How Students and Algorithms Learn to Filter:

Investigating Students' Understanding of Signal Processing Concepts and

Bilevel Methods for Learning Filters for Image Reconstruction

by

Caroline E. Crockett

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Electrical Engineering and Computer Science)
in the University of Michigan
2022

Doctoral Committee:

Professor Jeffrey A. Fessler, Co-Chair
Professor Cynthia J. Finelli, Co-Chair
Professor Mark Guzdial
Professor Harry C. Powell, University of Virginia
Professor Qing Qu

Caroline E. Crockett

cecroc@umich.edu

ORCID iD: 0000-0003-0604-9000

ACKNOWLEDGEMENTS

There are so many people who have supported and guided me over the years that this acknowledgements section is necessarily incomplete. I cannot sufficiently thank those people named here and I include by name of a few of the many people who should be included.

First, I would like to thank my advisers, Cindy Finelli and Jeff Fessler. Cindy reached out to me when I applied to the University of Michigan and asked if I was interested in being co-advised by her and doing engineering education research. At that time, I did not understand what engineering education research was, but she made it sound interesting and I said yes! Education research has been a fantastic fit with my personal interests and I am so grateful to Cindy for introducing me to this growing field and then mentoring and helping me to dive into it. I joined Jeff's research group at the end of my first year of graduate studies and, in addition to our research together, I was fortunate to take three of his graduate courses and to be a teaching assistant for him twice. There was a great harmony between our research and the coursework–our research into filter learning for piece-wise constant signals was even inspired by an example problem in one of those courses. Cindy and Jeff, I could not have asked for a better set of advisers. You have both taught me so much about research and how to stay curious and ask interesting questions. You have helped me become a better researcher, writer, and teacher. I truly admire the lab culture of caring, patience, and openness to new ideas that both of you have cultivated and I hope I can imitate your mentoring style for my future students.

I am also thankful to the other member of my committee for their support of this unusual dissertation. Harry Powell was kind enough to share years worth of concept inventory data with me and was invaluable for interpreting the results of our analysis, especially in thinking about implications for practice. Mark Guzdial introduced me to many new frameworks in engineering education research and his guidance helped to direct and strengthen part 1 of this dissertation. Mark was especially helpful in shaping the methodology for Chapter 4. Finally, Qing Qu introduced me to new ideas in optimization that are incorporated throughout part 2 of this dissertation and helped me to connect the review of bilevel methods to other areas of the machine learning literature.

This research in this dissertation was made possible in part due to the support of NIH grant R01 EB023618, NSF grant IIS 1838179, and the Rackham Graduate School at the University of Michigan. This research would also not exist if not for the instructors who helped me to recruit student participants, especially Stephane Lafortune, Achilleas Anastasopoulos, Robin Fowler, and

Christian Casper. Kristen Thornton and Kim Novak were also instrumental to recruiting students, helping me with administrative tasks, and generally making my life so much easier–thank you! I also thank Kathleen Wage and John Buck, for allowing me to use the signals and systems concept inventory and for all of their insight and recommendations. Finally, I thank all the participants who took the time to respond to surveys and participate in interviews or focus groups. Although I cannot name you, know that you made this research possible and I am forever grateful for your contributions.

Many other researchers in the community provided me with feedback and support. My co-authors from the systematic literature review on how students respond to active learning introduced me to the field of engineering education research and mentored me on research methodology. They are Kevin Nguyen, Maura Borrego, Matt DeMonbrun, Prat Shekhar, Sneha Tharayil, Robyn Rosenberg, and Cindy Waters. Shivani Sakri helped conduct and code some of the focus groups in Chapter 5. Il Yong Chun and David Hong helped me to start research on analysis operator learning and taught me a lot about optimization methods. Among those scholars who provided insight on bilevel methods are Lindon Roberts, Mike McCann, Avrajit Ghosh, and Saiprasad Ravishankar. I also would like to thank the anonymous reviewers for all of my submitted papers whose suggestions strengthened and clarified my writing.

I was fortunate to be part of two amazing lab groups. In particular, Rachel Vitali, Aaron Johnson, Matt DeMonbrun, and Jessica Swenson were wonderful mentors and friends who influenced my early research directions in education; David Hong was a mentor and role model for the start of my graduate journey in image reconstruction; and Melissa Haskell was a mentor and great friend for the second half of my graduate journey. I also thank my fellow labmates and post-docs for always making the office a fun place to work and collaborate. Finally, I thank the larger engineering education research and electrical engineering community at the University of Michigan for all the thought-provoking conversations and friendships. To the many others who I do not have space to name here, I hope to thank you in person someday.

Finally, and most importantly, I would like to thank my family. My amazing support system includes my in-laws, who have welcomed me into their family; Danny Medrano and the full Medrano family; my two older sisters, Laura and Theresa, who have always set a good example for me; my brothers-in-law, nephews, and niece; my extended family of grandparents, aunts, uncles, and cousins; and my dog, Oreo, who ensured I took multiple walks every day. Mom and dad, thank you for guiding me and supporting my entire life. You taught me to work hard, encouraged me to find and pursue work I enjoyed, and were there always there for me no matter what. Finally, to my husband Chris, thank you for everything–I could not have done this without you.

TABLE OF CONTENTS

CHAPTER

LIST OF TABLES

TABLE

LIST OF FIGURES

FIGURE

LIST OF APPENDICES

APPENDIX

LIST OF ABBREVIATIONS

BCM Block coordinate minimization

CAOL Convolutional Analysis Operator Learning

CDF Cumulative density function

CG Conjugate gradient

CI Concept Inventory

CNN Convolutional neural network

CT Computed Tomography

CU Conceptual Understanding

DCT Discrete Cosine Transform

DT Discrete Time

EE Electrical Engineering

EER Engineering Education Research

EFD Extended Finite Difference

FT Fourier transform

GPA Grade point average

IFT Implicit function theorem

IRB Institutional Review Board

IRT Item Response Theory

KiP Knowledge in Pieces

KKT Karush–Kuhn–Tucker

LT Laplace transform

LTI Linearity and time invariance

MoEP Model of Educational Productivity

MRI Magnetic Resonance Imaging

MSE Mean squared error

PDF Probability density function

PK Procedural Knowledge

PSNR Peak signal to noise ratio

PWC piece-wise constant

RMSE root mean square error

SES socioeconomic status

S&S Signals & Systems

SSCI Signals and Systems Concept Inventory

SSIM Structural Similarity

TV Total Variation

UM University of Michigan

ABSTRACT

Signals and systems (S&S) concepts are the theoretical foundation of machine learning and signal processing, cutting-edge fields with real-world applications in many domains. This dissertation combines two projects on S&S in the fields of engineering education research and image reconstruction.

Within the field of engineering education research, this dissertation discusses which S&S concepts students understand and what factors–such as motivation, choice of upper-level electives, or use of evidence-based instructional practices like active learning–influence their understanding. This research project involved three phases. The first phase used quantitative methods to measure CU and investigated factors that predict CU of students at the end of their S&S course. This phase found that measures of ability and motivation are significantly predictive of CU. Phase one also served as a pilot project for the following two phases that concentrate on CU of senior undergraduate students. The second phase used think-aloud interviews and a concept inventory to measure CU of S&S. The results show that many seniors understand some topics, such as filtering and time invariance, but struggle with other S&S concepts, such as linearity and convolution. The third phase used interviews and qualitative data analysis methods to investigate what factors impact CU over the course of an undergraduate degree. The results provide recommendations for how instructors and curriculum designers can improve students' CU of S&S, such as emphasizing the purpose of concepts, using contrasting examples in lectures, translating mathematics, and repeating concepts across multiple courses.

The second part of this dissertation applies concepts from S&S to image reconstruction. Image reconstruction is the process of taking input data from one signal space and producing an interpretable image. In medical image reconstruction, state-of-the-art methods use advances in machine learning and training datasets to learn parameters that can be used to reconstruct high-quality images with fewer measurements, thus decreasing radiation exposure for patients while providing doctors with high-quality images to properly diagnose and treat many diseases. The image reconstruction project in this dissertation motivates and reviews bilevel methods for learning image reconstruction parameters. Bilevel methods are task-based, so that learned parameters are expected to perform best at reconstructing; are explainable and interpretable, thus improving the likelihood that doctors will trust and adopt them; and allow for different measures of image quality, including traditional mean square error metrics that are easy to use and metrics that more accurately capture human perception. The results demonstrate that parameters learned in a common non-bilevel formulation under-perform handcrafted parameters due to the structure of the learning problem and that bilevel methods help to address this gap.

CHAPTER 1

Introduction and Overview

1.1 Opportunities for Advancing Signals and Systems

S&S is the focus of an electrical engineering (EE) science course at most universities. Topics in S&S, such as filtering and Fourier transforms, are fundamental to rapidly growing fields such as control theory, signal processing, communications, and machine learning.

This dissertation focuses on two aspects of S&S. Part 1.5 discusses how students learn concepts in S&S, with the ultimate goal of understanding how we can teach S&S better. This part is anchored in the field of Engineering Education Research (EER); to answer the research questions, I used theoretical frameworks of how students learn, collected and analyzed qualitative interview data and quantitative survey data, and related the findings to other research in the EER literature. Part 6.3 discusses using S&S concepts to improve on state-of-the-art image reconstruction methods. This part is anchored in the more common EE research tradition; to answer the research questions, I implemented parameter learning methods with existing data sets, compared the methods using primarily quantitative metrics, and connected the conclusions to other findings in the EE literature.

The remainder of this introductory chapter overviews both parts of the dissertation and how they relate to each other. After the introduction, readers should be able to read Part 1.5 or Part 6.3 stand-alone or in either order.

1.2 Part I: How Students Understand Signals and Systems

Despite the importance educators place on S&S concepts, previous studies have shown that students generally understand few of the concepts at the end of a S&S course [1]. Students generally take the introductory S&S course during their second or third year of an undergraduate degree. Part 1.5 of this dissertation aims to determine if undergraduate students later come to understand

S&S concepts, and, if they do, at what point they reach that understanding and what factors help them reach it. While Part 1.5 considers S&S concepts, I anticipate the study will provide insights into CU over time in other engineering disciplines.

Few studies report on students' CU of S&S topics one or more semesters after they completed a S&S course, even though developing CU can take longer than a single semester [2]. Part 1.5 of this dissertation helps to fill this gap by examining CU of senior students, who are typically one or two years removed from their S&S course. Further, this study considers what factors, ranging from student characteristics (*e.g.*, motivation or ability) to instructional characteristics (*e.g.*, quantity of instruction or use of active learning), might help students gain CU. My research questions are:

RQ#1 What is students' CU of S&S concepts at the end of an undergraduate S&S course? What factors predict how many S&S concepts students learn in a S&S course?

RQ#2 What is the CU of S&S concepts among senior students?

RQ#3 What instructional factors influence CU of S&S for senior students?

Chapter 1.5 provides background for the EER part of this dissertation. Specifically, it defines and reviews previous works on CU and overviews the concepts in the standard S&S curriculum. Chapter 3 addresses RQ#1 using a quantitative methodology, and serves as a pilot study to lay the groundwork for research to address RQ#2 and RQ#3. Chapter 4 mixes additional quantitative survey data with qualitative interview data to address RQ#2. Chapter 5 focuses on the last research question, RQ#3, using a qualitative methodology. Finally, Chapter 6 concludes Part 1.5 with a discussion of the findings and a summary of implications for practice.

The following subsections preview the methods and results of the three studies in Part 1.5, depicted in Fig. 1.1. This overview of the methodology pulls heavily from an overview of the methodology published in Crockett, Finelli, and Powell [3]. However, there are many differences in what is presented in the finalized methodology presented here and the plan laid out in [3]; many of the differences stem from the impact of COVID-19 on our data collection.

In each study, students were incentivized by a mixture of course credit, free food, gift cards, raffle prizes, and the gratitude of the research team. All participant interaction was approved by the University of Virginia (UVA) or University of Michigan (UM) institutional review board (approval numbers IRB-SBS #3566 and HUM00167323 respectively).

1.2.1 RQ#1: Conceptual Understanding during Signals and Systems

The first study served as a pilot for the second and third research questions; it considers both measuring CU and investigating factors that may affect CU, but in a simpler, better-studied context. Specifically, unlike the follow-on studies, the pilot study considers CU only within a S&S

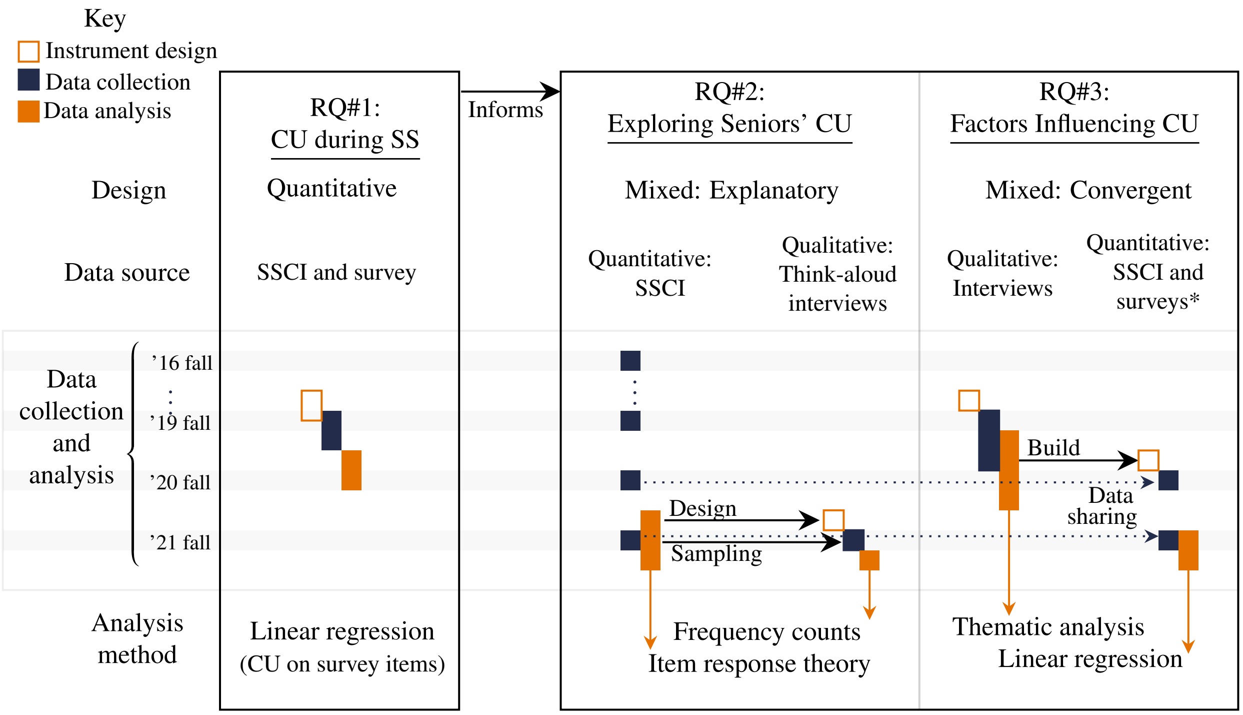


Figure 1.1: Overview of the three studies in Part 1.5 of this dissertation, covered in order in Chapters 3, 4, and 5. The results from the first study informed the design of the two later studies. Likewise, for the second and third study, the initial data analysis influenced the second data sources (think-aloud interviews and surveys, respectively), as indicated by the arrows. In both the first and third study, the linear regression analysis modeled the relationship between CU (the dependent variable) and hypothesized factors (independent variables) that were measured using surveys. *=Due to low participation, this dissertation does not discuss results from the planned linear regression for the third study.

course, for which there are many existing studies in the literature to compare results against, see Section 2.2.3. These previous studies generally show that many engineering undergraduates lack CU of S&S.

Study 1 involved undergraduates at UM in the introductory S&S course, EECS 216, which is aimed at second and third year students. The course emphasizes continuous time analysis and has an associated lab section that meets roughly five times a semester. Students in this course in Fall 2019 and Winter 2020 took the SSCI for extra credit near the beginning of the semester (the pre-test) and at the end of the semester about a week before their final exam (the post-test). The SSCI is an existing test that has 25 multiple-choice questions on background mathematics, system properties, convolution, Fourier and Laplace transforms, and filtering concepts; Section 2.2.3 describes the SSCI. This research uses version 5 of the continuous time SSCI. When students took the post-test, they were also given a short survey to measure factors that may predict their conceptual understanding. Specifically, the pilot study tests how well a subset of factors from the Model of Educational Productivity (student ability and motivation, instructional quality and quantity, and home, peer, and classroom environment) explain the variance in signals and systems conceptual understanding at the end of an introductory undergraduate course.

Chapter 3 overviews the Model of Educational Productivity, presents statistics from the post-test SSCI data ($n = 158$) to measure students' CU, and then discusses the results from a linear regression model on the surveys and concept inventories data ($n = 124$) to investigate factors that correlate with CU. The results show the hypothesized factors explained 28% of variance in post-test conceptual understanding. Further, two of the factors were significantly predictive of CU: ability ($p < 0.01$) and motivation ($p < 0.10$). The results in Chapter 3 expand on the results presented in Crockett and Finelli [4].

The lessons learned in the pilot study informed the approach to the following two studies. For example, analyzing the SSCI data to measure students' CU required me to become very familiar with the SSCI questions, how each measured a different concept, and connections between the questions. This prepared me to identify questions to use on the interviews as part of answering RQ#2. Also, based on the linear regression results in the pilot study, I clarified survey items and added questions about certain factors in interviews to answer RQ#3.

1.2.2 RQ#2: Conceptual Understanding of Senior Undergraduates

The second study moves from measuring CU of students in a S&S course to measuring CU of senior students. We use the term "senior" throughout to refer to students who are expected to complete their undergraduate degree within a year, as determined by them reaching a set number of credits determined by the university. While previous studies across multiple subjects show engineering students have low CU at the end of courses, little is known about CU semesters after a course. Does CU increase as seniors have time to digest concepts and perhaps see concepts repeated in upper-level courses? Or does CU decrease as seniors forget what they learned in S&S?

Study 2 is a mixed methods study and uses quantitative SSCI data ($n = 467$) and think-aloud interviews ($n = 12$) to measure CU. The data come from senior students at two universities: UM and UVA. To analyze the data, we use an item response theory analysis of the SSCI data; this analysis orders the SSCI questions from most to least difficult, while accounting for student ability. We then discuss what the difficulty of each questions suggests about students' CU of specific S&S concepts (linearity and time invariance, convolution, Fourier transform, and filtering). The think-aloud interviews investigate how students approach conceptual problems and test hypothesis from the SSCI data about students' CU.

Chapter 4 presents the results of this study. The discussion in Chapter 4 expands on Crockett, Powell, and Finelli [5]. Briefly, we found that seniors' scores on the concept inventory are typical of scores presented at the end of a S&S course. Many struggled with the concept of linearity, made a common error when finding the maximum value in graphical convolution, and had low confidence on relating frequencies in time to a FT representation, but seniors had relatively high CU of time invariance and filtering. We also observed a large variation in SSCI scores among the senior students. This naturally leads to RQ#3 on what factors may impact CU and cause differences in CU between students.

1.2.3 RQ#3: Factors Influencing Conceptual Understanding

The third study, used an exploratory qualitative approach to build on a literature review about what instructional factors influence CU of S&S for senior undergraduate engineering students. Previous results show students in S&S courses typically gain little CU, though evidence-based instructional practices, such as active learning, can increase gains in CU. However, few studies consider factors on CU of senior students or other instructional practices that increase CU.

To explore possible factors, I interviewed two faculty members, eight undergraduate seniors, five graduate students, and four practicing engineers. In Fall 2019 at UM, I conducted two focus groups (one each with undergraduate and with graduate students), conducted an instructor interview, and informally spoke with instructors. In summer 2020, I conducted an undergraduate focus group and an instructor interview at UVA. I additionally interviewed four engineers working in industry with a range of industry experience in Summer 2020.

Chapter 5 presents the results of analyzing the transcribed interviews using a constant comparative method along with many participant quotes as evidence for each theme, expanding on the discussion in Crockett, Powell, and Finelli [6]. Briefly, participants identified lectures presenting CU along-side mathematical expressions; lectures emphasizing purpose and connections; hands-on activities where students have control, receive immediate feedback, or where they have to apply and synthesize concepts; and repetition of concepts across multiple courses as factors that helped build CU. Grades that emphasize procedural knowledge over CU and heavy workloads were noted as hindrances to CU.

Unfortunately, due to low participation, Chapter 5 does not present the planned linear regression results shown in Fig. 1.1. Similar to the linear regression in the pilot study, these results would have shown how factors measured with a survey (survey 2 in Fig. 6.1) correlate with SSCI scores. However, Section 5.4.3 presents a preliminary analysis and suggestions for future work in this direction.

1.3 Part II: Bilevel Methods for Image Reconstruction

Part 6.3 of this dissertation applies S&S topics in image reconstruction methods. Image reconstruction is the process of taking input data from one signal space (*e.g.*, the data collected by an Magnetic Resonance Imaging (MRI) or Computed Tomography (CT) machine) and producing an image that humans (*e.g.*, doctors) or an image analysis software can interpret. Part 6.3 concentrates on image denoising and medical image reconstruction, though the methods easily apply to other image reconstruction problems.

There are many existing image reconstruction methods roughly corresponding to different assumptions about the reconstructed images. Historically, the assumptions are based on characteristics of the desired output image that humans can easily observe, such as a tendency to have smooth regions with few edges or to have some form of sparsity [7]. Recently, more researchers are using machine learning techniques to discover image characteristics to use in the image reconstruction process. Although machine learning, and particularly deep learning, can give state-of-the-art results, medical professionals are often wary of it because it can be hard to explain compared to handcrafted features based on observable image characteristics. Further, deep learning techniques often have few, if any, theoretical guarantees for image reconstruction applications.

Bilevel methods are one way to bridge the gap [8]. Part 6.3 examines a bilevel method that uses training data to learn sparsifying convolutional filters that yield good reconstructed training images, while allowing for different measures of what constitutes a good output image. Part 6.3 reviews and motivates bilevel methods for image reconstruction. My research questions are:

RQ#4 Why do handcrafted sparsifying filters sometimes outperform learned filters?

RQ#5 How does the bilevel method compare to handcrafted filters and filters learned in a non-task-based method?

RQ#6 What are the current trends in the literature on bilevel methods for image reconstruction?

Chapter 6.3 motivates the image reconstruction problem, defines notation, and introduces the bilevel problem considered throughout Part 6.3. Chapter 8 provides background on image reconstruction, loss function design, and hyperparameter optimization strategies. Chapter 9 addresses RQ#4 using a series of case studies. Chapter 10 begins the literature review for RQ#6 by describing optimization methods for the bilevel problem. Chapter 11 address RQ#5 by revisiting one of the case studies from Chapter 9 with the proposed bilevel method. Chapter 12 continues the literature review of bilevel methods by discussing previous applications of the bilevel method in image recovery problems and connecting bilevel methods to other machine learning approaches. Finally,

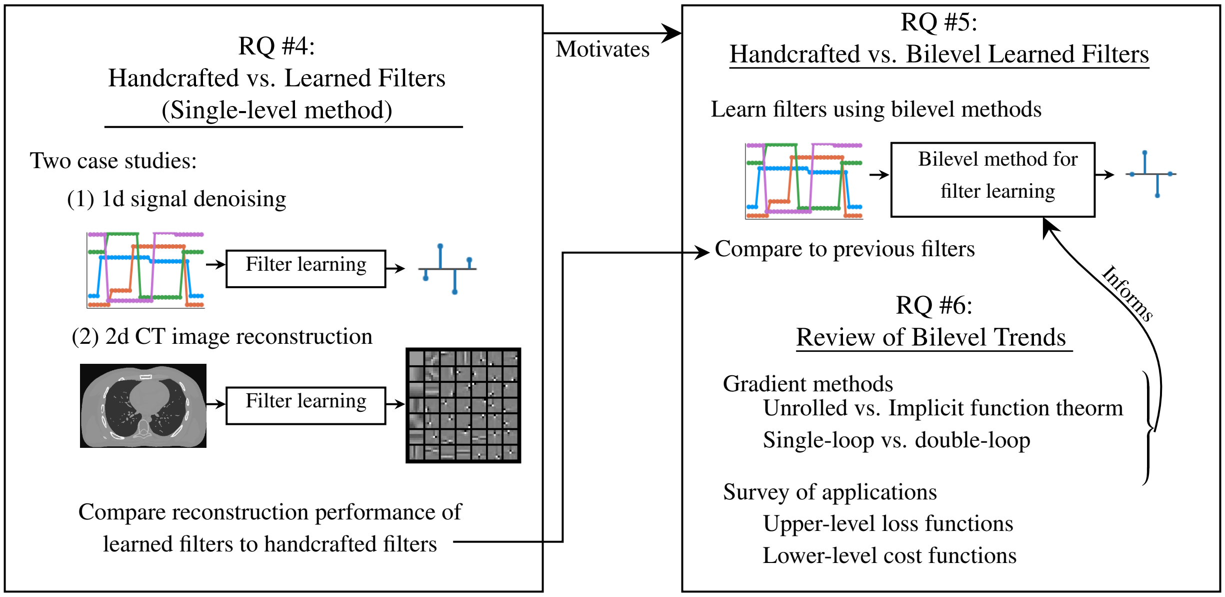

RQ #4:
Handcrafted vs. Learned Filters
(Single-level method)
Two case studies:
(1) 1d signal denoising
Filter learning
(2) 2d CT image reconstruction
Filter learning
Compare reconstruction performance of
learned filters to handcrafted filters
Motivates
RQ #5:
Handcrafted vs. Bilevel Learned Filters
Learn filters using bilevel methods
Bilevel method for
filter learning
Compare to previous filters
Informs
RQ #6:
Review of Bilevel Trends
Gradient methods
Unrolled vs. Implicit function theorm
Single-loop vs. double-loop
Survey of applications
Upper-level loss functions
Lower-level cost functions

Figure 1.2: Overview of the three research questions in Part 6.3 of this dissertation. Chapter 9 discusses RQ#4 and motivates the next two research questions. Chapter 10 reviews optimization methods for the bilevel problem as part of addressing RQ#6; these methods are used in Chapter 11 to answer RQ#5. Finally, Chapter 12 returns to RQ#6 and overviews previous applications of bilevel methods in the literature.

Chapter 13 offers summarizing commentary on the benefits and drawbacks of bilevel methods for computational imaging and proposes future directions for the field.

The following subsections preview the methods and results for each of the research questions, depicted in Fig. 1.2.

1.3.1 RQ#4: Handcrafted Versus Learned Filters

RQ#4 stemmed from an observation that, when we learned filters using a specific training methodology, the learned filters sometimes yielded noisier (worse) signals than handcrafted filters when later used in a denoising method. Learning filters from training data requires compute time and power; thus, it does not make sense to learn filters unless the learned filters are better in some way than handcrafted filters. The machine learning literature is full of examples of learned hyperparameters achieving lower test errors than handcrafted parameters. Therefore, RQ#4 does not ask whether learning is useful, but rather it asks why our specific training methodology yielded filters that did not perform as well as handcrafted filters.

Chapter 9 includes two experiments that investigate why handcrafted filters sometimes outperform learned filters. The first experiment considers learning sparsifying filters for simple, piece-wise constant 1D signals. By using piece-wise constant signals, we were able to handcraft a sparsifying filter based on finite differences that achieved reasonable denoising performance. The results show that a common method for learning the sparsifying filters tends to learn overly smoothed filters. The exact denoising error of the learned filters varies with a tuning parameter, but Chapter 9 includes an example where the learned filter resulted in denoised signals with 38% more error than the signals denoised using the handcrafted filter (see Tab. 9.2). In other words, although these smoother filters minimize the training objective, they do not denoise test signals as well as the handcrafted, "sharp" finite differencing filter.

The second experiment in Chapter 9 presents the work from Crockett, Hong, Chun, *et al.* [9] on learning sparsifying filters for CT image reconstruction. The proposed algorithm, called convolutional analysis operator learning with handcrafted filters, allowed us to compare learning a set number of filters to handcrafting a subset of the filters while learning the remaining filters. As in the simple 1D experiment, CT images are approximately piece-wise constant and finite differencing filters can yield reconstructed images with relatively small errors, so we examined the impact of handcrafting a varying number of finite differencing filters. Similar to the results from the simple 1D experiment, some of the filter banks with handcrafted filters were able to reconstruct the test images comparably to the learned filters while decreasing the training time.

In both experiments, Chapter 9 considers learning filters based on sparsifying training signals. The chapter discusses that this training objective does not account for the application of the learned filters to the denoising or image reconstruction task where they are tested. In answer to RQ#4, we conclude that both the specific structure of the training objective and the mismatch between the training objective (sparsifying signals) and the testing criteria (denoising or reconstructing signals) are what cause the handcrafted filters to perform better than or comparable to the learned filters in terms of the test criteria. The results from Chapter 9 thus motivate the use of bilevel, task-based learning methods and drive RQ#5 and RQ#6.

1.3.2 RQ#5: Learning Filters Using Bilevel Methods

The second research question in Part 6.3 is a direct follow-on to the first research question: it asks whether a bilevel, task-based method for learning sparsifying filters results in learned filters that match or improve on the denoising performance of handcrafted filters. While previous studies show the benefit of machine learning, we are unaware of a previous study that includes such a direct comparison of these two learning methods.

One can view the bilevel problem as formalizing hyperparameter optimization, as bridging machine learning and cost function based optimization methods, or as a method to learn variables best suited to a specific task. More formally, bilevel problems attempt to minimize an upper-level loss function, where variables in the upper-level loss function are themselves minimizers of a lower-level cost function.

Chapter 11 addresses RQ#5 and expands on the work of Crockett and Fessler [10]. The chapter considers the same 1D signals as in Chapter 9, but now with a bilevel method for filter learning. The results show that, averaged over multiple random initializations, filters learned using a bilevel method result in the two denoising test signals having less error than when the signals are denoised using the filters learned in Chapter 9 based on sparsifying training signals. Thus, as predicted, the results show the benefit of task-based learning. However, the filters learned using the bilevel method still denoised signals worse than a handcrafted finite differencing filter by 7-24%, with an average increase of error of 16% (see Tab. 11.1).

Although the bilevel-learned filters for the simple 1D experiment still did not achieve the denoising performance of the handcrafted filters, the results in Chapter 11 show the potential of a bilevel method, especially when compared to non-task-based learning methods. Compared to the training methods used to investigate RQ#4, bilevel methods require more design decisions. For example, Chapter 11 compares using different numbers of iterations to estimate the denoised signal and investigates the impact of initializing the learnable filters. For many other design decisions, Chapter 11 considers a relatively simple bilevel method design. More advanced bilevel methods, especially ones that use a more sophisticated image reconstruction cost function, should further improve the denoising performance.

1.3.3 RQ#6: A Literature Review of Bilevel Methods

The last research question in this dissertation involved a literature review of bilevel methods, as presented in Crockett and Fessler [11]. Part 6.3 references the bilevel literature review [11] throughout, but the two chapters that primarily address RQ#6 are Chapter 10 and 12. The goal of these two chapters is to make bilevel methods more easily accessible to different audiences.

Chapter 10 discusses different methods for optimizing the bilevel problem and the advantages and disadvantages of the variety of the methods. The chapter focuses on gradient-based optimization methods. Classic hyperparameter optimization strategies such as grid search or Bayesian methods consider the testing objective task, *e.g.*, image reconstruction, as a black-box. In contrast, gradient-based methods use knowledge of the structure of the task to compute a gradient of the upper-level loss with respect to the hyperparameters of interest. By doing so, gradient-based methods are able to scale to large numbers of hyperparameters. Chapter 10 is split into two primary sections. The first discusses how to find this gradient and the second discusses how to use the gradient to optimize the bilevel problem.

Chapter 12 overviews previous applications of bilevel methods and connects bilevel methods to other machine learning methods. The first section discusses lower-level cost functions that represent different image reconstruction tasks and a variety of upper-level loss functions for judging the quality of the hyperparameters. The second section compares and contrasts bilevel methods with popular machine learning methods: unrolled networks, fixed-point networks, and plug-and-play priors.

1.4 Positionality Statement

Science is often touted as objective–the views of the scientist do not impact the results of a study. If true, one could use science to uncover truth and it would not matter which researcher does the work. In contrast to this viewpoint, more people are recognizing that researchers impact their research. Henrich, Heine, and Norenzayan [12] discuss how a person is, in many ways, inseparable from their culture–culture can even impact our perception of optical illusions. Given how culture can greatly change these seemingly objective experiences, it is difficult to imagine that research is not also impacted by the researcher's culture and by their personal experiences and beliefs.

In EER, and especially in qualitative methodologies, this trend toward seeing research as more subjective is evidenced by the prevalence of positionality statements. The purpose of a positionality statement is to explain how a researcher's background and/or identity influenced the study design and data analysis. Such statements are becoming more common-place, *e.g.*, the Journal of Women and Minorities in Science and Engineering requires positionality statements in journal submissions [13]. The motivation behind including positionality statements is that a researcher's beliefs influence the research and that, while one may strive to be neutral, one cannot remove one's influence entirely. Therefore, it is a methodological best practice to discuss researcher beliefs that are relevant to the current research project and how they might influence the research, including research design, analysis, and interpretation.

The first obvious impact of my experiences on the research is in the selection of the research questions. My research in Part 6.3 is an application of S&S concepts and as a graduate student I have come to value CU of these concepts. Thus, my research questions for Part 1.5 presume that CU is important rather than ask *if* CU is important. As an undergraduate, I came to appreciate S&S concepts more deeply after my undergraduate S&S course. This experience made me more interested in the evolution of CU and of measuring CU years after a S&S course. In turn, my interest in education and becoming an instructor influenced my choice of research questions in Part 6.3, particularly RQ#6. My interest in reviewing the literature on bilevel methods stems from my interest in making technical content accessible.

A second way that my identity impacts the research is in what data I collect. For Part 6.3, this is most obvious in my choice of training and testing data sets. Because I was embedded in a research lab for medical image reconstruction, I tended to use medical imaging examples. Because I was interested in understanding and interpreting the image reconstruction system (rather than achieving state-of-the-art results on a specific application), my other data sets tended to consist of very simple signals where we can predict how an ideal system might respond. For Part 1.5, how I could relate to participants in qualitative interviews is a clear way that my identity impacts the data. My identity as a graduate student who previously worked in industry helped me transition between student interviews, faculty interviews, and interviews with practicing engineers by allowing me to develop rapport with each group by emphasizing our shared experiences. As someone who majored in and studies EE, most of participants likely considered me an "insider" in terms of the engineering community. Some benefits of being perceived as an insider were that it was easier to recruit participants, I could base my questions on my own experience to make the questions more detailed, I might have been more trusted by participants, I could speak the same language as participants, and I was not shocked by responses since I was familiar with many aspects of the culture and participants' experiences [14]. Corresponding disadvantages of being perceived as an insider were that I might have been unknowingly biased and I did not bring a fresh perspective on the subject, participants might not have said things they think should be obvious to me, I could not ask overly simple questions legitimately, and participants might have been more likely to cater their responses to what they think I want to hear [14]. As a specific example, I could not ask what a Fourier transform (FT) is and expect participants to explain it the same way and using the same language that they would with an interviewer from a different discipline.

Another inevitable influence I have as a researcher is during during data analysis and interpretation, especially for the qualitative data analysis in Part 1.5. Unlike the above influences, which shape the study but are not a direct concern for the study's quality, I took specific steps to minimize my personal influence on data analysis so as to improve the study's quality. Analyzing qualitative data requires the researcher to interpret participants' statements. I have my own perspective on what helped and hindered my CU and about what concepts are hard and why they are hard. Participants' ideas that align with mine were likely to be more obvious in the data. For example, my previous research on active learning [15]–[18] means I was more attuned to seeing that as a theme in the data. To minimize the impact of my experiences and to make sure my data analysis reflected the data, I followed best practices such as discussing the coding and interpretation of quotes with other researchers and memoing [19].

Finally, my experiences, particularly those of studying engineering, are generally aligned with post-positivism. Briefly, post-positivism is associated with the scientific method, rationality, and quantitative methods [20]. One of its main tenets is that there is a single "truth" and that researchers should attempt to mitigate and eliminate their biases such that their research is objective and value-free. Willis [21] discusses how positivism's history is intertwined with science. In contrast, the interpretive paradigm (which is frequently associated with qualitative research) focuses on social meaning and perspective. Interpretive approaches acknowledge that there can be multiple "truths." The positivist and interpretive paradigms are only two examples of research paradigms; [22] describes many of the more recent post-modernism and critical frameworks. Over the course of my degree, I have come to value qualitative data and have adopted a more interpretive framework to research. This is evident in the design trend of the three studies in Part 1.5: this first study is quantitative, the second study mixes quantitative and qualitative data, and the third study is qualitative. This dissertation may appeal to a diverse readership, many of whom are unfamiliar with qualitative research and the interpretive paradigm (as I was when entering my graduate studies). I initially found it useful to view the interpretive paradigm as a tool: even if you do not necessarily agree with its views, adopting it can be helpful. Through application, you may then find yourself agreeing that there can be more than one "truth."

1.5 Bridging the Two Parts

This dissertation considers two aspects of SS. First, Part 1.5 investigates how students learn concepts in SS, with the goal of improving the curriculum to prepare graduates to make significant contributions in the increasing number of SS-related jobs. Part 6.3 uses filtering (a major SS concept) as a basis for an image reconstruction model to advance the field of medical imaging, thus demonstrating one of the many areas where students who understand SS concepts can chose to contribute to research and application. The following list notes which chapters present the results of the primary papers included in this dissertation:

- Chapter 3:
  [4] C. Crockett and C. Finelli, "Factors influencing conceptual understanding in a signals and systems course," in *2021 ASEE Virtual Annual Conference Content Access*, Jul. 2021. [Online]. Available: https://peer.asee.org/37175

- Chapter 4:
  [5] C. Crockett, H. C. Powell, and C. J. Finelli, "Conceptual understanding of signals and systems in senior undergraduate students," Submitted to: *IEEE Transactions on Education*, 2022

- Chapter 5:
  [6] C. Crockett, H. C. Powell, and C. J. Finelli, "Factors influencing conceptual understanding of signals and systems of senior engineering students," Submitted to: *European Journal of Engineering Education*, 2022

- Chapter 9:
  [9] C. Crockett, D. Hong, I. Y. Chun, and J. A. Fessler, "Incorporating handcrafted filters in convolutional analysis operator learning for ill-posed inverse problems," in *2019 IEEE 8th*

International Workshop on Computational Advances in Multi-Sensor Adaptive Processing, CAMSAP, Dec. 2019, pp. 316–320. DOI: 10.1109/CAMSAP45676.2019.9022669

- Chapter 9 and 11:
  [10] C. Crockett and J. A. Fessler, "Motivating bilevel approaches to filter learning: A case study," in *2021 IEEE International Conference on Image Processing (ICIP)*, IEEE, Sep. 19, 2021, pp. 2803–2807, ISBN: 978-1-66544-115-5.
  DOI: 10.1109/ICIP42928.2021.9506489

- Chapter 10 and 12:
  [11] C. Crockett and J. A. Fessler, "Bilevel methods for image reconstruction," *Foundations and Trends® in Signal Processing*, vol. 15, no. 2-3, pp. 121–289, May 5, 2022, ISSN: 1932-8346, 1932-8354. DOI: 10.1561/2000000111

The two parts of the dissertation inform each other. My experience researching cutting-edge SS concepts for the image reconstruction part helped me to relate to students learning SS for the first time and to reflect on what experiences have helped me understand the concepts. For instance, learning about primal-dual formulations (see Appendix C) reminded me of learning about the FT (see Section 2.2.1). Both are tools that allow an engineer to transform a problem into another representation or domain in a way that makes the problem easier to solve. Once solved in the transformed representation, both tools provide a way to relate the solution back to the original variables. This idea of transforming a problem to another representation where the problem is easier to solve is a threshold concept in EE [2]; learning this can transform the way a student thinks. Although I previously learned the FT, seeing primal-dual formulations as an instance of this threshold concept took time and practice. Time and practice are common themes throughout Part 1.5 for how students develop CU.

My experience researching how students reach understanding for the engineering education part has in turn influenced my approach to my image reconstruction research. For example, while participating in a educational workshop on how to bring evidence in the literature to practice in the classroom ("evidence-to-practice"), I was introduced to the idea of backward design. The task-based nature of bilevel methods (that learned parameters are those that best perform some task) aligns with this backward design theory. In backward design, an instructor first identifies the learning objectives then designs an assessment, *e.g.*, a test, to measure how well students met the objectives [23, Ch. 1]. In bilevel methods, an engineer similarly must identify the goal, *e.g.*, to reconstruct a specific class of images, and design an assessment function (called the upper-level loss function, see Chapter 6.3) that measures how well the parameters perform at the given task. The next step in both bilevel methods and backward design is figuring out how to achieve the goal. In a classroom, this involves designing materials for class and teaching. In a bilevel method, this involves designing the image reconstruction system (called the lower-level cost function) and optimizing the resulting problem. Seeing the connection between bilevel methods and backward design is one reason I was motivated to research bilevel methods rather than other machine learning methods.

The main connection between Part 1.5 and Part 6.3 is in the recognition of the importance of S&S concepts. Hopefully the results of the first part on how to improve the S&S curriculum help to develop talented engineers who can further the work on applications like considered in the second part and the results of the second part act as motivation to students that these concepts are useful to solve real-world problems.

Part I: Conceptual Understanding of Signals and Systems

CHAPTER 2

Background

This chapter provides a literature review on ideas used throughout Part 1.5. It introduces and defines many terms used throughout the dissertation.

2.1 Conceptual Understanding

This section reviews previous work on CU, with a focus on how it may apply in undergraduate engineering classrooms. First, we define and discuss the importance of studying CU. Next, we discuss theoretical perspectives on CU and how it is reached. The theoretical perspectives inform the discussion of empirical results in undergraduate classrooms on how to measure CU.

2.1.1 Defining Conceptual Understanding

In [24], diSessa provides a historical overview of conceptual change research and acknowledges that there is still disagreement on what "conceptual understanding" means. To complicate the matter, many researchers use the phrase without offering an explicit definition. However, there are a few common ways of characterizing CU, such as how it is often defined in contrast to Procedural Knowledge (PK), see, *e.g.*, Hiebert and Lefevre [25] and Streveler, Brown, Herman, *et al.* [26].

Roughly speaking, PK is *how* to do something and includes knowing the precise formulas, steps, and techniques to solve a given problem. CU is knowing *relations* between pieces of information in a way that allows the information to be transferred to new contexts [25]. For example, Rittle-Johnson [27] defined CU as "understanding principles governing a domain and the *interrelations* between units of knowledge in a domain" [emphasis added] and Rao, Fan, Brame, *et al.* [28] defined CU as relating "mathematical representations and tangible physical interpretations." Other definitions of CU emphasize understanding contextual information; Montfort, Brown, and Pollock [29] defined CU as "an understanding of the phenomena underlying a calculation, including the context, purpose, necessary assumptions, and range of reasonable values expected." The following section presents a proposed definition for CU specific to engineering informed by these definitions.

Researchers hypothesize there is a positive feedback loop between CU and PK. Theorized benefits of increasing students' CU include increasing PK by providing structure to help students recall and select the correct procedure, more easily transferring procedures to new contexts, and developing more expert-like knowledge structures [25], [30]. PK can, in turn, increase CU; PK gives students the tools to solve problems that, especially in the realm of signal processing, are often mathematically complex. Knowing these mathematical tools can free cognitive resources to concentrate on understanding [25]. Similarly, PK, and specifically symbolic knowledge, can improve CU by representing complex concepts in more easily digestible ways. Both PK and CU are important and it can even be hard to fully separate them. For example, selecting the correct procedure to apply to a problem may be CU or PK, depending on which mental processes a student uses to make the selection.

Educators and researchers have discussed PK and CU in the mathematics curriculum spanning back to the late 1800s [25]. More recent work (starting roughly in the 1980s) has focused on:

- the interplay between CU and PK,
- a theoretical perspective of how students acquire CU, and
- contexts other than the elementary school classroom.

This project focuses on the latter two themes.

2.1.2 Proposed Definition of Conceptual Understanding

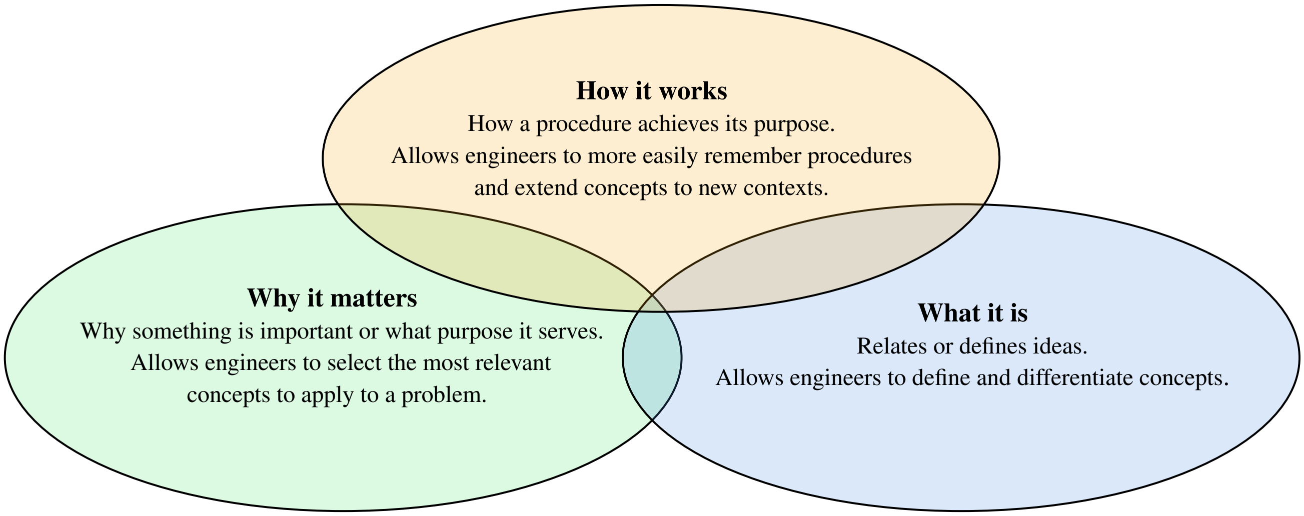
This section proposes a definition of CU, which is an expansion of the definition proposed in Crockett, Powell, and Finelli [5]. The definition is informed by a review of the literature (see Section 2.1.2) and my experiences working with the SSCI in the pilot study (see Chapter 3).

Recall from the previous section that common definitions of CU emphasize how topics relate to one another [27], [28], [31] and/or how they relate to contextual information [29]. Perhaps one reason for the lack of a clear definition of CU in *engineering* education literature is the historical dominance of CU research in the natural science field. Another reason may be that engineering is often thought of as applying science and mathematics concepts. Therefore, engineering specific "conceptual domains might not be seen necessary" [32]. Although engineers must understand underlying science concepts, engineering CU should also reflect the practical, applied nature of the field. Thus, we define engineering CU as *being able to reason about, relate, or apply concepts,* where we define three levels of concepts, summarized in Fig. 2.1: (1) why it matters concepts, (2) what it is concepts, and (3) how it works concepts.

As in most definitions, our proposed definition of engineering CU includes understanding the relationships between concepts. However, unlike the definitions of CU in Section 2.1.1 , this definition extends CU to explicitly include the application of and reasoning about concepts. Further, the proposed definition differentiates concepts into three levels. Tab. 2.1 illustrates some S&S concepts at each concept level.

(1) **Why it matters** concepts require knowing why something is important or what purpose it serves. These concepts thus allow engineers to select the most relevant concepts to apply to a possibly novel problem. An example concept from Tab. 2.1 is that the FT often yields a more convenient signal representation for analysis. This concept shows the importance of considering the full definition of CU because students' ability to state this concept does not mean they have CU of it. CU requires that one can reason about, relate, or apply this concept. An application of the concept would be selecting the FT to analyze an appropriate problem (rather than other possible tools) *because* the student recognizes the advantage of the Fourier representation. The "because" in the preceding sentence demonstrates the difficulty of measuring CU: it is challenging to know why a student decides to use the FT. Why it matters concepts most closely align with understanding the purpose of a calculation in the definition of CU from [29].

(2) **What it is** concepts relate or define ideas. These types of concepts therefore allow engineers to characterize and differentiate concepts. An example "what it is" concept is the definition of Linearity and time invariance (LTI). To repeat the point in the previous paragraph, simply stating the definition would fall under memorized knowledge or PK, while understanding the definition as a concept is characterized by the ability to reason about, relate, or apply it. For example, question 24 on the SSCI tests CU of LTI by presenting students with three graphical input/output pairs and asking if the system could be linear and/or time invariant. Standard homework problems in most S&S textbooks [33]–[35] ask students to determine if systems are LTI based on a mathematical


How it works
How a procedure achieves its purpose.
Allows engineers to more easily remember procedures and extend concepts to new contexts.

Why it matters
Why something is important or what purpose it serves.
Allows engineers to select the most relevant concepts to apply to a problem.

What it is
Relates or defines ideas.
Allows engineers to define and differentiate concepts.

Figure 2.1: Types of concepts in the proposed definition of conceptual understanding.

relation between the input and output, so the graphical presentation of question 24 is atypical for students. Thus, students have to reason about what LTI means and apply the definition in the graphical format.

Taken together, knowing "what it is" and "why it matters" helps engineers recognize when concepts may apply to a problem. We hypothesize that "why it matters" and "what it is" concepts are both relatively introductory-level concepts and that students can often learn them in either order with little previous exposure. However, as the previous examples show, even if the initial concept is easy for students to learn to repeat in words, it can take practice and seeing the concept in different applications for students to reach deep levels of CU for these concepts.

(3) **How it works** concepts encapsulate how a procedure achieves its purpose. These concepts allow engineers to more easily remember procedures and to extend concepts to new contexts. One example is that the FT performs a change of basis. As with the previous two types of concepts, students may be able to quickly learn to state the "how it works" concepts in Tab. 2.1, but reaching CU is more challenging. For this example concept, a student with CU could relate the FT to other linear transforms and would better understand coordinate systems and how to change coordinate systems more broadly. This example also shows CU and PK are not disjoint–some information may fall in both categories.

Table 2.1: Example engineering concepts in signals and systems at the three proposed concept levels. Examples of PK include the ability to write and compute a FT integral, perform a convolution, and check if a system is LTI.

"Why it matters"	"What it is" concepts	"How it works" concepts
• The FT often yields a more convenient signal representation for analysis. • Checking if a system is LTI determines if the output can be computed using a convolution. • Convolution is the correct operation to find the output of LTI systems. • Pole-zero plots are useful to determine system stability and causality. • Verifying a system's stability and causality is important for real-world systems.	• The FT relates time to frequency space. • A linear system is one which satisfies the homogeneity and scaling properties. • The Fourier transform relates convolution in one domain to multiplication in the dual domain. • Convolution is commutative. • Poles in the right half of the plane mean a system is unstable.	• Complex exponentials form a basis for signals. The FT is the corresponding change of basis. • Why the superposition principle applies to LTI systems and that convolution with the impulse response performs this superposition. • Understanding why the FT dualities hold, such as convolution-multiplication.

While students may learn "what it is" or "why it matters" concepts in any order, we hypothesize that "how it works" concepts are generally harder to achieve and often require the previous two levels of CU alongside a high level of PK. Many of these "how it works" concepts may not be the goal of introductory level (or even upper-level) undergraduate courses.

2.1.3 Models of Conceptual Understanding and Change

Historically, conceptual change research concentrated on identifying common conceptual errors rather than building a unifying theory of CU. The theoretical perspectives presented below were proposed since the 1980s and build on ideas about cognitivism. In turn, cognitivism was a response to its predecessor: behaviorism. For context, this section first briefly reviews behaviorism and cognitivism. The behaviorist and cognitivist frameworks are two ways of thinking about how students learn. They have different definitions for what constitutes knowledge and learning and therefore suggest different best practices for teaching styles, instructor and student roles, and assessment techniques [36].

The behaviorist framework was originally proposed in 1913 and remained dominant through the 1960s [36]. This framework considers the learner as a black-box, where the instructor can only observe the inputs (*e.g.*, lessons) and outputs (*e.g.*, exam responses). The instructor's goal is to find which inputs yield the desired outputs, then reinforce that behavior over multiple repetitions. When designing courses and curricula, behaviorists recommend that instructors:

- first design instructional objectives, then design the course to meet the objectives;
- break large tasks down into sub-tasks and tackle each piece on its own; and
- ensure that each learner reaches mastery of one piece before continuing.

Although behaviorism is commonly associated with lecturing, truly meeting the repetition and mastery elements requires self-pacing [36].

Cognitivism, introduced in the 1950s, is the current dominant educational theory [36], [37]. Rather than viewing the learner as a black-box, cognitivists try to understand the mechanisms behind how students learn and understand knowledge. In doing so, cognitivists describe student knowledge as belonging to a model, and they frequently recommend that instructors activate these models before teaching so students can focus on either validating or correcting them. Under the cognitivist framework, "*knowing* consists of having mental models that have been created and stored in the learner's long-term memory as a function of interacting with the environment [and] ...*learning* is the process of creating those models" [emphasis added] [36]. Cognitivism is frequently associated with demonstrations, inquiry learning, and other forms of active learning.

Taking the cognitivist viewpoint that understanding how students learn is important (rather than viewing them as black-boxes as in the behaviourist tradition), we now turn to discussing the theories for how conceptual knowledge is structured and how conceptual change occurs [26]. The following sections describe two prominent theories: framework theory and knowledge in pieces theory.

While framework theory and knowledge in pieces theory are two of the most commonly cited theories on conceptual change, there are additional theories. For example, schema theory and script theory both consider knowledge that is organized into structures based on context. Rumelhart [38] suggested that people organize knowledge about objects and relationships at all levels of specificity into schemata. For example, a schema on the Fourier transform might include general characteristics of the FT (the FT involves a complex integral, it uses $\mathcal{F}\{\cdot\}$ as a symbol, etc.), how the FT is commonly used (in filtering problems, to transform a time domain signal to a frequency domain signal), when one typically sees a FT (a signals and systems class), specific examples of the FT (perhaps from homework problems or in-class examples), and knowledge of the FT as an option in the "engineer's toolkit" (either very valuable or of dubious value). People make predictions in new situations based on what seems most likely using the information stored in the schema.

Script theory is very similar to schema theory, but with a heavier focus on the sequential nature of events. Both theories align with the definition of CU in [29]: "an understanding of the phenomena underlying a calculation, including the context, purpose, necessary assumptions, and range of reasonable values expected." Salzman and Strobel [32] provide a more thorough overview of additional theories and further discuss what CU is, how conceptual change occurs, how stable CU is, and how students react when they learn something that challenges their current CU.

2.1.3.1 Framework Theory

Framework theory [39] posits that students develop their own fairly coherent ideas of the world through everyday experiences (*e.g.*, blocks move when pushed). These ideas form a "naïve physics" framework. Students then add information they learn in school (*e.g.*, forces cause motion) into their naïve framework. Since some new information often contradicts naïve physics (*e.g.*, gravity is acting on the block even if it is not moving), the learning process can cause fragmentation and leave problematic reasonings. "Synthetic frameworks" are the intermediate frameworks that students create as they mold naïve physics to match the physics taught in classrooms.

Coherence is a central tenet of framework theory. Vosniadou and Skopeliti [39, p. 1430] claimed that "categorization is the most fundamental learning mechanism, a mechanism which most of the time promotes learning but which, in cases where conceptual change is required, can inhibit it." Imagine students' minds like a well-organized folder system, where every topic is nested within a parent topic. If students already have a folder created for "forces," adding a sub-folder for "gravity" may be easy, because it will inherit all the traits of other forces. However, if students misfiled gravity elsewhere, it could be hard for them to move all concepts related to gravity over to the correct folder. Framework theory emphasizes coherence, but the framework is still loose, with room for fragmentation. In fact, framework theory predicts that synthetic frameworks will be fragmented, partially because students do not have the mental capacity to realize the incoherence (recall that framework theory was largely developed by studying young elementary school children).

Applying framework theory, one can predict the slow process of conceptual change as students move from a coherent-but-naïve framework, through multiple synthetic frameworks, and eventually (hopefully) to an expert-like, coherent framework. Framework theory further explains that problematic reasonings often stem from topics that involve abstract concepts or concepts that deny naïve intuition (*e.g.*, gravity is acting on a block, even if the block is not moving). Instructors can therefore use framework theory to predict what concepts students will struggle with and how they might corrupt teachings to match naïve ideas. As another example, [39] describes how many children, when told the Earth is round, will picture the Earth as a pancake. Their everyday experience is of a flat world, but they have "learned" that the Earth is actually round. With this in mind, instructors can plan to elaborate on what they mean by "round."

Building on framework theory, Chi [40] suggests that "category mistakes" make it difficult to learn concepts that are incorrectly placed in one category of a framework because the concepts must be moved to the correct category. Chi concentrates on the example of how emergent processes are harder for students to understand than sequential processes. The classic diffusion example [41] provides an illustration: Imagine that you drop a small amount of dye into a glass of water. Over time, the molecules will move, interact randomly, and the dye will spread out. Children often explain this in a sequential way: the dye wants to go where it is less crowded, so it moves away and diffuses. However, the reality is an emergent process, where the micro scale elements of the system (the molecules) act in such a way to produce what may appear to be a macro scale phenomenon (diffusion). Other examples of emergent processes that are often misconceived as sequential processes are heat transfer as "hot particles" leaving and geese's V-formation as the goal of the flock [41]. In addition to being useful to predict certain topics that students will struggle with, Chi's theory cautions instructors when building on students' prior knowledge when the ontological categories do not align between the prior and new knowledge.

2.1.3.2 Knowledge in Pieces

diSessa's Knowledge in Pieces (KiP) theory [42] is on the opposite side of framework theory in the coherence versus fragmentation of knowledge debate. KiP proposes that naïve knowledge is composed of thousands of relatively independent "phenomenological primitives" or "p-prims" [43]. Here, phenomenological refers to the fact that these naïve ideas, originate from everyday, real-world experiences, similar to in framework theory. Primitive means that the ideas are usually evoked as a whole and are "explanatorily primitive;" the only answer a student can give you about why the belief holds is “because that’s how things are” [44]. Example p-prims, which are intuitive based on a child’s experience but untrue, are “increased effort begets greater results” and “multiplication makes numbers bigger” [42]. With instruction, these p-prims tend to organize and form a somewhat coherent framework.

Both framework theory and KiP recognize that students start with a naïve understanding of the world and that this understanding changes as they learn. The major distinction between the two is the degree of coherence versus fragmentation. In framework theory, students start out with a relatively coherent, albeit naïve, framework that often fragments as they learn and incorporate new knowledge until they reach an expert-like, coherent framework. In KiP, fragmentation is the initial state since “p-prims are many, loosely organized, and sometimes highly contextual” [44, p. 9]. Then, as students learn, coherence increases since “integration (increase coherence) is virtually the definition of conceptual advancement” [44, p. 10].

A large focus of KiP is the context-dependent nature of knowledge [43]. A student may apply a p-prim in one situation but not another, despite the obvious (to the instructor or researcher) connections between the two. For example, diSessa, Gillespie, and Esterly [43] observed that young children change their response about whether a block in a diagram experiences a force depending on the block’s color. Language, students’ moods, time of the day, and many other contextual variables might impact which p-prims students call upon to respond to questions.

In contrast to framework theory, implications of KiP to instructional practice suggest it is unrealistic for instructors to try to confront every one of hundreds or thousands of naïve conceptions that students have or to reliably predict how students will respond to a certain lesson. This position is aligned with constructivist theory, since every learner will bring their own background to a lesson and this background will change how they view and interpret the new material. The KiP theory also predicts that students will have trouble undergoing conceptual change for certain topics because they need to first gain enough knowledge to develop the concept [44].

2.1.3.3 Evidence for Conceptual Understanding Models

Framework theory [39] and the KiP theory [42] exist on a continuum; framework theory argues that knowledge is relatively coherent, while knowledge in pieces argues that knowledge is relatively fragmented. The authors of each theory do not expect every instance of conceptual change to follow their theory exactly; age, subject matter, and other contextual factors likely impact how conceptual change occurs [43]. Further, neither framework theory nor KiP is extreme in their coherence versus fragmentation stance. For example, KiP allows for some coherence: many p-prims are connected, some have wide scope, and children can start to form a coherent framework before schooling. However, there are so many independent elements (p-prims) in a knowledge system, that one cannot succinctly describe it as a single, coherent framework [43].

Both theories have supporting empirical evidence [39], [43], so the choice to apply one theory or the other should be based on context. For example, for SS concepts, [45] found students struggle with continuous versus discrete categories, similar to Chi's argument about category mistakes, and that the p-prim that "time is continuous" may especially challenge students. Ref. [29], [46], [47] similarly suggest that persistent incorrect student reasonings in circuits, fluid dynamics, and mechanics courses are due to category mistakes. Other incorrect reasonings, are dependent on the problem context and not fully explained by category mistakes; Brown, Montfort, Perova-Mello, *et al.* [46] interprets such a finding with framework theory, though KiP can similarly explain this contextually-dependent application of concepts.

The CU theories grounded our understanding of CU, but I did not explicitly adopt a single theory of CU for two primary reasons. First, it is challenging (if not impossible) to measure the amount of coherence or fragmentation of students' mental models [32]. Second, while theories are based on philosophical arguments and backed by empirical data, they were created to paint broad strokes; they aim to understand general mechanisms of understanding and how students think. In their generality, the theories lose context-specific information that may impact exactly how CU is developed in particular sub-fields of engineering.

2.1.4 Measuring Conceptual Understanding

Section 2.1.1 presented varying perspectives on how to define CU; Section 2.1.2 builds on those definitions from the literature and proposes a definition of CU specific to engineering. After defining CU, the next challenge for an empirical study is deciding how to measure it. This section discusses two common methods: (1) think-aloud interviews and (2) concept inventories. Tab. 2.2 summarizes the main points for both methods.

2.1.4.1 Think-aloud Interviews

Think-aloud interviews involve asking participants to solve a problem while saying what they are thinking. These interviews are one type of concurrent verbal reporting, which means that participants talk as they are doing the task, rather than providing a retrospective report of their reasoning after completing the task. Much of the literature on think-aloud interviews comes from usability interviews aimed at assessing how participants interface with a new product. Charters [48] discusses how the methodology can be used in qualitative research in an education setting and many previous engineering education studies use think-aloud interviews, *e.g.*, [29], [49]–[51].

The label "think-aloud interview" and "clinical interview" are sometimes used interchangeably, with the exact methodology varying considerably between studies. Typically, in comparison to think-aloud interviews, clinical interviews involve more interaction between the interviewer and

Table 2.2: Summary of the two primary ways to measure conceptual understanding.

	Think-aloud interviews	**Concept inventories**
What	Participants talk through their thought process as they solve problems	Participants take a multiple-choice test
Typical analysis	Qualitative: Speech communication theories, quantizing qualitative data	Quantitative: Item response theory, statistical tests
Disadvantages	• Do not fully reflect participant knowledge • Do not only reflect participant knowledge • Small sample size	• Possibility of guessing can skew results • Shallow view on participant understanding
Advantages	Deeper view on participant understanding	Repeatable and easy to scale

participant, and interviewers often ask for clarification, elaboration, and confirmation throughout the interview [52]. The goal of a clinical interview is for the interviewer and interviewee to work together to understand the interviewee's reasoning and clinical interviews may be "formative and exploratory," [52] unlike think-aloud interviews where interviewers are typically testing a set of preformed hypotheses. Researchers may also choose to follow a think-aloud interview with a clinical exit interview as a method of data triangulation; see Charters [48] for further discussion of this point.

Ericsson and Simon [53] pioneered the theoretical framework behind think-aloud interviews in the early 1980s [48]. They argue that asking participants to think-aloud does not alter participants' task performance or the structure or sequence of mental processes, and it slows them down "only moderately." (The slow-down in task completion is typically negligible. However, it does mean certain time-sensitive tasks, like juggling, are not suitable to the think-aloud method.) Further, the authors contend that think-aloud data is reliable if it is collected and analyzed according to their methodology [54]. The think-aloud methodology proposed by [53] is very strict in requiring interviewers to avoid social interaction. For example, [54] recommends that the interviewer sit behind the participant to emphasize that the interviewer does not expect any social interaction and to use the reminder "keep talking" rather than "tell me what you're thinking" when necessary to avoid making a social request.

Boren and Ramey [55] suggest that most think-aloud practice is incompatible with the theory from [53]. Instead of blaming bad practice, [55] argues the strict methodology may be too limiting for many research questions and that, even when an interviewer can and does avoid social interaction, interviewees cannot completely ignore the presence of interviewers. Boren and Ramey [55] propose a new theoretical backing for analyzing verbal report data based on speech communication theories to better align think-aloud theory with practice. Although [55] considers usability interviews, the proposed methodology is similar to think-aloud interviews in previous engineering education research studies, *e.g.*, [29], [49]–[51]. The authors suggest making the participant the expert, defining roles at the start of the interview, making the recording as unobtrusive as possible, and interrupting the participant as little as possible. Rather than a potentially brusque "keep talking" command, interviewers should use gentle reminders to think-aloud when needed. Although diSessa [52] does not cite [55], he recommends similar interviewer practices for clinical interviews, and this section weaves in suggestions from both works despite the note above about technical differences between think-aloud and clinical interviews.

Regardless of the theoretical framework, interviews should be carefully designed to minimize the impact of methodological disadvantages. In particular, there are two, related common criticisms of think-aloud interviews:

1. Think-aloud interviews do not *fully* reflect participant knowledge
2. Think-aloud interviews do not *only* reflect participant knowledge.

Both concerns apply similarly for clinical interviews, with the first being less of a concern and the second being more of a concern due to increased interaction between interviewer and interviewee. We summarize recommendations from numerous articles below on how to minimize these concerns.

The first concern, that **think-aloud interviews do not *fully* reflect participant knowledge,** is that participants do not reveal all of their knowledge during an interview and that the data are thus incomplete. In other words, the researcher can only analyze what people say–not their actual understanding.

There are many reasons participants might not reveal all their knowledge. First, if participants do not understand the purpose of the study, they may assume the goal is to "display only normatively correct knowledge, as is commonplace in schools," in which case the "interviewer fails" because participants do not talk through their full thought processes [43]. Second, the specific, and perhaps unusual, setting of an interview means participants may use different knowledge than what they would use in practice; diSessa [52] calls this "ecological validity." For educational contexts where the interview objective is to understand what concepts students use to solve problems, as in [49], the interview context and context of interest are very similar. The artificiality of the interview can also be beneficial because it allows interviewers to present problems that are rare in the real-world [52].

Interview data depend on other contextual variables, *e.g.*, environmental factors such as time of day, what the interviewee was previously working on or thinking about [26], or even on contextual factors that are relevant to the student but not obvious to the researcher as discussed in Section 2.1.3.2. Thus, participants might not use thought patterns in an interview that they might have in another context and the qualitative data reflects "revealed knowledge" rather than directly measuring CU [51].

To lessen the impact of interviews not fully capturing participant knowledge, interviewers can:

1. pilot their protocol and pick questions that elicit good verbal data;
2. if an interview protocol allows for social interaction, ask students for clarification on their thought processes as needed;
3. to help participants engage more fully in the interview, reiterate that the purpose of the interview is to see how the participant thinks, not to judge them or get a correct answer;
4. follow standard courtesies and interview methods to ensure the participant is comfortable during the interview; and
5. create a setting that is as close to possible to the setting of interest.

Finally, as Ericsson and Simon [54] notes, although it may be a "naïve hope that the full detail of cognitive processes could be made overt," interview data can still be useful, especially when used to test or create a hypothesis or in conjunction with other data.

The second concern, that **think-aloud interviews do not *only* reflect participant knowledge,** is related to the point above that other contextual variables influence participants' responses. Of particular concern is that the data can capture the impact of interactions between the interviewer and interviewee. Changes in facial expression, tone, repeating a question, or other signs of interest can all be interpreted by the interviewee and change the results, even when unintended by the interviewer. This is why Ericsson and Simon [54] requires such a strict methodology with minimal social interaction. However, as mentioned above, this is infeasible for many research designs.

To lessen the undesired impact of interviewers themselves on the data, interviewers can:

1. avoid expressing judgement about ways of thinking;
2. minimize interruptions during the interview;
3. set expectations about the role of the interviewer/interviewee at the start of the conversation; and
4. be careful during data analysis to not impose their own viewpoints/ideas.

Good evidence of the success of these strategies is if the interviewee responds "with a range of reactions to questions and offered alternatives," *i.e.*, interviewees sometimes change their positions, sometimes defend, and are sometimes uncertain in response to questions posed to them [52].

Another characteristic not yet mentioned of think-aloud interviews is common to most qualitative methods: because of the labor-intensive methodology, **think-aloud interviews typically involve a relatively small number of participants**. Researchers thus must sample strategically to get high quality data. For example, for a think-aloud interview investigating CU, a strategic sample might include students with a range of levels of understanding. Because of the level of cooperation needed from participants in think-aloud interviews, interviewers need to prioritize working with participants who are interested in and willing to participate in the interview over sampling to achieve diversity across a wide range of variables [48].

Finally, because think-aloud interviews are typically only done with a small number of participants, researchers should only use this methodology when it matches their research question. In engineering education, think-aloud interviews are typically used to more deeply understand how students think or to identify problematic reasonings, *e.g.*, as is [29], [56]. Other methodologies, such as concept inventories, are better suited to testing large populations of students and generalizing results.

After the interview, think-aloud data are typically transcribed then analyzed using any qualitative analysis method. For example, [29] used the constant comparative method and pattern coding. If the questions asked during a think-aloud interview are structured to have a small number of common approaches or expected errors, then the think-aloud data also lends itself to quantization. To analyze their think-aloud data, Wage, Buck, and Hjalmarson [49] tabluated characteristics of each participants' response to each question. Specifically, they listed the participants' final answers, their overall understanding of the concept for that question (right, partial, muddled, or wrong), the type of language used in their responses (technical, non-technical, or both), the overall problem solving strategy (guess, inspection, process, or elimination), and their overall confidence of their answer (high, medium, or low).

2.1.4.2 Concept Inventories

A quantitative approach to measuring CU is to use a Concept Inventory (CI). CIs are collections of validated, standardized conceptual questions. Writing a good CI often starts with think-aloud interviews for content and construct validity. After the initial validation, the questions can be given to a larger group of students, without the need to interview each student individually. With undergraduate students, CIs are a common way to: quantify how much students learned during the semester (*e.g.*, using a pre- and post- test format for test administration), determine which concepts are most difficult for students, analyze common errors (based on frequently chosen wrong answers), and investigate how performance on the test correlates with other variables (*e.g.*, grades or demographics) [26].

Although they can take many formats, CIs are typically timed, multiple choice question tests.

The questions often require no or very little use of formulas or mathematics, and can be answered by reasoning about concepts only. Most questions typically test one concept, while a few test combining concepts. Concepts are often tested in multiple questions and incorrect options can represent common errors.

CIs derive their their name from the Force Concept Inventory (FCI, [57]) [30], which was introduced in 1992. Fig. 2.2 shows two example questions from the FCI [57]. Since the FCI, researchers have developed many inventories to test understanding of other subjects. However, CIs are still more prevalent in physics and mechanical engineering than in EE.

Two metal balls are about the same size, but one weighs twice as much as the other. The balls are dropped from the top of a two story building at the same instant of time. The time it takes the balls to reach the ground below will be:

A about half as long for the heavier ball.
B about half as long for the lighter ball.
C about the same time for both balls.
D considerably less for the heavier ball, but not necessarily half as long.
E considerably less for the lighter ball, but not necessarily half as long.

Two people, a large man and a boy, are pulling as hard as they can on two ropes attached to a crate as illustrated in the diagram to the right. Which of the indicated paths (A-E) would most likely correspond to the path of the crate as they pull it along?

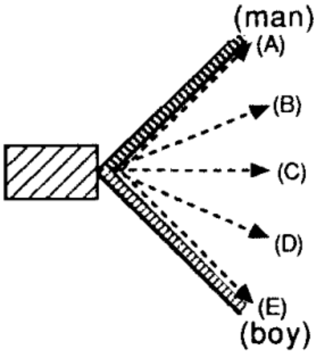


Figure 2.2: Two example questions from the FCI. The correct answers are C and B respectively.

Below is a list of a few EE CIs with the number of citations according to Google Scholar as of January 2022 as a simple proxy for the popularity of the CIs. This is not meant to be a statement about the test quality, as some subject matters are simply studied more often than others.

- DIRECT (Determining and Interpreting Resistive Electric Circuits Concept Test) [58] (cited 680 times). Engelhardt and Beichner [58] developed two versions of the test: one similar to the format described above and one with open-ended questions. The paper discusses validity testing using both high school and undergraduate students.

- SECDT (Simple Electric Circuits Diagnostic Test) [59] (cited 390 times). Although SECDT was developed with high-schoolers, some researchers may find it useful for a beginning undergraduate course. To help with data analysis, SECDT uses a three-tier format: (tier 1) the student's answer to a question, (tier 2) reason why the student chose that answer, and (tier 3) a certainty of response index.

- SSCI (Signals and Systems Concept Inventory) [1] (cited 179 times). Section 2.2.3 describes the SSCI.

- EMCI (Electromagnetics Concept Inventory) [60] (cited 65 times).

There are also CIs that are closely related to EE in physics and mechanics, *e.g.*, electricity and magnetism [61] and controls [62]. However, these CIs tend to focus on examples from physics and mechanics and would likely require modifications before use in an EE setting.

CIs are relatively easy to use, repeatable, and often have already been validated. However, there are limitations to the CI format. As seen in the second example FCI question above, CI questions can require assumptions or outside knowledge. Although the exact mechanism is unknown, conceptual questions can also unfairly advantage different student populations, *e.g.*, [63] shows certain FCI questions advantage men while others advantage women. Another limitation of CIs is that the questions are typically designed to measure single concepts. Thus, CIs do not test concept synthesis (or they only test synthesis of a select few concepts), students' use of scientific practices, or overall problem solving ability [64], [65].

A major limitation of CIs is that students can use process of elimination to answer correctly even when they do not have a full understanding of the concept tested by that question [66]. By analyzing interview data and open-ended final exam problems, researchers concluded that students who have good CU tend to get the corresponding SSCI question correct, but students can still get the SSCI question correct when they do not have full CU [56], [67]. We view this limitation as a known bias: students' scores on the SSCI may over-inflate their CU but are unlikely to underestimate understanding.

One way to combat the problem of students guessing would be to require students to explain their answers. Analyzing the resulting textual data is challenging in large classes and results in data that is less comparable across contexts. New natural language processing algorithms are making it easier to semi-automatically grade student explanations [66], [68]. These systems may allow for combining the main benefit of think-aloud interviews (a detailed picture of students' CU) with the main benefit of CIs (scalability/repeatability). However, such text analysis systems would still be limited in how well they can assess students CU. For example, Goris and Dyrenfurth [47] hypothesized that "misconceptions [in seniors] became more difficult to detect and were possibly hidden under scientific terminology and well developed scientific vocabulary." Montfort, Brown, and Pollock [29] similarly found that students (especially seniors and graduate students) were more confident in their ability and were able to hide a lack of CU behind mathematical skills and PK.

The quantitative data from CIs can be analyzed using many statistical methods. Following the analysis of Hake [69], concepts inventories are often given in a pre/post-test format and researchers typically report normalized gain statistics,

$$\langle g \rangle = \sum_i \frac{\text{post}_i - \text{pre}_i}{\text{max-score}_i - \text{pre}_i}, \tag{2.1}$$

where the sum over i averages the gain of all students in the sample. Normalized gain is an average measure of how many concepts students learned as a fraction of how many that they did not know at the time of the pre-test. Because $\langle g \rangle$ is normalized, it can be used across course contexts "as a rough measure of the effectiveness of a course in promoting conceptual understanding" [69]. One can also use tools from psychometrics such as classical test theory and item response theory to examine individual questions, *e.g.*, as used in [63], [70].

2.2 The Signals and Systems Curriculum

This dissertation concentrates on S&S, which is a standard electrical engineering (EE) undergraduate course covering LTI system properties, impulse and system responses, Fourier transforms (FT), Laplace transforms, and filtering. S&S provides core concepts that are "of fundamental importance in all engineering disciplines" [34, p. xvii]. Although this work concentrates on S&S, the discussion and results likely transfer to many other disciplines with the same challenges.

Despite the importance educators place on concepts in S&S, studies show that many students learn less than half of new concepts in a S&S course [1], and that students can derive the correct answer on procedural questions without being able to explain the underlying concepts [66], [71]. For example, students may be able to use convolution to derive the output of a LTI system given an input and impulse response, without understanding how the math is fundamentally relying on the properties of LTI systems, which is one of the first major concepts in S&S courses.

This section first overviews the S&S concepts that are the main focus of this dissertation: convolution, LTI, FT, and filtering, to provide background to readers who are unfamiliar with S&S concepts. It then reasons why S&S concepts may be especially challenging to students. Finally, this section describes the SSCI–the concept inventory to measure CU in S&S which we use throughout Part 1.5.

2.2.1 The Concepts in Signals and Systems

The introductory undergraduate S&S course typically involves three hours of lecture per week, and is sometimes accompanied by an additional discussion or lab time. The course covers a lot of material, including signal representations, continuous-time systems and system properties, frequency domain concepts, Fourier series, Fourier transforms, and Laplace transforms. This section provides a *brief* background on some of the main concepts from SS and presents previous findings about conceptual understanding that are specific to each concept. We specifically concentrate on LTI, convolution, FT, and filtering concepts; these concepts are covered in detail by standard S&S textbooks [33]–[35].

This section uses the following common notation:

- The radian frequency variable is ω.
- Time functions are always lowercase, *i.e.*, $x(t)$, and Fourier/Laplace transforms are always uppercase, *i.e.*, $X(\omega)$, $X(s)$.
- Input functions are denoted by $x(t)$, impulse responses by $h(t)$, and outputs by $y(t)$.
- A generic system is represented as $y(t) = \mathcal{S}\{x(t)\}$.
- Transform pairs are always indicated by the same letter, *i.e.*, $X(\omega)$ is the Fourier transform of $x(t)$.

All signals are continuous time signals.

2.2.1.1 Linear and Time Invariant

A system is time invariant if any time shift in any input signal yields a corresponding time shift in the output signal but does not otherwise change the output signal, *i.e.*, if $\mathcal{S}\{x(t)\} = y(t)$ then $\mathcal{S}\{x(t-\tau)\} = y(t-\tau)$ for all signals x and y and for any time shift τ. A system is linear if it satisfies the additivity and homogeneity properties for all possible input-output pairs:

$$\begin{aligned} \text{Additivity :} \quad & \mathcal{S}\{x_1(t) + x_2(t)\} = y_1(t) + y_2(t) \\ \text{Homogeneity :} \quad & \mathcal{S}\{\alpha x(t)\} = \alpha y(t) \quad \forall \alpha. \end{aligned}$$

Taken together, the additivity and homogeneity properties yield the superposition property:

$$\mathcal{S}\{\alpha x_1(t) + \beta x_2(t)\} = \alpha y_1(t) + \beta y_2(t). \tag{2.2}$$

Systems that are both linear and time invariant are LTI; S&S courses focus (almost exclusively) on LTI systems.

In addition to understanding the above definition of the linear and time invariant properties, S&S courses often emphasize other concepts related to LTI systems, including:

- What types of systems are LTI and how to verify whether a system is LTI.

- Few physical systems are truly LTI. However, LTI systems provide a good approximation for many physical systems.

- LTI systems are analytically convenient because the output is completely determined by the input and impulse response. Further, there are many tools, such as the FT, that make LTI analysis easier than analysis of non-linear or time varying systems.

The next section on convolution expands on the last point.

Students rate the difficulty of understanding system properties lower than instructors do [72], perhaps because students associate LTI concepts with relatively easy procedures. Nasr, Hall, and Garik [71] found that students are likely to predict the correct output to an LTI system given an input/output pair and a new input (they can perform the procedure), but they describe their response as self-evident and show no understanding of how the answer is grounded in LTI properties [71]. Students similarly could find the correct step response given the output from a finite duration pulse, but they used intuitive claims of symmetry and extrapolation without demonstrating any understanding of superposition. Although experts likely use the same "tricks" as students, [71] argues that instructors check their assumptions while students rely on them without verification.

2.2.1.2 Convolution

Convolution is the mathematical operation that calculates the output of an LTI system $y(t)$ based on the input signal $x(t)$ and the impulse response $h(t)$. Specifically, the convolution integral is

$$y(t) = x(t) \circledast h(t) = \int_{-\infty}^{\infty} x(\tau)h(t-\tau)d\tau. \tag{2.3}$$

The mathematical notation in (2.3) is confusing on its own; the way convolution is typically written as $y(t) = x(t) \circledast h(t)$ suggests convolution is a point-wise operation. A recognized, but less prevalent, notation is $y(t) = (x \circledast h)(t)$, which emphasizes that the output at time t can depend on the signal and impulse response values for all time. (Section 2.2.2 discusses how language may be similarly misleading or confusing.)

To solve a graphical convolution problem, students are typically taught to use the "reflect-and-shift" method. For a single time value t, the output is the integral of the input $x(\tau)$ point-wise scaled by the reflected impulse response function shifted by t. The canonical convolution example in S&S is two rectangular signals, which results in a triangle if the rectangles are the same width or a trapezoid otherwise. If students have CU of the convolution procedure, then they should understand what properties of the rectangular signals determine the maximum height and start/end times of the output. The SSCI tests these abilities in Q13 and Q15.

Some example concepts from S&S are:

- Convolution is distributive, *i.e.*, $x(t) \circledast (h_1(t) + h_2(t)) = x(t) \circledast h_1(t) + x(t) \circledast h_2(t)$. Therefore, the output of a parallel connection of LTI systems is the sum of their individual outputs.

- Convolution is associative, *i.e.*, $(x(t) \circledast h_1(t)) \circledast h_2(t) = x(t) \circledast (h_1(t) \circledast h_2(t))$. Therefore, the impulse response of the equivalent system to a series of LTI systems is the convolution of the impulse responses for the individual systems.
- Convolution is commutative, *i.e.*, $x(t) \circledast h(t) = h(t) \circledast x(t)$. Therefore, a series of LTI operations has the equivalent effect when applied in any order.
- Convolution with a time-shifted impulse shifts the signal by the same time-shift.

Some connections between system properties and the impulse response can also be understood when considering how convolution works. For example, with a full understanding of the convolution integral (2.3), one can show that a LTI system is memoryless if its impulse response is a scaled impulse at $t = 0$ and that a LTI system is bounded input, bounded output stable if the impulse response is absolutely integrable.

Convolution is often ranked high in difficulty among S&S topics because it is a multi-step process that students often do not connect to a concept or application [72]. Students struggle to determine the limits of integration and the time domain over which the output occurs, and they often incorrectly assume no contribution to the integral when one signal is negative because there is no graphical overlap [71]. Even when students performed part of the procedure correctly, for example adding the start and end times of the inputs to get the extent of the output, [56] found they had memorized "tricks" and did not justify their approach.

Without CU backing the tricks, students tend to overgeneralize examples seen in class [56], [71]. For example, if in-class problems always have the same limits of integration (often 0 to t) or have a unit-amplitude, students may not be able to convolve more general signals. As evidence that students can recognize patterns without underlying CU, [56] found students know the output of the convolution of two rectangles should be a trapezoid, but they did not show understanding of what determines the maximum amplitude nor the slope of the trapezoid. Ref. [71] similarly found that students struggled when a system was non-causal or when a signal differed from typical in-class example problems, *e.g.*, if it did not start at $t = 0$, was defined differently over multiple intervals, or had negative values.

2.2.1.3 Fourier Transform

The FT is a major focus of S&S courses; the overview here is necessarily just a small piece of the very important topic. The mathematical definition of a FT and an inverse FT is

$$\mathcal{F}\{x(t)\} = X(\omega) = \int_{-\infty}^{\infty} x(t)e^{-j\omega t}dt \tag{2.4}$$

$$\mathcal{F}^{-1}\{X(\omega)\} = \frac{1}{2\pi}\int_{-\infty}^{\infty} X(\omega)e^{j\omega t}d\omega. \tag{2.5}$$

Different writers use different notation for the FT of a signal, typically either $X(\omega)$ [35] or $X(j\omega)$ [33], [34], without changing the equations above (one can also use a hat or tilde rather than a capital letter for the FT of a signal, but the main distinction here is what argument the FT takes). Again, the mathematical notation itself can be a source of confusion as the latter notation, $X(j\omega)$, does not align with standard functional notation!

Some example FT concepts are

- When does the FT of a signal exist?
- All physically realizable signals have a FT.
- Properties of the FT including linearity, time shift, time scaling, etc. All these operations in time have a dual operation on the FT, *e.g.*, for time shift $\mathcal{F}\{x(t-t_0)\} = X(\omega)e^{-j\omega t_0}$. The time-convolution property, $x(t) \circledast h(t) = X(\omega)H(\omega)$, is one of the most important such properties.
- The FT is a change of basis from a time domain to a frequency domain representation.

The FT is directly connected to many of the other concepts in S&S, for example, the FT of a system's impulse response, $h(t) \leftrightarrow H(\omega)$, relates the FT to filtering concepts.

Students commonly identify the FT as one of the harder topics in S&S (along with convolution) [51]. Students perceive the FT as hard because: challenging and abstract mathematics are required, students struggled translating the graphical understanding to the symbolic mathematical representation, students did not understand its application/purpose, the topic was not used elsewhere in the class or previous classes, and students struggled to understand the FT units [2], [51], [72], [73]. In contrast to the perceived difficulty of the FT, students see the FT properties as an easier topic [51]. One possible explanation is that, similar to the LTI concepts, students see these properties only at the level of procedural knowledge.

FT concepts readily demonstrate how it is difficult to separate PK from CU by simple definitions or when designing problems for a concept inventory. For example, Fayyaz [51] asked students to find the FT of a signal written as the sum of sinusoids and cosines of varying frequencies and amplitudes. One could attempt to use the FT definition (2.4) and PK to solve the problem. Alternatively, one can recognize that sinusoids are the basis elements of the FT and correspondingly identify the coefficients by inspection. Using qualitative methods, [51] found that students were not able to use CU to recognize the convenient signal representation, but a correct response on a concept inventory would not clarify which methods students used.

2.2.1.4 Filtering

Although there are many reasons and ways to filter a signal, many of the example problems in S&S consider filters that remove certain frequency components of a signal and allow other frequency components to pass through unaltered or with some gain/attenuation. More generally, one can interpret the effects of any LTI system as a filter where the output of the system $y(t)$ is

$$y(t) = \mathcal{F}^{-1}\{X(\omega)H(\omega)\},$$

where $H(\omega) = \mathcal{F}\{h(t)\}$ is the FT of the impulse response and is called the frequency response. Because of the convolution-multiplication duality of the FT, filtering concepts are usually taught in the FT domain and the FT representation is typically more convenient for analysis. Thus, filtering is a very important application of the LTI, convolution, and FT concepts.

Some filter concepts include:

- What does the cutoff frequency mean?
- When is it important for a filter to be causal?
- What are the advantages and disadvantages of different filter design methods?
- There is generally a trade-off between the filter's roll-off steepness and design complexity.
- What is bandwidth?

S&S courses are often taught after a circuits course and textbooks generally relate many of these filtering concepts to RC (resistor-capacitor) circuits.

It is easy to think of many real-world applications of filtering, and the concept of filtering does not add much mathematical complexity beyond the FT. Thus, it is unsurprising that students rate filters as one of the easier S&S topics [72]. However, [49] found students had many problematic reasonings about frequency responses, which are a major concept in filtering. Wage, Buck, and Hjalmarson [49] coined the term "Aitchofjayomegaphobia" (H-of-j-omega-phobia) based on students' fear of frequency responses.

One problematic reasoning about filtering likely stems from how filtering is often introduced by defining ideal low-pass, high-pass, bandpass, and bandstop filters in the Fourier domain. Wage, Buck, Nelson, *et al.* [56] found students over-generalized these simple, example filters; students did not check the gain of the filter nor that the frequency of an input signal was less than the cutoff frequency of a low-pass filter before stating the signal passed through. Some students further took the idea of a filter as a "mask" to mean anything in $X(\omega)$ above $H(\omega)$ would be cut-off (thus if $X(\omega_0) = 2$ and $H(\omega_0) = 1$, a student might say $Y(\omega_0) = 1$) [56].

2.2.2 Why Signals and Systems is Difficult

This section presents an overview of reasons behind common errors in S&S, primarily grouped by the theoretical discussion in Section 2.1.3 and the findings from [50]. Montfort, Herman, Brown, *et al.* [50] analyzed over 250 interviews from four engineering disciplines (material science, transportation engineering, fluid mechanics, and digital logic) to search for patterns in which concepts students find difficult. The authors use a qualitative, emergent analysis approach, followed by thematic analysis to synthesize information across the four engineering areas [50].

One of the most often cited reasons that students struggle with S&S is the highly theoretical, abstract, or **mathematical nature of the material**, *e.g.*, [51], [73]–[75]. The mathematical nature of many S&S procedures means students easily "get lost in the complexity of the calculations without understanding the effects of the process and the concepts involved" [76]. The abstract nature may be especially challenging for students who prefer visualizing concepts. In the EE context, Fayyaz [51] likewise concludes that S&S is hard for students largely because of the inability to directly observe many of the phenomena and the disconnect from daily life.

A second reason S&S might be difficult comes from framework theory and the idea that students struggle to correct **miscategorized concepts**. Montfort, Herman, Brown, *et al.* [50] found that "students often inappropriately group dissimilar phenomena, processes or features." For example, because multiplexers and decoders have mirrored circuit diagrams, students think one is the opposite of the other. This incorrect grouping leads to problematic reasonings and difficulty solving problems. This finding could be consistent with the synthetic framework stage of framework theory or students trying to form a coherent framework through instruction in the KiP theory.

Fayyaz [51] similarly found students incorrectly categorized topics in S&S. Category mistakes in S&S include confusing the differences between the continuous and discrete domain, periodic and aperiodic signals, and finite and infinite duration signals. For example, [51] found that some interviewees mistakenly equated signals in the time and frequency domains: they said that since a constant in frequency has no ω in the expression, it has no frequency and thus the time domain signal is also a constant (the correct response is a time domain impulse). In addition to the framework theory perspective, p-prims, discussed in Section 2.1.3.2 as part of the KiP theory of CU, could also explain the categorization errors. For example, [51] suggests the p-prim that time is continuous makes it challenging for students to understand discrete time.

Another possible source of difficulty with S&S material is that students may use **overly simplified reasoning**. Ref. [50] also found that students reason "using simplified causal relationships." For example, students understood the if-A-then-B in Boolean logic as a cause and effect relationship, without understanding that the statement says nothing about whether B can be true in the absence of A. This finding expands on Chi's theory about emergent processes since it claims that students use causal reasoning in situations beyond just emergent processes. Along with the example of grouping electronic components according to their circuit diagram symbol, this finding suggest that students use overly simplistic reasoning. Therefore, [50] suggests instructors provide more help to students digesting complex systems. This is echoed by [51], who suggests that instructors spend more time telling students how each piece of S&S knowledge fits into the larger system. Specifically, the authors recommend making an analogy between the material and a (relatable) complex system, like an assembly line.

A related explanation for why S&S is difficult is that it involves **many threshold concepts**, which are concepts that transform the way students think (possibly in a way that transforms them as engineers) [2], [77]. For example, one frequently cited threshold concept in signal processing is how one can look at a problem in different domains, *e.g.*, the Fourier or time domain [2], [73]. Because of the transformative nature of the concepts, it can be difficult for instructors to understand students' confusion as they grapple with threshold concepts. Many instructors think of the Fourier-time domain duality as a (simple) change of basis or one of many analytic tools on their tool belt and correspondingly use whichever domain is most convenient. In contrast, students struggle with "mapping from the mathematical abstraction (where negative frequencies, complex exponentials and non-causal systems reside) to the physical reality (where systems have to be causal, frequencies are positive, and signals are real)" [73].

Another common theme in the literature is that S&S **terminology is misleading and confusing**. For example, [78] observed that students attempted to visualize the area between the curve and minus infinity as a literal interpretation of "area under the curve" when asked to do an unfamiliar integral. Students also confuse the technical and literal meaning of time invariant, thus mistakenly believing that a system that shifts the input left/right is time-varying since it is impacting the time of the signal [51]. Further, students may only look at spatial "overlap" when performing graphical convolution (note there is no *physical* overlap if one signal is positive and the other is negative), rather than the integral of the product, because of the word choice in the explanation of the procedure [75]. Wage, Buck, Nelson, *et al.* [56] further cautions instructors about using informal language when defining procedures in class because doing so may lead students to think of the procedures and concepts as magical tricks that they should memorize rather than understand.

Finally, some instructors cite students' **lack of prerequisite knowledge** as a barrier to gaining CU. For example, [73] claims that students enter with "insufficient mastery of pre-requisite knowledge (esp. complex number algebra)." In a related argument, Fayyaz [51] suggests that the main difficulty with S&S is not math background but rather being able to process information at high levels on Bloom's taxonomy. The need to combine math and S&S concepts and translate between representations and domains may thus explain why even students who do well in math prerequisites often struggle with S&S [51].

Regardless of the reason, many students come to fear S&S, and it is considered a weed-out course at many universities [49]. Instructors have tried including more hands-on activities or using research-based instructional practices to help students learn and decrease the fear surrounding S&S, but conceptual understanding remains low [49], [79].

2.2.3 Signals and Systems Concept Inventory

The Signals and Systems Concept Inventory (SSCI) is an instrument designed to measure conceptual understanding of common S&S concepts and to analyze common reasoning errors [1]. Because of the extensive initial study (involving 7 schools and over 900 students), the SSCI offers a benchmark for comparison across institutional contexts. This study uses version 5 of the continuous time SSCI, which we simply refer to as the SSCI [1]. The SSCI has 25 multiple-choice questions on background mathematics, LTI, convolution, Fourier and Laplace transform representations, and filtering (see Section 2.2.1 for an overview of most of these concepts). The Discrete Time (DT) version additionally covers sampling. The SSCI website[1] provides an overview of the development of the SSCI, links to many publications, and contact information for researchers and instructors who would like to use the inventory.

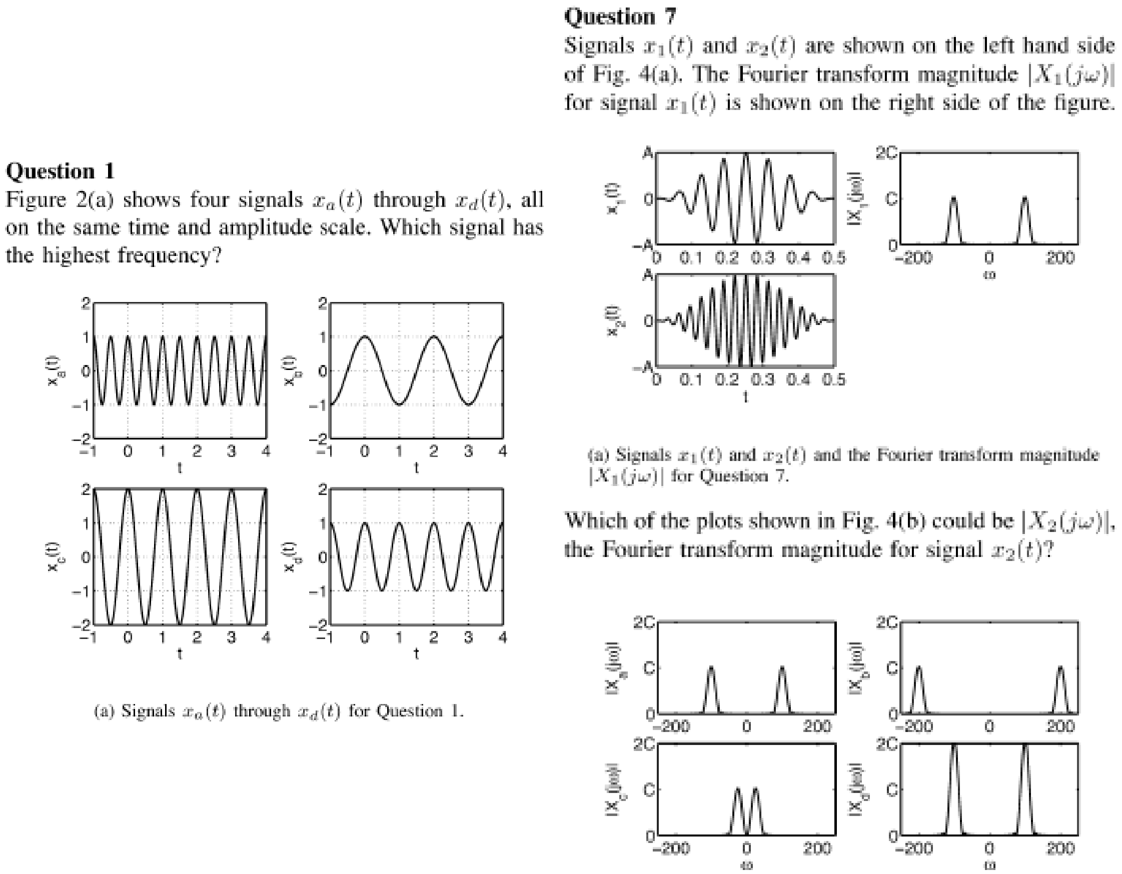
The SSCI takes no more than an hour to complete, emphasizes CU over PK, and presents more problems using figures than equations. There are only a few questions that require synthesizing multiple concepts or that require reverse-reasoning (defined as thinking about a problem in a manner that is not how the material is typically covered in-class). Fig. 2.3 shows two sample questions from a previous version of the SSCI.

CIs became popular with the introduction of the Force Concept Inventory (FCI) [57]. Using the FCI, [69] found $\langle g \rangle = 0.23 \pm 0.04$ in traditional, lecture-based physics classrooms ($N = 2,084$) and $\langle g \rangle = 0.48 \pm 0.14$ in classrooms that used active learning ($N = 4,458$). The low fraction (0.23 or 0.48) of new concepts learned during students' physics course is discouraging. However, the results are encouraging for proponents of active learning, as the introduction of active learning pedagogy led to learning gains two standard deviations above the learning seen in lecture-based classrooms.

The discussion surrounding the surprising FCI results spurred the development of the SSCI, with Wage, Buck, Wright, *et al.* [1] developing the SSCI partially to see if low CU gains were also a problem in S&S courses. The SSCI authors found very similar results: across 2,389 students in 69 courses, the average SSCI gain was $\langle g \rangle = 0.23 \pm 0.11$ in lecture-based classrooms and 0.39 ± 0.08 in active learning classrooms [56], suggesting that students learn an average of 23-39% of the new concepts in their S&S course.

Ref. [67], [80] examined the content and construct validity of the SSCI by comparing student responses on the SSCI to their free-response answers on final exam questions. The authors found a statistically significant correlation between overall SSCI scores and final exam scores. They also found a statistically significant correlation between scores on specific SSCI questions and exam questions on the same concept. Buck, Wage, Hjalmarson, *et al.* [80] also compared student

[1] http://signals-and-systems.org/



Question 1
Figure 2(a) shows four signals $x_a(t)$ through $x_d(t)$, all on the same time and amplitude scale. Which signal has the highest frequency?

(a) Signals $x_a(t)$ through $x_d(t)$ for Question 1.

Question 7
Signals $x_1(t)$ and $x_2(t)$ are shown on the left hand side of Fig. 4(a). The Fourier transform magnitude $|X_1(j\omega)|$ for signal $x_1(t)$ is shown on the right side of the figure.

(a) Signals $x_1(t)$ and $x_2(t)$ and the Fourier transform magnitude $|X_1(j\omega)|$ for Question 7.

Which of the plots shown in Fig. 4(b) could be $|X_2(j\omega)|$, the Fourier transform magnitude for signal $x_2(t)$?

Figure 2.3: Two example questions on the SSCI. The first question is in the background math sub-test and is the easiest on the exam. The seventh question is in the Fourier transform sub-test. These questions illustrate the generally plot-heavy, equation-light nature of the SSCI.

responses on the SSCI to interview data and found that CU was well-measured by the relevant SSCI questions. However, [67] also found insignificant correlations between other SSCI questions and related exam questions (specifically on the question about Bode plots and some convolution questions).

2.3 Conceptual Understanding in Signals and Systems

As mentioned in the previous section, Wage, Buck, Wright, *et al.* [1] found similar gains in CU as Hake [69]: students learned an average of 22±7% new concepts in lecture-bases courses and 39±6% of new concepts in courses with active learning. Other studies report similar gains in S&S courses in a variety of institutional contexts, *e.g.*, [74], [81], [82]. Across these studies, the mean pre-test scores were 35-50% and the mean post-test scores were 50-70%. Taken together, the previous studies of CU using CIs show that students do not learn many of the concepts in core engineering and science courses [1], [47], [58], [69].

There are fewer studies that study the evolution of CU over the course of an undergraduate degree. Such studies have mixed results with different studies showing that (1) students gain CU, (2) students maintain similar levels of CU, (3) students forget or lose CU, or (4) that the evolution of students' CU varies by the type of concept. This section emphasizes EE results, but also includes results from closely related fields, *e.g.*, physics, mechanical engineering, and mathematics.

(1) Some studies found that advanced **students retained or gained** conceptual understanding relative to less experienced students. For example, [47] found senior students had fewer problematic reasonings about circuits than freshmen and sophomores. Multiple other studies found active learning helped students retain physics CU years after their introductory physics course [83]–[85].

(2) Montfort, Brown, and Pollock [29] found **similar CU** for senior students and introductory students. The authors differentiated between PK and CU: senior students were better able to solve problems because they were more successful at completing the correctly identified procedures, but those senior students did not demonstrate more CU. Using qualitative methods, [29] hypothesized the finding was due to students' inability to "reconcile that knowledge with their intuition." Fayyaz [51] similarly proposed that upper-level students may come to question the concepts they previously "learned" and whether they understand them.

(3) Multiple studies found that **students forgot or lost CU** after a relevant course. In a qualitative study, [51] found that many students declined in their understanding of the Fourier representation of a signal one year after S&S. Using the SSCI, [81] similarly observed a "tendency for scores to drop over time," but that scores increased when students took related upper-level electives. However, these S&S studies involve small sample sizes and/or only students in S&S-related courses, making it difficult to generalize the results. In physics, [86] similarly found students tended to forget concepts over time and that repeated exposure helped. The results in [86] are more discouraging since [86] found that repeated exposure decreased the amount students forgot but it did not improve CU above the level of freshman physics students.

(4) Some results suggest that the **evolution of students' CU varies by the type of concept**. For example, [29], [47] generally found that senior or graduate students tended to have persistent incorrect reasonings for certain concepts. Specifically, using the DIRECT and think-aloud interviews, Goris and Dyrenfurth [47] found seniors had significantly fewer problematic reasonings than novices, but similar errors as novices on some topics (the physical aspects of circuits, lending credence to Chi's emergent processes theory) [47]. Fayyaz [51] found that students who took follow-on S&S-related courses had trouble mostly with translating representations, *e.g.*, they tended to think the product of a function and an impulse was a constant rather than a scaled im-

pulse.

2.4 Conclusions

This chapter reviewed the literature on CU and S&S. These ideas are central to the remaining chapters in Part 1.5.

Despite the theorized benefits of and the research on CU, there is still disagreement on, *e.g.*, how students obtain conceptual knowledge [24], the relative importance of CU and procedural knowledge [87], and how CU evolves over time [30]. Further, empirical studies support that there is still much work to be done if we want to help students achieve CU in undergraduate engineering classrooms, *e.g.*, [1], [29], [47], [58].

While the remainder of Part 1.5 emphasizes CU, we do not argue that it should come at the expense of PK. Both types of learning are important, in addition to many other goals of the engineering curriculum such as learning how to learn, gaining laboratory experience, and learning skills such as teamwork and clear writing. However, the focus on CU is supported by multiple large research studies that have shown that students generally do not learn core concepts in their courses [1], [69] and that students lack CU even as they gain PK [29].

Studies show low CU after a specific course, typically taken in a student's early undergraduate career. But there are fewer results on CU near the end of undergraduate engineering degree. It is possible that students do in fact learn the concepts, but not until after a course in the subject when most studies measure CU. Understanding the CU of students years after a course is a main goal of this study, with our case study being S&S concepts.

CHAPTER 3

RQ#1: Conceptual Understanding During the Signals and Systems Course

The first study in Part 1.5 of this dissertation addresses our first research question, RQ#1: "**What is students' CU of S&S concepts at the end of an undergraduate S&S course? What factors predict how many S&S concepts students learn in a S&S course?**" This study served as a pilot study for the following two chapters, which similarly consider measuring CU and factors that impact CU, but which concentrate on senior undergraduate students. This chapter solely considers students in an undergraduate S&S course, who are typically in their second or third year.

The methodology and results on factors influencing CU in this chapter are presented in [4]:

> C. Crockett and C. Finelli, "Factors influencing conceptual understanding in a signals and systems course," in *2021 ASEE Virtual Annual Conference Content Access*, Jul. 2021. [Online]. Available: `https://peer.asee.org/37175`

This chapter expands on that publication, notably by including additional results on measuring CU.

Fig. 3.1 depicts the high-level methodology for this chapter. The remainder of this chapter is organized as follows: The background section overviews the Model of Educational Productivity as an initial framework for predictive factors. The methods section describes the participants, the survey to measure factors, and the analysis techniques. The results section presents the statistics from the SSCIs and the results of a linear regression analysis. Finally, the chapter concludes with a discussion of main findings and how this work leads into Chapter 4 and 5.

3.1 Background: Model of Educational Productivity

There are many theories about what factors influence learning and why some students learn more than others. One such empirically validated model is the Model of Educational Productivity (MoEP) [88]. Based on a synthesis of national science achievement test and a survey given to 3,049 17-year-olds as part of the National Assessment of Educational Progress, [88] found nine

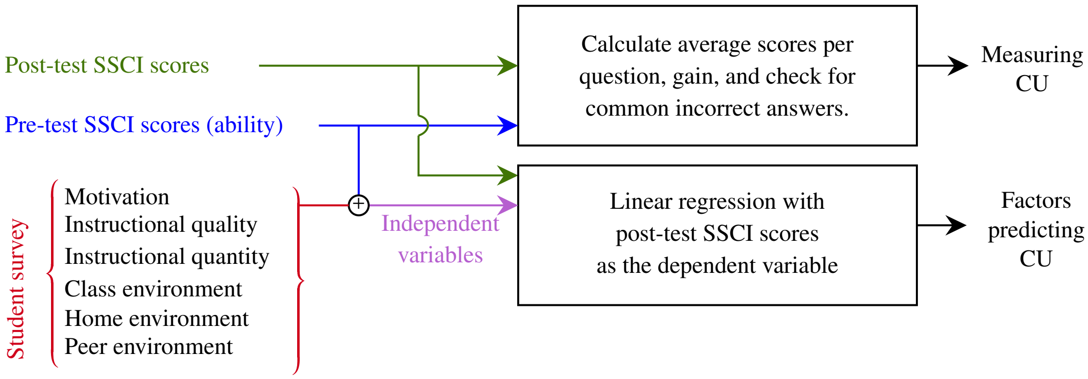


Figure 3.1: High-level view of the methodology for the pilot study to answer RQ#1.

significant factors on test scores (summarized in Tab. 3.1). The original model is

$$\text{Learning} = \alpha \prod_i (f_i)^{\beta_i}, \tag{3.1}$$

where the nine factors, f_i for $i \in [1, 9]$, are commonly described in three groups: three student variables, two instructional variables, and four environmental variables. The nine MoEP factors are summarized in Tab. 3.1 The outcome variable in the MoEP is typically learning, as measured by a standardized test. Other studies have also used the MoEP to investigate student attitudinal outcomes [89] and career aspiration [90].

Table 3.1: Summary of MoEP factors that predict learning [88].

Category	**Factors**
Student factors	(1) Age, (2) ability, and (3) motivation
Instructional variables	(4) Quality and (5) quantity of instruction
Environmental variables	Social psychological environment of the (6) class and (7) home, (8) peer group environment, and (9) exposure to mass media

The three independent student variables are development, ability, and motivation. Briefly, each of these has been defined as:

1. Student *development* is typically measured by student age, though it can be defined as the stage of maturation for students where age may not be a good measure of their development. This variable is commonly omitted in studies with students that are all of roughly the same age [88], [90], [91].

2. Student *ability* measures prior achievement or knowledge. Example measures of student

ability are grades in prerequisite courses or prior Grade point average (GPA) [92], [93]. Because many of the initial tests of the MoEP used existing datasets, researchers had to use non-optimal operationalizations of the factors based on what variables the existing data included, *e.g.*, some of the original studies [88], [94] used socioeconomic status (SES) as a "poor surrogate for IQ or prior achievement tests" [94, p. 288]. Despite this poor measure, the authors still demonstrated the usefulness of the MoEP.

3. In the original MoEP studies, student *motivation* was defined as a willingness to persevere. Example survey items asked students how often they do various activities related to learning when not required [88], [89], [95] and if they try to do their best in class [91]. However, other authors measure student motivation as whether a student thinks it is okay to miss or be late to class (reverse-coded) [90], if students say they work hard in school [91], or using an expectancy-value theory [92].

   Expectancy-value theory proposes that motivation is influenced by the expectation of success and the value of the task [96]. Task value is further divided into attainment value or importance of doing well on the task, intrinsic value or enjoyment of the task, utility value or the task's usefulness for achieving future goals, and the opportunity cost of pursuing the task [96]. The version of expectancy-value theory considered in [92] comes from [97] which considers the expectancy component (can I do the task?), a value component (which combines the intrinsic and utility value), and an affective component (related to the intrinsic value). Bruinsma and Jansen [92] do not consider the attainment value or cost component in the full expectancy-value theory.

The two instructional variables in the MoEP are instructional quality and instructional quantity:

4. *Instructional quantity* is the amount of time a student spends learning, and can include in-class and out-of-class time. For schools with a fixed amount of class time, this factor is often measured by self- or parent-reported number of hours students spent doing homework [88]–[91]. For schools with a varying amount of classes in a subject area, quantity can be measured as credit hours in the subject of interest [89], [92]. The rate of skipping classes can also measure quantity of instruction [90] (alternatively it can measure students' willingness to miss a class and thus their motivation [91]).

5. *Instructional quality* can be measured at the student or class level. At the student level, [88] measured didactic quality of instruction and use of student-centered instruction methods with 16 items on a student survey, [90] included survey items on if the teaching is "good" at school and if teachers listen to students, and [92] included survey items on the quality of the presentation, structure/organization, assessment, and pace of the course. At the class level,

[91] used teacher reports of how often the class did activities like experiments and writing reports. Fraser, Walberg, Welch, *et al.* [89] used an existing dataset, and included both student and class level information to operationalize the quality of instruction; the authors used average science teaching budget per student and a five item student survey on student attitude toward their teacher.

Finally, the third category of independent variables in the MoEP consists of environmental variables and includes exposure to mass media, home environment, classroom environment, and peer environment:

6. *Exposure to mass media* is usually measured as the number of hours a student spends watching television, sometimes split into weekend and weekday hours [88]–[90]. Under this definition, researchers assume that students are generally not watching educational programs and thus expect (and find) a negative relationship between the mass media factor and learning outcomes. In a more recent study of university students, Brouwer, Jansen, Hofman, *et al.* [98] updated the definition of mass media from hours watching television to time spent on social media. Unlike the earlier studies, the authors found a positive connection between mass media and amount of time spent studying. They hypothesized that students are using social media to motivate each other and to ask questions as part of their studying. The other recent study of the MoEP in a university setting did not include a variable for mass media [92].

7. Ideally, the *home environment* factor would measure the amount of intellectual stimulation a student receives while at home. In practice, this factor typically measures various home characteristics such as the presence of an encyclopedia [88], the highest level of parental education [89]–[91], or socioeconomic status [90]. In research with university students, the home environment factor may additionally, or alternatively, be operationalized to account for most students no longer living with their parents. For example, [92] expanded the definition of home environment to include undergraduate students' living environment: in addition to asking about parental educational status, they asked whether students were employed while attending classes. Bruinsma and Jansen [92] found little effect of home environment in their study.

8. The *classroom environment* refers to the social psychological environment of the classroom. Measures of classroom environment vary widely. Studies typically use student survey questions to measure this factor, with questions ranging from asking about morale, such as whether classes are interesting, [88], [90]; how students feel during class (uncomfortable, curious, student, confident, successful, and unhappy) [89], [93]; student attitudes toward

their teachers (*e.g.*, if teachers were able to explain difficult subjects) [92]; or if students feel put-down by teachers or students during class [90]. Reynolds and Walberg [91] takes a different approach and used the number of students that get a college degree and how many students take science as an elective to measure classroom environment. This last definition overlaps with the typical definition of the peer environment variable.

9. Peers can improve learning outcomes directly, by peer instruction, or indirectly, by creating an environment that encourages learning [89]. The *peer environment* factor in the MoEP studies measures the latter: the social psychological influence of peers. Example survey items ask if schooling and good grades are important to friends [90], [91] and how much peer support a students receives [92]. The peer environment variable may be harder to measure in existing datasets; [88] had to use the primary parent's occupation as a poor operalization and [89] was not able to include the peer-group environment variable.

The basis of the MoEP is in economics. One interpretation of (3.1) is that increasing any one variable (other than mass media, which is predicted and shown to have a negative coefficient) will improve learning, but with diminishing returns. Increasing the factor that is currently the smallest will yield the largest increase in learning per unit increase in the factor.

The goal of the MoEP is to model learning in a way that is parsimonious (using only a few factors to explain learning), generalizabe to different student populations/contexts, and repeatable by other researchers [89]. Therefore, the model does not take into account factors that are further removed from learning, such as political characteristics of a school. The model similarly does not include gender or race, as Walberg, Pascarella, Haertel, *et al.* [88] believed that the effect of these demographic variables should only be through the variables in the model. Studies using the MoEP still commonly include gender and race as control variables to help compensate for poor measurement of the independent factors, or due to other theoretical backings that suggest the variables are significant. Another variation on the MoEP is to include mediating terms to the standard direct-effects MoEP (3.1). Reynolds and Walberg [91] proposed one such model with mediating terms, the Reynolds-Walberg MoEP, with the mediating terms motivated by other theories on learning and previous results. The authors tested the proposed model using longitudinal data from 3,116 seventh and eight grade students. The Reynolds-Walberg MoEP explained only 2% more variance in learning outcomes than the direct-effects model (from 52% to 54%), but the fit statistics for the Reynolds-Walberg model were better and the new model aligned well with existing theories [91].

While the MoEP only predicts correlations, not causal links, the nine factors are grounded in many other theories that predict causal connections. Fraser, Walberg, Welch, *et al.* [89] provides a thorough overview of the theoretical frameworks that align with the MoEP, discusses previous studies that look at a subset of the factors, and performs a meta-analysis on 134 research studies that looked at factors that influence achievement outcomes.

Because of factor definition issues due to working with an existing dataset, [88] presented only preliminary results in support of the MoEP. Similarly, [89, Chapter 5] used existing survey data as a test of the model; the authors used the National Assessment in Science survey with roughly 2,000 each of 17, 13, and 9-year-olds. Both studies found that the MoEP explained a surprising percentage of the variation in learning considering the operalization issues, and hypothesized that a better operalization would lead to more explanatory power.

Although a number of studies have confirmed the usefulness of the MoEP, only a handful have used it in the higher education setting. Ref. [95] tested the model at a community college and [92] found that the Reynolds-Walberg MoEP (removing the mass media variable, as they hypothesized it would not matter as much for college students) transferred well to the higher education setting, explaining 23% of variance in grades among 62 first-year students in a mathematics and natural sciences department in The Netherlands. Because of the community college context and small sample size of these studies, the research offers preliminary, but not definitive, evidence that the MoEP extends to a university setting.

Chapter 5 discusses other models and studies of factors that may influence CU. Briefly, in addition to the factors predicted by [88], studies in engineering and physics have hypothesized (with varying amounts of evidence) that longitudinal CU is influenced by: which courses students take as part of their major or as electives [81], [86], whether the course uses a graphical representation of systems (*e.g.*, in S&S, LabView's graphical interface might help student learn better than Matlab's text-based interface [99]), whether students view the concepts as important [72], and the instruction style [83]–[85].

3.2 Methods

We measured students' CU using the SSCI and the MoEP factors using a short research survey. We then used regression models to test for significant predictors of CU at the end of a S&S course and to determine how much variance in CU the MoEP factors explain.

3.2.1 Data Sample

This study includes undergraduates at the University of Michigan (UM) who took the main undergraduate S&S course (EECS 216) in Fall 2019 or Winter 2020. Most of the students were in their second or third year of undergraduate studies, but were classified as third or forth years in terms of the number of credits they have taken. The course emphasized continuous time signal analysis and was taught using the free online textbook by Ulaby and Yagle [100]. In addition to the topics covered by the SSCI, the course briefly discussed sampling theory and the relationship between continuous and discrete time representations.

The S&S course was lecture-based and accompanied by a required lab section that met five times for labs on impulse response, envelope detector, frequency modulation discriminator, amplitude modulation radio, and feedback control. Due to the COVID-19 pandemic, the Winter 2020 section moved to remote instruction half-way through the semester and students' grades defaulted to pass/fail for all courses (students could elect to show a letter grade for the course, but we do not have data on how many of them chose to do so).

Students took the pre-test SSCI in-person during their assigned lab section in the second week of classes (in previous offerings, there was no scheduled lab meeting this week). Students took the post-test SSCI and an additional survey during their last class before finals as part of a review session in Fall 2019. In Winter 2020, students took the SSCI in the last week of classes before finals preparation. Due to the move to online instruction, the Winter 2020 post-test was online and had the answer options randomized. Students received a small amount of course credit for completing the SSCI; they were not graded based on their score. There was no incentive for students to take the research survey in Fall 2019. Due to the grading changes, students who participated in the research in Winter 2020 had a chance to win a small gift card. All student interaction was reviewed by the UM institution review board.

Tab. 3.2 summarizes the number of students who completed the pre-test, post-test and the research survey in Fall 2019 and Winter 2020 out of the 134 students enrolled at the start of both semesters. The numbers exclude students who did not finish the post-test (defined as not answering the last five SSCI questions or more) and students who skipped questions on the survey. Only two students, both from Winter 2020, did not complete the post-test. For analyzing gains in CU during the S&S course, our sample includes the $n = 180$ students who completed the pre-test and post-test and signed the consent form. For the second part of RQ#1 on factors that impact CU, this study

	Pre-test	Post-test	Pre- and Post-test	Pre- and post-test and survey
Fall 2019	118	91	91	78
Winter 2020	114	90	89	46
Total	232	183	180	124

Table 3.2: Summary of the number of students who signed the consent form and completed the pre-test SSCI, post-test SSCI, and/or the research survey during the S&S class at UM in Fall 2019 and Winter 2020. The $n = 180$ students who signed the consent form and took the pre-test and post-test make up the data sample for measuring CU and the $n = 124$ students from that group who additionally completed the survey make up the sample for the linear regression analysis of factors the predict CU.

considers only the sub-sample of the participants who also completed the survey. This decreases the total number of included participants to $n = 124$.

3.2.2 Survey for Predictive Variables

This study uses the MoEP as a framework for factors that may influence CU. The outcome (or dependent) variable was CU at the end of the S&S course, as measured by students' raw post-test SSCI scores. We measured the MoEP factors using a survey that students took immediately after finishing the SSCI post-test. The survey was in the same format as the SSCI (paper in Fall 2019 and online in Winter 2020). For the online survey, we used Qualtrics.

Following the framework of the MoEP, we include student, instructional, and environmental independent variables. All questions are coded such that we predict a positive correlation coefficient between the factors and our outcome variables. All Likert style questions had 5-options and followed the design principles suggested by [101]. Tab. 3.3 summarizes the measures used. The full text of our final survey questions are in Appendix A.2.

In the MoEP, the three independent student variables are age (or level of maturity), ability, and motivation. We do not include the *age* variable, as all students are roughly the same age. This is fairly common in studies that use the MoEP, *e.g.*, [88], [90], [93]. We use students' scores on the pre-test to measure *ability*. The pre-test captures information about a students background and whether they have previously seen S&S concepts, though we recognize that it does not capture a broader view on students' mathematical or even more general academic prior abilities. For student *motivation*, we designed seven Likert style survey questions to capture the intrinsic value and utility value components of motivation as defined in expectancy-value theory [96]. The questions asked how likely students are to major in EE, if learning S&S in interesting, and if students think learning the individual S&S topics (LTI, convolution, FT, Laplace transform, and filtering) will benefit their career. We did not include the background mathematics topic in this question due to its broad definition. Before testing the regression models, we first tested if the seven survey questions may be grouped into a single factor that measures the underlying motivation construct.

The two instructional variables are instructional quality and instructional quantity. For both, we use subjective, individual student opinions rather than a more objective measure of the instructors teaching style or amount of homework assigned. Thus, our commentary on instructional quality and quantity is not meant to reflect on the given instructor. For the *instructional quality* variable, we use responses to a Likert style question that asked students to rate the overall quality of instruction in S&S. Again, our definition most closely follows [92], though we only included a single survey item due to space limitations. For *instructional quantity*, we asked students to self-report the average numbers of hours they spent on homework each week and what percentage of lectures they attended. These are very typical measures. However, in Fall 2019, our research instrument was ambiguous, and some students may have counted time spent on pre-/post-labs toward homework time while others did not. Likewise, students who watched lectures online may or may not have included that in their attendance responses. We clarified the language for these questions in Winter 2020 to specifically ask students to include time spent on pre-/post-labs toward their homework time and online lectures toward their attendance.

The four environmental variables in the MoEP are exposure to mass media, home environment, classroom environment, and peer environment. Following [92], we do not include the mass media

Table 3.3: Example previous measures of the MoEP factors and summary of measures for this study. Surveys additionally asked students for their gender identity and which racial and ethnic group(s) they identify with.

	Factors	Previous definitions/measures	Our measures
Student	**Age**	Often excluded when participants are similar ages [88], [90], [93]	Not included
	Ability	Grades in prerequisite courses or prior GPA [92], [93]	SSCI pre-test score
	Motivation	Participation in optional, course-related activities [88], [93] Expectancy-value theory (measures self-efficacy, interest, and positive feelings) [92]	The average score of 7 questions asking: if students want to graduate in EE, if S&S is interesting, and if understanding convolution, LTI, FT, Laplace transform (LT), and filtering will benefit their career
Instructional	**Quality**	Use of didactic or student-centered instruction methods [88], [94] Quality of presentation, organization, assessment, and pace [92]	Students rate overall quality of instruction of S&S
	Quantity	Hours students spent on homework in a typical week (self-reported)	Avg. hours spent on S&S homework Percentage of lectures attended
Environmental	**Classroom**	Class morale [88], [90] How students feel in class (curious, uncomfortable, stupid, confident, successful, unhappy) [93]	If the learning environment was comfortable
	Home	Highest educational status of parents/guardians [90], [93] If they had an encyclopedia or a newspaper in the home [88], [94]	Highest education status of students' parent(s)/guardian(s)
	Peer-group	If schooling and grades are important to friends [90], [91]	How often peers helped their understanding of S&S
	Mass media	Hours watching television [88]	Not included

variable. We use the highest educational status of students' parents/guardians to measure *home environment*, as in [90], [93]. As seen in the full survey in Section A, the categories for home environment were did not finish high school, high school degree, Associates degree, Bachelor's degree, Master's degree, and Doctoral or Professional degree. Like our measure of instructional quality, we took a direct approach and designed a question asking students if the S&S course learning environment was comfortable[1]. This allows each student to individually interpret learning environment; which is both an advantage and disadvantage of our approach. Finally, we designed a Likert style question that asked if peers helped students' understanding of S&S material to measure *peer environment*. A more typical measure of peer environment is whether schooling and grades are important to friends [90], [91].

We followed best survey practices discussed by Fernandez, Godwin, Doyle, *et al.* [102] to order and phrase the questions. In terms of ordering the questions, we placed the demographic questions at the end of study to decrease issue of stereotype threat. We also adopted the language from [102] on how to ask about gender, race/ethnicity, and parents' education. We used standard labels for the Likert questions, following the suggestions from Qualtrics and [103].

3.2.3 Model Testing

The main contribution of this study is testing the MoEP as a model for students' S&S CU. We test the direct regression model, with no moderating or mediating variables, as in the original MoEP publications [88]. Our outcome variable is CU at the end of S&S and our key independent variables are student ability, student motivation, instructional quality, instructional quantity, home environment, classroom environment, and peer-group environment.

For all Likert style questions, we assume that responses are discretized measures of a latent variable. A latent variable is any variable that cannot be measured directly. CU of S&S is one latent variable of interest in this study. In contrast, blood pressure and heart rate are examples of observable variables. Although we do not know the thresholds for each Likert response, we assume a monotonic relationship between the responses and the latent factor. This allows us to treat the Likert data as if it were a continuous variable in our regression model.

To further test our models, we compare the results against models that include race/ethnicity, gender, and S&S semester as control variables. Walberg, Pascarella, Haertel, *et al.* [88] found that adding race/ethnicity and gender increased the amount of learning variance explained from 25% to 34-36% (in the calibration and validation datasets). The authors hypothesize that the the increased explanatory power of the model comes from the race and gender variables serve as a

[1] We chose the phrase "learning environment" instead of "classroom environment" to encourage students to think about the social-psychological environment, rather than the temperature settings and how much they liked the seats and chairs.

proxy for poorly measured factors. (Due to data limitations, [88] measured student ability using SES, so there were definitely measurement errors in the factors.) Further, they hypothesize that, if the factors are more accurately measured, then adding the race/ethnicity and gender variables will not greatly increase the explanatory power. The semester variable in our study could similarly capture differences in instruction, study population, or class environment that our survey does not fully measure.

For all the models tested, we report which of the independent variables are significant, R^2, and adjusted R^2 values. R^2 measures the proportion of variance in the predicted variable (CU of S&S as measured by the post-test score) that is explained by the independent variables (the MoEP factors measured by the survey and pre-test score). The R^2 statistic is thus a commonly reported measure of how well a regression model fits the data. However, R^2 can be a misleading statistic because it will always[2] increase as the number of independent variables increases. Further, a regression model can perfectly fit *any* data (including random noise) when the number of independent variables is equal to the number of samples.

Adjusted R^2 similarly captures how well the independent variables fit the data in a linear regression model, but it adjusts for the number of independent variables. The adjusted R^2 value takes into account that adding more variables will always increase the amount of variance explained; this measure increases only if the added variables explain more variance than is expected by chance, *i.e.*, if the independent variable is useful in explaining differences in the outcome. Adjusted R^2 is upper-bounded by R^2. Given R^2, the adjusted R^2 is

$$R^2_{\text{adj}} = 1 - (1 - R^2)\left(\frac{n-1}{n-k}\right), \tag{3.2}$$

where n is the number of samples ($n = 124$) and k is the number of independent variables. For the regression model without any control terms, $k = 12$ because there are

- (5) five factors in the MoEP captured by a single independent variable (student ability, student motivation, instructional quality, classroom environment, and peer group environment),
- (2) a factor measured by two independent variables (instructional quantity),
- (4) a categorical factor with a base category and four alternative categories (home environment), and
- (1) the constant offset term in the linear regression model.

For the home environment variable, the base category corresponded to a student's parent/guardian having a high school degree as their highest degree. The four alternate categories were for an Associates degree, Bachelor's degree, Master's degree, and Doctoral or Professional degree (no

[2] It can technically remain the same if the additional independent variable is an identical copy of an independent variable already incorporated in the model.

students in our sample responded that their guardian did not complete high school). Modeling the home environment using four categorical variables allows for non-linear and/or non-monotonic relations between the levels of a guardian's education and the outcome variable of interest in the linear regression.

3.2.4 Data Limitations

We face a number of data limitations in this study. First and foremost is that our data set is relatively small and homogeneous; all of the participants took one of two S&S classes. The small sample size means our statistical tests have low power, so we can expect few of the coefficients in our regressions models to reach statistical significance. The homogeneity of our sample population means there is little variance in some of the independent variables, further decreasing the likelihood that they will be significant.

Second, some variables are defined differently than what is common in the literature, *e.g.*, peer environment. Further, we only had a short survey, so the quality of instruction, peer environment, home environment, and class environment variables are measured by a single question rather than a multiple-question Likert scale. In our regression, we treat home environment as a categorical (discrete) variable, but we treat the other variables as continuous measures. The underlying assumption is that student responses capture a discretized measure of the underlying construct and that the spacing between items is roughly equal.

Ideally, we would have followed more rigorous survey design methods, included more questions to measure each variable, and considered the scale's validity and reliability, *i.e.*, following the practices described in [104]. At the time of designing the survey for this pilot study, I was unaware of the importance of this survey design literature. Section A shows the survey planned for senior students as part of addressing RQ#3; in many ways, the senior survey improves on the survey in this pilot study. For instance, the senior survey more clearly defines instructional quality as the use of student-centered instruction and thus measures instructional quality by asking students to rate how often they were engaged in their courses and how often their courses included active learning. This definition and question structure follows [94]. Due to low participation during semesters with online instruction, this dissertation does not analyze the data from the senior survey, which was originally intended to address RQ#3. Future work should consider using previously validated survey scales to measure each of the variables.

Another limitation is that not all students in the course chose to take the SSCI and to sign the consent form, so it is possible there is an overall bias in our results based on which students participated. This is especially true for the students who took S&S in Winter 2020, when our participation rate was noticeably lower on the research survey.

Finally, the pre-test and post-test SSCI are not perfect measures of ability or CU. As mentioned in Section 2.1.4.2, limitations of concept inventories include that students can guess the correct answer without full CU and the inventory could include questions that unfairly advantage a particular group of students [63]. Further, because students were given credit for completing the SSCI regardless of their score, students were not incentivized to try hard when taking the SSCI. It is possible that students will not take the pre-/post-test seriously, and, therefore, that their scores will not accurately reflect their ability or their problematic reasonings. However, [105] suggests that low-stakes assessments at UM are a valid measure of student CU, so we believe the impact of this limitation is likely small for the students in S&S in Fall 2019. Students in Winter 2020 may not have been incentivized to learn the S&S material as well due to the option to take the course pass/fail. The impact of COVID-19, and particularly the unexpected shift to online instruction and pass/fail grading, likely impacts the Winter 2020 data. We discuss how this limitation seems to have an impact by describing how the semester students took S&S is correlated with many other variables further in the results section.

3.3 Results

3.3.1 Measuring Conceptual Understanding

We first briefly present the raw SSCI statistics. For the S&S classes at the UM in Fall 2019 and Winter 2020, the average and standard deviation of the SSCI was 12.2 ± 2.9 (48.8% ± 11.7%) on the pre-test and 17.9 ± 3.8 (71.6% ± 15.3) on the post-test. Fig. 3.2 shows a histogram of student pre- and post-test scores out of a maximum possible score of 25 and Tab. 3.4 presents the raw data. Average student scores improved when averaging the questions in each of the six SSCI topics. Fig. 3.3 shows the percentage of students who got each question correct on the pre- and post-test organized by the concept categories.

The following list discusses the results for the six concept categories, including the SSCI questions that had common incorrect answers (defined as answers selected by more than 50% of the students who get a question wrong). A summary of the SSCI questions and the fraction of students responding to each correctly is presented in Tab. 3.4.

1. Background math

    - Excluding question 3, students scored 88-99% on the background math questions on the pre-test, suggesting that they enter S&S with the prerequisite information.
    - Students answered question 3 in the background math category correctly only 52% of the time on the post-test, while they scored 96-99% on the remaining three background

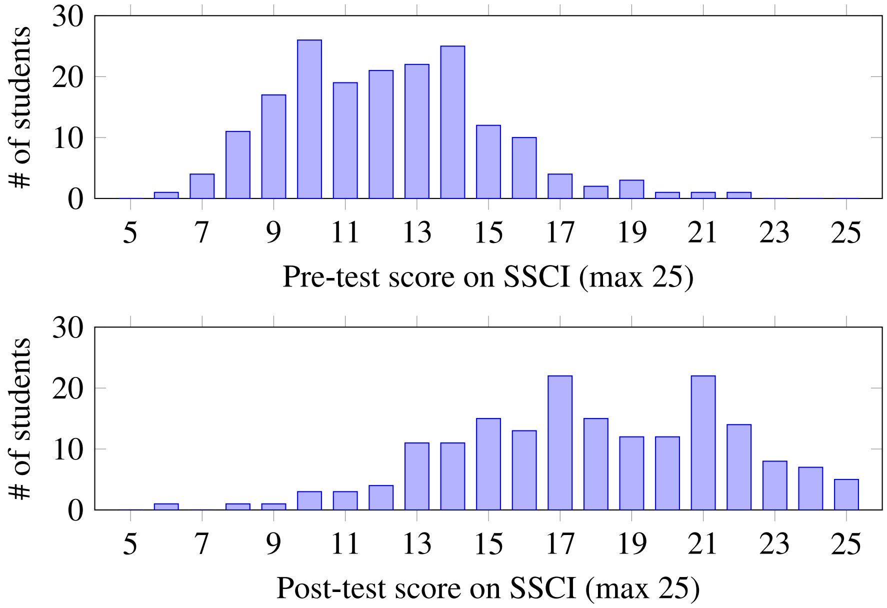


Figure 3.2: Histogram of SSCI scores in S&S on the pre-test (top) and post-test (bottom) for the $n = 180$ students in Fall 2019 and Winter 2020 who completed both tests. The pre-test average and standard deviation was 12.2 ± 2.9 and the post-test average and standard deviation was 17.9 ± 3.8 (both out of a maximum of 25 points).

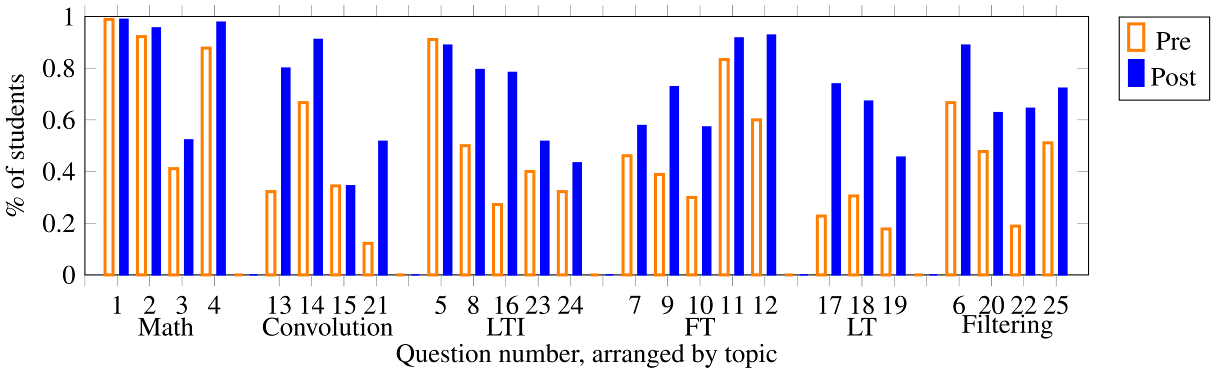


Figure 3.3: Percent of students who got each SSCI question correct on the pre- and post-test. Questions are grouped by sub-test. The raw data for this figure is given in Tab. 3.4.

math questions. Question #3 involved both flipping and shifting a signal. Forty-three percent of students correctly reversed the signal but shifted it the wrong direction.

2. Convolution

   - The convolution question with the highest pre-test score (question 14) tested if students know convolution is commutative. Students may have seen this concept in previous courses, such as differential equations. The other convolution questions require students know what an impulse response or a FT is, which they are less likely to have previously seen.
   - The only convolution question with a common incorrect answer on the post-test is question #15, which asks for the output signal given a rectangular impulse response and input signal. The incorrect answer, selected by 53% of students, has the correct ramp-up and ramp-down times but the incorrect maximum output amplitude. Section 4.3.2 discusses the incorrect answer in more depth as this error is persistent in the population of senior students.

3. LTI

   - Question 5 was the only SSCI question that students did worse on for the post-test (a decrease from 91% to 89%). This questions asks for the output of an LTI system when the input is delayed; the answer may be intuitive for students who have not formally been introduced to the definition of time invariance.
   - The only LTI question with a common incorrect answer is question 24. This questions asks students to determine if a system could be linear and/or time invariant based on three input-output pairs. Based on their responses, 73% of students responded correctly regarding the time invariance but only 51% responded correctly regarding the linearity of the system. Section 4.3.1 investigates question 24 further with senior students.

4. FT

   - Three of the FT questions have incorrect answers. Question 7 asks students to identify the Fourier series given a plot of a periodic signal $x(t)$; the common incorrect answer (22% of students) had the correct non-zero coefficients but with an incorrect magnitude relation. Anecdotally, multiple students asked about this question during review sessions. On question 9, 16% of students selected the FT for a signal with doubled amplitude instead of the requested FT of a signal with twice the frequency. On question 10, 38% of students convolved the Fourier transform instead of multiplying it when asked for the plot of $R(\omega)$ when $r(t) = p(t) \circledast p(t)$.

5. Laplace transform

    - The LT questions had the lowest average of the SSCI topics on the pre-test, likely since few students have seen pole-zero plots before a S&S course. None of the LT questions had common incorrect answers.

6. Filtering

    - The only filtering question with a common incorrect answer was question 20. This questions required Bode plot concepts, which students were not tested on in S&S.

The common errors on questions 3, 10, 15, 20, and 24 are also common errors in that population of senior students (questions 7 and 9 no longer have common incorrect answers in the senior sample). Chapter 4 goes into further detail on the likely errors that students make on many of the concept inventory questions (with emphasis on questions 9, 12, 13, 15, 24, and 25) and how these questions reflect students' CU.

Fig. 3.4 shows a histogram of the individual student gains. Using the SSCI in a pre/post-test format, we observed an average gain of 45.1% ± 28.6% among the $n = 180$ students who completed both tests in EECS 216 at UM. Eleven students had a negative gain, indicating that they scored worse on the post-test than the pre-test. These students had an average pre-test score of 14.3/25 (an additional two questions correct on average than the full sample) and an average post-test score of 12.3 (an additional 5.6 questions incorrect on average than the full sample).

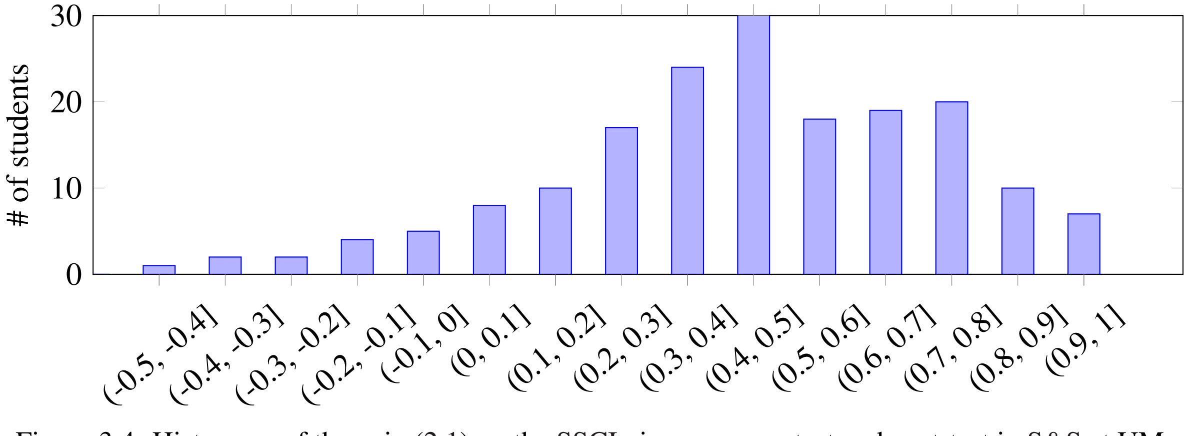


Figure 3.4: Histogram of the gain (2.1) on the SSCI given as a pre-test and post-test in S&S at UM.

Table 3.4: Summary of SSCI questions, including the relevant concept, the main topic of the question, and the fraction of students in S&S that answered the question correctly on the pre-test and post-test (n = 180).

Question	Topic	Concept	Pre	Post
Q1	Math	The definition of frequency.	0.99	0.99
Q2	Math	Time-reversal in the signal domain.	0.92	0.96
Q3	Math	Recognize a time-reversed and shifted signal.	0.41	0.52
Q4	Math	How to find the difference of a signal and its time-shifted version.	0.88	0.98
Q5	LTI	The definition of time invariance.	0.91	0.89
Q6	Filt	The interpretation of a magnitude and phase diagram for a filter.	0.67	0.89
Q7	FT	The definition of the Fourier series.	0.46	0.58
Q8	LTI	Sinusoids are eigenfunctions of LTI systems.	0.50	0.79
Q9	FT	Increasing the frequency of a signal in the time domain correspondingly increases the frequency in the FT domain.	0.39	0.73
Q10	FT	Convolution-multiplication duality of the FT.	0.30	0.57
Q11	FT	The FT is homogeneous.	0.83	0.92
Q12	FT	Convolution-multiplication duality and the FT of a cosine. Or, how multiplication with a carrier wave impacts the FT.	0.60	0.93
Q13	Conv	Convolution computes the output for an LTI system. Graphical convolution of rectangular pulses.	0.32	0.80
Q14	Conv	Convolution is commutative.	0.67	0.91
Q15	Conv	Convolution computes the output for an LTI system. Graphical convolution of rectangular pulses.	0.34	0.34
Q16	LTI	How to determine if a system is causal based on its impulse response.	0.27	0.78
Q17	LT	How to interpret a pole-zero plot to determine a system's causality and stability.	0.22	0.74
Q18	LT	The relation between a system's pole-zero plot and its impulse response.	0.31	0.67
Q19	LT	The relation between a system's pole-zero plot and its frequency response.	0.18	0.46
Q20	Filt	How adding a pole to a frequency response impacts the corresponding Bode plot.	0.48	0.63
Q21	Conv	Convolution-multiplication duality.	0.12	0.52
Q22	Filt	Time-phase shift duality.	0.19	0.64
Q23	LTI	The relation between the impulse response of a system and whether the system is causal. Parallel and cascade connections of systems.	0.40	0.52
Q24	LTI	Graphical interpretation of linearity and time invariance.	0.32	0.43
Q25	Filt	Low pass filtering of windowed signals.	0.51	0.72

3.3.2 Model of Educational Productivity Factors

Before presenting the linear regression results, this section discusses descriptive statistics for the independent variables, results from combining the seven motivation questions to for a single motivation variable, and correlations between the independent variables.

Considering only the $n = 124$ students who completed the research survey, the average class SSCI pre-test scores was 49.5% ± 12.0%, the post-test score average was 73.4% ± 14.4%, and the class gain was 47.3%. The SSCI scores are significantly higher ($p < 0.05$) than the scores presented in Section 3.3.1, which considered students regardless of if they completed the survey on MoEP factors. Thus, there may be some sample bias as students who completed the research survey on average scored better on the SSCI than those who did not.

3.3.2.1 Student Motivation Variable

Tab. 3.6 summarizes the responses to each of the seven survey questions relating to motivation. There was one question on intent to get an EE degree (EE), one on their interest in S&S, and one question each on how much students thought understanding convolution (conv), LTI, FT, Laplace transforms (LT), and filtering (filt) would benefit them in their careers. Looking at the distribution of the responses in Tab. 3.6, we note that most students were interested in learning S&S, even though many did not expect to graduate with a EE degree (many computer engineering majors at UM elect to take S&S from a list of possible core elective requirements). Students thought convolution was the topic least likely to benefit them in their careers, followed by LTI, while they thought filtering was the most likely to be beneficial (only one person disagreed that it would benefit them). Finally, students responded very similarly to the question about LT and FT, rating them in between filtering and LTI on perceived future usefulness.

Table 3.6: Summary of responses to Likert questions measuring student motivation. The questions asked if students planed to graduate in EE, if students thought S&S is interesting, and if students thought understanding each of the S&S topics will be beneficial in their career. Likert response options are strongly disagree (SD), disagree (D), neither agree nor disagree (N), agree (A), and strong agree (SA). The most common response on each question is bolded. The reported mean is calculated by numbering the responses from 1 to 5.

	SD	D	N	A	SA	Mean
Plan to graduate in EE	19	26	10	15	**54**	3.48
SS is interesting	2	9	14	**63**	36	3.98
Beneficial to career: Convolution	7	23	34	**42**	18	3.33
Beneficial to career: LTI	3	12	25	**61**	23	3.71
Beneficial to career: FT	2	8	16	**54**	44	4.05
Beneficial to career: LT	2	5	15	**59**	43	4.10
Beneficial to career: Filtering	0	1	7	41	**75**	4.53

We first considered if these seven survey questions could be combined to form a composite variable. To do so, we used the principal component factor analysis method [106]. The Kaiser-Meyer-Olkin (KMO) measure indicates if a factor analysis is warranted. A KMO close to 1 suggests the variables are linearly dependent and all measure the same underlying construct, while a value below 0.5 or 0.6 is usually considered unacceptable [107]. The KMO for the seven motivation survey questions was 0.84, suggesting that the items merited a factor analysis. There was only a single eigenvalue greater than one, suggesting a single variable, which we call student motivation. Every question loaded heavily onto the variable, with loadings between 0.59 (plan to graduate in EE) and 0.87 (understanding FT is beneficial to career). Cohen's alpha was 0.84 (there was a 0.47 average covariance with seven items in the scale).

We formed the student motivation variable as the average of the seven survey items. This approach to forming the variable had a 0.99 correlation coefficient with the variable generated using the loadings as weights for the items, suggesting that our results are unlikely to differ between the two methods. The final student motivation variable has a mean of 3.9, a standard deviation of 0.75, and a range of 1.4-5.0.

3.3.2.2 Independent Variables: Descriptive statistics

This section summarizes the independent variables in the MoEP and our control variables.

The two student independent variables were student ability and student motivation. We used the pre-test score to measure student ability and the average of seven Likert survey questions to measure student motivation. Tab. 3.7 presents high-level statistics for both variables.

The responses to the two instructional quantity questions were diverse (see Tab. 3.7). Students reported spending 1.5 to 48 hours on homework in an average week (average 8.4 hours, standard deviation 5.7 hours) and attendance ranged the full scale from 0 to 100% (average 70%, standard deviation 33.8%). As mentioned in the limitations section, the two survey questions regarding quantity of instruction are more prone to measurement error due to inexact question wording.

Table 3.7: Summary of the student ability, student motivation, and instructional quantity independent variables. The ability score is the SSCI pre-test score, out of a maximum possible 25 points. The motivation variable is the average of seven Likert questions, as described in Section 3.3.2.1 and has a scale of 1-5.

	Student Ability:	**Instructional Quantity:**		**Student**
	Pre-test	**HW hours**	**Attendance (%)**	**Motivation**
Mean	12.4	8.4	70.8	3.9
Standard deviation	3.0	5.7	33.4	0.75
Minimum value	6	1.5	0	1.4
Maximum value	21	48	100	5.0

For instructional quality (Tab. 3.8), 85% of students thought that the instruction was either good or excellent; no students thought it was poor. Likewise, most (69%) students agreed that the classroom environment made them feel comfortable, with 11% disagreeing and the remaining 20% responding neutrally. The responses to how much peers helped their understanding were more spread out; 29% of students responding neutrally, 32% thought peers helped only a little or not at all, and 39% thought peers helped a lot or a great deal. Finally, for the home environment variable, roughly an equal number of students (26-33%) said the highest education status of their parent(s)/guardian(s) was a bachelor's degree, a master's degree, or a professional degree. The remaining 10% of students were divided between their parents having an associate degree and a high school degree.

For the demographic variables, 100 students were male and 24 were female. Most students were either white or Asian. To protect student anonymity, we defined a categorical race variable that divided students into white (n = 71), Asian (n = 34), and other (n = 19) as the possible races. The other category included all students that responded that they were another race or biracial.

Table 3.8: Summary of responses to Likert questions measuring instructional quality, peer environment, and classroom environment. Likert response options are strongly disagree (SD), disagree (D), neither agree nor disagree (N), agree (A), and strong agree (SA). The most common response on each question is bolded. The reported mean is calculated by numbering the responses from 1 to 5.

	SD	D	N	A	SA	Mean
Instructional quality	0	2	17	**53**	52	4.25
Peer environment	12	28	**36**	33	15	3.09
Classroom environment	0	14	25	**58**	27	3.79

3.3.2.3 Simple Correlations

Having considered descriptive statistics for all the variables separately, this section now presents correlations between the independent variables. Ideally, the independent variables would be uncorrelated, suggesting that they measure different underlying constructs.

Tab. 3.9 shows the significant (p<0.10) pair-wise correlations between the independent variables. The two highest correlations are (1) between instructional quality and classroom environment (0.50) and (2) between student motivation and instructional quality (0.45). (1) The relation between instructional quality and classroom environment is easy to imagine: the quality of the lecture can easily impact how students feel in the classroom, and students who feel comfortable or uncomfortable with the learning environment are be likely to attribute art of that to the instructor. (2) The relation between student motivation and instructional quality can be similarly explained by the instructors impact on students, *e.g.*, a good instructor may help students see the value in the class material. Conversely, students who are dissatisfied with their instructor may be more likely

Table 3.9: Significant (p<0.10) pair-wise correlations between independent variables in the MoEP (non-significant correlations are not listed). Bold numbers denote variables that are significantly correlated with p<0.01. For correlations between two continuous variables (ability and instructional quantity), we report standard correlation coefficients. For correlations involving one or two discrete variables (instructional quality and the three environmental variables), we report Spearman correlations.

Factor	(1)	(2)	(3)	(4a)	(4b)	(5)	(6)	(7)	(8)
(1) Student ability	1			**-0.26**		0.20		-0.17	
(2) Motivation	.	1	**0.45**		**0.26**		**0.28**	**0.26**	0.23
(3) Instructional quality	.	.	1	-0.20	0.18		**0.50**	**0.27**	**-0.27**
(4a) Instructional quantity: homework	.	.	.	1	**0.21**		-0.20		**0.42**
(4b) Instructional quantity: attendance	.	.	.	.	1		0.18		**0.28**
(5) Home environment	.	.	.	.	.	1			0.20
(6) Classroom environment	.	.	.	.	.	.	1	**0.28**	**-0.31**
(7) Peer group	.	.	.	.	.	.	.	1	
(8) Semester	.	.	.	.	.	.	.	.	1

to consider dropping out of EE or the signal processing track in EE. It is also possible that students who are more interested in learning the material seek out additional instruction either by asking questions in class or attending office hours, and then rate instructional quality higher because they received targeted attention.

Tab. 3.9 shows that the semester variable is significantly correlated with most of the other variables. Semester is positively correlated with student motivation, both measures of instructional quantity, and home environment, suggesting students in Winter 2020 were more motivated, worked on S&S more per week, and had parent(s)/guardian(s) with higher education levels. The negative correlations suggest students in Winter 2020 perceived the instructional quality and classroom environment as poorer than their Fall 2019 peers.

We hypothesize that the students in the Fall 2019 and Winter 2020 were very similar coming into S&S (supported by the statistically insignificant correlation between semester and the pre-test student ability measure). However, because of the interruption in instruction and switch to pass/fail grading due to COVID-19, the population of students who chose to take the survey in Winter 2020 is likely different. COVID-19 likely also explains the lower ratings of instructional quality and classroom environment for Winter 2020. Although there was a different instructor, both instructors historically have received high ratings and, while there are certainly differences in their teaching, these differences are likely overshadowed by the impact of the transition to online learning. As evidence for this argument, three students commented in the survey's free response field that the unexpected switch to online instruction negatively impacted their perception of the class and made it harder to learn.

Tab. 3.10 summarizes the independent variables and presents how each independent variable is correlated with the post-test SSCI score (the outcome variables). Two of the variables have a negative correlation: instructional quantity and peer group. Although many of the other variables have the expected positive correlation sign, many are only weakly correlated. Rather than discussing these results in depth, we turn to considering regression models, which allow us to control for the effect of the other variables.

Table 3.10: Correlation between factors in the MoEP and the SSCI post-test score. The third column is the mean value for the student ability, student motivation, and the two instructional quantity variables. The third and fourth column are the mode value and Spearman correlation for the remaining variables, which are all categorical or measured using a single Likert question.

Factor	**Measurement**	**Mean/Mode**	**Correlation**
Student ability	Score on the pre-test (out of 25)	12.4	0.47
Student motivation	Average of 7 Likert questions	3.9	0.22
Instructional quality	Rate overall quality of instruction	4	0.15
Instructional quantity	Avg. weekly homework hours	8.4	-0.13
	Self-reported attendance	70.8%	0.02
Home environment	Highest education of parent(s)/gaurdian(s)	Masters	0.10
Classroom environment	Comfort of learning environment	4	0.14
Peer group	Frequency peers helped with their understanding	3	-0.13
Semester	Fall 2019 or Winter 2020	Fall 2019	0.05

3.3.3 Regression Models

We now present our main results from the linear regression to test which factors from the MoEP predict CU. Our first regression model included the seven independent variables from the MoEP. All of the independent variables, except home environment, were continuous regressors. The second model additionally included the categorical race/ethnicity, gender, and semester control variables. Both models used robust statistics to correct standard errors for possible heteroskedastic noise. Tab. 3.11 summarizes the models.

The model without any control variables (see the middle column of Tab. 3.11) explains 28.3% of the variance in post-test SSCI score and is significant at the α=0.01 level. Student ability (β=0.53, t(df=112)=5.04, p<0.01) and student motivation (β=0.86, t(df=112)=1.90, p=0.06) are the only two statistically significant predictors, both with the expected positive coefficients. The coefficients of the several other independent variables are negative, but none are significant. None of the control variables are significant ($p > 0.20$).

When we include the control variables (see the rightmost column of Tab. 3.11), the expanded model explains 30.4% of the variance. However, R^2 will always increase when we add variables to a linear regression model, so R^2 is not an ideal measure. The adjusted R^2 value actually decreases slightly when we add the control variables as seen in the last row of Tab. 3.11: R^2 for the model

Table 3.11: Coefficients (with p-values) and R^2 values for the regression models. Insignificant coefficients ($p > 0.10$) are shown in gray text. The home environment variable is not included in the table it was not significant. The coefficient β represents the predicted student score on the SSCI if all of the independent variables were zero. Although the result is statistically insignificant, consider the coefficient for the semester variable in the third column as an example of how to interpret the control variable results. The 0.45 coefficient suggests that, if all other variables in the model were equal, a student in S&S in winter 2020 would be expected to score 0.45 points higher on the SSCI than a student in S&S in fall 2019. All of the models shown are for the raw post-test score as the output (on a scale of 0-25) and are significant at the $\alpha < 0.01$ level.

Variable	**Without control variables**	**With control variables**	**Two variable**
Student ability	0.53 (0.01)	0.51 (0.01)	0.55 (0.01)
Student motivation	0.86 (0.06)	0.87 (0.08)	0.89 (0.03)
Instructional quality	0.06 (0.91)	0.09 (0.85)	·
Instructional quantity (homework)	0.02 (0.66)	0.02 (0.70)	·
Instructional quantity (attendance)	0.00 (0.84)	0.00 (0.85)	·
Peer environment	−0.39 (0.20)	−0.35 (0.26)	·
Classroom environment	0.45 (0.22)	0.55 (0.16)	·
Constant offset term (β)	6.99 (0.01)	7.36 (0.01)	8.14 (0.01)
Semester: Winter 2020	·	0.45 (0.60)	·
Gender: Female	·	0.14 (0.85)	·
Race/ethnicity: Asian	·	-0.49 (0.49)	·
Race/ethnicity: Other	·	-1.44 (0.10)	·
R^2	28.3%	30.4%	25.7%
Adjusted R^2	21.3%	20.8%	24.5%

without control factors is 21.3%, and this decreases to 20.8% when including control factors. This suggests that these variables are not worth including.

After finding that student ability and motivation were significant predictors, we tested if either was moderated by any of the control variables, or if either student ability or motivation had a quadratic effect. We found no evidence of the control variables moderating the impact of student ability or motivation nor of a quadratic effect.

For comparison, we ran a final regression model with only the two significant independent variables, student ability and student motivation, as the independent variables. The model (at $p < 0.01$) and both variables (at $p < 0.01$ and $p < 0.05$ respectively) were all statistically significant. This model explained 25.7% of the variance in post-test SSCI scores and had an adjusted R^2 value of 24.5%.

3.4 Discussion

This study addressed the first research question: “What is students’ CU of S&S concepts at the end of an undergraduate S&S course? What factors predict how many S&S concepts students learn in a S&S course?” In addition to considering significant factors in our regression results, the following sections discusses the overall explanatory power of the linear regression model and the descriptive statistics on the SSCI scores.

3.4.1 Conceptual Understanding

By collecting pre-post SSCI data, we were able to compare CU gain to other studies. Both the pre-test scores (mean 49%) and post-test scores (mean 72%) are higher than those reported in [1] (40% and 53% respectively). The only other research paper we found with a similar pre-test average in a S&S course was [74], who reported a 49% pre-test score, a 68% post-test score, and an average gain of 38% among 19 students in an interactive third-year course. Note that many other papers present similar, or higher, pre-test scores, but they are in upper-level electives such as digital signal processing, *e.g.*, [81].

Our observed gain of 45 ± 29% is also higher than the average gain reported for lectured-based or interactive S&S courses (22 ± 7% and 39 ± 6% respectively) [49]. Other studies report similar [99] or much larger [81] gains (70% or more), but these are in upper-level digital signal processing courses. One possible explanation for the higher scores and gains is that the later versions of the SSCI removed some of the more challenging questions [67], [70].

Another hypothesised reason for the high pre-test scores and gain is the effect of prerequisites, and specifically differential equations at UM. Like at many institutions, differential equations can be either a prerequisite or a corequisite with S&S, but 67% of UM students take it as a prerequisite. The qualitative data presented in Chapter 5 suggests that this mathematical background, particularly being introduced to convolution, helped students in S&S. As a specific example of how a prerequisite could increase both the pre-test score and gain, we hypothesize that students may understand convolution at the beginning of the semester, thus raising their pre-test scores, but not yet know it is the correct operation to use to compute the output of a LTI system given the input and impulse response. As evidence that students have some background on convolution, students scored 74% on question 14 of the pre-test, which tests whether students understand the commutative property of convolution. As corresponding evidence that students do not fully understand how to apply convolution to LTI systems, students scored 39% on a convolution question (#13) asking students to recognize the output of an LTI system given plots of the input and the impulse response. Since students understand much of what is needed to answer questions 13, it is easier for them to gain the CU by filling in a small gap in their knowledge. This study did not gather data to test the proposed hypothesis, so we leave this as a direction for future work.

3.4.2 Factors Influencing Understanding

To understand the dataset, we first considered simple correlations between each independent variable and the post-test score. Tab. 3.10 shows that all the correlations were positive except the average homework hours and peer group. The negative correlation with homework hours may be capturing that students who struggle in S&S need more time to complete their homework, but still perform worse on the SSCI. Although we cannot test causality with our data-set, this hypothesis is supported by our finding of a statistically significant negative correlation ($p < 0.01$) between student ability and average hours spent on homework.

The simple correlation results are helpful to understand the data, but the regression results allow us to test the predictive power of each independent variable, while controlling for the other independent variables. Our regression results (Tab. 3.11) show that only student motivation and student ability were significant when we controlled for all the variables. Further, when we ran a regression model with only these two variables, we found they were able to explain 25.7% of the variance in post-test scores. This suggests one could predict outcomes in S&S CU based only on measurements of student ability and student motivation almost as well as basing a prediction on all variables in the MoEP. However, the finding that these variables remain significant in the full model is arguably more important because it suggests that student ability and student motivation are significant *even when* we control for the other variables in the MoEP. We found no evidence of a quadratic effect of student ability or student motivation nor of a moderating effect of race and gender with the student ability or student motivation variable.

There are two likely explanations for why only two of the predicted seven independent variables were significant. First, as discussed in Section 3.2.4, the survey is incomplete and contains measurement errors due to poorly worded questions and variables measured by a single question. Second, our sample population is students from UM who had one of two instructors, similar homework assignments, and similar lecture material (though students were split into two sections in both semesters). Although we explicitly measure student perception rather than directly measuring the environment, there is still very little variance in the instructional setting because both courses are from the same university and share many resources such as the textbook and homework problems. These instructional and environmental independent variables might become significant if we were to survey students across multiple courses at different universities.

3.4.3 Explanatory Power of Regression Models

I hypothesized that the Model of Educational productivity would have explanatory power. The regression models with the seven independent variables explained 28-30% of the variance in post-test score, depending on whether we included the demographics and semester control variables. This is less than the amount explained in the early studies with high school students [91], but more than the amount explained in the original tests of the MoEP [88] and in the tests in other university settings [92]. Thus, we found no evidence to reject our hypothesis.

As the MoEP was originally tested for measuring learning in high school students, we consider the 28% of variance in CU among undergraduate students to be high. Adding variables specific to undergraduate students would likely help explain more of the variance. Similar to the claim in [88], we also expect designing additional survey questions to better measure the seven factors would increase the explanatory power of the model.

Walberg [88] suggests that race and gender variables should not increase the explanatory power of the MoEP if the environmental variables are properly measured. We see only a small increase increase in explanatory power when including gender, race, and semester, and none of these variables are significant. Further, the R^2 value of the model decreased when we added the control variables, suggesting that including them is not useful. Because our environmental variables are not significant, it is difficult to say if the changes in the coefficients between models with and without demographics reflect poorly measured environmental variables.

3.5 Summary and Conclusion

The first part of RQ#1 asked: **What is students' CU of S&S concepts at the end of an undergraduate S&S course?** Using the SSCI, we found that students learned almost 50% of new concepts during the semester they took S&S, which is a larger gain than typically reported during an introductory S&S course. However, we noted a large standard deviation (29%) in our relatively small sample size.

This naturally leads to the second part of RQ#1: **What factors predict how many S&S concepts students learn in a S&S course?** The goal in addressing this question is to discover factors with a positive influence which instructors, curriculum designers, or students can control. We used the Model of Educational Productivity as a basis for collecting survey data on seven possible factors: student motivation, student ability, instructional quality, instructional quantity, home environment, peer group environment, and classroom environment. We excluded two variables from the original MoEP, student age and mass media, based on our sample population of undergraduate students.

The regressions with the MoEP factors explained 28-30% of variance in CU at the end of S&S. We conclude that the model is a good starting place for understanding CU in an undergraduate setting. However, the model could be improved by tailoring factors to undergraduate students. For example, course selection might have a significant influence for college students since they generally have some freedom in their curriculum. We recommend future studies test other models of learning to see if a different model may more successfully explain differences in CU.

We found two of the factors, student ability and student motivation, were significant across models, even when controlling for all other factors and demographic variables. Future work should investigate if the relationship is causal (higher levels of student ability and student motivation lead to more CU) or if it is mediated by another variable. If the relationship is found to be causal, instructors can use this finding to focus more attention on enforcing prerequisites or providing materials for students without sufficient background knowledge. Instructors can also motivate students with real-world applications and examples of how S&S concepts may be useful to students in their future classes or careers.

CHAPTER 4

RQ#2: Measuring Conceptual Understanding of Senior Undergraduates

This chapter considers CU of senior undergraduate engineering students in S&S. Section 2.2.1 provides background on the concepts in S&S and Section 2.2.2 discusses reasons why S&S concepts may be difficult for students. This chapter is primarily drawn from the following publication [5]:

> C. Crockett, H. C. Powell, and C. J. Finelli, "Conceptual understanding of signals and systems in senior undergraduate students," Submitted to: *IEEE Transactions on Education*, 2022

Students typically enroll in S&S during their second or third year, and little is known about CU of students years after they complete a S&S course. Does CU increase as students have time to process ideas and see concepts repeated in upper-level courses? Does CU decrease as students do not use them on a daily basis and forget what they learned (or crammed) for their S&S final exam? Is there a more nuanced relation? Motivated by these questions, this chapter considers the second research question: **What is the CU of S&S concepts among senior students?** Gaining a better understanding of seniors' CU can help instructors and curriculum designers identify specific concepts that need emphasis.

This study mixes qualitative data from think-aloud interviews and quantitative data from the SSCI to measure CU. Think-aloud interviews involve asking students to solve a problem while saying what they are thinking. These interviews help researchers understand students' thought processes and identify specific concepts that student struggle to apply correctly. Section 2.1.4.1 discusses the theoretical backing for think-aloud interviews and the three primary limitations of think-aloud interviews: that participants may not reveal all of their knowledge, that the interview data may reflect things other than the participant's knowledge, and that think-aloud interviews generally involve a small number of participants. The procedural diagram in Fig. 4.1 summarizes the mixed methodology for this study.

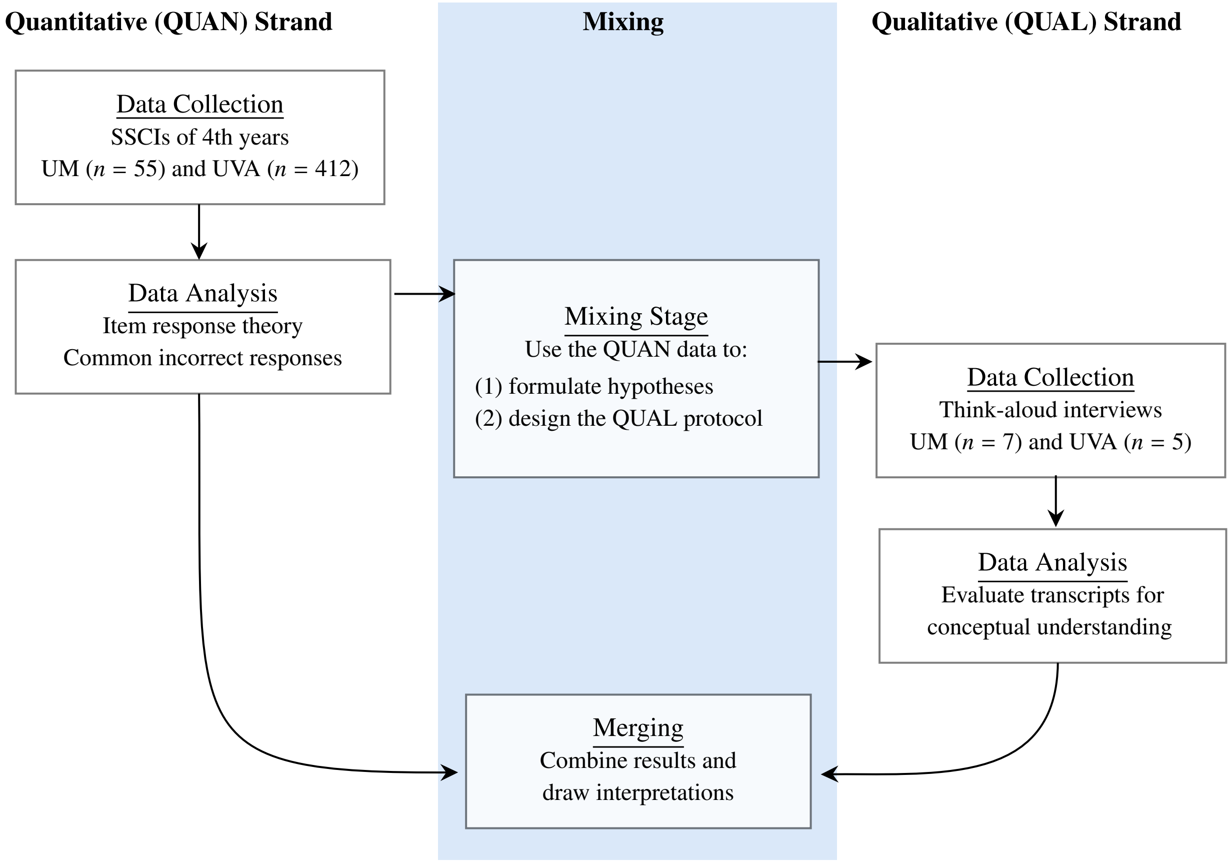


Figure 4.1: Procedural diagram for RQ#2: What is senior students' conceptual understanding of signals and systems? The arrows indicate the flow of data.

Unlike think-aloud interviews, CIs are easy to give to a large number of students. However, the inventories are limited by the multiple choice format [56], [66] as students may be able to guess the correct answer without having full CU. Further, students do not have to explain concepts in their own words, making it harder to identify the true level of CU and the source of any incorrect answers. Section 2.2.3 describes the SSCI used in this study. Because of the extensive initial studies involving seven schools and over 900 students [1], [49] and follow-on validity studies [67], [70], the SSCI offers a benchmark for comparison across institutional contexts.

This chapter largely extends from, and compares to, the work of Wage *et al.* [56] by investigating how their results generalize to a population of seniors. In [56], the authors of the SSCI collate results from the original SSCI study, "data of opportunity" from their courses and other studies, and video homework problems where students are required to explain their reasoning. Similar to the think-aloud interviews in this study, the video homework problems provided [56] with more insight into how well students understood the concepts and how students approached problems, *e.g.*, whether they used concepts, procedures, or "tricks" to answer. This chapter compare to their findings throughout Section 4.3 and 4.4.

4.1 Background: Item Response Theory

Psychometrics is “the science of measuring latent[1] variables” [108]. Item Response Theory (IRT) is a psychometric tool to analyze test responses and validate the reliability of the test. IRT recognizes that an individual’s performance on a test is dependent on both the individual’s ability (defined here are their CU) and the test questions themselves. van der Linden [108] describes the history and current trends in IRT.

As part of testing the first wide-spread version of the SSCI (version 3), Buck, Wage, and Hjalmarson [70] used IRT to analyze 1,276 pre/post-test continuous-time SSCI exams. To estimate the difficulty of each question β_i and the ability of each student θ_s, [70] used the single parameter binary logistic IRT model,

$$P_{i,s} = \text{Prob}(X_{i,s} = 1 \,|\, \theta_s, \beta_i) = \frac{e^{\theta_s - \beta_i}}{1 + e^{\theta_s - \beta_i}}, \tag{4.1}$$

where $X_{i,s}$ is a binary variable that indicates whether student s correctly answered question i, θ_s is the overall conceptual understanding of student s, and β_i is the difficulty of question i. Following a common random effects IRT model [108], θ_s is typically assumed to follow a zero-mean normal distribution with unit variance. When a student’s ability exactly matches the item difficulty ($\theta_s = \beta_i$), the model predicts a 50% chance of the student responding correctly. A negative β_i (low difficulty) means that a majority of the student population is expected to answer that question correctly while a positive β_i (high difficulty) suggests less than half will respond correctly. $P_{i,s}$ increases, *i.e.*, there is a higher probability that student s will get item i correct, as θ_s increases or β_i decreases.

An extension to the IRT model (4.1) is to also estimate item discrimination scores, α_i. Under this two parameter model, the probability that student s correctly answered item i is modeled as the logistic function

$$P_{i,s} = \frac{e^{\alpha_i(\theta_s - \beta_i)}}{1 + e^{\alpha_i(\theta_s - \beta_i)}}. \tag{4.2}$$

The item discrimination score determines how steeply $P_{i,s}$ increases around $\theta_s \approx \beta_i$. Larger discrimination scores imply questions that better differentiate between students who do and do not understand the tested concept.

To help interpret the item discrimination and difficulty scores, Fig. 4.2 shows two item characteristic curves for question 9 and 15 on the SSCI based on the results presented in Section 4.3. Item characteristic curves plot student ability versus the estimated probability of answering cor-

[1] Section 3.2.3 defined a latent variable as any variable that cannot be measured directly.

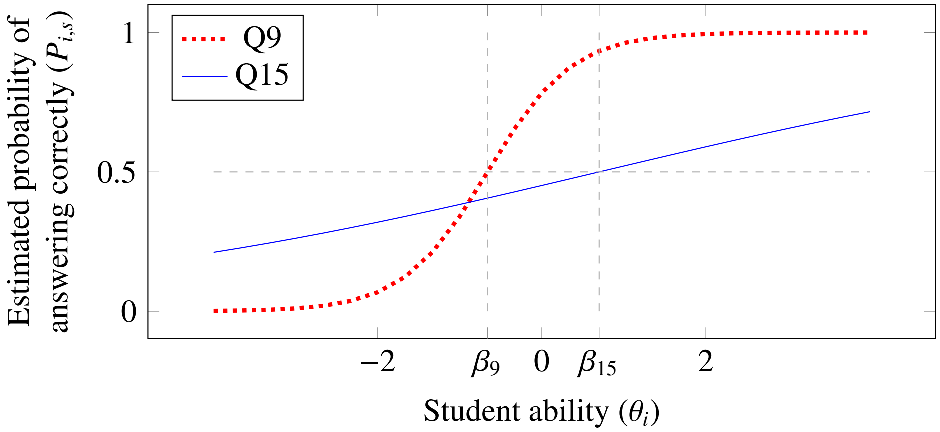


Figure 4.2: Example item characteristic curves. The estimated probability of answering correctly is 50% at the item difficulty β_i and the slope at that point is given by the item discrimination α_i. For question 9 (dotted, red line), α_9 = 1.95 and β_9 = -0.66. For question 15 (solid, blue line), α_{15} = 0.25 and β_{15} = 0.70.

rectly based on the IRT model (4.2). As student ability increases on the x-axis, the chances they answer the question correctly increases on the y-axis. Question 9 has a relatively low difficulty (β_9 = -0.66), so it intersects the line $P_{i,s}$ = 0.5 on the left-hand side of the plot. (The line $P_{i,s}$ = 0.5 marks where a student is estimated to have a 50% chance of answering correctly.) In contrast, question 15 has a high difficulty (β_{15} = 0.70); the curve for question 15 intersects the $P_{i,s}$ = 0.5 line at a higher ability. The slope of the curve at the item difficulty is determined by the item discrimination; question 9 has a larger item discrimination (α_9 = 1.95) than question 15 (α_{15} = 0.25). When writing exams, instructors may chose to look for high discrimination scores because such questions better distinguish between students who do and do not know the material.

The IRT model (4.2) will never exactly match the data; consider that $P_{i,s}$ in (4.2) is continuous while the observed data is discrete. Therefore, given a set of responses to a test such as the SSCI, finding the student abilities, item discrimination scores, and item difficulties is an optimization problem. The goal is to find a set of parameters, $(\theta_s, \alpha_i, \beta_i)$, such that the IRT model (4.2) best predicts the observed values, *i.e.*, $P_{i,s}$ should generally be high when a student answers a question correctly. Weissman [109] and Guo and Zheng [110] discuss approaches to fitting IRT models.

4.2 Methods

This mixed methods study uses SSCI data from senior undergraduate engineering students at UVA and UM. Undergraduates at both universities are typically full-time, non-transfer students and the test score data for first-year students indicates that admissions for both universities is selective [111]. The SSCI data is supported by qualitative data from think-aloud interviews that further investigate how the SSCI responses reflect CU. Fig. 4.1 depicts the overall mixed methodology.

Table 4.1: Example SS concepts.

"What it is" concepts	"Why it matters" concepts
• The definition of a linear and/or time invariant system (Section 4.3.1, Q24). • Convolution requires flipping and shifting a signal then computing the integral of the product of signals (Section 4.3.2, Q13, Q15). • The FT relates the frequency and time domains (Section 4.3.3, Q9). • The convolution-multiplication duality of the FT operation (Section 4.3.3, Q12). • The FT of the output of a system is the multiplication of the frequency response and FT of the input signal (Section 4.3.3, Q25). • The definition of an ideal low pass filter (Section 4.3.3, Q25).	• Convolution is the correct operation to find the output of LTI systems (Section 4.3.2, Q13, Q15). • Multiplying a signal by a cosine centers the original FT at the frequency of the cosine and is useful in applications such as radio broadcasting (Section 4.3.3, Q12). • Low pass filters are useful to remove high frequency signals (Section 4.3.4, Q25).

Unlike the participants in [56], who were in a S&S course, the seniors in this study are generally multiple semesters removed from their S&S course and have thus taken additional courses. Considering a limited sub-sample[2] of the study population for whom course data was available (n=53 from UM, n=90 from UVA), 75% of UM seniors had taken or were currently taking a S&S related elective, *e.g.*, digital signal or image processing, communications, or a control course. In contrast, only 19% of the UVA students had taken a S&S related elective (there are more computer engineering majors than electrical engineering majors at UVA). Although course data for all participants was not available for analysis, the authors expect these percentages are reflective of the entire study population.

This chapter uses the definition of CU proposed in Section 2.1.2. Tab. 4.1 describes the specific "what it is" and "why it matters" concepts that this chapter focuses on, the results section that further discusses each concept, and the corresponding question number (Q#) on the SSCI.

4.2.1 Quantitative: Concept Inventories

Tab. 4.2 lists the main concepts in the 25 SSCI questions along with the IRT results presented in Section 4.3. Students may use different concepts or PK for a given question, so the listed concepts are non-unique. For example, students may use the formula for convolution and procedural knowledge to answer question 13, which asks for the output signal given a rectangular input sig-

[2] These are students who took the SSCI in Fall 2020 or Fall 2021 and completed a survey asking them about which courses they took.

nal and impulse response. Or, students may use CU of convolution and the features of the input and impulse response to determine the start and end time, maximum amplitude, and ramp-up and ramp-down length of the output signal. Tab. 4.2 and Section 4.3 include short descriptions of the questions for context, but we avoid details of the questions and answers to preserve the integrity of the SSCI as a test and research instrument. See [1] for example questions and further description.

The 467 participants who took the SSCI include 412 UVA students and 55 UM students. The sampling strategy was to target a representative sample of seniors in terms of students' interests within EE. Thus, in contrast to many previous studies of the SSCI with senior students [66], [74], [81], we do not sample in a S&S-related upper-level elective such as digital signal processing. Giving the SSCI in a course such as digital signal processing is helpful to determine if students who elect the courses have the pre-requisite material and to test how the course improves CU. However, we sought to examine CU of S&S across the senior population, including students who did not continue to take S&S-related electives.

UVA students learned S&S as part of a series of three Fundamentals courses that intermix the curriculum typical in Linear Circuits, Electronics, and S&S courses. Students took the Fundamental courses during their second and third years. The classes emphasized connections between subjects and mixed lectures and labs in a studio-based format. For more information on the course design at UVA, see [112]. Seniors took the SSCI roughly halfway through the fall semester of their senior year for a small completion grade as part of a required capstone course at UVA between 2016-2021.

At UM, students took a single continuous-time S&S class in their second or third year, though most students were classified as third or fourth years based on number of credits. The course was lecture-based, used the free online textbook by Ulaby and Yagle [100], and was accompanied by a required lab section that met five times per semester. At UM, the SSCI was not tied to any class, but the research project was advertised to students in a required senior-level course between 2020-2021. To decrease self-selected sampling bias, UM students were not told the tested concepts were from S&S. Students were given $10 gift cards for completing the SSCI.

Tab. 4.2 shows the percentage of students answering each SSCI question correctly μ_i and the results of an IRT analysis of the SSCI data. We report μ_i over only UVA students as the UM sample is likely biased because a small percentage of students participated so their scores are unlikely to generalize to the average student population. In contrast, the IRT model accounts for differences in student ability, so we are able to include both UVA and UM students and still expect a reasonable estimate of the difficulties of each SSCI question for senior students. We used Stata to estimate the two-parameter logistic model (4.2) with robust standard errors [113].

Preliminary IRT results also informed the sub-sample of SSCI questions included in the think-aloud interviews, which are highlighted in Tab. 4.2. These preliminary results came from SSCI exams given before 2021 and exams given during the S&S course at UM from the pilot study described in Chapter 3, for a total of $n = 406$ SSCI exams. The next section describes how we selected the final think-aloud questions.

4.2.2 Qualitative: Think-aloud Interviews

This section describes the think-aloud methodology for this study; see Section 2.1.4.1 for a discussion of the theory behind think-aloud interviews.

Think-aloud interviews in EER generally follow the structure of presenting problems to students, asking them to solve them while explaining their approach, and recording the interview for later analysis. Further, the interviews generally follow most of basic recommendations in [114, Ch. 4]. For example, [114] explains that it is important to make sure the participant feels at ease by explaining the purpose of the interview and that there are no hidden motives. This explanation can help to create an atmosphere of confidence and easiness, which is important in think-aloud interviews because participants may be embarrassed by the way they approach problems or because they do not not have the knowledge they think they should have.

However, think-aloud interviews differ in the amount of interaction between the interviewer and interviewee. Ríos, Pollard, Dounas-Frazer, *et al.* [115] used interviews with relatively little interaction: the interviewer initially read a prepared prompt, reminded the student to think aloud, and then the student worked on the given problems. The interviewer asked follow-up questions only after the interviewee completed the tasks. Wage, Buck, and Hjalmarson [49] used semi-structured interviews, allowing for ad hoc probes during the interview. Fayyaz [51, Ch. 3] also used a semi-structured interview; the author asked students to think-aloud to collect concurrent data then asked probes after each question (as needed) to collect retrospective data. Montfort, Brown, and Pollock [29] took yet another approach with more interviewer-interviewee interaction. The interviewer from [29] initially made participants more comfortable by starting with introductions and general discussion and explicitly addressing the possibility of participants feeling uncomfortable. For every question, the interviewer described the diagrams verbally, asked if the question was clear, encouraged the student to ask questions and take their time answering, and attempted to encourage and relax the student in between questions. The interviewer occasionally asked broader questions to test a specific hypothesis during the interview.

The differences in level of interaction can be attributed to differences in research question and theoretical framework for think-aloud data. Interviews with less interaction are better aligned with the theories of Ericsson and Simon [53], while the theories of Boren and Ramey [55] allow for increased interaction; see Section 2.1.4.1. For this study, we did not want probes from the interviewer to change how students approached the later questions on the protocol. Therefore, to

Table 4.2: Summary of SSCI questions, including the relevant concept, the IRT discrimination (α_i) and difficulty (β_i) estimates and corresponding standard errors, and the fraction of students answering correctly (μ_i). Highlighted questions are used in the think-aloud interviews, starred questions were used in [56], and questions with RR require reverse reasoning.

Question	Concept	Discrimination α_i	Difficulty β_i	μ_i
Q1	The definition of frequency.	1.20 ± 0.48	-3.69±1.14	0.98
Q2	Time-reversal in the signal domain.	1.01 ± 0.20	-1.39±0.24	0.78
Q3	Recognize a time-reversed and shifted signal.	0.36 ± 0.17	0.48 ± 0.30	0.45
Q4	How to find the difference of a signal and its time-shifted version.	1.70 ± 0.28	-1.21±0.14	0.79
Q5	The definition of time invariance.	N/A	N/A	0.93
Q6*	The interpretation of a magnitude and phase diagram for a filter.	0.85 ± 0.15	-1.39±0.24	0.73
Q7	The definition of the Fourier series.	1.49 ± 0.22	-0.69±0.10	0.67
Q8	Sinusoids are eigenfunctions of LTI systems.	1.33 ± 0.20	-0.60±0.10	0.64
Q9*	Increasing the frequency of a signal in the time domain correspondingly increases the frequency in the FT domain.	1.95 ± 0.29	-0.66±0.09	0.67
Q10-RR	Convolution-multiplication duality of the FT.	0.72 ± 0.15	1.43 ± 0.29	0.25
Q11	The FT is homogeneous.	0.72 ± 0.19	-2.42±0.56	0.83
Q12	Convolution-multiplication duality and the FT of a cosine. Or, how multiplication with a carrier wave impacts the FT.	0.90 ± 0.16	-0.82±0.16	0.63
Q13*	Convolution computes the output for an LTI system. Graphical convolution of rectangular pulses.	0.93 ± 0.16	-1.03±0.18	0.69
Q14-RR	Convolution is commutative.	0.74 ± 0.17	-2.01±0.65	0.82
Q15*	Convolution computes the output for an LTI system. Graphical convolution of rectangular pulses.	0.28 ± 0.11	0.70 ± 0.43	0.47
Q16	How to determine if a system is causal based on its impulse response.	1.12 ± 0.16	-0.04±0.10	0.47
Q17	How to interpret a pole-zero plot to determine a system's causality and stability.	0.71 ± 0.15	-0.29±0.15	0.54
Q18	The relation between a system's pole-zero plot and its impulse response.	0.94 ± 0.16	0.45 ± 0.14	0.40
Q19	The relation between a system's pole-zero plot and its frequency response.	0.86 ± 0.16	0.99 ± 0.20	0.31
Q20	How adding a pole to a frequency response impacts the corresponding Bode plot.	0.40 ± 0.13	-2.01±0.65	0.70
Q21-RR	Convolution-multiplication duality.	0.88 ± 0.14	0.51 ± 0.14	0.39
Q22-RR	Time-phase shift duality.	0.71 ± 0.14	0.51 ± 0.17	0.40
Q23	The relation between the impulse response of a system and whether the system is causal. Parallel and cascade connections of systems.	0.26 ± 0.12	1.83 ± 0.87	0.38
Q24	Graphical interpretation of linearity and time invariance.	0.31 ± 0.12	1.28 ± 0.56	0.40
Q25*	Low pass filtering of windowed signals.	1.25 ± 0.19	-0.87±0.13	0.70

avoid altering the participants' thought processes during their first pass on the concept questions, I saved any follow-up questions until the participant completed all six questions. However, I did not attempt to minimize all social interaction. For example, for one of the participants who was more nervous, I reminded them to think-aloud by saying, "If you just say whatever it is you're thinking about. Again, I'm not here to judge if your though process is correct. [Just say] whatever it is." Phrasing this reminder as a social request follows the suggestions of Boren and Ramey [55] rather than Ericsson and Simon [53] and can help to make the conversation feel more natural to participants.

When I conducted the think-aloud interviews, I followed standard best practices for interviewing [55], [116], [117]. To help put students at ease, I introduced myself, reviewed the consent form, and explained that the purpose of the research was to understand different approaches to the problems. I emphasized that we were more interested in their thought process rather than whether they ended up selecting the correct answer and that I was not there to judge or test them. The first interviewee asked if I had written the problems; in the following interviews, I explained that other researchers wrote the problems. Participants were told they, not the interviewer, were the expert on how they thought and that they need not ask for verification of anything during the interview. Finally, I explained how to use the tablet computer and shared the plan for the interview session: that students would be asked to think aloud while solving the six problems and that we might return to some of the problems after they are done to ask follow-up questions.

The think-aloud interviews included six questions from the SSCI. One question tested LTI concepts (Q24), two tested convolution (Q13 and Q15), two tested FT (Q9 and Q12), and one tested filtering (Q25). Tab. 4.4 summarizes the concepts. Based on the IRT results and the results from [56], we made hypotheses about CU specific to each concept to test in the think-aloud interviews; these hypotheses are introduced in Section 4.3.

We selected the think-aloud questions based on multiple criteria. First, we decided to use many of the same questions analyzed in [56] to see how their results transferred to our setting. The only question from [56] that we did not use was question 6, which had a very low difficulty score on the initial IRT; we anticipated it may be difficult to recruit students for the think-aloud interview who responded incorrectly to question 6 and there may be too little variation in the data. We then added two additional questions: 12 and 24. By adding question 12 to the think-aloud, we hoped to see if students were more likely to use a "why it matters" concept or a "what it is" concept approach. Most other SSCI questions seem to favor a specific approach, while we were unsure which approach students would use on question 12; Section 4.3.2 discusses this question further.

When selecting the think-aloud questions, we were also analyzing the qualitative data for phase 3 of the overall EER study (see Fig. 1.1). Chapter 5 discusses this data further, but the conversations regarding LTI were intriguing and made us want to include a LTI question in the think-aloud interviews. Of the LTI SSCI questions, 24 was the most difficult according to the preliminary IRT results. The other questions may have been too simple to produce quality think-aloud data; Charters [48] explains that think-aloud questions should be neither too simple nor too complex.

Think-aloud participants worked on the six SSCI questions on the tablet computer (an iPad pro with pencil), which recorded the audio and screen annotations for later transcription and analysis. Students were encouraged to work at their own pace and were allowed to skip and revisit questions if desired. If the student asked for verification about a step or question, I instructed them to make their best guess. Some participants initially asked for verification (either explicitly or by pausing and looking up) after completing the first question. Once reassured that they were helping with the research as long as they continued to talk aloud, most participants completed the questions with no other significant interaction. The think-alouds lasted 30 minutes; all students finished the six problems and most finished in time to allow for follow-up questions.

All students who completed the SSCI were asked if they were interested and available to participate in the think-aloud interviews. If too many students had been interested, our plan was to select interviewees based on specific SSCI answers to achieve a purposeful sample from among the participants who indicated interest and availability. However, we were able to invite all interested students. Participants were incentivized by a small gift card. In total, seven UVA students and five UM students participated. Half of the think-aloud participants had taken or were taking a S&S related elective course. For every think-aloud question, at least six participants had answered correctly when taking the full SSCI and at least two had answered incorrectly. The range of overall SSCI scores among think-aloud participants was 11-24 out of 25.

After transcribing the data, the research team analyzed the think-aloud data across questions and across students for themes on which concepts students used and whether they used each concept correctly. Qualitative analysis involves "coding", which is the iterative process of attaching a label to a segment of the interview. The labels, or codes, can come from the data itself, from a prior hypothesis, or from a theoretical framework [118]. It was relatively easy to create codes for each question because the SSCI questions often have a single or a few common approaches students can use, the incorrect answers often capture particular errors, and we entered the think-aloud interviews with specific hypothesis for each question. For example, for question 12, one code distinguished whether students used the "why it matters" or "what it is" approach, using the definitions for these concepts presented in Fig. 2.1. For question 25 on filtering, we coded if students described the filter as a low-pass filter. Other example codes captured if students were guessing, if they said they were unsure of their answer, and if they tried to use a formula. When analyzing each question, we also open coded the responses to allow for new codes, *e.g.*, we created a code for if students mentioned the convolution of two rectangles is a trapezoid. Section 4.3 discusses the common codes for the think-aloud questions based on the specific hypothesis tested by each question.

4.2.3 Limitations

As in Chapter 3, students were not incentivized to try hard when taking the SSCI; they were given course credit at UVA and a gift card at UM for completion. Thus, it is possible that students did not take the pre/post-test seriously and that their scores do not accurately reflect their CU. Chapter 3 argued that the results from [105] apply to the S&S setting, suggesting that the low-stakes assessment in the S&S course is likely a valid measure of student CU. Whether the same argument applies in this case for the population of senior students is less certain because the SSCI is not directly relevant to a required senior course. Tab. 6.1 shows the fraction of students in S&S and of seniors answering each SSCI question correctly. Some of the decreases in accuracy suggest seniors may not have been trying as hard as the S&S students. For example, 79% of seniors correctly answered the background mathematics question 4 (on finding the difference of a signal and its time-shifted version) while 98% of students in S&S answered correctly. Although it is possible that senior students lose that background knowledge due to non-use, it seems more likely that seniors made careless errors by answering quickly than that they no longer recall how to shift and subtract signals. A counter-point to this argument is that the S&S students may have solved or verified their solution to problems like question 4 more using PK (perhaps to test their own knowledge before the S&S final exam). If true, the SSCI from seniors may be a better measure of CU without mixing effects from PK. How accurately the SSCI reflects CU of students when administered outside of the S&S course and when students are not graded on their answers would be a good area for future work.

As shown in Fig. 4.1, this research is a mixed methods study. Each component carries its own methodological limitations. As noted in the background, SSCI scores may not accurately reflect CU even when students put forth their best effort. The think-aloud interviews help verify which concepts students use for a small number of SSCI questions, but these interviews involved a small sample of both questions and students.

Due to a data collection error, SSCI responses for Q5 are not included for the students from UVA before 2021. When reporting score percentages throughout this chapter, we divide the number of correct responses by 24 instead of 25 for the impacted participants.

The COVID-19 pandemic was unprecedented and it is hard to estimate its impact on CU. Both universities adopted pass/fail grading during the Spring 2020 semester, and anecdotal evidence suggests students did not learn as much in their courses that semester. The results section looks at historical SSCI data from UVA to estimate the impact of this limitation.

4.3 Results

SSCI scores for senior students at both UM and UVA ranged from 5 to 25, with a mean score and standard deviation of 59.3 ± 16.9%. Fig. 4.3 shows a histogram of the scores and Fig. 4.5 shows the fraction of UVA students who answered each question correctly. Tables 4.2 and 4.4 include the fraction of UVA students ($n = 412$) who answered each question correctly μ_i. As briefly mentioned in Section 4.2.1, we do not include UM students when reporting μ_i because these students are likely a biased sample of the overall EE student population. Tables 4.2 and 4.4 also present the results of the IRT analysis: the difficulty β_i and discrimination α_i for each question. The difficulty and discrimination scores use the SSCI responses from all $n = 467$ UVA and UM seniors who participated in the study because, unlike reporting μ_i, the IRT model (4.2) accounts for student ability. (Tab. 4.2 is sorted by question number while Tab. 4.4 is sorted by question difficulty.) Question 5 had only 98 non-missing observations due to the data collection error and thus did not have a good IRT fit. One pattern across concepts is that students struggle with three of the four reverse-reasoning (RR) questions (Q10, Q21, and Q22), with only 25%, 40%, and 39% responding correctly.

To examine the impact of COVID-19, we performed a t-test to determine if the SSCI scores differed significantly between the group of UVA students who took online classes during COVID-19 (2020 and 2021 cohort, n=119) and those who did not (the cohorts before 2020, n=293). We do not have scores from UM students pre-2020 to do a similar comparison. The group impacted by COVID-19 scored significantly worse ($p < 0.01$), with an average of one additional incorrect answer. Fig. 4.4 shows how the SSCI scores vary over time.

The following subsections use the IRT results alongside the think-aloud data to examine CU of convolution, LTI, and FT and filtering. Each sub-section summarizes the difficulty of the SSCI questions and the most common incorrectly chosen answers (defined as answers selected by more

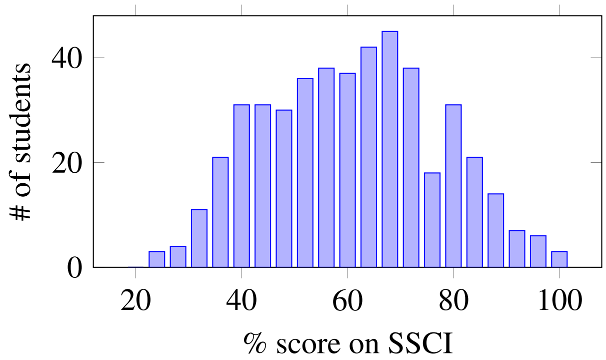


Figure 4.3: Histogram of SSCI scores for all participants.

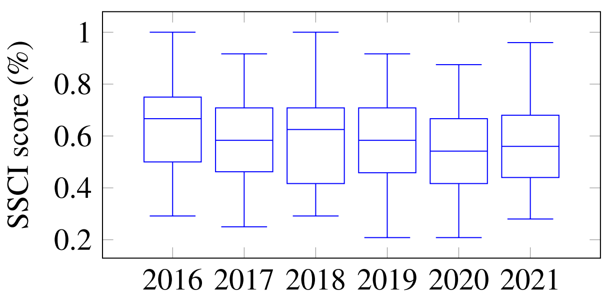


Figure 4.4: UVa SSCI scores by year. The boxes mark the first and third quartile and the whiskers show the minimum and maximum.

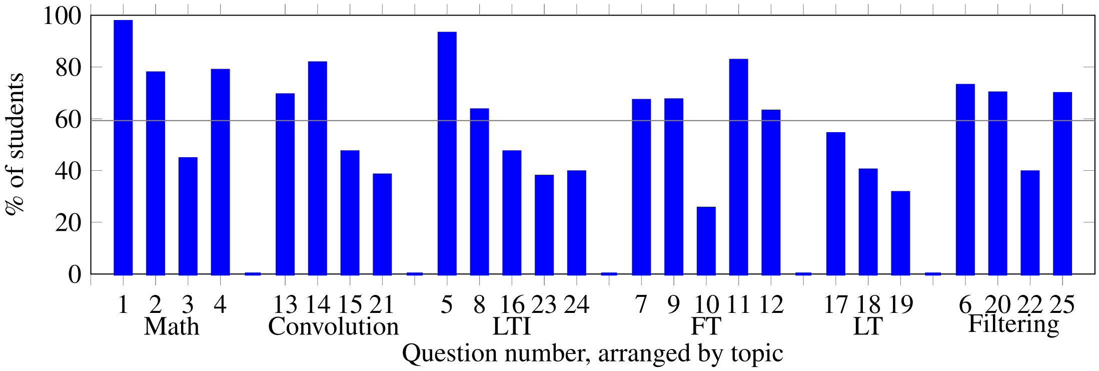


Figure 4.5: Percentage of senior UVA students who answered each SSCI question correctly.

than 50% of the students who get a question wrong), presents a hypothesis entering the think-aloud interviews, and discusses how the think-aloud data aligned with the hypothesis.

4.3.1 Linearity and Time Invariance

SSCI questions 5, 8 and 24 focus on LTI concepts. Most students answer Q5 and Q8 correctly (93% and 64% respectively) suggesting they can apply LTI concepts to relatively simple questions. Q5 asks students to recognize the output of an LTI system when the input is delayed; this question is easy in that it likely mimics the format of common homework problems. Q8 asks students to recognize a possible output of an LTI system when the input is a sinusoid; this format is likely less familiar to students, but a majority of students still select the correct answer. Neither Q5 nor Q8 has commonly selected incorrect answers.

In contrast, only 40% of students correctly answered Q24 and Q24 is the third most difficult SSCI question according to the IRT results. This question requires students to infer if a system could be linear and/or time invariant based on three input/output pairs. Looking at the four multiple-choice answers, 72% of students chose a response that is correct on the time invariance (TI) of the system while only 49% answered correctly regarding the linearity. Based on these data, our hypothesis entering the think-aloud interviews was:

(H-LTI) *Students have lower CU of linearity than TI and thus struggle to apply the linearity concept to a graphical problem with a novel format.*

This hypothesis was tested by Q24 on the interviews.

All think-aloud participants who did not use test-taking strategies (such as process of elimination) or explicitly say they were guessing used “what it is” concepts to answer Q24. Of the 12 think-aloud participants, ten checked for TI with correct reasoning. One participant guessed the answer, one confused TI for memoryless, and one (who later corrected their work) confused TI for causality.

In contrast, only half the participants reasoned correctly about linearity. The most common error (three participants) was thinking linearity meant the input/output pairs had to be proportional. Two participants recognized the system was not LTI, and used that to justify it being non-linear, suggesting they did not remember how to separately test linearity and TI. A couple of participants initially tried to reverse-engineer the system based on the input-output pairs, *e.g.*, one attempted to find the impulse response and then use that to determine if the system is LTI. This circular logic (assuming the system is LTI to understand what it does to then determine if it is LTI) suggests how accustomed students are to assuming LTI.

Overall, the think-aloud results support (H-LTI) that participants better understand TI in graphical form than linearity. A follow-up question for a future research project would be to determine whether students who do not properly check linearity from graphical input/output pairs recall the mathematical definition of linearity.

4.3.2 Convolution

Questions 13, 14, 15, 21, and 23 test convolution. However, Q21 requires reverse reasoning and Q23 requires synthesizing convolution and causality concepts, so neither provides a clear picture of convolution CU. Of the remaining three questions, Q14, which tests if students remember that convolution is commutative by reversing the roles of the input and impulse response, is the easiest according to the IRT results. The common incorrect answer (12% of participants) correctly identifies the shape of the requested signal, but incorrectly adds a time shift.

The other two convolution questions, Q13 and Q15, ask for the output of an LTI system given rectangular pulses as the input and impulse response, thus testing if students know convolution is the correct operation and if they can recognize a common graphical convolution. Looking at the problem statement, instructors likely see the questions as redundant[3], but the correct answer to Q13 only requires that students know the output is a trapezoid (none of the distractors are trapezoids) whereas all possible answers to Q15 are trapezoids or triangles with the same start and end times. To answer Q15 correctly, Q15 requires determining the maximum height and ramp-up slope/length of the output. The IRT results reflect that Q15 is more challenging: β_{15}=0.70 while β_{13}=-1.03. Fig. 4.6 demonstrates the key difference between the two questions.

The common incorrect answer to Q15 (33% of students) suggests that students do not recognize that a unit-height, wide rectangle has a maximum overlap of more than one units when convolved with a longer unit-height rectangle. *Wage et al.* [56] found the same common incorrect response.

[3] One instructor asked why both questions were included given that they (appeared to) test the same concepts. At the time, I did not have a good answer! In hindsight, I would have responded that Q15 is more challenging because of the answer choices and that Q15 tests if students know how to convolve two pulses both with width greater than one time unit.

Table 4.4: Summary of SSCI questions, the IRT estimates and corresponding standard errors, and the fraction of students answering correctly (μ_i). Highlighted questions are used in the think-aloud interviews and starred questions were used in [56].

Question	Concept	Discrimination α_i	Difficulty β_i	μ_i
Q5	The definition of time invariance.	N/A	N/A	0.93
Q1	The definition of frequency.	1.20 ± 0.48	-3.69±1.14	0.98
Q11	The FT is homogeneous.	0.72 ± 0.19	-2.42±0.56	0.83
Q14-RR	Convolution is commutative.	0.74 ± 0.17	-2.01±0.65	0.82
Q20	How adding a pole to a frequency response impacts the corresponding Bode plot.	0.40 ± 0.13	-2.01±0.65	0.70
Q2	Time-reversal in the signal domain.	1.01 ± 0.20	-1.39±0.24	0.78
Q6*	The interpretation of a magnitude and phase diagram for a filter.	0.85 ± 0.15	-1.39±0.24	0.73
Q4	How to find the difference of a signal and its time-shifted version.	1.70 ± 0.28	-1.21±0.14	0.79
Q13*	Convolution computes the output for an LTI system. Graphical convolution of rectangular pulses.	0.93 ± 0.16	-1.03±0.18	0.69
Q25*	Low pass filtering of windowed signals.	1.25 ± 0.19	-0.87±0.13	0.70
Q12	Convolution-multiplication duality and the FT of a cosine. Or, how multiplication with a carrier wave impacts the FT.	0.90 ± 0.16	-0.82±0.16	0.63
Q7	The definition of the Fourier series.	1.49 ± 0.22	-0.69±0.10	0.67
Q9*	Increasing the frequency of a signal in the time domain correspondingly increases the frequency in the FT domain.	1.95 ± 0.29	-0.66±0.09	0.67
Q8	Sinusoids are eigenfunctions of LTI systems.	1.33 ± 0.20	-0.60±0.10	0.64
Q17	How to interpret a pole-zero plot to determine a system's causality and stability.	0.71 ± 0.15	-0.29±0.15	0.54
Q16	How to determine if a system is causal based on its impulse response.	1.12 ± 0.16	-0.04±0.10	0.47
Q18	The relation between a system's pole-zero plot and its impulse response.	0.94 ± 0.16	0.45 ± 0.14	0.40
Q3	Recognize a time-reversed and shifted signal.	0.36 ± 0.17	0.48 ± 0.30	0.45
Q21-RR	Convolution-multiplication duality.	0.88 ± 0.14	0.51 ± 0.14	0.39
Q22-RR	Time-phase shift duality.	0.71 ± 0.14	0.51 ± 0.17	0.40
Q15*	Convolution computes the output for an LTI system. Graphical convolution of rectangular pulses.	0.28 ± 0.11	0.70 ± 0.43	0.47
Q19	The relation between a system's pole-zero plot and its frequency response.	0.86 ± 0.16	0.99 ± 0.20	0.31
Q24	Graphical interpretation of linearity and time invariance.	0.31 ± 0.12	1.28 ± 0.56	0.40
Q10-RR	Convolution-multiplication duality of the FT.	0.72 ± 0.15	1.43 ± 0.29	0.25
Q23	The relation between the impulse response of a system and whether the system is causal. Parallel and cascade connections of systems.	0.26 ± 0.12	1.83 ± 0.87	0.38

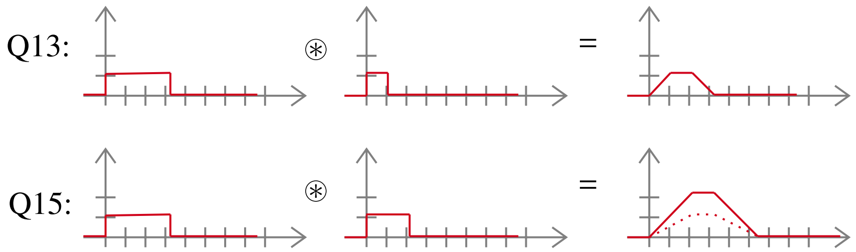


Figure 4.6: Example problems illustrating the main difference between question #13 (top) and #15 (bottom) on the SSCI. In both questions, students are given the input signal and impulse response (the two signals on the left side of the equality) and asked to select the plot of the output signal. The dotted line in the second row shows the common amplitude error made by students. These example problems do not use the exact input and impulse response from SSCI questions #13 and #15; the SSCI questions include a written prompt and the signals have different lengths and non-zero offsets.

They further observed that students used a memorized "trick" of adding start and end times to answer Q13, rather than using CU. Ref. [56] hypothesized that students over-generalize in-class examples, which primarily involve unit-width pulses. Our hypothesis entering the think-aloud interviews was:

(H-Conv) *Students have low CU of what input and impulse response features determine the output signal features,*

i.e., that the findings from [56] would generalize to our study population of seniors. This was tested with Q13 and Q15 on the think-aloud interviews.

On Q13 and Q15 of the think-aloud interviews, most (but not all) students recognized they needed to convolve the input and impulse response (the "why it matters" concept part of the question). Three did not mention convolution but recalled some of the procedure. One had forgotten convolution and guessed.

Next, students had to use CU or procedural knowledge to find the output of the graphical convolution. Two students attempted to use PK, with one of them successfully computing the convolutions for both Q13 and Q15. Ten students noted that they picked the only trapezoid on Q13; most of these used a "what it is" concept of convolution to recognize that the output needs to ramp up and down. Only one added start and end times to find the output duration; they said they discovered this "trick" on their own when previously solving a convolution problem and they demonstrated CU during a follow-up probe.

As in [56], the most common reasoning error in Q15 was in computing the maximum area of overlap: only four students correctly reasoned about this using "what it is" convolution concepts out of the seven students who considered the maximum value when answering. The other students either guessed or reasoned based on ramp-up length/slope. Three students explicitly recalled convolution involving multiplication and incorrectly said the maximum signal value was one since both inputs had unit amplitudes. In contrast, all eight students who reasoned about the ramp-up length or slope did so correctly. The way students talked suggested that they had CU of how the width of the input impacted these points. Thus, there is only partial support for (H-Conv)–students often did not understand what controlled the maximum value for the output signal, but they largely demonstrated CU of the start/end times and breakpoints and did not use memorized "tricks" as in [56].

4.3.3 Fourier Transform

Questions 7, 9, 10, 11, 12, and 22 on the SSCI emphasize FT concepts, with Q10 and Q22 requiring reverse reasoning. The common distractor on Q10 (56% of participants) corresponds to convolving the frequency response with itself (rather than multiplying it) when asked to convolve the time domain signal with itself. Considering the four remaining questions (Q7, Q9, Q11, and Q12), students responded correctly 67-83% of the time, suggesting a relatively high level of CU. Of these, Q12, which asks for the FT of $p(t)\cos(2w_o t)$ given a plot of $P(j\omega)$, was the only forward-reasoning FT question question with a common incorrect answer. The common incorrect answer (20% of participants) corresponds to convolving $P(j\omega)$ with itself.

Despite the high percentages, in interviews, [56] found students relied on tricks rather than CU to answer Q9. Q9 gives plots of a windowed sinusoid and its Fourier transform, then asks students to recognize the transform corresponding to the plot of a higher-frequency windowed sinusoid. This question thus tests if students understand the representational connection between time and frequency – one of the most critical "what it is" concepts in S&S.

Q9 and Q12 on the think-aloud tested the following hypotheses, respectively:

(H-FT1) *Students answer Q9 correctly without full CU.*

(H-FT2) *Students who answer Q12 correctly are more likely to recall why carrier waves are useful, rather than using the multiplication-convolution duality of the FT.*

Hypothesis (H-FT1) follows from [56] while (H-FT2) is based on my experiences.

For Q9, only six think-aloud students answered confidently with the correct reasoning. Five more answered correctly, but said they were unsure of their answer or used test taking strategies (three of whom selected different answers during the full SSCI). The one student who answered incorrectly was "thinking too fast" and confused high and low frequency; they otherwise had the correct reasoning and answered Q9 correctly on the full SSCI. Q9 was the first question on the think-aloud interview, so some of the lack of confidence might stem from students getting used to the think-aloud process. However, the fact that three students selected different answers when taking the full SSCI suggests that at least some participants were genuinely not sure of their answer. Although the seniors used "what it is" CU of the FT to answer Q9 rather than relying on memorized tricks as in [56], the data overall supports (H-FT1) as almost half of participants were unsure of their (correct) answer and many reasoned based on incomplete CU of the FT.

For Q12, there were an equal number of students (five) who used process of elimination as their primary strategy and who recalled, some only vaguely, that multiplying by a cosine acted as a carrier and centered the $P(j\omega)$ around $\pm 2\omega_o$. Four students tried to use the multiplication-convolution duality approach to answer Q12, but only one student recalled the FT of a cosine to answer that way (another student noted they would typically look it up in a table). Only one student picked the common distractor answer for Q12; they used test-taking strategies. Other students who used test-taking strategies often quickly eliminated other answers, suggesting this distractor may be common due to test taking strategies and not because students default to convolution. Whether students pick that answer because it is $P(j\omega)$ convolved with itself would be an interesting hypothesis for a future study.

The Q12 think-aloud results support (H-FT2) since students better recalled the "why it matters" concept of carriers, even if they no longer had full CU, than the multiplication-convolution duality and the FT of cosine. However, as with Q9, many students were able to answer Q12 correctly by process of elimination using partial knowledge. Also, almost the same number of students attempted the duality approach, but not knowing the FT of cosine made this approach less successful than the students who used the "why it matters" approach.

4.3.4 Filtering

Questions 6 and 25 on the SSCI focus on filtering. These questions were among the 10 easiest questions on the SSCI according to the IRT results and had some of the highest percentage of students responding correctly, 73% and 70% respectively of students responded correctly. There were no common incorrect answers. Using these questions, [56] found many students who responded correctly did not check if the signal was in a filter's passband and students assumed that at least one signal should pass through a filter, likely over-generalizing from examples in class that rarely (if ever) involve filtering a signal without a component in the passband. Similarly, many students did not check if the filter influenced the magnitude of the signal; they treated the filter as a "mask" that either fully passed or fully rejected the signal components. Based on these results, we hypothesized:

(H-Filt1) *Students do not check the passband for filters.*

(H-Filt2) *Students think of filters as a mask and do not check the filter magnitude.*

Q25 on the think-aloud interviews tested both hypothesis. Time permitting, after the student finished answering, I asked whether students' answers to Q25 would change if the frequency of the input signal were increased (by a factor that pushed it outside the pass band) or if the amplitude of the filter were doubled.

Tab. 2.1 indicates that there are multiple approaches to Q25 based on different concepts. Most of the participants recognized the question as a standard filtering problem, with two students explicitly noting that the frequency response corresponded to a low pass filter. These students were able to quickly identify that the output would consist of only the low frequency pulse using either the "why it matters" concept behind low pass filters or by applying the definition of a low pass filter (a "what it is" concept). Two students reasoned through the problem by multiplying the FT of the input signal with the system's frequency response to find the FT of the output then connecting the resulting FT to the low frequency pulse in the time domain. These students used the "what it is" concept for a frequency response rather than CU of filtering to arrive at the correct answer. Overall, most students demonstrated CU in some form on the think-aloud for Q25. Only two students had significant conceptual errors: one related zero frequency to zero time and the other thought the symmetry of the filter's frequency response implied the output time signal should be symmetric.

Unlike the results presented in [56], nine participants in this study checked the passband when initially answering (often graphically), an additional student checked the passband when asked a follow-on question about how the answer would change if the frequency of the input pulses was increased, and one never checked the passband but did recognize that a higher frequency input would result in zero output. Only two students checked the magnitude of the filter on their first pass, but, of the five students that we asked a follow-up question, all correctly accounted for the impact of a non-unit magnitude filter. Considering how many students used process of elimination for the other think-aloud questions, students were thorough when answering Q25 and the think-aloud data does not support (H-Filt1) nor (H-Filt2).

4.4 Discussion

This study was largely an extension of the CU of S&S work by Wage, Buck, Nelson, *et al.* [56] to a population of electrical and computer engineering senior students at UVA and UM. Unlike the participants in [56], the senior students in this study are generally one or more semesters removed from their primary S&S course. Seniors may forget concepts over time, or they may gain CU as they have time to process concepts and they see concepts reinforced in upper-level courses. Tab. 4.6 summarizes the main findings from this study and the literature.

Table 4.6: Summary of the hypotheses, main findings, how they compare with findings from the literature, and questions for future work.

	Hypotheses	Findings	Connection to the literature	Questions for future work
LTI	(H-LTI): Students have lower CU of linearity than TI and thus struggle to apply the linearity concept to a graphical problem with a novel format.	Students correctly predicted the output of LTI systems. Supporting (H-LTI): Students better understood TI than linearity, and confused linearity with proportionality.	Ref. [71]: Students can answer LTI questions without full CU. Students tend to assume LTI.	Do students who struggle with the graphical representation of linearity recall the formal definition? If yes, can they apply the definition?
Convolution	(H-Conv): Students have low CU of what input and impulse response features determine the output signal features.	SSCI scores show mixed CU. Students recall that convolving two rectangles yields a trapezoid. Mixed support for (H-Conv): Students understood how to find breakpoints in the output signal but fewer understood what determines the maximum amplitude. Some students confused convolution and multiplication.	Ref. [56]: students struggled to determine both breakpoints and the maximum amplitude when both input signals had an amplitude of one.	Does demonstrating CU on the SSCI questions transfer to novel convolution questions that involve shapes other than rectangles?
FT	(H-FT1): Students answer Q9 correctly without full CU. (H-FT2): Students who answer Q12 correctly recall why carrier waves are useful, rather than using the multiplication-convolution duality of the FT.	SSCI scores suggest relatively high CU of FT. Supporting (H-FT1): Students are not confident about relating time domain and frequency domain representations in Q9. Students do not recall the FT of a cosine. Tentative support for (H-FT2): An equal number of students attempted the “what it is” and “why it matters” approach to Q12. The students using the “why it matters” approach answered correctly more often.	Ref. [56] found students answered Q9 correctly without full CU. Ref. [51] found students struggled to write the FT of a signal comprised of a sum of sinusoids.	Is the observed lack of confidence due to Q9 being the first think-aloud question? Would students recognize the FT of a cosine if it were a stand-alone question?
Filtering	(H-Filt1): Students do not check the passband for filters. (H-Filt2): Students think of filters as a mask and do not check the filter magnitude.	SSCI scores suggest high CU of filtering and no common errors. Opposing (H-Filt1): Students mostly checked the passband of a low-pass filter. Opposing (H-Filt2): Most students did not explicitly check the filter magnitude during their first response, but did when asked a follow-up question.	Ref. [56]: Students did not check the passband or filter magnitude. Ref. [49]: Students think of a filter as either completely passing or blocking a signal component.	How do the results compare if using an initial think-aloud question that requires students check the filter passband and magnitude?

Seniors averaged 59% on the SSCI, which is in the middle of the 50-70% range of post-test scores reported in previous literature for students at the end of a S&S course [74], [81], [82]. The fraction of students answering each SSCI question correctly varied considerably, from 25% to 98%. Tab. 4.4 describes the SSCI questions in order of difficulty for our population of students. For comparison, [72] ranked list of topics that students *think* are hard, with FT and convolution ranking highest. Interestingly, [72] found that system properties (such as LTI) are more frequently described as hard by instructors, not students.

Overall, our results agree with [56] in that we found "many students arrived at correct answers despite incorrect and incomplete understanding." For example, not all of the seniors recalled convolution as the correct operation to find the output of an LTI system given the impulse response and input in Q13 and Q15. Nor were they all able to confidently connect high and low frequency pulses to their FT representations in Q9; supporting hypothesis (H-FT1). For convolution concepts, our results are also similar to [56], in that students struggled to determine the maximum area of overlap and mistakenly thought that the unit amplitude of the input determined the maximum output without considering the width of the pulse, partially supporting hypothesis (H-Conv).

However, our data differs in many ways from [56]. First, participants remembered and showed CU for how to find the breakpoints in a convolution problem, thus partially opposing (H-Conv). Students also almost always checked the passband of the filter in Q25 and did not think of the filter as a mask, arguing against hypotheses (H-Filt1) and (H-Filt2). Further, the think-aloud data does not suggest many errors stemmed from over-generalizing in-class examples as in [56]. For Q15, errors tended to come from students whose memory of convolution had faded (or never fully formed) and who reasoned that one times one is one because they recalled that convolution involved multiplication. Students used similar reasonings and test taking strategies on many other think-aloud questions. This difference between our results and those in [56] makes sense for the two different study populations as seniors are more removed from specific in-class examples.

The fact that many students correctly answered questions despite incomplete CU highlights the inherent limit of multiple-choice CIs. However, we noted for think-aloud participants that higher SSCI scores tended to mean higher CU–guessing and test taking strategies are limited. Thus, one should not conclude that students understand a concept based on a SSCI answer, but the performance of a large group of students can still paint a picture. Recent progress in machine learning for analyzing textual data [66] is promising for analyzing CU with more detail for large student populations.

Using think-aloud interviews, this study also investigated which levels of concepts, according to the framework in Fig. 2.1, students used. Most SSCI questions lend themselves toward "what it is" concepts. For most of the think-aloud questions, students who used CU (instead of test-taking strategies or guessing) used "what it is" concepts. Very few students used PK in the interviews.

Q12, which tested FT concepts, was particularly interesting as it allows for a "what it is" or a "why it matters" approach, thus allowing us to test (H-FT2). Most think-aloud participants who answered Q12 correctly used the "why it matters" approach, though some only partially recalled the concept. Students who tried to use a PK-based approach on Q12 got stuck because they did not recall the FT of cosine, which the students talked about as a memorized piece of information. The FT of a cosine is itself something students could figure out with CU of FTs, as tested in Q9.

The proposed definition of CU in Section 2.1.2 also requires that students can reason about, relate, or apply concepts. Students not thinking through the FT of a cosine in Q12 and their overall poor performance on reverse-reasoning questions suggests general difficulty in reasoning about and applying concepts, as was observed in previous studies [56], [75], [78]. As a counter-example to this finding, many think-aloud participants reasoned through Q24, though some used an incorrect concept in the process and thus selected the wrong answer. One student (who had high CU) was even excited about the novel format and said they "actually really like this problem." As another example of students showing the ability to reason, a different think-aloud participant correctly reasoned through the output start time on Q13 based on their CU of what an impulse response represents, showing a high level of reasoning and CU of impulse responses (although a low CU of convolution).

Finally, [56] did not separately consider LTI concepts, but our LTI results connect with other results from the literature. The IRT and think-aloud results suggest students can generally answer questions regarding LTI systems, but may not truly understand what LTI, and especially linearity, means, supporting hypothesis (H-LTI). These findings agree with those of Nasr, Hall, and Garik [71], who found that students predicted the correct output to an LTI system given an input/output pair and a new input without understanding how the answer is grounded in LTI properties. Further, for a simple, theoretical problem, students automatically assumed the system was LTI to determine the output [71]. One possible explanation is that students see so many examples of LTI systems in their classes that they can manipulate LTI systems without having full CU. This explanation is supported by some students using LTI assumptions to analyze a system to determine if that system is LTI.

Another possible explanation for students' struggle with linearity from the think-aloud data is that some students were confused by the term "linear." Colloquially, a linear system would be one such that that only scales inputs. Previous studies [75], [78] found students had similar confusions with aspects of convolution when words did not match how they are used in engineering or were used imprecisely. Wage, Buck, and Hjalmarson [49] also found some students were confused by the term "filter" because common uses of the word "filter," *e.g.*, an air filter, tend to evoke the meaning of completely passing or blocking certain components.

4.4.1 Future work

This chapter concentrates on measuring CU so instructors can see which concepts students understand and which they do not. However, the large standard deviation of 17% suggests students vary quite a bit in their CU. A natural next question is thus what factors, such as student motivation or choice of upper-level elective courses, influence CU and how can instructors and curriculum designers help increase CU. Chapter 5 addresses this question.

This study measures CU at a single point in time in two universities. Future work should investigate CU over time by sampling the same population (preferably using paired samples) at multiple points in the curriculum across a variety of institutional contexts. Such a study would be able to investigate important questions, such as if SSCI scores tend to increase or decrease for some sub-population(s).

Another avenue for future work is to develop a new instrument that measures CU according to the proposed definition and according to what concepts are most important for students to understand. Questions that ask students to select the correct tool to analyze a certain problem may better target "why it matters" concepts than current SSCI questions. Future studies should also expand the think-aloud interviews to better understand how seniors approach questions testing the Laplace transform and causality concepts.

Finally, Tab. 4.6 identifies questions for future work for each of the concepts included in this study. Other questions arising from the SSCI results that we did not get to examine in the think-aloud interviews were what was the reasoning was behind the common incorrect answer on Q12, why do students struggle with the reverse reasoning questions, and how students approach the background mathematics and LT questions.

4.5 Conclusion

Even if students forget mathematical details over time, CU provides mental scaffolding for deciding a method to approach a problem and for filling in forgotten procedural steps. Many previous studies consider CU during a S&S course, but few investigate students' CU of S&S concepts years after a S&S course. This study found an average SSCI score of 59% for seniors, which is in the middle of the range of previously reported SSCI scores of students after a S&S course. Chapter 6 compares the results of measuring CU from this study of senior students to the results of studying students in S&S from Chapter 3.

The approach seniors took in think-aloud interviews was similar to the S&S students from [56] in many ways: students made the same common error when finding the maximum overlap in a graphical convolution problem and students were not confident in their answers on basic FT concept questions. However, unlike [56], we found seniors checked the passband for filters and did not describe filters as masks. Further, the seniors did not use memorized tricks. Instead, many used test taking strategies and partial recollections if they did not have full CU. Some even mentioned that taking the SSCI helped them recall certain concepts.

This study also considered the proposed definition for engineering CU from Section 2.1.1 with three levels of concepts, "what it is," "why it matters," and "how it works" concepts, and looked at how the SSCI questions related to this definition. Most SSCI questions lend themselves to "what it is" approaches, but students tended to use the "why it matters" approach on the one think-aloud question that could be approached two primary ways, suggesting they might better remember this type of concept.

For think-aloud participants who demonstrated a high CU on the two convolution questions, we asked them to predict the most common incorrect answer. Half said the answer corresponding to the wrong maximum amplitude. Half said the answer with the wrong ramp-up length/slope. This sample is small and anecdotal, but aligns with the my experiences talking with colleagues: once you reach CU, it is difficult to recall conceptual challenges and to predict what others might struggle with. The results of this study will hopefully help instructors better understand seniors' CU of specific concepts in SS and in turn address the most challenging S&S concepts.

CHAPTER 5

RQ#3: What Factors Influence Conceptual Understanding?

Chapter 4 measured CU of senior students in electrical and computer engineering. One additional line of research identified in that chapter is investigating why senior students might differ in their CU. This chapter addresses that suggestion using a qualitative approach

The aim of CU studies is to increase students' CU and improve engineering education [26]. For example, Hake [69] and Wage *et al.* [1] show how active learning techniques correlate with larger gains in students' CU. These previous CU studies tend to assess students' CU, categorize concepts based on difficulty, and suggest or analyze strategies for making the challenging concepts more attainable, typically in the context of a specific course. Our chapter takes a different, complementary approach: we ask what instructional practices, *over the course of an undergraduate degree*, help the CU of senior students.

Our research question is: **What instructional factors influence CU of S&S for senior students?** The following section provides more background on factors that influence learning. Section 5.2 then describes the methodology for the study, which includes qualitative interviews with students, practicing engineers, and EE faculty who had previous taught S&S to explore what factors may influence CU. Section 5.3 presents the qualitative results, organized by instructional strategy. We conclude with a discussion in Section 5.4 and 5.5 of how the identified factors compare to theories in the literature and a list of concrete strategies for instructors.

The study in this chapter is presented in the following paper [6]:

> C. Crockett, H. C. Powell, and C. J. Finelli, "Factors influencing conceptual understanding of signals and systems of senior engineering students," Submitted to: *European Journal of Engineering Education*, 2022

5.1 Background: Factors Influencing Understanding

There are many theories on factors that influence learning. One empirically validated model for learning is the Model of Educational Productivity (MoEP) [88], which Chapter 3 used as a basis for studying variables that predict CU at the end of a S&S course. Building on a synthesis of national science achievement test and a survey given to 3,049 17-year-olds as part of the National Assessment of Educational Progress, [88] found nine significant factors on test scores. The nine factors are commonly described in three groups: the first three, age/development, ability, and motivation, relate to the student; the next two, quantity and quality of instruction, are dependent on instruction; and the final four factors, exposure to mass media and home, classroom, and peer environment, are environmental variables. Section 3.1 provides further background on the MoEP.

Many prominent theories on learning focus on instructional strategies that could be used to foster student learning; these would generally fall under the instructional quality factor of the MoEP. For example, previous results show that active learning improves student learning [119] and improves CU specifically [1], [69]. Bloom [120] found that the individualized, high-quality instruction students receive from a tutor leads to tutored students learning two standard deviations more material than the average student taught using traditional lecture-based instruction. Many of the instructional strategies to close this "two sigma" gap fall under the category of active learning strategies [120]. Considering instructional strategies that are more targeted for S&S courses, [99] suggests the graphical interface of LabView helps students gain CU more than the textual interface of Matlab and [51, p. 204] recommends that instructors spend more time telling students where each piece fits into the larger picture, make the lab complement the lecture and use consistent terminology, lead group discussions for robust problematic reasonings, and teach concepts in multiple stages to cover them from different angles.

Considering specifically CU, the Cognitive Reconstruction of Knowledge Model (CRKM) [121] proposes that the depth of cognitive engagement determines how much conceptual change a student will undergo. The degree of cognitive engagement is in turn influenced by the student characteristics and message (or lesson) characteristics. The influential student characteristics are: students' motivation and students' strength, coherence, and level of commitment to prior knowledge. Motivation is defined by students' dissatisfaction with their current understanding, personal relevance of the new message, and social context (similar to the peer environment factor in the MoEP). The influential message characteristics measure how comprehensible, coherent, plausible, and rhetorically compelling the message is. The CRKM was created by combining elements from the cognitive psychology, science education, and social psychology literatures [121]. The CRKM recognizes that conceptual change is an iterative process and the dynamic between the student and message characteristics change over time. For example, the model does not consider motivation to be an inherent, unchangeable student characteristic but rather a trait that is defined by the interaction between the student and the message which can change as the students learns more about the message.

Taasoobshirazi, Heddy, Bailey, *et al.* [122] tested the CRKM as a model for conceptual change in physics. The authors used the FCI to measure CU and items from the following five validated survey instruments to measure the student variables:

1. The physics motivation questionnaire [123]. This measures intrinsic motivation, extrinsic motivation, relevant of learning to personal goals, self-determination, and self-efficacy. All questions are specific to motivation for learning physics.

2. The approaches to learning instrument [124]. This measures deep cognitive engagement.

3. The achievement goal questionnaire [125]. This measures how much students tend toward mastery-goal orientation (the goal of learning to reach understanding) versus performance-goal orientation (the goal of learning to reach a milestone such as a good grade).

4. The need for cognition scale [126]. This scale measures a student's tendency to seek out and enjoy cognitively challenging problems. One example item is "I find satisfaction in deliberating hard and for long hours" [126].

5. The achievement emotions questionnaire [127]. This measured enjoyment, boredom, and anxiety.

Using structural equation modeling to analyze the data ($n = 117$), [122] found that, related to variable five, enjoyment was the only emotion that played a significant role in the model (anxiety and boredom did not). The motivation and goal orientation variables, items one and three in the list above, also played significant roles in the overall model, while need for cognition and engagement, variables two and four, did not. However, many of the variables that played a significant role were only indirectly linked to conceptual change through course grade. Further, the authors note that their findings, especially that engagement was not significant, may be due to the definition and measurement of the variables. The recommendations in [122] are that instructors should focus their efforts on increasing student enjoyment, increasing student motivation, and fostering a combination of a mastery and performance goal orientation.

In an engineering context, Felder and Brent [128] offer eight strategies for encouraging a deep learning approach, which the authors define very similarly to CU or a mastery goal orientation: deep learning requires students to "not simply rely on memorization of course material but focus instead on understanding it." The strategies are:

(FB1) make sure students are interested in and prepared for the material, (FB2) state expectations and provide clear feedback,

(FB3) structure grades to encourage deep learning over procedural knowledge,

(FB4) encourage students to be actively engaged in learning over the long-term,

(FB5) provide students with opportunities to influence the course content and learning methods,

(FB6) show care for the students learning,

(FB7) keep the workload reasonable, and

(FB8) encouraging a deep learning approach in one course will encourage a similar approach in future courses.

These points generally align with the focus on student motivation in the CRKM [121].

There are fewer results that consider CU over a longer period of time than a single course. Greene [82] and McKell and Danowitz [129] offer initial, largely anecdotal, evidence that active learning and standards-based grading may help with long-term retention in S&S. Based on interviews with students, postgraduate students, and academics, [2] suggests that time, along with self-regulated learning, is critical for processing concepts. Male and Baillie [77] concentrates on "threshold concepts," which are ideas that transform a way a student thinks. These concepts are often gateway concepts because students cannot understand later concepts without first understanding the given threshold concept. Male and Baillie [77] identifies time-frequency transformation and discretization as two threshold concepts in S&S. In addition to self-study and time, the interviewees in [77] cited using a concept for their work or in their teaching as what forced them to overcome a threshold.

Similar to self-studying a concept over time or applying a concept at work, previous results suggest that students who take related upper-level electives are more likely to develop CU in S&S. Although the small sample sizes make it hard to draw strong conclusions, the SSCI data in [81] shows a trend of scores dropping over time, suggesting students forget concepts after a S&S course. However, scores increased if students saw the same material multiple times, suggesting that the concepts are reinforced and solidified in upper-level courses. Likewise, [51] found that students who continued to take SS-related courses better understood the relation between convolution and multiplication and the definition of time-invariance than their counterparts who only took an introductory S&S course. Although both studies show a general increase in CU, upper-level students may continue to struggle with certain topics (per the discussion in Section 2.3).

Section 5.4 discusses how our results compare to the presented theories on factors that influence CU.

5.2 Methods

To get a variety of perspectives on what helped/hindered students from gaining CU, we interviewed faculty (F), practicing engineers (PE), undergraduate (UG), and graduate (G) students. Participants were purposefully selected to include EE and non-EE students, international and domestic students, students who focused on SS-related tracks and those who focused on other areas of EE, and a range of technical focus for practicing engineers. Tab. 5.1 provides basic information about the participants. To maintain participant confidentiality, we include limited demographic and degree information. All discussions in Fall 2019 (F ’19) took place in-person while those in Spring/Summer 2020 (S ’20) took place online with video conferencing due to COVID-19 restrictions.

Table 5.1: Summary of interview and focus group participants.

Category	N (time)	Description
(F) EE faculty	1 at UM (F ’19) 1 at UVA (S ’20)	Both taught S&S and related courses.
(UG) Undergrad. students	4 at UM (F ’19) 4 at UVA (S ’20)	2 male, 2 female. 3 EE majors, 1 computer science major. 1 male, 3 female. All EE or computer engineering majors.
(G) Graduate students	5 at UM (F ’19)	All male. 3 in signal processing related research, 1 in controls, 1 in biomedical imaging research. 3 with international undergraduate degrees.
(PE) Practicing engineers	4 (S ’20)	2 male, 2 female. 1 early career, 1 mid-career (5-8 years), and 2 in middle to upper management positions.

We used group interviews for undergraduate and graduate students because we expected the existing camaraderie to help students compare, contrast, and relate to each other’s experiences [117, Ch. 21]. All interviews were one hour long and students were either given free food (for in-person interviews) or $10 Amazon gift cards (for remote interviews) to thank them for participating. The undergraduate students provided the most immediate feedback on the undergraduate experience while the doctoral students were more removed from their initial S&S class, meaning that they likely forgot details of how they felt the first time the tried to learn certain topics. However, they had the benefit of hindsight; they could reflect on how well they thought they understood topics as undergraduates compared to their current understanding.

For faculty and practicing engineers, we used interviews since these participants had very different experiences from each other and to decrease the time commitment for each participant. Both faculty and practicing engineers provided a longer-term perspective. They were able to reflect on the undergraduate education, what they thought they knew then, and what they have since realized they knew. The faculty members also talked about what concepts their students struggled with or learned easily, and how they thought any teaching changes impacted students' CU. The faculty interviews were an hour, with 15 minutes at the end reserved for the participant to ask questions about the study. The practicing engineer interviews were 30 minutes, with five minutes reserved for participants to ask questions at the end. We did not provide any incentive for interview participants, so reserving time to answer their questions was a small gesture of gratitude[1].

All interviews were semi-structured: we developed a protocol to guide the overall session but allowed for different follow-up questions based on participants' responses. I followed best practices for interviews as summarized in [116], [117], [130] such as mixing descriptive and structural questions [131], crediting participants who brought up ideas when asking probing/follow-up questions [132], and measured use of humor and silence to make participants comfortable and allow them to expand on ideas [116]. The full protocols are included in Appendix A.

The general structure for all protocols was:

1. Explain the interview plan and review the consent form. The first step follows good human research practices and helps participants feel comfortable.

2. Ask everyone to introduce themselves. This promoted interaction and allowed everyone to talk at least once early on in the discussion [132]. I introduced myself first to set an expectation for how much everyone will talk and to build rapport with the participants by acknowledging that there are concepts in S&S that I did not learn during my course [132].

3. Describe CU and provide a list of example concepts.

4. Ask what factors helped/hindered learning concepts during participants' S&S course. Do not mention any example factors initially to allow participants to use their own words. After participants have a chance to answer, probe for hypothesized factors if not mentioned.

5. Ask the second focus question on impacting factors after S&S. Start by leaving the question open to interpretation but then probe for specific possible factors.

6. Ask a question to allow participants to reflect on the conversation and speak one more time. Ask students to summarize one thing that most helped and most hindered their CU and ask practicing engineers and faculty members what they would change about their undergraduate S&S education.

[1] Most participants took the entire time to ask questions and some requested to continue this conversation past our scheduled time. They were largely interested in what engineering education research is, how it fits into an EE graduate degree, and the current CU findings.

7. Thank the participants, reiterate that they were free to reach out, and ask for any last thoughts. For the practicing engineers and faculty members, reserve time at the end for them to ask about the study.

For the main questions (4 and 5), I allowed participants to interpret and answer the question in their own words before asking follow-up probes based on previous responses, factors from the MoEP, and preliminary analysis of previous interviews. For example, the graphical depiction of convolution as two overlapping, sliding rectangles came up in our initial student interviews, and I specifically asked the faculty members for their viewpoints on that method of teaching convolution in the interviews.

After transcribing the interviews, we coded the transcripts using the constant comparative method. This involves iterating between coding transcripts (attaching a label to a segment of the interview), memoing and reflection, and refining the codebook until the data and code align [117, Ch. 33]. After each of three major iterations, the research team met to discuss code definitions, agree on the coding of sampled quotes, and reflect on emerging themes. The first round of qualitative codes (completed between the Fall 2019 and Spring/Summer 2020 data interviews) included in vivo codes (from the data) and hypothesis codes (generated before analysis based on the MoEP); these codes types are described further in [118]. Another researcher coded two of the transcripts and helped refine code definitions for the first round. Later iterations refined these codes and added new in vivo codes based on additional interview data.

The final codebook was organized into the following five categories, each containing three to seven codes: concepts (Fourier transform, Laplace transform, convolution, linear and time invariant, filtering, discrete time, math), class components (class environment, grades, homework, lab, visuals), instructional quality and quantity (quantity, repetition, pace, and style), interest/motivation (purpose, motivate, abstract), outcomes and reactions (ability, easy, false confidence, familiarity, hate, importance, procedural), and outside influences (peers, work, workload). Tab. 5.3 defines example codes and provides an example quote, selected to help clarify the code definition and to represent a diverse sample of participants. Tab. B.1 in Appendix B shows the full codebook with example quotes.

5.3 Results

Engineers' views of CU is not a research question of this study, but participants' views of CU shape our results. At the start of each interview, I briefly contrasted CU with procedural knowledge. In one interview, I said that "a concept might be that the Fourier transform goes between signal space and frequency space. So, [I am] more interested in, 'do students understand that' and less

Table 5.3: Sample codes for analyzing the interviews.

Code	Description	Example quote
Concepts: **FT**	Fourier Transform, Fourier Series, and the frequency domain.	"The FT is pretty intuitive in that, at least for me, when there's things like music visualizers, that is exactly showing the spectrum of your sound." (G)
Class components: **hw**	Homework problems.	"[If] the questions are designed in such a way that they test your understanding, maybe that helps more. But then our homeworks are also similar, it's very, very procedural." (G)
Instructional Quantity: **pace**	The pace of a SS course and constraints on how much material must be covered.	"I'm afraid in the labs, they're so time-pressured, that they're going through the, you know, the steps, but they don't really have enough time to really sit down and figure out why they're doing it all." (F)
Interest: **purpose**	Seeing or wanting to see the big picture. Asking why something is the way it is.	"I loved it because it really connected. It was an engineering course.. It was, it was the whole system. Like you had to think big picture." (PE)
Outcomes: **familiarity**	Knowing a concept exists and how to find out more about it if needed.	"I understand it worse now but I think, I think it would come back, it wouldn't be too hard for it to come back to me if I like reviewed the material" (UG)

interested in 'can they do the mathematics to take the Fourier integral."' Other than this short example, participants were allowed to construct their own meanings of CU.

Most participants discussed CU as seeing "the bigger picture" (all types of participants used this phrasing) or *why* something was important. The "big picture" refers to intuition behind what a procedure does; some participants called this "translating" the mathematical formulas and procedures. Many participants continued to contrast CU with procedural knowledge and agreed with the literature's claim that procedures dominates CU in undergraduate engineering courses. As one graduate student said, "I was just like plugging in formulas you know. And I feel like it's very easy to do that. Um, yeah. And to like actually miss like the bigger, more salient points" (G).

Another commonality in participants' view of CU was being sufficiently familiar with a concept to know when they needed to use it and being able to re-learn the details on their own. For example, participants would say "I can pick it up pretty quickly" (UG) or "if I looked it up, I could do it" (G) or they would talk about something as being an "easy Google search" (PE). These comments were all in the context of participants recognizing they do not have a full understanding, but they have enough experience to know how to find the information that they need.

These views of CU align most closely with the "what it is" and "why it matters" aspects of the proposed CU definition in Section 2.1.1. Taken together, "what it is" and "why it matters" concepts help participants to select a concept for a problem. A few participants discussed the "how it works" aspect of CU. For example, a graduate student pushed back on the intuition-based view that the other students were using:

> people want to find the shortcut of intuition... [there's] a tendency for me that 'oh, I want to skip like three years of training to understand what it is' and then I think that's when there's actually like really conceptual breakdown because I just think for certain things there's just not really that easy of shortcut... certain things, like music, maybe give you a sense of it. But I don't think that that's necessarily like what Fourier transform and other things like, if you want to understand it, I don't think there's necessarily [a] short cut. (G)

This graduate student argued that there is no "shortcut" to CU–a student needs to fully understand the mathematics before reaching an expert-level of CU.

The following sections discuss the themes around influential instructional variables related to lectures, hands-on activities, coursework, and repetition and upper-level courses that emerged during the constant comparative data analysis. After presenting the evidence, Tab. 5.5 summarizes these themes.

5.3.1 Lectures

Many participants credited lectures with providing an initial level of CU. The following subsections describe two primary mechanisms.

5.3.1.1 Presenting CU Along-side Math

A reoccurring point in the interviews was the benefit of instructors providing intuition for mathematical concepts. A graduate student summarized the point succinctly: "a lot of those things, I guess, have intuitive explanations and as long as they're provided to you at the time, you can at least understand it intuitively" (G). For many, this intuition came in the form of analogies to every-day experiences with music.

Other participants had the opposite experience: they remembered topics being presented only as formulas. As a specific example, a graduate students recalls that the "Fourier Transform itself was like a big, scary thing. It was presented as a formula. Which made no sense" (G). But the student later took an online course and

> that professor actually gives a very quick and nice insight into [the] Fourier Transform if you read his book. That's when I started understanding FT and it started making sense. But before that it was presented as a formula. (G)

Other participants (including multiple students, a practicing engineer, and a faculty member) had similar experiences of seeing topics, especially convolution or the Fourier transform, in S&S only as formulas and procedures.

Participants had a range of mathematical abilities entering their S&S courses. Some had taken multiple math courses and were confident in their ability to perform calculations while others admitted struggling in earlier math courses and with the math in SS. Seeing these formulas was not sufficient for students to gain intuition, even when a student understood the heavy mathematics in SS. For example, one of the practicing engineers said he was "fine with math and with my background" (PE) but that he did not understand the purpose behind the math procedures:

> I think for me, um, I mean, like the FFTs[2], just like [the] integral, none of that was too tricky. I think for me it was mostly like, when do I apply these concepts... As far as the signals and systems, I think that I was fine with math and with my background. (PE)

The student group interviews similarly questioned the "why" behind convolution and LTI concepts even when they understood the procedures. A graduate student captured the tone of multiple comments regarding convolution in saying

> I had people like draw things on like pieces of transparent paper and like rub them across each other and I was like 'okay, I mean I guess I see what you're doing, but that's not a very good explanation.' And I knew how to do it but, yeah, I'd say that's it. (G)

If they were not provided with a bigger picture view, students did not form any intuition from mathematical expression on their own during their undergraduate courses.

For students who did not feel mathematically prepared for SS, the mathematical emphasis was even more challenging. Students noted that the "intuition is completely lost in the steps of how to do it" (G) and that they tended to "get lost in the [convolution] integral" (UG).

This is not an argument against a mathematically rigorous presentation of S&S material; many students talked favorably about math-heavy courses (especially upper-level courses; see Section 5.3.4) when the material was presented alongside CU. Participants of varying math abilities noted a few examples of how instructors presented CU along-side procedures. In contrast to the graduate student above, many students appreciated specific visuals that helped them understand concepts like convolution. One student wished instructors would talk at a higher-level about the impact of signal features on a procedure, such as how the areas of signals impacted the result of a convolution. Another undergraduate explained that an instructor would

[2] Fast Fourier Transform.

> translate the math in a way to actually understand and read the math and what the mathematical definition of the things were really trying to say... I actually understood things ridiculously well and I just hadn't thought to actually consider it from that perspective before. (UG)

Math is a foreign language and one in which students are less fluent than instructors [133] so students may need more frequent translations than instructors expect; one undergraduate student appreciated when the instructor made a list of "all the big picture concepts" at the end of every lecture. By changing the student's perspectives through visuals, higher-level discussions of procedures, and (repeated) translation, these instructors helped students gain intuition from math expressions that were previously perceived only as procedures.

5.3.1.2 Emphasizing Purpose and Connections

Both faculty members mentioned they aimed to promote CU by explaining the importance of concepts and connecting concepts to one another, *e.g.*, they built-up the convolution formula through LTI principles in SS, following the progression of major S&S textbooks, *e.g.*, [34], [35]. This progression is a key reason why LTI concepts are so important – they allow engineers to model systems with (relatively) easy input-output relations. However, students consistently failed to see the importance of LTI concepts:

> it really just seemed like a checkmark thing of like, oh, 'Can you do this?' Like, 'it's a thing. Make sure it's a thing.' ...Rather than why and how. But, like, what's its use? (UG)

A strategy suggested by multiple participants for emphasizing the importance of concepts is through contrasting examples. The students who later came to appreciate the importance of LTI concepts did so through exposure to non-LTI systems in elective courses (typically control systems courses) or working in industry. For example, a practicing engineer noted that during classes, "all we dealt with was linear time invariant systems, [so] I didn't really understand what the significance of that was" (PE) until working with nonlinear systems at work. Similarly, an undergraduate with research experience noted

> nonlinear circuits is a big part of EE that are never, like, we weren't exposed to at all. So not being exposed to it makes it hard to like really appreciate linear circuits and what they can do for you. And then that idea extends kind of throughout the entire curriculum, even in diff-EQ[3] and whatnot, it's all linear systems of differential equations. But when you get out into the real world or you start doing research at like

[3] A common abbreviation for differential equations.

> the Masters or PhD level, you quickly realize that, no, life isn't linear. And then you're suddenly like ill-prepared for the differential equations and the circuits that come.
>
> (UG)

Instructors have this larger perspective from "the real world" and are aware that non-LTI systems are common and generally much more difficult to work with. Despite instructors stating this comparison, without their own experience, students are unlikely to appreciate the idea that LTI systems are easier to work with, especially when presented with the challenging convolution formula.

A similar strategy that helped some students (and that others requested) was connecting material. Students appreciated when instructors explained the relation between different chapters in the course, such as the Fourier series and Fourier transform, as part of a story. Students likewise appreciated connection to real-life, which could be as simple as using "physically possible" example problems (PE) or using realistic units to "make things a little more practical" (F).

5.3.2 Hands-on Activities

Lecture alone is not enough for students to gain full CU. As a faculty member said, students can "hear stuff in lecture, maybe some of it sticks, but 'til you have to solve something, and as you know, until you teach something, you really don't know it" (F). This section turns to hands-on methods of learning that complement lecture strategies and allow students to construct their own knowledge. The subsections are roughly ordered by level of student involvement.

5.3.2.1 Interactive Simulations

Many participants mentioned a virtual platform where they could "play around with" (UG) a simulated system. For example, participants said they liked being able to "change parameters and see what happened" (PE) or "do something to an object on the computer and then like it changes and you see, like, see or hear, how it changed" (G). Specific examples of interactive simulations included multiple graphical user interfaces (GUIs) in MATLAB, the Fourier series animation tool from Desmos[4], and MultiSim simulations.

Interactive simulations are a relatively low barrier to entry among hands-on activities, yet they had a large influence on students' CU. Participants credited how the simulations offered them control and provided quick feedback. The quotes emphasized that students controlled some aspect of the simulation software that made it easy to quickly try many different options such as resistor or capacitor values or a slider for the number of frequency components. Then the simulation provided

[4] https://www.desmos.com/calculator/lab9nylxsi

immediate feedback on the impact of the change. For example, one of the graduate students said there was a program[5] where

> [I] can literally say 'this is my frequency, I want to hear this... I'm going to add this and this frequency and hear this one.' And then you can scroll through and see, 'oh, these are actually Fourier components that we are doing all the math for. It's not all random.' Which makes it very intuitive. (G)

Other students similarly describe how simulations changed "based on what you did" (G) or how "you would slide the slider and then you would see the [Fourier] representation" (UG). By giving control to students, simulations made the material feel "tangible" and "less abstract" (PE).

5.3.2.2 Design Problems

This theme captures that the participants are engineers; they gained CU by applying concepts to solve real-world problems. One of the practicing engineers contrasted the classic textbook problems in S&S with application-focused problems in an upper-level elective course:

> I loved it because it really connected, it was, it was an *engineering* course, you know, it wasn't just like this little, like, 'Okay, here's how to like write a low pass filter in DSP.' It was the whole system. Like you had to think big picture. And it was great. (PE)

There were many similar examples of students getting this real-world engineering experience in internships or early in their industry career. As an example, one practicing engineer commented that seeing how a concept "impacts our deliverable was a big part of solidifying the concepts" (PE).

Participants' explanations for how design problems improved CU fell into two main categories. First, real-world applications connect concepts to a purpose to make them less abstract, thus forcing students to consider the "why it matters" portion of CU. For example, at an internship working with audio signals, an undergraduate appreciated how the work involved

> actually apply[ing] a filter and then hear[ing] the changes. But for [SS] it kind of felt, it felt really imaginary. We were applying like a high-pass filter or a low-pass filter and I wasn't really sure exactly what was happening. Like I was looking at the graphs and I was like, 'I guess?' But, I wasn't 100% sure what was happening. (UG)

A graduate student similarly experienced that concepts connected to something tangible were easier to understand: "Pole-zero plots were pretty easy for me because I was an analog guy, I understood what-what poles and zeros were. That wasn't a big deal, that-that's all translatable" (G).

[5] Although he did not mention a specific program name, the description is similar to the program at https://www.codebymath.com/index.php/welcome/lesson/sound-sines.

Conversely, when concepts were not connected to applications, participants thought "the concepts were so abstract that I couldn't understand why I should care about them" (PE). Students mostly cited design problems (from homework problems, labs, industry, or extracurricular activities) for this real-world perspective, though instructors also provided example applications in lecture. As illustration, one faculty member mentioned how S&S has "applications in modeling of medical systems or human physiology or space dynamics or chemical processing plant or whatever" and that talking about these applications can help show students "why is this material useful, where is it useful" (F).

Secondly, open-ended design problems forced students to grapple with concepts more deeply than is required by standard homework or lab problems, thus helping students to gain the "what it is" part of CU. Undergraduate students said that, for the concept of sampling, they "didn't understand it in [SS] but we had to understand it for [an upper-level course] because we were actually building something" (UG), that "because we had more guidance over the design aspect, we had more meaningful discussions about what every concept really meant" (UG), and that they understood concepts better in upper-level courses because they "got to do everything on our own basically" (UG). One of the graduate students similarly discussed how an experience in industry forced a deeper understanding because "you understand a concept only when you use it for – when you really need it you go back and look at it" (G). He continues,

> There's never a point at the end of every course you remember everything perfectly. So, when you go back and actually use some of it for some-maybe to be maybe as your research work or as part of your job or whatever. There's when I had to design some called sort of notch filter and was like 'oh, this is what some of those frequencies were.' I think that helped me more than actually taking a course and stuff. (G)

5.3.2.3 Lab

Because the undergraduate students were from two universities and the graduate students completed their undergraduate education at five different universities, the participant pool included a wide variety of lab experiences. (Few practicing engineers recalled specific lab experiences.) There is a clear divide in the data between students who gained CU from lab and those who did not. In short, students who found labs helpful were those who engaged with the lab beyond simply following a set of procedures.

Students who had a lab that felt like "completing steps, just getting to the end" (UG) said the lab did not help CU. They admit the labs showed how concepts can be applied in hardware (which interested a few participants), but they did not learn about concepts or understand the applications. As stated by one of the graduate students,

> I don't think those circuits, like set up capacitors [and] inductors in the right way... I don't think that necessarily helped me. Like in terms of if the goal is to really help me conceptualize it. As opposed to actually just 'oh, it actually works' kind of thing... (G)

An undergraduate similarly stated that the labs "felt like you built a black box, you've applied things to it, but that doesn't mean you know what's going on inside of it" (UG).

One way to encourage students to engage more deeply with lab material is through lab reports. Lab reports had a mixed impact in our data, again depending on how students approached them. Some students had lab reports that required little processing or connecting of information. They could complete the lab report without gaining and CU, like this undergraduate student:

> the lab and like lab questions were like 'what results did you see.' ... If it was like 'explain,' [then] it's like 'I don't know what I was supposed to be looking at, so, I saw this and that's what happened.' (UG)

Students who mentioned labs helping their CU talked positively about the lab report or lab structure. Some lab reports required synthesizing information such as mentioned by another undergraduate:

> while you're writing the lab report, a lot of things actually come together because it's like you do learn a lot during the lab. But what you learn might not necessarily like come across at the point of like doing it. (UG)

Other students with similarly helpful lab writing assignments said the reports required students to test concepts and "think about what you just observed" (G).

Although structuring a lab to encourage student evaluation and synthesis of concepts helps, there was also variation within students at the same university as to how much labs helped their CU. Students who spent more time discussing lab results with peers or instructors (generally due to group dynamics) felt labs were more helpful. The faculty members also noted the interplay of time demands and CU with lab work:

> the labs have it [a design component]. But I'm afraid in the labs, they're so time-pressured, that they're going through the, you know, the steps, but they don't really have enough time to really sit down and figure out why they're doing it all. (F)

In other words, students who were required to "figure out why" from a lab report assignment benefited, but those without such a requirement moved on to other work without spending the time to interpret lab results. The following section explores the impact of workload in courses more generally.

5.3.3 Coursework

The previous lecture and hands-on activities factors concentrated mostly on positive student experiences or suggestions. In contrast, the subsections below describe factors that students typically identified as hindrances to their CU.

5.3.3.1 Grades that Emphasize Procedural Knowledge

Students need motivation to process ideas and form mental models of concepts. Motivation may be extrinsic, coming from graded assignments like design projects. Motivation may also be intrinsic or driven by students' work goals. For example, a graduate student explained that he does

> not just sit in class and [let the] professor teach and I don't really think too much about it. But more like in my spare time ask myself like 'oh, why is this important?' Like ask myself this question and try to find an answer myself. (G)

As opposed to his procedural-focused approach to undergraduate courses, he is more curious and intrinsically motivated as a graduate student.

Even when students mentioned being curious or intrinsically motivated, they prioritized completing graded assignments. By way of example, students said "I was just trying to get the homework done all the time" (UG) and that they were forced to adopt a "it's-a-requirement-so-just-get-it-over kind of mindset" (G). Some students talked about the concentration on procedures as a strategy, *e.g.*,

> if the goal is just pass the class, actually it's more effective to treat it as procedure, (G)

while others talked about it as a necessity, *e.g.*,

> the timeline and just the density of what we had to do in a single lab didn't allow for us to like introspect or think about what we were exactly collecting. (UG)

In both cases, participants noted that assignments were often focused on procedures, so focusing on graded assignments negatively impacted CU more than procedural knowledge.

Participants recognized the difficulty of designing assignments focused on CU. Even on questions designed to test concepts, students can sometimes earn full credit by copying procedures. On an exam problem on aliasing, a graduate "got full points for that problem and I still don't know what it means" (G). Previous results also suggest students can correctly answer concept questions by generalizing in-class procedures without understanding them [56].

5.3.3.2 Heavy Workloads

Students felt they gained more CU when they reflected and processed information. However, students often did not have time to process information. One of the graduate students said,

> A huge factor in my undergrad as to like whether or not I actually learned something or enjoyed it, was like how much I had going on. (G)

Similarly, an undergraduate said, “I felt like before I had the time to just process everything we were moving on” (UG). Conversely, when given more time for a design project in an upper-level elective, multiple undergraduate students mentioned using the time to talk with group members and to understand important design decisions. The time pressure is not only limited to coursework; as an undergraduate, you are also “taking a lot of courses, you’re trying to find time to go party, you’re trying, you know, you’re trying to do all this stuff right” (PE), which both faculty members also acknowledged.

A large workload can negate the positive impact of other strategies to help students gain CU. For example, one group of undergraduate students recalled a real-world homework problem that was designed to be motivating and make concepts concrete. However, students described it as being “like a whole four pages worth of like concept stuff for like history” (UG). Other quotes from the group suggest that they did want more of those connections to real-world applications and design problems, but the “shared trauma” (UG) of completing homeworks outweighed the motivational aspect of the problem.

Another concern with large workloads is that students may skip assignments. If tasks to build CU appear longer than procedure-focused problems (such as the four page homework problem), students are likely to skip them in favor of seemingly shorter problems and miss out on any benefits. Despite the complaints about workload, one of the students reluctantly admitted that the homeworks were helpful:

> it’s terrible and it’s long. But it’s the thing that causes you to be like ‘yeah, okay, I know where it is and I kinda understand’ or ‘I don’t understand at all.’ And I feel that was probably like the thing that was most helpful with grasping concepts. (UG)

By skipping many homework problems, the other students missed this benefit. Deciding on workload requires careful balancing as students also learn by repetition – the next section explores this point further.

5.3.4 Repetition and Upper-level Courses

As they were still taking courses, the undergraduate students comments about gaining CU after S&S concentrated on upper-level courses, internships, or extracurricular activities. One said, “I’d say I also remember Laplace and Fourier the most ‘cause we use it in other classes also, so, it’s hard to forget” (UG) and another said that after seeing a concept repeated in many classes, it seemed like “common sense” (UG). Others experienced forgetting specific concepts that were not reinforced, as stated by one undergraduate student:

> I would say that my knowledge grew after taking other classes as well. Um, but also very specific to what areas I ended up taking more classes in, because I would advance in those areas but not advance at all in other EE areas and maybe forget some stuff too. (UG)

A very common experience among the practicing engineers and graduate students is that they gained CU between S&S and graduating. Then they forgot details about topics that they did not use in their work/research while they maintained or gained CU in areas that they used. A fairly representative quote from one of the practicing engineers was: “most of what I remember is what I use on more or less a day-to-day now that I’m at work” (PE). Participants maintained some familiarity with concepts and were largely satisfied with understanding a concept sufficiently that they could recognize when they needed to use it (a type of “why is matters” CU) and then re-learn it. For example, one practicing engineer describes the timeline of her CU:

> Third year for signals and systems, I kind of grasped the knowledge. And then fourth year doing some more of it in a programming style, I think conceptually, I just get that better. And so it’s probably like kind of understanding it, really understanding it, and then like a really very consistent drop off after that. Because I just did not use it. (PE)

This participant points to the “programming style” of upper-level courses as helping her CU; the sections below describe other ways repetition and upper-level courses increased CU.

5.3.4.1 New Perspectives

Upper-level courses commonly helped many participants gain CU by presenting material differently than their S&S course, though the specific differences in presentation varied. For example, a programming focus helped the practicing engineer quoted above and emphasis on translating math helped one undergraduate (see Section 5.3.1.2).

A commonly cited new perspective was seeing more concrete applications in upper-level courses (or internships or extracurricular activities). An example of a concrete application came from students who took a course in image processing; one undergraduate expanded on a point made by another:

> I also took that class and just the idea of like preserving edges versus like smoothing and all of that stuff low-pass, high-pass filters just was taught to me in that class. I was like, 'oh, of course. Like that makes so much sense. Why, like, why didn't I know that as concisely before?' (UG)

The group agreed that the use of visuals in the image processing course made the effect of filters tangible. A communications course had a similar impact for concepts like the FT and noise properties for one of the practicing engineers:

> it was just presented in a way that was tangible in reality for me and-and I could think about it, and in a way that made more sense to me is more of an analog guy. What is a noise floor mean? What is noise spectral density mean? How to apply that to a model. It just was way less abstract, but ironically the mathematics were actually heavier. But it but it just, it just really connected for me. (PE)

As mentioned in Section 5.3.1, the students did not mind that these upper-level courses were heavily mathematical.

While courses like image processing helped students understand FT and filtering, other participants discussed how controls courses solidified their understanding of LTI and LT concepts. Participants credited controls courses with exposing them to non-LTI systems, which in turn helped them understand why LTI assumptions are important. One graduate student gave the following example of how controls courses require students to think in terms of an application:

> You're always thinking in s domain. Frequency domain. Always. So, okay, at this frequency, this is my frequency of interest, this is my bandwidth. So you're always doing 'okay, I want to increase the bandwidth of my controller.' Before that, it's bandwidth did not make any sense: 'Okay, bandwidth is like speed. It is speed, but what does it mean and why-why is it so important?' (G)

This quote highlights both the application and the repetition aspect of upper-level courses.

Offering multiple perspectives and example applications is possible within a S&S course. As a specific example, an undergraduate and a practicing engineer recalled their group members and teaching assistants helping them to think about ideas from different perspectives:

> if one team member has a really, really solid understanding of something, they might take lead in writing that part of the lab because that then helps others kind of fully see things and like I know, for me at least, I ended up writing the conclusion quite a few times and I would always look to what my teammates had written in the previous sections to, like, make sure that what I was writing was in line with what they'd really

> like gotten from each of the experiments and everything and just, like, make sure that all blended together, and especially if I like was missing a certain component of things in my understanding, at least. (UG)

However, presenting more applications and perspectives in S&S takes time. One possible compromise suggested by an undergraduate student was for instructors to pique student interest by advertising relevant upper-level courses when covering topics in SS.

5.3.4.2 Student Ability

Another explanation for the benefit of repetition in upper-level courses is that students are better prepared to understand S&S material after taking other courses. For example, a graduate student explained how complex numbers were a threshold concept for him:

> it took me probably until like my third year of graduate school to understand that complex numbers were interpreted as rotations in a plane. And then that's the critical link that had me actually understand what the formula meant. (G)

Once he understood complex numbers, he was able to more fully understand the 'how it works' concept behind the FT.

Students can and do benefit from previous exposure even without reaching full understanding. Multiple participants noted that seeing convolution before taking S&S (typically in a differential equations course), helped them in SS. Others said not having prior exposure to convolution made S&S more challenging. The combination of exposure and time to process the convolution process may be key, as suggested by this undergraduate:

> It was like you need the time to be able to like finally like catch on to something and be like 'okay now I see I can apply it.' And by taking [differential equations] first, you get the math, and in that semester worth of time, you have the amount of time for your brain to like 'okay, I got this' or 'I don't like this at all.' And then go into [SS] like 'I still hate this, but I know what's going on' or like 'I know now this is where I'm gonna be going' and stuff like that. (UG)

A member from industry similarly noted that she focused better on topics the second time because she could "focus less on the 'why are we doing this' and more on the, like, 'this is how we do this'" (PE).

5.4 Discussion

Tab. 5.5 summarizes the main instructional factors that participants identified as aiding or hindering their CU of SS. Each factor directly corresponds to an instructional strategy.

The lecturing techniques, such as relating math expressions back to concepts and emphasizing the purpose of ideas and how ideas connect, are easy ways for instructors to help students gain the 'what it is' and 'why it matters' part of CU. Students recognized these strategies as helping them gain a basic intuition for the concepts. Further, participants tended to retain this intuition, which often came in the form of being familiar with when a concept is relevant to an application, even if they forgot many details. The lecturing techniques do not directly match any of the eight strategies from Felder and Brent [128] (summarized in Section 5.1). However, they compliment the sugges-

Table 5.5: Summary of themes that emerged while analyzing the interviews.

	Aids	**Notes**
Lectures	Presenting CU along-side math	Use analogies to relatable experiences. Relate math expressions back to CU through visuals, higher-level discussions of procedures, or (repeatedly) translating math equations.
	Emphasizing purpose and connections	Provide experience with contrasting examples. Connect material within the course and to real-life applications.
Hands-on activities	Interactive simulations	Select a simulation where students have control over some setting and they get immediate feedback.
	Design problems	Help students to see the purpose of concepts by applying them to real-world problems and give students practice manipulating concepts.
	Labwork	Ensure students engage, reflect, and think about the lab work rather than only following a set of procedures, such as through a lab report that requires synthesizing concepts or explaining results.
Repetition and upper-level courses	New perspectives	Use visuals, concrete applications, and repetition to help students gain a new perspective.
	Student ability	Students may process ideas between semesters or learn new concepts in other courses that allow them to reach deeper CU when presented with SS concepts again in upper-level course.
	Hindrances	**Notes**
Coursework	Grades that emphasize procedures	Students prioritized graded assignments, which often focused on procedural knowledge more so than CU.
	Heavy workloads	Students did not have time to process information and form CU. Students may skip assignments, including motivating ones aimed at increasing CU.

tion to (FB1) make students interested in and prepared for the material, since providing intuition for topics helps students connect to and be interested in the material. The lecturing techniques do align with the CRKM model [121] in that they encourage making the message characteristics more comprehensible and compelling.

Lecturing is not sufficient for full CU. Aligned with much recent research on active learning, participants identified hands-on activities as where they truly started to understand concepts. Even simple activities, like interactive simulations on a computer, were helpful because students had an element of control and received immediate feedback. Design problems, either on homework or as part of a larger course project, allowed students to engage with concepts more and help them see where the concepts are useful. For S&S courses with lab sections, the lab is an obvious place for students to get hands-on experience. A clear divide in the data was between students who found lab helpful and those who did not. Those who did not described lab as blindly following a set of procedures and writing up results. In contrast, participants who gained CU from lab decided to or were forced to reflect on the lab results, often in a lab report. This result aligns with the CRKM [121] in that only labs that forced a high level of cognitive engagement led to conceptual change. A few of the students who reported the lab helping them gain CU cited conversations with their peers, similar to the recommendations in [77] for students to form a learning community.

The common hindrances to CU were related to workload and grades. A large workload meant students had to prioritize effort and spent their time completing graded assignments, which often emphasized procedural knowledge. Having a lot of work did not leave students time to process information and build CU. Students credited having more time as one way long-term design projects (typically in upper-level courses or internships) helped them gain CU. These hindrances match the recommendations from [128] that instructors should (FB3) structure grades to encourage deep learning over procedural knowledge and (FB7) keep the workload reasonable. The recognition that CU requires time also aligns with the findings and recommendations in [77].

This study asked participants about their CU over time. Although many of the strategies in Tab. 5.5 could be implemented in a S&S course, many students identified upper-level courses, internships, extracurricular, or industry work as where they started to really understand concepts. Participants identified the new perspectives and their higher starting ability as helping them gain more CU in experiences after their introductory S&S course. These results suggest the importance of the suggestion to (FB4) encourage students to be actively engaged in learning over the long-term. Further, if (FB8) encouraging a deep learning approach in one course will encourage a similar approach in future courses, S&S instructors can help students (eventually) gain CU, even if it takes students more time than they have in one semester.

Even without the other strategies, and without reaching CU the first time they saw a concept, participants thought repetition benefited CU (within or across courses). Although most students identified repetition, not just the passing of time, this observation raises the question of if their experience could be related to the impact of diffuse learning, or allowing students time between seeing a topic without explicitly trying to form CU [134]. Some data from [81] shows an increase in scores on a conceptual test after summer break – the authors propose lower stress as an explanation, but diffuse learning would also explain this finding. Untangling the impact of time and repetition on CU would be an interesting avenue for future work.

In Tab. 5.5, there is a clear tension between keeping a reasonable workload and providing students with new perspectives, repetition, and hands-on activities. Instructors generally cannot cover everything they want to in a course and must decide what material makes the cut. As one illustration of this trade-off, our results suggest it is preferable to decrease the number of assignments or labs with more time for each for students to gain CU. However, courses have other goals, *e.g.*, procedural knowledge or learning how to use lab equipment, that are often better achieved with more frequent, repetitive practice. This is something every instructor must balance. For topics that are (necessarily) covered briefly in a course, students preferred the topic to emphasize CU. They enjoyed being introduced to concepts in introductory courses at a high-level, especially if motivated by explaining their purpose and if the instructor told the class which courses would go into more detail. In contrast, participants in one student focus group described a more procedure-focused introduction to a topic; the students did not get to understand the full procedure and they were left more confused and had to un-learn the material in a later course.

The remaining suggestions from [128] are to (FB2) state expectations and provide clear feedback, (FB5) provide students with opportunities to influence the course content and learning methods, and (FB6) show care for the students learning. These did not explicitly appear in our data analysis. However, the general take-away from our results, that instructional quality and quantity can impact CU, suggest showing care for students learning is important. The results also generally support the idea of including these instructional factors in models for CU, such as ones based on the MoEP.

Finally, one point that was only briefly mentioned in the interviews, but which aligns with the threshold concept theory from [77], is that it can be challenging for instructors to recall how confusing concepts are and how they came to learn a concept. One of the graduate students noted that

> there's like sometimes use of these examples or conventions that are ubiquitous in the field but when you see it for the first time, it's like, the person presenting it or explaining it is so familiar and is so, like, they think that that's just the way it's presented and they fail to challenge it or explain why. (G)

Once learned, threshold concepts transform the way one thinks; this transformation can make concepts appear obvious that were once confusing.

5.4.1 Connections to Theory

Two prominent theories for CU are framework theory [39] and knowledge in pieces (KiP) [43]. These theories exist on a continuum: framework theory argues that knowledge is relatively coherent, while KiP argues that knowledge is relatively fragmented. Age, subject matter, and other contextual factors likely impact how conceptual change occurs and which theory is more applicable [43]. Section 2.1.3 describes these and other theories of CU in further detail. In agreement with the results in Tab. 5.5, both theories predict that students learn better when they see a concept multiple times from different perspectives, either as they move through progressively more expert-like synthetic frameworks (framework theory) or as they encounter new situations and confront more of their p-prims (KiP).

Participants' request for contrasting examples is particularly interesting to view in relation to Chi's "category mistakes" theory [40]. Participants noted that when they saw LTI systems in SS, they perceived LTI as a procedural check-list. They typically only understood the importance of LTI concepts when they later encountered non-LTI systems. One plausible explanation is that, rather than mis-categorizing LTI concepts as in Chi's theory, without contrast, participants were unable to categorize the concepts at all. This observation may be a direction for future research.

The interview protocol did not include asking if students understood the S&S concepts, though most participants talked briefly about their CU. Thus, we comment only briefly on students' mental models. We found KiP to be a helpful framework for interpreting these results. One undergraduate student noted he knew the definition of linearity in one context, but could not remember how it applied to systems. This reveals the contextual (and perhaps fragmented) nature of his knowledge. We can similarly use KiP to interpret some of the think-aloud results from Chapter 4. For example, students were able to select the FT of a windowed sinusoid on the first think-aloud question (Q9 on the SSCI) but did not recall the FT of a cosine to use on a later question (Q12).

No participants mentioned struggling to overcome naive concepts. In contrast, students mentioned analogies or examples from areas like music helping them to gain CU. One possible explanation is that students do not form a naive framework (as is common for physics concepts) because S&S concepts are not as obvious in everyday life. This observation agrees with the claim from Salzman and Strobel [32] that "the general lack of strongly grounded alternative conceptions or misconceptions about engineering and its processes limit the applicability of revolutionary models of conceptual change."

Even if students do not have initial naive conceptions for SS, the confusing terms used in S&S may present challenges. Although none of the student participants named this as a hindering factor, one faculty member talked at length about confusing terms, *e.g.*,

> the very use of the term signal is a little ambiguous, I suspect, to students. We tend to

> think of it as an alert or stop sign or something like that is a signal. But the context in which we use it is just a representation of a time series... It's a mathematical abstraction of some physical quantity. (F)

The faculty member also noted that students think of linearity as meaning "if you increase the amplitude, does the output grow proportionally?" A few students described linearity in this way, following the more common-place meaning of the word, which is incomplete in the S&S context. Others studies have also found that the language of SS (*e.g.*, the "overlap" step for computing convolution integrals and filters as "masks") can confuse students [56], [71] . If naive conceptions are less applicable in engineering [32], students' misapplying a familiar definition may be a more common equivalent of a naive p-prim or framework.

A future study investigating students' level of CU and how they formed their mental models of concepts over time could further discuss if either KiP or framework theory explained the development of students' CU in SS.

5.4.2 Limitations and Future Work

This research study involved a small number of participants, most of whom completed an undergraduate degree at a small number of universities in the United States. There was a single interviewer, whose identity, especially that of a graduate student in a SS-related field, likely impacted interactions with participants; see the discussion in Section 1.4 Finally, the interviews can only capture what participants perceived as influencing their CU – participants may not have realized or remembered the impact of various factors.

One limitation to the generalizability of our findings is that this study concentrates on S&S concepts. Future work should consider a similar research question in other engineering disciplines to look for themes and commonalities. Such qualitative studies could further theory on what instructional factors influence long-term CU. Once a more solid theory is established across multiple studies, a research team could design a survey to measure the factors that this and other studies hypothesize influence CU and relate those factors to a measure of CU such as the S&S concept inventory [1]. Such a quantitative design would allow for including a larger number of participants and testing the statistical significance of each variable. As briefly mentioned in Chapter 1, we planned this study to have a quantitative component, with students taking a survey to measure factors that we hypothesized would influence CU, similar to the survey and analysis from Chapter 3. However, we did not have enough student participation in 2021 and 2022 to do a full analysis of survey responses. Section 5.4.3 presents a preliminary, cursory analysis of the survey data.

Finally, participants' association of CU with a concept's purpose influenced the results of this study; many of the strategies in Tab. 5.5 emphasize demonstrating where a concept is used or the importance of the concept. Studies that introduce CU with different definitions are likely to find different factors.

5.4.3 Factors that Correlate with Conceptual Understandings

One obvious possible factor related to students' SSCI score is how many SS-related electives seniors took. Of the participants who completed the SSCI in 2020 and 2021 as part of the study in Chapter 4, $n = 143$ (90 from UVA, 53 from UM) took an additional survey that collected information about upper-level electives and grades. The full survey is given in Appendix A.2. UM participants took the survey and SSCI as part of the research (the SSCI was not part of a class for them), while UVA students were incentivized to take the additional survey with a raffle for $15 gift cards in fall 2020 and 2021.

Although we did not have sufficient participation to do a full analysis of factors impacting SSCI scores, as a brief, initial analysis, we performed two t-tests to see if the amount of exposure students had to S&S concepts after SS was correlated with their SSCI score. First, we tested if scores on the LTI and Laplace transform questions (Q8, Q17-19, and Q24) differed for students who did and did not take a controls course. LTI and Laplace transforms concepts are integral to controls courses, so we predicted that students who took a controls course would reach better CU of these concepts. Second, we tested if scores on the convolution and FT questions (Q6, Q7, Q9-12, Q22, and Q25) differed for students who did and did not take a signal processing and/or communications course. Similar to how controls courses tend to emphasize LTI and Laplace concepts, signal processing and communications courses emphasize convolution and the FT, so we expect students who complete these courses to have better CU. In both cases, students with a relevant course scored significantly higher (p<0.01) on the related concept questions, with an average score improvement of 21% and 16% respectively. Bartlett's test of equal variances came back insignificant, suggesting that the assumption of the t-test that the two groups have equal variances is reasonable.

We further used a linear regression to see if the impact of taking a SS-related course remained significant if we accounted for differences in student grades. The two independent variables were (1) the number of SS-related electives a student took and (2) their self-reported typical grades in engineering courses, with "mostly As" coded as 4.0, "mostly A and B" coded as 3.5, etc, and the dependent variable was their SSCI score. The linear regression model was significant (p<0.01) and explained 27% of variance in scores. The regression coefficients suggest that scores increased by 6% with every additional SS-related elective course and 13% with every increase in average letter grade. However, the results do not control for many other possible confounding variables nor do they suggest causality–students may choose courses in areas in which they already have higher

CU.

5.5 Conclusion

To address the third research question (RQ#3: What instructional factors influence CU of S&S for senior students?), we interviewed two faculty members, 8 undergraduate students, 5 graduate students, and 4 practicing engineers and used a constant comparative analysis on the transcript data. Participants noted multiple experiences that helped them gain CU: instructors presenting CU in parallel with mathematical expression in lectures, instructors emphasizing the purpose of ideas and how ideas connect in the course, interactive simulations where they had control and received immediate feedback, design problems, lab-work that forced them to reflect on ideas, new perspectives from upper-level courses, and repetition of concepts across the curriculum. They also noted that emphasizing procedures in graded assignments and heavy workloads made it harder for them to gain CU. Although this study focused on S&S concepts, we expect these findings to generalize to other engineering areas, particularly those with heavy mathematical content.

Few studies have investigated students' CU of S&S concepts multiple semesters after a S&S course. We hope this chapter provides concrete ideas for instructors on how to increase CU and encourages curriculum designers to consider how repetition across multiple courses can help students gain more advanced CU that they often do not have time to develop in a single S&S course. We end with a quote for thought: an undergraduate stated that S&S

> is nice for giving you kind of like the introduction for, 'hey, this is stuff that is in electrical engineering.' And then your upper level would be like, 'this is *why* this is in electrical engineering.' (UG)

Perhaps, if students gain CU in upper-level electives (or other experiences), the commonly reported low CU at the end of a S&S course [56] is less discouraging for S&S instructors.

CHAPTER 6

Part I: Summary, Contributions, and Conclusions

Part 1.5 of this dissertation looked at conceptual understanding of signals and systems in three phases. The three research questions were:

RQ#1 What is students' CU of S&S concepts at the end of an undergraduate S&S course? What factors predict how many S&S concepts students learn in a S&S course?

RQ#2 What is the CU of S&S concepts among senior students?

RQ#3 What instructional factors influence CU of S&S for senior students?

Chapter 3, 4, and 5 addressed each of these questions in turn. This conclusion reviews the main findings from each study and how the themes that emerge across the studies. Fig. 6.1 shows a timeline for collecting data for Part 1.5 and how each data source related to the three research questions.

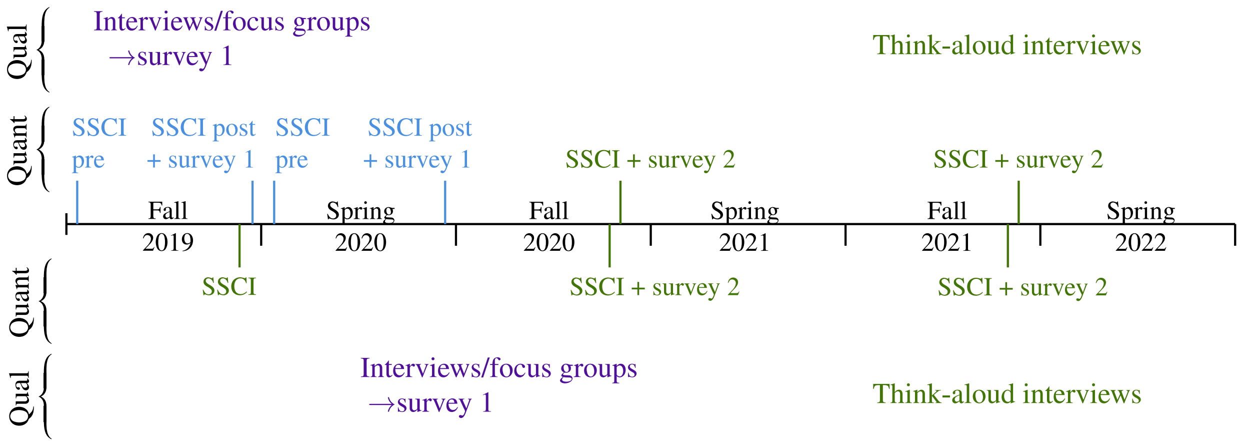


Figure 6.1: Summary of the data collection for Part 1.5 of this dissertation. The top half of the diagram is for UM and the bottom half is for UVA. The data from the SSCI and the survey given in S&S courses at UM answered RQ#1. The data from the SSCI given to seniors and the think-aloud interviews answered RQ#2 (the SSCI given to seniors at UVA from Fall 2016-2018 is not depicted). Finally, the exploratory interviews and focus groups influenced the design of the surveys and answered RQ#3.

6.1 Concepts in Signals and Systems

This section discusses results specific to each S&S concept that I studied and includes data from the interviews from Chapter 5 where participants talked about their understanding of each concept. For background, Section 2.2.1 introduced the concepts. Section 6.1.3 combines the discussion of FT and filtering as many interview participants talked about these almost interchangeably. Measuring CU was not a main focus of these interviews, so I did not do a formal qualitative analysis of the quotes where participants talked about their CU. However, I include some of the themes that emerged from those conversations where the interviews overlapped with our main results.

This section also briefly discusses how concepts are presented in common S&S textbooks [33]–[35]. In general, textbooks present many S&S concepts as "what it is" concepts initially and include real-world motivating applications either in the introduction chapter, the introduction section of each chapter, the homework problems, or as pointers in the bibliography. In the edits for the second version of the textbook, Oppenheim, Willsky, and Nawab [34, p. xvii] explained that they wanted to increase the emphasis on applications. For example, they moved the discussion of frequency-domain filtering earlier "to provide both motivation and insight." Textbooks are not generally designed for self-study; they are meant to be used in conjunction with other materials as part of a course. Therefore, what textbooks present is not necessarily indicative of how concepts are presented in a course. However, textbooks reflect community values and are minimally indicative of how we expect students to use textbooks to learn. I briefly comment on textbooks as supporting evidence and context for the other results; analyzing textbooks thoroughly is outside the scope of this discussion and would be an interesting avenue for future work.

6.1.1 Linearity and Time Invariance

Textbooks generally start with the concept of time invariance (TI) before linearity. For example, Phillips, Parr, and Riskin [35] give the definition of TI in words and in mathematical notation, discuss how to test it, provide mathematical examples, and give an example of a time-varying real-world system (the booster stage of the NASA shuttle) before doing the same for linearity. When introducing linearity, all three of the common textbooks I reviewed emphasize the importance of the superposition principle. The full explanation of why superposition is an important concept is typically saved for later in the book, after introducing other concepts like convolution.

Students generally rate LTI as easier than instructors rate it [72], and our results show that students generally score well on most of the LTI concept questions on the SSCI. For example, even on the pre-test in S&S, over 90% of students correctly answered question 5 on the concept of TI (note the pre-test was two weeks into the semester, and some students had seen TI in lecture although they had not yet turned in a homework on the concept). In interviews with the participants from Chapter 5, participants also typically described LTI as easy or intuitive.

Although students thought of LTI as an easy concept and they were able to apply it to problems, our data supports the finding from Nasr, Hall, and Garik [71] that students did not fully understand the concept. In contrast to the high scores on the other LTI questions, in Chapter 4, we observed students struggling to answer question 24 about LTI concepts on the SSCI. Most were able to correct reason about TI, but only about half showed CU of linearity. The SSCI results from the post-test in S&S suggest many students have a similar gap in their CU at the end of the S&S course. The interview data from Chapter 5 also support that students think LTI is easy, but that they do not have full understanding. One undergraduate student participant confused linearity for continuity and another recalled the mathematical definition but wasn't sure how it would apply to systems: "I think of the meaning as it being closed under scalar multiplication and addition but I don't remember how to, like, meld that definition to systems. Like, something like that. It's gotta be related to that though."

One conversation about LTI in a focus groups with undergraduates was particularly interesting. One student talked at length about how they felt LTI was not covered well in the undergraduate curriculum and that they only came to see the fundamental importance of the concept from physics and mathematics courses and being exposed to systems that were non-linear. In response, one of the other participants nicely summed up the theme of the conversation:

> I think on a surface level LTI concepts are like the easiest thing - on a surface level - the easiest thing I've ever learned, because in every single almost every single EE class I've been in, the first homework is LTI. And they have you memorize like the four rules to tell if it's LTI, and you do-like you go through like six equations of systems, and you write is this system LTI, and then we'd never touch it again. And so, on a surface level, it's a very easy concept because you're like, "I just need to follow these rules, it's just simple math." But, listening to [other participant], I realized, I never learned why it's good for a system to be linear or LTI or-or time invariant. Why would you - why is it more desirable to design a system like that? What benefits does that give you? So I guess I had like a sense of false confidence that I really understood this stuff because our first unit every class was LTI, but I truly did not understand on a more deeper level why that's kind of a big deal.

When saying LTI is easy, the participants talked about the procedural part because that is how they saw LTI. Only the participants who mentioned having experience with non-LTI systems (either in an upper-level elective, in industry, or in extra-curricular activities) talked about deeper CU of, or appreciation for, LTI concepts. This contrast also held for the four practicing engineers we interviewed: two recalled LTI concepts well and two did not at all (one said that they recalled that LTI was “a thing” and the other asked themselves if it was “linear versus time invariance” before stating that they did not remember it at all).

A related finding from the think-aloud interviews from Chapter 4 was that some students struggled to separate linearity from TI. A few tried to assume the system was LTI in the process of testing if it was LTI. Although the data is limited, this observation suggests that students are so accustomed to working with LTI systems that they do not know how to approach a non-LTI system. During the same conversation from the undergraduate focus group on LTI mentioned above, one of the students described being asked to analyze a non-LTI system as going to the “forbidden zone.” Nasr, Hall, and Garik [71] also found that students tended to automatically assume that a theoretical system presented in mathematical form was LTI to determine the output, even if that was not specified or not true. In contrast, when aerospace engineering undergraduates were given a real-world aerospace system that they knew was non-linear, they more often (correctly) said the output could not be determined from the given information [71].

From Section 2.2.2, a possible reason that students struggle with S&S concepts is that the terminology is misleading and confusing. For LTI, the think-aloud interviews showed a few students confused linearity and proportionality. One of the instructors in the interviews from Chapter 5 noted exactly this difficulty:

> people tend to think of linear is like an audio amplifier is linear, if you increase the amplitude does the output grow proportionally. And it’s more than that.

Jia, Bennett, Nguyen, *et al.* [78] discusses other examples of confusing terminology such as the “area under the curve” and students thinking that a system that shifts an input in time is time varying.

6.1.2 Convolution

Convolution is generally ranked as one of the most difficult concepts in S&S [72], likely due to its mathematical nature. The interviewees from Chapter 5 agreed with this. Many participants said that they did not understand convolution and cited the heavy mathematics as part of the problem. But even participants who felt they understood the math thought the concept was difficult or lost in the procedural steps during their S&S course.

The concepts in Section 2.2.1 are from standard S&S textbooks [33], [35]. Most of the concepts are at the “what it is” level. For example, one of the convolution concepts was:

> Convolution is associative, *i.e.*, $(x(t) \circledast h_1(t)) \circledast h_2(t) = x(t) \circledast (h_1(t) \circledast h_2(t))$. Therefore, the impulse response of the equivalent system to a series of LTI systems is the convolution of the impulse responses for the individual systems.

Although some might consider the second sentence a “why is matters” concept, it is translating the “what it is” concept from mathematical representation to words. The three common textbooks I reviewed [33]–[35] all present the concept that convolution is associative in block diagram form. The “what it is” concepts are generally presented with the corresponding proof or mathematical explanation behind the concept.

One of the more important “how it works” concepts relating convolution and LTI is understanding how convolution calculates the output of LTI systems due to the principle of superposition: it sums up copies of the input signal delayed and scaled according to the impulse response. This concept is mentioned but not emphasized in the introductory sections on convolution in [33], [35]. Ref. [34], [35] uses the goal of finding the output of an LTI system and the superposition property to build-up the convolution integral and emphasizes the importance of the result.

None of the convolution concepts in Section 2.2.1 emphasize the “why it matters” level. Further, the example problems in [33, Ch. 2] do not demonstrate why the properties are useful beyond solving standard homework problems. Ref. [35] similarly includes typical mathematical examples, such as finding the impulse response of a integrator, and the authors motivate the example systems as being useful in signal processing applications and control systems. Continuing the above associative concept example, a full “why it matters” should explain the practical significance of the associative property of convolution (not just of convolution or LTI systems in general), preferably with a specific, real-world application.

Perhaps the most basic “why it matters” concept behind convolution is why convolution is important. First, convolution is important because it is used to calculate the output of LTI systems and many real-world systems are well-modeled by LTI systems. This concept is presented and emphasized with numerous examples in textbooks and students generally remember this purpose [56]. However, the convolution procedure is mathematically challenging, so students are unlikely to appreciate this as a boon and therefore miss the motivational aspect of this “why it matters” concept. As mentioned in the previous section, the benefit of convolution and LTI is more apparent when one considers non-linear or non-time-invariant systems, which are considerably more challenging. However, second or third year students rarely see such systems in undergraduate courses.

A second viewpoint on why convolution is important is to consider specific applications. Such examples are typically presented in later chapters in textbooks due to incorporating other course material, such as Fourier transforms. One way to show “why it matters” concepts to students is with example convolutions. For example, convolving with a rectangular (low-pass) function takes a moving average and smooths the input signal; this is helpful in applications such as weather reporting or reading measurements from a sensor where any one measure may be noisy, but the time-average is more reliable. Another example is convolving with a differencing (high-pass) filter; this is helpful for edge detection. Both examples connect to FT concepts but are easily seen directly from the convolution integral.

Despite convolution being the building block for the FT and filtering, students may not see convolution as important. Convolution was rated the least likely concept to benefit students in their future careers on the survey in Chapter 3. In contrast, Nelson, Hjalmarson, Wage, *et al.* [72] found that students thought convolution was one of the more important concepts in S&S. The difference in findings is likely attributable to the definition of importance. The survey from Chapter 3 asked about importance of understanding the concept for their future career while [72] noted that students who said convolution was important did so because convolution appeared in other courses.

The SSCI results in Tab. 6.1 show that a large percentage of students correctly answered the question testing if they understand that convolution is commutative and the question that requires students recall that the convolution of two rectangles is a trapezoid (questions 14 and 13 respectively). In contrast, the results for question 15 suggest low CU. As discussed in Section 4.3.2, the think-aloud data for questions 15 suggest is similar to the findings in [56], [71] that students did not know how to handle novel convolution problems. Wage, Buck, Nelson, *et al.* [56] and Nasr, Hall, and Garik [71] found students struggled with convolution problems when the input signal did not start at time 0, the input signal had a magnitudes other than one, and when the input signal with not unit width. Both students in S&S and seniors had the same common incorrect answer on question 15, likely stemming from not accounting for the input signal having a width greater than one unit when determining the maximum amplitude of the output signal.

The interviews from Chapter 5 brought up two additional points on convolution for reflection and discussion. First, there was disagreement between students and instructors on whether the graphical representation of convolution is: (1) a way to provide intuition for the superposition integral and a tool for finding the break-points in a convolution problem (instructor viewpoint) or (2) an important procedure that students should know (student impression). The two viewpoints are not incompatible, but they reveal different emphases. I do not argue that one viewpoint is correct or better than the other, but instructors should clearly define what they want students to learn and make sure these goals align with homework problems and exam questions.

Second, when thinking about convolution from the student perspective of the graphical convolution procedure being important on its own (not in service of findings the bounds for more complex convolution problems), students questioned the importance of teaching the concept. Students were roughly split on whether the graphical convolution examples helped their CU of the superposition integral, and many participants made comments similar to this graduate student: “the focus was definitely on just being able to calculate something. Which may or may not be useful at all. I mean, it’s probably not because who’s ever going to do convolution by hand? Nobody!” Even if one thinks graphical convolution is not important beyond its use in finding breakpoints and understanding the superposition integral, graphical convolution questions can reveal problematic reasonings. For example, the common error on question 15 (which required students to recognize the answer for a graphical convolution problem) revealed that many students did not understand how the width of the input signal impacts the maximum amplitude of the output signal (see Section 4.3.2). If the procedure can reveal this type of error, it may also help instructors identify ways to help students to learn the associated concepts.

6.1.3 Fourier Transform and Filtering

Many of the results on the FT echo the results on convolution, while the results on filtering are often quite different. Similar to convolution, students generally think of the FT as a difficult concept, likely because of the mathematical nature [72]. As with convolution, some interview participants from Chapter 5 said they did not understand the FT in S&S even if they understood the mathematics and could compute the integral. In contrast, students perceive filtering as one of the easier topics in S&S [72].

FT properties are presented in textbooks [33]–[35] as equations and symbols with proofs or mathematical examples; there were generally few physical examples or explanations relevant to specific applications such as visualizing audio frequencies. Immediately after introducing the FT, [34] explains the mathematics behind and the goal of filtering and removing certain frequencies. Phillips, Parr, and Riskin [35] introduces filtering by presenting ideal high-, low-, and band-pass filters. The authors present filters with varying amplitudes in the pass band, but start with filters with unit magnitude. They then note that such ideal filters are not physically possible and discuss RC (resistor capacitor) low-pass filters.

Looking at the SSCI results in Tab. 6.1, a majority of students in S&S and seniors correctly answered FT questions (questions 7, 9, 11, and 12; excluding the FT questions that require reverse reasoning). Even though most students answered correctly, the think-aloud results for question 9 suggested that seniors were not confident in their answer; see Section 4.3.3. Question 9 tests one of the most fundamental concepts in S&S: it asks students to identify how the FT of a signal changes when the frequency of the signal changes. Further, the results for question 12 on the think-aloud interviews showed that multiple students were unable to recall the FT of a cosine. One student mentioned that they would normally look this fact up on a FT table. Interestingly, the students in S&S did much better on question 12 than the seniors (93% answered correctly as opposed to 63% of seniors). Recall from Section 4.3.3 that we included question 12 on the think-aloud interviews to see if students used a “what it is” approach (this approach required knowing the FT of a cosine) or a “why it matters” approach (recalling the purpose of carrier waves). An interesting avenue for future work is to do think-aloud interviews with students in S&S to see if they are better at recalling

Table 6.1: Summary of SSCI questions, including the relevant concept, the fraction of students in S&S that answered the question correctly on the pre-test and post-test ($n = 180$), and the fraction of senior students answering correctly ($n = 412$). Highlighted questions are used in the think-aloud interviews, starred questions were used in [56], and questions with RR require reverse reasoning.

Question	Concept	SS: Pre	SS: Post	Senior
Q1	The definition of frequency.	0.99	0.99	0.98
Q2	Time-reversal in the signal domain.	0.92	0.96	0.78
Q3	Recognize a time-reversed and shifted signal.	0.41	0.52	0.45
Q4	Convolution is commutative.	0.88	0.98	0.79
Q5	Convolution computes the output for an LTI system. Graphical convolution of rectangular pulses.	0.91	0.89	0.93
Q6*	How to determine if a system is causal based on its impulse response.	0.67	0.89	0.73
Q7	The definition of the Fourier series.	0.46	0.58	0.67
Q8	Sinusoids are eigenfunctions of LTI systems.	0.50	0.79	0.64
Q9*	Increasing the frequency of a signal in the time domain correspondingly increases the frequency in the FT domain.	0.39	0.73	0.67
Q10-RR	Convolution-multiplication duality of the FT.	0.30	0.57	0.25
Q11	The FT is homogeneous.	0.83	0.92	0.83
Q12	Convolution-multiplication duality and the FT of a cosine. Or, how multiplication with a carrier wave impacts the FT.	0.60	0.93	0.63
Q13*	Convolution computes the output for an LTI system. Graphical convolution of rectangular pulses.	0.32	0.80	0.69
Q14-RR	Convolution is commutative.	0.67	0.91	0.82
Q15*	Convolution computes the output for an LTI system. Graphical convolution of rectangular pulses.	0.34	0.34	0.47
Q16	How to determine if a system is causal based on its impulse response.	0.27	0.78	0.47
Q17	How to interpret a pole-zero plot to determine a system's causality and stability.	0.22	0.74	0.54
Q18	The relation between a system's pole-zero plot and its impulse response.	0.31	0.67	0.40
Q19	The relation between a system's pole-zero plot and its frequency response.	0.18	0.46	0.31
Q20	How adding a pole to a frequency response impacts the corresponding Bode plot.	0.48	0.63	0.70
Q21-RR	Convolution-multiplication duality.	0.12	0.52	0.39
Q22-RR	Time-phase shift duality.	0.19	0.64	0.40
Q23	The relation between the impulse response of a system and whether the system is causal. Parallel and cascade connections of systems.	0.40	0.52	0.38
Q24	Graphical interpretation of linearity and time invariance.	0.32	0.43	0.40
Q25*	Low pass filtering of windowed signals.	0.51	0.72	0.70

(possibly from memorization) the FT of cosine, if they are able to deduce the FT of a cosine, or if they are more likely to use the "why it matters" approach.

Students similarly tend to correctly answer the SSCI questions on filtering (questions 6 and 25). Unlike with the FT questions, the filtering question on the think-aloud interviews suggested that most students understood that the filtering concepts being tested. Our results, presented in Section 4.3.4, are in contrast to the think-aloud results from [56], who found that students were not thorough in answering the filtering question. Specifically, [56] found that students tended not to check the passband of the filter nor the filter magnitude.

The comments from the interview participants from Chapter 5 appear contrary to the SSCI results on FT but seem to support the results on filtering. Most participants felt they understood the concept of frequency content of time signals by the time they were undergraduate seniors. While many participants talked about how the math was challenging and they may not have gained intuition for the FT during the S&S course, many also talked about how they eventually saw the frequency-time relation as intuitive. Using natural language processing tools such as those in [68] to examine how a large group of students explain their answers to–and rate their confidence on–problems similar to question 9 on the SSCI would be an interesting area for future work.

6.2 Implications for Practice

Section 5.4.1 discusses the implications for theory for many of the findings from Part 1.5. This section concentrates on implications for practice, *i.e.*, how the findings from Part 1.5 might impact teaching. These implications are summarized in Tab. 6.3.

First, measuring students' CU is a first step to improving students' CU. By knowing which concepts students struggle with and their common errors, instructors can more effectively target those specific concepts. For example, the results in Tab. 6.1 show that most students were able to interpret a magnitude and phase diagram for a filter (question 6) but that they struggled to determine if a system of series and parallel connections of systems is causal given in the input responses of the individual systems in the connection diagram (question 23).

Second, Chapter 5 identifies many instructor strategies that may help improve CU. These are summarized in Tab. 5.5. The strategies included presenting CU along-side math equations in lectures, repeating the interpretation of mathematical expressions multiple times, emphasizing the purpose of concepts and connections between course material, and incorporating active learning. When incorporating active learning, participants noted that quick interactive simulations, which have a lower barrier to entry than full lab experiments, can help students develop CU. The key features for these simulations are that they give students a sense of control and provide immediate feedback. Lab experiments were very helpful for some students but not for others. To help develop

Table 6.3: Summary of implications for practice from Part 1.5.

Implication	Details	See also
Anticipate student difficulty with certain concepts.	Students at the end of S&S and senior undergraduate engineers both struggled with SSCI questions on flipping and shifting a time signal (Q3), how to the width of the input signal impacts the maximum amplitude of the output signal in a graphical convolution problem (Q15), and the definition of a linear system separate from a linear *and* time invariant system (Q24). Students also tended to score lower on concept questions that required reverse reasoning or synthesizing concepts, *e.g.*, Q10, Q21, Q22, and Q23.	Sec. 3.3.1 Chap. 4 Tab. 6.1
Contrast LTI with non-linear systems.	To help students appreciate the importance of LTI systems and see that such systems are easier to analyze, contrast the analysis of LTI systems with non-linear systems.	Sec. 4.3.1 Sec. 5.3.1
Explain the purpose of graphical convolution problems.	Instructors should clearly define what they want students to learn from doing graphical convolution problems and make sure these goals align with homework problems and exam questions.	Sec. 4.3.2 Sec. 6.1.2
Improved learning in pre-requisite courses aids in gaining CU in S&S.	Pre-test SSCI scores was a significant predictor of post-test SSCI scores and interviewees felt being exposed to convolution before S&S helped them to better learn the signal processing concepts by allowing them to concentrate on the application of convolution rather than being overwhelmed with the procedural steps.	Sec. 3.4 Sec. 5.3.4.2
Provide motivation for students to care about concepts.	Motivation was a significant predictor of post-test SSCI scores and Chap. 5 discusses how interviewees appreciated when their instructor motivated concepts by presenting the "bigger picture" or purpose behind them.	Sec. 3.4 Sec. 5.3.1
Use lectures to provide students with an initial level of CU.	Students especially appreciated instructors presenting "why it matters" concepts and connections between concepts. Instructors can use techniques such as analogies, translating math expressions, and contrasting examples to help students develop initial CU during lecture.	Sec. 5.3.1
Integrate hands-on activities to help students construct their own deeper level of CU.	These activities could come from a laboratory section or in a project-based course; the interviewees suggested that labs and projects should require students to engage with material and avoid having students follow a set of procedures. Students also appreciated quick activities such as interactive computer simulations where they control some element of the simulation and receive immediate feedback.	Sec. 5.3.2
Repeat concepts within a course and across a curriculum.	Repetition helps students gain CU by providing them with new perspectives and by re-introducing them to concepts when they have higher ability. Sec. 5.4.3 also shows preliminary evidence that upper-level electives significantly increase CU of certain emphasized concepts, while CU of other concepts is left unchanged.	Sec. 5.3.4 Sec. 5.4.3
Allow students time to develop CU.	Students may develop less CU if the assignments with the most weight emphasize procedural knowledge, especially if their workloads are high enough that they do not have any time to spend on courses beyond time spent to complete these assignments.	Sec. 5.3.3

CU, lab experiments should encourage students to synthesize and reflect on concepts; following a set of procedures and making calculations was insufficient for participants to identify labs as helpful for CU. Chapter 5 also warns against a heavy workload and notes that too much work can negate the positive impact of other activities such as application-focused homework problems.

Finally, a major focus of this dissertation was on the CU of senior students. The second and third phases both focused on this population of students, motivated by the lack of data in the literature about CU multiple semesters after students take a S&S course. Chapter 4 showed that seniors scored similarly to the scores reported in the literature for students taking a post-test in S&S and Chapter 5 found that many participants credited time as aiding their CU. Time to process ideas can aid students in developing CU [134], but many students specifically credited applying S&S concepts in follow-on classes, extracurricular activities, or at work as helping them to gain a new perspective on S&S concepts and thus improving their CU. An implication of this result is that designing the curriculum such that students see concepts repeated in multiple courses could improve CU of upper-level students. Section 5.4.3 presents preliminary results that suggest that specific upper-level courses improve CU of specific S&S concepts.

6.3 Future Work

Ultimately, the goal of this research is to shed light on how instructors and curriculum designers can help students reach long-term conceptual understanding. The results suggest multiple avenues for future research, such as:

- What caused the large observed conceptual gain in the S&S course in Chapter 3? One hypothesis stemming from the interview data from Chapter 5 is that the large gain is partly due to students taking differential equations as a prerequisite rather than a co-requisite and thus being better prepared for the concepts in S&S.

- How does CU of S&S evolve over time? Chapter 4 measured CU of senior students, but a future study could survey the same participants over multiple years in a longitudinal study to collect paired data and better see the impact of individual student experiences.

- What factors predict CU of senior undergraduate students? Chapter 5 presents exploratory qualitative data as a step toward answering this question and Section 5.4.3 showed that upper-level courses and grades were correlated with CU for senior students. A future study could build on these results and investigate how well a model such as the Cognitive Reconstruction of Knowledge Model (CRKM) [121] explains CU of senior students.

In addition to these directions, Tab. 5.5 suggests future work questions for individual S&S concepts.

Another area for future work is to develop new questions to test CU based on the findings from the existing questions. For example, a question on filtering that reveals if students verify the passband of the filter and the filter's magnitude would help determine if the results from Section 4.3.4 generalize to a larger student population. New concept questions could also use the definition of CU proposed in Section 2.1.2 to target different concept levels. For example, if there were two paired questions with one testing a "why it matters" concept and the second testing the corresponding "what it is" concept, a future study could see if one concept had more "staying power" with students, *i.e.*, if students were more likely to recall one of the concepts multiple semesters or years after taking a relevant course.

Finally, this dissertation considered concepts from signals and systems (SS) because it is pervasive across electrical engineering program and because of the expansive literature showing that students struggle with many of the concepts [1], [45], [56]. S&S is a core class for many subdisciplines in electrical engineering, including signal and image processing and machine learning. We anticipate the results will provide insight for other engineering disciplines, especially those with similarly heavy mathematical content. Hopefully others will conduct similar studies in other engineering areas so that the research community can identify common factors, similar to how [50] identified common conceptual challenges across engineering disciplines.

Part II: Image Reconstruction

CHAPTER 7

Introduction

This chapter motivates the image reconstruction problem, provides an introduction to the notation in Part 6.3, and introduces a running example bilevel problem used throughout Part 6.3. This chapter is presented in Ch. 1 of [11]:

> C. Crockett and J. A. Fessler, "Bilevel methods for image reconstruction," *Foundations and Trends® in Signal Processing*, vol. 15, no. 2-3, pp. 121–289, May 5, 2022, ISSN: 1932-8346, 1932-8354. DOI: 10.1561/2000000111

7.1 Motivation: Image Reconstruction

Methods for image recovery aim to estimate a good-quality image from noisy, incomplete, or indirect measurements. Such methods are also known as computational imaging. For example, image denoising and image deconvolution attempt to recover a clean image from a noisy and/or blurry input image, and image inpainting tries to complete missing measurements from an image. Medical image reconstruction aims to recover images that humans can interpret from the indirect measurements recorded by a system like a Magnetic Resonance Imaging (MRI) or Computed Tomography (CT) scanner. Such image reconstruction applications are a type of inverse problem [135].

New methods for image reconstruction attempt to lower complexity, decrease data requirements, or improve image quality for a given input data quality. For example, in CT, one goal is to provide doctors with information to help their patients while reducing radiation exposure [136]. To achieve these lower radiation doses, the CT system must collect data with lower beam intensity or fewer views. Similarly, in MRI, collecting fewer k-space samples can reduce scan times. Such "undersampling" leads to an under-determined problem, with fewer knowns (measurements from a scanner) than unknowns (pixels in the reconstructed image), requiring advanced image reconstruction methods.

Existing reconstruction methods make different assumptions about the characteristics of the images being recovered. Historically, the assumptions are based on easily observed (or assumed) characteristics of the desired output image, such as a tendency to have smooth regions with few edges or to have some form of sparsity [7]. More recent machine learning approaches use training data to discover image characteristics. These learning-based methods often outperform traditional methods, and are gaining popularity in part because of increased availability of training data and computational resources [137], [138].

There are many design decisions in learning-based reconstruction methods. How many parameters should be learned? What makes a set of parameters “good?” How can one learn these good parameters? Using a bilevel methodology is one systematic way to address these questions.

Bilevel methods are so named because they involve two “levels” of optimization: an upper-level loss function that defines a goal or measure of goodness (equivalently, badness) for the learnable parameters and a lower-level cost function that uses the learnable parameters, typically as part of a regularizer. The main benefits of bilevel methods are learning task-based hyperparameters in a principled approach and connecting machine learning techniques with image reconstruction methods that are defined in terms of optimizing a cost function, often called model-based image reconstruction methods. Conversely, the main challenge with bilevel methods is the computational complexity. However, like with neural networks, that complexity is highest during the training process, whereas deployment has lower complexity because it uses only the lower-level problem.

Part 6.3 focuses on formulations and applications where the lower-level problem is an image reconstruction cost function that uses regularization based on analysis sparsity. The application of bilevel methods to image reconstruction problems is relatively new, but there are a growing number of promising research efforts in this direction. We hope the review of bilevel methods serves as a primer and unifying treatment for readers who may already be familiar with image reconstruction problems and traditional regularization approaches but who have not yet delved into bilevel methods.

For overviews of machine learning in image reconstruction, see [138], [139]. For an overview of image reconstruction methods, including classical, variational, and learning-based methods, see [140]. Finally, for historical overviews of bilevel optimization and perspectives on its use in a wide variety of fields, see [8], [141]. Within the image recovery field, bilevel methods have also been used, *e.g.*, in learning synthesis dictionaries [142].

7.2 Notation

Part 6.3 focuses on continuous-valued, discrete space signals. Some papers, *e.g.*, [143], [144], analyze signals in function space, arguing that the goal of high resolution imagery is to approximate a continuous space reality and that analysis in the continuous domain can yield insights and optimization algorithms that are resolution independent. However, the majority of bilevel methods are motivated and described in discrete space. Problems in discrete-valued settings, such as image segmentation, often require different techniques to optimize the lower-level cost function, although some recent work uses dual formulations to bridge this gap [145], [146].

The literature is inconsistent in how it refers to variables in machine learning problems. For consistency within this document, we define the following terms:

- **Hyperparameters**: Any adjustable parameters that are part of a model. Tuning parameters and model parameters are both sub-types of hyperparameters. This document uses $\boldsymbol{\gamma}$ to denote a vector of hyperparameters.
- **Tuning parameters**: Scalar parameters that weight terms in a cost function to determine the relative importance of each term. We use β to denote individual tuning parameters.
- **Model parameters**: Parameters, generally in vector or matrix form, that are used in the structure of a cost or loss function, typically as part of the regularization term. In the running example in the next section, the model parameters are typically filter coefficients, denoted $\boldsymbol{c}$.

We write vectors as column vectors and use bold to denote matrices (uppercase letters) and vectors (lowercase letters). Subscripts index vector elements, so x_i is the ith element in $\boldsymbol{x}$. For functions that are applied element-wise to vectors, we use notation following the Julia programming language [147], where $f.(\boldsymbol{x})$ denotes the function f applied element wise to its argument:

$$\boldsymbol{x} \in \mathbb{F}^N \implies f.(\boldsymbol{x}) = \begin{bmatrix} f(x_1) \\ \vdots \\ f(x_N) \end{bmatrix} \in \mathbb{F}^N.$$

We will often use this notation in combination with a transposed vector of ones to sum the result of a function applied element-wise to a vector, *i.e.*,

$$\mathbf{1}' f.(\boldsymbol{x}) = \sum_{i=1}^{N} f(x_i). \tag{7.1}$$

For example, the standard Euclidean norm is equivalent to $\mathbf{1}' f.(\boldsymbol{x})$ when $f(x) = |x|^2$ and and the vector 1-norm can be similarly written when $f(x) = |x|$. This notation is helpful for regularizers that do not correspond to norms. The field $\mathbb{F}$ can be either $\mathbb{R}$ or $\mathbb{C}$, depending on the application.

Convolution between a vector, $\boldsymbol{x}$, and a filter, $\boldsymbol{c}$, is denoted as $\boldsymbol{c} \circledast \boldsymbol{x}$. We assume all convolutions use circular boundary conditions. Thus, convolution is equivalent to multiplication with a square, circulant matrix:

$$\boldsymbol{c} \circledast \boldsymbol{x} = \boldsymbol{C}\boldsymbol{x}.$$

The conjugate mirror reversal of $\boldsymbol{c}$ is denoted as $\tilde{\boldsymbol{c}}$ and its application is equivalent to multiplying with the adjoint of $\boldsymbol{C}$:

$$\tilde{\boldsymbol{c}} \circledast \boldsymbol{x} = \boldsymbol{C}'\boldsymbol{x},$$

where the prime indicates the Hermitian transpose operation.

Finally, for partial derivatives, we use the notation that

$$\begin{aligned} \nabla_{\boldsymbol{x}} f(\boldsymbol{x}, \boldsymbol{y}) &= \frac{\partial f(\boldsymbol{x}, \boldsymbol{y})}{\partial \boldsymbol{x}} \in \mathbb{F}^N, \\ \nabla_{\boldsymbol{xy}} f(\boldsymbol{x}, \boldsymbol{y}) &= \left[\frac{\partial^2 f(\boldsymbol{x}, \boldsymbol{y})}{\partial x_i \partial y_j} \right] \in \mathbb{F}^{N \times M}, \text{ and} \\ \nabla_{\boldsymbol{xy}} f(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}) &= \nabla_{\boldsymbol{xy}} f(\boldsymbol{x}, \boldsymbol{y}) \Big|_{\boldsymbol{x}=\hat{\boldsymbol{x}}, \boldsymbol{y}=\hat{\boldsymbol{y}}} \in \mathbb{F}, \end{aligned} \tag{7.2}$$

where $f : \mathbb{F}^N \times \mathbb{F}^M \to \mathbb{F}$.

Tables 7.2 and 7.4 summarize our frequently used notation for variables and functions.

7.3 Defining a Bilevel Problem

This section introduces a generic bilevel problem; the next presents a specific bilevel problem that serves as a running example throughout Part 6.3. Later chapters discuss many of the ideas presented here more thoroughly. Our hope is that an early introduction to the formal problem motivates readers and that this section acts as a quick-reference guide to our notation.

Part 6.3 considers the image reconstruction problem where the goal is to form an estimate $\hat{\boldsymbol{x}} \in \mathbb{F}^N$ of a (vectorized) latent image, given a set of measurements $\boldsymbol{y} \in \mathbb{F}^M$. For denoising problems, $N = M$, but the two dimensions may differ significantly in more general image reconstruction problems. The forward operator, $\boldsymbol{A} \in \mathbb{F}^{M \times N}$ models the physics of the system such that one would expect $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$ in an ideal (noiseless) system. We focus on linear imaging systems here, but the concepts generalize readily to nonlinear forward models. When known (in a supervised training setting), we denote the true, underlying signal as $\boldsymbol{x}^{\text{true}} \in \mathbb{F}^N$. Most bilevel methods are supervised, but Section 12.1.2 presents a few examples of unsupervised bilevel methods.

We focus on model-based image reconstruction methods where the goal is to estimate $\boldsymbol{x}$ from $\boldsymbol{y}$ by solving an optimization problem of the form

$$\hat{\boldsymbol{x}} = \hat{\boldsymbol{x}}(\boldsymbol{\gamma}) = \underset{\boldsymbol{x} \in \mathbb{F}^N}{\operatorname{argmin}}\, \Phi(\boldsymbol{x}\,; \boldsymbol{\gamma}, \boldsymbol{y}). \tag{7.3}$$

To simplify notation, we drop $\boldsymbol{y}$ from the list of Φ arguments except where needed for clarity. The quality of the estimate $\hat{\boldsymbol{x}}$ can depend greatly on the choice of the hyperparameters $\boldsymbol{\gamma}$. Historically there have been numerous approaches pursued for choosing $\boldsymbol{\gamma}$, such as cross validation [148],

Variable	Dim	Description
$\boldsymbol{x}_j^{\text{true}}$	N	One of J clean, noiseless training signals. Often used in a supervised training set-up.
$\boldsymbol{A}$	$M \times N$	Forward operator for the system of interest.
$\boldsymbol{y}_j$	M	During the bilevel learning process, $\boldsymbol{y}_j$ refers to simulated measurements, where $\boldsymbol{y}_j = \boldsymbol{A}\boldsymbol{x}_j^{\text{true}} + \boldsymbol{n}_j$. Once $\boldsymbol{\gamma}$ is learned, $\boldsymbol{y}$ refers to collected measurements.
$\boldsymbol{n}_j$	N	A noise realization.
$\hat{\boldsymbol{x}}_j$	N	A reconstructed image.
$\boldsymbol{\gamma}$	R	The vector of parameters to learn using bilevel methods. This often includes $\boldsymbol{c}_k$ and/or β_k.
$\boldsymbol{c}_k$	S	One of K convolutional filters. A 2D filter might be $\sqrt{S} \times \sqrt{S}$.
$\tilde{\boldsymbol{c}}_k$	S	Conjugate mirror reversal of filter $\boldsymbol{c}_k$.
$\boldsymbol{C}_k$	$N \times N$	The convolution matrix such that $\boldsymbol{C}_k\boldsymbol{x} = \boldsymbol{c}_k \circledast \boldsymbol{x}$ and $\boldsymbol{C}_k'\boldsymbol{x} = \tilde{\boldsymbol{c}}_k \circledast \boldsymbol{x}$.
β_k	$\mathbb{R}$	The tuning parameter associated with $\boldsymbol{c}_k$.
β_0	$\mathbb{R}$	An overall regularization (tuning) parameter, appearing as e^{β_0} in (Ex).
$\boldsymbol{\Omega}$	$F \times N$	A matrix with filters in each row. For the stacked convolution matrices in (8.7) $F = KN$.
z	Varies	A sparse vector, often from $\boldsymbol{C}_k\boldsymbol{x}$.
ϵ	$\mathbb{R}_+$	Parameter used to define ϕ. Typically determines the amount of corner-rounding.
t	$0, \ldots, T$	Iteration counter for the lower-level optimization iterates, *e.g.*, $\boldsymbol{x}^{(t)}$ is the estimate of the lower-level optimization variable $\boldsymbol{x}$ at the tth iteration.
u	$0, \ldots, U$	Iteration counter for the upper-level optimization iterates, *e.g.*, $\boldsymbol{\gamma}^{(u)}$.

Table 7.2: Overview of frequently used symbols.

Function	Description
$\ell(\boldsymbol{\gamma}) \mapsto \mathbb{R}$ or $\ell(\boldsymbol{\gamma}, \boldsymbol{x}) \mapsto \mathbb{R}$	Upper-level loss function used as a fitness measure of $\boldsymbol{\gamma}$. Although ℓ is a function of $\boldsymbol{\gamma}$, it is often helpful to write it with two inputs, where typically $\boldsymbol{x} = \hat{\boldsymbol{x}}$.
$\Phi(\boldsymbol{x}\,; \boldsymbol{\gamma}) \mapsto \mathbb{R}$	Lower-level cost function used for reconstructing an image.
$R(\boldsymbol{x}) \mapsto \mathbb{R}$	Regularization function. Incorporates prior information about likely image characteristics.
$d(\boldsymbol{x}, \boldsymbol{y}) \mapsto \mathbb{R}$	Data-fit term.
$\phi(z) \mapsto \mathbb{R}$	Sparsity promoting function, *e.g.*, 0-norm, 1-norm, or corner-rounded 1-norm. Typically used in R.

Table 7.4: Overview of frequently used functions.

generalized cross validation [149], the discrepancy principle [150], and Bayesian methods [151], among others.

Bilevel methods provide a framework for choosing hyperparameters. A bilevel problem for learning hyperparameters $\boldsymbol{\gamma}$ has the following "double minimization" form:

$$\hat{\boldsymbol{\gamma}} = \underset{\boldsymbol{\gamma} \in \mathbb{F}^R}{\operatorname{argmin}} \, \underbrace{\ell(\boldsymbol{\gamma}\,;\, \hat{\boldsymbol{x}}(\boldsymbol{\gamma}))}_{\ell(\boldsymbol{\gamma})} \text{ where} \tag{UL}$$

$$\hat{\boldsymbol{x}}(\boldsymbol{\gamma}) = \underset{\boldsymbol{x} \in \mathbb{F}^N}{\operatorname{argmin}} \, \Phi(\boldsymbol{x}\,;\, \boldsymbol{\gamma}). \tag{LL}$$

Fig. 7.1 depicts a generic bilevel problem for image reconstruction. The upper-level (UL) loss function, $\ell : \mathbb{R}^R \times \mathbb{F}^N \mapsto \mathbb{R}$, quantifies how (not) good is a vector $\boldsymbol{\gamma}$ of learnable parameters. The upper-level depends on the solution to the lower-level (LL) cost function, Φ, which depends on $\boldsymbol{\gamma}$. The upper-level can also be called the outer optimization, with the lower-level being the inner optimization. Another terminology is leader-follower, as the minimizer of the lower-level follows where the upper-level loss leads. We will also write the upper-level loss function with a single parameter as $\ell(\boldsymbol{\gamma}) := \ell(\boldsymbol{\gamma}\,;\, \hat{\boldsymbol{x}}(\boldsymbol{\gamma}))$.

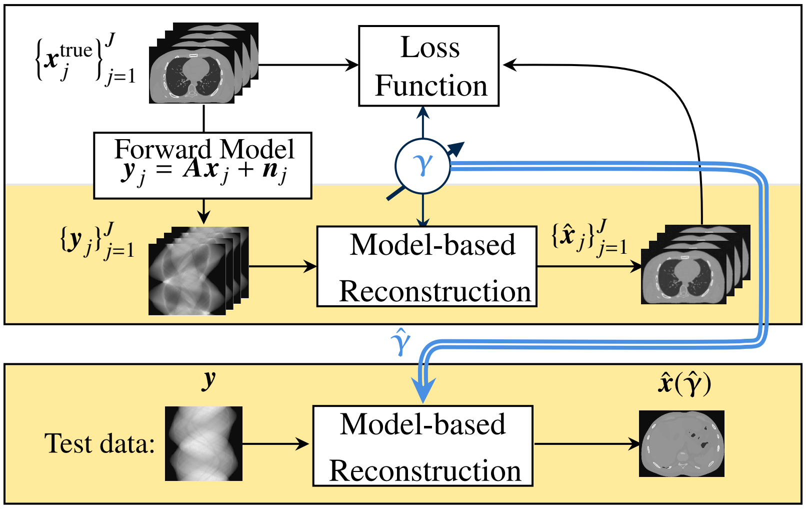

$\{\boldsymbol{x}_j^{\text{true}}\}_{j=1}^J$
Loss Function
Forward Model
$\boldsymbol{y}_j = \boldsymbol{A}\boldsymbol{x}_j + \boldsymbol{n}_j$
γ
$\{\boldsymbol{y}_j\}_{j=1}^J$
Model-based Reconstruction
$\{\hat{\boldsymbol{x}}_j\}_{j=1}^J$
$\hat{\gamma}$
$\boldsymbol{y}$
$\hat{\boldsymbol{x}}(\hat{\boldsymbol{\gamma}})$
Test data:
Model-based Reconstruction

Figure 7.1: Depiction of a typical bilevel problem for image reconstruction, illustrated using XCAT phantom from [152]. The upper box represents the training process, with the upper-level loss and lower-level cost function. During training, one minimizes the upper-level loss with respect to a vector of parameters, $\boldsymbol{\gamma}$, that are used in the image reconstruction task. Once learned, $\hat{\boldsymbol{\gamma}}$ is typically deployed in the same image reconstruction task, shown in the lower box.

We write the lower-level cost as an optimization problem with "argmin" and thus implicitly assume that Φ has unique minimizer, $\hat{\boldsymbol{x}}$. The lower-level is guaranteed to have a unique minimizer when Φ is a strictly convex function of $\boldsymbol{x}$. (See Section 10.1 for more discussion of this point). More generally, there may be a set of lower-level minimizers, each having some possibly distinct upper-level loss function value. For more discussion, [8] defines optimistic and pessimistic versions of the bilevel problem for the case of multiple lower-level solutions.

Bilevel methods typically use training data. Specifically, one often assumes that a given set of J good quality images $\boldsymbol{x}_1^{\text{true}}, \dots, \boldsymbol{x}_J^{\text{true}} \in \mathbb{F}^N$ are representative of the images of interest in a given application. (For simplicity of notation we assume the training images have the same size, but they can have different sizes in practice.) We typically generate corresponding simulated measurements for each training image using the imaging system model:

$$\boldsymbol{y}_j = \boldsymbol{A}\boldsymbol{x}_j^{\text{true}} + \boldsymbol{n}_j, \quad j = 1, \dots, J, \tag{7.4}$$

where $\boldsymbol{n}_j \in \mathbb{F}^M$ denotes an appropriate random noise realization[1]. In (7.4), we add one noise realization to each of the J images; in practice one could add multiple noise realizations to each $\boldsymbol{x}_j^{\text{true}}$ to augment the training data. We then use the training pairs $(\boldsymbol{x}_j^{\text{true}}, \boldsymbol{y}_j)$ to learn a good value of $\boldsymbol{\gamma}$. After those parameters are learned, we reconstruct subsequent test images using (7.3) with the learned hyperparameters $\hat{\boldsymbol{\gamma}}$.

An alternative to the upper-level formulation (UL) is the following stochastic formulation of bilevel learning:

$$\hat{\boldsymbol{\gamma}} = \underset{\boldsymbol{\gamma} \in \mathbb{F}^R}{\operatorname{argmin}} \; \underbrace{\mathbb{E}\left[\ell(\boldsymbol{\gamma})\right]}_{\approx \frac{1}{J}\sum_{j=1}^{J} \ell(\boldsymbol{\gamma}\,;\,\hat{\boldsymbol{x}}_j(\boldsymbol{\gamma}))} \tag{7.5}$$

$$\text{where } \hat{\boldsymbol{x}}_j(\boldsymbol{\gamma}) = \underset{\boldsymbol{x} \in \mathbb{F}^N}{\operatorname{argmin}} \, \Phi(\boldsymbol{x}\,;\boldsymbol{\gamma}, \boldsymbol{y}_j). \tag{7.6}$$

The expectation, taken with respect to the training data and noise distributions, is typically approximated as a sample mean over J training examples.

The definition of bilevel methods used in (UL) is not universal in the literature. In some works, bilevel methods refer to nested optimization problems with two levels, even when the two levels result from reformulating a single-level problem, *e.g.*, [153]. That definition is much more encompassing, and includes primal-dual reformulations, Lagrangian reformulations of constrained optimization problems, and alternating methods that introduce then minimize over an auxiliary variable.

Another term in the literature, sometimes used interchangeably with a bilevel problem, is a mathematical program with equilibrium constraints (MPEC). As shown in Section 10.1, many bilevel optimization methods start by transforming the two-level problem into an equivalent single-level problem by replacing the lower-level optimization with a set of constraints based on optimally

[1] A more general system model allows the noise to depend on the data and system model, *i.e.*, $\boldsymbol{n}_j(\boldsymbol{A}, \boldsymbol{x}_j)$. This generality is needed for applications with certain noise distributions such as Poisson noise.

conditions. Bilevel problems are thus a subset of MPECs. MPECs are generally challenging due to their non-convex nature; even when the lower-level cost function is convex, the upper-level loss function is rarely convex. Importantly, $\ell(\cdot,\cdot)$ is often convex with respect to both arguments. However, $\ell(\boldsymbol{\gamma}) = \ell(\boldsymbol{\gamma}\,;\hat{\boldsymbol{x}}(\boldsymbol{\gamma}))$ is generally non-convex in $\boldsymbol{\gamma}$ due to how the lower-level minimizer depends on $\boldsymbol{\gamma}$. There is a large literature on MPEC problems, *e.g.*, [8], [154], [155], and on non-convex optimization more generally [156]. Bilevel methods are one sub-field in this large literature.

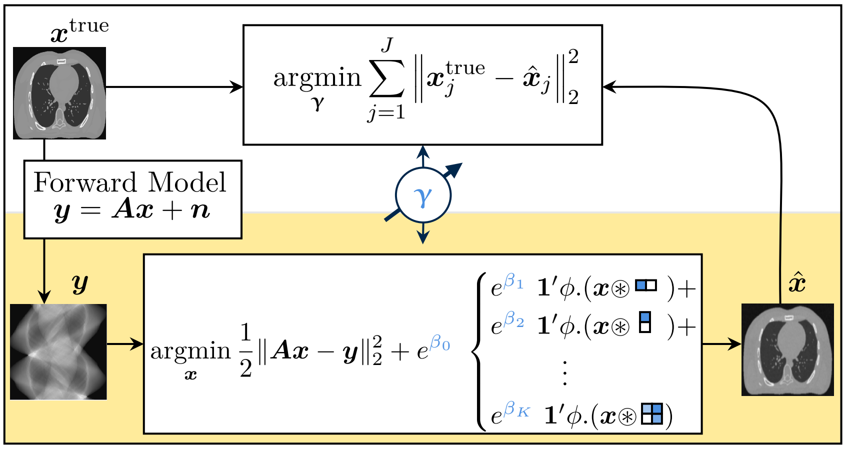
7.4 Running Example

To offer a concrete example, Part 6.3 will frequently refer to the following running example (Ex), a filter learning bilevel problem:

$$\hat{\boldsymbol{\gamma}} = \operatorname*{argmin}_{\boldsymbol{\gamma}\in\mathbb{F}^R} \frac{1}{2}\|\hat{\boldsymbol{x}}(\boldsymbol{\gamma}) - \boldsymbol{x}^{\text{true}}\|_2^2, \text{ where}$$

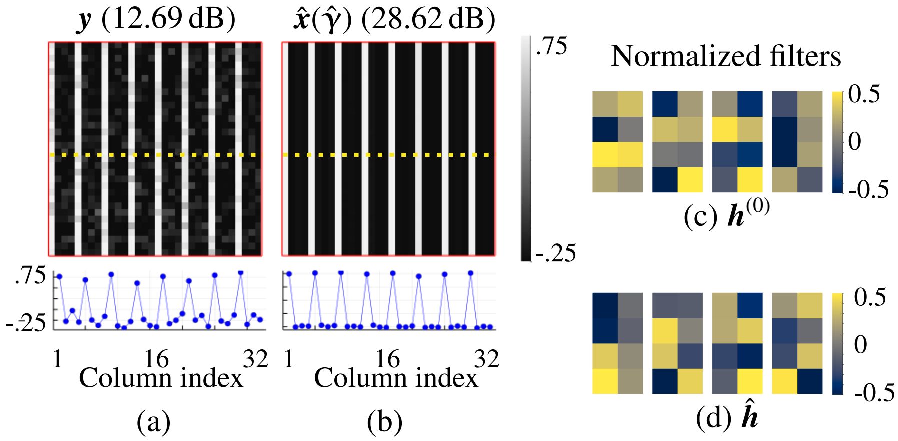
$$\hat{\boldsymbol{x}}(\boldsymbol{\gamma}) = \operatorname*{argmin}_{\boldsymbol{x}\in\mathbb{F}^N} \frac{1}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2 + e^{\beta_0}\sum_{k=1}^{K} e^{\beta_k}\mathbf{1}'\phi.(\boldsymbol{c}_k \circledast \boldsymbol{x}; \epsilon), \qquad \text{(Ex)}$$

where $\boldsymbol{\gamma} \in \mathbb{F}^R$ contains all variables that we wish to learn: the filter coefficients $\boldsymbol{c}_k \in \mathbb{F}^S$ and tuning parameters $\beta_k \in \mathbb{R}$ for all $k \in [1, K]$. We include an auxiliary tuning parameter, $\beta_0 \in \mathbb{R}$, for easier comparison to other models. Fig. 7.2 depicts the running example and Fig. 7.3 shows example learned filters for a toy training image. Ref. [157] demonstrates how a spectral analysis of learned filters and penalty functions can be interpreted to provide insight into real-world problems.



$\boldsymbol{x}^{\text{true}}$
$\operatorname*{argmin}_{\boldsymbol{\gamma}} \sum_{j=1}^{J} \left\|\boldsymbol{x}_j^{\text{true}} - \hat{\boldsymbol{x}}_j\right\|_2^2$
Forward Model
$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{n}$
$\boldsymbol{\gamma}$
$\boldsymbol{y}$
$\hat{\boldsymbol{x}}$
$\operatorname*{argmin}_{\boldsymbol{x}} \frac{1}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2 + e^{\beta_0}$
$e^{\beta_1}\ \mathbf{1}'\phi.(\boldsymbol{x}\circledast\ )+$
$e^{\beta_2}\ \mathbf{1}'\phi.(\boldsymbol{x}\circledast\ )+$
$\vdots$
$e^{\beta_K}\ \mathbf{1}'\phi.(\boldsymbol{x}\circledast)$

Figure 7.2: Bilevel problem in (Ex). The vector of learnable hyperparameters, $\boldsymbol{\gamma}$, includes the tuning parameters, β_k, and the filter coefficients, $\boldsymbol{c}_k$, shown as example filters. Although we only consider learning filters of a single size, the figure depicts how the framework easily extends to 2d filters of different sizes.



(a) (b) (c) $\boldsymbol{h}^{(0)}$ (d) $\hat{\boldsymbol{h}}$

Figure 7.3: Example learned filters for a simple training image, normalized for easier visualization. The true image is zero-mean and repeats three columns of signal value -0.25 and one column of signal value 0.75. (a) Noisy image. The lower plot shows a profile of one row of the image (marked by a dotted line). The signal-to-noise ratio, as defined in (8.14), is given in parenthesis. (b) The denoised image using learned filters as in (Ex). (c) Randomly initialized filters for the bilevel method ($K = 4$ and $S = 4 \cdot 2$). (d) Corresponding learned filters. As expected based on the training image, the learned filters primarily involve vertical differences. Appendix F provides further details including the regularization strength of each learned filter.

The learnable hyperparameters can also include the sparsifying function ϕ, its corner rounding parameter ϵ, the forward model $\boldsymbol{A}$, or some aspect of the data-fit term. For example, [157], [158] learn the regularization functional and [159], [160] learn part of the forward model. Such examples are relatively rare in the bilevel methods literature to date.

Unlike many learning problems (see examples in Section 9.1), the running example (Ex) does not include any constraints on $\boldsymbol{\gamma}$. Learned filters should be those that are best at the given task, where "best" is defined by the upper-level loss function. Therefore, a zero mean or norm constraint is not generally required, though some authors have found such constraints helpful, *e.g.*, [161], [162]. Following previous literature, *e.g.*, [163], the tuning parameters in (Ex) are written in terms of an exponential function to ensure positivity. One could re-write (Ex) without this exponentiation "trick" and then add a non-negativity constraint to the upper-level problem; most of the methods reviewed in Chapter 10 generalize to this common variation by substituting gradient methods for projected gradient methods. The exponential function means that the effective tuning parameter, e^{β_k}, cannot exactly reach zero. However, one can introduce a pruning strategy to remove or re-initialize filters that have an effective tuning parameter below a given threshold.

In (Ex), we drop the sum over J training images for simplicity; the methods easily extend to multiple training signals. For ease of notation, we further simplify by considering $\boldsymbol{c}_k$ to be of length S for all k, *e.g.*, a 2D filter might be $\sqrt{S} \times \sqrt{S}$. In practice, the filters may be of different lengths with minimal impact on the bilevel methods.

The function ϕ in (Ex) is a sparsity-promoting function. If we were to choose $\phi(z) = |z|$, then the regularizer would involve 1-norm terms of the type common in compressed sensing formulations:

$$\mathbf{1}'\phi.(\boldsymbol{c}_k \circledast \boldsymbol{x}) = \|\boldsymbol{c}_k \circledast \boldsymbol{x}\|_1 .$$

However, to satisfy differentiability assumptions (see Section 10.1), we will often consider ϕ to denote the following "corner rounded" 1-norm having the shape of a hyperbola with the corresponding first and second derivative:

$$\begin{aligned} \phi(z) &= \sqrt{z^2 + \epsilon^2} \\ \dot{\phi}(z) &= \frac{z}{\sqrt{z^2 + \epsilon^2}} \in [0, 1) \\ \ddot{\phi}(z) &= \frac{\epsilon^2}{\left(z^2 + \epsilon^2\right)^{3/2}} \in (0, \frac{1}{\epsilon}], \end{aligned} \tag{CR1N}$$

where ϵ is a small, relative to the expected range of z, parameter that controls the amount of corner rounding. (Here, we use a dot over the function rather than ∇ to indicate a derivative because ϕ has a scalar argument.)

7.5 Conclusion

Bilevel methods for selecting hyperparameters offer many benefits. Previous papers motivate them as a principled way to approach hyperparameter optimization [141], [164], as a task-based approach to learning [144], [158], [165], and/or as a way to combine the data-driven improvements from learning methods with the theoretical guarantees and explainability provided by cost function-based approaches [143], [161], [166]. A corresponding drawback of bilevel methods are their computational cost; see Chapter 10 for further discussion.

The task-based nature of bilevel methods is a particularly important advantage; Chapter 9 and 11 exemplify why by comparing the bilevel problem to single-level, non-task-based approaches for learning sparsifying filters. Task-based refers to the hyperparameters being learned based on how well they work in the lower-level cost function–the image reconstruction task in our running example. The learned hyperparameters can also adapt to the training dataset and noise characteristics. The task-based nature yields other benefits, such as making constraints or regularizers on the hyperparameters generally unnecessary; Section 12.1.2 presents some exceptions and [141] further discusses bilevel methods for applications with constraints.

There are three main elements to a bilevel approach. First, the lower-level cost function in a bilevel problem defines a goal, such as image reconstruction, including what hyperparameters can be learned, such as filters for a sparsifying regularizer. Section 8.1 provides background on this element specifically for image reconstruction tasks, such as the one in (Ex). Section 12.1.1 reviews example cost functions used in bilevel methods.

Second, the upper-level loss function determines how the hyperparameters should be evaluated. While the squared error loss function in the running example is a common choice, Section 8.2 discusses other loss functions based on supervised and unsupervised image quality metrics. Section 12.1.2 then reviews example loss functions used in bilevel methods.

While less apparent in the written optimization problem, the third main element for a bilevel problem is the optimization approach, especially for the upper-level problem. Section 8.4 briefly discusses various hyperparameter optimization strategies, then Chapter 10 present multiple gradient-based bilevel optimization strategies. Throughout Part 6.3, we refer to the running example to show how the bilevel optimization strategies apply.

CHAPTER 8

Background on Image Reconstruction and Hyperparameter Optimization

This chapter provides background for the three main elements of the bilevel method: the lower-level cost function, the upper-level loss function, and the optimization method. This chapter is presented in Ch. 2 and 3 of [11]:

> C. Crockett and J. A. Fessler, "Bilevel methods for image reconstruction," *Foundations and Trends® in Signal Processing*, vol. 15, no. 2-3, pp. 121–289, May 5, 2022, ISSN: 1932-8346, 1932-8354. DOI: 10.1561/2000000111

8.1 Cost Functions and Image Reconstruction

This dissertation focuses on bilevel problems having image reconstruction as the lower-level problem. Image reconstruction involves undoing any transformations inherent in an imaging system, *e.g.*, a camera or CT scanner, and removing measurement noise, *e.g.*, thermal and shot noise, to realize an image that captures an underlying object of interest, *e.g.*, a patient's anatomy. Fig. 8.1 shows an example image reconstruction pipeline for CT data. The following sections formally define image reconstruction, discuss why regularization is important, and overview common approaches to regularization.

8.1.1 Image Reconstruction

Although the true object is in continuous space, image reconstruction is almost always performed on sampled, discretized signals [167]. Without going into detail of the discretization process, we define $\boldsymbol{x}^{\text{true}} \in \mathbb{F}^N$ as the "true," discrete signal. The goal of image reconstruction is to recover an estimate $\hat{\boldsymbol{x}} \approx \boldsymbol{x}^{\text{true}}$ given corrupted measurements $\boldsymbol{y} \in \mathbb{F}^M$. Although we define the signal as a one-dimensional vector for notational convenience, the mathematics generalize to arbitrary dimensions.

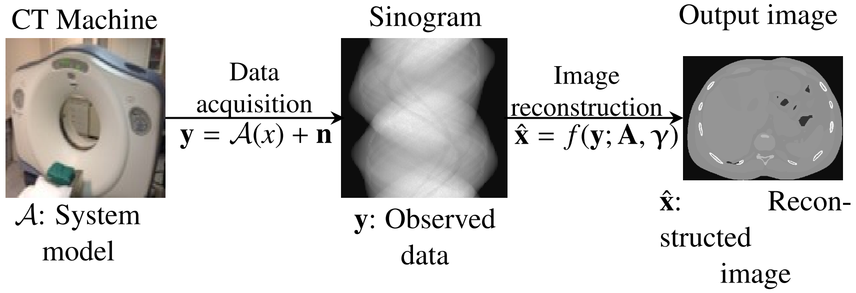

CT Machine
Sinogram
Output image
Data acquisition
$\mathbf{y} = \mathcal{A}(x) + \mathbf{n}$
Image reconstruction
$\hat{\mathbf{x}} = f(\mathbf{y}; \mathbf{A}, \gamma)$
$\mathcal{A}$: System model
$\mathbf{y}$: Observed data
$\hat{\mathbf{x}}$: Reconstructed image

Figure 8.1: Example image reconstruction pipe-line, illustrated using XCAT phantom from [152]. Here $\mathcal{A}$ denotes the actual physical mapping of the imaging system and $\boldsymbol{A}$ denotes the numerical system matrix used for reconstruction.

To find $\hat{\boldsymbol{x}}$, image reconstruction involves minimizing a cost function, $\Phi(\boldsymbol{x}\,;\boldsymbol{\gamma})$, with two terms:

$$\hat{\boldsymbol{x}} = \operatorname*{argmin}_{\boldsymbol{x}\in\mathbb{F}^N} \underbrace{\overbrace{d(\boldsymbol{x}\,;\boldsymbol{y})}^{\text{Data-fit}} + \beta\overbrace{R(\boldsymbol{x}\,;\boldsymbol{\gamma})}^{\text{Regularizer}}}_{\Phi(\boldsymbol{x}\,;\gamma)} \tag{8.1}$$

The first term, $d(\boldsymbol{x}\,;\boldsymbol{y})$, is a data-fit term that captures the physics of the ideal (noiseless) system using the matrix $\boldsymbol{A} \in \mathbb{F}^{M\times N}$; that matrix models the physical system such that we expect an observation, $\boldsymbol{y}$, to be $\boldsymbol{y} \approx \boldsymbol{A}\boldsymbol{x}$.

The most common data-fit term penalizes the square Euclidean norm of the "measurement error," $d(\boldsymbol{x}\,;\boldsymbol{y}) = ||\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}||_2^2$. This intuitive data-fit term can be derived from a maximum likelihood perspective, assuming a white Gaussian noise distribution [168]. Using the system model (7.4) and assuming the noise is normally distributed with zero-mean and variance σ^2, the maximum likelihood estimate $\hat{\boldsymbol{x}}_{\text{MLE}}$ is the image that is most likely given the observation $\boldsymbol{y}$, *i.e.*,

$$\hat{\boldsymbol{x}}_{\text{MLE}} = \operatorname*{argmax}_{\boldsymbol{x}\in\mathbb{F}^N} \text{Prob}(\boldsymbol{x}\,;\,\boldsymbol{y}, \sigma^2).$$

Substituting the assumed Gaussian distribution (and ignoring constants independent of $\boldsymbol{x}$),

$$\hat{\boldsymbol{x}}_{\text{MLE}} = \operatorname*{argmax}_{\boldsymbol{x}\in\mathbb{F}^N} e^{\frac{-1}{2\sigma^2}||\boldsymbol{A}\boldsymbol{x}-\boldsymbol{y}||^2} = \operatorname*{argmin}_{\boldsymbol{x}\in\mathbb{F}^N} \frac{1}{2}\,||\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}||^2 = \boldsymbol{A}^+\boldsymbol{y},$$

where $\boldsymbol{A}^+$ is the pseudo-inverse of $\boldsymbol{A}$.

The regularization term in (8.1) can be motivated by maximum *a posteriori* probability (MAP) estimation [168]. Rather than maximizing the likelihood of $\boldsymbol{x}$, the MAP estimate $\hat{\boldsymbol{x}}_{\text{MAP}}$ maximizes the conditional probability of $\boldsymbol{x}$ given the observation $\boldsymbol{y}$

$$\begin{aligned}\hat{\boldsymbol{x}}_{\mathrm{MAP}} &= \underset{\boldsymbol{x}\in\mathbb{F}^N}{\operatorname{argmax}}\,\mathrm{Prob}(\boldsymbol{x}|\boldsymbol{y}) \\ &= \underset{\boldsymbol{x}\in\mathbb{F}^N}{\operatorname{argmax}}\,\mathrm{Prob}(\boldsymbol{y}|\boldsymbol{x})\mathrm{Prob}(\boldsymbol{x})\end{aligned}$$

by Bayes theorem. A MAP estimator requires assuming a prior distribution on $\boldsymbol{x}$. Taking the logarithm and substituting the assumed Gaussian distribution for $\mathrm{Prob}(\boldsymbol{y}\,|\,\boldsymbol{x}\,;\,\sigma^2)$ yields

$$\hat{\boldsymbol{x}}_{\mathrm{MAP}} = \underset{\boldsymbol{x}\in\mathbb{F}^N}{\operatorname{argmin}}\,\frac{1}{2\sigma^2}\,\|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{y}\|^2 - \log\left(\mathrm{Prob}(\boldsymbol{x})\right),$$

where the regularization term in (8.1) comes from the log probability of $\boldsymbol{x}$, *i.e.*, the two are equivalent when one assumes the probability model $\mathrm{Prob}(\boldsymbol{x}) = \frac{1}{Z(\gamma)}\exp\{-R(\boldsymbol{x}\,;\,\gamma)\}$, where $Z(\gamma)$ is a scalar such that the probability integrates to one. The MLE estimate is equivalent to the MAP estimate when the prior on $\boldsymbol{x}$ is an (unbounded) “uniform” distribution.

While MAP estimation provides a useful perspective, common regularizers do not correspond to proper probability models. Further, the connection between the regularization perspective and the Bayesian perspective is simplest when the parameters γ are given. To learn γ, Bayesian formulations must consider the partition function $Z(\gamma)$; that complication is avoided for bilevel formulations using a regularized lower-level problem.

Many image reconstruction problems have linear system models. In image denoising problems, one takes $\boldsymbol{A} = \boldsymbol{I}$. For image inpainting, $\boldsymbol{A}$ is a diagonal matrix of 1’s and 0’s, where the 0’s correspond to sample indices of missing data [169]. In MRI, the system matrix is often approximated as a diagonal matrix times a discrete Fourier transform matrix, though more accurate models are often needed [170]. In some settings, one can learn $\boldsymbol{A}$ [171], or at least parts of $\boldsymbol{A}$ [172], as part of the estimation process. Although the bilevel method generalizes to learning $\boldsymbol{A}$, the majority of papers in the field assume $\boldsymbol{A}$ is known; Chapter 12.1 discusses a few exceptions.

Using the system model (7.4), if $\boldsymbol{n}$ were known and $\boldsymbol{A}$ were invertible, we could simply compute $\hat{\boldsymbol{x}} = \boldsymbol{x}^{\mathrm{true}} = \boldsymbol{A}^{-1}(\boldsymbol{y}-\boldsymbol{n})$. However, $\boldsymbol{n}$ is random and, while we may be able to model its characteristics, we never know it exactly. Further, the system matrix, $\boldsymbol{A}$, is often not invertible because the reconstruction problem is frequently under-determined, with fewer knowns than unknowns ($M < N$). Therefore, we must include prior assumptions about $\boldsymbol{x}^{\mathrm{true}}$ to make the problem feasible. These assumptions about $\boldsymbol{x}^{\mathrm{true}}$ are captured in the second, regularization term in (8.1), which depends on γ. The following section further discusses regularizers.

In sum, image reconstruction involves finding $\hat{\boldsymbol{x}}$ that matches the collected data *and* satisfies a set of prior assumptions. The data-fit term encourages $\hat{\boldsymbol{x}}$ to be a good match for the data; without this term, there would be no need to collect data. The regularization term encourages $\hat{\boldsymbol{x}}$ to match the prior assumptions. Finally, the tuning parameter, β, controls the relative importance of the two terms. The cost function can be minimized using different optimization techniques depending on the form of each term.

This section is a very short overview of image reconstruction methods. See [140] for a more thorough review of biomedical image reconstruction.

8.1.2 Sparsity-Based Regularizers

The regularization, or prior assumption, term in (8.1) often involves assumptions about sparsity [7], [173]. The basic idea behind sparsity-based regularization is that the true signal is sparse in some representation, while the noise or corruption is not. Thus, one can use the representation to separate the noise and signal, and then keep only the sparse signal component. In fact, a known sparsifying representation for a signal can help to "reconstruct a signal from far fewer measurements than required by the Shannon-Nyquist sampling theorem" [173].

The regularization design problem therefore requires determining what representation best sparsifies the signal. There are two main types of sparsity-based regularizers corresponding to two representational assumptions: synthesis and analysis [139], [168]; Fig. 8.2 depicts both. While both are popular, we concentrate on analysis regularizers, which are more widely represented in the bilevel image reconstruction literature. This section briefly compares the analysis and synthesis formulations. Here we simplify the formulas by considering $\boldsymbol{A} = \boldsymbol{I}$; the discussion generalizes to reconstruction by including $\boldsymbol{A}$. For more thorough discussions of analysis and synthesis regularizers, see [139], [168], [174].

8.1.2.1 Synthesis Regularizers

Synthesis regularizers model a signal being composed of building blocks, or "atoms." Small subsets of the atoms span a low dimensional subspace and the sparsity assumption is that the signal

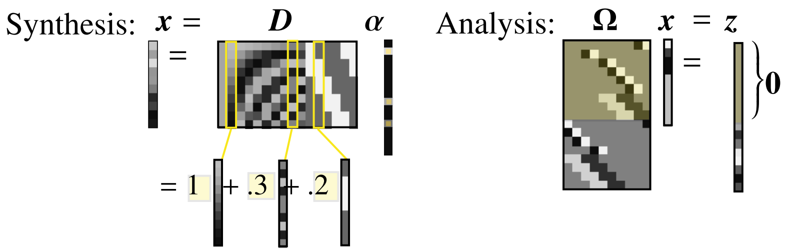

Synthesis: $\boldsymbol{x}$ = $\boldsymbol{D}$ $\boldsymbol{\alpha}$
=
= 1 + .3 + .2
Analysis: $\boldsymbol{\Omega}$ $\boldsymbol{x}$ = $\boldsymbol{z}$
=
$\boldsymbol{0}$

Figure 8.2: Depiction of synthesis and analysis sparsity. Under the synthesis model of sparsity (left), $\boldsymbol{x}$ is a linear combination of a few dictionary atoms. The dictionary, $\boldsymbol{D}$, is typically wide, with more atoms (columns) than elements in $\boldsymbol{x}$. Under the analysis model of sparsity (right), $\boldsymbol{x}$ is orthogonal to many filters. The filter matrix, $\boldsymbol{\Omega}$, is typically tall, with more filters (rows) than elements in $\boldsymbol{x}$.

requires using only a few of the atoms. More formally, the synthesis model is $\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{n}$, where the signal $\boldsymbol{x} = \boldsymbol{D}\boldsymbol{z}$ and $\boldsymbol{z}$ is a sparse vector. The columns of $\boldsymbol{D} \in \mathbb{F}^{N \times K}$ contain contain the K dictionary atoms and form a low dimensional subspace for the signal. If $\boldsymbol{D}$ is a wide matrix ($N < K$), the dictionary is over-complete and it is easier to represent a wide range of signals with a given number of dictionary atoms. The dictionary is complete when $\boldsymbol{D}$ is square (and full rank) and under-complete if $\boldsymbol{D}$ is tall (an uncommon choice).

Assuming one knows or has already learned $\boldsymbol{D}$, one can use the sparsity synthesis assumption to denoise a noisy signal $\boldsymbol{y}$ by optimizing

$$\hat{\boldsymbol{x}} = \boldsymbol{D} \cdot \underbrace{(\operatorname*{argmin}_{\boldsymbol{z} \in \mathbb{F}^K} \frac{1}{2} \|\boldsymbol{D}\boldsymbol{z} - \boldsymbol{y}\|^2 + \mathbf{1}'\phi.(\boldsymbol{z}))}_{\hat{\boldsymbol{z}}}. \tag{8.2}$$

The estimation procedure involves finding the sparse codes, $\hat{\boldsymbol{z}}$, from which the image is synthesized via $\hat{\boldsymbol{x}} = \boldsymbol{D}\hat{\boldsymbol{z}}$. Common sparsity-inducing functions, ϕ, are the absolute value or a non-zero indicator function, equivalent to the 1-norm and 0-norm respectively. The 2-norm is occasionally used in the regularizer, but it does not yield true sparse codes and it over-penalizes large values [175].

As written in (8.2), the synthesis formulation constrains the signal, $\boldsymbol{x}$, to be in the range of $\boldsymbol{D}$. This "strict synthesis" model can be undesirable in some applications, *e.g.*, when one is not confident in the quality of the dictionary. An alternative formulation is

$$\begin{aligned} \hat{\boldsymbol{x}} &= \operatorname*{argmin}_{\boldsymbol{x} \in \mathbb{F}^N} \frac{1}{2} \|\boldsymbol{x} - \boldsymbol{y}\|^2 + \beta R(\boldsymbol{x}), \\ R(\boldsymbol{x}) &= \min_{\boldsymbol{z} \in \mathbb{F}^K} \frac{1}{2} \|\boldsymbol{x} - \boldsymbol{D}\boldsymbol{z}\|^2 + \mathbf{1}'\phi.(\boldsymbol{z}), \end{aligned} \tag{8.3}$$

which no longer constrains $\boldsymbol{x}$ to be exactly in the range of $\boldsymbol{D}$. One can also learn $\boldsymbol{D}$ while solving (8.3) [176].

Both synthesis denoising forms have equivalent sparsity constrained versions; one can replace $\mathbf{1}'\phi.(\boldsymbol{z})$ with a characteristic function that is 0 within some desired set and infinite outside it, *e.g.*,

$$\psi(\boldsymbol{z}) = \begin{cases} 0 & \text{if } \|\boldsymbol{z}\|_0 \leq \kappa \\ \infty & \text{else,} \end{cases} \tag{8.4}$$

for some sparsity constraint given by the hyperparameter $\kappa \in \mathbb{N}$.

See [175], [177] for discussions of when the synthesis model can guarantee accurate recovery of signals. The minimization problem in (8.3) is called sparse coding and is closely related to the LASSO problem [178]. One can think of the entire dictionary $\boldsymbol{D}$ as a hyperparameter that can be learned with a bilevel method [179].

8.1.2.2 Analysis Regularizers

Analysis regularizers model a signal as being sparsified when mapped into another vector space by a linear transformation, often represented by a set of filters. More formally, an analysis model assumes the signal satisfies $\mathbf{\Omega}\boldsymbol{x} = \boldsymbol{z}$ for a sparse coefficient vector $\boldsymbol{z}$. Often the rows of the matrix $\mathbf{\Omega} \in \mathbb{F}^{K\times N}$ are thought of as filters and the rows of $\mathbf{\Omega}$ where $[\mathbf{\Omega}\boldsymbol{x}]_k = 0$ span a subspace to which $\boldsymbol{x}$ is orthogonal. The analysis operator is called over-complete if $\mathbf{\Omega}$ is tall ($N < K$), complete if $\mathbf{\Omega}$ is square (and full rank), and under-complete if $\mathbf{\Omega}$ is wide.

A particularly common analysis regularizer is based on a discretized version of total variation (TV) [180], and uses finite difference filters (or, more generally, filters that approximate higher-order derivatives). The finite difference filters sparsify any piece-wise constant (flat) regions in the signal, leaving the edges that are often approximately sparse in natural images. Other common analysis regularizers include the discrete Fourier transform (DFT), curvelets, and wavelet transforms [181].

The literature is less consistent in analysis regularizer vocabulary, and $\mathbf{\Omega}$ has been called an analysis dictionary, an analysis operator, a filter matrix, and a cosparse operator. The term "cosparse" comes from the sparsity holding in the codomain of the transformation $T\{\boldsymbol{x}\} = \mathbf{\Omega}\boldsymbol{x}$. The cosparsity of $\boldsymbol{x}$ with respect to $\mathbf{\Omega}$ is the number of zeros in $\mathbf{\Omega}\boldsymbol{x}$ or $K - ||\mathbf{\Omega}\boldsymbol{x}||_0$ [174]. Correspondingly, "cosupport" describes the indices of the rows where $\mathbf{\Omega}\boldsymbol{x} = 0$. We find the phrase "analysis operator" intuitive for general $\mathbf{\Omega}$'s and "filter matrix" more descriptive when referring to the specific (common) case when the rows of $\mathbf{\Omega}$ are dictated by a set of convolutional filters.

Assuming one knows, or has already learned, $\mathbf{\Omega}$, one can use the analysis sparsity assumption to denoise a noisy signal, $\boldsymbol{y}$, by optimizing

$$\hat{\boldsymbol{x}} = \underset{\boldsymbol{x}\in\mathbb{F}^N}{\operatorname{argmin}} \frac{1}{2} ||\boldsymbol{x} - \boldsymbol{y}||^2 + \beta \mathbf{1}' \phi.(\mathbf{\Omega}\boldsymbol{x}). \tag{8.5}$$

An alternative version is

$$\begin{aligned} \hat{\boldsymbol{x}} &= \underset{\boldsymbol{x}\in\mathbb{F}^N}{\operatorname{argmin}} \frac{1}{2} ||\boldsymbol{x} - \boldsymbol{y}||^2 + \beta R(\boldsymbol{x}) \\ R(\boldsymbol{x}) &= \min_{\boldsymbol{z}\in\mathbb{F}^K} \frac{1}{2} ||\mathbf{\Omega}\boldsymbol{x} - \boldsymbol{z}||^2 + \mathbf{1}' \phi.(\boldsymbol{z}). \end{aligned} \tag{8.6}$$

As in the synthesis case, both analysis formulations have equivalent sparsity-constrained forms using a characteristic function as in (8.4).

See [181] for an error bound on the estimated signal $\hat{\boldsymbol{x}}$ when using a 1-norm as the regularization function.

8.1.2.3 Comparing Analysis and Synthesis Approaches

The analysis and synthesis models are equivalent when the dictionary and analysis operator are invertible, with $\boldsymbol{D} = \boldsymbol{\Omega}^{-1}$ [168]. Furthermore, in the denoising scenario where the system matrix $\boldsymbol{A}$ is identity, the two are almost equivalent in the under-complete case, with the lack of full equivalence stemming from the analysis form not constraining $\boldsymbol{x}$ to be in the range space $\boldsymbol{D}$ [168].

As shown in [Example 3.1]chambolle:2016:introductioncontinuousoptimization, the analysis model can more generally be related to a Lasso-like problem using Legendre-Fenchel conjugates and convex duality. Appendix C briefly reviews duality and the main results from primal-dual analysis used throughout this dissertation. Considering the analysis operator learning problem (8.5), when the sparsity promoting function ϕ is convex and $\phi(z) < \infty$ for some z, the dual problem corresponding to (8.5) is

$$\hat{\boldsymbol{d}} = \underset{\boldsymbol{d} \in \mathbb{F}^K}{\operatorname{argmin}} \frac{1}{2} \|\boldsymbol{\Omega}' \boldsymbol{d} - \boldsymbol{y}\|^2 + \phi^*(\boldsymbol{d}),$$

where $\boldsymbol{d}$ is the dual variable and ϕ^* is the conjugate function of ϕ. (The primal solution $\hat{\boldsymbol{x}}$ can be computed from $\hat{\boldsymbol{d}}$ using (C.11).) This dual problem is similar in form to the inner minimization in the strict synthesis formulation (8.2). This relation between the analysis model and its dual formulation is limited to cases where ϕ is convex.

Whether analysis-based or synthesis-based regularizers are generally preferable is an open question, and the answer likely depends on the application and the relative importance of reconstruction accuracy and speed [168]. Synthesis regularization is perhaps easier to interpret because of its generative nature. In contrast, bilevel analysis filter learning is a discriminative learning approach: the task-based filters must learn to distinguish “good” and “bad” image features.

The synthesis approach used to be “widely considered to provide superior results” [168, p. 950]. However, [168] goes on to show that an analysis regularizer produced more accurate reconstructed images in experiments on real images. Later analysis-based results also show competitive, if not superior, quality results when compared to similar synthesis models [182], [183]. See [184] for a survey of optimization methods for MRI reconstruction and a comparison of the computational challenges for cost functions with synthesis and analysis-based regularizers.

The analysis and synthesis regularizers in (8.2) and (8.6) quickly yield infeasibly large operators as the signal size increases. In practice, both approaches are usually implemented with patch-based formulations. For the synthesis approach, the patches typically overlap and there is an averaging effect. Analysis regularizers that have rows corresponding to filters, called the convolutional analysis model, extend very naturally to a global image regularizer. For example, in the lower-level cost function of our running filter learning example (Ex), we can define an analysis regularizer matrix as follows:

$$\boldsymbol{\Omega} = \begin{bmatrix} \boldsymbol{C}_1 \\ \vdots \\ \boldsymbol{C}_K \end{bmatrix} \in \mathbb{F}^{KN \times N}. \tag{8.7}$$

Imposing this convolutional structure on $\boldsymbol{\Omega}$ helps make learning problems feasible as one only has to learn the S coefficients of each of the K filters rather than learning the full $\boldsymbol{\Omega}$ matrix. This structure also ensures translation invariance of the regularizer. See [162] and [185] for discussion of the connections between global models and patch-based models for analysis regularizers. The running example in this survey focuses on bilevel learning of convolutional analysis regularizers.

8.1.3 Brief History of Analysis Regularizer Learning

In 2003, Haber and Tenorio [158] proposed using bilevel methods to learn part of the regularizer in inverse problems. The authors motivate the use of bilevel methods through the task-based nature, noting that "the choice of good regularization operators strongly depends on the forward problem." They consider learning tuning parameters, space-varying weights, and regularization operators (comparable to defining ϕ), all for regularizers based on penalizing the energy in the derivatives of the reconstructed image. Their framework is general enough to handle learning filters. Ref. [158] was published a few years earlier than the other bilevel methods we consider in our review of the bilevel literature and was not cited in most other early works; [186] calls it a "groundbreaking, but often overlooked publication."

In 2005, Roth and Black [187] proposed the Field of Experts (FoE) model to learn filters. Although the FoE is not formulated as a bilevel method, many papers on bilevel methods for filter learning cite FoE as a starting or comparison point. The FoE model is a translation-invariant analysis operator model, built on convolutional filters. It is motivated by the local operators and presented as a Markov random field model, with the order of the field determined by the filter size.

Under the FoE model, the negative log[1] of the probability of a full image, $\boldsymbol{x}$, is proportional to

$$\sum_k \beta_k \, \phi.(\boldsymbol{c}_k \circledast \boldsymbol{x}) \text{ where } \phi(z) = \log\left(1 + \frac{1}{2}z^2\right). \tag{8.8}$$

This (non-convex) choice of sparsity function ϕ stems from the Student-t distribution. Ref. [187] learns the filters and filter-dependent tuning parameters such that the model distribution is as close as possible (defined using Kullback-Leibler divergence) to the training data distribution.

In 2007, Tappen, Liu, Adelson, *et al.* [188] proposed a different model based on convolutional

[1] By taking the log of the probability model in [187], the connection between the FoE and the regularization term in the lower-level of the running filter learning example (Ex) is more evident.

filters: the Gaussian Conditional Random Field (GCRF) model. Rather than using a sparsity promoting regularizer, the GCRF uses a quadratic function for ϕ. The authors introduce space-varying weights, $\boldsymbol{W}$, so that the quadratic model does not overly penalize sharp features in the image. The general idea behind $\boldsymbol{W}$ is to use the given (noisy) image to guess where edges occur, and correspondingly penalize those areas less to avoid blurring edges. The likelihood for GCRF model is thus (to within a proportionality constant and monotonic function transformations):

$$\sum_k \|\boldsymbol{c}_k \circledast \boldsymbol{x} - e_k\{\boldsymbol{x}\}\|^2_{\boldsymbol{W}_k},$$

where the term $e_k\{\boldsymbol{x}\}$ captures the estimated value of the filtered image. For example, [188] used one averaging filter and multiple differencing filters for the $\boldsymbol{c}_k$'s. The corresponding estimated values are $\boldsymbol{x}$ for the averaging filter and zero for the differencing filters.

The filters, $\boldsymbol{c}_k$, are pre-determined in the GCRF model; the learned element is how to form the weights as a function of image features. Specifically, each $\boldsymbol{W}_k$ is formed as a linear combination of the (absolute) responses to a set of edge-detecting filters, with the linear combination coefficients learned from training data. Rather than maximizing the likelihood of training data as in [187], [188] learns these coefficients to minimize the (corner-rounded) l_1 norm of the error of the predicted image, which is a form of bilevel learning even though not described with that terminology.

Apparently one of the first papers to explicitly propose using bilevel methods to learn filters appeared in 2009, where Samuel and Tappen [163] considered a bilevel formulation where the upper-level loss was the squared Euclidean norm of training data and the lower-level cost was a denoising task based on filter sparsity equivalent to (Ex). The method builds on the FoE model, using the same ϕ as in [187], but now learning the filters using a bilevel formulation rather than by maximizing a likelihood.

In 2011, Peyré and Fadili [165] proposed a similar bilevel method to learn analysis regularizers. The authors generalized the denoising task to use an analysis operator matrix and a wider class of sparsifying functions. Their results concentrate on the convolutional filter case with a corner-rounded 1-norm for ϕ.

Both [163] and [165] focus on introducing the bilevel method for analysis regularizer learning, with denoising or inpainting as illustrations. Chapter 10.1 further discusses the methodology of both papers. Many bilevel based papers build on one or both of their efforts. Chapter 10 and 12 summarize other bilevel based papers; here, we highlight some of papers in the non-bilevel thread of the literature for context and comparison.

Ophir, Elad, Bertin, *et al.* [189] proposed another approach to learning an analysis operator. The method learns the operator one row at a time by searching for vectors orthogonal to the training signals. Algorithm parameters were chosen empirically without an upper-level loss function as a guide.

Between 2011 [190] and 2013 [191], Yaghoobi, Nam, Gribonval, *et al.* were among the first to formally present analysis operator learning as an optimization problem. Their conference paper [190] considered noiseless training data and proposed learning an analysis operator as

$$\operatorname*{argmin}_{\boldsymbol{\Omega}} \|\boldsymbol{\Omega}\boldsymbol{X}^{\text{true}}\|_1 \text{ s.t. } \boldsymbol{\Omega} \in \mathcal{S} \tag{8.9}$$

for some constrained set $\mathcal{S}$. Each column of $\boldsymbol{X}^{\text{true}} \in \mathbb{F}^{N \times J}$ contains a training sample. The authors discussed varying options for $\mathcal{S}$, including a row norm, full rank, and tight frame constrained set.

Without any constraint on $\boldsymbol{\Omega}$, the trivial solution to (8.9) would be to learn the zero matrix, which is not informative for any problem such as image denoising. Section 9.1 discusses in more detail the need for constraints and the various constraint options proposed for filter learning.

Ref. [191] extends (8.9) to the noisy case where one does not have access to $\boldsymbol{X}^{\text{true}}$. The proposed cost function is

$$\operatorname*{argmin}_{\boldsymbol{\Omega}, \boldsymbol{X}} \|\boldsymbol{\Omega}\boldsymbol{X}\|_1 + \frac{\beta}{2}\|\boldsymbol{X} - \boldsymbol{Y}\|^2 \text{ s.t. } \boldsymbol{\Omega} \in \mathcal{S}, \tag{8.10}$$

where each column of $\boldsymbol{Y}$ contains a noisy data vector. Ref. [191] minimized (8.10) by alternating updating $\boldsymbol{X}$, using alternating direction method of multipliers (ADMM), and $\boldsymbol{\Omega}$, using a projected subgradient method for various constraint sets $\mathcal{S}$, especially Parseval tight frames.

In the same time-frame, Kunisch and Pock [192] started to analyze the theory behind the bilevel problem, building off the ideas in [163], [165]. Among the theoretical analysis, [192] proves the existence of upper-level minimizers when the bilevel problem takes the form of (Ex), $\boldsymbol{\gamma}$ is the tuning parameters (the β_k values), and ϕ corresponds to the squared 2-norm or the 1-norm. When $\phi(z) = z^2$, there is an analytic solution to the lower-level problem and a corresponding closed-form solution to the gradient of the upper-level problem; [192] uses this fact to discuss qualitative properties of the minimizer. Ref. [192] also proposed an efficient semi-smooth Newton algorithm for finding $\hat{\boldsymbol{\gamma}}$ (using corner rounding for the 1-norm case) and used this algorithm to make empirical comparisons of multiple sparsifying functions (2-norm, 1-norm, and $p = 1/2$-norm) and different pre-defined filter banks.

Also in 2013, Ravishankar and Bresler [183] made a distinction between the analysis model, where one models $\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{n}$ with $\boldsymbol{z} = \boldsymbol{\Omega}\boldsymbol{x}$ being sparse, and the transform model, where $\boldsymbol{\Omega}\boldsymbol{y} = \boldsymbol{z} + \boldsymbol{n}$ where $\boldsymbol{z}$ is sparse. The analysis version models the measurement as being a cosparse signal plus noise; the transform version models the measurement as being approximately cosparse. Another perspective on the distinction is that, if there is no noise, the analysis model constrains $\boldsymbol{y}$ to be in the range space of $\boldsymbol{\Omega}$, while there is no such constraint on the transform model. The corresponding transform learning problem is

$$\underset{\boldsymbol{\Omega}}{\operatorname{argmin}} \min_{\boldsymbol{Z}} ||\boldsymbol{\Omega Y} - \boldsymbol{Z}||_2^2 + R(\boldsymbol{\Omega}) \quad \text{s.t. } ||\boldsymbol{Z}_i||_0 \leq \alpha \ \forall i, \tag{8.11}$$

where i indexes the columns of $\boldsymbol{Z}$. Ref. [183] considers only square matrices $\boldsymbol{\Omega}$. The regularizer, R, promotes diversity in the rows of $\boldsymbol{\Omega}$ to avoid trivial solutions, similar to the set constraint in (8.10).

A more recent development is directly modeling the convolutional structure during the learning process. In 2020, [193] proposed Convolutional Analysis Operator Learning (CAOL) to learn convolutional filters without patches. The CAOL cost function is

$$\underset{[\boldsymbol{c}_1,\ldots,\boldsymbol{c}_K]}{\operatorname{argmin}} \sum_{k=1}^{K} \min_{z} \frac{1}{2} ||\boldsymbol{c}_k \circledast \boldsymbol{x} - z||_2^2 + \beta ||z||_0 \text{ s.t. } [\boldsymbol{c}_1 \ldots \boldsymbol{c}_K] \in \mathcal{S}. \tag{8.12}$$

Unlike the previous cost functions, which typically require patches, CAOL can easily handle full-sized training images $\boldsymbol{x}$ due to the nature of the convolutional operator. Section 9.3.1 describes CAOL in more detail.

At the same time that model-based methods were being developed in the signal processing literature, Convolutional neural network (CNN) models were being advanced and trained in the machine learning and computer vision literature [194] [195] [196]. The filters used in CNN models like U-Nets [197] can be thought of as having analysis roles in the earlier layers, and synthesis roles in the final layers [198]. See also [199] for further connections between analysis and transform models within CNN models. CNN training is usually supervised, and the supervised approach of bilevel learning of filters strengthens the relationships between the two approaches. A key distinction is that CNN models are generally feed-forward computations, whereas bilevel methods of the form (LL) have a cost function formulation. See Section 12.2 for further discussion of the parallels between CNNs and bilevel methods.

8.2 Formulating the Hyperparameter Optimization Problem

Most inverse problems involve at least one hyperparameter. For example, the general reconstruction cost function (8.1) requires choosing the tuning parameter β that trades-off the influence of the data-fit and regularization terms. The field of hyperparameter optimization is large and encompasses categorical hyperparameters, such as which optimizer to use; conditional hyperparameters, where certain hyperparameters are relevant only if others take on certain values; and integer or real-valued hyperparameters [200]. Here, we focus on learning real-valued, continuous hyperparameters.

A hyperparameter's value can greatly influence the properties of the minimizer and a tuned hyperparameter typically improves over a default setting [200]. Fig. 8.3 illustrates how changing a tuning parameter can dramatically impact the visual quality of the reconstructed image. If β is too low, not enough weight is on the regularization term, and the minimizer is likely to be corrupted by noise in the measurements. If β is too high, the regularization term dominates, and the minimizer will not align with the measurements.

Generalizing to an arbitrary learning problem that could have multiple hyperparameters, the goal of hyperparameter optimization is to find the "best" set of hyperparameters, $\hat{\boldsymbol{\gamma}}$, to meet a goal, described by a loss function ℓ. Specifically, we wish to solve

$$\hat{\boldsymbol{\gamma}} = \underset{\boldsymbol{\gamma}\in\Gamma}{\operatorname{argmin}}\, \mathbb{E}\left[\ell(\boldsymbol{\gamma})\right], \tag{8.13}$$

where Γ is the set of all possible hyperparameters and the expectation is taken with respect to the distribution of the input data. If evaluating ℓ uses the output of another optimization problem, *e.g.*, $\hat{\boldsymbol{x}}$, then (8.13) is a bilevel problem as defined in (UL).

There are two key tasks in hyperparameter optimization.

1. The first is to quantify how good a hyperparameter is; this step is equivalent to defining ℓ in (8.13). Section 8.3 focuses on a high-level discussion of loss functions in the broader image quality assessment (IQA) literature. Section 12.1.2 builds on this discussion by reviewing specific loss functions used in bilevel methods.

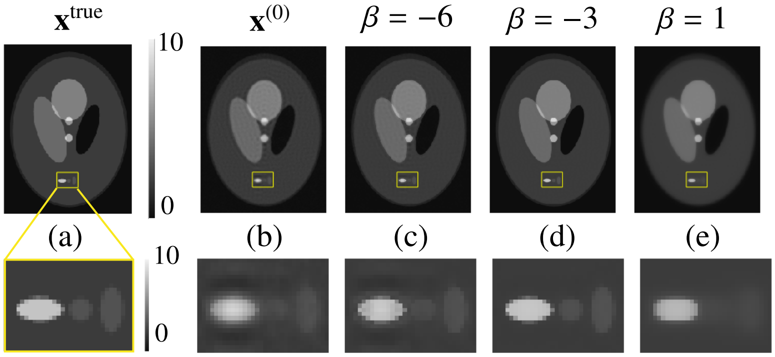


Figure 8.3: Example reconstructed simulated MRI images that demonstrate the importance of tuning parameters. (a) The original image, $\boldsymbol{x}^{\text{true}} \in R^N$, is a SheppLogan phantom [201] and N is the number of pixels. (b) A simplistic reconstruction $\frac{1}{N}\boldsymbol{A}'\boldsymbol{y}$ of the noisy, undersampled data, $\boldsymbol{y}$. This image is used as initialization, $\boldsymbol{x}^{(0)}$, for the following reconstructions. (c-e) Reconstructed images, found by optimizing $\operatorname{argmin}_{\boldsymbol{x}} \frac{1}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2 + 10^{\beta} N\phi(\boldsymbol{C}\boldsymbol{x})$, where $\boldsymbol{C}$ is an operator that takes vertical and horizontal finite differences. The reconstructed images correspond to (c) $\beta = -6$, resulting in an image that contains ringing artifacts, (d) $\beta = -3$, resulting in a visually appealing $\hat{\boldsymbol{x}}$, and (e) $\beta = 1$, resulting in a blurred image. The demonstration code and more details about the reconstruction set-up are available on github [202].

2. The second step is finding a good hyperparameter, which is equivalent to designing an optimization algorithm to minimize (8.13). Section 8.4 introduces common approaches, all of which have computational requirements that scale at least linearly with the number of hyperparameters. This scaling quickly becomes infeasible for large $\boldsymbol{\gamma}$, which motivates the focus on gradient-based bilevel methods in the remainder of this review.

The next two sections address each of these tasks in turn.

8.3 Image Quality Metrics

This section concentrates on the part of the upper-level loss function that compares the reconstructed image, $\hat{\boldsymbol{x}}(\boldsymbol{\gamma})$, to the true image, $\boldsymbol{x}^{\text{true}}$. As mentioned in Chapter 6.3, bilevel methods rarely require additional regularization for $\boldsymbol{\gamma}$, but it is simple to add a regularization term to any of the loss functions if useful for a specific application. To discuss only the portion of the loss function that measures image quality, we use the notation $\ell(\boldsymbol{\gamma}\,; \hat{\boldsymbol{x}}(\boldsymbol{\gamma})) = l(\hat{\boldsymbol{x}},\, \boldsymbol{x}^{\text{true}})$.

Picking a loss function is part of the engineering design process. No single loss function is likely to work in all scenarios; users must decide on the loss function that best fits their system, data, and goals. Consequently, there are a wide variety of loss functions proposed in the literature and some approaches combine multiple loss functions [138], [203].

One important decision criteria when selecting a loss function is the end purpose of the image. Much of the IQA literature focuses on metrics for images of natural scenes and is often motivated by applications where human enjoyment is the end-goal [204], [205]. In contrast, in the medical image reconstruction field, image quality is not the end-goal, but rather a means to achieving a correct diagnosis. Thus, the perceptual quality is less important than the information content.

There are two major classes of image quality metrics in the IQA literature, called full-reference and no-reference IQA[2]. The principles are somewhat analogous to supervised and unsupervised approaches in the machine learning literature. This section discusses some of the most common full-reference and no-reference loss functions; see [206] for a comparison of 11 full-reference IQA metrics and [207] for additional no-reference IQA metrics.

Perhaps surprisingly, the bilevel filter learning literature contains few examples of loss functions other than squared error or slight variants (see Section 12.1.2). While this is likely at least partially due to the computational requirements of bilevel methods (see Chapter 10), exploring additional loss functions is an interesting future direction for bilevel research.

[2] There are also reduced-reference image quality metrics, but we will not consider those here.

8.3.1 Full-Reference Image Quality Assessment

Full-reference IQA metrics assume that you have a noiseless image, $\boldsymbol{x}^{\text{true}}$, for comparison. Some of the simplest (and most common) full-reference loss functions are:

- Mean squared error (MSE or ℓ_2 error):

$$l_{\text{MSE}}(\hat{\boldsymbol{x}}, \boldsymbol{x}^{\text{true}}) = \frac{1}{N} \left\| \hat{\boldsymbol{x}} - \boldsymbol{x}^{\text{true}} \right\|_2^2$$

- Mean absolute error (or ℓ_1 error): $l_{\text{MAE}}(\hat{\boldsymbol{x}}, \boldsymbol{x}^{\text{true}}) = \frac{1}{N} \left\| \hat{\boldsymbol{x}} - \boldsymbol{x}^{\text{true}} \right\|_1$
- Signal to Noise Ratio (SNR, commonly expressed in dB):

$$l_{\text{SNR}}(\hat{\boldsymbol{x}}, \boldsymbol{x}^{\text{true}}) = 10\log\left(\frac{\left\| \boldsymbol{x}^{\text{true}} \right\|_2^2}{\|\hat{\boldsymbol{x}} - \boldsymbol{x}^{\text{true}}\|_2^2} \right) \tag{8.14}$$

- Peak SNR (Peak signal to noise ratio (PSNR), in dB): $l_{\text{PSNR}}(\hat{\boldsymbol{x}}, \boldsymbol{x}^{\text{true}}) = 10\log\left(\frac{N\left\| \boldsymbol{x}^{\text{true}} \right\|_\infty}{\|\hat{\boldsymbol{x}} - \boldsymbol{x}^{\text{true}}\|_2^2} \right)$.

The Euclidean norm is also frequently used as the data-fit term for reconstruction.

Mean squared error (MSE) (and the related metrics SNR and PSNR) are common in the signal processing field; they are intuitive and easy to use because they are differentiable and operate point-wise. However, these measures do not align well with human perceptions of image quality [206], [208]. For example, scaling an image by 2 leads to the same visual quality but causes 100% MSE. Fig. 8.4 shows a clean image and five images with different degradations. All five degraded images have almost equivalent squared errors, but humans judge their qualities as very different.

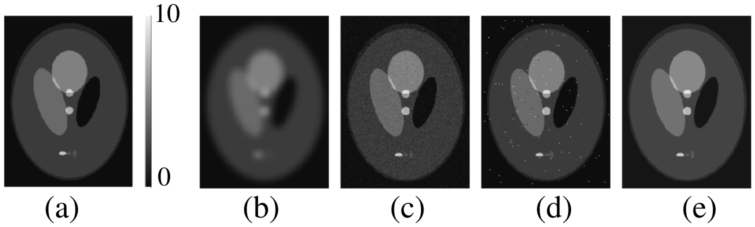


Figure 8.4: Example distortions that yield images with identical normalized squared error values: $\left\| \boldsymbol{x}^{\text{true}} - \boldsymbol{x} \right\| / \left\| \boldsymbol{x}^{\text{true}} \right\| = 0.17$. (a) The original image, $\boldsymbol{x}^{\text{true}}$, is a SheppLogan phantom [201]. The remaining images are displayed with the same colormap and have the following distortions: (b) blurred with an averaging filter, (c) additive, white Gaussian noise, (d) salt and pepper noise, and (e) a constant value added to every pixel.

Tuning parameters using MSE as the loss function tends to lead to images that are overly-smoothed, sacrificing high frequency information [209], [210]. High frequency details are particularly important for perceptual quality as they correspond to edges in images. Therefore, some authors use the MSE on edge-enhanced versions of images to discourage solutions that blur edges.

For example, [211] used a “high frequency error norm” metric consisting of the MSE of the difference of $\hat{\boldsymbol{x}}$ and $\boldsymbol{x}^{\text{true}}$ after applying a Laplacian of Gaussian (LoG) filter.

Another common full-reference IQA is Structural SIMilarity (Structural Similarity (SSIM)) [204] that attempts to address the issues with MSE discussed above. SSIM is defined in terms of the local luminance, contrast, and structure in images. A multiscale extension of SSIM, called MS-SSIM, considers these features at multiple resolutions [212]. The method computes the contrast and structure measures of SSIM for downsampled versions of the input images and then defines MS-SSIM as the product of the luminance at the original scale and the contrast and structure measures at each scale. However, SSIM and MS-SSIM may not correlate well with human observer performance on radiological tasks [213].

Recent works, *e.g.*, [207], [214], consider using (deep) CNN models for IQA. CNN methods are increasingly popular and their use as a model for the human visual system [215] makes them an attractive tool for assessing images. For example, [214] proposed a CNN with convolutional and pooling layers for feature extraction and fully connected layers for regression. They used VGG [216], a frequently-cited CNN design with 3 × 3 convolutional kernels, as the basis of the feature extraction portion of their network. Ref. [214] showed that deeper networks with more learnable parameters were able to better predict image quality. However, datasets of images with quality labels remain relatively scarce, making it difficult to train deep networks.

8.3.2 No-reference Image Quality Assessment

No-reference, or unsupervised, IQA metrics attempt to quantify an image’s quality without access to a noiseless version of the image. These metrics rely on modeling statistical characteristics of images or noise. Many no-reference IQA metrics assume the noise distribution is known.

The discrepancy principle is a classic example of an IQA metric that uses an assumed noise distribution to characterize the expected relation between the reconstructed image and the noisy data. For additive zero-mean white Gaussian noise with known variance σ^2, the discrepancy principle uses the fact that the expected MSE in the data space is the noise variance [150]:

$$\mathbb{E}\left[\frac{1}{M}\|\boldsymbol{A}\hat{\boldsymbol{x}}(\boldsymbol{\gamma})-\boldsymbol{y}\|_2^2\right]=\sigma^2.$$

The discrepancy principle can be used as a stopping criteria in machine learning methods or as a loss function, *e.g.*,

$$\ell(\boldsymbol{\gamma}\,;\hat{\boldsymbol{x}}(\boldsymbol{\gamma}))=\left(\frac{1}{M}\|\boldsymbol{A}\hat{\boldsymbol{x}}(\boldsymbol{\gamma})-\boldsymbol{y}\|_2^2-\sigma^2\right)^2.$$

However, images of varying quality can yield the same noise estimate, as seen in Fig. 8.4. Related methods have been developed for Poisson noise as well [217].

Paralleling MSE's popularity among supervised loss metrics, Stein's Unbiased Risk Estimator (SURE) [218] is an unbiased estimate of MSE that does not require noiseless images. Let $\boldsymbol{y} = \boldsymbol{x}^{\text{true}} + \boldsymbol{n}$ denote a signal plus noise measurement where $\boldsymbol{n}$ is, as above, Gaussian noise with known variance σ^2. The SURE estimate of the MSE of a denoised signal, $\hat{\boldsymbol{x}}$, is

$$\frac{1}{N} \|\hat{\boldsymbol{x}}(\boldsymbol{y}) - \boldsymbol{y}\|_2^2 - \sigma^2 + \frac{2\sigma^2}{N} \operatorname{Tr}\left(\nabla_y \hat{\boldsymbol{x}}(\boldsymbol{y})\right), \tag{8.15}$$

where we write $\hat{\boldsymbol{x}}$ as a function of $\boldsymbol{y}$ to emphasize the dependence and $\operatorname{Tr}(\cdot)$ denotes the trace operation. For large signal dimensions N, such as is common in image reconstruction problems, the law of large numbers suggests SURE is a fairly accurate approximation of the true MSE.

It is often impractical to evaluate the divergence term in (8.15), due to computational limitations or not knowing the form of $\hat{\boldsymbol{x}}(\boldsymbol{y})$. A Monte-Carlo approach to estimating the divergence [219] uses the following key equation:

$$\operatorname{Tr}\left(\nabla_y \hat{\boldsymbol{x}}(\boldsymbol{y})\right) = \lim_{\epsilon \to 0} \mathbb{E}\left[\boldsymbol{b}' \cdot \frac{\hat{\boldsymbol{x}}(\boldsymbol{y} + \epsilon \boldsymbol{b}) - \hat{\boldsymbol{x}}(\boldsymbol{y})}{\epsilon}\right], \tag{8.16}$$

where $\boldsymbol{b}$ is a independent and identically distributed (i.i.d.) random vector with zero mean, unit variance, and bounded higher order moments. Theoretical and empirical arguments show that a single noise vector can well-approximate the divergence [219], so only two calls to the lower-level solver $\hat{\boldsymbol{x}}(\boldsymbol{y})$ are required. This method treats the lower-level problem like a blackbox, thus allowing one to estimate the divergence of complicated functions, including those that may not be differentiable.

See [220]–[222] for examples of applying the Monte-Carlo estimation of SURE to train deep neural networks, and [223], [224] for two examples of learning a tuning parameter using a bilevel approach with SURE as the upper-level loss function. For extensions to inverse problems (where $\boldsymbol{A} \neq \boldsymbol{I}$) and to noise from exponential families, see [225]–[227].

While SURE and the discrepancy principle are popular no-reference metrics in the signal processing literature, there are many additional no-reference metrics in the image quality assessment literature. These metrics typically depend on modeling one (or more) of three things [205]:

- image source characteristics,
- image distortion characteristics, *e.g.*, blocking artifact from JPEG compression, and/or
- human visual system perceptual characteristics.

As an example of a strategy that can capture both image source and human visual system characteristics, natural scene[3] statistics characterize the distribution of various features in natural scenes,

[3] Natural scenes are those captured by optical cameras (not created by computer graphics or other artificial processes) and are not limited to outdoor scenes.

typically using some filters [205], [228]. If a feature reliably follows a specific statistical pattern in natural images but has a noticeably different distribution in distorted images, one can use that feature to assign quality scores to images. Some IQA metrics attempt to first identify the type of distortion and measure features specific to that distortion, while others use the same features for all images.

In addition to their use in full-reference IQA, CNN models have be trained to perform no-reference IQA [214], [229]. For example, [229] proposes a CNN model that extracts small (32×32) patches from images, estimates the quality of each one, and averages the scores over all patches to get a quality score for the entire image. Briefly, their method involves local contrast normalization for each patch, applying (learned) convolutional filters to extract features, maximum and minimum pooling, and fully connected layers with rectified linear units (ReLUs). As with most no-reference IQAs, [229] trained their CNN on a dataset of human encoded image quality scores (see [230] for a commonly used collection of publicly available test images with quality scores). Unlike most other IQA approaches, [229] used backpropagation to learn all the CNN weights rather than learning a transformation from handcrafted features to quality scores.

Interestingly, some of the no-reference IQA metrics [205], [228], [229] approach the performance of the full-reference IQAs in terms of their ability to match human judgements of image quality. This observation suggests that there is room to improve full-reference IQA metrics and that assessing image quality is a very challenging problem!

8.4 Parameter Search Strategies

After selecting a metric to measure how good a hyperparameter is, the next task is devising a strategy to find the best hyperparameter according to that metric. Search strategies fall into three main categories: (i) model-free, ℓ-only; (ii) model-based, ℓ-only; and (iii) gradient-based, using both ℓ and $\nabla\ell$. Model-free strategies do not assume any information about about the hyperparameter landscape, whereas model-based strategies use historical ℓ evaluations to predict the loss function at untested hyperparameter values.

The following sections describe common model-free and model-based hyperparameter search strategies that only use ℓ. See [141, Ch. 13 and Ch. 20.6] for discussion of additional gradient-free methods for bilevel problems, *e.g.*, population-based evolutionary algorithms, and [231] for a general discussion of derivative-free optimization methods.

The third class of hyperparameter optimization schemes are approaches based on gradient descent of a bilevel problem. The high-level strategy in bilevel approaches is to calculate the gradient of the upper-level loss function ℓ with respect to $\boldsymbol{\gamma}$ and then use any gradient descent method to minimize $\boldsymbol{\gamma}$. Although this approach can be computationally challenging, it generalizes well to a large number of hyperparameters. Chapter 10 discuss this point further and go into depth on different methods for computing this gradient.

8.4.1 Model-free Hyperparameter Optimization

The most common search strategy is probably an empirical search, where a researcher tries different hyperparameter combinations manually. A punny, but often accurate, term for this manual search is GSD: grad[uate] student descent [232]. Bergstra and Bengio [233] hypothesizes that manual search is common because it provides some insight as the user must evaluate each option, it requires no overhead for implementation, and it can perform reliably in very low dimensional hyperparameter spaces.

Grid search is a more systematic alternative to manual search. When there are only one or two continuous hyperparameters, or the possible set of hyperparameters, $\mathbf{\Gamma}$, is small, a grid search (or exhaustive search) strategy may suffice to find the optimal value, $\hat{\boldsymbol{\gamma}}$, to within the grid spacing. However, the complexity of grid search grows exponentially with the number of hyperparameters. Regularizers frequently have many hyperparameters, so one generally requires a more sophisticated search strategy.

One popular approach is random search, which [233] shows is superior to a grid search, especially when some hyperparameters are more important than others. There are also variations on random search, such as using Poisson disk sampling theory to explore the hyperparameter space [234]. The simplicity of random search makes it popular, and, even if one uses a more complicated search strategy, random search can provide a useful baseline or an initialization strategy. However, random search, like grid search, suffers from the curse of dimensionality, and is less effective as the hyperparameter space grows.

Another group of model-free blackbox strategies are population-based methods such as evolutionary algorithms. A popular population-based method is the covariance matrix adaption evolutionary strategy (CMA-ES) [235]. In short, every iteration, CMA-ES involves sampling a multivariate normal distribution to create a number of “offspring” samples. Mimicking natural selection, these offspring are judged according to some fitness function, a parallel to the upper-level loss function. The fittest offspring determine the update to the normal distribution and thus “pass on” their good characteristics to the next generation.

8.4.2 Model-based Hyperparameter Optimization

Model-based search strategies assume a model (or prior) for the hyperparameter space and use only loss function evaluations (no gradients). This section discusses two common model-based strategies: Bayesian methods and trust region methods.

8.4.2.1 Bayesian Approaches

Bayesian methods fit previous hyperparameter trials' results to a model to select the hyperparameters that appear most promising to evaluate next [236]. For example, a common model for the hyperparameters is the Gaussian Process prior. Given a few hyperparameter and cost function points, a Bayesian method involves the following steps.

1. Find the mean and covariance functions for the Gaussian Process. The mean function will generally interpolate the sampled points. The covariance function is generally expressed as a kernel function, often using squared exponential functions [237].
2. Create an acquisition function. The acquisition function captures how desirable it is to sample ("acquire") a hyperparameter setting. Thus, it should be large (desirable) for hyperparameter values that are predicted to yield small loss function values or that have high enough uncertainty that they may yield low losses. The design of the acquisition function thus trades-off between exploring new areas of the hyperparameter landscape with high uncertainty and a more locally focused exploitation of the current best hyperparameter settings. See [237] for a discussion of specific acquisition function designs.
3. Maximize the acquisition function (typically designed to be easy to optimize) to determine which hyperparameter point to sample next.
4. Evaluate the loss function at the new hyperparameter candidate.

These steps repeat for a given amount of time or until convergence.

8.4.2.2 Trust-region Methods

Another derivative-free optimization method that uses only loss function evaluations is a trust-region method. This section describes the specific trust-region method as presented in [159] (see references therein for previous, similar methods). An outline for a TRM is

1. Create a quadratic model for the upper-level loss function.

   (a) Select a set of upper-level interpolating points and (approximately) evaluate the upper-level at each one.

   (b) Estimate the upper-level gradient by interpolating a set of R samples (recall $\boldsymbol{\gamma} \in \mathbb{F}^R$) of the upper-level loss function.

   (c) Model the upper-level by it with its tangent-plane approximation using the estimated gradient from the previous step.

2. Minimize the model within some trust region to find the next candidate set of upper-level parameters. By construction, this is a simple convex-constrained quadratic problem.

3. Accept or reject the updated parameters and update the trust region. If the ratio between the actual reduction and predicted reduction is low, the model may no longer be a good fit, the update is rejected, and the trust region shrinks.

The derivative-free, trust-region method (TRM) [238] is similar to Bayesian optimization in that it involves fitting an easier to optimize function to the loss function of interest, ℓ, and then minimizing the easier, surrogate function (the "model"). Thus, TRM requires only function evaluations, not gradients, to construct and then minimize the model. However, unlike most Bayesian optimization-based approaches, TRM uses a local (often quadratic) model for ℓ around the current iterate, rather than a surrogate that fits all previous points. In taking a step based on this local information, TRM resembles gradient-based approaches.

The "trust-region" in TRM captures how well the model matches the observed ℓ values and determines the maximum step at every iteration. The "goodness" of the model is typically quantified as the ratio of the actual decrease in ℓ (based on observed function evaluations) to the predicted decrease (based on the model). If this ratio is relatively large (close to one), then the model is a good approximation of ℓ and the trust-region grows for the next iteration. If this ratio is close to zero, then the observed decrease is much less than predicted and the trust-region shrinks.

Recall that evaluating ℓ is typically expensive in bilevel problems as each upper-level function evaluation involves optimizing the lower-level cost. Thus, even constructing the model for a TRM can be expensive. To mitigate this computational complexity, [159] incorporated a dynamic accuracy component, with the accuracy for the lower-level cost initially set relatively loose (leading to rough estimates of ℓ) but increasing with the upper-level iterations (leading to refined estimates of ℓ as the algorithm nears a stationary point). One can use any optimization method for the lower-level cost; [159] used a gradient method for the lower-level optimization method with well-known convergence results to facilitate establishing convergence and computational complexity results.

The upper-level loss function considered in [159] is additively separable and quadratic:

$$\ell(\boldsymbol{\gamma}) = \frac{1}{J}\sum_{j=1}^{J} \ell(\boldsymbol{\gamma}\,; \hat{\boldsymbol{x}}_j(\boldsymbol{\gamma})) = \frac{1}{J}\sum_{j=1}^{J} \underbrace{\left(r(\boldsymbol{\gamma}\,; \hat{\boldsymbol{x}}_j(\boldsymbol{\gamma}))\right)^2}_{\text{Equivalently, } r_j(\boldsymbol{\gamma})^2},$$

where r is typically $\hat{\boldsymbol{x}}_j(\boldsymbol{\gamma}) - \boldsymbol{x}_j^{\text{true}}$. (Although we define the sum to be over the number of training samples, this expression easily generalizes to include a regularization term on $\boldsymbol{\gamma}$ by defining an additional r_{J+1} term.)

Given a current value for the hyperparameters, $\boldsymbol{\gamma}^*$, the TRM models the local upper-level loss function by creating a linear model such that

$$r_j(\boldsymbol{\gamma}^* + \boldsymbol{\delta}) \approx m_j(\boldsymbol{\delta}) := r_j(\boldsymbol{\gamma}^*) + \boldsymbol{g}_j'\boldsymbol{\delta},$$

where $\boldsymbol{g}_j \in \mathbb{F}^R$ approximates $\nabla r_j(\boldsymbol{\gamma}^*)$. Then, the overall model for the upper-level problem is quadratic:

$$\ell(\boldsymbol{\gamma}^* + \boldsymbol{\delta}) \approx \sum_j m_j^2(\boldsymbol{\delta}) = \ell(\boldsymbol{\gamma}^*) + \frac{1}{J} \sum_j 2r_j(\boldsymbol{\gamma}^*)\boldsymbol{g}_j'\boldsymbol{\delta} + (\boldsymbol{g}_j'\boldsymbol{\delta})^2. \tag{8.17}$$

One can estimate the gradients, $\boldsymbol{g}_j$, by interpolating a set of R samples (recall $\boldsymbol{\gamma} \in \mathbb{F}^R$) of the upper-level loss function. This process involves choosing a set of interpolating points, $\{\boldsymbol{\gamma}^* + \boldsymbol{\delta}^{(1)}, \ldots, \boldsymbol{\gamma}^* + \boldsymbol{\delta}^{(R)}\}$, (approximately) evaluating r at each one, then solving

$$\underbrace{\begin{bmatrix} r_j(\boldsymbol{\gamma}^* + \boldsymbol{\delta}^{(1)}) - r_j(\boldsymbol{\gamma}^*) \\ \vdots \\ r_j(\boldsymbol{\gamma}^* + \boldsymbol{\delta}^{(R)}) - r_j(\boldsymbol{\gamma}^*) \end{bmatrix}}_{R} = \underbrace{\begin{bmatrix} \left(\boldsymbol{\delta}^{(1)}\right)' \\ \vdots \\ \left(\boldsymbol{\delta}^{(R)}\right)' \end{bmatrix}}_{R \times R} \boldsymbol{g}_j$$

for $j \in [1 \ldots J]$.

After forming the quadratic model for the upper-level loss, the TRM minimizes the model (8.17) within some trust region, which is a simple convex-constrained quadratic problem After computing $\hat{\boldsymbol{\delta}}$, the TRM accepts the step and updates the hyperparameters ($\boldsymbol{\gamma}^{(i+1)} = \boldsymbol{\gamma}^{(i)} + \hat{\boldsymbol{\delta}}$) if the actual reduction (based on the estimated loss function values) to predicted reduction (based on the quadratic model of the loss function) ratio is large enough. Otherwise, if the ratio is low, the update step is rejected and the trust region shrinks.

While the TRM appears to involve R upper-level function evaluations every iteration to construct the gradient estimates, after an initialization, one can generally reuse samples, gradually replacing old samples with the samples at new hyperparameter iterates. Ref. [159] discusses requirements on the interpolation set to guarantee a good geometry and conditions for re-setting the interpolation sample if the model is not sufficiently accurate.

A main result from [159] is a bound on the number of iterations to reach an ϵ-optimal point (defined as $\min_u \|\nabla_{\boldsymbol{\gamma}} \ell(\boldsymbol{\gamma}^{(u)})\| < \epsilon$, where u indexes the upper-level iterates). The bound derivation assumes (i) Φ is differentiable in $\boldsymbol{x}$, (ii) Φ is μ-strongly convex, *i.e.*, $\Phi(\boldsymbol{x}) - \frac{\mu}{2} \|\boldsymbol{x}\|^2$ is convex for $\mu > 0$, (iii) the derivative of Φ is Lipschitz continuous, and (iv) the first and second derivative of the lower-level cost with respect to $\boldsymbol{x}$ exist and are continuous. These requirements are satisfied by the example filter learning problem (Ex), when $\boldsymbol{A}$ has full column rank, and more generally when there are certain constraints on the hyperparameters. The iteration bound is a function of the following:

- the tolerance ϵ,
- the trust region parameters (parameters that control the increase and decrease in trust re-

gion size based on the actual to predicted reduction, the starting trust region size, and the minimum possible trust region size),

- the initialization for $\boldsymbol{\gamma}$, and
- the maximum possible error between the gradient of the upper-level loss function and the gradient of the model for the upper-level loss within a trust region (when the gradient of ℓ is Lipschitz continuous, this bound is the corresponding Lipschitz constant).

The number of iterations required to reach such an ϵ-optimal point is $\mathcal{O}\left(\frac{1}{\epsilon^2}\right)$ [159] and the number of required upper-level loss function evaluations depends more than linearly on N [239]. The growth with the number of hyperparameters impedes its use in problems with many hyperparameters. However, new techniques such as [240] may be able to decrease or remove the dependency, making TRMs promising alternatives to the gradient-based bilevel methods described in the remainder of this review.

8.5 Summary

The first part of this background chapter focused on the lower-level problem: image reconstruction with a sparsity-based regularizer. After defining the problem and the need for regularization, Section 8.1.3 reviewed the history of analysis regularizer learning and included many examples of methods to learn hyperparameters.

Bilevel methods are just one, task-based way to learn such hyperparameters. Section 9.1 further expands on this point, but we can already see benefits of the task-based nature of bilevel methods. Without the bilevel approach, filters are often learned such that they best sparsify training data. These sparsifying filters can then be used in a regularizer for image reconstruction tasks. However, they are learned to *sparsify*, not necessarily to best *reconstruct*. In contrast, the bilevel approach aims to learn filters that best reconstruct images (or whatever other task is desired), even if those filters are not the ones that best sparsify. Although this distinction may seem subtle, [241] shows that different filters work better for image denoising versus image inpainting.

Turning from the discussion of the lower-level problem in Section 8.1, the second part of this background chapter concentrated on the other two aspects of bilevel problems: the upper-level loss function and the optimization strategy.

The loss function defines what a "good" hyperparameter is, typically using a metric of image quality to compare $\hat{\boldsymbol{x}}(\boldsymbol{\gamma})$ to a clean, training image, $\boldsymbol{x}^{\text{true}}$. Variations on squared error are the most common upper-level loss functions. Section 8.3 discussed many other full-reference and no-reference options, including ones motivated by human judgements of perceptual quality, from the image quality assessment literature; Section 12.1.2 gives examples of bilevel methods that use some of these other loss functions.

Section 8.4 concentrated on model-free and model-based hyperparameter search strategies. The grid search, CMA-ES, and trust region methods described above all scale at least linearly with the number of hyperparameters. Similarly, Bayesian optimization is best-suited for small hyperparameter dimensions; [237] suggests it is typically used for problems with 20 or fewer hyperparameters.

The remainder of this dissertation considers gradient-based strategies for hyperparameter optimization. The main benefit of gradient-based methods is that they can scale to the large number of hyperparameters that are commonly used in machine learning applications. Correspondingly, the main drawbacks of a gradient-based method are the implementation complexity, the per-iteration computational complexity, and the typical differentiability requirement. Chapter 10 discuss multiple options for gradient-based methods.

CHAPTER 9

RQ#4: Motivating Task-Based Learning Approaches

Chapter 6.3 introduced bilevel learning methods and claimed that one of the primary benefits of bilevel methods is the task-based nature. This chapter considers two case studies to support the importance of task-based learning, *i.e.*, learning parameters such that they best perform a lower-level task. Specifically, this chapter demonstrates the sub-optimally of filters learned using non-bilevel methods. Chapter 11 revisits one of the case studies to examine how the task-based bilevel method compares to the methods examined in this chapter.

This chapter considers our research question: **Why do handcrafted sparsifying filters sometimes outperform learned filters?** First, this chapter presents a simple experiment: learning a single sparsifying filter for piece-wise constant signals. The simplicity of the experiment allows us to handcraft a filter and then demonstrate how a common learning method makes the learned filter perform worse than an obvious handcrafted filter. This simple experiment builds on the results presented in [10]:

> C. Crockett and J. A. Fessler, "Motivating bilevel approaches to filter learning: A case study," in *2021 IEEE International Conference on Image Processing (ICIP)*, IEEE, Sep. 19, 2021, pp. 2803–2807, ISBN: 978-1-66544-115-5.
> DOI: 10.1109/ICIP42928.2021.9506489

Next, this chapter introduces the handcrafted Convolutional Analysis Operator Learning (CAOL) algorithm, which allows for incorporating specific handcrafted filters into the CAOL learning process, and is thus suited to examining the trade-off between handcrafting filters and learning filters. These results are presented in [9]:

> C. Crockett, D. Hong, I. Y. Chun, *et al.*, "Incorporating handcrafted filters in convolutional analysis operator learning for ill-posed inverse problems," in *2019 IEEE 8th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing*, CAMSAP, Dec. 2019, pp. 316–320.
> DOI: 10.1109/CAMSAP45676.2019.9022669

Throughout this chapter, we refer to "transforms" $\boldsymbol{T}$ and "sparsifying filters" $\boldsymbol{c}$ as different mathematical perspectives of the same underlying phenomenon. Typically, the literature uses "transforms" to refer to matrices that, when left-multiplied, sparsify patches of the signals, *i.e.*, to model $\boldsymbol{T}(\boldsymbol{P}_l\boldsymbol{x})$ being sparse for many l, where $\boldsymbol{P}_l$ extracts the lth patch from the signal $\boldsymbol{x}$. The phrase "sparsifying filters" commonly refers to a filter that, when convolved with the signals, sparsifies the signals, *i.e.*, to model $\boldsymbol{c} \circledast \boldsymbol{x}$ being sparse. With corresponding boundary conditions on the convolution operator and patch extraction matrix, the two models are equivalent, and we can view each row of $\boldsymbol{T}$ as either a row in a transform matrix or as a sparsifying filter that could be applied using convolution.

9.1 Background: Single-Level Parameter Learning

Section 8.1.3 briefly discussed some approaches to learning analysis operators. This section further motivates the task-based bilevel set-up by discussing the filter learning constraints imposed in single-level hyperparameter learning methods.

As summarized in Section 8.1.3, the earliest methods for learning analysis regularizers had no constraints on the analysis operators. Those approaches learned filters from training data to make a prior distribution match the observed data distribution. In contrast, more recent approaches to filter learning minimize a cost function that requires either a penalty function or constraint on the operators to ensure filter diversity. For reference, the cost functions mentioned in Section 8.1.3 were:

$$\begin{aligned}
\text{AOL} &: \operatorname*{argmin}_{\boldsymbol{\Omega}, \boldsymbol{X}} \|\boldsymbol{\Omega X}\|_1 + \frac{\beta}{2}\|\boldsymbol{Y} - \boldsymbol{X}\|^2 \text{ s.t. } \boldsymbol{\Omega} \in \mathcal{S}, \\
\text{TL} &: \operatorname*{argmin}_{\boldsymbol{\Omega} \in \mathbb{F}^{S \times S}, \boldsymbol{X}} \|\boldsymbol{\Omega Y} - \boldsymbol{X}\|_2^2 + R(\boldsymbol{\Omega}) \text{ s.t. } \left\|\boldsymbol{X}_{:,i}\right\|_0 \leq \alpha \ \forall i, \\
\text{CAOL} &: \operatorname*{argmin}_{[\boldsymbol{c}_1, \ldots, \boldsymbol{c}_K]} \min_{z} \sum_{k=1}^{K} \frac{1}{2}\|\boldsymbol{c}_k \circledast \boldsymbol{x} - \boldsymbol{z}\|_2^2 + \beta \|\boldsymbol{z}_k\|_0 \text{ s.t. } [\boldsymbol{c}_1, \ldots, \boldsymbol{c}_K] \in \mathcal{S},
\end{aligned}$$

where AOL is analysis operator learning [191], TL is transform learning [183], and CAOL is convolutional analysis operator learning [193]. In the following discussion of constraint sets, the equivalent filter matrix for CAOL has the convolutional kernels as rows:

$$\boldsymbol{\Omega}_{\text{CAOL}} = \begin{bmatrix} \boldsymbol{c}_1' \\ \vdots \\ \boldsymbol{c}_K' \end{bmatrix}.$$

While there are many other proposed cost functions in the literature, using different norms or including additional variables, these three examples capture the most common structures for filter learning.

In all the above cost functions, if one removed the constraint or regularizer, then the trivial solution would be to learn zero filters for $\mathbf{\Omega}$. Furthermore, a simple row norm constraint on $\mathbf{\Omega}$ would be insufficient, as then the minimizer would contain a single filter that is repeated many times. (In contrast, a unit norm constraint typically suffices for dictionary learning.) A row norm constraint plus a full rank constraint is also insufficient because $\mathbf{\Omega}$ can have full rank while being arbitrarily close to the rank-1 case of having a single repeated row.

The choice of constraint set $\mathcal{S}$ is important in single-level learning. Many methods constrain analysis operators to satisfy a tight frame constraint. A matrix $\boldsymbol{A}$ is a tight frame if there is a positive constant, α, such that

$$\|\boldsymbol{A}'\boldsymbol{x}\|_2^2 = \sum_i |\langle \boldsymbol{q}_i, \boldsymbol{x} \rangle|^2 = \alpha \, \|\boldsymbol{x}\|_2^2 \,, \; \forall \boldsymbol{x}$$

where $\boldsymbol{q}_i$ is the ith column of $\boldsymbol{A}$. This tight frame condition is equivalent to $\boldsymbol{A}\boldsymbol{A}' = \alpha \boldsymbol{I}$ for some positive constant α. Most analysis operators are defined with filters in their rows, so a tight frame requirement on the filters appears as the constraint $\mathbf{\Omega}'\mathbf{\Omega} = \alpha \boldsymbol{I}$.

Under the tight frame constraint for the filters, $\mathbf{\Omega}$ must be square or tall, so the filters are complete or over-complete. However, [191] found that the frame constraint was insufficient when learning over-complete operators, as the “excess” rows past full-rank tended to be all zeros. Therefore, [191] imposed a uniformly-normalized tight frame constraint: each row of the $\mathbf{\Omega}$ had to have unit norm and the filters had to form a tight frame.

Ref. [182] similarly constrained $\mathbf{\Omega}$ to have unit-norm rows with the filters forming a frame (though not tight). Such loosening of the tight frame constraint to a frame constraint could lead to the problem of learning almost identical rows, as discussed above. To prevent this issue, [182] additionally included a penalty that encourages distinct rows:

$$-\sum_k \sum_{\tilde{k}<k} \log\left(1 - (\omega_{\tilde{k}}' \omega_k)^2\right). \tag{9.1}$$

One possible concern with a tight frame constraint is that it requires the filters to span all of $\mathbb{F}^N$, so every spatial frequency can pass through at least one filter. However, most images are not zero-mean and have piece-wise constant regions, so the zero frequency component is not sparse. Ref. [191] modified the tight-frame constraint to require $\mathbf{\Omega}$ to span some space (*e.g.*, the space orthogonal to the zero frequency term). Likewise, [9] extended the CAOL algorithm to include handcrafted filters, such as a zero frequency term, that can then be used or discarded when reconstructing images. In the bilevel literature, [162], [163] similarly ensured that learned filters had no zero frequency component by learning coefficients for a linear combination of filter basis vectors, rather than learning the filters directly; see Section 12.1.1.

As an alternative to imposing a strict constraint on the filters, one can penalize $\mathbf{\Omega}$ to encourage filter diversity, as in (9.1). Using a penalty has the advantage of being able to learn any size (under- or over-complete) $\mathbf{\Omega}$ and not *requiring* the filters to represent all frequencies. For example, as an alternative to the tight frame constraint, [193] proposed a version of CAOL using the following regularizer (to within scaling constants)

$$R(\mathbf{\Omega}) = \beta \left\|\mathbf{\Omega}'\mathbf{\Omega} - \boldsymbol{I}\right\|^2$$

and a unit norm constraint on the filters. Ref. [185] included a similar penalty to (9.1), but with the inner product being divided by the norm of the filters as the filters were not constrained to unit norm. All such variations on this penalty are to encourage filter diversity.

To ensure a square $\mathbf{\Omega}$ is full rank, while also encouraging it to be well-conditioned, [183] used a regularizer that includes a term of the form

$$R(\mathbf{\Omega}) = \text{-}\beta_1 \log\left(|\mathbf{\Omega}|\right).$$

The log determinant term is known as a log barrier; it forces $\mathbf{\Omega}$ to have full rank because of the asymptote of the log function. Ref. [185] includes a similar log barrier regularization term in terms of the eigenvalues of $\mathbf{\Omega}$ to ensure it is left-invertible.

As another example of a filter penalty regularizer, both [183] and [185], include the following regularization term

$$R(\mathbf{\Omega}) = \beta_2 \left\|\mathbf{\Omega}\right\|_F^2,$$

rather than constraining the norm of the filters. This Frobenius norm addresses the scale ambiguity in the analysis and transform formulations and ensures the filter coefficients do not grow too large in magnitude.

Yet another approach to encouraging filter diversity is to consider the frequency response of the set of filters. Pfister and Bresler [185] discuss different constraint options for filter banks based on convolution strides to ensure perfect reconstruction. When the stride is one and one considers circular boundary conditions, the filters can perfectly reconstruct any signal as long as they pass the N discrete Fourier transform frequencies. Tight frames satisfy this constraint, but the constraint is more relaxed than a tight frame constraint.

Section 12.1 discusses some relatively rare bilevel problems with penalties on the learned hyperparameters, but, notably, there are no constraints nor penalties on the filters in the bilevel method (Ex)! Because of its task-based nature, filters learned via the bilevel method should be those that are best for image reconstruction. Thus, one should not have to worry about redundant filters, zero filters, or filters with excessively large coefficients. This property is one of the key benefits of bilevel methods.

9.2 Transform Learning: A Simple Experiment

The model in co-sparse transform learning is that a transform matrix, when left-multiplied, sparsifies patches of a signal, *i.e.*, $\boldsymbol{T}\boldsymbol{x}_j^{\text{true}}$ tends to be sparse, where $\boldsymbol{x}_j^{\text{true}}$ is one of J training patches. The first step in most transform learning problems is thus to learn a transform that sparsifies training signals, which are assumed to be noiseless. In other words, we would like to find

$$\hat{\boldsymbol{T}} = \underset{\boldsymbol{T} \in \mathbb{T}}{\operatorname{argmin}} \sum_{j=1}^{J} \left\| \boldsymbol{T}\boldsymbol{x}_j^{\text{true}} \right\|_0 = \underset{\boldsymbol{T} \in \mathbb{T}}{\operatorname{argmin}} \left\| \boldsymbol{T}\boldsymbol{X}^{\text{true}} \right\|_0 , \tag{9.2}$$

where $\mathbb{T} \subseteq \mathbb{F}^{K \times D}$ is the user-defined set of allowable transforms, K is the number of transforms to learn, D is the patch size, and J is the number of training patches. The jth training patch is $\boldsymbol{x}_j^{\text{true}} \in \mathbb{F}^D$ and $\boldsymbol{X}^{\text{true}} \in \mathbb{F}^{D \times J}$ is a matrix with one training patch per column. To avoid trivial solutions such as the zero filter or repeated filters, $\mathbb{T} \subseteq \mathbb{F}^{K \times D}$ may be defined, *e.g.*, as the set of matrices with orthonormal rows [193]. Section 9.1 discussed other options for constraints and the corresponding penalty forms of the constraints. The co-sparse filter learning model is equivalent to (9.2) for corresponding boundary conditions on the convolution and patch extraction. Specifically, the filter perspective views each row of $\boldsymbol{T}$ as a filter, $\boldsymbol{c}_k$, where $\boldsymbol{c}_k \circledast \boldsymbol{x}$ is assumed sparse.

Although (9.2) models a transform that sparsifies the data, the problem is difficult to solve. Rather than solve (9.2) directly, engineers often relax the problem by splitting the argument into two terms [183]:

$$\hat{\boldsymbol{T}} = \underset{\boldsymbol{T} \in \mathbb{T}}{\operatorname{argmin}} \sum_{j=1}^{J} \min_{\boldsymbol{z}_j} \frac{1}{2} \left\| \boldsymbol{T}\boldsymbol{x}_j^{\text{true}} - \boldsymbol{z}_j \right\|_2^2 + \lambda \left\| \boldsymbol{z}_j \right\|_0 . \tag{9.3}$$

Instead of directly modeling that $\boldsymbol{T}$ should sparsify $\boldsymbol{x}_j^{\text{true}}$, the split version models that $\boldsymbol{T}\boldsymbol{x}_j^{\text{true}}$ is close (in a 2-norm sense) to a sparse code, $\boldsymbol{z}_j$. The tuning parameter λ trades-off enforcing sparsity of $\boldsymbol{z}_j$ (larger λ) and enforcing $\boldsymbol{T}\boldsymbol{x}_j^{\text{true}} \approx \boldsymbol{z}_j$ (smaller λ).

This section concentrates on the impact of simplifying (9.2) by introducing the sparse code variables, $\boldsymbol{z}_j$, and tuning parameter, λ, in (9.3). To do so, we consider a very simple problem to gain an intuitive understanding for how learning approaches work. Specifically, our goal is to learn a transform that sparsifies 1D piece-wise constant (PWC) signals.

This section does not purport to improve on state-of-the-art results or to offer an especially novel denoising method. Instead, this section investigates *how the structure of a learning problem impacts the learned solutions* by examining a simple class of signals. These insights could apply in more complex image reconstruction tasks. In particular, the results motivate the use of bilevel approaches.

9.2.1 Methods

The split version of transform learning in (9.3) may initially look more difficult to solve because of the additional variables and terms. However, we can use block coordinate minimization to alternatively update the expression for both variables. At iteration i, the updates are:

$$\boldsymbol{z}_j^{(i)} = \operatorname*{argmin}_{z} \frac{1}{2} \left\| \boldsymbol{T}^{(i-1)} \boldsymbol{x}_j^{\text{true}} - z \right\|_2^2 + \lambda \|z\|_0 = \text{prox}.(\boldsymbol{T}^{(i-1)} \boldsymbol{x}_j^{\text{true}}) \tag{9.4}$$

$$\boldsymbol{T}^{(i)} = \operatorname*{argmin}_{\boldsymbol{T} \in \mathbb{T}} \left\| \boldsymbol{T} \boldsymbol{X}^{\text{true}} - \boldsymbol{Z}^{(i)} \right\|_2^2, \tag{9.5}$$

where $\boldsymbol{Z} \in \mathbb{F}^{K \times J}$ contains the sparse codes in its columns.

The sparse code update (9.4) is a proximal problem. The proximal operator for the 0-norm in (9.4) is element-wise hard-thresholding [242]

$$\boldsymbol{z}_j^{(i)} = \tau.\left(\boldsymbol{T}^{(i-1)} \boldsymbol{x}_j^{\text{true}}, \sqrt{2\lambda} \right) \text{ where } \tau(y, \alpha) = \begin{cases} y & |y| > \alpha \\ 0 & \text{else.} \end{cases}$$

Hard thresholding is cheap to compute despite being non-convex. A generalization is to replace the 0-norm with a generic sparsity-encouraging function $\phi(z)$–often a convex one with a similarly easy-to-compute proximal operator.

When $\mathbb{T}$ describes matrices with orthonormal filters, the transform update (9.5) is almost a standard Procrustes problem,

$$\hat{\boldsymbol{Q}} = \operatorname*{argmin}_{\boldsymbol{Q} : \boldsymbol{Q}'\boldsymbol{Q} = \boldsymbol{Q}\boldsymbol{Q}' = \boldsymbol{I}} \|\boldsymbol{B} - \boldsymbol{Q}\boldsymbol{A}\|_{\text{F}}^2, \tag{9.6}$$

with solution $\hat{\boldsymbol{Q}} = \boldsymbol{U}\boldsymbol{V}'$ where $\boldsymbol{U}$ and $\boldsymbol{V}$ are the left and right singular vectors of $\boldsymbol{B}\boldsymbol{A}'$. However $\boldsymbol{T}$ is often rectangular, and thus not unitary. When $\boldsymbol{T}$ has orthonormal columns, (9.5) is a generalized Procrustes problem [243]. However, to compare to TV approaches, we want to consider cases where $\boldsymbol{T}$ is wide.

Alg. 1 solves (9.5) when $\boldsymbol{T}$ is wide with orthonormal rows ($K < D$) by learning a unitary, $D \times D$ transform with the last $D - K$ rows containing "dummy" (irrelevant) filters. The $\boldsymbol{T}$ update in line 8 uses the standard Procrustes problem (9.6). Mathematically, this approach defines $\tilde{\boldsymbol{T}} \in \mathbb{F}^{D \times D}$ such that the first K rows contain the filters from $\boldsymbol{T}$ and the remaining rows contain the dummy filters.

Algorithm 1 Learning a wide transform matrix with orthonormal rows. Inputs: an initialization for the transform ($\boldsymbol{T}^{(0)} \in \mathbb{F}^{K \times D}$ where $K \leq D$), a matrix of training signal patches ($\boldsymbol{X}^{\text{true}} \in \mathbb{F}^{D \times L}$), the number of iterations to perform (N), and the proximal operator of ϕ (prox).

```
procedure PROCRUSTES-WIDE($\boldsymbol{T}^{(0)}$, $\boldsymbol{X}^{\text{true}}$, $N$, prox)
    $\boldsymbol{Q}, \boldsymbol{R} = \text{qr}(\left(\boldsymbol{T}^{(0)}\right)')$                ▷ QR decomposition
    $\tilde{\boldsymbol{T}}^{(0)} = \boldsymbol{Q}'$
    for $n = 1$ to $N$ do                                        ▷ Perform $N$ iterations
        $\boldsymbol{Z} = \tilde{\boldsymbol{T}}^{(n-1)} \boldsymbol{X}^{\text{true}}$
        $\boldsymbol{Z}_{1:K,:} = \text{prox.}(\boldsymbol{Z}_{1:K,:})$
        $\boldsymbol{U}, \boldsymbol{\Sigma}, \boldsymbol{V} = \text{svd}\left(\boldsymbol{Z}\left(\boldsymbol{X}^{\text{true}}\right)'\right)$
        $\tilde{\boldsymbol{T}}^{(n)} = \boldsymbol{U}\boldsymbol{V}'$
    end for
    return $\tilde{\boldsymbol{T}}^{(N)}_{1:K,:}$                              ▷ Remove dummy rows
end procedure
```

In terms of $\tilde{\boldsymbol{T}}$, the split transform learning optimization problem (9.3) is

$$\widehat{\tilde{\boldsymbol{T}}} = \operatorname*{argmin}_{\tilde{\boldsymbol{T}} \in \tilde{\mathbb{T}}} \min_{z_j \in \mathbb{F}^D} \sum_{j=1}^{J} \frac{1}{2} \left\| \tilde{\boldsymbol{T}} \boldsymbol{x}_j^{\text{true}} - \boldsymbol{z}_j \right\|^2 + \lambda \left\| \boldsymbol{W} \boldsymbol{z}_j \right\|_0, \tag{9.7}$$

where $\boldsymbol{W} = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \end{bmatrix} \in \mathbb{R}^{K \times D}$ selects the first K elements of $\boldsymbol{z}_j$ and $\tilde{\mathbb{T}}$ is the set of $D \times D$ unitary matrices.

Once the transform is learned, the next step is to use the transform to denoise test signals. In practice, we would not have the ground truth for these signals. However, to quantify how well the learned transform performs, we typically generate test data the same way we generate training data.

After learning the transform $\hat{\boldsymbol{T}}$, the cost function for denoising a noisy test sample $\boldsymbol{y}$ generally has a data-fit term and a regularizer that corresponds to the training cost function, *e.g.*,

$$\hat{\boldsymbol{x}}(y) = \operatorname*{argmin}_{\boldsymbol{x}} \frac{1}{2} \|\boldsymbol{x} - \boldsymbol{y}\|^2 + \beta R(\boldsymbol{x}), \tag{9.8}$$

where $R(\boldsymbol{x})$ might be, much like in the training stage, $\left\|\hat{\boldsymbol{T}}\boldsymbol{x}\right\|_0$ or $\min_{\boldsymbol{z}} \left\|\hat{\boldsymbol{T}}\boldsymbol{x} - \boldsymbol{z}\right\|^2 + \alpha\phi(\boldsymbol{z})$.

9.2.2 Experiment Set-up

Our training data set is 1,024 PWC 1D signals. Each signal has 32 elements with exactly three "jumps", which are indices where the left finite difference is non-zero. The jumps are spaced by at least three elements, so that there is at most one jump in any given length-4 patch. We assume circular boundary conditions, so each signal has exactly two values. The training data is noiseless, representing the ideal supervised setting for filter learning. The two distinct values in each signal are uniformly distributed between [-1, 1]. Fig. 9.1 shows some example training signals.

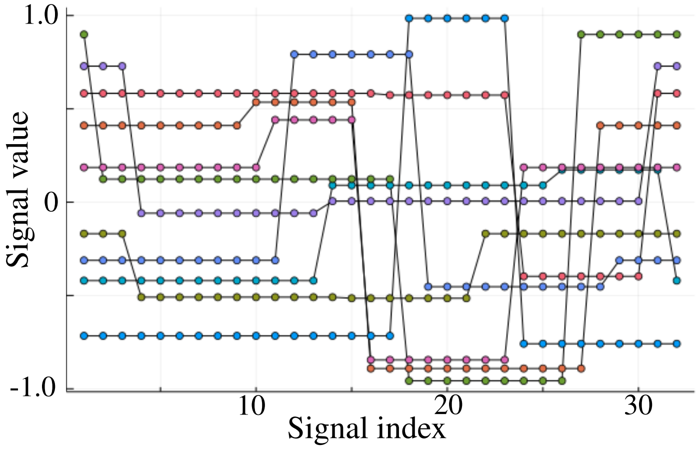


Figure 9.1: Example piece-wise constant training signals used throughout Section 9.2. The signals are in $\mathbb{R}^{32}$, but are plotted with connecting lines for easier visualization.

Because the signals are circularly symmetric, any circular shift of the columns of $\boldsymbol{T}$ will produce an equivalent transform. We do not continually state this fact, but we account for it in our calculations. For example, when we state that we compute the distance between a learned filter and a handcrafted filter, we find the minimum distance between the learned filter and all possible circularly shifted versions of the handcrafted filter.

To start, we assume that $\mathbb{T}$ is the set of single, length-4 filters with unit norm, *i.e.*, $K = 1$ and $D = 4$. The end of this section briefly discusses expanding the experiment to two orthonormal filters. We quantify disparities between learned and handcrafted filters using the formula for the angle between vectors:

$$\theta(z_1, z_2) = \cos^{-1}\left(\frac{|\langle z_1, z_2 \rangle|}{\|z_1\| \, \|z_2\|}\right). \tag{9.9}$$

Our test data is a collection of 128 signals created in the same way as the training data but with a different random seed. All test signal values are uniformly distributed over [-1, 1]. The corresponding noisy input signals mean zero Gaussian noise with a standard deviation of 0.1. We report the average root mean square error (RMSE) normalized by the expected signal strength and averaged over all testing signals,

$$\frac{1}{\mathbb{E}\{\|\boldsymbol{x}^{\text{true}}\|\}} \frac{1}{J} \sum_{j=1}^{J} \left\| \boldsymbol{x}_j^{\text{true}} - \hat{\boldsymbol{x}}(\boldsymbol{y}_j) \right\|. \tag{9.10}$$

Given the uniform distribution for the signals, $\mathbb{E}\{\|\boldsymbol{x}^{\text{true}}\|\} = \sqrt{N/3}$ where $N = 32$ is the signal dimension.

9.2.3 Effect of Introducing Sparse Code Variables

For the PWC signals considered here, we expect that the best sparsifying filter corresponds to the "Total Variation (TV) transform:"

$$\boldsymbol{T}_{\text{FD}} = \frac{1}{\sqrt{2}}\left[0 \ 1 \ \text{-}1 \ 0\right], \tag{9.11}$$

which is the minimizer of the transform learning problem (9.2). However, we empirically found that the learned transform is a slightly smoothed version of the TV filter, even when initializing with $\boldsymbol{T}_{\text{FD}}$. By smoothed, we mean that, in absolute value, the center coefficients are smaller than $\frac{1}{\sqrt{2}}$ and the edge coefficients are larger than 0, which is like using a blurring filter on $\boldsymbol{T}_{\text{FD}}$. This empirical observation was the initial inspiration for this simple experiment.

Using the training data set described above and $\boldsymbol{T}_{\text{FD}}$ as the filter initialization, we learned filters for different values of λ. The solid lines in Fig. 9.3 show the absolute value of the two larger filter coefficients versus λ. When $\lambda = 0$, the learned filter is exactly $\boldsymbol{T}_{TV}$ (more generally, it is equal to the initialization[1]). As λ increases, the learned filter becomes smoother, eventually reaching a limit. For example, when $\lambda = 0.5$, we found $\hat{\boldsymbol{T}} = \frac{1}{\sqrt{2}}\left[\text{-}0.37 \ 0.92 \ \text{-}0.93 \ 0.39\right]$.

Although the minimizer of (9.3) is not exactly $\boldsymbol{T}_{\text{FD}}$, we noticed it tended to take the approximate form

$$\left[\text{-}c \ d \ \text{-}d \ c\right] c = \pm\sqrt{\frac{1-2d}{2}}.$$

Here, $\frac{1}{\sqrt{2}} \geq |d| \geq c \geq 0$. We call this an approximate form because, as exemplified by the example $\hat{\boldsymbol{T}}$ for $\lambda = 0.5$ given above, the middle two coefficients (and the edge coefficients) are not *exactly* equal in magnitude for filters learned from training data.

In-line with these empirical observations, a grid search over the three[2] free variables in $\boldsymbol{T}$ showed that

$$\hat{\boldsymbol{T}} = \left[\text{-}\sqrt{\tfrac{1-2d^2}{2}} \ d \ \text{-}d \ \sqrt{\tfrac{1-2d^2}{2}}\right] \tag{9.12}$$

for large training sets. This learned transform is a smoothed version of the $\boldsymbol{T}_{\text{FD}}$ transform. The small difference between the $\hat{\boldsymbol{T}}$ found using a grid search and gradient descent likely stems from the non-convexity of (9.3). Without loss of generality, we can assume $\frac{1}{2} \leq d \leq \frac{1}{\sqrt{2}}$ because of the

[1] Following (9.4), and taking $\lambda = 0$, the first z_l update is hard.$(\boldsymbol{T}^{(i-1)}\boldsymbol{x}_l, 0) = \boldsymbol{T}^{(i-1)}\boldsymbol{x}_l$. Then, using (9.5), the $\boldsymbol{T}$ update is $\text{argmin}_{\boldsymbol{T}\in\mathbb{T}} \left\|\boldsymbol{T}\boldsymbol{X} - \boldsymbol{Z}^{(i)}\right\|^2 = \text{argmin}_{\boldsymbol{T}\in\mathbb{T}} \left\|\boldsymbol{T}\boldsymbol{X} - \boldsymbol{T}^{(i-1)}\boldsymbol{X}\right\|^2 = \boldsymbol{T}^{(0)}$, assuming that $\boldsymbol{T}^{(0)} \in \mathbb{T}$. The algorithm thus converges without any change from any initialization that lies in $\mathbb{T}$.

[2] The fourth filter coefficient is decided by the first three due to the unit norm constraint.

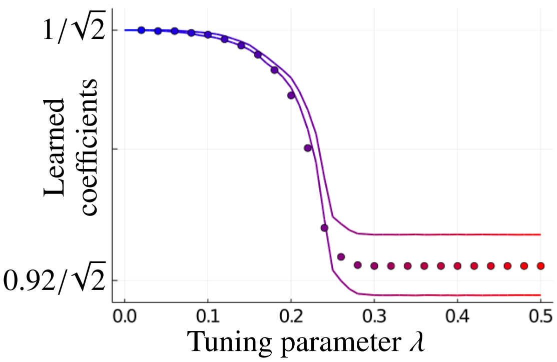


Figure 9.2: The solid lines show $\hat{T}_2$ and $|\hat{T}_3|$ versus λ, where $\hat{\boldsymbol{T}} = \left[\hat{T}_1\ \hat{T}_2\ \hat{T}_3\ \hat{T}_4\right]$ is the transform learned according to (9.3) on 1,024 PWC training signals. The points show $\hat{T}_2 = |\hat{T}_3|$ for the minimizer of the expected value of (9.3) (the value of d in (9.12)). As λ increases, the learned and expected minimizing transform becomes smoother.

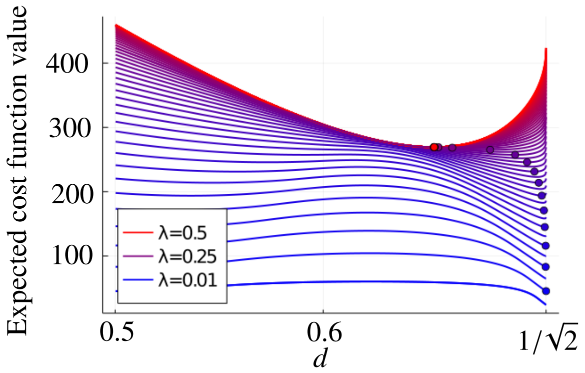


Figure 9.3: Plot of the expected value of the cost function value in (9.3) versus $T_2 = d$, where $\boldsymbol{T} = \left[-\sqrt{\frac{1-2d}{2}}\ d\ -d\ \sqrt{\frac{1-2d}{2}}\right]$. Each line corresponds to a different λ value. The points mark, $\hat{d}$, which determines the minimizer of the cost function, for each λ value, and correspond to the similarly colored points in Fig. 9.2.

circular shift invariance.

Taking (9.12) as the correct form for the glboal minimizer of (9.3), finding $\hat{\boldsymbol{T}}(\lambda)$ is a 1-D problem. Therefore, for a given λ, it is easy to sweep over d, compute the cost function, and find the global minimizer $\hat{d}$ (which fully determines $\hat{\boldsymbol{T}}$) for the class of unit-norm filters. Further, as shown in the following section, we can write down the expected value of the cost function given our assumptions about the training data then calculate, plot, and minimize this expected value.

Fig. 9.3 shows the expected value of the cost function, using the derivation in the following section, and the minimizers, $\hat{d}$, for various λ values. The points in Fig. 9.2 and Fig. 9.3 show the same $\hat{d}(\lambda)$ values with the same color mapping. As in Fig. 9.2, we can see the trend of $\hat{d}$ decreasing (the filter getting smoother) as λ increases (goes from red to blue). This behavior corresponds to the learned filter moving from $\boldsymbol{T}_{\text{FD}}$ to 22.5 degrees away from $\boldsymbol{T}_{\text{FD}}$, as seen in Fig. 9.4.

Fig. 9.2 further shows that the analytically derived values of $\hat{d}$, using the assumed form of the filter in (9.12), are the average of the empirically observed $|\hat{T}_2|$ and $|\hat{T}_3|$ elements. Finally, Fig. 9.3 demonstrates that the cost function is very flat in certain regions, especially for small λ values. The flatness could lead to a slow convergence or stopping before reaching convergence, depending on the convergence criteria.

Derivation of Smoothed TV Filter

This section derives the expected value of the cost function in (9.3) assuming the minimizer, $\hat{\boldsymbol{T}}$, takes the form (9.12). By taking the expected value, we remove the dependence on randomness in

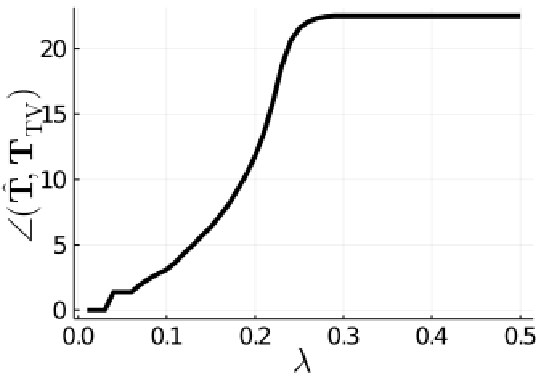


Figure 9.4: Plot of the angle (in degrees) between the learned transform and $\boldsymbol{T}_{\mathrm{FD}}$ versus the tuning parameter in (9.3).

the training samples. We can then find the expected minimizer, which is completely defined by $\hat{d}$ due to the unit norm constraint on $\boldsymbol{T}$, by minimizing the expected value of the cost function. This section is included for completeness; we used this derivation to find the $\hat{d}$ values in the previous section and confirm that they align with our empirically observed filter coefficients.

We start by proving some probability properties; these will be useful for simplifying the expected value of our cost function. A random variable is denoted by a capital (non-bold) letter, *e.g.*, X. The corresponding lower letter, x, represents one possible value of the random variable. The Probability density function (PDF) is denoted as $p_X(x)$ and the Cumulative density function (CDF) is $P_X(x) = \int_{-\infty}^{x} p_X(x)$. Recall that we assumed the values in the training signals are uniformly distributed between -1 and 1. Let a and b be two such values pulled at random from this distribution. Thus, the PDF for a is $p_a(t) = \frac{1}{2}\mathrm{rect}(t/2)$ and similarly for b. Since a and b are independent,

$$f_{a-b}(t) = (f_a \circledast f_b)(t) = \frac{1}{2}\mathrm{tri}(\frac{t}{2}).$$

Further, if $|\tau| \leq 2$,

$$\Pr[|a-b| < \tau] = \int_{-\tau}^{\tau} \frac{1}{2}\mathrm{tri}(\frac{t}{2})dt = \int_{0}^{\tau}(1 - \frac{t}{2})dt = \tau - \frac{1}{4}\tau^2. \tag{9.13}$$

Then, since $\Pr[|a-b| > \tau] = 1 - \Pr[|a-b| < \tau]$, we have that

$$\Pr[|a-b| > \tau] = (\frac{1}{2}\tau - 1)^2. \tag{9.14}$$

In addition to the two probabilities above, finding the expected value of the cost function requires the expectation of $(a-b)^2$ given that $|a-b| < \tau$. Substituting the PDF for $a-b$ into the expectation definition and simplifying yields:

$$\begin{aligned}\mathbb{E}[(a-b)^2 \| a-b| < \tau] &= \mathbb{E}[y|y < \tau] \text{ where } y = |a-b| \text{ so } f_y(t) = \text{tri}(\frac{t}{2})u(t) \\ &= \int_0^\tau y^2 \frac{f_Y(y)}{F_Y(\tau)} dy = \frac{1}{\tau - \frac{1}{4}\tau^2} \int_0^\tau y^2(1 - \frac{y}{2})dy \text{ by (9.13)} \\ &= \frac{8\tau^2 - 3\tau^3}{24 - 6\tau} \end{aligned} \tag{9.15}$$

With these probabilities defined, we now consider our training data and cost function. Because we assumed that the jumps in the training signals are spaced by at least three elements, any single length-4 patch has at most one jump. The filter sparsifies any constant patches because it has mean 0 and does not pass DC[3]. Tab. 9.1 shows the three possible non-constant patch configurations and their corresponding regularization values.

$\boldsymbol{x}_l$	$t = \boldsymbol{T}\boldsymbol{x}_j^{\text{true}}$		z_j	$\frac{1}{2}\|\boldsymbol{T}\boldsymbol{x}_l - z_l\|^2 +$	$\lambda \|z_l\|$
$\begin{bmatrix}a & a & a & b\end{bmatrix}$ or $\begin{bmatrix}a & b & b & b\end{bmatrix}$	$c(b-a)$	If $\|t\| \geq \sqrt{2\lambda}$:	$c(a-b)$	0	λ
		If $\|t\| < \sqrt{2\lambda}$:	0	$\frac{1}{2}c^2(b-a)^2$	0
$\begin{bmatrix}a & a & b & b\end{bmatrix}$	$(c-d)(b-a)$	If $\|t\| \geq \sqrt{2\lambda}$:	$(c-d)(b-a)$	0	λ
		If $\|t\| < \sqrt{2\lambda}$:	0	$\frac{1}{2}(c-d)^2(b-a)^2$	0

Table 9.1: The possible patch configurations and regularization values given the set-up described in Section 9.2.2. Here, a and b are any two values pulled from the uniform [-1, 1] distribution.

To find $\hat{d}$, we find an expression for the expected value of R in (9.3) as a function d, take the derivative and set it to zero, then solve for $\hat{d}$. The expectation averages over all possible patches, so we drop the subscript notation in the derivation. Using the values in Tab. 9.1 and the law of total

[3] DC stands for Direct Current for historical reasons. However, it has come to refer to 0 frequency or mean value of a signal.

probability, the expected value of the regularizer is:

$$\begin{aligned}
\mathbb{E}\left(R\left(\boldsymbol{x}\right)\right) &= \frac{2}{3}\,\mathbb{E}\left(R\left(\begin{bmatrix}a\ b\ b\ b\end{bmatrix}\right)\right) + \frac{1}{3}\,\mathbb{E}\left(R\left(\begin{bmatrix}a\ a\ b\ b\end{bmatrix}\right)\right)\\
&= \frac{2}{3}\left(\Pr\left(|c(b-a)| \geq \sqrt{2\lambda}\right)\cdot\lambda + \Pr\left(|c(b-a)| < \sqrt{2\lambda}\right)\cdot\mathbb{E}\left(\frac{1}{2}c^2(b-a)^2 \,\middle|\, |c(b-a)| < \sqrt{2\lambda}\right)\right) +\\
&\quad \frac{1}{3}\Big[\Pr\left(|(c-d)(b-a)| \geq \sqrt{2\lambda}\right)\cdot\lambda +\\
&\qquad \Pr\left(|(c-d)(b-a)| < \sqrt{2\lambda}\right)\cdot\mathbb{E}\left(\frac{1}{2}(c-d)^2(b-a)^2 \,\middle|\, |(c-d)(b-a)| < \sqrt{2\lambda}\right)\Big]\\
&= \frac{2}{3}\left(\Pr\left(|(b-a)| \geq \sqrt{2\lambda}/c\right)\cdot\lambda + \Pr\left(|(b-a)| < \sqrt{2\lambda}/c\right)\cdot\frac{1}{2}c^2\,\mathbb{E}\left((b-a)^2 \,\middle|\, |(b-a)| > \sqrt{2\lambda}/c\right)\right) +\\
&\quad \frac{1}{3}\Big[\Pr\left(|(b-a)| \geq \sqrt{2\lambda}/(d-c)\right)\cdot\lambda +\\
&\qquad \Pr\left(|(b-a)| < \sqrt{2\lambda}/(d-c)\right)\cdot\frac{1}{2}(c-d)^2\,\mathbb{E}\left((b-a)^2 \,\middle|\, |(b-a)| < \sqrt{2\lambda}/(d-c)\right)\Big].
\end{aligned} \tag{9.16}$$

We can substitute (9.13), (9.14), and (9.15) into the final expression to find the expected value of the regularizer as a function of only d[4].

Impact on Denoising Performance

While it is interesting from a theoretical perspective to consider which filters best sparsify a given set of signals, we typically learn sparsifying filters for a specific purpose. Here, we consider whether the sharp, handcrafted filters or smoothed, learned filters denoise a signal better.

We consider the denoising cost function:

$$\hat{\boldsymbol{x}}(\boldsymbol{y}) = \underset{\boldsymbol{x}}{\operatorname{argmin}}\ \frac{1}{2}\,\|\boldsymbol{x}-\boldsymbol{y}\|_2^2 + \beta\sum_j \min_{z_j}\left\|\boldsymbol{T}\boldsymbol{P}_j\boldsymbol{x} - z_j\right\|_2^2 + \alpha\left\|z_j\right\|_0, \tag{9.17}$$

where $\boldsymbol{P}_j$ is the matrix that extracts the jth patch from $\boldsymbol{x}$, with circular boundary conditions. The filter matrix, $\boldsymbol{T}$, is either the handcrafted filter $\boldsymbol{T}_{\text{FD}}$ defined in (9.11) or the expected best learned filter derived in the previous section, taking the form given in (9.12). We specifically test the learned filter with $d = 0.67$ (corresponding to $\lambda = 0.23$), so $\hat{\boldsymbol{T}}_{\lambda=0.23} = \begin{bmatrix}\text{-0.24 0.67 -0.67 0.24}\end{bmatrix}$.

We use Block coordinate minimization (BCM) to optimize (9.17). The $\boldsymbol{x}$ update is the solution to the least squares problem

$$\begin{aligned}
\boldsymbol{x}^{(i+1)} &= \underset{\boldsymbol{x}}{\operatorname{argmin}}\ \frac{1}{2}\,\|\boldsymbol{x}-\boldsymbol{y}\|_2^2 + \beta\sum_j \left\|\boldsymbol{T}\boldsymbol{P}_j\boldsymbol{x} - \boldsymbol{z}_j^{(i)}\right\|_2^2\\
&= \left(\boldsymbol{I} + \frac{1}{\beta}\sum_j \boldsymbol{P}_j'\boldsymbol{T}'\boldsymbol{T}\boldsymbol{P}_j\right)^{\text{-}1}\left(\boldsymbol{y} + \beta\sum_j \boldsymbol{P}_j'\boldsymbol{T}'\boldsymbol{z}_j^{(i)}\right).
\end{aligned}$$

We can further simplify the $\boldsymbol{x}$ update because $\boldsymbol{T}'\boldsymbol{T} = \boldsymbol{I}$ by the orthnormal constraint on the rows of $\boldsymbol{T}$ and $\sum_j \boldsymbol{P}_j'\boldsymbol{P}_j = D\boldsymbol{I}$

[4] The expression is quite messy, so we omit it here.

since $\boldsymbol{P}$ creates patches in a circular manner:

$$\boldsymbol{x}^{(i+1)} = \frac{1}{1+\beta D}\left(\boldsymbol{y} + \beta \sum_j \boldsymbol{P}_j' \boldsymbol{T}' \boldsymbol{z}_j^{(i)}\right).$$

The $\boldsymbol{z}_j$ update is simply hard thresholding applied to $\boldsymbol{T}\boldsymbol{P}_j\boldsymbol{x}_j^{(i+1)}$. We alternate the $\boldsymbol{x}$ and $\boldsymbol{z}$ updates until convergence, which we defined as when $\left\|\boldsymbol{x}^{(i+1)} - \boldsymbol{x}^{(i)}\right\| < 10^{-5}\,\mathbb{E}\{\|\boldsymbol{x}^{\text{true}}\|\}$.

We did a grid search over α and β (the grid search was over $\alpha \in e^{-7:0.5:-1}$ and $\beta \in e^{1:0.5:6}$) to find the tuning parameter setting that yields the lowest MSE. In practice, this would require a validation data set, but we use a test signal to get an optimistic error. For $\boldsymbol{T}_{\text{FD}}$, the best tuning parameters are $\alpha^* = e^{-4}$ and $\beta^* = e^{3.5}$. For $\hat{\boldsymbol{T}}_{\lambda=0.23}$, they are $\alpha^* = e^{-5.5}$ and $\beta^* = e^{5.0}$. Tab. 9.2 shows the resulting denoising error for the test signals. One could also run a grid search over λ, creating a bilevel transform learning problem. While this would be feasible in our simple experiment, it would be impractical for models and tasks such as those in [193].

	$\boldsymbol{T}_{\text{FD}}$	$\hat{\boldsymbol{T}}_{\lambda=0.23}$
RMSE	0.081 ± 0.035	0.131 ± 0.035

Table 9.2: Average and standard deviation of the RMSE as defined in (9.10) for the 128 denoised test signals using (9.17) with $\boldsymbol{T}$ being $\boldsymbol{T}_{\text{FD}}$ or learned according to (9.3) for $\lambda = 0.23$. Other values of λ (not shown) also yield higher RMSE values than $\boldsymbol{T}_{\text{FD}}$. The (smoothed) learned transform yields denoised signals with, on average, 38% more RMSE than $\boldsymbol{T}_{\text{FD}}$.

Fig. 9.5 shows a segment of the true signal, noisy signal, and the denoised signal using the two different filters. The handcrafted filter does a better job at reconstructing a constant signal (see in particular the first two constant segments of the signal). However, both filters struggle to reconstruct the smaller jumps in the middle of the signal.

The learned filter achieves a lower cost function in the training stage than $\boldsymbol{T}_{\text{FD}}$, which should suggest that it is somehow "better." However, since the handcrafted filter $\boldsymbol{T}_{\text{FD}}$ results in a lower MSE for the denoised signal, we would rather use it for a denoising task. This seeming contradiction is in part due to the structure of the training cost; the introduction of sparse code variables leads to the suboptimal smoothness of the learned filters. It is also because the

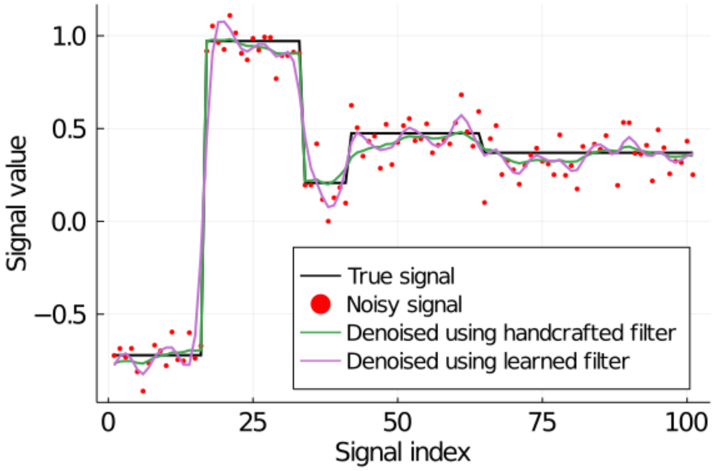


Figure 9.5: Example of how the handcrafted and learned filters perform for denoising a PWC signal.

training cost learns filters that minimize some sparsity-based cost function for the training signals, not filters that are good at denoising signals! This observation naturally leads to the idea of task-based training, where one learns filters designed for a particular purpose, such as denoising. In Chapter 11, we compare the techniques used in this section to task-based filter learning approaches.

9.2.4 Expanding to Two Filters

In this section, we consider minimizing (9.3) where now the possible transforms, $\mathbb{T}$, is the set of transform matrices with two orthonormal rows. With two filters and the orthogonality constraint, it is more difficult to guess the global minimizer or to prove a specific form the learned filters take. However, we hypothesize the filters will be some combination of $\boldsymbol{T}_{\text{FD}}$ and

$$\boldsymbol{T}_{\text{EFD}} = \begin{bmatrix} 0.5 & 0.5 & \text{-}0.5 & \text{-}0.5 \end{bmatrix}, \tag{9.18}$$

which we call, Extended Finite Difference (EFD). For example, good handcrafted filter options for sparsifying the data might be

$$\boldsymbol{T}_1 = \begin{bmatrix} \frac{\text{-}1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 \\ 0.5 & 0.5 & \text{-}0.5 & \text{-}0.5 \end{bmatrix} \text{ or } \boldsymbol{T}_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} \text{-}1 & 1 & 0 & 0 \\ 0 & 0 & \text{-}1 & 1 \end{bmatrix}. \tag{9.19}$$

The second option is the TV filter repeated twice. Note that even though the first and second filters in $\boldsymbol{T}_2$ are effectively the same, they are shifted such that they are still orthogonal. Thus, we see an example of how the orthogonality constraint does not necessarily promote useful filter diversity!

We compare these handcrafted filters to filters learned using the same BCM approach as in the previous section (see equations (9.5) and (9.4) for the iterative updates). We used the data set described in Section 9.2.2 for learning filters. For each value of λ that we tested, we created 100 random filter initializations and ran BCM on (9.3) for up to 5,000 iterations or until convergence (defined as when the norm of the change in $\boldsymbol{T}$ is less than 10^{-4}).

We use two metrics to quantify how close the learned filters are to the handcrafted filters, both from Rubinstein, Peleg, and Elad [244]. The first metric,

$$\frac{1}{2} \sum_{i=1}^{2} \min_{a} |1 - \tilde{\boldsymbol{T}}_i' \boldsymbol{T}_a|,$$

finds the angle between each of the learned filters and the closest handcrafted filter in $\boldsymbol{T}$ then averages over the learned filters. The second metric is the percentage of learned filters that match one of the handcrafted filters. We consider two filters to match if they are less than 10 degrees apart. Because the problem is invariant to circular shifts of $\boldsymbol{T}$, we use all possible shifts of the two filters in $\boldsymbol{T}_{\text{FD}}$ and $\boldsymbol{T}_{\text{EFD}}$ when calculating the two metrics. We report the average metrics over all 100 random initializations.

Fig. 9.6 shows the average angle between the learned filters and the two handcrafted filters for various λ values. Here, lower values suggest smaller angles and therefore more similar filters. The difference between the blue and orange curves shows that the learned filters are, on average, much closer to $\boldsymbol{T}_{\text{FD}}$ than $\boldsymbol{T}_{\text{EFD}}$.

Fig. 9.7 shows the average percentage of learned filters that match the handcrafted filters versus λ. Here, larger values are better. Again, we more often learn $\boldsymbol{T}_{\text{FD}}$ than $\boldsymbol{T}_{\text{EFD}}$, but this figure shows that, for some values of λ, roughly 10% of the learned filters are close to $\boldsymbol{T}_{\text{EFD}}$.

Fig. 9.7, shows that setting λ too low or too high leads to fewer matching filters. Based on our previous results learning a single filter, we hypothesized that (1) filters learned with small values of λ might be closer to their random

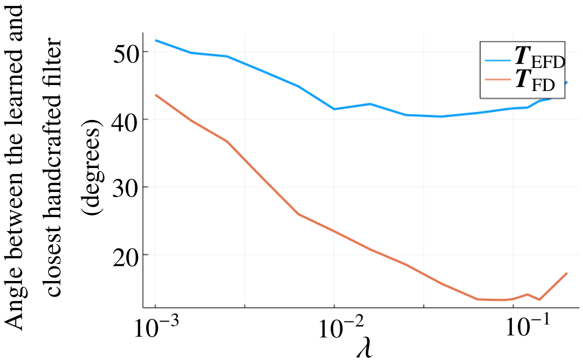


Figure 9.6: The average angle between the learned filters and $\boldsymbol{T}_{\text{FD}}$ and $\boldsymbol{T}_{\text{EFD}}$ versus λ, averaged over 100 random initialization. Smaller values suggest the learned filters are closer to the handcrafted filters.

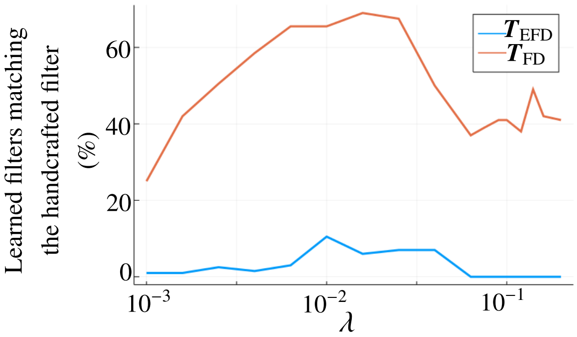


Figure 9.7: The percentage of learned filters that are within 10 degrees of $\boldsymbol{T}_{\text{FD}}$ or $\boldsymbol{T}_{\text{EFD}}$. Higher values suggest more of the learned filters are similar to the handcrafted filters.

initializations and (2) filters learned with large values of λ might be more smoothed.

As some evidence of the first hypothesis, Fig. 9.8 shows a clear correlation between $\log(\lambda)$ and the angle between the learned and initial filters. In short, filters learned with small values of λ did not move as far (measured by degrees) from their initialization.

To test our second hypothesis, we looked at filters that were within 10 degrees of $\boldsymbol{T}_{\text{FD}}$ and extracted the "equivalent d" value for these filters. This measure takes the average of the magnitude of the two largest magnitude elements after removing the mean from the filter. We call it "equivalent d" in reference to (9.12). The equivalent d for the $\boldsymbol{T}_{\text{FD}}$ filter is $\frac{1}{\sqrt{2}}$, and any smaller value represents some amount of smoothing. Fig. 9.9 shows that the equivalent d decreases for large λ values, supporting our hypothesis.

The other effect of λ is on convergence rate. As seen in Fig. 9.10, smaller values of λ typically require more iterations to converge.

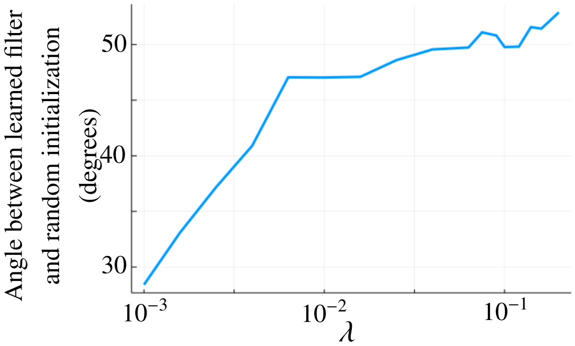


Figure 9.8: The angle between the random filter initialization and the learned filters versus λ. The correlation coefficient between $\log(\lambda)$ and the angle is 0.92.

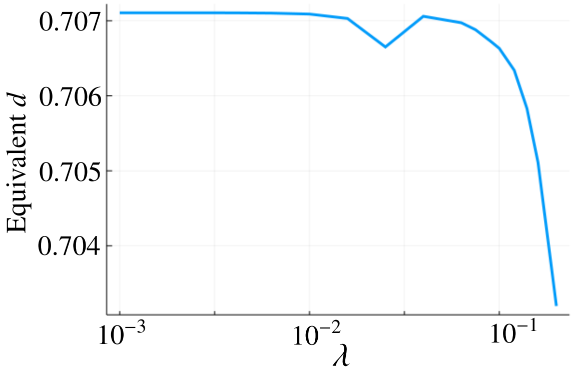


Figure 9.9: The average of the two larger magnitude elements in the learned filters after de-meaning. This plot only considers filters that are within 10 degrees of $\boldsymbol{T}_{\text{FD}}$, so larger values suggest a sharper filter (closer to $\boldsymbol{T}_{\text{FD}}$) while smaller values suggest a smoother filter.

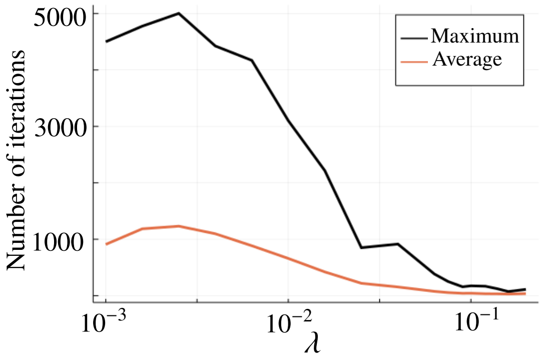


Figure 9.10: Number of iterations to learn two filters versus the tuning parameter λ in (9.3).

Above, we examined how many of the learned filters are close to the handcrafted filters. To test the usefulness of the filters, we next compared the denoising performance of the learned filters and handcrafted filters introduced in (9.19) using the same test signal as we did for testing the single filter. We chose an example learned filter with $\lambda = 0.1$ to compare to the two handcrafted filters:

$$\boldsymbol{T}_{\lambda=0.1} = \frac{1}{\sqrt{2}}\begin{bmatrix} 0.037 & 0.035 & -1.01 & 0.987121 \\ 0.987 & -1.011 & 0.035 & 0.035 \end{bmatrix}. \tag{9.20}$$

To tune α and β in (9.17), we again performed a grid search.

	$\boldsymbol{T}_1$	$\boldsymbol{T}_2$	$\hat{\boldsymbol{T}}_{\lambda=0.1}$
α^*	$e^{-4.5}$	e^{-4}	e^{-4}
β^*	$e^{1.5}$	$e^{2.5}$	e^{2}
RMSE	0.080 ± 0.033	0.081 ± 0.033	0.087 ± 0.027

Table 9.3: Tuned hyperparameter values and corresponding average and standard deviation of the RMSE as defined in (9.10) for the 128 denoised signals using (11.2) when learning two sparsifying transforms. The rows of the sparsifying transform $\boldsymbol{T}$ are the finite differencing filter shifted and repeated ($\boldsymbol{T}_1$), the finite differencing filter and the extended finite differencing filter ($\boldsymbol{T}_2$) or an example learned transform ($\hat{\boldsymbol{T}}$). The handcrafted filters are defined in (9.19) and the learned filter is defined in (9.20).

Tab. 9.3 shows the best tuning parameter values from the grid search and the RMSE of the corresponding denoised signal. For comparison, recall that the RMSE for the same signal using a single filter was 0.081 for $\boldsymbol{T}_{\text{FD}}$ and 0.131 for the learned (smoothed) version of $\boldsymbol{T}_{\text{FD}}$ with $\lambda = 0.23$. As when learning a single filter, the handcrafted filters outperform the learned filter on the denoising task. This is further motivation for the task-based learning approach. Also note that $\boldsymbol{T}_1$, which incorpoates both $\boldsymbol{T}_{\text{FD}}$ and $\boldsymbol{T}_{\text{EFD}}$, denoises the signal better than $\boldsymbol{T}_2$, which has two shifted versions of $\boldsymbol{T}_{\text{FD}}$, even though we rarely learn a transform that includes $\boldsymbol{T}_{\text{EFD}}$ as one of the two filters[5].

[5] $\boldsymbol{T}_1$ could denoise even better if we allowed the sparsity tuning parameter, α, to vary between filters.

9.2.5 Conclusions

We started this investigation to figure out why we did not learn $\boldsymbol{T}_{\text{FD}}$ when learning a single sparsifying filter on noiseless PWC training signals. Assuming that the learned filter takes the general form $\left[-\sqrt{\frac{1-2d}{2}}\ d\ -d\ \sqrt{\frac{1-2d}{2}}\right]$, we were able to derive the expected value of the cost function versus d and find the filter, fully determined by $\hat{d}$, that is expected to minimize the cost. In doing so, we showed that the filter becomes smoother as the tuning parameter in (9.3) increases. This smoothness is a direct result of approximating our original sparsity problem by one with auxiliary sparse code variables.

When we learned two filters, we similarly found that larger values of the tuning parameter, λ, tended to yield smoother filters. Further, the filters tended to both be approximations of $\boldsymbol{T}_{\text{FD}}$, but shifted to be orthonormal. For denoising, as predicted, having two $\boldsymbol{T}_{\text{FD}}$ filters produced no benefit over having a single one. Thus, while we tend to learn smoothed versions of shifted $\boldsymbol{T}_{\text{FD}}$ filters, the best filter we observed for denoising was $\boldsymbol{T}_1$, which had one filter as $\boldsymbol{T}_{\text{FD}}$ and the other as $\boldsymbol{T}_{\text{EFD}}$. Therefore, this experiment is an example of when learning filters is worse (for denoising performance) than handcrafting filters.

9.3 Handcrafted Convolutional Analyasis Operator Learning

The above section compares learned and handcrafted sparsifying filters in 1D. This section considers 2D filters for CT image reconstruction. Specifically, this section presents a generalization of the CAOL method that allows system designers to designate handcrafted filters, thus integrating domain-specific knowledge, while learning the remaining set of filters and thus adapting to the training data.

Continuing this chapter's theme of asking "when do learned filters outperform handcrafted filters", we then investigate how the number of handcrafted filters impacts training time and CT image reconstruction quality. Our numerical experiments show how handcrafting general purpose filters can trade-off between training time and CT reconstruction quality, and how handcrafting a few filters using domain-specific knowledge can lead to shorter training times while maintaining reconstruction quality. Thus, like the earlier sections of this chapter, we see the benefit of handcrafting.

9.3.1 Background: Convolutional Analysis Operator Learning

Convolutional dictionary learning (CDL) methods are reported to achieve lower redundancy in sparse representation and therefore to be more memory efficient than synthesis patch-based dictionary learning methods [245], [246]. This feature allows convolutional methods to to train from larger data-sets. Although benefits of the CDL model on its own are yet unknown in sparse-view CT model-based image reconstruction (MBIR), a combination of a "blind" CDL model and total variation (TV) penalty was successfully applied to sparse-view CT MBIR [247]. However, the model has large computational costs because it optimizes both a convolutional dictionary and corresponding sparse representations.

An alternative convolutional learning approach to CDL is convolutional *analysis* operator learning (CAOL). CAOL is more amenable to large data-sets than CDL, and has theoretical benefits from using more training samples [248]. In ill-posed inverse problems like sparse-view CT, applying learned convolutional analysis operators to MBIR has yielded significantly more accurate image reconstruction than existing MBIR with non-trained regularizers, *e.g.*, edge-preserving regularizers [249], and better generalization (and explainability) than existing non-MBIR deep neural network approaches [250]. Section 8.1.2.3 discusses the differences between regularizers that model analysis sparsity and those that model synthesis sparsity more generally.

As briefly mentioned in Section 8.1.3, CAOL [193] learns K filters, $\boldsymbol{c}_1, \dots \boldsymbol{c}_K \in \mathbb{F}^K$ from N training samples. Like the convolutional filters learned using (Ex), CAOL learns filters that sparsify, rather than synthesize, the signals. For example, in 1D, the finite difference filter $\boldsymbol{c} = [1, \ \text{-}1]$ sparsifies piecewise constant signals. The cost function for learning these filters is [193]:

$$\hat{\boldsymbol{H}} = \underset{\boldsymbol{H}}{\operatorname{argmin}} \min_{\boldsymbol{Z}} \sum_{j=1}^{J} \sum_{k=1}^{K} \frac{1}{2} \left\| \boldsymbol{c}_k \circledast \boldsymbol{x}_j - z_{k,j} \right\|_2^2 + \lambda \left\| z_{k,j} \right\|_0 \text{ s.t. } \ \boldsymbol{H}\boldsymbol{H}' = \frac{1}{K}\boldsymbol{I}, \tag{9.21}$$

where $\boldsymbol{x}_j \in \mathbb{F}^N$ is the jth training sample, $\boldsymbol{H} = [\boldsymbol{c}_1, \dots, \boldsymbol{c}_K]$ is a matrix of the vectorized filters, $z_{k,j} \in \mathbb{F}^N$ is a sparse code, and λ is a regularization parameter. Note that the matrix of filters $\boldsymbol{H}$ in CAOL is defined as the transpose of the filter matrix $\boldsymbol{\Omega}$ in (8.7); the columns of $\boldsymbol{H}$, not the rows, contain the convolutional filters that should sparsify the training data.

The tight-frame constraint in (9.21) encourages filter diversity. Without any constraint, $\boldsymbol{H} = \boldsymbol{0}$ would be a trivial solution. With a constraint such as $\|\boldsymbol{c}_k\| = 1 \ \forall k$, the same filter could be learned for all k. However, the tight-frame constraint allows for shifted versions of the same filter, which provide no additional benefit. Section 9.1 describes other constraint and penalty options for learning sparsifying filters.

Similar to the transform learning cost in Section 9.2.1, the CAOL training cost (9.21) can be optimized using block coordinate minimization (BCM). BCM alternates between minimizing with respect to $z_{k,j}$ and with respect to $\boldsymbol{C}$. The sparse code update is separable, yielding:

$$\begin{aligned} z_{k,j}^{(i+1)} &= \underset{z_{k,j}}{\operatorname{argmin}} \frac{1}{2} \left\| \boldsymbol{c}_k^{(i)} \circledast \boldsymbol{x}_j - z_{k,j} \right\|_2^2 + \alpha \left\| z_{k,j} \right\|_0 && (9.22) \\ &= \tau(\boldsymbol{c}_k^{(i)} \circledast \boldsymbol{x}_j, \sqrt{2\alpha}), && (9.23) \end{aligned}$$

where τ is the element-wise hard thresholding operator (9.6).

Letting $\boldsymbol{X}_j$ denote the data matrix for which $\boldsymbol{X}_j \boldsymbol{c}_k$ is equivalent to $\boldsymbol{x}_j \circledast \boldsymbol{c}_k$, the filter update can be written in matrix form [248]:

$$\begin{gathered} \underset{\boldsymbol{H}}{\operatorname{argmin}} \left\| \boldsymbol{X}\boldsymbol{H} - \boldsymbol{Z}^{(i+1)} \right\|_{\mathrm{F}}^2, \quad \text{s.t. } \boldsymbol{H}\boldsymbol{H}' = \frac{1}{K}\boldsymbol{I}, \text{ where} \\ \boldsymbol{X} = \begin{bmatrix} \boldsymbol{X}_1 \\ \vdots \\ \boldsymbol{X}_j \end{bmatrix} \text{ and } \boldsymbol{Z}^{(i+1)} = \begin{bmatrix} \boldsymbol{z}_{1,1}^{(i+1)} \dots \boldsymbol{z}_{K,1}^{(i+1)} \\ \vdots \\ \boldsymbol{z}_{1,J}^{(i+1)} \dots \boldsymbol{z}_{K,J}^{(i+1)} \end{bmatrix}. \end{gathered} \tag{9.24}$$

Defining $\boldsymbol{Q} = \sqrt{K}\boldsymbol{H}'$, $\boldsymbol{B} = \left(\boldsymbol{Z}^{(i+1)}\right)'$, and $\boldsymbol{A} = \frac{1}{\sqrt{K}}\boldsymbol{X}'$ yields the standard Procrustes problem [251]:

$$\begin{aligned} \hat{\boldsymbol{Q}} &= \underset{\boldsymbol{Q}}{\operatorname{argmin}} \|\boldsymbol{B} - \boldsymbol{Q}\boldsymbol{A}\|_{\mathrm{F}}^2, \quad \text{s.t. } \boldsymbol{Q}'\boldsymbol{Q} = \boldsymbol{I} && (9.25) \\ &= \boldsymbol{U}\boldsymbol{V}', \quad \text{where } \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}' = \operatorname{svd}(\boldsymbol{B}\boldsymbol{A}'). \end{aligned}$$

Therefore, the update equation for the filters is

$$\boldsymbol{H}^{(i+1)} = \frac{1}{\sqrt{K}}\boldsymbol{U}\boldsymbol{V}', \text{ where } \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}' = \operatorname{svd}\left(\boldsymbol{X}'\boldsymbol{Z}^{(i+1)}\right). \tag{9.26}$$

The BCM algorithm for the CAOL problem alternates between the sparse code update (9.22) and the filter update

(9.26), where memory efficient implementations form $\boldsymbol{X}'\boldsymbol{Z}^{(i+1)}$ incrementally to avoid storing all the sparse codes. See [193] for details about initialization, stopping criteria, and a generalization to a non-square filter matrix.

9.3.2 Motivation

The tight-frame constraint in (9.21) means that the learned filter bank, $\boldsymbol{H}$, passes all spatial frequencies. However, for certain applications, such as CT, we know that the images are non-negative. Therefore, the DC (0-frequency) component will not be sparse. This causes a model mis-match; CAOL models that the DC component is sparse, but we know it is not for applications like CT.

This can cause problems when we attempt to use the learned filters for, *e.g.*, image reconstruction. Consider one possible reconstruction cost function using learned sparsifying filters, $\boldsymbol{c}_k$,

$$\hat{\boldsymbol{x}} = \underset{\boldsymbol{x}}{\operatorname{argmin}} \frac{1}{2} \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|^2 + \beta \min_{z_k} \sum_{k=1}^{K} \frac{1}{2} \|\boldsymbol{c}_k \circledast \boldsymbol{x} - z_k\|_2^2 + \alpha_k \|z_k\|_0 ,$$

where $\boldsymbol{A}$ is the system matrix and $\boldsymbol{y}$ is the observed signal. A straightforward solution to the model mis-match problem would be to set $\alpha_k = 0$ for whichever k value(s) captures the DC component. However, this requires knowing which filters capture the DC component and, if multiple filters capture varying amounts of the DC component, it could lead to us discarding other useful information captured by those filters.

The initial impetus for creating a handcrafted filter generalization of CAOL was to guarantee that the first filter in $\boldsymbol{H}$ fully captured the DC component. Then, for a reconstruction problem set-up like the one above, it is simple to set $\alpha_1 = 0$ and not enforce any sparsity in the DC component for the reconstruction. The derivation below generalizes this idea to allow an arbitrary filter (or a collection of orthogonal filters) to be handcrafted.

9.3.3 Derivation

This section defines an efficient approach to CAOL with handcrafted filters (CAOL-HF): a modification to CAOL that constrains the first P filters to be handcrafted ("predefined") and learns the remaining $L = K - P$ filters from training data.

Using the standard Procrustes variables and the same mapping of variables we introduced in Section 9.3.1, we assume the initialization $\boldsymbol{Q}^{(0)}$ has the scaled handcrafted filters in the first P rows and satisfies the tight frame constraint. To incorporate handcrafted filters, we include an additional constraint in the filter update:

$$\underset{\boldsymbol{Q}_L}{\operatorname{argmin}} \ \|\boldsymbol{B} - \boldsymbol{Q}\boldsymbol{A}\|_{\mathrm{F}}^2, \quad \text{s.t. } \boldsymbol{Q}'\boldsymbol{Q} = \boldsymbol{I} \text{ and } \boldsymbol{Q} = \begin{bmatrix} \boldsymbol{Q}_P \\ \boldsymbol{Q}_L \end{bmatrix}, \tag{9.27}$$

where $\boldsymbol{Q}_P \in \mathbb{F}^{P\times K}$ contains the "predefined" filters and $\boldsymbol{Q}_L \in \mathbb{F}^{L\times K}$ contains learned filters.

The tight-frame constraint in (9.27) forces $\boldsymbol{Q}$ to be a unitary matrix. Thus, $\boldsymbol{I}_K = \boldsymbol{Q}'\boldsymbol{Q} = \boldsymbol{Q}\boldsymbol{Q}'$ if and only if

(Condition (i)) $\boldsymbol{Q}_P\boldsymbol{Q}_L' = \boldsymbol{0}$ and

(Condition (ii)) $\boldsymbol{Q}_L\boldsymbol{Q}_L' = \boldsymbol{I}_L$.

We now introduce a change of variables:

$$\boldsymbol{W} = \boldsymbol{Q}_L\left(\boldsymbol{Q}^{(0)}\right)' \iff \boldsymbol{Q}_L = \boldsymbol{W}\boldsymbol{Q}^{(0)}.$$

We use $\boldsymbol{Q}^{(0)}$ to define $\boldsymbol{W}$, but any matrix satisfying the tight frame condition and containing the handcrafted filters in the first P rows works. By condition (i) and the definition of $\boldsymbol{Q}^{(0)}$,

$$\begin{aligned}\boldsymbol{W} &= \begin{bmatrix}\boldsymbol{W}_P & \boldsymbol{W}_L\end{bmatrix} = \begin{bmatrix}\boldsymbol{Q}_L\boldsymbol{Q}'_P & \boldsymbol{Q}_L\left(\boldsymbol{Q}_L^{(0)}\right)'\end{bmatrix}\\ &= \begin{bmatrix}\boldsymbol{0}_L & \boldsymbol{Q}_L\left(\boldsymbol{Q}_L^{(0)}\right)'\end{bmatrix}.\end{aligned}$$

Furthermore, in terms of $\boldsymbol{W}$, condition (ii) becomes

$$\begin{aligned}\boldsymbol{I}_L = \boldsymbol{Q}_L\boldsymbol{Q}'_L &= \left(\boldsymbol{W}\boldsymbol{Q}^{(0)}\right)\left(\boldsymbol{W}\boldsymbol{Q}^{(0)}\right)'\\ &= \boldsymbol{W}\boldsymbol{W}' = \boldsymbol{W}_P\boldsymbol{W}'_P + \boldsymbol{W}_L\boldsymbol{W}'_L.\end{aligned}$$

Therefore, in terms of our new variable, the two conditions require that $\boldsymbol{W}_P = \boldsymbol{0}$ and $\boldsymbol{W}_L \in \mathbb{F}^{L\times L}$ be unitary.

Applying these two conditions, the minimization in terms of $\boldsymbol{W}_L$ is an orthogonal Procrustes problem:

$$\begin{aligned}\hat{\boldsymbol{W}}_L &= \underset{\boldsymbol{W}_L}{\operatorname{argmin}}\ \left\|\boldsymbol{B}_L - \boldsymbol{W}_L\boldsymbol{Q}_L^{(0)}\boldsymbol{A}\right\|_{\mathrm{F}}^2, \quad \text{s.t. } \boldsymbol{W}'_L\boldsymbol{W}_L = \boldsymbol{I}_L\\ &= \boldsymbol{U}\boldsymbol{V}', \quad \text{where } \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}' = \operatorname{svd}\left(\boldsymbol{B}_L\left(\boldsymbol{Q}_L^{(0)}\boldsymbol{A}\right)'\right),\end{aligned}$$

where $\boldsymbol{B}_L$ contains the last L rows of $\boldsymbol{B}$. Substituting for $\boldsymbol{Q}$, the final expression for the minimizer to (9.27) is:

$$\begin{aligned}\hat{\boldsymbol{Q}}_L &= \boldsymbol{W}_L\boldsymbol{Q}_L^{(0)} = \boldsymbol{U}\boldsymbol{V}'\boldsymbol{Q}_L^{(0)}, \quad \text{where}\\ \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}' &= \operatorname{svd}\left(\boldsymbol{B}_L\boldsymbol{A}'\left(\boldsymbol{Q}_L^{(0)}\right)'\right).\end{aligned} \tag{9.28}$$

Finally, substituting for the original CAOL variables, the modified filter update equation is:

$$\begin{aligned}\boldsymbol{H}_L^{(i+1)} &= \boldsymbol{H}_L^{(0)}\boldsymbol{U}\boldsymbol{V}', \quad \text{where}\\ \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}' &= \operatorname{svd}\left(\left(\boldsymbol{H}_L^{(0)}\right)'\boldsymbol{X}'\boldsymbol{Z}_L^{(i+1)}\right),\end{aligned} \tag{9.29}$$

where $\boldsymbol{Z}_L$ contains the last L columns of $\boldsymbol{Z}$ that correspond to the sparse codes of the learned filters. In this form, one can verify that the learned filters are constrained to be in the range of $\boldsymbol{H}_L^{(0)}$, which is the range orthogonal to $\boldsymbol{H}_P$. Alg. 2 summarizes CAOL-HF. We use the SVD to initialize $\boldsymbol{H}_L^{(0)}$ to satisfy the tight-frame condition.

9.3.4 Computational Benefit

One can use an accumulator to store only one sparse code at a time since the lth column of $\boldsymbol{X}'\boldsymbol{Z}_L^{(i+1)}$ is $\sum_{j=1}^{J}\boldsymbol{X}'_j z_{P+l,j}$, for $l = 1\ldots,L$. Assuming one uses an accumulator in both implementations, CAOL-HF and CAOL have the same memory complexity: $O(\min(M, KL))$, where $L = K$ for CAOL and typically $KL \ll M$.

When $K \leq MJL$ (which holds for large data sets), the per-iteration computational cost of CAOL-HF is smaller than the $O(MJK^2)$ required by CAOL. CAOL-HF avoids $O(MJPK)$ operations by not calculating $z_{k,j}$ for $k \leq P$ (Alg. 2 line 6) and again when evaluating $\boldsymbol{X}'\boldsymbol{Z}_L$ (Alg. 2 line 8). Thus, the time complexity of each CAOL-HF BCD iteration is $O(MJLK)$. The following section empirically examines the number of iterations required to reach convergence.

Algorithm 2 CAOL with handcrafted filters.

```
procedure CAOL-HF($\boldsymbol{H}_P$, tol, $I_{\max}$)
    $i = 0$
    $\boldsymbol{H}_L^{(0)} = (1/\sqrt{K})\text{null}(\boldsymbol{H}_P')$
    while $i < I_{\max}$ and $\frac{\left\|\boldsymbol{H}_L^{(i)} - \boldsymbol{H}_L^{(i-1)}\right\|}{\left\|\boldsymbol{H}_L^{(i-1)}\right\|} > \text{tol}$ do
        for $k = (P+1) : K,\ j = 1 : J$ do
            $\boldsymbol{z}_{k,j}^{(i+1)} = \tau.(\boldsymbol{c}_k^{(i)} \circledast \boldsymbol{x}_j, \alpha)$        ▹ From (9.22)
        end for
        $\boldsymbol{U\Sigma V}' = \text{svd}((\boldsymbol{H}_L^{(0)})'\boldsymbol{X}'\boldsymbol{Z}_L^{(i+1)})$
        $\boldsymbol{H}_L^{(i+1)} = \boldsymbol{H}_L^{(0)}\boldsymbol{UV}'$        ▹ From (9.29)
        $i = i + 1$
    end while
    return $\boldsymbol{H}_L^{(i)}$
end procedure
```

9.3.5 Application to Sparse-View CT

This section examines the effect of the number of handcrafted filters on training time and sparse-view CT reconstruction quality. All CT images are presented in modified Hounsfield units (HU), where air is 0 HU and water is 1000 HU. Training code is available at [252].

Training Setup and Results

The training process involves learning $\hat{\boldsymbol{H}}$ via Alg. 2 from high quality CT images. We used $J = 10$ XCAT phantom 512×512 slices [152] spaced by five slices (3.125 mm) and normalized to [0,1] (see Fig. 9.11 for example slices). We set $K = 7 \times 7$, $\lambda = 5 \cdot 10^{-4}$ (selected by visually comparing to the filters presented in [249]), a convergence tolerance of 10^{-6}, and 2000 maximum iterations.

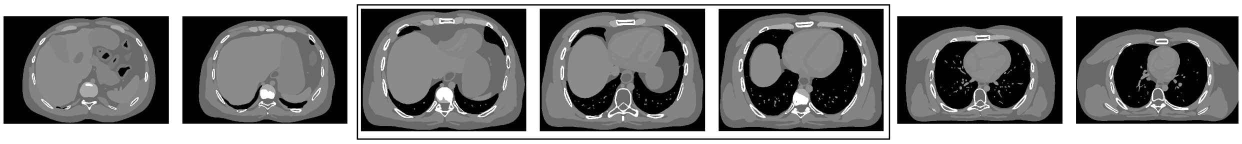


Figure 9.11: Example training and testing images, arranged by slice order in the phantom (abdominal slices are toward the left, chest slices are toward the right; display window is [800, 1200] HU). Left images: testing images 1 and 2 from the abdominal region. Center box: images from the beginning, middle, and end of the training data set. Right images: testing images 3 and 4 from the chest region.

To examine the effect of handcrafting filters for sparse-view CT image reconstruction, we used two sets of filters. First, we used the 2D Discrete Cosine Transform (DCT), ordered from low to high frequency. We learned filters for $P \in \{0, 1, 3, 6, \ldots, 43, 46, 49\}$, which is equivalent to all filters up to the ith anti-diagonal in the usual DCT arrangement. Second, we used EFD filters (*e.g.*, [1, -1] and [1, 1, -1, -1] in both the horizontal and vertical directions). To initialize the EFD filter matrix, we replaced the first nine DCT filters with our EFD filters and applied the Gram-Schmidt

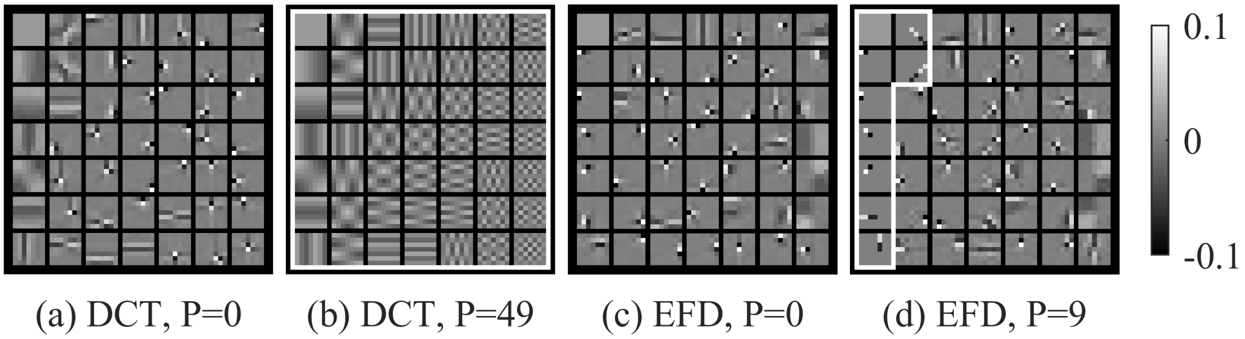


Figure 9.12: Example filters for the DCT (a-b) and EFD (c-d) cases. (a) and (c) show the case of all-learned filters ($P = 0$) while (b) and (d) show the case of the maximum number of handcrafted filters ($P = 49$ for DCT and $P = 9$ for EFD). Handcrafted filters are outlined by white borders.

procedure to obtain an orthogonal matrix. We learned filters for $P \in \{0, 1, 5, 9\}$. Fig. 9.12 shows four example $\boldsymbol{H}$'s as a grid of filters arranged in column-major order.

Fig. 9.13 shows the number of iterations and the time per iteration versus P. The time per iteration decreases linearly with P as discussed previously. The number of iterations to convergence is less predictable, but the overall trend is that the number of iterations decreases as P increases.

CT Reconstruction Formulation

Reconstruction recovers a linear attenuation coefficient image $\hat{\boldsymbol{x}} \in \mathbb{R}^N$ from a post-log measurement $\boldsymbol{y} \in \mathbb{R}^M$ [253], [254] by optimizing [193]:

$$\hat{\boldsymbol{x}} = \underset{\boldsymbol{x} \geq 0}{\operatorname{argmin}} \frac{1}{2} \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_{\boldsymbol{W}}^2 + \gamma \min_{\boldsymbol{Z}} \sum_{k=1}^{K} \frac{1}{2} \|\boldsymbol{c}_k \circledast \boldsymbol{x} - \boldsymbol{z}_k\|_2^2 + \alpha \|\boldsymbol{\psi} \odot \boldsymbol{z}_k\|_0 . \tag{9.30}$$

Here, $\odot$ is the Hadamard product, $\boldsymbol{A} \in \mathbb{R}^{M \times N}$ is the system matrix that captures CT physics [255]; $\boldsymbol{W} \in \mathbb{R}^{M \times M}$ is a diagonal matrix with $\boldsymbol{W}_{i,i} = \frac{\rho_i^2}{\rho_i + \sigma^2}$ based on a Poisson-Gaussian model for the pre-log measurements $\boldsymbol{\rho}$ with electronic readout noise variance σ^2 [253], [254]; γ and α are regularization parameters; and $\boldsymbol{\psi}$ is a binary mask that is one only inside the circle inscribing $\boldsymbol{x}$ [254], [256]. To rapidly solve (9.30) while guaranteeing convergence to a critical point, we applied the block proximal extrapolated gradient method using a majorizer [249].

Reconstruction Setup and Results

To simulate sparse-view CT sinograms, we used 4 XCAT phantom 840×840 slices [152] in 1/mm units with pixel size 0.4883 mm, using 888 detectors and 123 views (out of a possible 984 views, yielding a 87.5% reduction in radiation), an incident intensity of $1 \cdot 10^5$, and added noise with $\sigma^2 = 25$. The reconstructed image $\hat{\boldsymbol{x}}$ is 420×420 with a pixel size of 0.9766 mm. The test images are separated from the training slices by between 18.75 and 53.125 mm.

To minimize (9.30), we used $\gamma = 13 \cdot 10^6$ (suggested in [249]), $\alpha = 5 \cdot 10^{-9}$ (based on a rough grid search and a RMSE criteria), a convergence tolerance of 10^{-6}, and 5000 maximum iterations. We initialized with the conventional filtered back-projection (FBP) image with a Hamming window. We evaluated the quality of reconstructed images against the true image using RMSE inside a region of interest defined by $\boldsymbol{\psi}$. Fig. 9.15 plots the RMSE versus P and

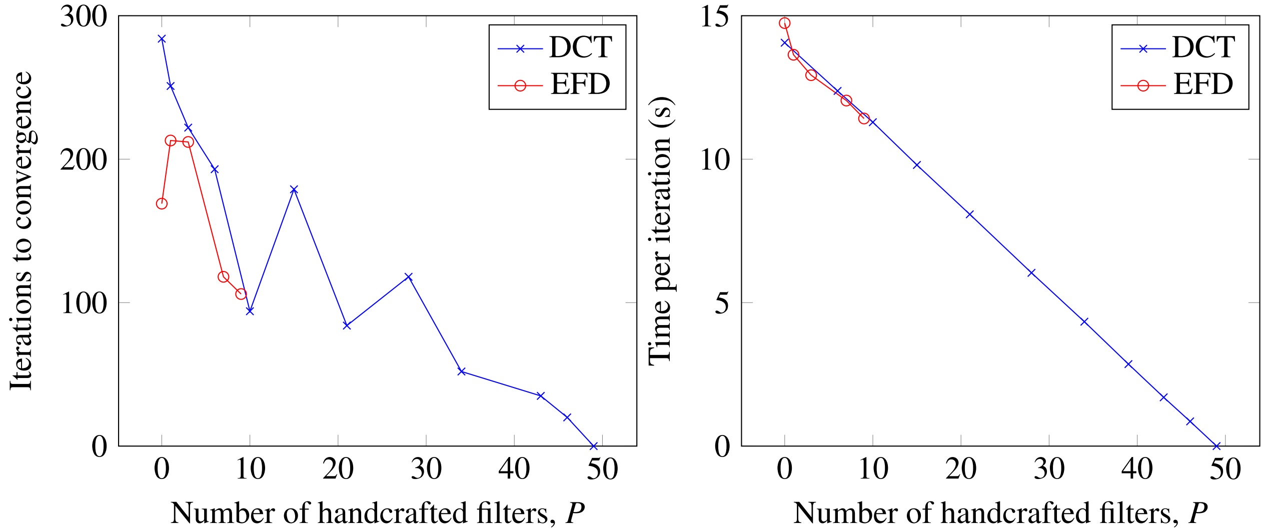


Figure 9.13: Training results for the CAOL-HF algorithm for DCT and EFD filters. (left) Number of CAOL-HF iterations required to reach convergence versus the number of handcrafted filters. (right) Time per iteration versus the number of handcrafted filters.

Fig. 9.14 shows example reconstructed images.

For comparison, Fig. 9.15 reports the RMSE of images reconstructed using a total variation (TV) regularizer with corner rounding. We implemented TV using (9.30) by replacing α with α_k, having $\boldsymbol{c}_1$ and $\boldsymbol{c}_2$ take vertical and horizontal differences and satisfy the tight-frame condition, and setting $\alpha_k = 0$ for $k \geq 3$. Based on a rough grid search, we chose $\alpha_1 = \alpha_2 = 10^{-7}$ and $\beta = 10^8$.

The DCT filter results (Fig. 9.15) suggest a trade-off: as P increases, both the iterations to convergence and reconstruction quality tend to decrease (though neither is monotonic). This trend is more noticeable for test images 3 and 4, where the RMSE increases by an average of 5.01 when comparing $P = 0$ to $P = 49$. In comparison, test images 1 and 2 have an average RMSE increase of only 0.85.

The EFD filters had lower RMSEs than the DCT filters and, unlike the DCT filters, the RMSE decreases as P increases. Unlike the DCT files, the EFD filters were designed based on our domain knowledge, so it is unsurprisingly that they perform better as handcrafted filters than the DCT filters. The fact that the RMSE is still decreasing suggests we may have been able to handcraft additional filters. We do not present the results of handcrafting additional filters because, after viewing the learned filters, it would be "cheating" to return to our initial hypothesis and handcraft additional filters. To illustrate this point, we could, of course, take the output of CAOL without any handcrafted filters, use all of those filters as "handcrafted" filters, and converge to a solution that is just as good as the learned solution with no iterations! However, engineers with more CT domain knowledge could likely construct additional handcrafted filters that would be worth testing.

Both the DCT (for small P) and EFD filters improve on the TV regularizer for images 3-4 but not for images 1-2. We hypothesize that learned filters led to lower RMSEs for images 3-4 because those images have more high-contrast regions, similar to the majority of our training data set. If we learn filters on slices similar to test images 1-2, we may outperform TV for these low-contrast images.

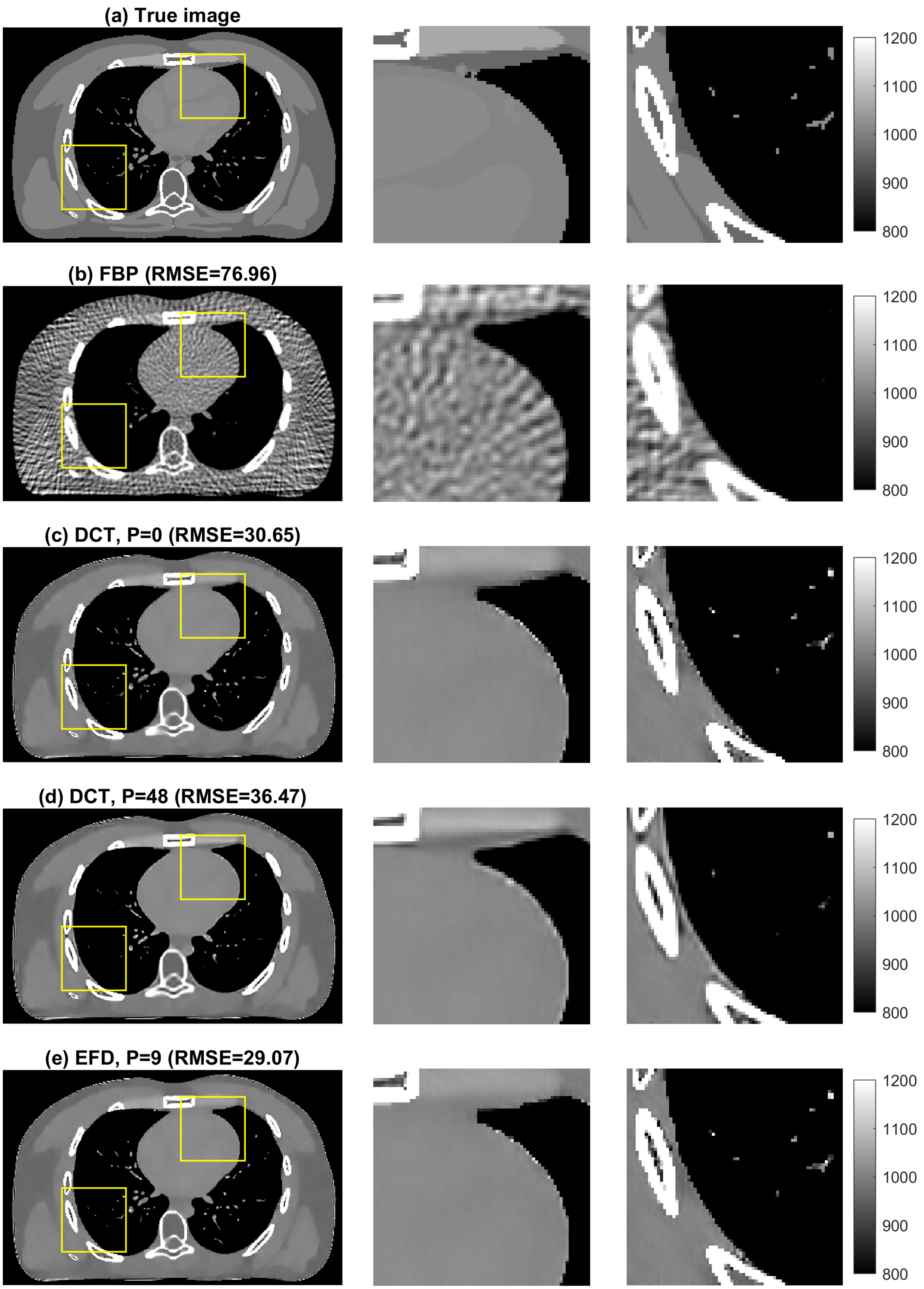


Figure 9.14: Reconstruction results for test image 3 (display window is [800, 1200] HU). The first column shows the full image and the second and third columns zoom in on the highlighted regions for easier visual comparison between the reconstructed images.

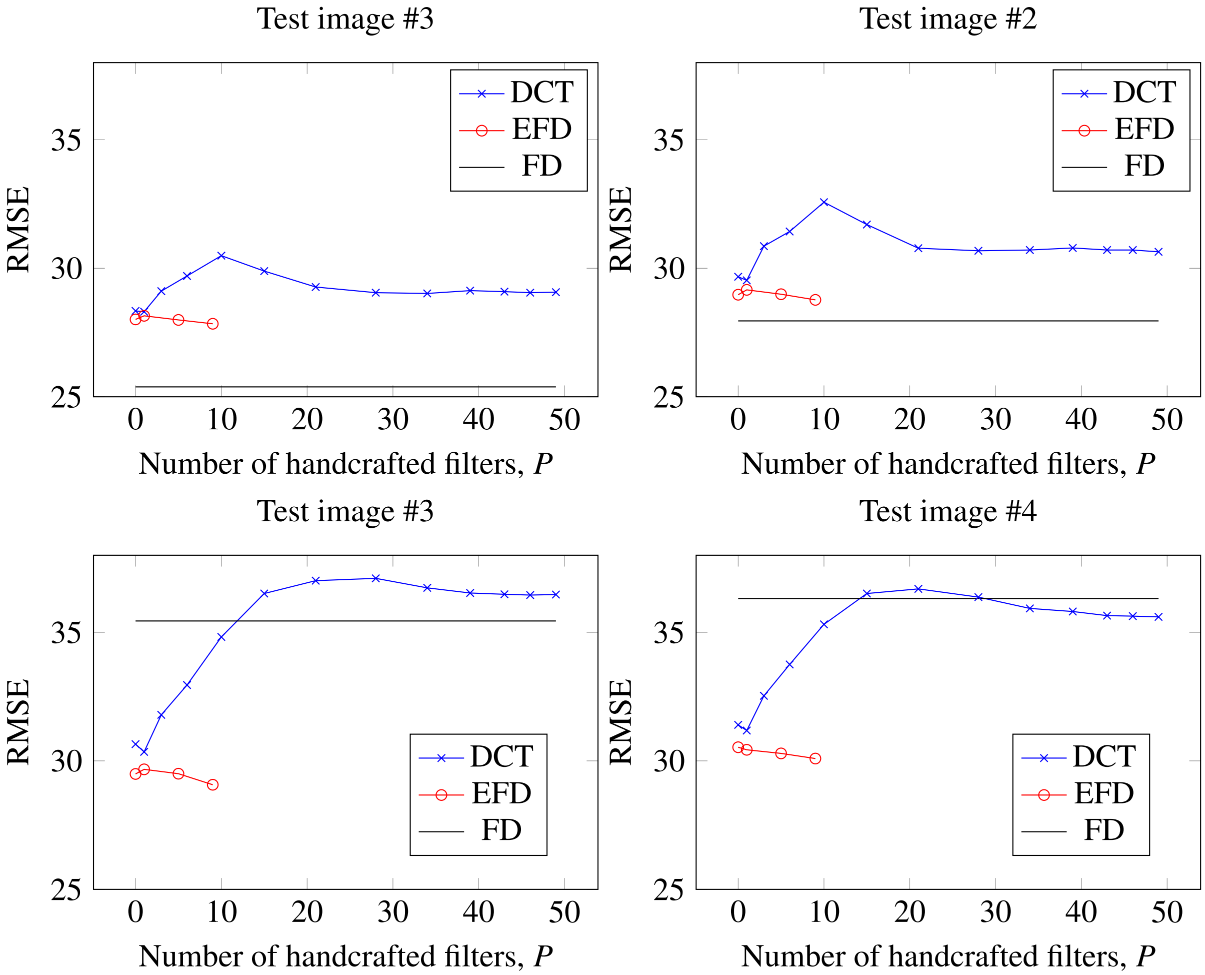


Figure 9.15: RMSE of the reconstructed CT test images versus the number of handcrafted filters, *P*. The finite differencing results are plotted for comparison, but do not vary with *P*.

9.3.6 Discussion

This section examined how incorporating handcrafted filters into CAOL affects training time and CT reconstruction quality. Our proposed algorithm, CAOL-HF, has lower per iteration time complexity as the number of handcrafted filters increases, though the overall time complexity is hard to analyze due to the varying number of iterations to convergence. We hypothesize that handcrafting well-designed filters generally leads to fewer iterations, though proving this remains future work. For reconstruction quality, we observed a decrease in quality when handcrafting DCT filters but a slight increase in quality when handcrafting EFD filters that are more appropriate for CT. Future work should consider how to design/learn filters for both high and low contrast CT slices.

Although our experiments are specific to sparse-view CT, the ideas transfer to other signal processing tasks. The presented modification of the Procrustes problem could be used in more domains to understand the trade-off between learning and handcrafting as well as to decrease training time while possibly maintaining or improving reconstruction quality.

9.4 Conclusion

Machine learning, and particularly deep learning, generally outperforms handcrafted approaches to problems [257]. However, there are still situations when a handcrafted approach is preferable. Generally, researchers may wish to select a handcrafted model when there is limited training data available or if compute power is limited. End-uses, such as doctors in the case of medical image reconstruction, may also prefer handcrafted approaches for the additional explainability and possible theoretical guarantees.

This chapter examined learned and handcrafted filters in the common setting of learning a transform that sparsifies training data. To make the problem tractable, we followed previous work [183] and incorporated sparse code variables and split the original cost function into two terms. Section 9.2 first looked at a simple 1d denoising experiment where all the training signals were noiseless and PWC. In this setting, we know the solution to the original, non-split problem is the finite differencing filter, $\boldsymbol{T}_{\text{FD}}$. However, our experimental results showed that, because of the addition of the sparse codes, the learned filters tended to be smoothed versions of the finite differencing filter. Further, when learning two filters, we found that we frequently learned two shifted versions of the finite differencing filter. Learning this filter twice makes sense since it minimizes the training cost. However, the learned information is redundant and provides no added benefit when using the filters in a denoising or reconstruction problem.

Section 9.3 then looked at the CT reconstruction with 2D filters based on the CAOL algorithm and found similar results. We were able to introduce a small number of handcrafted filters and achieve similar (or, in a few cases, better) reconstruction results.

Learning filters takes training time, and there is no reason to spend the training time to learn filters that perform worse than handcrafted filters. However, we know that machine learning techniques yield state-of-the-art results in image denoising and reconstruction. Thus, our question is not "is learning useful?"[6], but rather "*when* is learning useful?"

In answer, we claim that learning is generally useful. However, there are three cautionary notes that come from our experiments:

1. If the signals are simple enough that one can handcraft a solution to an exact model, without having to approximate the model to run an optimization algorithm, then that handcrafted solution may be best. In the case

[6] To which the answer is a resounding yes!

for our simple 1D, PWC signals, and, to a lesser extent, the PWC CT images, we could easily see the finite differencing filter would sparsify the signals. However, we cannot as easily guess what other filters would best sparsify the CT images. Training signals are often much more complicated than these relatively simple examples.

2. Model simplifications are typically necessary to solve problems. However, these simplifications, like adding the sparse code variables, change the minimizers. The new minimizers may be less well-suited for the original goal.

3. The training task determines what filters are learned. Thus, it is vital to set-up the training task to learn filters that will be useful in the end application. In our examples, the training task involved filters making the training data approximately match sparse signals. We then used the learned filters to try to separate noise (which theoretically should not be sparsified by the filters) from signal (which theoretically is sparse once filtered) in our test data set and denoise or reconstruct the original signals. However, the training set-up in this chapter did not specifically encourage the filters to perform best in our denoising and reconstruction models.

The last point motivates task-based bilevel approaches to filter learning. The following chapter describes how to learn parameters in a bilevel manner. Chapter 11 then revisits the simple experiment from Section 9.2 using the presented bilevel methods.

CHAPTER 10

RQ#6: A Review of Bilevel Methods

This chapter addresses one piece of RQ#6: What are the current trends in the literature on bilevel methods for image reconstruction? Specifically, this chapter consists of a literature review on bilevel optimization methods.

When the lower-level optimization problem (LL) has a closed-form solution, $\hat{\boldsymbol{x}}$, one can substitute that solution into the upper-level loss function (UL). In this case, the bilevel problem is equivalent to a single-level problem and one can use classic single-level optimization methods to minimize the upper-level loss. (See [192] for analysis and discussion of some simple bilevel problems with closed-form solutions for $\hat{\boldsymbol{x}}$.) This review focuses on the more typical bilevel problems that lack a closed-form solution for $\hat{\boldsymbol{x}}$.

Although there are a wide variety of optimization methods for this challenging category of bilevel problems, many methods are built on gradient descent of the upper-level loss. The primary challenge with gradient-based methods is that the gradient of the upper-level function depends on a variable that is itself the solution to an optimization problem involving the hyperparameters of interest. Section 10.1 describes two common approaches for overcoming this challenge. The first approach uses the fact that the gradient of the lower-level cost function is zero at the minimizer to compute an exact gradient at the exact minimizer. The second approach uses knowledge of the update scheme for the lower-level cost function to calculate the exact gradient for an approximation to the minimizer after a specific number of lower-level optimization steps. With this (approximation of the) gradient of the lower-level optimization variable with respect to the hyperparameters, one can compute the gradient of the upper-level loss function with respect to the hyperparameters, $\boldsymbol{\gamma}$.

Section 10.2 uses the building blocks from Section 10.1 to explain various gradient-based bilevel optimization methods. Bilevel gradient methods fall into two broad categories. Most gradient-based approaches to the bilevel problem fall under the first category: double-loop algorithms. These methods involve (i) optimizing the lower-level cost, either to some convergence tolerance if using a minimizer approach or for a certain number of iterations if using an unrolled approach, (ii) calculating the upper-level gradient, (iii) taking a gradient step in $\boldsymbol{\gamma}$, and (iv) iterating. The first step is itself an optimization algorithm and may involve many inner iterations, thus the categorization as a "double-loop algorithm." The second category, "single-loop" algorithms, involve one loop, with each iteration containing one gradient step for both the lower-level optimization variable, $\boldsymbol{x}$, and the upper-level optimization variable, $\boldsymbol{\gamma}$. Single-loop algorithms may alternate updates or update the variables simultaneously.

This material in this chapter is presented in chapters 3-4 of [11]:

> C. Crockett and J. A. Fessler, "Bilevel methods for image reconstruction," *Foundations and Trends® in Signal Processing*, vol. 15, no. 2-3, pp. 121–289, May 5, 2022, ISSN: 1932-8346, 1932-8354.
> DOI: 10.1561/2000000111

10.1 Gradient-Based Bilevel Methodology: The Groundwork

Recall from Section 7.3 that a generic bilevel problem is

$$\underset{\boldsymbol{\gamma}}{\operatorname{argmin}} \; \underbrace{\ell(\boldsymbol{\gamma}\,;\hat{\boldsymbol{x}}(\boldsymbol{\gamma}))}_{\ell(\boldsymbol{\gamma})} \text{ where } \hat{\boldsymbol{x}}(\boldsymbol{\gamma}) = \underset{\boldsymbol{x}}{\operatorname{argmin}}\, \Phi(\boldsymbol{x}\,;\boldsymbol{\gamma}). \tag{10.1}$$

For simplicity, hereafter we focus on the case $\mathbb{F} = \mathbb{R}$. Using the chain rule, the gradient of the upper-level loss function with respect to the hyperparameters is

$$\nabla\ell(\boldsymbol{\gamma}) = \nabla_{\boldsymbol{\gamma}}\ell(\boldsymbol{\gamma}\,;\hat{\boldsymbol{x}}(\boldsymbol{\gamma})) + \left(\nabla_{\boldsymbol{\gamma}}\hat{\boldsymbol{x}}(\boldsymbol{\gamma})\right)' \nabla_{\boldsymbol{x}}\ell(\boldsymbol{\gamma}\,;\hat{\boldsymbol{x}}(\boldsymbol{\gamma})), \tag{10.2}$$

where on the right hand side $\nabla_{\boldsymbol{\gamma}}$ and $\nabla_{\boldsymbol{x}}$ denote partial derivatives w.r.t. the first and second arguments of $\ell(\boldsymbol{\gamma}\,;\boldsymbol{x})$, respectively. We typically select the loss function such that it is easy to compute these partials. For example, if ℓ is the squared error training loss, *i.e.*, $\ell(\boldsymbol{\gamma}\,;\hat{\boldsymbol{x}}(\boldsymbol{\gamma})) = \frac{1}{2}\left\|\hat{\boldsymbol{x}}(\boldsymbol{\gamma}) - \boldsymbol{x}^{\text{true}}\right\|_2^2$, then

$$\nabla_{\boldsymbol{\gamma}}\ell(\boldsymbol{\gamma}\,;\hat{\boldsymbol{x}}(\boldsymbol{\gamma})) = 0 \text{ and } \nabla_{\boldsymbol{x}}\ell(\boldsymbol{\gamma}\,;\hat{\boldsymbol{x}}(\boldsymbol{\gamma})) = \hat{\boldsymbol{x}}(\boldsymbol{\gamma}) - \boldsymbol{x}^{\text{true}}.$$

The following sections survey methods to find the remaining, more challenging piece in (10.2): the Jacobian $\nabla_{\boldsymbol{\gamma}}\hat{\boldsymbol{x}}(\boldsymbol{\gamma}) \in \mathbb{F}^{N\times R}$ for a given value of $\boldsymbol{\gamma}$.

10.1.1 Minimizer Approach

The first approach finds the Jacobian $\nabla_{\boldsymbol{\gamma}}\hat{\boldsymbol{x}}(\boldsymbol{\gamma})$ by assuming the gradient of Φ at the minimizer is zero. There are two ways to arrive at the final expression: the implicit function theorem (Implicit function theorem (IFT)) perspective (as in [163], [258]) and the Lagrangian/KKT transformation perspective (as in [162], [164]). This section presents both perspectives in sequence. The end of the section summarizes the required assumptions and discusses computational complexity and memory requirements.

The first step in both perspectives is to assume we have computed $\hat{\boldsymbol{x}}(\boldsymbol{\gamma})$ and that the lower-level problem 10.1 is unconstrained (*e.g.*, no non-negativity or box constraints). Therefore, the gradient of Φ with respect to $\boldsymbol{x}$ and evaluated at $\hat{\boldsymbol{x}}$ must be zero:

$$\nabla_{\boldsymbol{x}}\Phi(\boldsymbol{x}\,;\boldsymbol{\gamma})\Big|_{\boldsymbol{x}=\hat{\boldsymbol{x}}(\boldsymbol{\gamma})} = \nabla_{\boldsymbol{x}}\Phi(\hat{\boldsymbol{x}}\,;\boldsymbol{\gamma}) = \mathbf{0}. \tag{10.3}$$

After this point, the two perspectives diverge.

10.1.1.1 Implicit Function Theorem Perspective

In the IFT perspective, we apply the IFT (*cf*. [259]) to define a function h such that $\hat{\boldsymbol{x}}(\boldsymbol{\gamma}) = h(\boldsymbol{y}, \boldsymbol{\gamma})$. If we could write h explicitly, then the bilevel problem could be converted to an equivalent single-level problem. However, per the IFT, we do not need to define h, we only state that such an h exists. Combining this definition with (10.3) yields

$$\mathbf{0} = \nabla_{\boldsymbol{x}}\Phi(h(\boldsymbol{y}, \boldsymbol{\gamma}); \boldsymbol{\gamma}). \tag{10.4}$$

Using the chain rule, we differentiate both sides of (10.4) with respect to $\boldsymbol{\gamma}$. The $\boldsymbol{I}$ in the equation below follows from the chain rule because $\nabla_{\boldsymbol{\gamma}} \boldsymbol{\gamma} = \boldsymbol{I}$. We then rearrange terms to solve for the desired quantity, noting that $\nabla_{\boldsymbol{\gamma}} \hat{\boldsymbol{x}}(\boldsymbol{\gamma}) = \nabla_{\boldsymbol{\gamma}} h(\boldsymbol{y}, \boldsymbol{\gamma})$. Thus, evaluating all terms at $\hat{\boldsymbol{x}}$ leads to the Jacobian expression of interest:

$$\begin{aligned} 0 =& \nabla_{\boldsymbol{xx}} \Phi(h(\boldsymbol{y}, \boldsymbol{\gamma}); \boldsymbol{\gamma}) \nabla_{\boldsymbol{\gamma}} h(\boldsymbol{y}, \boldsymbol{\gamma}) + \boldsymbol{I} \cdot \nabla_{\boldsymbol{x\gamma}} \Phi(h(\boldsymbol{y}, \boldsymbol{\gamma}); \boldsymbol{\gamma}) \\ \nabla_{\boldsymbol{\gamma}} h(\boldsymbol{y}, \boldsymbol{\gamma}) =& - [\nabla_{\boldsymbol{xx}} \Phi(h(\boldsymbol{y}, \boldsymbol{\gamma}); \boldsymbol{\gamma})]^{-1} \cdot \nabla_{\boldsymbol{x\gamma}} \Phi(h(\boldsymbol{y}, \boldsymbol{\gamma}); \boldsymbol{\gamma}) \\ \nabla_{\boldsymbol{\gamma}} \hat{\boldsymbol{x}}(\boldsymbol{\gamma}) =& - [\nabla_{\boldsymbol{xx}} \Phi(\hat{\boldsymbol{x}}; \boldsymbol{\gamma})]^{-1} \cdot \nabla_{\boldsymbol{x\gamma}} \Phi(\hat{\boldsymbol{x}}; \boldsymbol{\gamma}). \end{aligned} \tag{10.5}$$

When Φ is strictly convex, the Hessian of Φ is positive definite and $\nabla_{\boldsymbol{xx}} \Phi(\hat{\boldsymbol{x}}; \boldsymbol{\gamma})$ is invertible.

Substituting (10.5) into (10.2) yields the following expression for the gradient of the upper-level loss function with respect to $\boldsymbol{\gamma}$:

$$\nabla \ell(\boldsymbol{\gamma}) = \nabla_{\boldsymbol{\gamma}} \ell(\boldsymbol{\gamma}\,; \hat{\boldsymbol{x}}(\boldsymbol{\gamma})) - \left(\nabla_{\boldsymbol{x\gamma}} \Phi(\hat{\boldsymbol{x}}; \boldsymbol{\gamma})\right)' (\nabla_{\boldsymbol{xx}} \Phi\, (\hat{\boldsymbol{x}}\,; \boldsymbol{\gamma}))^{-1} \nabla_{\boldsymbol{x}} \ell(\boldsymbol{\gamma}\,; \hat{\boldsymbol{x}}).$$

If there is a closed-form solution to the lower-level problem, one can verify that the IFT gradient agrees with the analytic gradient; see [258] for examples.

10.1.1.2 KKT Conditions

In the Lagrangian perspective, (10.3) is treated as a constraint on the upper-level problem, creating a single-level problem with N equality constraints:

$$\underset{\boldsymbol{\gamma}}{\operatorname{argmin}}\, \ell(\boldsymbol{\gamma}\,; \boldsymbol{x}) \text{ subject to } \nabla_{\boldsymbol{x}} \Phi(\boldsymbol{x}\,; \boldsymbol{\gamma}) = \boldsymbol{0}_N. \tag{10.6}$$

Using the KKT conditions to transform the bilevel problem into a single-level, constrained problem is sometimes called the "KKT transformation" of the bilevel problem. This transformation relates bilevel optimization to mathematical programs with equilibrium constraints (MPEC), see [141, Ch. 12], and some authors use approaches from the broader MPEC literature to approach bilevel problems [260]. The Lagrangian corresponding to (10.6) is

$$L(\boldsymbol{x}, \boldsymbol{\gamma}, \boldsymbol{\nu}) = \ell(\boldsymbol{\gamma}\,; \boldsymbol{x}) + \boldsymbol{\nu}^T \nabla_{\boldsymbol{x}} \Phi(\boldsymbol{x}\,; \boldsymbol{\gamma})$$

where $\boldsymbol{\nu} \in \mathbb{F}^N$ is a vector of Lagrange multipliers associated with the N equality constraints in (10.6).

The Lagrange reformulation is generally well-posed because many bilevel problems, such as (Ex), satisfy the linear independence constraint qualification (LICQ) [8], [261]. The LICQ requires that the matrix of derivatives of the constraint has full row rank [261], *i.e.*,

$$\operatorname{rank}\left(\begin{bmatrix} \nabla_{\boldsymbol{x\gamma}} \Phi(\boldsymbol{x}\,; \boldsymbol{\gamma}) & \nabla_{\boldsymbol{xx}} \Phi(\boldsymbol{x}\,; \boldsymbol{\gamma}) \end{bmatrix}\right) = N.$$

Strict convexity of $\Phi(\boldsymbol{x}\,; \boldsymbol{\gamma})$ is therefore a sufficient condition for LICQ to hold. (Note the similarity to the IFT perspective, where strict convexity is sufficient for the Hessian to be invertible.) Ref. [262] explores more generally how bilevel problems relate to MPECs and when the global and local minimizers of the KKT reformulation are minimizers of the original bilevel problem.

The first Karush–Kuhn–Tucker (KKT) condition states that, at the optimal point, the gradient of the Lagrangian with respect to $\boldsymbol{x}$ must be $\mathbf{0}$. We can use this fact to solve for the vector of optimal Lagrangian multipliers, $\hat{\boldsymbol{\nu}}$:

$$\begin{aligned} \nabla_{\boldsymbol{x}} L(\hat{\boldsymbol{x}}, \boldsymbol{\gamma}, \hat{\boldsymbol{\nu}}) &= \nabla_{\boldsymbol{x}} \ell(\boldsymbol{\gamma}\,;\hat{\boldsymbol{x}}) + \nabla_{\boldsymbol{x}\boldsymbol{x}} \Phi(\hat{\boldsymbol{x}}\,;\boldsymbol{\gamma})\hat{\boldsymbol{\nu}} = \mathbf{0} \\ \hat{\boldsymbol{\nu}} &= \text{-}(\nabla_{\boldsymbol{x}\boldsymbol{x}} \Phi\,(\hat{\boldsymbol{x}}\,;\boldsymbol{\gamma}))^{\text{-}1} \nabla_{\boldsymbol{x}} \ell(\boldsymbol{\gamma}\,;\hat{\boldsymbol{x}}). \end{aligned}$$

Substituting the expression for $\hat{\boldsymbol{\nu}}$ into the gradient of the Lagrangian with respect to $\boldsymbol{\gamma}$ yields

$$\begin{aligned} \nabla_{\boldsymbol{\gamma}} L(\hat{\boldsymbol{x}}, \boldsymbol{\gamma}, \hat{\boldsymbol{\nu}}) &= \nabla_{\boldsymbol{\gamma}} \ell(\boldsymbol{\gamma}\,;\hat{\boldsymbol{x}}) + \left(\nabla_{\boldsymbol{x}\boldsymbol{\gamma}} \Phi(\hat{\boldsymbol{x}}\,;\boldsymbol{\gamma})\right)' \hat{\boldsymbol{\nu}} \\ &= \nabla_{\boldsymbol{\gamma}} \ell(\boldsymbol{\gamma}\,;\hat{\boldsymbol{x}}) - \left(\nabla_{\boldsymbol{x}\boldsymbol{\gamma}} \Phi(\hat{\boldsymbol{x}}\,;\boldsymbol{\gamma})\right)' (\nabla_{\boldsymbol{x}\boldsymbol{x}} \Phi\,(\hat{\boldsymbol{x}}\,;\boldsymbol{\gamma}))^{\text{-}1} \nabla_{\boldsymbol{x}} \ell(\boldsymbol{\gamma}\,;\hat{\boldsymbol{x}}), \end{aligned}$$

which is equivalent to the IFT result.

Ref. [164] generalized the Lagrangian approach to the case where the forward model is defined only implicitly, *e.g.*, as the solution to a differential equation. The authors write the lower-level problem as

$$\hat{\boldsymbol{x}} = \operatorname*{argmin}_{\boldsymbol{x}} \min_{\tilde{\boldsymbol{y}}} \|\boldsymbol{y} - \tilde{\boldsymbol{y}}\|_2^2 + R(\boldsymbol{x}) \text{ s.t. } e(\tilde{\boldsymbol{y}}, \boldsymbol{x}) = 0, \tag{10.7}$$

where the constraint function, e, incorporates the implicit system model. For example, when the forward model is linear ($\boldsymbol{A}\boldsymbol{x}$), taking $e(\tilde{\boldsymbol{y}}, \boldsymbol{x}) = \|\boldsymbol{A}\boldsymbol{x} - \tilde{\boldsymbol{y}}\|_2^2$ shows the equivalence of the approach here to the one in [164].

10.1.1.3 Summary of the Minimizer Approach

In summary, the upper-level gradient expression for the minimizer approach (*i.e.*, when one "exactly" minimizes the lower-level cost function) is

$$\nabla \ell(\boldsymbol{\gamma}) = \nabla_{\boldsymbol{\gamma}} \ell(\boldsymbol{\gamma}\,;\hat{\boldsymbol{x}}) - \left(\nabla_{\boldsymbol{x}\boldsymbol{\gamma}} \Phi(\hat{\boldsymbol{x}};\boldsymbol{\gamma})\right)' (\nabla_{\boldsymbol{x}\boldsymbol{x}} \Phi\,(\hat{\boldsymbol{x}}\,;\boldsymbol{\gamma}))^{\text{-}1} \nabla_{\boldsymbol{x}} \ell(\boldsymbol{\gamma}\,;\hat{\boldsymbol{x}}). \tag{10.8}$$

Thus, for a given loss function and cost function, calculating the gradient of the upper-level loss function (with respect to $\boldsymbol{\gamma}$) requires the following components all evaluated at $\boldsymbol{x} = \hat{\boldsymbol{x}}$: $\nabla_{\boldsymbol{\gamma}} \ell(\boldsymbol{\gamma}\,;\boldsymbol{x}) \in \mathbb{F}^R$, $\nabla_{\boldsymbol{x}\boldsymbol{\gamma}} \Phi(\boldsymbol{x}\,;\boldsymbol{\gamma}) \in \mathbb{F}^{N\times R}$, $\nabla_{\boldsymbol{x}\boldsymbol{x}} \Phi(\boldsymbol{x}\,;\boldsymbol{\gamma}) \in \mathbb{F}^{N\times N}$, and $\nabla_{\boldsymbol{x}} \ell(\boldsymbol{\gamma}\,;\boldsymbol{x}) \in \mathbb{F}^N$.

Continuing the specific example of learning filter coefficients and tuning parameters (Ex), the components are:

$$\begin{aligned} \nabla_{\boldsymbol{x}} \Phi(\hat{\boldsymbol{x}}\,;\boldsymbol{\gamma}) &= \boldsymbol{A}'(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}) + e^{\beta_0} \sum_{k=1}^{K} e^{\beta_k} \tilde{\boldsymbol{c}}_k \circledast \dot{\phi}.(\boldsymbol{c}_k \circledast \boldsymbol{x}; \epsilon) \\ \nabla_{\boldsymbol{x}\beta_k} \Phi(\hat{\boldsymbol{x}}\,;\boldsymbol{\gamma}) &= e^{\beta_0+\beta_k} \tilde{\boldsymbol{c}}_k \circledast \dot{\phi}.(\boldsymbol{c}_k \circledast \hat{\boldsymbol{x}}) \\ \nabla_{\boldsymbol{x} c_{k,s}} \Phi(\hat{\boldsymbol{x}}\,;\boldsymbol{\gamma}) &= e^{\beta_0+\beta_k} \left(\dot{\phi}.((\boldsymbol{c}_k \circledast \hat{\boldsymbol{x}})^{\langle \boldsymbol{s}\rangle}) + \tilde{\boldsymbol{c}}_k \circledast \left(\ddot{\phi}.(\boldsymbol{c}_k \circledast \hat{\boldsymbol{x}}) \odot \hat{\boldsymbol{x}}^{\langle \text{-}\boldsymbol{s}\rangle}\right)\right) \\ \nabla_{\boldsymbol{x}\boldsymbol{x}} \Phi(\hat{\boldsymbol{x}}\,;\boldsymbol{\gamma}) &= \boldsymbol{A}'\boldsymbol{A} + e^{\beta_0} \sum_k e^{\beta_k} \boldsymbol{C}_k' \operatorname{diag}(\ddot{\phi}.(\boldsymbol{c}_k \circledast \hat{\boldsymbol{x}}))\boldsymbol{C}_k \\ \nabla_{\boldsymbol{\gamma}} \ell(\boldsymbol{\gamma}\,;\boldsymbol{x}) &= 0 \\ \nabla_{\boldsymbol{x}} \ell(\boldsymbol{\gamma}\,;\hat{\boldsymbol{x}}) &= \hat{\boldsymbol{x}}(\boldsymbol{\gamma}) - \boldsymbol{x}^{\text{true}}. \end{aligned} \tag{10.9}$$

Here, the notation $\boldsymbol{x}^{\langle \boldsymbol{i}\rangle}$ means circularly shifting the vector $\boldsymbol{x}$ by $\boldsymbol{i}$ elements, and $c_{k,\boldsymbol{s}}$ denotes the $\boldsymbol{s}$th element of the kth filter $\boldsymbol{c}_k$, where $\boldsymbol{s}$ is a tuple that indexes each dimension of $\boldsymbol{c}_k$. Appendix E gives examples of using the $\boldsymbol{x}^{\langle \boldsymbol{i}\rangle}$ notation and derives $\nabla_{c_{k,s}} (\tilde{\boldsymbol{c}}_k \circledast f.(\boldsymbol{c}_k \circledast \boldsymbol{x}))$, which is the key step to expressing $\nabla_{\boldsymbol{x} c_{k,s}} \Phi(\hat{\boldsymbol{x}}\,;\boldsymbol{\gamma})$. The other components follow directly from $\nabla_{\boldsymbol{x}} \Phi(\hat{\boldsymbol{x}}\,;\boldsymbol{\gamma})$ using standard gradient tools for matrix expressions [263].

The minimizer approach to finding $\nabla\ell(\boldsymbol{\gamma})$ uses the following assumptions:

1. Both the upper-level and lower-level optimization problems have no inequality constraints.
2. $\hat{\boldsymbol{x}}$ is the minimizer to the lower-level cost function, not an approximation of the minimizer. This constraint ensures that (10.3) holds.
3. The cost function Φ is twice-differentiable in $\boldsymbol{x}$ and differentiable with respect to $\boldsymbol{x}$ and $\boldsymbol{\gamma}$.
4. The Hessian of the lower-level cost function, $\nabla_{\boldsymbol{xx}}\Phi(\boldsymbol{x}\,;\boldsymbol{\gamma})$, is invertible; this is guaranteed when Φ is strictly convex.

The first condition technically excludes applications like CT imaging, where the image is typically constrained to be non-negative. However, non-negativity constraints are rarely required when good regularizers are used, so the resulting non-constrained image can still be useful in practice [259].

The second constraint is often the most challenging since the lower-level problem typically uses an iterative algorithm that runs for a certain number of iterations or until a given convergence criteria is met. As previously noted, if there were a closed-form solution for $\hat{\boldsymbol{x}}$, then we would not have needed to use the IFT or Lagrangian to find the partial derivative of $\hat{\boldsymbol{x}}$ with respect to $\boldsymbol{\gamma}$. Since one usually does not reach the exact minimizer, the calculated gradient will have some error in it, depending on how close the final iterate is to the true minimizer $\hat{\boldsymbol{x}}$. Thus, the practical application of this method is more accurately called Approximate Implicit Differentiation (AID) [264], [265]. Section 10.1.4 further discusses gradient accuracy.

The third condition disqualifies sparsity-promoting functions such as the 0-norm and 1-norm as choices for ϕ.

Finally, the fourth (strict convexity) condition is easily satisfied in denoising problems where $\boldsymbol{A} = \boldsymbol{I}$ whenever ϕ is convex. Common convex ϕ choices include (CR1N) and the Fair potential [266]. However, in applications like compressed sensing where $\boldsymbol{A}'\boldsymbol{A}$ is not positive definite, the strict convexity of Φ depends non-trivially on $\boldsymbol{\gamma}$. The condition is likely to hold in practice for "good" values of $\boldsymbol{\gamma}$. Specifically, if ϕ is strictly convex, then the condition will hold for any value of $\boldsymbol{\gamma}$ such that the null-space of the regularization term is disjoint from the null-space of $\boldsymbol{A}$ and the regularization parameters are sufficiently large (e^{β_k} cannot approach 0). To interpret this condition, recall that regularization helps compensate for the under-determined nature of $\boldsymbol{A}$ (Section 8.1.1). Values of $\boldsymbol{\gamma}$ that do not sufficiently "fill-in" the null-space of $\boldsymbol{A}$ will leave the lower-level cost function under-determined. The task-based nature of the bilevel problem should discourage these "bad" values, but this intuition is insufficient to claim that the minimizer approach is well-defined at all iterations. To ensure that the lower-level problem is strongly convex, one could include a term like $\|\boldsymbol{x}\|_2^2$ with a small positive regularization parameter, like is done with elastic-net regularization [267].

10.1.1.4 Computational Costs

The largest cost in computing the gradient of the upper-level loss using (10.8) is often finding (an approximation of) $\hat{\boldsymbol{x}}$. However, this cost is difficult to quantify, as the IFT approach is agnostic to the lower-level optimization methodology. To compare the bilevel gradient methods, we will later assume the cost is comparable to the gradient descent calculations used in the unrolled approach (described in Section 10.1.3). However, this is an over-estimation of the cost, as the IFT approach is not constrained to smooth lower-level updates, and one can use optimization methods with, *e.g.*, warm starts and restarts to reduce this cost.

When the lower-level problem satisfies the assumptions above, and assuming one has already found $\hat{\boldsymbol{x}}$, a straightforward approach to computing the gradient (10.8) would be dominated by the $\mathcal{O}\left(N^3\right)$ operations required to compute the Hessian's inverse. For many problems, N is large, and that matrix inversion is infeasible due to computation or memory requirements. Instead, as described in [268], one can use a conjugate gradient (Conjugate gradient (CG)) method to compute the matrix-vector product

$$(\nabla_{\boldsymbol{xx}} \Phi\,(\hat{\boldsymbol{x}}\,;\boldsymbol{\gamma}))^{-1} \nabla_{\boldsymbol{x}} \ell(\boldsymbol{\gamma}\,;\hat{\boldsymbol{x}}) \tag{10.10}$$

because the Hessian is symmetric and positive definite (see assumption #4 in the previous section). For a generic $\boldsymbol{A}$, each CG iteration requires multiplying the Hessian by a vector, which has a computational complexity that is $\mathcal{O}\left(N^2\right)$.

CG takes N iterations to converge fully (ignoring finite numerical precision), so the final complexity is still $\mathcal{O}\left(N^3\right)$ in general. However, the Hessian often has a special structure that simplifies computing the matrix-vector product. Consider the running example of learning filters per (Ex). The Hessian, as given in (10.9), multiplied with any vector $\boldsymbol{v} \in \mathbb{F}^N$ is

$$\nabla_{\boldsymbol{xx}} \Phi(\hat{\boldsymbol{x}}; \boldsymbol{\gamma}, \boldsymbol{y}) \cdot \boldsymbol{v} = \underbrace{\boldsymbol{A}'(\boldsymbol{A}\boldsymbol{v})}_{2N^2} + e^{\beta_0} \sum_k e^{\beta_k} \underbrace{\boldsymbol{C}_k' \cdot}_{NS} \overbrace{\operatorname{diag}(\ddot{\phi}.(\underbrace{\boldsymbol{c}_k \circledast \hat{\boldsymbol{x}}}_{NS}))\cdot}^{N} \underbrace{(\boldsymbol{C}_k \boldsymbol{v})}_{NS}\,. \tag{10.11}$$

The annotations show the multiplications required for each component, where we used the simplifying assumption that the number of measurements matches the number of unknowns ($M = N$).

As written, (10.11) does not make any assumptions on $\boldsymbol{A}$, so the first term is still computationally expensive. If $\boldsymbol{A}$ is the identity matrix (as in denoising), the N^2 term could instead be zero cost. If $\boldsymbol{A}'\boldsymbol{A}$ is circulant, *e.g.*, if $\boldsymbol{A}$ is a MRI sampling matrix that can be written in terms of a discrete Fourier transform, then the cost is $N\log(N)$. More generally, the computational cost for one (of N) iterations of CG is $\mathcal{O}(c_{\boldsymbol{A}} N)$ where $c_{\boldsymbol{A}} \in [0, N]$ is some constant dependent on the structure of $\boldsymbol{A}$.

For the second addend in (10.11), we assume that $S \ll N$, so direct convolution is most efficient and the matrix-vector product requires $\mathcal{O}(NS)$ multiplies. When the filters are relatively large, one can use Fourier transforms for the filtering, and the cost is $\mathcal{O}(N\log(N))$. The final cost of the Hessian-vector product for (Ex) is $\mathcal{O}(c_{\boldsymbol{A}} N + RN)$. This cost includes a multiplication by K to account for the sum over all filters, which simplifies since SK is[1] $\mathcal{O}(R)$.

If N is small enough that storing the inverse Hessian is feasible, then one can estimate the Hessian inverse rather than computing it directly. Consider using a quasi-Newton algorithm to find $\hat{\boldsymbol{x}}$, which involves estimating the inverse Hessian as a pre-conditioning matrix for the gradient steps. This inverse Hessian estimate can be "shared" to efficiently approximate the inverse Hessian-vector product in (10.8) [219]. Ref. [269] used this strategy and also incorporated information from the upper-level loss function to improve the estimated inverse Hessian vector product while maintaining the super-linear convergence rate of the quasi-Newton algorithm.

10.1.2 Translation to a Single-Level

Before discussing the other widely used approach to calculating the gradient of the upper-level loss, we summarize a specialized approach for 1-norm regularizers. Like the minimizer approach described above, this approach assumes we have computed an (almost) exact minimizer of the lower-level cost function. It writes the minimizer as an (almost everywhere) differentiable function in terms of that $\hat{\boldsymbol{x}}$, then substitutes this expression for the minimizer into the upper-level loss to create a single-level optimization problem that is suitable for one hyperparameter update step.

Ref. [270] proposed the translation to a single-level approach to solve a bilevel problem with both synthesis and analysis operators. Refs. [271], [272] more recently presented versions specific to analysis operators. The bilevel

[1] The full parameter dimension includes the filters and tuning parameters, so $R = S(K + 1) + 1$.

problem considered in [271], [272] is:

$$\begin{aligned}&\underset{\boldsymbol{\gamma}}{\operatorname{argmin}} \sum_j \frac{1}{2} \|\hat{\boldsymbol{x}}_j(\boldsymbol{\gamma}) - \boldsymbol{x}_j^{\text{true}}\|_2^2 \\ \hat{\boldsymbol{x}}_j(\boldsymbol{\gamma}) = &\underset{\boldsymbol{x} \in \mathbb{F}^N}{\operatorname{argmin}} \frac{1}{2} \left\|\boldsymbol{x} - \boldsymbol{y}_j\right\|_2^2 + \left\|\boldsymbol{\Omega}_{\boldsymbol{\gamma}} \boldsymbol{x}\right\|_1, \end{aligned} \tag{10.12}$$

where $\boldsymbol{\Omega}_{\boldsymbol{\gamma}} \in \mathbb{F}^{F \times N}$ is a matrix constructed based on $\boldsymbol{\gamma}$. We write $\boldsymbol{\Omega}$ without the $\boldsymbol{\gamma}$ subscript and $\hat{\boldsymbol{x}}_j(\boldsymbol{\gamma})$ without the j subscript in the following discussion to simplify notation. As in the minimizer approach, the first step is to compute $\hat{\boldsymbol{x}}(\boldsymbol{\gamma})$ for the current guess of $\boldsymbol{\gamma}$, *e.g.*, using ADMM. After optimizing for $\hat{\boldsymbol{x}}(\boldsymbol{\gamma})$, [271], [272] both used the known sign pattern of the filtered signal, $\boldsymbol{\Omega}\hat{\boldsymbol{x}}(\boldsymbol{\gamma})$ to rewrite the lower-level problem (10.12) in a simpler, (almost everywhere) differentiable form. By rewriting the problem, the translation to a single-level approaches handle the non-smooth 1-norm in (10.12) directly–they do not require any corner rounding as in the minimizer approach.

One way to rewrite the lower-level problem is to split the 1-norm into its positive and negative elements, *e.g.*,

$$\|\boldsymbol{\Omega}\hat{\boldsymbol{x}}(\boldsymbol{\gamma})\|_1 = \sum_{i \in \mathcal{I}_+(\boldsymbol{\gamma})} [\boldsymbol{\Omega}\hat{\boldsymbol{x}}(\boldsymbol{\gamma})]_i - \sum_{i \in \mathcal{I}_-(\boldsymbol{\gamma})} [\boldsymbol{\Omega}\hat{\boldsymbol{x}}(\boldsymbol{\gamma})]_i,$$

where $\mathcal{I}_+(\boldsymbol{\gamma})$ and $\mathcal{I}_-(\boldsymbol{\gamma})$ denote the set of indices where $\boldsymbol{\Omega}\hat{\boldsymbol{x}}(\boldsymbol{\gamma})$ is positive and negative, respectively. Ref. [271] used this approach and defined a diagonal sign matrix, $\boldsymbol{S}(\boldsymbol{\gamma}) = \operatorname{diag}(\operatorname{sign}(\boldsymbol{\Omega}\hat{\boldsymbol{x}}(\boldsymbol{\gamma})))$, having positive and negative diagonal elements at the appropriate indices. For a single training image, the lower-level problem (10.12) is thus equivalent to

$$\hat{\boldsymbol{x}}(\boldsymbol{\gamma}) = \underset{\boldsymbol{x} \in \mathbb{F}^N}{\operatorname{argmin}} \frac{1}{2} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2 + \beta \boldsymbol{1}' \boldsymbol{S}(\boldsymbol{\gamma}) \boldsymbol{\Omega} \boldsymbol{x}, \text{ s.t. } [\boldsymbol{\Omega}\boldsymbol{x}]_{\mathcal{I}_0(\boldsymbol{\gamma})} = \boldsymbol{0}, \tag{10.13}$$

where $\mathcal{I}_0(\boldsymbol{\gamma})$ denotes the set of indices where $[\boldsymbol{\Omega}\hat{\boldsymbol{x}}(\boldsymbol{\gamma})]_i = 0$. The rewritten problem (10.13) it is a quadratic cost function with a linear equality constraint and thus has a closed-form solution. Ref. [271] states that $\hat{\boldsymbol{x}}(\boldsymbol{\gamma})$ is differentiable everywhere except a set of measure zero when $\boldsymbol{A} = \boldsymbol{I}$ and when the rows of $\boldsymbol{\Omega}$ corresponding to $\mathcal{I}_0(\boldsymbol{\gamma})$ are linearly independent.

Another way to rewrite (10.12) uses the results from [273]. The lower-level problem (10.12) can be transformed into the dual problem

$$\min_{\boldsymbol{d} \in \mathbb{R}^F} \frac{1}{2} \left\|\text{-}\boldsymbol{\Omega}' \boldsymbol{d} + \boldsymbol{y}\right\|^2 - \frac{1}{2} \|\boldsymbol{y}\|^2 \text{ s.t. } |d_i| \leq 1 \; \forall i. \tag{10.14}$$

where the dual variable $\boldsymbol{d}$ is related to the filtered signal by

$$\boldsymbol{d}_i \in \begin{cases} \operatorname{sign}([\boldsymbol{\Omega}\boldsymbol{x}]_i) & \text{if } [\boldsymbol{\Omega}\hat{\boldsymbol{x}}]_i \neq 0 \\ [\text{-}1, 1] & \text{if } [\boldsymbol{\Omega}\hat{\boldsymbol{x}}]_i = 0 \end{cases} \tag{10.15}$$

(compare to (C.9) and (C.13) in Appendix C). Ref. [273] defines boundary indices as the set of indices where the dual variable is at the edges of its allowed range: $B := \{i : |\boldsymbol{d}_i| = 1\}$. The complement to this set is $\bar{B} := \{i : |\boldsymbol{d}_i| \neq 1\}$ and contains all coordinates where $\boldsymbol{d}$ is in the interior of its allowed range. Let $\boldsymbol{\Omega}_{\boldsymbol{d}} \in \mathbb{F}^{|B| \times N}$ contain the rows of $\boldsymbol{\Omega}$ that correspond to B and similarly for $\boldsymbol{\Omega}_{\bar{B}}$. By taking the gradient of the Lagrangian of the dual formulation and then substituting the dual variable minimizer into (C.11), [273] derives the following closed-form expression for $\hat{\boldsymbol{x}}$

$$\hat{\boldsymbol{x}} = (\boldsymbol{I} - \boldsymbol{\Omega}_{\bar{B}}^{+} \boldsymbol{\Omega}_{\bar{B}}) \left(\boldsymbol{y} - \boldsymbol{\Omega}_B \operatorname{sign}(\boldsymbol{\Omega}_B \hat{\boldsymbol{x}})\right), \tag{10.16}$$

which is a projection onto the null space of $\boldsymbol{\Omega}_{\bar{B}}$. Thus, similar to splitting the 1-norm based on the sign of $\boldsymbol{\Omega}\hat{\boldsymbol{x}}$, splitting the dual variable into boundary and interior indices yields a rewritten problem with a simpler structure.

Ref. [272] used (10.16) to rewrite the lower-level problem (10.12) and then used matrix gradient relations to derive a closed-form expression for $\nabla_{\boldsymbol{\gamma}} \hat{\boldsymbol{x}}(\boldsymbol{\gamma})$. Unlike [271], the final upper-level gradient $\nabla \ell(\boldsymbol{\gamma})$ in [272] does not require that the rows of $\boldsymbol{\Omega}$ that are orthogonal to $\hat{\boldsymbol{x}}(\boldsymbol{\gamma})$ are linearly independent.

In both (10.13) and (10.16), the rewritten problem has the same minimizer as the original problem (10.12), but the reformulated problem has a simpler structure. Recall that the rewriting process requires $\hat{\boldsymbol{x}}(\boldsymbol{\gamma})$, so one cannot use this equivalence to optimize the lower-level problem. However, the closed-form expressions can be differentiated. Because of the discontinuity of the sign function, both methods require the sign pattern of $\boldsymbol{\Omega}\hat{\boldsymbol{x}}$ to be constant within a region to compute an accurate gradient [271], [272]. The authors have shown that this condition holds in various empirical settings [274].

In summary, the translation to a single-level approach involves computing $\hat{\boldsymbol{x}}$, creating a closed-form expression for $\hat{\boldsymbol{x}}$, and then differentiating the closed-form expression to compute the desired Jacobian, $\nabla_{\boldsymbol{\gamma}} \hat{\boldsymbol{x}}(\boldsymbol{\gamma})$. As in the minimizer approach, $\nabla_{\boldsymbol{\gamma}} \hat{\boldsymbol{x}}(\boldsymbol{\gamma})$ is related to the upper-level gradient by the chain rule (10.2). In terms of computation, both translation to a single-level approaches require optimizing the lower-level cost sufficiently precisely to ensure the sign pattern converges; [272] used thousands of iterations of ADMM. Ref. [272] demonstrates that evaluating the closed-form expression for $\nabla \ell(\boldsymbol{\gamma})$ is faster than using automatic differentiation tools that rely on backpropagation.

10.1.3 Unrolled Approaches

A popular approach to finding $\nabla_{\boldsymbol{\gamma}} \hat{\boldsymbol{x}}(\boldsymbol{\gamma})$ is to assume that the lower-level cost function is approximately minimized by applying T iterations of some (sub)differentiable optimization algorithm, where we write the update step at iteration $t \in [1 \dots T]$ as

$$\boldsymbol{x}^{(t)} = \Psi(\boldsymbol{x}^{(t-1)} \,; \boldsymbol{\gamma}),$$

for some mapping $\Psi : \mathbb{F}^N \mapsto \mathbb{F}^N$ that should have the fixed-point property $\Psi(\hat{\boldsymbol{x}}(\boldsymbol{\gamma}) \,; \boldsymbol{\gamma}) = \hat{\boldsymbol{x}}(\boldsymbol{\gamma})$. For example, GD has $\Psi(\boldsymbol{x} \,; \boldsymbol{\gamma}) = \boldsymbol{x} - \alpha_\Phi \nabla \Phi(\boldsymbol{x} \,; \boldsymbol{\gamma})$ for some step size α_Φ. We write the update here only in terms of $\boldsymbol{x}$; the idea easily extends to updates in terms of a state vector that allows one to include momentum terms, weights, and other accessory variables in $\boldsymbol{\gamma}$ [275].

In contrast to the two approaches described above, the "unrolled" approach no longer assumes the solution to the lower-level problem is an exact minimizer. Instead, the unrolled approach reformulates the bilevel problem (LL) as

$$\underset{\boldsymbol{\gamma}}{\operatorname{argmin}} \underbrace{\ell\left(\boldsymbol{\gamma} \,; \boldsymbol{x}^{(T)}(\boldsymbol{\gamma})\right)}_{\ell(\boldsymbol{\gamma})} \text{ s.t.} \qquad (10.17)$$

$$\boldsymbol{x}^{(t)}(\boldsymbol{\gamma}) = \Psi(\boldsymbol{x}^{(t-1)} \,; \boldsymbol{\gamma}), \quad \forall t \in [1 \dots T],$$

where $\boldsymbol{x}^{(0)}$ is an initialization, *e.g.*, $\boldsymbol{A}'\boldsymbol{y}$. One can then take the (sub)gradient of a finite number of iterations T of Ψ, hoping that $\boldsymbol{x}^{(T)}$ approximately minimizes the lower-level function Φ.

The chain rule for derivatives is the foundation of the unrolled method. The gradient of interest, $\nabla \ell(\boldsymbol{\gamma})$, depends on the gradient of the optimization algorithm step with respect to $\boldsymbol{x}$ and $\boldsymbol{\gamma}$. For readability, define the following matrices for the tth unrolled iteration

$$\boldsymbol{H}_t := \nabla_{\boldsymbol{x}} \Psi\left(\boldsymbol{x}^{(t-1)} \,; \boldsymbol{\gamma}\right) \in \mathbb{F}^{N \times N} \text{ and } \boldsymbol{J}_t := \nabla_{\boldsymbol{\gamma}} \Psi\left(\boldsymbol{x}^{(t-1)} \,; \boldsymbol{\gamma}\right) \in \mathbb{F}^{N \times R},$$

for $t \in [1, T]$. We use these letters because, when using gradient descent as the optimization algorithm, $\nabla_{\boldsymbol{x}} \Psi(\boldsymbol{x} \,; \boldsymbol{\gamma})$

is closely related to the Hessian of Φ and $\nabla_{\gamma}\Psi(\boldsymbol{x}\,;\boldsymbol{\gamma})$ is proportional to the Jacobian of the gradient[2]. Thus, when Ψ corresponds to GD, an unrolled approach involves computing the same quantities as required by the IFT approach (10.8).

By the chain rule, the gradient of (10.17) is

$$\nabla \ell(\boldsymbol{\gamma}) = \nabla_{\boldsymbol{\gamma}} \ell(\boldsymbol{\gamma}\,; \boldsymbol{x}^{(T)}) + \left(\sum_{t=1}^{T} (\boldsymbol{H}_T \cdots \boldsymbol{H}_{t+1})\, \boldsymbol{J}_t \right)' \nabla_{\boldsymbol{x}} \ell(\boldsymbol{\gamma}\,; \boldsymbol{x}^{(T)}) \in \mathbb{F}^R. \tag{10.18}$$

One can derive this gradient expression using a reverse or forward perspective, with parallels to back-propagation through time and real-time recurrent learning respectively [275]. Appendix D describes the reverse and forward approaches to unrolling.

Most unrolled implementations use the reverse-mode approach (backpropagation) due to its lower computational burden, but unrolling with reverse mode differentiation may have prohibitively high memory requirements if T is large or if the training dataset includes large images [241]. A strategy to trade-off the memory and computation requirements is checkpointing, which stores $\boldsymbol{x}$ every few iterations. Checkpointing is an active research area; see [276] for an overview. Another option is to use (some or all) reversible network layers [277] to trade off the memory and computational requirements.

The following sections overview some design decisions for unrolling and draw some parallels to unrolled methods as used in the (non-bilevel specific) machine learning literature. Section 12.2.1 further discusses the relation between bilevel problems and unrolling methods common in the broader literature.

10.1.3.1 Number of Iterations

Unlike the minimizer approach, where the goal is to run the lower-level optimization until (close to) convergence so that an optimally condition holds and one can use implicit differentiation to find $\nabla\ell(\boldsymbol{\gamma})$, most unrolling methods set the number of lower-level iterations T in advance. The set number of lower-level iterations mimics the depth of neural networks and allows a precise estimate of how much computational effort each lower-level optimization takes. The chosen number of iterations is important as, at test time, “one cannot deviate from the choice of [number of unrolled iterations] and expect good performance” [278].

Although it is generally not equal to the gradient of the original bilevel problem (UL), the unrolled gradient is exact for the reformulated problem (10.17). Therefore, when T is small enough that the lower-level optimizer is far from convergence, the unrolled method is only loosely tied to the original bilevel optimization problem. To maintain a stronger connection to the bilevel problem while avoiding setting T larger than necessary for convergence, [279] used a convergence criterion to determine the number of Ψ iterations rather than pre-specifying a number of iterations. Unrolling until convergence is also used in deep equilibrium or fixed point networks, see Section 12.2.1.

A subtle point in unrolling gradient-based methods for the lower-level cost function is that the Lipschitz constant of $\nabla_{\boldsymbol{x}}\Phi$ is a function of the hyperparameters, so the step size range that ensures convergence cannot be pre-specified. Many unrolled methods use a fixed step size alongside a fixed T and allow the learned parameters to adapt to these set values. An alternative approach is to compute a new step-size as a function of the current parameters, $\boldsymbol{\gamma}^{(u)}$, every upper-level iteration. For example, from (E.5), for a given value $\boldsymbol{\gamma}$ of the tuning parameters and filter coefficients, a

[2] When $\Psi(\boldsymbol{x}\,;\boldsymbol{\gamma}) = \boldsymbol{x} - \alpha_\Phi \nabla_{\boldsymbol{x}}\Phi(\boldsymbol{x}\,;\boldsymbol{\gamma})$, then $\nabla_{\boldsymbol{x}}\Psi(\boldsymbol{x}\,;\boldsymbol{\gamma}) = \boldsymbol{I} - \alpha_\Phi \nabla_{\boldsymbol{xx}}\Phi(\boldsymbol{x}\,;\boldsymbol{\gamma})$ and $\nabla_{\boldsymbol{\gamma}}\Psi(\boldsymbol{x}\,;\boldsymbol{\gamma}) = \text{-}\alpha_\Phi \nabla_{\boldsymbol{x\gamma}}\Phi(\boldsymbol{x}\,;\boldsymbol{\gamma})$.

Lipschitz constant of the lower-level gradient for (Ex) is

$$L = \sigma_1^2(\boldsymbol{A}) + e^{\beta_0} L_{\dot{\phi}} \sum_k e^{\beta_k} \|\boldsymbol{c}_k\|_1^2 , \tag{10.19}$$

where $L_{\dot{\phi}}$ is a Lipschitz constant for $\dot{\phi}(z)$ (for (CR1N), $L_{\dot{\phi}} = 1/\epsilon$). A reasonable step size for the classical gradient descent method would be $1/L$. It is relatively inexpensive to update this L as $\boldsymbol{\gamma}$ evolves.

The adaptive approach to setting the step size ensures that any theoretical guarantees of the lower-level optimizer hold. This approach may be beneficial when using a convergence criteria for the lower-level optimization algorithm or when running sufficiently many lower-level iterations to essentially converge. However, updating the step-size every upper-level iteration is incompatible with fixing the number of unrolled iterations. To illustrate, consider an upper-level iteration where the tuning parameters increase, leading to a larger L and a smaller step size. In a fixed number of iterations, the smaller step size means the lower-level optimization algorithm will be farther from convergence, and the estimated minimizer, $\hat{\boldsymbol{x}}(\boldsymbol{\gamma}^{(u+1)})$, may be worse (as judged by the upper-level loss function) than $\hat{\boldsymbol{x}}(\boldsymbol{\gamma}^{(u)})$, even if the updated hyperparameters are better when evaluated with the previous (larger) step-size or more lower-level iterations. Fig. 11.1 and 11.2 demonstrate this phenomenon.

Another approach is to learn the step-size and/or number of iterations. For example, [157] provides a continuous-time perspective on the unrolling approach and learns the stopping time, which translates to the number of iterations in the discrete approach.

The continuous time perspective on unrolling models the lower-level problem as a differential equation with an initial condition enforcing that $\boldsymbol{x}$ at time 0 is $\boldsymbol{x}_0$ [157], [280]. Just as the unrolled approach better approximates the bilevel problem as the number of iterations approaches infinity, the continuous perspective on unrolling approaches the bilevel problem as the stopping time $T \to \infty$. The discretization of the continuous-time gradient flow corresponds to an unrolled optimization algorithm (or, more generally, to a variational network with shared weights) and back-propagation can be seen as a discretization of the continuous-time adjoint equation [157], [280]. Solving the differentiable adjoint equation does not require saving the forward-pass output at every "step," making the backward pass feasible for large problems such as 3D CT image reconstruction [281].

Like many other bilevel methods for filter learning, [157] uses a regularizer based on the Field of Experts [187] and the standard data-fit term. The lower-level problem in [157] is

$$\begin{aligned} &\text{State equation: } \frac{d\boldsymbol{x}(t)}{dt} = \text{-}\boldsymbol{A}'(\boldsymbol{A}\boldsymbol{x}(t) - \boldsymbol{y}) - \sum_k \boldsymbol{C}_k' \phi_k(\boldsymbol{C}_k \boldsymbol{x}(t)) \\ &\text{Initial condition: } \boldsymbol{x}(0) = \boldsymbol{x}_0, \end{aligned}$$

where [157] learns a separate penalty function for each filter. Ref. [157] found that beyond a certain depth, increasing the number of layers did not significantly decrease the upper-level loss. Further, following intuition, the learned stopping time increased with higher noise levels or blur strengths in the denoising and deblurring problem settings [157].

10.1.3.2 Application to Non-smooth Cost Functions

An important distinction between the minimizer approach and the unrolled approach is that the unrolled approach depends on the optimization algorithm. Therefore, in addition to the number of iterations and step size, one must select an optimization algorithm to unroll. The choice is typically driven by parameters such as memory availability and desired run-time, with the one requirement being that Ψ be differentiable in both $\boldsymbol{x}$ and $\boldsymbol{\gamma}$. For certain cost functions, a resulting advantage of the unrolling method is that one can use a smooth Ψ to optimize a non-smooth cost function, removing the need for smoothing techniques such as used in (CR1N).

Ochs *et al.* [146] describe one such smooth update algorithm for a non-smooth cost function. At a high-level, their approach is to:

1. transform the lower-level cost function to a primal-dual, saddle-point problem, using the Legendre-Fenchel conjugate of ϕ (defined in Appendix C),
2. use a forward-backward splitting algorithm to alternatively update the primal ($\boldsymbol{x}$) and dual ($\boldsymbol{d}$) variables, and
3. replace the Euclidean norm in the proximal operator in the dual variable update equation with a Bregman divergence measure.

If the Bregman divergence measure is chosen carefully, the resulting update is smooth and standard backpropagation tools can compute $\nabla\ell(\boldsymbol{\gamma})$. This section overviews how the approach in [146] applies to (Ex). Ref. [146] derives the full backpropagation formula and uses Bregman divergences to unroll non-smooth cost functions in a multi-label segmentation problem, but the approach generalizes to image reconstruction as shown here.

Using the stacked convolutional matrix notation for the learned filters defined in (8.7) and selecting ϕ to be the absolute value function[3], the lower-level optimization problem is

$$\underset{\boldsymbol{x}}{\operatorname{argmin}} \frac{1}{2}\|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{y}\|^2 + \|\boldsymbol{\Omega}\boldsymbol{x}\|_1 .$$

From (C.8), the corresponding saddle-point formulation is

$$\underset{\boldsymbol{x}}{\operatorname{argmin}} \min_{\boldsymbol{d}} \frac{1}{2}\|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{y}\|^2 - \langle \boldsymbol{d}, \boldsymbol{\Omega}\boldsymbol{x}\rangle \text{ s.t. } |d_i| \leq 1 \; \forall i,$$

where $\boldsymbol{d}$ is the dual variable. The minimum cost value and corresponding minimizer, $\hat{\boldsymbol{x}}$, of the saddle-point problem are equivalent to those of the original problem because the 1-norm is convex.

To optimize the saddle-point problem, one can alternate $\boldsymbol{x}$ and $\boldsymbol{z}$ updates. Ref. [146] uses the primal-dual algorithm from [282] that introduces a proximity function to each update step:

$$\begin{aligned}
\boldsymbol{x}^{(t+1)} &= \underset{\boldsymbol{x}}{\operatorname{argmin}} \frac{1}{2}\|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{y}\|^2 - \langle \boldsymbol{d}^{(t)}, \boldsymbol{\Omega}\boldsymbol{x}\rangle + \frac{1}{\alpha_{\mathrm{x}}}\mathbf{1}' D.(\boldsymbol{x}, \boldsymbol{x}^{(t)}) \\
\boldsymbol{d}^{(t+1)} &= \underset{\boldsymbol{d}}{\operatorname{argmin}} \frac{1}{\alpha_{\mathrm{d}}}\mathbf{1}' D.(\boldsymbol{d}, \tilde{\boldsymbol{d}}) - \langle \boldsymbol{d}, \boldsymbol{\Omega}\tilde{\boldsymbol{x}}\rangle \text{ s.t. } |\boldsymbol{d}_i| \leq 1 \; \forall i,
\end{aligned} \tag{10.20}$$

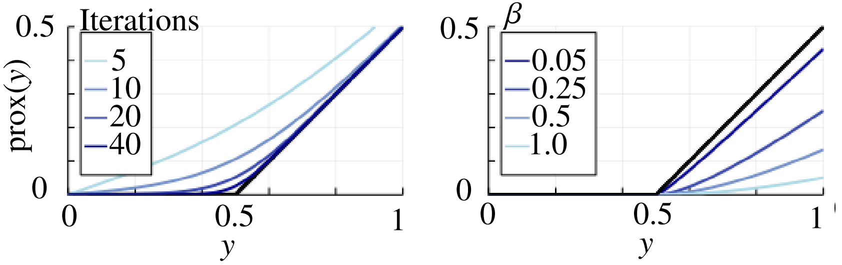
where $\tilde{\boldsymbol{x}}$ and $\tilde{\boldsymbol{d}}$ are defined in terms of previous iterates, *e.g.*, when including momentum, and α_{x} and α_{d} are step size parameters chosen according to the theory in [282]. The $\boldsymbol{x}$ update is a smooth, quadratic problem and is straightforward. However, the standard dual update involves a non-smooth projection; in particular, if the proximal distance function is the standard Euclidean 2-norm, i.e., $D(d, \tilde{d}) = \frac{1}{2}(d - \tilde{d})^2$, then the $\boldsymbol{d}$ update is the projection

$$\boldsymbol{d}^{(t+1)} = \operatorname{sign}.(\tilde{\boldsymbol{d}} + \alpha_{\mathrm{d}}\boldsymbol{\Omega}\tilde{\boldsymbol{x}}) \odot \min.(1, |\tilde{\boldsymbol{d}} + \alpha_{\mathrm{d}}\boldsymbol{\Omega}\tilde{\boldsymbol{x}}|),$$

which is non-smooth.

To make the $\boldsymbol{d}$ update smooth, [146] replaces the standard Euclidean norm in the proximity operator with a

[3] When using the absolute value, one can absorb the tuning parameters β_k into the filter magnitudes, conveniently reducing the dimension of $\boldsymbol{\gamma}$.



(a) Estimate using the Bregman divergence.

(b) Proximal operator with $p = 3/2$ term.

Figure 10.1: Proximal operators for $R(x) = \frac{1}{2}|x|$ and some smooth relatives. The black line in both plots is the soft thresholding function, which is the proximal operator for the absolute value function, *i.e.*, $\text{prox}(y) = \text{argmin}_x \frac{1}{2}(x-y)^2 + \frac{1}{2}|x|$. (a) As described in [146], the number of iterations of the primal-dual algorithm with the Bregman proximity function acts as a smoothing parameter for the proximal operator estimate and the estimate improves as the number of iterations increases (from light to dark lines). (b) Smooth proximal operator for the non-smooth penalty function (10.23) for $p = 3/2, \beta = 0.5$, and four different values of $\tilde{\beta}$. The proximal operator is closer to soft thresholding for smaller values of $\tilde{\beta}$ (darker lines).

Bregman divergence. For the 1-norm regularizer, [146] considers the divergence measure

$$D(d, \tilde{d}) = \psi(d) - \psi(\tilde{d}) - \nabla\psi(\tilde{d})'(d - \tilde{d}) \tag{10.21}$$

where $\psi(d) = \frac{1}{2}((d+1)\log(d+1) + (1-d)\log(1-d))$. Similar to standard distance metrics, this Bregman divergence is zero when $d = \tilde{d}$. However, it is not symmetric, *i.e.*, $D(d, \tilde{d}) \neq D(\tilde{d}, d)$ in general. Using this definition for D, one can differentiate and solve for the minimizer in the $\boldsymbol{d}$ update (10.20) [146]. Because all the functions are separable, the update can be done independently for each $\boldsymbol{d}$ coordinate:

$$d_i^{(t+1)} = \frac{e^{2\alpha_{\text{d}}[\boldsymbol{\Omega x}]_i} - \frac{1-\tilde{d}_i}{1+\tilde{d}_i}}{e^{2\alpha_{\text{d}}[\boldsymbol{\Omega x}]_i} + \frac{1-\tilde{d}_i}{1+\tilde{d}_i}}. \tag{10.22}$$

When the step-size α_{d} approaches infinity, $d_i^{(t+1)}$ approaches ±1 (its extreme values). When α_{d} approaches 0, $d_i^{(t+1)} = \tilde{d}_i$. The updated coordinate is guaranteed to satisfy the constraint $|d_i| \leq 1$ whenever $\tilde{d}_i$ does, so there is no need for a (non-smooth) projection. Although this approach allows for applying the unrolled method to non-smooth cost functions, [146] comments that "the [equivalent of a] 'smoothing parameter' in our approach is the number of iterations of the algorithm that replaces the lower level problem." Fig. 10.1 demonstrates how the number of iterations impacts the effective smoothing for a simple version of the problem where $\boldsymbol{A} = \boldsymbol{I}$ and $\boldsymbol{\Omega} = \boldsymbol{I}$.

Ref. [241] uses the same saddle-point problem as in [146] to propose another approach to computing $\nabla\ell(\boldsymbol{\gamma})$. Instead of unrolling an algorithm and then back-propagating, [241] uses a sensitivity analysis and introduces additional adjoint variables that allow for simultaneously computing $\nabla\ell(\boldsymbol{\gamma})$ in the same forward iteration as $\hat{\boldsymbol{x}}(\boldsymbol{\gamma})$, without incurring the large matrix-matrix multiplications costs as in the forward-mode method of computing (10.18). Although the theoretical analysis of the resulting "piggy-backing" optimization algorithm is for smooth functions, [241] found it worked well empirically in non-smooth settings.

Christof [283] shows another approach to achieving a smooth optimization algorithm for a non-smooth cost function. Ref. [283] specifically considers cost functions with penalty functions of the form

$$\phi(z) = \beta|z| + 2\tilde{\beta}\frac{|z|^p}{p} \text{ for } 1 < p < 2. \tag{10.23}$$

As a simple demonstration, in the case where there are no convolutional filters and $p = 3/2$, the lower-level cost function is the proximal operator

$$\text{prox}_{\phi}(y) = \underset{x}{\text{argmin}}\, \frac{1}{2}(x-y)^2 + \phi(x).$$

Differentiating and solving for the minimizer yields

$$\text{prox}_{\phi}(y) = \begin{cases} \text{sign}(y)\left(\sqrt{\tilde{\beta}^2 + |y| - \beta} - \tilde{\beta}\right)^2 & \text{if } |y| > \beta \\ 0 & \text{else,} \end{cases}$$

which is continuous and differentiable everywhere with respect to y despite the non-differential absolute value function in ϕ! Fig. 10.1 shows this proximal operator alongside the proximal operator when $\phi(z) = |z|$ (soft thresholding). Ref. [283] proves that this simple example generalizes to the bilevel problem of learning filters.

10.1.4 Summary

This section focused on computing $\nabla\ell(\boldsymbol{\gamma})$, the gradient of the upper-level loss function with respect to the learnable parameters. Section 10.2 builds on this foundation to consider optimization methods for bilevel problems. Many of those optimization methods can be used in conjunction with the minimizer, translation to a single-level, or unrolled approaches to compute $\nabla\ell(\boldsymbol{\gamma})$. Thus, how one selects an approach may depend on the structure of the specific bilevel problem, how closely tied one wishes to be to the original bilevel problem, computational cost, and/or gradient accuracy.

The translation to a single-level approach is tailored to a specific type of bilevel problem. A benefit of the translation approach is the ability to use the 1-norm (without any corner rounding) in the lower-level cost function. However, the corresponding drawback is the (current) lack of generality in the minimizer approach; the closed-form expression derived in [270]–[272] is specific to using the 1-norm as ϕ. Expanding this approach to regularizers other than the 1-norm is a possible avenue for future work.

One difference among the methods is whether they depend on the lower-level optimization algorithm; while the unrolled approach depends on the specific optimization algorithm, the minimizer approach and the translation to a single-level approach do not. A resulting downside of unrolling is that one cannot use techniques such as warm starts and non-differentiable restarts, so $\boldsymbol{x}^{(T)}$ may be farther from the minimizer than the approximation from a similar number of iterations of a more sophisticated, non-differentiable update method. However, the unrolled method's dependence on Ψ is also a benefit, as an unrolled method can be applied to non-smooth cost functions, as long as the resulting update mapping Ψ is smooth. Further, defining Ψ and the initial starting point ensures that $\boldsymbol{x}^{(T)}$ is unique, avoiding concerns about non-unique minimizers.

Another advantage of unrolling is that one can run a given number of iterations of the optimization algorithm, without having to reach convergence, and still calculate a valid gradient. Particularly in image reconstruction problems, where finding $\hat{\boldsymbol{x}}$ exactly can be time intensive, the benefit of a more flexible run-time could outweigh the disadvantages. However, the corresponding downside of unrolling is that the learned hyperparameters are less clearly tied to the original cost function than when one uses the minimizer approach. Section 12.2.1 further discusses this point in connection to how unrolling for bilevel methods can differ from (deep) learnable optimization algorithms.

One way to connect the minimizer and unrolling strategies is to consider the limit as the number of unrolled iterations approaches infinity. Assuming the optimization algorithm converges, this "fixed point" approach is strongly related to the minimizer approach. For instance, [284] shows that backpropagating through the last $\tilde{T}$ iterations of a converged unrolled algorithm can be viewed as approximating the matrix inverse in the minimizer gradient equation (10.8) with an order-$\tilde{T}$ Taylor series. Section 12.2.1 further discusses how fixed point networks (or "equilibrium networks") relate the unrolled-to-convergence and minimizer approaches.

Gradient accuracy and computational cost are, unsurprisingly, trade-offs. Tab. 10.2 summarizes the cost of the minimizer and unrolled approaches, derived in Section 10.1.1.4 and Appendix D respectively, but the total computation will depend on the required gradient accuracy. By accuracy, we mean error from the true bilevel gradient

$$\|\underbrace{\hat{\nabla}_T \ell(\boldsymbol{\gamma})}_{\substack{\text{Estimated}\\ \text{gradient}}} - \underbrace{\nabla \ell(\boldsymbol{\gamma})}_{\substack{\text{True bilevel}\\ \text{gradient}}}\|,$$

where T denotes the number of lower-level optimization steps. The unrolled gradient is always accurate for the unrolled mapping, but not for the original bilevel problem. Therefore, unrolling may be more computationally feasible when one cannot run a sufficient number of lower-level optimization steps to reach close enough to a minimizer to assume the gradient in (10.3) is approximately zero [161].

In all of the approaches considered, the accuracy of the estimated hyperparameter gradient in turn depends on the solution accuracy or number of unrolled iterations of the lower-level cost function. Ref. [271] notes that their translation to a single-level approach failed if they did not optimize the lower-level problem to a sufficient accuracy level. However, [270]–[272] did not investigate how the solution accuracy of the lower-level problem impacts the upper-level gradient estimate.

For the minimizer and unrolled approaches, [264], [265] found that the gradient estimate from the minimizer approach converges to the true gradient faster than the unrolled approach (in terms of computation). To state the bounds, [264], [265] assert conditions on the structure of the bilevel problem. They assume that $\hat{\boldsymbol{x}}(\boldsymbol{\gamma})$ is the unique minimizer of the lower-level cost function, the Hessian of the lower-level is invertible, the Hessian and Jacobian of Φ are Lipschitz continuous with respect to $\boldsymbol{x}$, the gradients of the upper-level loss are Lipschitz continuous with respect

	Minimizer	Unrolled: reverse	Unrolled: forward
Memory	0	$\mathcal{O}(TN)$	$\mathcal{O}(NR)$
Hessian-vector products	0	$\mathcal{O}(T)$	$\mathcal{O}(TR)$
Hessian-inverse vector products	1	0	0
Other multiplications	NR	$\mathcal{O}(TNR)$	$\mathcal{O}(NR)$

Table 10.2: Memory and computational complexity of the minimizer approach (10.8), reverse-mode unrolled approach (D.2), and forward-mode unrolled approach (D.3) to computing $\nabla_{\boldsymbol{\gamma}} \ell(\boldsymbol{\gamma}\,; \hat{\boldsymbol{x}}(\boldsymbol{\gamma}))$. Computational costs do not include running the optimization algorithm (typically expensive but often comparable across methods), computing $\nabla_{\boldsymbol{x}} \ell(\boldsymbol{\gamma}\,; \boldsymbol{x}^{(T)})$ (typically cheap), or computing $\nabla_{\boldsymbol{\gamma}} \ell(\boldsymbol{\gamma}\,; \boldsymbol{x})$ (frequently zero). Memory requirements do not include storing a single copy of $\boldsymbol{x}$, $\boldsymbol{A}$, $\boldsymbol{\gamma}$, $\boldsymbol{H}$, and $\boldsymbol{J}$. Recall $\boldsymbol{x} \in \mathbb{F}^N$, $\boldsymbol{\gamma} \in \mathbb{F}^R$, and there are T iterations of the lower-level optimization algorithm for the unrolled method. Hessian-vector products (first row) and Hessian-inverse-vector products (middle row) are listed separately from all other multiplications (last row) as the computational cost of Hessian operations can vary widely; see discussion in Section 10.1.1.4.

to $\boldsymbol{x}$, the norm of $\boldsymbol{x}$ is bounded, and the lower-level cost is strongly convex and Lipschitz smooth for every $\boldsymbol{\gamma}$ value. Section 10.2.3.1 discusses similar investigations that use these conditions, how easy or hard they are to satisfy, and how they apply to (Ex).

Ref. [265] initializes the lower-level iterates for both the unrolled and minimizer approach with the zero vector, *i.e.*, $\boldsymbol{x}^{(0)} = \boldsymbol{0}$. Under their assumptions, [265] prove that both the unrolled and minimizer gradients converge linearly in the number of lower-level iterations when the lower-level optimization algorithm and conjugate gradient algorithm for the minimizer approach converge linearly. Although the rate of the approaches is the same, the minimizer approach converges at a faster linear rate and [265] generally recommends the minimizer approach, though they found empirically that the unrolled approach may be more reliable when the strong convexity and Lipschitz smooth assumptions on the lower-level cost do not hold.

Ref. [264] extended the analysis from [265] to consider a warm start initialization for the lower-level optimization algorithm. They similarly find that the minimizer approach has a lower complexity than the unrolled approach. Section 10.2.3.2 and 10.2.3.3 further discuss complexity results after introducing specific bilevel optimization algorithms.

10.2 Gradient-Based Bilevel Optimization Methods

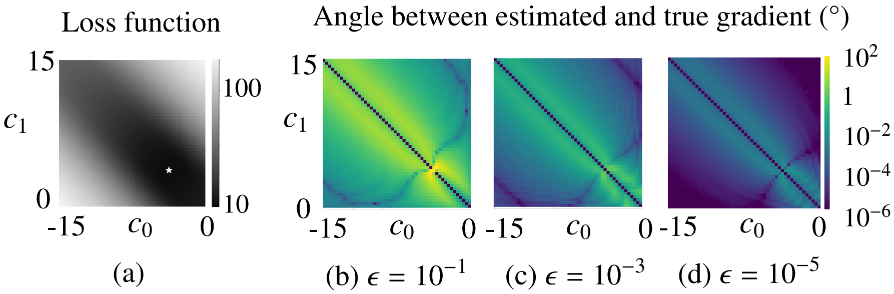
The previous section discussed different approaches to finding $\nabla \ell(\boldsymbol{\gamma})$, the gradient of the upper-level loss function with respect to the learnable parameters. Building on those results, we now consider approaches for optimizing the bilevel problem. In particular, this section concentrates on gradient-based algorithms for optimizing the hyperparameters. While there is some overlap with single-level optimization methods, this section focuses on the challenges due to the bilevel structure. Therefore, we do not discuss the lower-level optimization algorithms in detail; for overviews of single-level optimization, see, *e.g.*, [173], [285].

Gradient-based methods for bilevel problems are an alternative to the approaches described in Section 8.2, *e.g.*, grid or random search, Bayesian optimization, and trust region methods. By incorporating gradient information, the methods presented in this section can scale to problems having many hyperparameters. In fact, Section 10.2.3 reviews papers that provide bounds on the number of upper-level gradient descent iterations required to reach a point within some user-defined tolerance of a solution. While the bounds depend on the regularity of the upper-level loss and lower-level cost functions, they do not depend directly on the number of hyperparameters nor the signal dimension. Although having more hyperparameters will increase computation per iteration, using a gradient descent approach means the number of iterations need not scale with the number of hyperparameters, R.

As mentioned in the introduction of this chapter, most gradient-based bilevel methods fall into two broad categories: double-loop algorithms or single-loop algorithms. The following section discuss each category in turn. Section 10.1 used t to denote the lower-level iteration counter; this section introduces u as the iteration counter for the upper-level iterations and as the single iteration counter for single-loop algorithms.

10.2.1 Double-Loop Algorithms

After using one of the approaches in Section 10.1 to compute the hyperparameter gradient $\nabla \ell(\boldsymbol{\gamma})$, typical double-loop algorithms for bilevel problems run some type of gradient descent on the upper-level loss. Alg. 3 shows an example double-loop algorithm [286]. Line 10 of Alg. 3 uses the CG method to compute the product of the Hessian inverse with a vector in (10.8). Thus, Alg. 3 actually involves three loops. However, the third, CG loop is often left as an implementation detail and we will continue to use the term “double-loop” for the overall strategy. There is similarly



(a) (b) $\epsilon = 10^{-1}$ (c) $\epsilon = 10^{-3}$ (d) $\epsilon = 10^{-5}$

Figure 10.2: Error in the upper-level gradient, $\nabla \ell(\boldsymbol{\gamma})$, for various convergence thresholds for the lower-level optimizer. The bilevel problem is (Ex) with a single filter, $\boldsymbol{c} = \begin{bmatrix} c_0 & c_1 \end{bmatrix}$, $e^{\beta_0} = 0$, $e^{\beta_1} = -5$, and $\phi(z) = z^2$ so there is an analytic solution for $\nabla \ell(\boldsymbol{\gamma})$. The training data is piece-wise constant 1d signals and the learnable hyperparameters are the filter coefficients. (a) Upper-level loss function, $\ell(\boldsymbol{\gamma})$. The cost function is low (dark) where $c_1 \approx c_0$, corresponding to approximate finite differences. The star indicates the minimum. (b-d) Error in the estimated gradient angle using the minimizer approach (10.8), defined as the angle between $\hat{\nabla} \ell(\boldsymbol{\gamma})$ and $\nabla \ell(\boldsymbol{\gamma})$, when the lower-level optimization is run until $\|\nabla_{\boldsymbol{x}} \Phi(\boldsymbol{x}\,; \boldsymbol{\gamma})\|_2 < \epsilon$.

a third, hidden loop in approaches that use the reverse mode method for backpropogation in the unrolled approaches described in Section 10.1.3.

The final iterate of a lower-level optimizer is only an approximation of the lower-level minimizer. However, the minimizer approach to calculating the upper-level gradient $\nabla \ell(\boldsymbol{\gamma})$ from Section 10.1.1 assumes $\nabla_{\boldsymbol{x}} \Phi(\hat{\boldsymbol{x}}\,; \boldsymbol{\gamma}) = \boldsymbol{0}$. Any error stemming from not being at an exact critical point can be magnified in the full calculation (10.8), and the resulting hyperparameter gradient will be an approximation of the true gradient, as illustrated in Fig. 10.2. Thus, how accurately one optimizes the lower-level problem can greatly impact the quality of the learned parameters, $\hat{\boldsymbol{\gamma}}$ [287]. Alternatively, if one uses the unrolled approach with a set number of iterations (10.17), the gradient is accurate for that specific number of iterations, but the lower-level optimization sequence may not have converged and the overall method may not accurately approximate the original bilevel problem.

Due to such inevitable inexactness when computing $\nabla \ell(\boldsymbol{\gamma})$, one may wonder about the convergence of double-loop algorithms for bilevel problems. Considering the unrolled method of computing $\nabla \ell(\boldsymbol{\gamma})$, [288] showed that the sequence of hyperparameter values in a double-loop algorithm, $\boldsymbol{\gamma}^{(u)}$, converges as the number of unrolled iterations increases. To prove this result, [288] assumed the hyperparameters were constrained to a compact set, $\ell(\boldsymbol{\gamma}\,; \boldsymbol{x})$ and $\Phi(\boldsymbol{x}\,; \boldsymbol{\gamma})$ are jointly continuous, there is a unique solution $\hat{\boldsymbol{x}}(\boldsymbol{\gamma})$ to the lower-level cost for all $\boldsymbol{\gamma}$; and $\hat{\boldsymbol{x}}(\boldsymbol{\gamma})$ is bounded for all $\boldsymbol{\gamma}$. These conditions are satisfied for problems with strictly convex lower-level cost functions and suitable box constraints on $\boldsymbol{\gamma}$. Section 10.2.3.2 further discusses convergence results for double-loop algorithms.

Pedregosa [286] proved a similar result for the minimizer formula (10.8) using CG to compute (10.10). Specifically, [286] showed that the hyperparameter sequence convergences to a stationary point if the sequence of positive tolerances, $\{\epsilon^{(u)}, u = 1, 2, \ldots\}$ in Alg. 3, is summable. The convergence results are for the algorithm version shown in Alg. 3 that uses a Lipschitz constant of $\ell(\boldsymbol{\gamma})$, which is generally unknown. Although [286] discusses various empirical strategies for setting the step size, the convergence theory does not consider those variations. Thus, the double-loop algorithm [286] requires multiple design decisions.

There are four key design decisions for double-loop algorithms:

1. How accurately should one solve the lower-level problem?
2. What upper-level gradient descent algorithm should one use?
3. How does one pick the step size for the upper-level descent step?
4. What stopping criteria should one use for the upper-level iterations?

Algorithm 3 Hyperparameter optimization with approximate gradient (HOAG) from [286]. As written below, the HOAG algorithm is impractical because it uses $\hat{\boldsymbol{x}}(\boldsymbol{\gamma}^{(u)})$ in the convergence criteria; however, for strongly convex lower-level problems, the convergence criteria, $\|\hat{\boldsymbol{x}}(\boldsymbol{\gamma}^{(u)}) - \boldsymbol{x}^{(t)}(\boldsymbol{\gamma}^{(u)})\|$, is easily upper-bounded.

```
procedure HOAG($\{\epsilon^{(u)}, u = 1, 2, \ldots\}$, $\boldsymbol{\gamma}^{(0)}$, $\boldsymbol{x}^{(0)}$, $\boldsymbol{y}$)
    for $u$ do=0,1,...                                   ▷ Upper-level iteration counter
        $t = 0$                                          ▷ Lower-level iteration counter
        while $\|\hat{\boldsymbol{x}}(\boldsymbol{\gamma}^{(u)}) - \boldsymbol{x}^{(t)}(\boldsymbol{\gamma}^{(u)})\| \geq \epsilon^{(u)}$ do
            $\boldsymbol{x}^{(t+1)} = \Psi(\boldsymbol{x}^{(t)}\,; \boldsymbol{\gamma}^{(u)})$        ▷ Lower-level optimization step
            $t = t + 1$
        end while
        Compute gradient $\nabla_{\boldsymbol{x}} \ell(\boldsymbol{\gamma}^{(u)}\,; \boldsymbol{x}^{(t)})$ and
        Jacobian $\nabla_{\boldsymbol{x}\boldsymbol{\gamma}} \Phi(\boldsymbol{x}^{(t)}\,; \boldsymbol{\gamma}^{(u)})$
        Using CG, find $\boldsymbol{q}$ such that
        $\|\nabla_{\boldsymbol{x}\boldsymbol{x}} \Phi(\boldsymbol{x}^{(t)}\,; \boldsymbol{\gamma}^{(u)})\boldsymbol{q} - \nabla_{\boldsymbol{x}} \ell(\boldsymbol{\gamma}^{(u)}\,; \boldsymbol{x}^{(t)})\| \leq \epsilon^{(u)}$
        $\boldsymbol{g} = \nabla_{\boldsymbol{\gamma}} \ell(\boldsymbol{\gamma}^{(u)}\,; \boldsymbol{x}^{(t)}) - \left(\nabla_{\boldsymbol{x}\boldsymbol{\gamma}} \Phi(\boldsymbol{x}^{(t)}\,; \boldsymbol{\gamma}^{(u)})\right)' \boldsymbol{q}$      ▷ From (10.8)
        $\boldsymbol{\gamma}^{(u+1)} = \boldsymbol{\gamma}^{(u)} - \frac{1}{L}\boldsymbol{g}$      ▷ $L$ is a Lipschitz constant of $\nabla \ell(\boldsymbol{\gamma})$
    end for
    return $\boldsymbol{\gamma}^{(u+1)}$
end procedure
```

This section first reviews some (largely heuristic) approaches to these design decisions and presents example bilevel gradient descent methods with no (or few) assumptions beyond those made in Section 10.1. Without any further assumptions, the answers to the questions above are based on heuristics, with few theoretical guarantees but often providing good experimental results. Section 10.2.3.2 discusses recent methods with stricter assumptions on the bilevel problem and their theory-backed answers to the above questions.

The first step in a double-loop algorithm is to optimize the lower-level cost, for which there are many optimization approaches. The only restriction is computability of the gradient of the upper-level loss $\nabla \ell(\boldsymbol{\gamma})$, which typically includes a smoothness assumption (see Section 10.1 for discussion). Many bilevel methods use a standard optimizer for the lower-level problem, although others propose new variants, *e.g.*, [166].

The **first design decision** (how accurately to solve the lower-level problem) involves a trade-off between computational complexity and accuracy. Example convergence criteria are fairly standard to the optimization literature, *e.g.*, the Euclidean norm of the lower-level gradient [162], [289] or the normalized change in the estimate $\boldsymbol{x}$ [290] being less than some threshold. For example, [162] used a convergence criteria of $\|\nabla_{\boldsymbol{x}} \Phi(\boldsymbol{x}^{(t)}\,; \boldsymbol{\gamma})\|_2 \leq 10^{-3}$ (where the image scale is 0-255). As mentioned above, [286] uses a sequence of convergence tolerances so that the lower-level cost function is optimized more accurately as the upper-level iterations continue.

Ref. [287] investigated the importance of lower-level optimization accuracy. The authors use the same training model as in [163], which is the bilevel extension of the Field of Experts [187], but varied the convergence criteria for the lower-level problem. When using a convergence tolerance of $\|\nabla_{\boldsymbol{x}} \Phi(\boldsymbol{x}^{(t)}\,; \boldsymbol{\gamma})\|_2 / \sqrt{N} \leq 10^{-5}$, [287] found an average improvement of 0.65dB in the PSNR for test images over [163], who ran their lower-level optimization algorithm for a set number of iterations. Ref. [287] also plots the test PSNR and training loss versus the lower-level convergence criteria and shows how test PSNR increases and training loss decreases with increased lower-level solution accuracy for this specific filter learning bilevel problem.

Many publications do not report a specific threshold or discuss how they chose a convergence criteria or number of lower-level iterations. However, a few note the importance of such decisions. For example, [271] found that their learning method fails if the lower-level optimizer is insufficiently close to the minimizer and [162] stated their results are "significantly better" than [163] because they solve the lower-level problem "with high[er] accuracy."

After selecting a level of accuracy, finding (an approximation of) $\hat{\boldsymbol{x}}$, and calculating $\nabla\ell(\boldsymbol{\gamma})$ using one of the approaches from Section 10.1, one must make the **second design decision**: which gradient-based method to use for the upper-level problem. Many bilevel methods suggest a simple gradient-based method such as plain gradient descent (GD) [165], GD with a line search (see the third design decision), projected GD [279], or stochastic GD [271]. These methods update $\boldsymbol{\gamma}$ based on only the current upper-level gradient; they do not have memory of previous gradients nor require/estimate any second-order information.

Methods that incorporate some second-order information use more memory and computation per iteration, but may converge faster than basic GD methods. For example, Broyden-Fletcher-Goldfarb-Shanno (BFGS) and L-BFGS (the low-memory version of BFGS) [291] are quasi-Newton algorithms that store and update an approximate Hessian matrix that serves as a preconditioner for the gradient. The $R \times R$ size of the Hessian grows as the number hyperparameters increases, but quasi-Newton methods like L-BFGS use practical rank-1 updates with storage $\mathcal{O}(R)$. Adam [292] is a popular GD method, especially in the machine learning community, that tailors the step size (equivalently the learning rate) for each hyperparameter based on moments of the gradient. Although Adam requires its own parameters, the parameters are relatively easy to set and the default settings often perform adequately. Example bilevel papers using methods with second-order information include those that use BFGS [164], L-BFGS [162], Gauss-Newton [293], and Adam [166].

Many gradient-based methods require selecting a step size parameter, *e.g.*, one must choose a step size α_ℓ in classical GD:

$$\boldsymbol{\gamma}^{(u+1)} = \boldsymbol{\gamma}^{(u)} - \alpha_\ell \, \nabla\ell\left(\boldsymbol{\gamma}^{(u)}\right).$$

This choice is the **third design decision**. Bilevel problems are generally non-convex[4], and typically a Lipschitz constant is unavailable, so line search strategies initially appear appealing. However, any line search strategy that involves attempting multiple values quickly becomes computationally intractable for large-scale problems. The upper-level loss function in bilevel problems is particularly expensive to evaluate because it requires optimizing the lower-level cost! Further, recall that the upper-level loss is typically an expectation over multiple training samples (UL), so evaluating a single step size involves optimizing the lower-level cost J times (or using a stochastic approach and selecting a batch size).

Despite these challenges, a line search strategy may be viable if it rarely requires multiple attempts. For example, the backtracking line search in [289] that used the Armijo–Goldstein condition required 57-59 lower-level evaluations (per training example) over 40 upper-level gradient descent steps, so most upper-level steps required only one lower-level evaluation. Other bilevel papers that used backtracking with Armijo-type conditions include [143], [164], [290]; [294] used the Barzilai-Borwein method for picking an adaptive step size.

Other approaches to determining the step size are: (i) normalize the gradient by the dimension of the data and pick a fixed step size [271], (ii) pick a value that is small enough based on experience [165], or (iii) adapt the step size based on the decrease from the previous iteration [286].

The **fourth design decision** is the convergence criteria for the upper-level loss. As with the lower-level convergence criteria, few publications include a specific threshold, but most bilevel methods tend to use traditional convergence criteria such as the norm of the hyperparameter gradient falling below some threshold [164], the norm of the

[4] Although there are exceptions for simple functions, for common upper and lower-level functions, non-convexity is easily verified by plotting a cross-section of the cost function.

change in parameters falling below some threshold [162], and/or reaching a maximum iteration count (many papers). One specific example is to terminate when the normalized change in learned parameters, $\|\boldsymbol{\gamma}^{(u+1)} - \boldsymbol{\gamma}^{(u)}\|/\|\boldsymbol{\gamma}^{(u)}\|$, is below 0.01 [290]. The normalized change bound is convenient because it is unitless and thus invariant to scaling of $\boldsymbol{\gamma}$.

Fig. 10.3 shows example upper-level convergence plots for a double-loop algorithm for the bilevel problem (Ex). After an initial first run of OGM to get the lower-level initialization $\hat{\boldsymbol{x}}(\boldsymbol{\gamma}^{(0)})$ such that $\frac{1}{\sqrt{N}} \left\| \nabla_{\boldsymbol{x}} \Phi \left(\hat{\boldsymbol{x}}(\boldsymbol{\gamma}^{(0)}) \,;\, \boldsymbol{\gamma}^{(0)} \right) \right\|_2 < 10^{-7}$, the lower-level optimizer consisted of 10 iterations of OGM [295], initialized with the estimate from the previous upper-level iteration. The upper-level optimizer is Adam [292] with the default parameters, negating the need for a separate upper-level step-size parameter. We ran 10,000 outer-loop iterations. The final norm of the upper-level gradient, $\frac{1}{\sqrt{R}} \|\nabla(\boldsymbol{\gamma}^{(U)})\|$ was 0.08 when learning the filter coefficients and tuning parameters and $5 \cdot 10^{-4}$ when learning only $\boldsymbol{\beta}$. Fig. 12.2 shows the corresponding denoised images and Appendix F further details the experiment settings.

10.2.2 Single-Loop Algorithms

Unlike double-loop algorithms, single-loop algorithms take a gradient step in $\boldsymbol{\gamma}$ without optimizing the lower-level problem each step. Two early bilevel method papers [158], [192] proposed single-loop approaches based on solving the system of equations that arises from the Lagrangian.

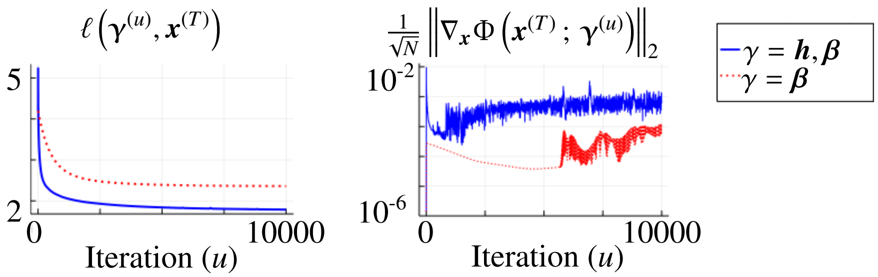
The system of equations approach in [158], [192] closely follows the KKT perspective on the minimizer approach in Section 10.1.1.2. Recall that the gradient of the lower-level problem is zero at a minimizer, $\hat{\boldsymbol{x}}$, and one can use this equality as a constraint on the upper-level loss function. The corresponding Lagrangian is

$$L(\boldsymbol{x}, \boldsymbol{\gamma}, \boldsymbol{\nu}) = \ell(\boldsymbol{\gamma} \,;\, \boldsymbol{x}) + \boldsymbol{\nu}^T \nabla_{\boldsymbol{x}} \Phi(\boldsymbol{x} \,;\, \boldsymbol{\gamma}), \tag{10.24}$$

where $\boldsymbol{\nu}$ is a vector of Lagrange multipliers. For the filter learning example (Ex), the Lagrangian is

$$L(\boldsymbol{x}, \boldsymbol{\gamma}, \boldsymbol{\nu}) = \frac{1}{2} \|\boldsymbol{x} - \boldsymbol{x}^{\text{true}}\|_2^2 + \boldsymbol{\nu}^T \left(\boldsymbol{A}'(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}) + e^{\beta_0} \sum_{k=1}^{K} e^{\beta_k} \tilde{\boldsymbol{c}}_k \circledast \phi.(\boldsymbol{c}_k \circledast \boldsymbol{x}; \epsilon) \right).$$

As in Section 10.1.1.2, we consider derivatives of the Lagrangian with respect to $\boldsymbol{\nu}$, $\boldsymbol{x}$, and $\boldsymbol{\gamma}$. Here are the general expressions and the specific equations for the filter learning example (Ex) when considering the element of $\boldsymbol{\gamma}$



(a) (b)

Figure 10.3: Example convergence plots for a double-loop bilevel method when $\boldsymbol{\gamma}$ includes $\boldsymbol{h}$ and $\boldsymbol{\beta}$ (solid lines) and when $\boldsymbol{\gamma} = \boldsymbol{\beta}$ (dotted lines). (a) Estimated upper-level loss function evaluated at the current estimate of the lower-level minimizer, $\boldsymbol{x}^{(T)} = \boldsymbol{x}^{(T)}(\boldsymbol{\gamma}^{(u)})$, versus upper-level iteration u. (b) Lower-level convergence metric, averaged over all training samples, versus upper-level iteration. The estimated lower-level minimizer remains close to convergence throughout the double-loop method.

corresponding to β_k:

$$\begin{aligned}
\nabla_{\boldsymbol{\nu}} L(\boldsymbol{x}, \boldsymbol{\gamma}, \boldsymbol{\nu}) &= \nabla_{\boldsymbol{x}} \Phi(\boldsymbol{x}\,; \boldsymbol{\gamma}) \\
&= \boldsymbol{A}'(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}) + e^{\beta_0} \sum_{k=1}^{K} e^{\beta_k} \tilde{\boldsymbol{c}}_k \circledast \phi.(\boldsymbol{c}_k \circledast \boldsymbol{x}; \epsilon) \\
\nabla_{\boldsymbol{x}} L(\boldsymbol{x}, \boldsymbol{\gamma}, \boldsymbol{\nu}) &= \nabla_{\boldsymbol{x}} \ell(\boldsymbol{\gamma}\,; \boldsymbol{x}) + \nabla_{\boldsymbol{x}\boldsymbol{x}} \Phi(\boldsymbol{x}\,; \boldsymbol{\gamma})\boldsymbol{\nu} \\
&= \boldsymbol{x} - \boldsymbol{x}^{\text{true}} + \boldsymbol{A}'\boldsymbol{A}\boldsymbol{\nu} + e^{\beta_0} \sum_{k} e^{\beta_k} \boldsymbol{C}_k' \operatorname{diag}(\ddot{\phi}.(\boldsymbol{c}_k \circledast \hat{\boldsymbol{x}}))\boldsymbol{C}_k \boldsymbol{\nu} \\
\nabla_{\boldsymbol{\gamma}} L(\boldsymbol{x}, \boldsymbol{\gamma}, \boldsymbol{\nu}) &= \nabla_{\boldsymbol{\gamma}} \ell(\boldsymbol{\gamma}\,; \, \boldsymbol{x}) + \boldsymbol{\nu}^T \nabla_{\boldsymbol{x}\boldsymbol{\gamma}} \Phi(\boldsymbol{x}\,; \boldsymbol{\gamma}) \\
&= \boldsymbol{\nu}^T \left(e^{\beta_0} e^{\beta_k} \tilde{\boldsymbol{c}}_k \circledast \dot{\phi}.(\boldsymbol{c}_k \circledast \hat{\boldsymbol{x}}) \right) \text{ when } \boldsymbol{\gamma} = \beta_k.
\end{aligned}$$

These expressions are equivalent to the primal, adjoint, and optimality conditions respectively in [192].

Here the minimizer and single-loop approach diverge. Section 10.1.1.2 used the above Lagrangian gradients to solve for $\hat{\boldsymbol{\nu}}$, substitute $\hat{\boldsymbol{\nu}}$ into the gradient of the Lagrangian with respect to $\boldsymbol{\gamma}$, and thus find the minimizer expression for $\nabla \ell(\boldsymbol{\gamma})$. The single-loop approach instead considers solving the system of gradient equations directly:

$$\boldsymbol{G}(\boldsymbol{x}, \boldsymbol{\gamma}, \boldsymbol{\nu}) = \begin{bmatrix} \nabla_{\boldsymbol{\nu}} L(\boldsymbol{x}, \boldsymbol{\gamma}, \boldsymbol{\nu}) \\ \nabla_{\boldsymbol{x}} L(\boldsymbol{x}, \boldsymbol{\gamma}, \boldsymbol{\nu}) \\ \nabla_{\boldsymbol{\gamma}} L(\boldsymbol{x}, \boldsymbol{\gamma}, \boldsymbol{\nu}) \end{bmatrix} = \boldsymbol{0}.$$

For example, [192] proposed a Newton algorithm using the Jacobian of the gradient matrix $\boldsymbol{G}$.

Another approach to single-loop algorithms is to replace the "while" loop in Alg. 3 line 4 with a single gradient step in the lower-level optimization variables. Two single-loop algorithms are the two-timescale stochastic approximation (TTSA) method [296] and the Single Timescale stochAstic BiLevEl optimization (STABLE) method [297]. Alg. 4 shows TTSA as an example single-loop algorithm. Both TTSA and STABLE alternate between one gradient step for the lower-level cost and one gradient step for the upper-level problem.

There are two main challenges in designing such a single loop algorithm for bilevel optimization. Because both TTSA and STABLE use the minimizer approach (10.8) to finding the upper-level gradient, the first challenge is ensuring the current lower-level iterate is close enough to the minimizer to calculate a useful upper-level gradient. TTSA addresses this challenge by taking larger steps for the lower-level problem while STABLE addresses this using a lower-level update that better predicts the next lower-level minimizer, $\hat{\boldsymbol{x}}(\boldsymbol{\gamma}^{(u+1)})$.

The second main challenge is estimating the upper-level gradient, even given stochastic estimates of $\nabla_{\boldsymbol{x}\boldsymbol{x}}\Phi$ and $\nabla_{\boldsymbol{x}\boldsymbol{\gamma}}\Phi$, because the minimizer equation (10.8) is nonlinear. The theoretical results about TTSA are built on the assumption that the upper-level gradient is biased due to this nonlinearity. In contrast, STABLE uses recursion to update estimates of the gradients and thus reduce variance. Section 10.2.3.3 goes into more detail about both algorithms.

10.2.3 Complexity Analysis

A series of recent papers established finite-time sample complexity bounds for stochastic bilevel optimization methods based on gradient descent for the upper-level loss and lower-level cost. Ref.s [264], [298] use double-loop approaches and [296], [297] use single-loop algorithms. Unlike most of the methods discussed in Section 10.2.1, these papers make additional assumptions about the upper and lower-level functions then select the upper and lower-level step sizes to ensure convergence.

Algorithm 4 Two-Timescale Stochastic Approximation (TTSA) method from [296]. TTSA includes a possible projection of the hyperparameter after each gradient step onto a constraint set, not shown here. The tildes denote stochastic approximations for the corresponding expressions.

```
procedure TTSA($\boldsymbol{\gamma}^{(0)}, \boldsymbol{x}^{(0)}, \alpha_\ell^{(u)}, \alpha_\Phi^{(u)}$)
    for $u = 1, \ldots$ do
        $\boldsymbol{x}^{(u+1)} = \boldsymbol{x}^{(u)} - \alpha_\Phi^{(u)} \tilde{\nabla}_{\boldsymbol{x}} \Phi(\boldsymbol{x}^{(u)}; \boldsymbol{\gamma}^{(u)})$
        $\boldsymbol{g} = \nabla_{\boldsymbol{\gamma}} \ell^{(u)} - \left(\tilde{\nabla}_{\boldsymbol{x}\boldsymbol{\gamma}} \Phi^{(u)}\right)' \left(\tilde{\nabla}_{\boldsymbol{x}\boldsymbol{x}} \Phi^{(u)}\right)^{\text{-1}} \nabla_{\boldsymbol{x}} \ell^{(u)}$
        $\boldsymbol{\gamma}^{(u+1)} = \boldsymbol{\gamma}^{(u)} - \alpha_\ell^{(u)} \boldsymbol{g}$
    end for
end procedure
```

Table 10.3: Finite-time sample complexities for the stochastic bilevel problem in the common scenario where ℓ is non-convex when using BA [298], stocBiO [264], TTSA [296], and STABLE [297]. When ℓ is strongly convex, the sample complexity of STABLE is $\mathcal{O}\left(\frac{1}{\epsilon^1}\right)$ (for the upper- and lower-level gradients), which is the same as single level stochastic gradient algorithms. See cited papers for other complexity results when ℓ is strongly convex.

In these works, "finite-time sample complexity" refers to big-O bounds on a number of iterations that ensures one reaches a minimizer to within some desired tolerance. In contrast to asymptotic convergence analysis, finite-time bounds provide information about the estimated hyperparameters, $\boldsymbol{\gamma}^{(u)}$, after a finite number of upper-level iterations. These bounds depend on problem-specific quantities, such as Lipschitz constants, but not on the hyperparameter or signal dimensions.

To summarize the results, this section returns to the notation from the introduction where the upper-level loss may be deterministic or stochastic, *e.g.*, the bilevel problem is

$$\hat{\boldsymbol{\gamma}} = \underset{\boldsymbol{\gamma}}{\operatorname{argmin}}\, \ell(\boldsymbol{\gamma}) \text{ with } \ell(\boldsymbol{\gamma}) = \begin{cases} \ell(\boldsymbol{\gamma}, \hat{\boldsymbol{x}}(\boldsymbol{\gamma})) & \text{deterministic} \\ \mathbb{E}\left[\ell(\boldsymbol{\gamma}, \hat{\boldsymbol{x}}(\boldsymbol{\gamma}))\right] & \text{stochastic.} \end{cases} \tag{10.25}$$

The expectation in (10.25) can have different meanings depending on the setting. When one has J training images with one noise realization per image, one often picks a random subset ("minibatch") of those J images for each update of $\boldsymbol{\gamma}$, corresponding to stochastic gradient descent of the upper-level loss. In this setting, the randomness is a property of the algorithm, not of the upper-level loss, and the expectation reduces to the deterministic case. Section 13.2 discusses other possible definitions of the stochastic bilevel formulation.

The complexity results (summarized in Tab. 10.3) are all in terms of finding $\boldsymbol{\gamma}_\epsilon$, defined as an ϵ-optimal solution. In the (atypical) setting where $\ell(\boldsymbol{\gamma})$ is convex, $\boldsymbol{\gamma}_\epsilon$ is an ϵ-optimal solution if it satisfies either $\ell(\boldsymbol{\gamma}_\epsilon) - \ell(\hat{\boldsymbol{\gamma}}) \leq \epsilon$ [264], [296], [298] or $\left\|\hat{\boldsymbol{\gamma}} - \boldsymbol{\gamma}_\epsilon\right\|^2 \leq \epsilon$ [297]. (These conditions are equivalent if ℓ is strongly convex in $\boldsymbol{\gamma}$, but can differ otherwise.) In the (common) non-convex setting, $\boldsymbol{\gamma}_\epsilon$ is typically called an ϵ-stationary point if it satisfies $\left\|\nabla \ell(\boldsymbol{\gamma}_\epsilon)\right\|^2 \leq \epsilon$ [264], [297], [298]. In the stochastic setting, $\boldsymbol{\gamma}_\epsilon$ must satisfy these conditions in expectation.

The following sections briefly describe the BA, stocBiO, TTSA, and STABLE algorithms. The literature in this area is quickly evolving; between the writing and editing of this work, new double-loop and single-loop methods appeared with improved complexity results. For example, [299], [300] concurrently proposed bilevel optimization methods that leverage momentum and variance reduction techniques to reduce the bound on the number of iterations to $\widetilde{\mathcal{O}}\left(\frac{1}{\epsilon^{1.5}}\right)$ for both upper-level and lower-level gradients. Ref. [299] achieved this complexity result for both a double-loop method and a single-loop method.

Whether double-loop or single-loop methods are preferred is an open question. Refs. [264], [299] find that double-loop methods converge faster (in terms of wall time) than single-loop methods. The authors hypothesize that $\nabla \ell(\boldsymbol{\gamma})$ is sensitive enough to changes in the estimate of the lower-level optimizer that the increased accuracy of the double-loop estimates of $\nabla \ell(\boldsymbol{\gamma})$ is worth the additional lower-level optimization time. Future work should test this hypothesis in different experimental settings and establish guidelines on when to use a double-loop or single-loop algorithm.

10.2.3.1 Assumptions

References [264], [296]–[298] all make similar assumptions about ℓ and Φ to derive theoretical results for their proposed bilevel optimization methods. We first summarize the set of sufficient conditions from [298], and later note any additional assumptions used by the other methods. The conditions in [298] on the upper-level function, $\ell(\boldsymbol{\gamma}\,;\,\boldsymbol{x})$, are:

Aℓ1. $\forall \boldsymbol{\gamma} \in \mathbb{F}^R$, $\nabla_{\boldsymbol{\gamma}} \ell(\boldsymbol{\gamma}, \boldsymbol{x})$ and $\nabla_{\boldsymbol{x}} \ell(\boldsymbol{\gamma}, \boldsymbol{x})$ are Lipschitz continuous with respect to $\boldsymbol{x}$, with corresponding Lipschitz constants $L_{\boldsymbol{x},\nabla_{\boldsymbol{\gamma}}\ell}$ and $L_{\boldsymbol{x},\nabla_{\boldsymbol{x}}\ell}$. (These constants are independent of $\boldsymbol{x}$ and $\boldsymbol{\gamma}$.)

Aℓ2. The gradient with respect to $\boldsymbol{x}$ is bounded, *i.e.*,
$\|\nabla_{\boldsymbol{x}} \ell(\boldsymbol{\gamma}, \boldsymbol{x})\| \leq C_{\nabla_{\boldsymbol{x}}\ell}, \ \forall \boldsymbol{x} \in \mathbb{F}^N$.

Aℓ3. $\forall \boldsymbol{x} \in \mathbb{F}^N$, $\nabla_{\boldsymbol{x}} \ell(\boldsymbol{\gamma}, \boldsymbol{x})$ is Lipschitz continuous with respect to $\boldsymbol{\gamma}$, with corresponding Lipschitz constant $L_{\boldsymbol{\gamma},\nabla_{\boldsymbol{x}}\ell}$.

The conditions in [298] on the lower-level function, $\Phi(\boldsymbol{x}\,;\boldsymbol{\gamma})$, are:

AΦ1. Φ is continuously twice differentiable in $\boldsymbol{\gamma}$ and $\boldsymbol{x}$.

AΦ2. $\forall \boldsymbol{\gamma} \in \mathbb{F}^R$, $\nabla_{\boldsymbol{x}} \Phi(\boldsymbol{x}\,;\boldsymbol{\gamma})$ is Lipschitz continuous with respect to $\boldsymbol{x}$ with corresponding constant $L_{\boldsymbol{x},\nabla_{\boldsymbol{x}}\Phi}$.

AΦ3. $\forall \boldsymbol{\gamma} \in \mathbb{F}^R$, $\Phi(\boldsymbol{x}\,;\boldsymbol{\gamma})$ is strongly convex with respect to $\boldsymbol{x}$, *i.e.*, $\mu_{\boldsymbol{x},\Phi} \boldsymbol{I} \preceq \nabla^2_{\boldsymbol{x}} \Phi(\boldsymbol{\gamma}\,;\boldsymbol{x})$, for some $\mu_{\boldsymbol{x},\Phi} > 0$.

AΦ4. $\forall \boldsymbol{\gamma} \in \mathbb{F}^R$, $\nabla_{\boldsymbol{x}\boldsymbol{x}} \Phi(\boldsymbol{x}\,;\boldsymbol{\gamma})$ and $\nabla_{\boldsymbol{\gamma}\boldsymbol{x}} \Phi(\boldsymbol{x}\,;\boldsymbol{\gamma})$ are Lipschitz continuous with respect to $\boldsymbol{x}$ with Lipschitz constants $L_{\boldsymbol{x},\nabla_{\boldsymbol{x}\boldsymbol{x}}\Phi}$ and $L_{\boldsymbol{x},\nabla_{\boldsymbol{\gamma}\boldsymbol{x}}\Phi}$.

AΦ5. The mixed second gradient of Φ is bounded, *i.e.*,
$\left\|\nabla_{\boldsymbol{\gamma}\boldsymbol{x}} \Phi(\boldsymbol{x}\,;\boldsymbol{\gamma})\right\| \leq C_{\nabla_{\boldsymbol{\gamma}\boldsymbol{x}}\Phi}, \quad \forall \boldsymbol{\gamma}, \boldsymbol{x}$.

AΦ6. $\forall \boldsymbol{x} \in \mathbb{F}^N$, $\nabla_{\boldsymbol{\gamma}\boldsymbol{x}} \Phi(\boldsymbol{x}\,;\boldsymbol{\gamma})$ and $\nabla_{\boldsymbol{x}\boldsymbol{x}} \Phi(\boldsymbol{x}\,;\boldsymbol{\gamma})$ are Lipschitz continuous with respect to $\boldsymbol{\gamma}$ with Lipschitz constants $L_{\boldsymbol{\gamma},\nabla_{\boldsymbol{\gamma}\boldsymbol{x}}\Phi}$ and $L_{\boldsymbol{\gamma},\nabla_{\boldsymbol{x}\boldsymbol{x}}\Phi}$.

In addition to the assumptions above on ℓ and Φ, analyses of optimization algorithms for the stochastic bilevel problem assume that (i) all estimated gradients are unbiased and (ii) the variance of the estimation errors is bounded by $\sigma^2_{\nabla_{\boldsymbol{\gamma}}\ell}$, $\sigma^2_{\nabla_{\boldsymbol{x}}\ell}$, $\sigma^2_{\nabla_{\boldsymbol{x}}\Phi}$, $\sigma^2_{\nabla_{\boldsymbol{\gamma}\boldsymbol{x}}\Phi}$, and $\sigma^2_{\nabla_{\boldsymbol{x}\boldsymbol{x}}\Phi}$. The stochastic methods discussed here are all based on the minimizer approach to finding the upper-level gradient. Therefore, the methods use estimates of $\nabla_{\boldsymbol{\gamma}} \ell(\boldsymbol{\gamma}\,;\,\boldsymbol{x})$, $\nabla_{\boldsymbol{x}} \ell(\boldsymbol{\gamma}\,;\,\boldsymbol{x})$, $\nabla_{\boldsymbol{x}} \Phi(\boldsymbol{x}\,;\boldsymbol{\gamma})$, $\nabla_{\boldsymbol{\gamma},\boldsymbol{x}} \Phi(\boldsymbol{x}\,;\boldsymbol{\gamma})$, and $\nabla_{\boldsymbol{x},\boldsymbol{x}} \Phi(\boldsymbol{x}\,;\boldsymbol{\gamma})$. We denote the estimates of these gradient using tildes, *e.g.*, $\tilde{\nabla}_{\boldsymbol{\gamma}} \ell(\boldsymbol{\gamma}\,;\,\boldsymbol{x})$. Following (10.8), an estimate of the upper-level gradient approximation is thus

$$\hat{\nabla} \ell(\boldsymbol{\gamma}) = \tilde{\nabla}_{\boldsymbol{\gamma}} \ell(\boldsymbol{\gamma}, \boldsymbol{x}) - (\tilde{\nabla}_{\boldsymbol{x}\boldsymbol{\gamma}} \Phi(\boldsymbol{x}\,;\boldsymbol{\gamma}))' (\tilde{\nabla}_{\boldsymbol{x}\boldsymbol{x}} \Phi(\boldsymbol{x}\,;\boldsymbol{\gamma}))^{-1} \tilde{\nabla}_{\boldsymbol{x}} \ell(\boldsymbol{\gamma}, \boldsymbol{x}).$$

As an example of the bounded variance assumption, [298] assumes

$$\mathbb{E}\left[\|\nabla_{\boldsymbol{\gamma}} \ell(\boldsymbol{\gamma}\,;\,\boldsymbol{x}) - \tilde{\nabla}_{\boldsymbol{\gamma}} \ell(\boldsymbol{\gamma}\,;\boldsymbol{x})\|^2\right] \leq \sigma^2_{\nabla_{\boldsymbol{\gamma}}\ell} \quad \forall \boldsymbol{x}, \boldsymbol{\gamma}.$$

Algorithm 5 Bilevel Approximation (BA) Method from [298]. The differences for the AID-BiO and ITD-BiO methods from [264] are: (1) when $u > 0$, the BiO methods replace line 3 with $\boldsymbol{x}^{(0)} = \boldsymbol{x}^{(T_{u-1})}$, (2) T_i does not vary with upper-level iteration, (3) the upper-level gradient calculation in line 7 can use the minimizer approach (10.8) or backpropagation (D.2), and (4) the hyperparameter update is standard gradient descent, so line 8 becomes $\boldsymbol{\gamma}^{(u+1)} = \boldsymbol{\gamma}^{(u)} - \alpha_\ell \boldsymbol{g}$.

```
1:  procedure BA($\boldsymbol{\gamma}^{(0)}$, $\boldsymbol{x}^{(0)}$, $\alpha_\ell$, $\alpha_\Phi$, $T_u \; \forall u$)
2:      for $u = 1, \ldots$ do                                          ▷ Upper-level iterations
3:          $\boldsymbol{x}^{(0)} = \boldsymbol{x}^{(0)}$                                 ▷ Included for comparison with [264]
4:          for $t = 1 : T_u$ do                                        ▷ $T$ lower-level iterations
5:              $\boldsymbol{x}^{(t)} = \boldsymbol{x}^{(t-1)} - \alpha_\Phi \nabla_{\boldsymbol{x}} \Phi(\boldsymbol{\gamma}, \boldsymbol{x}^{(t-1)})$
6:          end for
7:          $\boldsymbol{g} = \nabla_{\boldsymbol{\gamma}} \ell(\boldsymbol{\gamma}^{(u)}, \boldsymbol{x}^{T_i})$                        ▷ Use minimizer result (10.8)
8:          $\boldsymbol{\gamma}^{(u+1)} = \underset{\boldsymbol{\gamma}}{\operatorname{argmin}} \left\{ \frac{1}{2} \|\boldsymbol{\gamma} - \boldsymbol{\gamma}^{(u)}\|^2 + \alpha_\ell \langle \boldsymbol{g}, \boldsymbol{\gamma} \rangle \right\}$
9:      end for
10: end procedure
```

To consider how the complexity analysis bounds may apply in practice, Appendix E examines how assumptions Aℓ1-Aℓ3 and assumptions AΦ1-AΦ6 apply to the running filter learning example (Ex). Although a few of the conditions are easily satisfied, most are not. Appendix E shows that the conditions are met if one invokes box constraints on the variables $\boldsymbol{x}$ and $\boldsymbol{\gamma}$. Although imposing box constraints requires modifying the algorithms, *e.g.*, by including a projection step, the iterates remain unchanged if the constraints are sufficiently generous. However, such generous box constraints are likely to yield large Lipschitz constants and bounds, leading to overly-conservative predicted convergence rates. Further, any differentiable upper-level loss and lower-level cost function would meet the conditions above with such box constraints. Generalizing the following complexity analysis for looser conditions is an important avenue for future work.

10.2.3.2 Double-loop

Ghadimi and Wang [298] were the first to provide a finite-time analysis of the bilevel problem. The authors proposed and analyzed the Bilevel Approximation (BA) method (see Alg. 5). BA uses two nested loops. The inner loop minimizes the lower-level cost to some accuracy, determined by the number of lower-level iterations; the more inner iterations, the more accurate the gradient will be, but at the cost of more computation and time. The outer loop is (inexact) projected gradient steps on ℓ. Ref. [298] used the minimizer result (10.8) (with the IFT perspective for the derivation) to estimate the upper-level gradient.

To bound the complexity of BA, [298] first related the error in the lower-level solution to the error in the upper-level gradient estimate as

$$\|\underbrace{\hat{\nabla}_{\boldsymbol{\gamma}} \ell(\boldsymbol{\gamma}, \boldsymbol{x}^{(T)})}_{\text{Estimated gradient}} - \underbrace{\nabla_{\boldsymbol{\gamma}} \ell(\boldsymbol{\gamma}, \hat{\boldsymbol{x}}(\boldsymbol{\gamma}))}_{\text{True gradient}}\| \leq C_{\text{GW}} \underbrace{\left\| \boldsymbol{x}^{(T)} - \hat{\boldsymbol{x}}(\boldsymbol{\gamma}) \right\|}_{\text{Error in lower-level}},$$

where C_{GW} is a constant that depends on many of the bounds defined in the assumptions above [298]. Combing the above error bound with known gradient descent bounds for the accuracy of the lower-level problem yields bounds on the accuracy of the upper-level gradient. The standard lower-level bounds can vary by the specific algorithm ([298]

uses plain GD), but are in terms of $Q_\Phi = \frac{L_{\boldsymbol{x},\nabla_{\boldsymbol{x}}\Phi}}{\mu_{\boldsymbol{x},\Phi}}$ (the "condition number" for the strongly convex lower-level function) and the distance between the initialization and the minimizer.

Ref. [298] shows that $\hat{\boldsymbol{x}}(\boldsymbol{\gamma})$ is Lipschitz continuous in $\boldsymbol{\gamma}$ under the above assumptions, which intuitively states that the lower-level minimizer does not change too rapidly with changes in the hyperparameters. Further, $\nabla\ell(\boldsymbol{\gamma})$ is Lipschitz continuous in $\boldsymbol{\gamma}$ with a Lipschitz constant, $L_{\boldsymbol{\gamma},\nabla_{\boldsymbol{\gamma}}\ell}$, that depends on many of the constants given above.

The main theorems from [298] hold when the lower-level GD step size is $\alpha_\Phi = \frac{2}{L_{\boldsymbol{x},\nabla_{\boldsymbol{x}}\Phi}+\mu_{\boldsymbol{x},\Phi}}$ and the upper-level step size satisfies $\alpha_\ell \leq \frac{1}{L_{\boldsymbol{\gamma},\nabla_{\boldsymbol{\gamma}}\ell}}$. Then, the distance between the uth loss function value and the minimum loss function value, $\ell(\boldsymbol{\gamma}^{(u)}, \hat{\boldsymbol{x}}(\boldsymbol{\gamma}^{(u)})) - \ell(\hat{\boldsymbol{\gamma}}, \hat{\boldsymbol{x}}(\hat{\boldsymbol{\gamma}}))$, is bounded by a constant that depends on the starting distance from a minimizer (dependent on the initialization of $\boldsymbol{\gamma}$ and $\boldsymbol{x}$), Q_Φ, C_{GW}, the number of inner iterations, and the upper-level step size. The bound differs for strongly convex, convex, and possibly non-convex upper-level loss functions. Tab. 10.4 summarizes the sample complexity required to reach an ϵ-optimal point in each of these scenarios.

Table 10.4: Sample complexity to reach an ϵ-optimal solution of the deterministic bilevel problem using BA [298], for various assumptions on the upper-level loss function. Usually $\ell(\boldsymbol{\gamma})$ is non-convex and that case has the worst-case order results. The complexities show the total number of partial gradients of the upper-level loss (equal to the number of lower-level Hessians needed for estimating $\nabla\ell(\boldsymbol{\gamma})$ using (10.8)) and the partial gradients of the lower-level. The convex results use the accelerated BA method, which uses acceleration techniques similar to Nesterov's method [301] applied to the upper-level gradient step in Alg. 5.

Ji, Yang, and Liang [264] proposed two methods for Bilevel Optimization that improve on the sample complexities from [298] for non-convex loss functions under similar assumptions. The first, ITD-BiO (ITerative Differentiation), uses the unrolled method for calculating the upper-level gradient (see Section 10.1.3). The second, AID-BiO (Approximate Implicit Differentiation), uses the minimizer method with the implicit function theory perspective (see Section 10.1.1). Tab. 10.5 summarizes the sample complexities [264]. Much of the computational advantage of ITD-BiO and AID-BiO is in improving the iteration complexity with respect to the condition number (not shown in the summary table).

One of the main computational advantages of the AID-BiO and IFT-BiO methods in [264] over the BA algorithm Alg. 5 is a warm restart for the lower-level optimization. Although the hyperparameters change every outer iteration, the change is generally small enough that the stopping point of the previous lower-level descent is a better initialization than the noisy data (recall that [298] showed the lower-level minimizer is Lipschitz continuous in $\boldsymbol{\gamma}$). One can account for this warm restart when using automatic differentiation tools (backpropagation) [264]. The caption for Alg. 5 summarizes the other differences between BA and the BiO methods.

Table 10.5: A comparison of the finite-time sample complexity to reach an ϵ-solution of the deterministic bilevel problem when the upper-level loss function is non-convex using BA [298], AID-BiO [264], and ITD-BiO [264]. $\widetilde{\mathcal{O}}(\cdot)$ = order omits any $\log{(\epsilon)}^{\text{-1}}$ term.

The Bilevel Stochastic Approximation (BSA) method replaces the lower-level update in BA (see Alg. 5) with standard stochastic gradient descent. The corresponding upper-level step in BSA is a projected gradient step with stochastic estimates of all gradients. Another difference in the stochastic versions of the BA [298] and BiO [264] methods is that they use an inverse matrix theorem (based on the Neumann series) to estimate the Hessian inverse. Ref. [264] simplifies the inverse Hessian calculation to replace expensive matrix-matrix multiplications with matrix-vector multiplications. This same strategy makes backpropagation more computationally efficient than the forward mode computation for the unrolled gradient; see Appendix D.

10.2.3.3 Single-Loop

Recently, [296], [297] extended the double-loop analysis of [264], [298] to single-loop algorithms that alternate gradient steps in $\boldsymbol{x}$ and $\boldsymbol{\gamma}$.

Alg. 4 summarizes the single-loop algorithm TTSA [296]. The analysis of TTSA uses the same lower-level cost function assumptions as mentioned above for BSA [298] and one additional upper-level assumption: that ℓ is weakly convex with parameter $\mu_\ell > 0$, *i.e.*,

$$\ell(\boldsymbol{\gamma} + \boldsymbol{\delta}) \geq \ell(\boldsymbol{\gamma})\langle \nabla \ell(\boldsymbol{\gamma}), \boldsymbol{\delta} \rangle + \mu_\ell \left\|\boldsymbol{\delta}\right\|^2, \quad \forall \boldsymbol{\gamma}, \boldsymbol{\delta} \in \mathbb{R}^R.$$

TTSA assumes the lower-level gradient estimate is still unbiased and that its variance is now bounded as

$$\mathbb{E}\left[\|\nabla_{\boldsymbol{x}} \Phi(\boldsymbol{x}, \boldsymbol{\gamma}) - \tilde{\nabla}_{\boldsymbol{x}} \Phi(\boldsymbol{x}, \boldsymbol{\gamma})\|^2 \right] \leq \sigma^2_{\nabla_{\boldsymbol{x}} \Phi} \, (1 + \|\nabla_{\boldsymbol{x}} \Phi(\boldsymbol{x}, \boldsymbol{\gamma})\|^2).$$

Further, the stochastic upper-level gradient estimate, $\tilde{\nabla}_{\boldsymbol{\gamma}} \ell(\boldsymbol{\gamma}^{(u)}, \boldsymbol{x}^{(u+1)})$, includes a bias that stems from the nonlinear dependence on the lower-level Hessian. This bias decreases as the batch size increases.

The "two-timescale" part of TTSA comes from using different upper and lower step size sequences. The lower-level step size is larger and bounds the tracking error (the distance between $\hat{\boldsymbol{x}}$ and the $\boldsymbol{x}$ iterate) as the hyperparameters change (at the upper-level loss's relatively slower rate). Thus, [296] chose step-sizes such that $\alpha_\ell(u)/\alpha_\Phi(u) \to 0$. Specifically, if ℓ is strongly convex, then α_ℓ is $\mathcal{O}\left(u^{-1}\right)$ and α_Φ is $\mathcal{O}\left(u^{-2/3}\right)$. If ℓ is convex, then α_ℓ is $\mathcal{O}\left(u^{-3/4}\right)$ and α_Φ is $\mathcal{O}\left(u^{-1/2}\right)$.

Chen, Sun, Xiao, *et al.* [297] improved the sample complexity of TTSA. By using a single timescale, their algorithm, STABLE, achieves the "same order of sample complexity as the stochastic gradient descent method for the single-level stochastic optimization" [297]. However, the improved sample complexity comes at the cost of additional computation per iteration as STABLE can no longer trade a matrix inversion (of size $R \times R$) for matrix-vector products, as done in the [264]. Ref. [297] therefore recommended STABLE when sampling is more costly than computation or when R is relatively small.

The analysis of STABLE uses the same upper-level loss and lower-level cost function assumptions as listed above for BSA. Additionally, STABLE assumes that, $\forall \boldsymbol{x}$, $\nabla_{\boldsymbol{\gamma}} \ell(\boldsymbol{\gamma}\,;\, \boldsymbol{x})$ is Lipschitz continuous in $\boldsymbol{\gamma}$. This condition is easily satisfied as many upper-level loss functions do not regularize $\boldsymbol{\gamma}$. Further, those that do often use a squared 2-norm, *i.e.*, Tikhonov-style regularization, that has a Lipschitz continuous gradient. Additionally, rather than bounding the gradient norms as in assumptions Aℓ2 and AΦ5, [296] assumes the following moments are bounded:

- the second and fourth moment of $\nabla_{\boldsymbol{\gamma}} \ell(\boldsymbol{\gamma}\,;\, \boldsymbol{x})$ and $\nabla_{\boldsymbol{x}} \ell(\boldsymbol{\gamma}\,;\, \boldsymbol{x})$ and
- the second moment of $\nabla_{\boldsymbol{\gamma x}} \Phi(\boldsymbol{x}\,;\, \boldsymbol{\gamma})$ and $\nabla_{\boldsymbol{xx}} \Phi(\boldsymbol{x}\,;\, \boldsymbol{\gamma})$,

ensuring that the upper-level gradient is Lipschitz continuous.

Like the previous algorithms discussed, STABLE evaluates the minimizer result (10.8) at non-minimizer lower-level iterates, $\boldsymbol{x}^{(T)}(\boldsymbol{\gamma}^{(u)})$, to estimate the hyperparameter gradient. However, it differs in how it estimates and uses the gradients. STABLE replaces the upper-level gradient in TTSA line 4 with

$$\boldsymbol{g} = \nabla_{\boldsymbol{\gamma}} \ell^{(u)} - \underbrace{(\Delta_{\boldsymbol{x\gamma}}^{(u)})'}_{\text{Prev. } \tilde{\nabla}_{\boldsymbol{x\gamma}} \Phi^{(u)}} \underbrace{(\Delta_{\boldsymbol{xx}}^{(u)}}_{\text{Prev. } \tilde{\nabla}_{\boldsymbol{xx}} \Phi^{(u)}})^{-1} \nabla_{\boldsymbol{x}} \ell^{(u)}. \tag{10.26}$$

Taking inspiration from variance reduction techniques for single-level optimization problems, *e.g.*, [302], STABLE recursively updates the newly defined matrices as follows:

$$\Delta_{\boldsymbol{x}\boldsymbol{\gamma}}^{(u)} = \mathcal{P}_{\|\Delta\| \leq C_{\nabla_{\boldsymbol{\gamma}\boldsymbol{x}}\Phi}} \left((1-\tau_u) \underbrace{(\Delta_{\boldsymbol{x}\boldsymbol{\gamma}}^{(u-1)} - \tilde{\nabla}_{\boldsymbol{x}\boldsymbol{\gamma}} \Phi^{(u-1)})}_{\text{Recursive update}} + \underbrace{\tilde{\nabla}_{\boldsymbol{x}\boldsymbol{\gamma}} \Phi^{(u)}}_{\text{New estimate}} \right)$$

$$\Delta_{\boldsymbol{x}\boldsymbol{x}}^{(u)} = \mathcal{P}_{\Delta \succeq \mu_{\boldsymbol{x},\Phi} \boldsymbol{I}} \left((1-\tau_u) \overbrace{(\Delta_{\boldsymbol{x}\boldsymbol{x}}^{(u-1)} - \tilde{\nabla}_{\boldsymbol{x}\boldsymbol{x}} \Phi^{(u-1)})} + \overbrace{\tilde{\nabla}_{\boldsymbol{x}\boldsymbol{x}} \Phi^{(u)}} \right).$$

In the $\Delta_{\boldsymbol{x}\boldsymbol{\gamma}}^{(u)}$ update, the projection onto the set of matrices with a maximum norm helps ensure stability by not allowing the gradient to get too large. The projection in the $\Delta_{\boldsymbol{x}\boldsymbol{x}}^{(u)}$ update is an eigenvalue truncation that ensures positive definiteness of the estimated Hessian in this Newton-based method. After computing the gradient $\boldsymbol{g}$ (10.26), the upper-level update is a standard descent step as in Alg. 4 line 5.

STABLE [297] also uses the recursively estimated gradient matrices in the lower-level cost function descent. It replaces the standard gradient descent step in Alg. 4 line 3 with one that uses second order information:

$$\boldsymbol{x}^{(u+1)} = \boldsymbol{x}^{(u)} - \underbrace{\alpha_\Phi(u) \tilde{\nabla}_{\boldsymbol{x}} \Phi(\boldsymbol{x}^{(u)}; \boldsymbol{\gamma}^{(u)})}_{\text{Standard GD step}} - \underbrace{(\Delta_{\boldsymbol{x}\boldsymbol{x}}^{(u)})^{-1} (\Delta_{\boldsymbol{\gamma}\boldsymbol{x}}^{(u)})' (\boldsymbol{x}^{(u+1)} - \boldsymbol{x}^{(u)})}_{\text{New term}}.$$

With these changes, STABLE is able to reduce the iteration complexity relative to TTSA as summarized in Tab. 10.3.

10.2.4 Summary of Methods

There are many variations of gradient-based methods for optimizing bilevel problems, especially when one considers that many of the upper-level descent strategies can work with either the minimizer or unrolled approach discussed in Section 10.1. There is no clear single "best" algorithm for all applications; each algorithm involves trade-offs.

Building on the minimizer and unrolled methods for finding the upper-level gradient with respect to the hyperparameters, $\nabla \ell(\boldsymbol{\gamma})$, double-loop algorithms are an intuitive approach. Although optimizing the lower-level problem every time one takes a gradient step in $\boldsymbol{\gamma}$ is computationally expensive, the lower-level problem is is embarrassingly parallelizable across samples. Specifically, one can optimize the lower-level cost for each training sample independently before averaging the resulting gradients to take an upper-level gradient step. In the typical scenario when training is performed offline, training wall-time can therefore be dramatically reduced by using multiple processors.

Single-loop algorithms remove the need to optimize the lower-level cost function multiple times. The single-loop algorithms that consider a system of equations often accelerate convergence using Newton solvers [143], [192]. However, the optimality system grows quickly when there are multiple training images, and may become too computationally expensive as J increases [162]. Another type of single-loop algorithm uses alternating gradient steps in $\boldsymbol{x}$ and $\boldsymbol{\gamma}$ [296], [297]. Although each method has slight variations (such as whether it uses momemtum), these single-loop methods are generally equivalent to considering $T = 1$ in the double-loop methods.

This section organized algorithms based on the number of for-loops; double-loop algorithms have two loops while single-loop algorithms have one[5]. However, there are many other ways in which bilevel optimization methods differ and not all methods fall cleanly into one group. One such example is the Penalty method [303]. The Penalty method forms a single-level, constrained optimization problem, with the constraint that the gradient of the lower-level cost function should be zero, $\nabla_{\boldsymbol{x}} \Phi(\boldsymbol{x}\,; \boldsymbol{\gamma}) = \boldsymbol{0}$. (This step is similar to the derivation of the minimizer approach via KKT conditions; see Section 10.1.1.2.) Rather than forming the Lagrangian as in (10.24), [303] penalizes the norm of the

[5] As noted at the start of the section, this loop counting does not include the loop in CG or in backpropagation.

gradient, with increasing penalties as the upper iterations increase. Thus, the Penalty cost function[6] at iteration u is

$$p(\boldsymbol{\gamma}, \boldsymbol{x}) = \ell(\boldsymbol{\gamma}; \hat{\boldsymbol{x}}(\boldsymbol{\gamma})) + \lambda^{(u)} \|\nabla_{\boldsymbol{x}} \Phi(\boldsymbol{x}; \boldsymbol{\gamma})\|_2^2 .$$

The penalty variable sequence, $\lambda^{(u)}$, must be positive, non-decreasing, and divergent ($\lambda^{(u)} \to \infty$).

Penalty [303] incorporates elements of both double-loop and single-loop algorithms. Similar to the double-loop algorithms, Penalty takes multiple gradient descent steps in the lower-level optimization variable, $\boldsymbol{x}$, before calculating and updating the hyperparameters. However, Penalty forms a single-level optimization problem that could be optimized using techniques such as those used in single-loop algorithms.

Another variant on a double-loop bilevel optimization method is to optimize a lower-level surrogate function $\tilde{\Phi}(\boldsymbol{x}; \boldsymbol{\gamma}^{(u)})$ instead of optimizing $\Phi(\boldsymbol{x}; \boldsymbol{\gamma}^{(u)})$. For example, [304] replaces Φ with its first-order approximation around the current solution point $(\boldsymbol{\gamma}^{(u)}, \hat{\boldsymbol{x}}(\boldsymbol{\gamma}^{(u)}))$. Because this approximation is only reliable in the neighborhood of $(\boldsymbol{\gamma}^{(u)}, \hat{\boldsymbol{x}}(\boldsymbol{\gamma}^{(u)}))$, [304] adds the proximal term $\lambda\|\boldsymbol{\gamma} - \boldsymbol{\gamma}^{(u)}\|^2$ to the upper-level loss function at each outer iteration, where λ is a positive tuning parameter.

The finite-time complexity analyses [264], [296]–[299] justify the use of gradient-based bilevel methods for problems with many hyperparameters, as none of the sample complexity bounds involved the number of hyperparameters. This is in stark contrast with the hyperparameter optimization strategies in Section 8.2. However, the per-iteration cost for bilevel methods is still large and increasing with the hyperparameter dimension. Further, the conditions on the lower-level cost function AΦ1-AΦ6 seem restrictive and may not be satisfied in practice. Complexity analysis based on more relaxed conditions could be very valuable.

Because of the restrictive conditions in the complexity analysis, it is generally infeasible to compute theoretically justified step-sizes and other algorithm parameters in the single-loop and double-loop methods [264], [296]–[299]. Thus, one must often resort to grid searches or use heuristics, such as those discussed in Section 10.2.1, to select these algorithm parameters. Ref. [299] comments on one example of how empirical practice can differ from theory. Although their theory requires that the number of iterates of the Neumann series used to approximate the inverse Hessian matrix grows with the desired solution accuracy, the authors found that using a few iterates was sufficient (and faster) in practice.

Gradient-based and other hyperparameter optimization methods are active research areas, and the trade-offs continue to evolve. Although it currently seems that gradient-based bilevel methods make sense for problems with many hyperparameters, new methods may overtake or combine with what is presented here. For example, many bilevel methods (and convergence analyses thereof) use classical gradient descent for the lower-level optimization algorithm, whereas [305] showed that the Optimized Gradient Method (OGM) has better convergence guarantees and is optimal among first-order methods for smooth convex problems [306]. These advances provide opportunities for further acceleration of bilevel methods.

[6] This is a simplification; [303] allows for constraints on $\boldsymbol{x}$ and $\boldsymbol{\gamma}$.

CHAPTER 11

RQ#5: Revisiting the Simple Filter Learning Experiment

This chapter applies the bilevel methodology defined in Chapter 10 to the filter learning experiment from Section 9.2. Recall that Section 9.2 examined a simple experiment with PWC signals where learned transforms did not denoise test data as well as a handcrafted transform. The learned transforms were trained to sparsify the training data, but, to efficiently minimize the training cost, we had to introduce an auxiliary variable and tuning parameter. The impact of introducing these variables and "splitting" the training cost, was to learn smoother transforms that did not denoise the test data as well as the sharp, handcrafted transform.

This chapter addresses RQ#5: **How does the bilevel method compare to handcrafted filters and filters learned in a non-task-based method?** We hypothesized that

1. the task-based nature of the bilevel problem will ensure that filters learned in a bilevel manner outperform (on average) handcrafted filters and
2. the bilevel method would learn filters that are more similar to the handcrafted TV-based filters than the sparsity approach in Section 9.2.

This chapter expands on the results presented in [10]:

> C. Crockett and J. A. Fessler, "Motivating bilevel approaches to filter learning: A case study," in *2021 IEEE International Conference on Image Processing (ICIP)*, IEEE, Sep. 19, 2021, pp. 2803–2807, ISBN: 978-1-66544-115-5. DOI: 10.1109/ICIP42928.2021.9506489

11.1 Methods

This chapter considers the example bilevel problem (Ex) in the denoising setting ($\boldsymbol{A} = \boldsymbol{I}$). The sparsifying function is (CR1N) with $\epsilon = 0.1$ and the learning parameters are either $\boldsymbol{\gamma} = (\beta_1, \boldsymbol{c})$ or $\boldsymbol{\gamma} = \boldsymbol{c}$. We consider initializing $\boldsymbol{c}$ with finite differencing filters and with random initializations.

Note that (Ex) does not have any constraints on the tuning parameters. Unlike (9.21), there are no trivial solutions; we should not learn the $\mathbf{0}$ filter because that does not provide any benefit to denoising. Likewise, the task-based approach means that we should not learn equivalent, but shifted, versions of the same filters, as we saw in the simple 1D experiment, where we more often learned two shifted versions of the finite differencing filter rather than learning an extended finite differencing filter, *i.e.*, we tended to learn transforms similar to $\boldsymbol{T}_2$ in (9.19). However, as we will discuss further in each of our experiments, the bilevel problem is highly non-convex and the amount of diversity in the learned filters depends on how we initialize the upper-level optimization algorithm.

As before, we test learning one or two length-4 filters from 1D PWC training data and compare the learned filters with the TV-based handcrafted filters. However, because the bilevel problem takes much longer to train, we only use 128 signals instead of 1,024. We use the set of 128 test signals as defined in Section 9.2.2.

To optimize the lower-level cost function, we use gradient descent with the step size $\frac{1}{L}$ with L defined in (10.19). Note that L changes every time we take a gradient step on the hyperparameters because the Lipschitz constant of $\nabla_{\boldsymbol{x}} \Phi$ is a function of $\boldsymbol{\beta}$ and $\boldsymbol{c}$. We could easily imagine setting a norm constraint on $\boldsymbol{c}$. However, setting an upper limit on $\boldsymbol{\beta}$ and the norm of $\boldsymbol{c}$ is likely to either yield poor denoising performance (if the limit is too small) or slow convergence of the lower-level cost function (if the limit is too large). Thus, we cannot reasonably upper-bound L. Another approach to avoiding the L update would be to simply declare a value and allow the bilevel problem to learn $\boldsymbol{c}$ and β_k that works with the set value. However, this approach risks losing theoretical convergence guarantees on the lower-level problem.

We use reverse-mode backpropagation (defined in Appendix D) to compute $\nabla \ell(\boldsymbol{\gamma})$ and Adam [292] with the default settings to take a gradient step with respect to $\boldsymbol{\gamma}$. Section 11.2.1 discusses the influence of T; after that section, we use $T = 100$ to learn a single filter, a single filter and its corresponding tuning parameter, and two filters. For each of the configurations in the sections below, unless otherwise noted, we ran 7,000 upper-level iterations and then selected the hyperparameter with the lowest upper-level loss as the returned minimizer.

We used the same metrics as in Section 9.2 to measure how similar the learned filters are from the handcrafted filters. However, we additionally have to consider the norm of the filters, since there is no longer a unit-norm constraint.[1] Therefore, we also present the norm of the learned filters.

11.2 Results

11.2.1 Effect of the Number of Lower-level Iterations

The number of iterations of the lower-level optimization algorithm, T, determines how closely we approximate $\hat{\boldsymbol{x}}$, which in turn influences the gradient calculation for $\nabla_{\boldsymbol{\gamma}} \ell(\boldsymbol{\gamma}\,; \hat{\boldsymbol{x}}(\boldsymbol{\gamma}))$. Further, recall that the proposed bilevel method for analysis filters updates the Lipschitz constant for the lower-level algorithm after every hyperparameter gradient step.

Fig. 11.1 and Fig. 11.2 show the upper-level error per bilevel iteration when $T = 10$ where we evaluate the error at the current estimate of the denoised signal $\tilde{\boldsymbol{x}}(\boldsymbol{\gamma}^{(u)}) = \boldsymbol{x}^{(10)}(\boldsymbol{\gamma}^{(u)})$. We initialize $\boldsymbol{c}_1$ with $\boldsymbol{c}_{\text{FD}} = 1/\sqrt{2}\begin{bmatrix}0 & \text{-1} & 1 & 0\end{bmatrix}$. Fig. 11.1 considers when $\boldsymbol{\gamma} = \boldsymbol{c}$ and Fig. 11.2 considers when $\boldsymbol{\gamma} = (\boldsymbol{c}, \beta_1)$. After initially decreasing, the loss function tends to steadily *increase*. Although Adam does not guarantee a monotonically decreasing loss function [292], a steady increase is concerning.

Upon investigating some of the iterations where the loss function increased, we found the gradient calculation was correct, and a gradient step in $\boldsymbol{\gamma}$ would decrease the loss function *if L were held constant*. However, by updating L after every gradient step in the hyperparameters, without accounting for this in the gradient calculation, the loss function tended to increase. Specifically, the loss function increased when the hyperparameter update caused L to increase. Larger values of L correspond to a smaller step size for the lower-level image denoising problem, and thus a worse approximation to $\hat{\boldsymbol{x}}$ given a fixed number of iterations.

We considered two high-level strategies to ensure the loss function did not steadily increase. One option is to fix the step size[2] and allow the bilevel problem to adjust the norm of $\boldsymbol{c}$ and $\boldsymbol{\beta}$ appropriately. When T is small enough

[1] The scale factor does not influence the similarity and distance metrics.

[2] In this approach, it makes more conceptual sense to think about fixing the step size, rather than fixing L because L would no longer represent the Lipschitz constant of the cost function.

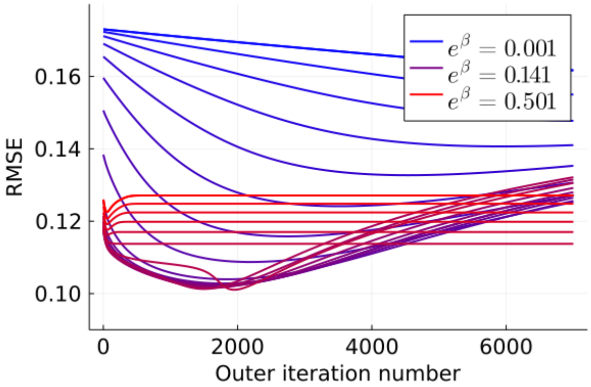


Figure 11.1: Upper-level RMSE as defined in (9.10) averaged over the 128 training signals versus the upper-level iteration when learning $\boldsymbol{c}$ and when T = 10. The different colored lines show different set values of $\boldsymbol{\beta}$.

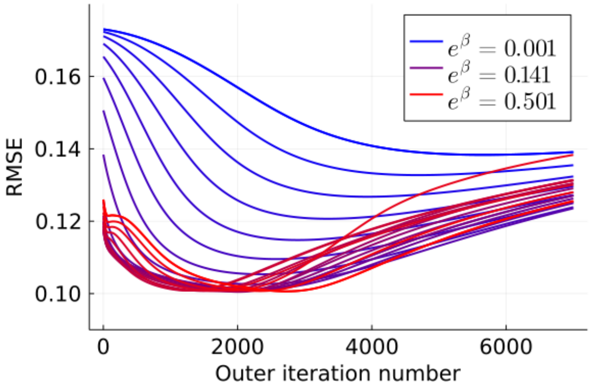


Figure 11.2: RMSE as defined in (9.10) averaged over the 128 training signals versus the upper-level iteration when learning $\boldsymbol{c}$ and $\boldsymbol{\beta}$ when T = 10. The different colored lines show different initializations of $\boldsymbol{\beta}$.

that the lower-level cost function does not reach convergence, this approach would allow the optimization algorithm to take larger step sizes than the step size corresponding to the Lipschitz constant.

The second option is to run the lower-level optimization algorithm to convergence either by selecting a large enough value for T or iterating until some lower-level convergence criteria is met. This approach requires us to set a convergence tolerance and it could take much longer, since the lower-level problem may take many iterations to converge. However, one can use a warm start as in the methods from [264] to decrease the total number of lower-level iterations; see Section 10.2.3.2. A benefit of this approach is that, once the hyperparameters are learned, the corresponding step size for the lower-level problem has all the theoretical convergence guarantees of the chosen optimization method.

We take the second approach of setting T large enough for the lower-level to be close to convergence due to its additional theoretical convergence guarantees. To measure convergence of the cost function, we examine the normalized change in $\boldsymbol{x}$ on the last iteration:

$$\frac{\|\boldsymbol{x}_T - \boldsymbol{x}_{T-1}\|}{\boldsymbol{x}_T}. \tag{11.1}$$

Fig. 11.3 and Fig. 11.4 show a histogram of this convergence measure over many training signals at the start of the bilevel algorithm and at the end. For these plots, we initialized the bilevel algorithm with a random filter and learned both $\boldsymbol{c}$ and $\boldsymbol{\beta}$. In both plots, the convergence measure (11.1) is small, suggesting that the optimization algorithm is close to convergence. Further, the convergence measure is smaller at the end of the bilevel algorithm, when the filters are tuned for denoising the signals, than at the start of the algorithm, when the filters are mostly random noise.

After verifying that the lower-level optimization algorithm was relatively close to convergence after $T = 100$ iterations, we re-examined the estimated upper-level loss function. In contrast to the plots in Fig. 11.1 and 11.2 for $T = 10$, the errors in Fig. 11.5 and 11.6 steadily decreases with the upper-level iteration when $T = 100$. Using more lower-level iterations (100 instead of 10) is thus sufficient for the upper-level loss function to decrease in this simple experiment because the lower-level estimate of the minimizer, $\tilde{\boldsymbol{x}}^{(T)}(\boldsymbol{\gamma}^{(u)})$, remains close enough to convergence.

For the remainder of this chapter, we use $T = 100$.

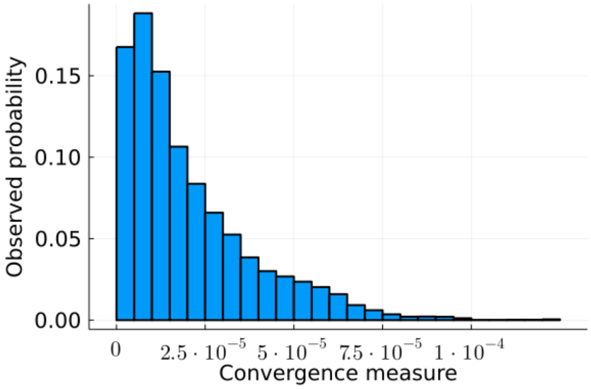


Figure 11.3: Histogram of the convergence measure (11.1) for 128 signals for the *first* 100 Adam iterations when learning $\boldsymbol{c}$ and $\boldsymbol{\beta}$ from a random initialization.

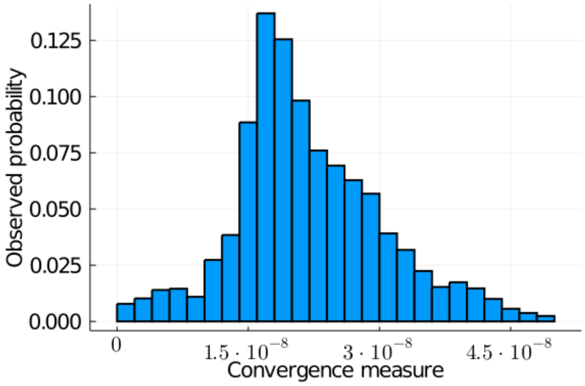


Figure 11.4: Histogram of the convergence measure (11.1) for 128 signals for the *last* 100 Adam iterations (out of a total of 7,000 iterations) when learning $\boldsymbol{c}$ and $\boldsymbol{\beta}$. Note the much smaller horizontal scale than in Fig. 11.3.

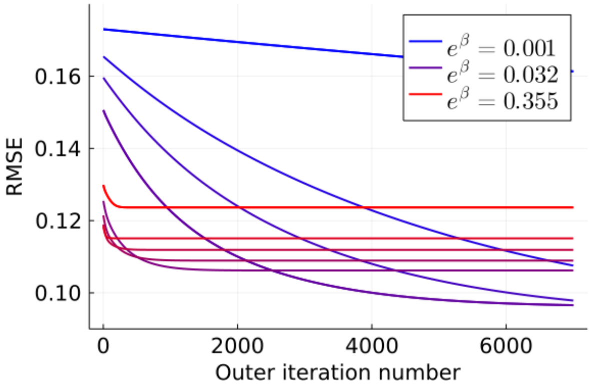


Figure 11.5: Upper-level RMSE as defined in (9.10) averaged over the 128 training signal when learning $\boldsymbol{c}$ and when $T = 100$. The different colored lines show different set values of $\boldsymbol{\beta}$.

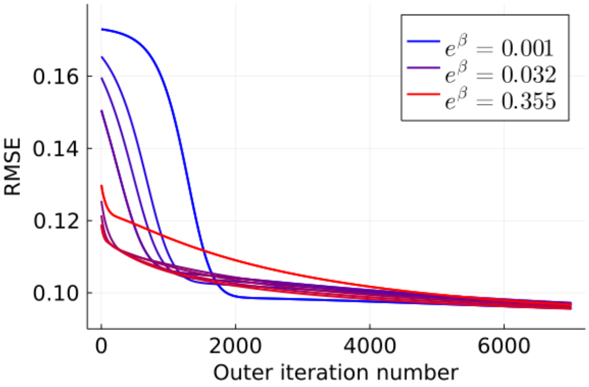


Figure 11.6: Upper-level RMSE as defined in (9.10) averaged over the 128 training signal when learning $\boldsymbol{c}$ and $\boldsymbol{\beta}$ when $T = 100$. The different colored lines show different initializations of $\boldsymbol{\beta}$.

11.2.2 Learning One Filter

We next examine the learned filters with $T = 100$. Our first test involved setting β and learning only the filter. We initialized the filter with $\boldsymbol{c}_{\mathrm{FD}}$ and tested nine values of e^{β_1} between 0.001 and 0.35 (we set $e^{\beta_0} = 1$ so that it has no effect).

With this informed initialization for $\boldsymbol{c}$, the mean and maximum angle between the learned filters and $\boldsymbol{c}_{\mathrm{FD}}$ are 1.3 degrees and 1.5 degrees respectively. Further, the minimum upper-level loss function shows no obvious trend with the initialization of the tuning parameter, β_1. Although the bilevel filter is not forced to take the simple form given in (9.12), we compared our results by forming an approximation to d by normalizing the filters and taking the mean of the two larger absolute value elements (after normalization). For the bilevel filters, this approximation to d is between 0.704 and 0.707 for all tuning parameter settings (with a mean of 0.706). In comparison, using the single-level split sparsity training cost in Section 9.2, d was as low as 0.65 (see Fig. 11.7). This means that the task-based bilevel filters are less smoothed than the previous, non-task-based filters, despite our relaxation of the 0-norm to the corner rounded 1-norm!

There is no discernible effect of β on the angle between the learned filter or on the approximation of d. However, as seen in Fig. 11.7, the minimum loss function value depends on β[3], The loss function measures the training error; this is our expected test error assuming that the bilevel problem does not over-fit the training data. Considering that our goal is to minimize the loss function, we should therefore either initialize β carefully or incorporate β into $\boldsymbol{\gamma}$ and learn it. We take the latter approach below.

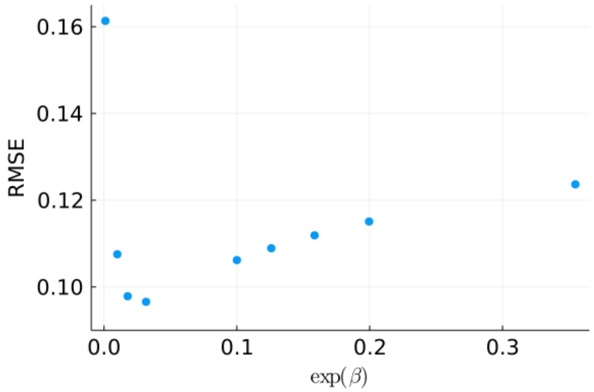


Figure 11.7: Upper-level RMSE as defined in (9.10) versus β_1 for learning a single filter using the bilevel set-up in (Ex).

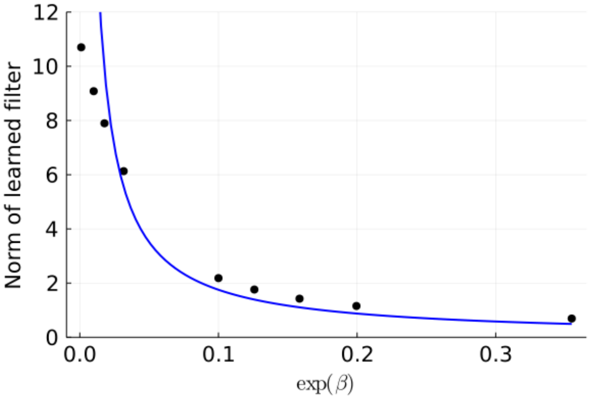


Figure 11.8: The scattered points show the norm of the learned filter versus different, set values of the tuning parameter, β_1. For comparison, the solid line plots the expected relation for a 1-norm sparsifying function, *i.e.*, $||\boldsymbol{c}|| = \alpha / e^{\beta_1}$ where α is a constant that describes the inverse relationship between β_1 and the norm of the learned filter.

Before investigating minimizing (Ex) with both $\boldsymbol{c}$ and β in $\boldsymbol{\gamma}$, we make two more observations about the results from learning $\boldsymbol{c}$ with β set. First, Fig. 11.8 shows the filter 2-norm versus the tuning parameter, β_1. There is a general trend for the norm to decrease as β_1 increases. Because ϕ is a corner-rounded 1-norm, we cannot quite absorb

[3] This might suggest that using angle to $\boldsymbol{c}_{\mathrm{FD}}$ is a bad measure of quality for the learned filters. However, we can also interpret these results as learning a good filter, but without a good corresponding tuning parameter. The results in the next section support this interpretation.

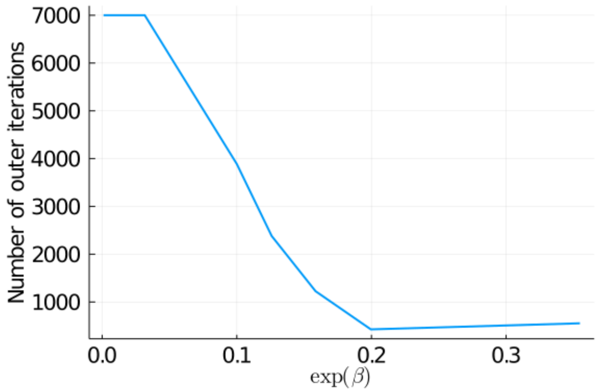


Figure 11.9: Number of outer (Adam) iterations to reach the lowest loss function evaluation. Note that for the smallest values of the tuning parameter, β, the loss function was still descending at iteration 7,000.

the change in filter magnitude into β_1. However, the reverse relationship between the filter magnitude and tuning parameter value has a similar effect.

Finally, we note that the bilevel problem tends to converge faster with larger values of β, as seen in Fig. 11.9. Thus, filters learned with relatively large values of β are still close to $\boldsymbol{c}_{\mathrm{FD}}$, while taking less time to learn. Although the filters are close to $\boldsymbol{c}_{\mathrm{FD}}$, some achieve a lower loss function value largely due to the difference in the norm of the learned filter.

11.2.3 Learning One Filter and One Tuning Parameter

Now we consider adding β_1 to the $\boldsymbol{\gamma}$ hyperparameter vector in the bilevel problem (Ex), while still learning only a single filter. Our initialization for these tests is the same: $\boldsymbol{c}_{\mathrm{FD}}$ for the filters and a value of e^{β_1} between 0.001 and 0.35.

As when we optimized for only $\boldsymbol{c}$, when we optimized $\boldsymbol{c}$ and β, there was no apparent trend in the measures of closeness to the $\boldsymbol{c}_{\mathrm{FD}}$ filter and the initialization for β. The equivalent d value for the filters (the average of the two largest in absolute value elements after normalization) is between $\frac{1}{\sqrt{2}}0.997$ and $\frac{1}{\sqrt{2}}$, with a mean value just below $\frac{1}{\sqrt{2}}$. The mean and maximum angle between the learned filters and $\boldsymbol{c}_{\mathrm{FD}}$ are 1.3 degrees and 1.5 degrees respectively. Thus, the filters are closer, on average, to $\boldsymbol{c}_{\mathrm{FD}}$ than the filters learned when only descending on $\boldsymbol{c}$.

Learning β has additional benefits. As seen in Fig. 11.6, the upper-level RMSE when we also learn the tuning parameter has a range of 0.0956 to 0.0972, which is less than the corresponding loss function value for learning only $\boldsymbol{c}$ (which has a range of 0.0966 to 0.1613).

The downside of learning β is the increased training time. All experiments were still decreasing after the full 7,000 iterations, though the flatness at the end of the curves in Fig. 11.6 suggests we were close to convergence.

11.2.4 Random Initialization

The above experiments all initialized $\boldsymbol{c}$ to $\boldsymbol{c}_{\mathrm{FD}}$. A more interesting test of the non-convex bilevel problem is a random initialization. For this, we initialized $\boldsymbol{c}$ with 100 normalized Gaussian noise realizations, set β_1 = -1, and ran 10,000 upper-level iterations. Fig. 11.12 shows that there is a strong positive correlation (correlation coefficient of 0.77) between the minimum loss function value and the angle between the learned filter and $\boldsymbol{c}_{\mathrm{FD}}$, suggesting the angle from $\boldsymbol{c}_{\mathrm{FD}}$ is a reasonable indicator of the denoising performance during training.

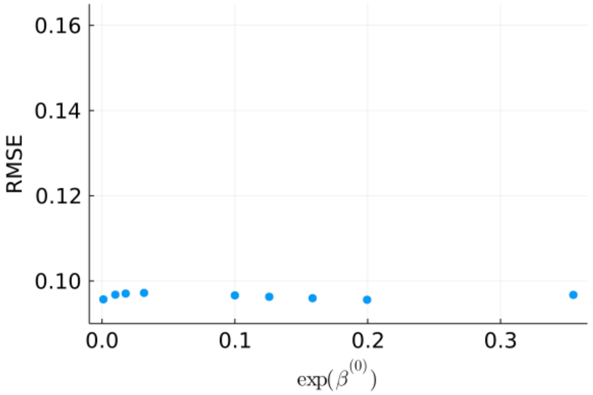


Figure 11.10: Upper-level RMSE as defined in (9.10) versus β_1 when learning a single filter and its tuning parameter using the bilevel set-up in (Ex). Shown on the same scale as Fig. 11.7.

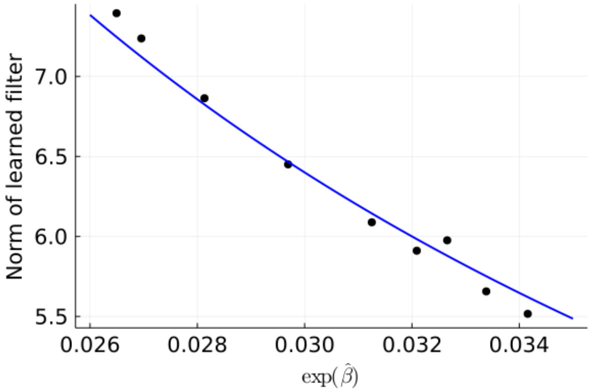


Figure 11.11: The scattered points show the norm of the learned filter versus different, learned values of the tuning parameter, $\hat{\beta}_1$. For comparison, the solid line plots the expected relation for a 1-norm sparsifying function, *i.e.*, $\|\boldsymbol{c}\| = \alpha/e^{\beta_1}$ where α is a constant that describes the inverse relationship between β_1 and the norm of the learned filter.

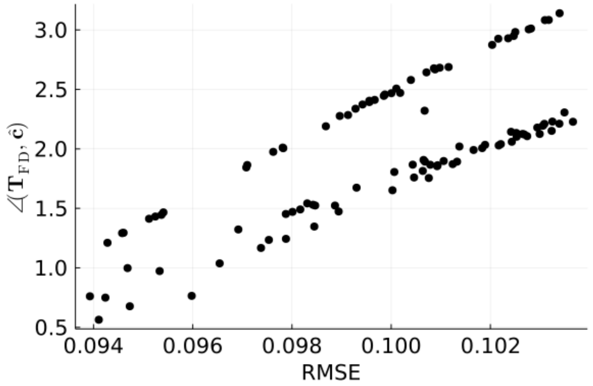


Figure 11.12: Scatter plot of the angle between learned filter and $\boldsymbol{c}_{\mathrm{FD}}$ and the RMSE on the test data as defined in (9.10) evaluated at the learned filter. The correlation coefficient is 0.77, suggesting that the angle from $\boldsymbol{c}_{\mathrm{FD}}$ is a relatively good indicator of the denoising performance during training of the learned filter.

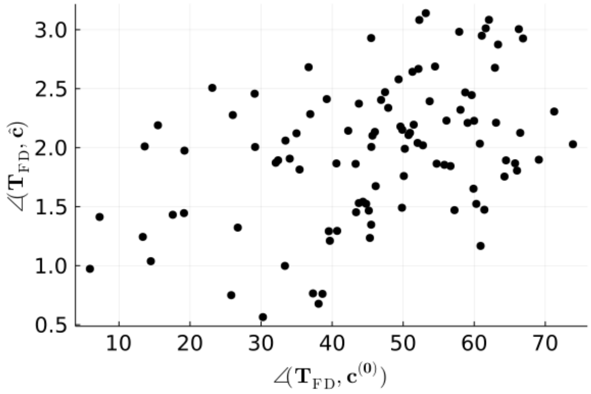


Figure 11.13: Scatter plot of the angle between the learned filter $\hat{\boldsymbol{c}}$ and $\boldsymbol{c}_{\mathrm{FD}}$ and the angle between the randomly initialized filter $\boldsymbol{c}^{(0)}$ and $\boldsymbol{c}_{\mathrm{FD}}$. The correlation coefficient is 0.4, suggesting there is some relation between the starting angle and final angle.

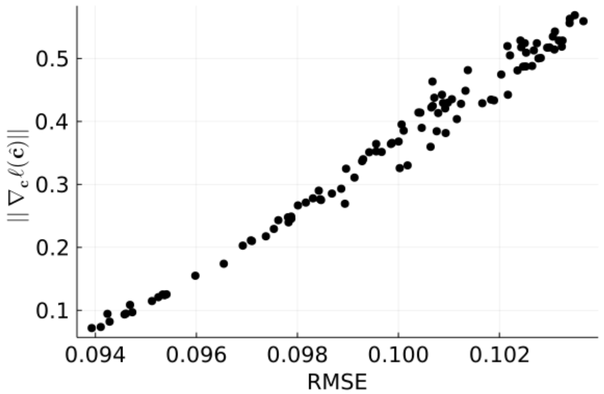


Figure 11.14: Scatter plot a metric of convergence versus the denoising performance during training of the learned filter. Here, convergence is measured as the norm of the upper-level gradient with respect to the filter coefficients evaluated at the learned filter ($\hat{\boldsymbol{c}} = \boldsymbol{c}^{(10,000)}$). The correlation coefficient is 0.99, suggesting a strong relation between the filter initialization and the norm of the gradient.

On average, the learned filters are separated from $\boldsymbol{c}_{\text{FD}}$ by 2.0 degrees. Fig. 11.13 shows that, for a wide range of random initial filters, all but two of the learned filters are within 0.56 to 3.14 degrees of $\boldsymbol{c}_{\text{FD}}$. This result is promising considering the highly non-convex nature of the bilevel problem.

However, Fig. 11.13 also shows a weak correlation between how close the random initialization is to $\boldsymbol{c}_{\text{FD}}$ and how close the learned filter is to $\boldsymbol{c}_{\text{FD}}$. Fig. 11.14 suggests this relation is related to filters that were randomly initialized to be further from $\boldsymbol{c}_{\text{FD}}$ being further from convergence after 10,000 upper-level iterations.

11.2.5 Denoising Performance

To test denoising performance of the learned filters, we used the lower-level cost function from (Ex) with the same corner-rounded 1-norm sparsifying function as in training process. When initialized with $\boldsymbol{c}_{\text{FD}}$, the gradient descent nature of the bilevel problem suggests that learned filters will perform better than $\boldsymbol{c}_{\text{FD}}$ on the training data.

The right half of Tab. 11.1 shows the results for the learned filters corresponding to the smallest ($\hat{\boldsymbol{c}}_{\text{best}}$) and largest ($\hat{\boldsymbol{c}}_{\text{worst}}$) RMSE values on the test data across all 100 learned filters from the random initialization experiment described in the previous section. Fig. 11.15 shows a histogram of the average RMSEs for all the random initializations. For comparison, Tab. 11.1 also reports the denoising performance of $\boldsymbol{c}_{\text{FD}}$ for the same lower-level cost function, with the filter norm and β_1 tuned (150 and -6.5 respectively) using a grid search to minimize RMSE.

	Transforms: Denoised using (11.2)		Filters (Bilevel method): Denoised using (Ex)		
	$\boldsymbol{T}_{\text{FD}}$	$\hat{\boldsymbol{T}}_{\lambda=0.23}$	$\boldsymbol{c}_{\text{FD}}$	$\hat{\boldsymbol{c}}_{\text{best}}$	$\hat{\boldsymbol{c}}_{\text{worst}}$
RMSE	0.081 ± 0.035	0.131 ± 0.035	0.083 ± 0.026	0.089 ± 0.022	0.103 ± 0.022

Table 11.1: Average and standard deviation of the RMSE as defined in (9.10) for the 128 test signals. **Left columns**: denoising using (11.2) with $\boldsymbol{T}$ being $\boldsymbol{T}_{\text{FD}}$ or learned according to (9.3) with the training tuning parameter λ set as 0.23. **Right columns**: denoising using the lower-level cost function in (Ex) with $\boldsymbol{c}_{\text{FD}}$ and the best and worst performing filters (judged on the test data) learned using the bilevel method with random initializations. The mean denoising RMSE across all random initializations was 0.097.

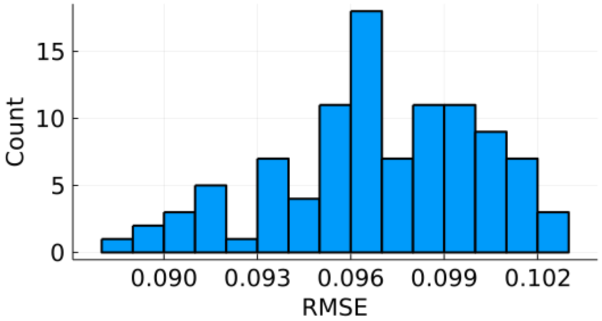


Figure 11.15: Histogram of the average RMSE for the test signals for all 100 random initializations $\boldsymbol{c}^{(0)}$ in the set-up bilevel filter learning.

The left half of Tab. 11.1 repeats the denoising results from Section 9.2. Recall that the denoising cost in that section was

$$\hat{\boldsymbol{x}}(\boldsymbol{y}) = \operatorname*{argmin}_{\boldsymbol{x}} \frac{1}{2} \|\boldsymbol{x} - \boldsymbol{y}\|_2^2 + \beta \sum_j \min_{z_j} \left\|\boldsymbol{T}\boldsymbol{P}_j\boldsymbol{x} - z_j\right\|_2^2 + \alpha \left\|z_j\right\|_0, \tag{11.2}$$

where $\boldsymbol{T}$ was either a learned transform or $\boldsymbol{T}_{\text{FD}}$. The lower-level cost function for the transforms uses the 0-norm to encourage sparsity while the cost function for the filters uses (CR1N). This difference in the cost function definitions explains the increase in error from 0.081 when using $\boldsymbol{T}_{\text{FD}}$ to 0.803 when using $\boldsymbol{c}_{\text{FD}}$. Namely, the increased error illustrates the cost of replacing the split 0-norm with a convex, smooth sparsity penalty function. Recall that we used (CR1N) in the bilevel formulation because of the smoothness requirements to compute the upper-level gradient; see Section 10.1.3. An interesting avenue for future work would be to compare the denoising performance of the bilevel methods that do not require smoothing (the unrolled methods in Section 10.1.3.2 and the translation to a single level methods from Section 10.1.2). Another interesting comparison would be to use a non-convex sparsifying functions such as $\phi_{\text{GR}}(z) = \alpha|z|/(1 + \alpha|z|)$ [307] or the corresponding smoothed version achieved by rounding the corners of the absolute value functions.

Although the handcrafted filter $\boldsymbol{c}_{\text{FD}}$ performs worse at denoising than the handcrafted transform $\boldsymbol{T}_{\text{FD}}$ does with (11.2) due to the structure of the regularizer in the lower-level cost function, the learned filters perform better than the learned transform. In other words, the learned filters denoise signals better than the learned transforms despite an inferior cost function design. We attribute this advantage to the task-based nature of the bilevel learning method.

Finally, counter to our hypothesis that the task-based nature of the bilevel training method would ensure that the learned filters outperformed the handcrafted filter, Tab. 11.1 shows that the filters learned using bilevel methods do not denoise better than $\boldsymbol{c}_{\text{FD}}$. Specifically, the test RMSEs using the filters learned from 100 different initializations are 7-24% higher than the RMSE using $\boldsymbol{c}_{\text{FD}}$. In comparison, the learned transform with $\lambda = 0.23$ resulted in denoised signals with 38% more error than the finite differencing transform. The strong denoising ability of $\boldsymbol{c}_{\text{FD}}$ stems from the simple, piece-wise constant structure of the test signals.

11.3 Conclusion

We started the investigation in Section 9.2 based on an observation that we did not learn $\boldsymbol{T}_{\text{FD}}$ using the split transform learning training cost with noiseless PWC training signals. From this observation, we asked: Why do handcrafted sparsifying filters sometimes outperform learned filters? By construction, the learned transform achieves a lower training loss (9.3) than $\boldsymbol{T}_{\mathrm{FD}}$. Though this might suggest that the learned transform is "better," in fact the handcrafted transform better denoises the test signals. The disparity is due to the structure of the training cost: the transform is learned to make training data approximately match sparse codes, which is best accomplished by a smooth transform, rather than to separate signal and noise for denoising. Section 9.2 speficially showed that the smoothness in the learned transform results from splitting the cost function as in (9.3), and that the smoothness increases as the tuning parameter increases.

The observation that handcrafted filters can outperform learned filters due to the mismatch of the training cost and test criteria naturally leads to the task-based bilevel formulation. This chapter addressed RQ#5: How does the bilevel method compare to handcrafted filters and filters learned in a non-task-based method? Although the learned filters did not denoise signals better than $\boldsymbol{c}_{\mathrm{FD}}$, our simple experimental results show the learned filters perform more similar to $\boldsymbol{c}_{\mathrm{FD}}$ than the learned transforms from Section 9.2 that perform noticeably worse than $\boldsymbol{T}_{\mathrm{FD}}$. Further, the results in Section 11.2.4 suggest that the learned filters using the bilevel method would continue to be better at denoising if we ran additional upper-level iterations. Also, the learned task-based filters denoise better than the transforms learned to (approximately) sparsify training signals, despite the relaxation from the 0-norm in the transform learning problem to the corner-rounded 1-norm in the bilevel problem.

The results from Chapter 9 and this chapter exemplify the benefit of the task-based approach for simple experiments where we specifically designed the training and test signals so that we expected $\boldsymbol{T}_{\mathrm{FD}}$ to denoise very well. In more complicated problems, there is typically no obvious minimizer and one cannot hand-craft ideal filters. Thus, the benefit of task-based learning will likely be amplified in real-world settings.

CHAPTER 12

RQ#6: Survey of Applications and Connections

Along with Chapter 10, this chapter addresses RQ#6: What are the current trends in the literature on bilevel methods for image reconstruction? Section 12.1 surveys applications in the bilevel methods literature and Section 12.2 connects the bilevel literature and other machine learning techniques.

This material in this chapter is presented in chapters 6 and 7 of [11]:

> C. Crockett and J. A. Fessler, "Bilevel methods for image reconstruction," *Foundations and Trends® in Signal Processing*, vol. 15, no. 2-3, pp. 121–289, May 5, 2022, ISSN: 1932-8346, 1932-8354.
> DOI: 10.1561/2000000111

12.1 Survey of Applications

Bilevel methods have been used in many image reconstruction applications, including 1D signal denoising [165], image denoising (see following sections), compressed sensing [166], spectral CT image reconstruction [290], and MRI image reconstruction [166]. Bilevel methods are also used for classification problems. For example, [Sec. 6]nowozin:2011:structuredlearningprediction shows how the structured support vector machine (SSVM) is a convex surrogate for the bilevel model when the lower-level cost is linear in γ. This section discusses trends and highlights specific applications to provide concrete examples of bilevel methods for image reconstruction.

Many papers present or analyze bilevel optimization methods for general upper-level loss functions and lower-level cost functions, under some set of assumptions about each level. Chapter 10 summarized many of these methods. Although there are cases when the choice of a loss function and/or cost impacts the optimization strategy, many bilevel problems could use any optimization method. Thus, this section concentrates on the specific applications, rather than methodology.

This section is split into a discussion of lower-level cost and upper-level loss functions. (Lower-level cost functions that involve CNNs are discussed separately; see Section 12.2.1.) The conclusion section discusses examples where the loss function is tightly connected to the cost function.

12.1.1 Lower-level Cost Function Design

Once a bilevel problem is optimized to find $\hat{\gamma}$, the learned parameters are typically deployed in the same lower-level problem as used during training but with new, testing data. Thus, it is the lower-level cost function that specifies the application of the bilevel problem, *e.g.*, CT image reconstruction or image deblurring.

Denoising applications consider the case where the forward model is an identity operator ($\boldsymbol{A} = \boldsymbol{I}$). This case has the simplest possible data-fit term in the cost function and requires the least amount of computation when computing gradients or evaluating Φ. Because bilevel methods are generally already computationally expensive, it is unsurprising that many papers focus on denoising, even if only as a starting point towards applying the proposed bilevel method to other applications.

More general image reconstruction problems consider non-identity forward models. Few papers learn parameters for image reconstruction in the fully task-based manner described in (UL), likely due to the additional computational cost. Some papers, *e.g.*, [161], [162], [241] consider learning parameters for denoising, and then apply $\hat{\boldsymbol{\gamma}}$ in a reconstruction problem with the same regularizer but introducing the new $\boldsymbol{A}$ to the data-fit term. These "crossover experiments" [241] test the generalizability of the learned parameters, but they sacrifice the specific task-based nature of the bilevel method.

Recall from Chapter 8.1 that the regularizer (with its learned parameters) can be related to a prior for $\boldsymbol{x}$ in a maximum *a posteriori* probability perspective. If this perspective is valid, then the $\hat{\boldsymbol{\gamma}}$ should generalize to other system matrices. However, the exact connection between the regularizer and the probability distribution is not straight-forward [308] and previous results suggest that $\hat{\boldsymbol{\gamma}}$ varies with different $\boldsymbol{A}$'s [157], [241]. Further, $\boldsymbol{A}$ is often an imperfect model for the true underlying phenomena and $\hat{\boldsymbol{\gamma}}$ may end up compensating for modeling errors that are specific to a given $\boldsymbol{A}$, and thus may not generalize to other imaging system models.

Many bilevel methods, especially in image denoising [162], [163], [165], [192], [293], but also in image reconstruction [164], use the same or a very similar lower-level cost as the running example in this review. From Section 7.3, the running example cost function is:

$$\hat{\boldsymbol{x}}(\boldsymbol{\gamma}, \boldsymbol{y}) = \underset{\boldsymbol{x}}{\operatorname{argmin}} \overbrace{\frac{1}{2} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2 + \underbrace{e^{\beta_0} \sum_{k=1}^{K} e^{\beta_k} \mathbf{1}' \phi(\boldsymbol{c}_k \circledast \boldsymbol{x}; \epsilon)}_{R(\boldsymbol{x}\,;\boldsymbol{\gamma})}}^{\Phi(\boldsymbol{x}\,;\boldsymbol{\gamma})} . \tag{12.1}$$

The learned hyperparameters, $\boldsymbol{\gamma}$, include the tuning parameters, β_k and/or the filter coefficients, $\boldsymbol{c}_k$. The image reconstruction example in [164] generalized (12.1) for implicitly defined forward models by using a different data-fit term, as given in (10.7). Their two example problems involve learning parameters to estimate the diffusion coefficient or forcing function in a second-order elliptic partial differential equation.

Two common variations among applications using (12.1) are (1) the choice of which tuning parameters to learn and (2) what sparsifying function, ϕ, to use. Some methods [164], [192], [293] learn only the tuning parameters; these methods typically use finite differencing filters or discrete cosine transform (DCT) filters (excluding the DC filter) as the $\boldsymbol{c}_k$'s. Other methods learn only filter coefficients [165]. Fig. 12.1 shows filters learned from patches of the "cameraman" image when $\boldsymbol{\gamma} = (\boldsymbol{\beta}, \boldsymbol{h})$ and shows filter strengths when $\boldsymbol{\gamma} = \boldsymbol{\beta}$. The corresponding bilevel problem is (Ex) with ϕ given in (CR1N). Fig. 12.2 shows the corresponding denoised image and Appendix F describes the experiment settings and additional results.

A slight variation on learning the filters is to learn coefficients for a linear combination of filter basis elements [162], [163], *i.e.*, learning $a_{k,i}$ where

$$\boldsymbol{c}_k = \sum_i a_{k,i} \boldsymbol{b}_i,$$

for some set of basis filter elements, $\boldsymbol{b}_i$. One benefit of imposing a filter basis is the ability to ensure the filters lie in a given subspace. For example, [162], [163] use the DCT as a basis and remove the constant filter so that all learned filters are guaranteed to have zero-mean.

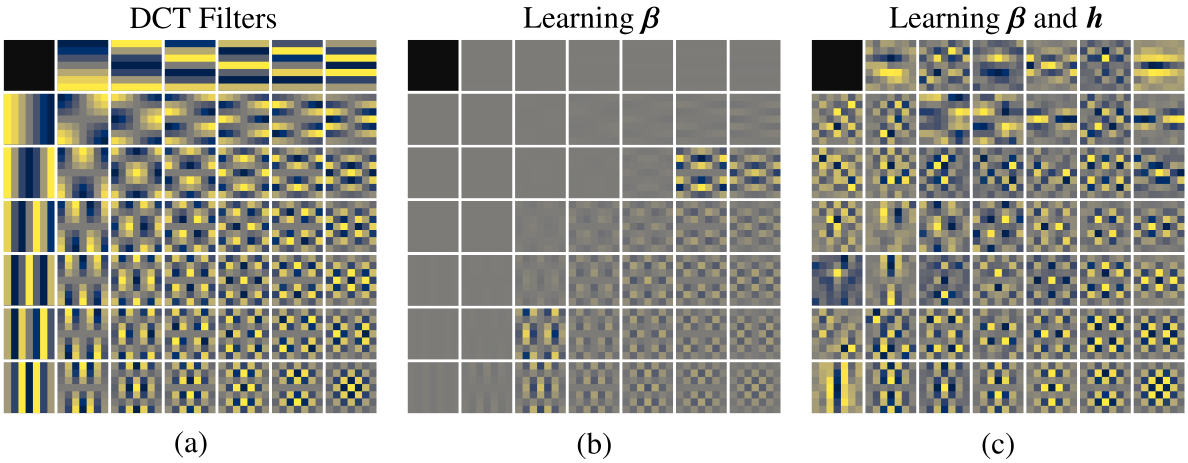

DCT Filters
Learning $\boldsymbol{\beta}$
Learning $\boldsymbol{\beta}$ and $\boldsymbol{h}$
(a)
(b)
(c)

Figure 12.1: The DCT filter bank and example learned filters for (Ex) with training data from the "cameraman" image. (a) The 48 non-constant 7×7 DCT filters used to initialize $\boldsymbol{\gamma}$. The dark, top-left square represents the removed DC filter. (b) The DCT filters multiplied by their respective tuning parameter β_k when $\boldsymbol{\gamma} = \boldsymbol{\beta}$. The range of $e^{\beta_0+\beta_k}$ is 0.001-1.08. The learned tuning parameters emphasize the higher-frequency DCT filters. (c) Learned filters when $\boldsymbol{\gamma} = (\boldsymbol{\beta}, \boldsymbol{h})$ (scaled to have unit-norm for visualization).

In terms of sparsifying functions, [165], [293] used the same corner rounded 1-norm as in (CR1N), [163] used $\phi = \log\left(1 + z^2\right)$ to relate their method to the Field of Experts framework [187], [164] used a quadratic penalty, and [162], [192] both consider multiple ϕ options to examine the impact of non-convexity in ϕ. Ref. [192] compared p-norms, $\|\boldsymbol{c}_k \circledast \boldsymbol{x}\|_p^p$, for $p \in \{\frac{1}{2}, 1, 2\}$, where the $p = \frac{1}{2}$ and $p = 1$ cases are corner-rounded to ensure ϕ is smooth. (The $p = \frac{1}{2}$ case is non-convex.) Ref. [162] compared the convex corner-rounded 1-norm in (CR1N) with two non-convex choices: the log-sum penalty $\log\left(1 + z^2\right)$, and the Student-t function $\log\left(10\epsilon + \sqrt{z^2 + \epsilon^2}\right)$.

Both [162], [192] found that non-convex penalty functions led to denoised images with better (higher) PSNR. They hypothesize that the improvement is due to the non-convex penalty functions better matching the heavy-tailed distributions in natural images. As further evidence of the importance of non-convexity, [162] found that untrained 7×7 DCT filters (excluding the constant filter) with learned tuning parameters and a non-convex ϕ outperformed learned filter coefficients with a convex ϕ, despite the increased data adaptability when learning filter coefficients. The trade-off for using non-convex penalty functions is the possibility of local minimizers of the lower-level cost.

Chen, Ranftl, and Pock [162] also investigated how the number of learned filters and the size of the filters impacted denoising PSNR. They concluded that increasing the number of filters to achieve an over-complete filter set may not be worth the increased computational expense and that increasing the filter size past 11×11 is unlikely to improve PSNR. Using 48 filters of size 7×7 and the log-sum penalty function, [162] achieved denoising results on natural images comparable to algorithms such as BM3D [309], as seen in Fig. 12.3. Although results will vary between applications and training data sets, the results from [162] provide motivation for filter learning and an initial guide for designing bilevel methods.

In addition to variations on the running example for Φ (12.1), a common regularizer for the lower-level cost is Total Generalized Variation with order 2 (TGV^2) [310]. Whereas TV encourages images to be piece-wise constant, TGV^2 is a generalization of TV designed for piece-wise linear images. Another generalization of TV for piece-wise linear images is Infimal Convolutional Total Variation (ICTV) [311]. Bilevel papers that investigate ICTV include [143], [144]; these papers also investigate TGV^2. See [312] for a comparison of the two.

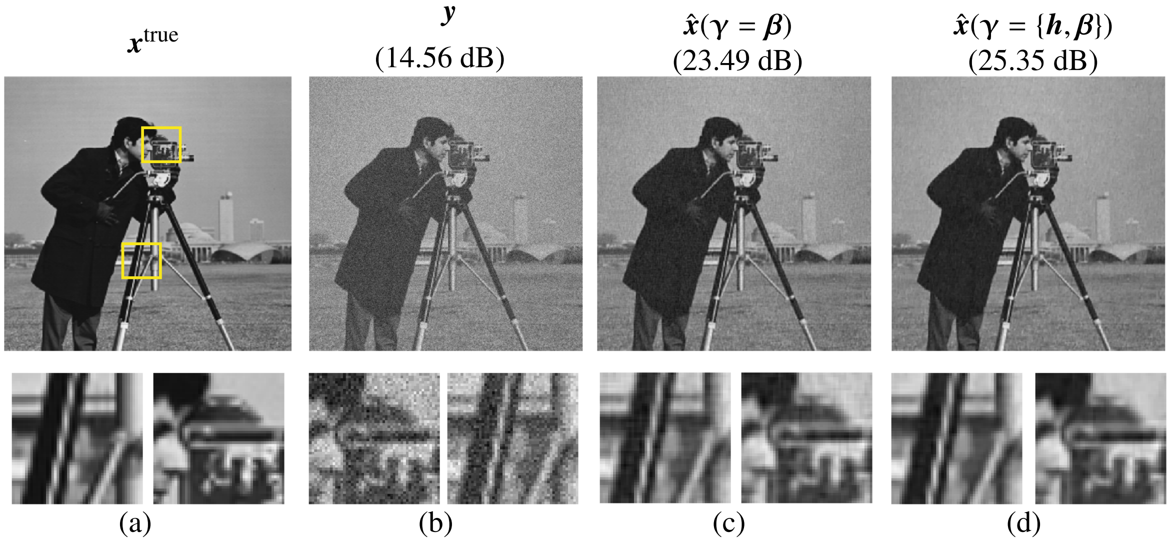

$\boldsymbol{x}^{\text{true}}$
$\boldsymbol{y}$
(14.56 dB)
$\hat{\boldsymbol{x}}(\boldsymbol{\gamma} = \boldsymbol{\beta})$
(23.49 dB)
$\hat{\boldsymbol{x}}(\boldsymbol{\gamma} = \{\boldsymbol{h}, \boldsymbol{\beta}\})$
(25.35 dB)
(a)
(b)
(c)
(d)

Figure 12.2: Example denoising results for the full "cameraman" test image and two of the training patches. (a) Noiseless training "cameraman" test image. (b) Noisy image and its SNR. (c) Denoised image using the learned tuning parameters that weight the DCT filters as shown in Fig. 12.1b. (d) Denoised image using the learned filter coefficients and tuning parameters as shown in Fig. 12.1c. For comparison, the denoised image using BM3D [309] has a SNR of 26.87. See Appendix F for more details.

TGV cost functions are typically expressed in the continuous domain, at least initially, but then discretized for implementation, *e.g.*, [313], [314]. One discrete approximation of the TGV^2 regularizer is:

$$R_{\text{TGV}}(\boldsymbol{x}) = \min_{\boldsymbol{z}} e^{\beta_1} \|\boldsymbol{c}_{\text{FD}} \circledast \boldsymbol{x} - \boldsymbol{z}\|_1 + e^{\beta_2} \|\partial \boldsymbol{z}\|_1 ,$$

where $\boldsymbol{c}_{\text{FD}}$ is a filter that takes finite differences and ∂ is a filter that approximates a symmetrized gradient. In TV, one usually thinks of $\boldsymbol{z}$ as a sparse vector; here $\boldsymbol{z}$ is a vector whose finite differences are sparse, so $\boldsymbol{z}$ is approximately piece-wise constant. Encouraging $\boldsymbol{z}$ to be piece-wise constant in turn makes $\boldsymbol{x}$ approximately piece-wise linear, since $\boldsymbol{c}_{\text{FD}} \circledast \boldsymbol{x} \approx \boldsymbol{z}$ from the first term. Bilevel methods for learning β_1 and β_2 for the TGV^2 regularizer include [143], [144]. An extension to the TGV^2 regularizer model is to learn a space-varying tuning parameter [289].

As an example of how the regularizer should be chosen based on the application, [289] found that standard TV with a learned tuning parameter performed best (in terms of SSIM) for approximately piece-wise constant images while TGV^2 with learned tuning parameters performed best for approximately piece-wise linear images.

12.1.2 Upper-Level Loss Function Design

From some of the earliest bilevel methods, *e.g.*, [158], [165], to some of the most recent bilevel methods, *e.g.*, [161], [279], square error or mean squared error (MSE) remains the most common upper-level loss function. In the unsupervised setting, [223], [224] used SURE (an estimate of the MSE, see Section 8.3) as the upper-level loss function. Unlike many perceptually motivated image quality measures, MSE is convex in $\boldsymbol{x}$ and it is easy to find $\nabla_{\boldsymbol{x}} \ell(\boldsymbol{\gamma}\,; \hat{\boldsymbol{x}}(\boldsymbol{\gamma}))$. However, MSE does not capture perceptual quality nor image utility (see Section 8.3). This section discusses a few bilevel methods that used different loss functions.

Ref. [144] compared a squared error upper-level loss function with a Huber (corner rounded 1-norm) loss func-

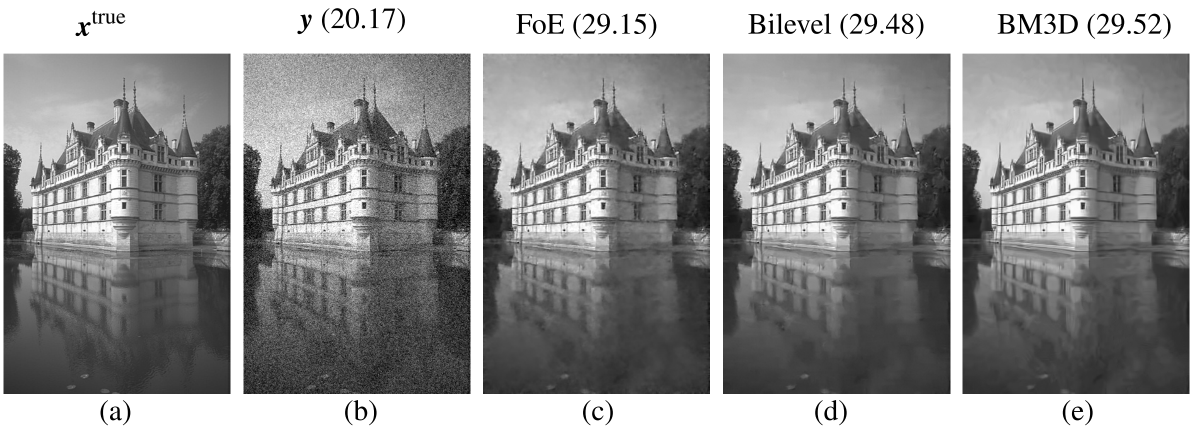


Figure 12.3: Example denoising results from [162] comparing filters learned using bilevel methods to other denoising methods. (a) The original image $\boldsymbol{x}^{\text{true}}$. (b) The noisy image $\boldsymbol{y}$. (c-d) Denoised images using FoE [187], BM3D [309], and a bilevel approach using a set-up equivalent to (Ex) with a non-convex penalty function, $\phi(z) = \log\left(1 + z^2\right)$ [162]. The PSNR values in dB are given in parenthesis. ©2014 IEEE. Reprinted, with permission, from [162].

tion. The corresponding lower-level problem was a denoising problem with a standard 2-norm data-fit term and three different options for a regularizer: TV, TGV^2, and ICTV. The authors learned tuning parameters for a natural image dataset using both upper-level loss function options for each of the lower-level regularizers.

Since SNR is equivalent to MSE, the MSE loss will always perform the best according to any SNR-based metric (assuming the bilevel model is well-trained). However, [144] found the tuning parameters learned using the Huber loss yielded denoised images with better qualitative properties and better SSIM, especially at low noise levels. Like MSE, the Huber loss operates point-wise and is easy to differentiate. Thus, the authors conclude that the Huber loss is a good trade-off between tractability and improving on MSE as an image quality measure.

A set of loss functions in [289], [290], [293] consider the unsupervised or "blind" bilevel setting, where one wishes to reconstruct an image without clean samples. Therefore, rather than using an image quality metric that compares a reconstructed image, $\hat{\boldsymbol{x}}$, to some true image, $\boldsymbol{x}^{\text{true}}$, these loss function consider the estimated residual,

$$\hat{\boldsymbol{n}} = \hat{\boldsymbol{n}}(\boldsymbol{\gamma}) = \boldsymbol{A}\hat{\boldsymbol{x}}(\boldsymbol{\gamma}) - \boldsymbol{y},$$

where $\boldsymbol{\gamma}$ is learned using only noisy data. Unsupervised bilevel methods may be beneficial when there is no clean data and one has more knowledge of noise properties than of expected image content. All three methods [289], [290], [293] assume the noise variance, σ^2, is known.

The earliest example [293], learned tuning parameters $\boldsymbol{\gamma}$ such that $\hat{\boldsymbol{n}}$ matched the second moment of the assumed Gaussian distribution for the noise. Their lower-level cost is comparable to (Ex), but re-written in terms of $\boldsymbol{n}$ and with pre-defined finite differencing or 5 × 5 DCT filters, *i.e.*, they learn only the tuning parameters, β_k. Their upper-level

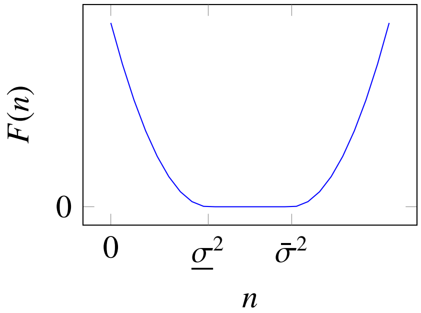


Figure 12.4: Noise corridor function (12.2) used as part of the upper-level loss function for the unsupervised bilevel method in [289].

loss encourages the empirical variances of the noise in different frequency bands to match the expected variances:

$$\ell(\boldsymbol{\gamma}\,;\boldsymbol{n}(\boldsymbol{\gamma})) = \frac{1}{2}\sum_i \frac{\left(\left\|\boldsymbol{f}_i \circledast \boldsymbol{n}\right\|_2^2 - \mu_i\right)^2}{v_i}$$

$$\mu_i = \mathbb{E}\left[\left\|\boldsymbol{f}_i \circledast \boldsymbol{n}\right\|_2^2\right] \text{ and } v_i = \mathrm{Var}\left[\left\|\boldsymbol{f}_i \circledast \boldsymbol{n}\right\|_2^2\right],$$

where $\boldsymbol{f}_i$ are predetermined filters that select specific frequency components. By using bandpass filters that partition Fourier space, the corresponding means and variances of the second moments of the filtered noise are easily computed, with

$$\mu_i = N\sigma^2\left\|\boldsymbol{f}_i\right\|^2 \quad \text{and} \quad v_i = N\sigma^4\left\|\boldsymbol{f}_i\right\|^4.$$

Although the experimental results are promising, [293] does not claim state-of-the-art results since their lower-level denoiser is relatively simple.

As an alternative to the Gaussian-inspired approach in [293], [289] and [290] use loss functions that penalize noise outside a set "noise corridor." Both methods learn space-varying tuning parameters, and the upper-level loss consists of a data-fit term (that measures noise properties) and a regularizer on $\boldsymbol{\gamma}$. The data-fit term in the upper-level loss function in [293] defines the noise corridor between a maximum variance, $\bar{\sigma}^2$, and a minimum variance, $\underline{\sigma}^2$:

$$\begin{aligned} &\mathbf{1}' F.\left(\boldsymbol{w} \odot (\boldsymbol{n}(\boldsymbol{\gamma}) \odot \boldsymbol{n}(\boldsymbol{\gamma}))\right) \text{ for} \\ &F(n) = \frac{1}{2}\max(n - \bar{\sigma}^2, 0)^2 + \frac{1}{2}\min(n - \underline{\sigma}^2, 0)^2, \end{aligned} \tag{12.2}$$

where $\boldsymbol{w}$ is a predetermined weighting vector. The noise corridor function, $F(n)$, penalizes any noise outside of the expected range as shown in Fig. 12.4. Ref. [290] uses the same noise corridor function, but extends the bilevel method for images with Poisson noise; [290] thus estimates the noisy image using the Kullback-Leibler distance. In addition to the noise corridor function as the data-fit component of the upper-level loss function, [289], [290] include a smoothness-promoting regularizer on $\boldsymbol{\gamma}$, which is a spatially varying tuning parameter vector in both methods.

The task-based nature of bilevel typically makes regularizers or constraints on $\boldsymbol{\gamma}$ unnecessary (see Section 9.1 for common options for other forms of learning). However, there are two general cases where a regularizer on $\boldsymbol{\gamma}$ is useful in the upper-level loss function. First, a regularizer can help avoid over-fitting when the amount of training data is insufficient for the number of learnable hyperparameters. This is often the case when learning space-varying parameters that have similar dimensions as the input data, *e.g.*, [158], [289], [290], [315]. In such cases, the regularization often takes the form of a 2-norm on the learned hyperparameters, $\|\boldsymbol{\gamma}\|_2^2$.

Second, some problems require application-specific constraints, *e.g.*, [241] incorporates constraints in the upper-level loss to ensure that the learned parameters are valid interpolation kernels. Many other hyperparameter constraints do not require a regularization term, For example, non-negativity constraints on tuning parameters are easily handled by redefining the tuning parameter in terms of an exponential, as in (Ex), and box constraints are common and easy to incorporate with a projection step if using a gradient-based method. Constraints that require sparsity on the learned parameters may benefit from regularization in the upper-level loss function.

An example of an application-specific constraint is found in [159], [160], which consider MRI reconstruction with a data-fit term and a variational regularizer. Both papers extend the bilevel model in (Ex) to include part of the forward model in the learnable parameters, $\boldsymbol{\gamma}$. Specifically, [159], [160] learned the sparse sampling matrix for MRI. (Ref. [160] additionally learns tuning parameters for predetermined filters, whereas [159] sets the tuning parameters and filters and learns only the sampling matrix.) Here, the forward model is

$$\boldsymbol{A} = \operatorname{diag}(\underbrace{s_1, s_2, \ldots, s_M}_{\boldsymbol{s}(\boldsymbol{\gamma})})\boldsymbol{F},$$

where $\boldsymbol{F}$ is the DFT matrix and s_i are learned binary values that specify whether a frequency location should be sampled.

The motivation for learning a sparse sampling matrix comes from the lower-level MRI reconstruction problem; designing more effective sparse sampling patterns in MRI can decrease scan time and thus improve patient experience, decrease cost, and decrease artifacts from patient movement. This goal requires the learned parameters, s_i, to be binary, which in turn influences the upper-level loss function design. Thus, [159], [160] include regularization in the upper-level to encourage $\boldsymbol{s}$ to be sparse, *e.g.*, [160] uses an upper-level loss with a squared error term and regularizer on $\boldsymbol{s}$:

$$\ell(\boldsymbol{\gamma}\,;\hat{\boldsymbol{x}}(\boldsymbol{\gamma})) = \|\hat{\boldsymbol{x}}(\boldsymbol{\gamma}) - \boldsymbol{x}^{\text{true}}\|_2^2 + \lambda \sum_i (s_i + s_i(1 - s_i)), \tag{12.3}$$

where λ is a upper-level tuning parameter that one must set manually. (In experiments, they thresholded the learned s_i values to be exactly binary.) An alternative approach is to constrain the number of samples [316], though that formulation requires other optimization methods.

12.1.3 Conclusion

This section split the discussion of lower-level cost and upper-level loss functions to discuss trends in both areas. However, when designing a bilevel problem, design decisions can impact both levels. For example, the unsupervised nature of [290], [293] clearly impacted their choice of upper-level loss function to use noise statistics rather than squared error calculated with ground-truth data. Since it can be challenging to learn many good parameters from noisy training data, the unsupervised nature also likely impacted the authors' decision to learn only tuning parameters and set the filters manually. Another example of coupling between lower-level and upper-level design is when one enforces application-specific constraints on the learned parameters, *e.g.*, using a regularizer like (12.3) in the upper-level loss to promote sparsity of the MRI sampling matrix [159], [160].

In addition to design decisions influencing both levels, bilevel methods may adopt common techniques for the upper-level loss function and lower-level cost function. For example, a common theme is the tendency to use smooth functions, such as replacing the 1-norm with a corner-rounded 1-norm. This approach requires setting a smoothing

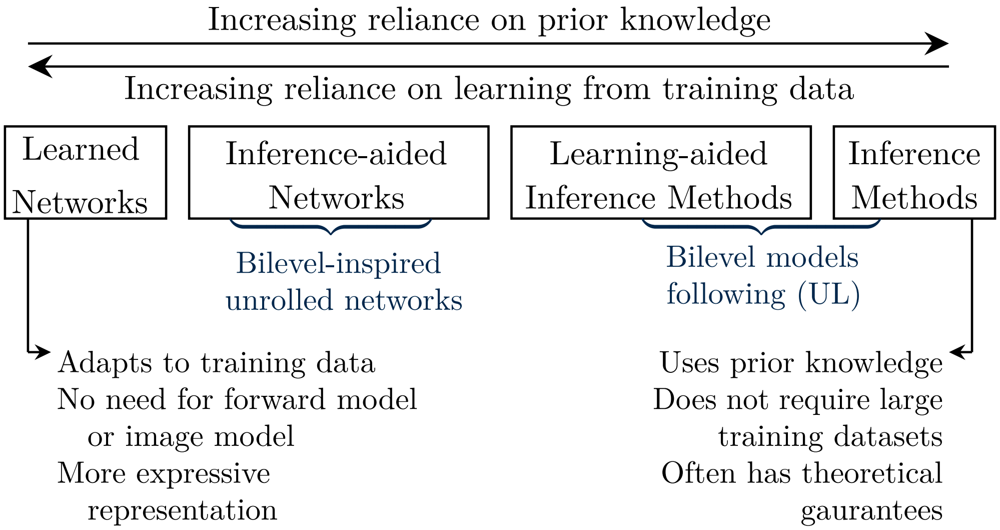

Increasing reliance on prior knowledge
Increasing reliance on learning from training data
Learned Networks
Inference-aided Networks
Learning-aided Inference Methods
Inference Methods
Bilevel-inspired unrolled networks
Bilevel models following (UL)
Adapts to training data
No need for forward model or image model
More expressive representation
Uses prior knowledge
Does not require large training datasets
Often has theoretical gaurantees

Figure 12.5: Spectrum of learning to inference-based methods from [317].

parameter, *e.g.*, ϵ in (CR1N), which in turn impacts the Lipschitz constant and optimization speed. More accurate approximations generally lead to larger Lipschitz constants and slower convergence. One approach to trading-off the accuracy of the smoothing with optimization speed is to use a graduated approach and approximate the non-smooth term more and more closely as the optimization progresses [166].

The prevalence of smoothing is unsurprising considering that this review focuses on gradient-based bilevel methods. Rare exceptions include [271], [272], which used the (not corner-rounded) one-norm to define ϕ to learn convolutional filters using the translation to a single level approach described in Section 10.1.2. The impact of smoothing and how accurately one should approximate a non-differentiable point remains an open question.

From an image quality perspective, ideally one would independently design the lower-level cost function and upper-level training loss. The lower-level cost would depend on the imaging physics and would incorporate regularizers that expected to provide excellent image quality when tuned appropriately, and the upper-level loss would use terms that are meaningful for the imaging tasks of interest. As we have seen, in practice one often makes compromises to facilitate optimization and reduce computation time.

12.2 Connections

This section uses the spectrum of learning methods from Shlezinger *et al.* [317] as a framework for comparing bilevel methods to other learning-based approaches that combine inferences or prior knowledge[1] and deep learning. Inferences can include information about the structure of the forward model, $\boldsymbol{A}$, or about the object $\boldsymbol{x}$ being imaged. For example, any known statistical properties of the object of interest could be used to design a regularizer that encourages the minimizer $\hat{\boldsymbol{x}}$ to be compatible with that prior information. At one extreme, inference-based approaches rely on a relatively small number of handcrafted regularizers with a few, if any, tuning parameters learned from training data. At the other extreme, fully learned approaches assume no information about the application or data and learn all hyperparameters from training data. Fig. 12.5 depicts the spectrum [317].

[1] Ref. [317] uses the term "model-based", but this review uses "inferences" to differentiate from other definitions of model-based learning in the literature.

Ref. [317] proposed two general categories for methods that mix elements of inference-based and learning-based methods. The first category, inference-aided networks, includes deep neural networks (DNNs) with architectures based on an inference-based method. For example, in deep unrolling, one starts with a fixed number of iterations of an optimization algorithm derived from a cost function and then learns parameters that may vary between iterations, or "layers," or may be shared across such iterations. Section 12.2.1 further discusses unrolling, which is a common inference-aided network design strategy, and the connection to the bilevel unrolling method described in Section 10.1.3.

The second general category is DNN-aided inference methods [317]. These methods incorporate a deep learning component into traditional inference-based techniques (typically a cost function in image reconstruction). The learned DNN component(s) can be trained separately for each iteration or end-to-end. Because prior knowledge takes a larger role than in the inference-aided networks, these methods typically require smaller training datasets, with the amount of training data required varying with the number of hyperparameters. Section 12.2.3 discusses how bilevel methods compare to Plug-and-Play, which is an example DNN-aided inference model.

While [317] focused on DNNs due to their highly expressive nature and the abundance of interest in them, the idea of trading off prior knowledge and learning components applies to machine learning more broadly. Section 12.2.1 through 12.2.3 describe how bilevel methods fit into the framework from [317] and relates bilevel methods to other methods in the framework.

12.2.1 Learnable Optimization Algorithms

Learning parameters in unrolled optimization algorithms to create an inference-aided network, often called a Learnable Optimization Algorithm (LOA), is a quickly growing area of research [318]. The first such instance was a learned version of the Iterative Shrinkage and Thresholding Algorithm (ISTA), called LISTA [319]. Similar to the bilevel unrolling method, a LOA typically starts from a traditional, inference-based optimization algorithm, unrolls multiple iterations, and then learns parameters using end-to-end training.

There are many unrolled methods for image reconstruction [318]. Two examples that explicitly state the bilevel connection are [166], [320]; both set-up a bilevel problem with a DNN as a regularizer and then allow the parameters to vary by iteration, *i.e.*, learning $\boldsymbol{c}_k^{(t)}$ where t denotes the lower-level iteration. Ref. [320] motivated the use of an unrolled DNN over more inference-based methods by the lack of an accurate forward model, specifically coil sensitivity maps, for MRI reconstruction. Other examples of unrolled networks are [321], which unrolls the Field of Experts model [187] (see Section 8.1.3 and 12.1.1 for how the Field of Experts model has inspired many bilevel methods); [322], which unrolls the convolutional analysis operator model [193] (see (8.12)); and [288], which discusses the connection to meta-learning.

Unlike the unrolled approach to bilevel learning described in Section 10.1.3, many LOAs depart from their base cost function and "only superficially resemble the steps of optimization algorithms" [166]. For example, unrolled algorithms may "untie" the gradient from the original cost function, *e.g.*, using $\widetilde{\boldsymbol{A}}'(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y})$, instead of $\boldsymbol{A}'(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y})$ for the gradient of the common 2-norm data-fit term, where $\tilde{\boldsymbol{A}}'$ is learned or otherwise differs from the adjoint of $\boldsymbol{A}$. LOAs that allow the learned parameters to vary every unrolled iteration or learn step size and momentum parameters further depart from a cost function perspective.

In addition to selecting which variables to learn, one must decide how many iterations to unroll for both bilevel unrolled approaches and LOAs. Most methods pick a set number of iterations in advance, perhaps based on previous experience, initial trials, or the available computational resources. Using a set number of iterations yields an algorithm with predictable run times and allows the learned parameters to adapt to the given number of iterations. Further, picking a small number of iterations can act as implicit regularization, comparable to early stopping in machine learning, which may be helpful when the amount of training data is small relative to the number of hyperparameters in the unrolled algorithm [288].

One can also use a convergence criteria to determine the number of iterations to evaluate, rather than selecting a number in advance [279]. This convergence-based method more closely follows classic inference-based optimization algorithms. A benefit of running the lower-level optimization algorithm until convergence is that one could switch optimization algorithms between training and testing, especially for strictly convex lower-level cost functions, and still expect the learned parameters to perform similarly. This ability to switch optimization algorithms means one could use faster, but not differentiable, algorithms at test-time, such as accelerated gradient descent methods with adaptive restart [295]. We are unaware of any bilevel methods that have exploited this possibility.

Even within the unrolling methodology, one must make several design decisions. To remain most closely tied to the original optimization algorithm, an unrolled method might fix a large number of iterations or run the optimization algorithm until convergence, use the same parameters every layer, and calculate the step size based on the Lipschitz constant every upper-level iteration (see discussion in Section 10.1.3.1 and 11.2.1). Like all design decisions, there are trade-offs and the literature shows many successful methods that benefit from the increased generality of designing LOAs that are further removed from their cost function roots [318]. Echoing the ideas from [317], the design should be based on the specific application and relative availability, reliability, and importance of prior knowledge and training data.

This survey focuses on unrolled methods that are closely tied to the original bilevel formulation; [318] reviews LOAs more broadly. A benefit of maintaining the connection to the original cost function and optimization algorithm is that, once trained, the lower-level problem in an unrolled bilevel method inherits any theoretical and convergence results from the corresponding optimization method. The corresponding benefit for LOAs is increased flexibility in network architecture.

12.2.2 Equilibrium-based Networks

Equilibrium-based, or fixed point, networks are related to both LOAs and the minimizer approach from Section 10.1.1. The idea was proposed only recently in [323], but has received much attention. From the unrolled perspective, equilibrium networks consider what happens when the number of unrolled iterations approaches infinity. Alternatively, they can be viewed as a single, implicit layer; as in the minimizer approach, the output is the solution to a nonlinear equation.

We first consider the unrolled perspective. If an algorithm Ψ is a contraction, *i.e.*,

$$||\Psi(\boldsymbol{x}_1\,;\,\boldsymbol{\gamma}) - \Psi(\boldsymbol{x}_2\,;\,\boldsymbol{\gamma})|| \leq \delta\,||\boldsymbol{x}_1 - \boldsymbol{x}_2||\,,\ \forall \boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathbb{F}^N$$

for some parameter $\delta \in [0, 1)$, then the sequence of iterates will eventually converge to a fixed-point of Ψ. If the optimization algorithm optimizes a cost function with a data-fit and regularization term, then the equilibrium network approach is equivalent to a bilevel method. For a given value of $\boldsymbol{\gamma}$, the contraction condition is typically easy to satisfy by selecting an appropriate step-size in algorithms like gradient descent. Ref. [278] provides conditions on deep equilibrium models specific to optimization algorithms based on gradient descent, proximal gradient descent, and ADMM that ensure convergence.

Re-using some of our bilevel notation, let $\hat{\boldsymbol{x}}(\boldsymbol{\gamma})$ denote a fixed-point of an equilibrium network. The derivation for finding $\nabla_{\boldsymbol{\gamma}} \hat{\boldsymbol{x}}(\boldsymbol{\gamma}) \in \mathbb{F}^{N \times R}$ follows similar steps to the IFT perspective on the bilevel minimizer approach in Section 10.1.1.1. The key difference is that, rather than using the first-order optimally condition as in the minimizer approach (10.3), the equilibrium method considers the lower-level minimizer to be a fixed point of an optimization algorithm.

When the goal of the lower level problem is to find a fixed point, the bilevel problem becomes

$$\operatorname*{argmin}_{\boldsymbol{\gamma}} \underbrace{\ell\,(\boldsymbol{\gamma}\,;\,\hat{\boldsymbol{x}}(\boldsymbol{\gamma}))}_{\ell(\boldsymbol{\gamma})} \text{ s.t. } \underbrace{\hat{\boldsymbol{x}}(\boldsymbol{\gamma}) = \Psi(\hat{\boldsymbol{x}}(\boldsymbol{\gamma})\,;\boldsymbol{\gamma})}_{\text{Fixed point equation}}. \tag{12.4}$$

Similar to the IFT perspective, one can differentiate both sides of the fixed point equation using the chain rule

$$\nabla_{\boldsymbol{\gamma}}\hat{\boldsymbol{x}}(\boldsymbol{\gamma}) = (\nabla_{\boldsymbol{x}}\Psi(\hat{\boldsymbol{x}}(\boldsymbol{\gamma})\,;\boldsymbol{\gamma}))\,\nabla_{\boldsymbol{\gamma}}\hat{\boldsymbol{x}}(\boldsymbol{\gamma}) + \nabla_{\boldsymbol{\gamma}}\Psi(\hat{\boldsymbol{x}}(\boldsymbol{\gamma})\,;\boldsymbol{\gamma})$$

and then rearrange to derive an expression for $\nabla_{\boldsymbol{\gamma}}\hat{\boldsymbol{x}}(\boldsymbol{\gamma})$

$$\nabla_{\boldsymbol{\gamma}}\hat{\boldsymbol{x}}(\boldsymbol{\gamma}) = (\boldsymbol{I} - \underbrace{(\nabla_{\boldsymbol{x}}\Psi(\hat{\boldsymbol{x}}(\boldsymbol{\gamma})\,;\boldsymbol{\gamma})))}_{\hat{\boldsymbol{J}}}^{-1}\nabla_{\boldsymbol{\gamma}}\Psi(\hat{\boldsymbol{x}}(\boldsymbol{\gamma})\,;\boldsymbol{\gamma}). \tag{12.5}$$

The matrix $\hat{\boldsymbol{J}}$ is the Jacobian of the optimization algorithm, evaluated at the fixed point $\hat{\boldsymbol{x}}(\boldsymbol{\gamma})$.

Substituting (12.5) into the expression for the upper-level gradient (10.2) yields

$$\nabla\ell(\boldsymbol{\gamma}) = \nabla_{\boldsymbol{\gamma}}\ell(\boldsymbol{\gamma}\,;\hat{\boldsymbol{x}}(\boldsymbol{\gamma})) + \left(\nabla_{\boldsymbol{\gamma}}\Psi(\hat{\boldsymbol{x}}(\boldsymbol{\gamma})\,;\boldsymbol{\gamma})\right)'(\boldsymbol{I} - \hat{\boldsymbol{J}})^{-1}\nabla_{\boldsymbol{x}}\ell(\boldsymbol{\gamma}\,;\hat{\boldsymbol{x}}(\boldsymbol{\gamma})). \tag{12.6}$$

If the optimization is standard gradient descent, *i.e.*, $\Psi(\boldsymbol{x}\,;\boldsymbol{\gamma}) = \boldsymbol{x} - \alpha_\Phi\nabla_{\boldsymbol{x}}\Phi(\boldsymbol{x}\,;\boldsymbol{\gamma})$, then

$$\begin{gathered}\nabla_{\boldsymbol{\gamma}}\Psi(\hat{\boldsymbol{x}}(\boldsymbol{\gamma})\,;\boldsymbol{\gamma}) = \text{-}\alpha_\Phi\nabla_{\boldsymbol{x}\boldsymbol{\gamma}}\Phi(\boldsymbol{x}\,;\boldsymbol{\gamma}) \text{ and}\\ \nabla_{\boldsymbol{x}}\Psi(\hat{\boldsymbol{x}}(\boldsymbol{\gamma})\,;\boldsymbol{\gamma}) = \boldsymbol{I} - \alpha_\Phi\nabla_{\boldsymbol{x}\boldsymbol{x}}\Phi(\boldsymbol{x}\,;\boldsymbol{\gamma}).\end{gathered}$$

Substituting these expressions into (12.5) yields the gradient as derived using the IFT perspective in the minimizer approach (10.5), showing the close connection between the equilibrium and minimizer approach.

Similar to the minimizer approach, one can use any algorithm to find a fixed point $\hat{\boldsymbol{x}}(\boldsymbol{\gamma})$ of Ψ. For example, [323] used a quasi-Newton method and [278] used a standard fixed-point accelerated method. One can use any fixed point algorithm to find $\hat{\boldsymbol{x}}(\boldsymbol{\gamma})$; the algorithm used need not correspond to Ψ in (12.4). For example, Ψ could be standard gradient descent, even if one uses a more advanced algorithm to initially compute $\hat{\boldsymbol{x}}(\boldsymbol{\gamma})$. Another similarity to the minimizer approach is that the learned parameters are optimal at convergence of the lower-level problem, rather than after a fixed number of lower-level iterations. Therefore, the end-user can trade-off accuracy and compute requirements at test time, unlike in unrolled approaches where the number of iterations is pre-decided.

Although the equilibrium model is the limit as the number of unrolled iterations approaches infinity, computing $\nabla\ell(\boldsymbol{\gamma})$ does not require backpropagation nor storing any intermediate matrices. The trade-off is that (12.6) requires multiplying $(\boldsymbol{I} - \hat{\boldsymbol{J}})^{-1}$ by a vector. The remaining computations in the full upper-level gradient (12.6) are straightforward. Similar to the required Hessian inverse-vector product in the minimizer approach, one can use an iterative algorithm to approximate the matrix inverse. Ref. [278] notes that the inverse matrix-vector product

$$\boldsymbol{v} = (\boldsymbol{I} - \hat{\boldsymbol{J}})^{-1}\nabla_{\boldsymbol{x}}\ell(\boldsymbol{\gamma}\,;\hat{\boldsymbol{x}}(\boldsymbol{\gamma})),$$

is a fixed point of the equation

$$\boldsymbol{v} = \hat{\boldsymbol{J}}\boldsymbol{v} + \nabla_{\boldsymbol{x}}\ell(\boldsymbol{\gamma}\,;\hat{\boldsymbol{x}}(\boldsymbol{\gamma})).$$

Therefore, one can use any fixed-point solver to compute the matrix-vector product. Another way to decrease the computational cost of the Jacobian product is to use the method from [269]: if a quasi-Newton algorithm is used to estimate the Jacobian for the forward step of computing $\hat{\boldsymbol{x}}(\boldsymbol{\gamma})$, then one can "re-use" this estimated Jacobian to find $\nabla\ell(\boldsymbol{\gamma})$.

Fixed point networks can also be viewed from the perspective of unrolled methods. Although it is often infeasible to backpropagate through the large number of iterations required to reach a fixed point, backpropagating through the last few iterations yields a valid gradient estimate for $\nabla_{\boldsymbol{\gamma}}\hat{\boldsymbol{x}}(\boldsymbol{\gamma})$ [284]. Ref. [284] proves that this "truncated backpropagation" approach converges to a stationary point of the upper-level loss when the lower-level cost function is locally strongly convex around $\hat{\boldsymbol{x}}(\boldsymbol{\gamma})$ because the backpropagation gradient error decays exponentially with reverse depth. A similar approach is to use $\hat{\boldsymbol{x}}(\boldsymbol{\gamma})$ at every backpropagation step rather than previous iterates. Ref. [324] shows this is equivalent to approximating the matrix inverse in the minimizer approach using a Neumann series.

Recently, [325] proposed a Jacobian-free method to find $\nabla\ell(\boldsymbol{\gamma})$ that takes the approach from [284] to the extreme case: it considers unrolling a single layer. The approach in [325] is equivalent to viewing the deep equilibrium network as a single layer network where the initialization is the fixed-point, *i.e.*, using $\hat{\boldsymbol{x}}(\boldsymbol{\gamma}) = \Psi(\boldsymbol{x}^{(0)}\,;\,\boldsymbol{\gamma})$ in the unrolled method with $\boldsymbol{x}^{(0)} = \hat{\boldsymbol{x}}(\boldsymbol{\gamma})$. With this new perspective, it is easy to use existing backpropagation tools to compute the derivative through the single layer network. Assuming that the network is Lipschitz, contractive, and differentiable and that the upper-level loss is differentiable, [325] shows the Jacobian-free gradient is a descent direction for estimates of $\hat{\boldsymbol{x}}(\boldsymbol{\gamma})$ that are within some error bound of the true fixed point.

Deep equilibrium networks can be fully learned or they can incorporate physics-based models into their network architecture and move into the inference-aided networks category in Fig. 12.5. For example, [278], [326] incorporated system matrices into fixed point networks and applied them to MRI and CT image reconstruction problems.

12.2.3 Plug-and-play Priors

The Plug-and-Play (PNP) framework [327] is an example of a DNN-aided inference method. It is similar to bilevel methods in its dependence on the forward model. However, unlike bilevel methods, the PNP framework need not be connected to a specific lower-level cost function and it leverages pre-trained denoisers rather than training them for a specific task.

As a brief overview of the PNP framework, consider rewriting the generic data-fit plus regularizer optimization problem (8.1) with an auxiliary variable:

$$\hat{\boldsymbol{x}} = \operatorname*{argmin}_{\boldsymbol{x}\in\mathbb{F}^N} \underbrace{\overbrace{d(\boldsymbol{x}\,;\boldsymbol{y})}^{\text{Data-fit}} + \;\beta\, \overbrace{R(\boldsymbol{z}\,;\boldsymbol{\gamma})}^{\text{Regularizer}}}_{\Phi(\boldsymbol{x}\,;\boldsymbol{\gamma})} \qquad \text{s.t. } \boldsymbol{x} = \boldsymbol{z}. \tag{12.7}$$

Using ADMM [328] to solve this constrained optimization problem and rearranging variables yields the following iterative optimization approach for (12.7):

$$\begin{aligned}
\boldsymbol{x}^{(u+1)} &= \operatorname*{argmin}_{\boldsymbol{x}} d(\boldsymbol{x}\,;\boldsymbol{y}) + \frac{\lambda}{2}\|\boldsymbol{x} - \underbrace{(\boldsymbol{z}^{(u)} - \boldsymbol{u}^{(u)})}_{\tilde{\boldsymbol{x}}}\|_2^2 && = \operatorname{prox}_{\frac{1}{\lambda}d(\boldsymbol{x}\,;\boldsymbol{y})}(\tilde{\boldsymbol{x}}) \\
\boldsymbol{z}^{(u+1)} &= \operatorname*{argmin}_{\boldsymbol{z}} \beta R(\boldsymbol{z}\,;\boldsymbol{\gamma}) + \frac{\lambda}{2}\|\boldsymbol{z} - \underbrace{(\boldsymbol{x}^{(u)} + \boldsymbol{u}^{(u)})}_{\tilde{\boldsymbol{z}}}\|_2^2 && = \operatorname{prox}_{\frac{\beta}{\lambda}R(\boldsymbol{z}\,;\boldsymbol{\gamma})}(\tilde{\boldsymbol{z}}) \\
\boldsymbol{u}^{(u+1)} &= \boldsymbol{u}^{(u)} + (\boldsymbol{x}^{(u+1)} - \boldsymbol{z}^{(u+1)}),
\end{aligned}$$

where λ is an ADMM penalty parameter that effects the convergence rate (but not the limit, for convex problems). The first step is a proximal update for $\boldsymbol{x}$ that uses the forward model but does not depend on the regularizer. Conversely, the second step is a proximal update for the split variable $\boldsymbol{z}$ that depends on the regularizer, but is agnostic of the forward model. This step acts as a denoiser. The final step is the dual variable update and encourages $\boldsymbol{x}^{(u)} \approx \boldsymbol{z}^{(u)}$ as $u \to \infty$.

The key insight from [327] is that the above update equations separate the forward model and denoiser. Thus, one can substitute, or "plug in," a wide range of denoisers for the $\boldsymbol{z}$ update, in place of its proximal update, while keeping the data-fit update independent.

Whereas in the original ADMM approach, the parameter λ has no effect on the final image for convex cost functions, in the PNP framework that parameter does affect image quality. Thus, one could also use training data to tune the λ in a bilevel manner. Although PNP allows one to substitute a pre-trained denoiser, one could additionally tune the parameters in the denoiser. Ref. [329] provides one such example of starting from a PNP framework then learning denoising parameters and λ that vary by iteration.

A large motivation for the PNP framework is the abundance of advanced denoising methods, including ones that are not associated with an optimization problem such as BM3D [309]. However, using existing denoisers sacrifices the ability to learn parameters to work well with the specific forward model, as is done in task-based methods. As simple examples of how learned parameters may differ when $\boldsymbol{A}$ changes, [241] found that different filters worked better for image denoising versus image inpainting and [157] found that unrolled deblurring methods required more upper-level iterations than unrolled denoising methods. A more complicated example is using bilevel methods to learn some aspect of $\boldsymbol{A}$ alongside some aspect of the regularizer, *e.g.*, [160] learned a sparse sampling matrix and tuning parameter for MRI that are adaptive to the regularization for the image reconstruction problem.

12.3 Conclusion

This chapter discussed a variety of bilevel methods and how bilevel methods compare to other machine learning methods. It is meant to provide perspective, ideas, and connections to increase understanding for readers who may be more familiar with a different area of the literature. It is not meant to claim one perspective or definition is superior or to narrow the definition or application of any method.

Section 12.1 considered different upper-level and lower-level formulations for bilevel methods in the literature. Here, the variety across methods is most obvious in the choice of the upper-level loss function and lower-level cost function. Many methods consider filter learning similar to (Ex), but existing bilevel approaches are far from limited to this particular set-up. As just one example of the extent of the differences between methods, much of Part 6.3 refers to the bilevel method as a supervised method, but Section 8.3 includes examples of unsupervised bilevel methods.

The loss function and cost function should be designed based on a specific image reconstruction problem. Offering guidelines on how to approach this design problem will differ by application and is beyond the scope of this work. However, one theme we noted from the studies comparing multiple sparsifying functions is the benefit of non-convexity [162], [192]. An interesting avenue for future work is testing if there is a similar benefit to non-smooth sparsifying functions; Section 13.2 expands on this point.

Section 12.2 showed how bilevel methods compare to a variety of other machine learning methods. Some authors refer to inference-aided or learned networks as being bilevel methods. This is true if one takes a broad perspective on the definition of bilevel methods and defines any method that uses training data to tune hyperparameters based on a performance metric, *e.g.*, as in cross-validation, as a bilevel method. We took a more narrow view of bilevel methods; (UL) specifies an upper-level loss function that measures quality using the result of a model-based lower-level cost function. Thus, for example, unrolled algorithms that untie parameters between layers might be bilevel-inspired but do not fit our definition for a bilevel method.

Finally, although not covered in the framework in Fig. 12.5, a connection noted throughout Part 6.3 is the comparison between bilevel methods and "single-level" learning methods. Section 9.1 reviews single-level learning methods, where hyperparameters are learned to sparsify training data or to model the distribution of the training data. Like most bilevel methods, single-level methods learn hyperparameters in a supervised manner. However, they generally learn parameters that sparsify the training images, $\{\boldsymbol{x}_j^{\text{true}}\}$, and do not use the noisy data, $\{\boldsymbol{y}_j\}$. Chapter 11 demonstrates the benefit of task-based approaches over the single-level methods from Chapter 9 for a simple experiment.

CHAPTER 13

Part II: Summary, Contributions, Conclusions, and Future Work

Part 6.3 of this dissertation focused on bilevel methods for image reconstruction. The three research questions were:

RQ#4 Why do handcrafted sparsifying filters sometimes outperform learned filters?

RQ#5 How does the bilevel method compare to handcrafted filters and filters learned in a non-task-based method?

RQ#6 What are the current trends in the literature on bilevel methods for image reconstruction?

Chapter 9 and Chapter 11 addressed the first two questions using a series of case studies. For RQ#6, Chapter 10 reviewed bilevel methods and Chapter 12 discussed applications of bilevel methods and how bilevel methods compare to other machine learning approaches. This conclusion reviews the main contributions of Part 6.3 and summarizes ideas for future directions. We end Part 6.3 with a summary of the advantages and disadvantages of bilevel methods.

13.1 Summary of Contributions

The main contribution of Part 6.3 is motivating bilevel methods. At the start of my graduate studies, I was working on extending models such as those presented in Chapter 9 where the training objective was to sparsify a set of training signals. The initial impetus for learning about bilevel methods stemmed from concern over the constraints required in the training process. We developing the handcrafted CAOL algorithm (see Section 9.3) to avoid sparsifying the DC component, but the tight-frame filter constraint in CAOL still seemed overly restrictive.

There are many constraint and penalty options for single-level filter learning problems (see Section 9.1), but ultimately they all led us to the same question: is this the correct training objective? Chapter 9 learned filters using a common optimization method based on approximately sparsifying training signals. The results showed that these learned filters did not denoise a signal or reconstruct an image as well as handcrafted filters. One reason for the disparity is that the training task may not match the end application for the filters.

The training objective in bilevel methods considers the end application. For example, in words, the training objective might be “learn filters that, on average, perform best at denoising these example signals.” In comparison, the single-level objectives from Chapter 9 were “learn filters that sparsify these training signals.” The task-based set-up of the bilevel optimization problem clearly connected the training objective to the end application and it removed the need for hyperparameter constraints! Chapter 11 demonstrated how filters learned using a bilevel method denoised signals better than those learned using the single-level training objective.

Having discovered the benefit of bilevel methods, we set out to design a bilevel image reconstruction method. In many ways, bilevel methods serve as a bridge from the (traditional) single-level training objectives and the machine learning literature on convolutional and deep neural networks. Like the single-level objectives, the bilevel method has a model-based lower-level cost function that one can design to have a chosen set of theoretical guarantees. Like the neural networks, the bilevel method learns hyperparameters to minimize a loss function. While familiar with methods from both of the adjoining fields, I was not familiar with bilevel methods.

Thus, we embarked on a review of the literature that turned into a formal literature review [11]. The hope for the literature review, largely presented in Chapter 10 and 12, is that it aids other engineers who are similarly new to the bilevel literature to learn about the available tools and the existing applications. The next section shows that that are many promising avenues for future work on bilevel methods, particularly in fields such as medical image reconstruction where explainability and theoretical guarantees are highly valued, there are limited training datasets, and engineers have good models for the physics of the imaging system.

13.2 Future Directions

Throughout Part 6.3, we mentioned a few areas for future work on bilevel methods. This section highlights some of the avenues that we think are particularly promising.

Advancing upper-level loss function design is identified as future work in many bilevel papers. Despite the abundance of research on image quality metrics (see Section 8.3), most bilevel methods use squared error for the upper-level loss function (see Section 12.1.2 for exceptions). Using loss functions that better match the end-application of the images is a clear future direction for bilevel methods that nicely aligns with their task-based nature. For example, in the medical imaging field there is a large literature on objective measures of image quality [330], often based on mathematical observers designed to emulate human performance on signal detection tasks, *e.g.*, in situations where a lesion's location is unknown [331]. To our knowledge, there has been little if any work to date on using such mathematical observers to define loss functions for bilevel methods or for training CNN models, though there has been work on CNN-based observers [332]. Using task-based metrics for bilevel methods and CNN training is a natural direction for future work that could bridge the extensive literature on such metrics with the image reconstruction field.

Unsupervised bilevel problems are exceptions to the trend of using squared error for the upper-level loss function. Section 12.1.2 considered a few unsupervised bilevel methods that use noise statistics to estimate the quality of the reconstructed images, *e.g.*, [289], [290], [293] [223], [224]. One extension to the unsupervised setting is the semi-supervised setting, where one might have access to a few clean training samples and additional, noisy training samples.

A related opportunity for future work is to use bilevel methods to learn patient-adaptive parameters. The population-based learning approach considered in (7.5) learns hyperparameters that are best *on average* over the set of training images. In contrast, a patient-adaptive approach tunes hyperparameters for every input image. For example, one could learn filters and initial tuning parameters offline from a training dataset and then adjust the tuning parameters when reconstructing a specific image, *e.g.*, using approaches such as the unsupervised approaches in Section 12.1.2. An alternative approach for adapting hyperparameters at test time is to learn a mapping from the input data to the set of hyperparameters [186], [333].

Just as considering more advanced image quality metrics for the upper-level loss function is a promising area for future work, bilevel methods can likely be improved by using more advanced lower-level cost functions. For example, one could use bilevel methods to learn multi-scale filters, increasing the receptive field of a regularizer and providing a more natural representation for data that is inherently multiscale [334], [335]. Perhaps due to the already challenging and non-convex nature of bilevel problems, most methods consider relatively simple convex lower-level cost functions. Papers that examine non-convex regularizers, *e.g.*, [162], [192], conclude that non-convex regularizers lead to more accurate image reconstructions, likely due to better matching the statistics of natural images. This observation aligns with the simple denoising experimental results in [10], where learned filters with (CR1N) as the regularizer yielded noisier signals than signals denoised with a hand-crafted filter with the non-convex 0-norm regularizer. In other words, the structure of the regularizer matters in addition to how one learns the filters.

In addition to non-convexity, future bilevel methods could consider non-smooth cost functions. Many bilevel methods require the lower-level cost to be smooth. Exceptions include the translation to a single level approach (Section 10.1.2), which uses the 1-norm as the lower-level regularizer, and unrolled methods, which can be applied to non-smooth cost functions as long as the optimization algorithm has smooth updates (Section 10.1.3.2). The impact of smoothing the cost function on the perceptual quality of the reconstructed image is largely unknown.

Another avenue for future work is based on the fact that $\boldsymbol{x}^{\text{true}}$ is really a continuous-space function. A few methods, *e.g.*, [143], [144], develop bilevel methods in continuous-space. However, the majority of methods use discretized forward models without considering the impact of this simplification (as done in this dissertation). Future investigations of bilevel methods should strive to avoid the "inverse crime" [336] implicit in (7.4) where the data is synthesized using the same discretization assumed by the reconstruction method.

Future work may also consider how to more closely tie the bilevel method to a statistical modeling framework and leverage progress made in that field. Many bilevel methods for filter learning use the Field of Experts [187] as a starting point. Ref. [187] takes a maximum-likelihood perspective and learns parameters to model the training data distribution. In contrast, bilevel methods such as (Ex) have their roots in a maximum *a posteriori* perspective. While this approach is motivated by and aligns with the task-based nature of bilevel methods [163], it is not clear how well the learned parameters reflect a prior or how to use the learned parameters to generate model uncertainties. Ideas from the Bayesian statistics literature, such as Monte Carlo methods, may be a promising avenue for future research.

Related to connecting bilevel methods and statistical processes, an interesting opportunity for a stochastic bilevel formulation is to add different noise realizations in (7.4), providing an uncountable ensemble of $(\boldsymbol{x}, \boldsymbol{y})$ training tuples, where the expectation in (7.5) is over the distribution of noise realizations. Yet another possibility is to have a truly random set of training images $\boldsymbol{x}^{\text{true}}$ drawn from some distribution. For example, [337] trained a CNN-based CT reconstruction method using an ensemble of images consisting of randomly generated ellipses. Other variations, such as random rotations or warps, have also been used for data augmentation [338]. One could combine such a random ensemble of images with a random ensemble of noise realizations, in which case the expectation in (7.5) would be taken over both the image and noise distributions. We are unaware of any bilevel methods for imaging that exploit this full generality. Future literature on stochastic methods should clearly state what expectation is used and may consider exploiting a more general definition of randomness.

13.3 Summary of Advantages and Disadvantages

Like the methods described in Shlezinger, Whang, Eldar, *et al.* [317], bilevel methods for computational imaging involve mixing inference-based optimization approaches with learning-based approaches to leverage benefits of both techniques.

Inference-based approaches use prior knowledge, usually in the form of a forward model and an object model, to reconstruct images. Typically the forward model, $\boldsymbol{A}$, is under-determined, so some form of regularization based on the object model is essential. Regularizers always involve some number of adjustable parameters; traditionally inference-based methods select such parameters empirically or using basic image properties like resolution and noise [256], [259]. The regularization parameters may also be learned from training to maximize SNR [339] or detection task performance [340] in a bilevel manner (often using a grid or random search due to the relatively small number of learnable parameters). When the forward model and object model are well-known and easy to incorporate in a cost function, inference-based methods can yield accurate reconstructions without the need for large datasets of clean training data.

Learning-based approaches use training datasets to learn a prior. Recently, learning-based approaches have achieved remarkable reconstruction accuracy in practice, largely due to the increased availability in computational resources and larger, more accessible training datasets [137], [138]. However, many (deep) learning methods lack theoretical guarantees and explainability and finding sufficient training data is still challenging in many applications. Both of these challenges may impede adoption of learning-based methods in clinical practice for some applications, such as medical image reconstruction [257]. Some deep learning methods for CT image reconstruction were approved for clinical use in 2019 [341]; early studies have shown such methods can significantly reduce noise but may also compromise low-contrast spatial resolution [342].

Combining inference-based and learning-based approaches allows the integration of learning from training data while using smaller training datasets by incorporating prior knowledge. Such mixed methods often maintain interpretability from the inference-based roots while using learning to provide adaptive regularization. Thus, the benefits of bilevel methods mentioned in Chapter 6.3 introduction are generally shared among the methods described in [317]: theoretical guarantees, competitive performance in terms of reconstruction accuracy, and similar performance to learned networks with a fraction of the free parameters, *e.g.*, [161], [166].

What distinguishes bilevel methods from the other methods in the inference-based to learning-based spectrum in Fig. 12.5? While one can argue that the conventional CNN and deep learning approach is always bilevel in the sense that the hyperparameters are trained to minimize a loss function, Part 6.3 considered bilevel methods with the cost function structure (LL). The regularization term in (LL) could be based on a DNN [166], but we followed the bilevel literature that focuses on priors/regularizers, such as in (Ex), maintaining a stronger connection to traditional cost function design.

Another lens for understanding bilevel methods is extending single-level hyperparameter optimization approaches to be task-based, bilevel approaches. Single-level approaches to image reconstruction, such as those using dictionary learning [211], convolutional analysis operator learning [193], and convolutional dictionary learning [245], [343], generally aim to learn characteristics of a training dataset, with the idea that these characteristics can then be used in a prior for an image reconstruction task. While such an approach may learn more general information, [10], [271] showed that a common single-level optimization strategy resulted in learning a regularizer that was suboptimal for the simple task of signal denoising.

As further evidence of the benefit of task-based learning, [271] found that the lack of constraints in the bilevel filter learning problem is important; the learned filters used the flexibility of the model and were not orthonormal, whereas orthonormality is a constraint often imposed in single-level models (see Section 9.1). Ref. [192] showed how the task-based nature adapts to training data; total variation based regularization works well for piece-wise constant images but less so for natural images. Beyond adapting to the training dataset, bilevel methods are task-based in terms of adapting to the level of noise; [159] found the learned tuning parameters for image denoising go to 0 as the noise goes to 0, since no regularization is needed in the absence of noise for well-determined problems.

A primary disadvantage cited for most bilevel methods is the computational cost compared to single-level hyperparameter optimization methods or other methods with a smaller learning component. In turn, the main driver behind the large computational cost of gradient descent based bilevel optimization methods is that one typically has to optimize the lower-level cost function many times, either to some tolerance or for a certain number of iterations. The computational cost involves a trade-off because how accurately one optimizes the lower-level problem can impact the quality of the learned parameters. For example, [162], [192] both claim better denoising accuracy than [163] because they optimize the lower-level problem more accurately. Similarly, [271] notes that learning will fail if the lower-level cost is not optimized to sufficient accuracy.

There are various strategies to decrease the computational cost for bilevel methods. Some are relatively intuitive and applicable to a wide range of problems in machine learning. For example, [271] used larger batch size as the iterations continue, [143] increased the batch size if a gradient step in $\boldsymbol{\gamma}$ does not sufficiently improve the loss function, and [159] tightened the accuracy requirement for the gradient estimation over iterations. These strategies all save computation by starting with rougher approximations near the beginning of the optimization method, when $\boldsymbol{\gamma}^{(u)}$ is likely far from $\hat{\boldsymbol{\gamma}}$, while using a relatively accurate solution by the end of the algorithm.

Another disadvantage of bilevel methods is that, while the optimization algorithm for the lower-level problem often has theoretical convergence guarantees, and the lower-level cost is often designed to be strictly convex, the full bilevel problem (UL) is usually non-convex, so the quality of the learned hyperparameters can depend on initialization. Thus, in practice, one requires a strategy for initializing $\boldsymbol{\gamma}$. For example, for (Ex), one may decide to use a single-level filter learning technique such as the Field of Experts [187] to initialize the hyperparameters. Or, one can use a handcrafted set of filters, such as the DCT filters (or a subset thereof). Other hyperparameters often have similar warm start options. Despite the non-convexity, papers that tested multiple initializations generally found similarly good solutions surprisingly often, *e.g.*, [159], [162], [289].

There is no one correct answer for how much a method should use prior information or learning techniques, and it is unlikely that any single approach can be the best for all image reconstruction applications. Like most engineering problems, the trade-off is application-dependent. One should (minimally) consider the amount of training data available, how representative the training data is of the test data, how under-determined the forward model is (*i.e.*, how strong of regularization is needed), how well-known the object model is, the importance of theoretical guarantees and explainability, and the available computational resources at training time and at test time. Bilevel methods show particular promise for applications where training data is limited and/or explainability is highly valued, such as in medical imaging.

APPENDICES

APPENDIX A

Full Survey Instruments

A.1 Qualitative Protocols

The protocols in the following sections all refer to a table of concepts that we offered participants. We provided this in a formatted table for in-person interactions and as a list in the chat window for the interactions over Zoom.

Note that the table simply lists what the concepts are, without describing any in much detail. The purpose of the table was to job participants' memories (especially those of engineers working in industry) of what is covered in S&S and what they already know about those concepts; we did not want the table to explain concepts and accidentally increase participants' knowledge.

Table A.1: Table of concepts in S&S provided to interview and focus group participants.

Topic	Description
Background mathematics	Function manipulation in the time domain (for example, subtracting functions from each other or shifting a function in time)
Linearity and time invariance	Definition of linear and time invariant (LTI) and properties of systems that are LTI
Convolution	The procedure for simple graphical convolution, how convolution can be used to determine the system output given the impulse response, and the relationship with multiplication and the Fourier transform
Fourier transform	How the Fourier transform (FT) maps time to frequency space and basic properties of the FT
Pole-zero plots	How pole-zero plots can be used to determine system causality and stability
Filtering	The procedure for using the frequency response to determine the output of a system given the input

A.1.1 Focus group protocol

We used the same basic protocol for both the undergraduate and graduate focus groups, with only minor modifications. The focus groups lasted one hour each, with about five minutes allowed for everyone arriving and getting settled.

1. **Background:** The purpose of the first step is to make participants feel comfortable with the format of the focus group and to follow good Institutional Review Board (IRB) practices.

    - I'm Caroline. I've organized this focus group to talk about how students learn signals and systems. You've all taken [course name/number here for signals and systems at given university], so I want to get your opinions on your experience. I might step in occasionally to redirect the conversation to make sure we get through everything in the time we have or to pose new questions, but I really want to hear from all of you because you've had different experiences. It's helpful if you talk to each other, not just to me, and there aren't any right or wrong answers here, so it's okay to disagree with each other.

    - Just a reminder, the conversation's being audio-recorded for me to transcribe and look at later. However, it's only me that will ever hear the tape, and nobody will be named on the transcript. I don't expect to talk about anything too sensitive today, but please respect the other people and don't talk about what others share here after we leave. Any questions before we get started?

2. **Introduction to the research:** Next is a simple intro question to promote interaction and get everyone to talk at least once. I talk first to set an expectation for how much everyone will talk and to further build rapport with the participants by acknowledging that there are concepts in S&S that I did not learn during my course.

    - I want to start by just getting to know each other a little. Can we all go around and say your name, year, when you took [signals and systems course], and one example of something you remember or something you have completely forgotten from the course.

    - I go first by introducing myself and my experience in signals and systems.

3. **Formally introduce concepts:** We provided a list of concepts to help them recall what S&S covered.

    - Okay, I want to just give you a little more detail on our topic for the day. I really want to focus on concepts in signals and systems, not procedures. So, for example, I'm more interested in how you learned that the Fourier transform converts signal space to frequency space and less interested if you can still do the math and take a Fourier transform by hand.

    - With that said, here's a cheat sheet for what concepts I am interested in talking about today (see Tab. A.1). Leave a minute for them to read it.

    - Do the descriptions of the concepts are clear to you?

4. **Focus in-course:** The first major goal of the focus group is to understand what factors helped/hindered students learning concepts during their S&S course.

    - While you were taking the course, what do you think helped or hindered you from understanding these concepts?

    - If the following ideas do not come up, probe for the possibility of: format of the class, interest in the topic, perceived value of topic to future career, use of the topic outside of class.

    - If needed to stimulate conversation, you can also probe for how they learned specific concepts.

5. **Focus longitudinal:** The second major goal of the focus group is to understand what factors made students learn, retain, or forget concepts after their S&S course.

- Do you think you have a better or worse understanding of these concepts now versus right after you finished [signals and systems]?
- If needed to stimulate conversation: Is there any reason you remember one concept but not another?
- Was there anything that you feel like helped or hindered you learning more of the concepts after you finished [signals and systems]?
- As before, if these items do not come up, you can probe for: interest in the topic, perceived value of the topic, use of the topic outside of class, seeing the topic again in upper-level courses

6. **Summarizing big-picture question:** This question is more open-ended and should encourage students to think more broadly about their S&S education.

   - Okay, I feel like we had a lot of good things come out of that conversation. I want to just ask a final summarizing question. If you had to name just one thing that helped and one thing that hindered your learning of signals and systems, what would it be? It can be something we've already talked about or something new.

7. **Conclusion:** Thank the participants and make sure they have nothing else to add.

   - I have no more questions to ask but is there anything else you all would like to bring up, or ask about, before we finish this session?
   - Thanks for coming and being a part of the conversation. It's really helpful for us to hear from current students and get your perspectives. If there is something you didn't have a chance to say, I'll send a follow-up email just to thank you and feel free to send a response with anything you thought of.

A.1.2 Industry interview

We scheduled industry interviews for 30 minutes, and reserved the last 5 minutes to allow our participants to ask any questions about the research (because we did not reward industry participants with gift cards or free food, this was a small gesture of thanks). We did all industry interviews over Zoom.

1. **Background:** I start the interview by introducing myself and reviewing the most important parts of the IRB consent form.

   - I'm Caroline. As part of my dissertation, I'm studying conceptual understanding of signals and systems. I've organized this interview to get your opinions as someone in industry. There are no right or wrong answers.
   - Just a reminder, the conversation's being audio recorded for me to transcribe and look at later. However, it's only me that will ever hear the tape, and you will be anonymous in any reporting of the data.
   - Any questions before we get started?

2. **Concepts**:

   - I really want to focus on concepts in signals and systems, not procedures. So, for example, I'm more interested in how students learn that the Fourier transform converts signal space to frequency space and less interested if they can still do the math and take a Fourier transform by hand.

- With that said, here are some common concepts in signals and systems. Briefly talk through the hand-out (see A.1).
- Any questions about conceptual understanding or these concepts?

3. **Student years:**

   - I want to start by asking you about when you were a student. Can you first briefly tell me about your undergraduate and how your signals and systems class fit into the curriculum and what it covered?
   - When you were first learning signals and systems, which concepts did you find particularly easy or hard to understand?
   - Why? Probe as needed for impact from teaching style, interest/motivation, previous classes/known material, etc.
   - As you continued in undergraduate classes, do you remember any concept making more sense? Or perhaps making less sense?
   - Probe as needed for impact of other courses and passing of time helping/hurting?

4. **Industry years:**

   - I want to transition to talking about your experiences in industry.
   - Can you first tell me a little about what your job entails and maybe about the other positions you've had over the years? [Note: from experience, this usually comes up naturally earlier in the interview and I can skip this question.]
   - How often do you find yourself using material from signals and systems? In what way do you use it? Which parts of it come up most?
   - Do you feel like you understand signals and systems better or worse now than when you were an undergrad? (probe for why and various possible factors)
   - How do you feel your undergraduate S&S related courses prepared you for your work in industry?

5. **Summarizing big-picture question**

   - I feel like we've talked about a lot of good things. I want to ask some big-picture summarizing questions.
   - If you could change one thing about your undergraduate EE education, what would be your change? What if you had to change something about the signals and systems curriculum? (Note if its purpose to help conceptual understanding?)

6. **Conclusion:**

   - Thanks again for talking with me. It's really helpful for me to get your perspective.
   - Do you have anything else you want to add?
   - I like to leave the last few minutes in case you want to ask me any questions about the research.

A.1.3 Instructor Interview

We scheduled instructor interviews for 60 minutes, but planned to take only 45 (we reserved the last 15 minutes for answering their questions about our study). We did the UM interview in-person and the UVA interview over Zoom due to COVID-19.

1. **Background:** I start the interview by introducing myself and reviewing the most important parts of the IRB consent form.

    - I'm Caroline. As part of my dissertation, I'm studying conceptual understanding of signals and systems. I've organized this interview to get your opinions as someone in industry. There are no right or wrong answers.
    - Just a reminder, the conversation's being audio recorded for me to transcribe and look at later. However, it's only me that will ever hear the tape, and you will be anonymous in any reporting of the data.
    - Any questions before we get started?

2. **Concepts**:

    - I really want to focus on concepts in signals and systems, not procedures. So, for example, I'm more interested in how students learn that the Fourier transform converts signal space to frequency space and less interested if they can still do the math and take a Fourier transform by hand.
    - With that said, here are some common concepts in signals and systems. Briefly talk through the hand-out (see A.1).
    - Any questions about conceptual understanding or these concepts?

3. **Student years:** By asking the instructors to think back to when they were students, we hope to help them recall struggling with certain concepts that they now know so well and to get them more engaged in our interview questions, rather than giving a pre-planned lecture on students understanding of conceptual understanding.

    - I want to start by asking you about when you were a student. Can you first briefly tell me about your undergraduate and how your signals and systems class fit into the curriculum and what it covered?
    - When you were first learning signals and systems, which concepts did you find particularly easy or hard to understand?
    - Why? Probe as needed for impact from teaching style, interest/motivation, previous classes/known material, etc.
    - As you continued in undergraduate classes, do you remember any concept making more sense? Or perhaps making less sense?
    - Probe as needed for impact of other courses and passing of time helping/hurting?

4. **Teaching years:** This is the primary focus of the interview.

    - I want to transition to talking about your experiences teaching signals and systems.
    - I want to start by asking about your overall philosophy regarding [signals and systems course]. How have you structured your course and why did you do it that way?

- Probe for: Lectures, lab design, homework problems, changes over the years, graphical vs mathematical representation (e.g., of convolution), and point of each component of the class
- What concepts do you think students find easiest or hardest? Why?
- Probe for interest/perceived usefulness, prior classes, level of abstraction, etc.
- Have you tried modifying the class or your teaching to help with the harder concepts? If so, how did it work out?

5. **Summarizing big-picture question**:

    - I feel like we've talked about a lot of good things. I want to ask some big-picture summarizing questions.
    - If you could redesign the signals and systems curriculum without any time, financial, or department-imposed constraints, what would be your biggest change?

6. **Conclusion:**

    - Thanks again for talking with me. It's really helpful for me to get your perspective.
    - Do you have anything else you want to add?
    - I like to leave the last few minutes in case you want to ask me any questions about the research.

A.2 Surveys

Note that the formatting presented below is not a representation of the survey given to students. All Likert questions had 5-point scales. Here, we only present the two extremes of the Likert response options. Most surveys (those other than Fall 2019 in EECS 216 at UM) were given in Qualtrics.

All surveys were proceeded by a statement that the survey would not impact their grade and that their answers would be used anonymously.

A.2.1 Survey #1

This is the survey given to students at the end of their S&S class (EECS 216 at UM and FUN 3 at UVA). For text that differs between the UM survey and UVA survey, we present both versions of the text in brackets, *i.e.*, [UM survey text/UVA survey text]

1. I would like to graduate with a major in Electrical Engineering [1]. [Strongly disagree – Strongly agree]
2. After graduating, I would like to... (*mark all that apply*)

    - be in a technical role
    - be in a managerial role
    - be a systems engineer
    - teach
    - work in a service- focused role

[1] This question was not included in the UVA survey as most students have already declared their major by the time they take FUN 3

- do research (in industry or academia)
- get a Master's degree in [text entry line]
- get a PhD degree in [text entry line]
- Other: [text entry line]

3. Learning signals and systems is interesting. [Strongly disagree – Strongly agree]
4. For each of the following, please respond to the statement: Understanding [2] *this topic* will benefit me in my career. [Strongly disagree – Strongly agree]
    - Convolution
    - Linear and Time Invariance
    - Fourier transform
    - Laplace transform
    - Filtering
5. How would you rate the overall quality of instruction in [EECS 216/the three Fundamentals courses]? [Very poor – Excellent]
6. How often did your peers help your understanding of the [EECS 216/Fundamentals] material? [Never – Always]
7. The [EECS 216/Fundamentals] learning environment made me feel comfortable. [Strongly disagree – Strongly agree]
8. In a typical week, how many hours did you spend on [EECS 216/Fundamentals] homework (including work completed for lab outside of your lab session) [3]? [text entry line]
9. What percentage of EECS 216 lectures did you attend in-person or watch recorded [4]? [text entry line]
10. What is the highest educational status achieved by your parent(s)/guardian(s)?
    - Did not finish high school
    - High school degree
    - Associates degree
    - Bachelor's Degree
    - Master's degree
    - Doctoral or Professional degree
11. How do you describe your gender identity? [text entry line]
12. With which racial and ethnic group(s) do you identify? (*Mark all that apply*)
    - American Indian or Alaska Native

[2] "Learning" was used in place of "Understanding" on the Fall 2019 survey

[3] The parenthetical statement was not included in the Fall 2019 survey nor in the UVA survey

[4] This question was just "What percentage of EECS 216 lectures did you attend" on the Fall 2019 survey

- Asian
- Black or African American
- Hispanic, Latino, or Spanish origin
- Middle Eastern or North African
- Native Hawaiian or Other Pacific Islander
- White
- Another race or ethnicity not listed: [text entry line]

13. Is there anything you would like to add?[5]

A.2.2 Survey #2

This is the survey given to students in their final year of undergraduate studies. For text that differs between the UM survey and UVA survey, we present both versions of the text in brackets, *i.e.*, [UM survey text/UVA survey text]

1. After graduating, I would like to... (*mark all that apply*)
    - be in a technical role
    - be in a managerial role
    - be a systems-level designer
    - teach
    - work in a service-focused role
    - Go into a non-traditional engineering role (such as banking, finance, law, medicine, or sales)
    - do research (in industry or academia)
    - get a Master's degree in [text entry line]
    - get a PhD degree in [text entry line]
    - Other: [text entry line]
2. I am majoring in:
    - EE
    - CpE
    - Another major: [text entry line]
3. For each of the following topics, please respond to the statement: Understanding *this signals and systems topic* will benefit me in my career. [Strongly disagree – Strongly agree]
    - Convolution
    - Linear and Time Invariance

[5] We did not explicitly ask this on the paper version of the survey in Fall 2019, though there was room for students to write comments.

   - Fourier transform
   - Laplace transform
   - Filtering

4. I am excited to learn about Signals and Systems. [Strongly disagree – Strongly agree]
5. How would you rate the overall quality of instruction in [EECS 216 (or equivalent signals and systems class if you took it elsewhere)/FUN 2 and FUN 3] ? [Terrible – Excellent]
6. How much did your peers help your understanding of the [signals and systems/FUN 2 and FUN 3] material? [None at all – A great deal]
7. The [EECS 216/FUN 2 and FUN 3] learning environment made me feel comfortable. [Strongly disagree – Strongly agree]
8. In a typical week, [and averaging across the two semesters,] how many hours did you spend on [EECS 216/FUN 2 and FUN 3] homework (including time spent on lab assignments out of class)? [text entry line]
9. In any extracurriculars and/or internships you did, how often did you use the concepts from signals and systems? [None at all – A great deal]
10. Did you complete a course on differential equations (such as [MATH 216/APMA 2130]) before beginning [EECS 216/FUN 2]?
    - Yes
    - I'm not sure, but I think so
    - I'm not sure, but I think not
    - No (please select this if you took them in the same semester)
11. - Mostly A
    - About half A and half B
    - Mostly B
    - About half B and half C
    - Mostly C or below
12. My grade in [EECS 216/FUN 2, 3[6]] was (your best guess is fine if you do not remember):
    - A+, A, or A-
    - B+, B, or B-
    - C+, C, or C-
    - D+ or below
13. UM only: Which semester did you take EECS 216 (or equivalent signals and systems class if you took it elsewhere)? Your best guess is fine if you do not remember.

[6] Asked as two separate questions

- Fall 2017 or earlier
- Winter 2018
- Fall 2018
- Winter 2019
- Winter 2020
- Fall 2020[7]
- Winter 2021
- Other: [test box entry]

14. Which of the following courses have you taken or are you currently taking?

    - UM list: EECS 351: Intro to digital signal processing, EECS 442: Computer vision, EECS 445: Machine learning, EECS 452: Digital signal processing design lab, EECS 455: Wireless communications systems, EECS/BIOMEDE 458: Biomedical instrumentation and design, EECS 460: Control system analysis and design, EECS 461: Embedded control
    - UVA list: Science of information (ECE 2066), How the iPhone works (specific section of ECE 2066), The math of information (specific section of ECE 2066), Wireless devices (ECE 4209), Communications (ECE 4710), Digital signal processing (ECE 4750), Wireless communications (ECE 4784/6784), Digital image processing (a section of ECE 4501 or ECE 6782), Linear control systems (ECE 4850)

15. Considering your experience in all courses that cover concepts in signals and systems:[8] [Never – Always]

    - The overall instruction was high-quality
    - I was engaged in the classes
    - The courses included hands-on activities, demonstrations, or open-ended projects

16. Considering your experience in all courses that cover concepts in signals and systems: [Strongly disagree – Strongly agree]

    - On average, students felt comfortable participating in classes (for example, asking or answering questions)
    - The classes made me interested in learning signals and systems
    - The classes made me more confident in my engineering knowledge

17. Growing up, I was aware of my family's educational expectations for me. [Strongly disagree – Strongly agree]

18. My family believes a college education is important for my future. [Strongly disagree – Strongly agree]

19. While I was growing up, my family encouraged me to take classes that would challenge me. [Strongly disagree – Strongly agree]

20. In general, my friends value receiving good grades. [Strongly disagree – Strongly agree]

[7] Fall 2020 and Winter 2021 were only options on the Fall 2021 survey.

[8] This questions and the following were deliberately placed immediately after the question that listed courses that used signals and systems concepts.

21. In general, my friends enjoy engineering. [Strongly disagree – Strongly agree]

22. What is the highest educational status achieved by your parent(s)/guardian(s)?

    - Did not finish high school
    - High school degree
    - Associates degree
    - Bachelor's Degree
    - Master's degree
    - Doctoral or Professional degree

23. How do you describe your gender identity (such as "female")[9]? [text entry line]

24. With which racial and ethnic group(s) do you identify? (*Mark all that apply*)

    - American Indian or Alaska Native
    - Asian
    - Black or African American
    - Hispanic, Latino, or Spanish origin
    - Middle Eastern or North African
    - Native Hawaiian or Other Pacific Islander
    - White
    - Another race or ethnicity not listed: [text entry line]

25. In Fall 2021 only:

    - UVA: Are you interested in participating in an interview on Nov. 3 or 4? This would involve talking through some questions from the concept inventory for 30 minutes. You would get a $10 gift card to thank you for your time. No preparation required. Select yes if interested. This is a chance to learn more, not a commitment. Prof. Powell is not involved and will not know whether you decide to participate. Research study ID: UVA IRB-SBS #4661.
    - UM:

26. Is there anything you would like to add?

[9] This example was added because some of the responses from the first survey were "heterosexual" or similar.

APPENDIX B

Qualitative Codebook

The following table is the full codebook for the qualitative analysis from Chapter 5, including example quotes from the interviewed undergraduate students (UG), graduate students (G), faculty (F), and practicing engineers (PE).

Table B.1: Codebook for the focus groups and interviews.

Code	Description	Example quote
	Concepts in SS	
FT	Fourier Transform and Fourier Series. General comments about the frequency domain.	"Like the FT is pretty intuitive in that, at least for me, when there's things like music visualizers, that is exactly showing the spectrum of your sound" (G).
LT	Laplace Transform and pole-zero plots.	"I mean, I remember how to take a Laplace transform – well, if I looked it up, I could do it" (G)
Conv	Convolution. Includes comments about impulse or frequency response.	"Students have a heck of a time doing convolution. And in real life, no one does convolution, which is interesting" (F)
LTI	Basic system properties (linearity, time invariance, and stability or their opposites)	"I feel like everybody just kinda assumes LTI. There's never been like a conversation where you're just like kinda like 'what if it's not LTI?' ... It's like don't go to the forbidden zone" (UG).
Filtering	Filtering or Bode plots.	"One thing that helped me a lot... was getting an intuition of what kind of filtering is the most useful for what your goal is... Like any kind of low pass filter means you're blurring out edges" (UG).
DT	Anything contrasting continuous and discrete time or talking specifically about discrete time.	"I was super confused at how to do the Fourier transform on the analog version before doing it on the discrete version. And then once we did it in code it just sort of clicked for me" (PE).

Table B.1 continued from previous page

Code	Description	Example quote
Math	The math curriculum or mathematics in signals and systems.	“Peoples’ exposure to complex variables prior to college is probably like a maybe a month or two in honors, pre-calc, or in a calculus course, probably in high school... And then all of electrical engineering is complex variables” (UG).
Class components		
class	Class environment, *e.g.*, if students felt comfortable participating. Includes office hours, discussion sections, and online interactions.	“That perceived sense of diversity [of students’ majors in the class] made it a lot easier to accept that it was okay to not know what was going on because I felt like everyone else didn’t and we’re all in the same boat even if that boat is sort of sinking” (G).
grades	Grades, feedback on assignments, the grading weights, or exams.	“[homeworks] were only like 10% of the grade when they were like a huge part of the work of the class” (UG).
hw	Homework problems.	“[If] the questions are designed in such a way that they test your understanding, maybe that helps more. But then our homeworks are also similar, it’s very, very procedural” (G).
lab	All comments about lab sections.	“I wanted the students to appreciate that these techniques were very powerful and they actually allow you to build and analyze real things. And this was the hope for the lab” (F).
visuals	Visuals and how they complement or compare with mathematical formulas.	“I certainly visualize things a lot better than like hearing them spoken to me... especially in the Signals and system space, pictures are really useful to sort of describe what’s going on” (PE).
Instructional quality and quantity		
quantity	Amount of work for the class or time in class spent on concepts/a specific concept. Specific to S&S classes or concepts.	“I just really really disliked those homeworks, just cause like they were too long... I feel like you could have gotten close to the same understanding it they had been half the length” (UG).
quantity: repetition	Seeing material again in a separate class, an extracurricular activity, or in industry.	“I would say that my knowledge grew after taking other classes as well. Um, but also very specific to what areas I ended up taking more classes in, because I would advance in those areas but not advance at all in other EE areas and maybe forget some stuff too” (UG).
pace	The pace of a S&S course and constraints on how much material must be covered during the course.	“I’m afraid in the labs, they’re so time-pressured, that they’re going through the, you know, the steps, but they-they don’t really have enough time to really sit down and figure out why they’re doing it all” (F).

Table B.1 continued from previous page

Code	Description	Example quote
quality: style	The quality of instruction, including active learning strategies and how the student liked the style of the class, *e.g.*, emphasis on theory vs. application.	"I had just good professors for all of them that I would say resonated with myself... I think my junior level signals and systems course was very much just like a very strict, it's like we're going through this mathematically, textbook style, going through it. That's helpful for me" (PE).
	Interest	
purpose	Seeing or wanting to see the big picture or wondering why something is the way it is. May or may not cause interest or motivation.	"I loved it because it really connected. It was an engineering course.. It was, it was the whole system. Like you had to think big picture. And it was great" (PE).
motivate	Any comments related to student motivation or motivating students that does not otherwise fall under purpose or grades. Includes statements about interest.	"...they told you like, 'hey, we're building this, and you can test this, and you can play with it and do things with it.' That's what keeps you motivated to keep working at the projects" (UG).
abstract	Discussion of the ideas in S&S being abstract or, the opposite, concrete. Includes mentions of real examples of systems, how designing systems impacted learning, and how building systems in hardware impacted learning.	"Pole-zero plots were pretty easy for me because I was an analog guy, I understood what-what poles and zeros were. That wasn't a big deal, that-that's all translatable" (PE).
	Outcomes and reactions	
ability	General student ability to reason/think or anything that prepared students (or didn't) for S&S, such as former courses or extracurricular activities.	"I have to imagine that a lot of the people that either struggled or succeeded in that course, really hinged on that first bullet point of the - just the background mathematics" (PE).
easy	Comments about a concept being relatively easy or hard to understand.	"there were things that I later learned were actually pretty easy to understand. But just in that one point in time, the way it was presented just wasn't the right fit for me or I just, it just didn't click for some reason" (UG).
false confidence	Realization that they previously thought they understood something, but later realized they did not. Includes comments about "unlearning" a concept because they did not learn it correctly the first time.	"On a surface level LTI concepts are like the easiest thing... But, listening to [name removed], I realized, I never learned why it's good for a system to be linear or LTI... So I guess I had like a sense of false confidence that I really understood this stuff" (UG).
familiar	Knowing enough about a topic to know it exists and how to find out more about it if they need to in the future. Includes comments about forgetting specifics, especially when discussed over time.	"I understand it worse now but I think, I think it would come back, it wouldn't be too hard for it to come back to me if I like reviewed the material" (UG).

Table B.1 continued from previous page

Code	Description	Example quote
hate	Hatred, fear, or love of a concept, S&S, or a S&S instructor.	"I really thought about jumping to a computer science degree because I was so flustered by my undergraduate signals and systems course.. Which again, the irony is that I love this stuff" (PE).
importance	Importance of CU in S&S or more broadly.	"what I constantly try to emphasize is conceptual... Because if you know that, you can fill in the details" (F).
Procedural	Comparing S&S to a math course or any comments about the procedural nature of S&S knowledge or S&S courses.	"I've had numerous comments over the year of course evaluations and things. 'This seems like just a math course.' I think that is one perception that could easily be held" (F).
	Outside influences	
peers	Interaction with or influence from peers. For example, comments about friends enjoying (or disliking) certain classes, valuing good grades, cheating, and working together on assignments.	"Interaction with my teammates and interactions with the TAs has just really helped me" (UG).
work	How internships or work experience influence CU and learning.	"The very use of the term signal is a little ambiguous, I suspect, to students. ... the context in which we use it is just a representation of a time series continuous time, or could be in space could be an image or something like that. It's a mathematical abstraction of some physical quantity" (F).
workload	Students responsibilities outside of the S&S class. Includes other classes, social life, etc.	"it definitely was a huge factor in my undergrad as to like whether or not I actually learned something or enjoyed it, was like how much I had going on" (G).

APPENDIX C

Background: Primal-Dual Formulations

This appendix is presented Crockett and Fessler [11, App. A].

This appendix briefly reviews primal-dual analysis as it applies to (Ex). Section 3.3 in [173] provides a more general but brief introduction to the notion of conjugate functions and duality and [344] goes into more depth on duality.

The conjugate of a function $f : \mathbb{R}^N \to \mathbb{R} \cup \{\text{-}\infty, \infty\}$ is denoted $f^* : \mathbb{R}^N \to \mathbb{R} \cup \{\text{-}\infty, \infty\}$, and is defined as

$$f^*(\boldsymbol{d}) = \sup_{\boldsymbol{x} \in \text{domain}(f)} \boldsymbol{d}'\boldsymbol{x} - f(\boldsymbol{x}), \tag{C.1}$$

where $\boldsymbol{d} \in \mathbb{R}^N$ is a dual variable. The derivations below use the following two conjugate function relations.

1. When $f(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{x} - \boldsymbol{y}\|^2$ for $\boldsymbol{y} \in \mathbb{R}^N$, the conjugate function is

   $$f^*(\boldsymbol{d}) = \sup_{\boldsymbol{x} \in \mathbb{R}^N} \boldsymbol{d}'\boldsymbol{x} - \frac{1}{2}\|\boldsymbol{x} - \boldsymbol{y}\|^2.$$

   The maximizer of the quadratic cost function f^* is

   $$\hat{\boldsymbol{x}} = \boldsymbol{y} + \boldsymbol{d} \tag{C.2}$$

   and the maximum value simplifies to

   $$f^*(\boldsymbol{d}) = \frac{1}{2}\,\|\boldsymbol{d} + \boldsymbol{y}\|^2 - \frac{1}{2}\|\boldsymbol{y}\|^2. \tag{C.3}$$

2. When $\phi(z) = |z|$ is defined on $\mathbb{R}$, the conjugate function is

   $$\phi^*(d) = \sup_{z \in \mathbb{R}} dz - |z|.$$

   One can verify that the conjugate is

   $$\phi^*(d) = \begin{cases} 0 & \text{if } |d| \le 1 \\ \infty & \text{else} \end{cases} \tag{C.4}$$

and the corresponding sets of suprema are

$$\operatorname*{argmax}_{z \in \mathbb{R}} dz - |z| = \begin{cases} \operatorname{sign}(d) \cdot \infty & \text{if } |d| > 1 \\ 0 & \text{if } |d| < 1 \\ [0, \infty) & \text{if } d = 1 \\ (\text{-}\infty, 0] & \text{if } d = \text{-}1. \end{cases} \tag{C.5}$$

Generalizing (C.4) to a vector, the conjugate function of the 1-norm is a characteristic function that is infinity if any element of the input vector is larger than 1 in absolute value.

Ref. [344, p. 50] provides a table with many more conjugate functions.

The biconjugate, denoted f^{**}, is the conjugate of f^*, *i.e.*,

$$f^{**}(\boldsymbol{x}) = \sup_{\boldsymbol{d} \in \operatorname{domain}(f^*)} \boldsymbol{x}'\boldsymbol{d} - f^*(\boldsymbol{d}), \tag{C.6}$$

and is the largest convex, lower semi-continuous function below f. When f is convex and lower semi-continuous, the biconjugate is equal to the original function, *i.e.*, $f^{**} = f$. One can use the equality of the original function and the biconjugate to derive the saddle point and dual problems when f is convex.

Consider the specific lower-level problem with an analysis-based regularizer

$$\operatorname*{argmin}_{\boldsymbol{x} \in \mathbb{R}^N} \frac{1}{2} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|^2 + \boldsymbol{1}'\phi_{.}(\boldsymbol{\Omega}\boldsymbol{x}), \tag{C.7}$$

where $\boldsymbol{\Omega} \in \mathbb{R}^{F \times N}$. When ϕ is convex, the corresponding saddle-point problem is

$$\operatorname*{argmin}_{\boldsymbol{x} \in \mathbb{R}^N} \frac{1}{2} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|^2 + \underbrace{\sup_{\boldsymbol{d} \in \mathbb{R}^F} \langle \boldsymbol{d}, \boldsymbol{\Omega}\boldsymbol{x} \rangle - \boldsymbol{1}'\phi^*.(\boldsymbol{d})}_{\boldsymbol{1}'\phi^{**}_{.}(\boldsymbol{\Omega}\boldsymbol{x})},$$

where $\langle \cdot, \cdot, \rangle$ is the standard inner product. Under very mild conditions (satisfied for the absolute value function) [173], one can swap the minimum and supremum operations and write the **saddle-point problem** as

$$\sup_{\boldsymbol{d} \in \mathbb{R}^F} \min_{\boldsymbol{x} \in \mathbb{R}^N} \frac{1}{2} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|^2 + \langle \boldsymbol{d}, \boldsymbol{\Omega}\boldsymbol{x} \rangle - \boldsymbol{1}'\phi^*.(\boldsymbol{d}).$$

Substituting the conjugate of the 1-norm (C.4), the saddle-point problem is thus

$$\min_{\boldsymbol{x} \in \mathbb{R}^N} \min_{\boldsymbol{d} \in \mathbb{R}^F} \frac{1}{2} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|^2 - \langle \boldsymbol{d}, \boldsymbol{\Omega}\boldsymbol{x} \rangle \text{ s.t. } |d_i| \leq 1 \ \forall i. \tag{C.8}$$

We hereafter assume $\boldsymbol{A} = \boldsymbol{I}$ to derive the dual problem from the saddle-point problem. By grouping terms and re-arranging negative signs, the dual problem can be derived from the saddle point problem. For a general ϕ, the saddle-point problem is equivalent to

$$\begin{aligned}
&\max_{\boldsymbol{d} \in \mathbb{R}^F} \text{-}\boldsymbol{1}'\phi^*.(\boldsymbol{d}) + \left(\min_{\boldsymbol{x} \in \mathbb{R}^N} \langle \boldsymbol{d}, \boldsymbol{\Omega x} \rangle + \frac{1}{2} \|\boldsymbol{x} - \boldsymbol{y}\|^2 \right) \\
&= \max_{\boldsymbol{d} \in \mathbb{R}^F} \text{-}\boldsymbol{1}'\phi^*.(\boldsymbol{d}) - \underbrace{\left(\max_{\boldsymbol{x} \in \mathbb{R}^N} \langle \text{-}\boldsymbol{\Omega}'\boldsymbol{d}, \boldsymbol{x} \rangle - \frac{1}{2} \|\boldsymbol{x} - \boldsymbol{y}\|^2 \right)}_{f^*(\text{-}\boldsymbol{\Omega}'\boldsymbol{d})},
\end{aligned}$$

where the last line follows from properties of inner products. The expression in parenthesis is the conjugate function for the data-fit term, given in (C.3). Therefore, the dual problem for a general, convex ϕ is

$$\max_{\boldsymbol{d} \in \mathbb{R}^F} \text{-}\boldsymbol{1}'\phi^*.(\boldsymbol{d}) - f^*(\text{-}\boldsymbol{\Omega}'\boldsymbol{d}) = \text{-} \min_{\boldsymbol{d} \in \mathbb{R}^F} \boldsymbol{1}'\phi^*.(\boldsymbol{d}) + f^*(\text{-}\boldsymbol{\Omega}'\boldsymbol{d}).$$

Substituting the conjugates for the data-fit term (C.3) and the conjugate for the 1-norm regularizer (C.4), the **dual problem** for (C.7) with $\phi(z) = |z|$ becomes

$$\min_{\boldsymbol{d} \in \mathbb{R}^F} \frac{1}{2} \left\| \text{-}\boldsymbol{\Omega}'\boldsymbol{d} + \boldsymbol{y} \right\|^2 - \frac{1}{2} \|\boldsymbol{y}\|^2 \text{ s.t. } |d_i| \leq 1 \ \forall i. \tag{C.9}$$

When we require only the minimizer (not the minimum), an equivalent dual problem is

$$\hat{\boldsymbol{d}} = \operatorname*{argmin}_{\boldsymbol{d} \in \mathbb{R}^F} \frac{1}{2} \left\| \text{-}\boldsymbol{\Omega}'\boldsymbol{d} + \boldsymbol{y} \right\|^2 \text{ s.t. } |d_i| \leq 1 \ \forall i. \tag{C.10}$$

This dual problem is a constrained least squares problem and can be solved with a projected gradient descent method, optionally with momentum [295]. From (C.2), the primal minimizer can be recovered from the dual minimizer by

$$\hat{\boldsymbol{x}} = \boldsymbol{y} - \boldsymbol{\Omega}'\hat{\boldsymbol{d}}. \tag{C.11}$$

Finally, from (C.5), the dual variable is related to the filtered signal by

$$d_i \in \begin{cases} 1 & \text{if } [\boldsymbol{\Omega}\hat{\boldsymbol{x}}]_i > 0 \\ \text{-}1 & \text{if } [\boldsymbol{\Omega}\hat{\boldsymbol{x}}]_i < 0 \\ [0, \infty) & \text{if } [\boldsymbol{\Omega}\hat{\boldsymbol{x}}]_i = 1 \\ (\text{-}\infty, 0] & \text{if } [\boldsymbol{\Omega}\hat{\boldsymbol{x}}]_i = \text{-}1. \end{cases} \tag{C.12}$$

Ref. [273] provides a more general version of the dual function for non-identity system matrices.

Above, we derived the saddle-point and dual problems using the equality of the biconjugate and the original function for a convex regularizer. The dual problem can also be derived using Lagrangian theory, as shown in [273]. Define an auxiliary (split) variable that is constrained to equal the filtered signal, *i.e.*, $\boldsymbol{z} = \boldsymbol{\Omega x}$. Considering the specific case of the 1-norm regularizer, the Lagrangian of the constrained version of (C.7) is

$$\frac{1}{2} \|\boldsymbol{x} - \boldsymbol{y}\|^2 + \|\boldsymbol{z}\|_1 + \boldsymbol{d}'(\boldsymbol{\Omega x} - \boldsymbol{z}),$$

where $\boldsymbol{d} \in \mathbb{R}^F$ is a vector of Lagrange multipliers and we have omitted the KKT conditions. Minimizing the Lagrangian with respect to $\boldsymbol{x}$ and $\boldsymbol{z}$ yields the conjugate functions for the data-fit term and 1-norm and thus the dual problem.

Using the Lagrangian perspective to derive the dual problem yields a useful relation between the filtered signal and the dual variable [273]. Because the split variable $\boldsymbol{z}$ is constrained to equal $\boldsymbol{\Omega x}$, $[\boldsymbol{\Omega x}]_i > 0$ implies $z_i > 0$. From (C.5), z_i is only positive and finite when $d_i = 1$. A similar argument holds for $[\boldsymbol{\Omega x}]_i < 0$. Therefore, the dual variable and $\hat{\boldsymbol{x}}$ are related by

$$d_i \in \begin{cases} \operatorname{sign}([\boldsymbol{\Omega x}]_i) & \text{if } [\boldsymbol{\Omega} \hat{\boldsymbol{x}}]_i \neq 0 \\ [\text{-}1, 1] & \text{if } [\boldsymbol{\Omega} \hat{\boldsymbol{x}}]_i = 0. \end{cases} \tag{C.13}$$

The second case follows from observing that d_i can take any value in its constrained range when $z_i = 0$ as the minimum in (C.9) will be 0 regardless of d_i.

The primal-dual results reviewed in this appendix are referenced in Section 8.1.2.3 to relate analysis and synthesis regularizers, Section 10.1.2 to rewrite the lower-level minimizer as a differentiable function of itself and $\boldsymbol{\gamma}$, and in Section 10.1.3.2 to unroll a differentiable algorithm for a non-smooth cost function.

APPENDIX D

Forward and Reverse Approaches to Unrolling

This appendix is presented Crockett and Fessler [11, App. B].

This appendix provides background on the forward and backward approaches to the unrolled gradient computation introduced in Section 10.1.3. From (10.18), the gradient of interest is:

$$\nabla \ell(\boldsymbol{\gamma}) = \nabla_{\boldsymbol{\gamma}} \ell(\boldsymbol{\gamma}\,; \boldsymbol{x}^{(T)}) + \left(\sum_{t=1}^{T} (\boldsymbol{H}_T \cdots \boldsymbol{H}_{t+1})\, \boldsymbol{J}_t \right)' \nabla_{\boldsymbol{x}} \ell(\boldsymbol{\gamma}\,; \boldsymbol{x}^{(T)}) \in \mathbb{F}^R. \tag{D.1}$$

If one uses a gradient descent based algorithm to optimize the lower-level cost function Φ, then $\boldsymbol{H}_t = \nabla_{\boldsymbol{x}} \Psi(\boldsymbol{x}^{(t-1)}\,; \boldsymbol{\gamma}) \in \mathbb{F}^{N \times N}$ is closely related to the Hessian of Φ and $\boldsymbol{J}_t = \nabla_{\boldsymbol{\gamma}} \Psi(\boldsymbol{x}^{(t-1)}\,; \boldsymbol{\gamma}) \in \mathbb{F}^{N \times R}$ is proportional to the Jacobian of the gradient.

To compare the forward and reverse approaches to gradient computation for unrolled methods, we introduce notation for an ordered product of matrices. We indicate the arrangement of the multiplications by the set endpoints, $s \in [s_1 \leftrightarrow s_2]$ with the left endpoint, s_1, corresponding to the index for the left-most matrix in the product and the right endpoint, s_2, corresponding to the right-most matrix. Thus, for any sequence of square matrices $\{\boldsymbol{A}\}_i$:

$$\prod_{s \in [t \leftrightarrow T]} \boldsymbol{A}_s := \boldsymbol{A}_t \boldsymbol{A}_{t+1} \cdots \boldsymbol{A}_T = (\boldsymbol{A}'_T \boldsymbol{A}'_{T-1} \cdots \boldsymbol{A}'_t)' = \left(\prod_{s \in [T \leftrightarrow t]} \boldsymbol{A}'_s \right)'.$$

The above double arrow notation does not indicate order of operations. In the following notation the arrow direction does not affect the product result (ignoring finite precision effects), but rather signifies the direction (order) of calculation:

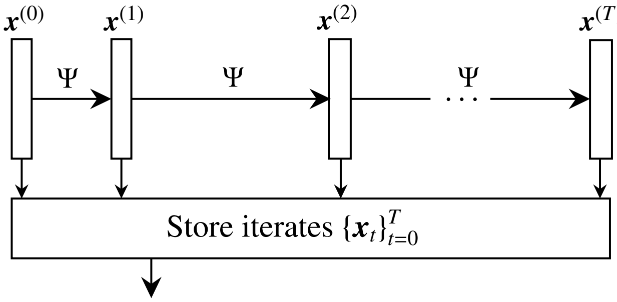
$$\prod_{s \in [T \leftarrow t]} \boldsymbol{A}_s := \boldsymbol{A}_T \,(\boldsymbol{A}_{T-1} \cdots (\boldsymbol{A}_{t+1}\,(\boldsymbol{A}_t)))$$

$$\prod_{s \in [T \rightarrow t]} \boldsymbol{A}_s := (((\boldsymbol{A}_T \boldsymbol{A}_{T-1}) \cdots)\, \boldsymbol{A}_{t+1})\, \boldsymbol{A}_t.$$

We use a similar arrow notation to denote the order that terms are computed for sums; as above, the order is only important for computational considerations and does not affect the final result.

Using this notation, the reverse gradient calculation of (D.1) is

$$\nabla_{\boldsymbol{\gamma}} \ell(\boldsymbol{\gamma}\,; \boldsymbol{x}^{(T)}) + \sum_{t \in [T \rightarrow 1]} \boldsymbol{J}_t{}' \left(\prod_{s \in [(t+1) \leftarrow T]} \boldsymbol{H}'_s \right) \nabla_{\boldsymbol{x}} \ell(\boldsymbol{\gamma}\,; \boldsymbol{x}^{(T)}). \tag{D.2}$$



```
$\boldsymbol{\Delta} = \nabla_{\boldsymbol{x}}\ell\left(\boldsymbol{\gamma}\,;\,\boldsymbol{x}^{(T)}\right)$
$\boldsymbol{r} = 0$
For $t = T-1 \ldots 0$
    Use $\boldsymbol{x}^{(t)}$ to compute $\boldsymbol{J}_{t+1}$ and $\boldsymbol{H}_{t+1}$
    $\boldsymbol{r} = \boldsymbol{r} + \boldsymbol{J}_{t+1}'\boldsymbol{\Delta}$
    $\boldsymbol{\Delta} = \boldsymbol{H}_{t+1}'\boldsymbol{\Delta}$
```

Figure D.1: Reverse mode computation of the unrolled gradient from (D.1). The first gradient computation requires $\boldsymbol{x}^{(T)}$, so all computations occur after the lower-level optimization algorithm is complete. The final gradient is $\nabla\ell(\boldsymbol{\gamma}) = \nabla_{\boldsymbol{\gamma}}\ell(\boldsymbol{\gamma}\,;\boldsymbol{x}^{(T)}) + \boldsymbol{r}$.

This expression requires $\prod_{s\in[(T+1)\leftarrow T]}\boldsymbol{H}_s' = \boldsymbol{I}$, because $\boldsymbol{H}_{T+1}$ is not defined. For example, for $T=3$, we have

$$\nabla_{\boldsymbol{\gamma}}\ell(\boldsymbol{\gamma}\,;\boldsymbol{x}^{(3)}) + \underbrace{\boldsymbol{J}_3'(\boldsymbol{I})\boldsymbol{g}}_{t=3} + \underbrace{\boldsymbol{J}_2'\left(\boldsymbol{H}_3'\right)\boldsymbol{g}}_{t=2} + \underbrace{\boldsymbol{J}_1'\left(\boldsymbol{H}_2'\boldsymbol{H}_3'\right)\boldsymbol{g}}_{t=1},$$

where $\boldsymbol{g}$ is shorthand for $\nabla_{\boldsymbol{x}}\ell(\boldsymbol{\gamma}\,;\boldsymbol{x}^{(T)})$ here. This version is called reverse as all computations (arrows) begin at the end, T.

The primary benefit of the reverse mode comes from the ability to group $\nabla_{\boldsymbol{x}}\ell(\boldsymbol{\gamma}\,;\boldsymbol{x}^{(T)})$ with the right-most $\boldsymbol{H}_T$, such that all products are matrix-vector products, as seen in Fig. D.1 Further, one can save the matrix-vector products for use during the next iteration and avoid duplicating the computation. Continuing the example for $T=3$, we have

$$\nabla_{\boldsymbol{\gamma}}\ell(\boldsymbol{\gamma}\,;\boldsymbol{x}^{(3)}) + \underbrace{\boldsymbol{J}_3'(\boldsymbol{I})\boldsymbol{g}}_{t=1} + \underbrace{\boldsymbol{J}_2'(\overbrace{\boldsymbol{H}_3'\boldsymbol{g}}^{\boldsymbol{\Delta}})}_{t=2} + \underbrace{\boldsymbol{J}_1'(\boldsymbol{H}_2'\overbrace{\left(\boldsymbol{H}_3'\boldsymbol{g}\right)}^{\boldsymbol{\Delta}})}_{t=3},$$

where one only needs to compute $\boldsymbol{\Delta}$ once. This ability to rearrange the parenthesis to compute matrix-vector products greatly decreases the computational requirement compared to matrix-matrix products. Excluding the costs of the optimization algorithm steps and forming the $\boldsymbol{H}_s$ and $\boldsymbol{J}_t$ matrices (these costs will be the same in the forward mode computation), reverse mode requires $\mathcal{O}(T)$ Hessian-vector multiplies and $\mathcal{O}(TNR)$ additional multiplies. The trade-off is that reverse mode requires storing all T iterates, $\boldsymbol{x}^{(t)}$, so that one can compute the corresponding Hessians and Jacobians from them as needed, and thus has a memory complexity $\mathcal{O}(TN)$.

The forward mode calculation of (D.1), depicted in Fig. D.2, has all computations (arrows) starting at the earlier

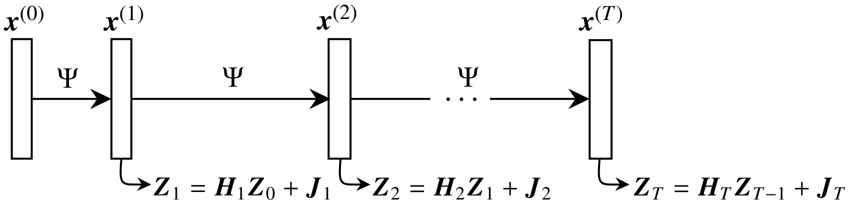

$\boldsymbol{x}^{(0)}$
$\boldsymbol{x}^{(1)}$
$\boldsymbol{x}^{(2)}$
$\boldsymbol{x}^{(T)}$
Ψ
Ψ
Ψ
$\cdots$
$\boldsymbol{Z}_1 = \boldsymbol{H}_1 \boldsymbol{Z}_0 + \boldsymbol{J}_1$
$\boldsymbol{Z}_2 = \boldsymbol{H}_2 \boldsymbol{Z}_1 + \boldsymbol{J}_2$
$\boldsymbol{Z}_T = \boldsymbol{H}_T \boldsymbol{Z}_{T-1} + \boldsymbol{J}_T$

Figure D.2: Forward mode computation of the unrolled gradient from (D.3). The intermediate computation matrix, $\boldsymbol{Z}$, is initialized to zero ($\boldsymbol{Z}_0 = \boldsymbol{0}$) then updated every iteration. The final gradient is $\nabla \ell(\boldsymbol{\gamma}) = \nabla_{\boldsymbol{\gamma}} \ell(\boldsymbol{\gamma}\,; \boldsymbol{x}^{(T)}) + \boldsymbol{Z}_T' \nabla_{\boldsymbol{x}} \ell(\boldsymbol{\gamma}\,; \boldsymbol{x}^{(T)})$.

iterate:

$$\nabla_{\boldsymbol{\gamma}} \ell(\boldsymbol{\gamma}\,; \boldsymbol{x}^{(T)}) + \left(\sum_{t \in [1 \to T]} \left(\prod_{s \in [T \leftarrow (t+1)]} \boldsymbol{H}_s \right) \boldsymbol{J}_t \right)' \nabla_{\boldsymbol{x}} \ell(\boldsymbol{\gamma}\,; \boldsymbol{x}^{(T)}). \tag{D.3}$$

As before, $\boldsymbol{H}_{T+1}$ is not defined, so we take $\prod_{s \in [T \leftarrow (T+1)]} \boldsymbol{H}_s = \boldsymbol{I}$. For example, for $T = 3$ we have

$$\nabla_{\boldsymbol{\gamma}} \ell(\boldsymbol{\gamma}\,; \boldsymbol{x}^{(T)}) + \left(\underbrace{((\boldsymbol{H}_3 \boldsymbol{H}_2) \boldsymbol{J}_1)'}_{t=1} + \underbrace{((\boldsymbol{H}_3) \boldsymbol{J}_2)'}_{t=2} + \underbrace{((\boldsymbol{I}) \boldsymbol{J}_3)'}_{t=3} \right) \boldsymbol{g}.$$

How the forward mode avoids storing $\boldsymbol{x}$ iterates is evident after rearranging the parenthesis to avoid duplicate calculations, as illustrated in Fig. D.2. Continuing the example for $T = 3$, we have

$$\nabla_{\boldsymbol{\gamma}} \ell(\boldsymbol{\gamma}\,; \boldsymbol{x}^{(T)}) + \left[\underbrace{\boldsymbol{H}_3 \left(\overbrace{\boldsymbol{H}_2 \underbrace{(\boldsymbol{H}_1 \cdot \boldsymbol{0} + \boldsymbol{J}_1)}_{\boldsymbol{Z}_1} + \boldsymbol{J}_2}^{\boldsymbol{Z}_2} \right) + \boldsymbol{J}_3}_{\boldsymbol{Z}_3} \right]' \boldsymbol{g},$$

where $\boldsymbol{Z}_s = \boldsymbol{H}_s \boldsymbol{Z}_{s-1} + \boldsymbol{J}_s \in \mathbb{F}^{N \times R}$ stores the intermediate calculations. The above formula also illustrates why $\boldsymbol{H}_1$ is not needed in (10.17); $\nabla_{\boldsymbol{\gamma}} \boldsymbol{x}^{(0)} = \boldsymbol{0}$ is the last element from applying the chain rule.

There is no way to rearrange the terms in the forward mode formula to achieve matrix-vector products (while preserving the computation order). Therefore, the computation requirement is much higher at $\mathcal{O}(TR)$ Hessian-vector multiplications. The corresponding benefit of the forward mode method is that it does not require storing iterates, thus decreasing (in the common case when $T > R$) the memory requirement to $\mathcal{O}(NR)$ for storing the intermediate matrix $\boldsymbol{Z}_s$ during calculation.

As with the minimizer approach in Section 10.1.1, the computational complexity of the unrolled approach is lower than the generic bound when we consider the specific example of learning convolutional filters according to (Ex). Nevertheless, the general comparison that reverse mode takes more memory but less computation holds true. See Tab. 10.2 for a comparison of the computational and memory complexities.

APPENDIX E

Additional Running Example Results

This appendix is adapted from Crockett and Fessler [11, App. C].

This appendix derives some results that are relevant to the running example used throughout Part 6.3 of this dissertation.

E.1 Derivatives for Convolutional Filters

This section proves the result

$$\frac{\partial}{\partial c_{\boldsymbol{s}}} \left(\tilde{\boldsymbol{c}}_k \circledast f.(\boldsymbol{c}_k \circledast \boldsymbol{x})\right) = f.(\boldsymbol{c}_k \circledast \boldsymbol{z}^{\langle \boldsymbol{s} \rangle}) + \tilde{\boldsymbol{c}}_k \circledast \left(\dot{f}.(\boldsymbol{c}_k \circledast \boldsymbol{x}) \odot \boldsymbol{x}^{\langle -\boldsymbol{s} \rangle}\right), \tag{E.1}$$

when considering $\mathbb{F} = \mathbb{R}$. This equation is key to finding derivatives of the lower-level cost function in (Ex) with respect to the filter coefficients.

To simplify notation, we drop the indexing over k, so $\boldsymbol{c}$ is a single filter and $c_{\boldsymbol{s}}$ denotes the $\boldsymbol{s}$th element in the filter for $\boldsymbol{s} \in \mathbb{Z}^D$. Here, $\boldsymbol{s}$ indexes every dimension of $\boldsymbol{c}$, *e.g.*, for a two-dimensional filter, we could equivalently write $\boldsymbol{s}$ as $\langle s_1, s_2 \rangle$. Recall that the notation $\tilde{\boldsymbol{c}}$ signifies a reversed version of $\boldsymbol{c}$, as needed for the adjoint of convolution.

Define the notation $\boldsymbol{x}^{\langle \boldsymbol{i} \rangle}$ as the vector $\boldsymbol{x}$ circularly shifted according to the index $\boldsymbol{i}$. Thus, if $\boldsymbol{x}$ is 0-indexed and we use circular indexing,

$$(\boldsymbol{x}^{\langle \boldsymbol{s} \rangle})_{\boldsymbol{i}} = \boldsymbol{x}_{\boldsymbol{i}-\boldsymbol{s}}.$$

As two examples,

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{N-1} \\ x_N \end{bmatrix} \to \boldsymbol{x}^{\langle \text{-}1 \rangle} = \begin{bmatrix} x_2 \\ x_3 \\ \vdots \\ x_N \\ x_1 \end{bmatrix},$$

and, in two dimensions, if $\boldsymbol{X} \in \mathbb{F}^{M \times N}$

$$\boldsymbol{X}^{\langle 1,2 \rangle} = \begin{bmatrix} x_{M,N-1} & x_{M,N} & x_{M,1} & \dots & x_{M,3} \\ x_{1,N-1} & x_{1,N} & x_{1,1} & \dots & x_{1,3} \\ x_{2,N-1} & x_{2,N} & x_{2,1} & \dots & x_{2,3} \\ \vdots & & \ddots & & \vdots \\ x_{M-1,N-1} & x_{M-1,N} & x_{M-1,1} & \dots & x_{M-1,3} \end{bmatrix}.$$

This circular shift notation is useful in the derivation and statement of the desired gradient.

Define $\boldsymbol{z} = \boldsymbol{c} \circledast \boldsymbol{x}$, where $\boldsymbol{c}$ and $\boldsymbol{x}$ are both N-dimensional. By the definition of convolution, $\boldsymbol{z}$ is given by

$$\boldsymbol{z} = \sum_{i_1} \cdots \sum_{i_N} c_{i_1,\dots,i_N} \boldsymbol{x}^{\langle \text{-}i_1,\dots,\text{-}i_N \rangle} := \sum_{i_1,\dots,i_N} c_{i_1,\dots,i_N} \boldsymbol{x}^{\langle \text{-}\boldsymbol{i} \rangle},$$

where, for each sum, the indexing variable i_n iterates over the size of $\boldsymbol{c}$ in the ith dimension and we simplify the index for circularly shifting vectors, $i_1, \dots, i_N$, as simply $\langle \boldsymbol{i} \rangle$. This expression shows that the derivative of $\boldsymbol{c} \circledast \boldsymbol{x}$ with respect to the $\boldsymbol{s}$th filter coefficient is the -$\boldsymbol{s}$th coefficient in $\boldsymbol{x}$, *i.e.*,

$$\frac{\partial}{\partial c_{\boldsymbol{s}}} (\boldsymbol{c} \circledast \boldsymbol{x}) = \boldsymbol{x}^{\langle -\boldsymbol{s} \rangle}. \tag{E.2}$$

We can now find the partial derivative of interest:

$$\begin{aligned} \tilde{\boldsymbol{c}} \circledast f.(\boldsymbol{z}) &= \sum_{i_1,\dots,i_N} [\tilde{\boldsymbol{c}}]_{i_1,\dots,i_N} f.(\boldsymbol{z})^{\langle \text{-}\boldsymbol{i} \rangle} && \text{by the convolution formula} \\ &= \sum_{i_1,\dots,i_N} [\tilde{\boldsymbol{c}}]_{i_1,\dots,i_N} f. \left(\boldsymbol{z}^{\langle \text{-}\boldsymbol{i} \rangle} \right) && \text{since } f \text{ operates point-wise} \\ &= \sum_{i_1,\dots,i_N} c_{\text{-}i_1,\dots,\text{-}i_N} f. \left(\boldsymbol{z}^{\langle \text{-}\boldsymbol{i} \rangle} \right) && \text{by definition of } \tilde{\boldsymbol{c}} \\ &= \sum_{i_1,\dots,i_N} c_{i_1,\dots,i_N} f. \left(\boldsymbol{z}^{\langle \boldsymbol{i} \rangle} \right) && \text{reverse summation order.} \end{aligned}$$

Recall that $\boldsymbol{z}$ is a function of $c_{\boldsymbol{s}}$. Therefore, using the chain rule to take the derivative,

$$\begin{aligned} \frac{\partial}{\partial c_{\boldsymbol{s}}} \left(\tilde{\boldsymbol{c}} \circledast f.(\boldsymbol{z}) \right) &= f.(\boldsymbol{z}^{\langle s \rangle}) + \sum_{i_1} \cdots \sum_{i_N} c_{i_1,\dots,i_N} \dot{f}.(\boldsymbol{z}^{\langle i_1,\dots,i_N \rangle}) \odot \nabla_{c_s} \left(\boldsymbol{z}^{\langle \boldsymbol{i} \rangle} \right) \\ &= f.(\boldsymbol{z}^{\langle \boldsymbol{s} \rangle}) + \sum_{i_1} \cdots \sum_{i_N} [\tilde{\boldsymbol{c}}]_{\text{-}i_1,\dots,\text{-}i_N} \dot{f}.(\boldsymbol{z}^{\langle i_1,\dots,i_N \rangle}) \odot \boldsymbol{x}^{\langle \boldsymbol{i} - \boldsymbol{s} \rangle}, \end{aligned}$$

where the second equality follows from (E.2) and the definition of $\tilde{\boldsymbol{c}}$. Recognizing the convolution formula in the second summand, the expression can be simplified to

$$f.(\boldsymbol{z}^{\langle \boldsymbol{s} \rangle}) + \tilde{\boldsymbol{c}} \circledast \left(\dot{f}.(\boldsymbol{z}) \odot \boldsymbol{x}^{\langle -\boldsymbol{s} \rangle} \right).$$

This proves the claim. Note that the provided formula is for a single element in $\boldsymbol{c}$. One can concatenate the partial derivative result for each value of $\boldsymbol{s}$ to get the full Jacobian.

E.2 Evaluating Assumptions for the Running Example

To better understand the upper-level assumptions Aℓ 1-Aℓ 3 and lower-level assumptions AΦ 1-AΦ 6 in Section 10.2.3.1, this section examines whether the filter learning example (Ex) meets each assumption.

E.2.1 Upper-level Loss Assumptions

Recall the upper-level loss function in (Ex) is squared error:

$$\ell(\boldsymbol{\gamma}\,;\,\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{x} - \boldsymbol{x}^{\text{true}}\|_2^2, \tag{E.3}$$

where ℓ is typically evaluated at $\boldsymbol{x} = \hat{\boldsymbol{x}}(\boldsymbol{\gamma})$.

The loss function (E.3) satisfies Aℓ 1. Because there is no dependence on $\boldsymbol{\gamma}$ in the upper-level, $L_{\boldsymbol{x},\nabla_{\boldsymbol{\gamma}}\ell} = 0$. The gradient with respect to $\boldsymbol{x}$ is $\nabla_{\boldsymbol{x}}\ell(\boldsymbol{\gamma}\,;\,\boldsymbol{x}) = \boldsymbol{x} - \boldsymbol{x}^{\text{true}}$, so $L_{\boldsymbol{x},\nabla_{\boldsymbol{x}}\ell} = 1$.

The norm of the upper-level gradient with respect to $\boldsymbol{x}$,

$$\|\nabla_{\boldsymbol{x}}\ell(\boldsymbol{\gamma}\,;\,\boldsymbol{x})\| = \left\|\boldsymbol{x} - \boldsymbol{x}^{\text{true}}\right\|,$$

can grow arbitrarily large, so condition Aℓ2 is not met in general. However, in most applications, one can assume an upper bound (possibly quite large) on the elements of $\boldsymbol{x}^{\text{true}}$ and impose that bound as a box constraint when computing $\hat{\boldsymbol{x}}$. Then the triangle inequality provides a bound on $\left\|\boldsymbol{x} - \boldsymbol{x}^{\text{true}}\right\|$ for all $\boldsymbol{x}$ within the constraint box.

Finally, Aℓ3 is met by any loss function, including (E.3), that lacks cross terms between $\boldsymbol{x}$ and $\boldsymbol{\gamma}$. We are unaware of any bilevel method papers using such cross terms.

E.2.2 Lower-level Cost Assumptions

One property used below in many of the bounds for the lower-level cost function is that

$$\sigma_1(\boldsymbol{C}_k) = \|\boldsymbol{c}_k\|_1\,, \tag{E.4}$$

where $\sigma_1(\cdot)$ is a function that returns the first singular value of its matrix argument. This property follows from Young's inequality and is related to bounded-input bounded-output stability of linear and time invariant systems [345].

As with the upper-level assumptions considered above, (Ex) meets the lower-level assumptions AΦ1-AΦ6 if we impose additional constraints on the maximum norm of variables. In addition to bounding the elements in $\boldsymbol{x}$, as we did to ensure Aℓ 2, imposing bounds on $\|\boldsymbol{c}_k\|$ and $|\beta_k|$ is sufficient to meet all the lower-level assumptions. We now examine each condition individually.

Recall from (Ex) that the example lower-level cost function is

$$\hat{\boldsymbol{x}}(\boldsymbol{\gamma}) = \underset{\boldsymbol{x}\in\mathbb{F}^N}{\operatorname{argmin}}\, \frac{1}{2}\,\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2 + e^{\beta_0}\sum_{k=1}^{K} e^{\beta_k}\boldsymbol{1}'\phi.(\boldsymbol{c}_k \circledast \boldsymbol{x}; \epsilon),$$

where ϕ is a corner-rounded 1-norm (CR1N).

As described in Section 10.1.1, the minimizer approach requires Φ to be twice differentiable. Thus, Φ satisfies AΦ1. This condition limits the choices of ϕ to twice differentiable functions.

Considering AΦ2, the gradient of Φ with respect to $\boldsymbol{x}$ is Lipschitz continuous in $\boldsymbol{x}$ if the norm of the Hessian,

$\|\nabla_{\boldsymbol{xx}} \Phi(\boldsymbol{x}\,;\boldsymbol{\gamma})\|_2$, is bounded. Using (10.9) and assuming the Lipschitz constant of the derivative of ϕ is $L_{\dot{\phi}}$ (for (CR1N), $L_{\dot{\phi}} = \frac{1}{\epsilon}$), a Lipschitz constant for $\nabla_{\boldsymbol{x}} \Phi$ is

$$\begin{aligned} L_{\boldsymbol{x},\nabla_{\boldsymbol{x}}\Phi} &= \sigma_1^2(\boldsymbol{A}) + L_{\dot{\phi}} e^{\beta_0} \sum_k e^{\beta_k} \sigma_1(\boldsymbol{C}_k' \boldsymbol{C}_k) \\ &= \sigma_1^2(\boldsymbol{A}) + L_{\dot{\phi}} e^{\beta_0} \sum_k e^{\beta_k} \|\boldsymbol{c}_k\|_1^2 \text{ by (E.4).} \end{aligned} \tag{E.5}$$

The Lipschitz constant $L_{\boldsymbol{x},\nabla_{\boldsymbol{x}}\Phi}$ depends on the values in $\boldsymbol{\gamma}$ and therefore does not strictly satisfy AΦ2. Here if β_0, β_k, and $\boldsymbol{c}_k$ have upper bounds, then one can upper bound $L_{\boldsymbol{x},\nabla_{\boldsymbol{x}}\Phi}$. All of the bounds below have similar considerations.

To consider the strong convexity condition in AΦ3, we consider the Hessian,

$$\nabla_{\boldsymbol{xx}} \Phi(\boldsymbol{x}\,;\boldsymbol{\gamma}) = \underbrace{\boldsymbol{A}'\boldsymbol{A}}_{\text{From data-fit term}} + \underbrace{e^{\beta_0} \sum_k e^{\beta_k} \boldsymbol{C}_k' \operatorname{diag}(\ddot{\phi}.(\boldsymbol{c}_k \circledast \boldsymbol{x}))\boldsymbol{C}_k}_{\text{From regularizer}}\,. \tag{E.6}$$

We assume that $\ddot{\phi}(z) \geq 0\,\forall z$, as is the case for the corner rounded 1-norm. If $\boldsymbol{A}'\boldsymbol{A}$ is positive-definite with $\sigma_N(\boldsymbol{A}'\boldsymbol{A}) > 0$ (this is equivalent to $\boldsymbol{A}$ having full column rank), then the Hessian is positive-definite and $\mu_{\boldsymbol{x},\Phi} = \sigma_N^2(\boldsymbol{A})$ suffices as a strong convexity parameter. In applications like compressed sensing, $\boldsymbol{A}$ does not have full column rank. In such cases, $\sigma_N(\boldsymbol{A}'\boldsymbol{A}) = 0$ and as $e^{\beta_0} \to 0$ the regularizer term vanishes, so there does not exist any universal $\mu_{\boldsymbol{x},\Phi} > 0$ for all $\boldsymbol{\gamma} \in \mathbb{F}^R$, so the strong convexity condition AΦ3 is not satisfied. However, as discussed in Section 10.1.1.3, the condition may hold in practice for many values of $\boldsymbol{\gamma}$. How to adapt the complexity theory to rigorously address these subtleties is an open question.

The fourth condition, AΦ4, is that $\nabla_{\boldsymbol{xx}} \Phi(\boldsymbol{x}\,;\boldsymbol{\gamma})$ and $\nabla_{\boldsymbol{\gamma x}} \Phi(\boldsymbol{x}\,;\boldsymbol{\gamma})$ are Lipschitz continuous with respect to $\boldsymbol{x}$ for all $\boldsymbol{\gamma}$. For the first part part, a Lipschitz constant results from bounding the difference in the Hessian evaluated at two points, $\boldsymbol{x}^{(1)}$ and $\boldsymbol{x}^{(2)}$:

$$\left\|\nabla_{\boldsymbol{xx}} \Phi(\boldsymbol{x}^{(1)}\,;\boldsymbol{\gamma}) - \nabla_{\boldsymbol{xx}} \Phi(\boldsymbol{x}^{(2)}\,;\boldsymbol{\gamma})\right\|_2 = \left\| e^{\beta_0} \sum_k e^{\beta_k} \boldsymbol{C}_k' \operatorname{diag}(\ddot{\phi}.(\boldsymbol{c}_k \circledast \boldsymbol{x}^{(1)}) - \ddot{\phi}(\boldsymbol{c}_k \circledast \boldsymbol{x}^{(2)}))\boldsymbol{C}_k \right\|_2 .$$

Since every element of $\ddot{\phi}$ is bounded in $(0, L_{\dot{\phi}})$, the difference between any two evaluations of $\ddot{\phi}$ is at most $L_{\dot{\phi}}$. Thus

$$\begin{aligned} \left\|\nabla_{\boldsymbol{xx}} \Phi(\boldsymbol{x}^{(1)}\,;\boldsymbol{\gamma}) - \nabla_{\boldsymbol{xx}} \Phi(\boldsymbol{x}^{(2)}\,;\boldsymbol{\gamma})\right\|_2 &\leq e^{\beta_0} L_{\dot{\phi}} \sum_k e^{\beta_k} \left\|\boldsymbol{C}_k' \boldsymbol{C}_k\right\|_2 \\ &\leq e^{\beta_0} L_{\dot{\phi}} \sum_k e^{\beta_k} \|\boldsymbol{c}_k\|_1^2 \,. \end{aligned}$$

The final simplification again uses (E.4). Thus,

$$L_{\boldsymbol{x},\nabla_{\boldsymbol{xx}}\Phi} = e^{\beta_0} L_{\dot{\phi}} \sum_k e^{\beta_k} \|\boldsymbol{c}_k\|_1^2 \,.$$

For the second part of AΦ4, we must look at the tuning parameters and filter coefficients separately. When considering learning a tuning parameter, β_k,

$$\nabla_{\beta_k \boldsymbol{x}} \Phi(\boldsymbol{x}\,;\boldsymbol{\gamma}) = e^{\beta_0 + \beta_k} \boldsymbol{C}_k' \dot{\phi}.(\boldsymbol{C}_k \boldsymbol{x}).$$

To find a Lipschitz constant, consider the Jacobian:

$$\nabla_{\boldsymbol{x}} \left(\nabla_{\beta_k \boldsymbol{x}} \Phi(\boldsymbol{x}\,;\boldsymbol{\gamma}) \right) = e^{\beta_0+\beta_k} \boldsymbol{C}_k' \mathrm{diag}(\ddot{\phi}.(\boldsymbol{C}_k \boldsymbol{x})) \boldsymbol{C}_k.$$

A Lipschitz constant of $\nabla_{\beta_k \boldsymbol{x}} \Phi(\boldsymbol{x}\,;\boldsymbol{\gamma})$ is given by the bound on the norm of this matrix (we chose to use the matrix 2-norm, also called the spectral norm). Using similar steps as above to simplify the expression, $L_{\boldsymbol{x},\nabla_{\beta_k \boldsymbol{x}}\Phi} = e^{\beta_0+\beta_k} L_{\dot{\phi}} \left\|\boldsymbol{c}_k\right\|_1^2$.

When considering learning the $\boldsymbol{s}$th element of the kth filter,

$$\begin{aligned} \nabla_{c_{k,s} \boldsymbol{x}} \Phi(\boldsymbol{x}\,;\boldsymbol{\gamma}) &= e^{\beta_0+\beta_k} \left(\dot{\phi}.((\boldsymbol{C}_k \boldsymbol{x})^{\langle s \rangle}) + \boldsymbol{C}_k' \left(\ddot{\phi}.(\boldsymbol{C}_k \boldsymbol{x}) \odot \boldsymbol{x}^{\langle \text{-}s \rangle} \right) \right) \\ &= e^{\beta_0+\beta_k} \left(\underbrace{\dot{\phi}.(\boldsymbol{R}_1 \boldsymbol{C}_k \boldsymbol{x})}_{\text{Expression 1}} + \underbrace{\boldsymbol{C}_k' \left(\ddot{\phi}.(\boldsymbol{C}_k \boldsymbol{x}) \odot \boldsymbol{R}_2 \boldsymbol{x} \right)}_{\text{Expressions 2-3}} \right) \in \mathbb{F}^N, \end{aligned}$$

where $\boldsymbol{R}_1$ and $\boldsymbol{R}_2$ are rotation matrices that depends on $\boldsymbol{s}$ such that $\boldsymbol{R}_1 \boldsymbol{x} = \boldsymbol{x}^{\langle s \rangle}$ and $\boldsymbol{R}_2 \boldsymbol{x} = \boldsymbol{x}^{\langle \text{-}s \rangle}$. For taking the gradient, it is convenient to note that the last term can be expressed in multiple ways:

$$\ddot{\phi}.(\boldsymbol{C}_k \boldsymbol{x}) \odot \boldsymbol{x}^{\langle \text{-}s \rangle} = \underbrace{\mathrm{diag}(\ddot{\phi}.(\boldsymbol{C}_k \boldsymbol{x})) \boldsymbol{R}_2 \boldsymbol{x}}_{\text{Expression 2}} = \underbrace{\mathrm{diag}(\boldsymbol{R}_2 \boldsymbol{x}) \ddot{\phi}.(\boldsymbol{C}_k \boldsymbol{x})}_{\text{Expression 3}}.$$

Using the alternate expressions to perform the chain rule with respect to the $\boldsymbol{x}$ term that is not in the diag$(\cdot)$ statement, the gradient with respect to $\boldsymbol{x}$ is:

$$\begin{aligned} \nabla_{\boldsymbol{x}} \left(\nabla_{c_{k,s} \boldsymbol{x}} \Phi(\boldsymbol{x}\,;\boldsymbol{\gamma}) \right) = e^{\beta_0+\beta_k} (& \underbrace{\boldsymbol{C}_k' \boldsymbol{R}_1' \mathrm{diag}(\ddot{\phi}.(\boldsymbol{R}_1 \boldsymbol{C}_k \boldsymbol{x}))}_{\text{Expression 1}} \\ & + \underbrace{\boldsymbol{C}_k' \mathrm{diag}(\ddot{\phi}.(\boldsymbol{C}_k \boldsymbol{x})) \boldsymbol{R}_2}_{\text{Expression 2}} \\ & + \underbrace{\boldsymbol{C}_k' \mathrm{diag}(\dddot{\phi}(\boldsymbol{C}_k \boldsymbol{x})) \mathrm{diag}(\boldsymbol{R}_2 \boldsymbol{x})' \boldsymbol{C}_k}_{\text{Expression 3}}). \end{aligned}$$

The bound on the spectral norm of the first and second expressions are both $\sigma_1(\boldsymbol{C}_k) L_{\dot{\phi}}$ because, for any $\boldsymbol{z} \in \mathbb{F}^N$,

$$\left\|\mathrm{diag}(\ddot{\phi}.(\boldsymbol{z}))\right\|_2 \leq \max_z |\ddot{\phi}(z)| = L_{\dot{\phi}}.$$

The third expression is bounded by $\sigma_1^2(\boldsymbol{C}_k) \left\|\boldsymbol{x}\right\|_2 L_{\ddot{\phi}}$, which requires a bound on the norm of $\boldsymbol{x}$, similar to Aℓ2. Summing the three expressions and including the tuning parameters gives the final Lipschitz constant

$$L_{\boldsymbol{x},\nabla_{c_{k,s} \boldsymbol{x}}\Phi} = e^{\beta_0+\beta_k} \sigma_1(\boldsymbol{C}_k)(2L_{\dot{\phi}} + \sigma_1(\boldsymbol{C}_k) L_{\ddot{\phi}} \left\|\boldsymbol{x}\right\|_2). \tag{E.7}$$

The fifth assumption, AΦ5 states that the mixed second gradient of Φ is bounded. For the tuning parameters, the mixed second gradient is given in (10.9) as

$$\nabla_{\beta_k \boldsymbol{x}} \Phi(\hat{\boldsymbol{x}}\,;\boldsymbol{\gamma}) = e^{\beta_0} e^{\beta_k} \tilde{\boldsymbol{c}}_k \circledast \dot{\phi}.(\boldsymbol{c}_k \circledast \hat{\boldsymbol{x}}).$$

The bound given in AΦ5 follows easily by considering that

$$\left\|\mathrm{diag}(\dot{\phi}.(\boldsymbol{c}_k \circledast \hat{\boldsymbol{x}}))\right\|_2 \leq \max_z |\dot{\phi}(z)| = L_{\phi}.$$

For a filter coefficient, the mixed second gradient is more complicated:

$$\nabla_{c_{k,s}\boldsymbol{x}} \Phi(\hat{\boldsymbol{x}}\,;\boldsymbol{\gamma}) = e^{\beta_0+\beta_k}\Big(\underbrace{\dot{\phi}.((\boldsymbol{c}_k \circledast \hat{\boldsymbol{x}})^{\langle \boldsymbol{s} \rangle})}_{\text{Bounded by } L_\phi} + \tilde{\boldsymbol{c}}_k \circledast \Big(\underbrace{\ddot{\phi}.(\boldsymbol{c}_k \circledast \hat{\boldsymbol{x}})}_{\text{Bounded by } L_{\dot{\phi}}} \odot \hat{\boldsymbol{x}}^{\langle \text{-}\boldsymbol{s} \rangle}\Big)\Big).$$

Assuming that the bounds L_ϕ and $L_{\dot{\phi}}$ exist (they are 1 and $\frac{1}{\epsilon}$ respectively for (CR1N)), a bound on the norm of the mixed gradient is

$$\|\nabla_{c_{k,s}\boldsymbol{x}} \Phi(\hat{\boldsymbol{x}}\,;\boldsymbol{\gamma})\|_2 \leq e^{\beta_0+\beta_k} \left(L_\phi + L_{\dot{\phi}} \, \|\boldsymbol{c}_k\|_1 \, \|\boldsymbol{x}\|_2\right).$$

The sixth assumption, AΦ6, is that $L_{\boldsymbol{\gamma},\nabla_{\boldsymbol{\gamma}\boldsymbol{x}}\Phi}$ and $L_{\boldsymbol{\gamma},\nabla_{\boldsymbol{x}\boldsymbol{x}}\Phi}$ exist. Lipschitz constants for the tuning parameters are

$$L_{\beta_k,\nabla_{\beta_k \boldsymbol{x}}\Phi} = e^{\beta_0+\beta_k} \, \|\boldsymbol{c}_k\|_1 \, L_\phi \text{ and } L_{\beta_k,\nabla_{\boldsymbol{x}\boldsymbol{x}}\Phi} = e^{\beta_0+\beta_k} \, \|\boldsymbol{c}_k\|_1^2 \, L_{\dot{\phi}}.$$

Using similar derivations as shown above, corresponding Lipschitz constants for the filter coefficients are

$$\begin{aligned} L_{c_{k,s},\nabla_{c_{k,s}\boldsymbol{x}}\Phi} &= e^{\beta_0+\beta_k} \left(L_\phi + \|\boldsymbol{x}\|_2 \left(L_{\dot{\phi}} + L_{\ddot{\phi}} \, \|\boldsymbol{c}_k\|_1 \, \|\boldsymbol{x}\|_2\right)\right) \\ L_{c_{k,s},\nabla_{\boldsymbol{x}\boldsymbol{x}}\Phi} &= e^{\beta_0+\beta_k} \left(2L_{\dot{\phi}} \, \|\boldsymbol{c}_k\|_1 + L_{\ddot{\phi}} \, \|\boldsymbol{c}_k\|_1^2 \, \|\boldsymbol{x}\|_2\right). \end{aligned}$$

This is the last lower-level condition in Section 10.2.3.1 for the single-loop and double-loop bilevel optimization method analysis.

APPENDIX F

Implementation Details

This appendix is presented Crockett and Fessler [11, App. D].

This appendix describes the experimental settings for the results in Fig. 7.3 and for the series of figures using the cameraman image (Fig. 10.3, Fig. 12.1, and Fig. 12.2). We first present the common settings; the following sub-sections detail any differences or additional settings. The code for all experiments is available on github [346].

The experiments consider the denoising problem ($\boldsymbol{A} = \boldsymbol{I}$) and use (CR1N) as the sparsifying function ϕ with $\epsilon = 0.01$. The training data is typically on the scale [0, 1] and noisy samples are generated from the clean training data using (7.4) with zero-mean Gaussian noise with a standard deviation of $\sigma = 25/255$, following [162].

The lower-level optimizer is the optimized gradient method (OGM) with gradient-based restart [295]. We calculate the step-size based on the Lipschitz constant of the lower-level gradient using (E.5) every upper-level iteration. Each experiment sets a maximum number of lower-level iterations, but the lower-level optimization will terminate early if it converges, defined as if $||\nabla_{\boldsymbol{x}} \Phi(\boldsymbol{x} \,; \boldsymbol{\gamma})|| < 10^{-5}$.

The upper-level optimizer follows the general structure of the double-loop procedure outlined in Alg. 5. To compute $\nabla \ell(\boldsymbol{\gamma})$, we use the minimizer formulation (10.8), with the conjugate gradient (CG) method to compute the Hessian-inverse-vector product (10.10). As suggested in [264], the initialization for the lower-level optimization is the estimated minimizer from the previous outer loop iteration, $\boldsymbol{x}^{(T)}(\boldsymbol{\gamma}^{(u-1)})$ and the initialization for the CG method is the solution from the previous CG iteration. Following [166] and other bilevel works, the experiments use Adam with the default parameters [292] to determine the size of the upper-level gradient descent; this choice avoids introducing the tuning parameter α_ℓ.

The learnable parameters include the filter coefficients and the tuning parameters β_k for $k \in [1, K]$. The experiments either use random or DCT filters to initialize $\boldsymbol{h}$. An initial grid search determines the tuning parameter β_0; β_k for $k \in [1, K]$ are initialized as 0 such that $e^{\beta_k} = 1$.

F.1 Vertical Bar Training Image

This section describes additional details for Fig. 7.3. This simple proof of concept used 50 lower-level iterations ($T = 50$) and 4,000 upper-level iterations ($U = 4,000$). The initial grid search for β_0 yielded -4.6.

When $\phi(z) = |z|$, one can absorb the kth filter's magnitude into the tuning parameter β_k because $||\boldsymbol{c}_k \circledast \boldsymbol{x}||_1 = ||\boldsymbol{c}_k||_2 \left\| \frac{1}{||\boldsymbol{c}_k||_2} \boldsymbol{c}_k \circledast \boldsymbol{x} \right\|_1$. When using (CR1N), this equality no longer holds, but

$$e^{\beta_0 + \beta_k} \, ||\boldsymbol{c}_k||_2 \tag{F.1}$$

still provides a reasonable approximation for the overall regularization strength for the kth filter. From left to right, the approximate regularization strengths of the filters in Fig. 7.3 are 0.77, 0.49, 0.17, and 0.05.

The learned filters reflect that the training data is constant along the columns. Visually, the filters resemble vertical (extended) finite differences. This matches our expectations as a filter that takes vertical finite differences will exactly sparsify the noiseless signal. Further, the maximum sum of the columns of the learned filters is 10^{-5}. In contrast, the sum of the rows of the learned filters varies from -2.6 to 3.0.

F.2 Cameraman Training Image

This section describes the experimental settings for Fig. 10.3, Fig. 12.2, and Fig. 12.1.

To reduce computation, we selected three 50×50 patches from the "cameraman" image in Fig. 12.2 to use as the training data. We hand selected the training patches to contain structure. Fig. F.1 shows the training image patches.

We set the lower-level initialization $\hat{\boldsymbol{x}}(\boldsymbol{\gamma}^{(0)})$ by optimizing the lower-level cost function until the norm of the gradient fell below a threshold for each training patch, *i.e.*, until $\frac{1}{\sqrt{N}} \left\| \nabla_{\boldsymbol{x}} \Phi \left(\hat{\boldsymbol{x}}_j(\boldsymbol{\gamma}^{(0)}) \,;\, \boldsymbol{\gamma}^{(0)} \right) \right\|_2 < 10^{-7}$ for $j \in [1, J]$. The lower-level optimizer consisted of 10 iterations of OGM [295].

As shown in Fig. 12.1, the initial filters are the 48 non-constant DCT filters of size 7×7. The initial grid search for β_0 yielded -4. In summary, the settings are $J = 3$, $N = 50 \cdot 50$, $S = 7 \cdot 7$, $K = 48$, $R = 48(49 + 1) = 2400$, $\beta_0 = -4$, $T = 10$, and $U = 10{,}000$.

Fig. 12.1 shows the learned filters. To visualize the filters when $\boldsymbol{\gamma}$ includes $\boldsymbol{h}$, Fig. 12.1c scales each learned filter $\hat{\boldsymbol{c}}_k$ to have unit norm. Fig. F.2 shows the learned filters with the effective regularization strength printed above each filter.

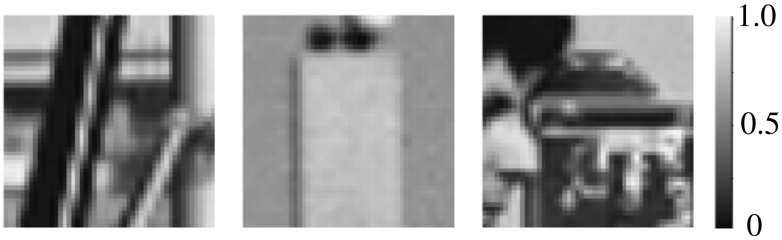


Figure F.1: Patches from the cameraman test images used as the training dataset.

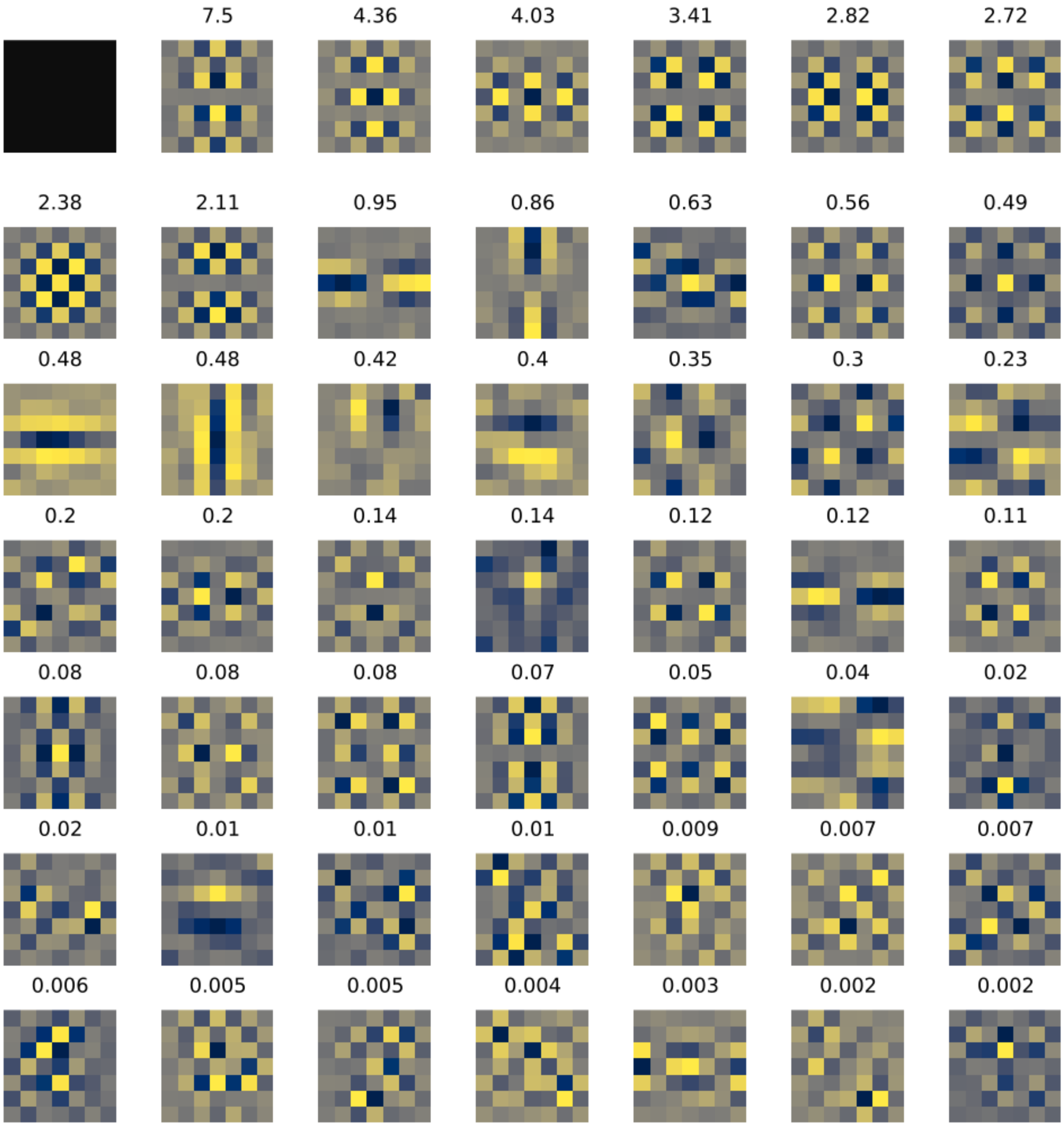


Figure F.2: Learned filers for (Ex) when $\boldsymbol{\gamma}$ includes $\boldsymbol{h}$ and $\boldsymbol{\beta}$, ordered by their effective regularization strength $e^{\beta_k}\,||\boldsymbol{c}_k||_2$, which is printed above each filter. This effective regularization does not include the influence of e^{β_0}, which is uniform across all filters.

BIBLIOGRAPHY

[1] K. Wage, J. Buck, C. Wright, and T. Welch, "The signals and systems concept inventory," *IEEE Transactions on Education*, vol. 48, no. 3, pp. 448–461, Aug. 2005, ISSN: 0018-9359. DOI: 10.1109/TE.2005.849746.

[2] S. A. Male, R. B. Togneri, and L. C. Jin, "Novice to postgraduate researcher perceptions of threshold concepts and capabilities in signal processing: Understanding students' and researchers' perspectives," *IEEE Signal Processing Magazine*, vol. 38, no. 3, pp. 30–36, May 2021, ISSN: 1053-5888, 1558-0792. DOI: 10.1109/MSP.2021.3055201.

[3] C. Crockett, C. J. Finelli, and H. C. Powell, "Work in progress: A longitudinal study of students' conceptual understanding of signals and systems," in *2020 ASEE Virtual Annual Conference Content Access Proceedings*, ASEE Conferences, Jun. 2020, p. 35 595. DOI: 10.18260/1-2--35595.

[4] C. Crockett and C. Finelli, "Factors influencing conceptual understanding in a signals and systems course," in *2021 ASEE Virtual Annual Conference Content Access*, Jul. 2021. [Online]. Available: https://peer.asee.org/37175.

[5] C. Crockett, H. C. Powell, and C. J. Finelli, "Conceptual understanding of signals and systems in senior undergraduate students," Submitted to: *IEEE Transactions on Education*, 2022.

[6] ——, "Factors influencing conceptual understanding of signals and systems of senior engineering students," Submitted to: *European Journal of Engineering Education*, 2022.

[7] Y. Eldar and G. Kutyniok, *Compressed sensing: Theory and applications*. Cambridge, 2012. DOI: 10.1017/CBO9780511794308.

[8] S. Dempe, "Annotated bibliography on bilevel programming and mathematical programs with equilibrium constraints," *Optimization*, vol. 52, no. 3, pp. 333–359, Jun. 2003, ISSN: 0233-1934, 1029-4945. DOI: 10.1080/0233193031000149894.

[9] C. Crockett, D. Hong, I. Y. Chun, and J. A. Fessler, "Incorporating handcrafted filters in convolutional analysis operator learning for ill-posed inverse problems," in *2019 IEEE 8th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing*, CAMSAP, Dec. 2019, pp. 316–320. DOI: 10.1109/CAMSAP45676.2019.9022669.

[10] C. Crockett and J. A. Fessler, "Motivating bilevel approaches to filter learning: A case study," in *2021 IEEE International Conference on Image Processing (ICIP)*, IEEE, Sep. 19, 2021, pp. 2803–2807, ISBN: 978-1-66544-115-5. DOI: 10.1109/ICIP42928.2021.9506489.

[11] ——, "Bilevel methods for image reconstruction," *Foundations and Trends® in Signal Processing*, vol. 15, no. 2-3, pp. 121–289, May 5, 2022, ISSN: 1932-8346, 1932-8354. DOI: 10.1561/2000000111.

[12] J. Henrich, S. J. Heine, and A. Norenzayan, "The weirdest people in the world?" *Behavioral and Brain Sciences*, vol. 33, no. 2-3, pp. 61–83, Jun. 2010, ISSN: 0140-525X, 1469-1825. DOI: 10.1017/S0140525X0999152X.

[13] (). "Journal of Women and Minorities in Science and Engineering: Author instructions," begell, [Online]. Available: https://www.begellhouse.com/forauthors/journals/journal-of-women-and-minorities-in-science-and-engineering.html.

[14] A. G. D. Holmes, "Researcher Positionality - A Consideration of Its Influence and Place in Qualitative Research - A New Researcher Guide," *Shanlax International Journal of Education*, vol. 8, no. 4, pp. 1–10, Sep. 1, 2020, ISSN: 2582-1334, 2320-2653. DOI: 10.34293/education.v8i4.3232.

[15] M. Borrego, K. A. Nguyen, C. Crockett, M. DeMonbrun, P. Shekhar, S. Tharayil, C. J. Finelli, R. S. Rosenberg, and C. Waters, "Systematic literature review of students' affective responses to active learning: Overview of results," in *2018 IEEE Frontiers in Education Conference (FIE)*, IEEE, Oct. 2018, pp. 1–7, ISBN: 978-1-5386-1174-6. DOI: 10.1109/FIE.2018.8659306.

[16] C. Crockett, C. J. Finelli, M. Demonbrun, K. A. Nguyen, S. Tharayil, P. Shekhar, and R. S. Rosenberg, "Common characteristics of high-quality papers studying student response to active learning," *International Journal of Engineering Education*, vol. 37, no. 2, 2021. [Online]. Available: https://www.ijee.ie/contents/c370221.html.

[17] P. Shekhar, M. Borrego, M. DeMonbrun, C. Finelli, C. Crockett, and K. Nguyen, "Negative student response to active learning in STEM classrooms: A systematic review of underlying reasons," *Journal of College Science Teaching*, vol. 49, no. 6, pp. 45–54, 2020, ISSN: 0047231X. [Online]. Available: https://www.nsta.org/journal-college-science-teaching/journal-college-science-teaching-julyaugust-2020/negative-student.

[18] K. A. Nguyen, M. Borrego, C. J. Finelli, M. DeMonbrun, C. Crockett, S. Tharayil, P. Shekhar, C. Waters, and R. Rosenberg, "Instructor strategies to aid implementation of active learning: A systematic literature review," *International Journal of STEM Education*, vol. 8, no. 9, p. 18, 2021. DOI: 10.1186/s40594-021-00270-7.

[19] J. Walther, N. W. Sochacka, and N. N. Kellam, "Quality in interpretive engineering education research: Reflections on an example study," *Journal of Engineering Education*, vol. 102, no. 4, pp. 626–659, Oct. 2013, ISSN: 10694730. DOI: 10.1002/jee.20029.

[20] S. N. Hesse-Biber, "Chapter 2: Paradigmatic approaches to qualitative research," in *The Practice of Qualitative Research: Engaging Students in the Research Process*, Third edition, SAGE, 2017, ISBN: 978-1-4522-6808-8.

[21] J. Willis, "Chapter 2: History and context of paradigm development," in *Foundations of Qualitative Research: Interpretive and Critical Approaches*, SAGE Publications, Inc., 2007, ISBN: 978-1-4129-2741-3. DOI: 10.4135/9781452230108.

[22] C. A. Capper, *Organizational Theory for Equity and Diversity: Leading Integrated, Socially Just Education*, 1st ed. Routledge, Oct. 17, 2018, ISBN: 978-1-315-81861-0. DOI: 10.4324/9781315818610.

[23] G. P. Wiggins and J. McTighe, *Understanding by Design*, Expanded 2nd ed. Association for Supervision and Curriculum Development, 2005, 370 pp., ISBN: 978-1-4166-0035-0.

[24] A. A. diSessa, “A History of Conceptual Change Research,” in *The Cambridge Handbook of the Learning Sciences*, ser. Cambridge Handbooks in Psychology, R. K. Sawyer, Ed., Cambridge University Press, 2005, pp. 265–282, ISBN: 978-0-511-81683-3. DOI: 10.1017/CBO9780511816833.017.

[25] J. Hiebert and P. Lefevre, “Conceptual and procedural knowledge in mathematics: An introductory analysis.,” in *Conceptual and Procedural Knowledge: The Case of Mathematics*. Lawrence Erlbaum Associates, Inc, 1986, pp. 1–27, ISBN: 0-89859-556-8 (Hardcover).

[26] R. A. Streveler, S. Brown, G. L. Herman, and D. Montfort, “Conceptual Change and Misconceptions in Engineering Education,” in *Cambridge Handbook of Engineering Education Research*, A. Johri and B. M. Olds, Eds., Cambridge University Press, 2013, pp. 83–102, ISBN: 978-1-139-01345-1. DOI: 10.1017/CBO9781139013451.008.

[27] B. Rittle-Johnson, “Promoting transfer: Effects of self-explanation and direct instruction,” *Child Development*, vol. 77, no. 1, pp. 1–15, Feb. 2006, ISSN: 0009-3920, 1467-8624. DOI: 10.1111/j.1467-8624.2006.00852.x.

[28] A. S. Rao, J. Fan, C. Brame, and B. Landman, “Improving conceptual understanding of signals and systems in undergraduate engineering students using collaborative in-class laboratory exercises,” in *2014 ASEE Annual Conference & Exposition Proceedings*, ASEE Conferences, Jun. 2014, pp. 24.715.1–24.715.14. DOI: 10.18260/1-2--20607.

[29] D. Montfort, S. Brown, and D. Pollock, “An investigation of students’ conceptual understanding in related sophomore to graduate-level engineering and mechanics courses,” *Journal of Engineering Education*, vol. 98, no. 2, pp. 111–129, Apr. 1, 2009, ISSN: 1069-4730. DOI: 10.1002/j.2168-9830.2009.tb01011.x.

[30] R. A. Streveler, T. A. Litzinger, R. L. Miller, and P. S. Steif, “Learning conceptual knowledge in the engineering sciences: Overview and future research directions,” *Journal of Engineering Education*, vol. 97, no. 3, pp. 279–294, Jul. 2008, ISSN: 10694730. DOI: 10.1002/j.2168-9830.2008.tb00979.x.

[31] W. Boles, D. Jayalath, and A. Goncher, “Categorising Conceptual Assessments under the Framework of Bloom’s Taxonomy,” p. 11, 2015.

[32] N. Salzman and J. Strobel, “Conceptual change in precollege engineering,” presented at the Research in Engineering Education Symposium (REES) 2011, 2011, p. 9.

[33] S. S. Haykin and B. Van Veen, *Signals and Systems*, 2nd ed. Wiley, 2002, 802 pp., ISBN: 978-0-471-16474-6.

[34] A. V. Oppenheim, A. S. Willsky, and S. H. Nawab, *Signals & Systems*, 2nd ed, ser. Prentice-Hall Signal Processing Series. Prentice Hall, 1997, 957 pp., ISBN: 978-0-13-814757-0.

[35] C. L. Phillips, J. M. Parr, and E. A. Riskin, *Signals, Systems, and Transforms*, 4th ed. Pearson/Prentice Hall, 2008, 774 pp., ISBN: 978-0-13-198923-8.

[36] W. C. Newstetter and M. D. Svinicki, “Learning Theories for Engineering Education Practice,” in *Cambridge Handbook of Engineering Education Research*, A. Johri and B. M. Olds, Eds., Cambridge University Press, 2014, pp. 29–46, ISBN: 978-1-107-01410-7. DOI: 10.1017/CBO9781139013451.005.

[37] P. W. Thompson, "Constructivism in Mathematics Education," in *Encyclopedia of Mathematics Education*, S. Lerman, Ed., Springer International Publishing, 2020, pp. 127–134, ISBN: 978-3-030-15789-0. DOI: 10.1007/978-3-030-15789-0_31.

[38] D. E. Rumelhart, "Schemata: The Building Blocks of Cognition," in *Theoretical Issues in Reading Comprehension*, Routledge, 1980, ISBN: 978-1-315-10749-3.

[39] S. Vosniadou and I. Skopeliti, "Conceptual change from the framework theory side of the fence," *Science & Education*, vol. 23, no. 7, pp. 1427–1445, Jul. 1, 2014, ISSN: 1573-1901. DOI: 10.1007/s11191-013-9640-3.

[40] M. T. H. Chi, "Commonsense Conceptions of Emergent Processes: Why Some Misconceptions Are Robust," *Journal of the Learning Sciences*, vol. 14, no. 2, pp. 161–199, Apr. 2005, ISSN: 1050-8406, 1532-7809. DOI: 10.1207/s15327809jls1402_1.

[41] J. B. Henderson, E. Langbeheim, and M. Chi, "Addressing robust misconceptions through the ontological distinction between sequential and emergent processes," *Converging Perspectives on Conceptual Change: Mapping an Emerging Paradigm in the Learning Sciences*, pp. 26–33, Jan. 1, 2017. DOI: 10.4324/9781315467139.

[42] A. A. diSessa, "A Friendly Introduction to "Knowledge in Pieces": Modeling Types of Knowledge and Their Roles in Learning," in *Invited Lectures from the 13th International Congress on Mathematical Education*, G. Kaiser, H. Forgasz, M. Graven, A. Kuzniak, E. Simmt, and B. Xu, Eds., Springer International Publishing, 2018, pp. 65–84, ISBN: 978-3-319-72170-5.

[43] A. A. diSessa, N. M. Gillespie, and J. B. Esterly, "Coherence versus fragmentation in the development of the concept of force," *Cognitive Science*, vol. 28, no. 6, pp. 843–900, Nov. 1, 2004, ISSN: 0364-0213. DOI: 10.1016/j.cogsci.2004.05.003.

[44] A. A. diSessa, "A history of conceptual change research," in *The Cambridge Handbook of the Learning Sciences*, R. K. Sawyer, Ed., 2nd ed., Cambridge University Press, 2014, pp. 88–108, ISBN: 978-1-139-51952-6. DOI: 10.1017/CBO9781139519526.007.

[45] F. Fayyaz, R. A. Streveler, A. Iqbal, and M. Kamran, "Category Mistakes, Knowledge in Pieces, or Something Else? Problems in Conceptually Learning Signal Analysis," *International Journal of Engineering Education*, vol. 31, pp. 58–71, 1(A) 2015.

[46] S. Brown, D. Montfort, N. Perova-Mello, B. Lutz, A. Berger, and R. Streveler, "Framework theory of conceptual change to interpret undergraduate engineering students' explanations about mechanics of materials concepts: Conceptual change in mechanics of materials," *Journal of Engineering Education*, vol. 107, no. 1, pp. 113–139, Jan. 2018, ISSN: 10694730. DOI: 10.1002/jee.20186.

[47] T. V. Goris and M. J. Dyrenfurth, "How electrical engineering technology students understand concepts of electricity. Comparison of misconceptions of freshmen, sophomores, and seniors," in *2013 ASEE Annual Conference & Exposition*, Jun. 2013, p. 20. [Online]. Available: https://peer.asee.org/19682.

[48] E. Charters, "The Use of Think-aloud Methods in Qualitative Research An Introduction to Think-aloud Methods," *Brock Education Journal*, vol. 12, no. 2, Jul. 1, 2003, ISSN: 2371-7750, 1183-1189. DOI: 10.26522/brocked.v12i2.38.

[49] K. Wage, J. Buck, and M. Hjalmarson, "Analyzing misconceptions using the signals and systems concept inventory and student interviews," in *2006 IEEE 12th Digital Signal Processing Workshop & 4th IEEE Signal Processing Education Workshop*, IEEE, Sep. 2006, pp. 123–128, ISBN: 978-1-4244-0535-0. DOI: 10.1109/DSPWS.2006.265451.

[50] D. Montfort, G. L. Herman, S. Brown, H. M. Matusovich, R. A. Streveler, and O. Adesope, "Patterns of student conceptual understanding across engineering content areas," *International Journal of Engineering Education*, vol. 31, pp. 1587–1604, 6(A) 2015. [Online]. Available: http://publish.illinois.edu/glherman/files/2016/03/2015-IJEE-Conceptual-Understanding-Across-Disciplines.pdf.

[51] F. Fayyaz, "A qualitative study of problematic reasonings of undergraduate electrical engineering students in Continuous Time Signals and Systems courses," Ph.D. dissertation, Purdue University, 2014, 301 pp. [Online]. Available: https://docs.lib.purdue.edu/open_access_dissertations/266/.

[52] A. A. diSessa, "An Interactional Analysis of Clinical Interviewing," *Cognition and Instruction*, vol. 25, no. 4, pp. 523–565, Dec. 2007, ISSN: 0737-0008, 1532-690X. DOI: 10.1080/07370000701632413.

[53] K. A. Ericsson and H. A. Simon, "Preface to the revised edition," in *Protocol Analysis: Verbal Reports as Data*, Rev. ed, MIT Press, 1993, ISBN: 978-0-262-55023-9.

[54] ——, "Chapter 1: Introduction and summary," in *Protocol Analysis: Verbal Reports as Data*, Rev. ed, MIT Press, 1993, ISBN: 978-0-262-55023-9.

[55] T. Boren and J. Ramey, "Thinking aloud: Reconciling theory and practice," *IEEE Transactions on Professional Communication*, vol. 43, no. 3, pp. 261–278, 2000, ISSN: 03611434. DOI: 10.1109/47.867942.

[56] K. E. Wage, J. R. Buck, J. K. Nelson, and M. A. Hjalmarson, "What were they thinking?: Refining conceptual assessments using think-aloud problem solving," *IEEE Signal Processing Magazine*, vol. 38, no. 3, pp. 85–93, May 2021, ISSN: 1053-5888, 1558-0792. DOI: 10.1109/MSP.2021.3060382.

[57] D. Hestenes, M. Wells, and G. Swackhamer, "Force concept inventory," *The Physics Teacher*, vol. 30, no. 3, pp. 141–158, Mar. 1, 1992, ISSN: 0031-921X. DOI: 10.1119/1.2343497.

[58] P. V. Engelhardt and R. J. Beichner, "Students' understanding of direct current resistive electrical circuits," *American Journal of Physics*, vol. 72, no. 1, pp. 98–115, Dec. 12, 2003, ISSN: 0002-9505. DOI: 10.1119/1.1614813.

[59] H. Peşman and A. Eryılmaz, "Development of a Three-Tier Test to Assess Misconceptions About Simple Electric Circuits," *The Journal of Educational Research*, vol. 103, no. 3, pp. 208–222, Feb. 16, 2010, ISSN: 0022-0671. DOI: 10.1080/00220670903383002.

[60] B. Notaros, "Concept inventory assessment instruments for electromagnetics education," in *IEEE Antennas and Propagation Society International Symposium*, vol. 1, Jun. 2002, 684–687 vol.1. DOI: 10.1109/APS.2002.1016436.

[61] M. McColgan, R. Finn, D. Broder, and G. Hassel, “Assessing students’ conceptual knowledge of electricity and magnetism,” *Physical Review Physics Education Research*, vol. 13, Oct. 23, 2017. DOI: 10.1103/PhysRevPhysEducRes.13.020121.

[62] M. Bristow, K. Erkorkmaz, J. P. Huissoon, S. Jeon, W. S. Owen, S. L. Waslander, and G. D. Stubley, “A Control Systems Concept Inventory Test Design and Assessment,” *IEEE Transactions on Education*, vol. 55, no. 2, pp. 203–212, May 2012, ISSN: 1557-9638. DOI: 10.1109/TE.2011.2160946.

[63] A. Traxler, R. Henderson, J. Stewart, G. Stewart, A. Papak, and R. Lindell, “Gender fairness within the Force Concept Inventory,” *Physical Review Physics Education Research*, vol. 14, no. 1, p. 010 103, Jan. 18, 2018, ISSN: 2469-9896. DOI: 10.1103/PhysRevPhysEducRes.14.010103.

[64] J. T. Laverty and M. D. Caballero, “Analysis of the most common concept inventories in physics: What are we assessing?” *Physical Review Physics Education Research*, vol. 14, no. 1, p. 010 123, Apr. 12, 2018, ISSN: 2469-9896. DOI: 10.1103/PhysRevPhysEducRes.14.010123.

[65] P. Steif, “Comparison between performance on a concept inventory and solving of multifaceted problems,” in *33rd Annual Frontiers in Education, 2003. FIE 2003.*, vol. 1, Nov. 2003, T3D–T3D. DOI: 10.1109/FIE.2003.1263339.

[66] A. M. Goncher and W. Boles, “Enhancing the effectiveness of concept inventories using textual analysis: Investigations in an electrical engineering subject,” *European Journal of Engineering Education*, vol. 44, no. 1-2, pp. 222–233, Mar. 4, 2019, ISSN: 0304-3797, 1469-5898. DOI: 10.1080/03043797.2017.1410523.

[67] K. E. Wage, J. R. Buck, M. A. Hjalmarson, and J. K. Nelson, “Signals and systems assessment: Comparison of responses to multiple choice conceptual questions and open-ended final exam problems,” in *2011 Digital Signal Processing and Signal Processing Education Meeting (DSP/SPE)*, IEEE, Jan. 2011, pp. 198–203, ISBN: 978-1-61284-226-4. DOI: 10.1109/DSP-SPE.2011.5739211.

[68] R. Somers, S. Cunningham-Nelson, and W. Boles, “Applying natural language processing to automatically assess student conceptual understanding from textual responses,” *Australasian Journal of Educational Technology*, vol. 37, no. 5, pp. 98–115, Dec. 6, 2021, ISSN: 1449-5554, 1449-3098. DOI: 10.14742/ajet.7121.

[69] R. R. Hake, “Interactive-engagement versus traditional methods: A six-thousand-student survey of mechanics test data for introductory physics courses,” *American Journal of Physics*, vol. 66, no. 1, pp. 64–74, Jan. 1998, ISSN: 0002-9505, 1943-2909. DOI: 10.1119/1.18809.

[70] J. R. Buck, K. E. Wage, and M. A. Hjalmarson, “Item response analysis of the continuous-time signals and systems concept inventory,” in *2009 IEEE 13th Digital Signal Processing Workshop and 5th IEEE Signal Processing Education Workshop*, Jan. 2009, pp. 726–730. DOI: 10.1109/DSP.2009.4786017.

[71] R. Nasr, S. R. Hall, and P. Garik, "Understanding naïve reasonings in signals and systems: A foundation for designing effective instructional material," in *2009 IEEE 13th Digital Signal Processing Workshop and 5th IEEE Signal Processing Education Workshop*, IEEE, Jan. 2009, pp. 720–725. DOI: 10.1109/DSP.2009.4786016.

[72] J. K. Nelson, M. A. Hjalmarson, K. E. Wage, and J. R. Buck, "Students' interpretation of the importance and difficulty of concepts in signals and systems," in *2010 IEEE Frontiers in Education Conference (FIE)*, IEEE, Oct. 2010, T3G-1-T3G–6, ISBN: 978-1-4244-6261-2. DOI: 10.1109/FIE.2010.5673121.

[73] R. Togneri and S. Male, "Signals and Systems: Casting It as an Action-adventure Rather than a Horror Genre," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, May 2019, pp. 7859–7863, ISBN: 978-1-4799-8131-1. DOI: 10.1109/ICASSP.2019.8682794.

[74] J. A. Hurtado, J. Quiroga, and B. Masiero, "Motivating and envolving projects in signal processing class," in *2016 12th International CDIO Conference*, CDIO Conference, Jun. 2016, p. 10. [Online]. Available: http://www.cdio.org/knowledge-library/documents/motivating-and-envolving-projects-signal-processing-class.

[75] R. Nasr, S. Hall, and P. Garik, "Student misconceptions in signals and systems and their origins," in *Proceedings Frontiers in Education 35th Annual Conference*, IEEE, 2005, T4E-26-T4E–31, ISBN: 978-0-7803-9077-5. DOI: 10.1109/FIE.2005.1611980.

[76] S. Pamplona, I. Seoane, and J. Bravo-Agapito, "Assessing conceptual knowledge in three online engineering courses: Theory of computation and compiler construction, operating systems, and signal and systems," in *Proceedings of the First International Conference on Data Science, E-learning and Information Systems*, ACM, Oct. 2018, pp. 1–6, ISBN: 978-1-4503-6536-9. DOI: 10.1145/3279996.3280005.

[77] S. Male and C. Baillie, "Threshold concepts," presented at the CHEER UP - Cambridge Handbook of Engineering Education Research - Updated Perspectives (Online), Jul. 14, 2020.

[78] C. Jia, A. Bennett, D.-H. Nguyen, N. Rebello, and S. Warren, "Teaching-learning interviews to understand and remediate student difficulties with Fourier series concepts," in *2011 ASEE Annual Conference & Exposition Proceedings*, ASEE Conferences, Jun. 2011, pp. 22.1409.1–22.1409.16. DOI: 10.18260/1-2--18564.

[79] J. Buck and K. Wage, "Active and cooperative learning in signal processing courses," *IEEE Signal Processing Magazine*, vol. 22, no. 2, pp. 76–81, Mar. 2005, ISSN: 1053-5888. DOI: 10.1109/MSP.2005.1406489.

[80] J. R. Buck, K. E. Wage, M. A. Hjalmarson, and J. K. Nelson, "Comparing student understanding of signals and systems using a concept inventory, a traditional exam and interviews," in *2007 37th Annual Frontiers in Education Conference - Global Engineering: Knowledge without Borders, Opportunities without Passports*, IEEE, Oct. 2007, S1G-1-S1G–6. DOI: 10.1109/FIE.2007.4418043.

[81] W. T. Padgett, M. A. Yoder, and S. A. Forbes, "Extending the usefulness of the Signals and Systems Concept Inventory (SSCI)," in *2011 Digital Signal Processing and Signal Processing Education Meeting (DSP/SPE)*, IEEE, Jan. 2011, pp. 204–209, ISBN: 978-1-61284-226-4. DOI: 10.1109/DSP-SPE.2011.5739212.

[82] C. Greene, "Studio based instruction in signals and systems," in *2007 ASEE Annual Conference & Exposition Proceedings*, ASEE, 2007. [Online]. Available: https://peer.asee.org/studio-based-instruction-in-signals-and-systems.pdf.

[83] S. J. Pollock, "Longitudinal study of student conceptual understanding in electricity and magnetism," *Physical Review Special Topics - Physics Education Research*, vol. 5, no. 2, p. 020 110, Dec. 15, 2009, ISSN: 1554-9178.
DOI: 10.1103/PhysRevSTPER.5.020110.

[84] G. E. Francis, J. P. Adams, and E. J. Noonan, "Do they stay fixed?" *The Physics Teacher*, vol. 36, no. 8, pp. 488–490, Nov. 1998, ISSN: 0031-921X. DOI: 10.1119/1.879933.

[85] J. Bernhard, "Does active engagement curricula give long-lived conceptual understanding?" In *Physics Teacher Education Beyond 2020*, R. Pinto and S. Surinach, Eds., Elsevier, 2001, pp. 749–752. [Online]. Available: https://www.per-central.org/items/detail.cfm?ID=12221.

[86] A. Pawl, A. Barrantes, D. E. Pritchard, and R. Mitchell, "What do seniors remember from freshman physics?" *Physical Review Special Topics - Physics Education Research*, vol. 8, no. 2, p. 020 118, Dec. 10, 2012, ISSN: 1554-9178.
DOI: 10.1103/PhysRevSTPER.8.020118.

[87] S. A. Brown and M. S. Barner, "Board # 13 : Examining engineering concepts in practice: Is conceptual understanding relevant to practice?," presented at the 2017 ASEE Annual Conference & Exposition, Jun. 24, 2017. [Online]. Available: https://peer.asee.org/board-13-examining-engineering-concepts-in-practice-is-conceptual-understanding-relevant-to-practice.

[88] H. J. Walberg, E. Pascarella, G. D. Haertel, L. K. Junker, and F. D. Boulanger, "Probing a model of educational productivity in high school science with national assessment samples," *Journal of Educational Psychology*, vol. 74, no. 3, pp. 295–307, 1982, ISSN: 1939-2176. DOI: 10.1037/0022-0663.74.3.295.

[89] B. J. Fraser, H. J. Walberg, W. W. Welch, and J. A. Hattie, "Syntheses of educational productivity research," *International Journal of Educational Research*, vol. 11, no. 2, pp. 147–252, Jan. 1987, ISSN: 08830355. DOI: 10.1016/0883-0355(87)90035-8.

[90] X. Ma and J. Wang, "A confirmatory examination of Walberg's Model of Educational Productivity in student career aspiration," *Educational Psychology*, vol. 21, no. 4, pp. 443–453, Dec. 2001, ISSN: 0144-3410, 1469-5820.
DOI: 10.1080/01443410120090821.

[91] A. J. Reynolds and H. J. Walberg, "A structural model of science achievement," *Journal of Educational Psychology*, vol. 83, no. 1, pp. 97–107, Mar. 1991, ISSN: 0022-0663.
DOI: 10.1037/0022-0663.83.1.97.

[92] M. Bruinsma and E. P. W. A. Jansen, "Educational productivity in higher education: An examination of part of the Walberg educational productivity model," *School Effectiveness and School Improvement*, vol. 18, no. 1, pp. 45–65, Mar. 2007, ISSN: 0924-3453, 1744-5124. DOI: 10.1080/09243450600797711.

[93] H. J. Walberg, B. J. Fraser, and W. W. Welch, "A test of a Model of Educational Productivity among senior high school students," *The Journal of Educational Research*, vol. 79, no. 3, pp. 133–139, Jan. 1986, ISSN: 0022-0671, 1940-0675. DOI: 10.1080/00220671.1986.10885664.

[94] H. J. Walberg and T. Weinstein, "The production of achievement and attitude in high school social studies," *The Journal of Educational Research*, vol. 75, no. 5, pp. 285–293, May 1982, ISSN: 0022-0671, 1940-0675. DOI: 10.1080/00220671.1982.10885396.

[95] M. L. Johnson and H. J. Walberg, "Factors Influencing Grade Point Averages at a Community College," *Community College Review*, vol. 16, no. 4, pp. 50–60, Apr. 1989, ISSN: 0091-5521, 1940-2325. DOI: 10.1177/009155218901600407.

[96] A. Wigfield and J. S. Eccles, "Expectancy–value theory of achievement motivation," *Contemporary Educational Psychology*, vol. 25, no. 1, pp. 68–81, Jan. 2000, ISSN: 0361476X. DOI: 10.1006/ceps.1999.1015.

[97] P. R. Pintrich and E. V. D. Groot, "Motivational and self-regulated learning vomponents of vlassroom scademic performance," *Journal of Educational Psychology*, vol. 82, no. 1, pp. 33–40, 1990.

[98] J. Brouwer, E. Jansen, A. Hofman, and A. Flache, "Early tracking or finally leaving? Determinants of early study success in first-year university students," *Research in Post-Compulsory Education*, vol. 21, no. 4, pp. 376–393, Oct. 2016, ISSN: 1359-6748, 1747-5112. DOI: 10.1080/13596748.2016.1226584.

[99] M. Yoder and B. Black, "Teaching DSP first with LabVIEW," in *2006 IEEE 12th Digital Signal Processing Workshop & 4th IEEE Signal Processing Education Workshop*, IEEE, Sep. 2006, pp. 278–280, ISBN: 978-1-4244-0535-0. DOI: 10.1109/DSPWS.2006.265390.

[100] F. Ulaby and A. Yagle, *Signals & Systems: Theory and Applications*. Michigan Publishing, 2018, 666 pp., ISBN: 978-1-60785-487-6. [Online]. Available: http://ss2.eecs.umich.edu/.

[101] S. A. McLeod. (Aug. 3, 2019). "Likert scale," Simply Psychology, [Online]. Available: https://www.simplypsychology.org/likert-scale.html.

[102] T. Fernandez, A. Godwin, J. Doyle, D. Verdin, H. Boone, A. Kirn, L. Benson, and G. Potvin, "More Comprehensive and Inclusive Approaches to Demographic Data Collection," in *2016 ASEE Annual Conference & Exposition Proceedings*, ASEE Conferences, Jun. 2016, p. 25 751. DOI: 10.18260/p.25751.

[103] (). "Likert Scale Definition, Examples and Analysis — Simply Psychology," [Online]. Available: https://www.simplypsychology.org/likert-scale.html.

[104] Xiufeng Liu, *Using and Developing Measurement Instruments in Science Education: A Rasch Modeling Approach*, ser. Science and Engineering Education Sources. Information Age Publishing, 2010, ISBN: 978-1-61735-003-0. [Online]. Available: http://proxy.lib.umich.edu/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=e000xna&AN=521953&site=ehost-live&scope=site.

[105] R. V. Vitali, N. C. Perkins, and C. J. Finelli, "Comparing student performance on low-stakes and high-stakes evaluations of conceptual understanding," in *2018 IEEE Frontiers in Education Conference (FIE)*, Oct. 2018, pp. 1–4.
DOI: 10.1109/FIE.2018.8658449.

[106] StataCorp, "Factor analysis," in *Stata 16 Base Reference Manual*, Stata Press, 2019. [Online]. Available: https://www.stata.com/manuals/mvfactor.pdf#mvfactor.

[107] C. D. Dziuban and E. C. Shirkey, "When is a correlation matrix appropriate for factor analysis? Some decision rules," *Psychological Bulletin*, vol. 81, no. 6, pp. 358–361, Jun. 1974, ISSN: 0033-2909. DOI: 10.1037/h0036316.

[108] W. J. van der Linden, Ed., *Handbook of Item Response Theory. Volume 1: Models*, ser. Chapman & Hall/CRC Statistics in the Social and Behavioral Sciences Series. CRC Press, Taylor & Francis Group, 2016, 595 pp., ISBN: 978-1-4665-1431-7.

[109] A. Weissman, "Optimizing information using the EM algorithm in item response theory," *Annals of Operations Research*, vol. 206, no. 1, pp. 627–646, Jul. 1, 2013, ISSN: 1572-9338. DOI: 10.1007/s10479-012-1204-4.

[110] S. Guo and C. Zheng, "The Bayesian expectation-maximization-maximization for the 3PLM," *Frontiers in Psychology*, vol. 10, 2019, ISSN: 1664-1078. [Online]. Available: https://www.frontiersin.org/article/10.3389/fpsyg.2019.01175.

[111] (). "Carnegie Classifications — Definitions," The Carnegie Classification of Institutions of Higher Education, [Online]. Available: https://carnegieclassifications.iu.edu/definitions.php.

[112] H. C. Powell, R. W. Williams, M. Brandt-Pearce, and R. Weikle, "Restructuring an electrical and computer engineering curriculum: A vertically integrated laboratory/lecture approach," presented at the 2015 ASEE Southeast Section Conference, American Society for Engineering Education, Apr. 2015. [Online]. Available: http://se.asee.org/proceedings/ASEE2015/papers2015/53.pdf.

[113] StataCorp, "IRT (item response theory) — Stata," in *Stata 16 Base Reference Manual*, Stata Press, 2019. [Online]. Available: https://www.stata.com/manuals/irtirt.pdf.

[114] P. Solomon, "The think aloud method: A practical guide to modelling cognitive processes," *Information Processing & Management*, vol. 31, no. 6, pp. 906–907, Nov. 1995, ISSN: 03064573. DOI: 10.1016/0306-4573(95)90031-4.

[115] L. Ríos, B. Pollard, D. R. Dounas-Frazer, and H. J. Lewandowski, "Using think-aloud interviews to characterize model-based reasoning in electronics for a laboratory course assessment," *Physical Review Physics Education Research*, vol. 15, no. 1, p. 010 140, Jun. 12, 2019, ISSN: 2469-9896. DOI: 10.1103/PhysRevPhysEducRes.15.010140.

[116] S. Kvale, "Chatper 5: Conducting an interview," in *Doing Interviews*, Jul. 21, 2020, pp. 51–65. DOI: 10.4135/9781849208963.

[117] L. Cohen, L. Manion, and K. Morrison, *Research Methods in Education*, 7th ed. Routledge, 2011, 758 pp., ISBN: 978-0-415-58335-0.

[118] M. B. Miles, A. M. Huberman, and J. Saldaña, "Chapter 4: Fundamentals of qualitative data analysis," in *Qualitative Data Analysis : A Methods Sourcebook*, SAGE Publications, 2014, 69=104, ISBN: 978-1-4522-5787-7.

[119] M. Prince, "Does Active Learning Work? A Review of the Research," *Journal of Engineering Education*, vol. 93, no. 3, pp. 223–231, Jul. 2004, ISSN: 10694730. DOI: 10.1002/j.2168-9830.2004.tb00809.x.

[120] B. S. Bloom, "The 2 Sigma Problem: The Search for Methods of Group Instruction as Effective as One-to-One Tutoring," *Educational Researcher*, p. 13, 1984.

[121] J. A. Dole and G. M. Sinatra, "Reconceptalizing change in the cognitive construction of knowledge," *Educational Psychologist*, vol. 33, no. 2-3, pp. 109–128, Mar. 1998, ISSN: 0046-1520, 1532-6985. DOI: 10.1080/00461520.1998.9653294.

[122] G. Taasoobshirazi, B. Heddy, M. Bailey, and J. Farley, "A multivariate model of conceptual change," *Instructional Science*, vol. 44, no. 2, pp. 125–145, Apr. 2016, ISSN: 0020-4277, 1573-1952. DOI: 10.1007/s11251-016-9372-2.

[123] S. M. Glynn, G. Taasoobshirazi, and P. Brickman, "Nonscience majors learning science: A theoretical model of motivation," *Journal of Research in Science Teaching*, vol. 44, no. 8, pp. 1088–1107, Oct. 2007, ISSN: 00224308, 10982736. DOI: 10.1002/tea.20181.

[124] B. A. Greene, R. B. Miller, H. Crowson, B. L. Duke, and K. L. Akey, "Predicting high school students' cognitive engagement and achievement: Contributions of classroom perceptions and motivation," *Contemporary Educational Psychology*, vol. 29, no. 4, pp. 462–482, Oct. 2004, ISSN: 0361476X. DOI: 10.1016/j.cedpsych.2004.01.006.

[125] A. J. Elliot and K. Murayama, "On the measurement of achievement goals: Critique, illustration, and application.," *Journal of Educational Psychology*, vol. 100, no. 3, pp. 613–628, Aug. 2008, ISSN: 1939-2176, 0022-0663. DOI: 10.1037/0022-0663.100.3.613.

[126] J. T. Cacioppo, R. E. Petty, and C. Feng Kao, "The efficient assessment of need for cognition," *Journal of Personality Assessment*, vol. 48, no. 3, pp. 306–307, Jun. 1984, ISSN: 0022-3891, 1532-7752. DOI: 10.1207/s15327752jpa4803_13.

[127] R. Pekrun, T. Goetz, A. C. Frenzel, P. Barchfeld, and R. P. Perry, "Measuring emotions in students' learning and performance: The Achievement Emotions Questionnaire (AEQ)," *Contemporary Educational Psychology*, vol. 36, no. 1, pp. 36–48, Jan. 2011, ISSN: 0361476X. DOI: 10.1016/j.cedpsych.2010.10.002.

[128] R. M. Felder and R. Brent, "Understanding student differences," *Journal of Engineering Education*, vol. 94, no. 1, pp. 57–72, Jan. 2005, ISSN: 10694730. DOI: 10.1002/j.2168-9830.2005.tb00829.x.

[129] K. C. McKell and A. Danowitz, "Exploring the effect of standards-based grading on student learning," in *2020 IEEE Frontiers in Education Conference (FIE)*, IEEE, Oct. 21, 2020, pp. 1–7, ISBN: 978-1-72818-961-1. DOI: 10.1109/FIE44824.2020.9273889.

[130] J. P. Spradley, *The Ethnographic Interview*. Holt, Rinehart and Winston, 1979, 247 pp., ISBN: 978-0-03-044496-8.

[131] J. Spradley, "Step seven: Asking structural questions," in *The Ethnographic Interview*, ser. Anthropology / Harcourt College, Holt, Rinehart and Winston, 1979, pp. 120–131, ISBN: 978-0-03-044496-8. [Online]. Available: https://books.google.com/books?id=XP5_AAAAMAAJ.

[132] ——, "Step 4: Asking descriptive questions," in *The Ethnographic Interview*, ser. Anthropology / Harcourt College, Holt, Rinehart and Winston, 1979, pp. 78–91, ISBN: 978-0-03-044496-8. [Online]. Available: https://books.google.com/books?id=XP5_AAAAMAAJ.

[133] D. V. Wakefield, "Math as a Second Language," *The Educational Forum*, vol. 64, no. 3, pp. 272–279, Sep. 30, 2000, ISSN: 0013-1725, 1938-8098. DOI: 10.1080/00131720008984764.

[134] B. A. Oakley, B. Rogowsky, and T. J. Sejnowski, *Uncommon Sense Teaching: Practical Insights in Brain Science to Help Students Learn*. TarcherPerigee, an imprint of Penguin Random House LLC, 2021, 322 pp., ISBN: 978-0-593-32973-3.

[135] H. W. Engl, M. Hanke, and A. Neubauer, *Regularization of inverse problems*. Dordrecht: Kluwer, 1996.

[136] C. H. McCollough, A. C. Bartley, R. E. Carter, *et al.*, "Low-dose CT for the detection and classification of metastatic liver lesions: Results of the 2016 Low Dose CT Grand Challenge," *Med. Phys.*, vol. 44, no. 10, e339–52, Oct. 2017. DOI: 10.1002/mp.12345.

[137] G. Wang, "A perspective on deep imaging," *IEEE Access*, vol. 4, 8914–24, Nov. 2016. DOI: 10.1109/ACCESS.2016.2624938.

[138] K. Hammernik and F. Knoll, "Machine learning for image reconstruction," in *Handbook of Medical Image Computing and Computer Assisted Intervention*, Elsevier, 2020, pp. 25–64, ISBN: 978-0-12-816176-0. DOI: 10.1016/B978-0-12-816176-0.00007-7.

[139] S. Ravishankar, J. C. Ye, and J. A. Fessler, "Image reconstruction: From sparsity to data-adaptive methods and machine learning," *Proc. IEEE*, vol. 108, no. 1, 86–109, Jan. 2020. DOI: 10.1109/JPROC.2019.2936204.

[140] M. T. McCann and M. Unser, "Biomedical image reconstruction: From the foundations to deep neural networks," *Foundation and Trends in Signal Processing*, vol. 13, no. 3, pp. 283–359, 2019. DOI: 10.1561/2000000101.

[141] S. Dempe and A. Zemkoho, Eds., *Bilevel Optimization: Advances and next Challenges*, ser. Springer Optimization and Its Applications. Springer International Publishing, 2020, vol. 161, ISBN: 978-3-030-52118-9. DOI: 10.1007/978-3-030-52119-6.

[142] J. Mairal, F. Bach, and J. Ponce, "Task-driven dictionary learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 4, pp. 791–804, Apr. 2012, ISSN: 1939-3539. DOI: 10.1109/TPAMI.2011.156.

[143] L. Calatroni, C. Chung, J. C. De los Reyes, C.-B. Schönlieb, and T. Valkonen, "Bilevel approaches for learning of variational imaging models," in *Variational Methods in Imaging and Geometric Control*, ser. Radon Series on Computational and Applied Mathematics, vol. 18, De Gruyter, 2017. [Online]. Available: http://arxiv.org/abs/1505.02120.

[144] J. C. De los Reyes, C.-B. Schönlieb, and T. Valkonen, "Bilevel parameter learning for higher-order total variation regularisation models," *Journal of Mathematical Imaging and Vision*, vol. 57, no. 1, pp. 1–25, Jan. 2017, ISSN: 0924-9907, 1573-7683. DOI: 10.1007/s10851-016-0662-8.

[145] P. Knöbelreiter, C. Sormann, A. Shekhovtsov, F. Fraundorfer, and T. Pock, "Belief propagation reloaded: Learning BP-layers for labeling problems," presented at the The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Jun. 2020, pp. 7897–7906. DOI: 10.1109/CVPR42600.2020.00792.

[146] P. Ochs, R. Ranftl, T. Brox, and T. Pock, "Techniques for gradient-based bilevel optimization with non-smooth lower level problems," *Journal of Mathematical Imaging and Vision*, vol. 56, no. 2, pp. 175–194, Oct. 2016, ISSN: 0924-9907, 1573-7683. DOI: 10.1007/s10851-016-0663-7.

[147] J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah, "Julia: A fresh approach to numerical computing," *SIAM Review*, vol. 59, no. 1, 65–98, 2017. DOI: 10.1137/141000671.

[148] M. Stone, "Cross-validation: A review," *Math Oper Stat Ser Stat.*, vol. 9, no. 1, 127–139, 1978. DOI: 10.1080/02331887808801414.

[149] G. H. Golub, M. Heath, and G. Wahba, "Generalized cross-validation as a method for choosing a good ridge parameter," *Technometrics*, vol. 21, no. 2, 215–23, May 1979. [Online]. Available: http://www.jstor.org/stable/1268518.

[150] D. L. Phillips, "A technique for the numerical solution of certain integral equations of the first kind," *J. Assoc. Comput. Mach.*, vol. 9, no. 1, 84–97, Jan. 1962. DOI: 10.1145/321105.321114.

[151] S. S. Saquib, C. A. Bouman, and K. Sauer, "ML parameter estimation for Markov random fields, with applications to Bayesian tomography," *IEEE Trans. Im. Proc.*, vol. 7, no. 7, 1029–44, Jul. 1998. DOI: 10.1109/83.701163.

[152] W. P. Segars, G. Sturgeon, S. Mendonca, J. Grimes, and B. M. W. Tsui, "4D XCAT phantom for multimodality imaging research," *Medical Physics*, vol. 37, no. 9, pp. 4902–15, Aug. 2010. DOI: 10.1118/1.3480985.

[153] C. Poon and G. Peyré, “Smooth Bilevel Programming for Sparse Regularization,” in *35th Conference on Neural Information Processing Systems*, 2021. [Online]. Available: https://proceedings.neurips.cc/paper/2021/hash/0bed45bd5774ffddc95ffe500024f628-Abstract.html.

[154] R. Fletcher and S. Leyffer, “Numerical experience with solving MPECs as NLPs,” Department of Mathematics and Computer Science, University of Dundee, Dundee, 2002. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.57.6674.

[155] B. Colson, P. Marcotte, and G. Savard, “An overview of bilevel optimization,” *Annals of Operations Research*, vol. 153, no. 1, pp. 235–256, Jun. 6, 2007. DOI: 10.1007/s10479-007-0176-2.

[156] P. Jain and P. Kar, “Non-convex optimization for machine learning,” *Found. & Trends in Machine Learning*, vol. 10, no. 3-4, 142–336, 2017. DOI: 10.1561/2200000058.

[157] A. Effland, E. Kobler, K. Kunisch, and T. Pock, “Variational networks: An optimal control approach to early stopping variational methods for image restoration,” *Journal of Mathematical Imaging and Vision*, vol. 62, no. 3, pp. 396–416, Apr. 2020, ISSN: 0924-9907, 1573-7683. DOI: 10.1007/s10851-019-00926-8.

[158] E. Haber and L. Tenorio, “Learning regularization functionals a supervised training approach,” *Inverse Problems*, vol. 19, no. 3, pp. 611–626, Jun. 1, 2003, ISSN: 0266-5611, 1361-6420. DOI: 10.1088/0266-5611/19/3/309.

[159] M. J. Ehrhardt and L. Roberts, “Inexact derivative-free optimization for bilevel learning,” *Journal of Mathematical Imaging and Vision*, vol. 63, pp. 580–600, Feb. 6, 2021. DOI: 10.1007/s10851-021-01020-8.

[160] F. Sherry, M. Benning, J. C. De los Reyes, M. J. Graves, G. Maierhofer, G. Williams, C.-B. Schonlieb, and M. J. Ehrhardt, “Learning the sampling pattern for MRI,” *IEEE Transactions on Medical Imaging*, vol. 39, no. 12, pp. 4310–4321, Dec. 2020, ISSN: 0278-0062, 1558-254X. DOI: 10.1109/TMI.2020.3017353.

[161] E. Kobler, A. Effland, K. Kunisch, and T. Pock, “Total deep variation: A stable regularization method for inverse problems,” *IEEE transactions on pattern analysis and machine intelligence*, vol. PP, Nov. 2, 2021, ISSN: 1939-3539. DOI: 10.1109/TPAMI.2021.3124086. PMID: 34727026.

[162] Y. Chen, R. Ranftl, and T. Pock, “Insights into analysis operator learning: From patch-based sparse models to higher order MRFs,” *IEEE Transactions on Image Processing*, vol. 23, no. 3, pp. 1060–1072, Mar. 2014, ISSN: 1057-7149, 1941-0042. DOI: 10.1109/TIP.2014.2299065.

[163] K. G. G. Samuel and M. F. Tappen, “Learning optimized MAP estimates in continuously-valued MRF models,” in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2009, pp. 477–484, ISBN: 978-1-4244-3992-8. DOI: 10.1109/CVPR.2009.5206774.

[164] G. Holler, K. Kunisch, and R. C. Barnard, "A bilevel approach for parameter learning in inverse problems," *Inverse Problems*, vol. 34, no. 11, p. 115 012, Nov. 1, 2018, ISSN: 0266-5611, 1361-6420. DOI: 10.1088/1361-6420/aade77.

[165] G. Peyré and J. M. Fadili, "Learning analysis sparsity priors," in *IEEE Intl. Conf. on Sampling Theory and Appl. (SampTA)*, 2011. [Online]. Available: https://hal.archives-ouvertes.fr/hal-00542016.

[166] Y. Chen, H. Liu, X. Ye, and Q. Zhang, "Learnable descent algorithm for nonsmooth nonconvex image reconstruction," *SIAM Journal on Imaging Sciences*, vol. 14, no. 4, pp. 1532–1564, 2021. DOI: 10.1137/20M1353368.

[167] R. M. Lewitt and S. Matej, "Overview of methods for image reconstruction from projections in emission computed tomography," *Proc. IEEE*, vol. 91, no. 10, 1588–611, Oct. 2003. DOI: 10.1109/JPROC.2003.817882.

[168] M. Elad, P. Milanfar, and R. Rubinstein, "Analysis versus synthesis in signal priors," *Inverse Problems*, vol. 23, no. 3, pp. 947–68, Jun. 2007.
DOI: 10.1088/0266-5611/23/3/007.

[169] C. Guillemot and O. Le Meur, "Image inpainting: Overview and recent advances," *IEEE Sig. Proc. Mag.*, vol. 31, no. 1, 127–44, Jan. 2014.
DOI: 10.1109/MSP.2013.2273004.

[170] J. A. Fessler, "Model-based image reconstruction for MRI," *IEEE Sig. Proc. Mag.*, vol. 27, no. 4, 81–9, Jul. 2010. DOI: 10.1109/MSP.2010.936726.

[171] G. H. Golub and C. F. Van Loan, "An analysis of the total least squares problem," *SIAM J. Numer. Anal.*, vol. 17, no. 6, 883–93, Dec. 1980. DOI: 10.1137/0717073.

[172] L. Ying and J. Sheng, "Joint image reconstruction and sensitivity estimation in SENSE (JSENSE)," *Mag. Res. Med.*, vol. 57, no. 6, 1196–1202, Jun. 2007.
DOI: 10.1002/mrm.21245.

[173] A. Chambolle and T. Pock, "An introduction to continuous optimization for imaging," *Acta Numerica*, vol. 25, pp. 161–319, May 1, 2016, ISSN: 0962-4929, 1474-0508.
DOI: 10.1017/S096249291600009X.

[174] S. Nam, M. Davies, M. Elad, and R. Gribonval, "The cosparse analysis model and algorithms," *Applied and Computational Harmonic Analysis*, vol. 34, no. 1, pp. 30–56, Jan. 2013, ISSN: 10635203. DOI: 10.1016/j.acha.2012.03.006.

[175] M. Elad, *Sparse and redundant representations: from theory to applications in signal and image processing*. Berlin: Springer, 2010. DOI: 10.1007/978-1-4419-7011-4.

[176] G. Peyre, "A review of adaptive image representations," *IEEE J. Sel. Top. Sig. Proc.*, vol. 5, no. 5, 896–911, Sep. 2011. DOI: 10.1109/JSTSP.2011.2120592.

[177] E. Candes, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Transactions on Information Theory*, vol. 52, no. 2, pp. 489–509, Feb. 2006, ISSN: 0018-9448.
DOI: 10.1109/TIT.2005.862083.

[178] R. Tibshirani, "Regression shrinkage and selection via the Lasso," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, Jan. 1996, ISSN: 00359246. DOI: 10.1111/j.2517-6161.1996.tb02080.x.

[179] P. Zhou, C. Zhang, and Z. Lin, "Bilevel model-based discriminative dictionary learning for recognition," *IEEE Trans. Im. Proc.*, vol. 26, no. 3, 1173–87, Mar. 2017. DOI: 10.1109/tip.2016.2623487.

[180] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithm," *Physica D*, vol. 60, no. 1-4, 259–68, Nov. 1992. DOI: 10.1016/0167-2789(92)90242-F.

[181] E. J. Candès, Y. C. Eldar, D. Needell, and P. Randall, "Compressed sensing with coherent and redundant dictionaries," *Applied and Computational Harmonic Analysis*, vol. 31, no. 1, pp. 59–73, Jul. 2011, ISSN: 10635203. DOI: 10.1016/j.acha.2010.10.002.

[182] S. Hawe, M. Kleinsteuber, and K. Diepold, "Analysis operator learning and its application to image reconstruction," *IEEE Transactions on Image Processing*, vol. 22, no. 6, pp. 2138–2150, Jun. 2013, ISSN: 1057-7149. DOI: 10.1109/TIP.2013.2246175.

[183] S. Ravishankar and Y. Bresler, "Learning sparsifying transforms," *IEEE Transactions on Signal Processing*, vol. 61, no. 5, pp. 1072–1086, Mar. 2013, ISSN: 1941-0476. DOI: 10.1109/TSP.2012.2226449.

[184] J. A. Fessler, "Optimization methods for MR image reconstruction," *IEEE Sig. Proc. Mag.*, vol. 37, no. 1, 33–40, Jan. 2020. DOI: 10.1109/MSP.2019.2943645.

[185] L. Pfister and Y. Bresler, "Learning filter bank sparsifying transforms," *IEEE Transactions on Signal Processing*, vol. 67, no. 2, pp. 504–519, Jan. 2019, ISSN: 1941-0476. DOI: 10.1109/TSP.2018.2883021.

[186] B. M. Afkham, J. Chung, and M. Chung, "Learning regularization parameters of inverse problems via deep neural networks," *Inverse Problems*, vol. 37, no. 10, p. 105 017, Sep. 2021, ISSN: 0266-5611. DOI: 10.1088/1361-6420/ac245d.

[187] S. Roth and M. Black, "Fields of experts: A framework for learning image priors," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 2, 2005, pp. 860–867. DOI: 10.1109/CVPR.2005.160.

[188] M. F. Tappen, C. Liu, E. H. Adelson, and W. T. Freeman, "Learning gaussian conditional random fields for low-level vision," in *2007 IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2007, pp. 1–8. DOI: 10.1109/CVPR.2007.382979.

[189] B. Ophir, M. Elad, N. Bertin, and M. D. Plumbley, "Sequential minimal eigenvalues - an approach to analysis dictionary learning," *19th European Signal Processing Conference*, pp. 1465–1469, 2011. [Online]. Available: https://ieeexplore.ieee.org/document/7074010.

[190] M. Yaghoobi, S. Nam, R. Gribonval, and M. E. Davies, "Analysis operator learning for overcomplete cosparse representations," presented at the 2011 19th European Signal Processing Conference, IEEE, 2011, pp. 1470–1474. [Online]. Available: https://ieeexplore.ieee.org/document/7074220.

[191] M. Yaghoobi, S. Nam, R. Gribonval, and M. E. Davies, "Constrained overcomplete analysis operator learning for cosparse signal modelling," *IEEE Transactions on Signal Processing*, vol. 61, no. 9, pp. 2341–2355, May 2013. DOI: 10.1109/TSP.2013.2250968.

[192] K. Kunisch and T. Pock, "A bilevel optimization approach for parameter learning in variational models," *SIAM Journal on Imaging Sciences*, vol. 6, no. 2, pp. 938–983, Jan. 2013, ISSN: 1936-4954. DOI: 10.1137/120882706.

[193] I. Y. Chun and J. A. Fessler, "Convolutional analysis operator learning: Acceleration and convergence," *IEEE Transactions on Image Processing*, vol. 29, pp. 2108–2122, 2020, ISSN: 1941-0042. DOI: 10.1109/TIP.2019.2937734.

[194] S. Haykin, "Neural networks expand SP's horizons," *IEEE Sig. Proc. Mag.*, vol. 13, no. 2, 24–49, Mar. 1996. DOI: 10.1109/79.487040.

[195] J.-N. Hwang, S.-Y. Kung, M. Niranjan, and J. C. Principe, "The past, present, and future of neural networks for signal processing," *IEEE Sig. Proc. Mag.*, vol. 14, no. 6, 28–48, Nov. 1997. DOI: 10.1109/79.637299.

[196] A. Lucas, M. Iliadis, R. Molina, and A. K. Katsaggelos, "Using deep neural networks for inverse problems in imaging: Beyond analytical methods," *IEEE Sig. Proc. Mag.*, vol. 35, no. 1, 20–36, Jan. 2018. DOI: 10.1109/msp.2017.2760358.

[197] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention*, 2015, 234–41. DOI: 10.1007/978-3-319-24574-4_28.

[198] J. C. Ye, Y. Han, and E. Cha, "Deep convolutional framelets: A general deep learning framework for inverse problems," *SIAM J. Imaging Sci.*, vol. 11, no. 2, 991–1048, Jan. 2018. DOI: 10.1137/17m1141771.

[199] B. Wen, S. Ravishankar, L. Pfister, and Y. Bresler, "Transform learning for magnetic resonance image reconstruction: From model-based learning to building neural networks," *IEEE Sig. Proc. Mag.*, vol. 37, no. 1, 41–53, Jan. 2020. DOI: 10.1109/MSP.2019.2951469.

[200] M. Feurer and F. Hutter, "Chapter 1: Hyperparameter optimization," in *Automated Machine Learning: Methods, Systems, Challenges*, ser. The Springer Series on Challenges in Machine Learning, F. Hutter, L. Kotthoff, and J. Vanschoren, Eds., Springer International Publishing, 2019, pp. 3–33, ISBN: 978-3-030-05317-8. DOI: 10.1007/978-3-030-05318-5.

[201] L. A. Shepp and B. F. Logan, "The Fourier reconstruction of a head section," *IEEE Trans. Nuc. Sci.*, vol. 21, no. 3, 21–43, Jun. 1974. DOI: 10.1109/TNS.1974.6499235.

[202] J. A. Fessler, *MIRT-demo: 01-recon*, Jul. 25, 2020. [Online]. Available: https://github.com/JeffFessler/mirt-demo/blob/master/isbi-19/01-recon.jl.

[203] C. You, Q. Yang, H. Shan, *et al.*, "Structure-sensitive multi-scale deep neural network for low-dose CT denoising," *IEEE Access*, vol. 6, pp. 41 839–41 855, 2018, ISSN: 2169-3536. DOI: 10.1109/ACCESS.2018.2858196.

[204] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, Apr. 2004, ISSN: 1057-7149. DOI: 10.1109/TIP.2003.819861.

[205] Z. Wang and A. Bovik, "Reduced- and no-reference image quality assessment," *IEEE Signal Processing Magazine*, vol. 28, no. 6, pp. 29–40, Nov. 2011, ISSN: 1053-5888. DOI: 10.1109/MSP.2011.942471.

[206] L. Zhang, L. Zhang, X. Mou, and D. Zhang, "A comprehensive evaluation of full reference image quality assessment algorithms," in *2012 19th IEEE International Conference on Image Processing*, Sep. 2012, pp. 1477–1480, ISBN: 978-1-4673-2533-2. DOI: 10.1109/ICIP.2012.6467150.

[207] W. Zhang, K. Ma, J. Yan, D. Deng, and Z. Wang, "Blind image quality assessment using a deep bilinear convolutional neural network," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 1, pp. 36–47, Jan. 2020, ISSN: 1051-8215, 1558-2205. DOI: 10.1109/TCSVT.2018.2886771.

[208] A. Mason, J. Rioux, S. E. Clarke, A. Costa, M. Schmidt, V. Keough, T. Huynh, and S. Beyea, "Comparison of objective image quality metrics to expert radiologists' scoring of diagnostic quality of MR images," *IEEE Transactions on Medical Imaging*, vol. 39, no. 4, pp. 1064–1072, Apr. 2020, ISSN: 1558-254X. DOI: 10.1109/TMI.2019.2930338.

[209] M. Gholizadeh-Ansari, J. Alirezaie, and P. Babyn, "Deep learning for low-dose CT denoising using perceptual loss and edge detection layer," *J. Digital Im.*, vol. 33, no. 2, 504–15, 2020. DOI: 10.1007/s10278-019-00274-4.

[210] G. Seif and D. A., "Edge-based loss function for single image super-resolution," in *Proc. IEEE Conf. Acoust. Speech Sig. Proc.*, 2018, 1468–72. DOI: 10.1109/ICASSP.2018.8461664.

[211] S. Ravishankar and Y. Bresler, "MR image reconstruction from highly undersampled k-space data by dictionary learning," *IEEE Transactions on Medical Imaging*, vol. 30, no. 5, pp. 1028–1041, May 2011, ISSN: 1558-254X. DOI: 10.1109/TMI.2010.2090538.

[212] Z. Wang, E. Simoncelli, and A. Bovik, "Multiscale structural similarity for image quality assessment," in *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, IEEE, 2003, pp. 1398–1402, ISBN: 978-0-7803-8104-9. DOI: 10.1109/ACSSC.2003.1292216.

[213] G. P. Renieblas, A. T. Nogués, A. M. González, N. G. León, and E. G. . Castillo, "Structural similarity index family for image quality assessment in radiological images," *J. Med. Im.*, vol. 4, no. 3, p. 035 501, Jul. 2017. DOI: 10.1117/1.JMI.4.3.035501.

[214] S. Bosse, D. Maniry, K.-R. Muller, T. Wiegand, and W. Samek, "Deep neural networks for no-reference and full-reference image quality assessment," *IEEE Transactions on Image Processing*, vol. 27, no. 1, pp. 206–219, Jan. 2018, ISSN: 1057-7149, 1941-0042. DOI: 10.1109/TIP.2017.2760518.

[215] G. W. Lindsay, "Convolutional neural networks as a model of the visual system: Past, present, and future," *Journal of Cognitive Neuroscience*, pp. 1–15, Feb. 6, 2020, ISSN: 0898-929X, 1530-8898. DOI: 10.1162/jocn_a_01544.

[216] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations*, May 2015. [Online]. Available: http://arxiv.org/abs/1409.1556.

[217] T. J. Hebert and R. Leahy, "Statistic-based MAP image reconstruction from Poisson data using Gibbs priors," *IEEE Trans. Sig. Proc.*, vol. 40, no. 9, 2290–303, Sep. 1992. DOI: 10.1109/78.157228.

[218] C. M. Stein, "Estimation of the mean of a multivariate normal distribution," *The Annals of Statistics*, vol. 9, no. 6, Nov. 1, 1981, ISSN: 0090-5364. DOI: 10.1214/aos/1176345632.

[219] S. Ramani, T. Blu, and M. Unser, "Monte-carlo sure: A black-box optimization of regularization parameters for general denoising algorithms," *IEEE Transactions on Image Processing*, vol. 17, no. 9, pp. 1540–1554, Sep. 2008, ISSN: 1057-7149. DOI: 10.1109/TIP.2008.2001404.

[220] S. Soltanayev and S. Y. Chun, "Training deep learning based denoisers without ground truth data," in *Neural Information Processing Systems*, vol. 31, 2018. [Online]. Available: https://papers.nips.cc/paper/7587-training-deep-learning-based-denoisers-without-ground-truth-data.

[221] K. Kim, S. Soltanayev, and S. Y. Chun, "Unsupervised training of denoisers for low-dose CT reconstruction without full-dose ground truth," *IEEE J. Sel. Top. Sig. Proc.*, vol. 14, no. 6, 1112–25, Oct. 2020. DOI: 10.1109/JSTSP.2020.3007326.

[222] M. Zhussip, S. Soltanayev, and S. Y. Chun, "Training deep learning based image denoisers from undersampled measurements without ground truth and without image prior," in *Proc. IEEE Conf. on Comp. Vision and Pattern Recognition*, 2019, 10247–56. DOI: 10.1109/CVPR.2019.01050.

[223] H. Zhang, X. Chen, X. Zhang, and X. Zhang, "A bi-level nested sparse optimization for adaptive mechanical fault feature detection," *IEEE Access*, vol. 8, pp. 19 767–19 782, 2020, ISSN: 2169-3536. DOI: 10.1109/ACCESS.2020.2968726.

[224] C.-A. Deledalle, S. Vaiter, J. Fadili, and G. Peyré, "Stein Unbiased GrAdient estimator of the Risk (SUGAR) for multiple parameter selection," *SIAM Journal on Imaging Sciences*, vol. 7, no. 4, pp. 2448–2487, Jan. 2014, ISSN: 1936-4954. DOI: 10.1137/140968045.

[225] Y. C. Eldar, "Rethinking biased estimation: Improving maximum likelihood and the Cramer-Rao bound," *Found. & Trends in Sig. Pro.*, vol. 1, no. 4, 305–449, 2008. DOI: 10.1561/2000000008.

[226] ——, "Generalized SURE for exponential families: Applications to regularization," *IEEE Transactions on Signal Processing*, vol. 57, no. 2, pp. 471–481, Feb. 2009, ISSN: 1053-587X, 1941-0476. DOI: 10.1109/TSP.2008.2008212.

[227] R. Giryes, M. Elad, and Y. C. Eldar, "The projected GSURE for automatic parameter tuning in iterative shrinkage methods," *Applied and Computational Harmonic Analysis*, vol. 30, no. 3, 407–22, May 2011. DOI: 10.1016/j.acha.2010.11.005.

[228] A. Mittal, R. Soundararajan, and A. C. Bovik, “Making a “completely blind” image quality analyzer,” *IEEE Signal Processing Letters*, vol. 20, no. 3, pp. 209–212, Mar. 2013, ISSN: 1070-9908, 1558-2361. DOI: 10.1109/LSP.2012.2227726.

[229] L. Kang, P. Ye, Y. Li, and D. Doermann, “Convolutional neural networks for no-reference image quality assessment,” in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2014, pp. 1733–1740, ISBN: 978-1-4799-5118-5. DOI: 10.1109/CVPR.2014.224.

[230] T. U. of Texas at Austin: Laboratory for Image and Video Engineering. (). “Image & video quality assessment at LIVE,” [Online]. Available: http://live.ece.utexas.edu/research/quality/.

[231] J. Larson, M. Menickelly, and S. M. Wild, “Derivative-free optimization methods,” *Acta Numerica*, vol. 28, pp. 287–404, May 1, 2019, ISSN: 0962-4929, 1474-0508. DOI: 10.1017/S0962492919000060.

[232] O. Gencoglu, M. van Gils, E. Guldogan, C. Morikawa, M. Süzen, M. Gruber, J. Leinonen, and H. Huttunen, *HARK side of deep learning – From grad student descent to automated machine learning*, Apr. 16, 2019. arXiv: 1904.07633.

[233] J. Bergstra and Y. Bengio, “Random search for hyper-parameter optimization,” *Journal of Machine Learning Research*, vol. 13, pp. 281–305, Feb. 2012, ISSN: 1532-4435. DOI: 10.5555/2188385.2188395.

[234] G. Muniraju, B. Kailkhura, J. J. Thiagarajan, and T. Bremer, “Controlled random search improves sample mining and hyper-parameter optimization,” in *Thirty-Third AAAI Conference on Artificial Intelligence*, 2019. [Online]. Available: https://www.osti.gov/servlets/purl/1497973.

[235] H.-G. Beyer, *The Theory of Evolution Strategies*, ser. Natural Computing Series. Springer, 2001, ISBN: 978-3-540-67297-5.

[236] A. Klein, S. Falkner, S. Bartels, P. Hennig, and F. Hutter, “Fast Bayesian hyperparameter optimization on large datasets,” *Electron. J. Statist.*, vol. 11, no. 2, pp. 4945–68, 2017. DOI: 10.1214/17-EJS1335SI.

[237] P. I. Frazier, *A tutorial on bayesian optimization*, Jul. 8, 2018. arXiv: 1807.02811.

[238] A. R. Conn, N. I. M. Gould, and P. L. Toint, *Trust Region Methods*, ser. MOS-SIAM Series on Optimization. Society for Industrial and Applied Mathematics, Jan. 1, 2000, 960 pp., ISBN: 978-0-89871-460-9. DOI: 10.1137/1.9780898719857.

[239] L. Roberts, “Inexact DFO for Bilevel Learning: Dimension Question,” E-mail, Jul. 11, 2021.

[240] C. Cartis and L. Roberts, *Scalable subspace methods for derivative-free nonlinear least-squares optimization*, Feb. 23, 2021. arXiv: 2102.12016.

[241] A. Chambolle and T. Pock, “Learning consistent discretizations of the total variation,” vol. 14, no. 2, pp. 778–813, 2021. DOI: 10.1137/20M1377199.

[242] S. Ravishankar and Y. Bresler, “Efficient blind compressed sensing using sparsifying transforms with convergence guarantees and application to MRI,” *SIAM J. Imaging Sci.*, vol. 8, no. 4, 2519–57, 2015. DOI: 10.1137/141002293.

[243] P. H. Schönemann, "A generalized solution of the orthogonal procrustes problem," *Psychometrika*, vol. 31, no. 1, pp. 1–10, Mar. 1966, ISSN: 0033-3123, 1860-0980. DOI: 10.1007/BF02289451.

[244] R. Rubinstein, T. Peleg, and M. Elad, "Analysis K-SVD: A Dictionary-Learning Algorithm for the Analysis Sparse Model," *IEEE Transactions on Signal Processing*, vol. 61, no. 3, pp. 661–677, Feb. 2013, ISSN: 1941-0476. DOI: 10.1109/TSP.2012.2226445.

[245] I. Y. Chun and J. A. Fessler, "Convolutional dictionary learning: Acceleration and convergence," *IEEE Trans. Im. Proc.*, vol. 27, no. 4, pp. 1697–712, Apr. 2018. DOI: 10.1109/TIP.2017.2761545.

[246] I. Y. Chun and J. A. Fessler, "Convergent convolutional dictionary learning using adaptive contrast enhancement (CDL-ACE): Application of CDL to image denoising," in *Proc. Sampling Theory and Appl.*, Jul. 2017, pp. 460–464. DOI: 10.1109/SAMPTA.2017.8024378.

[247] P. Bao, W. Xia, K. Yang, J. Zhou, and Y. Zhang, "Sparse-view CT reconstruction via convolutional sparse coding," in *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*, Apr. 2019, pp. 1446–1449. DOI: 10.1109/ISBI.2019.8759260.

[248] I. Y. Chun, D. Hong, B. Adcock, and J. A. Fessler, "Convolutional analysis operator learning: Dependence on training data," *IEEE Signal Processing Letters*, vol. 26, no. 8, pp. 1137–1141, Aug. 2019, ISSN: 1070-9908. DOI: 10.1109/LSP.2019.2921446.

[249] I. Y. Chun and J. A. Fessler, "Convolutional analysis operator learning: Application to sparse-view CT," in *Proc., IEEE Asilomar Conf. on Signals, Systems, and Comp.*, Oct. 2018, pp. 1631–5. DOI: 10.1109/ACSSC.2018.8645500.

[250] X. Zheng, I. Y. Chun, Z. Li, Y. Long, and J. A. Fessler, "Sparse-view X-Ray CT reconstruction using L1 prior with learned transform," submitted, Feb. 2019. [Online]. Available: http://arxiv.org/abs/1711.00905.

[251] P. H. Schonemann, "A generalized solution of the orthogonal Procrustes problem," *Psychometrika*, vol. 31, no. 1, pp. 1–10, Mar. 1966.

[252] D. Hong, C. Crockett, and I. Y. Chun, *Convolutional operator learning (for Julia)*, GitHub repository, 2019. [Online]. Available: https://github.com/dahong67/ConvolutionalOperatorLearning.jl.

[253] I. Y. Chun and T. M. Talavage, "Efficient compressed sensing statistical X-ray/CT reconstruction from fewer measurements," in *Proc. Intl. Mtg. on Fully 3D Image Recon. in Rad. and Nuc. Med*, Jun. 2013, pp. 30–3.

[254] I. Y. Chun, X. Zheng, Y. Long, and J. A. Fessler, "Sparse-view X-ray CT reconstruction using L1 regularization with learned sparsifying transform," in *Proc. Intl. Mtg. on Fully 3D Image Recon. in Rad. and Nuc. Med*, Jun. 2017, pp. 115–9. DOI: 10.12059/Fully3D.2017-11-3109002.

[255] Y. Long, J. A. Fessler, and J. M. Balter, "3D forward and back-projection for X-ray CT using separable footprints," *IEEE Transactions on Medical Imaging*, vol. 29, no. 11, pp. 1839–50, Nov. 2010. DOI: 10.1109/TMI.2010.2050898.

[256] J. A. Fessler and W. L. Rogers, “Spatial resolution properties of penalized-likelihood image reconstruction methods: Space-invariant tomographs,” *IEEE Trans. Im. Proc.*, vol. 5, no. 9, pp. 1346–58, Sep. 1996. DOI: 10.1109/83.535846.

[257] B. Sahiner, A. Pezeshk, L. M. Hadjiiski, X. Wang, K. Drukker, K. Cha, R. Summers, and M. L. Giger, “Deep learning in medical imaging and radiation therapy,” *Medical Physics*, Nov. 2018. DOI: 10.1002/mp.13264.

[258] S. Gould, B. Fernando, A. Cherian, P. Anderson, R. S. Cruz, and E. Guo, *On differentiating parameterized argmin and argmax problems with application to bi-level optimization*, Jul. 20, 2016. arXiv: 1607.05447.

[259] J. A. Fessler, “Mean and variance of implicitly defined biased estimators (such as penalized maximum likelihood): Applications to tomography,” *IEEE Trans. Im. Proc.*, vol. 5, no. 3, pp. 493–506, Mar. 1996. DOI: 10.1109/83.491322.

[260] M. Hintermüller and T. Wu, “Bilevel optimization for calibrating point spread functions in blind deconvolution,” *Inverse Problems & Imaging*, vol. 9, no. 4, pp. 1139–1169, 2015, ISSN: 1930-8345. DOI: 10.3934/ipi.2015.9.1139.

[261] S. Scholtes and M. Stöhr, “How stringent is the linear independence assumption for mathematical programs with complementarity constraints?” *Mathematics of Operations Research*, vol. 26, no. 4, pp. 851–863, Nov. 2001, ISSN: 0364-765X, 1526-5471. DOI: 10.1287/moor.26.4.851.10007.

[262] S. Dempe and J. Dutta, “Is bilevel programming a special case of a mathematical program with complementarity constraints?” *Mathematical Programming*, vol. 131, no. 1-2, pp. 37–48, Feb. 2012, ISSN: 0025-5610, 1436-4646. DOI: 10.1007/s10107-010-0342-1.

[263] K. B. Petersen and M. S. Pedersen, *The Matrix Cookbook*. Technical University of Denmark, Nov. 2012. [Online]. Available: http://www2.imm.dtu.dk/pubdb/views/publication_details.php?id=3274.

[264] K. Ji, J. Yang, and Y. Liang, “Bilevel optimization: Convergence analysis and enhanced design,” in *Proceedings of the 38th International Conference on Machine Learning*, Jul. 2021, pp. 4882–4892. [Online]. Available: http://proceedings.mlr.press/v139/ji21c.html.

[265] R. Grazzi, L. Franceschi, M. Pontil, and S. Salzo, “On the iteration complexity of hypergradient computation,” in *Proceedings of the 37th International Conference on Machine Learning*, 2020, p. 11. [Online]. Available: http://proceedings.mlr.press/v119/grazzi20a.html.

[266] P. W. Holland and R. E. Welsch, “Robust regression using iteratively reweighted least-squares,” *Comm. in Statistics—Theory and Methods*, vol. 6, no. 9, 813–27, 1977. DOI: 10.1080/03610927708827533.

[267] H. Zou and T. Hastie, “Regularization and variable selection via the elastic net,” *J. Royal Stat. Soc. Ser. B*, vol. 67, no. 2, 301–20, 2005. DOI: 10.1111/j.1467-9868.2005.00503.x.

[268] C.-s. Foo, C. B., and A. Ng, “Efficient multiple hyperparameter learning for log-linear models,” in *Advances in Neural Information Processing Systems*, vol. 20, Curran Associates, Inc., 2007. [Online]. Available: https://proceedings.neurips.cc/paper/2007/hash/851ddf5058cf22df63d3344ad89919cf-Abstract.html.

[269] Z. Ramzi, F. Mannel, S. Bai, J.-L. Starck, P. Ciuciu, and T. Moreau, “SHINE: SHaring the INverse Estimate from the forward pass for bi-level optimization and implicit models,” in *International Conference on Learning Representations*, 2022. [Online]. Available: https://openreview.net/forum?id=-ApAkox5mp.

[270] P. Sprechmann, R. Litman, T. B. Yakar, A. M. Bronstein, and G. Sapiro, “Supervised sparse analysis and synthesis operators,” in *Neural Information Processing Systems*, 2013, pp. 908–916. [Online]. Available: https://papers.nips.cc/paper/2013/hash/7380ad8a673226ae47fce7bff88e9c33-Abstract.html.

[271] M. T. McCann and S. Ravishankar, “Supervised learning of sparsity-promoting regularizers for denoising,” *arXiv Computing Research Repository*, Jun. 9, 2020. arXiv: 2006.05521.

[272] A. Ghosh, M. T. Mccann, and S. Ravishankar, *Bilevel learning of l1-regularizers with closed-form gradients (BLORC)*, Nov. 21, 2021. arXiv: 2111.10858.

[273] R. J. Tibshirani and J. Taylor, “The solution path of the generalized lasso,” *The Annals of Statistics*, vol. 39, no. 3, Jun. 1, 2011, ISSN: 0090-5364. DOI: 10.1214/11-AOS878.

[274] A. Ghosh, “Questions about BLORC,” E-mail, Feb. 21, 2022.

[275] L. Franceschi, M. Donini, P. Frasconi, and M. Pontil, “Forward and reverse gradient-based hyperparameter optimization,” in *Proceedings of the International Conference on Machine Learning*, PMLR, Dec. 12, 2017, pp. 1165–1173. [Online]. Available: http://proceedings.mlr.press/v70/franceschi17a.html.

[276] B. Dauvergne and L. Hascoet, “The data-flow equations of checkpointing in reverse automatic differentiation,” in *International Conference on Computational Science*, 2006, pp. 566–573. DOI: 10.1007/11758549_78.

[277] M. Kellman, K. Zhang, E. Markley, J. Tamir, E. Bostan, M. Lustig, and L. Waller, “Memory-efficient learning for large-scale computational imaging,” *IEEE Transactions on Computational Imaging*, vol. 6, pp. 1403–1414, 2020, ISSN: 2333-9403, 2334-0118, 2573-0436. DOI: 10.1109/TCI.2020.3025735.

[278] D. Gilton, G. Ongie, and R. Willett, *Deep equilibrium architectures for inverse problems in imaging*, Jun. 2, 2021. arXiv: 2102.07944 [cs, eess]. [Online]. Available: http://arxiv.org/abs/2102.07944.

[279] H. Antil, Z. Di, and R. Khatri, “Bilevel optimization, deep learning and fractional laplacian regularization with applications in tomography,” *Inverse Problems*, Mar. 18, 2020, ISSN: 0266-5611, 1361-6420. DOI: 10.1088/1361-6420/ab80d7.

[280] T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud, “Neural ordinary differential equations,” in *Advances in Neural Information Processing Systems*, vol. 31, Curran Associates, Inc., 2018. [Online]. Available: https://papers.nips.cc/paper/2018/hash/69386f6bb1dfed68692a24c8686939b9-Abstract.html.

[281] M. Thies, F. Wagner, M. Gu, L. Folle, L. Felsner, and A. Maier, *Learned cone-beam CT reconstruction using neural ordinary differential equations*, Jan. 19, 2022. arXiv: 2201.07562.

[282] A. Chambolle and T. Pock, “On the ergodic convergence rates of a first-order primal—dual algorithm,” *Mathematical Programming: Series A and B*, vol. 159, no. 1-2, pp. 253–287, Sep. 1, 2016, ISSN: 0025-5610. DOI: 10.1007/s10107-015-0957-3.

[283] C. Christof, “Gradient-based solution algorithms for a class of bilevel optimization and optimal control problems with a nonsmooth lower level,” *SIAM Journal on Optimization*, vol. 30, no. 1, pp. 290–318, Jan. 2020, ISSN: 1052-6234, 1095-7189. DOI: 10.1137/18M1225707.

[284] A. Shaban, C.-A. Cheng, N. Hatch, and B. Boots, “Truncated back-propagation for bilevel optimization,” in *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, PMLR, Apr. 11, 2019, pp. 1723–1732. [Online]. Available: https://proceedings.mlr.press/v89/shaban19a.html.

[285] D. P. Palomar and Y. C. Eldar, *Convex optimization in signal processing and communications*. Cambridge, 2011. DOI: 10.1017/CBO9780511804458.

[286] F. Pedregosa, “Hyperparameter optimization with approximate gradient,” in *Proceedings International Conference on Machine Learning*, M. F. Balcan and K. Q. Weinberger, Eds., vol. 48, PMLR, Jun. 20–22, 2016, pp. 737–46. [Online]. Available: http://proceedings.mlr.press/v48/pedregosa16.html.

[287] Y. Chen, T. Pock, R. Ranftl, and H. Bischof, “Revisiting loss-specific training of filter-based MRFs for image restoration,” in *Pattern Recognition*, J. Weickert, M. Hein, and B. Schiele, Eds., Springer Berlin Heidelberg, 2013, pp. 271–281, ISBN: 978-3-642-40602-7. DOI: 10.1007/978-3-642-40602-7_30.

[288] L. Franceschi, P. Frasconi, S. Salzo, R. Grazzi, and M. Pontil, “Bilevel programming for hyperparameter optimization and meta-learning,” in *International Conference on Machine Learning*, PMLR, Jul. 3, 2018, pp. 1568–1577. [Online]. Available: http://proceedings.mlr.press/v80/franceschi18a.html.

[289] M. Hintermüller, K. Papafitsoros, C. N. Rautenberg, and H. Sun, “Dualization and automatic distributed parameter selection of total generalized variation via bilevel optimization,” *Numerical Functional Analysis and Optimization*, pp. 1–46, 2022. DOI: 10.1080/01630563.2022.2069812.

[290] B. Sixou, “Adaptative regularization parameter for poisson noise with a bilevel approach: Application to spectral computerized tomography,” *Inverse Problems in Science and Engineering*, pp. 1–18, Dec. 22, 2020. DOI: 10.1080/17415977.2020.1864348.

[291] R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu, “A limited memory algorithm for bound constrained optimization,” *SIAM J. Sci. Comp.*, vol. 16, no. 5, 1190–208, 1995.
DOI: 10.1137/0916069.

[292] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in *3rd International Conference on Learning Representations*, vol. abs/1412.6980, May 2015. arXiv: 1412.6980.

[293] J. Fehrenbach, M. Nikolova, G. Steidl, and P. Weiss, “Bilevel image denoising using gaussianity tests,” in *International Conference on Scale Space and Variational Methods in Computer Vision*, vol. 9087, 2015, pp. 117–128, ISBN: 978-3-319-18460-9.
DOI: 10.1007/978-3-319-18461-6_10.

[294] B. Lecouat, J. Ponce, and J. Mairal, “A flexible framework for designing trainable priors with adaptive smoothing and game encoding,” in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 15 664–15 675. [Online]. Available: https://papers.nips.cc/paper/2020/hash/b4edda67f0f57e218a8e766927e3e5c5-Abstract.html.

[295] D. Kim and J. A. Fessler, “Adaptive restart of the optimized gradient method for convex optimization,” *J. Optim. Theory Appl.*, vol. 178, no. 1, 240–63, Jul. 2018.
DOI: 10.1007/s10957-018-1287-4.

[296] M. Hong, H.-T. Wai, Z. Wang, and Z. Yang, *A two-timescale framework for bilevel optimization: Complexity analysis and application to actor-critic*, Dec. 20, 2020. arXiv: 2007.05170.

[297] T. Chen, Y. Sun, Q. Xiao, and W. Yin, “A Single-Timescale Method for Stochastic Bilevel Optimization,” in *Proceedings of the 25th International Conference on Artificial Intelligence and Statistics (AISTATS) 2022*, vol. 151, 2022, pp. 2466–2488. [Online]. Available: https://proceedings.mlr.press/v151/chen22e.html.

[298] S. Ghadimi and M. Wang, *Approximation methods for bilevel programming*, Feb. 6, 2018. arXiv: 1802.02246.

[299] J. Yang, K. Ji, and Y. Liang, “Provably faster algorithms for bilevel optimization,” in *35th Conference on Neural Information Processing Systems (NeurIPS 2021)*, 2021. [Online]. Available: https://proceedings.neurips.cc/paper/2021/hash/71cc107d2e0408e60a3d3c44f47507bd-Abstract.html.

[300] P. Khanduri, H.-T. Wai, S. Zeng, M. Hong, Z. Wang, and Z. Yang, “A near-optimal algorithm for stochastic bilevel optimization via double-momentum,” in *35th Conference on Neural Information Processing Systems (NeurIPS 2021)*, 2021, p. 13. [Online]. Available: https://proceedings.neurips.cc/paper/2021/hash/fe2b421b8b5f0e7c355ace66a9fe0206-Abstract.html.

[301] Y. Nesterov, “A method of solving a convex programming problem with convergence rate $O(1/k^2)$,” *Soviet Math. Dokl.*, vol. 27, no. 2, 372–76, 1983.

[302] L. M. Nguyen, J. Liu, K. Scheinberg, and M. Takác, "SARAH: A novel method for machine learning problems using stochastic recursive gradient," in *34th International Conference on Machine Learning*, 2017, p. 9. [Online]. Available: https://proceedings.mlr.press/v70/nguyen17b.html.

[303] A. Mehra and J. Hamm, "Penalty method for inversion-free deep bilevel optimization," in *Proceedings of The 13th Asian Conference on Machine Learning*, PMLR, Nov. 28, 2021, pp. 347–362. [Online]. Available: https://proceedings.mlr.press/v157/mehra21a.html.

[304] L. Hoeltgen, S. Setzer, and J. Weickert, "An optimal control approach to find sparse data for Laplace interpolation," in *Energy Minimization Methods in Computer Vision and Pattern Recognition*, ser. Lecture Notes in Computer Science, A. Heyden, F. Kahl, C. Olsson, M. Oskarsson, and X.-C. Tai, Eds., red. by D. Hutchison, T. Kanade, J. Kittler, *et al.*, vol. 8081, Springer Berlin Heidelberg, 2013, pp. 151–164, ISBN: 978-3-642-40394-1. DOI: 10.1007/978-3-642-40395-8_12.

[305] D. Kim and J. A. Fessler, "On the convergence analysis of the optimized gradient method," *Journal of Optimization Theory and Applications*, vol. 172, no. 1, pp. 187–205, Jan. 2017, ISSN: 0022-3239, 1573-2878. DOI: 10.1007/s10957-016-1018-7.

[306] Y. Drori, "The exact information-based complexity of smooth convex minimization," *J. Complexity*, vol. 39, 1–16, Apr. 2017. DOI: 10.1016/j.jco.2016.11.001.

[307] D. Geman and G. Reynolds, "Constrained restoration and the recovery of discontinuities," *IEEE Trans. Patt. Anal. Mach. Int.*, vol. 14, no. 3, 367–83, Mar. 1992. DOI: 10.1109/34.120331.

[308] M. Nikolova and ,CMLA, ENS Cachan, CNRS, PRES UniverSud, 61 Av. President Wilson, F-94230 Cachan, "Model distortions in Bayesian MAP reconstruction," *Inverse Problems & Imaging*, vol. 1, no. 2, pp. 399–422, 2007, ISSN: 1930-8345. DOI: 10.3934/ipi.2007.1.399.

[309] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-D transform-domain collaborative filtering," *IEEE Transactions on Image Processing*, vol. 16, no. 8, pp. 2080–2095, Aug. 2007, ISSN: 1057-7149, 1941-0042. DOI: 10.1109/TIP.2007.901238.

[310] K. Bredies, K. Kunisch, and T. Pock, "Total generalized variation," *SIAM Journal on Imaging Sciences*, vol. 3, no. 3, pp. 492–526, Jan. 2010, ISSN: 1936-4954. DOI: 10.1137/090769521.

[311] A. Chambolle and P.-L. Lions, "Image recovery via total variation minimization and related problems," *Numerische Mathematik*, vol. 76, no. 2, pp. 167–188, Apr. 1, 1997, ISSN: 0029-599X, 0945-3245. DOI: 10.1007/s002110050258.

[312] M. Benning, C. Brune, M. Burger, and J. Müller, "Higher-order TV methods—Enhancement via Bregman iteration," *Journal of Scientific Computing*, vol. 54, no. 2-3, pp. 269–310, Feb. 2013, ISSN: 0885-7474, 1573-7691. DOI: 10.1007/s10915-012-9650-3.

[313] F. Knoll, K. Bredies, T. Pock, and R. Stollberger, “Second order total generalized variation (TGV) for MRI,” *Mag. Res. Med.*, vol. 65, no. 2, 480–91, 2011. DOI: 10.1002/mrm.22595.

[314] S. Setzer, G. Steidl, and T. Teuber, “Infimal convolution regularizations with discrete ℓ1-type functionals,” *Comm. Math. Sci.*, vol. 9, no. 3, 797–827, 2011. DOI: 10.4310/CMS.2011.v9.n3.a7.

[315] M. D’Elia, J. C. De los Reyes, and A. M. Trujillo, *Bilevel parameter optimization for learning nonlocal image denoising models*, Apr. 29, 2020. arXiv: 1912.02347.

[316] B. Gozcu, R. K. Mahabadi, Y.-H. Li, E. Ilicak, T. Cukur, J. Scarlett, and V. Cevher, “Learning-based compressive MRI,” *IEEE Trans. Med. Imag.*, vol. 37, no. 6, 1394–406, Jun. 2018. DOI: 10.1109/TMI.2018.2832540.

[317] N. Shlezinger, J. Whang, Y. C. Eldar, and A. G. Dimakis, *Model-based deep learning*, Dec. 15, 2020. arXiv: 2012.08405.

[318] V. Monga, Y. Li, and Y. C. Eldar, “Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing,” *IEEE Signal Processing Magazine*, vol. 38, no. 2, pp. 18–44, Mar. 2021. DOI: 10.1109/MSP.2020.3016905.

[319] K. Gregor and Y. LeCun, “Learning fast approximations of sparse coding,” in *Proc. Intl. Conf. Mach. Learn*, 2010. [Online]. Available: http://yann.lecun.com/exdb/publis/pdf/gregor-icml-10.pdf.

[320] W. Bian, Y. Chen, and X. Ye, “Deep parallel MRI reconstruction network without coil sensitivities,” in *Machine Learning for Medical Image Reconstruction*, F. Deeba, P. Johnson, T. Würfl, and J. C. Ye, Eds., ser. Lecture Notes in Computer Science, Springer International Publishing, 2020, pp. 17–26, ISBN: 978-3-030-61598-7. DOI: 10.1007/978-3-030-61598-7_2.

[321] K. Hammernik, T. Klatzer, E. Kobler, M. P. Recht, D. K. Sodickson, T. Pock, and F. Knoll, “Learning a variational network for reconstruction of accelerated MRI data,” *Magnetic Resonance in Medicine*, vol. 79, no. 6, pp. 3055–3071, 2018, ISSN: 1522-2594. DOI: 10.1002/mrm.26977.

[322] H. Lim, I. Y. Chun, Y. K. Dewaraja, and J. A. Fessler, “Improved low-count quantitative PET reconstruction with an iterative neural network,” *IEEE Transactions on Medical Imaging*, vol. 39, no. 11, pp. 3512–3522, Nov. 2020, ISSN: 0278-0062, 1558-254X. DOI: 10.1109/TMI.2020.2998480.

[323] S. Bai, J. Z. Kolter, and V. Koltun, “Deep equilibrium models,” in *Advances in Neural Information Processing Systems*, vol. 32, Curran Associates, Inc., 2019. [Online]. Available: https://proceedings.neurips.cc/paper/2019/hash/01386bd6d8e091c2ab4c7c7de644d37b-Abstract.html.

[324] J. Lorraine, P. Vicol, and D. Duvenaud, “Optimizing millions of hyperparameters by implicit differentiation,” in *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, PMLR, Jun. 3, 2020, pp. 1540–1552. [Online]. Available: https://proceedings.mlr.press/v108/lorraine20a.html.

[325] S. W. Fung, H. Heaton, Q. Li, D. McKenzie, S. Osher, and W. Yin, “JFB: Jacobian-free backpropagation for implicit networks,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022. arXiv: 2103.12803.

[326] H. Heaton, S. Wu Fung, A. Gibali, and W. Yin, “Feasibility-based fixed point networks,” *Fixed Point Theory and Algorithms for Sciences and Engineering*, vol. 2021, no. 1, p. 21, Dec. 2021, ISSN: 2730-5422. DOI: 10.1186/s13663-021-00706-3.

[327] S. V. Venkatakrishnan, C. A. Bouman, and B. Wohlberg, “Plug-and-Play priors for model based reconstruction,” in *2013 IEEE Global Conference on Signal and Information Processing*, IEEE, Dec. 2013, pp. 945–948, ISBN: 978-1-4799-0248-4. DOI: 10.1109/GlobalSIP.2013.6737048.

[328] J. Eckstein and D. P. Bertsekas, “On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators,” *Mathematical Programming*, vol. 55, no. 1-3, 293–318, Apr. 1992. DOI: 10.1007/BF01581204.

[329] J. He, Y. Yang, Y. Wang, D. Zeng, Z. Bian, H. Zhang, J. Sun, Z. Xu, and J. Ma, “Optimizing a parameterized plug-and-play ADMM for iterative low-dose CT reconstruction,” *IEEE Transactions on Medical Imaging*, vol. 38, no. 2, pp. 371–382, Feb. 2019, ISSN: 0278-0062, 1558-254X. DOI: 10.1109/TMI.2018.2865202.

[330] H. H. Barrett, “Objective assessment of image quality: Effects of quantum noise and object variability,” *J. Opt. Soc. Am. A*, vol. 7, no. 7, 1266–1278, Jul. 1990. DOI: 10.1364/JOSAA.7.001266.

[331] A. Yendiki and J. A. Fessler, “Analysis of observer performance in unknown-location tasks for tomographic image reconstruction,” *J. Opt. Soc. Am. A*, vol. 24, no. 12, B99–109, Dec. 2007. DOI: 10.1364/JOSAA.24.000B99.

[332] F. K. Kopp, M. Catalano, D. Pfeiffer, A. A. Fingerle, E. J. Rummeny, and P. B. Noel, “CNN as model observer in a liver lesion detection task for x-ray computed tomography: A phantom study,” *Med. Phys.*, vol. 45, no. 10, 4439–47, Oct. 2018. DOI: 10.1002/mp.13151.

[333] J. Xu and F. Noo, “Patient-specific hyperparameter learning for optimization-based CT image reconstruction,” *Physics in Medicine & Biology*, vol. 66, no. 19, 19NT01, Sep. 20, 2021, ISSN: 0031-9155. DOI: 10.1088/1361-6560/ac0f9a.

[334] J. Mairal, G. Sapiro, and M. Elad, “Learning multiscale sparse representations for image and video restoration,” *Multiscale Modeling & Simulation*, vol. 7, no. 1, pp. 214–241, Jan. 2008, ISSN: 1540-3459, 1540-3467. DOI: 10.1137/070697653.

[335] T. Liu, A. Chaman, D. Belius, and I. Dokmanić, *Learning multiscale convolutional dictionaries for image reconstruction*, Aug. 19, 2021. arXiv: 2011.12815.

[336] J. Kaipioa and E. Somersalo, “Statistical inverse problems: Discretization, model reduction and inverse crimes,” *J. Comp. Appl. Math.*, vol. 198, no. 2, 493–504, Jan. 2007. DOI: 10.1016/j.cam.2005.09.027.

[337] K. H. Jin, M. T. McCann, E. Froustey, and M. Unser, “Deep convolutional neural network for inverse problems in imaging,” *IEEE Trans. Im. Proc.*, vol. 26, no. 9, 4509–22, Sep. 2017. DOI: 10.1109/TIP.2017.2713099.

[338] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *J. Big Data*, vol. 6, no. 1, p. 60, Jul. 2019.
DOI: 10.1186/s40537-019-0197-0.

[339] J. Qi and R. H. Huesman, "Penalized maximum-likelihood image reconstruction for lesion detection," *Phys. Med. Biol.*, vol. 51, no. 16, 4017–30, Aug. 2006.
DOI: 10.1088/0031-9155/51/16/009.

[340] L. Yang, J. Zhou, A. Ferrero, R. D. Badawi, and J. Qi, "Regularization design in penalized maximum-likelihood image reconstruction for lesion detection in 3D PET," *Phys. Med. Biol.*, vol. 59, no. 2, 403–20, Jan. 2014. DOI: 10.1088/0031-9155/59/2/403.

[341] FDA, *510k premarket notification of Deep Learning Image Reconstruction (GE Medical Systems)*, 2019. [Online]. Available: https://www.accessdata.fda.gov/scripts/cdrh/cfdocs/cfpmn/pmn.cfm?ID=K183202.

[342] J. Solomon, P. Lyu, D. Marin, and E. Samei, "Noise and spatial resolution properties of a commercially available deep learning-based CT reconstruction algorithm," *Med. Phys.*, vol. 47, no. 9, 3961–71, 2020. DOI: 10.1002/mp.14319.

[343] C. Garcia-Cardona and B. Wohlberg, "Convolutional dictionary learning: A comparative review and new algorithms," *IEEE Transactions on Computational Imaging*, vol. 4, no. 3, pp. 366–381, Sep. 2018, ISSN: 2333-9403, 2334-0118, 2573-0436.
DOI: 10.1109/TCI.2018.2840334.

[344] J. Borwein and A. Lewis, "Fenchel Duality," in *Convex Analysis and Nonlinear Optimization: Theory and Examples*, ser. CMS Books in Mathematics, Springer, 2006, pp. 33–63, ISBN: 978-0-387-31256-9. DOI: 10.1007/978-0-387-31256-9_3.

[345] M. Unser and T. Blu, "Generalized smoothing splines and the optimal discretization of the Wiener filter," *IEEE Trans. Sig. Proc.*, vol. 53, no. 6, 2146–59, Jun. 2005.
DOI: 10.1109/TSP.2005.847821.

[346] C. Crockett, *BilevelFilterLearningForImageRecon*, 2022. [Online]. Available: https://github.com/cecroc/BilevelFilterLearningForImageRecon.