Extracting Compact Knowledge From Massive Data

by

Dejiao Zhang

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Electrical Engineering: Systems)
in The University of Michigan
2019

Doctoral Committee:

Assistant Professor Laura Balzano, Chair
Professor Mário A.T. Figueiredo, University of Lisbon, Portugal
Assistant Professor David Fouhey
Professor Alfred Hero
Assistant Professor Jenna Wiens

Dejiao Zhang

dejiao@umich.edu

ORCID iD: 0000-0002-1324-9388

To my parents.

ACKNOWLEDGEMENTS

The completion of this thesis marks the end of the beginning of a journey. First and foremost, I thank my advisor, Professor Laura Balzano. Ever since our first Skype meeting, Laura has been an excellent mentor and role-model. This dissertation would not be possible without her invaluable advice, continuous encouragement, and tremendous support. I am grateful for the freedom she has given to me to explore my research interests, and for her insights and immense patience to guide me through the research challenges. I am also thankful for those conversations, both mathematical and philosophical, with Laura, which have shaped my perspectives and will continue influence my road ahead.

I sincerely thank my excellent committee who have inspired and challenged me, including Professors Mário Figueiredo, Alfred Hero, David Fouhey, and Jenna Wiens. A special thank you goes to Professor Mário Figueiredo, for his guidance for Chapter III. Those meetings with him have always being inspiring and rewarding, thanks to his efforts in sharing and explaining his insights and vision with me. I am also grateful to Professor Jun He at NUIST for opening the door of research for me, and for his inspiration and patient guidance for my first research project. Lastly, I want to thank Professors Demos Teneketzis and Clayton Scott for their excellent teaching, which has profound influence on the way I interpret the research problems.

The past six years at Michigan as a graduate student has been very fruitful and enjoyable due to many peers. The first thank you goes to John Lipor and David

Hong. I am fortunate to start the Ph.D. adventure with them, and have benefited a lot from those enlightening discussions and fun chats with them. I also thank the other brilliant members of the SPADA lab, including Amanda Bower, Yutong Wang, Zhe Du, Kyle Gilman, Alex Ritchie, and Ali Soltani-Tehrani. And thanks especially to Amanda Bower and Yutong Wang, for their friendships. I have learned a lot from Amanda and Yutong, those conversations with them often prompt me to examine myself and help me find out the parts of my personality that require further improvement. Those times with them have become one of most precious memories of this journey. Lastly, I want to thank my wonderful collaborators, including Eric Zhao, Yitong Sun, Haozhu Wang, and Greg Ongie. Those long meetings with you have always been so inspiring and intellectually enjoyable.

I would also like to thank my friends outside of the SPADA lab, who have been the constant source of support and greatly enriched my life. First, I want to thank Ling Zhang, my best friend, not only for always listening and supporting me, but also for constantly asking me thought-provoking questions and challenging me to grow. I would also like to thank Hui Gao and Tatyana Saleski, for being energetic and inspiring, and for always encouraging me to achieve my own goals and reminding me of the good sides when I fell down. Most of all, I sincerely thank Fei Wen, for always listening and challenging me. Those good-natured debates with her always encourage me to deeply think about what I believe in and prepare me to stand-up for what I truly believe.

Finally, this dissertation is dedicated to my parents, for their unconditional love, support, and belief in me, and for always giving me enough freedom and support to make my own choices and take adventures.

TABLE OF CONTENTS

LIST OF TABLES

Table

LIST OF FIGURES

Figure

ABSTRACT

Over the past couple decades, we have witnessed a huge explosion in data generation from almost every perspective on our lives. Along with such huge volumes of data come more complex models, *e.g.,* deep neural networks (DNNs). This increase in complexity demands new trends in both modeling and analysis of data, among which low dimensionality and sparsity lie at the core. In this thesis, we follow this avenue to address some problems and challenges raised by modern data and models.

High-dimensional data are often not uniformly distributed in the feature space, but instead they lie in the vicinity of a low dimensional subspace. Identifying such low-dimensional structures cannot only give better interpretability of the data, but also significantly reduce the storage and computation costs for algorithms that deal with the data. The second chapter of this thesis focuses on low-rank linear subspace models, and we particularly focus on improving and analyzing an efficient subspace estimation method in the context of streaming data with emphasis on data being undersampled.

On the other hand, real word data are in general non-linear and involve much more complex dependencies, which motivates the development of DNNs. With massive amounts of data and computation power, the high capacity and the hierarchical structure of DNNs allow them to learn extremely complex non-linear dependencies and features. However, the successes achieved by DNNs are marred by the inscrutability of models, poor generalizability, and high demands on data and computational resources,

especially given that the size and the complexity of DNNs keeps increasing. To combat these challenges, we specifically focus on two perspectives, model compression and disentangled representation learning.

DNNs are often over-parameterized with many parameters being redundant and non-critical, hence successfully removing these connections is expected to improve both efficiency and generalization. In Chapter III, we go a step further by presenting a new method for compressing DNNs, which encourages sparsity while simultaneously identifying strongly correlated neurons and setting the corresponding weights to a common value. The ability of our method to identify correlations within the network not only helps further reduce the complexity of DNNs, but also allows us to cope with and gain more insights on the highly correlated neurons instead of being negatively affected by them.

From another perspective, many believe that the poor generalization and interpretability of DNNs can be resolved if the model can, in the setting of unsupervised learning, identify and separate out the underlying explanatory factors of data into different factors of its learned representation. Such representations are more likely to be used across a variety of tasks, with each particular task being relevant with a different subset or combination of all representation factors. In Chapter 4, we present an information theoretic approach for jointly learning a hybrid discrete-continuous representation, where the goal is to uncover the underlying categories of data while simultaneously separating the continuous representation into statistical independent components with each encoding a specific variation in data.

CHAPTER I

Introduction

1.1 Motivation

Data are currently being generated from ubiquitous sources with an unprecedented volume and variety, which together with the massive amount of computational power allow us to train more complex machine learning models, *e.g.,* deep learning models. Deep learning is built upon the computation technique of neural networks, with the hope to mimic the human brain and analyze data in a way similar to what a human would do. Over the past decade, deep learning has become one of the primary research areas in developing artificial intelligence, and it has been the working horse in solving a variety of practical problems, in particular for those in supervised learning [84, 122, 70, 74] and reinforcement learning [97, 98, 99, 117, 75].

Although deep learning has achieved tremendous success, we still need to overcome several challenges before deep learning is widely adopted in solving real life problems. In general, these challenges fall into three categories. First, deep learning is highly demanding on data and computational power. Deep learning models are usually of massive size, that is, a huge amount of parameters need to be tuned. More parameters generally require more data and more computational resources. Second, deep learning models suffer from the butterfly effect. As was initially observed in

[123], a small perturbation on the input data can cause even state-of-the-art neural networks to output completely incorrect predictions with high confidence, though such perturbations are imperceptible for humans. This vulnerability allows hackers to attack the model even without accessing the internal mechanism. Third, deep neural networks are essentially a black box. Due to the nonlinearity and the hierarchical structure, it's very hard to interpret the decision-making process of a deep learning model. This significantly limits the usage of deep learning models in mission-critical applications like financial services and clinical decision-making.

This thesis work takes a step towards resolving these challenges by targeting three different machine learning problems, namely, subspace learning from massive data, deep neural network compression, and interpretable representation learning, with the unified goal of extracting compact and interpretable knowledge from massive data. The underlying motivations can be summarized as the following:

- *Don't solve a harder problem than you have to.* Many real data are more likely to concentrate in the vicinity of a subspace of much lower dimension embedded in the original high dimensional space where data live. This motivates the usage of low rank subspace models in solving a wide range of problems in science and engineering for the sake of efficiency, ease of analysis, and better interpretability. Therefore, the low-dimensional linear subspace model is popular because of its ubiquitous success, and complicated nonlinear dimensionality reduction is not desirable unless it's absolutely necessary.

- *Less is more.* Compressing dense neural networks to sparse ones is often desired. Neural network compression not only tackles potential over-fitting but also accelerate the development of intelligent devices by significantly reducing the burden on computation and memory. Moreover, neural networks can be further

simplified by identifying strongly correlated input features for each layer and tying the associated connection weights together. By doing so, we thereby take a step towards opening the black box by identifying and coping with the highly correlated neurons instead of being negatively affected by them.

- *Build an explanatory model of data.* The vision of artificial intelligence is that the future world will be populated with intelligent agents that can fundamentally understand the world around us and reason and make decisions in an interpretable way. A widely believed idea is that such a goal can be achieved if the agents can, without a teacher (annotations), identify and disentangle the underlying explanatory factors of data into disjoint parts of the learned representations.

  In addition to providing more interpretability of the inner decision-making mechanism of the agent, such representations are more likely to be used over a variety of tasks, *i.e.,* each task depends on a different subset or combination of the whole set of representations factors.

1.2 Subspace Identification from Streaming Data

Many practical datasets exhibit low-dimensional structure. By leveraging low-dimensional structure of the given data, we are expected to (i) play with many fewer degrees of freedom, which can help improve efficiency when solving many important problems in machine learning and statistics; (ii) capture useful representations that are less variant to most local changes of the input, *e.g.,* in the presence of noise and outliers; (iii) simplify the analysis of complex (nonlinear) observation processes; (iv) gain more interpretability about the data.

Consider the following three illustrative examples. First, for surveillance video data analysis, background subtraction is a useful technique that can help us identify and

track the activities of moving objects in the foreground [68, 33]. In this scenario, if we stack the video frames as columns of a matrix, the static background with lighting changes naturally corresponds to a low-dimensional subspace. A second example is face recognition. With only the illumination changing, the human face can be well represented by a low-dimensional subspace. Identifying this subspace is crucial in many applications like face recognition [135] and image alignment [69, 105]. For a final example, consider recommender systems, where the goal is to use incomplete rankings provided by other users on some products to anticipate the preference of any user on any other products. The low dimensional assumption is key for solving this problem [82]. The philosophy behind this assumption is that we can use fewer categories to model diverse human behavior.

Therefore, subspace estimation is of great use in a variety of settings, *i.e.,* instead of the massive data itself, the subspace spanned by the these principal components is the object of interest and requires less storage and communication complexity. However, to efficiently identify the subspace, we need to handle the following three challenges arising regularly in modern machine learning problems and data-intensive analysis. First of all, data can be high-dimensional and observed in a streaming way. The velocity of high-dimensional data being generated is continuing to reach unprecedented levels, *e.g.,* huge volumes of photos and video data are generated from ubiquitous sources, moreover, each single frame may contain thousands or ten thousands of pixels. Secondly, data are likely incomplete. Two typical cases are missing data and compressed sensing data, which can arise due to the limitations on the way data are being collected [132, 82], or privacy settings in social networks, or the technology used by a specific area like Magnetic resonance imaging (MRI) for medical diagnosis [9]. The last challenge is the *theoretical guarantees for proposed*

algorithms. Until recently, most known methods with rigorous guarantees for solving this type of problem use a batch data model [34, 106, 77, 79, 88, 78], which requires expansive storage and computation costs and thus prevents efficient processing of large data.

In Chapter II, we focus on extending an existing algorithm for subspace estimation from streaming data to a more general sampling scheme and providing theoretical guarantees correspondingly.

1.3 Simultaneous Sparsification and Parameter Tying in Deep Learning

Deep neural networks (DNNs) have recently revolutionized machine learning by dramatically advancing state-of-the-art performance in many applications, including computer vision [70, 62], natural language processing [13, 49], playing video games [96] and healthcare [95]. The key behind this widespread success is that a typical DNN usually contains millions or even billions of parameters/weights, which together with the hierarchical structure of DNN and massive amounts of data allow it to learn very complex mappings and perform inferences at multiple levels. On the other hand, although a larger DNN usually yields better performance [61, 51], such high capacity makes both storage and computation very expensive, resulting in fundamental challenges in deploying DNNs in devices with limited resources, *e.g.,* cell phones, smart wearable devices, drones and self-driving cars. However, it has been shown that many popular deep neural network architectures are over-parameterized [47], where a large fraction of the parameters can be predicted from the remaining ones with no accuracy loss. This motivates a surge of research interest to compress neural networks with benefits being twofold: (i) significantly reducing the complexity of DNNs so as to improve efficiency and generation; (ii) facilitating the deployment

of a larger base architecture to obtain a better baseline for further compression.

A natural approach for neural network compression is to remove redundant connections. Two typical methods are pruning [65, 89] and sparsity regularization [146, 134]. Parameter sharing/tying is another well-known approach for controlling the complexity of DNNs by forcing certain sets of weights to share a common value. Some forms of weight sharing are hard-wired to express certain invariances; a notable example is the shift-invariance of convolutional layers. However, other groups of weights may be tied together during the learning process to further reduce the network complexity. In Chapter III, we adopt a recently proposed regularizer, GrOWL (*group ordered weighted* ℓ_1), which encourages sparsity and, simultaneously, learns which groups of parameters should share a common value.

GrOWL has proven effective in linear regression, being able to identify and cope with strongly correlated covariates. Unlike standard sparsity-inducing regularizers (*e.g.*, ℓ_1 a.k.a. Lasso), GrOWL not only eliminates unimportant neurons by setting all their weights to zero, but also explicitly identifies strongly correlated neurons by tying the corresponding weights to a common value. This ability of GrOWL motivates the following two-stage procedure: (i) using the GrOWL regularizer during training to simultaneously identify significant neurons and groups of parameters that should be tied together; (ii) retraining the network, enforcing the structure that was unveiled in the previous phase, *i.e.*, keeping only the significant neurons and enforcing the learned tying structure. As shown in Chapter III, being capable of identifying highly correlated neurons and tying the assoicated parameters/weights together yields two desirable effects in terms of better stability and prediction. First, unlike the standard ℓ_1 type regularization, GrOWL is capable of selecting all relevant features instead of a random subset of them, which guarantees a more stable learning process and better

interpretability of the model. Second, the parameter tying property helps denoise the input data and alleviate co-adaption in DNNs so as to improve prediction.

1.4 Interpretable Representation Learning

Building explanatory models of data lies at the core of artificial intelligence (AI) [23]. Such explainable models have profound impact on various important topics, including safe AI, fairness and bias in social science, and automatic scientific discovery. This has led to a surge of interests in the deep learning community in learning disentangled representation. Although there is no generally agreed upon definition of disentanglement, many believe learning a factorial representation is a good starting point, given the hypothesis that data has been generated by a number of independent factors through a stochastic random process. Specifically, such a factorial representation is expected to be semantically meaningful in a way that a change in one dimension of the representation corresponds to a change in one true factor of variation in data, while being invariant to changes in other factors. Successfully learning the disentangled representations is useful in a variety of tasks, including enabling interpretable decision making in downstream tasks like supervised learning and reinforcement learning, allowing explainable knowledge transfer in transfer learning, and performing controllable data generation in generative models.

A large amount of research has focused on learning disentangled representations in supervised learning and semi-supervised learning where the annotations can either explicitly or implicitly guide what disentanglement means. However, unsupervised learning is more desired, sometimes even necessary, given that it's more human-like and annotations are costly to obtain and hence are scarce in practice. Recently, deep generative models have shown great promise in unsupervised learning of disentangled

representations, where variational autoencoders (VAE) [81] and generative adversarial networks (GAN) [63] based approaches are arguably the two most influential lines. The original objectives of both VAE and GAN solely target data reconstruction/generation fidelity, thus tending to encode different variations in data in a highly entangled way. Many follow up works augment the original objectives with various disentanglement-encouraging terms, with the goal of finding the optimal trade-off between reconstruction and disentanglement [71, 32, 80, 35, 37, 52, 92, 55].

In Chapter IV, we propose the Regularized Information Maximizing Auto-Encoding (RIMAE), an information theoretic approach to learning hybrid discrete-continuous representations in an unsupervised setting. Instead of building upon the generative models, we start with a very natural criterion that good representations, at least to some degree, should be informative about the data. This motivates us to maximize the mutual information between data and its representations, which is the substance of the InfoMax principle [90]. Although the mutual information between data and its representations can be trivially maximized by simply memorizing the data, proper constraints are naturally implied by our target. We show that the proposed objective provides a principled framework for understanding the relationships among the informativeness of each representation factor, disentanglement of representations, and decoding quality.

1.5 Outline and Publications

This thesis proposal is organized as follows. Chapter II provides the theoretical guarantees of a Grassmannian gradient descent algorithm for subspace estimation from both fully sampled and undersampled data. Chapter III describes a novel method for compressing neural networks by simultaneously encouraging sparisification

and parameter tying. Chapter IV presents an information theoretic approach for simultaneously learning discrete and continuous representations of data by leveraging stochastic autoencoder. Chapter V provides conclusion and discussions on future work.

Publications

Chapter II

- Dejiao Zhang and Laura Balzano. "Convergence of a Grassmannian Gradient Descent Algorithm for Subspace Estimation from Undersampled Data". *In Preparation for Journal of the Society for the Foundations of Computational Mathematics (FOCM).*
- Dejiao Zhang and Laura Balzano. "Global Convergence of a Grassmannian Gradient Descent Algorithm for Subspace Estimation". *In Proceedings of the 19th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2016.
- Dejiao Zhang and Laura Balzano. "Matched Subspace Detection Using Compressively Sampled Data". *In Proceedings of the 42nd IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017.
- Greg Ongie, David Hong, Dejiao Zhang, Laura Balzano. Online estimation of coherent subspaces with adaptive sampling. *In IEEE Statistical Signal Processing Workshop (SSP)*, 2018.
- Greg Ongie, David Hong, Dejiao Zhang, Laura Balzano. Enhanced Online Subspace Estimation via Adaptive Sensing. *In 51st Asilomar Conference on Signals, Systems, and Computers*, Asilomar 2017.

Chapter III

- Dejiao Zhang, Haozhu Wang, Mário A.T. Figueiredo, Laura Balzano. "Learning to share: Simultaneous Parameter Tying and Sparsification in Deep Learning". *In the Sixth International Conference on Learning Representations (ICLR)*, 2018.
- Dejiao Zhang, Julian Katz-Samuels, Mário A.T. Figueiredo, Laura Balzano. "Simultaneous sparsity and parameter tying for deep learning using ordered weighted L1 regularization". *In IEEE Statistical Signal Processing Workshop (SSP)*, 2018.

Chapter IV

- Dejiao Zhang and Laura Balzano. "Regularized Information Maximization Auto-Encoding". *In preparation for JMLR.*
- Dejiao Zhang, Tianchen Zhao, Laura Balzano. "Information Maximization Auto-Encoding". *Workshop on Bayesian Deep Learning (BDL)*, NeurIPS 2018.

Other

- Dejiao Zhang, Yifan Sun, Brian Eriksson, Laura Balzano. "Deep Unsupervised Clustering with Mixture of Autoencoders". *UMich Deep Blue Technical Report*, 2018.
- Tianchen Zhao, Dejiao Zhang, Zeyu Sun, Honglak Lee. "Information Regularized Neural Networks". *Workshop on Integration of Deep Learning Theories*, NeurIPS 2018.

CHAPTER II

Convergence of a Grassmannian Gradient Descent Algorithm for Subspace Estimation From Undersampled Data

2.1 Introduction

Low-rank matrix factorization is an essential tool for high-dimensional inference with fewer measurements than variables of interest, where low-dimensional models are necessary to perform accurate and stable inference. Many modern problems fit this paradigm, where signals are undersampled because of sensor failure, resource constraints, or privacy concerns. Suppose we wish to factorize a matrix $M = UW^T$ when we only get a small number of linear measurements of M. Solving for the subspace basis U can be computationally burdensome in this undersampled problem and related regularized problems. Many algorithms that attempt to speed up computation are solving a non-convex optimization problem, and therefore come with few guarantees.

The Singular Value Decomposition (SVD) provides the solution to the non-convex matrix factorization problem formulation with full data, and there are several highly successful algorithms for solving it [59]. Unfortunately, these algorithms cannot easily be extended to problems with incomplete observations of the matrix. Recently, several results have been published with first-of-their-kind guarantees for a variety of different gradient-type algorithms on non-convex matrix factorization problems

[11, 25, 39, 46, 76, 77, 145]. These new algorithms, being gradient-based, are well-suited to extensions of the SVD where the matrix is not fully sampled and where we include different cost functions or regularizers. For example, with gradient methods to solve the SVD we may be able to solve Robust PCA [33, 68, 137], Sparse PCA [44], or even ℓ_1 PCA [29] with gradient methods as well. However, almost none of these results gives guarantees in *streaming problem*, where data can only be accessed one partial column vector at a time. This is a critical problem in the modern machine learning context with massive data and comparatively limited memory, or in applications where data are collected continuously and must be processed in realtime. The existing theoretical results for the streaming problem significantly overestimate the number of samples needed for convergence for typical algorithms.

Our contribution is to provide a global convergence result for d-dimensional subspace estimation using an incremental gradient algorithm performed on the Grassmannian, the space of all d-dimensional subspaces of $\mathbb{R}^n$, denoted by $\mathcal{G}(n, d)$. Subspace estimation is a special case of matrix factorization with orthogonality constraints, where we seek to estimate only the subspace spanned by the columns of the left matrix factor $U \in \mathbb{R}^{n \times d}$. Our result demonstrates that, for fully sampled data without noise, this gradient algorithm *converges globally to the global minimizer* almost surely, *i.e.,* it converges from any random initialization to the global minimizer. For undersampled data, including compressively sampled data and missing data, we provide results showing monotonic improvement in expectation on the metric of convergence for each iteration.

This chapter is organized as follows. The problem formulation and the GROUSE algorithm are described in Section 2.2. The global convergence result for fully sampled data is presented in Section 2.4, the convergence behavior of GROUSE

with undersampled data is studied in Section 2.5, and the corresponding proofs are provided in Sections 2.8.1, 2.8.2 and 2.8.3. Experiment results are in Section 2.6.

2.2 Problem Setting

in this chapter, we consider the problem of learning a low dimensional subspace representation from streaming data. Specifically, we are given a sequence of observations $x_t = A_t v_t$ where $A_t \in \mathbb{R}^{m \times n}$ $(m \leq n)$ are sampling matrices that are given for each observation; $v_t \in \mathbb{R}^n$ are drawn from a continuous distribution with support on the true subspace, spanned by $\bar{U} \in \mathbb{R}^{n \times d}$ with orthonormal columns, *i.e.,* $v_t = \bar{U} s_t, s_t \in \mathbb{R}^d$. in this chapter, we study three different sampling frameworks: the fully sampled case with A_t being the identity matrix, the compressively sampled case with $A_t \in \mathbb{R}^{m \times n}$ $(m \ll n)$ being random Gaussian matrices, and the missing data case where each row of A_t $(m \ll n)$ is uniformly sampled from the identity matrix.

We formulate subspace estimation as a non-convex optimization problem as follows. Let $U \in \mathbb{R}^{n \times d}$ be a matrix with orthonormal columns. Then we want to solve:

$$\begin{aligned} \underset{U \in \mathbb{R}^{n \times d}}{\text{minimize}} \quad & \sum_{t=1}^{T} \min_{w_t} \|A_t U w_t - x_t\|_2^2 \\ \text{subject to} \quad & \text{span}\,(U) \in \mathcal{G}(n, d) \end{aligned} \tag{2.1}$$

This problem is non-convex firstly because of the product of the two variables U and w_t and secondly because the optimization is over the Grassmannian $\mathcal{G}(n, d)$, the non-convex set of all d-dimensional subspaces in $\mathbb{R}^n$. We study an online algorithm to solve the above problem, where we process one observation at a time and perform a rank-one update to generate a sequence of estimates U_t with the goal that $R(U_t) \to R(\bar{U})$, where $R(\cdot)$ denotes the column range.

We can see the relationship between our problem and the well studied low-rank matrix recovery problem. Let $W \in \mathbb{R}^{d\times T}$ and $M = [v_1, \ldots, v_T] \in \mathbb{R}^{n\times T}$, then (2.1) is equivalent to

$$\begin{aligned} \underset{U\in\mathbb{R}^{n\times d}, W\in\mathbb{R}^{d\times T}}{\text{minimize}} \quad & \|\mathcal{A}(UW) - \mathcal{A}(M)\|_2^2 \\ \text{subject to} \quad & \text{span}(U) \in \mathcal{G}(n,d) \end{aligned} \tag{2.2}$$

where $\mathcal{A} : \mathbb{R}^{n\times T} \to \mathbb{R}^{mT}$ is a linear operator. Our algorithm can be thought of as an incremental algorithm to solve this problem as well. Fueled by the great deal of recent success of directly solving non-convex factorization problems (as we discuss in related work below), we study the natural incremental gradient descent algorithm [24] applied to (2.1) directly. Since the optimization variable in our problem is a subspace, we constrain the gradient descent to the Grassmannian $\mathcal{G}(n,d)$. The resulting algorithm is called GROUSE (Grassmannian Rank-One Update Subspace Estimation) algorithm and is described in Algorithm 1. This description differs from its initial introduction in [15] in that it extends the missing data case to a more general sampling framework.

Algorithm 1 GROUSE: Grassmannian Rank-One Update Subspace Estimation

Given U_0, an $n \times d$ matrix with orthonormal columns, with $0 < d < n$;
Set $t := 0$;
repeat
  Given sampling matrix $A_t : \mathbb{R}^n \to \mathbb{R}^m$ and observation $x_t = A_t v_t$;
  Define $w_t := \arg\min_a \|A_t U_t a - x_t\|^2$;
  Define $p_t := U_t w_t$ and $\widetilde{r}_t := x_t - A_t p_t$, $r_t := A_t^T \widetilde{r}_t$;
  Using step size

$$\theta_t = \arctan\left(\frac{\|r_t\|}{\|p_t\|}\right) \tag{2.3}$$

  update with a gradient step on the Grassmannian:

$$U_{t+1} := U_t + \left(\frac{y_t}{\|y_t\|} - \frac{p_t}{\|p_t\|}\right)\frac{w_t^T}{\|w_t\|} \tag{2.4}$$

  where

$$\frac{y_t}{\|y_t\|_2} = \frac{p_t}{\|p_t\|_2}\cos(\theta_t) + \frac{r_t}{\|r_t\|_2}\sin(\theta_t)$$

  $t := t + 1$;
until termination

2.2.1 Algorithm

At each step, the GROUSE algorithm receives a vector $x_t = A_t v_t$, and tries to minimize the inconsistency between $R(U)$ and the true subspace $R(\bar{U})$ with respect to the information revealed in the sampled vector x_t, *i.e.,*

$$\mathcal{F}(U;t) = \min_a \|A_t U a - x_t\|^2 \tag{2.5}$$

In order to do so, GROUSE forms the gradient of $\mathcal{F}$ with respect to U evaluated at the current estimate U_t, and takes a step in the direction of the negative gradient restricted to the Grassmannian. The derivation of the incremental gradient descent update rule on the Grassmannian is found in [15, 18], and we summarize it here.

To compute the gradient of $\mathcal{F}$ on the Grassmannian, we first need to compute the derivative of $\mathcal{F}$ with respect to U and evaluate it at U_t. As we will prove later, under mild conditions, $A_t U_t$ has full column rank with high probability. Therefore, the derivative is

$$\frac{d\mathcal{F}}{dU} = -2A_t^T \widetilde{r}_t w_t^T \tag{2.6}$$

where $\widetilde{r} := x_t - A_t U_t w_t$ denotes the residual vector with respect to the sampled vector x_t, and w_t is the least-squares solution of (2.5). Using Equation (2.70) in [53], the gradient of $\mathcal{F}$ on the Grassmannian then follows as

$$\begin{aligned} \nabla\mathcal{F} = \left(I - U_t U_t^T\right) \frac{d\mathcal{F}}{dU} &= -2\left(I - U_t U_t^T\right) A_t^T \widetilde{r}_t w_t^T \\ &= -2A_t^T \widetilde{r}_t w_t^T \ . \end{aligned} \tag{2.7}$$

The final equality follows by $\widetilde{r}_t \perp A_t U_t$, which can be verified using the definitions of w_t and $\widetilde{r}_t$. According to Eq (2.65) in [53], a gradient step along the geodesic with tangent vector $-\nabla\mathcal{F}$ can be then formed as a function of the singular values and

singular vectors of $\nabla\mathcal{F}$. For this specific case of our rank one $\nabla\mathcal{F}$ given in 2.7, the update rule follows as

$$U(\eta) = U_t + \left[(\cos(\eta_t\sigma_t) - 1)\frac{U_t w_t}{\|w_t\|} + \sin(\eta_t\sigma_t)\frac{A_t^T\widetilde{r}_t}{\|A_t^T\widetilde{r}_t\|}\right]\frac{w_t^T}{\|w_t\|} \tag{2.8}$$

where $\eta_t > 0$ is the chosen step size at iteration t, $p_t := U_t w_t$ is the predicted value of the projection of the vector v_t onto $R(U_t)$ and $\sigma_t = \|A_t^T\widetilde{r}_t\|\|p_t\|$. By leveraging the fact that $\widetilde{r}_t \perp A_t U_t$ and $p_t \in R(U_t)$, it's easy to verify that the rank-one update (2.8) maintains orthogonality $U(\eta)^T U(\eta) = \mathbb{I}_d$, and tilts $R(U_t)$ to a new point on Grassmannian.

In summary, for each observation the GROUSE algorithm works as follows: it projects the data vector onto the current estimate of the true subspace with respect to the sampling matrix A_t, to get either the exact (when $A_t = \mathbb{I}_n$) or approximated projection p_t and residual $r_t = A_t^T\widetilde{r}_t$. Then GROUSE updates the current estimate with a rank-one step as described by (2.4). In the present work, we propose an adaptive stepsize framework that sets the stepsize only based on the sampled data and the algorithm outputs. More specifically, at each iteration a stepsize η_t is chosen such that $\eta_t\sigma_t = \arctan\left(\frac{\|r_t\|}{\|p_t\|}\right)$. As shown in Section 2.4, the proposed stepsize scheme is greedy for the fully sampled data, *i.e.,* it maximizes the improvement of our defined convergence metric at each iteration. For the undersampled data, we establish a local convergence result by showing that, with the proposed stepsize, GROUSE moves the current estimated subspace towards the true subspace with high probability despite the nonconvex nature of the problem and undersampled data.

2.2.2 Related Work

Many recent results have shown theoretical support for directly solving non-convex matrix factorization problems with gradient or alternating minimization

methods. Among the incremental methods [46] is the one closest to ours, where the authors consider recovering a positive semidefinite matrix with undersampled data. They propose a step size scheme with which they prove global convergence results from a randomly generated initialization. However, their convergence results contain a obscure term, and their choice of step size depends on the knowledge of some parameters that are likely to be unknown in practical problems. Without this knowledge, the results only hold with sufficiently small step size that implies significantly slower convergence. In contrast, while our work applies more narrowly to the subspace estimation problem, we provide an explicit expression for the expected improvement at each iteration, using a step size that only depends on the observations and outputs of the algorithms. Based on that, we posit a conjecture on the global convergence rate for fully sampled data. Although we have not yet established a complete proof of this conjecture, it's a very promising result that matches what we have seen in practice. We present our current proof of the conjecture in Appendix 2.8.2, and discuss the missing steps for finally validating it. Other work that has looked at incremental methods has focused only on fully sampled vectors. For example, [14] invokes a martingale-based argument to derive the global convergence rate of the proposed incremental PCA method to the single top eigenvector in the fully sampled case. In contrast, [12] estimates the best d-dimensional subspace in the fully sampled case and provides a global convergence result by relaxing the non-convex problem to a convex one. We seek to identify the d dimensional subspace by solving the non-convex problem directly.

The results in this chapter are very closely related to our previous work [17]. In [17], we prove that, within a local region of the true subspace, an expected improvement of their defined convergence metric for each iteration of GROUSE can be obtained.

In contrast, we establish global convergence results to a global minimizer from any random initialization for fully sampled data, and extend the local convergence results to compressively sampled data. We also expand the local convergence results in [17] to a much less conservative region, and we provide a much simpler analysis framework that can be applied to different sampling strategies. Moreover, for each iteration of the GROUSE algorithm, the expected improvement on the convergence metric defined in [17] only holds locally in both theory and practice, while our theoretical result provides a tighter bound for the global convergence behavior of GROUSE over a variety of simulations. This suggests that our result has more promise to be extended to a global result for both missing data and compressively sampled data.

Turning to batch methods, [108, 77] provided the first theoretical guarantee for an alternating minimization algorithm for low-rank matrix recovery in the undersampled case. Under typical assumptions required for the matrix recovery problems [106], they established geometric convergence to the global optimal solution. Earlier work [78, 102] considered the same undersampled problem formulation and established convergence guarantees for a steepest descent method (and a preconditioned version) on the full gradient, performed on the Grassmannian. [39, 25, 145] considered low rank semidefinite matrix estimation problems, where they reparamterized the underlying matrix as $M = UU^T$, and update U via a first order gradient descent method. However, all these results require batch processing and a decent initialization that is close enough to the optimal point, resulting in a heavy computational burden and precluding problems with streaming data. We study random initialization, and our algorithm has fast, computationally efficient updates that can be performed in an online context.

Lastly, several convergence results for optimization on general Riemannian mani-

folds, including several special cases for the Grassmannian, can be found in [2]. Most of the results are very general; they include global convergence rates to local optima for steepest descent, conjugate gradient, and trust region methods, to name a few. We instead focus on solving the problem in equation 2.1 and provide global convergence rates to the global minimum.

Before we present the main results, we first call out the following notation which we use throughout this chapter. For notational convenience, we will drop the iteration subscript except our convergence metric ζ_t defined in Definition 1 hereafter.

Notation We use $R(M)$ to denote the column space of a matrix M and $\mathcal{P}_M$ to denote the orthogonal projection onto $R(M)$. $\mathbb{I}_n$ denotes the identity matrix in $\mathbb{R}^{n\times n}$ and M_i denotes the i^{th} row of matrix M. in this chapter, without specification, $\|\cdot\|$ denotes the ℓ_2 norm. $R(\bar{U})$ and $R(U)$ denote the true subspace and our estimated subspace respectively, here both $\bar{U}$ and U are matrices in $\mathbb{R}^{n\times d}$ with orthonormal columns. Also we use $v_\|$ and $v_\perp$ to denote the projection and residual of the underlying full vector $v \in \mathbb{R}^n$ onto the estimated subspace $R(U)$, *i.e.,* $v_\| = UU^T v, v_\perp = v - v_\|$. Note that these two quantities are in general unknown for the undersampled data case. We define them so as to relate the intermediate quantities, determined by the algorithm and sampled data, to the improvement on our defined convergence metric.

2.3 Preliminaries

In this section, we first define our convergence metric and describe an assumption on the streaming data needed to establish our results. Subsequently, we state a fundamental result that is essential to quantify the improvement on the convergence metric over GROUSE iterates.

Definition 1 (Determinant similarity)**.** *Our measure of similarity between $R(U)$ and*

$R(\bar{U})$ is $\zeta \in [0,1]$, defined as

$$\zeta := \det(\bar{U}^T U U^T \bar{U}) = \prod_{k=1}^{d} \cos^2 \phi_k \, .$$

where ϕ_k denotes the k^{th} principal angle between $R(\bar{U})$ and $R(U)$ (See [[120], Chapter 5]) by $\cos \phi_k = \sigma_k(\bar{U}^T U)$ with σ_k denoting the k^{th} singular value of $\bar{U}^T U$.

The convergence metric ζ increases to one when our estimate $R(U)$ converges to $R(\bar{U})$, *i.e.,* all principal angles between the two subspaces equal zero. Compared to other convergence metrics defined either as $\|(I - \bar{U}\bar{U}^T)U\|_F^2 = d - \|\bar{U}^T U\|_F^2 = \sum_{k=1}^{d} \sin^2 \phi_k$ or $1 - \|\bar{U}^T U\|_2^2 = \sin^2 \phi_1$, our convergence metric ζ measures the similarity instead of the discrepancy between $R(U)$ and $R(\bar{U})$. In other words, ζ achieves its maximum value one when $R(U)$ converges to $R(\bar{U})$, while the typical subspace distance is zero when the subspaces are equal. Also note that $\zeta = 0$ iff at least one of the principal angles is a right angle. That is, all stationary points U_{stat} of the full data problem except the true subspace have $\det\left(\bar{U}^T U_{stat} U_{stat}^T \bar{U}\right) = 0$ [138, 18].

Assumption 1. *For the underlying data $v = \bar{U}s$, we assume the entries of s are independent, and identically distributed symmetrically about zero, and each entry has zero-mean and unit variance.*

Given this assumption, we have the following lemma which relates the projection $v_\parallel$ and the projection residual $v_\perp$ to the improvement on our convergence metric ζ_t. As we will show in the following sections, this lemma is crucial for us to establish the expected improvement on our defined convergence metric ζ_t for all the sampling frameworks considered in this work. The proof is provided in Section 2.8.1.

Lemma 2.3.1. *Let $v_\parallel$ and $v_\perp$ denote the projection and residual of the full data sample v onto the current estimate $R(U)$. Then given Assumption 1, for each iteration*

of GROUSE we have

$$\mathbb{E}\left[\frac{\|v_\perp\|^2}{\|v_\|\|^2}\middle| U\right] \geq \mathbb{E}\left[\frac{\|v_\perp\|^2}{\|v\|^2}\middle| U\right] \geq \frac{1-\zeta_t}{d}\ . \tag{2.9}$$

Although both projection ($v_\|$) and projection residual ($v_\perp$) are in general unknown for the undersampled data, we can relate the approximated projection residual $A^T\widetilde{r}$ to the true one $v_\perp$ by leveraging either random matrix theory or the incoherence property of the underlying subspace $R(\bar{U})$. Therefore, the above lemma provides a unifying step to quantify the improvement on the convergence metric for all cases considered in the present work.

2.4 Fully Sampled Data

In this section, we consider fully sampled data, *i.e.,* $A = \mathbb{I}_n$. The corresponding proofs for these results can be found in Section 2.8.2. We start by deriving a greedy step size scheme for each iteration t that maximizes the improvement on our convergence metric ζ_t. For each update we prove the following:

$$\frac{\zeta_{t+1}}{\zeta_t} = \left(\cos\theta + \frac{\|v_\perp\|}{\|v_\|\|}\sin\theta\right)^2 . \tag{2.10}$$

It then follows that

$$\theta^* = \arg\max_\theta \frac{\zeta_{t+1}}{\zeta_t} = \arctan\left(\frac{\|v_\perp\|}{\|v_\|\|}\right). \tag{2.11}$$

This is equivalent to (2.3) in the fully sampled setting $A_t = \mathbb{I}_n$. Using θ^*, we obtain monotonic improvement on the determinant similarity that can be quantified by the following lemma.

Lemma 2.4.1 (Monotonicity for the fully sampled noiseless case)**.** *For fully sampled data, choosing step size* $\theta^* = \arctan\left(\frac{\|v_\perp\|}{\|v_\|\|}\right)$*, after one iteration of GROUSE we obtain*

$$\frac{\zeta_{t+1}}{\zeta_t} = 1 + \frac{\|v_\perp\|^2}{\|v_\|\|^2} \geq 1\ .$$

To gain more insight into the improvement on ζ_t for each iteration of GROUSE, we call out the following lemma, which is a natural result of Lemma 2.3.1 and Lemma 2.4.1.

Lemma 2.4.2 (Expected improvement on ζ_t)**.** *When fully sampled data satisfying Assumption 1 are input to the GROUSE (Algorithm 1), the expected improvement after one update step is given as:*

$$\mathbb{E}\left[\zeta_{t+1}\middle|U\right] \geq \left(1 + \frac{1-\zeta_t}{d}\right)\zeta_t \ .$$

Under the mild assumption that each data vector is randomly sampled from the underlying subspace, we obtain strict improvement on ζ_t for each iteration provided $\|v_\perp\| > 0$ and $\|v_\parallel\| > 0$. Therefore, Lemma 2.4.1 provides insight into how the GROUSE algorithm converges to the global minimum of a non-convex problem formulation: GROUSE is not attracted to stationary points that are not the global minimum. As we mentioned previously, all other stationary points U_{stat} have $\det(\bar{U}^T U_{stat} U_{stat}^T \bar{U}) = 0$, because they have at least one direction orthogonal to $\bar{U}$ [18]. Therefore, if the initial point U_0 has determinant similarity with $\bar{U}$ strictly greater than zero, then we are guaranteed to stay away from other stationary points, since GROUSE increases the determinant similarity monotonically, according to Lemma 2.4.1. This together with Lemma 2.4.2 yields the following convergence result of GROUSE.

Theorem 2.4.1 (Convergence of GROUSE)**.** *Initialize the starting point U_0 of GROUSE such that $\zeta_0 > 0$. Let $1 \geq \zeta^* \geq \zeta_0$ be the desired accuracy of our estimated*

subspace. Then for any $\rho > 0$, after

$$K \geq \left(\frac{d}{\zeta_0} + 1\right) \log\left(\frac{1}{\rho(1-\zeta^*)}\right)$$

iterations of GROUSE Algorithm 1,

$$\mathbb{P}\left(\zeta_K \geq \zeta^*\right) \geq 1 - \rho \; .$$

.

Notice that if we initialize GROUSE with U_0 drawn uniformly from the Grassmannian, *e.g.,* as the orthonormal basis of a random matrix $V \in R^{n\times d}$ with entries being independent standard Gaussian variables, this guarantees $\zeta_0 > 0$ with probability one. Therefore, Theorem 2.4.1 provides a global convergence result of GROUSE despite the non-convexity of our objective. However, with this randomly initialized U_0, the value of the associated determinant similarity ζ_0 is $\mathcal{O}\left(\left(\frac{d}{n}\right)^d\right)$. Thereby, GROUSE requires $\mathcal{O}\left(d\left(\frac{n}{d}\right)^d\right)$ iterations to converge to the required precision, which is too pessimistic compared to the actual number of iterations required by GROUSE. To narrow this gap, we call out the following conjecture on the global convergence rate for GROUSE.

Conjecture 2.4.1 (Global Convergence of GROUSE)**.** *Let $1 \geq \zeta^* > 0$ be the desired accuracy of our estimated subspace. With the initialization (U_0) of GROUSE as the range of an $n \times d$ matrix with entries being i.i.d standard normal random variables, then for any $\rho > 0$, after*

$$\begin{aligned} K &\geq K_1 + K_2 \\ &= \left(\frac{2d^2}{\rho} + 1\right) \tau_0 \log(n) + 2d \log\left(\frac{1}{2\rho(1-\zeta^*)}\right) \end{aligned}$$

iterations of GROUSE Algorithm 1,

$$\mathbb{P}\left(\zeta_K \geq \zeta^*\right) \geq 1 - 2\rho \; ,$$

where $\tau_0 = 1 + \frac{\log \frac{(1-\rho/2)}{C} + d\log(e/d)}{d\log n}$ *with* C *be a constant approximately equal to* 1.

The is still incomplete. We present our current strategy for proving it in Section 2.8.2 and discuss the missing steps there. We show that the iteration complexity can potentially be a combination of iterations required by two phases: $K_1 = \left(\frac{2d^2}{\rho} + 1\right)\tau_0 \log(n)$ is the number of iterations required by GROUSE to achieve $\zeta_t \geq 1/2$ from a random initialization U_0; and $K_2 = 2d\log\left(\frac{1}{2\rho(1-\zeta^*)}\right)$ is the number of additional iterations required by GROUSE to converge to the given accuracy ζ^* from $\zeta_{K_1} = 1/2$.

We want to comment that conjecture 2.4.1 requires fully observed noiseless data, which is not very practical in many cases. However, it can potentially be the first convergence guarantee for the Grassmannian gradient descent based method for subspace estimation with streaming data. It is a very important initial step for further studies on more general cases, including undersampled data and noisy data with outliers. In the following section, we will analyze the convergence behavior of GROUSE for undersampled data. We leave the corrupted data case as future work.

2.5 Undersampled Data

In this section, we consider undersampled data where each vector v is subsampled by a sampling matrix $A \in \mathbb{R}^{m\times n}$ with the number of measurements being much smaller than the ambient dimension ($m \ll n$). We study two typical cases, the compressively sampled data where A are random Gaussian matrices, and the missing data where each row of A is uniformly sampled from the identity matrix, $\mathbb{I}_n \in \mathbb{R}^{n\times n}$.

We first outline several elementary facts that can help us understand how the GROUSE algorithm navigates on the Grassmannian with undersampled data. The proofs can be found in Section 2.8.3.

Suppose AU has full column rank, then the projection coefficients w are found by the squares solution of $w =: \arg\min_a \|AUa - x\|^2$, *i.e.,* $w = (U^T A^T AU)^{-1} U^T A^T x$. Note that $x = Av$, therefore we can further decompose the projection coefficients w as $w = w_\parallel + w_\perp$ where

$$w_\parallel = \left(U^T A^T AU\right)^{-1} U^T A^T A v_\parallel \,, \qquad w_\perp = \left(U^T A^T AU\right)^{-1} U^T A^T A v_\perp \,. \tag{2.12}$$

This decomposition explicitly shows the perturbation induced by the undersampling framework, *i.e.,* $Av_\perp$ is not perpendicular to AU in general, though $v_\perp$ is orthogonal to $R(U)$. Now we are going to use this perturbation to show how the approximated projection p and residual r deviate from the exact ones obtained by projecting the full data sample v onto the current estimate $R(U)$.

Lemma 2.5.1. *Given Eq equation 2.12, let* $p = p_\parallel + p_\perp$ *with* $p_\parallel = Uw_\parallel$ *and* $p_\perp = Uw_\perp$, *then*

$$p_\parallel = v_\parallel \quad \textit{and} \quad r = A^T A v_\perp - A^T \mathcal{P}_{AU}(Av_\perp) \,. \tag{2.13}$$

Proof. Let $a = U^T v_\parallel$, then a is the unique solution to $Uw = v_\parallel$ given that U has full column rank. Since AU also has full column rank, $b = \left(U^T A^T AU\right)^{-1} U^T A^T A v_\parallel$ is also the unique solution to $AUw = Av_\parallel$. It then follows that $AUa = Av_\parallel = AUb$. Therefore, $a = b$. As for the second statement, it simply follows due to the fact that $Av_\parallel = AUw_\parallel \in R(AU)$. Hence $\widetilde{r} = (\mathbb{I}_m - \mathcal{P}_{AU})\, Av = (\mathbb{I}_m - \mathcal{P}_{AU})\, Av_\perp$, recall that $\mathcal{P}_{AU}$ denotes the orthogonal projection operator onto the column space of AU. This together with $r = A^T \widetilde{r}$ completes the proof. □

Below we lower bound the improvement on ζ_t as a function of the key quantities $r, \widetilde{r}$ and p. Compared to Lemma 2.4.1, Lemma 2.5.1 and Lemma 2.5.2 highlight

the how the perturbations induced by the undersampling framework influence the improvement on ζ_t for each iteration. Being able to analyze and bound the quantities that include the perturbations is the key to establish the expected improvement on ζ_t for undersampled data.

Lemma 2.5.2. *Suppose AU has full column rank, then for each iteration of GROUSE we have*

$$\frac{\zeta_{t+1}}{\zeta_t} \geq 1 + \frac{2\left\|\widetilde{r}\right\|^2 - \|r\|^2}{\|p\|^2} + 2\frac{\Delta}{\|p\|^2} \tag{2.14}$$

where $\Delta = w_\perp^T \left(\bar{U}^T U\right)^{-1} \bar{U}^T r$ *with* $w_\perp = \left(U^T A^T A U\right)^{-1} U^T A^T A v_\perp$.

The above lemma highlights the main hurdle in establishing global convergence for undersampled data. As is indicated by (2.14), there is no guarantee on monotonicity of the improvement on ζ_t. Indeed the uncertainty and perturbations introduced by the undersampling framework can even prevent us from establishing monotonically expected improvement on ζ_t. However, we are still able to bound the key quantities in Lemma 2.5.2 and provide more insights on the convergence behaviors of GROUSE for both compressively sampled data and missing data.

2.5.1 Compressively Sampled Data

This section presents convergence results for compressively sampled data. We use an approach that merges linear algebra with random matrix theory to establish an expected rate of improvement on the determinant similarity ζ_t at each iteration. We show that, under mild conditions, the determinant similarity increases in expectation with a rate similar to that of the fully sampled case, roughly scaled by $\frac{m}{n}$. Detailed proofs for this section are provided in Section 2.8.3.

Theorem 2.5.1. *Suppose each sampling matrix A has i.i.d Gaussian entries distributed as* $\mathcal{N}(0, 1/n)$. *Let* ϕ_d *denote the largest principal angle between* $R(U)$ *and*

$R(\bar{U})$, then with probability exceeding $1 - \exp\left(-\frac{d\delta^2}{8}\right) - \exp\left(-\frac{m\delta^2}{32} + d\log\left(\frac{24}{\delta}\right)\right) - (4d+2)\exp\left(-\frac{m\delta^2}{8}\right)$ *we obtain*

$$\mathbb{E}_v\left[\zeta_{t+1}\middle|U\right] \geq \left(1 + \gamma_1\left(1 - \gamma_2\frac{d}{m}\right)\frac{m}{n}\frac{1-\zeta_t}{d}\right)\zeta_t\ ,$$

where $\gamma_1 = \frac{(1-\delta)\left(1-2\delta\sqrt{\frac{m}{n}}\right)}{\left(1+\sqrt{\frac{1+\delta}{1-\delta}\frac{d}{m}}\right)^2}$ *and* $\gamma_2 = \left(1 + \frac{2\tan(\phi_d)+\delta\frac{d}{\cos(\phi_d)}}{\left(1-2\delta\sqrt{\frac{m}{n}}\right)\sqrt{(1+\delta)d/m}}\right)\frac{1+\delta}{1-\delta}$. *Now let* $\beta = \frac{8(1+\delta)}{(1-\delta)^2(1-2\delta)^2}$, *further suppose*

$$m \geq d\cdot\max\left\{\frac{32}{\delta^2}\log\left(\frac{24n^{2/d}}{\delta}\right), \beta\left(\tan\phi_d + \delta\cos\phi_d d\right)\left(\tan\phi_d + \delta\cos\phi_d d + \frac{1}{2}\right)\right\}\ ,$$

then with probability at least $1 - 2/n^2 - \exp\left(-d\delta^2/8\right)$ *we have*

$$\mathbb{E}_v\left[\zeta_{t+1}\middle|U\right] \geq \left(1 + \frac{1}{2\gamma_1}\frac{m}{n}\frac{1-\zeta_t}{d}\right)\zeta_t\ .$$

This theorem implies that, for each iteration of GROUSE, expected improvement on ζ_t can be obtained with high probability as long as the number of samples is enough. As shown in Theorem 2.5.1, our theory for GROUSE requires more measurements when $R(U)$ is far away from $R(\bar{U})$, in which case $\cos\phi_d =: \varepsilon$ is very small. In the high dimensional setting where $m \ll n$, compared to the fully sampled data case, the expected improvement on ζ_t is approximately scaled down by $\frac{m}{n}$. As we will show, this scaling factor is mainly determined by the relative amount of effective information stored in the approximated projection residual. On the other hand, due to the perturbation and uncertainty induced by the compressed sampling framework, the improvement on the determinant similarity given by the lower bound in Lemma 2.5.2 is neither monotonic nor global. As mentioned before, this is the main hurdle to pass before we can provide a global convergence result for undersampled data. However, despite of these difficulties, we are still able to establish Theorem 2.5.1 which shows that, with reasonable number of measurements, the expected improvement on

the convergence metric is monotonic with high probability as long as our estimate $R(U)$ is not too far away from the true subspace $R(\bar{U})$.

To prove Theorem 2.5.1, we provide the following intermediate results to quantify the key quantities in Lemma 2.5.2 with high probability, where probability is taken with respect to the random Gaussian sampling matrix A.

Lemma 2.5.3. *Under the same conditions as Theorem 2.5.1, with probability at least* $1 - \exp\left(-\frac{m\delta_2^2}{2}\right) - \exp\left(-\frac{m\delta_1^2}{8}\right) - \exp\left(-\frac{d\delta_1^2}{8}\right)$ *we obtain*

$$\|\widetilde{r}\|_2^2 \geq (1-\delta_1)\left(1-\beta\frac{d}{m}\right)\frac{m}{n}\|v_\perp\|_2^2 \tag{2.15}$$

$$2\|\widetilde{r}\|_2^2 - \|r\|_2^2 \geq (1-\delta_1)\left(1-2\delta_2\sqrt{\frac{m}{n}}\right)\left(1-\beta\frac{d}{m}\right)\frac{m}{n}\|v_\perp\|_2^2 \tag{2.16}$$

where $\delta_1, \delta_2 \in (0,1)$, *and* $\beta = \frac{1+\delta_1}{1-\delta_1}$.

To interpret the above results, note that

$$\|\widetilde{r}\|_2^2 = \|(\mathbb{I}_m - \mathcal{P}_{AU})\,Av_\perp\|_2^2 = \|Av_\perp\|_2^2 - \|\mathcal{P}_{AU}(Av_\perp)\|_2^2\ . \tag{2.17}$$

where the first equality follows by the fact that $(\mathbb{I}_m - \mathcal{P}_{AU})\,Av_\| = 0$ as we argued before, and the second equality holds since $\mathcal{P}_{AU}$ is an orthogonal projection onto $R(AU)$. Then by leveraging the concentration property of random projection, we can prove that $\|\widetilde{r}\|_2^2$ concentrates around its expectation $\frac{m-d}{n}\|v_\perp\|_2^2$ with high probability. Also note that $\|r\|_2^2 \leq \|A\|_2^2\|\widetilde{r}\|_2^2$, hence the second statement equation 2.16 can be established by the concentration result of $\|\widetilde{r}\|_2^2$ and that of $\|A\|_2^2$ according to the random matrix theory.

Next we establish high probability bounds on $\|p\|_2^2$ and Δ. Then Theorem 2.5.1 follows naturally by first replacing the key quantities in Lemma 2.5.2 with their high probability bounds, and then taking the expectation over the uncertainty of the underlying full data v_t.

Lemma 2.5.4. *With the same conditions as Theorem 2.5.1, for any $\delta_1 \in (0,1)$, we have*

$$\|p\|^2 \leq \left(1 + \sqrt{\frac{1+\delta_1}{1-\delta_1}\frac{d}{m}}\right)^2 \|v\|^2$$

with probability at least $1 - \exp\left(-\frac{d\delta_1^2}{8}\right) - \exp\left(-\frac{m\delta_1^2}{32} + d\log\left(\frac{24}{\delta_1}\right)\right)$.

Lemma 2.5.5. *With the same conditions as Theorem 2.5.1, let $\delta_1, \delta_3 \in (0,1)$, then*

$$\Delta \leq \sqrt{\frac{1+\delta_1}{1-\delta_1}\frac{d}{m}}\left(\tan(\phi_d) + \delta_3\frac{d}{\cos(\phi_d)}\right)\frac{m}{n}\|v_\perp\|^2$$

holds with probability at least $1-\exp\left(-\frac{d\delta_1^2}{8}\right)-\exp\left(-\frac{m\delta_1^2}{32} + d\log\left(\frac{24}{\delta_1}\right)\right)-4d\exp\left(-\frac{m\delta_3^2}{8}\right)$.

Lemma 2.5.4 shows that $\|p\|_2^2$ doesn't diverge significantly from $\|v\|_2^2$ as long as $m \geq d$. This together with Lemma 2.5.2 and Lemma 2.5.3 imply that the required number of measurements in Theorem 2.5.1 is mainly determined by that required by Lemma 2.5.5 so as to prevent Δ diverging too far from $\frac{m}{n}\|v_\perp\|_2^2$. As a result, the improvement on the determinant similarity is still dominated by the magnitude of the projection residual over that of the projection, which is proportional to that of the full data case scaled by the sampling density. On the other hand, Lemma 2.5.5 implies that, in order to guarantee Δ to be much smaller than $\frac{m}{n}\|v_\perp\|_2^2$, the number of required measurements increases along with first principal angle between the estimated subspace $R(U)$ and the true subspace $R(\bar{U})$.

For the sake of completeness, we sketch the proof of Theorem 2.5.1 here, and the detailed proof is provided in Section 2.8.3.

Proof sketch of Theorem 2.5.1. Let $\eta_1 = \frac{1+\delta}{1-\delta}\frac{d}{m}$, $\eta_2 = (1-\delta)\left(1 - 2\delta\sqrt{\frac{m}{n}}\right)$ and $\eta_3 = \tan(\phi_d) + \delta\frac{d}{\cos(\phi_d)}$, then plugging in the results in Lemmas 2.5.3, 2.5.4 and 2.5.5 into

Lemma 2.5.2 with $\delta_1 = \delta_2 = \delta_3 = \delta$ yields,

$$\frac{\zeta_{t+1}}{\zeta_t} \geq 1 + \gamma_1 \left(1 - \gamma_2 \frac{d}{m}\right) \frac{m}{n} \frac{\|v_\perp\|^2}{\|v\|^2} \geq 1 + \gamma_1 \left(1 - \gamma_2 \frac{d}{m}\right) \frac{m}{n} \frac{1-\zeta_t}{d} \tag{2.18}$$

where $\gamma_1 = \frac{(1-\delta)\left(1-2\delta\sqrt{\frac{m}{n}}\right)}{\left(1+\sqrt{\frac{1+\delta}{1-\delta}\frac{d}{m}}\right)^2}$ and $\gamma_2 = \left(1 + 2\frac{\tan(\phi_d)+\delta_3\frac{d}{\cos(\phi_d)}}{\left(1-2\delta\sqrt{\frac{m}{n}}\right)\sqrt{(1-\delta^2)d/m}}\right)\frac{1+\delta}{1-\delta}$.

The first probability bound is obtained by taking the union bound of those quantities used to generate Lemma 2.5.3 to Lemma 2.5.5, which can be lower bounded by

$$1 - \exp\left(-\frac{d\delta^2}{8}\right) - \exp\left(-\frac{m\delta^2}{32} + d\log\left(\frac{24}{\delta}\right)\right) - (4d+2)\exp\left(-\frac{m\delta^2}{8}\right) \tag{2.19}$$

Next we establish the complexity bound on m. As we will prove in Section 2.8.3, $\gamma_2 \frac{d}{m} < \frac{1}{2}$ is equivalent to the following,

$$m \geq \frac{8(1+\delta)}{(1-\delta)^2(1-2\delta)^2}\left(\varepsilon + \delta\sqrt{1+\varepsilon^2}d\right)\left(\varepsilon + \delta\sqrt{1+\varepsilon^2}d + \frac{1}{2}\right)d \tag{2.20}$$

To establish another bound on m, $m \geq \frac{32}{\delta^2}\log\left(\frac{24n^{2/d}}{\delta}\right)d$ implies the following,

$$\exp\left(-\frac{m\delta^2}{32} + d\log\left(\frac{24}{\delta}\right)\right) \leq \exp(-\log n^2) = \frac{1}{n^2} \tag{2.21}$$

$$(4d+2)\exp\left(-\frac{m\delta^2}{8}\right) \leq \frac{(4d+2)}{n^8}\left(\frac{\delta}{24}\right)^{4d} \ll \frac{1}{n^2} \tag{2.22}$$

equation 2.21 and equation 2.22 complete the proof for the bound on m and justify the simplification of the probability bound in equation 2.19. □

2.5.2 Missing Data

In this section, we study the convergence of GROUSE for the missing data case. We show that within the local region of the true subspace, we obtain an expected monotonic improvement on our defined convergence metric with high probability. We use Ω to denote the indices of observed entries for each data vector, and we assume Ω is uniformly sampled over $\{1, 2, \ldots, n\}$ with replacement. In other words, we assume

each row of the sampling matrices A is uniformly sampled from the rows of identity matrix $\mathbb{I}_n$ with replacement. We use the notation $Av =: v_\Omega, AU =: U_\Omega$. Again our results are with high probability with respect to A, in this case with respect to the random draw of rows of $\mathbb{I}_n$, and in expectation with respect to the random data v. Please refer to Section 2.8.3 for the proofs of this section.

Before we present our main results, we first call out the typical incoherence assumption on the underlying data.

Definition 2. *A subspace $R(U)$ is incoherent with parameter μ if*

$$\max_{i\in\{1,\dots,n\}} \|\mathcal{P}_U e_i\|_2^2 \leq \frac{\mu d}{n}$$

where e_i is the i^{th} canonical basis vector and $\mathcal{P}_U$ is the projection operator onto the column space of U.

Note that $1 \leq \mu \leq \frac{n}{d}$. According to the above definition, the incoherence parameter of a vector $z \in \mathbb{R}^n$ is defined as:

$$\mu(z) = \frac{n\|z\|_\infty^2}{\|z\|_2^2} \tag{2.23}$$

In this section, we assume the true subspace $R(\bar{U})$ is incoherent with parameter μ_0, and use $\mu(U)$, $\mu(v_\perp)$ to denote the incoherence parameter of $R(U)$ and $v_\perp$ respectively. We now show the expected improvement of ζ_t in a local region of the true subspace.

Theorem 2.5.2. *Suppose $\sum_{k=1}^{d} \sin^2\phi_k \leq \frac{d\mu_0}{16n}$ and $|\Omega| = m$. If*

$$m > \max\left\{\frac{128d\mu_0}{3}\log\left(\sqrt{2}dn\right), 64\mu(v_\perp)^2\log(n), 52\left(1+2\sqrt{\mu(v_\perp)\log(n)}\right)^2 d\mu_0\right\}$$

then with probability at least $1-\frac{3}{n^2}$ we have

$$\mathbb{E}_v\left[\zeta_{t+1}\middle|U\right] \geq 1 + \frac{1}{4}\frac{m}{n}\frac{1-\zeta_t}{d}\,.$$

This theorem shows that, within the local region of the true subspace, expected improvement on ζ_t can be obtained with high probability. As is implied by the theorem, this local region gets enlarged if the true subspace is more coherent, which may seem at first counterintuitive. However, the required number of measurements also increases as we increase μ_0. In the extreme case, when m increases to n, the local convergence results can be extended to a global result, as we proved for the full data case in Section 2.4. On the other hand, compared to Theorem 2.5.1, the convergence result for the missing data case holds within a more conservative local region of the true subspace. This gap is induced by the challenge of maintaining the incoherence property of our estimates $R(U)$, for which we had to consider the worst case. We leave the extension of the local convergence results to global results as future work.

In order to compare our result to the local convergence result in [Corollary 2.15, [17]], consider the following corollary.

Corollary 2.5.1. *Define the determinant discrepancy as $\kappa_t = 1 - \zeta_t$, then under the same conditions as Theorem 2.5.2, we have*

$$\mathbb{E}_v \left[\kappa_{t+1} \middle| \kappa_t \right] \leq \left(1 - \frac{1}{4} \left(1 - \frac{d\mu_0}{16n} \right) \frac{m}{nd} \right) \kappa_t$$

with probability exceeding $1 - 3/n^2$.

Recall that $1 \leq \mu_0 \leq \frac{n}{d}$, therefore the expected linear decay rate of κ_t is at least $1 - \frac{9}{16} \frac{m}{nd}$. In [17] (Corollary 2.15), a similar linear convergence result is established in terms of the Frobenius norm discrepancy between $R(\bar{U})$ and $R(U)$, denoted as $\epsilon_t = \sum_{i=1}^{d} \sin^2 \phi_d$. However, their result only holds when $\epsilon_t \leq (8 \times 10^{-6}) \frac{m}{n^3 d^2}$ which is more conservative than our assumption in Theorem 2.5.2. Moreover, as we mentioned previously, empirical evidence shows the lower bound in Theorem 2.5.2 holds for every iteration from any random initialization. In contrast, in [17], even for numerical

results expected linear improvements only hold within the local region of the true subspace.

Now we present the following intermediate results for the proof of Theorem 2.5.2. Note that in this missing data case, the projection residual r_Ω of v_Ω onto U_Ω is mapped back to $\mathbb{R}^n$ by zero padding the entries at the indices that are not in Ω. Therefore, unlike Lemma 2.5.5 of the compressively sampled data case, here $\|\widetilde{r}\| = \|r\| = \|r_\Omega\|$. Therefore, equation 2.14 becomes

$$\frac{\zeta_{t+1}}{\zeta_t} \geq 1 + \frac{\|r_\Omega\|^2}{\|p\|^2} + 2\frac{\Delta}{\|p\|^2} \,. \tag{2.24}$$

Now similarly to the compressively sampled data case, we proceed by establishing concentration results for the key quantities $\|r\|_2^2$, $\|p\|_2^2$ and Δ respectively.

Lemma 2.5.6 ([16], Theorem 1)**.** *Let $\delta > 0$, and suppose $m \geq \frac{8}{3} d\mu(U) \log(2d/\delta)$. Then, with probability exceeding $1 - 3\delta$,*

$$\|r_\Omega\|^2 \geq (1 - \alpha_0)\frac{m}{n}\|v_\perp\|^2$$

where $\alpha_0 = \sqrt{\frac{2\mu(v_\perp)^2}{m}\log\left(\frac{1}{\delta}\right)} + \frac{(\beta_1+1)^2}{1-\gamma_1}\frac{d\mu(U)}{m}$, $\beta_1 = \sqrt{2\mu(v_\perp)\log\left(\frac{1}{\delta}\right)}$, and $\gamma_1 = \sqrt{\frac{8d\mu(U)}{3m}\log(2d/\delta)}$.

Lemma 2.5.7. *Let $\delta > 0$. Under the same condition on m as Lemma 2.5.6, with probability at least $1 - 2\delta$ we have*

$$\|p\|^2 \leq \left(1 + \frac{\beta_1 + 1}{1 - \gamma_1}\sqrt{\frac{d\mu(U)}{m}}\right)^2 \|v\|^2$$

where β_1 and γ_1 equal to those defined in Lemma 2.5.6.

Lemma 2.5.8. *Let $\delta > 0$. Under the same condition on m as Lemma 2.5.6, with probability at least $1 - 3\delta$ we have*

$$|\Delta| \leq \frac{\eta_3}{\cos\phi_d}\sqrt{\sin^2\phi_d + \frac{d\mu_0}{m}}\sqrt{\frac{d\mu(U)}{m}}\frac{m}{n}\|v_\perp\|^2$$

where $\eta_3 = \frac{(1+\beta_1)(1+\beta_2)}{1-\gamma_1}$, $\beta_2 = \sqrt{2\mu(v_\perp)\log\left(\frac{1}{\delta}\right)\frac{d\mu_0}{d\mu_0+m\sin^2\phi_d}}$, *and* β_1 *and* γ_1 *equal to those defined in Lemma 2.5.6.*

Lemma 2.5.6 shows that the concentration of $\|r\|_2^2 = \|r_\Omega\|_2^2$ does not only depend on the sampling framework, but also on the incoherence property of the current estimate and the true projection residual, *i.e.,* $\mu(U)$ and $\mu(v_\perp)$. To see this clearly, recall that $\|r_\Omega\|_2^2 = \|v_{\perp,\Omega}\|_2^2 - \|\mathcal{P}_{U_\Omega}(v_{\perp,\Omega})\|_2^2$, hence the incoherence property of $v_\perp$ and $R(U)$ directly influences the concentration of $\|r_\Omega\|_2^2$. On the other hand, for compressive data, the Gaussian distributed sampling matrices yield tight concentration results for $\|p\|_2^2$, $\|r_\Omega\|_2^2$ and Δ. Therefore, the upper bounds of the key quantities established in Lemmas 2.5.6, 2.5.7 and 2.5.8 are not as tight as those for the compressive data except the extreme case where $\mu(U) = \mu(v_\perp) = 1$, *i.e.,* both $R(U)$ and $v_\perp$ are incoherent.

As shown in the above lemmas, in order to establish concentration of the key quantities in (2.24), it is essential for the subspaces generated by GROUSE to be incoherent over iterates. It has been proven in [17] that within the local region of $R(\bar{U})$, the incoherence of $R(U)$ can be bounded by that of $R(\bar{U})$.

Lemma 2.5.9 ([17], Lemma 2.5)**.** *Suppose* $\sum_{k=1}^{d}\sin^2\phi_k \le \frac{d}{16n}\mu_0$, *then* $\mu(U) \le 2\mu_0$.

Now we are ready to prove Theorem 2.5.2. We sketch the proof here, and a detailed proof is provided in Section 2.8.3.

Proof sketch of Theorem 2.5.2. Given the condition required by Theorem 2.5.2, we have $\sin\phi_d \le \sqrt{d\mu_0/16n}$ and $\cos\phi_d \ge \sqrt{1-d\mu_0/16n}$. This together with Lemma 2.5.9 and Lemma 2.5.8 yield $|\Delta| \le \frac{11}{5}\eta_3\frac{d\mu_0}{n}\|v_\perp\|^2$. Also for β_2 in Lemma 2.5.8, $\beta_2 \le \sqrt{2\mu(v_\perp)\log(1/\delta)} = \beta_1$. Hence,

$$|\Delta| \le \frac{11}{5}\frac{(1+\beta_1)^2}{1-\gamma_1}\frac{d\mu_0}{n}\|v_\perp\|^2 . \tag{2.25}$$

Letting $\eta_2 = \frac{(1+\beta_1)^2}{1-\gamma_1}\frac{d\mu_0}{m}$ and $\alpha_1 = \sqrt{\frac{2\mu(v_\perp)^2}{m}\log\left(\frac{1}{\delta}\right)}$, then applying this definition together with Lemma 2.5.9 to Lemma 2.5.7 and Lemma 2.5.6 yields

$$\|p\|^2 \leq \left(1+\sqrt{\frac{2\eta_2}{1-\gamma_1}}\right)^2 \|v\|^2 \tag{2.26}$$

$$\|r_\Omega\|^2 \geq (1-\alpha_1-2\eta_2)\frac{m}{n}\|v_\perp\|^2 \tag{2.27}$$

Now applying equation 2.25, equation 2.26 and equation 2.27 to equation 2.24 we have

$$\frac{\zeta_{t+1}}{\zeta_t} \geq 1 + \frac{(1-\alpha_1-\frac{32}{5}\eta_2)}{(1+\sqrt{2\eta_2/(1-\gamma_1)})^2}\frac{m}{n}\frac{\|v_\perp\|^2}{\|v\|^2} \tag{2.28}$$

with probability at least $1-3\delta$. The probability bound is obtained by taking the union bound of those generating Lemmas 2.5.6, 2.5.7 and 2.5.8, as we can see in the proofs in Section 2.8.3 this union bound is at least $1-3\delta$.

Letting $\eta_1 = \frac{(1-\alpha_1-\frac{32}{5}\eta_2)}{(1+\sqrt{2\eta_2/(1-\gamma_1)})^2}$, then $\eta_1 > 0$ is equivalent to $1-\alpha_1-\frac{32}{5}\eta_2 > 0$. This further gives that if m satisfies the condition in Theorem 2.5.2, then $\eta_1 > \frac{1}{4}$. Now taking expectation with respect to v yields,

$$\mathbb{E}_v\left[\zeta_{t+1}\middle|U\right] \geq \left(1+\frac{1}{4}\frac{m}{n}\mathbb{E}\left[\frac{\|v_\perp\|^2}{\|v\|^2}\middle|U\right]\right)\zeta_t \geq \left(1+\frac{1}{4}\frac{m}{n}\frac{1-\zeta_t}{d}\right)\zeta_t \tag{2.29}$$

where the last inequality follows from Lemma 2.3.1. Finally choosing δ to be $1/n^2$completes the proof. □

2.6 Numerical Results

In this section, we demonstrate that our theoretical results match the empirical convergence behavior of GROUSE. We generate the underlying data matrix $M = \begin{bmatrix} v_1 & v_2 & \dots & v_T \end{bmatrix}$ as $M = \bar{U}W$. For both the fully sampled data case and compressively sampled data case, the underlying signals are generated from a sparse subspace, demonstrating that incoherence assumptions are not required by our results

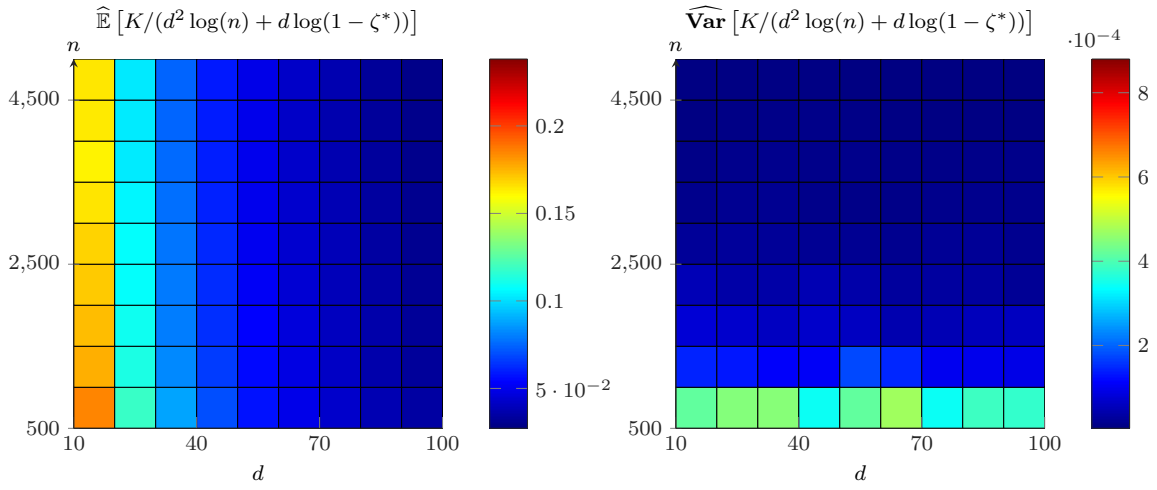


Figure 2.1: Illustration of the bounds on K in Conjecture 2.4.1 compared to their values in practice, averaged over 50 trials with different n and d. We show the ratio of K to the bound $d^2\log(n)+d\log(1-\zeta^*)$.

for these two cases. Specifically, the underlying subspace of each trial is set to be a sparse subspace, as the range of an $n\times d$ matrix $\bar{U}$ with sparsity on the order of $\frac{\log(n)}{n}$. For the missing data case, we generate the underlying subspace as the range of an $n\times d$ matrix with i.i.d standard normal distribution. The entries of the coefficient matrix W for all three cases are generated as i.i.d $\mathcal{N}(0,1)$ satisfying Assumption 1. We also want to mention that we run GROUSE with random initialization for all of the plots in this section.

We first examine our global convergence result, *i.e.,* Conjecture 2.4.1, for the fully sampled data in Figure 2.1. We run GROUSE to convergence for a required accuracy $\zeta^*=1-1e\text{-}4$ and show the ratio of K to the simplified bound of Conjecture 2.4.1, $d^2\log(n)+d\log\frac{1}{1-\zeta^*}$. We run GROUSE over 50 trials and show the mean and variance. We can see that, for fixed n, our conjecture becomes more and more loose as we increase the dimension of the underlying subspace. However, compared to the empirical mean, the empirical variance is very small. This indicates that the relationship between our conjectured upper bounds and the actual iterations required

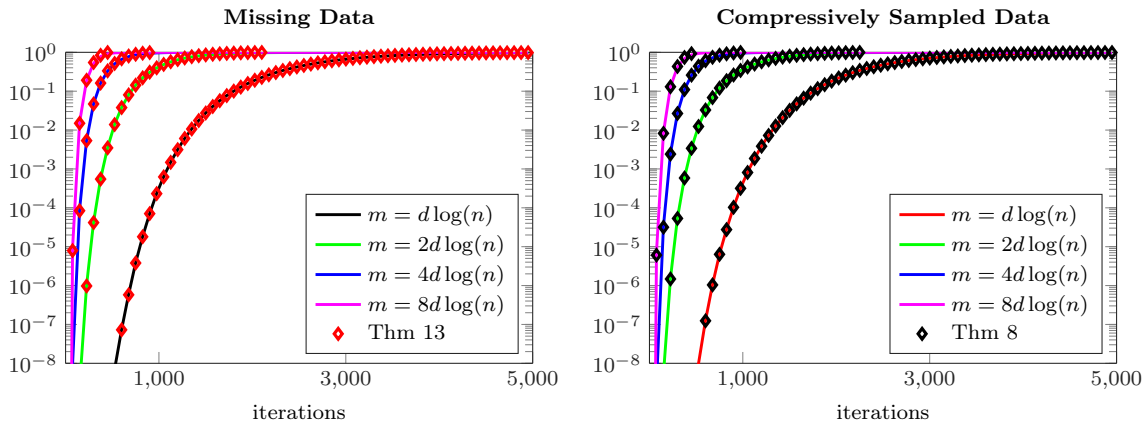


Figure 2.2: Illustration of expected improvement on ζ given by Theorem 2.5.1 (left) and Theorem 2.5.2 (right) over 50 trials. We set $n = 5000$, $d = 10$. The diamonds denote the lower bound on expected convergence rates described in Theorem 2.5.1 and Theorem 2.5.2.

by GROUSE is stable.

Next we examine our theoretical results (Theorem 2.5.1 and Theorem 2.5.2) for the expected improvement on ζ_t for the undersampled case in Figure 2.2. We set $n = 5000$ and $d = 10$. We run GROUSE over different sampling numbers m. The plots are obtained by averaging over 50 trials. We can see that our theoretical bounds on the expected improvement on ζ_t for both missing data and compressively sampled data are tight from any random initialization, although we have only established local convergence results for both cases. Also note that Theorem 2.5.1 and Theorem 2.5.2 indicate that the expected improvement on the determinant similarity has a similar form to that of the fully sampled case roughly scaled by the sampling density (m/n). These together motivate us to approximate the required iterations to achieve a given accuracy as that required by the fully sampled case times the reciprocal of sampling density, n/m:

$$(n/m) \cdot \left(d^2 \log(n) + d\log(1 - \zeta^*)\right) \ .$$

As we see in Figure 2.3, when m is slightly larger than d, the empirical mean of the ratio of the actual iterations required by GROUSE to our heuristic bound is similar

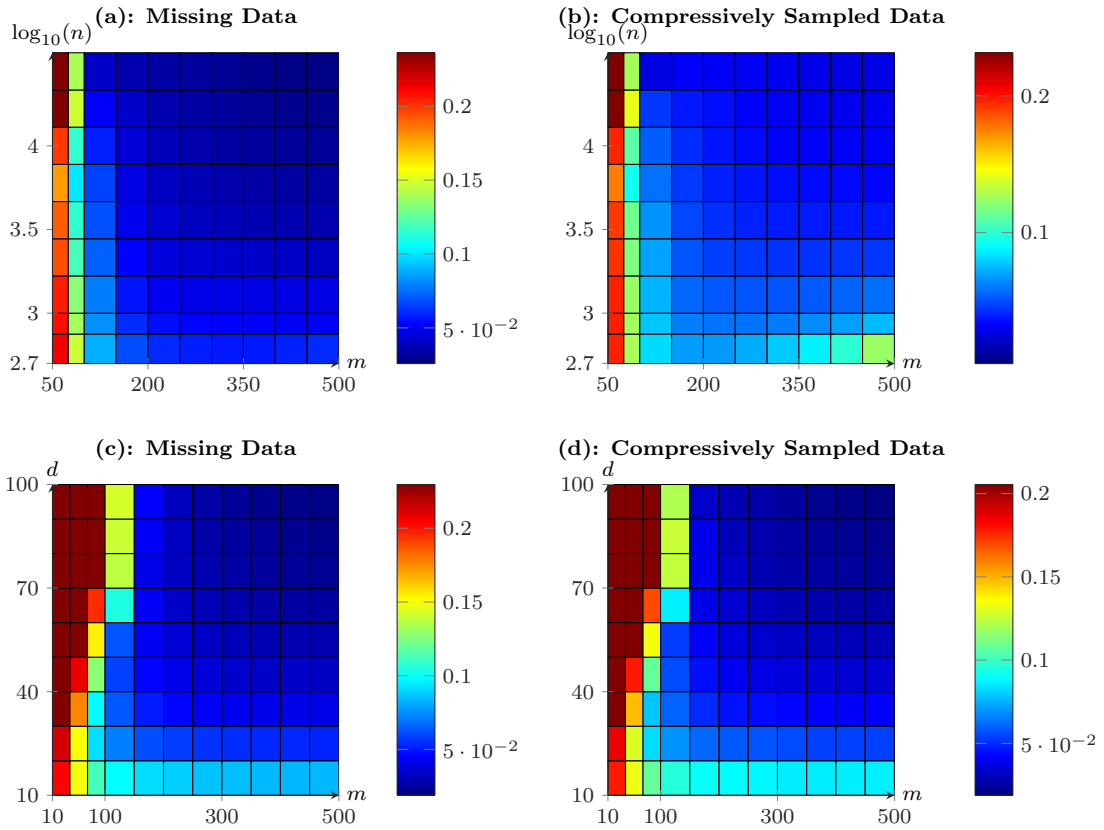


Figure 2.3: Illustration of our heuristic bounds on K (the actual iterations required by GROUSE to converge to the given accuracy) over different d, m and n, averaged over 20 trials. In this simulation, we run GROUSE from a random initialization to convergence for a required accuracy $\zeta^* = 1 - 1e\text{-}3$. We show the ratio of K to the heuristic bound $\frac{n}{m}\left(d^2 \log(n) + d \log(1-\zeta^*)\right)$. In (a) and (b), we set $d = 50$ and examine K over m and n for both missing data (a) and compressively sampled data (b). In (c) and (d), we set $n = 10000$ and examine K over m and d for both missing data (c) and compressively sampled data (d). In these plots, we use the dark red to indicate the failure of convergence.

to that of the full data case. We leave the rigorous proof of this heuristic as future work.

2.7 Conclusion

In this chapter, we analyze a manifold incremental gradient descent algorithm applied to a particular non-convex optimization formulation for recovering a low-dimensional subspace from streaming data sampled from that subspace. We provide a

simplified analysis as compared to [143], showing global convergence of the algorithm to the global minimizer for fully sampled data. However, the current convergence rate is loose compared to what we observed in practice. A future direction is to narrow the gap between our theory and the actual performance of GROUSE, for which Conjecture 2.4.1 shows great promise. We will complete the missing step in our current proof of Conjecture 2.4.1 and validate it in the near future.

With undersampled data, we show that expected improvement on our convergence metric ζ_t can be obtained with high probability for each iteration t. We prove that, comparing with fully sampled data, the expected improvement on determinant similarity is roughly proportional to the sampling density. With compressively sampled data this expected improvement holds from any random initialization, while it only holds within the local region of the true subspace for the missing data case.

2.8 Supplementary material

2.8.1 Preliminaries

We start by providing the following lemma that we will use regularly in the manipulation of the matrix $\bar{U}^T U$. It also provides us with more insight into our metric of determinant similarity between the subspaces. The proof can be found in [120].

Lemma 2.8.1 ([120], Theorem 5.2)**.** *Let $U, \bar{U} \in \mathbb{R}^{n\times d}$ with orthonormal columns, then there are unitary matrices Q, $\bar{Y}$, and Y such that*

$$Q\bar{U}\bar{Y} := \begin{matrix} \\ d \\ d \\ n-2d \end{matrix}\begin{matrix} d \\ \begin{pmatrix} I \\ 0 \\ 0 \end{pmatrix} \end{matrix} \quad \textit{and} \quad QUY := \begin{matrix} \\ d \\ d \\ n-2d \end{matrix}\begin{matrix} d \\ \begin{pmatrix} \Gamma \\ \Sigma \\ 0 \end{pmatrix} \end{matrix}$$

where $\Gamma = \operatorname{diag}(\cos\phi_1, \ldots, \cos\phi_d), \Sigma = \operatorname{diag}(\sin\phi_1, \ldots, \sin\phi_d)$ *with* ϕ_i *being the* i^{th} *principal angle between* $R(U)$ *and* $R(\bar{U})$ *defined in Definition 1.*

Now we are going to prove Lemma 2.3.1, which is essential for us to establish expected improvement on the determinant similarity for each iteration in the various sampling cases we consider. Before that, we present the following lemmas that are requried for the proof.

Lemma 2.8.2. *Given any matrix* $Q \in \mathbb{R}^{d\times d}$*, suppose that* $w \in \mathbb{R}^d$ *is a random vector whose components* w_i*,* $i = 1, \ldots, d$ *are zero-mean, independent, and identically distributed symmetrically about zero (*i.e., *the distribution of* w_i *is an even function). Then*

$$E\left[\frac{w^TQw}{w^Tw}\right] = \frac{1}{d}\,\boldsymbol{tr}(Q)\ .$$

Proof of Lemma 2.8.2.

$$E\left[\frac{w^TQw}{w^Tw}\right] = \sum_{i\neq j} E\left[\frac{w_iw_jQ_{ij}}{w^Tw}\right] + \sum_{i=1}^{d} E\left[\frac{w_i^2Q_{ii}}{w^Tw}\right]$$

$$= \sum_{i=1}^{d} Q_{ii}E\left[\frac{w_i^2}{w^Tw}\right] \tag{2.30}$$

$$= \frac{1}{d}\,\mathbf{tr}\,Q\ , \tag{2.31}$$

where Eqs (2.30) and (2.31) hold by the following two arguments. For Eq (2.30), let $f(w_1, \ldots, w_d)$ be the joint distribution among the coordinates, and without loss of

generality let $i = 1$ and $j \neq 1$, then

$$
\begin{aligned}
& E\left[\frac{w_1 w_j Q_{1j}}{w^T w}\right] \\
&= \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} \frac{w_1 w_j Q_{ij}}{w^T w} f(w_1, \ldots, w_d) dw_1 dw_2 \cdots dw_d \\
&= \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} \frac{w_1 w_j Q_{1j}}{w_1^2 + \sum_{k \neq i} w_k^2} f(w_1) f(w_2) \cdots f(w_d) dw_1 dw_2 \cdots dw_d \\
&= \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} \left(\int_{-\infty}^{\infty} \frac{w_1}{w_1^2 + \sum_{k \neq i} w_k^2} f(w_1) dw_1 \right) w_j Q_{1j} f(w_2) \cdots f(w_d) dw_2 \cdots dw_d \\
&= 0
\end{aligned}
$$

where the last inequality holds since $\frac{w_1}{w_1^2 + \sum_{k \neq i} w_k^2}$ is an odd function of w_1 and $f(w_1)$ is an even function of w_1, thereby the term in parentheses will integrate to zero. We note that if w_i is a discrete random variable, the argument would be similar.

To get Eq (2.31) we note that

$$
1 = E\left[\frac{\sum_i w_i^2}{\sum_j w_j^2}\right] = \sum_i E\left[\frac{w_i^2}{w^T w}\right] = dE\left[\frac{w_i^2}{w^T w}\right] \ , i = 1, \ldots, d \ ,
$$

where the last step holds because each w_i is identically distributed. □

Lemma 2.8.3 ([46], Lemma 16)**.** *Let $X = [X_1, \cdots, X_d]$ with $X_i \in [0, 1], i = 1, \ldots, d$, then*

$$
d - \sum_{i=1}^{d} X_i \geq 1 - \Pi_{i=1}^{d} X_i
$$

Proof of Lemma 2.3.1. According to Lemma 2.8.2 and Lemma 2.8.3 we have the following

$$
\begin{aligned}
\mathbb{E}\left[\frac{\|v_\perp\|^2}{\|v\|^2}\middle| U\right] = \mathbb{E}\left[\frac{\|\bar{U}s\|^2 - \|UU^T\bar{U}s\|^2}{\|\bar{U}s\|^2}\middle| U\right] &\overset{\vartheta_1}{=} \mathbb{E}\left[\frac{s^T\bar{Y}(I - \Gamma^2)\bar{Y}^T s}{s^T s}\middle| U\right] \\
&\overset{\vartheta_2}{=} \frac{1}{d}\,\mathbf{tr}\left(I - \Gamma^2\right) \overset{\vartheta_3}{\geq} \frac{1 - \zeta_t}{d}
\end{aligned} \tag{2.32}
$$

where ϑ_1 follows by Lemma 2.8.1 and $\|\bar{U}s\|^2 = \|s\|^2$, ϑ_2 from Lemma 2.8.2, and ϑ_3 from Lemma 2.8.3 with $X_i = \cos^2\phi_i$. □

2.8.2 Proof of Fully Sampled Data

In this section we prove the results of Section 2.4. We start by proving Eq 2.10, the deterministic expression for the change in determinant similarity from one step of the GROUSE algorithm to the next. Using this expression, we prove the GROUSE monotonic improvement of Lemma 2.4.1, expected improvement of Lemma 2.4.2, and finally the global convergence conjecture 2.4.1.

Recall that $\frac{y}{\|y\|} = \cos(\theta)\frac{v_\|}{\|v_\|\|} + \sin(\theta)\frac{v_\perp}{\|v_\perp\|}$ in Algorithm 1. Then according to the GROUSE update in 2.4 we have

$$\begin{aligned}
\det\left(\bar{U}^T U_{t+1}\right) &= \det\left(\bar{U}^T U + \left(\frac{\bar{U}^T y}{\|y\|} - \frac{\bar{U}^T v_\|}{\|v_\|\|}\right)\frac{w^T}{\|w\|}\right) \\
&\overset{\vartheta_1}{=} \det\left(\bar{U}^T U\right)\left(1 + \frac{w^T(\bar{U}^T U)^{-1}}{\|w\|}\left(\frac{\bar{U}^T y}{\|y\|} - \frac{\bar{U}^T v_\|}{\|v_\|\|}\right)\right) \\
&\overset{\vartheta_2}{=} \det\left(\bar{U}^T U\right)\frac{w^T(\bar{U}^T U)^{-1}\bar{U}^T y}{\|y\|\|w\|} \\
&\overset{\vartheta_3}{=} \det\left(\bar{U}^T U\right)\left(\cos\theta + \frac{\|v_\perp\|}{\|v_\|\|}\sin\theta\right) \qquad (2.33)
\end{aligned}$$

where ϑ_1 follows from the Schur complement, *i.e.,* that for any invertible matrix M we have $\det\left(M + ab^T\right) = \det(M)\left(1 + b^T M^{-1} a\right)$; ϑ_2 and ϑ_3 hold since $\|v_\|\|^2 = \|Uw\|^2 = \|w\|^2$ and the following

$$w^T(\bar{U}^T U)^{-1}\bar{U}^T v_\| \overset{w=U^T\bar{U}s}{=} v^T v_\| = \|v_\|\|^2 \qquad (2.34a)$$

$$w^T(\bar{U}^T U)^{-1}\bar{U}^T v_\perp \overset{w=U^T\bar{U}s}{=} v^T v_\perp = \|v_\perp\|^2. \qquad (2.34b)$$

Given this, the proof of Lemma 2.4.1 follows directly from the above proof and the greedy step size derived in Eq. 2.11.

Proof of Lemma 2.4.1. By using $\theta = \arctan\left(\frac{\|v_\perp\|}{\|v_\|\|}\right)$, we have $\cos\theta = \frac{\|v_\|\|}{\|v\|}$ and $\sin\theta = \frac{\|v_\perp\|}{\|v\|}$. This together with 2.33 gives $\det\left(\bar{U}^T U_{t+1}\right) = \det\left(\bar{U}^T U\right)\frac{\|v\|}{\|v_\|\|}$. Therefore, $\frac{\zeta_{t+1}}{\zeta_t} = \frac{\det\left(\bar{U}^T U_{t+1}\right)^2}{\det\left(\bar{U}^T U\right)^2} = \frac{\|v\|^2}{\|v_\|\|^2} = 1 + \frac{\|v_\perp\|^2}{\|v_\|\|^2}$. □

Proof of Lemma 2.4.2. Lemma 2.4.2 follows directly from 2.3.1 and 2.4.1, *i.e.,*

$$\begin{aligned}\mathbb{E}\left[\frac{\zeta_{t+1}}{\zeta_t}\middle| U\right] = 1 + \mathbb{E}\left[\frac{\|v_\perp\|^2}{\|v_\|\|^2}\middle| U\right] &\geq 1 + \mathbb{E}\left[\frac{\|v_\perp\|^2}{\|v\|^2}\middle| U\right] \\ &\geq 1 + \frac{1-\zeta_t}{d} \qquad (2.35)\end{aligned}$$

Note that, given U, ζ_t is a constant, hence completes the proof. □

With the above results, we are ready to prove Theorem 2.4.1.

Proof of Theorem 2.4.1. Let $\kappa_t = 1 - \zeta_t$ denote the determinant *discrepancy* between $R(\bar{U})$ and $R(U)$. According to Lemma 2.4.2 we have the following:

$$\mathbb{E}\left[\frac{\kappa_{t+1}}{\kappa_t}\middle| U\right] \leq 1 - \frac{1-\kappa_t}{d} \qquad (2.36)$$

Now according to Lemma 2.4.1, $\kappa_t \leq 1 - \zeta_0$ for all $t \geq 0$. So using Eq (2.36) we have the following:

$$\mathbb{E}\left[\kappa_{t+1}\middle| U\right] \leq \left(1 - \frac{1-\kappa_t}{d}\right)\kappa_t \leq \left(1 - \frac{\zeta_0}{d}\right)\kappa_t \ .$$

Taking expectation of both sides, we have

$$\mathbb{E}\left[\kappa_{t+1}\right] \leq \left(1 - \frac{\zeta_0}{d}\right)\mathbb{E}\left[\kappa_t\right] \ .$$

After $K \geq \frac{d}{\zeta_0}\log\frac{1}{\rho(1-\zeta^*)} \geq \frac{d}{\zeta_0}\log\frac{\mathbb{E}[\eta_{K_1}]}{\rho(1-\zeta^*)}$ iterations of GROUSE we obtain

$$\mathbb{E}\left[\kappa_{t+K_1}\right] \leq \left(1 - \frac{\zeta_0}{d}\right)^K \mathbb{E}[\kappa_0] \leq \left(1 - \frac{\zeta_0}{d}\right)^{\frac{d}{\zeta_0}\log\frac{\mathbb{E}[\kappa_0]}{\rho(1-\zeta^*)}} \mathbb{E}[\kappa_0] \leq \rho(1-\zeta^*) \ .$$

Therefore

$$\mathbb{P}\left(\zeta_K \geq \zeta^*\right) = 1 - \mathbb{P}\left(\kappa_K \geq 1 - \zeta^*\right) \geq 1 - \frac{\mathbb{E}\left[\kappa_K\right]}{1-\zeta^*} \geq 1 - \rho \ . \qquad (2.37)$$

□

To prove Conjecture 2.4.1, we need the following lemma.

Lemma 2.8.4. *[103] Initialize the starting point U_0 of GROUSE as the orthonormalization of an $n \times d$ matrix with entries being standard normal random variables. Then*

$$\mathbb{E}[\zeta_0] = \mathbb{E}\left[\det(U_0^T \bar{U}\bar{U}^T U_0)\right] = C\left(\frac{d}{ne}\right)^d$$

where $C > 0$ is a constant.

Now we are ready to show our initial proof of Conjecture 2.4.1.

Proof of Conjecture 2.4.1. Let $\kappa_t = 1 - \zeta_t$ denote the determinant *discrepancy* between $R(\bar{U})$ and $R(U)$. According to Lemma 2.4.2 we have the following:

$$\mathbb{E}\left[\frac{\zeta_{t+1}}{\zeta_t}\middle| U\right] \geq 1 + \frac{1-\zeta_t}{d} \tag{2.38a}$$

$$\mathbb{E}\left[\frac{\kappa_{t+1}}{\kappa_t}\middle| U\right] \leq 1 - \frac{1-\kappa_t}{d} \tag{2.38b}$$

Therefore, the expected convergence rate of ζ_t is faster when $R(U)$ is far away from $R(\bar{U})$, while that of κ_t is faster when $R(U)$ is close to $R(\bar{U})$. This motivates us to split the analysis into two phases, bounding the number of iterations in each phase. We first use Eq (2.38a) to get the necessary K_1 iterations for GROUSE to converge to a local region of global optimal point from a random initialization. From there, we obtain the necessary K_2 iterations for GROUSE to converge to the required accuracy by leveraging Eq (2.38b).

Let ρ be any number within the range $(0, 1]$. Let $\bar{\zeta}_t$ be a non-decreasing sequence with $\mathbb{E}[\bar{\zeta}_0] = \mathbb{E}[\zeta_0]$ and the expected increase rate being lower bounded as

$$\mathbb{E}\left[\bar{\zeta}_{t+1}\middle| U\right] \geq \left(1 + \frac{\rho}{2d}\right)\bar{\zeta}_t \,.$$

Taking expectation of both sides, we obtain the following:

$$\mathbb{E}\left[\bar{\zeta}_{t+1}\right] \geq \left(1 + \frac{\rho}{2d}\right)\mathbb{E}[\bar{\zeta}_t]$$

Therefore after $K_1 \geq (2d/\rho + 1) \log \frac{1-\frac{\rho}{2}}{\mathbb{E}[\zeta_0]}$ steps we have

$$\begin{aligned} \mathbb{E}\left[\bar{\zeta}_{K_1}\right] &\geq \left(1 + \frac{\rho}{2d}\right)^{K_1} \mathbb{E}[\zeta_0] \geq \left(\left(1 + \frac{\rho}{2d}\right)^{\frac{2d}{\rho}+1}\right)^{\log \frac{1-\frac{\rho}{2}}{\mathbb{E}[\zeta_0]}} \mathbb{E}[\zeta_0] \\ &\geq \mathbb{E}[\zeta_0] e^{\log \frac{1-\frac{\rho}{2}}{\mathbb{E}[\zeta_0]}} = 1 - \frac{\rho}{2} \end{aligned} \tag{2.39}$$

Assume the ζ_t produced by GROUSE converges faster than $\bar{\zeta}_t$, i.e.,

$$\mathbb{E}\left[\zeta_{K_1}\right] \geq \mathbb{E}\left[\bar{\zeta}_{K_1}\right] \geq 1 - \frac{\rho}{2} \tag{2.40}$$

Therefore,

$$\mathbb{P}\left(\zeta_{K_1} \geq \frac{1}{2}\right) = 1 - \mathbb{P}\left(1 - \zeta_{K_1} \geq \frac{1}{2}\right) \overset{\vartheta_1}{\geq} 1 - \frac{\mathbb{E}[1 - \zeta_{K_1}]}{1/2} \geq 1 - \rho \tag{2.41}$$

where ϑ_1 follows by applying Markov inequality to the nonnegative random variable $1 - \bar{\zeta}_{K_1}$.

Now with probability at least $1 - \rho$, $\zeta_t \geq \frac{1}{2}$ for all $t \geq K_1$, *i.e.,* $\kappa_t \leq \frac{1}{2}$ for all $t \geq K_1$. So using Eq (2.38b) we have the following:

$$\mathbb{E}\left[\kappa_{t+1} \middle| U\right] \leq \left(1 - \frac{1 - \kappa_t}{d}\right) \kappa_t \leq \left(1 - \frac{1}{2d}\right) \kappa_t \ .$$

Taking expectation of both sides, we have

$$\mathbb{E}\left[\kappa_{t+1}\right] \leq \left(1 - \frac{1}{2d}\right) \mathbb{E}\left[\kappa_t\right] \ .$$

After $K_2 \geq 2d \log \frac{1/2}{\rho(1-\zeta^*)} \geq 2d \log \frac{\mathbb{E}[\eta_{K_1}]}{\rho(1-\zeta^*)}$ additional iterations of GROUSE we obtain

$$\mathbb{E}\left[\kappa_{t+K_1}\right] \leq \left(1 - \frac{1}{2d}\right)^{K_2} \mathbb{E}[\kappa_{K_1}] \leq \left(1 - \frac{1}{2d}\right)^{2d \log \frac{\mathbb{E}[\kappa_{K_1}]}{\rho(1-\zeta^*)}} \mathbb{E}[\kappa_{K_1}] \leq \rho(1 - \zeta^*) \ .$$

Hence following a similar argument as before we have

$$\mathbb{P}\left(\zeta_{K_1+K_2} \geq \zeta^*\right) = 1 - \mathbb{P}\left(\kappa_{K_1+K_2} \geq 1 - \zeta^*\right) \geq 1 - \frac{\mathbb{E}\left[\kappa_{K_1+K_2}\right]}{1 - \zeta^*} \geq 1 - \rho \ . \tag{2.42}$$

(2.41) and (2.42) together complete the proof. □

Although we still need more rigorous analysis to justify our assumption, this proof provides a form of the convergence rate we can expect. We also want to emphasize that the above proof provides the local convergence rate for GROUSE. Specifically, as is indicated by the proof of the second phase, GROUSE requires at most $2d \log \frac{1/2}{\rho(1-\zeta^*)}$ iterations to converge from $\zeta_t = 1/2$ to any required accuracy $\zeta^* \in (1/2, 1)$.

2.8.3 Proof of Undersampled Data

In this section, we prove our main results for undersampled data. We again start by proving a result for the deterministic expression for the change in determinant similarity from one step of the GROUSE algorithm to the next, in this case a lower bound given by Lemma 2.5.2.

Proof of Lemma 2.5.2. Note that,

$$w^T(\bar{U}^T U)^{-1}\bar{U}^T p = w^T(\bar{U}^T U)^{-1}\bar{U}^T U w = \|p\|^2 \tag{2.43a}$$

$$w_1^T(\bar{U}^T U)^{-1}\bar{U}^T r \overset{\vartheta_1}{=} s^T\bar{U}^T U(\bar{U}^T U)^{-1}\bar{U}^T r = v^T A^T\widetilde{r} \overset{\vartheta_2}{=} \|\widetilde{r}\|^2 \tag{2.43b}$$

where ϑ_1 follows by Lemma 2.5.1 and ϑ_2 holds since $v^T A^T\widetilde{r} = v^T A^T\left(\mathbb{I}_m - \mathcal{P}_{AU}\right)\widetilde{r} = \|\widetilde{r}\|^2$. As a consequence, we have the following

$$\begin{aligned}
\det\left(\bar{U}^T U_{t+1}\right) &= \det\left(\bar{U}^T U + \bar{U}^T\left(\frac{p+r}{\|p+r\|} - \frac{p}{\|p\|}\right)\frac{w^T}{\|w\|}\right) \\
&\overset{\vartheta_3}{=} \det(\bar{U}^T U)\frac{w^T(\bar{U}^T U)^{-1}\bar{U}^T(p+r)}{\|p\|\sqrt{\|p\|^2+\|r\|^2}} \\
&= \det(\bar{U}^T U)\frac{\|p\|^2+\|r\|^2+\|\widetilde{r}\|^2-\|r\|^2+\Delta}{\|p\|\sqrt{\|p\|^2+\|r\|^2}}
\end{aligned}$$

where $\Delta = w_2^T\left(\bar{U}^T U\right)^{-1}\bar{U}^T r$; and ϑ_3 follows by the Schur complement $\det\left(M + ab^T\right) = \det(M)\left(1 + b^T M^{-1}a\right)$ for any invertible $M \in \mathbb{R}^{n\times n}$ and $a, b \in \mathbb{R}^n$. Hence

$$\frac{\bar{\zeta}_{t+1}}{\zeta_t} = \left(\frac{\det\left(\bar{U}^T U_{t+1}\right)}{\det\left(\bar{U}^T U\right)}\right)^2 \overset{\vartheta_4}{\geq} 1 + \frac{\|r\|^2}{\|p\|^2} + 2\frac{\|\widetilde{r}\| - \|r\|^2}{\|p\|^2} + 2\frac{\Delta}{\|p\|^2}$$

where ϑ_4 holds since $(c+d)^2 \geq c^2+2cd$ with $c = \frac{\|p\|^2+\|r\|^2}{\|p\|\sqrt{\|p\|^2+\|r\|^2}}$, $d = \frac{\|\widetilde{r}\|^2-\|r\|^2+\Delta}{\|p\|\sqrt{\|p\|^2+\|r\|^2}}$. $\square$

In the following sections, we proceed by establishing the convergence results of missing data and compressively sampled data by bounding the key quantities in Lemma 2.5.2.

Proof for Compressively Sampled Data We start by showing how the results on the key quantities in Lemmas 2.5.3, 2.5.4 and 2.5.5 lead to the main result of the compressively sampled data case.

Proof of Theorem 2.5.1. Let $\eta_1 = \frac{1+\delta}{1-\delta}\frac{d}{m}$, $\eta_2 = (1-\delta)\left(1-2\delta\sqrt{\frac{m}{n}}\right)$ and $\eta_3 = \tan(\phi_d)+\delta\frac{d}{\cos(\phi_d)}$, then plugging in the results in Lemma 2.5.3 to Lemma 2.5.5 into Lemma 2.5.2 with $\delta_1 = \delta_2 = \delta_3 = \delta$ yields,

$$\begin{aligned}
\frac{\zeta_{t+1}}{\zeta_t} &\geq 1 + \frac{2\|\widetilde{r}\|^2 - \|r\|^2}{\|p\|^2} + 2\frac{\Delta}{\|p\|^2} \\
&\geq 1 + \frac{1}{\left(1+\sqrt{\eta_1}\right)^2}\left(\eta_2(1-\eta_1) - 2\sqrt{\eta_1}\eta_3\right)\frac{m}{n}\frac{\|v_\perp\|^2}{\|v\|^2} \\
&= 1 + \gamma_1\left(1-\gamma_2\frac{d}{m}\right)\frac{m}{n}\frac{\|v_\perp\|^2}{\|v\|^2} \\
&= \left(1+\gamma_1\left(1-\gamma_2\frac{d}{m}\right)\frac{m}{n}\right)\frac{1-\zeta_t}{d} \qquad (2.44)
\end{aligned}$$

where $\gamma_2 = \left(1+2\frac{\eta_3}{\eta_2\sqrt{\eta_1}}\right)\frac{1+\delta}{1-\delta} = \left(1+2\frac{\tan(\phi_d)+\delta_3\frac{d}{\cos(\phi_d)}}{\left(1-2\delta\sqrt{\frac{m}{n}}\right)\sqrt{(1-\delta^2)d/m}}\right)\frac{1+\delta}{1-\delta}$, $\gamma_1 = \frac{\eta_2}{\left(1+\sqrt{\eta_1}\right)^2} = \frac{(1-\delta)\left(1-2\delta\sqrt{\frac{m}{n}}\right)}{\left(1+\sqrt{\frac{1+\delta}{1-\delta}\frac{d}{m}}\right)^2}$, and the last equality follows from Lemma 2.3.1.

The probability bound is obtained by taking the union bound of those quantities (in Lemma 2.8.5, Lemma 2.8.8, Lemma 2.8.7, Corollary 2.8.2, Lemma 2.8.16) used to generate Lemma 2.5.3 to Lemma 2.5.5. As we can see, this union bound is

$$\begin{aligned}
&1-\exp\left(-\frac{m\delta^2}{2}\right)-\exp\left(-\frac{d\delta^2}{8}\right)-\exp\left(-\frac{m\delta^2}{32}+d\log\left(\frac{24}{\delta}\right)\right)-(4d+1)\exp\left(-\frac{m\delta^2}{8}\right) \\
&\qquad > 1-\exp\left(-\frac{d\delta^2}{8}\right)-\exp\left(-\frac{m\delta^2}{32}+d\log\left(\frac{24}{\delta}\right)\right)-(4d+2)\exp\left(-\frac{m\delta^2}{8}\right) \qquad (2.45)
\end{aligned}$$

To get the complexity bound on m, let $\varepsilon = \tan(\phi_d)$, $\alpha_1 = \varepsilon + \delta\sqrt{1+\varepsilon^2}d$, $\alpha_2 = \frac{1+\delta}{1-\delta}$ and $\alpha_3 = \left(1-2\delta\sqrt{\frac{m}{n}}\right)\sqrt{1+\delta}$, then according to 2.54 we have $\gamma_2\frac{d}{m} < \frac{1}{2}$ is equivalent

to the following,

$$\begin{aligned}
&\alpha_2 d + \frac{2\alpha_1\alpha_2\sqrt{d}}{\alpha_3}\sqrt{m} < \frac{m}{2} \\
&\Leftrightarrow \left(\sqrt{\frac{m}{2}} - \frac{\alpha_1\alpha_2\sqrt{d}}{\alpha_3}\right)^2 > \left(\alpha_2 + \frac{\alpha_1^2\alpha_2^2}{\alpha_3^2}\right) d \\
&\overset{\vartheta_1}{\Leftarrow} m \geq 8\frac{\alpha_1^2\alpha_2^2}{\alpha_3^2} d + 4\sqrt{\alpha_2}\frac{\alpha_1\alpha_2}{\alpha_3} d \\
&\overset{\vartheta_2}{\Leftarrow} m \geq \beta\left(\varepsilon + \delta\sqrt{1+\varepsilon^2} d\right)\left(\varepsilon + \delta\sqrt{1+\varepsilon^2} d + \frac{1}{2}\right) d
\end{aligned} \tag{2.46}$$

where ϑ_1 follows from $\sqrt{\left(\alpha_2 + \frac{\alpha_1^2\alpha_2^2}{\alpha_3^2}\right) d} < \sqrt{\alpha_2 d} + \frac{\alpha_1\alpha_2}{\alpha_3}\sqrt{d}$; and ϑ_2 follows by $\beta = \frac{8(1+\delta)}{(1-\delta)^2(1-2\delta)^2}$.

To establish another bound on m we can see that $m \geq \frac{32}{\delta^2}\log\left(\frac{24n^{2/d}}{\delta}\right) d$ implies the following,

$$\exp\left(-\frac{m\delta^2}{32} + d\log\left(\frac{24}{\delta}\right)\right) \leq \exp(-\log n^2) = \frac{1}{n^2} \tag{2.47}$$

$$(4d+2)\exp\left(-\frac{m\delta^2}{8}\right) \leq \frac{(4d+2)}{n^8}\left(\frac{\delta}{24}\right)^{4d} \to 0 \tag{2.48}$$

Eqs (2.47) and (2.48) complete the proof for the bound on m and justify the simplification of the probability bound in Eq (2.45). □

Next we are going to prove the intermediate lemmas in Section 2.5.1, *i.e.,* bound the key quantities in Lemma 2.5.2, for which we need the following concentration results.

Lemma 2.8.5. *Let $A \in \mathbb{R}^{m\times n}$ with entries being i.i.d Gaussian random variables distributed as $\mathcal{N}(0, 1/n)$, $v \in \mathbb{R}^n$ is an vector. Then for any $\delta \in (0,1)$, with probability at least $1 - 2\exp^{-m\delta^2/8}$, we have*

$$\begin{aligned}
\mathbb{P}\left(\|Av\|_2^2 > (1+\delta)\frac{m}{n}\|v\|_2^2\right) &< \exp\left(-\frac{m\delta^2}{8}\right), \\
\mathbb{P}\left(\|Av\|_2^2 < (1-\delta)\frac{m}{n}\|v\|_2^2\right) &< \exp\left(-\frac{m\delta^2}{8}\right).
\end{aligned}$$

Proof. Note that Av is a random vector with i.i.d entries distributing as $\mathcal{N}(0, \|v\|_2^2/n)$. Therefore, $\frac{n\|Av\|_2^2}{\|v\|_2^2}$ is a chi-squared distribution with m degrees of freedom, which yields,

$$\mathbb{P}\left[\frac{n\|Av\|_2^2}{m\|v\|_2^2} - 1 > \delta\right] < \exp\left(-m\delta^2/8\right)$$
$$\mathbb{P}\left[\frac{n\|Av\|_2^2}{m\|v\|_2^2} - 1 < -\delta\right] < \exp\left(-m\delta^2/8\right)$$

□

Lemma 2.8.6. *Let $A \in \mathbb{R}^{m\times n}$ be a random matrix whose entries are independent and identically distributed Gaussian random variables with mean zero, and variance γ . Let $z_1, z_2 \in \mathbb{R}^n$ such that $z_1 \perp z_2$, then Az_1 and Az_2 are independent of each other.*

Proof. Let a_i^T denote the i^{th} row of A and $M = Az_1z_2^TA^T$. Then we have

$$\mathbb{E}[M]_{ii} = \mathbb{E}\left[a_i^Tz_1z_1^Ta_i\right] = z_1^T\mathbb{E}[a_ia_i^T]z_2 = \gamma z_1^Tz_2 = 0$$
$$\mathbb{E}[M]_{ij} = \mathbb{E}\left[a_i^Tz_1z_1^Ta_j\right] = z_1^T\mathbb{E}[a_ia_j^T]z_2 = 0$$

Therefore Az_1 and Az_2 are uncorrelated. This together with the fact that both Az_1 and Az_2 are Gaussian distributed random vectors imply that Az_1 and Az_2 are independent. □

Lemma 2.8.7 ([129], Corollary 5.35)**.** *Let A be an $n \times m$ matrix ($n \geq m$) whose entries are independent standard normal random variables. Then for every $h \geq 0$, with probability at least $1 - 2\exp(-h^2/2)$ one has*

$$\sqrt{n} - \sqrt{m} - h \leq \sigma_{min}(A) \leq \sigma_{max}(A) \leq \sqrt{n} + \sqrt{m} + h \tag{2.49}$$

where $\sigma_{min}, \sigma_{max}$ denote the smallest and largest singular values of A.

With the above results, we are able to call out the following intermediate result to quantify $\|\mathcal{P}_{AU}(Av_\perp)\|_2^2$, which is a key quantity that will be used for proving Lemmas 2.5.3, 2.5.4 and 2.5.5.

Lemma 2.8.8. *Let $A \in \mathbb{R}^{m\times n}$ with entries being i.i.d Gaussian random variables distributed as $\mathcal{N}(0, 1/n)$, then for any $\delta \in (0, 1)$ we have*

$$\|\mathcal{P}_{AU} Av_\perp\|_2^2 \leq (1+\delta)\frac{d}{n}\|v_\perp\|_2^2$$

hold with probability at least $1 - \exp\left(-\frac{d\delta^2}{8}\right)$.

Proof. Note that $Av_\perp$ is a Gaussian random vector with i.i.d entries distributed as $\mathcal{N}(0, \|v_\perp\|_2^2/n)$, and AU is a Gaussian random matrix with i.i.d entries distributed as $\mathcal{N}(0, 1/n)$. Then according to Lemma 2.8.6, AU and $Av_\perp$ are independent of each other. Therefore, $y = \mathcal{P}_{AU}(Av_\perp)$ is the projection of $Av_\perp$ onto a independent random d-dimensional subspace. According to the rotation invariance property of $Av_\perp$, $\|\mathcal{P}_{AU}(Av_\perp)\|$ is equivalent to the length of projecting $Av_\perp$ onto its first d coordinates. Hence,

$$\mathbb{P}\left(\|\mathcal{P}_{AU}(Av_\perp)\|_2^2 = \sum_{k=1}^{d} y_k^2 \leq (1+\delta)\frac{d}{n}\|v_\perp\|_2^2\right) \geq 1 - \exp\left(-\frac{d\delta^2}{8}\right) \tag{2.50}$$

Similar to the proof for Lemma 2.8.5, here the probability bound is followed from the concentration bound for Chi-squared distribution with degree d. □

Now we start by proving that Lemma 2.5.3 follows directly from Lemma 2.8.5 and Lemma 2.8.7.

Proof of Lemma 2.5.3. According to Lemmas 2.8.5 and 2.8.8, we have

$$\begin{aligned}\|\widetilde{r}\|_2^2 = \|\left(\mathbb{I}_m - \mathcal{P}_{AU}\right) Av_\perp\|_2^2 &= \|Av_\perp\|_2^2 - \|\mathcal{P}_{AU}(Av_\perp)\|_2^2 \\ &\geq (1-\delta_1)\frac{m}{n}\|v_\perp\|_2^2 - (1+\delta_1)\frac{d}{n}\|v_\perp\|_2^2 \\ &= (1-\delta_1)\left(1 - \frac{1+\delta_1}{1-\delta_1}\frac{d}{m}\right)\frac{m}{n}\|v_\perp\|_2^2 \end{aligned} \tag{2.51}$$

hold with probability at least $1-\exp\left(-\frac{m\delta_1^2}{8}\right)-\exp\left(-\frac{d\delta_1^2}{8}\right)$. As for the second part of Lemma 2.5.3, we have

$$\begin{aligned}2\|\widetilde{r}\|_2^2 - \|r\|_2^2 = 2\|\widetilde{r}\|_2^2 - \|A^T\widetilde{r}\|_2^2 &\geq (2-\sigma_{\max}^2(A^T))\|\widetilde{r}\|_2^2 \\ &\overset{\vartheta_1}{\geq} \left(1 - 2\delta_2\sqrt{\frac{m}{n}}\right)\|\widetilde{r}\|_2^2 \\ &\geq \left(1 - 2\delta_2\sqrt{\frac{m}{n}}\right)(1-\delta_1)\left(1 - \frac{1+\delta_1}{1-\delta_1}\frac{d}{m}\right)\frac{m}{n}\|v_\perp\|_2^2 \end{aligned} \tag{2.52}$$

here ϑ_1 follows from Lemma 2.8.7 with $A_{ij} \sim \mathcal{N}(0, 1/n)$ and $h = \delta\sqrt{m/n}$. The probability bound $1-\exp\left(-\frac{m\delta_1^2}{8}\right)-\exp\left(-\frac{d\delta_1^2}{8}\right)-\exp\left(-\frac{m\delta_2^2}{2}\right)$ is obtained by taking the union bound over 2.51 and ϑ_1. □

To prove Lemma 2.5.4 and Lemma 2.5.5, we need the following extra results which are implied by Lemma 2.8.5. The corresponding proofs are provided at the end of this section.

Corollary 2.8.1. *Under the conditions of Lemma 2.8.5, for $x, y \in \mathbb{R}^n$ and δ, with probability exceeding $1 - 4e^{-m\delta^2/8}$ we have*

$$\frac{m}{n}\left(x^Ty - \delta\|x\|\|y\|\right) \leq x^TA^TAy \leq \frac{m}{n}\left(x^Ty + \delta\|x\|\|y\|\right)$$

Corollary 2.8.2. *Under the condition of Lemma 2.8.5, for any vector $v \in R(U)$ we*

have

$$\mathbb{P}\left(\|Av\|_2^2 > (1+\delta)\frac{m}{n}\|v\|_2^2\right) < \exp\left(-\frac{m\delta^2}{32} - d\log(\delta) + d\log(24)\right) ,$$
$$\mathbb{P}\left(\|Av\|_2^2 < (1-\delta)\frac{m}{n}\|v\|_2^2\right) < \exp\left(-\frac{m\delta^2}{32} - d\log(\delta) + d\log(24)\right) .$$

Given Lemma 2.8.1 and Corollary 2.8.2, we prove Lemma 2.5.4 and Lemma 2.5.5 by first proving the following intermediate results to bound the key components of p and Δ.

Lemma 2.8.9. *Let* $w_2 = \left(U^T A^T A U\right)^{-1} U^T A^T A v_\perp$, *then*

$$\mathbb{P}\left(\|w_2\| \le \sqrt{\frac{1+\delta_1}{1-\delta_2}\frac{d}{m}}\|v_\perp\|\right)$$
$$\ge 1 - \exp\left(-\frac{d\delta_1^2}{8}\right) - \exp\left(-\frac{m\delta_2^2}{8} - d\log(\delta_2) + d\log(24)\right)$$

Proof. Given the fact that $U \in \mathbb{R}^{n\times d}$ with columns being orthonormal, we have $\|w_2\| = \|Uw_2\|$. It then follows that,

$$\|Uw_2\| \overset{\vartheta_1}{\le} \frac{\|AUw_2\|}{\sqrt{(1-\delta_2)m/n}} \overset{\vartheta_2}{\le} \sqrt{\frac{1+\delta_1}{1-\delta_2}\frac{d}{m}}\|v_\perp\|$$

where ϑ_1 follows from Corollary 2.8.2, and ϑ_2 followed by Lemma 2.8.8, *i.e.,*

$$\|AUw_2\| = \|\mathcal{P}_{AU}(Av_\perp)\| \le \sqrt{(1+\delta_1)\frac{d}{n}\|v_\perp\|^2}$$

The probability bound is obtained by applying the union bound over ϑ_1 and ϑ_2. □

Lemma 2.8.10. *Let* ϕ_d *denote the largest principal angle between* $R(U)$ *and* $R(\bar{U})$, *then*

$$\mathbb{P}\left(\left\|\bar{U}^T A^T A v_\perp\right\| \le (\sin\phi_d + d\delta)\frac{m}{n}\|v_\perp\|\right) \ge 1 - 4d\exp\left(-\frac{m\delta^2}{8}\right)$$

Proof of Lemma 2.8.10. Let $\bar{u}_k$ denote the k^{th} column of $\bar{U}$, and $\delta \in (0,1)$. Then

$$\begin{aligned}
\left\|\bar{U}^T A^T A v_\perp\right\| &= \left\|\bar{U}^T \left(A^T A - \frac{m}{n}\mathbb{I}_n\right) v_\perp + \frac{m}{n}\bar{U}^T v_\perp\right\| \\
&\leq \frac{m}{n}\left\|\bar{U}^T v_\perp\right\| + \left\|\bar{U}^T \left(A^T A - \frac{m}{n}\mathbb{I}_n\right) v_\perp\right\| \\
&= \frac{m}{n}\left\|\bar{U}^T v_\perp\right\| + \sqrt{\sum_{k=1}^{d}\left(\bar{u}_k^T A^T A v_\perp - \frac{m}{n}\bar{u}_k^T v_\perp\right)^2} \\
&\overset{\vartheta_1}{\leq} \frac{m}{n}\left\|\bar{U}^T v_\perp\right\| + \sqrt{\sum_{k=1}^{d}\left(\delta\frac{m}{n}\|\bar{u}_k\|\|v_\perp\|\right)^2} \\
&\overset{\vartheta_2}{\leq} \sin\phi_d \frac{m}{n}\|v_\perp\| + \frac{m}{n}d\delta\|v_\perp\| \qquad (2.53)
\end{aligned}$$

where ϑ_1 follows from Lemma 2.8.1; ϑ_2 holds from Lemma 2.8.16 and the fact that $\sqrt{\sum_{k=1}^{d}\left(\delta\frac{m}{n}\|\bar{u}_k\|\|v_\perp\|\right)^2} \leq d\delta\frac{m}{n}\|\bar{u}_k\|\|v_\perp\|$; and the probability bound is obtained by taking the union bound of that in Lemma 2.8.1. □

We are ready to prove Lemma 2.5.4 and Lemma 2.5.5.

Proof of Lemma 2.5.4. Let $\eta = \sqrt{\frac{1+\delta_1}{1-\delta_1}\frac{d}{m}}$, then according to Lemma 2.8.9 we have

$$\begin{aligned}
\|p\|^2 = \|Uw_1 + Uw_2\|^2 &\leq \left(\|v_\parallel\| + \|Uw_2\|\right)^2 \\
&\leq \left(\|v_\parallel\| + \eta\|v_\perp\|\right)^2 \\
&\leq (1+\eta)^2\|v\|^2
\end{aligned}$$

with probability at least

$$1 - \exp\left(-\frac{m\delta_1^2}{32} - d\log(\delta_1) + d\log(24)\right) - \exp\left(-\frac{d\delta_1^2}{8}\right) .$$

Here the probability bound is obtained by choosing $\delta_1 = \delta_2$ in Lemma 2.8.9, hence completes the proof. □

Proof of Lemma 2.5.5. According to the definition of Δ, we can see Lemma 2.5.5 is

a direct results of Lemma 2.8.9 and Lemma 2.8.16, that is

$$
\begin{aligned}
|\Delta| &= w_2^T \left(\bar{U}^T U\right)^{-1} \bar{U}^T A^T \left(\mathbb{I}_m - \mathcal{P}_{AU}\right) A v_\perp \\
&\leq \left\|w_2^T\right\| \left\|\left(\bar{U}^T U\right)^{-1}\right\| \left\|\bar{U}^T A^T \left(\mathbb{I}_m - \mathcal{P}_{AU}\right) A v_\perp\right\| \\
&\overset{\vartheta_1}{\leq} \|w_2\| \left\|\left(\bar{U}^T U\right)^{-1}\right\| \left\|\bar{U}^T A^T A v_\perp\right\| \\
&\overset{\vartheta_2}{\leq} \frac{1}{\cos(\phi_d)} \sqrt{\frac{1+\delta_1}{1-\delta_1}\frac{d}{m}} \|v_\perp\| \left(\sin\phi_d \frac{m}{n} + \frac{m}{n} d\delta_3\right) \|v_\perp\| \\
&= \frac{1}{\cos(\phi_d)} \sqrt{\frac{1+\delta_1}{1-\delta_1}\frac{d}{m}} \left(\sin(\phi_d) + d\delta_3\right) \frac{m}{n} \|v_\perp\|^2 \qquad (2.54)
\end{aligned}
$$

where ϑ_1 holds since $\left\|\bar{U}^T A^T \left(\mathbb{I}_m - \mathcal{P}_{AU}\right) A v_\perp\right\| \leq \left\|\bar{U}^T A^T A v_\perp\right\|$; ϑ_2 followed by Lemma 2.8.9 and Lemma 2.8.10; and the probability bound is obtained by taking the union bound that in Lemma 2.8.9 and Lemma 2.8.10. □

Finally, we are going to prove the auxiliary results Corollary 2.8.2 and Lemma 2.8.1. The key idea for proving Corollary 2.8.2 is using the covering numbers argument and applying Lemma 2.5.3 to a given d-dimensional subspace $R(U)$. This is a common strategy used for compress sensing.

Proof of Corollary 2.8.2. Without loss of generality we restrict $\|v\| = 1$. From covering numbers [121], there exists a finite set Q with at most $\left(\frac{24}{\delta}\right)^d$ points such that $Q \subset \mathbb{R}(U)$, $\|q\| = 1, \forall q \in Q$, and for all $x \in R(U)$ with $\|v\| = 1$ we can find a $q \in Q$ such that

$$\|v - q\| \leq \delta/8$$

Now applying Lemma 2.8.5 to the points in Q with $\varepsilon = \delta/2$ and using the standard union bound, then with probability at least $1 - 2\left(\frac{24}{\delta}\right)^d \exp\left(-\frac{\delta^2}{32}m\right)$ we have

$$(1 - \delta/2)\frac{m}{n}\|v\|^2 \leq \|Ax\|^2 \leq (1 + \delta/2)\frac{m}{n}\|v\|^2$$

which gives

$$\sqrt{1-\delta/2}\sqrt{\frac{m}{n}}\|v\| \leq \|Ax\| \leq \sqrt{1+\delta/2}\sqrt{\frac{m}{n}}\|v\| \tag{2.55}$$

Since $\|v\| = 1$, we define γ as the smallest number such that

$$\|Ax\| \leq \sqrt{1+\gamma}\sqrt{\frac{m}{n}} \quad \forall x \in R(U) \tag{2.56}$$

Since for any $x \in R(U)$ with $\|v\| = 1$ we can find a $q \in Q$ such that $\|x - q\| \leq \delta/8$, we have the following

$$\|Ax\| \leq \|Aq\| + \|A(x-q)\| \leq \sqrt{1+\delta/2}\sqrt{\frac{m}{n}} + \sqrt{1+H}\sqrt{\frac{m}{n}}\delta/8$$

Since γ is the smallest number (2.56) holds, we have $\sqrt{1+\gamma} \leq \sqrt{1+\delta/2} + \sqrt{1+\gamma}\delta/8$.

$$\sqrt{1+\gamma} \leq \frac{\sqrt{1+\delta/2}}{1-\delta/8} \leq \sqrt{1+\delta} \tag{2.57}$$

Similarly, the lower bound follows by

$$\begin{aligned}
\|Ax\| \geq \|Aq\| - \|A(x-q)\| &\geq \sqrt{1-\delta/2}\sqrt{\frac{m}{n}} - \sqrt{1+\gamma}\frac{\delta}{8}\sqrt{\frac{m}{n}} \\
&\geq \left(\sqrt{1-\delta/2} - \sqrt{1+\delta}\frac{\delta}{8}\right)\sqrt{\frac{m}{n}} \\
&\geq \sqrt{1-\delta}\sqrt{\frac{m}{n}}
\end{aligned}$$

This completes the proof. □

Proof of Lemma 2.8.1. Note that,

$$\frac{x^T A^T A y}{\|x\|\|y\|} = \frac{1}{4}\left(\left\|A\left(\frac{x}{\|x\|} + \frac{y}{\|y\|}\right)\right\|^2 - \left\|A\left(\frac{x}{\|x\|} - \frac{y}{\|y\|}\right)\right\|^2\right)$$

Applying Lemma 2.8.5 on both terms separately and applying the union bound we

have

$$\mathbb{P}\left[\frac{x^T A^T A y}{\|x\|\|y\|} \leq \frac{m}{n}\left(\frac{x^T y}{\|x\|\|y\|} - \delta\right)\right]$$
$$= \mathbb{P}\left[\frac{x^T A^T A y}{\|x\|\|y\|} \leq \frac{1}{4}\left((1-\delta)\frac{m}{n}\left\|\frac{x}{\|x\|} + \frac{y}{\|y\|}\right\|^2 - (1+\delta)\frac{m}{n}\left\|\frac{x}{\|x\|} - \frac{y}{\|y\|}\right\|^2\right)\right]$$

(2.58)
$$< 2\exp\left(-\frac{m\delta^2}{8}\right)$$

Similarly,

$$\mathbb{P}\left[\frac{x^T A^T A y}{\|x\|\|y\|} \geq \frac{m}{n}\left(\frac{x^T y}{\|x\|\|y\|} + \delta\right)\right]$$
$$= \mathbb{P}\left[\frac{x^T A^T A y}{\|x\|\|y\|} \geq \frac{1}{4}\left((1+\delta)\frac{m}{n}\left\|\frac{x}{\|x\|} + \frac{y}{\|y\|}\right\|^2 - (1-\delta)\frac{m}{n}\left\|\frac{x}{\|x\|} - \frac{y}{\|y\|}\right\|^2\right)\right]$$

(2.59)
$$< 2\exp\left(-\frac{m\delta^2}{8}\right)$$

holds with probability no more than 2.58 and 2.59 complete the proof. □

Proof of Missing Data Here we again bound the quantities in Lemma 2.5.2, Equation 2.14, this time assuming A represents an entry-wise observation operation and assuming incoherence on the signals of interest. As we show below, in the proof of Theorem 2.5.2, we put together bounds given by Lemmas 2.5.6, 2.5.7 and 2.5.8, which are all proved in this section too, along with Lemma 2.5.9 for completeness. We start by proving the main result for missing data.

Proof of Theorem 2.5.2. Given the condition required by Theorem 2.5.2, we have $\sin\phi_d \leq \sqrt{d\mu_0/16n}$ and $\cos\phi_d \geq \sqrt{1-d\mu_0/16n}$. This together with Lemma 2.5.9 and Lemma 2.5.8 yield $|\Delta| \leq \frac{\eta_3\sqrt{1+\frac{m}{16n}}}{\sqrt{1-d\mu_0/16n}}\frac{2d\mu_0}{n}\|v_\perp\|^2 \leq \frac{2\eta_3\sqrt{1+\frac{1}{16}}}{\sqrt{1-\frac{1}{16}}}\frac{d\mu_0}{n}\|v_\perp\|^2 \leq$

$\frac{11}{5}\eta_3\frac{d\mu_0}{n}\|v_\perp\|^2$. Also for β_2 in Lemma 2.5.8 we have $\beta_2 \leq \sqrt{2\mu(v_\perp)\log(1/\delta)} = \beta_1$. Therefore,

$$|\Delta| \leq \frac{11}{5}\frac{(1+\beta_1)^2}{1-\gamma_1}\frac{d\mu_0}{n}\|v_\perp\|^2 . \tag{2.60}$$

Letting $\eta_2 = \frac{(1+\beta_1)^2}{1-\gamma_1}\frac{d\mu_0}{m}$ and $\alpha_1 = \sqrt{\frac{2\mu(v_\perp)^2}{m}\log\left(\frac{1}{\delta}\right)}$, then applying this definition together with Lemma 2.5.9 to Lemma 2.5.7 Lemma 2.5.6 yields

$$\|p\|^2 \leq \left(1+\sqrt{\frac{2\eta_2}{1-\gamma_1}}\right)^2\|v\|^2 \tag{2.61}$$

$$\|r_\Omega\|^2 \geq (1-\alpha_1-2\eta_2)\frac{m}{n}\|v_\perp\|^2 \tag{2.62}$$

Now applying 2.60, 2.61 and 2.62 to 2.24 we obtain

$$\begin{aligned}\frac{\zeta_{t+1}}{\zeta_t} &\geq 1+\frac{(1-\alpha_1-2\eta_2)}{(1+\sqrt{2\eta_2/(1-\gamma_1)})^2}\frac{m}{n}\frac{\|v_\perp\|^2}{\|v\|^2} - \frac{22}{5}\frac{\eta_2}{(1+\sqrt{2\eta_2/(1-\gamma_1)})^2}\frac{m}{n}\frac{\|v_\perp\|^2}{\|v\|^2} \\ &\geq 1+\frac{(1-\alpha_1-\frac{32}{5}\eta_2)}{(1+\sqrt{2\eta_2/(1-\gamma_1)})^2}\frac{m}{n}\frac{\|v_\perp\|^2}{\|v\|^2} \qquad (2.63)\end{aligned}$$

which holds with probability at least $1-3\delta$. The probability bound is obtained by taking the union bound of those generating Lemmas 2.5.6, 2.5.7 and 2.5.8, as we can see in the proofs of them in this Section, this union bound is at least $1-3\delta$.

Letting $\eta_1 = \frac{(1-\alpha_1-\frac{32}{5}\eta_2)}{(1+\sqrt{2\eta_2/(1-\gamma_1)})^2}$, then $\eta_1 > 0$ is equivalent to $1-\alpha_1-\frac{32}{5}\eta_2 > 0$, for which we have the following: if

(2.64)

$$m > \max\left\{\frac{128d\mu_0}{3}\log\left(\frac{2d}{\delta}\right), 32\mu(v_\perp)^2\log\left(\frac{1}{\delta}\right), 52d\mu_0\left(1+\sqrt{2\mu(v_\perp)\log\left(\frac{1}{\delta}\right)}\right)^2\right\}$$

then $\eta_1 > \frac{1}{4}$.

Under this condition, taking expectation with respect to v yields,

$$\mathbb{E}_v\left[\frac{\zeta_{t+1}}{\zeta_t}\Big|U\right] \geq 1+\frac{1}{4}\frac{m}{n}\mathbb{E}\left[\frac{\|v_\perp\|^2}{\|v\|^2}\Big|U\right] \geq 1+\frac{1}{4}\frac{m}{n}\frac{1-\zeta_t}{d} \tag{2.65}$$

where the last inequality follows from Lemma 2.3.1. Finally choosing δ to be $1/n^2$completes the proof. □

We then prove Corollary 2.5.1, the result that allows comparison between our convergence rate and that in [17].

Proof of Corollary 2.5.1. Let $X = [X_1, \ldots, X_d]$ with $X_i = \sin^2 \phi_i$. Let $f(X) = 1 - \sum_{i=1}^{d} X_i - \Pi_{i=1}^{d}(1 - X_i)$, then $\frac{\partial f(X)}{\partial X_i} = -1 + \Pi_{j \neq i}(1 - X_j) \leq 0$. That is, $f(X)$ is a decreasing function of each component. Therefore, $f(X) \leq f(0) = 0$. It follows that

$$\zeta_t = \Pi_{i=1}^{d}(1 - X_i) \geq 1 - \sum_{i=1}^{d} X_i \geq 1 - \frac{d\mu_0}{16n} \tag{2.66}$$

With a slight modification of Theorem 2.5.2 we obtain

$$\mathbb{E}\left[\kappa_{t+1} \middle| \kappa_t\right] \leq \left(1 - \frac{1}{4}\frac{m}{n}\frac{\zeta_t}{d}\right)\kappa_t \, . \tag{2.67}$$

(2.66) and (2.67) together complete the proof. □

We now focus on proving the key lemmas for establishing Theorem 2.5.2, for which we need the following lemmas (the proofs can be found in [16]).

Lemma 2.8.11. *[16] Let $\delta > 0$. Suppose $m \geq \frac{8}{3} d\mu(U) \log(2d/\delta)$, then*

$$\mathbb{P}\left(\left\|\left(U_\Omega^T U_\Omega\right)^{-1}\right\| \leq \frac{n}{(1 - \gamma_1)m}\right) \geq 1 - \delta$$

where $\gamma_1 = \sqrt{\frac{8d\mu(U)}{3m} \log(2d/\delta)}$.

Lemma 2.8.12 ([16], Lemma 1)**.** *Let $\alpha = \sqrt{\frac{2\mu(v_\perp)^2}{m} \log(1/\delta)}$, then*

$$\mathbb{P}\left(\|v_{\perp,\Omega}\|^2 \geq (1 - \alpha)\frac{m}{n}\|v_\perp\|^2\right) \geq 1 - \delta$$

Lemma 2.8.13 ([16], Lemma 2)**.** *Let $\mu(U), \mu(v_\perp)$ denote the incoherence parameters of $R(U)$ and $v_\perp$, and let $\delta \in (0, 1)$ and $\beta_1 = \sqrt{2\mu(v_\perp)\log(1/\delta)}$, then*

$$\mathbb{P}\left(\left\|U_\Omega^T v_{\perp,\Omega}\right\|^2 \leq (\beta_1 + 1)^2 \frac{m}{n}\frac{d\mu(U)}{n}\|v_\perp\|^2\right) \geq 1 - \delta$$

Now we are ready for the proof of Lemmas 2.5.6, 2.5.7 and 2.5.8.

Proof of Lemma 2.5.6. According to Lemmas 2.8.12, 2.8.13 and 2.8.11, we have

$$\begin{aligned}\|r_\Omega\|^2 &= \|v_{\perp,\Omega}\|^2 - v_{\perp,\Omega}^T U_\Omega \left(U_\Omega^T U_\Omega\right)^{-1} U_\Omega^T v_{\perp,\Omega} \\ &\geq \|v_{\perp,\Omega}\|^2 - \left\|\left(U_\Omega^T U_\Omega\right)^{-1}\right\| \|U_\Omega^T v_{\perp,\Omega}\|^2 \\ &\overset{\vartheta_1}{\geq} \left(1 - \alpha - \frac{(\beta_1+1)^2}{1-\gamma_1}\frac{d\mu(U)}{m}\right)\frac{m}{n}\|v_\perp\|^2\end{aligned}$$

with probability at least $1-3\delta$. □

Proof of Lemma 2.5.7. Lemma 2.8.13 and Lemma 2.8.11 together give the following

$$\begin{aligned}\|Uw_2\|^2 = \left\|\left(U_\Omega^T U_\Omega\right)^{-1} U_\Omega^T v_{\perp,\Omega}\right\|^2 &\leq \left\|\left(U_\Omega^T U_\Omega\right)^{-1}\right\|^2 \left\|U_\Omega^T v_{\perp,\Omega}\right\|^2 \\ &\leq \frac{(\beta_1+1)^2}{(1-\gamma_1)^2}\frac{d\mu(U)}{m}\|v_\perp\|^2\end{aligned}$$

holds with probability exceeding $1-2\delta$. Therefore,

$$\|p\|^2 \leq \left(\|v_\parallel\| + \|Uw_2\|\right)^2 \leq \left(1 + \frac{\beta_1+1}{1-\gamma_1}\sqrt{\frac{d\mu(U)}{m}}\right)^2 \|v\|^2$$

□

We also need the following lemma for the proof of Lemma 2.5.8, the proof of which is provided at the end of this section.

Lemma 2.8.14. *Let* $\beta_2 = \sqrt{2\mu(v_\perp)\log\left(\frac{1}{\delta}\right)\frac{d\mu_0}{d\mu_0 + m\sin^2\phi_d}}$, *where again* μ_0 *denoting the incoherence parameter of* $R(\bar{U})$. *Then*

$$\mathbb{P}\left(\left\|\bar{U}_\Omega^T v_{\perp,\Omega}\right\| \leq (1+\beta_2)\sqrt{\frac{m}{n}\frac{d\mu_0}{n}}\sqrt{\frac{m\sin^2\phi_d}{d\mu_0}+1}\|v_\perp\|\right) \geq 1-\delta$$

Proof of Lemma 2.5.8. Note that $|\Delta| = \|\Delta\|$, for which we have the following,

$$\begin{aligned}
\|\Delta\| &= \left\|w_2^T(\bar{U}^T U)^{-1}\bar{U}^T r\right\| \\
&= \left\|v_{\perp,\Omega}^T U_\Omega \left(U_\Omega^T U_\Omega\right)^{-1}\left(\bar{U}^T U\right)^{-1}\bar{U}_\Omega^T\left(I - \mathcal{P}_{U_\Omega}\right)v_{\perp,\Omega}\right\| \\
&\leq \left\|v_{\perp,\Omega}^T U_\Omega\right\|\left\|\left(U_\Omega^T U_\Omega\right)^{-1}\right\|\left\|\left(\bar{U}^T U\right)^{-1}\right\|\left\|\bar{U}_\Omega^T\left(I - \mathcal{P}_{U_\Omega}\right)v_{\perp,\Omega}\right\| \\
&\overset{\vartheta_1}{\leq} \frac{1}{\cos\phi_d}\left\|v_{\perp,\Omega}^T U_\Omega\right\|\left\|\left(U_\Omega^T U_\Omega\right)^{-1}\right\|\left\|\bar{U}_\Omega^T v_{\perp,\Omega}\right\| \\
&\leq \frac{1}{\cos\phi_d}(\beta_1+1)\sqrt{\frac{m}{n}\frac{d\mu(U)}{n}}(1+\beta_2)\sqrt{\frac{m}{n}\frac{d\mu_0}{n}}\sqrt{\frac{m\sin^2\phi_d}{d\mu_0}+1}\frac{n}{m(1-\gamma_1)}\|v_\perp\|^2 \\
&\overset{\vartheta_2}{\leq} \frac{(1+\beta_1)(1+\beta_2)}{(1-\gamma_1)\cos\phi_d}\sqrt{\frac{m\sin^2\phi_d}{d\mu_0}+1}\sqrt{\frac{d\mu_0}{n}}\sqrt{\frac{d\mu(U)}{n}}\|v_\perp\|^2
\end{aligned}$$

where ϑ_1 holds since from the following:

$$\left\|\bar{U}_\Omega^T\left(I - \mathcal{P}_{U_\Omega}\right)v_{\perp,\Omega}\right\| \leq \left\|\bar{U}_\Omega^T v_{\perp,\Omega}\right\| \ , \qquad \left\|\left(U_\Omega^T U_\Omega\right)^{-1}\right\| \leq \frac{1}{\cos\phi_d}$$

and ϑ_2 follows by putting Lemmas 2.8.13, 2.8.11 and 2.8.14 together. □

We also prove Lemma 2.5.9 for completeness. Before that we first call out the following lemma, the proof of which can be found in [17].

Lemma 2.8.15. *[17] There exists an orthogonal matrix $V \in \mathbb{R}^{d\times d}$ such that*

$$\sum_{k=1}^{d}\sin^2\phi_k \leq \left\|\bar{U}V - U\right\|_F^2 \leq 2\sum_{k=1}^{d}\sin^2\phi_k$$

Proof of Lemma 2.5.9. According to Lemma 2.8.15 we have

$$\begin{aligned}
\|U_i\|_2 \leq \left\|\bar{U}_i\right\|_2 + \left\|\bar{U}_i V - U_i\right\|_2 &\leq \left\|\bar{U}_i\right\| + \sqrt{2\sum_{k=1}^{d}\sin^2\phi_k} \\
&\leq \left(1 + \frac{1}{2\sqrt{2}}\right)\sqrt{\frac{d\mu_0}{n}}
\end{aligned}$$

It hence follows that $\|U_i\|_2^2 \leq 2\frac{d\mu_0}{n}$. □

We need the following lemma and McDiarmid's inequality to prove Lemma 2.8.16.

Lemma 2.8.16. $\left\|\bar{U}^T v_\perp\right\|^2 \leq \sin^2(\phi_d)\|v_\perp\|^2$, *where* ϕ_d *denotes the largest principal angle between* $R(\bar{U})$ *and* $R(U)$.

Proof. According to the definition of $v_\perp$ and Lemma 2.8.1, we have

$$\begin{aligned}\left\|\bar{U}^T y\right\|^2 = \left\|\bar{U}^T\left(\mathbb{I} - UU^T\right)\bar{U}s\right\|^2 &= s^T\bar{Y}\Sigma^4\bar{Y}^4 s \\ &\overset{\vartheta_3}{\leq} \sin^2\phi_d s^T\bar{Y}\Sigma^2\bar{Y}^T s = \sin^2\phi_d\|v_\perp\|^2\end{aligned}$$

here $\bar{Y}$ and Σ are the same as those defined in Lemma 2.8.1, and the last equality holds since $\|v_\perp\|^2 = \|s\|^2 - v^T UU^T v = s^T\bar{Y}\Sigma^2\bar{Y}^T s$. □

Theorem 2.8.1. *(McDiarmid's Inequality [94]). Let* $X_1, \ldots, X_n$ *be independent random variables, and assume* f *is a function for which there exist* t_i, $i = 1, \ldots, n$ *satisfying*

$$\sup_{x_1,\ldots,x_n,\widehat{x}_i} |f(x_1, \ldots, x_n) - f(x_1, \ldots, \widehat{x}_i, \ldots, x_n)| \leq t_i$$

where $\widehat{x}_i$ *indicates replacing the sample value* x_i *with any other of its possible values. Call* $f(X_1, \ldots, X_n) := Y$. *Then for any* $\epsilon > 0$,

$$\begin{aligned}\mathbb{P}\left[Y \geq \mathbb{E}Y + \epsilon\right] &\leq \exp\left(-\frac{2\epsilon^2}{\sum_{i=1}^n t_i^2}\right) \\ \mathbb{P}\left[Y \leq \mathbb{E}Y - \epsilon\right] &\leq \exp\left(-\frac{2\epsilon^2}{\sum_{i=1}^n t_i^2}\right)\end{aligned}$$

Proof of Lemma 2.8.14. We use McDiarmid's inequality to prove this. For the simplicity of notation denote $v_\perp$ as y. Let $X_i = \bar{U}_{\Omega(i)} y_{\Omega(i)} \in \mathbb{R}^d$, and $f(X_1, \ldots, X_m) = \|\sum_{i=1}^m X_i\|_2 = \left\|\bar{U}_\Omega^T v_{\perp,\Omega}\right\|_2$, then $|f(x_1, \ldots, x_n) - f(x_1, \ldots, \widehat{x}_i, \ldots x_n|$ can be bounded

via

$$\left|\left\|\sum_{i=1}^{m} X_i\right\|_2 - \left\|\sum_{i\neq k}^{m} X_i + \widehat{X}_k\right\|_2\right| \leq \left\|X_k - \widehat{X}_k\right\|_2 \leq \|X_k\|_2 + \|\widehat{X}_k\|_2$$

$$\leq 2\|y\|_\infty \sqrt{d\mu_0/n} \tag{2.68}$$

We next calculate $\mathbb{E}\left[f(X_1,\ldots,X_m)\right] = \mathbb{E}\left[\|\sum_{i=1}^m X_i\|_2\right]$. Note that

$$\mathbb{E}\left[\left\|\sum_{i=1}^{m} X_i\right\|^2\right] = \mathbb{E}\left[\sum_{i=1}^{m}\|X_i\|^2 + \sum_{i=1}^{m}\sum_{j\neq i} X_i^T X_j\right] \tag{2.69}$$

Recall that we assume the samples are taken uniformly with replacement. This together with the fact that $\left\|\bar{U}_i\right\|_2 = \|\mathcal{P}_{R(\bar{U})}(e_i)\| \leq \sqrt{d\mu_0/n}$ yield the following

$$\begin{aligned}\mathbb{E}\left[\sum_{i=1}^{m}\|X_i\|^2\right] &= \sum_{i=1}^{m}\mathbb{E}\left[\left\|U_{\Omega(i)} y_{\Omega_{(i)}}\right\|^2\right] \\ &= \sum_{i=1}^{m}\sum_{k=1}^{n}\|\bar{U}_k\|^2 y_k^2 \mathbb{P}_{\{\Omega(i)=k\}} \leq \frac{m}{n}\frac{d\mu_0}{n}\|y\|^2 \end{aligned} \tag{2.70}$$

$$\begin{aligned}\mathbb{E}\left[\sum_{i=1}^{m}\sum_{j\neq i} X_i^T X_j\right] &= \sum_{i=1}^{m}\sum_{j\neq i}\sum_{k_1=1}^{n}\sum_{k_2=1}^{n} y_{k_1}\bar{U}_{k_1}^T \bar{U}_{k_2} y_{k_2}\mathbb{P}(\Omega_j = k_2)\mathbb{P}(\Omega_i = k_1) \\ &= \frac{m^2-m}{n^2}\|\bar{U}^T y\|^2 \leq \frac{m^2}{n^2}\sin^2\phi_d\|y\|^2 \end{aligned} \tag{2.71}$$

where the last inequality holds by Lemma 2.8.16.

Eqs (2.69) (2.70) and (2.71) together with the Jensen's inequality imply

$$\mathbb{E}\left[\left\|\sum_{i=1}^{m} X_i\right\|\right] \leq \sqrt{\frac{m}{n}}\sqrt{\frac{m}{n}\sin^2\phi_d + \frac{d\mu_0}{n}}\|y\| = \sqrt{\frac{m}{n}\frac{d\mu_0}{n}}\sqrt{\frac{m\sin^2\phi_d}{d\mu_0} + 1}\|y\| \tag{2.72}$$

Let $\epsilon = \beta_2\sqrt{\frac{m}{n}\frac{d\mu_0}{n}}\sqrt{\frac{m\sin^2\phi_d}{d\mu_0}+1}\|y\|$, then (2.68) and (2.72) together with Theorem

2.8.1 give

$$\begin{aligned}
&\mathbb{P}\left[\|U_\Omega y_\Omega\| \geq (1+\beta_2)\sqrt{\frac{m}{n}\frac{d\mu_0}{n}}\sqrt{\frac{m\sin^2\phi_d}{d\mu_0}+1}\|y\|\right] \\
&\qquad \leq \exp\left(\frac{-2\beta_2^2\frac{m}{n}\frac{d\mu_0}{n}\left(\frac{m\sin^2\phi_d}{d\mu_0}+1\right)\|y\|^2}{4m\|y\|_\infty^2\frac{d\mu_0}{n}}\right) \\
&\qquad = \exp\left(\frac{-\beta_2^2\left(\frac{m\sin^2\phi_d}{d\mu_0}+1\right)\|y\|^2}{2n\|y\|_\infty^2}\right) = \delta \qquad (2.73)
\end{aligned}$$

where the last inequality follows by submitting our definition of $\mu(y)$ Eq (2.23) and β_2. □

CHAPTER III

Learning to Share: Simultaneous Parameter Tying and Sparsification in Deep Learning

3.1 Introduction

Deep neural networks (DNNs) have recently revolutionized machine learning by dramatically advancing the state-of-the-art in several applications, ranging from speech and image recognition to playing video games [61]. A typical DNN consists of a sequence of concatenated layers, potentially involving millions or billions of parameters; by using very large training sets, DNNs are able to learn extremely complex non-linear mappings, features, and dependencies.

A large amount of research has focused on the use of regularization in DNN learning [61], as a means of reducing the generalization error. It has been shown that the parametrization of many DNNs is very redundant, with a large fraction of the parameters being predictable from the remaining ones, with no accuracy loss [47]. Several regularization methods have been proposed to tackle the potential over-fitting due to this redundancy. Arguably, the earliest and simplest choice is the classical ℓ_2 norm, known as *weight decay* in the early neural networks literature [111], and as *ridge regression* in statistics. In the past two decades, sparsity-inducing regularization based on the ℓ_1 norm (often known as Lasso) [124], and variants thereof, became standard tools in statistics and machine learning, including in deep learning [61].

Recently, [114] used group-Lasso (a variant of Lasso that assumes that parameters are organized in groups and encourages sparsity at the group level [139]) in deep learning. One of the effects of Lasso or group-Lasso regularization in learning a DNN is that many of the parameters may become exactly zero, thus reducing the amount of memory needed to store the model, and lowering the computational cost of applying it.

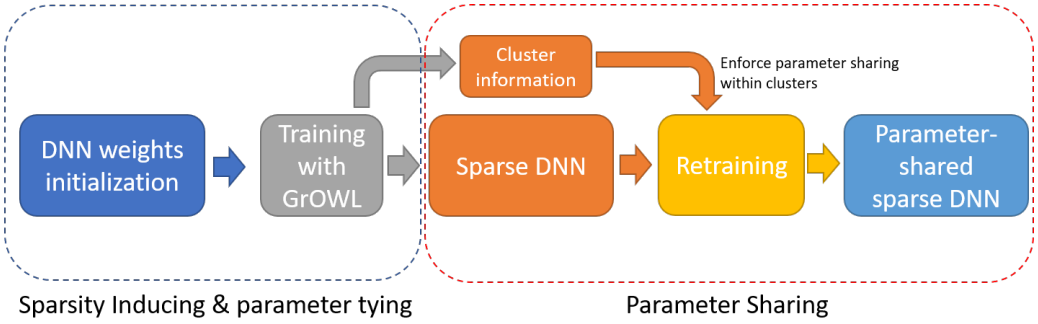


Figure 3.1: A DNN is first trained with GrOWL regularization to simultaneously identify the sparse but significant connectivities and the correlated cluster information of the selected features. We then retrain the neural network only in terms of the selected connectivities while enforcing parameter sharing within each cluster.

It has been pointed out by several authors that a major drawback of Lasso (or group-Lasso) regularization is that in the presence of groups of highly correlated covariates/features, it tends to select only one or an arbitrary convex combination of features from each group [27, 30, 56, 104, 147]. Moreover, the learning process tends to be unstable, in the sense that subsets of parameters that end up being selected may change dramatically with minor changes in the data or algorithmic procedure. In DNNs, it is almost unavoidable to encounter correlated features, not only due to the high dimensionality of the input to each layer, but also because neurons tend to co-adapt, yielding strongly correlated features that are passed as input to the subsequent layer [119].

In this work, we propose using, as a regularizer for learning DNNs, the group version

of the *ordered weighted* ℓ_1 (OWL) norm [56], termed group-OWL (GrOWL), which was recently proposed by [104]. In a linear regression context, GrOWL regularization has been shown to avoid the above mentioned deficiency of group-Lasso regularization. In addition to being a sparsity-inducing regularizer, GrOWL is able to explicitly identify groups of correlated features and set the corresponding parameters/weights to be very close or exactly equal to each other, thus taking advantage of correlated features, rather than being negatively affected by them. In deep learning parlance, this corresponds to adaptive *parameter sharing/tying*, where instead of having to define *a priori* which sets of parameters are forced to share a common value, these sets are learned during the training process. We exploit this ability of GrOWL regularization to encourage parameter sparsity and group-clustering in a two-stage procedure depicted in Fig. 3.1: we first use GrOWL to identify the significant parameters/weights of the network and, simultaneously, the correlated cluster information of the selected features; then, we retrain the network only in terms of the selected features, while enforcing the weights within the same cluster to share a common value.

The experiments reported below confirm that using GrOWL regularization in learning DNNs encourages sparsity and also yields parameter sharing, by forcing groups of weights to share a common absolute value. We test the proposed approach on two benchmark datasets, MNIST and CIFAR-10, comparing it with weight decay and group-Lasso regularization, and exploring the accuracy-memory trade-off. Our results indicate that GrOWL is able to reduce the number of free parameters in the network without degrading the accuracy, as compared to other approaches.

3.2 Related Work

In order to relieve the burden on both required memory and data for training and storing DNNs, a substantial amount of work has focused on reducing the number of free parameters to be estimated, namely by enforcing weight sharing. The classical instance of sharing is found in the convolutional layers of DNNs [61]. In fact, weight-sharing as a simplifying technique for NNs can be traced back to more than 30 years ago [85, 112].

Recently, there has been a surge of interest in compressing the description of DNNs, with the aim of reducing their storage and communication costs. Various methods have been proposed to approximate or quantize the learned weights after the training process. [48] have shown that, in some cases, it is possible to replace the original weight matrix with a low-rank approximation. Alternatively, [5] propose retraining the network layer by layer, keeping the layer inputs and outputs close to the originally trained model, while seeking a sparse transform matrix, whereas [60] propose using vector quantization to compress the parameters of DNNs.

Network pruning is another relevant line of work. In early work, [86] and [66] use the information provided by the Hessian of the loss function to remove less important weights; however, this requires expensive computation of second order derivatives. Recently, [64] reduce the number of parameters by up to an order of magnitude by alternating between learning the parameters and removing those below a certain threshold. [89] propose to prune filters, which seeks sparsity with respect to neurons, rather than connections; that approach relieves the burden on requiring sparse libraries or special hardware to deploy the network. All those methods either require multiple training/retraining iterations or a careful choice of thresholds.

There is a large body of work on sparsity-inducing regularization in deep learning. For example, [42] exploit ℓ_1 and ℓ_0 regularization to encourage weight sparsity; however, the sparsity level achieved is typically modest, making that approach not competitive for DNN compression. Group-Lasso has also been used in training DNNs; it allows seeking sparsity in terms of neurons [114, 8, 146, 100] or other structures, *e.g.*, filters, channels, filter shapes, and layer depth [134]. However, as mentioned above, both Lasso and group-Lasso can fail in the presence of strongly correlated features (as illustrated in Section 3.4, with both synthetic data and real data.

A recent stream of work has focused on using further parameter sharing in convolutional DNNs. By tying weights in an appropriate way, [50] obtain a convolutional DNN with rotation invariance. On the task of analyzing positions in the game *Go*, [40] showed improved performance by constraining features to be invariant to reflections along the x-axis, y-axis, and diagonal-axis. Finally, [36] used a hash function to randomly group the weights such that those in a hash bucket share the same value. In contrast, with GrOWL regularization, we aim to learn weight sharing from the data itself, rather than specifying it *a priori*.

Dropout-type methods have been proposed to fight over-fitting and are very popular, arguably due to their simplicity of implementation [119]. Dropout has been shown to effectively reduce over-fitting and prevent different neurons from co-adapting. Decorrelation is another popular technique in deep learning pipelines [22, 41, 109]; unlike sparsity-inducing regularizers, these methods try to make full use of the model's capacity by decorrelating the neurons. Although dropout and decorrelation can reduce over-fitting, they do not compress the network, hence do not address the issue of high memory cost. It should also be mentioned that our proposal can be seen as complementary to dropout and decorrelation: whereas dropout and decorrelation

can reduce co-adaption of nodes during training, GrOWL regularization copes with co-adaptation by tying together the weights associated to co-adapted nodes.

3.3 Group-OWL Regularization for Deep Learning

3.3.1 The Group-OWL Norm

We start by recalling the definition of the group-OWL (GrOWL) regularizer and very briefly reviewing some of its relevant properties [104].

Definition 3. *Given a matrix $W \in \mathbb{R}^{n\times m}$, let $w_{[i]\cdot}$ denote the row of W with the i-th largest ℓ_2 norm. Let $\lambda \in \mathbb{R}^n_+$, with $0 < \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n \geq 0$. The GrOWL regularizer (which is a norm) $\Omega_\lambda : \mathbb{R}^{n\times m} \to \mathbb{R}$ is defined as*

$$\Omega_\lambda(W) = \sum_{i=1}^{n} \lambda_i \left\| w_{[i]\cdot} \right\| \tag{3.1}$$

This is a group version of the OWL regularizer [56], also known as WSL1 (*weighted sorted ℓ1* [142]) and SLOPE [26], where the groups are the rows of its matrix argument. It is clear that GrOWL includes group-Lasso as a special case when $\lambda_1 = \lambda_n$. As a regularizer for multiple/multi-task linear regression, each row of W contains the regression coefficients of a given feature, for the m tasks. It has been shown that by adding the GrOWL regularizer to a standard squared-error loss function, the resulting estimate of W has the following property: rows associated with highly correlated covariates are very close or even exactly equal to each other [104]. In the linear case, GrOWL encourages correlated features to form predictive clusters corresponding to the groups of rows that are nearly or exactly equal. The rationale underlying this chapter is that when used as a regularizer for DNN learning, GrOWL will induce both sparsity and parameters tying, as illustrated in Fig. 3.2 and explained below in detail.

3.3.2 Layer-Wise GrOWL Regularization For Feedforward Neural Networks

A typical feed-forward DNN with L layers can be treated as a function f of the following form:

$$\begin{aligned} f(x,\theta) \equiv h_L &= f_L\left(h_{L-1}W_L + b_L\right) \\ h_{L-1} &= f_{L-1}\left(h_{L-2}W_{L-1} + b_{L-1}\right) \\ &\vdots \\ h_1 &= f_1\left(xW_1 + b_1\right) \end{aligned}$$

$\theta = \big(W_1, b_1, \ldots, W_L, b_L\big)$ denotes the set of parameters of the network, and each f_i is a component-wise nonlinear activation function, with the *rectified linear unit* (ReLU), the *sigmoid*, and the *hyperbolic tangent* being common choices for this function [61].

Given labelled data $\mathcal{D} = \big((x^{(1)}, y^{(1)}), ..., (x^{(m)}, y^{(m)})\big)$, DNN learning may be formalized as an optimization problem,

$$\min_{\theta} \mathcal{L}(\theta) + \mathcal{R}(\theta), \quad \text{with } \mathcal{L}(\theta) = \sum_{i=1}^{m} L\Big(y^{(i)}, f\big(x^{(i)}, \theta\big)\Big) \,, \tag{3.2}$$

where $L\big(y, \hat{y}\big)$ is the loss incurred when the DNN predicts $\hat{y}$ for y, and $\mathcal{R}$ is a regularizer. Here, we adopt as regularizer a sum of GrOWL penalties, each for each layer of the neural network, *i.e.,*

$$\mathcal{R}(\theta) = \sum_{l=1}^{L} \Omega_{\lambda^{(l)}}\big(W_l\big), \quad \lambda^{(l)} \in \mathbb{R}_+^{N_{l-1}} \,, \tag{3.3}$$

where N_l denotes the number of neurons in the l-th layer and $0 < \lambda_1^{(l)} \geq \lambda_2^{(l)} \geq \cdots \geq \lambda_{N_{l-1}}^{(l)} \geq 0$. Since $\mathcal{R}(\theta)$ does not depend on $b_1, ..., b_L$, the biases are not regularized, as is common practice.

As indicated in Eq. equation 3.3, the number of groups in each GrOWL regularizer is the number of neurons in the previous layer, *i.e.,* $\lambda^{(l)} \in \mathbb{R}^{N_{l-1}}$. In other words, *we*

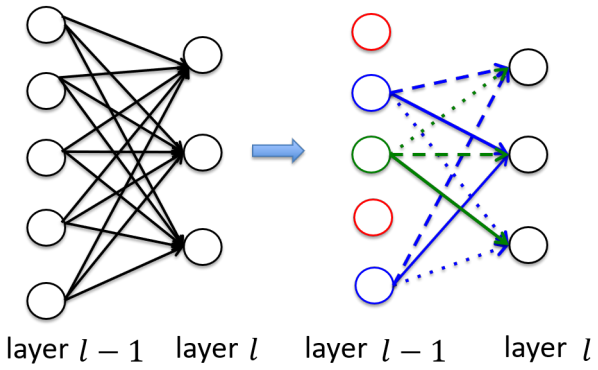
treat the weights associated with each input feature as a group. For fully connected layers, where $W_l \in \mathbb{R}^{N_{l-1} \times N_l}$, each group is a row of the weight matrix. In convolutional layers, where $W_l \in \mathbb{R}^{F_w \times F_h \times N_{l-1} \times N_l}$, with F_w and F_h denoting the width and height, respectively, of each filter, we first reshape W_l to a 2-dimensional array, *i.e.,* $W_l \rightarrow W_l^{2D}$, where $W_l^{2D} \in \mathbb{R}^{N_{l-1} \times (F_w F_h N_l)}$, and then apply GrOWL on the reshaped matrix. That is, if the l-th layer is convolutional, then

$$\mathcal{R}(W_l) = \Omega_{\lambda^{(l)}}\left(W_l^{2D}\right). \tag{3.4}$$

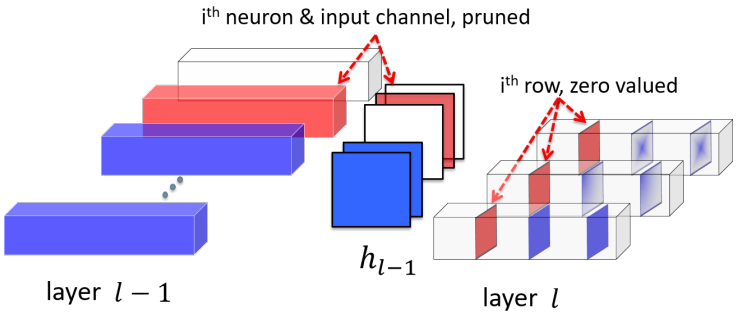
Each row of W_l^{2D} represents the operation on an input channel. The rationale to apply the GrOWL regularizer to each row of the reshaped weight matrix is that GrOWL can select the relevant features of the network, while encouraging the coefficient rows of each layer associated with strongly correlated features from the previous layer to be nearly or exactly equal, as depicted in Fig. 3.2. The goal is to significantly reduce the complexity by: **(i)** pruning unimportant neurons of the previous layer that correspond to zero rows of the (reshaped) weight matrix of the current layer; **(ii)** grouping the rows associated with highly correlated features of the previous layer, thus encouraging the coefficient rows in each of these groups to be very close to each other. As a consequence, in the retraining process, we can further compress the neural network by enforcing the parameters *within each neuron* that belong to the same cluster to share same values.

In the work of [8], each group is predefined as the set of parameters associated to a neuron, and group-Lasso regularization is applied to seek group sparsity, which corresponds to zeroing out redundant neurons of each layer. In contrast, we treat the filters corresponding to the same input channel as a group, and GrOWL is applied to prune the redundant groups and thus *remove the associated unimportant neurons of the previous layer*, while grouping associated parameters of the current layer that

correspond with highly correlated input features to different clusters. Moreover, as will be shown in Section 3.4, group-Lasso can fail at selecting all relevant features of previous layers, and for the selected ones the corresponding coefficient groups are quite dissimilar from each other, making it impossible to further compress the DNN by enforcing parameter tying.



(a) GrOWL regularization effect on fully connected layers.



(b) GrOWL regularization effect on convolutional layers.

Figure 3.2: GrOWL's regularization effect on DNNs. (a) Fully connected layers: for layer l, GrOWL clusters the input features from the previous layer, $l-1$, into different groups, *e.g.,* blue and green. Within each neuron of layer l, the weights associated with the input features from the same cluster (input arrows marked with the same color) share the same parameter value. The neurons in layer $l-1$ corresponding to zero-valued rows of W_l have zero input to layer l, hence get removed automatically. (b) Convolutional layers: each group (row) is predefined as the filters associated with the same input channel; parameter sharing is enforced among the filters within each neuron that corresponds with the same cluster (marked as blue with different effects) of input channels.

3.3.3 Proximal Gradient Algorithm

To solve (3.2), we use a *proximal gradient algorithm* [20], which has the following general form: at the t-th iteration, the parameter estimates are updated according to

$$\theta^{(t+1)} = \text{prox}_{\eta\mathcal{R}}\left(\theta^{(t)} - \eta\nabla_\theta\mathcal{L}(\theta^{(t)})\right), \tag{3.5}$$

where, for some convex function Q, prox_Q denotes its *proximity operator* (or simply "prox") [20], defined as $\text{prox}_Q(\xi) = \arg\min_\nu Q(\nu) + \frac{1}{2}\|\nu - \xi\|_2^2$. In Eq. equation 3.5, $\|\nu - \xi\|_2^2$ denotes the sum of the squares of the differences between the corresponding components of ν and ξ, regardless of their organization (here, a collection of matrices and vectors).

Since $\mathcal{R}(\theta)$, as defined in equation 3.3, is separable across the weight matrices of different layers and zero for $b_1, ..., b_L$, the corresponding prox is also separable, thus

$$W_l^{(t+1)} = \text{prox}_{\eta\Omega_\lambda^{(l)}}\left(W_l^{(t)} - \eta\,\nabla_{W_l}\mathcal{L}(\theta^{(t)})\right), \quad \text{for } l = 1, \ldots, L \tag{3.6}$$

$$b_l^{(t+1)} = b_l^{(t)} - \eta\,\nabla_{b_l}\mathcal{L}(\theta^{(t)}) \quad \text{for } \; l = 1, ..., L. \tag{3.7}$$

It was shown by [104] that the prox of GrOWL can be computed as follows. For some matrix $V \in \mathbb{R}^{N\times M}$, let $U = \text{prox}_{\Omega_\lambda}(V)$, and v_i and u_i denote the corresponding i-th rows. Then,

$$u_i = v_i\,(\text{prox}_{\Omega_\lambda}(\tilde{v}))_i/\|v_i\|, \tag{3.8}$$

where $\tilde{v} = [\|v_1\|, \|v_2\|, \cdots, \|v_N\|]$. For vectors in $\mathbb{R}^N$ (in which case GrOWL coincides with OWL), $\text{prox}_{\Omega_{\lambda^{(l)}}}$ can be computed with $O(n\log n)$ cost, where the core computation is the so-called *pool adjacent violators algorithm* (PAVA [45]) for isotonic regression. We provide one of the existing algorithms in Appendix 3.6.1; for details, the reader is referred to the work of [26] and [141]. in this chapter, we apply the

proximal gradient algorithm per epoch, which generally performs better. The training method is summarized in Algorithm 2.

Algorithm 2

Input: parameters of the OWL regularizers $\lambda^{(l)}, ..., \lambda^{(L)}$, learning rate η
for each epoch T **do**
  for each iteration t in epoch T **do**
    Update the parameters $\theta = \left(W_1, b_1, \ldots, W_L, b_L\right)$ via backpropagation (BP)
  end for
  Apply proximity operator via equation 3.6
end for

3.3.4 Implementation Details

Setting the GrOWL Weights GrOWL is a family of regularizers, with different variants obtained by choosing different weight sequences $\lambda_1, \ldots, \lambda_n$. in this chapter, we propose the following choice:

$$\lambda_i = \begin{cases} \Lambda_1 + (p - i + 1)\Lambda_2, & \text{for } \; i = 1, ..., p, \\ \Lambda_1, & \text{for } \; i = p + 1, ..., n, \end{cases} \tag{3.9}$$

where $p \in \{1, ...n\}$ is a parameter. The first p weights follow a linear decay, while the remaining ones are all equal to Λ_1. Notice that, if $p = n$, the above setting is equivalent to OSCAR [27]. Roughly speaking, Λ_1 controls the sparsifying strength of the regularizer, while Λ_2 controls the clustering property (correlation identification ability) of GrOWL [104]. Moreover, by setting the weights to a common constant beyond index p means that clustering is only encouraged among the p largest coefficients, *i.e.,* only among relevant coefficient groups.

Finding adequate choices for p, Λ_1, and Λ_2 is crucial for jointly selecting the relevant features and identifying the underlying correlations. In practice, we find that with properly chosen p, GrOWL is able to find more correlations than OSCAR. We explore different choices of p in Section 3.4.1.

Parameter Tying After the initial training phase, at each layer l, rows of W_l that corresponds to highly correlated outputs of layer $l-1$ have been made similar or even exactly equal. To further compress the DNN, we force rows that are close to each other to be identical. We first group the rows into different clusters [1] according to the *pairwise similarity* metric

$$\mathcal{S}_l(i,j) = \frac{W_{l,i}^T W_{l,j}}{\max\left(\|W_{l,i}\|_2^2, \|W_{l,j}\|_2^2\right)} \in [-1,1], \tag{3.10}$$

where $W_{l,i}$ and $W_{l,j}$ denote the i-th and j-th rows of W_l, respectively.

With the cluster information obtained by using GrOWL, we enforce parameter sharing for the rows that belong to a same cluster by replacing their values with the averages (centroid) of the rows in that cluster. In the subsequent retraining process , let $\mathcal{G}_k^{(l)}$ denote the k-th cluster of the l-th layer, then centroid $g_k^{(l)}$ of this cluster is updated via

$$\frac{\partial \mathcal{L}}{\partial g_k^{(l)}} = \frac{1}{\left|\mathcal{G}_k^{(l)}\right|} \sum_{W_{l,i} \in \mathcal{G}_k^{(l)}} \frac{\partial \mathcal{L}}{\partial W_{l,i}}. \tag{3.11}$$

3.4 Numerical Results

We assess the performance of the proposed method on two benchmark datasets: MNIST and CIFAR-10. We consider two different networks and compare GrOWL with group-Lasso and weight decay, in terms of the compression vs accuracy trade-off. For fair comparison, the training-retraining pipeline is used with the different regularizers. After the initial training phase, the rows that are close to each other are clustered together and forced to share common values in the retraining phase. We implement all models using Tensorflow [1]. We evaluate the effect of the different

[1] in this chapter, we use the built-in *affinity propagation* method of the scikit-learn package [31]. A brief description of the algorithm is provided in Appendix 3.6.2.

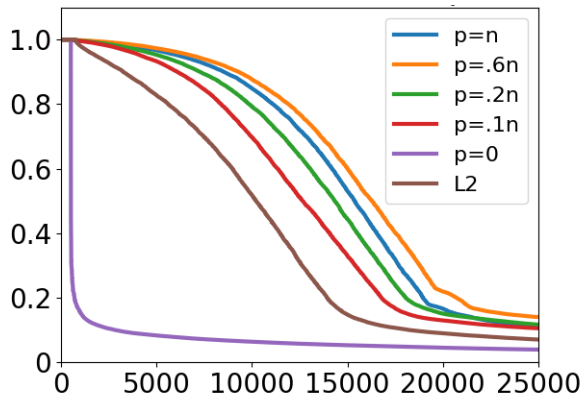


Figure 3.3: Regularization effect of GrOWL for different p values (Eq (3.9)). In this plot, the y-axis indicates the sorted values of $\mathcal{S}$ in Equation 3.10.

regularizers using the following quantities:

$$\text{sparsity} = (\#\text{zero params})/(\#\text{ total params}) \ ,$$

$$\text{compression rate} = (\#\text{ total params})/(\#\text{ unique params}) \ ,$$

$$\text{parameter sharing} = (\#\text{ nonzero params})/(\#\text{ unique params}) \ .$$

3.4.1 Different Choices of GrOWL Parameters

First, we consider a synthetic data matrix X with block-diagonal covariance matrix Σ, where each block corresponds to a cluster of correlated features, and there is a gap g between two blocks. Within each cluster, the covariance between two features X_i and X_j is $\text{cov}(X_i, X_j) = 0.96^{|i-j|}$, while features from different clusters are generated independently of each other. We set $n = 784, K = 10$, block size 50, and gap $g = 28$. We generate 10000 training and 1000 testing examples.

We train a NN with a single fully-connected layer of 300 hidden units. Fig 3.3 shows the first 25000 entries of the sorted *pairwise similarity* matrices (Eq 3.10) obtained by applying GrOWL with different p (Eq 3.9) values. By setting the weights beyond index p to a common constant implies that clustering is only encouraged among the p largest coefficients, *i.e.*, relevant coefficient groups; however, Fig. 3.3 shows that, with

properly chosen p, GrOWL yields more parameter tying than OSCAR ($p = n$). On the other hand, smaller p values allow using large Λ_2, encouraging parameter tying among relatively loose correlations. In practice, we find that for p around the target fraction of nonzero parameters leads to good performance in general. The intuition is that we only need to identify correlations among the selected important features.

Fig. 3.3 shows that weight decay (denoted as ℓ_2) also pushes parameters together, though the parameter-tying effect is not as clear as that of GrOWL. As has been observed in the literature [27], weight decay often achieves better generalization than sparsity-inducing regularizers. It achieves this via parameter shrinkage, especially in the highly correlated region, but it does not yield sparse models. In the following section, we explore the compression performance of GrOWL by comparing it with both group-Lasso and weight decay. We also explore how to further improve the accuracy vs compression trade-off by using sparsity-inducing regularization together with weight decay (ℓ_2). For each case, the baseline performance is provided as the best performance obtained by running the original neural network (without compression) after sweeping the hyper-parameter on the weight decay regularizer over a range of values.

3.4.2 Fully Connected Neural Network on MNIST

The MNIST dataset contains centered images of handwritten digits (0–9), of size 28×28 (784) pixels. Fig 3.4 (a) shows the (784×784) correlation matrix of the dataset (the margins are zero due to the redundant background of the images). We use a network with a single fully connected layer of 300 hidden units. The network is trained for 300 epochs and then retrained for an additional 100 epochs, both with momentum. The initial learning rate is set to 0.001, for both training and retraining, and is reduced by a factor of 0.96 every 10 epochs. We set $p = 0.5$, and Λ_1, Λ_2 are

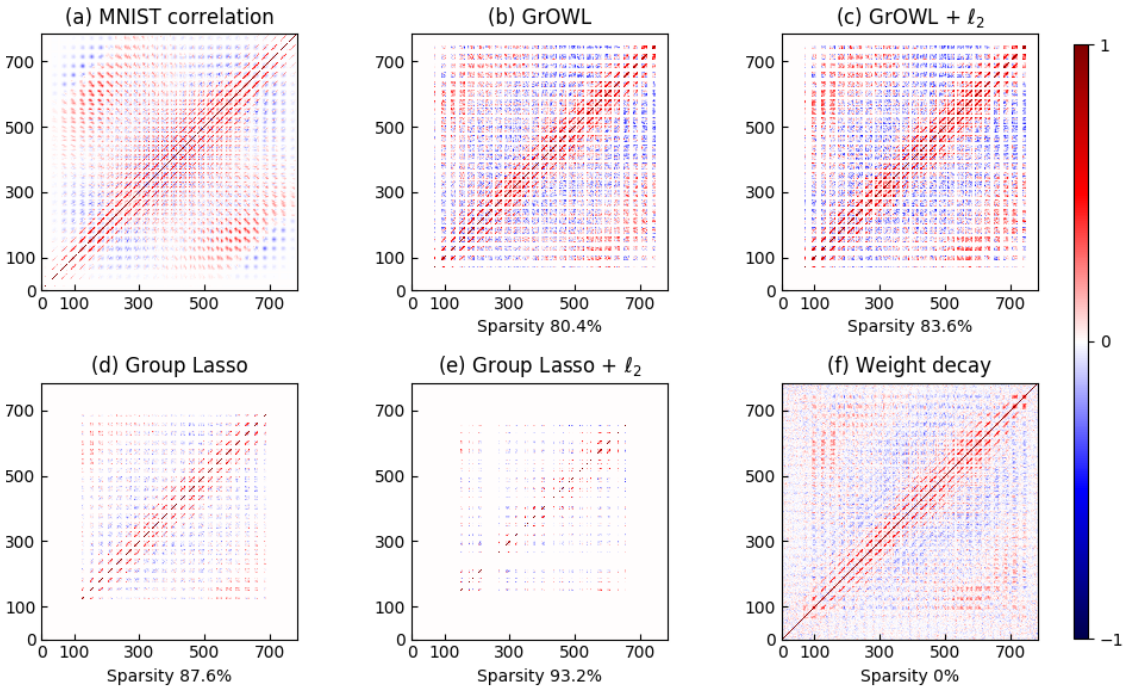


Figure 3.4: MNIST: comparison of the data correlation and the pairwise similarity maps (Eq (3.10)) of the parameter rows obtained by training the neural network with GrOWL, GrOWL+ℓ_2, group-Lasso, group-Lasso+ℓ_2 and weight decay (ℓ_2).

selected by grid search.

Pairwise similarities (see Eq. (3.10)) between the rows of the weight matrices learned with different regularizers are shown in Fig. 3.4 (b–f). As we can see, GrOWL ($+\ell_2$) identifies more correlations than group-Lasso ($+\ell_2$), and the similarity patterns in Fig. 3.4 (b, c) are very close to that of the data (Fig. 3.4(a)). On the other hand, weight decay also identifies correlations between parameter rows, but it does not induce sparsity. Moreover, as shown in Table 3.1, GrOWL yields a higher level of parameter sharing than weight decay, matching what we observed on synthetic data in Section 3.4.1.

The compression vs accuracy trade-off of the different regularizers is summarized in Table 3.1, where we see that applying ℓ_2 regularization together with group-Lasso or GrOWL leads to a higher compression ratio, with negligible effect on the accuracy.

Table 3.1: Sparsity, parameter sharing, and compression rate results on MNIST. Baseline model is trained with weight decay and we do not enforce parameter sharing for baseline model. We train each model for 5 times and report the average values together with their standard deviations.

Regularizer	**Sparsity**	**Parameter Sharing**	**Compression ratio**	**Accuracy**
none	$0.0 \pm 0\%$	1.0 ± 0	1.0 ± 0	$98.3 \pm 0.1\%$
weight decay	$0.0 \pm 0\%$	1.6 ± 0	1.6 ± 0	$98.4 \pm 0.0\%$
group-Lasso	$87.6 \pm 0.1\%$	1.9 ± 0.1	15.8 ± 1.0	$98.1 \pm 0.1\%$
group-Lasso+ℓ_2	$93.2 \pm 0.4\%$	1.6 ± 0.1	23.7 ± 2.1	$98.0 \pm 0.1\%$
GrOWL	$80.4 \pm 1.0\%$	3.2 ± 0.1	16.7 ± 1.3	$98.1 \pm 0.1\%$
GrOWL+ℓ_2	$83.6 \pm 0.5\%$	3.9 ± 0.1	24.1 ± 0.8	$98.1 \pm 0.1\%$

Table 3.1 also shows that, even with lower sparsity after the initial training phase, GrOWL (+ℓ_2) compresses the network more than group-Lasso (+ℓ_2), due to the significant amount of correlation it identifies; this also implies that group-Lasso only selects a subset of the correlated features, while GrOWL selects all of them. On the other hand, group-Lasso suffers from randomly selecting a subset of correlated features; this effect is illustrated in Fig. 3.5, which plots the indices of nonzero rows, showing that GrOWL (+ℓ_2) stably selects relevant features while group-Lasso (+ℓ_2) does not. The mean ratios of changed indices[2] are 11.09%, 0.59%, 32.07%, and 0.62% for group-Lasso, GrOWL, group-Lasso+ℓ_2, and GrOWL+ℓ_2, respectively.

3.4.3 VGG-16 on CIFAR-10

To evaluate the proposed method on large DNNs, we consider a VGG-like [118] architecture proposed by [140] on the CIFAR-10 dataset. The network architecture is summarized in Appendix 3.6.4; comparing with the original VGG of [118], their fully connected layers are replaced with two much smaller ones. A batch normalization layer is added after each convolutional layer and the first fully connected layer. Unlike [140], we don't use dropout. We first train the network under different regularizers for 150 epochs, then retrain it for another 50 epochs, using the learning rate decay

[2]The mean ratio of changed indices is defined as: $\frac{1}{n}\sum_{k=1}^{n} \|I_k - \bar{I}\|_0 / \|\bar{I}\|_0$, where n is the number of experiments, I_k is the index vector of kth experiment, and $\bar{I} = \frac{1}{n}\sum_{k=1}^{n} I_k$ is the mean index vector.

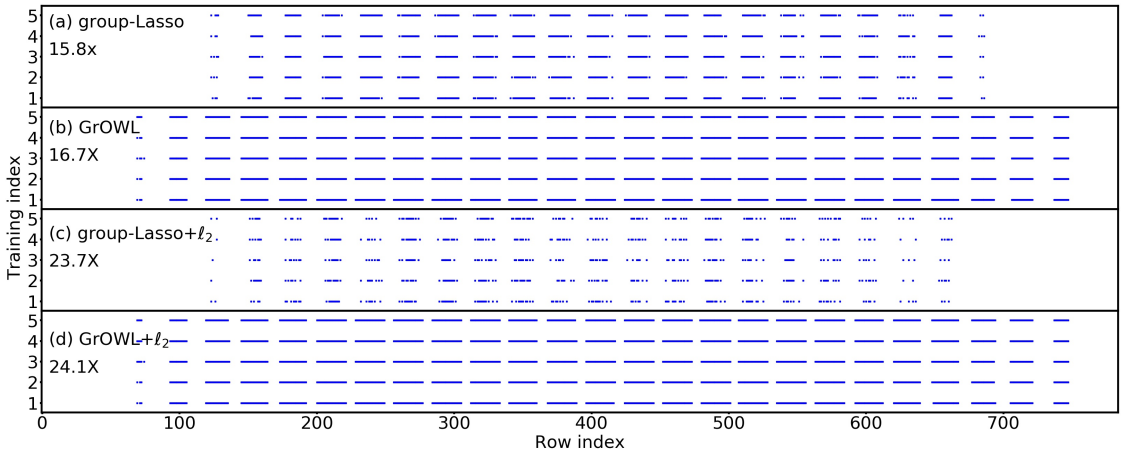


Figure 3.5: MNIST: sparsity pattern of the trained fully connected layer, for 5 training runs, using group-Lasso, GrOWL, group-Lasso+ℓ_2, GrOWL+ℓ_2.

scheme described by [70]: the initial rates for the training and retraining phases are set to 0.01 and 0.001, respectively; the learning rate is multiplied by 0.1 every 60 epochs of the training phase, and every 20 epochs of the retraining phase. For GrOWL ($+\ell_2$), we set $p = 0.1\,n$ (see Eq. equation 3.9) for all layers, where n denotes the number of rows of the (reshaped) weight matrices of each layer.

The results are summarized in Table 3.2. For all of the regularizers, we use the *affinity propagation* algorithm (with preference value[3] set to 0.8) to cluster the rows at the end of initial training process. Our experiments showed that it is hard to encourage parameter tying in the first 7 convolutional layers; this may be because the filters of these first 7 convolutional layers have comparatively large feature maps (from 32×32 to 8×8), which are only loosely correlated. We illustrate this reasoning in Fig. 3.6, showing the cosine similarity between the vectorized output channels of layers 1, 6, 10, and 11, at the end of the training phase; it can be seen that the outputs of layers 10 and 11 have many more significant similarities than that of layer 6. Although the output channels of layer 1 also have certain similarities, as seen in

[3] In Table 3.4 (Appendix 3.6.4), we explore the effect of different choices of this value.

Table 3.2: Sparsity (S1) and Parameter Sharing (S2) of VGG-16 on CIFAR-10. Layers marked by * are regularized. We report the averaged results over 5 runs.

Layers	Weight Decay (**S1**, **S2**)	group-Lasso (**S1**, **S2**)	group-Lasso + ℓ_2 (**S1**, **S2**)	GrOWL (**S1**, **S2**)	GrOWL + ℓ_2 (**S1**, **S2**)
conv1	0%, 1.0	0%, 1.0	0%, 1.0	0%,1.0	0%, 1.0
*conv2	0%, 1.0	34%, 1.0	40%, 1.0	20%, 1.0	34%, 1.0
*conv3	0%, 1.0	28%, 1.0	20%, 1.0	28%, 1.0	17%, 1.0
*conv4	0%, 1.0	34%, 1.0	29%, 1.0	30%, 1.0	27% 1.0
*conv5	0%, 1.0	12%, 1.0	11%, 1.0	8%, 1.0	14%, 1.0
*conv6	0%, 1.0	38%, 1.0	40%, 1.0	38%, 1.0	43%, 1.0
*conv7	0%, 1.0	46%, 1.0	51%, 1.0	40%, 1.0	50%, 1.0
*conv8	0%, 1.0	49%, 1.0	53%, 1.0	50%, 1.0	55%, 1.0
*conv9	0%, 1.0	78%, 1.0	78%, 1.0	74%, 1.1	75%, 1.2
*conv10	0%, 1.2	76%, 1.0	76%, 1.0	66%, 2.7	73%, 3.0
*conv11	0%, 1.2	84%, 1.0	87%, 1.0	81%, 3.7	88%, 3.7
*conv12	0%, 2.0	85%, 1.0	91%, 1.0	75%, 2.6	78%, 2.5
*conv13	0%, 2.1	75%, 1.1	90%, 1.1	78%, 1.9	71%, 4.2
*fc	0%, 4.2	78%, 1.0	91%, 1.1	69%, 2.7	81%, 2.2
softmax	0%, 1.0	0%,1.0	0%, 1.0	0%, 1.0	0%, 1.0
Compression	1.3 ± 0.1X	11.1 ± 0.5X	14.5 ± 0.5X	11.4 ± 0.5X	14.5 ± 0.5X
Accuracy	93.1 ± 0.0%	92.1 ± 0.2%	92.7 ± 0.1%	92.2 ± 0.1%	92.7 ± 0.1%
Baseline	Accuracy: 93.4 ± 0.2%, Compression: 1.0X				

Table 3.2, neither GrOWL ($+\ell_2$) nor weight decay tends to tie the associated weights. This may mean that the network is maintaining the diversity of the inputs in the first few convolutional layers.

Although GrOWL and weight decay both encourage parameter tying in layers 9-13, weight decay does it with less intensity and does not yield a sparse model, thus it cannot significantly compress the network. [89] propose to prune small weights after the initial training phase with weight decay, then retrain the reduced network; however, this type of method only achieves compression[4] ratios around 3. As mentioned by [89], layers 3-7 can be very sensitive to pruning; however, both GrOWL ($+\ell_2$) and group-Lasso ($+\ell_2$) effectively compress them, with minor accuracy loss.

On the other hand, similar to what we observed by running the simple fully-connected network on MNIST, the accuracy-memory trade-off improves significantly by applying GrOWL or group-Lasso together with ℓ_2. However, Table 3.2 also shows

[4]Although parameter sharing is not considered by [89], according to Table 3.2, pruning following weight decay together with parameter sharing still cannot compress the network as much as GrOWL does.

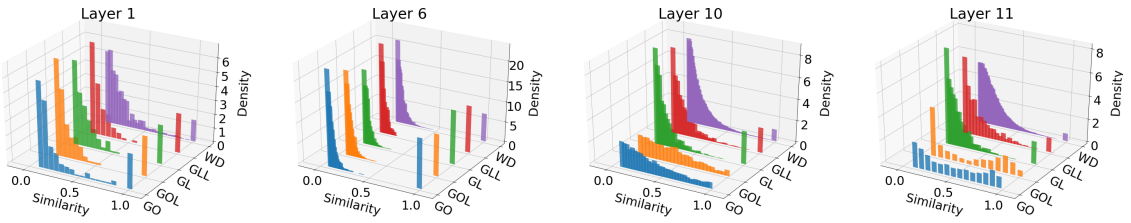


Figure 3.6: Output channel cosine similarity histogram obtained with different regularizers. Labels: GO:GrOWL, GOL:GrOWL+ℓ_2, GL:group-Lasso, GLL:group-Lasso+ℓ_2, WD:weight decay.

that the trade-off achieved by GrOWL (+ℓ_2) and group-Lasso (+ℓ_2) are almost the same. We suspect that this is caused by the fact that CIFAR-10 is simple enough that one could still expect a good performance after strong network compression. We believe this gap in the compression vs accuracy trade-off can be further increased in larger networks on more complex datasets. We leave this question for future research.

3.5 Conclusion

We have proposed using the recent GrOWL regularizer for simultaneous parameter sparsity and tying in DNN learning. By leveraging on GrOWL's capability of simultaneously pruning redundant parameters and tying parameters associated with highly correlated features, we achieve significant reduction of model complexity, with a slight or even no loss in generalization accuracy. We evaluate the proposed method on both a fully connected neural network and a deep convolutional neural network. The results show that GrOWL can compress large DNNs by factors ranging from 11.4 to 14.5, with negligible loss on accuracy.

The correlation patterns identified by GrOWL are close to those of the input features to each layer. This may be important to reveal the structure of the features, contributing to the interpretability of deep learning models. On the other hand, by automatically tying together the parameters corresponding to highly correlated

features, GrOWL alleviates the negative effect of strong correlations that might be induced by the noisy input or the co-adaption tendency of DNNs.

3.6 Supplementary Material

3.6.1 ProxGrOWL

Various methods have been proposed to compute the proximal mapping of OWL (ProxOWL). It has been proven that the computation complexity of these methods is $O(n \log n)$ which is just slightly worse than the soft thresholding method for solving ℓ_1 norm regularization. In this chapter, we use Algorithm 3 that was originally proposed in [26].

Algorithm 3 ProxGrOWL [26] for solving $\text{prox}_{\eta,\Omega_\lambda}(z)$

Input: z and λ
Let $\widetilde{\lambda} = \eta\lambda$ and $\widetilde{z} = |Pz|$ be a nonincreasing vector, where P is a permutation matrix.
while $\widetilde{z} - \widetilde{\lambda}$ is not nonincreasing: **do**
Identify strictly increasing subsequences, *i.e.,* segments $i : j$ such that

$$\widetilde{z}_i - \widetilde{\lambda}_i < \widetilde{z}_{i+1} - \widetilde{\lambda}_{i+1} < \widetilde{z}_j - \widetilde{\lambda}_j \tag{3.13}$$

Replace the values of $\widetilde{z}$ and $\widetilde{\lambda}$ over such segments by their average value: for $k \in \{i, i+1, \cdots, j\}$

$$\widetilde{z}_k \leftarrow \frac{1}{j-i+1} \sum_{i \le k \le j} \widetilde{z}_k, \qquad \widetilde{\lambda}_k \leftarrow \frac{1}{j-i+1} \sum_{i \le k \le j} \widetilde{\lambda}_k \tag{3.14}$$

end while
Output: $\widehat{z} = \text{sign}(z)\, P^T(\widetilde{z} - \widetilde{\lambda})_+$.

3.6.2 Affinity Propagation

Affinity Propagation is a clustering method based on sending messages between pairs of data samples. The idea is to use these messages to determine the most representative data samples, which are called exemplars, then create clusters using these exemplars.

Provided with the precomputed *data similarity* $s(i,j), i \neq j$ and *preference* $s(i,i)$, there are two types information being sent between samples iteratively: 1) *respon-*

sibility $r(i,k)$, which measures how likely that sample k should be the exemplar of sample i; 2) *availability* $a(k,i)$, which is the evidence that sample i should choose sample k as its exemplar. The algorithm is described in 4.

Algorithm 4 Affinity Propagation [57]

Initialization: $r(i,k) = 0, a(k,i) = 0$ for all i,k
while not converge **do**
Responsibility updates:

$$r(i,k) \leftarrow s(i,k) - \max_{j \neq k}(a(j,i) + s(i,j))$$

Availability updates:

$$a(k,k) \leftarrow \sum_{j \neq k} \max\{0, r(j,k)\}$$

$$a(k,i) \leftarrow \min\left(0, r(k,k) + \sum_{j \notin \{k,i\}} \max\{0, r(j,k)\}\right)$$

end while
Making assignments:

$$c_i^* \leftarrow \arg\max_k r(i,k) + a(k,i)$$

Unlike k-means or agglomerative algorithm, Affinity Propagation does not require the number of clusters as an input. We deem this as a desired property for enforcing parameter sharing in neural network compression because it's impossible to have the exact number of clusters as a prior information. In practice, the input *preference* of Affinity Propagation determines how likely each sample will be chosen as an exemplar and its value will influence the number of clusters created.

3.6.3 Network Architecture for Synthetic Data and MNIST

Table 3.3: Network Architecture for both MNIST and Synthetic Data

Input $\in \mathbb{R}^{784}$
FC 500 BatchNorm ReLU
FC 10 Softmax

3.6.4 VGG-16 on CIFAR-10

Table 3.4: VGG: Clustering rows over different preference values for running the *affinity propagation algorithm* (Algorithm 4). For each experiment, we report clustering accuracy (A), compression rate (C), and parameter sharing (S) of layers 9-14. For each regularizer, we use different preference values to run Algorithm 4 to cluster the rows at the end of initial training process. Then we retrain the neural network correspondingly. The results are reported as the averages over 5 training and retraining runs.

Preference Value	**0.6** (**A, C, S**)	**0.7** (**A, C, S**)	**0.8** (**A, C, S**)	**0.9** (**A, C, S**)
GrOWL	92.2%, 13.6, 3.5	92.2%, 12.5, 2.6	92.2%, 11.4, 2.1	92.2%, 10.9, 1.7
Group Lasso	92.2%, 12.1, 1.1	92.0%, 11.4, 1.1	92.1%, 11.0, 1.0	92.2%, 9.5, 1.0
GrOWL + ℓ_2	92.7%, 14.7, 2.3	92.5%, 15.4, 2.9	92.7%, 14.5, 2.3	92.6,%, 13.5, 1.8
GrLasso + ℓ_2	92.7%, 14.8, 1.2	92.7%, 14.5, 1.1	92.7%, 14.5, 1.0	92.6%, 14.3, 1.0
Weight Decay	93.2%, 1.8, 2.2	93.4%, 1.5, 1.7	93.1%, 1.3, 1.4	93.3%, 1.1, 1.1

The network architecture of VGG is summarized in Table 3.5, which is proposed by [140] on the CIFAR-10 dataset. Comparing with the original VGG of [118], their fully connected layers of this tiny VGG are replaced with two much smaller ones.

In Table 3.4, we access each regularizer over different preference values for running the *affinity propagation algorithm* (Algorithm 4). For each cell, the result is summarized over 5 training-retraining runs, one for each trial with only the preference value required by Algorithm 4 being different. As is shown in Table 3.4, both Group Lasso and GrLasso+ℓ_2 exhibit almost no parameter sharing effect over a wide range of preference values required by the clustering algorithm (Algorithm 4). This validates the fact, in existence of highly correlated features, Group Lasso tends to select one or a small random subset of all relevant features. In contrast, GrOWL is capable of selecting all of them and tying the corresponding parameters together. Although this ability of GrOWL is supposed to yield better performance and more stable learning process, we did not observe this by running VGG on CIFAR-10. We suspect this is due to the fact that we are using a complex model on a comparatively simpler dataset. We expect the superiority of GrOWL over Group Lasso would be more obvious when we move to more complex dataset, where being able to select all relevant features is crucial for guaranteeing stable learning process and better performance.

Table 3.5: Network statistics of VGG-16.

Layers	Output $w \times h$	#Channels in&out	#Params
conv1	32×32	3, 64	1.7E+03
*conv2	32×32	*64*, 64	3.7E+04
*conv3	16×16	*64*, 128	7.4E+04
*conv4	16×16	*128*, 128	1.5E+05
*conv5	8×8	*128*, 128	2.9E+05
*conv6	8×8	*128*, 256	5.9E+05
*conv7	8×8	*256*, 256	5.9E+05
*conv8	4×4	*256*, 512	1.2E+06
*conv9	4×4	*512*, 512	2.4E+06
*conv10	4×4	*512*, 512	2.4E+06
*conv11	2×2	*512*, 512	2.4E+06
*conv12	2×2	*512*, 512	2.4E+06
*conv13	2×2	*512*, 512	2.4E+06
*fc	1	*512*, 512	1.0E+06
sofrmax	1	512, 10	5.1E+03

CHAPTER IV

Regularized Information Maximization Auto-Encoding

4.1 Introduction

Recent years have witnessed great successes in deep learning, but these successes are marred by the inscrutability of models as their sizes and complexities keep growing. Learning interpretable representations so as to gain more understanding of the decision-making process has become ever more important, especially as complex models have reached many critical areas, including social science, financial services and healthcare. A large amount of research has focused on unsupervised learning of disentangled representations, which is motivated by the argument that a real intelligent agent can only be attained if it can learn to identify and separate out the underlying explanatory factors of data into disjoint parts of the learned representations [23].

Over the past decade, there has been a large body of work focusing on using autoencoders to power unsupervised representation [23, 128]. An autoencoder consists of an encoder and a decoder. Given the input data, the encoder first maps it to the representation space, then the decoder maps the representation back to the original space where the input lives. Autoencoders are typically used as a dimensionality reduction technique by restricting the latent space to a lower dimensionality than the

input space. Extensions of ordinary autoencoders have been proposed to encourage desired representation characteristics. Sparse autoencoders [101] impose sparsity constraints on the over-complete representation bases. Denoising autoencoders [131] propose to reconstruct the original clean input from its artificially corrupted version. These approaches are shown to extract useful representations with better compactness and robustness. However, it is still unclear how to further extend them to achieve more challenging goals, *e.g.,* when the goal is to simultaneously identify the underlying categories of data and disentangle the continuous representation factors.

In the parallel fashion, there has been a surge of interest in exploring the Variational AutoEncoder (VAE) [81, 107] framework for disentangled representation learning. From the optimization perspective, VAE can also be interpreted as an ordinary autoencoder regularized in a specific way, *i.e.,* by pushing the conditional probability of representation given data towards some prior that is often chosen according to our assumptions on how data is generated. As is proposed in β-VAE [71], better disentangled representations can be attained by using a larger weight on the regularization. However, by doing so, β-VAE also severely sacrifices the mutual information between data and its representations, resulting in a poor trade-off between disentanglement and reconstruction fidelity.

Many recent efforts have focused on attaining a better disentanglement and reconstruction trade-off, where the approaches generally fall into two lines. One of them revises the original objective of VAE and put larger weight on the so-called *total correlation* [133] term to promote statistical independence between representation factors [80, 58, 35, 55], while the other proposes to implicitly preserve the mutual information when increasing the regularization strength [92, 32, 7, 52, 144]. While various degrees of improvements have been achieved, these approaches still suffer from

two major shortcomings. First, they are shown to be incapable of uncovering the categorical information of the data. Although better priors may compensate for this weakness, finding such priors is itself a very challenging problem. Second, we still lack fundamental understanding or principles for effectively retaining the informativeness each representation component has about the data while simultaneously encouraging statistical independence between them.

In this work, we propose and explore a different strategy. Rather than directly targeting proper constraints for attaining desired representations characteristics, we step back to a very intuitive criterion that any good representation should, at least to some degree, retain a significant amount of information about the data. This suggests to maximize the mutual information between data and its representations, which is the substance of the *InfoMax* principle [90]. However, the mutual information can be trivially maximized by simply memorizing the data. InfoMax is thereby typically invoked with various constraints. A natural question arises as, building upon InfoMax, whether it's possible to derive proper constraints to yield the desired representation characteristics, *e.g.,* a disentangled continuous representation and interpretable categorical representation, yet maximally preserving the informativeness of representation. With this as our motivation and ultimate goal, we propose Regularized Information Maximization Auto-Encoding (RIMAE) for jointly learning a hybrid discrete and continuous representation. Our contributions lie in the following:

- Proposing an objective to simultaneously recover the underlying categories of data and disentangle the continuous representation factors in a way that each of them corresponds to a specific variation in data.
- While regularizations can naturally incur a loss on reconstruction, building upon InfoMax, the derived constraints allow us to achieve a better trade-off between

the desired representation characteristics and reconstruction fidelity.

- Providing a fundamental understanding on how to encourage statistically independent representation factors yet effectively preserving the informativeness of each factor.

4.2 Jointly Learning A Hybrid Categorical and Continuous Representation

Assume data $\boldsymbol{x} \in \mathbb{R}^d$ has been generated by combining a discrete factor with a fixed number of independent continuous factors, through a complex stochastic process. The discrete factor determines the category of the data, while the continuous latent factors correspond to the other variations in data. Let K_1^* and K_2^* denote the numbers of the underlying categories and continuous latent factors of data, respectively. Assume $K_1^*, K_2^* \ll d$, and we are primarily concerned with the reverse direction. That is, learning a hybrid discrete-continuous representation from observed data, using a stochastic autoencoder. Our goal is to uncover the underlying categories of data, while successfully separating the continuous latent factors into independent representation factors.

4.2.1 Classic Autoencoders and Beyond

Let $\boldsymbol{y} \in \{1, \ldots, K_1\}$ and $\boldsymbol{z} \in \mathbb{R}^{K_2}$, with $K_1, K_2 \ll d$, denote the discrete and continuous representation variables respectively, an ordinary autoencoder optimizes the following,

$$\text{maximize}_{\theta,\phi} \;\; \mathbb{E}_{q_\phi(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z})} \left[\log p_\theta(\boldsymbol{x}|\boldsymbol{y},\boldsymbol{z})\right] \;, \tag{4.1}$$

where q_ϕ and p_θ denote the stochastic encoder and decoder correspondingly. Notice that $q_\phi(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}) = q(\boldsymbol{x})q_\phi(\boldsymbol{y},\boldsymbol{z}|\boldsymbol{x})$, where $q(\boldsymbol{x})$ denotes the distribution of data $\boldsymbol{x}$, and $q_\phi(\boldsymbol{y},\boldsymbol{z}|\boldsymbol{x})$ denotes the conditional distribution, parameterized by the encoder,

of the representation given the data. The objective (4.1) is often termed as negative *reconstruction error*, though the output of decoder is generally not the exact reconstruction of the input $\boldsymbol{x}$, but instead being the probabilistic parameter(s) for a distribution $p_\theta(\boldsymbol{x}|\boldsymbol{y}, \boldsymbol{z})$ that may generate $\boldsymbol{x}$ with high probability. The choices for the likelihood distribution $p_\theta(\boldsymbol{x}|\boldsymbol{y}, \boldsymbol{z})$ are data dependent, two common ones are Gaussian distribution for real valued data and Bernoulli distribution for binary data.

Without any constraints on K_1 and K_2, the reconstruction error can be trivially minimized by simply learning an identity mapping. A natural solution is to use a bottleneck layer to produce under-complete representations, this often results in a loss on the reconstruction fidelity, and hence can be seen as *lossy compressed representation* learning. In this chapter, we follow this avenue by restricting $K_1, K_2 \ll d$. Nevertheless, the low dimensionality constraint alone is not enough to yield interpretable representations. This can be seen from two perspectives. First, only restricting $K_1, K_2 \ll d$ is not sufficient to uncover the true categories of data and disentangle the continuous representation in the desired way. Second, the model can simply memorize the data as long as the capacity of the autoencoder is high enough, which results in worse overfitting. Therefore, further regularization needs to be applied, which will naturally incur an additional loss on reconstruction. The goal is thereby finding proper constraints to attain a better trade-off between the reconstruction quality and the desired representation characteristics.

At the same time, it is evident that informative representations yield better reconstruction quality. This motivates us to revisit the InfoMax principle [90], which proposes to maximize the mutual information between data and its representations. However, the mutual information can be trivially maximized by simply memorizing the data. Instead of directly applying InfoMax, we take it as our starting point, upon

which we derive proper objective or constraints to preserve the informativeness of representation when seeking for the desired representation characteristics maximally. Next, we proceed by first showing that proper decomposition of the mutual information between data and its representation sheds more light on this goal.

4.2.2 Simultaneous Category Separation and Category Identification

Maximizing the mutual information $\mathrm{I}_\phi(\boldsymbol{x};\boldsymbol{y})$ between data $\boldsymbol{x}$ and its categorical representation $\boldsymbol{y}$ provides a very intuitive way for learning. To see this, recall that the mutual information between two random variables, *e.g.,* $\boldsymbol{x}$ and $\boldsymbol{y}$, can be decomposed as [43][1],

$$\mathrm{I}_\phi(\boldsymbol{x};\boldsymbol{y}) = \mathrm{H}_\phi(\boldsymbol{y}) - \mathrm{H}_\phi(\boldsymbol{y}|\boldsymbol{x}) \ . \tag{4.2}$$

Here $\mathrm{H}_\phi(\boldsymbol{y}) = \mathbb{E}_{q_\phi(\boldsymbol{y})}[-\log q_\phi(\boldsymbol{y})]$ denotes the entropy of $\boldsymbol{y}$ under the conditional probability $q_\phi(\boldsymbol{y}|\boldsymbol{x})$, and $\mathrm{H}_\phi(\boldsymbol{y}|\boldsymbol{x}) = \mathbb{E}_{q(\boldsymbol{x})q_\phi(\boldsymbol{y}|\boldsymbol{x})}[-\log q_\phi(\boldsymbol{y}|\boldsymbol{x})]$ denotes the conditional entropy of $\boldsymbol{y}$ given $\boldsymbol{x}$. Mutual information can thus be interpreted as the decrease in uncertainty of one random variable given another random variable [43]. In the context where $\boldsymbol{y}$ being the categorical representation, $\mathrm{H}_\phi(\boldsymbol{y})$ achieves the maximum if the marginal distribution of $\boldsymbol{y}$ is an uniform distribution over all categories, and $\mathrm{H}_\phi(\boldsymbol{y}|\boldsymbol{x})$ attains its minimum if the conditional distribution $q_\phi(\boldsymbol{y}|\boldsymbol{x})$ is deterministic for all $\boldsymbol{x}$. The entropy $\mathrm{H}_\phi(\boldsymbol{y})$ can thus be interpreted as the the *category balance* quantity, and the conditional entropy $\mathrm{H}_\phi(\boldsymbol{y}|\boldsymbol{x})$ can be seen as the *category separation* quantity. Therefore, maximizing $\mathrm{I}_\phi(\boldsymbol{x};\boldsymbol{y})$ is equivalent to uniformly assign data over all categories while simultaneously driving highly confident categorical identities for each data sample. Without any given priors on the distribution of categories, this is essentially the desired effect for unsupervised learning of categorical representation.

[1] Note (4.2) is generally true regardless of ϕ. We keep ϕ here for the consistency in notations.

Now the question is, given the conditional probabilistic model $q_\phi(\boldsymbol{y}|\boldsymbol{x})$, how we can effectively estimate and maximize the mutual information $\mathrm{I}_\phi(\boldsymbol{x};\boldsymbol{y})$. Mutual information is notoriously hard to compute, in particular in the high-dimensional and continuous settings. Fortunately, by leveraging the fact that the cardinality of the space of $\boldsymbol{y}$ (*i.e.,* the number of categories) is typically low, $\mathrm{I}_\phi(\boldsymbol{x};\boldsymbol{y})$ is computationally tractable. Specifically, let x_m denote a random sample of $\boldsymbol{x}$. Given M i.i.d samples $\{x_m\}_{m=1}^M$, let

$$\begin{aligned}\widehat{\mathrm{I}}_\phi(\boldsymbol{x};\boldsymbol{y}) &= \mathrm{H}\left(\frac{1}{M}\textstyle\sum_{m=1}^M q_\phi(\boldsymbol{y}|x_m)\right) - \frac{1}{M}\textstyle\sum_{m=1}^M \mathrm{H}\left(q_\phi(\boldsymbol{y}|x_m)\right)\\ &= \mathrm{H}\left(\widehat{q}_\phi(\boldsymbol{y})\right) - \widehat{\mathrm{H}}\left(q_\phi(\boldsymbol{y}|\boldsymbol{x})\right) \qquad (4.3)\end{aligned}$$

denote our estimation of $\mathrm{I}_\phi(\boldsymbol{x};\boldsymbol{y})$ under the conditional distribution $q_\phi(\boldsymbol{y}|x_m)$. As is indicated in Proposition 4.2.1, with a suitably large batch of samples, $\mathrm{I}_\phi(\boldsymbol{x};\boldsymbol{y})$ can be well estimated by it's estimate $\widehat{\mathrm{I}}_\phi(\boldsymbol{x};\boldsymbol{y})$ established in (4.3). This allows us a way to optimize $\mathrm{I}_\phi(\boldsymbol{x};\boldsymbol{y})$ that is amenable to stochastic gradient descent with minibatches of data. In other words, optimizing $\mathrm{I}(\boldsymbol{x};\boldsymbol{y})$ can be reduced to optimize $\widehat{\mathrm{I}}_\phi(\boldsymbol{x};\boldsymbol{y})$.

Proposition 4.2.1. *Let $\boldsymbol{y}$ be a discrete random variable that belongs to some categorical class $\mathcal{C}$. Suppose both the true and the estimated marginal distributions are bounded below, that is, $q_\phi(\boldsymbol{y}), \widehat{q}_\phi(\boldsymbol{y}) \in [1/(CK_1), 1]$ with some constant $C \geq 1$. Then for any $\delta \in (0,1)$, with probability at least $1-2\delta$ we obtain the following,*

$$\left|I_\phi(\boldsymbol{x};\boldsymbol{y}) - \widehat{I}_\phi(\boldsymbol{x};\boldsymbol{y})\right| \leq K_1\left(\max\{\log CK_1 - 1, 1\} + e\right)\sqrt{\frac{\log(2K_2/\delta)}{2M}}\,. \qquad (4.4)$$

Here M denotes the number of samples used to establish $\widehat{I}_\phi(\boldsymbol{x};\boldsymbol{y})$ as Eq (4.3).

4.2.3 Simultaneous Informativeness and Disentanglement for Continuous Representation

As for the continuous representation $\boldsymbol{z}$, to gain some insights into the relationship between the informativeness of each representation factor and the statistical

independence between the factors, we use the following decomposition of the mutual information $\mathrm{I}_\phi(\boldsymbol{x};\boldsymbol{z})$. Assume the conditional distribution $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ is factorial [2], let D_{KL} denote the KL divergence, then (we refer to Appendix 4.7.1 for the proof),

$$\mathrm{I}_\phi(\boldsymbol{x};\boldsymbol{z}) = \sum_{k=1}^{K_2}\mathrm{I}_\phi(\boldsymbol{x};\boldsymbol{z}_k) - \mathrm{D}_{KL}\left(q_\phi(\boldsymbol{z})\middle|\middle|\Pi_{k=1}^{K_2}q_\phi(\boldsymbol{z}_k)\right) \ . \tag{4.5}$$

The first term of the RHS of (4.5) quantifies how much information each representation factor $\boldsymbol{z}_k$ carries about the data $\boldsymbol{x}$. The second term is known as the *total correlation* of $\boldsymbol{z}$ [133], which quantifies the statistical dependence across the dimensions of $\boldsymbol{z}$ and achieves the minimum if and only if they are independent of each other. As is shown in (4.5), maximizing the mutual information $\mathrm{I}_\phi(\boldsymbol{x};\boldsymbol{z})$ can translate to maximizing the individual informativeness of each representation factor, while maximally reducing the statistical dependence between factors.

With data $\boldsymbol{x}$ being continuous, the mutual information between data and its continuous representation factor $\mathrm{I}_\phi(\boldsymbol{x};\boldsymbol{z}_k)$ can be trivially maximized by simply memorizing the data. This is due to the fact that $\mathrm{H}_\phi(\boldsymbol{z}_k|\boldsymbol{x})$ diverges to $-\infty$ when the conditional probability $q_\phi(\boldsymbol{z}_k|\boldsymbol{x})$ degenerates to Dirac delta distribution for each data sample. In our context, this can be seen according to the following proposition.

Proposition 4.2.2. *Suppose the conditional distribution $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ is a factorial Gaussian distribution with mean $\boldsymbol{\mu} \in \mathbb{R}^{K_2}$ and covariance diag$(\boldsymbol{\sigma}) \in \mathbb{R}^{K_2\times K_2}$. Then*[3]*,*

$$I_\phi(\boldsymbol{x};\boldsymbol{z}_k) \leq \frac{1}{2}\log\left(\mathbb{E}_{q(\boldsymbol{x})}\left[\boldsymbol{\sigma}_k^2|\boldsymbol{x}\right] + \mathrm{Var}_{q(\boldsymbol{x})}\left[\boldsymbol{\mu}_k|\boldsymbol{x}\right]\right) - \frac{1}{2}\mathbb{E}_{q(\boldsymbol{x})}\left[\log\boldsymbol{\sigma}_k^2|\boldsymbol{x}\right] \ . \tag{4.6}$$

The equality is attained if and only if $\boldsymbol{z}_k$ is Gaussian distributed.

Note here both $\boldsymbol{\mu}_k$ and $\boldsymbol{\sigma}_k$ are random variables. As we proved in Appendix 4.7.1, the variance of each representation factor can be established as $\mathrm{Var}(\boldsymbol{z}_k) =$

[2] In this work, we model $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ as factorial Gaussian distribution by explicitly parameterizing its mean and diagonal covariance accordingly in our implementation.

[3] While similar results have likely been established in the information theory literature, we include this proposition to motivate our objective design.

$\mathbb{E}_{q(\boldsymbol{x})}\left[\boldsymbol{\sigma}_k^2|\boldsymbol{x}\right] + \mathrm{Var}_{q(\boldsymbol{x})}\left[\boldsymbol{\mu}_k|\boldsymbol{x}\right]$. The inequality in (4.6) is incurred by the fact that, with any fixed value of $\mathrm{Var}(\boldsymbol{z}_k)$, the entropy $\mathrm{H}_\phi(\boldsymbol{z}_k)$ will achieve its maximum if and only if $\boldsymbol{z}_k$ is Gaussian distributed. A natural constraint raises as the variance of $\boldsymbol{z}_k$ should be finite. With any finite value of $\mathrm{Var}(\boldsymbol{z}_k)$, Proposition 4.2.2 indicates that maximizing $\mathrm{I}_\phi(\boldsymbol{x};\boldsymbol{z}_k)$ is equivalent to squeezing $\boldsymbol{z}_k$ within the domain of a Gaussian distribution while simultaneously decreasing the variance $\boldsymbol{\sigma}_k$ of the conditional distribution $q_\phi(\boldsymbol{z}_k|\boldsymbol{x})$. This matches the intuition that $\boldsymbol{z}_k$ is more informative about $\boldsymbol{x}$ if it has less uncertainty given $\boldsymbol{x}$, yet captures more variance in data, *i.e.,* $\boldsymbol{\sigma}_k$ is small while $\boldsymbol{\mu}_k$ dispersing within the domain the $\boldsymbol{z}$.

However, a vanished variance, *i.e.,* $\boldsymbol{\sigma}_k$ being zero for all data, results in the ordinary autoencoder setting. This is not desired given the following two major shortcomings: i) With limited training data and each data sample is mapped to a deterministic representation, the representation space can exhibit a certain degree of fragmentation, *i.e.,* being extremely non-smooth, and thereby overfitting. ii) The encoder can choose to remember whatever type of information, *e.g.,* noise specific to local patches or pixels, to improve the decoding quality maximally. To remedy this issue while achieving the upper bound in Proposition 4.2.2, a simple solution is to push $q_\phi(\boldsymbol{z}_k)$ towards a Gaussian distribution and simultaneously prevent $\boldsymbol{\sigma}_k$ from being too small. Therefore, we consider the following for maximizing $\mathrm{I}_\phi(\boldsymbol{x};\boldsymbol{z}_k)$ under proper constraint,

$$\text{maximize}_\phi \ -\sum_{k=1}^{K_2} D_{\mathrm{KL}}\left(q_\phi(\boldsymbol{z}_k)||p(\boldsymbol{z}_k)\right) - \alpha\sum_{k=1}^{K_2}\max(0,\sigma_k^* - \boldsymbol{\sigma}_k) \tag{4.7}$$

Here $p(\boldsymbol{z}_k)$ are i.i.d scaled normal distribution with finite variance. For the numerical results presented in this chapter, we set $\sigma_k^* = 0.01$ and $\alpha = 10^{-4}$ for all $k = 1, \dots K_2$. Using (4.7) as the proxy for maximizing $\mathrm{I}_\phi(\boldsymbol{x};\boldsymbol{z}_k)$ in (4.5) yields our objective for maximizing $\mathrm{I}_\phi(\boldsymbol{x};\boldsymbol{z})$.

4.2.4 Overall Objective

According to Sections 4.2.2&4.2.3, our overall objective can be summarized as,

$$\text{maximize}_{\theta,\phi}\ \mathbb{E}_{q_\phi(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z})}\left[\log p_\theta(\boldsymbol{x}|\boldsymbol{y},\boldsymbol{z})\right] + \text{InfoReg}(\boldsymbol{y}) + \text{InfoReg}(\boldsymbol{z})\ ,$$

$$\text{where}\quad \text{InfoReg}(\boldsymbol{y}) = \mathrm{I}_\phi(\boldsymbol{x};\boldsymbol{y})$$

$$\text{InfoReg}(\boldsymbol{z}) = -\textstyle\sum_{k=1}^{K_2} D_{\mathrm{KL}}\left(q_\phi(\boldsymbol{z}_k)||p(\boldsymbol{z}_k)\right) - \alpha\sum_{k=1}^{K_2}\max(0,\sigma_k^* - \boldsymbol{\sigma}_k)$$

$$- D_{\mathrm{KL}}\left[q_\phi(\boldsymbol{z})||\Pi_{k=1}^{K_2} q_\phi(\boldsymbol{z}_k)\right]$$

Recall that, both $-\left(\sum_{k=1}^{K_2} D_{\mathrm{KL}}\left(q_\phi(\boldsymbol{z}_k)||p(\boldsymbol{z}_k)\right) - \alpha\sum_{k=1}^{K_2}\max(0,\sigma_k^* - \boldsymbol{\sigma}_k)\right)$ and $\mathrm{I}_\phi(\boldsymbol{x};\boldsymbol{y})$ correspond to our information regularization (promoting) objective for each representation factor ($\boldsymbol{z}_k$ and $\boldsymbol{y}$), while $D_{\mathrm{KL}}\left[q_\phi(\boldsymbol{z})||\Pi_{k=1}^{K_2} q_\phi(\boldsymbol{z}_k)\right]$ characterizing the statistical independence between the continuous representation factors. Therefore, trade-off can be formalized between the informativeness of each latent factor and the statistical independence between the continuous representation factors. This motivates us to consider the following objective,

$$\text{maximize}_{\theta,\phi}\ \underbrace{\mathbb{E}_{q_\phi(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z})}\left[\log p_\theta(\boldsymbol{x}|\boldsymbol{y},\boldsymbol{z})\right]}_{\mathcal{L}_{\text{AutoEncoder}}} + \underbrace{\beta\mathrm{I}_\phi(\boldsymbol{x};\boldsymbol{y})}_{\text{Informativeness of } \boldsymbol{y}}$$

$$\underbrace{-\beta\left(\textstyle\sum_{k=1}^{K_2} D_{\mathrm{KL}}\left(q_\phi(\boldsymbol{z}_k)||p(\boldsymbol{z}_k)\right) - \alpha\sum_{k=1}^{K_2}\max(0,\sigma_k^* - \boldsymbol{\sigma}_k)\right)}_{\text{Informativeness of } \boldsymbol{z}_k} \underbrace{- \gamma D_{\mathrm{KL}}\left[q_\phi(\boldsymbol{z})||\Pi_{k=1}^{K_2} q_\phi(\boldsymbol{z}_k)\right]}_{\text{Dependence between } \boldsymbol{z}_k}\ .$$

Here $\beta, \gamma > 0$. As proposed in Eq (4.7), α is the regularization strength of preventing $\boldsymbol{\sigma}_k$ from being too small (below the prefixed threshold σ_k^*) so as to avoid degenerated solution.

4.3 Related Work

VAE and its variants. Variational autoencoder has shown great promise in learning disentangled representation. From the optimization perspective, VAE can be seen as

the ordinary autoencoder regularized in a specific way[4], indicated by Eq (4.8a). To better illustrate the focuses of VAE and its variants, we follow [73] and rewrite (4.8a) as (4.8b).

$$\text{(4.8a)}\quad \mathcal{L}_{\text{VAE}} = \underbrace{\mathbb{E}_{q_\phi(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z})}\left[\log p_\theta(\boldsymbol{x}|\boldsymbol{y},\boldsymbol{z})\right]}_{\mathcal{L}_{\text{AE}}} - \underbrace{D_{\text{KL}}\left[q_\phi(\boldsymbol{y}|\boldsymbol{x})||p(\boldsymbol{y})\right]}_{\mathcal{R}(\boldsymbol{y})} - \underbrace{D_{\text{KL}}\left[q_\phi(\boldsymbol{z}|\boldsymbol{x})||p(\boldsymbol{z})\right]}_{\mathcal{R}(\boldsymbol{z})}$$

$$\text{(4.8b)}\qquad = \mathcal{L}_{\text{AE}} - D_{\text{KL}}\left[q_\phi(\boldsymbol{y})||p(\boldsymbol{y})\right] - \text{I}_\phi(\boldsymbol{x};\boldsymbol{y}) - D_{\text{KL}}\left[q_\phi(\boldsymbol{z})||p(\boldsymbol{z})\right] - \text{I}_\phi(\boldsymbol{x};\boldsymbol{z})\ .$$

Here $p(\boldsymbol{y})$ and $p(\boldsymbol{z})$ denote the priors that are often chosen according to the assumption on how data is generated. The decomposition in (4.8) highlights the major drawback of the popular method β-VAE [71]. To be more specific, although using larger weights on $\mathcal{R}(\boldsymbol{y})$ and $\mathcal{R}(\boldsymbol{z})$ are expected to yield desired representation characteristics by heavily penalizing the divergence between $q_\phi(\boldsymbol{y})$ and $q_\phi(\boldsymbol{z})$ and their priors respectively, it also severely sacrifices the mutual information between data and its representation, and hence results in even less utilization of the representation and poor reconstruction.

Various methods have been proposed in recent work to address the limitations of β-VAE. [52, 7, 32, 92] propose to constrain the mutual information between the representation and the data by pushing its upper bounds, *i.e.,* $\mathcal{R}(\boldsymbol{y})$ and $\mathcal{R}(\boldsymbol{z})$ in Eq (4.8a), towards progressively increased target values. However, specifying and tuning the target values can itself be very challenging, which makes this method less practical. Alternatively, [144, 127] drop the mutual information terms in (4.8b). By pushing only the marginal distributions $q_\phi(\boldsymbol{y})$ and $q_\phi(\boldsymbol{z})$ towards their priors, they target the desired representation characteristics carried by the priors without explicitly penalizing the informativeness of the representation. Another relevant line of work [58, 80, 35, 55] proposes to learn disentangled continuous representation by minimizing the so-called *total correlation* term, either augmented as an extra term

[4]We augment the ordinary VAE objective by incorporating a discrete representation.

to (4.8) or obtained by further decomposing $\mathrm{D}_{KL}\left(q_\phi(\boldsymbol{z})||p(\boldsymbol{z})\right)$ in (4.8), as a way to encourage statistical independence between the representation components. Although these approaches can reduce the drawback of β-VAE to various degrees, they all suffer from uncovering the underlying categories of data. Better priors may compensate this weakness, however, finding such priors is itself a very challenging task.

Information theory based representation learning. The recent trend in leveraging information theory for representation learning was initially driven by supervised approaches. In this setting, the Information Bottleneck (IB) framework [125, 115, 126, 116, 113, 6, 4, 3] provides an elegant principle for representation learning as "keeping only what is relevant". To be more specific, the IB approach trades off the minimality of the representation by minimizing the informativeness it carries about the data, against the sufficiency of the representation for the task by maximizing the mutual information between the representation and the task.

The elephant in the room is that the IB framework focuses on supervised learning which often suffers from poor data availability and generalizability. One possible explanation is that the information required by a specific task is very limited. Thus the underlying structure of data is less explored. A long-standing idea in machine learning is that such limitation can be addressed by unsupervised learning. In the setting of unsupervised learning, maximizing the mutual information between data and its representation, *i.e.,* the so called *InfoMax* principle [90], is an important avenue [90, 21, 28, 19, 83, 72] for representation learning. Typically, InfoMax is invoked with some constraints to avoid trivial solutions, *e.g.,* preventing the model from simply memorizing the data.

In this chapter, we look at a challenging setting of unsupervised representation learning, where we target a hybrid discrete-continuous representation. Starting with the InfoMax principle, we derive key quantities and constraints that allow us to further sculpt the representation space, to achieve the desired representation characteristics while maximally preserving the informativeness of each representation factor.

4.4 Experimental Results

We compare RIMAE against various VAE based approaches that are summarized in Figure 4.1. We would like to demonstrate that RIMAE can (i) successfully learn a hybrid of continuous and discrete representations, with $\boldsymbol{y}$ matching the underlying categories $\boldsymbol{y}_{\mathbf{true}}$ well and $\boldsymbol{z}$ disentangling the explanatory factors hidden in data into its independent representation factors; (ii) outperform the VAE based models by achieving a better trade-off between the representation interpretability and the decoding quality.

$$\mathcal{L}_{\text{VAE}} = \underbrace{\mathcal{L}_{\text{AE}}}_{①} - \underbrace{D_{\text{KL}}\left[q_\phi(\boldsymbol{y})||p(\boldsymbol{y})\right]}_{②} - \underbrace{\mathrm{I}_\phi(\boldsymbol{x};\boldsymbol{y})}_{③} - \underbrace{D_{\text{KL}}\left[q_\phi(\boldsymbol{z})||p(\boldsymbol{z})\right]}_{④} - \underbrace{\mathrm{I}_\phi(\boldsymbol{x};\boldsymbol{z})}_{⑤} .$$

β-VAE: $① - \beta\,(② + ③) - \beta\,(④ + ⑤)$

InfoVAE: $① - \beta\ ③ - \beta\ ⑤$

Joint-VAE: $① - \beta\,|② + ③ - C_{\boldsymbol{y}}\ | - \beta\,|④ + ⑤ - C_{\boldsymbol{z}}|$

Figure 4.1: Relevant work. β-VAE modifies the original VAE objective by increasing the penalty on the KL divergence. InfoVAE drops the mutual information terms. JointVAE seeks to control the mutual information by pushing their upper bounds (*i.e.,* $② + ③$ and $④ + ⑤$) towards progressively increased values, $C_{\boldsymbol{y}}$&$C_{\boldsymbol{z}}$.

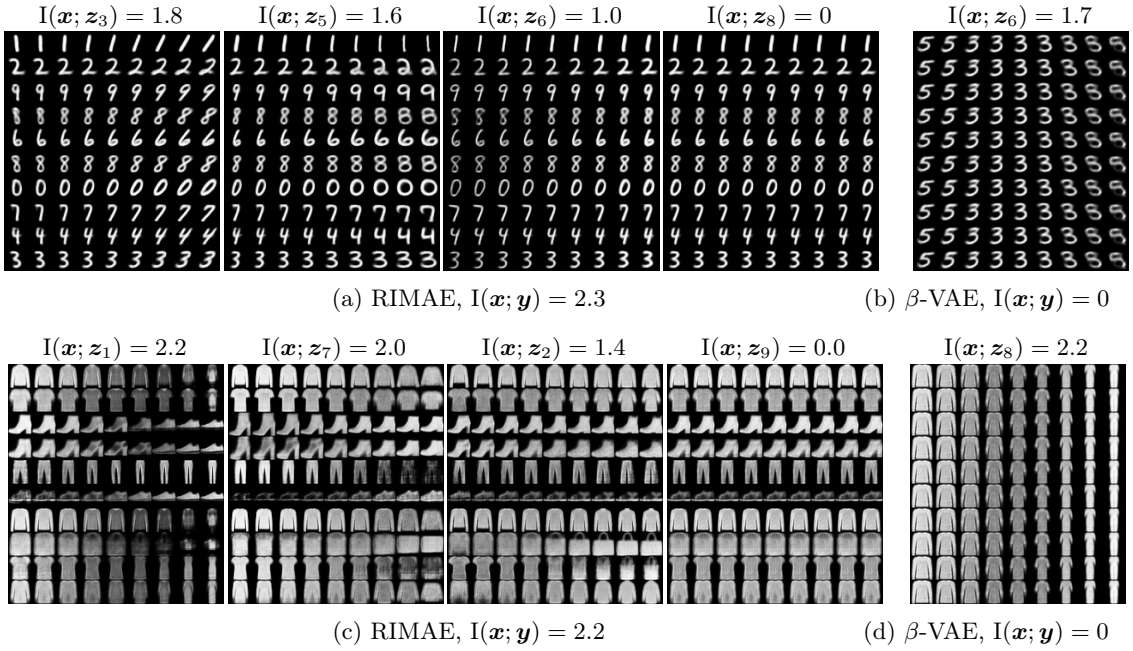
In this section, we perform quantitative evaluations on MNIST [87], Fashion MNIST [136] and dSprites [93]. We choose the priors $p(\boldsymbol{z})$ and $p(\boldsymbol{y})$ as isotropic Gaussian distribution and uniform distribution, respectively. Detailed experimental settings are provided in Appendix 4.7.3. For notational convenience, we drop the subscripts ϕ and θ hereafter.

4.4.1 MNIST and Fashion MNIST

We start by evaluating different methods on MNIST and Fashion MNIST in Figure 4.3. Both MNIST and Fashion MNIST contain 60000 binary 28×28 training images with only the underlying discrete factor (label) are given. Therefore, we quantitatively evaluate the interpretability of $\boldsymbol{y}$ only, which we report as the clustering accuracy. As for the continuous representation factors, we qualitatively assess their interpretability via latent space traversal.

Qualitative evaluation We start by qualitatively demonstrating that informative representations can potentially yield better interpretability. In each plot of Figure 4.2, we fix the discrete representation ($\boldsymbol{y}$) and traverse a continuous representation factor (indicated by the subtitle) in each row. Similarly, each column is obtained by fixing the value of the continuous representation $\boldsymbol{z}$ and traversing $\boldsymbol{y}$. As we can see, the informative continuous representation factors have uncovered intuitive factors of the variations in data, *e.g.,* angle, width, and thickness for MNIST, size, style, and brightness for Fashion MNIST. In contrast, traversing the continuous factors that achieve zero informativeness about the data shows no variation. We observe the same phenomenon for the discrete representation $\boldsymbol{y}$. Specifically, RIMAE can generate different digit (item) types with the same style, while the non-informative $\boldsymbol{y}$ learned by β-VAE completely fails to discover any useful categorical information of data.

Informative representations yield better reconstruction Figure 4.1 shows that, β-VAE implicitly pushes the marginal distributions of both continuous and discrete representations towards the associated priors respectively, *i.e.,* minimizing the divergences $D_{\mathrm{KL}}\left(q(\boldsymbol{y})||p(\boldsymbol{y})\right)$ and $D_{\mathrm{KL}}\left(q(\boldsymbol{z})||p(\boldsymbol{z})\right)$. Although using larger β values can



(a) RIMAE, $\mathrm{I}(\boldsymbol{x};\boldsymbol{y}) = 2.3$ (b) β-VAE, $\mathrm{I}(\boldsymbol{x};\boldsymbol{y}) = 0$

(c) RIMAE, $\mathrm{I}(\boldsymbol{x};\boldsymbol{y}) = 2.2$ (d) β-VAE, $\mathrm{I}(\boldsymbol{x};\boldsymbol{y}) = 0$

Figure 4.2: Maintaining informativeness of representation factors is necessary for capturing variations in data. In every plot above, each *row* is obtained by conditioning on a fixed value of $\boldsymbol{y}$ and traversing the associated $\boldsymbol{z}_k$ within range $[-2.5, 2.5]$; and each *column* shows the images generated by fixing the value of $\boldsymbol{z}$ and traversing $\boldsymbol{y} \in \{1, 2, \cdots, 10\}$. For each plot, the initial value of $\boldsymbol{z}$ is randomly sampled from the isotropic Gaussian distribution. As we can see, non-informative representation factors, *i.e.,* the non-informative $\boldsymbol{z}_k$ (learned by RIMAE) and the non-informative $\boldsymbol{y}$ (learned by β-VAE), completely fail at discovering any variations in data.

better minimize such divergences, as seen in Figure 4.1, it also increasingly diminishes the mutual information between data and its representation. This in turn results in less useful (informative) representations and poor reconstruction quality, as demonstrated in Figure 4.3. In contrast, Figure 4.3 also shows that, by either explicitly or implicitly preserving the mutual information between data and its representation, all the other three methods achieve much better reconstruction as we increase the penalty strength.

Figure 4.3 also implies that the reconstruction quality mainly depends on the informativeness of the continuous representation. This can be seen from two perspectives. First, when the regularization strength is small, all four models achieve the same

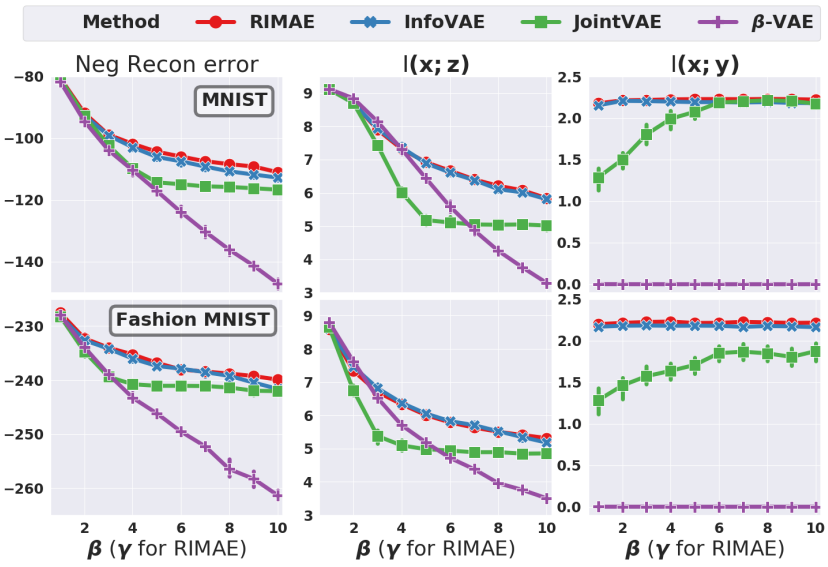


Figure 4.3: Informative representations yield better reconstruction. We train each model by sweeping β (γ for RIMAE) within the range $[1, 10]$. We set $\beta = \gamma/2$ for RIMAE. For each parameter value, we run each method on each dataset over 20 random initializations.

level of reconstruction quality, where the associated $\mathrm{I}(\boldsymbol{x}; \boldsymbol{z})$ are roughly the same while $\mathrm{I}(\boldsymbol{x}; \boldsymbol{y})$ are dramatically different. Second, the reconstruction error changes similarly as that of $\mathrm{I}(\boldsymbol{x}; \boldsymbol{z})$. One possible explanation is that the informativeness of the continuous representation $\mathrm{I}(\boldsymbol{x}; \boldsymbol{z})$ is much larger than that of the discrete representation $\mathrm{I}(\boldsymbol{x}; \boldsymbol{y})$. In the extreme case, $\mathrm{I}(\boldsymbol{x}; \boldsymbol{z})$ can be infinitely large, while $\mathrm{I}(\boldsymbol{x}; \boldsymbol{y})$ can always be upper bounded by $\log K_1$ and thereby is negligible. Therefore, in JointVAE, using larger target values of $C_{\boldsymbol{y}}$ for the discrete representation does not necessarily lead to better reconstruction, while a larger value of $C_{\boldsymbol{z}}$ for the continuous representation does yield high fidelity reconstruction. However, this comes along with poor independence between the continuous representation factors. This can be explained by Figure 4.1, where large $C_{\boldsymbol{z}}$ values not only promote the mutual information $\mathrm{I}(\boldsymbol{x}; \boldsymbol{z})$, but also drive $q(\boldsymbol{z})$ away from its factorial prior $p(\boldsymbol{z})$.

On I($\boldsymbol{x}; \boldsymbol{y}$) and interpretable categories of data We asses the performance of different methods on uncovering the underlying categories of data, which is quantitatively evaluated as the clustering accuracy. As seen in Figure 4.4, by simply pushing the conditional distribution $q(\boldsymbol{y}|\boldsymbol{x})$ towards a uniform distribution, β-VAE obtains a completely non-informative discrete representation $\boldsymbol{y}$. As a result, β-VAE is not able to recover the underlying categories. As a comparison, InfoVAE implicitly preserves the mutual information I($\boldsymbol{x}; \boldsymbol{y}$) by simply dropping it from the VAE objective. However, InfoVAE only optimizes the category balance of data by regularizing the marginal distribution $q(\boldsymbol{y})$ to be uniform. As a result, it cannot ensure good enough category separation, especially for those categories that are similar to each other. This is further demonstrated by the comparatively larger values of the conditional entropy H($\boldsymbol{y}|\boldsymbol{x}$) retained by InfoVAE. In contrast, RIMAE optimizes both category balance and category separation of data and hence performs better in uncovering the underlying categories of data.

JointVAE implicitly maximizes the mutual information I($\boldsymbol{x}; \boldsymbol{y}$) by pushing its upper bound, *i.e.,* $D_{\mathrm{KL}}(q(\boldsymbol{y}|\boldsymbol{x})||p(\boldsymbol{y}))$, towards a progressively increasing target value $C_{\boldsymbol{y}}$. However, Figure 4.4 implies that it can easily get stuck at some bad local optima, where the mutual information is still away from its maximum. A heuristic is that once JointVAE enters the local region of some local optima, progressively increasing $C_{\boldsymbol{y}}$ only induces oscillation within that region. On the other hand, even when JointVAE achieves comparatively larger mutual information I($\boldsymbol{x}; \boldsymbol{y}$), it can easily overfit. As is indicated by the comparatively smaller values of H($\boldsymbol{y}|\boldsymbol{x}$) in Figure 4.4, JointVAE tends to give very confident predictions on the category identities for each data sample, even when the clustering accuracy is poor.

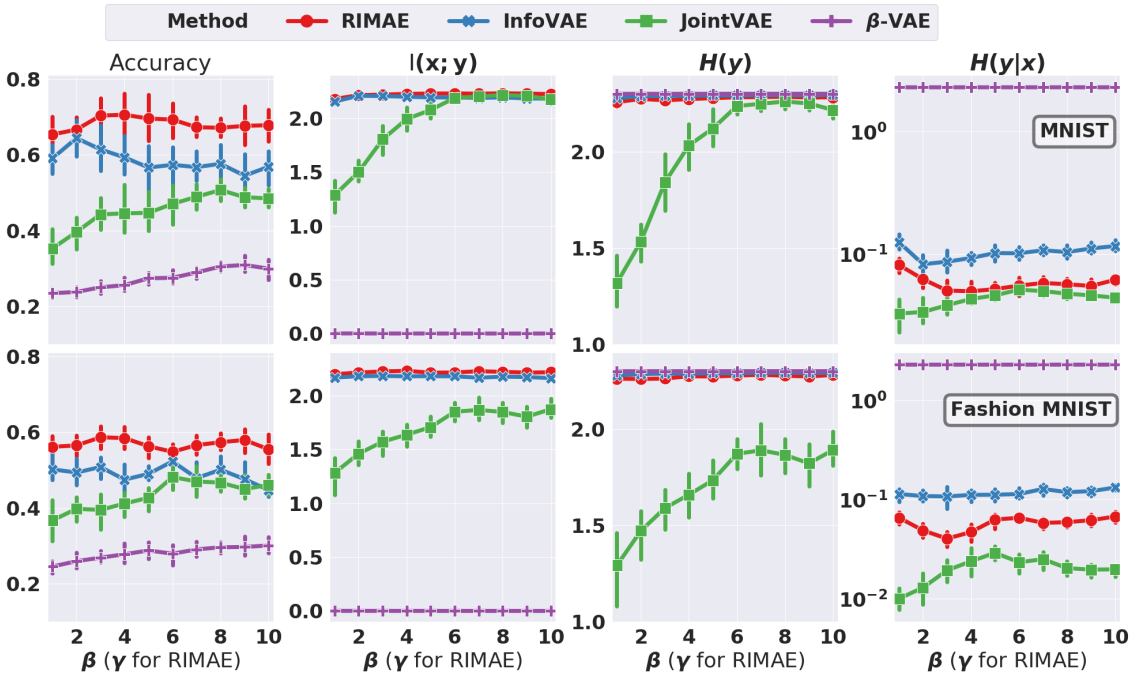
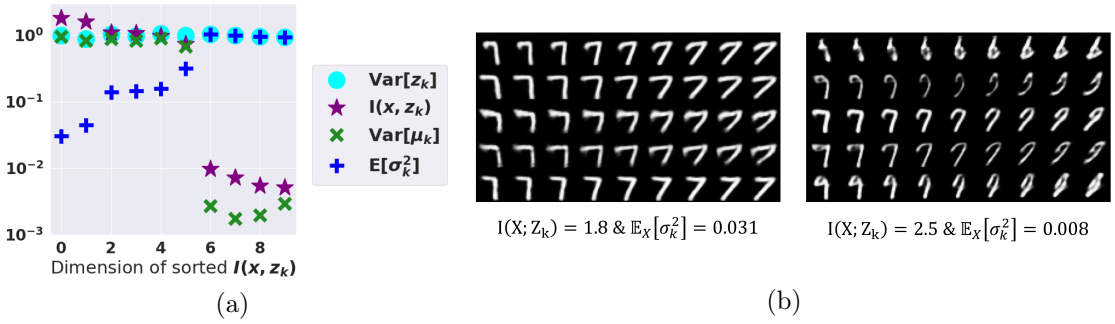


Figure 4.4: Quantitatively evaluation on the discrete representation $\boldsymbol{y}$. We train each model by sweeping β (γ for RIMAE) within the range $[1, 10]$. We set $\beta = \gamma/2$ for RIMAE. For each β value, we run each method on each dataset over 20 random initializations.

On $\mathbf{I}(\boldsymbol{x}; \boldsymbol{z}_k)$ and interpretable variations in data For each value of $\boldsymbol{x}$, the associated value of the mean $\boldsymbol{\mu}_k$ is such that the conditional distribution $q(\boldsymbol{z}_k|\boldsymbol{x})$ will concentrate around that value, while the corresponding value of $\boldsymbol{\sigma}_k$ quantifies the variation of $\boldsymbol{z}_k$ allowed to deviate from the mean $\boldsymbol{\mu}_k$. Figure 4.5(a) shows that the representation factor $\boldsymbol{z}_k$ is more informative about the data $\boldsymbol{x}$ if it (i) can effectively capture the variation in data by dispersing the mean values $\boldsymbol{\mu}_k$ of the conditional distribution $q(\boldsymbol{z}_k|\boldsymbol{x})$ across data samples, and (ii) has less uncertainty given $\boldsymbol{x}$, *i.e.,* $\boldsymbol{\sigma}_k$ being small for each data sample.

On the other hand, a vanishing variance, *i.e.,* $\boldsymbol{\sigma}_k$ being zero for all values of $\boldsymbol{x}$, degenerates the conditional distribution $q(\boldsymbol{z}_k|\boldsymbol{x})$ to the Dirac delta distribution. In this setting, each value of $\boldsymbol{x}$ (each data sample) is associated with a deterministic value of $\boldsymbol{z}_k$, hence the space of $\boldsymbol{z}_k$ may not be continuous, and it's prone to overfitting.



(a) (b)

Figure 4.5: On the informativeness of the continuous representation factors. (a) The continuous representation factor $\boldsymbol{z}_k$ is more informative about $\boldsymbol{x}$ if it has less uncertainty given $\boldsymbol{x}$ ($\boldsymbol{\sigma}_k$ being small), yet captures more variance in data ($\boldsymbol{\mu}_k$ dispersing across data samples). (b) Latent traverse of continuous representation factors, learned by RIMAE with different regularization strengths, that encode the angle of digits. For each plot, we fix the value of $\boldsymbol{y}$ and traverse the associated dimension $\boldsymbol{z}_k$ in each row. In each row, we initialize $\boldsymbol{z}$ by randomly sample a value from the isotropic gaussian distribution $\mathcal{N}(0, \mathbb{I}_d)$, then we traverse the dimension $\boldsymbol{z}_k$ within range $[-2.5, 2.5]$.

To better illustrate this, we traverse the representation factors that associate with different levels of variance $\boldsymbol{\sigma}_k$ and informativeness $\mathrm{I}(\boldsymbol{x}; \boldsymbol{z}_k)$ in Figure 4.5(b). As is implied, maintaining proper magnitude of the $\boldsymbol{\sigma}_k$ preserves the local smoothness of the space of $\boldsymbol{z}_k$, thereby small traversals generate images with small, consistent transforms in the angle of the digits. However, with $\boldsymbol{\sigma}_k$ too small, there exist noticeable representational discontinuities, for which small latent traversals generate images with inconsistent variation. The generated images near the boundaries of these fragments are often of poor quality and dramatically different from the training data.

On the disentanglement of discrete and continuous representations For the data where different categories share common variations, we want to disentangle the discrete representation $\boldsymbol{y}$ and the continuous representation $\boldsymbol{z}$. In particular, we seek to retain the underlying categorical information maximally, denoted as $\boldsymbol{y}_{\mathbf{true}}$, of data in $\boldsymbol{y}$, while removing it from $\boldsymbol{z}$.

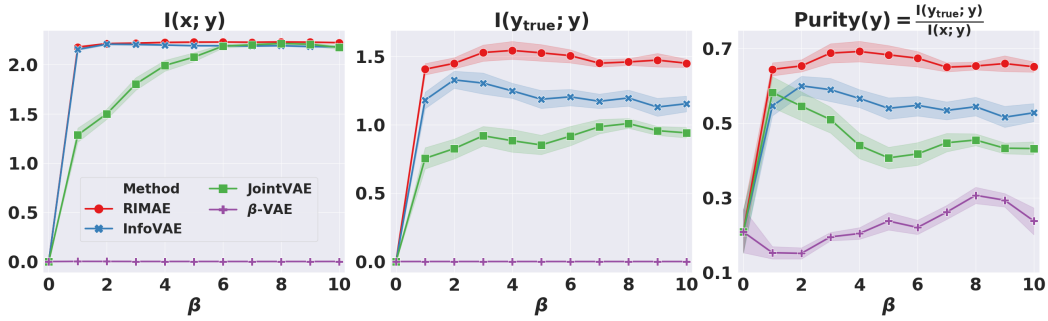
We first provide another perspective to highlight the importance of promoting

$\mathrm{I}(\boldsymbol{x};\boldsymbol{y})$. Figure 4.6 shows that, without any constraints, *i.e.,* β(or γ) $= 0$, the ordinary autoencoder tends to completely ignore the discrete representation, where the associated mutual information $\mathrm{I}(\boldsymbol{x};\boldsymbol{y})$ vanishes. In this setting, $\mathrm{I}(\boldsymbol{x};\boldsymbol{z})$ can be infinitely large when $\boldsymbol{z}$ is a deterministic function of continuous $\boldsymbol{x}$, so the usefulness of $\boldsymbol{y}$ is therefore negligible. This implies that, to obtain interpretable $\boldsymbol{y}$, we need to either promote $\mathrm{I}(\boldsymbol{x};\boldsymbol{y})$ or decrease $\mathrm{I}(\boldsymbol{x};\boldsymbol{z})$. However, Figure 4.6 shows that, with a wide range of regularization strength, $\mathrm{I}(\boldsymbol{x};\boldsymbol{z})$ still dominates $\mathrm{I}(\boldsymbol{x};\boldsymbol{y})$, hence the reconstruction quality mainly depends on $\mathrm{I}(\boldsymbol{x};\boldsymbol{z})$ (as is discussed in Section 4.4.1) and the usefulness of the discrete representation is negligible. This sheds insights into why we should promote the informativeness of $\boldsymbol{y}$ instead of further penalizing it.

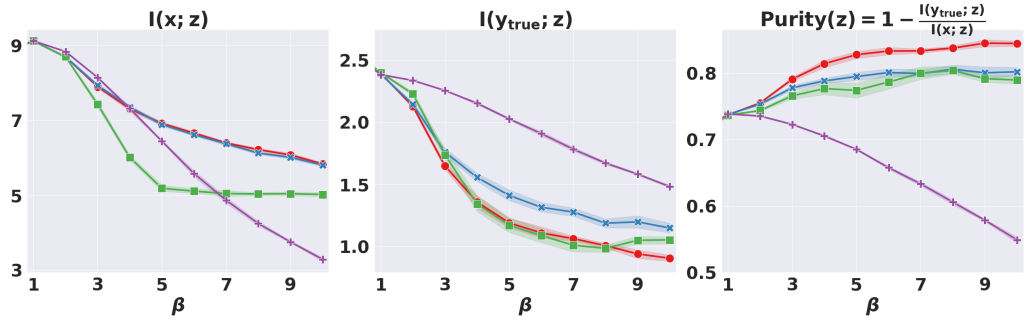
Next we show that penalizing the mutual information $\mathrm{I}(\boldsymbol{x};\boldsymbol{z})$ between data and its continuous representation can implicitly decrease the informativeness of $\boldsymbol{z}$ about the underlying categories of data, *i.e.,* $\mathrm{I}(\boldsymbol{y}_{\mathbf{true}};\boldsymbol{z})$. To be more specific, we can prove the following (the proof is provided in Appendix 4.7.1),

$$\mathrm{I}(\boldsymbol{y}_{\mathbf{true}};\boldsymbol{z}) \leq \mathrm{I}(\boldsymbol{x};\boldsymbol{z}) - \mathrm{H}(\boldsymbol{x};\boldsymbol{y}_{\mathbf{true}}) - \mathbb{E}_{q(\boldsymbol{y},\boldsymbol{z}|\boldsymbol{x})q(\boldsymbol{x})}[\log p_\theta(\boldsymbol{x}|\boldsymbol{y},\boldsymbol{z})] \tag{4.9}$$

Notice that, $\mathrm{H}(\boldsymbol{x}|\boldsymbol{y}_{\mathbf{true}})$ is a constant, and $\mathbb{E}_{q(\boldsymbol{y},\boldsymbol{z}|\boldsymbol{x})q(\boldsymbol{x})}[\log p(\boldsymbol{x}|\boldsymbol{y},\boldsymbol{z})]$ is the *negative reconstruction error*. Therefore, the informativeness of $\boldsymbol{z}$ about the underlying categories $\boldsymbol{y}_{\mathbf{true}}$ of data can be implicitly decreased by penalizing $\mathrm{I}(\boldsymbol{x};\boldsymbol{z})$ as well as the reconstruction error. This is further demonstrated in Figure 4.6(b), where $\mathrm{I}(\boldsymbol{x};\boldsymbol{y}_{\mathbf{true}})$ diminishes along with the decrease of $\mathrm{I}(\boldsymbol{x};\boldsymbol{z})$. However, the purity of the continuous representation, defined as $\mathrm{Purity}(\boldsymbol{z}) = 1 - \frac{\mathrm{I}(\boldsymbol{y}_{\mathbf{true}};\boldsymbol{z})}{\mathrm{I}(\boldsymbol{x};\boldsymbol{z})}$, increase for all models except β-VAE as we increase the regularization strength. One possible explanation is that β-VAE is incapble of encoding the categorical information $\boldsymbol{y}_{\mathbf{true}}$ into its discrete representation, hence the continuous representation needs to carry more information about $\boldsymbol{y}_{\mathbf{true}}$, which is significant for guaranteeing the reconstruction quality. This



(a) Without any regularization ($\beta = 0$), the informativeness of the discrete representation $\boldsymbol{y}$ is already very poor. The reconstruction quality is determined by the informativeness of the continuous representation, since $\mathrm{I}(\boldsymbol{x}; \boldsymbol{z})$ can be infinitely large, while $\mathrm{I}(\boldsymbol{x}; \boldsymbol{y})$ can always be upper bounded by $\log K_1$ and thereby is negligible. Therefore, to obtain interpretable $\boldsymbol{y}$, we should promote $\mathrm{I}(\boldsymbol{x}; \boldsymbol{y})$ instead of further penalizing it, which is what β-VAE does.



(b) Penalizing $\mathrm{I}(\boldsymbol{x}; \boldsymbol{z})$ implicitly decreases $\mathrm{I}(\boldsymbol{y}_{\mathbf{true}}; \boldsymbol{z})$. Compared to β-VAE, the other three models are able to retain the true categorical information ($\boldsymbol{y}_{\mathbf{true}}$) in $\boldsymbol{y}$ to various degrees (see (a)), and hence increase the purity of $\boldsymbol{z}$ in terms of driving $\boldsymbol{z}$ to be less informative about $\boldsymbol{y}$ while better preserving the informativeness of $\boldsymbol{z}$ regarding the other variations in data.

Figure 4.6: Disentanglement of $\boldsymbol{y}$ and $\boldsymbol{z}$ regarding the true categories of data, denoted as $\boldsymbol{y}_{\mathbf{true}}$. We track the mutual information between $\boldsymbol{y}_{\mathbf{true}}$ and (a) discrete representation $\boldsymbol{y}$, and **(b)** continuous representation $\boldsymbol{z}$.

implies that the disentanglement of $\boldsymbol{y}$ and $\boldsymbol{z}$ can be obtained by penalizing $\mathrm{I}(\boldsymbol{x}; \boldsymbol{z})$ while simultaneously encouraging the informativeness of $\boldsymbol{y}$.

4.4.2 2D Shapes

In this section, we quantitatively evaluate the disentanglement capability of RIMAE on dSprites. This dataset contains 737,280 binary 64×64 images with six ground truth factors: shape(3), scale(6), orientation(40), x-position(32), y-position(32). Parentheses contain the number of quantized values for each true latent factor. We set $\boldsymbol{y} \in \{1, 2, 3\}$ and $\boldsymbol{z} \in \mathbb{R}^{10}$, where we expect that $\boldsymbol{y}$ can recover the shapes of data,

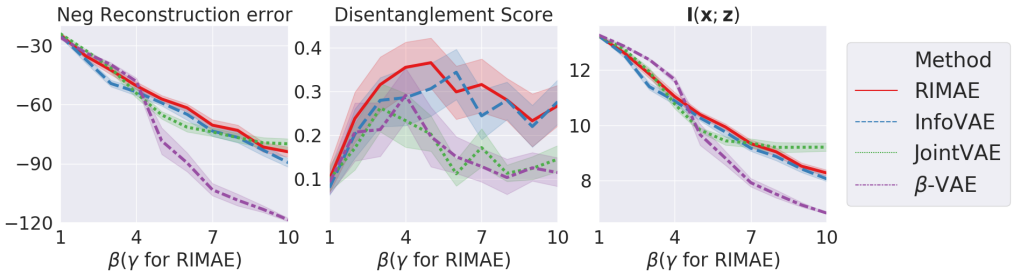


Figure 4.7: Disentanglement vs. reconstruction on dSprites. The results are reported by training each method with $\beta \in [1, 10]$, and we set $\beta = \gamma/2$ with $\gamma \in [1, 10]$ for RIMAE. For each β value, every method is trained over 20 random initializations. Shade regions indicate the 90% confidence intervals.

and $\boldsymbol{z}$ can separate the other underlying factors into its disjoint dimensions, *i.e.,* each $\boldsymbol{z}_k$ associates with a different underlying factor.

To asses the reconstruction vs. reconstruction trade-off, we use the disentanglement metric proposed by [35],

$$\text{MIG} = \frac{1}{J}\sum_{j=1}^{J}\frac{1}{\text{H}(v_j)}\left(\widehat{\text{I}}(\boldsymbol{z}_{k^{(j)}}; v_j) - \max_{k\neq k^{(j)}}\widehat{\text{I}}(\boldsymbol{z}_k; v_j)\right) , \tag{4.10}$$
$$\text{where } k^{(j)} = \arg\max_k \text{I}(\boldsymbol{z}_k; v_j) .$$

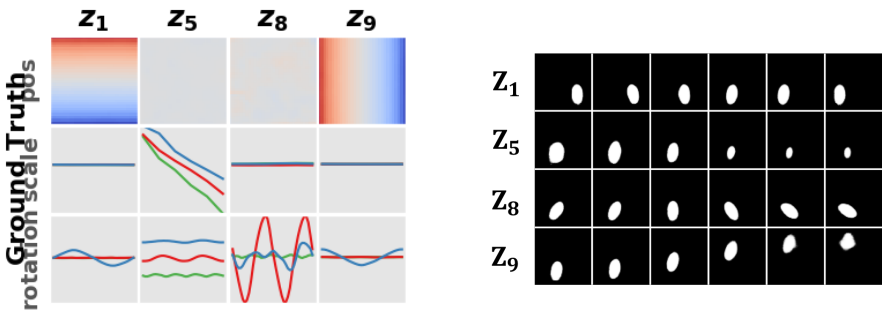
Here J is the number of ground truth factors, v_j denotes the j^{th} true factor, and $\widehat{\text{I}}(\boldsymbol{z}_k; v_j)$ denotes the empirical mutual information between a latent representation factor $\boldsymbol{z}_k$ and the ground truth factor v_j. As we can see, for each true factor v_j, MIG first measures the gap between the largest two informativeness among all of those achieved by the representation factors about v_j. The disentanglement score is then defined as the weighted average of the gaps that are obtained over all true factors. A high disentanglement score implies that each ground truth factor associated with one single representation factor that is more informative than the others.

Disentanglement vs. reconstruction Figure 4.7, shows that β-VAE struggles in achieving a good trade-off between disentanglement and reconstruction. This is

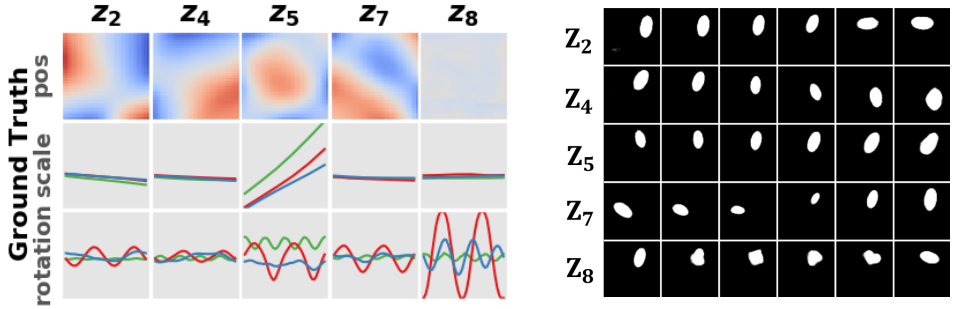
caused by the fact that, although using large β values can heavily push the marginal distribution $q(\boldsymbol{z})$ towards the factorial prior, it also significantly sacrifices the informativeness of the representation. To be more specific, using large β values can degrade the informativeness of each representation factor, which can scale down the difference between the top two informativeness in Eq (4.10). Moreover, using large β values can also decrease the number of informative representation factors (dimensions of $\boldsymbol{z}$). As a result, there may not exist enough informative representation factors to separate the underlying latent factors of data, and different variations in data can even be encoded into one single representation factor.

As a comparison, both InfoVAE and RIMAE achieve better disentanglement vs. reconstruction quality trade-off. Specifically, Figure 4.7 shows that RIMAE consistently obtains better disentanglement in the region of interest where both the decoding quality and the informativeness of representation are reasonably good, *i.e.,* when β ranges from 3 to 5. We attribute this to the effect of explicitly encouraging statistically independent representation factors by minimizing the total correlation term in our objective. On the other side, with large β values, InfoVAE further minimizes the divergence between the marginal distribution of the representation towards the factorial prior, leading to good disentanglement that is comparable to that achieved by RIMAE in the same region. However, in such a region, the associated decoding quality and the informativeness of the representation are both poor, which hinders the usefulness of the learned representations.

As has been observed in Figure 4.7, even with higher values of β, JointVAE can still maintain certain amount of mutual information that is very close to the target value $C_{\boldsymbol{z}}$. This is expected for JointVAE, since larger β values further minimize the divergence between the upper bound of the mutual information, *i.e.,* $D_{\mathrm{KL}}\left(q(\boldsymbol{z}|\boldsymbol{x})||p(\boldsymbol{z})\right)$, and



(a) Disentanglement score (MIG) = 0.44. High MIG indicates more disentangled representation. Left: groundth truth factors vs. representation factors; Right: traversing $\boldsymbol{z}_k$ conditioned on $\boldsymbol{y} = 1$.



(b) Disentanglement score (MIG) = 0.09. Low MIG value indicates more entangled representation. Left: groundth truth factors vs. representation factors; Right: traversing $\boldsymbol{z}_k$ conditioned on $\boldsymbol{y} = 1$.

Figure 4.8: RIMAE on dSprites. Left column: Relation between the continuous representation factors and the ground truth factors. Each row corresponds to a ground truth factor and each column to a latent variable. Each cell shows the relationship between the mean of a representation factor versus the quantized ground truth factor. For the position factor (first row), blue indicates high value and red indicates low value. The colored lines indicate object shape, oval(red), square (green), and heart (blue). We only plot those $\boldsymbol{z}_k$ where $\text{Var}[\boldsymbol{\mu}_k] \geq 0.01$. Right column: Traversing the learned representation factors listed in the left column. For each plot we randomly sample a isotropic Gaussian distributed vector $z \in \mathbb{R}^{K_2}$, then traverse each z_k within range $[-2.5, 2.5]$ for each row.

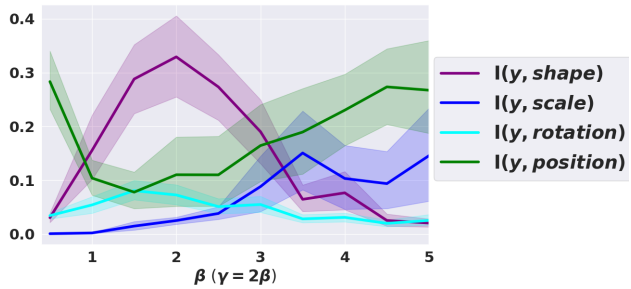
the target value $C_{\boldsymbol{z}}$. However, the disentanglement achieved by JointVAE is generally poor. The reason is that JointVAE indeed pushes the summation of the mutual information $\text{I}(\boldsymbol{x}; \boldsymbol{z})$ and the divergence $D_{\text{KL}}(q(\boldsymbol{z})||p(\boldsymbol{z}))$ towards a target value, hence there is no guarantees that the divergence $D_{\text{KL}}(q(\boldsymbol{z})||p(\boldsymbol{z}))$ would diminish to zero even at the convergence. Figure 4.7 indicates that, over a wide range of β values, the representation $\boldsymbol{z}$ learned by JointVAE is comparatively more correlated across its dimensions.

High disentanglement score implies better interpretability To assess whether higher disentanglement score (MIG) indeed implies more disentangled representation, we evaluate the learned continuous representations associated with different MIG scores.[5] Figure 4.8(a) shows that the representation that corresponds to higher MIG score successfully separates the underlying latent factors of data into different representation factors. In contrast, Figure 4.8(b) indicates that the representation achieves a lower MIG score exhibits high entanglement, where either one single representation factor learns more than one underlying factors of data, or one underlying factor of data is mapped to multiple representation factors.

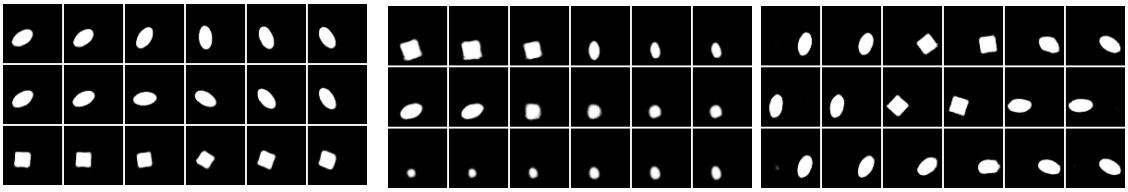
Non-prominent categorical information We also explore the discrete representation learned by RIMAE in Figure 4.9. Compared to the other variations in dSprites, especially the variations in position and scale, the variation of shape is not prominent. However, Figure 4.9 shows that RIMAE can still learn the shape information to some degree. On the other hand, when (β, γ) is comparatively smaller or larger, RIMAE tends to encode other variations in data rather than the shape information into the categorical representation $\boldsymbol{y}$. One possible explanation is that, in these two regions, the reconstruction term in the RIMAE objective affects the distribution of representation more.

To be more specific, when (β, γ) is small, the reconstruction term in the RIMAE objective weights more in sculpting the representation space. Figures 4.9(a)&(d) imply that RIMAE assigns the more prominent information, *i.e.,* position, to the discrete representation $\boldsymbol{y}$. As seen in Figure 4.9(d), the categorical representation $\boldsymbol{y}$ learned by RIMAE separates the location of data into two regions. We found that the subtle changes within each of such regions are encoded in the continuous

[5] For Figures (a) and (c), we borrow the idea from [35].



(a) I($\boldsymbol{y}$, true factors)



(b) I($\boldsymbol{y}$, shape) = 0.54
I($\boldsymbol{y}$, rotate) = 0.13

(c) I($\boldsymbol{y}$, shape) = 0.13
I($\boldsymbol{y}$, scale) = 0.47

(d) I($\boldsymbol{y}$, shape) = 0.01
I($\boldsymbol{y}$, position) = 0.68

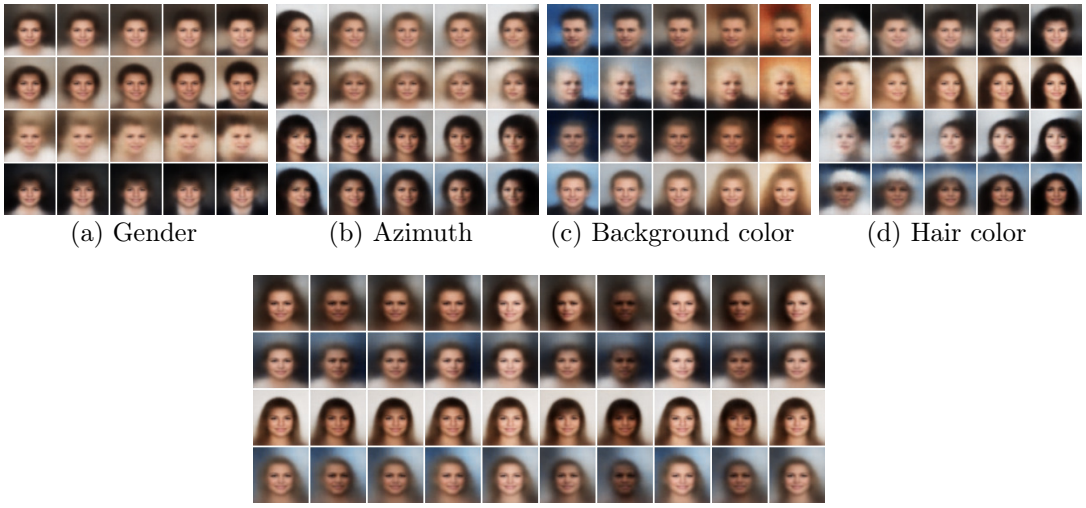
Figure 4.9: RIMAE on dSprites. (a): Tracking the mutual information between $\boldsymbol{y}$ and various ground truth latent factors over β values. The plots are generated over 20 random runs. Shade regions indicate the 90% confidence intervals. (b)-(d): For each plot, we traverse $\boldsymbol{y}$ in each column and traverse one continuous representation factor $\boldsymbol{z}_k$ in each row.

representations to guarantee good reconstruction quality. Similarly, large (β, γ) values strongly regularize the latent representation, which can incur loss on the reconstruction severely. To maximally preserve the reconstruction fidelity in this situation, RIMAE again encodes the more prominent information in data, *e.g.,* position and scale, into its discrete representation.

4.4.3 Qualitative Results on CelebA

To evaluate RIMAE on more realistic data, we consider CelebA [91]. For this dataset, neither the continuous nor the discrete latent factors are known. Therefore we qualitatively evaluate RIMAE by performing latent traverse on the learned representations. We set $\boldsymbol{z} \in \mathbb{R}^{32}$ and $\boldsymbol{y} \in \{1, 2, \ldots, 10\}$.

In Figure 4.10, we traverse both discrete and continuous representation factors learned by RIMAE. In each plot, each row is obtained by randomly sampling an



(a) Gender (b) Azimuth (c) Background color (d) Hair color

(e) Traversing $\boldsymbol{y} \in \{1, 2, \dots 10\}$ in each row.

Figure 4.10: RIMAE on CelebA. (a)–(d): Latent traverse of part of the continuous latent factors learned by RIMAE. For each row, we randomly sample an isotropic Gaussian random vector z, and then traverse one continuous factor (one dimension of z) within range $[-3, 3]$ while fixing the other dimensions. (e): Latent traverse of the discrete representation $\boldsymbol{y}$. Each row is obtained by randomly sampling an isotropic Gaussian random vector z, and then traversing $\boldsymbol{y} \in \{1, 2, \dots, 10\}$.

isotropic Gaussian random vector, and traversing the corresponded representation factor ($\boldsymbol{z}_k$ or $\boldsymbol{y}$). Figures 4.10(a)–(d) show that, RIMAE successfully uncovers and separate intuitive factors hidden in data into different representation factors. As for the discrete representation $\boldsymbol{y}$, instead of facial identity, the discrete representation $\boldsymbol{y}$ encodes the other variations in data, including azimuth, face brightness, or hairstyle. Following the discussion for dSprites, there exist several other prominent variations in CelebA, *e.g.,* hair color, background color, or azimuth. Therefore, it's less clear for the model which aspect of the data should be used to separate data into different categories.

4.5 Matching the intrinsic dimension of data

As for the continuous representation, our current work implicitly assumes the intrinsic dimension of data is known, *i.e.,* the number of true latent factors is

known, and we seek to uncover and separate these factors to different representation components while properly regularizing the informativeness of each component to capture useful variations in data. However, the actual number of latent factors is generally unknown, a natural solution could be starting with a large dimension for the continuous representation, and encouraging the model to discover the intrinsic dimension of true latent factors by using proper regularization. Instead of only preventing the variance of the conditional distribution from being too small, we consider the following regularization [6]:

$$\mathbf{Reg}(\boldsymbol{z}) = \sum_{k=1}^{K_2} \left| \log \left(\frac{\boldsymbol{\sigma}_k^2}{h_k} \right) \right| . \tag{4.11}$$

Here h_k is the variance of the prior $p(\boldsymbol{z}_k)$ associated with the k-th dimension of $\boldsymbol{z}$. As we can see, for each dimension k, $|\log(\boldsymbol{\sigma}_k^2)|$ ranges from $[0, \infty)$. Explicitly, it equals 0 if $\boldsymbol{\sigma}_k = h_k$ and diverges to ∞ when $\boldsymbol{\sigma}_k$ equals 0 or ∞.

Notice that, by squeezing the marginal distribution $q(\boldsymbol{z}_k)$ within the domain of a Gaussian distribution $p(\boldsymbol{z}_k)$ whose variance is h_k, we can upper bounded $\boldsymbol{\sigma}_k$ by h_k. Therefore, minimizing the $\mathbf{Reg}(\boldsymbol{z})$, defined in Eq (4.7), can effectively remove the redundant dimensions of $\boldsymbol{z}$ by pushing the associated $\boldsymbol{\sigma}_k$ towards h_k. Moreover, it can also prevent $\boldsymbol{\sigma}_k$ from being vanished (*i.e.,* $\boldsymbol{\sigma}_k = 0$), thereby effectively avoiding overfitting.

Instead of using the regularization in (4.7), we consider the following for regularizing the continuous representation $\boldsymbol{z}$,

$$\text{maximize}_\phi \ -\sum_{k=1}^{K_2} D_{\mathrm{KL}}\left(q_\phi(\boldsymbol{z}_k)||p(\boldsymbol{z}_k)\right) - \sum_{k=1}^{K_2} \left| \log \left(\frac{\boldsymbol{\sigma}_k^2}{h_k} \right) \right| \tag{4.12}$$

It can be of great practical interest to explore this new regularization to see whether it can better uncover the intrinsic dimension of the true latent continuous representation $\boldsymbol{z}$, and enable RIMAE to tackle more general problems effectively.

[6] This regularization was first proposed in [110] in the setting the wasserstein autoencoder.

4.6 Conclusion

We have proposed RIMAE, a novel approach for uncovering the underlying categories of data, while simultaneously disentangling the other hidden explanatory factors of data into disjoint parts of the learned continuous representation. Instead of directly targeting proper constraints for the ordinary autoencoders to obtain the desired representation characteristics, RIMAE steps back to a natural criterion that any good representations should be informative about the data, and proposes to maximizing the mutual information between data and its representations at the very beginning. We show that the information maximization objective provides a very natural way for unsupervised learning of categorical representation while providing a connection between the informativeness of each continuous representation factor and the statistical independence between these factors. The constraints are implied by our objective to avoid degenerated solutions.

Unsupervised joint learning of disentangled continuous and discrete representations is a challenging problem due to the lack of prior for semantic awareness and other inherent difficulties that arise in learning discrete representations. This work takes a step towards achieving this goal.

4.7 Supplementary Material

4.7.1 Proof of the Main Results

Proof of Equation (4.5) Recall that

$$I_\phi(\boldsymbol{x}; \boldsymbol{z}) = \mathrm{H}_\phi(\boldsymbol{z}) - \mathrm{H}_\phi(\boldsymbol{z}|\boldsymbol{x}) \; . \tag{4.13}$$

Let $\mathcal{X}, \mathcal{Z}, \mathcal{Z}_k$ denote the domains $\boldsymbol{x}, \boldsymbol{z}$ and $\boldsymbol{z}_k$ correspondingly. We can further decompose $\mathrm{H}_\phi(\boldsymbol{z})$ and $\mathrm{H}_\phi(\boldsymbol{z}|\boldsymbol{x})$ as,

$$\begin{aligned}
\mathrm{H}_\phi(\boldsymbol{z}|\boldsymbol{x}) &= -\int_{\mathcal{X}} q(x) \int_{\mathcal{Z}} q_\phi(z|x) \log q_\phi(z|x) dz dx \\
&\overset{\vartheta_1}{=} \sum_{k=1}^{K_2} -\int_{\mathcal{X}} q(x) \int_{\mathcal{Z}} q_\phi(z|x) \log q_\phi(z_k|x) dz dx \\
&= \sum_{k=1}^{K_2} H_\phi(\boldsymbol{z}_k|\boldsymbol{x}) \ , \qquad (4.14)
\end{aligned}$$

where ϑ_1 follows by the assumption that $q_\phi(z|x)$ is factorial. As for $\mathrm{H}_\phi(\boldsymbol{z})$, we have:

$$\begin{aligned}
\mathrm{H}_\phi(\boldsymbol{z}) &= -\int_{\mathcal{Z}} q_\phi(z) \log q_\phi(z) dz \\
&= -\int_{\mathcal{Z}} q_\phi(z) \log \frac{q_\phi(z)}{\Pi_{k=1}^{K_2} q_\phi(z_k)} dz - \sum_{k=1}^{K_2} \int_{\mathcal{Z}} q_\phi(z) \log q_\phi(z_k) dz \\
&= -D_{\mathrm{KL}} \left(q_\phi(\boldsymbol{z}) || \Pi_{k=1}^{K_2} q_\phi(\boldsymbol{z}_k) \right) + \sum_{k=1}^{K_2} H_\phi(\boldsymbol{z}_k) \ . \qquad (4.15)
\end{aligned}$$

Substituting Equations (4.14) & (4.15) into Equation (4.13) yields the result:

$$\begin{aligned}
I_\phi(\boldsymbol{x}; \boldsymbol{z}) &= \sum_{k=1}^{K_2} H_\phi(\boldsymbol{z}_k) - D_{\mathrm{KL}} \left(q_\phi(\boldsymbol{z}) || \Pi_{k=1}^{K_2} q_\phi(\boldsymbol{z}_k) \right) - \sum_{k=1}^{K_2} H_\phi(\boldsymbol{z}_k|\boldsymbol{x}) \\
&= \sum_{k=1}^{K_2} I_\phi(\boldsymbol{x}; \boldsymbol{z}_k) - D_{\mathrm{KL}} \left(q_\phi(\boldsymbol{z}) || \Pi_{k=1}^{K_2} q_\phi(\boldsymbol{z}_k) \right) \ . \qquad (4.16)
\end{aligned}$$

Proof of Proposition 4.2.2

Proof. We start by computing the expectation of $\boldsymbol{z}_k$. Recall that $\boldsymbol{\mu}_k$ and $\boldsymbol{\sigma}_k$ are the mean and variance of the conditional distribution $q_\phi(\boldsymbol{z}_k|\boldsymbol{x})$ correspondingly. Note here $\boldsymbol{\mu}_k$ and $\boldsymbol{\sigma}_k$ are both random variables, with each pair values of $(\boldsymbol{\mu}_k, \boldsymbol{\sigma}_k)$ being associated with a specific value x, which we denote as (μ_k, σ_k). Let $\mathcal{X}$ and $\mathcal{Z}_k$ denote

the domains of $\boldsymbol{x}$ and $\boldsymbol{z}_k$ respectively, then

$$\begin{aligned}
\mathbb{E}_{q_\phi(\boldsymbol{z}_k)}\left[\boldsymbol{z}_k\right] &= \int_{\mathcal{Z}_k} z_k \int_{\mathcal{X}} q_\phi(z_k|x)q(x)dxdz_k \\
&= \int_{\mathcal{X}} q(x) \int_{\mathcal{Z}_k} z_k q_\phi(z_k|x)dz_k dx \\
&= \int_{\mathcal{X}} q(x)\mu_k(x)dx \\
&= \mathbb{E}_{\boldsymbol{x}}\left[\boldsymbol{\mu}_k|\boldsymbol{x}\right] \ . \qquad (4.17)
\end{aligned}$$

Then the variance of z_k followed as:

$$\begin{aligned}
\mathrm{Var}_\phi\left[\boldsymbol{z}_k\right] &= \int_{\mathcal{Z}_k} z_k^2 \int_{\mathcal{X}} q_\phi(z_k|x)q(x)dxdz_k - \mathbb{E}_{\boldsymbol{x}}\left[\boldsymbol{\mu}_k|\boldsymbol{x}\right]^2 \\
&= \int_{\mathcal{X}} q(x) \int_{\mathcal{Z}_k} z_k^2 q_\phi(z_k|x)dz_k dx - \mathbb{E}_{\boldsymbol{x}}\left[\boldsymbol{\mu}_k|\boldsymbol{x}\right]^2 \\
&\overset{\vartheta_1}{=} \int_{\mathcal{X}} q(x)\left[\sigma_k^2(x) + \mu_k(x)^2\right]dx - \mathbb{E}_{\boldsymbol{x}}\left[\boldsymbol{\mu}_k|\boldsymbol{x}\right]^2 \\
&= \mathbb{E}_{\boldsymbol{x}}\left[\boldsymbol{\sigma}_k^2|\boldsymbol{x}\right] + \mathrm{Var}_{\boldsymbol{x}}\left[\boldsymbol{\mu}_k|\boldsymbol{x}\right] \ , \qquad (4.18)
\end{aligned}$$

where ϑ_1 holds since $\mathbb{E}[\boldsymbol{z}_k^2|x] = \mu_k(x)^2 + \sigma_k^2(x)$. Next, note that

$$\mathrm{I}_\phi(\boldsymbol{x};\boldsymbol{z}_k) = \mathrm{H}_\phi(\boldsymbol{z}_k) - \mathrm{H}_\phi(\boldsymbol{z}_k|\boldsymbol{x}) \ , \qquad (4.19)$$

for which we have the following,

$$\begin{aligned}
\mathrm{H}_\phi(\boldsymbol{z}_k|\boldsymbol{x}) &= -\int_{\mathcal{X}} q(x) \int_{\mathcal{Z}_k} q_\phi(z_k|x)\log q_\phi(z_k|x)dz_k dx \\
&= \frac{1}{2}\int_{\mathcal{X}} q(x)\log\left(2\pi e\sigma_k^2(x)\right)dx \\
&= \frac{1}{2}\left(\log(2\pi e) + \mathbb{E}_{\boldsymbol{x}}\left[\log\boldsymbol{\sigma}_k^2|\boldsymbol{x}\right]\right). \qquad (4.20)
\end{aligned}$$

For the entropy of $\boldsymbol{z}_k$, we leverage the fact that $\mathrm{H}_\phi(\boldsymbol{z}_k)$ is upper bounded by the entropy of a Gaussian distributed random variable with the same variance, that is

$$\mathrm{H}_\phi(\boldsymbol{z}_k) \leq \frac{1}{2}\left(\log 2\pi e + \log\left(\mathbb{E}_{\boldsymbol{x}}\left[\boldsymbol{\sigma}_k^2|\boldsymbol{x}\right] + \mathrm{Var}_{\boldsymbol{x}}\left[\boldsymbol{\mu}_k|\boldsymbol{x}\right]\right)\right) \qquad (4.21)$$

Substituting equations (4.20) & (4.21) into equation (4.19) completes the proof. □

Proof of Proposition 4.2.1

Proof. Let x_n denote a sample of $\boldsymbol{x}$, and $\widehat{q}_\phi(\boldsymbol{y}) = \frac{1}{N}\sum_{n=1}^{N} q_\phi(\boldsymbol{y}|x_n)$ denote the Monte Carlo estimate of the true marginal distribution $q_\phi(\boldsymbol{y}) = \int_{\mathcal{X}} q(x)q_\phi(\boldsymbol{y}|x)dx = \mathbb{E}_{\boldsymbol{x}}\left[q_\phi(\boldsymbol{y}|\boldsymbol{x})\right]$. Note that $q_\phi(\boldsymbol{y}|\boldsymbol{x}) \in [0,1]$ for all $\boldsymbol{x} \in \mathcal{X}$, then applying the Hoeffding's inequality for bounded random variables [Theorem 2.2.6, [130]] yields,

$$\begin{aligned}\mathbb{P}\left(|\widehat{q}_\phi(\boldsymbol{y}) - q_\phi(\boldsymbol{y})| \geq t\right) &= \mathbb{P}\left(\left|\frac{1}{N}\sum_{n=1}^{N} q_\phi(\boldsymbol{y}|x_n) - \mathbb{E}_{\boldsymbol{x}}\left[q_\phi(\boldsymbol{y}|\boldsymbol{x})\right]\right| \geq t\right) \\ &\leq 2\exp\left(-2Nt^2\right) \qquad (4.22)\end{aligned}$$

Let $\delta' = 2\exp\left(-2Nt^2\right)$, it then follows,

$$\mathbb{P}\left(|\widehat{q}_\phi(\boldsymbol{y}) - q_\phi(\boldsymbol{y})| < \sqrt{\frac{\log(2/\delta')}{2N}}\right) \geq 1 - \delta' \qquad (4.23)$$

Given Eq (4.23), we first establish the concentration results of the entropy $H_{\widehat{q}_\phi}(\boldsymbol{y})$ with respect to the estimate $\widehat{q}_\phi(\boldsymbol{y})$. Assume for each $k \in \{1, \ldots, K_1\}$, we have $q_\phi(\boldsymbol{y} = k)$ and $\widehat{q}_\phi(\boldsymbol{y} = k)$ bounded below by $1/(CK_1)$ for some constant $C \geq 1$. Consider the function $t\log t$ whose derivative is $1 + \log t \in [1 - \log CK_1, 1]$, therefore

$$\begin{aligned}|\widehat{q}_\phi(\boldsymbol{y})\log\widehat{q}_\phi(\boldsymbol{y}) - q_\phi(\boldsymbol{y})\log q_\phi(\boldsymbol{y})| &= \left|\int_{q_\phi(\boldsymbol{y})}^{\widehat{q}_\phi(\boldsymbol{y})} (1 + \log t)dt\right| \\ &\leq \left|\int_{q_\phi(\boldsymbol{y})}^{\widehat{q}_\phi(\boldsymbol{y})} |1 + \log t|dt\right| \\ &\leq \left|\int_{q_\phi(\boldsymbol{y})}^{\widehat{q}_\phi(\boldsymbol{y})} \max\{\log CK_1 - 1, 1\}dt\right| \\ &\leq \max\{\log CK_1 - 1, 1\}|\widehat{q}_\phi(\boldsymbol{y}) - q_\phi(\boldsymbol{y})| \qquad (4.24)\end{aligned}$$

Summing over $k = 1, \ldots, K_1$ gives

$$\left|\widehat{H}_\phi(\boldsymbol{y}) - H_\phi(\boldsymbol{y})\right| \leq K_1 \max\{\log CK_1 - 1, 1\}|\widehat{q}_\phi(\boldsymbol{y}) - q_\phi(\boldsymbol{y})|\,. \qquad (4.25)$$

Let $\delta = K_1\delta'$, then Eq (4.23) together with Eq (4.25) yield the following,

$$\mathbb{P}\left(\left|\widehat{H}_\phi(\boldsymbol{y}) - H_\phi(\boldsymbol{y})\right| < K_1 \max\{\log CK_1 - 1, 1\}\sqrt{\frac{\log(2K_1/\delta)}{2N}}\right) \geq 1 - \delta \tag{4.26}$$

Next we are going to bound the divergence between $\widehat{H}_\phi(\boldsymbol{y}|\boldsymbol{x})$ and $H_\phi(\boldsymbol{y}|\boldsymbol{x})$ that are defined as the following,

$$\widehat{H}_\phi(\boldsymbol{y}|\boldsymbol{x}) = -\frac{1}{N}\sum_{n=1}^{N}\sum_{k=1}^{K_1} q_\phi(\boldsymbol{y} = k|x_n)\log q_\phi(\boldsymbol{y} = k|x_n),$$

$$H_\phi(\boldsymbol{y}|\boldsymbol{x}) = -\int_{\mathcal{X}} q(x)\sum_{k=1}^{K_1} q_\phi(\boldsymbol{y} = k|x)\log q_\phi(\boldsymbol{y} = k|x)dx\ .$$

Note that $h\log h \in [-1/e, 0]$ for all $h \in [0, 1]$, then again applying [Theorem 2.2.6, [130]] yields,

$$\mathbb{P}\left(\left|\frac{1}{N}\sum_{n=1}^{N} q_\phi(\boldsymbol{y}|x_n)\log q_\phi(\boldsymbol{y}|x_n) - \mathbb{E}_{\boldsymbol{x}}\left[q_\phi(\boldsymbol{y}|\boldsymbol{x})\log q_\phi(\boldsymbol{y}|\boldsymbol{x})\right]\right| < t\right) \leq 2\exp\left(-2t^2e^2N\right) \tag{4.27}$$

Following the similar arguments as before, let $\delta' = 2\exp\left(-2t^2e^2N\right)$, we have

$$\mathbb{P}\left(\left|\frac{1}{N}\sum_{n=1}^{N} q_\phi(\boldsymbol{y}|x_n)\log q_\phi(\boldsymbol{y}|x_n) - \mathbb{E}_{\boldsymbol{x}}\left[q_\phi(\boldsymbol{y}|\boldsymbol{x})\log q_\phi(\boldsymbol{y}|\boldsymbol{x})\right]\right| < \sqrt{\frac{e^2\log(2/\delta')}{2N}}\right) \leq \delta' \tag{4.28}$$

Now let $\delta = K_1\delta'$, applying union bound yields,

$$\begin{aligned}|\widehat{H}_\phi(\boldsymbol{y}|\boldsymbol{x}) - H_\phi(\boldsymbol{y}|\boldsymbol{x})| &\leq \sum_{k=1}^{K_1}\left|\frac{1}{N}\sum_{n=1}^{N} q_\phi(\boldsymbol{y}|x_n)\log q_\phi(\boldsymbol{y}|x_n) - \mathbb{E}_{\boldsymbol{x}}\left[q_\phi(\boldsymbol{y}|\boldsymbol{x})\log q_\phi(\boldsymbol{y}|\boldsymbol{x})\right]\right| \\ &\leq K_1\sqrt{\frac{e^2\log(2K_1/\delta)}{2N}} \end{aligned} \tag{4.29}$$

hold with probability $1 - \delta$.

Concluding from equations (4.26) & (4.29), with probability exceeding $1 - 2\delta$, we have

$$\begin{aligned}\left|I_\phi(\boldsymbol{x};\boldsymbol{y}) - \widehat{I}_\phi(\boldsymbol{x};\boldsymbol{y})\right| &\leq \left|H_\phi(\boldsymbol{y}) - \widehat{H}_\phi(\boldsymbol{y})\right| + \left|H_\phi(\boldsymbol{y}|\boldsymbol{x}) - \widehat{H}_\phi(\boldsymbol{y}|\boldsymbol{x})\right| \\ &= K_1\left(\max\{\log CK_1 - 1, 1\} + e\right)\sqrt{\frac{\log(2K_1/\delta)}{N}}\ . \end{aligned} \tag{4.30}$$

□

Proof of Equation (4.9) To see this, note that $\mathrm{I}(\boldsymbol{y}_{\mathbf{true}}; \boldsymbol{z}) = \mathrm{I}(\boldsymbol{x}; \boldsymbol{z}) - \mathrm{I}(\boldsymbol{x}; \boldsymbol{z}|\boldsymbol{y}) + \mathrm{I}(\boldsymbol{z}; \boldsymbol{y}_{\mathbf{true}}|\boldsymbol{x})$. As for $\mathrm{I}(\boldsymbol{z}; \boldsymbol{y}_{\mathbf{true}}|\boldsymbol{x})$, we have,

$$\mathrm{I}(\boldsymbol{z}; \boldsymbol{y}_{\mathbf{true}}|\boldsymbol{x}) = \mathrm{H}(\boldsymbol{z}|\boldsymbol{x}) - \mathrm{H}(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{y}_{\mathbf{true}}) = \mathrm{H}(\boldsymbol{z}|\boldsymbol{x}) - \mathrm{H}(\boldsymbol{z}|\boldsymbol{x}) = 0 \tag{4.31}$$

It then follows,

$$\begin{aligned} \mathrm{I}(\boldsymbol{y}_{\mathbf{true}}; \boldsymbol{z}) &= \mathrm{I}(\boldsymbol{x}; \boldsymbol{z}) - \mathrm{I}(\boldsymbol{x}; \boldsymbol{z}|\boldsymbol{y}_{\mathbf{true}}) \\ &= \mathrm{I}(\boldsymbol{x}; \boldsymbol{z}) - \mathrm{H}(\boldsymbol{x}|\boldsymbol{y}_{\mathbf{true}}) + \mathrm{H}(\boldsymbol{x}|\boldsymbol{z}, \boldsymbol{y}_{\mathbf{true}}) \\ &\overset{\vartheta_1}{\leq} \mathrm{I}(\boldsymbol{x}; \boldsymbol{z}) - \mathrm{H}(\boldsymbol{x}|\boldsymbol{y}_{\mathbf{true}}) - \mathbb{E}_{q(\boldsymbol{y},\boldsymbol{z}|\boldsymbol{x})q(\boldsymbol{x})}[\log p(\boldsymbol{x}|\boldsymbol{y}, \boldsymbol{z})] \end{aligned} \tag{4.32}$$

where ϑ_1 holds since $-\mathbb{E}_{q(\boldsymbol{y},\boldsymbol{z}|\boldsymbol{x})q(\boldsymbol{x})}[\log p(\boldsymbol{x}|\boldsymbol{y}, \boldsymbol{z})] = \mathrm{H}(\boldsymbol{x}|\boldsymbol{y}_{\mathbf{true}}, \boldsymbol{z}) + D_{\mathrm{KL}}\left[q(\boldsymbol{x}|\boldsymbol{y}, \boldsymbol{z})||p(\boldsymbol{x}|\boldsymbol{y}, \boldsymbol{z})\right]$. Notice that, $\mathrm{H}(\boldsymbol{x}|\boldsymbol{y}_{\mathbf{true}})$ is a constant, and $\mathbb{E}_{q(\boldsymbol{y},\boldsymbol{z}|\boldsymbol{x})q(\boldsymbol{x})}[\log p(\boldsymbol{x}|\boldsymbol{y}, \boldsymbol{z})]$ is the *negative reconstruction error*. Therefore, the informativeness of $\boldsymbol{z}$ about the underlying categories $\boldsymbol{y}_{\mathbf{true}}$ of data can be implicitly decreased by penalizing $\mathrm{I}(\boldsymbol{x}; \boldsymbol{z})$ as well as the reconstruction error.

4.7.2 Approximation of the Marginal Distribution

Computing the marginal distributions of the continuous representation $\boldsymbol{z}$ and its k^{th} component $\boldsymbol{z}_k$ requires the entire dataset, *e.g.,* $q_\phi(\boldsymbol{z}) = \int_{\mathcal{X}} q_\phi(\boldsymbol{z}, x)dx \approx \frac{1}{N}\sum_{n=1}^{N} q_\phi(\boldsymbol{z}|x_n)$ with x_n being a sample of $\boldsymbol{x}$. To scale up our method to large datasets, we propose to estimate based on the minibatches data, *e.g.,* $q_\phi(\boldsymbol{z}) \approx \frac{1}{B}\sum_{b=1}^{B} q_\phi(\boldsymbol{z}|x_b)$.

Now consider the entropy $\mathrm{H}_\phi(\boldsymbol{z})$, which we approximate in the following way,

$$\begin{aligned} \mathrm{H}_\phi(\boldsymbol{z}) = \mathbb{E}_{\boldsymbol{z}}[\log q_\phi(\boldsymbol{z})] &\approx \frac{1}{B}\sum_{b=1}^{B} \log q_\phi(z_b) \\ &= \frac{1}{B}\sum_{b=1}^{B} \log \frac{1}{B}\sum_{b'=1}^{B} q_\phi(z_b|x_{b'}) \ . \end{aligned} \tag{4.33}$$

Here both b and b' are enumerated within the same minibath. Other quantities involved in our objective are estimated in a similar fashion.

4.7.3 Experimental Settings

Training procedure: We use Adam to train all models with learning rate $1e-3$.

- MNIST&Fashion MNIST:
  - epochs: 100
  - batch size: 1024
- dSprites:
  - epochs: 50
  - batch size: 2048
- CelebA:
  - epochs: 50
  - batch size: 512

Network architecture: The networks used by us for each dataset are summarized below, which are all implemented in PyTorch.

Table 4.1: Encoder and Decoder architecture for MNIST and Fashion MNIST.

Encoder			
Input vectorized 28×28 grayscale image			
	#Input	#Output	Activation function
Layer 1	784	500	ReLU
Layer 2	500	500	ReLU
Layer 3	500	20 $[10(\mu) + 10(\log \sigma)]$	Linear
	500	10 $[p(y)]$	Softmax

Decoder			
Input concatenated representation $\left(\boldsymbol{y} \in \mathbb{R}^{10}, \boldsymbol{z} \in \mathbb{R}^{10}\right)$			
	#Input	#Output	Activation function
Layer 1	20	500	ReLU
Layer 2	500	500	ReLU
Layer 3	500	784	Sigmoid

Table 4.2: Encoder and Decoder architecture for dSprites. For this dataset, we adopt the network architecture used in [35].

Encoder			
Input vectorized 64×64 grayscale image			
	#Input	#Output	Activation function
Layer 1	4096	1200	ReLU
Layer 2	1200	1200	ReLU
Layer 3	1200	20 $[10(\mu) + 10(\log \sigma)]$	Linear
	1200	3 $[p(y)]$	Softmax

Decoder			
Input concatenated representation $\left(\boldsymbol{y} \in \mathbb{R}^{3}, \boldsymbol{z} \in \mathbb{R}^{10}\right)$			
	#Input	#Output	Activation function
Layer 1	13	1200	Tanh
Layer 2	1200	1200	Tanh
Layer 3	1200	1200	Tanh
Layer 4	1200	4096	Sigmoid

Table 4.3: Encoder and Decoder architecture for CelebA.

Encoder			
Input 64×64 RGB image			
Layers	Output $w \times h$	#Channels in&out	Activation function
Conv 1	32×32	3, 32	ReLU
Conv 2	16×16	32, 32	ReLU
Conv 3	8×8	32, 32	ReLU
Conv 4	4×4	32, 32	ReLU
FC 5	1	512, 256	ReLU
FC 6	256	64 $[32(\mu) + 32(\log \sigma)]$	Linear
	256	10 $[p(y)]$	Softmax

Decoder			
Input concatenated representation $\left(\boldsymbol{y} \in \mathbb{R}^{10}, \boldsymbol{z} \in \mathbb{R}^{32}\right)$			
Layers	Output $w \times h$	#Channels in&out	Activation function
FC 1	1	42, 256	ReLU
Conv 2	4×4	256, 64	ReLU
Conv 3	8×8	64, 64	ReLU
Conv 4	16×16	64, 32	ReLU
Conv 5	32×32	32, 32	ReLU
Conv 5	64×64	32, 3	Sigmoid

CHAPTER V

Conclusion and Future Work

In this thesis, we target three different machine learning problems to tackle the challenges caused by the increasing complexity of both modern data and models. Although each presented approach is motivated from a different perspective, the underlying goal can be unified as extracting compact knowledge from data by leveraging the structure hidden in data. We now conclude with a brief summary of each chapter and discussions on the associated future directions.

5.1 Convergence of GROUSE for Both Fully Sampled and Undersampled Data

In Chapter II, we analytically study the convergence behavior of the GROUSE algorithm, and provide the first global convergence result for an incremental gradient descent method on the Grassmannian for fully sampled noise-free data. For the case of undersampled data, we establish monotonic expected improvement on the defined convergence metric for each iteration with high probability. To further narrow the gap between our global convergence and the actual convergence behavior of GROUSE, we propose a conjecture where we divide the convergence into two phases: the initial phase and the local phase. In the initial phase, we propose to use a different analysis strategy to bound the convergence rate more tighter. Through we do empirically

validate our conjecture on the global convergence of fully sampled data, establishing global convergence through more rigorous analysis is a very important direction for future work.

Another avenue of future work is to establish the convergence results for undersampled data. As is indicated by Lemma 2.5.2, the main hurdle to establish the global convergence result is the perturbation induced by the undersampling framework. A natural question hence arises as whether we can cross the hurdles in analysis by using a new sampling strategy? Our expectations are, with the new sampling strategy, we can resolve the problems caused by the typical undersampling framework, *i.e.,* subsampling the input vectors uniformly at random [missing data case, Section 2.5.2] or taking isotropic random Gaussian measurements [compressively sampled data, Section 2.5.1]. Inspired by the recent works on compressed sensing of sparse vectors [67, 10] and low-rank matrix completion [38, 54], adaptive sampling can be a promising direction of future work.

Last but not the least, extending our current analysis to noisy data would be of great practical and theoretical interest. With noiseless data, we propose a greedy step size scheme so as to maximally include the information in the projection residual for each update step of GROUSE. However, such a strategy can lead to worse convergence in the existence of noise, since the noise part will gradually dominate the projection residual as our estimate gets closer to the true subspace. Therefore, with noisy data, in order to incorporate less and less of the residual information into our estimate over time, a different step size scheme is required. A natural resolution can be to impose a diminishing weight in front of our current step size scheme for the noiseless data. By doing so, we can maximally incorporate the information in the projection residual, while gradually leaving out the information of noise.

In [143], we propose a step size regimen for fully sampled data, which is simply a weighted version of the step size for noise-free data, where the weights depend on the data and noise statistics. It would be interesting to develop similar strategy for undersampled data, in particularly for the cases where the noise (outliers) is sparse but can be of arbitrarily large magnitude.

5.2 Simultaneous Sparsity and Parameter Tying for Neural Networks Compression

In Chapter III, we propose using the recent GrOWL regularizer for simultaneous parameter sparsity and tying in DNN (Deep Neural Network) learning. Unlike the conventional sparsity-inducing regularizers, GrOWL simultaneously eliminates unimportant variables by setting their weights to zero, while also explicitly identifying highly correlated groups of variables by tying the corresponding weights to be very close or exactly equal to each other. Instead of defining a prior to decide which sets of parameters should be enforced to share a common value, GrOWL learns the parameter sharing structure from the data itself. This ability of GrOWL not only allows us more space for neural network compression but also helps us cope with strong correlations that might be induced by the noisy input or the co-adaptation tendency of DNNs.

As we numerically demonstrate in Chapter III, the correlation patterns identified by GrOWL are close to those of the input features to each layer. Therefore, many interesting directions can be explored by leveraging the correlation identification ability of GrOWL. Among these, the most exciting one would be, using GrOWL to identify and better understand the intrinsic correlations among the selected sparse features, contributing to revealing the structure of features and their relations with the corresponding task. By doing so, we can probably take a step towards improving

the interpretability of deep learning models. Moreover, instead of the hard-wired parameter sharing framework used by the convolutional neural network, GrOWL allows us to learn the parameter sharing structure from the data. Exploring whether GrOWL could give us a interpretable yet useful parameter sharing framework would be of great practical interest.

Another exciting direction is exploring the possibility of leveraging the denoising effect of GrOWL to improve the vulnerability of DNNs in the existence of adversarial examples. Adversarial examples can be obtained by perturbing the input data with small norm-bounded noise, which can induce strong correlations among the data. By leveraging GrOWL's ability in identifying such correlations and tying the associated parameters together, we can effectively denoise the input data to improve the vulnerability of DNNs, instead of being negatively affected by it.

Finally, as is discussed in the numerical section of Chapter III, the gap in the accuracy versus memory trade-off obtained by applying GrOWL and group-Lasso decreases as we move to large DNNs. Although we suspect this can be caused by running a much larger network on a simple dataset, it motivates us to explore different ways to apply GrOWL to compress neural networks. One possible approach for future work is to apply GrOWL within each neuron by predefining each 2D convolutional filter as a group (instead all 2D convolutional filters corresponding to the same input features). By doing so, we encourage parameter sharing among much smaller units, which in turn would further improve the diversity vs. parameter sharing trade-off.

5.3 Regularized Information Maximization AutoEncoding

In Chapter IV, we present an information-theoretical approach for disentangled representation learning in the setting of unsupervised learning, with the emphasis on

jointly learning a hybrid continuous and discrete representation. We show that the proposed RIMAE method is capable of discovering the underlying categories of data, while simultaneously identifying and separating the other explanatory factors hidden in data into disjoint part of the learned continuous representation. Building upon the great promise of RIMAE, we briefly discuss a few interesting future directions to improve the current method further.

First of all, maximizing the mutual information $\mathrm{I}(\boldsymbol{x}; \boldsymbol{y})$ between data $\boldsymbol{x}$ and its discrete representation $\boldsymbol{y}$ alone is not good enough to guarantee successful recovery of the underlying categories of data. Notice that, the mutual information $\mathrm{I}(\boldsymbol{x}; \boldsymbol{y})$ can be trivially maximized by uniformly distributing the whole data over all categories, while randomly assigning each data sample to one category without guaranteeing similar samples being assigned to the same category. Although the reconstruction criterion is supposed to prevent such useless discrete representation, given the existence of the continuous representation, it's easy for the model to ignore the discrete representation, since its capacity in determining the reconstruction fidelity is typically much smaller than that of the continuous representation. To remedy this issue, further constraints are required for learning useful discrete representation. Among these, a promising one is to enforce local smoothness of the model, which can potentially remove the flaw of our current model by enforcing each data sample and its perturbation, obtained by adding small norm-bounded noise to each data sample, being assigned to the same category.

Moreover, RIMAE assumes data is uniformly distributed over all categories. However, real-world data may contain imbalanced categories. In this setting, directly maximizing the mutual information $\mathrm{I}(\boldsymbol{x}; \boldsymbol{y})$ between data and its categorical representation is not proper, since optimizing $\mathrm{I}(\boldsymbol{x}; \boldsymbol{y})$ tends to push the distribution

of $\boldsymbol{y}$ towards uniform distribution. One possible resolution would be replacing the categorical balance term $\mathrm{H}(\boldsymbol{y})$ in $\mathrm{I}(\boldsymbol{x};\boldsymbol{y})$ with a cross entropy loss $\mathrm{H}(q(\boldsymbol{y});p(\boldsymbol{y}))$ where $p(\boldsymbol{y})$ denote the prior of the distribution of data over its categories.

Given categorical data that exhibit similar variations for each group, *e.g.,* MNIST shares the same writing style for each digit. It would be of great practical interest to explicitly encourage the statistical independence between the discrete representation and the continuous one. By doing so, we can obtain disentangled representation that is more faithful to the underlying latent factors of data, *i.e.,* successfully uncovering the categories of data, while simultaneously disentangling the continuous representation factors with each factor capturing a different variation in data, shared over categories.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. Corrado, A. Davis, J. Dean, M. Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv:1603.04467*, 2016.

[2] P-A Absil, Robert Mahony, and Rodolphe Sepulchre. *Optimization algorithms on matrix manifolds*. Princeton University Press, 2009.

[3] Alessandro Achille and Stefano Soatto. Emergence of invariance and disentanglement in deep representations. *The Journal of Machine Learning Research*, 19(1):1947–1980, 2018.

[4] Alessandro Achille and Stefano Soatto. Information dropout: Learning optimal representations through noisy computation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.

[5] A. Aghasi, N. Nguyen, and J. Romberg. Net-Trim: A layer-wise convex pruning of deep neural networks. *arXiv:1611.05162*, 2016.

[6] Alexander A Alemi, Ian Fischer, Joshua V Dillon, and Kevin Murphy. Deep variational information bottleneck. *arXiv preprint arXiv:1612.00410*, 2016.

[7] Alexander A Alemi, Ben Poole, Ian Fischer, Joshua V Dillon, Rif A Saurous, and Kevin Murphy. An information-theoretic analysis of deep latent-variable models. *arXiv preprint arXiv:1711.00464*, 2017.

[8] J. Alvarez and M. Salzmann. Learning the number of neurons in deep networks. In *Advances in Neural Information Processing Systems*, pages 2270–2278, 2016.

[9] Babak Ardekani, Jeff Kershaw, Kenichi Kashikura, Iwao Kanno, et al. Activation detection in functional mri using subspace modeling and maximum likelihood estimation. *Medical Imaging, IEEE Transactions on*, 18(2):101–114, 1999.

[10] Ery Arias-Castro, Emmanuel J Candes, and Mark A Davenport. On the fundamental limits of adaptive sensing. *IEEE Transactions on Information Theory*, 59(1):472–481, 2013.

[11] Diego Armentano, Carlos Beltrán, and Michael Shub. Average polynomial time for eigenvector computations. *arXiv preprint arXiv:1410.2179*, 2014.

[12] Raman Arora, Andy Cotter, and Nati Srebro. Stochastic optimization of PCA with capped MSG. In *Advances in Neural Information Processing Systems*, pages 1815–1823, 2013.

[13] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[14] Akshay Balsubramani, Sanjoy Dasgupta, and Yoav Freund. The fast convergence of incremental PCA. In *Advances in Neural Information Processing Systems*, pages 3174–3182, 2013.

[15] Laura Balzano, Robert Nowak, and Benjamin Recht. Online identification and tracking of subspaces from highly incomplete information. In *Communication, Control, and Computing (Allerton), 2010 48th Annual Allerton Conference on*, pages 704–711. IEEE, 2010.

[16] Laura Balzano, Benjamin Recht, and Robert Nowak. High-dimensional matched subspace detection when data are missing. In *2010 IEEE International Symposium on Information Theory*, pages 1638–1642. IEEE, 2010.

[17] Laura Balzano and Stephen J Wright. Local convergence of an algorithm for subspace identification from partial data. *Foundations of Computational Mathematics*, pages 1–36, 2014.

[18] Laura Kathryn Balzano. *Handling missing data in high-dimensional subspace modeling*. PhD thesis, UNIVERSITY OF WISCONSIN–MADISON, 2012.

[19] David Barber and Felix Agakov. The im algorithm: a variational approach to information maximization. In *Proceedings of the 16th International Conference on Neural Information Processing Systems*, pages 201–208. MIT Press, 2003.

[20] H. Bauschke and P. Combettes. *Convex analysis and monotone operator theory in Hilbert spaces*. Springer, 2011.

[21] Anthony J Bell and Terrence J Sejnowski. An information-maximization approach to blind separation and blind deconvolution. *Neural computation*, 7(6):1129–1159, 1995.

[22] Y. Bengio and J. Bergstra. Slow, decorrelated features for pretraining complex cell-like networks. In *Advances in neural information processing systems*, pages 99–107, 2009.

[23] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

[24] Dimitri P Bertsekas. Incremental gradient, subgradient, and proximal methods for convex optimization: A survey. *Optimization for Machine Learning*, 2010(1-38):3, 2011.

[25] Srinadh Bhojanapalli, Anastasios Kyrillidis, and Sujay Sanghavi. Dropping convexity for faster semi-definite optimization. *arXiv preprint arXiv:1509.03917*, 2015.

[26] M. Bogdan, E. van den Berg, C. Sabatti, W. Su, and E. Candès. Slope–adaptive variable selection via convex optimization. *The annals of applied statistics*, 9(3):1103, 2015.

[27] H. Bondell and B. Reich. Simultaneous regression shrinkage, variable selection, and supervised clustering of predictors with oscar. *Biometrics*, 64(1):115–123, 2008.

[28] John S Bridle, Anthony JR Heading, and David JC MacKay. Unsupervised classifiers, mutual information and'phantom targets. In *Advances in Neural Information Processing Systems*, pages 1096–1101, 1992.

[29] J Paul Brooks, JH Dulá, and Edward L Boone. A pure l1-norm principal component analysis. *Computational statistics & data analysis*, 61:83–98, 2013.

[30] P. Bühlmann, P. Rütimann, S. van de Geer, and C.-H. Zhang. Correlated variables in regression: clustering and sparse estimation. *Journal of Statistical Planning and Inference*, 143:1835–1871, 2013.

[31] L. Buitinck, G. Louppe, M. Blondel, F. Pedregosa, A. Mueller, O. Grisel, V. Niculae, P. Prettenhofer, A. Gramfort, J. Grobler, R. Layton, J. VanderPlas, A. Joly, B. Holt, and G. Varoquaux. API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pages 108–122, 2013.

[32] Christopher P Burgess, Irina Higgins, Arka Pal, Loic Matthey, Nick Watters, Guillaume Desjardins, and Alexander Lerchner. Understanding disentangling in β-vae. *arXiv preprint arXiv:1804.03599*, 2018.

[33] Emmanuel J Candès, Xiaodong Li, Yi Ma, and John Wright. Robust principal component analysis? *Journal of the ACM (JACM)*, 58(3):11, 2011.

[34] Emmanuel J Candès and Benjamin Recht. Exact matrix completion via convex optimization. *Foundations of Computational mathematics*, 9(6):717–772, 2009.

[35] Tian Qi Chen, Xuechen Li, Roger Grosse, and David Duvenaud. Isolating sources of disentanglement in variational autoencoders. *arXiv preprint arXiv:1802.04942*, 2018.

[36] W. Chen, J. Wilson, S. Tyree, K. Weinberger, and Y. Chen. Compressing neural networks with the hashing trick. In *Proceedings of ICML*, 2015.

[37] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2172–2180, 2016.

[38] Yudong Chen, Srinadh Bhojanapalli, Sujay Sanghavi, and Rachel Ward. Completing any low-rank matrix, provably. *arXiv preprint arXiv:1306.2979*, 2013.

[39] Yudong Chen and Martin J Wainwright. Fast low-rank estimation by projected gradient descent: General statistical and algorithmic guarantees. *arXiv preprint arXiv:1509.03025*, 2015.

[40] C. Clark and A. Storkey. Teaching deep convolutional neural networks to play Go. In *Proceedings of ICML*, 2015.

[41] M. Cogswell, F. Ahmed, R. Girshick, L. Zitnick, and D. Batra. Reducing overfitting in deep networks by decorrelating representations. *arXiv:1511.06068*, 2015.

[42] M. Collins and P. Kohli. Memory bounded deep convolutional networks. *arXiv:1412.1442*, 2014.

[43] Thomas M Cover and Joy A Thomas. *Elements of information theory*. John Wiley & Sons, 2012.

[44] Alexandre d'Aspremont, Francis Bach, and Laurent El Ghaoui. Optimal solutions for sparse principal component analysis. *The Journal of Machine Learning Research*, 9:1269–1294, 2008.

[45] J. de Leeuw, K. Hornik, and Patrick Mair. Isotone optimization in R: Pool adjacent-violators algorithm (PAVA) and active set methods. *Statistical Software*, 32:1–24, 2009.

[46] Christopher D De Sa, Christopher Re, and Kunle Olukotun. Global convergence of stochastic gradient descent for some non-convex matrix problems. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pages 2332–2341, 2015.

[47] M. Denil, B. Shakibi, L. Dinh, N. de Freitas, et al. Predicting parameters in deep learning. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2148–2156, 2013.

[48] E. Denton, W. Zaremba, J. Bruna, Y. LeCun, and R. Fergus. Exploiting linear structure within convolutional networks for efficient evaluation. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1269–1277, 2014.

[49] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[50] S. Dieleman, J. De Fauw, and K. Kavukcuoglu. Exploiting cyclic symmetry in convolutional neural networks. *Proceedings of ICML*, 2016.

[51] Simon S Du, Jason D Lee, Haochuan Li, Liwei Wang, and Xiyu Zhai. Gradient descent finds global minima of deep neural networks. *arXiv preprint arXiv:1811.03804*, 2018.

[52] Emilien Dupont. Joint-vae: Learning disentangled joint continuous and discrete representations. *arXiv preprint arXiv:1804.00104*, 2018.

[53] Alan Edelman, Tomás A Arias, and Steven T Smith. The geometry of algorithms with orthogonality constraints. *SIAM journal on Matrix Analysis and Applications*, 20(2):303–353, 1998.

[54] Armin Eftekhari, Michael B Wakin, and Rachel A Ward. Mcˆ 2: A two-phase algorithm for leveraged matrix completion. *arXiv preprint arXiv:1609.01795*, 2016.

[55] Babak Esmaeili, Hao Wu, Sarthak Jain, Siddharth Narayanaswamy, Brooks Paige, and Jan-Willem van de Meent. Hierarchical disentangled representations. *arXiv preprint arXiv:1804.02086*, 2018.

[56] M. Figueiredo and R. Nowak. Ordered weighted l1 regularized regression with strongly correlated covariates: Theoretical aspects. In *Proceedings of AISTATS*, pages 930–938, 2016.

[57] B. J Frey and D. Dueck. Clustering by passing messages between data points. *Science*, 315(5814):972–976, 2007.

[58] Shuyang Gao, Rob Brekelmans, Greg Ver Steeg, and Aram Galstyan. Auto-encoding total correlation explanation. *arXiv preprint arXiv:1802.05822*, 2018.

[59] Gene H Golub and Charles F Van Loan. *Matrix computations*. JHU Press, 4 edition, 2012.

[60] Y. Gong, L. Liu, M. Yang, and L. Bourdev. Compressing deep convolutional networks using vector quantization. *arXiv:1412.6115*, 2014.

[61] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016.

[62] Ian Goodfellow. Nips 2016 tutorial: Generative adversarial networks. *arXiv preprint arXiv:1701.00160*, 2016.

[63] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. pages 2672–2680, 2014.

[64] S. Han, H. Mao, and W. Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization, and Huffman coding. In *Proceedings of ICLR*, 2016.

[65] Song Han, Jeff Pool, John Tran, and William Dally. Learning both weights and connections for efficient neural networks. In *Advances in neural information processing systems*, 2015.

[66] B. Hassibi and D Stork. Second order derivatives for network pruning: Optimal brain surgeon. *Advances in Neural Information Processing Systems (NIPS)*, pages 164–164, 1993.

[67] Jarvis Haupt, Rui M Castro, and Robert Nowak. Distilled sensing: Adaptive sampling for sparse detection and estimation. *IEEE Transactions on Information Theory*, 57(9):6222–6235, 2011.

[68] Jun He, Laura Balzano, and Arthur Szlam. Incremental gradient on the grassmannian for online foreground and background separation in subsampled video. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1568–1575. IEEE, 2012.

[69] Jun He, Dejiao Zhang, Laura Balzano, and Tao Tao. Iterative grassmannian optimization for robust image alignment. *Image and Vision Computing*, 32(10):800–813, 2014.

[70] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of IEEE CVPR*, pages 770–778, 2016.

[71] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. 2016.

[72] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. In *International Conference on Learning Representations*, 2019.

[73] Matthew D Hoffman and Matthew J Johnson. Elbo surgery: yet another way to carve up the variational evidence lower bound. In *Workshop in Advances in Approximate Bayesian Inference, NIPS*, 2016.

[74] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018.

[75] Max Jaderberg, Volodymyr Mnih, Wojciech Marian Czarnecki, Tom Schaul, Joel Z Leibo, David Silver, and Koray Kavukcuoglu. Reinforcement learning with unsupervised auxiliary tasks. *arXiv preprint arXiv:1611.05397*, 2016.

[76] Prateek Jain, Chi Jin, Sham M Kakade, Praneeth Netrapalli, and Aaron Sidford. Streaming pca: Matching matrix bernstein and near-optimal finite sample guarantees for ojaâĂŹs algorithm. In *29th Annual Conference on Learning Theory*, pages 1147–1164, 2016.

[77] Prateek Jain, Praneeth Netrapalli, and Sujay Sanghavi. Low-rank matrix completion using alternating minimization. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 665–674. ACM, 2013.

[78] Raghunandan H Keshavan, Andrea Montanari, and Sewoong Oh. Matrix completion from a few entries. *Information Theory, IEEE Transactions on*, 56(6):2980–2998, 2010.

[79] Raghunandan Hulikal Keshavan. *Efficient algorithms for collaborative filtering*. PhD thesis, Stanford University, 2012.

[80] Hyunjik Kim and Andriy Mnih. Disentangling by factorising. *arXiv preprint arXiv:1802.05983*, 2018.

[81] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[82] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, (8):30–37, 2009.

[83] Andreas Krause, Pietro Perona, and Ryan G Gomes. Discriminative clustering by regularized information maximization. In *Advances in neural information processing systems*, pages 775–783, 2010.

[84] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[85] Y. LeCun. *Modèles connexionnistes de l'apprentissage*. PhD thesis, Université Pierre et Marie Curie, Paris, France, 1987.

[86] Y. LeCun, J. Denker, S. Solla, R. Howard, and L. Jackel. Optimal brain damage. In *Advances in Neural Information Processing Systems (NIPS)*, volume 2, pages 598–605, 1989.

[87] Yann LeCun and Corinna Cortes. MNIST handwritten digit database. 2010.

[88] Kiryung Lee, Yoram Bresler, H Munthe-Kaas, A Lundervold, S Gaubert, M Sharify, CJ Cotter, M Colombeau, M Hutzenthaler, A Jentzen, et al. Admira: Atomic decomposition for minimum rank approximation. 2009. *arXiv preprint arXiv:0905.0044*.

[89] H. Li, A. Kadav, I. Durdanovic, H. Samet, and H. Graf. Pruning filters for efficient convnets. *arXiv:1608.08710*, 2016.

[90] Ralph Linsker. Self-organization in a perceptual network. *Computer*, 21(3):105–117, 1988.

[91] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2015.

[92] Nate Kushman Ryota Tomioka Sebastian Nowozin Mary Phuong, Max Welling. The mutual autoencoder: Controlling information in latent code representations. 2018.

[93] Loic Matthey, Irina Higgins, Demis Hassabis, and Alexander Lerchner. dsprites: Disentanglement testing sprites dataset. https://github.com/deepmind/dsprites-dataset/, 2017.

[94] Colin McDiarmid. On the method of bounded differences. *Surveys in combinatorics*, 141(1):148–188, 1989.

[95] Riccardo Miotto, Fei Wang, Shuang Wang, Xiaoqian Jiang, and Joel T Dudley. Deep learning for healthcare: review, opportunities and challenges. *Briefings in bioinformatics*, 19(6):1236–1246, 2017.

[96] Andriy Mnih and Karol Gregor. Neural variational inference and learning in belief networks. *arXiv preprint arXiv:1402.0030*, 2014.

[97] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pages 1928–1937, 2016.

[98] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

[99] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529, 2015.

[100] K. Murray and D. Chiang. Auto-sizing neural networks: With applications to n-gram language models. *arXiv:1508.05051*, 2015.

[101] Andrew Ng et al. Sparse autoencoder. *CS294A Lecture notes*, 72(2011):1–19, 2011.

[102] Thanh Ngo and Yousef Saad. Scaled gradients on grassmann manifolds for matrix completion. In *Advances in Neural Information Processing Systems*, pages 1412–1420, 2012.

[103] Hoi H Nguyen, Van Vu, et al. Random matrices: Law of the determinant. *The Annals of Probability*, 42(1):146–167, 2014.

[104] U. Oswal, C. Cox, M. Lambon-Ralph, T. Rogers, and R. Nowak. Representational similarity learning with application to brain networks. In *Proceedings of ICML*, pages 1041–1049, 2016.

[105] Yigang Peng, Arvind Ganesh, John Wright, Wenli Xu, and Yi Ma. Rasl: Robust alignment by sparse and low-rank decomposition for linearly correlated images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(11):2233–2246, 2012.

[106] Benjamin Recht, Maryam Fazel, and Pablo A Parrilo. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM review*, 52(3):471–501, 2010.

[107] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. *arXiv preprint arXiv:1401.4082*, 2014.

[108] R.H.Keshavan. *Efficient algorithms for collaborative filtering*. PhD thesis, Stanford University, 2012.

[109] P. Rodríguez, J. Gonzàlez, G. Cucurull, J. Gonfaus, and X. Roca. Regularizing cnns with locally constrained decorrelations. *arXiv:1611.01967*, 2016.

[110] Paul K Rubenstein, Bernhard Schoelkopf, and Ilya Tolstikhin. On the latent space of wasserstein auto-encoders. *arXiv preprint arXiv:1802.03761*, 2018.

[111] D. Rumelhart, G. Hinton, and R. Williams. Learning representations by back-propagating errors. *Nature*, 323:533–536, 1986.

[112] D. Rumelhart and J. McClelland. *Parallel Distributed Processing*. MIT Press, 1986.

[113] Andrew Michael Saxe, Yamini Bansal, Joel Dapello, Madhu Advani, Artemy Kolchinsky, Brendan Daniel Tracey, and David Daniel Cox. On the information bottleneck theory of deep learning. 2018.

[114] S. Scardapane, D. Comminiello, A. Hussain, and A. Uncini. Group sparse regularization for deep neural networks. *Neurocomputing*, 241:81–89, 2017.

[115] Ohad Shamir, Sivan Sabato, and Naftali Tishby. Learning and generalization with the information bottleneck. *Theoretical Computer Science*, 411(29-30):2696–2711, 2010.

[116] Ravid Shwartz-Ziv and Naftali Tishby. Opening the black box of deep neural networks via information. *arXiv preprint arXiv:1703.00810*, 2017.

[117] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484, 2016.

[118] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv:1409.1556*, 2014.

[119] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[120] Gilbert W Stewart and Ji-guang Sun. *Matrix perturbation theory*. Academic press, 1990.

[121] Stanislaw J Szarek. Metric entropy of homogeneous spaces. *arXiv preprint math/9701213*, 1997.

[122] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[123] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[124] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society (B)*, 58:267–288, 1996.

[125] Naftali Tishby, Fernando C Pereira, and William Bialek. The information bottleneck method. *arXiv preprint physics/0004057*, 2000.

[126] Naftali Tishby and Noga Zaslavsky. Deep learning and the information bottleneck principle. In *2015 IEEE Information Theory Workshop (ITW)*, pages 1–5. IEEE, 2015.

[127] Ilya Tolstikhin, Olivier Bousquet, Sylvain Gelly, and Bernhard Schoelkopf. Wasserstein auto-encoders. *arXiv preprint arXiv:1711.01558*, 2017.

[128] Michael Tschannen, Olivier Bachem, and Mario Lucic. Recent advances in autoencoder-based representation learning. *arXiv preprint arXiv:1812.05069*, 2018.

[129] Roman Vershynin. Introduction to the non-asymptotic analysis of random matrices. *arXiv preprint arXiv:1011.3027*, 2010.

[130] Roman Vershynin. *High-Dimensional Probability: An Introduction with Applications in Data Science*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 2018.

[131] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of machine learning research*, 11(Dec):3371–3408, 2010.

[132] Meng Wang, Pengzhi Gao, Scott G Ghiocel, Joe H Chow, Bruce Fardanesh, George Stefopoulos, and Michael P Razanousky. Identification of âĂIJunobservableâĂİ cyber data attacks on power grids. In *Smart Grid Communications (SmartGridComm), 2014 IEEE International Conference on*, pages 830–835. IEEE, 2014.

[133] Satosi Watanabe. Information theoretical analysis of multivariate correlation. *IBM Journal of research and development*, 4(1):66–82, 1960.

[134] W. Wen, C. Wu, Y. Wang, Y. Chen, and H. Li. Learning structured sparsity in deep neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2074–2082, 2016.

[135] John Wright, Allen Y Yang, Arvind Ganesh, Shankar S Sastry, and Yi Ma. Robust face recognition via sparse representation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(2):210–227, 2009.

[136] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017.

[137] Huan Xu, Constantine Caramanis, and Sujay Sanghavi. Robust pca via outlier pursuit. In *Advances in Neural Information Processing Systems*, pages 2496–2504, 2010.

[138] Bin Yang. Projection approximation subspace tracking. *IEEE Transactions on Signal processing*, 43(1):95–107, 1995.

[139] M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67, 2006.

[140] S. Zagoruyko. 92.45% on CIFAR-10 in torch. `http://torch.ch/blog/2015/07/30/cifar.html`, 2015.

[141] X. Zeng and M. Figueiredo. The ordered weighted ℓ_1 norm: Atomic formulation, projections, and algorithms. *arXiv:1409.4271*, 2015.

[142] Xiangrong Zeng and Mário AT Figueiredo. Decreasing weighted sorted l1 regularization. *IEEE Signal Processing Letters*, 21(10):1240–1244, 2014.

[143] Dejiao Zhang and Laura Balzano. Global convergence of a grassmannian gradient descent algorithm for subspace estimation. In *AISTATS*, pages 1460–1468, 2016.

[144] Shengjia Zhao, Jiaming Song, and Stefano Ermon. Infovae: Information maximizing variational autoencoders. *arXiv preprint arXiv:1706.02262*, 2017.

[145] Qinqing Zheng and John Lafferty. A convergent gradient descent algorithm for rank minimization and semidefinite programming from random linear measurements. In *Advances in Neural Information Processing Systems*, pages 109–117, 2015.

[146] H. Zhou, J. Alvarez, and F. Porikli. Less is more: Towards compact cnns. In *European Conference on Computer Vision*, pages 662–677. Springer, 2016.

[147] H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society (series B)*, 67(2):301–320, 2005.