Deep Generative Models for Single-Cell Perturbation Experiments

by

Hengshi Yu

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Biostatistics)
in The University of Michigan
2022

Doctoral Committee:

Assistant Professor Joshua D. Welch, Co-Chair
Associate Professor Xiang Zhou, Co-Chair
Professor Veerabhadran Baladandayuthapani
Assistant Professor Jie Liu

Hengshi Yu

hengshi@umich.edu

ORCID iD: 0000-0001-9850-9347

To my parents, Jie Yu and Mingxia Zhang,

who showed me unconditional love and support,

and taught me the values of hard work and integrity.

ACKNOWLEDGEMENTS

It was my greatest honor and pleasure to pursue my Ph.D. degree in biostatistics at the University of Michigan. My Ph.D. life was challenging and rewarding, and having it was one of the best decisions that I have ever made. Over the past five years of pursuing my doctoral degree, I was extremely fortunate to have many mentors and friends by my side supporting and encouraging me to become a qualified researcher.

I would like to express my sincere gratitude to my co-advisor Prof. Joshua Welch, for his continuous guidance, instruction and support. I am deeply indebted to Prof. Joshua Welch who led me into the interesting world of deep learning and his exceptional research group as the first group member. Josh taught me how to define research problems, how to do research, how to write academic papers and give talks. I have thoroughly enjoyed working with Josh, who helped and motivated me to bring innovative solutions to challenging topics. Thank you for always being supportive and giving me the freedom to design various solutions for many interesting research problems.

The completion of my dissertation would not have been possible without the support and encouragement of my co-advisor, Prof. Xiang Zhou, who first introduced me to computational biology research with Bayesian modeling. I am extremely grateful to Prof. Xiang Zhou for his kind help with the resources necessary to thrive during my Ph.D. program, as well as for his generous guidance on my research problems and invaluable feedback on my research work.

I would also like to show my appreciation for my dissertation committee members,

Prof. Veera Baladandayuthapani and Prof. Jie Liu, for their time and effort to serve on my dissertation committee. Thanks for providing me with insightful comments, feedback and encouragements on my research along the way.

I want to thank all our biostatistics faculty members, especially for Profs. Thomas Braun, Peter Xuekun Song, Brisa Sánchez, Michael Boehnke, Trivellore Raghunathan, Kevin Zhi He, Susan Murray, Veronica Berrocal and Lili Zhao. I sincerely thank Prof. Lu Wang, for her help while I was a junior student. Further thanks go to our biostatistics staff members, Nicole Fenech, Fatma-Zohra Nedjari, Kerry Sprague, Amanda Larson, Andrea Hill and Tara Smith, for their help and support. I would also like to express my deep appreciation to Kirsten Herold for her help on my scientific writing.

I want to thank Prof. Richard Gonzalez from the statistics department for giving me great advice on bringing innovative dissertation topics. I would also like to thank Prof. Fan Li from Yale University and Prof. Elizabeth Turner from Duke University who supervised my master's thesis research and encouraged me to pursue my Ph.D. study in biostatistics. I am grateful to Prof. John Preisser from the University of North Carolina at Chapel Hill, who is a role model as an enthusiastic researcher. I also want to thank Joseph Replogle from Jonathan Weissman's lab at MIT for providing the genome-scale Perturb-seq data in Chapter IV.

I would like to extend my sincere thanks to my colleagues in the Welch Lab for discussions and encouragement. I have always been inspired to discuss with my fabulous colleagues on many interesting bio-AI projects and I am honored to be the first Ph.D. graduated from our research group. I would like to express my thanks to Jane Wiesner for her valuable and insightful advice for my dissertation. I want to thank Profs. Jun Li, Jeffrey Regier and Ivo Dinov for helpful discussions in research. I want to thank Ken Weiss and Brock Palen for assisting our computing clusters.

I want to thank my fellow biostatistics students and friends during my study at

the University of Michigan. I am grateful to have interesting academic discussions and work on homework problems with my fabulous classmates and friends. Amongst many others, this includes Yilun Sun, Yaoyuan Vincent Tan, Boxian Wei, Kelly Speth, Summer Xia, Christopher Lee, Lili Wang, Ming Tang, Nina Zhou, Yingchao Zhong, Yan-Cheng Chao, Jiaqiang Zhu, Dylan Sun, Wenbo Wu, Jung Yeon Won, Ketian Yu, Emily Roberts, Jonathan Boss, Andrew Whiteman, Daiwei David Zhang, Guangyu Yang, Xubo Yue, Holly Hartman, Woosub Shin, Abhay Hukku, Tianwen Ma, Catherine Smith, Yuqi Zhai, Pedro Orozco del Pino and Mengbing Li, for their help in and outside research and coursework. I would also like to extend my gratitude to the excellent academic and research environment of Michigan Biostatistics.

Lastly, I want to thank my parents, Jie Yu and Mingxia Zhang. Although my parents are on the other side of Earth, they are always by my side, giving me the best assistance and cheers. Thanks for always believing in me, and giving me unconditional love, support and joy. To my parents, I dedicate this dissertation.

The research work in this dissertation was partially supported by the National Institutes of Health (NIH) grant R01-HG010883. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health.

TABLE OF CONTENTS

LIST OF FIGURES

Figure

LIST OF TABLES

Table

LIST OF ABBREVIATIONS

DNA Deoxyribonucleic Acid

RNA Ribonucleic Acid

A Adenine

G Guanine

C Cytosine

T Thymine

U Uracil

tRNA Transfer RNA

NGS Next-Generation Sequencing

RNA-seq RNA Sequencing

cDNA Complementary DNA

scRNA-seq Single-Cell RNA-seq

ISH In Situ Hybridization

FISH Fluorescence In Situ Hybridization

smFISH Single Molecule FISH

seqFISH Sequential FISH

MERFISH Multiplexed Error-Robust FISH

HTS High-Throughput Screen

ADME Absorption, Distribution, Metabolism and Excretion

CRISPR Clustered Regularly Interspaced Short Palindromic Repeats

PAM Protospacer Adjacent Motif

CRISPRa CRISPR-Mediated Activation

CRISPRi CRISPR Interference

dCas9 Deactivated Cas9

ITE Individual Treatment Effect

VAEs Variational Autoencoders

GANs Generative Adversarial Networks

ELBO Evidence Lower Bound

KL Kullback-Leibler

MI Mutual Information

TC Total Correlation

MIG Mutual Information Gap

MLP Multilayer Perceptron

FC Fully-Connected

ReLU Rectified Linear Unit

SMILES Simplified Molecular-Input Line-Entry System

KNN k-Nearest Neighbors

cINN Conditional Invertible Neural Network

FID Fréchet Inception Distance

GRNs Gene Regulatory Networks

GO Gene Ontology

LSTM Long Short-Term Memory

XAI Explainable Artificial Intelligence

W2 Wasserstein-2

ABSTRACT

Recent developments in deep learning have enabled generation of novel and realistic images or sentences from low-dimensional representations. In addition, a revolution in biotechnology has enabled high-throughput measurement of gene expression in thousands to millions of single cells. Several deep generative models have been developed to learn latent representations of cells and generate realistic high-dimensional single-cell data. However, these deep generative models primarily generate data similar to that seen during training and have limited ability to predict gene expression of unseen cell states. In consequence, it constrains the applicability of deep generative models for single-cell data, which usually have a relatively small set of observed conditions. Therefore, this dissertation aims to develop flexible and accurate deep generative models for single-cell data that learn representations characterizing how cells respond to various perturbation conditions and predict unobserved cell states.

In Chapter II, we study two main classes of deep generative models for single-cell RNA-seq data: variational autoencoders (VAEs) and generative adversarial networks (GANs). We systematically assess their disentanglement and generation performance and show that VAEs excel at learning cellular representations, while GANs excel at generating realistic single-cell gene expression data. We also develop MichiGAN, a novel neural network architecture that combines the strengths of VAEs and GANs to sample from disentangled representations without sacrificing data generation quality. We learn disentangled representations of three large single-cell RNA-seq datasets and use MichiGAN to sample from these representations to manipulate semantically distinct aspects of cellular identity and predict single-cell gene expression responses

to drug treatments.

In Chapter III, we develop PerturbNet, a novel deep generative model to generate single-cell data under unseen drug treatments. Existing approaches attempt to learn drug effects independently of cell state and cannot predict results for unseen drug treatments. To address these limitations, our PerturbNet framework learns mapping from a continuous representation of drug treatment to cellular states. PerturbNet can then generate single-cell data for both observed and unseen drug treatments. We show that PerturbNet accurately predicts single-cell RNA-seq data resulting from unseen drug treatments. We also fine-tune PerturbNet using cellular properties to improve the continuous representations of drug treatments.

In Chapter IV, we extend PerturbNet to learn single-cell responses to genetic perturbations, including pooled CRISPR genetic inactivations and genetic mutations. Existing approaches attempt to learn genetic perturbation effects independently of cell state and rely on one-hot encodings of genetic perturbations. Although this type of representation allows different combinations of observed target genes to be learned, it cannot generalize to unseen target genes. We develop a GenotypeVAE model and also employ a state-of-the-art protein sequence embedding model to encode genetic perturbations into continuous representations, allowing prediction for both unseen genes and unseen gene combinations.

In Chapter V, we extend PerturbNet to design optimal perturbations and attribute perturbation outcomes to specific perturbation features. We consider the translation of a group of cells to a target cell state, and propose two algorithms to design perturbations that achieve this desired target cell state. We show that the algorithms are effective at designing perturbations that achieve the cell state translation of interest. We also employ model interpretability methods to attribute the effects of chemical or genetic perturbations to specific atoms or gene functional annotations.

CHAPTER I

Introduction

1.1 Molecular Biology Overview

The cell is the fundamental unit of biological life. A biological organism consists of one or multiple cells, corresponding to unicellular and multicellular organisms. A multicellular organism, the main focus of this dissertation, has tens to millions of cells with the same machinery for their most basic functions (*Alberts et al.*, 2002). Additionally, an organism has extraordinary capacity of reproducing itself called heredity when a parent organism hands down detailed information to specify offspring characteristics. These properties make a multicellular organism distinguishable from others, giving rise to many species on Earth.

The information stored in cells and passed from a parent multicellular organism to its offspring is encoded in the form of double-stranded nucleotide molecules of DNA. The complete set of DNA sequences in an organism serves as its genome, and the genetic constitution defines its genotype. The organism's genotype guides its basic functions by decoding the information in the genome to use for cells. The genome's guidance to cells forms the central dogma of molecule biology. Basically, the genetic information from the DNA sequences of most multicellular organisms is first transcribed to RNA molecules, and then the RNA molecules translate the information to proteins, which are building blocks of cells. With these two processes of the central

dogma, the genetic information flows from genome to cellular functional products.

Although a multicellular organism can consist of a large number of cells with the same genotype, its cells exhibit varying functions across time and space. For example, a red blood cell in a human body delivers oxygen to body tissues through the circulatory system, while a nerve cell transmits information through synapses. The varying characteristics of cells in a multicellular organism are diverse cellular phenotypes expressed from the same genotype, enabling the complex functions of life.

A key goal of molecular biology is to understand how a genotype encodes myriad and diverse phenotypes of cells. Both DNA and RNA are linear polymers made of four types of nucleotide subunits linked together by phosphodiester bonds. The four nucleotide subunits of a double-stranded DNA sequence are bases adenine (A), guanine (G), cytosine (C) and thymine (T). In contrast, RNA is a single-stranded sequence and its nucleotides are ribonucleotides with three similar bases and the fourth base of uracil (U) rather than thymine (T) in DNA. The bases of DNA can be paired with those of RNA by hydrogen-bonding so that the DNA bases A, G, C and T have their complementary RNA bases U, C, G and A. In the transcription phase, a small portion of the DNA double helix is exposed through opening and unwinding, and one of its two strands functions as a template to grow an RNA chain through base-pairing. Then the RNA chain is released from the DNA template and the DNA helix re-forms. Next in the translation phase, an RNA molecule is attached with ribosomes and its nucleotide sequence is converted to an amino acid sequence to form proteins. The conversion follows rules known as the genetic code, where a group of three consecutive nucleotides in RNA (a codon) is paired with a specific kind of molecule called a transfer RNA (tRNA) or defines a signal to terminate the translation. Each tRNA is bound with a type of amino acid and has a region of three consecutive nucleotides, forming an anti-codon that can pair with a codon. Therefore,

the nucleotide sequence in an RNA molecule is translated to an amino acid sequence through codon pairing with tRNAs (*Alberts et al.*, 2002). The amino acid sequences fold into proteins that serve enzymatic or structural functions within the cell.

Thus, in the two phases of the central dogma, RNA serves as an intermediate element to convey the genetic information from DNA to proteins. The final protein profiles across cells correspond to their diverse phenotypes. Measuring protein levels of individual cells can directly dissect their heterogeneous phenotypic patterns, but contemporary technologies have not been able to provide single-cell protein profiles. As proteins are generated in translation of RNA molecules, their levels are correlated with RNA profiles. Therefore, understanding the gene expression profile of RNA molecules facilitates studying the diverse cellular phenotypes in a multicellular organism. The gene expression profile of a cell defines its cell state or identity, and can be used to infer its phenotypic functions and properties.

1.2 Single-Cell Transcriptome

The gene expression profile of a cell is typically a query with each entry representing the number of RNA molecule copies from a gene. The expression profile of a set of cells is represented as an expression matrix by stacking the gene expression profiles across the cells. The gene expression matrices are usually high-dimensional and with large sample sizes. For example, Tabula Muris is a compendium with gene expression data for around 10^5 cells and $20,000$ genes collected from 20 organs and tissues (*Consortium et al.*, 2018). The abundance of gene expression profiles gives rich information for learning heterogeneity and diversity of cellular functions and phenotypes. The transcriptome is the complete set of RNA molecules from a biological sample. This collection is often a large and and high-dimensional dataset. Sequencing technologies have largely advanced transcriptome collection by determining the precise nucleotide sequences of many RNA molecules. The first-generation

sequencing technology originated in the 1970's and was based on the automated Sanger method (*Metzker*, 2005; *Hutchison III*, 2007). Despite many improvements, the first-generation Sanger sequencing has limited power for sequencing large numbers of molecules. Recent advances created the next-generation sequencing (NGS) technologies that combine procedures of template preparation, sequencing and imaging, and genome alignment and assembly methods to produce enormous datasets at a much lower expense (*Metzker*, 2010). The NGS technologies sequence massive numbers of DNA strands in parallel (*Goodwin et al.*, 2016).

NGS technologies greatly accelerate profiling of transcriptomes, enabling RNA sequencing (RNA-seq) methods to profile and interpret the functional elements of genome (*Wang et al.*, 2009). In RNA-seq, RNA molecules are converted to a library of complementary DNA (cDNA) molecules with amplification, and are subsequently sequenced in a high-throughput manner (*Tang et al.*, 2010). Numerous high-throughput RNA-seq methods are available to quantify RNA molecules for tens of thousands of genes in biological samples, bringing many biological applications (*Islam et al.*, 2011). The earliest RNA-seq experiments sequenced RNA molecules in bulk, measuring the average expression profile of an entire group of cells.

However, such bulk profiling obscures the distinct properties of cells within a heterogeneous population (*Lee et al.*, 2019). The improvement of sequencing technology has made it possible to bring many RNA-seq methods to single-cell resolution (*Shapiro et al.*, 2013). Single-cell RNA-seq (scRNA-seq) technologies have revolutionized transcriptome analysis by providing a comprehensive catalogue of gene expression in mixed cell populations or complex tissues such as the brain.

Figure 1.1 from *Lee et al.* (2019) shows the general procedures of single-cell transcriptome technology. First, cells are dissociated from solid tissues using mechanical or enzymatic dissociation methods, and are isolated to individual cells by pipetting, laser-capture microdissection or microfluidic approaches. RNA molecules in single

cells are converted to cDNA and amplified to create a cDNA library. Finally, the cDNA library is sequenced via NGS platforms to obtain the single-cell transcriptome. Single-cell sequencing was named the method of the Year 2013 by Nature Methods (*Anonymous*, 2014). Recent novel scRNA-seq methods such as Drop-seq (*Macosko et al.*, 2015; *Rodriques et al.*, 2019) have largely enhanced sequencing efficiency and scalability. In addition, many large scRNA-seq datasets have been generated, such as the Tabula Muris compendium (*Consortium et al.*, 2018).

Modern developments in scRNA-seq technologies have also enabled comprehensive single-cell profiling methods for simultaneously measuring multiple molecular modalities. Researchers have combined scRNA-seq with other biomolecular assays such as protein profiling, chromatin accessibility, DNA methylation and spatial positions (*Li et al.*, 2019; *Swanson et al.*, 2021; *Burgess*, 2019). Single-cell multimodal omics was named the method of the Year 2019 by Nature Methods (*Anonymous*, 2020).

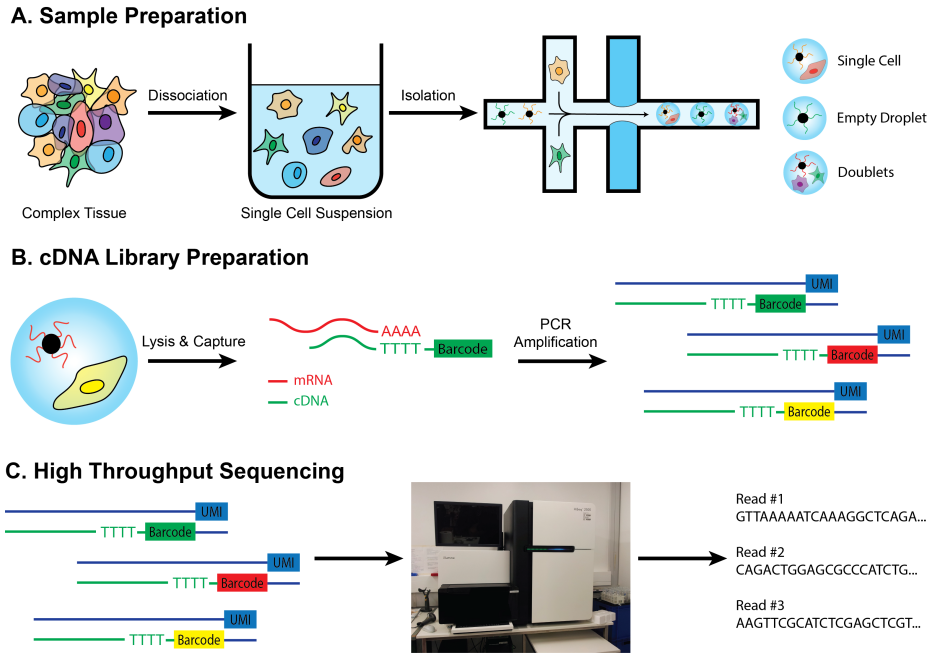

Figure 1.1: Overview of single-cell transcriptomic technology (*Lee et al.*, 2019).

1.3 Single-Cell Imaging

Single-cell imaging approaches represent a complementary approach to investigating cellular form and function. Fluorescent microscopy technologies employ fluorescence to study the physical appearance of cells (*Hamilton*, 2009). Fluorophore imaging of cells can be used to study molecular localization and subcellular structures. In situ hybridization (ISH) technology hybridizes a radioactive nucleotide sequence in solution and detects native nucleic acid sequences in cells by microautoradiography (*Pardue and Gall*, 1969; *John et al.*, 1969). Fluorescence in situ hybridization (FISH) further extends ISH to fluorescent labels rather than radioactive labels (*Pinkel et al.*, 1986, 1988). With fluorescent probes labeled with dyes, fluorescent microscopy technologies allow simultaneous visualization, identification, enumeration and localization of individual DNA or RNA molecules within single cells (*Pinkel et al.*, 1988; *Amann et al.*, 1990).

Many FISH extensions provide high-quality imaging modalities for cells. Single molecule FISH (smFISH) employs single molecules labeled with gene-specific probes to precisely quantify transcripts and locations (*Raj et al.*, 2008). The sequential FISH (seqFISH) protocol improved the accuracy, scalability and resolution of transcriptome imaging using multiple hybridization rounds (*Eng et al.*, 2019). The multiplexed error-robust FISH (MERFISH) approach uses an error-correcting code to measure subcellular localization of transcripts and spatial context of large numbers of cells (*Chen et al.*, 2015). Both seqFish and MERFISH give morphological profiles of single cells and can be applied to cultured cells or complex tissues (*Asp et al.*, 2020).

Hyperspectral imaging obtains a wide spectrum of light for each pixel in the image (*Grahn and Geladi*, 2007), and can provide diagnostic information about tissue composition, morphology, physiology, and diagnostic information (*Lu and Fei*, 2014). Immunofluorescence techniques utilize antibodies labeled with fluorescent dyes to visualize molecules under a light microscope (*Odell and Cook*, 2013). Immunofluo-

rescence can amplify chemical signals, detect and localize specific antigens in various tissue types (*Im et al.*, 2019).

Imaging-based screening can bring high-content cellular data and their image features bring multi-dimensional information (*Bickle*, 2010; *Singh et al.*, 2014). Cell painting is another type of cell imaging technology that quantifies a very large set of features in cell morphology (*Bray et al.*, 2016). It utilizes different dyes on a single microscopy-based assay to illuminate biological components or compartments in fluorescent channels. The single-cell resolution from cell painting experiments can provide multidimensional and rich profiles for detecting subtle phenotypes.

Single-cell imaging technologies generate abundant measurements of cellular structure, activities and subcellular localization. Therefore, the single-cell imaging data can be used to study responses to various conditions such as drug treatments or genetic perturbations (*Chandrasekaran et al.*, 2021).

1.4 Chemical Perturbations

Cells can be exposed to various conditions in high-throughput screen (HTS) experiments. A screen experiment usually involves multiple complex chemical or genetic perturbations that potentially change cells' phenotypic characteristics (*Devlin*, 1997). Chemical perturbations are widely used to impact cellular activities, and a chemical perturbation is usually delivered as a drug treatment on cells. The HTS technologies originated in natural product screening and were then utilized to identify molecules that modulated therapeutic targets (*Pereira and Williams*, 2007). HTS experienced steady improvements to incorporate absorption, distribution, metabolism and excretion (ADME) targets (*Banker et al.*, 2003). Advances of scalability and efficiency of HTS have enabled many new methods for pharmaceutical drug discovery (*Janzen*, 2001; *Minor*, 2006).

HTS technologies have been employed to identify cellular elements with prolifera-

tion properties (*Yu et al.*, 2016). Researchers have also implemented HTS experiments in imaging responses with high-content morphology responses (*Perlman et al.*, 2004; *Futamura et al.*, 2012). Additionally, HTS identifies optimal cell lines with highly specific readouts (*Kang et al.*, 2016).

1.5 Genetic Perturbations

Genetic perturbations are performed directly on genes to impact their functions. Clustered regularly interspaced short palindromic repeats (CRISPR) is a family of proteins that contribute to the immune system of prokaryotic organisms. CRISPR has been adapted to perform gene-editing with Cas9, a widely used protein in bacteria that can easily find and bind to most of the desired sequences with a piece of guide RNA (*Jinek et al.*, 2012). The CRISPR/Cas9 system has been shown to perform RNA-guided site-specific DNA cleavages (*Cong et al.*, 2013). As shown in Figure 1.2, the CRISPR/Cas9 method delivers Cas9, guide RNA, and an associated complex to cells. The specified guide RNA molecule can locate a particular segment of host DNA and bind with the protospacer adjacent motif (PAM) sequence to induce double-stranded breaks of a DNA sequence. The Cas9 enzyme then precisely knocks out target genomic loci. The cleaved DNA sequences are repaired to form new sequences. The CRISPR/Cas9 method also facilitates simultaneous cleavages of multiple target loci, improving the efficiency and flexibility of gene editing activity (*Bialk et al.*, 2015).

Figure 1.2 also shows two main derivatives of CRISPR/Cas9 including CRISPR-mediated activation (CRISPRa) and CRISPR interference (CRISPRi). The two methods use a deactivated version of Cas9 (dCas9) to bind target DNA but not cut the DNA (*Gilbert et al.*, 2014). CRISPRa directs dCas9 fused with transcriptional activators to promoter regions to upregulate expression (*Cheng et al.*, 2013; *Konermann et al.*, 2015). In contrast, CRISPRi uses dCas9 fused with a transcriptional repressor to interfere with the transcription process of target genes by generating

a DNA recognition complex (*Qi et al.*, 2013). The activation and repression effects from CRISPRa and CRISPRi are reversible and can also be applied to multiple target genes (*Horlbeck et al.*, 2016).

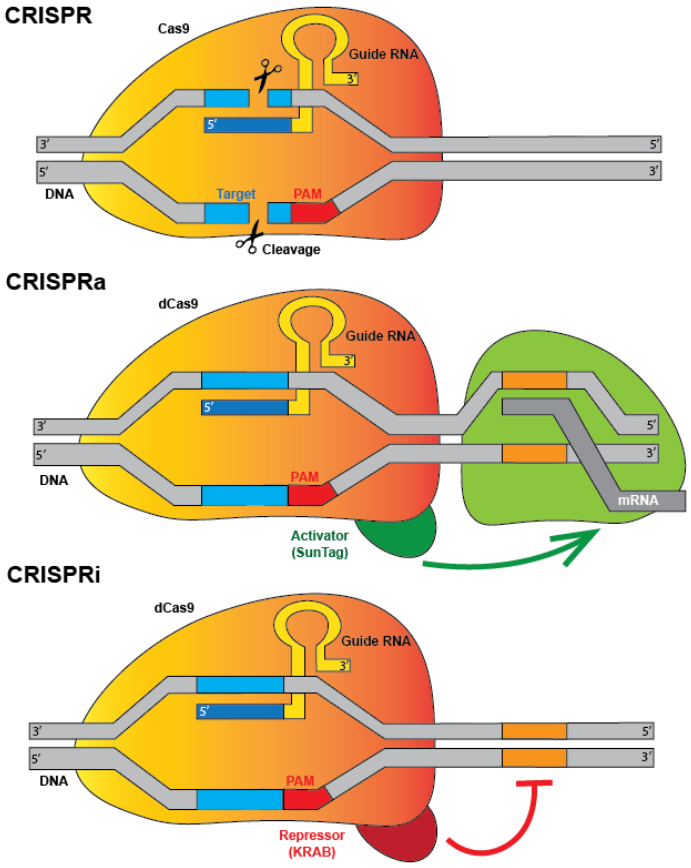

Figure 1.2: Overview of CRISPR/Cas, CRISPRa, CRISPRi technologies.

The genetic perturbations from CRISPR gene editing technologies usually involve complex experimental procedures, while the chemical perturbations from drug treatments have overall easier delivery procedures. CRISPR gene editing technologies have experienced rapid developments in human cells to precisely target genes and induce specific genomic effects (*Ran et al.*, 2013; *Shalem et al.*, 2014). Both the chemical and genetic perturbations can reveal fundamental biological properties of cells and modify cellular phenotypes. An optimal chemical or genetic perturbation can even cure disease by transforming cells to healthy cell states (*Meissner et al.*, 2022; *Jain et al.*,

2016). Publicly available chemical databases contain many kinds of drugs with different properties (*Gaulton et al.*, 2012), and detailed chemical information of millions of drugs (*Irwin and Shoichet*, 2005; *Kim et al.*, 2016). In addition, the set of possible genetic perturbations is huge. The human genome comprises approximately 3.2 billion nucleotides (*Brown*, 2018). Considering genetic perturbations with multiple target genes, the space of genetic perturbations is essentially infinite.

1.6 Measuring Single-Cell Responses to Perturbations

It has been challenging to bridge the gap between single-cell profiles and complex cellular perturbations, but recent pioneering experimental work has enabled high-throughput measurement of single-cell perturbation effects. On the one hand, large screen experiments designed with specific cellular perturbations can expose single cells to various biochemical conditions, where the cells' phenotypic changes are measured and examined. These perturbations provide single-cell measurements with various kinds of conditions, helping biomedical discovery and development. On the other hand, single-cell technologies bring rich high-throughput profiles for the cellular perturbations. The measured single-cell responses detail the high-dimensional perturbation effects. Measuring and learning single-cell responses to complex perturbations holds great promise for biological understanding and ultimately therapeutic intervention.

In a high-throughput chemical screen experiment, many cells receive drug treatments, and then their single-cell transcriptional responses are measured using single-cell technologies. A high-throughput chemical screen experiment such as sci-Plex can collect scRNA-seq responses to hundreds of drugs on multiple cell types (*Srivatsan et al.*, 2020). The single-cell responses to chemical perturbations are typically cell by gene matrices, with each entry representing the gene's transcription count in the cell. The identity and dose of the drug each cell receives are known.

Single-cell technologies also enable measuring the effects of using CRISPR gene editing methods to knock out, activate or knock down target genes. These CRISPR perturbations modify the activities of other genes through a complex network of genetic interactions in which genes regulate each other's expression levels (*Statello et al.*, 2021). As a result, a genetic perturbation exerts changes in the overall gene expression profile and molecular state of a cell. The Perturb-seq technologies combine scRNA-seq and CRISPR/Cas9 (*Dixit et al.*, 2016) or combine scRNA-seq and CRISPRi (*Adamson et al.*, 2016) to measure responses to genetic perturbations. As with sci-Plex, scRNA-seq responses to genetic perturbations are usually cell by gene matrices. The identity of the perturbation applied to each cell is known. Apart from Perturb-seq, other techniques such as CRISP-seq and CROP-seq also combine CRISPR perturbations with scRNA-seq, but have slightly different experimental and technical approaches (*Jaitin et al.*, 2016; *Datlinger et al.*, 2017).

The effects of CRISPR perturbations can also be measured with single-cell epigenomic profiling (*Rubin et al.*, 2019) and single-cell multimodal profiling (*Mimitou et al.*, 2019; *Frangieh et al.*, 2021; *Papalexi et al.*, 2021). CaRPool-seq combines CRISPR perturbations with multimodal single-cell profiling (*Wessels et al.*, 2022). In addition, single-cell imaging data with perturbations can reveal substantial information for mining and identifying cellular mechanisms (*Danuser*, 2011). CRISPR methods can also be paired with morphological profiling at scale in image-based screens (*Feldman et al.*, 2019). High-throughput cell painting experiments can also incorporate chemical or genetic perturbations to study cell responses and identify potential therapeutics (*Perlman et al.*, 2004).

1.7 Predicting Single-Cell Responses to Perturbations

The set of perturbations is usually limited in both chemical and genetic screen experiments. Although the delivery of a drug treatment is experimentally easier, a

multiplex chemical experiment usually involves several drug treatments and each drug treats a large number of cells with different experimental conditions (*Gehring et al.*, 2020). The CRIPSR genetic perturbations have much more byzantine procedures where the perturbations are usually delivered with deliberately planned biomedical processes. The single-cell CRISPR screen experiments generally have been limited to studying a few hundred genetic perturbations (*Replogle et al.*, 2022), while the genome contains tens of thousands genes, leaving behind massive combinations of genes. Therefore, a model that can predict single-cell responses to various unobserved perturbations would be tremendously valuable.

Predicting single-cell responses to perturbations, however, has some unique challenges. First, cells have heterogeneous responses to a perturbation *Altschuler and Wu* (2010); *Snijder and Pelkmans* (2011) and show varying individual perturbation responses. A more specific focus of perturbation responses might involve the individual treatment effect (ITE, *Shalit et al.*, 2017) in causal inference to predict single-cell perturbation responses of an individual cell or a cell subpopulation. Second, measuring single-cell responses usually destroys the cells, and each cell only provides a static snapshot of its profile after a specified perturbation. It is not immediately apparent to display the cell subpopulation within a perturbation. Third, the assignment of perturbations might be correlated with some cell state covariates characterizing the cell subpopulation under each perturbation, which further confounds attempts to model the perturbation responses. A model accurately predicting individual single-cell perturbation responses needs to debias the confounding effects from between-perturbation cellular characteristics.

1.8 Deep Generative Models

Many machine learning models have been developed for single-cell data. Two main branches of machine learning models are discriminative models and generative

models. A discriminative model, like a classification tree, predicts target variables from observable variables. On the other hand, a generative model such as a Gaussian mixture model (GMM) learns to generate the observed variables from latent variables. Generative models give a way of understanding the real-world data, and their latent variables serve as representations learned from the observable variables. However, most of the machine-learning-based generative models have limited performance to generate complex high-dimensional data such as those in single-cell transcriptome or imaging.

The deep learning era brought much more powerful generative models that are constructed with deep neural networks from latent variables to data samples, and are stochastically optimized. Some of the early deep generative models such as restricted Boltzmann machines (*Hinton*, 2002) and sigmoid belief networks (*Neal*, 1992) were very efficient in data modeling (*Welling et al.*, 2004) and their extensions have been applied to different types of data (*Hinton et al.*, 2006; *Hinton and Salakhutdinov*, 2009; *Salakhutdinov and Larochelle*, 2010; *Mnih and Gregor*, 2014). Three commonly-used classes of deep generative models are variational autoencoders (VAEs) (*Kingma and Welling*, 2013), generative adversarial networks (GANs) (*Goodfellow et al.*, 2014) and auto-regressive models such as PixelCNN and PixelRNN (*Oord et al.*, 2016a,b,c).

These state-of-the-art deep generative models have achieved tremendous success in the domains of images and natural language. Frameworks such as InfoGAN learn interpretable latent representations from images that manipulate semantically meaningful image generation (*Chen et al.*, 2016). Many derivatives of GANs such as WGAN (*Arjovsky et al.*, 2017), Progressive GAN (*Karras et al.*, 2017) and LOGAN (*Wu et al.*, 2019) improve the training stability and generation quality for high-resolution images. Conditional generation enables text-to-image synthesis (*Reed et al.*, 2016) or image-to-image translation (*Isola et al.*, 2017; *Zhu et al.*, 2017). VAEs coupled with normalizing flows improve variational inference for images (*Rezende and Mohamed*,

2015; *Kingma et al.*, 2016). Conditional VAEs can learn text attributes to perform attribute specific data generation or interpolation (*Yan et al.*, 2016). VAEs can also generate coherent novel sentences from interpretable latent representations (*Bowman et al.*, 2015). In addition, the normalizing flows of autoregressive models delineate the complex high-dimensional densities of images (*Dinh et al.*, 2014, 2016; *Grover et al.*, 2018). More recently, models such as transformers achieve excellent performance in both natural language processing and computer vision (*Vaswani et al.*, 2017; *Brown et al.*, 2020). Single-cell data have also utilized deep generative models to generate realistic samples and learn representations. The scVI framework (*Lopez et al.*, 2018) combines probabilistic modeling of scRNA-seq data and variational inference to learn low-dimensional latent representations and generate realistic single-cell responses. Other methods have also facilitated the prediction of single-cell responses to perturbations (*Lotfollahi et al.*, 2019). The current deep generative models for single-cell perturbation experiments, however, are usually unable to deal with the heterogeneity of perturbation responses, identification of pre-perturbation cell state distributions, or confounding bias in modeling perturbation responses. Additionally, most of existing methods only deal with limited numbers of perturbations, and can hardly extend to unseen perturbations. Therefore, new methods that effectively learn pre-perturbation cell state information, and accurately model single-cell perturbation responses for various perturbations, are needed.

1.9 Dissertation Overview

In this dissertation, we develop deep generative models for modeling single-cell perturbation data. We aim to accurately capture the cell state of individual cells from their single-cell perturbation responses. Then we precisely predict single-cell responses to various chemical or genetic perturbations.

In Chapter II, we study the two main branches of deep generative models–VAEs

and GANs–on single-cell data. We evaluate their disentanglement and generation performance on single-cell data. We propose an integrative framework with exceptional performance in both disentanglement and data generation. We then use our approach to learn semantically disentangled representations of cell states and predict unseen cell states.

In Chapter III, we propose a novel deep generative model to predict single-cell responses to unseen drug treatments. The deep generative model precisely captures drug information and cell state, as well as their relationships. We also propose a machine-learning method and predict scRNA-seq responses to drug treatments observed in the training data or from an unseen set using the proposed methods. We show that adjusting for cell state covariates can improve the prediction performance. We also develop a fine-tuning technique for the framework to improve its prediction performance.

In Chapter IV, we extend the proposed methods for drug treatments to predict single-cell responses to genetic perturbations. We consider two kinds of genetic perturbations: gene knockout/knockdown and sequence mutation. We predict single-cell perturbation responses for several large high-throughput screen datasets with genetic perturbations. We also fine-tune the framework and show improved performance of the methods.

In Chapter V, we develop methods to design optimal perturbations and interpret the trained model. We consider the scenarios of finding alternative perturbations to translate cells to approximate a target cell state. We extend our deep generative models for chemical and genetic perturbations to predict the counterfactual responses of cells. In addition, we also employ model interpretability methods to identify important atoms or gene ontology terms within chemical and genetic perturbations.

Finally, in Chapter VI, we summarize the previous chapters, discuss the implications of our work and propose potential future research.

CHAPTER II

Sampling from Disentangled Representations of Single-Cell Data using Generative Adversarial Networks

2.1 Introduction

Deep learning techniques have recently achieved remarkable successes, especially in vision and language applications (*LeCun et al.*, 2015; *Bengio et al.*, 2013). In particular, state-of-the-art deep generative models can generate realistic images or sentences from low-dimensional latent variables (*Theis et al.*, 2015). The generated images and text data are often nearly indistinguishable from real data, and data generating performance is rapidly improving (*Brock et al.*, 2018; *Wu et al.*, 2019). The two most widely used types of deep generative models are variational autoencoders (VAEs) and generative adversarial networks (GANs). VAEs use a Bayesian approach to estimate the posterior distribution of a probabilistic encoder network, based on a combination of reconstruction error and the prior probability of the encoded distribution (*Kingma and Welling*, 2013). In contrast, the GAN framework consists of a two-player game between a generator network and a discriminator network (*Goodfellow et al.*, 2014). GANs and VAEs possess complementary strengths and weaknesses: GANs generate much better samples than VAEs (*Goodfellow et al.*,

2016), but VAE training is much more stable and learns more useful "disentangled" latent representations (*Higgins et al.*, 2017). GANs outperform VAEs in generating sharp image samples (*Goodfellow et al.*, 2014), while VAEs tend to generate blurry images (*Larsen et al.*, 2016). GAN training is generally less stable than VAE training, but some recent derivations of GAN like Wasserstein GAN (*Arjovsky et al.*, 2017; *Gulrajani et al.*, 2017; *Heusel et al.*, 2017) significantly improve the stability of GAN training, which is particularly helpful for non-image data.

Achieving a property called "disentanglement", in which each dimension of the latent representation controls a semantically distinct factor of variation, is a key focus of recent research on deep generative models (*Desjardins et al.*, 2012; *Ridgeway*, 2016; *Denton et al.*, 2017; *Achille and Soatto*, 2018; *Eastwood and Williams*, 2018; *Locatello et al.*, 2019; *Higgins et al.*, 2018). Disentanglement is important for controlling data generation and generalizing to unseen latent variable combinations. For example, disentangled representations of image data allow prediction of intermediate images (*Berthelot et al.*, 2018) and mixing images' styles (*Karras et al.*, 2019). For reasons that are not fully understood, VAEs generally learn representations that are more disentangled than other approaches (*Hsu et al.*, 2017; *Dupont*, 2018; *Bai and Duan*, 2019; *Rolinek et al.*, 2019; *Esmaeili et al.*, 2019; *Khemakhem et al.*, 2020). The state-of-the-art methods for learning disentangled representations capitalize on this advantage by employing modified VAE architectures that further improve disentanglement, including β-VAE, FactorVAE, and β-TCVAE (*Higgins et al.*, 2017; *Kim and Mnih*, 2018; *Chen et al.*, 2018; *Gao et al.*, 2019). In contrast, the latent space of the traditional GAN is highly entangled. Some modified GAN architectures, such as InfoGAN (*Chen et al.*, 2016), encourage disentanglement using purely unsupervised techniques, but these approaches still do not match the disentanglement performance of VAEs (*Ramesh et al.*, 2018; *Kaneko et al.*, 2018; *Jeon et al.*, 2021; *Lin et al.*, 2019; *Kazemi et al.*, 2019; *Shen et al.*, 2020; *Liu et al.*, 2020; *Lee et al.*, 2020).

Disentanglement performance is usually quantitatively evaluated on standard image datasets with known ground-truth factors of variation (*Matthey et al.*, 2017; *Paysan et al.*, 2009; *Aubry et al.*, 2014; *Liu et al.*, 2015). In addition, disentangled representations can be qualitatively assessed by performing traversals or linear arithmetic in the latent space and visually inspecting the resulting images (*Burgess et al.*, 2018; *White*, 2016; *Laine*, 2018; *Dosovitskiy et al.*, 2015; *Sainburg et al.*, 2018).

Recently, molecular biology has seen the rapid growth of single-cell RNA-seq (scRNA-seq) technologies that can measure the expression levels of all genes across thousands to millions of cells (*Efremova and Teichmann*, 2020). Like image data, for which deep generative models have proven so successful, scRNA-seq datasets are large and high-dimensional. Thus, it seems likely that deep learning will be helpful for single-cell data. In particular, deep generative models hold great promise for distilling semantically distinct facets of cellular identity and predicting unseen cell states.

Several papers have already applied VAEs (*Lotfollahi et al.*, 2019; *Tan et al.*, 2014; *Gupta et al.*, 2015; *Way and Greene*, 2017; *Rampášek et al.*, 2019; *Deng et al.*, 2018; *Grønbech et al.*, 2018; *Wang and Gu*, 2018; *Ding et al.*, 2018; *Hu and Greene*, 2019; *Cui et al.*, 2020) and GANs (*Marouf et al.*, 2020) to single-cell data. A representative VAE method is scGen, which uses the same objective function as β-VAE (*Higgins et al.*, 2017). The learned latent values in scGen are utilized for out-of-sample predictions by latent space arithmetic. The cscGAN paper adapts the Wasserstein GAN approach for single-cell data and shows that it can generate realistic gene expression profiles, proposing to use it for data augmentation.

Assessing disentanglement performance of models on single-cell data is more challenging than image data, because humans cannot intuitively understand the data by looking at it as with images. Previous approaches such as scGen have implicitly used the properties of disentangled representations (*Lotfollahi et al.*, 2019), but disentanglement performance has not been rigorously assessed on single-cell data.

Here, we systematically assess the disentanglement and generation performance of deep generative models on scRNA-seq data. We show that the complementary strengths and weaknesses of VAEs and GANs apply to single-cell data in a similar way as image data. We develop MichiGAN, a neural network model that combines the strengths of VAEs and GANs to sample from disentangled representations without sacrificing data generation quality. We employ MichiGAN and other methods on simulated scRNA-seq data (*Zappia et al.*, 2017) and provide quantitative comparisons through several disentanglement metrics (*Chen et al.*, 2018). We also learn disentangled representations of three real scRNA-seq datasets (*Consortium et al.*, 2018; *Bastidas-Ponce et al.*, 2019; *Srivatsan et al.*, 2020) and show that the disentangled representations can control semantically distinct aspects of cellular identity and predict unseen combinations of cell states.

Our work builds upon that of *Lotfollahi et al.* (2019), who showed that a simple VAE (which they called scGen) can predict single-cell perturbation responses. They also showed several specific biological contexts in which this type of approach is useful. First, they predicted the cell-type-specific gene expression changes induced by treating immune cells with lipopolysaccharide. Second, they predicted the cell-type-specific changes that occur when intestinal epithelial cells are infected by *Salmonella* or *Heligmosomoides polygyrus*. Finally, they showed that scGen can use mouse data to predict perturbation responses in human cells or across other species. For such tasks, one can gain significant biological insights from the generated scRNA-seq profiles.

Our method, MichiGAN, can make the same kinds of predictions and yield the same kinds of biological insights as scGen, but we show that MichiGAN has significant benefits compared to scGen (including disentanglement and data generation performance). In addition, we show that MichiGAN can predict single-cell response to drug treatment, a biological application that was not demonstrated in the scGen paper.

2.2 Methods

2.2.1 Variational Autoencoders

VAEs have an encoder network with parameters $\boldsymbol{\phi}$, which maps the input data $\boldsymbol{X}$ to a latent space $\boldsymbol{Z}$, and a decoder network parameterized by $\boldsymbol{\theta}$, which reconstructs the high-dimensional data from the latent space.

Rather than learning a deterministic function for the encoder as in a conventional autoencoder, a VAE learns the mean and variance parameters of the posterior distribution $p_{\boldsymbol{\theta}}(\boldsymbol{X} \mid \boldsymbol{Z})$ over the latent variables and the data sample $\boldsymbol{X}$ are modeled to be drawn from $p_{\boldsymbol{\theta}}(\boldsymbol{X} \mid \boldsymbol{Z})$. However, even using a factorized Gaussian prior $p(\boldsymbol{Z}) = \mathcal{N}(\boldsymbol{Z} \mid \mathbf{0}, \boldsymbol{I})$, the posterior $p_{\boldsymbol{\theta}}(\boldsymbol{Z} \mid \boldsymbol{X})$ is intractable. Thus, VAEs perform parameter inference using variational Bayes, where the posterior distribution of latent $\boldsymbol{Z}$ given data $\boldsymbol{X}$ is approximated as $q_{\boldsymbol{\phi}}(\boldsymbol{Z} \mid \boldsymbol{X})$ (*Kingma and Welling*, 2013). Following a standard mean-field approximation, one can derive an evidence lower bound (ELBO). Given the data sampling distribution $q(\boldsymbol{X})$, the objective of VAE is to maximize the ELBO or minimize its opposite with respect to $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$:

$$L_{\text{VAE}}(\boldsymbol{\theta}, \boldsymbol{\phi}) = -\text{ELBO} = \mathbb{E}_{q(\boldsymbol{X})}\big[-E_{q_{\boldsymbol{\phi}}(\boldsymbol{Z}|\boldsymbol{X})}\big\{\log p_{\boldsymbol{\theta}}(\boldsymbol{X} \mid \boldsymbol{Z})\big\} + D_{\text{KL}}\big\{q_{\boldsymbol{\phi}}(\boldsymbol{Z} \mid \boldsymbol{X}) || p(\boldsymbol{Z})\big\}\big].$$

The ELBO has a nice interpretation: the first term is reconstruction error and the second term is the Kullback-Leibler (KL) divergence between the posterior and prior distributions of the latent variables $\boldsymbol{Z}$. If the prior distribution $p(\boldsymbol{Z})$ is factorized Gaussian or uniform distribution, the KL divergence encourages the latent factors to be statistically independent, which may contribute to the good disentanglement performance of VAEs. This effect can be further enhanced by introducing a weight β to place more emphasis on the KL divergence at the cost of reconstruction error, an approach called β-VAE (*Higgins et al.*, 2017).

2.2.2 β-TCVAE

The total correlation variational autoencoder (β-TCVAE) is a VAE extension that further promotes disentanglement. The KL divergence of VAE can be further decomposed into several parts:

$$\begin{aligned}\mathbb{E}_{q(\boldsymbol{X})}\big[D_{\mathrm{KL}}\big\{q_{\boldsymbol{\phi}}(\boldsymbol{Z} \mid \boldsymbol{X})||p(\boldsymbol{Z})\big\}\big] = &D_{\mathrm{KL}}\big\{q_{\boldsymbol{\phi}}(\boldsymbol{Z}, \boldsymbol{X})||q_{\boldsymbol{\phi}}(\boldsymbol{Z})q(\boldsymbol{X})\big\} \\ &+ D_{\mathrm{KL}}\big\{q_{\boldsymbol{\phi}}(\boldsymbol{Z})||\prod_j q_{\boldsymbol{\phi}}(Z_j)\big\} \\ &+ \sum_j D_{\mathrm{KL}}\big\{q_{\boldsymbol{\phi}}(Z_j)||p(Z_j)\big\}.\end{aligned}$$

The first part is referred to as the index-code mutual information (MI), the second part is the total correlation (TC) and the third part is the dimension-wise KL divergence (*Chen et al.*, 2018). The total correlation is the most important term for learning disentangled representations, while penalizing the two other parts does not directly improve the disentanglement performance, but increases the reconstruction error.

The β-TCVAE specifically penalizes the TC in the loss function:

$$L_{\beta\text{-TCVAE}}(\boldsymbol{\theta}, \boldsymbol{\phi}; \beta) = L_{\mathrm{VAE}}(\boldsymbol{\theta}, \boldsymbol{\phi}) + \beta D_{\mathrm{KL}}\big\{q_{\boldsymbol{\phi}}(\boldsymbol{Z})||\prod_j q_{\boldsymbol{\phi}}(Z_j)\big\},$$

where $\beta = 0$ gives the VAE loss function. There is no closed form for the total correlation of the latent representation, so β-TCVAE approximates it as follows:

$$\mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{Z})}\big\{\log q_{\boldsymbol{\phi}}(\boldsymbol{Z})\big\} \approx E_{q_{\boldsymbol{\phi}}(\boldsymbol{Z})}\big[\log \mathbb{E}_{q(\boldsymbol{X})}\{q_{\boldsymbol{\phi}}(\boldsymbol{Z} \mid \boldsymbol{X}) \mid \boldsymbol{Z}\}\big]$$

and

$$\mathbb{E}_{q_{\boldsymbol{\phi}}(Z_j)}\big\{\log q_{\boldsymbol{\phi}}(\boldsymbol{Z}_j)\big\} \approx \mathbb{E}_{q_{\boldsymbol{\phi}}(Z_j)}\big[\log \mathbb{E}_{q(\boldsymbol{X})}\big\{q_{\boldsymbol{\phi}}(Z_j \mid \boldsymbol{X}) \mid Z_j\big\}\big].$$

Estimating TC is difficult from a small minibatch, so we utilize the minibatch stratified sampling in *Chen et al.* (2018) to estimate $\mathbb{E}\{q_{\boldsymbol{\phi}}(\boldsymbol{Z} \mid \boldsymbol{X}) \mid \boldsymbol{Z}\}$ during training.

2.2.3 Generative Adversarial Networks

A generative adversarial network (GAN) consists of a generator network G and a discriminator network D. There are many types of GANs, but we specifically focus on Wasserstein GAN with gradient penalty (WGAN-GP, *Gulrajani et al.*, 2017), which significantly stabilizes GAN training. The discriminator loss function for WGAN-GP is

$$L_{\text{Discriminator}} = \mathbb{E}_{p(\boldsymbol{Z}),q(\boldsymbol{X})}\big[D(\boldsymbol{X}) - D\big\{G(\boldsymbol{Z})\big\} + \lambda\big\{||\nabla_{\widetilde{\boldsymbol{X}}} D(\widetilde{\boldsymbol{X}})||_2 - 1\big\}^2\big],$$

where $\nabla_{\widetilde{\boldsymbol{X}}} D(\widetilde{\boldsymbol{X}})$ is the gradient of the discriminator on input $\widetilde{\boldsymbol{X}}$ and $\widetilde{\boldsymbol{X}} = \epsilon \boldsymbol{X} + (1-\epsilon)G(\boldsymbol{Z})$ with ϵ sampled from a uniform distribution on $[0, 1]$. The generator loss function for WGAN-GP is

$$L_{\text{Generator}} = \mathbb{E}_{p(\boldsymbol{Z})}\big[D\{G(\boldsymbol{Z})\}\big].$$

Upon convergence, WGAN-GP gives the generated data distribution $\{G(\boldsymbol{Z}) : \boldsymbol{Z} \sim p(\boldsymbol{Z})\}$ that matches the real data distribution $q(\boldsymbol{X})$.

2.2.4 Conditional GAN and PCGAN

The conditional GAN extends GANs to respect the relationship between generated data and known labels (*Mirza and Osindero*, 2014). There are many different network architectures for conditional GAN (*Mirza and Osindero*, 2014; *Reed et al.*, 2016; *Odena et al.*, 2017), but found the conditional GAN with projection discriminator (PCGAN) (*Miyato and Koyama*, 2018) works best. A recent paper similarly found that PCGAN worked well for scRNA-seq data (*Marouf et al.*, 2020). The original PCGAN paper mentions that the projection discriminator works most effectively when the conditional distribution $p(\boldsymbol{C}|\boldsymbol{X})$ is unimodal. One theoretical reason why PCGAN may be well-suited for MichiGAN is that the posterior multivariate Gaussian distributions of latent variables from VAEs are, in fact, unimodal.

In implementing the PCGAN, we do not use the conditional batch normalization or spectral normalization mentioned in *Miyato and Koyama* (2018), but instead use standard batch normalization and Wasserstein GAN with gradient penalty. Thus we refer to this approach as PCWGAN-GP.

2.2.5 MichiGAN: Combining the Strengths of VAEs and GANs

As VAEs achieve better disentanglement performance, but GANs achieve better generation performance, we sought to develop an approach that combines the strengths of both techniques. Several previous approaches have combined variational and adversarial techniques (*Larsen et al.*, 2016; *Pu et al.*, 2017; *Mescheder et al.*, 2017). However, when we tested these approaches on single-cell data, we found that attempts to jointly perform variational and adversarial training compromised both training stability and generation performance. We also investigated the InfoGAN and semi-supervised InfoGAN with details in Supplementary Materials Section 2.5.3, but found that the disentanglement performance was still significantly worse than that of the VAE approaches as shown in Supplementary Figure 2.11.

We thus developed a different approach: We first train a VAE to learn a disentangled representation. Then we use the VAE encoder's latent representation $\boldsymbol{z}$ for each cell $\boldsymbol{x}$ as a given code and train a conditional GAN using the $(\boldsymbol{z}, \boldsymbol{x})$ pairs. After training, we can generate high-quality samples from the VAE's disentangled representation. Importantly, the training is no less stable than training VAE and GAN separately, and the GAN generation quality is not compromised by a regularization term encouraging disentanglement. In addition, any kind of representation–from non-linear methods like VAEs or linear methods like PCA–can be incorporated in our approach. Wanting to follow the convention that the names of many generative adversarial networks end with "GAN", but unable to devise a compelling acronym, we named our approach MichiGAN after our institution. Algorithm 1 summarizes

the MichiGAN framework:

Algorithm 1: MichiGAN

Input: scRNA-seq data $\boldsymbol{X}$.

1. Obtain disentangled representations $\boldsymbol{Z_X}$ from an approach such as PCA, VAE or β-TCVAE.
2. Utilize the representations $\boldsymbol{Z_X}$ as codes.
3. Train a conditional GAN (*Miyato and Koyama*, 2018) using the codes.

Result: a generator network that produces high-quality samples from a disentangled latent representation.

The MichiGAN architecture is also shown in Figure 2.1. We find that MichiGAN effectively achieves our goal of sampling from a disentangled representation without compromising generation quality (see results below). Although our approach is conceptually simple, there are several underlying reasons why it performs so well, and recognizing these led us to pursue this approach. First, training a conditional GAN maximizes mutual information between the condition variable and the generated data. This is a similar intuition as the InfoGAN but, unlike InfoGAN, MichiGAN does not need to learn its own codes, and thus the discriminator can focus exclusively on enforcing the relationship between code and data. A nearly optimal discriminator is crucial for maximizing this mutual information, but the Wasserstein loss also has this requirement, and we meet it by training the discriminator five times for every generator update. Second, the adversarial loss allows the GAN generator to capture complex, multi-modal distributional structure that cannot be modeled by the factorized Gaussian distribution of the VAE decoder. This is particularly helpful if multiple distinct types of cells map to a similar latent code, in which case the unimodal Gaussian distribution of the VAE decoder will generate the average of these cell types. In contrast, even though the GAN generates from the same latent representation as the VAE, the GAN can fit complex, multimodal distributions by minimizing the Wasserstein distance between generated and true data distributions. Addition-

ally, a data-dependent code (the posterior of the VAE encoder) allows the GAN to generate from a flexible latent space that reflects the data distribution, rather than an arbitrary distribution such as the commonly used standard normal. We believe this inflexibility contributes significantly to the relatively poor disentanglement performance of InfoGAN. For example, InfoGAN is highly sensitive to the number and distribution chosen for the latent codes; if classes are imbalanced in the real data but the prior has balanced classes, it cannot learn a categorical variable that reflects the true proportions.

Based on the results from our disentanglement comparison (see below), we chose to use the β-TCVAE to learn the latent representation for MichiGAN. We then use either the posterior means or the random samples from the posterior as the condition for the GANs; both choices have been utilized to evaluate disentanglement performance in previous studies (*Higgins et al.*, 2017; *Kim and Mnih*, 2018; *Chen et al.*, 2018).

The last step of MichiGAN involves training a conditional GAN. We found that a conditional Wasserstein GAN with projection discriminator (*Miyato and Koyama*, 2018) and gradient penalty (*Gulrajani et al.*, 2017) is most effective at enforcing the condition. We also assessed semi-supervised InfoGAN (*Spurr et al.*, 2017) and a conditional GAN based on simple concatenation, but found that these were less effective at enforcing the relationship between code and generated data (Supplementary Figure 2.14) and less stable during training.

2.2.6 Latent Space Vector Arithmetic

MichiGAN's ability to sample from a disentangled representation allows predicting unseen combinations of latent variables using latent space arithmetic. We perform latent space arithmetic as in *Lotfollahi et al.* (2019) to predict the single-cell gene expression of unseen cell states. Specifically, suppose we have m cell types $C_1, \ldots, C_m$ and n perturbation $D_1, \ldots, D_n$. Denote $\boldsymbol{Z}(C_i, D_j)$ as the latent value corresponding to the

expression data with combination (C_i, D_j) for $1 \leq i \leq m$ and $1 \leq j \leq n$. If we want to predict the unobserved expression profile for the combination $(C_{i'}, D_{j'})$, we can calculate the average latent difference between cell type $C_{i'}$ and another cell type C_k in the set of observable treatments $\boldsymbol{\Omega}$ that $\boldsymbol{\Delta}_{C_{i'},C_k} = \int_{\boldsymbol{\Omega}} \{Z(C_{i'}, D_s) - Z(C_k, D_s)\} dP(s)$ and then use the latent space $\boldsymbol{Z}(C_k, D_{j'})$ of observed combination $(C_k, D_{j'})$ to predict

$$\widehat{\boldsymbol{Z}}(C_{i'}, D_{j'}) = \boldsymbol{Z}(C_k, D_{j'}) + \boldsymbol{\Delta}_{C_{i'},C_k}.$$

The predicted $\widehat{\boldsymbol{Z}}(C_{i'}, D_{j'})$ is further used to generate predicted data of the unseen combination. The predicted latent space assumes the average latent difference across observed treatments is equal to the latent difference of the unobserved treatment, which may not hold if there is a strong cell type effect for the perturbation.

2.2.7 Latent Space Entropy

We developed a novel metric for assessing the accuracy of latent space arithmetic for a particular held-out cell type/perturbation combination. For a subset of the data $\boldsymbol{X} \sim g(\boldsymbol{X})$ and the latent space $\boldsymbol{Z} \sim \tau(\boldsymbol{Z})$, we define the latent space entropy as:

$$H\left\{\tau(\boldsymbol{Z}), g(\boldsymbol{X})\right\} = -\mathbb{E}_{\tau(\boldsymbol{Z})}\left[\log \mathbb{E}_{g(\boldsymbol{X})}\left\{q_{\boldsymbol{\phi}}(\boldsymbol{Z} \mid \boldsymbol{X}) \mid \boldsymbol{Z}\right\}\right].$$

Intuitively, H quantifies the concentration of $\boldsymbol{Z}$ with respect to $\boldsymbol{X}$. We can then compare the entropy of the latent embeddings for the held-out data and the latent values predicted by latent space arithmetic by calculating $\Delta H = H\{\tau_{Fake}(\boldsymbol{Z}), g(\boldsymbol{X})\} - H\{\tau_{Real}(\boldsymbol{Z}), g(\boldsymbol{X})\}$, where τ_{Fake} is calculated by latent space arithmetic and τ_{Real} is calculated using the encoder. The quantity ΔH then gives a measure of how accurately latent space arithmetic predicts the latent values for the held-out data. If ΔH is positive, then the latent space prediction is less concentrated (and thus more uncertain) than the encoding of the real data.

2.2.8 Related Work

To our knowledge, no approach like MichiGAN has been published. Several previous approaches have combined variational and adversarial techniques, including VAEGAN (*Larsen et al.*, 2016), adversarial symmetric variational autoencoder (*Pu et al.*, 2017) and adversarial variational Bayes (*Mescheder et al.*, 2017). InfoGAN and semi-supervised InfoGAN are also conceptually related to MichiGAN, but we found that none of these previous approaches produced good results on single-cell data. Concurrent to our work, another group released a preprint with an approach called ID-GAN, which also uses a pre-trained VAE to learn a disentangled representation (*Lee et al.*, 2020). However, they use the reverse KL divergence framework to enforce mutual information between the VAE representation and the generated data, which we previously tested and found does work as well as a conditional GAN with projection discriminator (*Miyato and Koyama*, 2018). Furthermore, ID-GAN uses a convolutional architecture and classic GAN loss for image data, whereas we use a multilayer perceptron architecture and Wasserstein loss for single-cell data.

2.3 Experiments

2.3.1 Variational Autoencoders Learn Disentangled Representations of Single-Cell Data

Real single-cell datasets usually have unknown, unbalanced, and complex ground-truth variables, and humans cannot readily distinguish single-cell expression profiles by eye, making it difficult to assess disentanglement performance by either qualitative or quantitative evaluations. Thus, we first performed simulation experiments to generate balanced single-cell data with several data generating variables using the Splatter R package (*Zappia et al.*, 2017). All the datasets were processed using the SCANPY software (*Wolf et al.*, 2018). Details of the simulation can be found in Sup-

plementary Materials Section 2.5.2. We measured the disentanglement performances of different methods on the simulated single-cell data using several disentanglement metrics and also provided qualitative evaluations on the learned representations using the real datasets with details in Supplementary Materials Section 2.5.1.

We first estimated simulation parameters to match the Tabula Muris dataset (*Consortium et al.*, 2018). Then, we set the differential expression probability, factor location, factor scale, and common biological coefficient of variation to be $(0.5, 0.01, 0.5, 0.1)$. We then used Splatter (*Zappia et al.*, 2017) to simulate gene expression data of $10,000$ cells with four underlying ground-truth variables: batch, path, step, and library size. Batch is a categorical variable that simulates linear differences among biological or technical replicates. Step represents the degree of progression through a simulated differentiation process, and path represents different branches of the differentiation process. We simulated two batches, two paths, and 20 steps. The batch and path variables have linear effects on the simulated expression data, while the step variable can be related either linearly or non-linearly to the simulated gene expression values. We tested the effects of this variable by separately simulating a purely linear and a non-linear differentiation process. We also included library size, the total number of expressed mRNAs per cell, as a ground-truth variable. A UMAP plot of the simulated data shows that the four ground-truth variables each have complementary and distinct effects on the resulting gene expression state (Figure 2.2a and Supplementary Figure 2.8a).

We compared the disentanglement performance of three methods: probabilistic principal component analysis (PCA) (*Tipping and Bishop*, 1999), β-VAE and β-TCVAE. The probabilistic PCA method assumes a linear relationship between data and representations, while VAE and β-TCVAE can learn non-linear representations. Note that we use probabilistic PCA to allow calculation of mutual information (see below). The β-TCVAE approach penalizes the total correlation of the latent represen-

tation, directly minimizing the mutual information between latent dimensions, which has been shown to significantly improve disentanglement performance on image data. The details of choosing β values and implementation are provided in Supplementary Materials Sections 2.5.6 and 2.5.7.

We used the three methods to learn a 10-dimensional latent representation of the simulated data (Figure 2.2b and Supplementary Figure 2.8b). Some latent variables learned by each method showed clear relationships with the ground-truth variables. For example, the first latent variable Z1 from PCA seemed related to library size, and Z3, Z4 and Z5 were related to batch, path and step, respectively. The VAE representations similarly showed some relationships with the ground-truth variables. Based on the UMAP plots, the latent variables from β-TCVAE appeared to show the strongest and most clear relationships with the ground-truth variables.

To quantify the disentanglement performance of the three methods, we calculated the Spearman correlation and normalized mutual information between each representation and a ground-truth variable (Figure 2.2c-d). Spearman correlation measures the strength of monotonic relatedness between two random variables. The normalized mutual information, on the other hand, is a more general and robust metric of statistical dependence. A disentangled representation should have a bar plot with only four distinct bars in this case, indicating that each ground-truth variable was captured by exactly one latent variable. PCA showed the best performance as measured by Spearman correlation (Figure 2.2c), likely because the metric does not fully characterize the complex statistical dependency between true and inferred latent variables for the VAE methods, which learn more complex non-linear relationships. Based on the normalized mutual information metric, both the PCA and VAE representations achieved some degree of disentanglement, but neither approach fully disentangled all ground-truth variables. Multiple PCA representations had measurable mutual information with step and library size quartile, while multiple VAE representations

identified batch and path and none of the VAE representations identified step. In contrast, exactly one β-TCVAE representation had significant mutual information for each ground-truth variable. Also, β-TCVAE was the only method with a unique representation for the non-linear step variable.

We also computed the Spearman correlation and normalized mutual information for the simulated data with linear step (Supplementary Figure 2.8c-d). The results for the simulated data with linear step were similar and β-TCVAE did the best at identifying only one representation for each ground-truth variable.

We further calculated the mutual information gap (MIG) metric used in *Chen et al.* (2018) and FactorVAE disentanglement metric (*Kim and Mnih*, 2018) to measure disentanglement. The MIG metric is defined as the average gap between the mutual information of the two latent variables that are most related to each ground-truth variable. If there is a single latent variable that has high mutual information with each ground-truth variable, the MIG will be high. The FactorVAE metric is based on the error rate of a linear classifier that identifies which ground-truth variable differs based on data points using latent dimensions. In addition, we calculated a Spearman correlation gap similar to MIG. We provide details in Supplementary Materials Section 2.5.4. Table 2.1 summarizes the correlation gap, FactorVAE metric and MIG of the three models over five runs for the two simulated datasets. As expected from the bar charts, the PCA representations have the largest Spearman correlation gap and β-TCVAE has the largest MIG, showing the best disentanglement performance for both simulated datasets. The FactorVAE metric also shows that β-TCVAE has the best disentanglement performance. We also evaluated InfoWGAN-GP on the simulated data in Supplementary Figure 2.11 and found that the representations are entangled with the ground-truth variables for simulated datasets with linear and non-linear step.

We also evaluated the disentanglement performance of the three methods with

four latent dimensions (the same as the number of ground-truth variables), for the simulated datasets in Supplementary Figures 2.15 and 2.16. The β-TCVAE representations still most effectively disentangle the ground-truth variables. Table 2.2 summarizes the disentanglement metrics of the three methods with four latent dimensions. Although FactorVAE metric shows similar values for the three methods, β-TCVAE consistently has much higher MIG than PCA and VAE.

In addition, we utilized the PROSSTT package (*Papadopoulos et al.*, 2019) to simulate three single-cell datasets. PROSSTT simulates cells undergoing a continuous process such as differentiation. As shown in Supplementary Figures 2.17a, 2.18a and 2.19a, the three PROSSTT-simulated datasets have 3, 4 or 5-way branching trajectories, respectively. The three PROSSTT-simulated datasets also have a continuous time variable. We use three ground-truth variables (branch, time and library size) to calculate mutual information with the learned latent variables (Supplementary Figures 2.17b, 2.18b and 2.19b). PCA and VAE have multiple latent dimensions with moderate mutual information with branch and time quartile, while β-TCVAE captures each of these quantities mostly in a single variable. We also summarized the disentanglement metrics of the three methods on the PROSSTT-simulated datasets in Table 2.3. β-TCVAE has the highest FactorVAE metric and MIG for each of the three datasets.

In summary, our assessment indicates that β-TCVAE most accurately disentangles the latent variables underlying single-cell data, consistent with its previously reported superior disentanglement performance on image data (*Chen et al.*, 2018).

2.3.2 GANs Generate More Realistic Single-Cell Expression Profiles than VAEs

We next evaluated the data generating performance of several deep generative models including VAE, β-TCVAE and Wasserstein GAN with gradient penalty (WGAN-

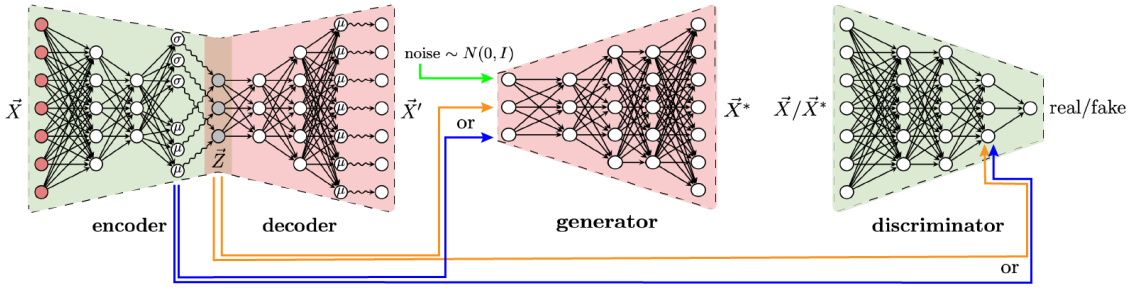

Figure 2.1: Overview of the MichiGAN architecture. We first train a model, such as β-TCVAE, to learn a disentangled representation of the real data. We then use the resulting latent codes to train a conditional GAN with projection discriminator, so that the GAN generator becomes a more accurate decoder. Because the VAE and GAN are trained separately, training is just as stable as training each one individually, but the combined approach inherits the strengths of each individual technique. After training, we can generate high-quality samples from the disentangled representation using the GAN generator.

Table 2.1: Disentanglement metrics for two splatter-simulated scRNA-seq datasets with four ground-truth variables. The mean and standard deviation over five runs are presented for each method. The dimensionality of the latent space was 10 for all three approaches.

		Spearman correlation gap ↑	FactorVAE metric ↑	MIG ↑
Linear step	PCA	**0.68** ± 0.00	0.35 ± 0.01	0.54 ± 0.00
	VAE	0.3 ± 0.04	0.4 ± 0.02	0.48 ± 0.13
	β-TCVAE	0.18 ± 0.05	**0.48** ± 0.03	**0.72** ± 0.02
Non-linear step	PCA	**0.72** ± 0.00	0.35 ± 0.01	0.55 ± 0.00
	VAE	0.27 ± 0.07	0.41 ± 0.02	0.43 ± 0.08
	β-TCVAE	0.16 ± 0.06	**0.51** ± 0.04	**0.66** ± 0.16

Table 2.2: Disentanglement metrics for two splatter-simulated scRNA-seq datasets with four ground-truth variables. The dimensionality of the latent space was four for all three approaches.

		Spearman correlation gap ↑	FactorVAE metric ↑	MIG ↑
Linear step	PCA	0.57	0.36	0.56
	VAE	0.37	**0.44**	0.39
	β-TCVAE	**0.65**	0.33	**0.72**
Non-linear step	PCA	**0.60**	0.36	0.58
	VAE	0.4	**0.38**	0.38
	β-TCVAE	0.55	0.34	**0.73**

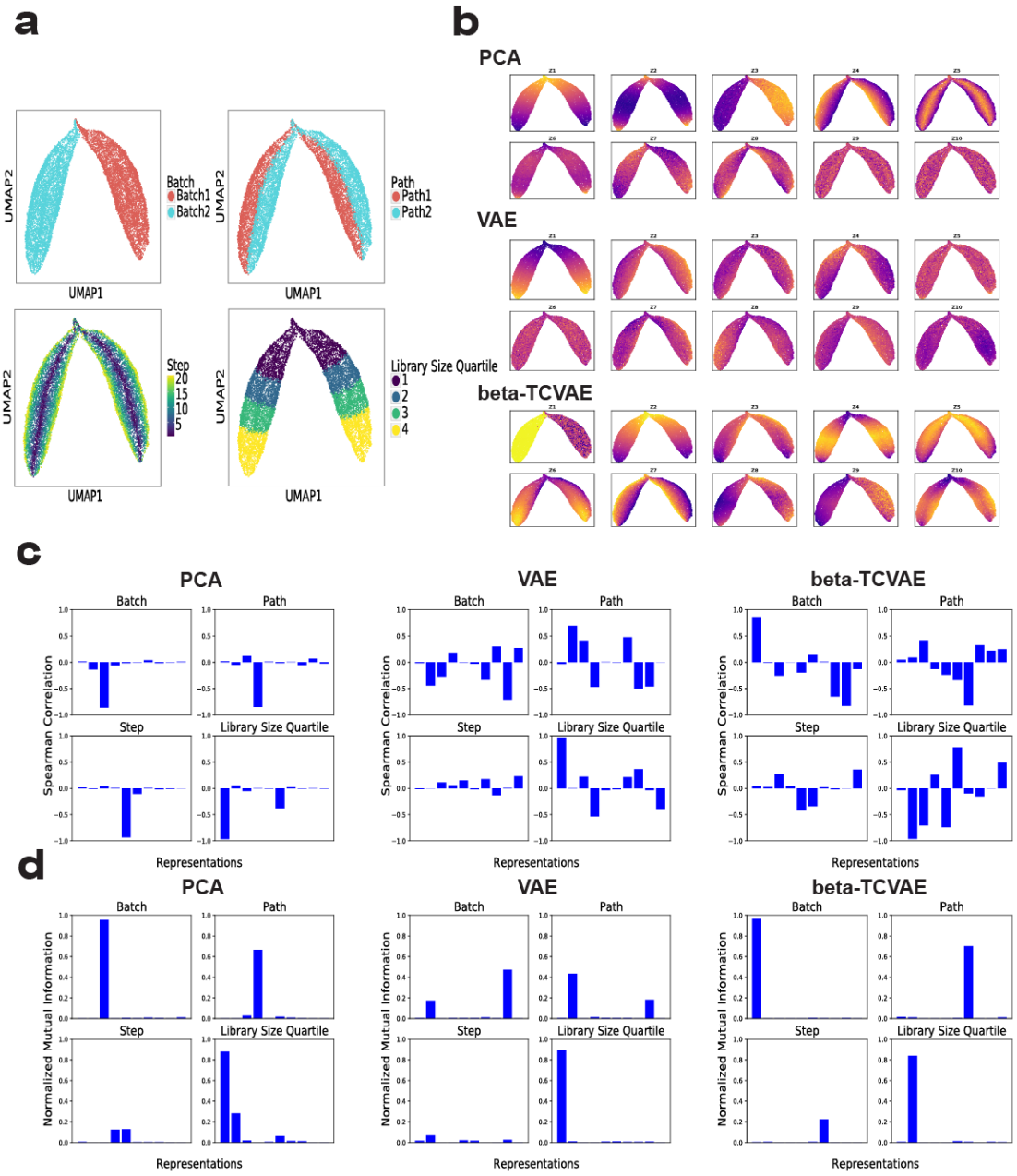

Figure 2.2: Evaluating disentanglement performance on simulated data. **a** UMAP plots of simulated data colored by batch, path, step and library size quartile. **b** UMAP plots of data colored by the 10 latent variables learned by PCA, VAE and β-TCVAE. **c** Bar plots of Spearman correlations between 10 latent variables and each of the four ground-truth variables for PCA, VAE and β-TCVAE. **d** Bar plots of normalized mutual information between 10 representations and each of the four ground-truth variables for PCA, VAE and β-TCVAE.

GP), as well as traditional methods of PCA and Gaussian mixture models (GMM) on the Tabula Muris dataset (*Consortium et al.*, 2018). This dataset contains a comprehensive collection of single-cell gene expression profiles from nearly all mouse tissues, and thus represents an appropriate dataset for evaluating data generation, analogous to the ImageNet dataset in computer vision. We also measured data generation performance on a subset of the Tabula Muris containing only cells from the mouse heart. We used two metrics to assess data generation performance: random forest error and inception score. Random forest error was introduced in the cscGAN paper (*Marouf et al.*, 2020), and quantifies how difficult it is for a random forest classifier to distinguish generated cells from real cells. A higher random forest error indicates that the generated samples are more realistic. We also computed inception score (*Barratt and Sharma*, 2018), a metric commonly used for quantifying generation performance on image data. Intuitively, to achieve a high inception score, a generative model must generate every class in the training dataset (analogous to recall) and every generated example must be recognizable as belonging to a particular class (analogous to precision). Details of these generation metrics can be found in Supplementary Materials Section 2.5.5.

We show the random forest errors over five runs of VAE, β-TCVAE and WGAN-GP during training for the Tabula Muris heart subset and the whole Tabula Muris in Figure 2.3a-b. We also evaluate simpler generative models, including PCA and GMM. WGAN-GP achieves the best generation performance, as measured by both metrics, on both the subset and full dataset. The deep generative models significantly outperform PCA and GMM. VAE achieves second-best generating performance and, as expected with an endeavour to pursue more disentangled representation, the quality of β-TCVAE generation is the worst of the three approaches. Figure 2.3c-d show the inception scores over five runs for the two datasets; this metric reveals the same trend as with random forest errors, indicating that WGAN-GP has the best genera-

tion performance and β-TCVAE generates the least realistic data. Additionally, the generation performance of the GAN is still significantly higher than that of the VAE even for the smaller Tabula Muris heart dataset. These results match well with previous results from the image literature, indicating that GANs generate better samples than VAEs, and VAE modifications to encourage disentanglement come at the cost of sample quality.

2.3.3 MichiGAN Samples from Disentangled Representations without Sacrificing Generation Performance

We evaluated the MichiGAN algorithm on the simulated single-cell data with the trained β-TCVAE models. Figure 2.4a shows the UMAP plots of real data colored by β-TCVAE latent representations and generated color-coded data using WGAN-GP and MichiGAN on the simulated data with non-linear step. The WGAN-GP representations are very entangled and none of the representations shows an identifiable coloring pattern. In contrast, the UMAP plots have consistent coloring patterns between the β-TCVAE and MichiGAN representations. Thus, the generator of MichiGAN preserves the relationship between latent code and generated data, effectively sampling from the disentangled representation learned by the β-TCVAE. Because there is no inference network for the generated data of either WGAN-GP or MichiGAN, we are unable to measure the mutual information for the generators. Therefore, we used Spearman correlation as an indicator of whether MichiGAN retains the relationship between disentangled latent representation and generated data. Figure 2.4b also shows the bar plots of Spearman correlations between representations and variables for the three methods. We used the correlations between each representation and ground-truth variables for β-TCVAE, WGAN-GP and MichiGAN. For GAN models, we trained a k-nearest neighbor regressor ($k = 3$) for each variable based on the real data and predicted the variables for the generated data. The WGAN-GP rep-

Table 2.3: Disentanglement metrics for three PROSSTT-simulated scRNA-seq datasets with three ground-truth variables

		FactorVAE metric ↑	MIG ↑
3 trajectories	PCA	0.54	0.10
	VAE	0.58	0.08
	β-TCVAE	**0.64**	**0.27**
4 trajectories	PCA	0.59	0.12
	VAE	0.61	0.12
	β-TCVAE	**0.72**	**0.15**
5 trajectories	PCA	0.59	0.06
	VAE	0.53	0.06
	β-TCVAE	**0.62**	**0.26**

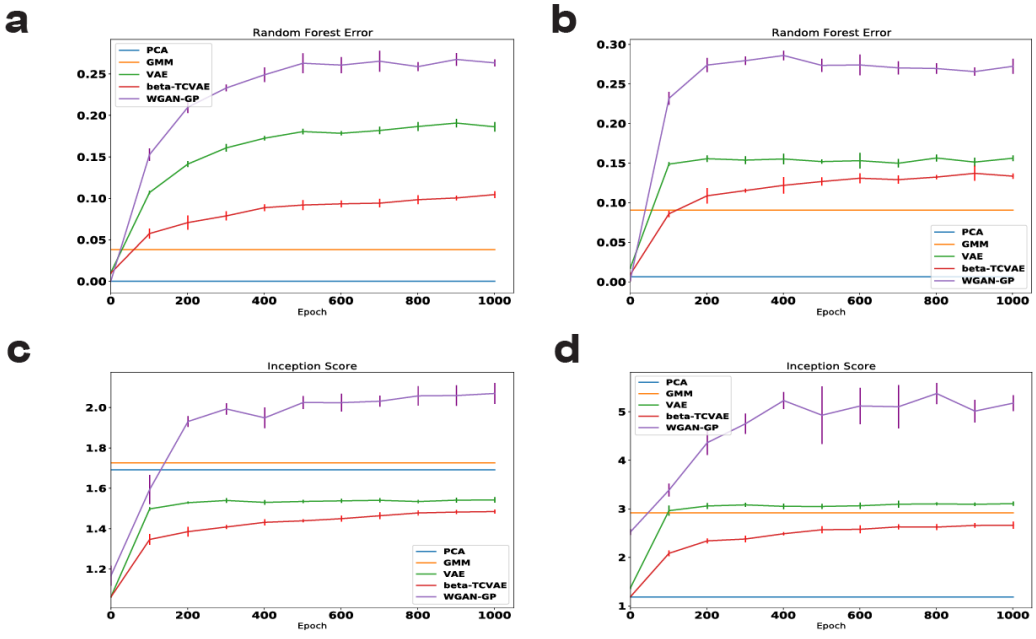

Figure 2.3: Generation performance of VAE, β-TCVAE, WGAN-GP, PCA and GMM on the Tabula Muris heart data and the whole Tabula Muris data. **a** Random forest error for the five methods on the Tabula Muris heart data during training. **b** Random forest error for the five methods on the whole Tabula Muris data during training. **c** Inception score for the five methods on the Tabula Muris heart data during training. **d** Inception score for the five methods on the whole Tabula Muris data during training. Error bars indicate standard deviation across five runs. For clarity, the error bars for PCA and GMM are omitted because of their small and large variability.

resentations do not show large correlation with any inferred ground-truth variable. In contrast, the representations for β-TCVAE and MichiGAN show nearly identical correlations to the true variables in the real data and predicted variables in the generated data, respectively.

We also trained MichiGAN using PCA to obtain the latent code, instead of β-TCVAE. Supplementary Figure 2.10a-b show the UMAP plots of real data colored by the PCA representations and generated data colored by the MichiGAN-PCA representations on the two simulated datasets. In addition, Supplementary Figure 2.10c-d show nearly identical Spearman correlation bar plots between PCA and MichiGAN. MichiGAN trained with principal components preserves the relationship between the latent representations and real data, underscoring the generalizability of our approach.

We present the UMAP plots colored by the representations as well as bar plots of correlations for the simulated data with linear step in Supplementary Figure 2.9a-b. The results for the simulated data with linear step also indicate that MichiGAN restores the disentanglement performance of β-TCVAE, while the WGAN-GP representations are entangled. We further summarize the correlation gaps for the three methods on two simulated datasets in Table 2.4. For each simulated dataset, the MichiGAN and β-TCVAE have very similar correlation gaps and WGAN-GP has a very small correlation gap, as expected.

We evaluated MichiGAN on the whole Tabula Muris dataset (Figure 2.4c). MichiGAN greatly improved the data generation performance based using the disentangled representations of β-TCVAE. The random forest error of MichiGAN was larger than VAE and nearly as good as the WGAN-GP, while still generating samples from a disentangled latent space.

Additionally, we applied PCA, GMM, VAE, β-TCVAE, WGAN-GP and MichiGAN on the pancreas endocrinogenesis dataset (*Bastidas-Ponce et al.*, 2019). We obtained the cells' latent time and cell cycle scores for G2M and S phases from *Bergen*

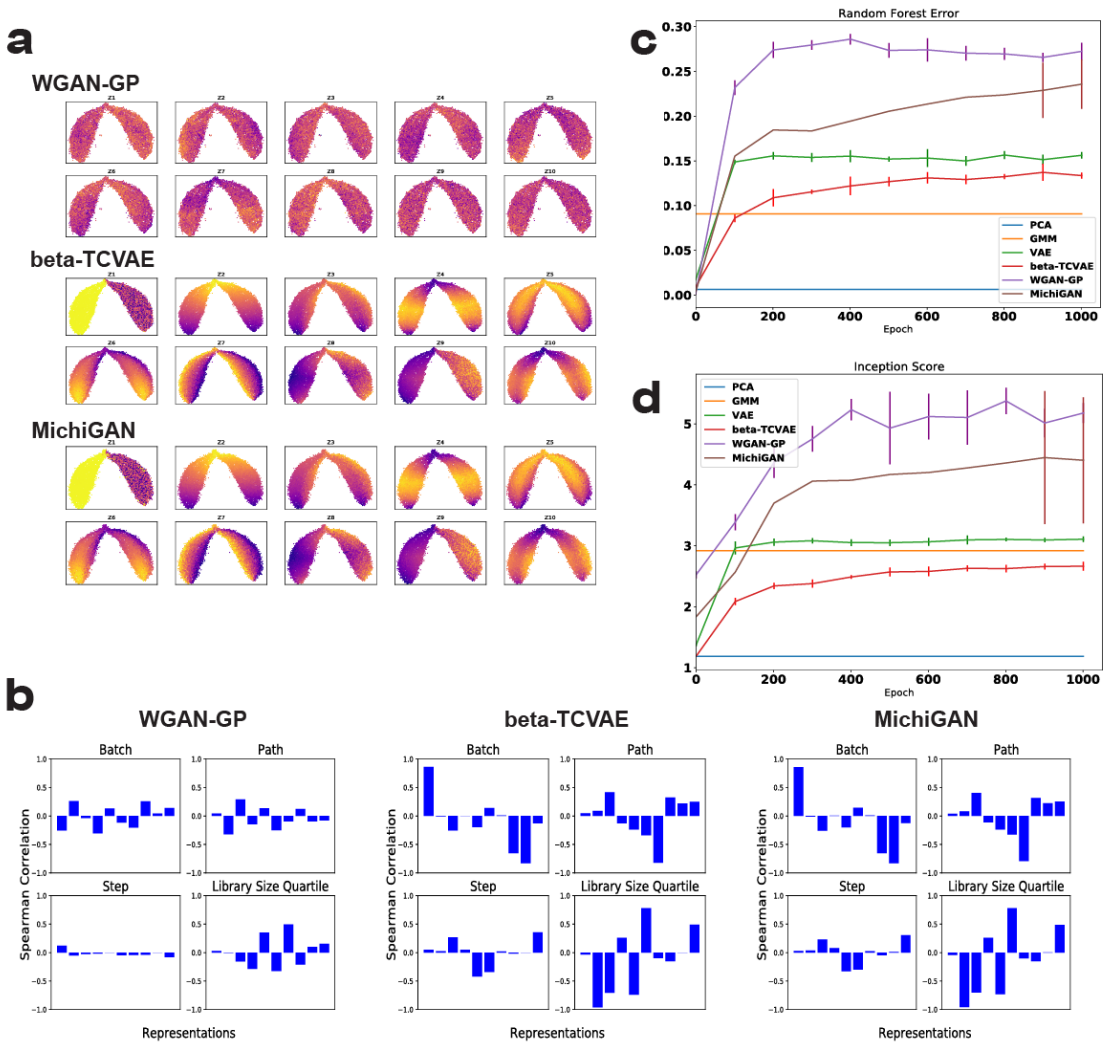

Figure 2.4: Disentanglement and generation performance of WGAN-GP, β-TCVAE and MichiGAN. **a** UMAP plots of real data colored by the 10 representations of β-TCVAE and generated data colored by the 10 representations of WGAN-GP and MichiGAN on the simulated data with non-linear step. The β-TCVAE panel is reproduced from Figure 2.2b for clarity. **b** Bar plots of Spearman correlations between 10 representations and each of the four ground-truth or inferred variables for WGAN-GP, β-TCVAE and MichiGAN on the simulated data with non-linear step. The β-TCVAE panel is reproduced from Figure 2.2c for clarity. **c** Random forest error of PCA, GMM, VAE, β-TCVAE, WGAN-GP and MichiGAN on the whole Tabula Muris data during training. **d** Inception score of PCA, GMM, VAE, β-TCVAE, WGAN-GP and MichiGAN on the whole Tabula Muris data during training. Error bars indicate standard deviation across five runs. For clarity, the error bars for MichiGAN are shown only for the last 100 epochs because the convergence speed in earlier epochs is variable, and the error bars for PCA and GMM are omitted because of their small and large variability.

Table 2.4: Spearman correlation gap for the methods of WGAN-GP, InfoWGAN-GP, PCA, MichiGAN-PCA, VAE, β-TCVAE and MichiGAN on the two splatter-simulated scRNA-seq datasets. The mean and standard deviation are presented for each method over five runs.

Model	Linear step	Non-linear step
WGAN-GP	0.07 ± 0.02	0.10 ± 0.06
InfoWGAN-GP	0.05 ± 0.05	0.04 ± 0.02
PCA	0.68 ± 0.00	0.72 ± 0.00
MichiGAN-PCA	0.65 ± 0.01	0.68 ± 0.00
VAE	0.3 ± 0.04	0.27 ± 0.07
β-TCVAE	0.18 ± 0.05	0.16 ± 0.06
MichiGAN	0.18 ± 0.04	0.15 ± 0.05

et al. (2020). Supplementary Figure 2.20a shows the UMAP plots of data colored by latent time and the difference between G2M and S scores. The β-TCVAE method gives qualitatively more disentangled representations (Supplementary Figure 2.20b), and gives much better disentanglement metrics (Supplementary Figure 2.20c). In addition, Supplementary Figure 2.20c also shows that MichiGAN significantly improves the data generation performance of β-TCVAE.

2.3.4 MichiGAN Enables Semantically Meaningful Latent Traversals

Disentangled representations of images are often evaluated qualitatively by performing latent traversals, in which a single latent variable is changed by holding the others fixed. Looking at the resulting changes in the generated images to see whether only a single semantic attribute changes provides a way of visually judging the quality of disentanglement. We wanted to perform a similar assessment of MichiGAN, but single-cell gene expression values are not individually and visually interpretable in the same way that images are. Thus, we devised a way of using UMAP plots to visualize latent traversals on single-cell data.

We performed latent traversals using both the Tabula Muris dataset and data from the recently published sci-Plex protocol (*Srivatsan et al.*, 2020). After training

on the Tabula Muris dataset (Supplementary Figure 2.12a), we chose a starting cell type, cardiac fibroblasts (Supplementary Figure 2.12b). We then varied the value of each latent variable from low to high, keeping the values of the other variables fixed to the latent embedding of a particular cell. For the sci-Plex dataset, which contains scRNA-seq data from cells of three types (A549, K562, MCF7; Supplementary Figure 2.12c) treated with one of 188 drugs, we subsampled the data to include one drug treatment from each of 18 pathways by selecting the drug with the largest number of cells (Supplementary Figure 2.12d). This gives one treatment for each pathway; the numbers of cells for each combination are shown in Supplementary Table 2.5. We then performed latent traversals on cells with cell type MCF7 and treatment S7259 (Supplementary Figure 2.12e).

To visualize the traversals, we plotted each of the generated cells on a UMAP plot containing all of the real cells and colored each generated cell by the value of the latent variable used to generate it. Figure 2.5a-b show how traversing the latent variables concentrates the generated values on each part of the UMAP plots for Tabula Muris data using the first 10 dimensions of 128-dimensional WGAN-GP and MichiGAN, respectively. Figure 2.5c-d are the latent-traversal plots for the sci-Plex data using WGAN-GP and MichiGAN. As shown in Figure 2.5b, all but three of the latent variables learned by the β-TCVAE behave like noise when we traverse them starting from the fibroblast cells, a property previously noted in assessments of disentangled latent variables learned by VAEs (*Kim and Mnih*, 2018). The remaining dimensions, Z3, Z6 and Z10, show semantically meaningful latent traversals. Latent variable Z3 shows high values for mesenchymal stem cells and fibroblasts, with a gradual transition to differentiated epithelial cell types from bladder, intestine and pancreas at lower values of Z3. This is intriguing, because the mesenchymal-epithelial transition is a key biological process in normal development, wound healing and cell reprogramming (*Pei et al.*, 2019). Latent variable Z6 generates mesenchymal and

endothelial cells at low values, and mammary epithelial and cardiac muscle cells at high values. Latent variable Z10 is clearly related to immune function, generating immune cells at low and medium values and traversing from hematopoietic stem and progenitor cells to monocytes, T cells and B cells. In contrast, latent traversals in the latent space of 128-dimensional WGAN-GP (Figure 2.5a) do not show semantically meaningful changes along each dimension.

Figure 2.5d also shows that MichiGAN's latent traversals gives meaningful changes on the sci-Plex data. Latent variable Z8 has lower values on MCF7 cells and gradually transitions to higher values on K562 cells. In addition, latent variable Z9 also shows an A549-MCF7 transition with lower values on the A549 cells. The latent traversals of the 128-dimensional WGAN-GP, however, do not provide interpretable changes across the UMAP plot along each dimension. We also provide the latent traversals using 10-dimensional WGAN-GP for the two datasets in Supplementary Figure 2.13a-b and find that the latent traversals are still not semantically meaningful.

2.3.5 MichiGAN Predicts Single-Cell Gene Expression Data under Unseen Drug Treatments

One of the most exciting applications of disentangled representations is predicting high-dimensional data from unseen combinations of latent variables. We next investigated whether MichiGAN can predict single-cell gene expression response to drug treatment for unseen combinations of cell type and drug.

We trained MichiGAN on data from the recently published sci-Plex protocol. The dataset contains scRNA-seq data from cells of three types (A549, K562, MCF7), each treated with one of 188 drugs. The drug is known for each scRNA-seq profile. We subsampled the data to include one drug treatment from each of 18 pathways by selecting the drug with the largest number of cells (Figure 2.6a). We then have one treatment for each pathway; the numbers of cells for each combination are shown in

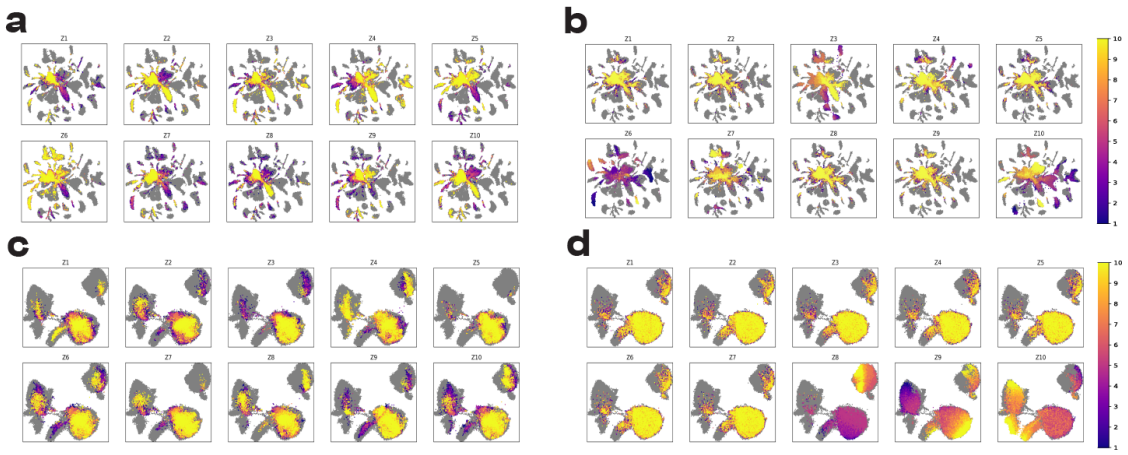

Figure 2.5: Latent traversals of WGAN-GP and MichiGAN on Tabula Muris and sci-Plex datasets. **a** UMAP plot of latent traversals of the 10 representations of latent values that generate data closest to fibroblast cells in the heart within the Tabula Muris data using WGAN-GP with 128 dimensions. **b** UMAP plot of latent traversals of the 10 representations of latent values of fibroblast cells in the heart within the Tabula Muris data using MichiGAN. **c** UMAP plot of latent traversals of the 10 representations of latent values that generate data closest to MCF7-S7259 cells within the sci-Plex data using WGAN-GP with 128 dimensions. **d** UMAP plot of latent traversals of the 10 representations of latent values of MCF7-S7259 cells within the sci-Plex data using MichiGAN.

Supplementary Table 2.5. We also held out three drug/cell type combinations (A549-S1628, K562-S1096 and MCF7-S7259) to test MichiGAN's out-of-sample prediction ability.

We predict single-cell gene expression for each drug/cell type combination in a two-step process. First, we estimate the mean latent difference between the target cell type and another control cell type for other treatments using either posterior means or posterior samples from the β-TCVAE encoder. We then add the average latent difference to the latent values with the same treatment and the control cell type. This latent space vector arithmetic assumes the mean cell type latent differences are homogeneous across different treatments. Note that this assumption may not hold if there is a strong interaction effect between cell type and drug treatment.

Because there are a total of three cell types, we have a total of six predictions for the three held-out drug/cell type combinations. Figure 2.6b shows UMAP plots for these six predictions. For all six predictions, the predicted values are closer to the true drug treated cells on the UMAP plot than the control cells used to calculate the latent vector. However, the predicted cells do not overlap with the treated cells for the combinations A549-S1628 and K562-S1096, while the two predictions for MCF7-S7259 appear to be more accurate. For both β-TCVAE and MichiGAN, we measure their random forest errors between the real and predicted cells for each combination. The random forest scatter plots for sampled representations are shown in Figure 2.6c. MichiGAN, with sampled representations, has significantly better random forest error than β-TCVAE ($p < 10^{-4}$, one-sided Wilcoxon test) and most of the points are above the diagonal line. We also show the random forest scatter plots for mean representations in Figure 2.6c, which does not show significantly larger random forest errors compared to β-TCVAE ($p > 0.05$, one-sided Wilcoxon test) and might be due to the remaining correlations among mean representations of β-TCVAE (*Locatello et al.*, 2019). Thus, MichiGAN, with sampled representations, is able to more accurately

make predictions from latent space arithmetic than β-TCVAE. However, some of the six predictions for the missing combinations show low random forest errors from both methods, and some of the predictions from MichiGAN are only marginally better than those of β-TCVAE.

2.3.6 Accuracy of Latent Space Arithmetic Influences MichiGAN Prediction Accuracy

We next examined factors influencing the accuracy of MichiGAN predictions from latent space arithmetic. We suspected that the prediction accuracy might depend on the accuracy of the latent coordinates calculated by latent space arithmetic, which could vary depending, for example, on whether the drug exerts a consistent effect across cell types.

The quantity ΔH measures how accurately latent space arithmetic predicts the latent values for the held-out data. Thus, we expect that MichiGAN should be able to more accurately predict drug/cell type combinations with a small ΔH.

As Figure 2.7a shows, ΔH is significantly correlated with the difference in random forest error between MichiGAN and β-TCVAE, when sampling from either the posterior distribution of the latent representations or the posterior means. This supports our hypothesis that accuracy of the latent space arithmetic influences MichiGAN performance. To further test this, we selected the three drug/cell type combinations with the lowest overall ΔH values, and re-trained the network using all combinations except these three. Figure 2.7b shows the predicted, real and control cells for the six predictions of the three new missing combinations based on MichiGAN using sampled representations. The predicted cells (green) overlap most parts of the real cells (blue) for all six predictions. As expected, MichiGAN predicted each of these low ΔH held-out combinations significantly more accurately than β-TCVAE (Figure 2.7c).

We also compared the performance of VAE and MichiGAN trained with VAE

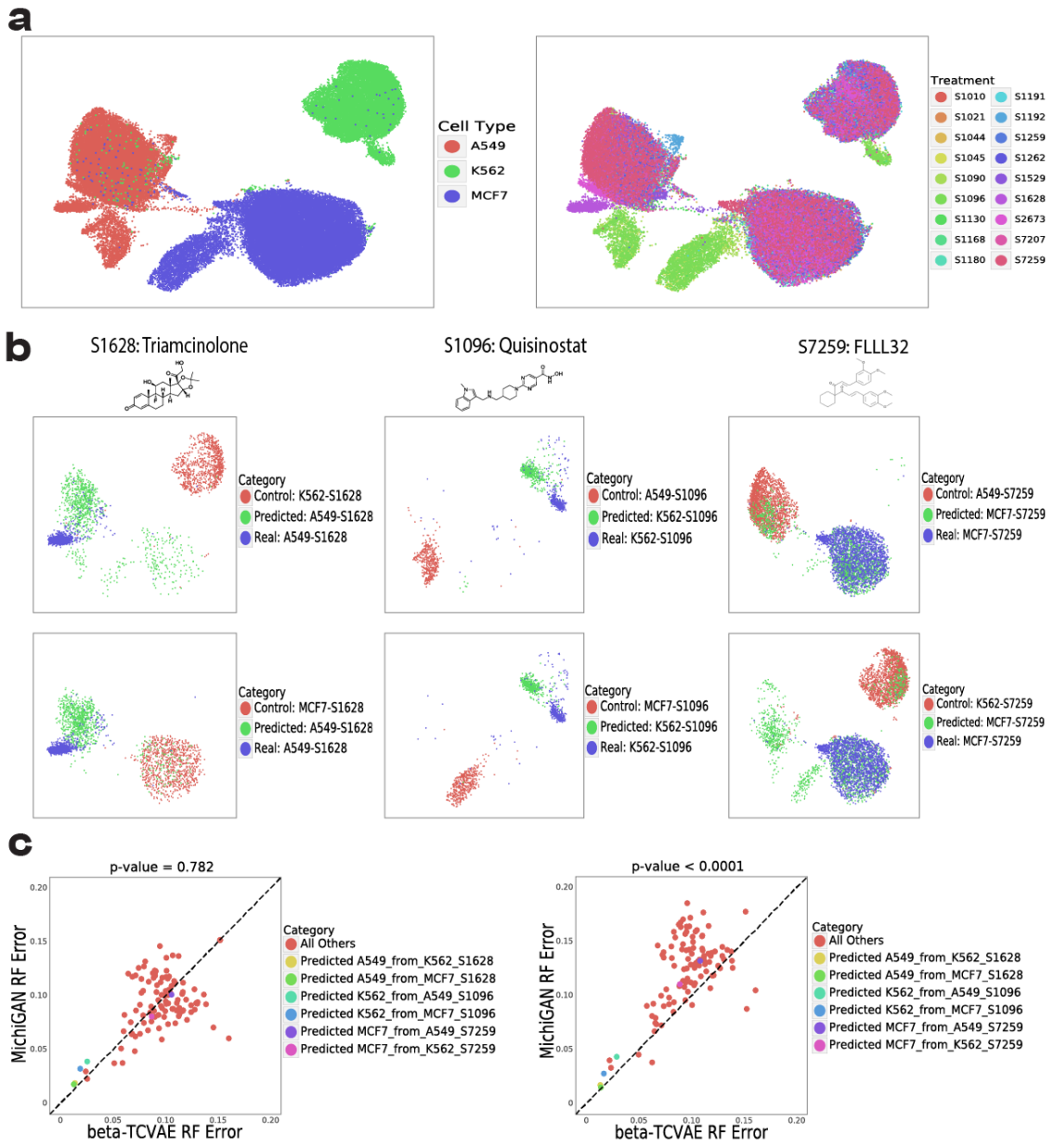

Figure 2.6: Predicting single-cell gene expression effects of unseen drugs using MichiGAN. **a** UMAP plots of sci-Plex dataset colored by cell type (left) and treatment (right). **b** UMAP plots of the predicted (green), real (blue) and control (red) cells for six predictions of three missing cell type/drug combinations (A549-S1628, K562-S1096 and MCF7-S7259). **c** Random forest errors between MichiGAN and β-TCVAE for all combinations. MichiGAN was trained using mean representations (left) or representations sampled from the posterior distribution (right).

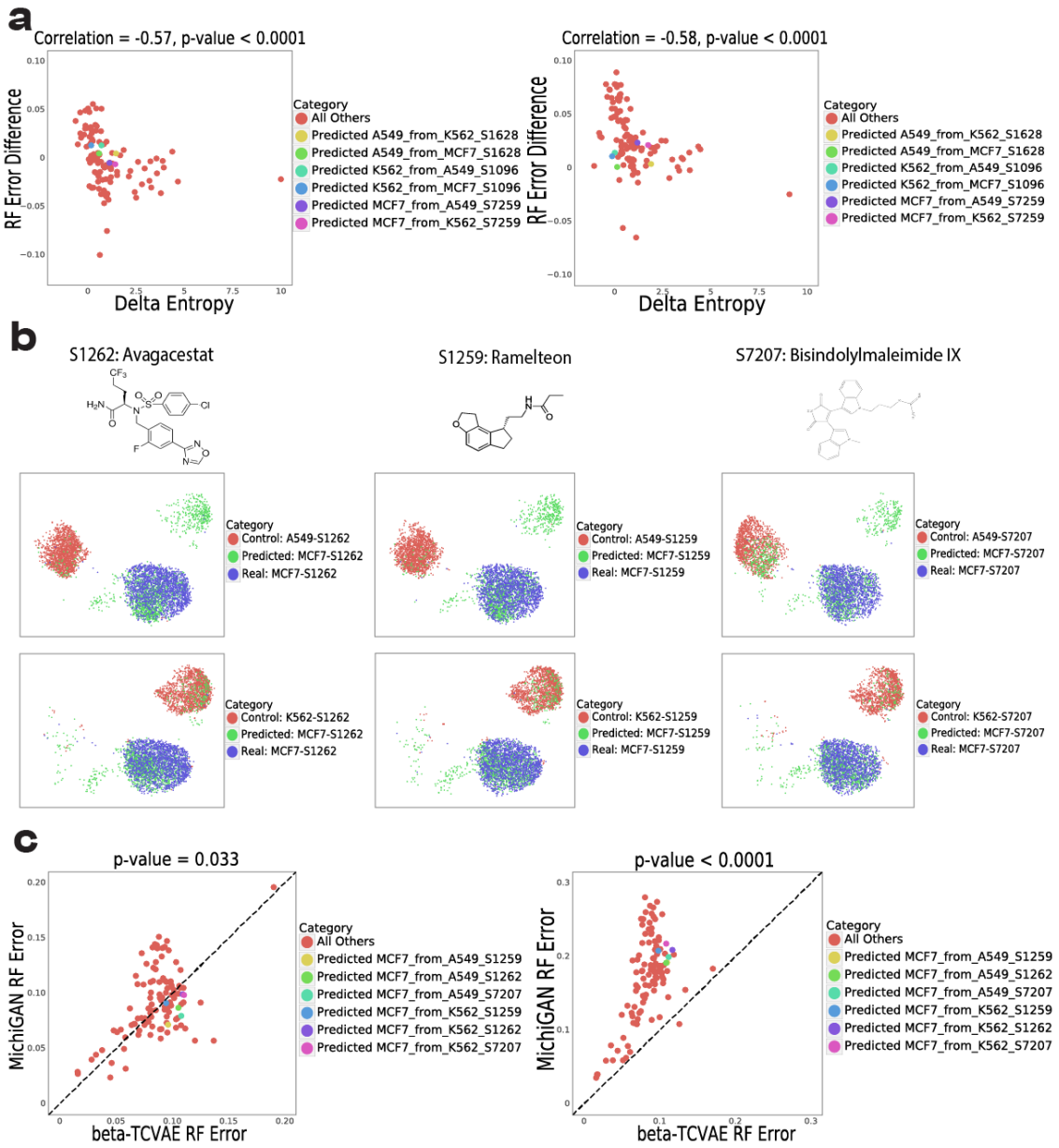

Figure 2.7: MichiGAN predicts unseen or observed combinations in the large screen sci-Plex data. **a** Scatter plots of random forest errors' difference between MichiGAN and β-TCVAE versus delta entropy for MichiGAN with mean representations (left) and sampled representations (right) on the large screen sci-Plex data without three combinations of A549-S1628, K562-S1096 and MCF7-S7259. **b** UMAP plots of the predicted (green), real (blue) and control (red) cells for six predictions of the three missing combinations of MCF7-S1262, MCF7-S1259 and MCF7-S7207. **c** Random forest errors between MichiGAN and β-TCVAE for MichiGAN with mean representations (left) and sampled representations (right) after selecting held-out combinations with low ΔH.

on the sci-Plex data after holding out the selected drug/cell type combinations with lowest overall ΔH values in Supplementary Figure 2.22. MichiGAN trained with VAE gives accurate prediction of the unseen combinations (Supplementary Figure 2.22a), and also has significantly higher random forest error than that of VAE to predict different drug/cell type combinations using the latent space vector arithmetic algorithm (Supplementary Figure 2.22b).

2.4 Discussion

Our work provides fundamental evaluations of disentanglement performances of deep generative models on scRNA-seq data. We show that combining GANs and VAEs can provide strong performance in terms of both data generation and disentanglement. MichiGAN provides an alternative to the current disentanglement learning literature, which focuses on learning disentangled representations through improved VAE-based or GAN-based methods, but rarely by combining them. Additionally, as the state of the art in disentangled representation advances, we can immediately incorporate new approaches in the MichiGAN framework, because the training of representation and GAN are completely separate.

We envision several exciting future directions. First, it would be interesting to investigate the representations learned by β-VAE or β-TCVAE across a range of biological contexts. Second, incorporating additional state-of-the-art GAN training techniques may further improve data generation quality. Additionally, there are many other biological settings in which predicting unseen combinations of latent variables may be helpful, such as cross-species analysis or disease state prediction.

2.5 Supplementary Materials

2.5.1 Real scRNA-seq Datasets

The Tabula Muris dataset is a compendium of single-cell transcriptomic data from the model organism Mus musculus (*Consortium et al.*, 2018). We processed the Tabula Muris data using SCANPY (*Wolf et al.*, 2018) and the dataset contains $41,965$ cells and 4062 genes from 64 cell types. The sci-Plex dataset has three cell types treated with 188 molecules targeting 22 pathways (*Srivatsan et al.*, 2020). We selected the 18 common pathways among the three cell types and chose the drug treatment from each pathway with the largest number of cells. We also use SCANPY to process the data and then have $64,050$ cells and 4295 genes. The pancreatic endocrinogenesis contains 3696 cells and $27,998$ genes (*Bastidas-Ponce et al.*, 2019). We filtered and normalized the pancreas data to 2000 genes using the scVelo package (*Bergen et al.*, 2020). We also obtained the latent time and G2M and S cell cycle scores for each cell.

2.5.2 Simulated scRNA-seq Datasets

To simulate data with Splatter (*Zappia et al.*, 2017), we first estimated simulation parameters to match the Tabula Muris data. We set the differential expression probability, factor location, factor scale and common biological coefficient of variation to be $(0.5, 0.01, 0.5, 0.1)$. We then used Splatter to simulate gene expression data of $10,000$ cells with four underlying ground-truth variables: batch, path, step and library size.

Using the PROSSTT package, we simulated 2000 genes across $10,500$ (three trajectories), $10,800$ (four trajectories) and $11,000$ cells (five trajectories). We followed the steps and parameter settings in the PROSSTT tutorial (`https://github.com/soedinglab/prosstt/blob/master/examples/many_branches_cells.ipynb`), varying only the numbers of branches, cells and genes.

2.5.3 InfoGAN and ssInfoGAN

The Information Maximizing Generative Adversarial Networks (InfoGAN) framework extends the regular GAN to encourage disentanglement (*Chen et al.*, 2016). The InfoGAN decomposes the latent variables into latent code $\boldsymbol{C}$ and noise $\boldsymbol{Z}$. To encourage disentanglement, InfoGAN maximizes the mutual information between the latent code and the generated data. To estimate mutual information, InfoGAN relies on an additional network Q that takes generated data as input and predicts the code $Q(\boldsymbol{C} \mid \boldsymbol{X})$ that generated the data. $Q(\boldsymbol{C} \mid \boldsymbol{X})$ is very similar to an encoder in a VAE and estimates a posterior distribution in the same way as the prior distribution of the code $p(\boldsymbol{C})$. InfoGAN then maximizes mutual information between the code and generated data with the following loss functions for the discriminator and generator:

$$\min_{G,Q} \max_{D} L(D, G, Q) = \min_{G,Q} \max_{D} \big\{ L_{\text{GAN}}(G, D) - \lambda_{\text{MI}} L_{\text{MI}}(G, Q) \big\},$$

where $L_{\text{MI}}(G, Q) = \mathbb{E}_{\boldsymbol{C} \sim p(\boldsymbol{C}), \boldsymbol{X} \sim G(\boldsymbol{C}, \boldsymbol{Z})} \{\log Q(\boldsymbol{C} \mid \boldsymbol{X})\} + H(\boldsymbol{C})$ is a lower bound for the mutual information between $\boldsymbol{C}$ and $\boldsymbol{X}$ and $H(\boldsymbol{C})$ is the entropy of the codes. We implemented InfoGAN with the Wasserstein distance, which we refer to as InfoWGAN-GP. We chose a factorized normal distribution with unit variance for $Q(\boldsymbol{C} \mid \boldsymbol{X})$ (the unit variance stabilizes InfoGAN training (*Chen et al.*, 2016; *Lin et al.*, 2019)).

InfoGAN architecture can also be extended to semi-supervised InfoGAN (ssInfoGAN), if labels are available for some or all of the data points (*Spurr et al.*, 2017). The ssInfoGAN maximizes mutual information not only between the generated data and the codes, but also between the real data and corresponding labels. This guides the learned codes to reflect the label information.

2.5.4 Disentanglement Metrics

2.5.4.1 Mutual Information Gap

Following *Chen et al.* (2018), we measure the disentanglement performance of the representations using MIG. Denote $p(V_k)$ and $p(\boldsymbol{X} \mid V_k)$ as the probability of a ground-truth variable V_k and the conditional probability of the data $\boldsymbol{X}$ under V_k. Given $q_{\boldsymbol{\phi}}(Z_j, V_k) = \int_{\boldsymbol{X}} p(V_k)p(\boldsymbol{X} \mid V_k)q_{\boldsymbol{\phi}}(Z_j \mid \boldsymbol{X})d\boldsymbol{X}$, the mutual information between a latent variable Z_j and a ground-truth variable V_k is defined as

$$I(Z_j, V_k) = \mathbb{E}_{q_{\boldsymbol{\phi}}(Z_j, V_k)}\big\{\log \int_{\boldsymbol{X} \in \mathcal{X}_{V_k}} q_{\boldsymbol{\phi}}(Z_j \mid \boldsymbol{X})p(\boldsymbol{X} \mid V_k)d\boldsymbol{X}\big\} + H(Z_j),$$

where $\mathcal{X}_{V_k}$ is the support of $p(\boldsymbol{X} \mid V_k)$ and $H(Z_j)$ is the entropy of Z_j. Due to the different variabilities of the ground-truth variables, the normalized mutual information is better when used with a normalization term of $H(V_k)$, the entropy of V_k. The posterior distribution $q_{\boldsymbol{\phi}}(Z_j \mid \boldsymbol{X})$ is obtained from the encoder (for VAEs) or the derived posterior distribution for probabilistic PCA (*Bishop*, 2006). With K ground-truth variables $\{V_1, \ldots, V_k\}$, the mutual information gap (MIG) is further defined as

$$\text{MIG} = \frac{1}{K}\sum_{k=1}^{K} \frac{1}{H(V_k)} \left\{ I(Z_{j^{(k)}}, V_k) - \max_{j \neq j^{(k)}} I(Z_j, V_k) \right\},$$

where $j^{(k)} = \arg\max_j I(Z_j, V_k)$.

The MIG metric is the average difference between the largest and the second largest normalized mutual information value across all ground-truth variables. Intuitively, this indicates how much each ground-truth variable is captured by a single latent variable. As described in *Chen et al.* (2018), the MIG metric has the axis-alignment property and is unbiased for all hyperparameter settings.

2.5.4.2 FactorVAE Metric

For completeness, we also calculated the disentanglement metric introduced in the FactorVAE paper (*Kim and Mnih*, 2018). In each of the multiple repetitions, we first randomly chose a ground-truth variable and then generate data, keeping this variable fixed and other variables at random. We normalized each dimension by the empirical standard deviation over the whole data and chose the dimension with the lowest empirical variance. The dimension with the lowest empirical variance and the fixed ground-truth variable are then used as (j, k) pairs to train a majority vote classifier. The FactorVAE disentanglement metric is defined as the accuracy of the resulting classifier.

2.5.4.3 Spearman Correlation

Inspired by the MIG metric, we also utilized the Spearman correlation to quantify disentanglement performance. Although the Spearman correlation is a more restricted metric of statistical dependence than mutual information, it has the advantage of being computed without a distributional estimate of a latent representation, which is not available for GAN models. Given the Spearman correlation $S = \text{cor}(Z_j, V_k)$ between inferred representation Z_j and ground-truth variable V_k, we define the corresponding correlation gap as $|\text{cor}(Z_{j^{(k)}}, V_k)| - \max_{j \neq j^{(k)}} |\text{cor}(Z_j, V_k)|$, where $j^{(k)} = \arg\max_j |\text{cor}(Z_j, V_k)|$.

2.5.5 Generation Metrics

2.5.5.1 Random Forest Error

We follow the random forest error metric introduced in the cscGAN paper (*Marouf et al.*, 2020) to quantify how difficult it is for a random forest classifier to distinguish generated cells from real cells. A higher random forest error indicates that the gen-

erated samples are more realistic. We randomly sample 3000 cells and generate 3000 additional cells. Then we train a random forest classifier on the 50 principal components of the 6000 cells to predict whether each cell is real or fake. We train with 5-fold cross validation and report the average error across the 5 folds.

2.5.5.2 Inception Score

We also define an inception score metric similar to the one widely used in evaluating performance on image data (*Barratt and Sharma*, 2018). Intuitively, to achieve a high inception score, a generative model must generate every class in the training dataset (analogous to recall) and every generated example must be recognizable as belonging to a particular class (analogous to precision). We train a random forest classifier on 3000 randomly-sampled real cells to predict their cell types. Based on the trained cell-type classifier, we are able to predict the probabilities of being different cell types for each generated cell. We then input the predicted probabilities to the calculations of the inception score.

2.5.6 Tuning β values in β-TCVAE

The β value is a hyperparameter in the β-TCVAE model that controls the relative importance of penalizing the total correlation of the learned representation. Because β is a hyperparameter in an unsupervised learning approach (no ground truth is available, in general), there is no direct way to pick a single best value for β. This is not a problem unique to the β-TCVAE, but is a general challenge with any unsupervised learning approach. Our best recommendation is to choose a value in the range of 10-50 and use whatever biological prior knowledge is available, such as annotations of cell time point, condition or cell type, to qualitatively assess the disentanglement of representations for different values. One of the best things one can hope for with unsupervised learning algorithms is that the results are robust to different hyperpa-

rameter settings. To show that this is true in this case, we measured disentanglement performance of VAE and β-TCVAE for $\beta = 10$ and 50 on the simulated datasets as shown in Supplementary Figure 2.21. We found that β-TCVAE with $\beta = 10$ or 50 consistently gives a higher MIG than VAE. In short, even if you do not choose the perfect value of β, it is still better to use β-TCVAE than VAE.

2.5.7 Implementation

The VAE-based methods use multilayer perceptron (MLP) units and have two fully-connected (FC) hidden layers with 512 and 256 neurons, followed by separate parameters for mean and variance of the latent representation. The first two hidden layers in the decoder have 256 and 512 neurons, while the last layer gives mean gene expression and has the same number of neurons as the number of genes. Each hidden layer utilizes batch normalization, activated by Rectified Linear Unit (ReLU) or Leaky ReLU. Each hidden layer employs dropout regularization, with a dropout probability of 0.2. We also experimented with three hidden layers for the VAE encoders, but found that the training became unstable. This is consistent with a previous report (*Hu and Greene*, 2019) that found that most VAEs for biological data have only two hidden layers. The GAN-based methods also use MLP for both generator and discriminator. There are three FC hidden layers with 256, 512 and 1024 neurons as well as three hidden layers with 1024, 512 and 10 neurons from data to output. The hidden layers of GANs also have Batch Normalization and ReLU or Leaky ReLU activation. The generator uses dropout regularization with dropout probability of 0.2 for each hidden layer. The VAE-based methods are trained with Adam optimization, while the GAN-based methods are trained with Adam and the gradient prediction method (*Yadav et al.*, 2017). All the hyperparameters of each method on different datasets are tuned for the optimal results.

We trained all models for 1000 epochs and used 10 latent variables. We used

$\beta = 10$ for β-TCVAE on all of the splatter-simulated scRNA-seq datasets, except that we used $\beta = 5$ for β-TCVAE with four latent dimensions on the simulated data with linear step. We used $\beta = 50$ for β-TCVAE on the PROSSTT-simulated datasets and pancreas dataset. For the two real scRNA-seq datasets, we used $\beta = 100$. We used 118-dimensional Gaussian noise for MichiGAN. All models were implemented in TensorFlow.

2.5.8 Supplementary Tables and Figures

Table 2.5: Number of cells for each the cell type/drug combinations selected from the sci-Plex dataset.

		Cell Type		
Pathway	Treatment	A549	K562	MCF7
Protein Tyrosine Kinase	S1010	1014	800	1548
Angiogenesis	S1021	791	506	1626
PI3K/Akt/mTOR	S1044	700	800	1530
Others	S1045	882	787	1849
Cytoskeletal Signaling	S1090	1934	643	1928
Epigenetics	S1096	879	450	1688
Apoptosis	S1130	216	694	79
Neuronal Signaling	S1168	1009	1006	1984
Stem Cells & Wnt	S1180	934	934	1854
Endocrinology & Hormones	S1191	957	982	1945
DNA Damage	S1192	808	803	1392
GPCR & G Protein	S1259	1061	1083	1682
Proteases	S1261	874	964	1759
Cell Cycle	S1529	1499	739	1077
Metabolism	S1628	902	1162	1899
MAPK	S2673	724	730	1823
TGF-beta/Smad	S7207	2195	759	1431
JAK/STAT	S7259	2846	875	2014

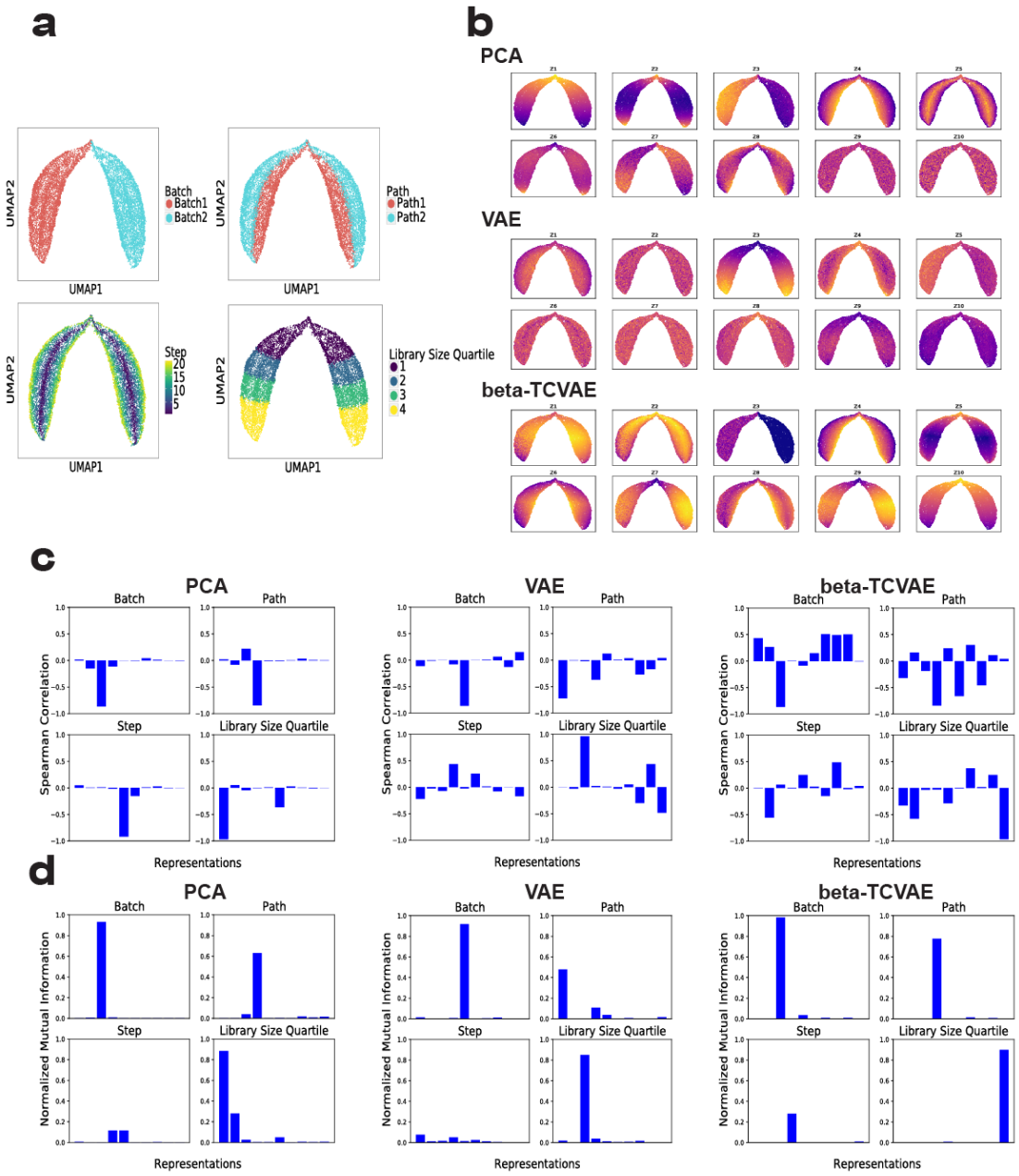

Figure 2.8: Evaluating disentanglement performance on a simulated dataset with linear step. **a** UMAP plots of simulated data colored by batch, path, step and library size quartile. **b** UMAP plots of data colored by the 10 latent variables learned by PCA, VAE and β-TCVAE. **c** Bar plots of Spearman correlations between 10 latent variables and each of the four ground-truth variables for PCA, VAE and β-TCVAE. **d** Bar plots of normalized mutual information between 10 latent variables and each of the four ground-truth variables for PCA, VAE and β-TCVAE.

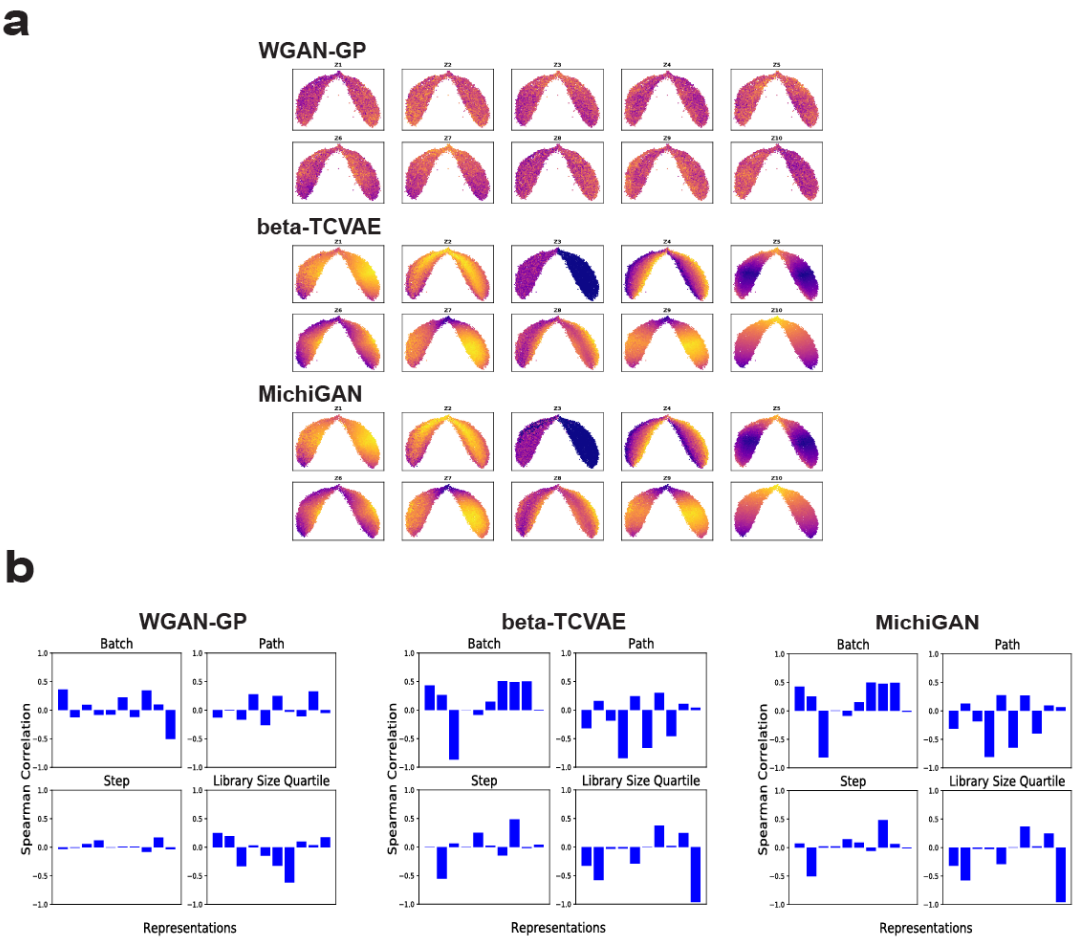

Figure 2.9: Disentanglement and generation performance of WGAN-GP, β-TCVAE and MichiGAN. **a** UMAP plots of real data colored by the 10 representations of β-TCVAE and generated data colored by the 10 representations of WGAN-GP and MichiGAN on the simulated data with linear step. The β-TCVAE panel is reproduced from Supplementary Figure 2.8b for clarity. **b** Bar plots of Spearman correlations between 10 representations and each of the four ground-truth or inferred variables for WGAN, β-TCVAE and MichiGAN on the simulated data with linear step. The β-TCVAE panel is reproduced from Supplementary Figure 2.8c for clarity.

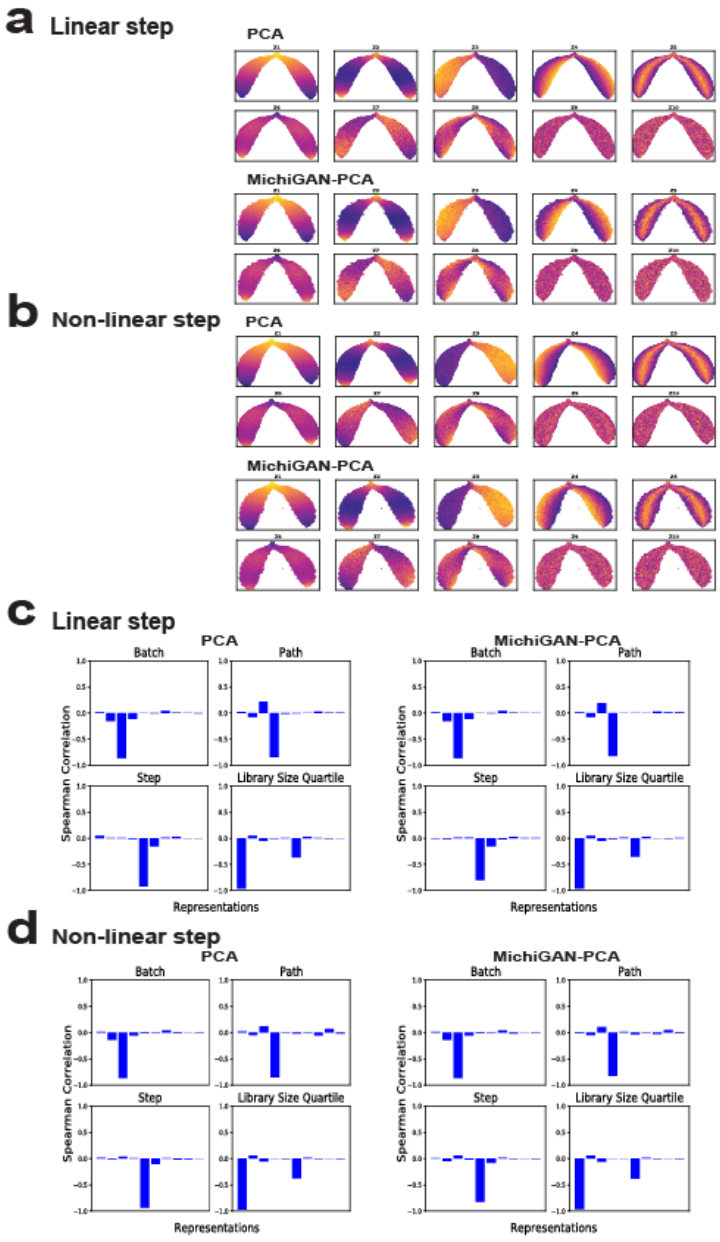

Figure 2.10: Disentanglement performance of PCA and MichiGAN-PCA. **a** UMAP plots of real data colored by 10 representations of PCA and generated data colored by the MichiGAN-PCA representations on the simulated data with linear step. **b** UMAP plots of real data colored by 10 representations of PCA and generated data colored by the MichiGAN-PCA representations on the simulated data with non-linear step. **c** Bar plots of Spearman correlations between 10 representations and each of the four ground-truth or inferred variables for PCA and MichiGAN-PCA on the simulated data with linear step. **d** Bar plots of Spearman correlations between 10 representations and each of the four ground-truth or inferred variables for PCA and MichiGAN-PCA on the simulated data with non-linear step. The PCA panels are reproduced from Figure 2.2b-c and Supplementary Figure 2.8b-c for clarity.

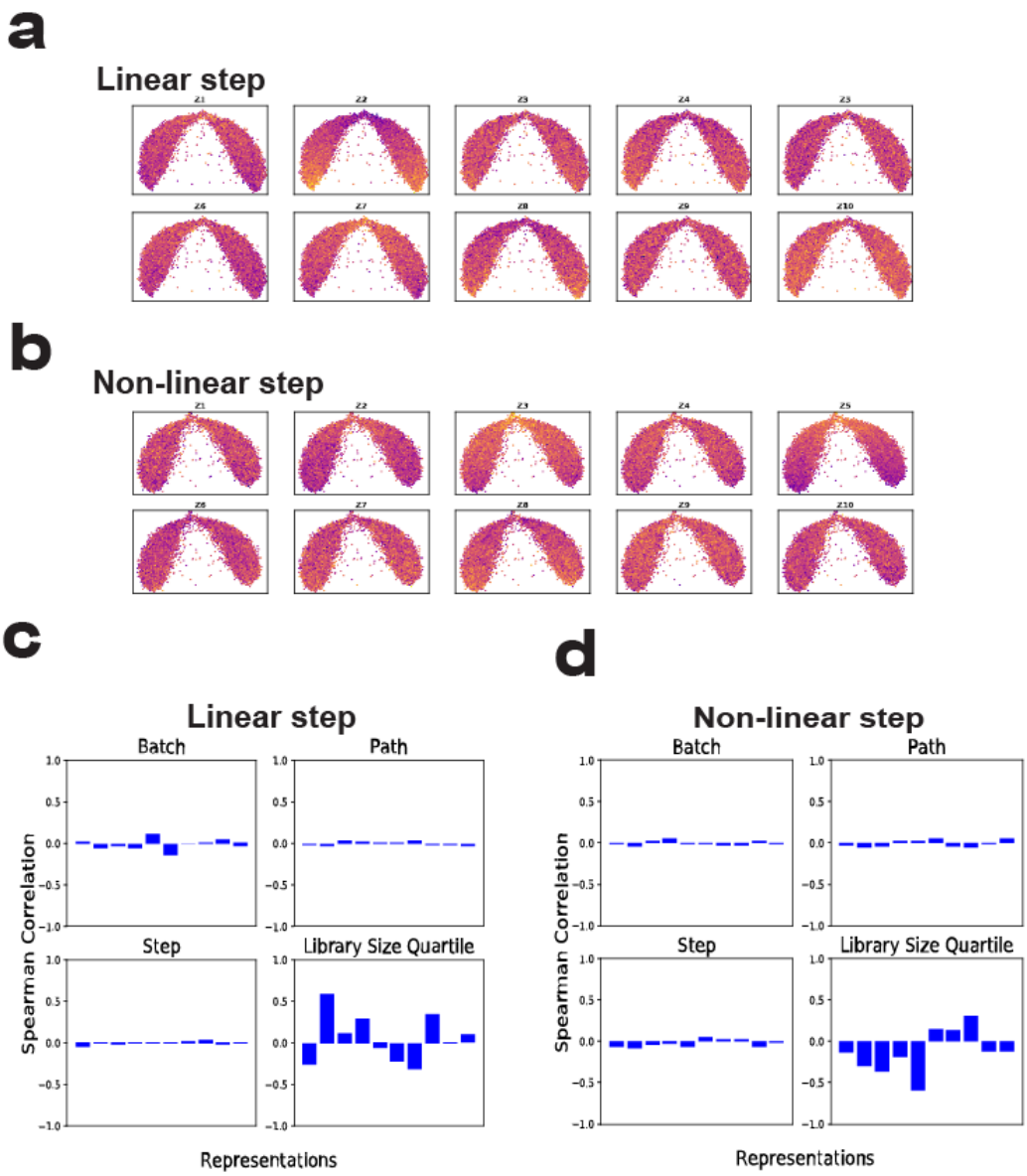

Figure 2.11: Representations learned by InfoWGAN-GP from the simulated single-cell data. **a** UMAP plots of the simulated data with linear step colored by the 10 representations learned by InfoWGAN-GP. **b** UMAP plots of the simulated data with non-linear step colored by the 10 representations learned by InfoWGAN-GP. **c** Bar plots of Spearman correlations between 10 representations and each of the four ground-truth variables for InfoWGAN-GP on the simulated data with linear step. **d** Bar plots of Spearman correlations between 10 representations and each of the four ground-truth variables for InfoWGAN-GP on the simulated data with non-linear step.

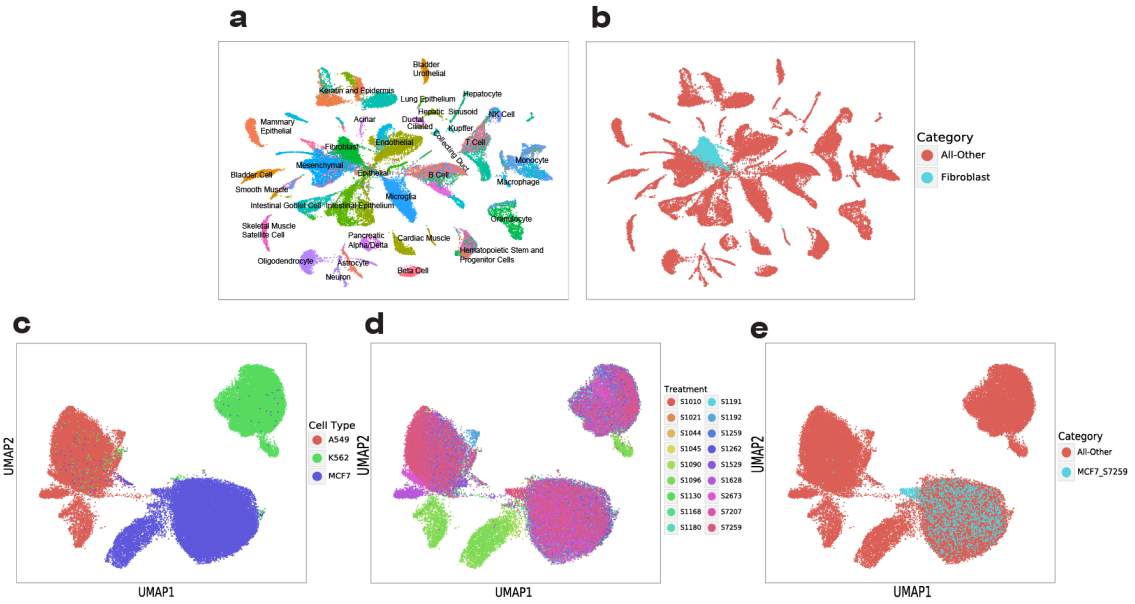

Figure 2.12: The whole Tabula Muris data and the large sci-Plex data. **a** UMAP plot of the whole Tabula Muris data colored by cell type. **b** UMAP plot of the 2026 fibroblast cells in the heart within the whole Tabula Muris data. **c** UMAP plot of the sci-Plex data colored by cell type. **d** UMAP plot of the sci-Plex data colored by drug treatment. **e** UMAP plot of the 2014 cells with MCF7 cell type and S7259 treatment within the sci-Plex data. For clarity, **c** and **d** are reproduced from Figure 2.6a.

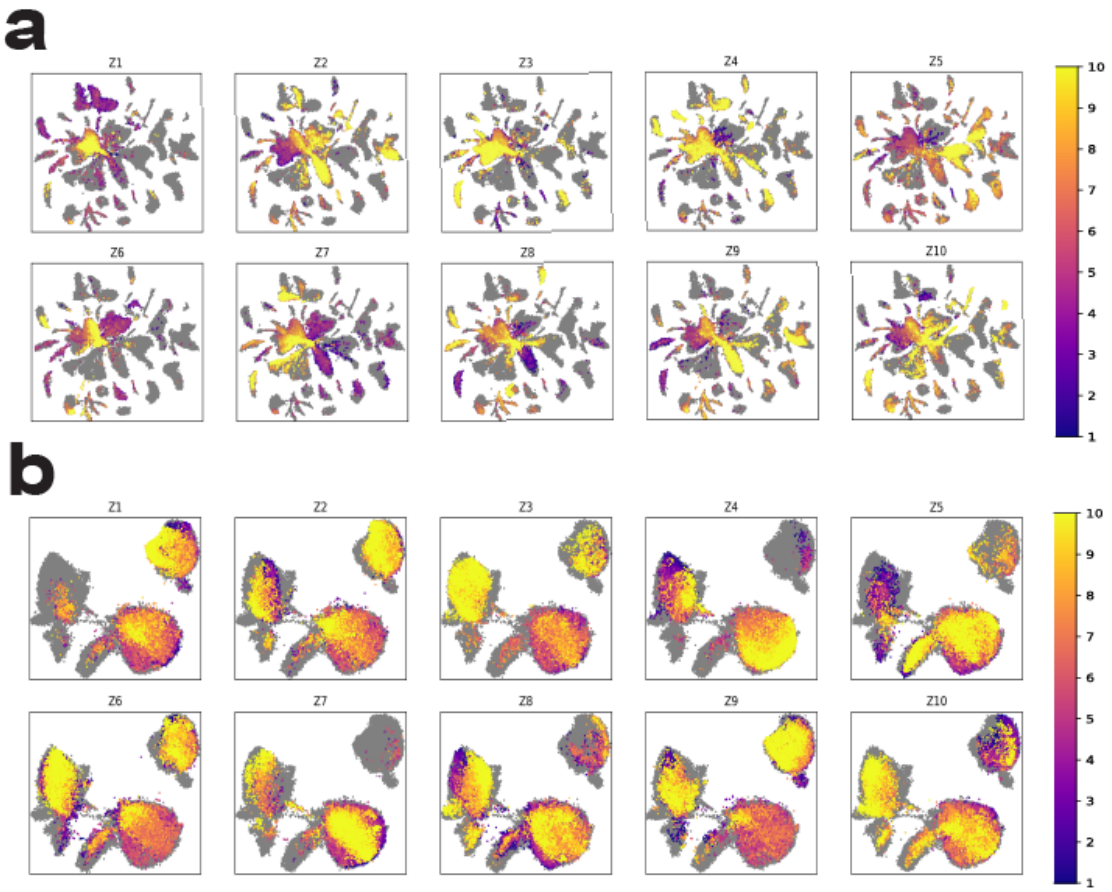

Figure 2.13: UMAP plots of data generated via latent traversals. **a** UMAP plot of latent traversals of the 10 representations of latent values that generate data closest to fibroblast cells in heart within the Tabula Muris data using WGAN-GP with 10 dimensions. **b** UMAP plot of latent traversals of the 10 representations of latent values that generate data closest to MCF7-S7259 cells within the sci-Plex data using WGAN-GP with 10 dimensions.

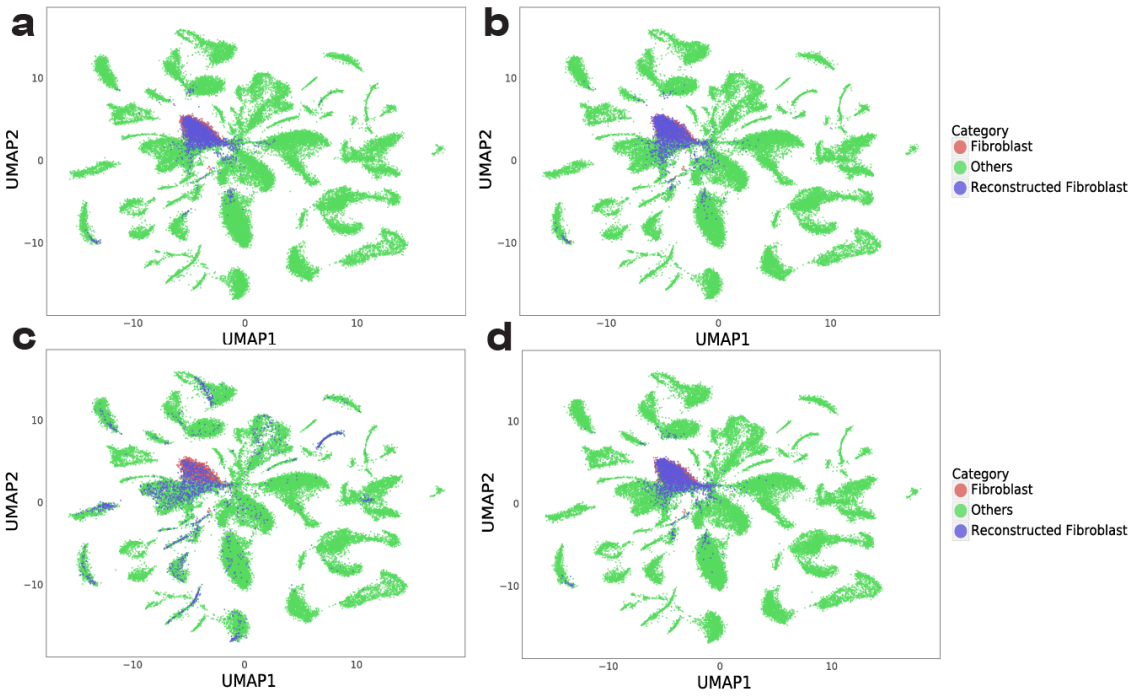

Figure 2.14: Comparison of conditional GAN strategies. **a** UMAP plots of reconstructed cardiac fibroblast cells using β-TCVAE. **b** UMAP plots of reconstructed cardiac fibroblast cells using MichiGAN with PCWGAN-GP. **c** UMAP plots of reconstructed cardiac fibroblast cells using MichiGAN with ssInfoWGAN-GP. **d** UMAP plots of reconstructed cardiac fibroblast cells using MichiGAN with CWGAN-GP.

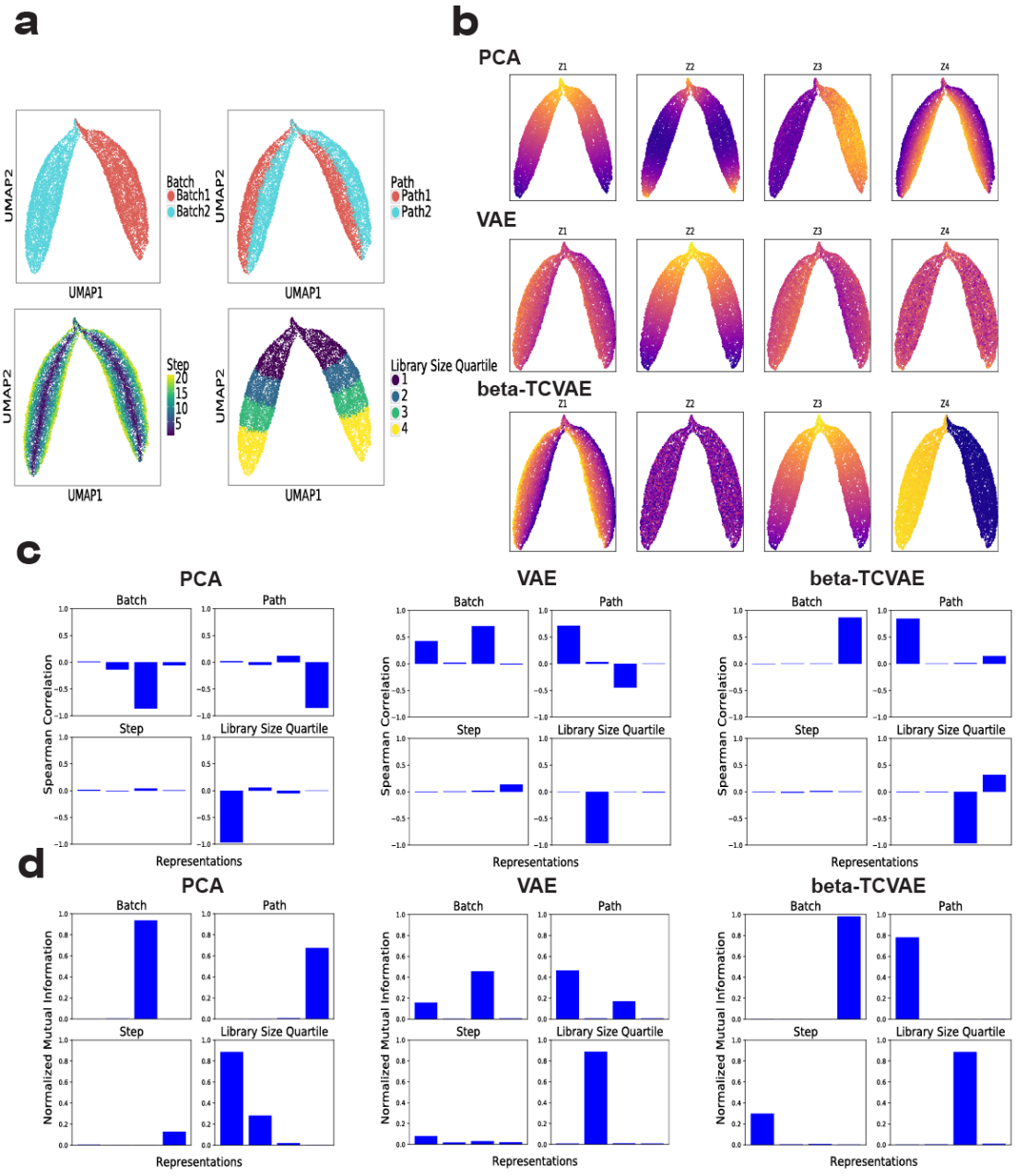

Figure 2.15: Evaluating disentanglement performance on simulated dataset with non-linear step. **a** UMAP plots of simulated data colored by batch, path, step and library size quartile. **b** UMAP plots of data colored by the four latent variables learned by PCA, VAE and β-TCVAE. **c** Bar plots of Spearman correlations between four latent variables and each of the four ground-truth variables for PCA, VAE and β-TCVAE. **d** Bar plots of normalized mutual information between four latent variables and each of the four ground-truth variables for PCA, VAE and β-TCVAE. For clarity, **a** is reproduced from Figure 2.2a.

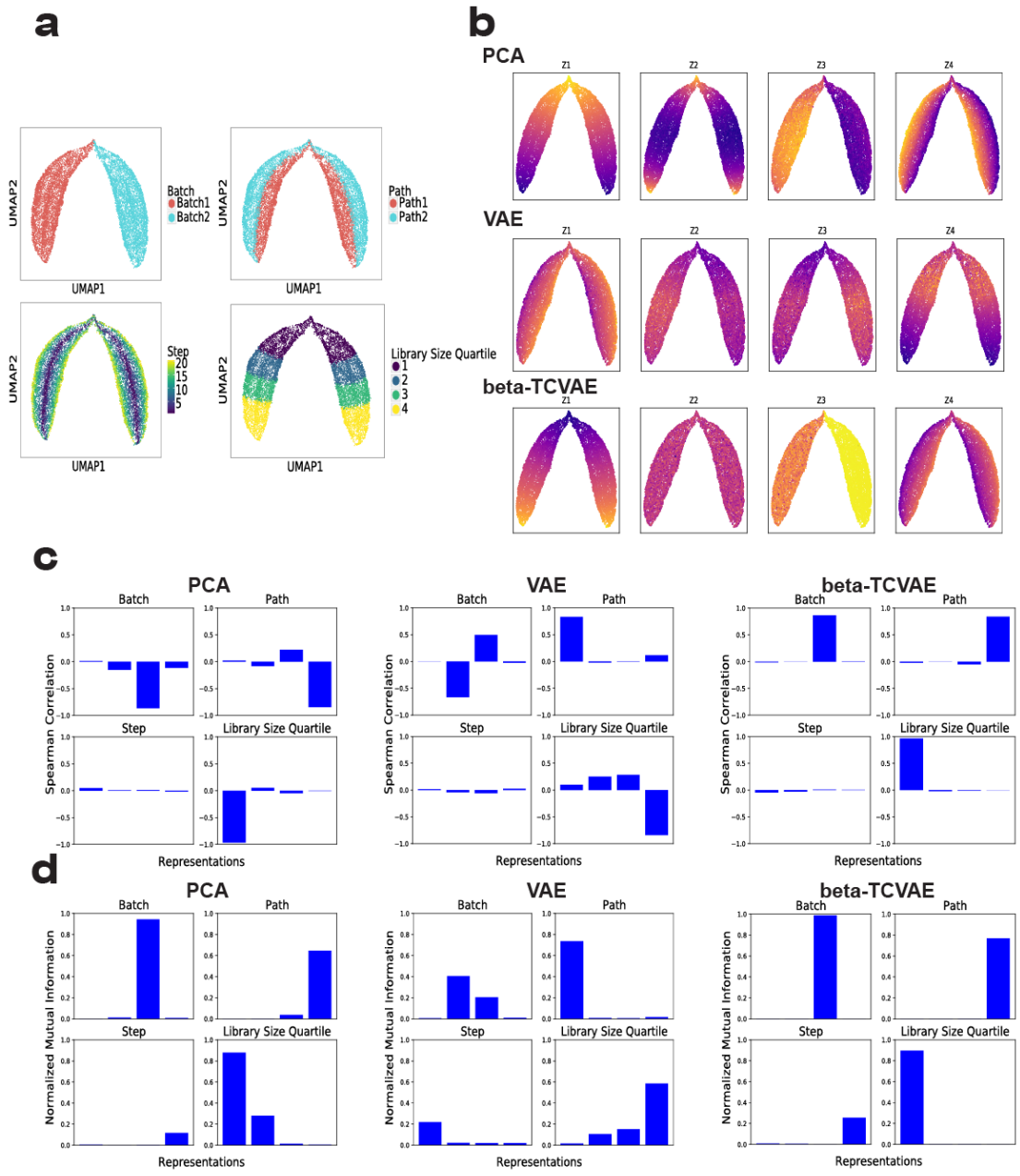

Figure 2.16: Evaluating disentanglement performance on simulated dataset with linear step. **a** UMAP plots of simulated data colored by batch, path, step and library size quartile. **b** UMAP plots of data colored by the four latent variables learned by PCA, VAE and β-TCVAE. **c** Bar plots of Spearman correlations between four latent variables and each of the four ground-truth variables for PCA, VAE and β-TCVAE. **d** Bar plots of normalized mutual information between four latent variables and each of the four ground-truth variables for PCA, VAE and β-TCVAE. For clarity, **a** is reproduced from Figure 2.8a.

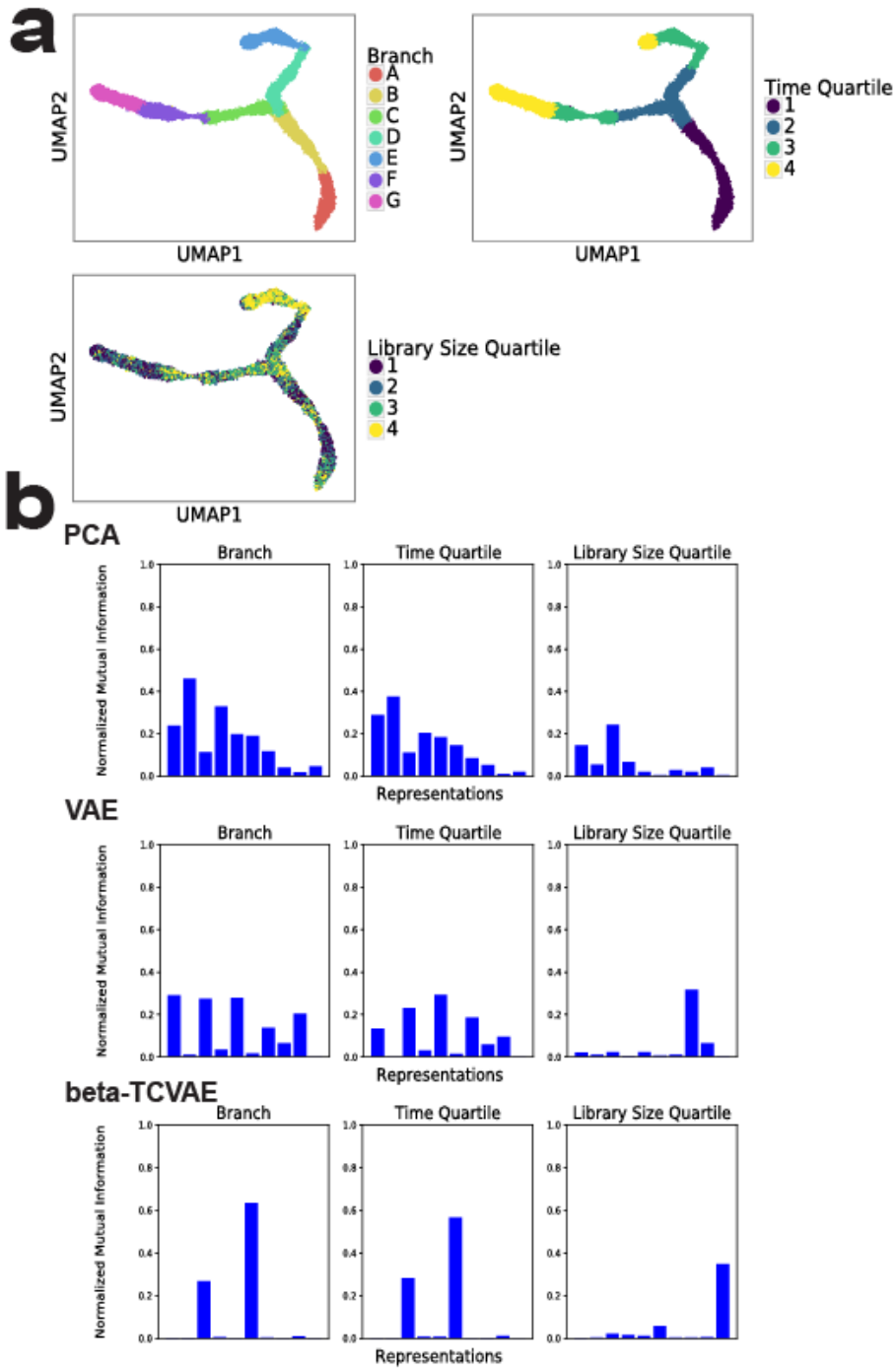

Figure 2.17: Evaluating disentanglement performance on simulated dataset by PROSSTT with three main trajectories. **a** UMAP plots of simulated data colored by branch, time quartile and library size quartile. **b** Bar plots of normalized mutual information between 10 latent variables and each of the three ground-truth variables for PCA, VAE and β-TCVAE.

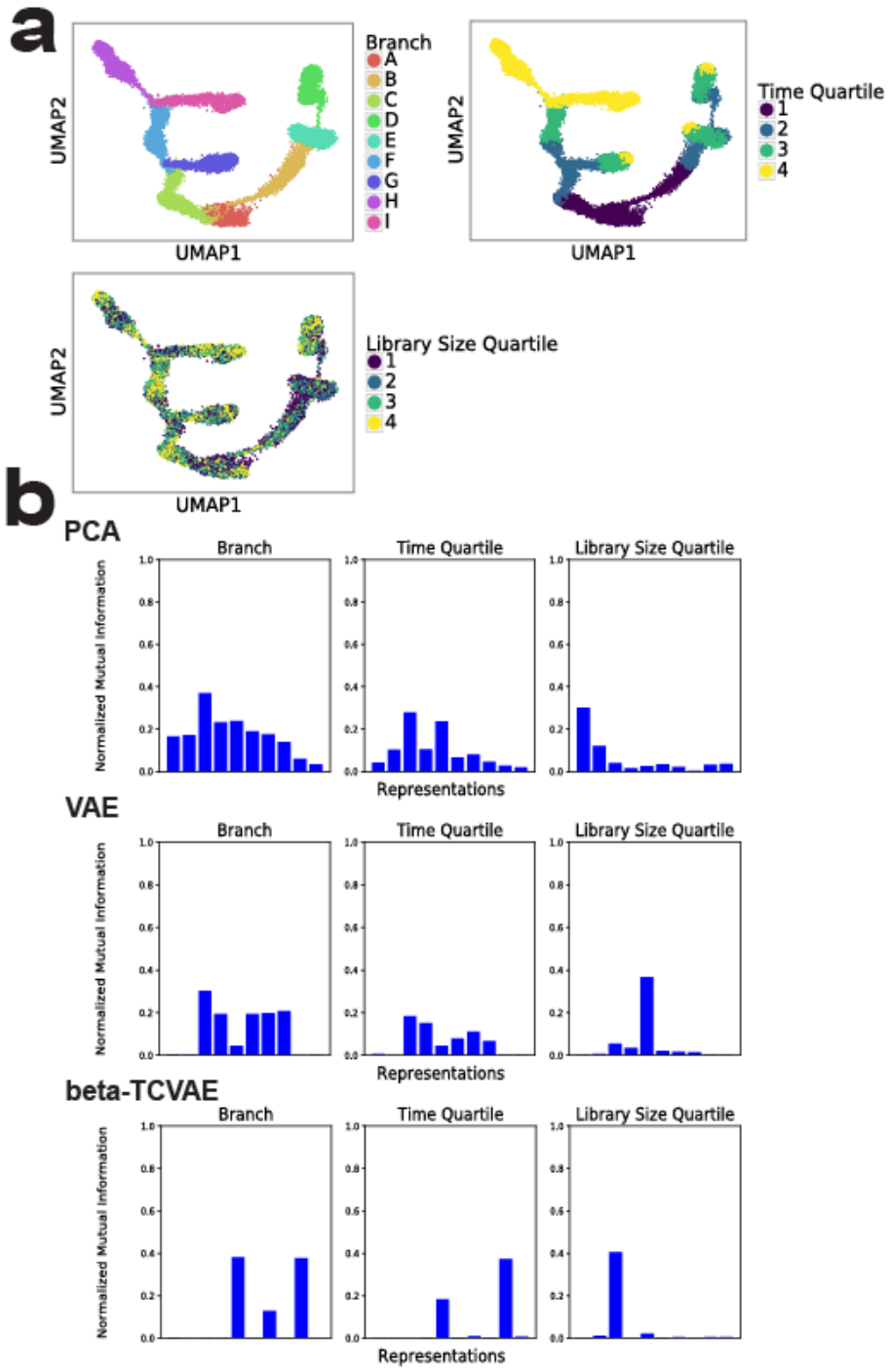

Figure 2.18: Evaluating disentanglement performance on simulated dataset by PROSSTT with four main trajectories. **a** UMAP plots of simulated data colored by branch, time quartile and library size quartile. **b** Bar plots of normalized mutual information between 10 latent variables and each of the three ground-truth variables for PCA, VAE and β-TCVAE.

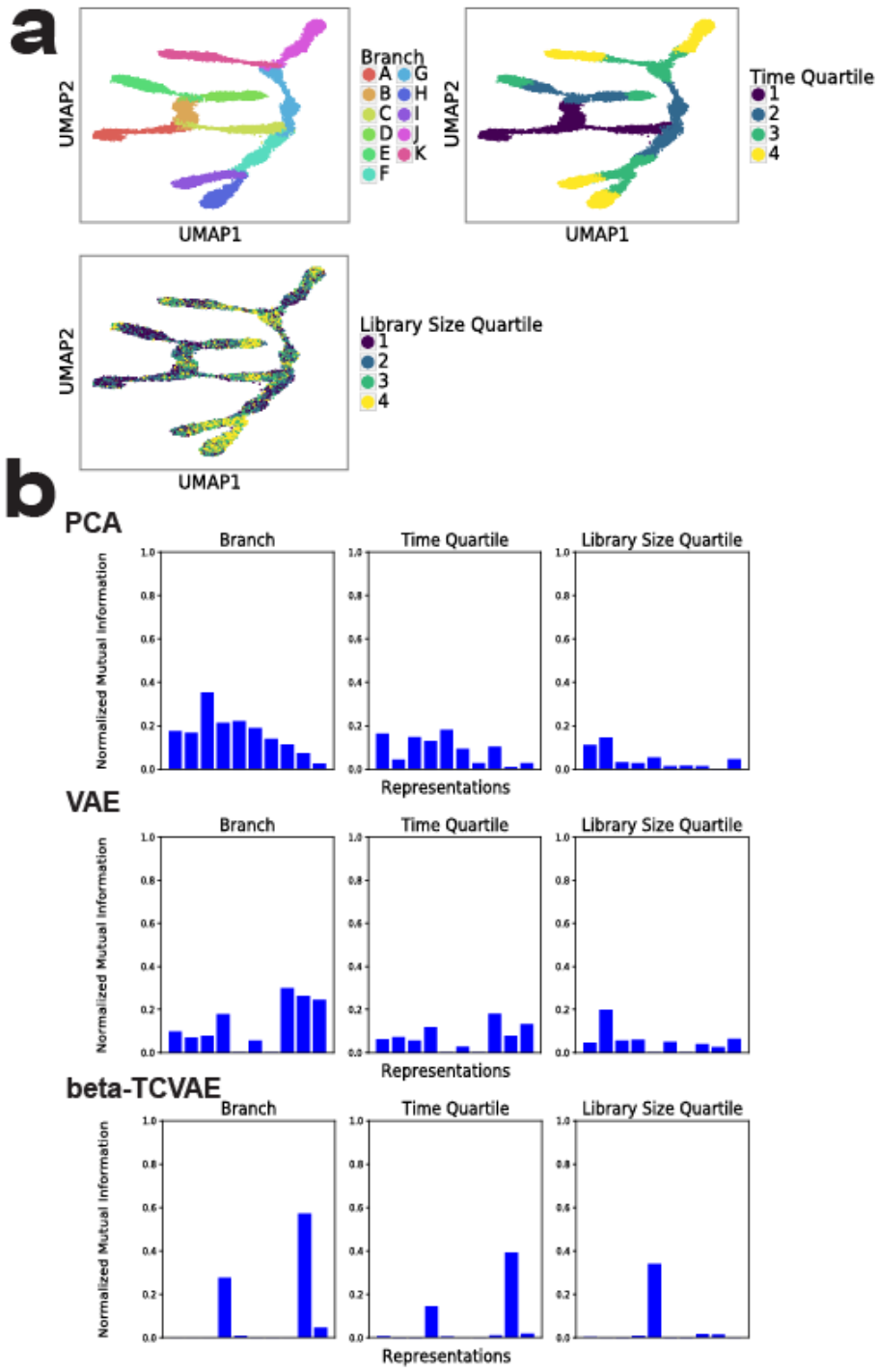

Figure 2.19: Evaluating disentanglement performance on simulated dataset by PROSSTT with five main trajectories. **a** UMAP plots of simulated data colored by branch, time quartile and library size quartile. **b** Bar plots of normalized mutual information between 10 latent variables and each of the three ground-truth variables for PCA, VAE and β-TCVAE.

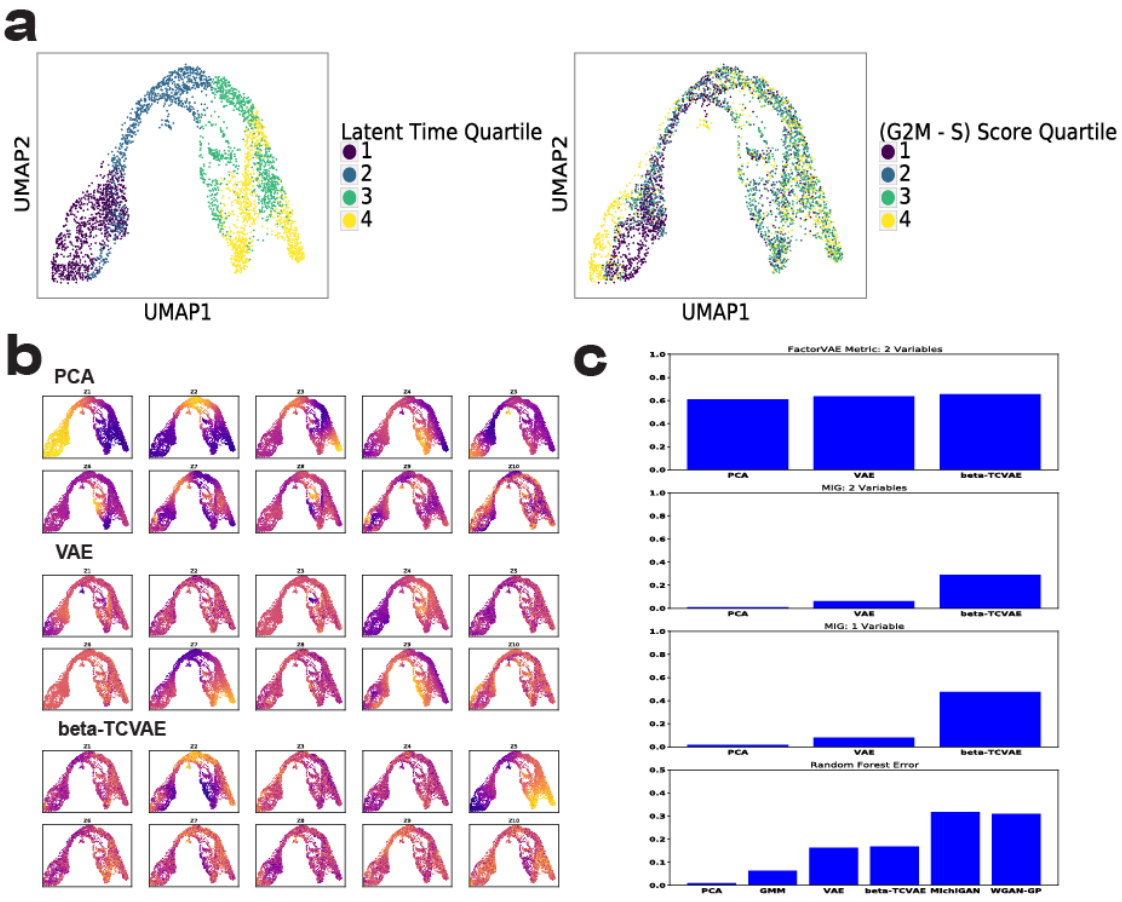

Figure 2.20: Disentanglement and generation performance on pancreas endocrinogenesis. **a** UMAP plots of data colored by latent time quartile and the quartile of the difference between the G2M and S cycle scores. **b** UMAP plots of data colored by the 10 latent variables learned by PCA, VAE and β-TCVAE. **c** Bar plots of FactorVAE metric, MIG for PCA, VAE and β-TCVAE, as well as random forest error for PCA, GMM, VAE, β-TCVAE, MichiGAN and WGAN-GP. For clarity, the 2 variables refer to latent time quartile and (G2M - S) score quartile, and the 1 variable means only latent time quartile.

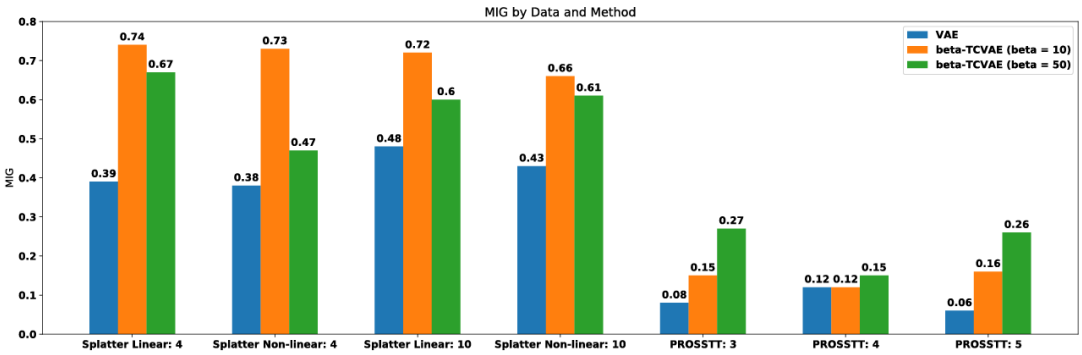

Figure 2.21: Robustness of disentanglement performance: MIG of VAE and β-TCVAE ($\beta = 10, 50$) on simulated datasets.

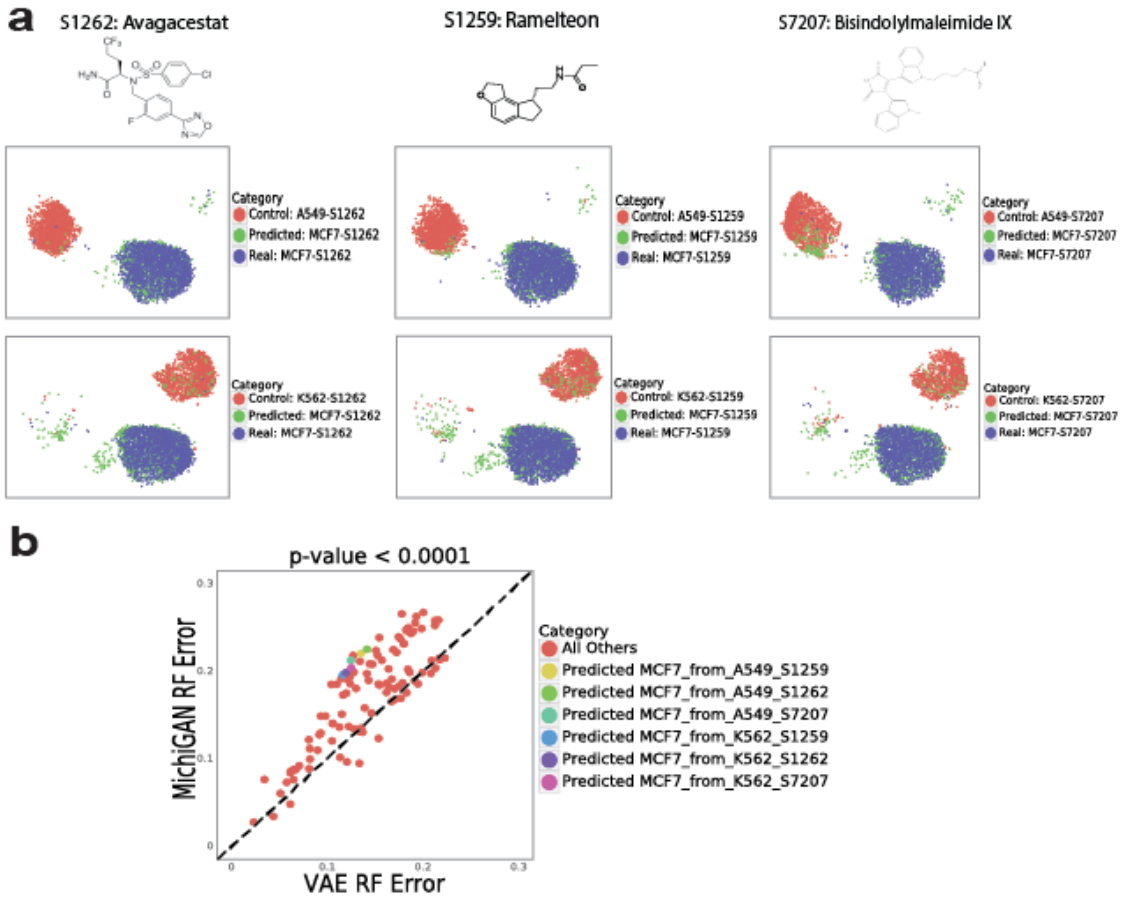

Figure 2.22: MichiGAN based on VAE predicts unseen or observed combinations in the sci-Plex dataset. **a** UMAP plots of the predicted (green), real (blue) and control (red) cells for six predictions of the three missing combinations of MCF7-S1262, MCF7-S1259 and MCF7-S7207. **b** Random forest errors values for MichiGAN trained on VAE and VAE alone after selecting held-out combinations with low ΔH.

CHAPTER III

Predicting Single-Cell Responses to Drug Perturbations

3.1 Introduction

Recent experimental developments have enabled high-throughput single-cell molecular measurement of response to drug treatment. A high-throughput chemical screen experiment usually involves a large number of cells and multiple treatments, where each cell receives a kind of drug treatment and is impacted in a distinct manner (*Gehring et al.*, 2020; *Srivatsan et al.*, 2020).

Understanding how drugs influence cellular responses helps discover treatments with desired effects, potentially benefiting a myriad of therapeutic applications. Various methods have been developed to predict cellular responses under drug treatments, including mechanistic models on protein level changes and phenotype changes (*Yuan et al.*, 2021), as well as deep learning models on disease outcomes (*Cheng et al.*, 2019; *Kuenzi et al.*, 2020).

In this study, we are interested in predicting the conditional distribution of cell states given different drug treatments. In a particular tissue under homeostatic conditions, there is a wild-type distribution of cellular gene expression states $p(\boldsymbol{X})$. However, treating cells with a drug G changes their cell state distribution. Our goal is

to shape these treatment-specific data distributions $p(\boldsymbol{X} \mid G)$ by developing deep generative models to sample from the cell state distribution for any drug treatment.

Several deep generative models have generated realistic single-cell data from drug treatments. *Lotfollahi et al.* (2020) proposed a conditional variational autoencoder (VAE) framework with representations under two treatment conditions balanced using a similarity score of their counterfactual inference developed by *Johansson et al.* (2016). In contrast, scGen predicts single-cell data through latent space vector arithmetic in an unsupervised learning fashion (*Lotfollahi et al.*, 2019). Another method of Dr.VAE (*Rampášek et al.*, 2019) also explored the dependency of the latent space on treatments. These methods, however, can only apply to drug treatment experiments with two treatment conditions, and they are unable to make predictions for unseen drug treatments. In response, *Lotfollahi et al.* (2021) proposed compositional perturbation autoencoder (CPA) to extract a basal state in the latent space of VAE, and to further predict latent values under drug treatments using a linear model. However, CPA assumes that it is possible to learn the effect of a perturbation independent from the cell state, which is probably invalid in many cases. For example, if a perturbation G selectively kills cells in state A, the model will incorrectly generate A cells under perturbation G. In contrast, our proposed method can model general relationships between drug treatment and cell state, whether dependent or independent.

In this chapter, we propose PerturbNet, a novel and flexible framework that predicts single-cell responses to different perturbations. The PerturbNet model connects drug treatment information and latent space through normalizing flows (*Papamakarios et al.*, 2021), enabling the translation between drug treatment domain and single-cell domain (*Baltrušaitis et al.*, 2018). The PerturbNet framework is generally applicable to any type of data with drug treatments, such as those of single-cell RNA-seq (scRNA-seq) data. Furthermore, PerturbNet can make predictions for both observed and unseen drug treatments.

3.2 Methods

3.2.1 Drug Treatment Encoder and ChemicalVAE

The commonly used one-hot encoding approach can transform drug treatment labels to a vector of 1's and 0's, but it needs pre-specifying the total number of possible drug treatments and cannot encode new treatments after the specification. Therefore, we consider flexible representations $\boldsymbol{Y}$ for drug treatments to predict drug treatment effects on single-cell data for unseen perturbations.

A drug treatment contains abundant information more than just a label such as 'S1096'. Its pharmacological properties are usually determined by its chemical structure. We thus aim to encode drugs' chemical structures to dense representations. We consider drug treatments' simplified molecular-input line-entry system (SMILES) strings, which distinctively represent chemical structures and treatment information. Although SMILES strings can be encoded to numerical representations through molecular Morgan fingerprints (*Rogers and Hahn*, 2010) or through language models (*Xu et al.*, 2017; *Chithrananda et al.*, 2020), the representations from these methods are deterministic, meaning that the representations remain the same in replicated encoding implementations. Given that a chemical screen experiment usually contains a limited number of distinct drug treatments, the use of stochastic representations of the drug treatments prevents possible model overfitting.

To improve the learning capacity, especially for representations of unseen treatments, we consider using a chemical variational autoencoder (ChemicalVAE) to generate the stochastic sampled representation $\boldsymbol{Y}$ of each drug's SMILES string (*Kusner et al.*, 2017; *Zhu et al.*, 2021). In essence, the ChemicalVAE first transforms and standardizes SMILES strings to their canonical forms and tokenizes each canonical SMILES to be encoded as a one-hot matrix. For a canonical SMILES string, the ith row of its one-hot matrix corresponds to its ith place, and has the jth column

being 1 and all other columns being 0's, if its ith place has the jth character in the collected chemical elements' library. The one-hot matrices of SMILES strings are then fitted into ChemicalVAE which provides representations $\boldsymbol{Y}$ for SMILES strings of drug treatments g. Figure 3.1 summarizes the ChemicalVAE architecture.

3.2.2 Baseline KNN and Random Models

From the perturbation representations, $\boldsymbol{Y}$, of drug treatments, we can learn the relationship of several drug treatments in their latent space. We assume that drug treatments with close latent values tend to also have similar single-cell responses. Thus, the distributions of perturbation responses $p(\boldsymbol{X} \mid G = g_1)$ and $p(\boldsymbol{X} \mid G = g_2)$ are similar if g_1 and g_2 have close representations of $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$.

We then propose our baseline model using the k-nearest neighbors (KNN) algorithm to predict single-cell data under drug treatments in Algorithm 2. From ChemicalVAE, we can obtain the representation $\boldsymbol{Y}$ for a set of treatments $\mathcal{G}$, each of which has measured single-cell samples. Then for a drug treatment $g \notin \mathcal{G}$ with representation $\boldsymbol{y}$, we can find its k nearest neighbors $\{g_{(1)}, \ldots, g_{(k)}\}$ from $\mathcal{G}$ based on $\boldsymbol{Y}$. We then sample single-cell samples treated with the k nearest treatments in proportion to their exponentiated negative distances to the treatment of interest in the latent space of $\boldsymbol{Y}$. The sampled single-cell data can be regarded as a baseline prediction for the single-cell data with the treatment of interest.

Algorithm 2: Baseline KNN Model

Input: Drug treatment of interest g and its representation $\boldsymbol{y}$. A set of drug treatments $\mathcal{G} = \{g_1, \ldots, g_m\}$ with their representations $\{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_m\}$ as well as single-cell sample sets $\{\mathcal{X}_1, \ldots, \mathcal{X}_m\}$.

1. Train KNN algorithm (k = 5) on $\{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_m\}$.
2. Obtain $\boldsymbol{y}$'s k neighbors $\{\boldsymbol{y}_{(1)}, \ldots, \boldsymbol{y}_{(k)}\}$ and their pairwise distances $(d_{(1)}, \ldots, d_{(k)})^T$ from the trained KNN algorithm.
3. Sample a number of cells $\mathcal{X}'$ through stratified sampling with replacement from $\{\mathcal{X}_{(1)}, \ldots, \mathcal{X}_{(k)}\}$. Each set $\mathcal{X}_{(i)}$ has a proportion of $\exp\{-d_{(i)}\} / \sum_{j=1}^{k} \exp\{-d_{(j)}\}$.

Result: predicted single cells $\mathcal{X}'$ under perturbation g.

The key assumption of the baseline KNN model is that the perturbation representation is informative to infer single-cell data. To test the informativeness assumption on the perturbation representation, we propose a naive baseline random model in Algorithm 3 that randomly samples single-cell samples under treatments other than the target treatment. If the perturbation representation is uninformative to inferring cell state or cellular response, the random model is likely to have a similar performance to the KNN model.

Algorithm 3: Baseline Random Model

Input: Drug treatment of interest g. A set of single-cell samples $\{\mathcal{X}_{-g}\}$ receiving drug treatments other than g.

1. Sample a number of cells $\mathcal{X}'$ with replacement from $\mathcal{X}_{-g}$.

Result: predicted single cells $\mathcal{X}'$ under perturbation g.

3.2.3 Conditional Invertible Neural Networks

We consider employing more complex normalizing flows of invertible neural networks to understand the relationship between perturbation representation and cellular responses. An affine coupling block (*Dinh et al.*, 2016) enables the input

$\boldsymbol{U} = (\boldsymbol{U}_1^T, \boldsymbol{U}_2^T)^T$ to be transformed to output $\boldsymbol{W} = (\boldsymbol{W}_1^T, \boldsymbol{W}_2^T)^T$ with:

$$\boldsymbol{W}_1 = \boldsymbol{U}_1 \odot \exp\{\text{scale}_1(\boldsymbol{U}_2)\} + \text{trans}_1(\boldsymbol{U}_2)$$

and

$$\boldsymbol{W}_2 = \boldsymbol{U}_2 \odot \exp\{\text{scale}_2(\boldsymbol{W}_1)\} + \text{trans}_2(\boldsymbol{W}_1),$$

where $\text{scale}_1(\cdot), \text{scale}_2(\cdot), \text{trans}_1(\cdot), \text{trans}_2(\cdot)$ are arbitrary scale and transformation neural networks, and $\odot$ is the Hadamard product or element-wise product. The inverse of the coupling blocking can be represented by

$$\boldsymbol{U}_2 = \{\boldsymbol{W}_2 - \text{trans}_2(\boldsymbol{W}_1)\} \oslash \exp\{\text{scale}_2(\boldsymbol{W}_1)\}$$

and

$$\boldsymbol{U}_1 = \{\boldsymbol{W}_2 - \text{trans}_1(\boldsymbol{U}_2)\} \oslash \exp\{\text{scale}_1(\boldsymbol{U}_2)\},$$

where $\oslash$ is the element-wise division. The affine coupling block allows bijective transformations between $\boldsymbol{U}$ and $\boldsymbol{W}$ with strictly upper or lower triangular Jacobian matrices. A conditional coupling block is further adapted to concatenate a conditioning variable with inputs in scale and transformation networks. A conditional coupling block preserves the invertibility of the block and the simplicity of the Jacobian determinant.

A conditional invertible neural network (cINN, *Ardizzone et al.*, 2019; *Rombach et al.*, 2020) is a type of conditional normalizing flow with conditional coupling blocks and actnorm layers (*Kingma and Dhariwal*, 2018), with both forward and inverse translations. Denote representations from two domains as $\mathbf{Y} \in \mathcal{D}_{\mathbf{Y}}$ and $\mathbf{Z} \in \mathcal{D}_{\mathbf{Z}}$. A cINN modeling $\boldsymbol{Z}$ over $\boldsymbol{Y}$ gives forward translation

$$\boldsymbol{Z} = f(\boldsymbol{V} \mid \boldsymbol{Y})$$

and inverse translation

$$\boldsymbol{V} = f^{-1}(\boldsymbol{Z} \mid \boldsymbol{Y}),$$

where $\boldsymbol{V} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$. cINN effectively models $p(\boldsymbol{Z} \mid \boldsymbol{Y})$, the probabilistic dependency of $\boldsymbol{Z}$ over $\boldsymbol{Y}$ with a residual variable $\boldsymbol{V}$. As a cINN seeks to extract the shared information from $\boldsymbol{Y}$ and add residual information $\boldsymbol{V}$ to generate $\boldsymbol{Z}$, the objective function to train a cINN is the Kullback-Leibler (KL) divergence between the residual's posterior $q(\boldsymbol{V} \mid \boldsymbol{Y})$ and its prior $p(\boldsymbol{V})$. The objective function can further be derived to

$$\begin{aligned}\mathbb{E}_{p(\boldsymbol{Y})}\left[D_{\mathrm{KL}}\{q(\boldsymbol{V} \mid \boldsymbol{Y})||p(\boldsymbol{V})\}\right] = \mathbb{E}_{p(\boldsymbol{Z},\boldsymbol{Y})}\left[-\log p\{f^{-1}(\boldsymbol{V} \mid \boldsymbol{Y})\} - |\det J_{f^{-1}}(\boldsymbol{Z} \mid \boldsymbol{Y})|\right] \\ - H(\boldsymbol{Z} \mid \boldsymbol{Y}),\end{aligned} \tag{3.1}$$

where $\det J_{f^{-1}}$ is the determinant of the Jacobian matrix of f^{-1} and H is a constant entropy. The optimal f that minimizes the objective function in Equation (3.1) gives $q(\boldsymbol{V} \mid \boldsymbol{Y}) = p(\boldsymbol{V})$. In addition, the objective is an upper bound of the mutual information $I(\boldsymbol{V}, \boldsymbol{Y})$. Therefore, a well-trained cINN effectively achieves independence between $\boldsymbol{V}$ and $\boldsymbol{Y}$. cINN has the same parameters for forward and inverse translations, reducing the number of model parameters while still preserving network details in both translation directions, and has been utilized to translate domain representations of images and texts (*Rombach et al.*, 2020).

3.2.4 PerturbNet

From ChemicalVAE, a drug treatment g is represented as a dense variable $\boldsymbol{Y}$; we propose a baseline KNN model to make predictions for single-cell perturbation responses by sampling cells treated by similar perturbations according to their representations. To predict single-cell responses, a more powerful predictive model can

be constructed with deep neural networks on perturbation representations. The deep learning predictive model can potentially give better predicting performance than the baseline KNN model, especially when it is trained with a large number of perturbations.

We thus propose our PerturbNet framework in Figure 3.2. The PerturbNet framework has a VAE-based model for single-cell data and ChemicalVAE for drug treatments. The single-cell VAE model can be scVI (*Lopez et al.*, 2018) for count data or regular VAE for normalized data. The pre-trained single-cell VAE model encodes single-cell sample $\boldsymbol{X}$ to cellular representation $\boldsymbol{Z}$. The pre-trained ChemicalVAE model encodes drug treatment G to perturbation representation $\boldsymbol{Y}$. Then, the perturbation representation $\boldsymbol{Y}$ and cellular representation $\boldsymbol{Z}$ are connected through a conditional invertible neural network (cINN), where residual representation $\boldsymbol{V}$ predicts $\boldsymbol{Z}$ with $\boldsymbol{Y}$ as the conditioning variable. The residual representation is independent of perturbation in predicting cellular representation and is regarded as noise to the perturbation effects on cell state.

In addition, we also consider conditioning on known cell state covariates in modeling cell representation. We concatenate perturbation representation and cell state covariates to serve as an overall condition of cINN and translate between residual or condition-invariant representation $\boldsymbol{V}$ and cellular representation. The extra conditioning on cell state covariates potentially debiases their confounding effects on modeling perturbation effects on cellular representation.

3.2.5 ChemicalVAE Fine-Tuning

As both KNN and PerturbNet methods predict cell state based on perturbation representation, it might enhance the prediction performance for cell state from perturbation to use perturbation representation that learns cellular representation information.

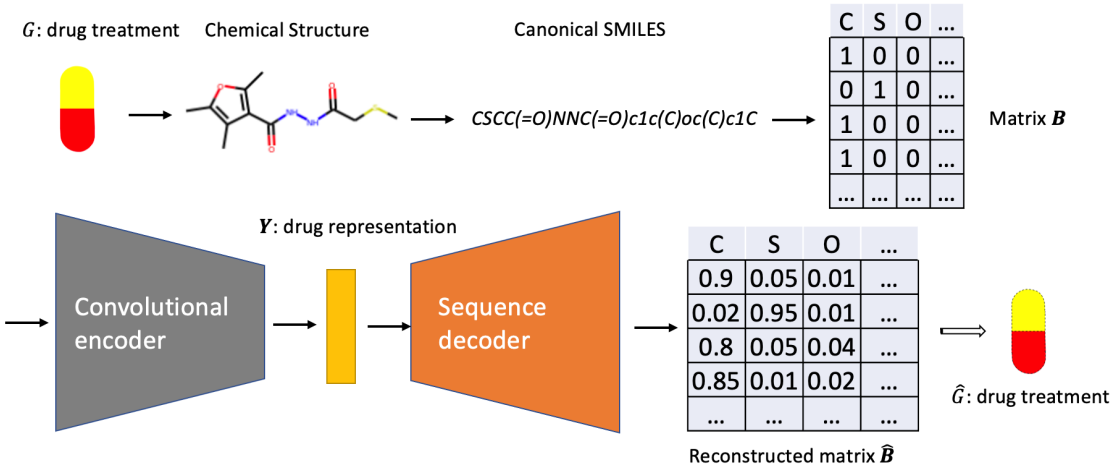

Figure 3.1: Overview of the ChemicalVAE architecture.

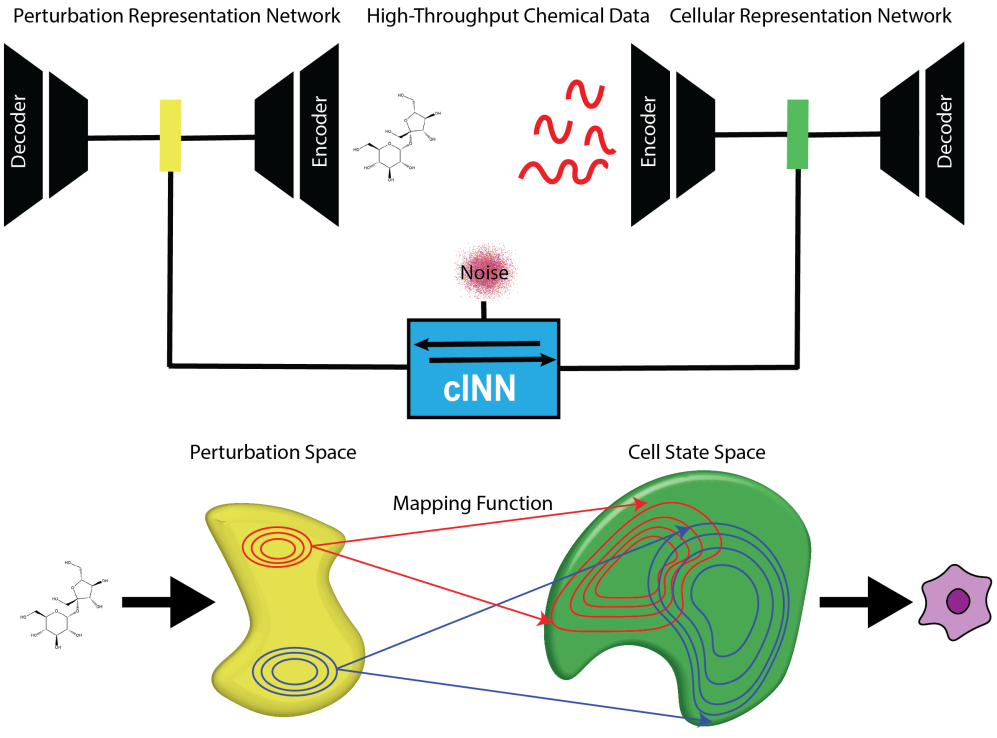

Figure 3.2: Overview of the PerturbNet architecture.

We propose an algorithm to fine-tune ChemicalVAE, by adding the evidence lower bound (ELBO) loss of ChemicalVAE with an extra term for a certain cellular property quantity (*Gómez-Bombarelli et al.*, 2018). In this study, we compute the Wasserstein-2 (W2) distance between cellular representations of each pair of perturbations and penalizing the quantity of the trace of $\boldsymbol{Y}$'s second moment weighted by the Laplacian matrix $\boldsymbol{L}$ of the adjacency matrix defined from pairwise distances (*Cai et al.*, 2010). Denote $\boldsymbol{y}$ and $\boldsymbol{L}$ as the perturbation representations and the Laplacian matrix of the perturbations. The penalizing quantity is defined as trace($\boldsymbol{y}^T\boldsymbol{L}\boldsymbol{y}$). By penalizing the proposed quantity, we expect perturbations with similar cell states to have closer perturbation representations from ChemicalVAE. To implement the fine-tuning algorithm with cell state property, we alternate the ChemicalVAE training with a batch of chemical SMILES strings from a large chemical database with the ELBO loss and another batch of pairs of SMILES strings and cellular representations from a single-cell chemical screen dataset with the penalized ELBO loss. We tune a hyperparameter λ on the extra term to adjust the ChemicalVAE fine-tuning performance. We summarize the ChemicalVAE fine-tuning in Algorithm 4.

Algorithm 4: ChemicalVAE Fine-Tuning

Input: Set of drug treatments $\mathcal{G}_s$ in ZINC or PubChem datasets. Single-cell samples with perturbations $\{(\boldsymbol{x}_1, g_1), \ldots, (\boldsymbol{x}_n, g_n)\}$. The perturbations' cell-state Laplacian matrix $\boldsymbol{L}$. Two Adam optimizers (*Kingma and Ba*, 2014) Adam_1, Adam_2.

1. Initialize parameters $(\boldsymbol{\phi}, \boldsymbol{\theta})$.
2. While $(\boldsymbol{\phi}, \boldsymbol{\theta})$ has not converged:
    1). Sample a batch $\{(\boldsymbol{x}_{(i)}, g_{(i)})\}_{i=1}^m$ from the single-cell samples.
    2). Obtain representations $\boldsymbol{y} = (\boldsymbol{y}_{(0)}, \ldots, \boldsymbol{y}_{(m)})^T$ for $\{g_{(i)}\}_{i=1}^m$.
    3). Obtain the Laplacian matrix $\boldsymbol{L}_g$ for $\{g_{(i)}\}_{i=1}^m$.
    4). Compute gradient $\boldsymbol{g}_{\boldsymbol{\phi},\boldsymbol{\theta}}^{\lambda} = \nabla_{\boldsymbol{\phi},\boldsymbol{\theta}}\{\text{-ELBO}(\boldsymbol{\phi}, \boldsymbol{\theta}) + \lambda \text{trace}(\boldsymbol{y}^T \boldsymbol{L}_g \boldsymbol{y})\}$
    5). Update parameters using $\boldsymbol{g}_{\boldsymbol{\phi},\boldsymbol{\theta}}^{\lambda}$ via Adam_1.
    6). Sample a batch $\{g_{(i)}\}_{i=1}^m$ from $\mathcal{G}_s$
    7). Compute gradient $\boldsymbol{g}_{\boldsymbol{\phi},\boldsymbol{\theta}} = \nabla_{\boldsymbol{\phi},\boldsymbol{\theta}}\{\text{-ELBO}(\boldsymbol{\phi}, \boldsymbol{\theta})\}$
    8). Update parameters using $\boldsymbol{g}_{\boldsymbol{\phi},\boldsymbol{\theta}}$ via Adam_2.

Result: fine-tuned ChemicalVAE with parameters $(\boldsymbol{\phi}, \boldsymbol{\theta})$.

3.2.6 Related Work

There has been limited previous work on predicting single-cell responses to multiple perturbations. A related work is compositional perturbation autoencoder (CPA, *Lotfollahi et al.*, 2021), a VAE-based framework, which also models cellular representation under cellular perturbation. The CPA framework assumes that perturbation and known cell state covariates independently influence cellular representation in a linear model. In contrast, PerturbNet models cellular representation with normalizing flows, which flexibly incorporates perturbation, cell state covariates and also their complex interaction effects. Additionally, the residual representation in PerturbNet preserves stochastic noise that is invariant to the conditioning representation in a single-cell sample, while CPA relies on encoding single-cell data to a basal cel-

lular representation to provide independent stochastic variability from perturbation and covariates for individual cells. The encoded basal cell representation of CPA, however, is very likely to entangle with perturbation and covariates in a one-stage latent-arithmetic modeling framework. In addition, the translations between residual representation and cellular representation of PerturbNet are based on complex normalizing flows, while those of CPA are linear functions which are unable to uncover the complex heterogeneous effects of the same perturbation on different cells.

More importantly, the PerturbNet framework models perturbation responses with dense perturbation representation, while CPA one-hot-encodes perturbation with cell state covariates. The dense perturbation representation of PerturbNet qualifies predicting unobserved perturbations outside of the training data, while labels utilized in CPA constrain the predictions to only the observed perturbations or their combinations. Thus, the functionality of predicting unseen perturbations greatly extends the application scope of PerturbNet.

3.3 Experiments

We focus on two datasets with chemical perturbations including the sci-Plex (*Srivatsan et al.*, 2020) and LINCS data (*Subramanian et al.*, 2017). The sci-Plex dataset has scRNA-seq measurements to 188 drug treatments, and LINCS is a microarray dataset with cellular measurements of chemical or genetic perturbations approximately at a single-cell resolution. Thus, we regard both the sci-Plex data and the LINCS subset with 20,065 chemical perturbations (LINCS-Drug) as single-cell responses to chemical perturbations. Table 3.1 summarizes the measurement information of the two datasets and we provide details of their data preprocessing steps in Supplementary Materials Section 3.5.1.

Table 3.1: High-Throughput Gene Expression Datasets with Chemical Perturbations.

Dataset	sci-Plex	LINCS-Drug
Source	scRNA-seq	Microarrays
Cell Lines	A549, K562, MCF7	∼100
Number of Measurements	648,857	689,831
Number of Genes	5087	978
Number of Perturbations	188	20,065

3.3.1 ChemicalVAE Gives Meaningful Perturbation Representations

Following *Gómez-Bombarelli et al.* (2018), we trained ChemicalVAE on the ZINC database (*Irwin and Shoichet*, 2005) with around 250,000 drug treatments. From the trained ChemicalVAE, we obtained the perturbation representations of the drug treatments in the sci-Plex and LINCS-Drug datasets. As the sci-Plex dataset possesses integer count scRNA-seq samples and LINCS-Drug has normally distributed microarray samples, we obtained their cellular representations by training scVI on the sci-Plex data and VAE on the LINCS data. Figure 3.3a shows the UMAP plots of the perturbation representations and the cellular representations of two drug treatments of S1628 and S1007 in the sci-Plex data. The two treatments have distinctive perturbation representations and their cell states are also different.

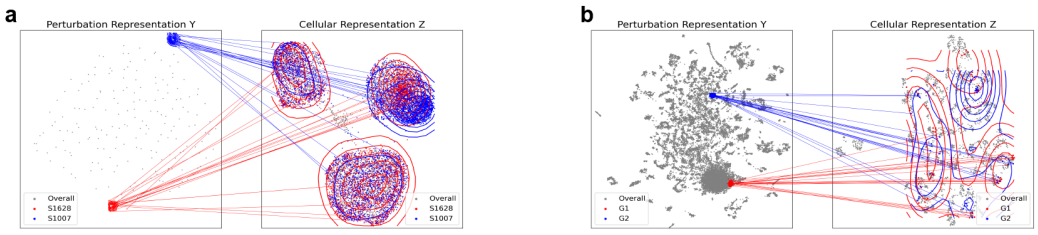

Figure 3.3: **a** UMAP plots of perturbation representations and cellular representations of S1628 and S1007 in the sci-Plex data. **b** UMAP plots of perturbation representations and cellular representations of two drugs in the LINCS-Drug data.

Figure 3.3b shows the UMAP plots of representations of two drug treatments of the LINCS-Drug data. The drugs of G1 and G2 have very different latent values

within the overall perturbation representation space, which map to two unique cell state distributions. Therefore, the perturbation representations from ChemicalVAE can reflect the particular perturbation effects on cell states.

3.3.2 KNN Models Have Better Generation than Random Models

We implemented the baseline models on the sci-Plex and LINCS-Drug data to predict single-cell responses to chemical perturbations. Both baseline KNN and random models predict single-cell perturbation responses by sampling from responses of other perturbations and have relatively consistent performance across perturbations, while PerturbNet tends to have different performances between perturbations utilized to train the model and unobserved perturbations. Therefore, we partitioned the set of perturbations to observed perturbations for model training and unseen perturbations. For the KNN model, we trained a KNN graph on the representations of the observed perturbations to select five nearest neighbors for each of the observed and unseen perturbations. To predict responses of each target perturbation in either the observed or unseen set using the random model, we randomly sampled cells with observed perturbations other than the target perturbation.

We employed the R squared and Fréchet inception distance (FID) metrics to evaluate prediction performance of single-cell responses using the baseline models. A higher R squared and lower FID values better reflect an alignment between the predicted samples and real samples. The details of R squared and FID metrics are shown in Supplementary Materials Section 3.5.3. We compared the prediction performances between the KNN model and the random model for both unseen and observed perturbations, with the one-sided Wilcoxon test. Figure 3.4 shows the performance of baseline models for both unseen and observed treatments of the sci-Plex and LINCS-Drug data. For the sci-Plex data, the KNN model has significantly higher R squared values than the random model, while the FID does not give a significant difference

between the two baseline models. For the LINCS-Drug data, KNN outperforms the random model in both R squared and FID for either the 2000 unseen or 18,065 observed perturbations.

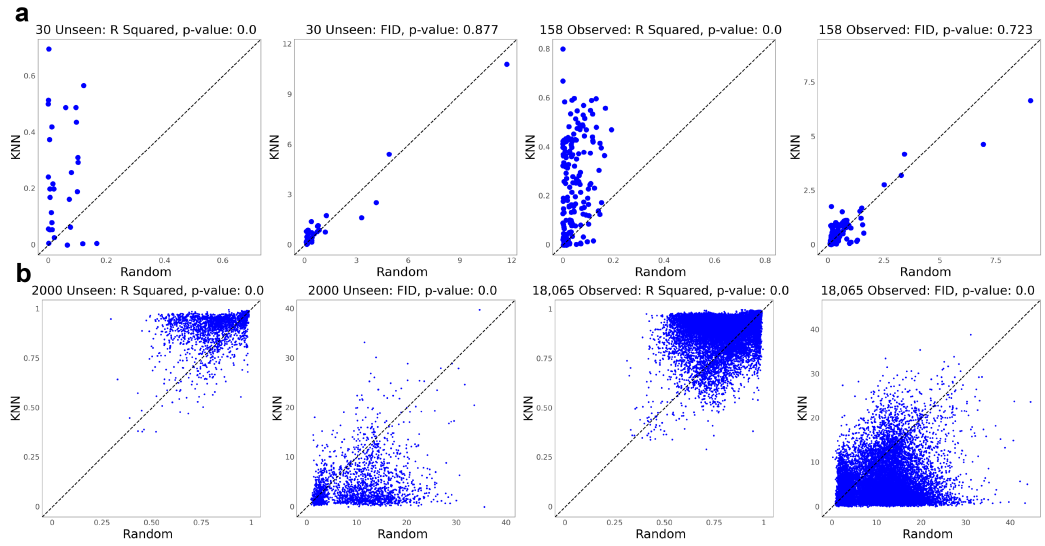

Figure 3.4: R squared and FID of KNN model over random model for unseen and observed drug treatments of the sci-Plex (**a**) and LINCS-Drug (**b**) data.

3.3.3 PerturbNet Predicts Single-Cell Perturbation Responses to Drug Treatments

We employed PerturbNet to predict single-cell responses to drug treatments. We trained scVI on the sci-Plex count data with 158 observed drug treatments, and VAE on the LINCS-Drug data with 18,065 observed drug treatments. We utilized pairs of perturbation and cellular representations of the single-cell subset with observed perturbations to establish the cINN translations of the PerturbNet trained with translations from perturbation to cell state and cellular response.

After constructing the PerturbNet framework on the two datasets, we predicted single-cell responses to each of the unseen and observed perturbations. We evaluated the performances of PerturbNet for the two datasets and compared them to those of the baseline random model (Figure 3.5). As can be seen, PerturbNet has overall

better predictions than the baseline random model. Although PerturbNet does not beat the random model in FID for the unseen perturbations of the sci-Plex data, it has significantly lower FID than the random model for the observed perturbations. For the LINCS-Drug data, PerturbNet outperforms the random model with significantly better R squared and FID. The LINCS-Drug dataset has many more perturbations than the sci-Plex data for training the cINN translations of the PerturbNet, and possibly achieves better out-of-sample predictions for unseen perturbations.

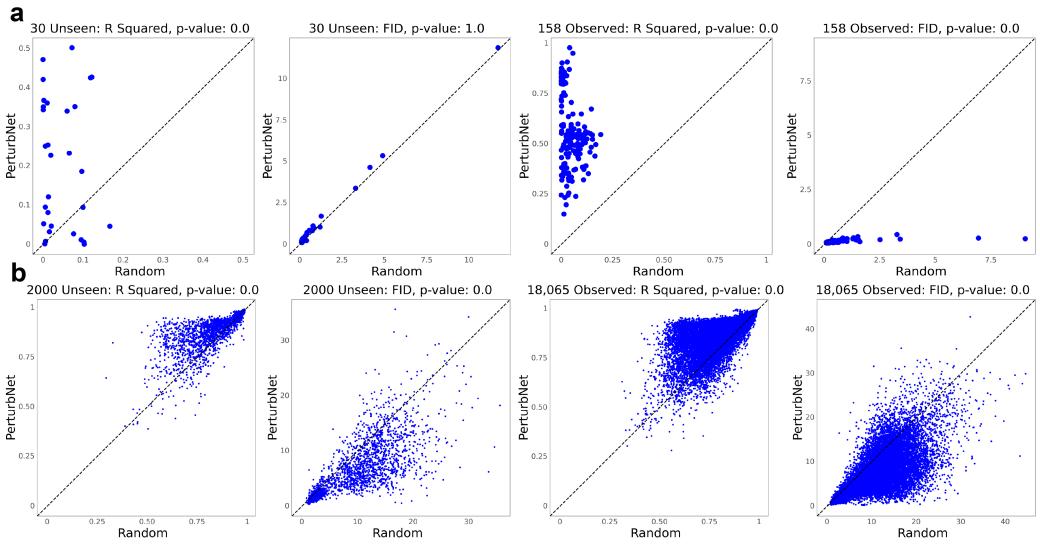

Figure 3.5: R squared and FID of PerturbNet over baseline random model for unseen and observed drug treatments of the sci-Plex (**a**) and LINCS-Drug (**b**) data.

We also compared the performances of PerturbNet and KNN in Supplementary Figure 3.13. The PerturbNet model has significantly better metric values than the KNN model for the observed perturbations of the sci-Plex data, while it does not exceed the KNN model for unseen perturbations of the sci-Plex, LINCS-Drug, or observed perturbations of the LINCS-Drug. The limited 30 unseen perturbations of the sci-Plex data might be insufficient to significantly distinguish the performances of KNN and PerturbNet. In addition, the LINCS-Drug dataset has a smaller variability with high prediction performances from the random model, and thus might enable the KNN model as a difficult baseline model to be overcome by PerturbNet.

3.3.4 Covariate Adjustment Gives Better Predictions for PerturbNet

As the sci-Plex dataset has two cell state covariates of cell type and dose, we adjusted these covariates in modeling cINN translations of PerturbNet. We encoded cell type and dose to the one-hot encodings, and concatenated them to the perturbation representation $\boldsymbol{Y}$ as a joint condition representation. Then we trained cINN with the joint representation of perturbation and covariates as conditions for translations between residual representation and cellular representation. We then predicted single-cell responses to a perturbation with the specific values of covariates.

We evaluated the prediction performance of PerturbNet adjusted for covariates on the unseen and observed perturbations with cell covariates' values, and compared its performance with that of the previous PerturbNet trained without the cell state covariates. As can be seen in Figure 3.6, the PerturbNet adjusted for cell state covariates significantly outperforms the PerturbNet without covariate adjustment for observed perturbations in both R squared and FID. The PerturbNet adjusted for covariates improves R squared for the unseen perturbations. The cell state covariates are correlated with perturbation assignment and also influence cellular responses, making them possess confounding effects in modeling perturbation responses. Therefore, adjusting for covariates in cINN modeling of the PerturbNet helps debias their confounding effects, and more accurately quantify perturbation effects.

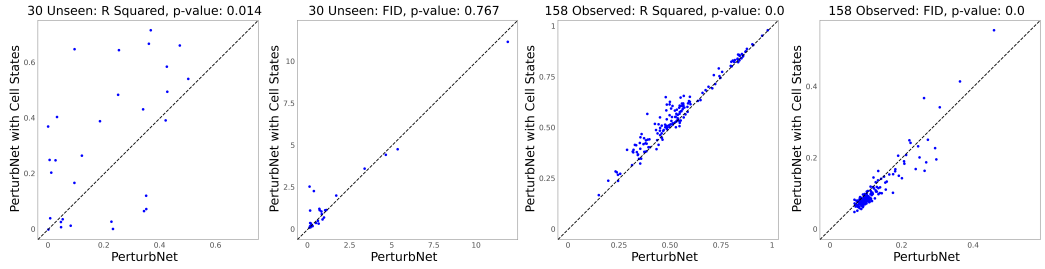

Figure 3.6: R squared and FID of PerturbNet adjusted for cell state covariates over the unadjusted PerturbNet for 30 unseen and 158 observed drug treatments of the sci-Plex data.

We compared the performance of PerturbNet adjusted for covariates with the

baseline models. As the PerturbNet adjusted for covariates takes additional covariate information other than perturbation, we performed a stratified prediction in each cell type by dose stratum to also adjust covariate information for the baseline models. Each perturbation has 12 strata with three cell types and four doses. We proceeded with the sampling procedures of the baseline KNN and random models within each cell by dose stratum, and made PerturbNet predictions with the corresponding covariates' values in the stratum. Figure 3.7 shows that PerturbNet consistently outperforms the random model for observed perturbations, while KNN is unable to beat the random model for either unseen or observed perturbations. As the stratified evaluations constrain cellular variability and sample size, which possibly narrows down the prediction performances of the KNN and random models, we also compared PerturbNet adjusted for covariates and KNN in stratified predictions (Supplementary Figure 3.14). As with their unstratified comparisons in Supplemenetary Figure 3.13, the PerturbNet has a better performance for observed perturbations but does not defeat KNN for unseen perturbations.

3.3.5 Adjusting Confounders of Perturbations in PerturbNet

In Section 3.3.4, we adjusted the covariates in PerturbNet to improve the prediction performance for both unseen and observed perturbations. We illustrated potential confounding effects of the cell state covariates in modeling perturbations on single-cell responses, and showed that PerturbNet adjusted for covariates achieved better predictions. In this section, we study impacts of confounding effects of cell state covariates in learning representations and predicting perturbation responses in the cINN modeling of the PerturbNet. We considered the sci-Plex subset example in Section 2.3.6 and *Yu and Welch* (2021) with 18 drug treatments and three cell types, where we implemented latent space vector arithmetic algorithm to predict single-cell data of cell type/drug treatment combinations, including three unseen combinations

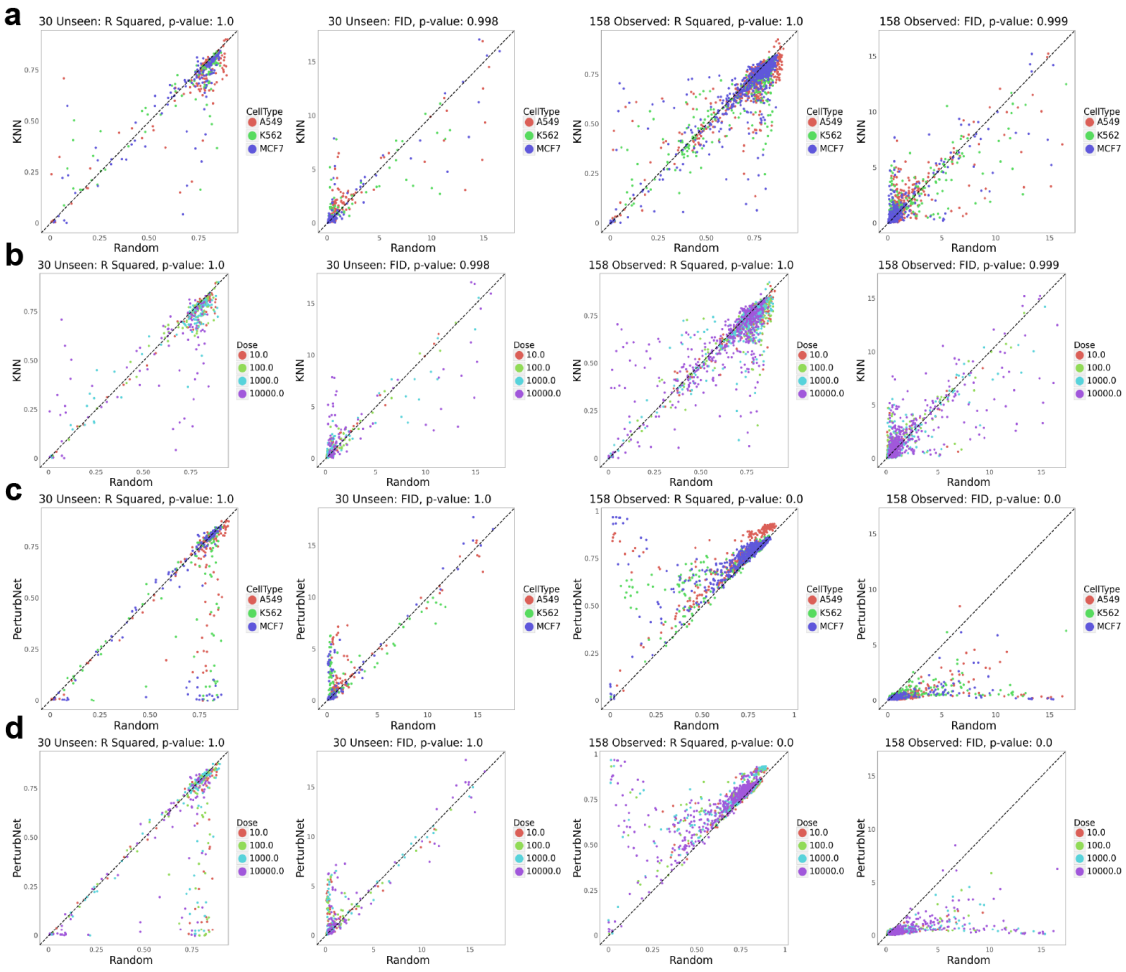

Figure 3.7: R squared and FID of KNN (**a, b**) and PerturbNet adjusted for covariates (**c, d**) over the random model for 30 unseen and 158 observed drug treatments in each stratum of cell type by dose of the sci-Plex data, visualized by cell type (**a, c**) and dose (**b, d**).

(MCF7-S1259, MCF7-S1262, MCF7-S7207).

We trained VAE on the sci-Plex subset without the three unseen combinations. We then one-hot-encoded cell type or perturbation, and constructed the cINN translations between one-hot-encoded cell type or one-hot-encoded perturbation and cellular representation. We then adapted these cINN transitions to perform a similar procedure to latent space vector arithmetic to predict single-cell data of each cell type/treatment combination. For the cINN trained between one-hot-encoded cell type and cellular representation, we utilized a group of control cells of another type and the same treatment to obtain their residual representations $\boldsymbol{V}$'s via the cINN inverse translation, and predict their counterfactual cellular representation and single-cell data through the cINN translation with the cell type of the target combination. We named this prediction cell type translation. For the cINN trained between one-hot-encoded perturbation and cellular representation, we utilized a group of control cells of other treatments and the same cell type to obtain their residual representations $\boldsymbol{V}$'s and predict their counterfactual cellular representation and single-cell data using the perturbation of the target combination through the cINN inverse and forward translations, respectively. We named this prediction as treatment translation. The two predictions assumed that the residual representations $\boldsymbol{V}$'s in the cell type cINN translations preserved perturbation information to predicted counterfactual cellular representation, and vice versa.

We also trained a PerturbNet model on the sci-Plex subset without the three unseen combinations with the cINN translations between concatenated one-hot-encoded representations of perturbation, cell type and dose. We then predicted each combination by generating cells from PerturbNet. In addition to cell type translation, treatment translation and PerturbNet, we also predicted each combination using the latent space vector arithmetic algorithm.

We evaluated the predicted data using the R squared metric and show the R

squared values of the four methods across the 54 combinations (Figure 3.8a). PerturbNet has the highest R squared among the four methods for almost all the combinations. Figure 3.8b also shows that PerturbNet predictions has significantly higher R squared than latent space vector arithmetic, cell type translation and treatment translation. It also has higher R squared for the three unseen combinations.

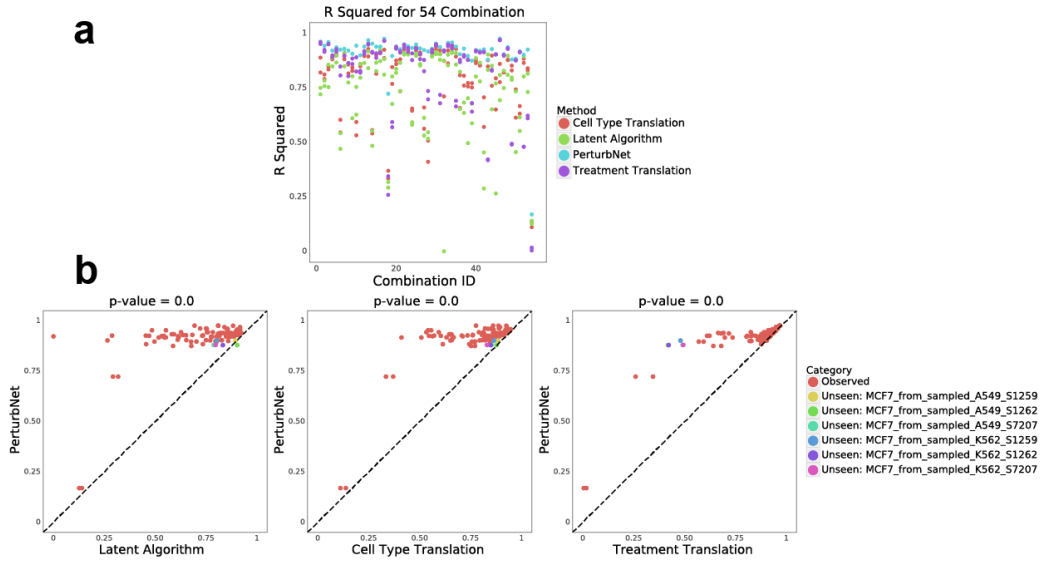

Figure 3.8: **a** R squared of predictions of cell type/treatment combinations using latent space vector arithmetic (latent algorithm), cell type translation, treatment translation and PerturbNet. **b** R squared of predicted cell type/treatment combinations between PerturbNet and each of latent algorithm, cell type translation and treatment translation. The p-values are from the one-sided Wilcoxon test.

Figure 3.9 shows the UMAP plots of predicted MCF7-S1259 unseen combination from the four methods. The latent space vector arithmetic and PerturbNet generally recover the real cells of the combination, while cell type translation generate cells of perturbations other than S1259 and treatment translation gives cells of cell types other than MCF7. This means that the residual representations $\boldsymbol{V}$'s in the cINN trained with only cell type, and no treatment, does not preserve the treatment information, nor did the residual representations in the cINN trained with only drug treatment, no cell type, fail to preserve the cell type. The PerturbNet fully adjusted for perturbation

and covariates has the best predicted cells of MCF7-S1259, which overlap well with the real cells.

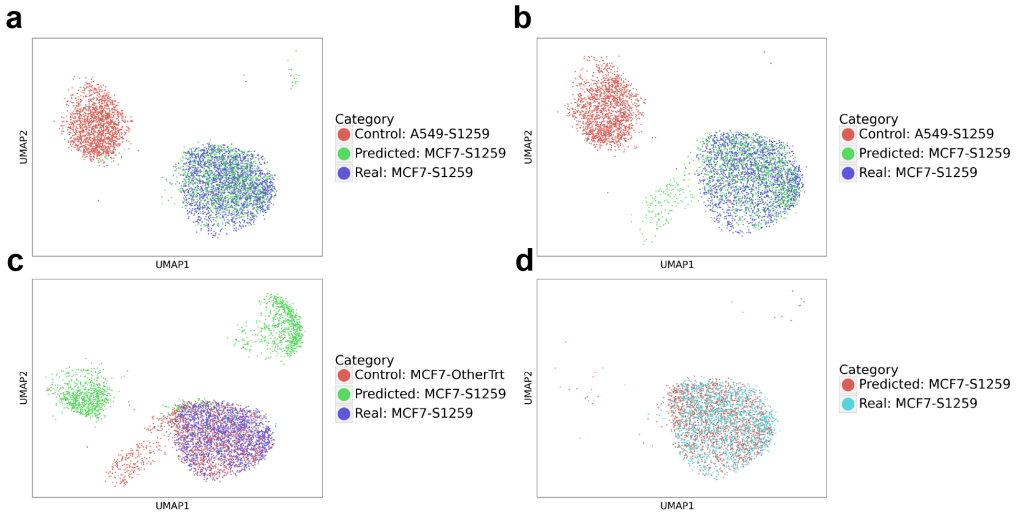

Figure 3.9: UMAP plots of predicted MCF7-S1259 using latent space vector algorithm (**a**), cell type translation (**b**) treatment translation (**c**) and PerturbNet (**d**).

Therefore, both cell type translation and treatment translation have a more parsimonious modeling procedure and do not need to adjust for other covariates in their cINN models, but their residual representations might not preserve meaningful information for individual cells. A possible reason for their entangled residual representations is that a cINN model assumes independence between condition and residual representation, while cell type is actually correlated with perturbation assignment in the sci-Plex subset. Thus, the residual representation does not carry meaningful perturbation or cell type information in the two translations for an individual cell. On the other hand, modeling cell type/treatment combinations using PerturbNet requires specifying covariates in the cINN modeling, and these covariates help make more unbiased predictions.

3.3.6 Fine-Tuned ChemicalVAE Improves the Performance of PerturbNet

We performed ChemicalVAE fine-tuning to improve the performance of PerturbNet. To construct a cell-state Laplacian matrix $\boldsymbol{L}$, we computed the Wasserstein-2 (W2) distance between cellular latent values of each pair of drug treatments. As the number of treatment pairs is extremely large in LINCS-Drug, we first fitted a KNN algorithm on the 20,065 perturbation representations and selected the 30 nearest neighbors for each drug treatment to compute their pairwise cellular latent distances. As the resulting pairwise cell latent distance matrix for all the 20,065 treatments was not symmetric, we took the average of the matrix and its transposed matrix. We then calculated the exponential of their opposite values and row-normalized the matrix to obtain the adjacency matrix with each entry as a transition probability. We then obtained the Laplacian matrix from the adjacency matrix.

We utilized the Laplacian sub-matrix for the observed drug treatments of LINCS-Drug to fine-tune ChemicalVAE. We considered values of λ in (0.1, 1, 5, 10, 100, 1000, 10,000) to implement the ChemicalVAE fine-tuning algorithm. After we fine-tuned the ChemicalVAE, we evaluated the KNN model on the perturbation representations from these fine-tuned ChemicalVAE. We also constructed the cINN model of the PerturbNet between the perturbation representations of the fined-tuned ChemicalVAE and cellular representations using cells with the observed perturbations. We evaluated the prediction performance of the fine-tuned KNN and PerturbNet models on the 2000 unseen perturbations of the LINCS-Drug data (Figure 3.10). Both R squared and FID of PerturbNet have small to medium fluctuations across increasing λ values, while those of KNN do not obviously change with varying λ values. Several λ values give slight increases of median R squared or decreases of median FID for PerturbNet over the non-fine-tuned one, such as $\lambda = 0.1, 1, 5, 10, 100$.

We compared the fine-tuned KNN and PerturbNet with $\lambda = 1$ to their non-

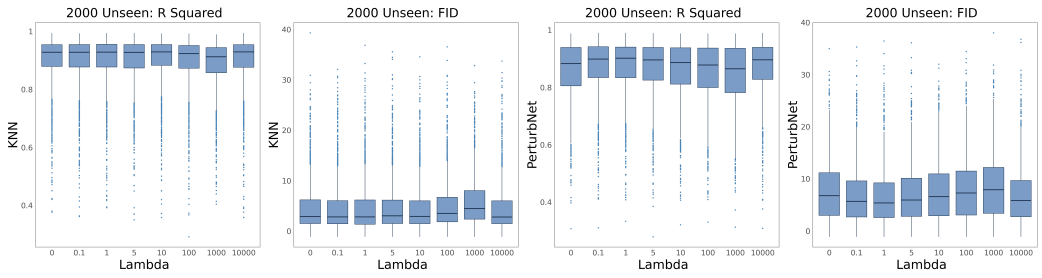

Figure 3.10: R squared and FID of KNN and PerturbNet with fine-tuned ChemicalVAE across different λ values for the 2000 unseen drug treatments of the LINCS-Drug data.

fine-tuned counterparts for the unseen perturbations (Figure 3.11). The fine-tuned PerturbNet has significant improvements in both R squared and FID, while fine-tuning ChemicalVAE does not significantly enhance KNN. A possible explanation is that the cINN of PerturbNet further enforces the prediction capacity from fine-tuned perturbation representation to cell state.

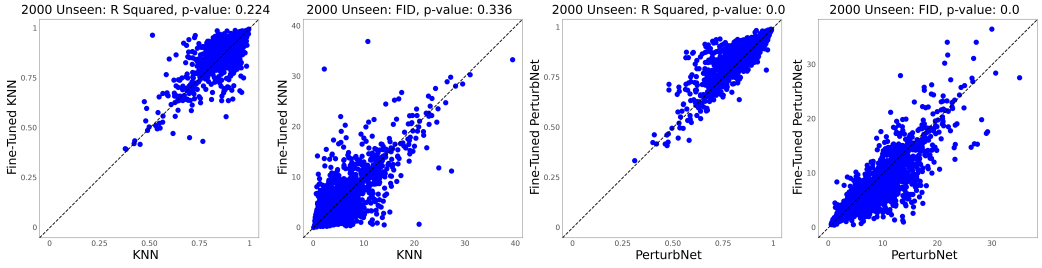

Figure 3.11: R squared and FID metrics of KNN and PerturbNet with fine-tuned ChemicalVAE of $\lambda = 1$ over non-fine-tuned PerturbNet for 2000 unseen drug treatments of the LINCS-Drug data.

3.3.7 PerturbNet Recovers the Perturbation and Cell Latent Spaces

We used PerturbNet to generate cellular representation from the perturbation representation of a drug treatment. We considered reconstructing cellular representations using $\boldsymbol{V}$'s inferred from real cells, or sampling cellular representations using $\boldsymbol{V}$'s from the prior distribution. We used the perturbations of the sci-Plex and LINCS-Drug data in Figure 3.3 to reconstruct and sample cellular representations and show

the UMAP plots of these cellular representations in Figure 3.12. As can be seen, the reconstructed representations recover the observed cellular representations of both the sci-Plex and LINCS-Drug data in Figure 3.3, while the sampled representations do not strictly reflect the observed latent distributions. The residual representations possess condition-invariant information to reconstruct individual cells, and those sampled from the prior distribution generate general cell states of a perturbation.

Therefore, the predicted single-cell responses to a perturbation via PerturbNet by sampling $\boldsymbol{V} \sim \mathcal{N}(\mathbf{0}, \mathbf{1})$ does not specifically recover the observed individual cells to the perturbation, as they possess specific residual information. PerturbNet predicts single-cell perturbation responses with a general residual distribution, which maps to the overall training data in the cINN modeling of PerturbNet. If we have prior information about the individual residual representation, we can make more precise predictions on individual perturbation responses for observed cells, especially for unseen perturbations. However, cells are usually measured and destroyed in single-cell experiments before this residual information can be inferred.

3.4 Discussion

In this chapter, we propose a deep generative model, PerturbNet, to predict single-cell responses to chemical perturbations. We encode cell samples and drug SMILES strings to dense latent representations using single-cell VAE and ChemicalVAE. We then connect two representations through cINN. PerturbNet gives a stable training process, which has two stages of ChemicalVAE and single-cell VAE trained separately and integrated through cINN. The PerturbNet framework can make predictions for both unseen and observed drug treatments. We perform experiments to show that PerturbNet has excellent prediction performance for single-cell responses to both unseen and observed drug treatments. In addition, our ChemicalVAE fine-tuning algorithm also improves the prediction performance using fine-tuned perturbation

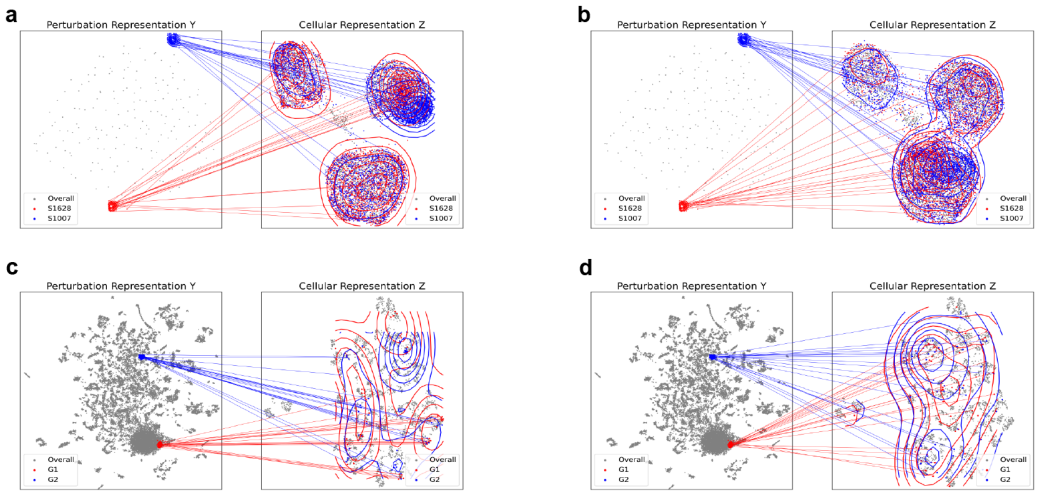

Figure 3.12: **a, b** UMAP plots of perturbation representations and reconstructed cellular representations (**a**) as well as sampled cellular representations (**b**) for S1628 and S1007 in the sci-Plex data. **c, d** UMAP plots of perturbation representations and reconstructed cellular representations (**c**) as well as sampled cellular representations (**d**) for two drugs in the LINCS-Drug data.

representation.

Our PerturbNet is closely related to the CPA framework, which also predicts single-cell perturbation responses to chemical perturbations (*Lotfollahi et al.*, 2021). The CPA framework also utilizes a single-cell VAE and models perturbation effects on latent values of cells. However, the encoder of CPA only gives a basal cell state, while perturbation is further incorporated to decode to single-cell data, making it a very unusual autoencoder framework lying between VAEs and conditional VAEs. In addition, the latent linear model in CPA takes drug treatments as categories, and thus cannot make predictions for unseen drug treatments. Therefore, we do not directly compare CPA and PerturbNet due to their different functionalities.

A limitation of our experiments is with the limited number of chemical perturbations. We find that LINCS-Drug has many more chemical perturbations and a slightly better prediction performance for unseen perturbations than the sci-Plex data. Having more chemical perturbations enables more powerful cINN connections

of the PerturbNet to infer the cellular representation for each unseen perturbation representation. Therefore, future research can implement PerturbNet on new high-throughput chemical screen datasets with many more drug treatments.

Another future improvement of PerturbNet is to train ChemicalVAE with larger chemical databases, such as PubChem (*Kim et al.*, 2016) with millions of drug treatments. In addition, having better preprocessing steps of chemical SMILES strings might further improve the training of ChemicalVAE to obtain better perturbation representations. Other advanced deep generative models for chemical perturbations can also be employed to replace the ChemicalVAE model of the PerturbNet. For example, *Jin et al.* (2020a) proposed a hierarchical VAE structure for graph generation of molecules and multi-layer representations of drug treatments. These multi-resolution perturbation representations can potentially give better single-cell predictions to unseen drug treatments.

3.5 Supplementary Materials

3.5.1 Datasets

We obtained the ZINC database with 250,000 compounds (*Irwin and Shoichet*, 2005) from the ChemicalVAE model (`https://github.com/aspuru-guzik-group/chemical_vae/tree/main/models/zinc`). We transformed the compounds to canonical SMILES following the ChemicalVAE tutorial (`https://github.com/aspuru-guzik-group/chemical_vae/blob/main/examples/intro_to_chemvae.ipynb`) via the RDKit package (*Landrum*, 2016). We also utilized the chemical elements' library from this tutorial to define the one-hot matrices of drug treatments, where we constrained the maximum length of canonical SMILES strings to be 120.

We processed the whole sci-Plex data (*Srivatsan et al.*, 2020) using SCANPY (*Wolf et al.*, 2018) with a lot of 648,857 cells and 5087 genes. There were 634,110

cells perturbed by 188 drug treatments in total, with 14,627 cells treated by unknown drug treatments and 120 unperturbed cells. We randomly selected 30 drug treatments as unseen perturbations and the other 158 drug treatments as observed perturbations.

We obtained the LINCS dataset (*Subramanian et al.*, 2017) from GEO accession ID GSE92742. The LINCS data had been processed with 1,319,138 cells and 978 landmark genes, containing the LINCS-Drug subset with 689,831 cells treated by 20,329 drug treatments denoted with their SMILES, 20,065 drug treatments of which had lengths smaller than 120. We randomly selected 2000 drug treatments as unseen perturbations and the other 18,065 drug treatments as observed perturbations. We transformed the SMILES strings of drug treatments of the sci-Plex and LINCS-Drug data to their one-hot matrices according to the chemical elements' library.

3.5.2 Neural Network Architectures

We followed the ChemicalVAE model utilized in *Gómez-Bombarelli et al.* (2018) and adapted it to PyTorch implementations. The ChemicalVAE model takes each input of size of 120 by 35, and has three one-dimensional convolution layers with the triplet of number of input channels, number of output channels and kernel size being (120, 9, 9), (9, 9, 9) and (9, 10, 11), respectively. There are a Tanh activation function and a batch normalization layer following each convolution layer. After these transformations, the input is then flattened to a fully-connected (FC) hidden layer with 196 neurons, and is subsequently activated by a Tanh function, followed by a dropout regularization with a dropout probability of 0.08 and a batch normalization layer. Then two hidden layers both with 196 neurons generate means and standard deviations of the latent variable. The decoder of the ChemicalVAE model has a FC hidden layer with 196 neurons, followed by a Tanh activation, a dropout regularization with a dropout probability of 0.08 and a batch normalization layer. Then the elements of the input are repeated 120 times to be put in a GRU layer with three hidden layers

of 488 hidden neurons, followed by a Tanh activation. The input is then transformed to a two-dimensional tensor to be put in a FC layer with 35 neurons and a softmax activation function. Then each input is reshaped to be the output tensor of 120 by 35. We implemented the ChemicalVAE training on the ZINC data with different learning rates. We finally had an optimal training with a batch size of 128 and a learning rate of 10^{-4} for 100 epochs.

We trained the cINN translations following *Rombach et al.* (2020), where a cINN consists of 20 invertible neural network blocks and an embedding module. Each block has an alternating affine coupling layer, an activation normalization (actnorm) layer and a fixed permutation layer. The embedding module consists FC hidden layers and Leaky Rectified Linear Unit (ReLU) activation functions to embed the conditioning variable into a 10-dimensional variable. We fixed the batch size of 128, the learning rate of 4.5×10^{-6} and varied different numbers of epochs for training cINN. We found the cINN training generally stabilized after 50 epochs across different datasets.

For cellular VAE models, we utilized scVI version 0.7.1 with 10-dimensional latent space. We trained scVI on both the whole sci-Plex data and its subset with observed perturbations for 700 epochs with scVI's default setting of learning rate of 10^{-3} and batch size of 128. We trained a regular VAE model for the LINCS-Drug data with a similar multilayer perceptron (MLP) model in Chapter II, which is based on TensorFlow version 1.14.0. The VAE has two fully-connected (FC) hidden layers with 512 and 256 neurons, followed by separate hidden layers for means and standard deviations of the latent variable. The decoder has two hidden layers with 256 and 512 neurons, and its output layer has the same number of neurons as the number of genes. Each hidden layer is followed by a batch normalization, a ReLU or Leaky ReLU activation, a dropout regularization with a dropout probability of 0.2. We tried different learning rates and found that a learning rate of 10^{-4} and a batch size of 128 work well with the LINCS data. We trained VAE on the whole LINCS data for 200

epochs, and trained VAE on the LINCS-Drug with observed perturbations for 150 epochs.

3.5.3 Prediction Metrics

3.5.3.1 R Squared

We follow the R Squared metric utilized in several frameworks to predict single-cell responses to perturbations (*Lotfollahi et al.*, 2019, 2020, 2021). We first obtain the normalized data of predicted and real single-cell responses to a perturbation for the sci-Plex data. We conduct similar processing steps to SCANPY (*Wolf et al.*, 2018). We first normalize the total number of counts of each cell to be 10^4, take log-transformation, and scale the values. We directly use LINCS samples as they have already been normalized. We compute the mean gene expression values of normalized data of both predicted and real cells to a drug treatment. We then fit a simple linear regression model on the real mean gene expression values over the predicted mean gene expression values. The R squared of the fitted linear regression is then reported to quantify the accuracy of predicted cells.

3.5.3.2 FID Score

We define an FID score metric similar to the FID metric utilized in image data (*Heusel et al.*, 2017). We train a single-cell VAE model on the whole single-cell dataset using either scVI or VAE depending on the data type. We obtain the cell latent values of the predicted and real cells to a perturbation. We then apply the Fréchet distance to the latent values of predicted and real cells with the Gaussian assumption

$$\text{FID} = \|\boldsymbol{\mu}_{\text{Real}} - \boldsymbol{\mu}_{\text{Predicted}}\|_2^2 + \text{trace}\{\boldsymbol{\Sigma}_{\text{Real}} + \boldsymbol{\Sigma}_{\text{Predicted}} - 2(\boldsymbol{\Sigma}_{\text{Real}}\boldsymbol{\Sigma}_{\text{Predicted}})^{1/2}\},$$

where $\boldsymbol{\mu}_{\text{Real}}$, $\boldsymbol{\mu}_{\text{Predicted}}$ are means of predicted and real latent values, and $\boldsymbol{\Sigma}_{\text{Real}}$, $\boldsymbol{\Sigma}_{\text{Predicted}}$ are covariance matrices of predicted and real latent values.

3.5.4 Supplementary Figures

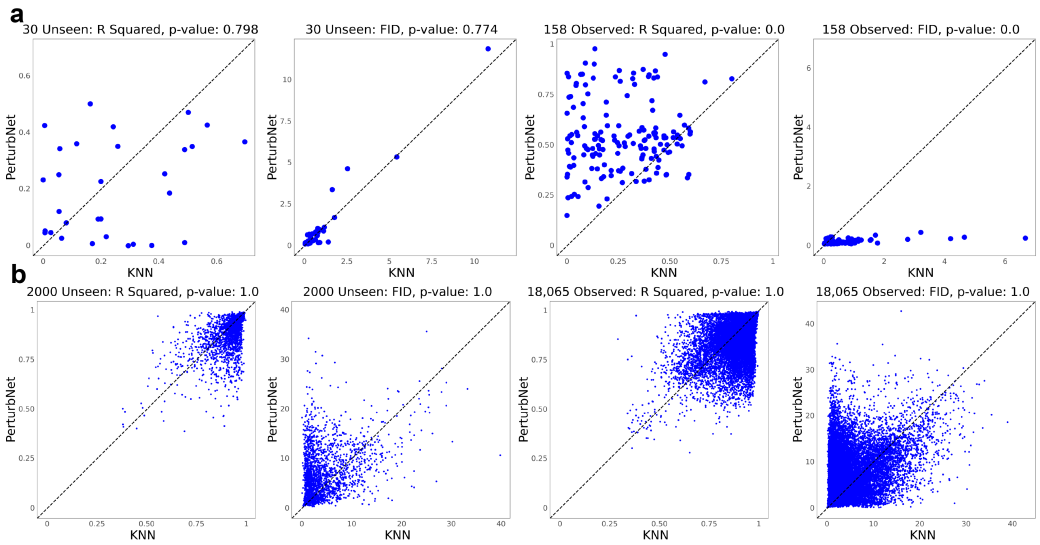

Figure 3.13: R squared and FID of PerturbNet over baseline KNN model for unseen and observed drug treatments of the sci-Plex (**a**) and LINCS-Drug (**b**) data.

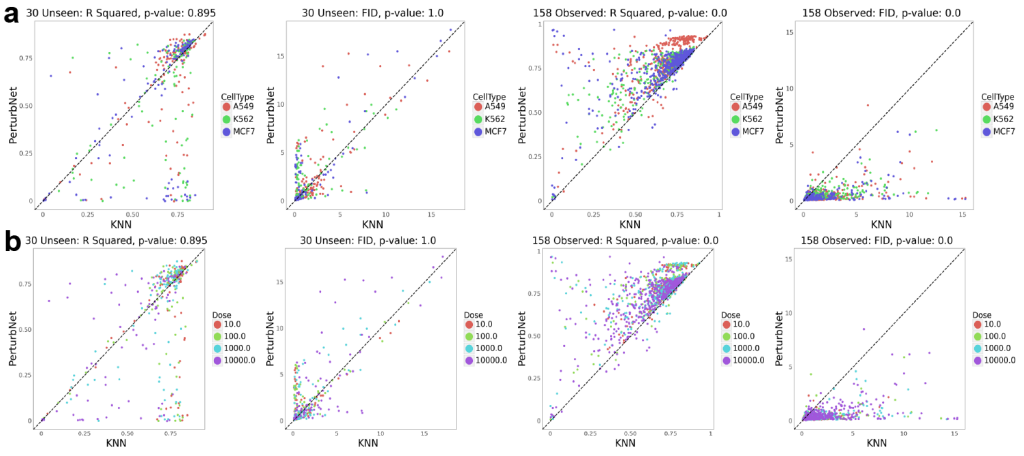

Figure 3.14: R squared and FID of PerturbNet adjusted for covariates over KNN for 30 unseen and 158 observed drug treatments in each stratum of cell type by dose of the sci-Plex data, visualized by cell type and dose.

CHAPTER IV

Predicting Single-Cell Responses to Genetic Perturbations

4.1 Introduction

Unlike chemical perturbations, whose direct gene targets are generally unknown, genetic perturbations are designed to directly knock out or activate one or several target genes. The genes' activation or knockout will not only influence their own expression, but also impact other genes through a complex network of downstream gene regulatory interactions. The Clustered Regularly Interspaced Short Palindromic Repeats (CRISPR) technology allows an easy design of precise genetic mutants through genome editing (*Doudna and Charpentier*, 2014). More recently, CRISPR has been combined with transcriptional activators (CRISPRa) or repressors (CRISPRi) tethered to dCas9 to enable activation or inhibition of target genes. The Perturb-seq technology combines CRISPR gene editing and single-cell RNA-sequencing (scRNA-seq) to measure single-cell responses to pooled CRISPR screens (*Dixit et al.*, 2016). Perturb-seq measures cellular responses at single-cell resolution, which reveals how cell states are impacted by genetic perturbations, and has been utilized for many biomedical applications (*Wang et al.*, 2015; *Adamson et al.*, 2016; *Datlinger et al.*, 2017; *Ursu et al.*, 2020; *Jin et al.*, 2020b). However, as Perturb-seq experiments can

only measure limited numbers of perturbation loci, while the human genome contains about 3.2 billion nucleotides of DNA (*Brown*, 2018), it is not feasible to measure single-cell responses for each potential genetic perturbation.

Various methods have been proposed to detect perturbations that have interpretable effects on cellular responses to genetic perturbations. *Norman et al.* (2019) used Perturb-seq data to identify genetic interaction from paired gene knockouts. *Burkhardt et al.* (2021) identified perturbation effects over the cellular manifold, using graph signal processing tools. *Yeo et al.* (2021) proposed a generative model using a diffusion process over a potential energy landscape to learn the underlying differentiation landscape from time-series scRNA-seq data and to predict cellular trajectories under perturbations. Linear models were also used to estimate the impact of perturbations on high-dimensional scRNA-seq data (*Dixit et al.*, 2016) or infer gene regulatory networks (GRNs) from perturbations (*Kamimoto et al.*, 2020). However, these linear models had limited predictive power on the non-linear gene expression profiles of perturbations and other cell state variables. The work most closely related to ours is the compositional perturbation autoencoder (CPA) framework (*Lotfollahi et al.*, 2021), which generates single-cell data under perturbations based on latent space linear models. However, as with drug treatment perturbations, CPA assumes that the effects of genetic perturbations can be estimated independently of a basal cell state. In addition, although CPA can make predictions on new combinations of observed target genes, it cannot predict single-cell responses to genetic perturbations with unseen target genes.

Therefore, in this chapter, we develop a deep generative model that predicts single-cell responses to combinatorial genetic perturbations, including both observed and unseen target genes. To do this, we extend the PerturbNet model using a neural network that learns representations of genetic perturbations. Our network can embed the two main classes of genetic perturbations: genome edits made with CRISPR/Cas9

(*Dixit et al.*, 2016) and gene knockdowns or activations using CRISPRi or CRISPRa (*Adamson et al.*, 2016). This genetic perturbation autoencoder allows us to translate from the perturbation latent space to the cell state space, then to generate realistic single-cell gene expression profiles from the cell state space.

4.2 Methods

We extend the PerturbNet framework to genetic perturbations by constructing an autoencoder for the target genes in combinatorial genetic perturbations. There are two main types of genetic perturbations. The genetic perturbations using CRISPR activation (CRISPRa) or CRISPR interference (CRISPRi) do not change the original DNA sequence (*Norman et al.*, 2019). In contrast, CRISPR/Cas9 directly modifies the DNA sequence, leading to changes in the protein-coding sequence or non-coding regulatory sequence. A genetic perturbation can be represented by either the identities of its target genes (*Dixit et al.*, 2016) or the final sequence induced by genome editing. For example, *Ursu et al.* (2020) performed Perturb-seq to assess the impacts of single amino acid changes in the proteins TP53 and KRAS. In this experiment, a TP53 variant of 'Q5R' means that the fifth amino acid of the TP53 protein sequence was changed from 'Q' to 'R'. We construct two types of genetic perturbation autoencoders–one for each type of perturbation representation (target gene identities or edited sequences).

4.2.1 Genetic Perturbations and GenotypeVAE

We first consider genetic perturbations represented by target genes. Most of the existing methods one-hot-encode the target genes across a set of genes (*Dixit et al.*, 2016) or all genes on a coding sequence (*Ma et al.*, 2018). However, this strategy cannot generalize to perturbations with an unseen target gene.

To encode genetic perturbations, we propose a more parsimonious framework and

refer to it as GenotypeVAE (Figure 4.1). Our key insight is that the numerous functional annotations of each gene (organized into a hierarchy in the gene ontology) provide features for learning a low-dimensional representation of individual genes and groups of genes. Using gene ontology (GO) terms, we can represent each target gene g as a one-hot vector $\boldsymbol{B}_g$, where 1's in the vector element correspond to a particular term indicating that the gene has the annotation. Our approach is inspired by *Chicco et al.* (2014). If we have a genetic perturbation with multiple target genes $\{g_1, \ldots, g_k\}$, we use annotation-wise union operations to generate a one-hot annotation vector for the genetic perturbation as follows:

$$\boldsymbol{B}_{g_1,\ldots,g_k} = \cup_{j=1}^{k} \boldsymbol{B}_{g_j}.$$

Then, we can train GenotypeVAE using one-hot representations of many possible genetic perturbations. We use the GO Consortium gene ontology annotation dataset of human genes. This resource annotates 18,832 genes with 15,988 annotation terms (after removing some annotations with insufficient information). We take the 15,988-dimensional annotation vector as the input to the GenotypeVAE encoder consisting of two hidden layers with 512 and 256 neurons, following output layers for means and standard deviations, both with 10 neurons. The GenotypeVAE decoder also has two hidden layers with 256 and 512 neurons, along with an output layer of 15,988 neurons activated by the sigmoid activation function. We also have a batch normalization layer, Leaky ReLU activation and a dropout layer with a dropout probability of 0.2 following each hidden layer of GenotypeVAE.

4.2.2 Protein Perturbations and ESM

We also extend the PerturbNet framework to predict single-cell responses to protein-coding sequence variants. A coding variant can be uniquely represented by the

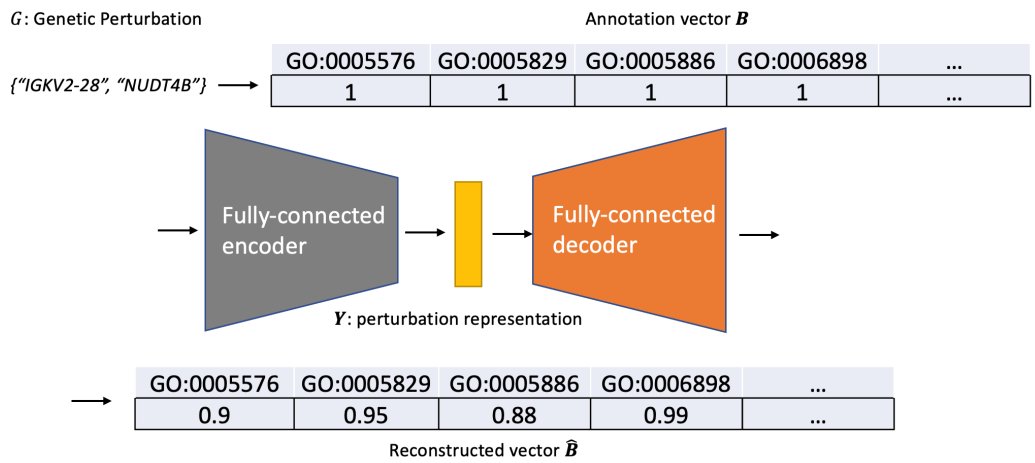

Figure 4.1: Overview of the GenotypeVAE architecture.

protein sequence resulting from the nucleotide alterations induced by CRISPR/Cas9 editing. Similar to chemical perturbations, protein perturbations can also be summarized as sequences of strings. A key difference is that each character of a protein sequence is a naturally occurring character sequence, whereas a chemical structure is actually a three-dimensional structure (even if it is sometimes represented as a string).

We therefore consider a state-of-the-art language model for protein sequences. Rather than designing our own model and training it from scratch, we employ the previously published Evolutionary Scale Modeling (ESM, *Rives et al.*, 2021) architecture shown in Figure 4.2. ESM is a self-supervised transformer model (*Devlin et al.*, 2018) and was previously shown to achieve better representations and prediction performance on protein sequences compared to other language models such as long short-term memory (LSTM) networks. As with other transformer models (*Vaswani et al.*, 2017), the ESM model was pre-trained on large protein sequence datasets (*Rao et al.*, 2021). We adopt a pre-trained ESM model specialized for prediction of single variant effects (*Meier et al.*, 2021), because this application is most similar to our scenario.

However, the representation obtained from ESM is deterministic for a given pro-

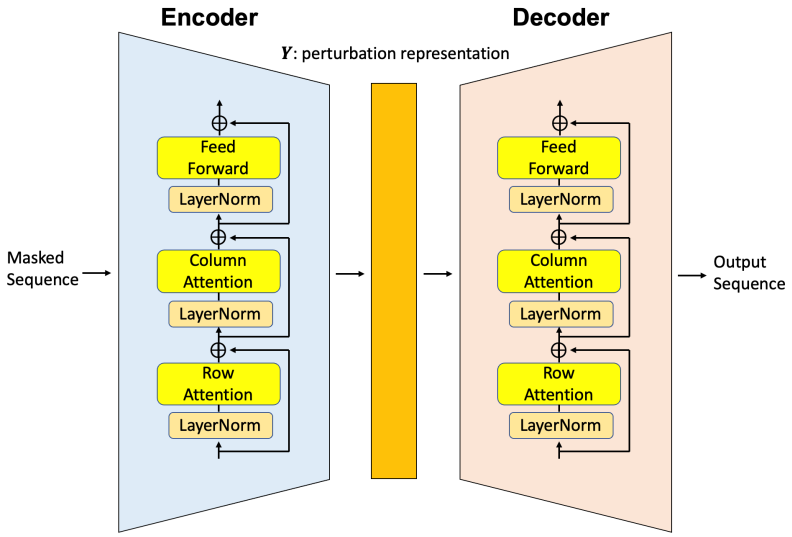

Figure 4.2: Sketch of the ESM architecture.

tein sequence. The fixed protein representations limit the amount of training data available for PerturbNet, especially when there is a small number of protein sequences. We therefore add low-variance noise $\boldsymbol{\epsilon}$ to the ESM representation $\boldsymbol{Y}_{\text{ESM}}$ from ESM. The final perturbation representation is thus computed as

$$\boldsymbol{Y} = \boldsymbol{Y}_{\text{ESM}} + \boldsymbol{\epsilon},$$

where $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\sigma}^2 \boldsymbol{I})$. We choose the variance $\boldsymbol{\sigma}^2$ to be a positive constant small enough that it does not significantly alter the relative distances between proteins in the ESM latent space.

4.3 Experiments

We applied different prediction methods to several datasets with genetic perturbation. We used the genetic CRISPR screen data to explore genetic interaction manifolds (GI, *Norman et al.*, 2019), LINCS data with 4,109 genetic perturbations (*Subramanian et al.*, 2017), genome-scale Perturb-seq data with 9499 perturbations (GSPS, *Replogle et al.*, 2022), as well as the Perturb-seq data with 1338 coding-

sequence variants (Ursu, *Ursu et al.*, 2020). The first three datasets have genetic perturbations of target gene identities, and the Ursu dataset has genetic perturbations of protein-coding sequence variants. The LINCS is a microarray dataset with cellular measurements approximately at a single-cell resolution, and we used its subset treated by genetic perturbations (LINCS-Gene) as single-cell responses. Table 4.1 summarizes the measurement information of the datasets, and we provide details of their data preprocessing steps in Supplementary Materials Section 4.5.1.

Table 4.1: High-Throughput Gene Expression Datasets with Genetic Perturbations.

Dataset	GI	LINCS-Gene	GSPS	Ursu et al.
Source	scRNA-seq	Microarrays	scRNA-seq	scRNA-seq
Cell Lines	K562	∼100	K562	A549
Number of Measurements	109,738	442,684	1,989,373	164,931
Number of Genes	2279	978	2000	1629
Number of Perturbations	230	4109	9499	1338
Perturbation Identity	Gene	Gene	Gene	Sequence

4.3.1 PerturbNet Models Latent Representations of Genetic Perturbations

We utilized the ELBO loss and trained GenotypeVAE with a learning rate of 10^{-4} for 300 epochs. During the training, we considered a probability of 0.5 for an iteration with a batch of genes to be genetic perturbations with single target genes, and another probability of 0.5 for the iteration to be used with the next batch of genes to be genetic perturbations with double target genes. We then evaluated the GenotypeVAE latent spaces for the genetic perturbations of the GI, LINCS-Gene and GSPS datasets. Both GI and GSPS have integer count scRNA-seq samples, while LINCS-Gene has normally distributed microarray samples. We therefore obtained their cellular representations by training scVI (*Lopez et al.*, 2018) on the GI and GSPS data, as well as VAE on the LINCS. Figure 4.3a shows the UMAP plots of perturbation representations

and cellular representations for the two selected genetic perturbations in each of the three datasets. Both pairs of (KLF1/ctrl, SLC4A1/ctrl) in the GI data and (ERG, ERBB3) in the LINCS-Gene data have very different perturbation representations and cell state distributions. The perturbation representations of the pair of (RPL3, PINK1) in the GSPS data show distinctive distributions, while the difference between their cellular representations is less obvious. The perturbation representations have meaningful mappings to the cellular representations for the GI and LINCS-Gene datasets. The smaller difference in the cellular representation for the GSPS data might be due to its high batch effects (*Replogle et al.*, 2022).

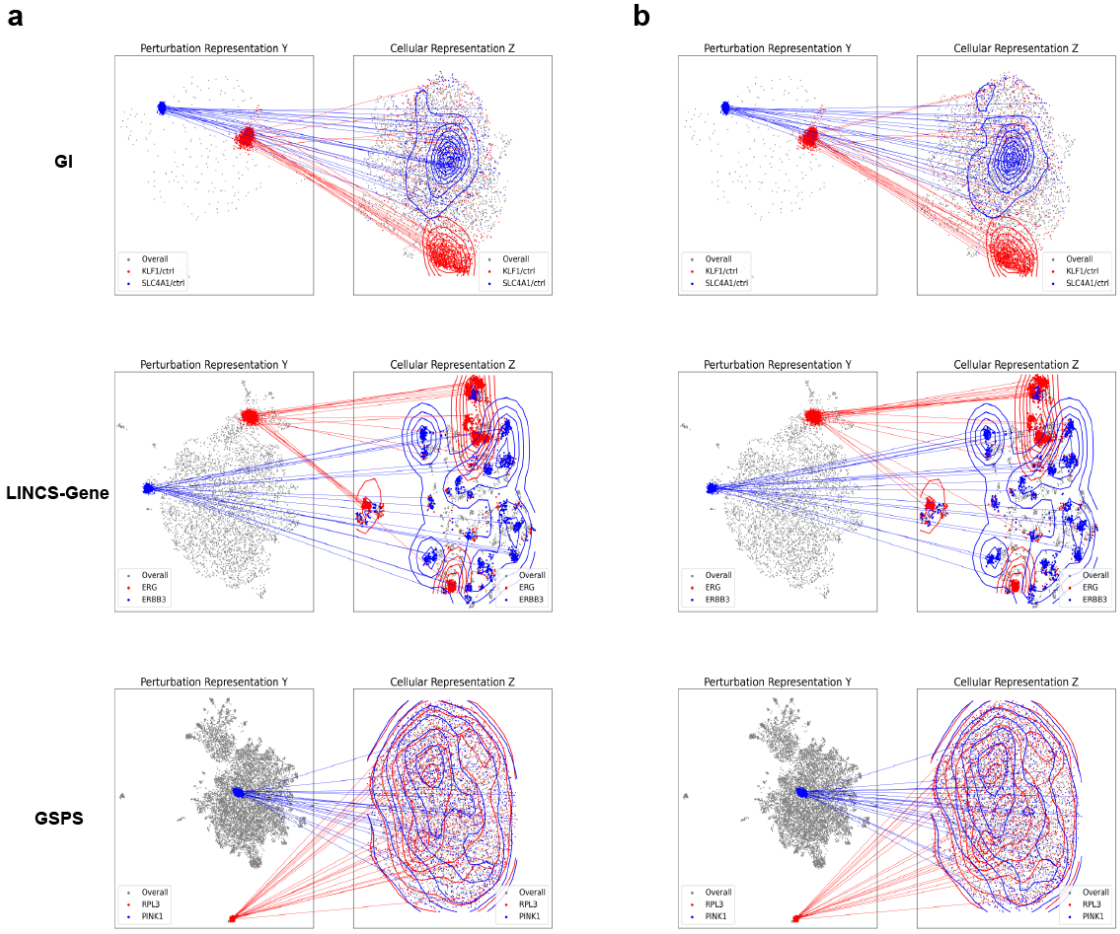

Figure 4.3: UMAP plots of perturbation representations and cellular representations (**a**) as well as reconstructed cellular representations (**b**) of three pairs of genetic perturbations in the GI, LINCS-Gene and GSPS datasets

We trained PerturbNet on the three datasets to recover the cellular representations from the perturbation representations. We first partitioned the genetic perturbations to observed and unseen perturbations for each dataset. We then used the cells with observed perturbations to train scVI on the GI and GSPS subset, as well as to train VAE on the LINCS-Gene subset. We then connected the GenotypeVAE latent space and each cell latent space by training a conditional invertible neural network (cINN) on the cells with observed perturbations. From these steps, we constructed PerturbNet for the three datasets. We then utilized PerturbNet to reconstruct the cell latent values for the three pairs of genetic perturbations in Figure 4.3b. The mappings between the perturbation representation and the cellular representation of the three perturbation pairs were precisely modeled by PerturbNet.

4.3.2 PerturbNet Predicts Single-Cell Response to Genetic Perturbations

We predicted single-cell responses to each genetic perturbation using the baseline KNN, random models and PerturbNet for the three datasets. Figure 4.4 shows the performance of predicted cell samples evaluated with R squared and FID metrics of KNN and PerturbNet over random on the GI data. Both KNN and PerturbNet significantly outperform the random model for the 180 observed perturbations. The two models are also significantly better than random for unseen perturbations in R squared, but the KNN model does not have significantly lower FID than the random model for the 50 unseen perturbations.

Figure 4.5 shows the prediction performance of KNN and PerturbNet over random for the LINCS-Gene data. Although the KNN model does not outperform the random model for either unseen or observed perturbations, PerturbNet has significantly lower FID than the random model for both unseen and observed perturbations, and also has higher R squared for the observed perturbations. The random model shows very high R squared values (around 0.75) for the LINCS-Gene data, possibly because

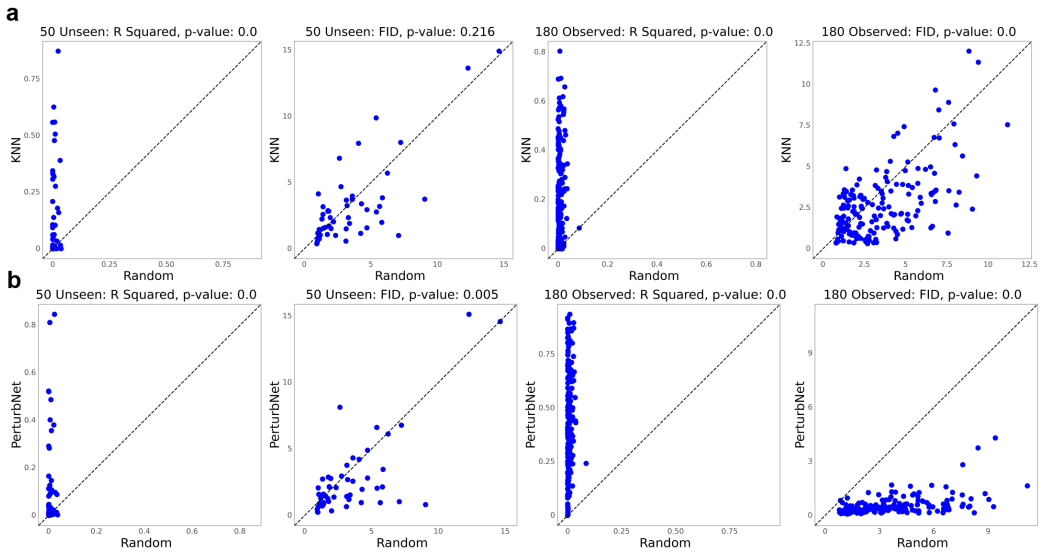

Figure 4.4: R squared and FID of KNN (**a**) and PerturbNet (**b**) over the random model for 50 unseen and 180 observed genetic perturbations of the GI data.

most genetic perturbations of the LINCS-Gene have small to medium perturbation effects. We expect that, if we focus on the subset of perturbations that have some detectable effect, the random model will be much less accurate and the PerturbNet will outperform it.

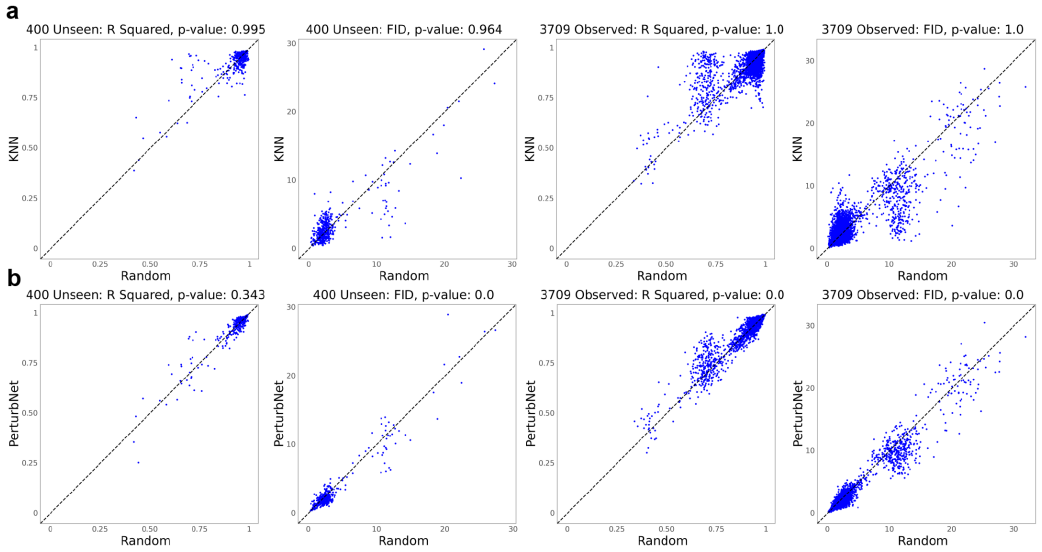

Figure 4.5: R squared and FID of KNN (**a**) and PerturbNet (**b**) over the random model for 400 unseen and 3709 observed genetic perturbations of the LINCS-Gene data.

We evaluated the three predictive models on the GSPS data (*Replogle et al.*, 2022), with a large number of target genes and a substantial proportion of perturbations with very few cells. We filtered out the genetic perturbations with fewer than or equal to 100 cells before evaluating the KNN and random models. We utilized the batch key of 'gemgroups' to train scVI models for PerturbNet and evaluations. Figure 4.6 shows the performance of the three models on the 802 unseen and 6859 observed genetic perturbations with more than 100 cells of the GSPS data. Both KNN and PerturbNet have significantly higher R squared than random for either unseen or observed perturbations. However, neither shows better FID than random, possibly due to complex batch effects in this large-scale scRNA-seq data (*Replogle et al.*, 2022).

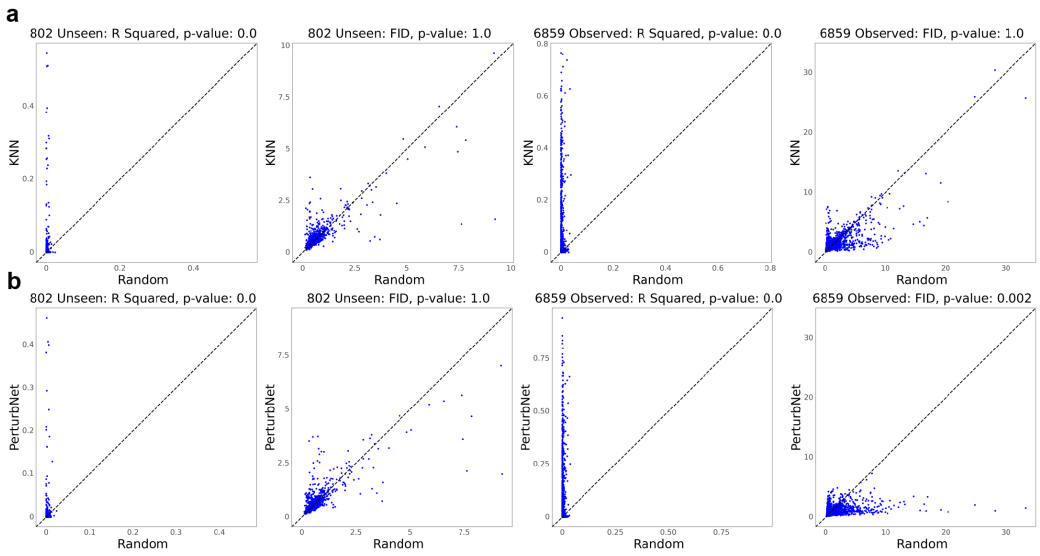

Figure 4.6: R squared and FID of KNN (**a**) and PerturbNet (**b**) over the random model for 802 unseen and 6859 observed genetic perturbations with more than 100 cells of the GSPS data.

We also compared the performance between KNN and PerturbNet for the three datasets of the GI, LINCS-Gene and GSPS data (Supplementary Figure 4.12a-c). PerturbNet shows better predictions than KNN for the observed perturbations of the three datasets. PerturbNet also gives significantly better R squared and FID for un-

seen perturbations of the LINCS-Gene data, and better FID for unseen perturbations of the GI and GSPS data. However, PerturbNet does not have a significant advantage in R squared over KNN for the unseen perturbations of the GI or GSPS data. The limited number of 180 observed genetic perturbations learned in the GI data might not extrapolate well to the 50 unseen perturbations, and the batch effects in the GSPS data might still have a negative impact on the predictions and evaluations of PerturbNet.

4.3.3 Fine-Tuned GenotypeVAE Improves the Performance of PerturbNet for Genetic Perturbations

We fine-tuned GenotypeVAE using the LINCS-Gene data following similar steps of the ChemicalVAE fine-tuning algorithm in Algorithm 4 of Chapter III. We first implemented the KNN algorithm on the perturbation representations of the genetic perturbations of the LINCS-Gene. We selected the five nearest neighbors of each perturbation, calculated the pairwise Wasserstein-2 (W2) distances between their cellular representations and set the distances between non-neighbors as 0's. We then averaged the distance matrix and its transposed matrix, calculated the exponential of their opposite values, and normalized to a unit sum for each row to obtain an adjacency matrix with each entry as a transition probability. We calculated the Laplacian $\boldsymbol{L}$ from the adjacency matrix, and used it as the graph to fine-tune GenotypeVAE. We alternated the GenotypeVAE training with a batch of genetic perturbations from the large GO annotation dataset using the evidence lower bound (ELBO) loss and another batch of genetic perturbations from the LINCS-Gene data using the loss as follows:

$$\text{Loss}^{\lambda}_{\boldsymbol{\phi},\boldsymbol{\theta}} = \text{-ELBO}(\boldsymbol{\phi},\boldsymbol{\theta}) + \lambda \text{trace}(\boldsymbol{y}^T \boldsymbol{L}_g \boldsymbol{y}),$$

where $\boldsymbol{\phi},\boldsymbol{\theta}$ were the parameters of the encoder and decoder of GenotypeVAE, and $\boldsymbol{L}_g$ and $\boldsymbol{y}$ were the Laplacian matrix and perturbation representations of the genetic

perturbations in the batch.

Figure 4.7 shows the R squared and FID of KNN and PerturbNet trained with fine-tuned GenotypeVAE ($\lambda = 0.1, 1, 5, 10, 100, 1000, 10000$). By comparing the evaluation metrics obtained from fine-tuned KNN and PerturbNet with different λ values, we determined that $\lambda = 1$ was the optimal hyperparameter. Figure 4.8 shows the scatter plots of R squared and FID of KNN and PerturbNet with fine-tuned GenotypeVAE of $\lambda = 1$ over those with non-fine-tuned GenotypeVAE. Fine-tuning GenotypeVAE significantly improves the performance of PerturbNet, especially for observed perturbations. Somewhat surprisingly, the fine-tuning algorithm improves only the PerturbNet, but does not significantly improve the performance of the KNN model.

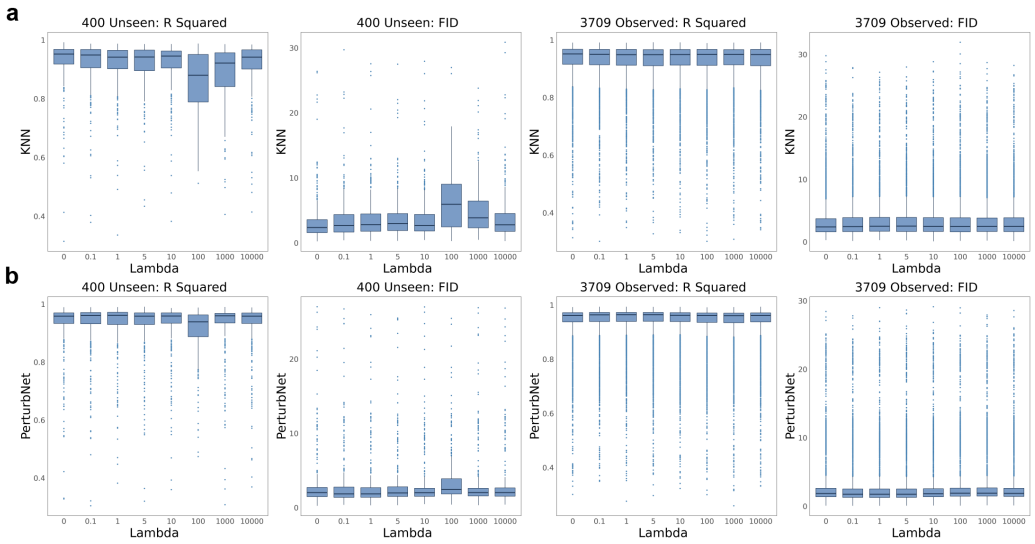

Figure 4.7: R squared and FID of KNN (**a**) and PerturbNet (**b**) with fine-tuned ChemicalVAE across different λ values for 400 unseen and 3709 observed genetic perturbations of the LINCS-Gene data.

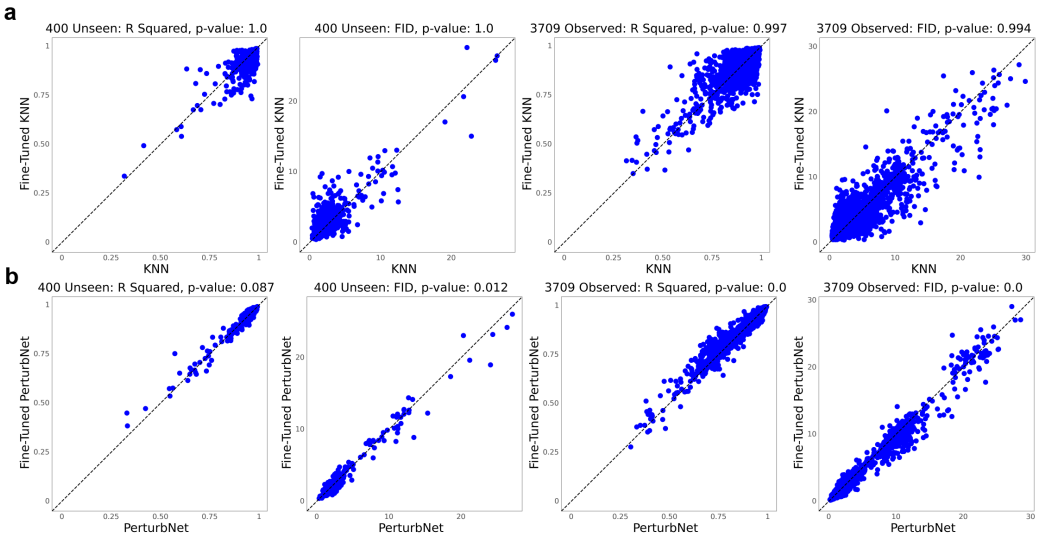

Figure 4.8: R squared and FID metrics of KNN and PerturbNet with fine-tuned GenotypeVAE of $\lambda = 1$ over non-fine-tuned PerturbNet for 400 unseen and 3709 observed genetic perturbations of the LINCS-Gene data.

4.3.4 PerturbNet Models Latent Representations of Protein Perturbations

We evaluated the performance of PerturbNet for predicting single-cell responses to protein-coding sequence variants. To do this, we used a Perturb-seq dataset of TP53 and KRAS variants introduced into the A549 cancer cell line (Ursu, *Ursu et al.*, 2020). We preprocessed the detected CRISPR guide RNA sequences to obtain a single, complete protein sequence for each of the perturbations. We then used the pretrained ESM transformer model to obtain deterministic representations of the protein perturbations. We experimented with the ESM representations by adding a small amount of Gaussian noise. We found that a noise term sampled from $\mathcal{N}(\mathbf{0}, 0.001\boldsymbol{I})$ effectively preserves the overall distribution of ESM representations (see Figure 4.9).

We trained scVI on the whole Ursu data and obtained the cellular representations of cells treated by a KRAS variant and a TP53 variant to visualize their mappings from perturbation representations to cellular representations (Figure 4.10a). Both the perturbation representations and cellular representations have distinct distribu-

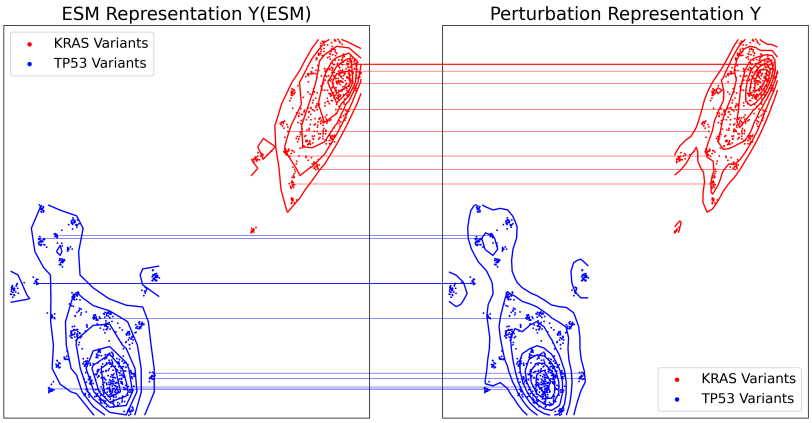

Figure 4.9: UMAP plots of ESM representations and perturbation representations for protein perturbations of the Ursu data.

tions for the two variants. We also partitioned the perturbations of coding variants into 1208 observed and 130 unseen perturbations. We trained scVI on the cells with the observed perturbations and constructed the cINN translations of the PerturbNet using the observed perturbation representation and cellular representation pairs. We utilized PerturbNet to reconstruct cellular representations from the perturbation representations (Figure 4.10b). The perturbation-cellular representation mappings of the two variants were accurately restored.

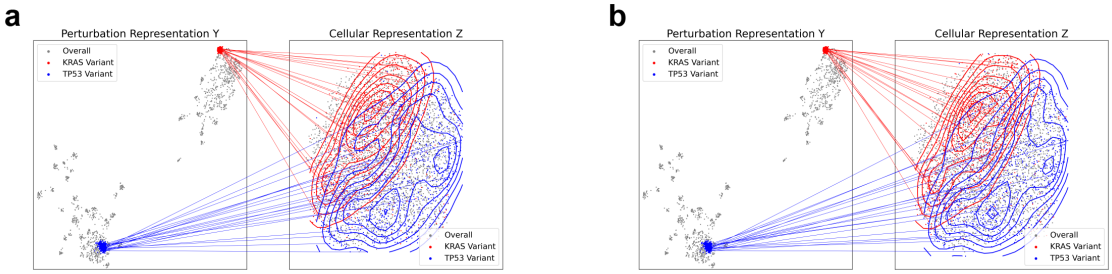

Figure 4.10: UMAP plots of perturbation representations and cellular representations (**a**) as well as reconstructed cellular representations (**b**) for two protein perturbations in the Ursu data.

4.3.5 PerturbNet Predicts Single-Cell Responses to Coding Sequence Mutations

We employed KNN and PerturbNet to predict the single-cell responses to protein perturbations in the Ursu data. We found a large variability for the number of cells per perturbation and small numbers of cells for a substantial proportion of variants. We filtered the variants to those with more than 400 cells for the baseline KNN and random models. Figure 4.11 shows the R squared and FID metrics of KNN and PerturbNet over the random model for both filtered unseen and filtered observed perturbations. Both KNN and PerturbNet have significantly better metric values than the random model for either unseen or observed perturbations.

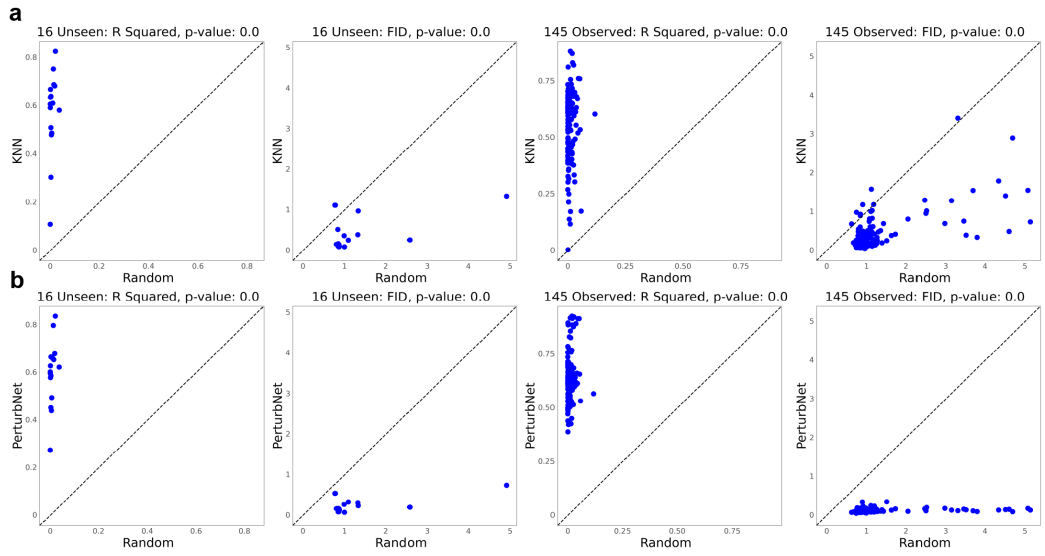

Figure 4.11: R squared and FID of KNN (**a**) and PerturbNet (**b**) over the random model for 16 unseen and 145 observed coding variants with more than 400 cells of the Ursu data.

We also compared the performance between KNN and PerturbNet for the Ursu data (Supplementary Figure 4.12d). PerturbNet shows better predictions than KNN for the observed perturbations, and has better FID than KNN for the unseen perturbations. However, PerturbNet does not perform better than KNN for the unseen perturbations in R squared, possibly due to their limited number (16).

4.4 Discussion

In this chapter, we extend PerturbNet to predict single-cell responses to genetic perturbations. We consider two types of genetic perturbations from CRISPR gene editing experiments, including one with a set of target genes, and another with an edited coding sequence. We develop the GenotypeVAE model to encode the first kind of genetic perturbations with GO annotation features for the target genes. We call the second type of genetic perturbations "protein perturbations," and employ a state-of-the-art transformer model (*Meier et al.*, 2021) to encode variant sequences. We demonstrate that both KNN and PerturbNet predict the single-cell responses to genetic perturbations. We also fine-tune the GenotypeVAE using the LINCS-Gene data and improve the performance of PerturbNet.

A limitation of the study is that the framework primarily focuses on predicting single-cell responses to genetic perturbations in Perturb-seq data, and does not generalize to other single-cell data. With new advances in generalizability between Perturb-seq data and general single-cell data, future research may infer single-cell perturbation responses for single-cell data such as the Tabula Muris compendium (*Consortium et al.*, 2018).

As representation learning techniques are rapidly evolving, future improvements of genetic perturbations can be utilized for PerturbNet. For example, *Van Den Oord et al.* (2017) introduced the VQ-VAE framework that learns discrete latent representations with very strong semantic meanings. In addition, VQ-VAE can model sequences with long term dependencies such as those in the coding variants of protein perturbations. The GenotypeVAE model could also be extended to model the hierarchical structure of GO terms (*Ma et al.*, 2018), rather than simply using a one-hot vector as we did here.

4.5 Supplementary Materials

4.5.1 Datasets

We obtained the GO annotation dataset for proteins of homo sapiens from GO Consortium at `http://geneontology.org/docs/guide-go-evidence-codes`. We removed the annotations of three sources without sufficient information: inferred from electronic annotation (IEA), no biological data available (ND) and non-traceable author statement (NAS). The filtered dataset had 15,988 possible annotations for 18,832 genes.

We obtained the GI data on GEO accession ID GSE133344 (*Norman et al.*, 2019). Each cell was perturbed with 0, 1 or 2 target genes. We processed the GI data using SCANPY (*Wolf et al.*, 2018) with 109,738 cells and 2279 genes. The processed GI data contained 236 unique genetic perturbations for 105 target genes and 11,726 cells were unperturbed. There were 230 out of 236 genetic perturbations that could be mapped to the GO annotation dataset. We randomly selected 50 genetic perturbations as unseen and the other 180 perturbations as observed.

We obtained the LINCS dataset (*Subramanian et al.*, 2017) from GEO accession ID GSE92742. The LINCS data had been processed with 1,319,138 cells and 978 landmark genes. The LINCS-Gene subset of the LINCS data contained 442,684 cells treated by 4371 genetic perturbations with single target genes. The 4109 out of 4371 genetic perturbations could be mapped to the GO annotation dataset, and we randomly selected 400 genetic perturbations as unseen perturbations and the other 3709 as observed perturbations.

We used SCANPY to preprocess the GSPS data (*Replogle et al.*, 2022) and to select the top 2000 highly-variable genes with respect to the batches of 'gemgroups'. The GSPS dataset contained 1,989,373 cells treated by 9867 genetic perturbations with single target genes. There were 9499 genetic perturbations that can be mapped

to the GO annotation library. We randomly selected 1000 genetic perturbations as unseen perturbations and the other 8499 as observed perturbations. There were 802 unseen and 6859 observed perturbations, each with more than 100 cells.

We obtained the Ursu data from GEO accession ID GSE161824, and filtered the raw data according to the processed datasets and concatenated the two datasets with KRAS variants and TP53 variants, using their common genes. We preprocessed the concatenated data using SCANPY, containing 164,931 cells and 1629 genes. We also collected the variants from the modifications on the original KRAS and TP53 protein sequences. We obtained 596 KRAS sequences and 742 TP53 protein sequences, and randomly selected 60 KRAS and 70 TP53 variants as unseen perturbations. There were 16 unseen and 145 observed variants with more than 400 cells.

4.5.2 Neural Network Architectures

The GenotypeVAE model has two fully-connected (FC) hidden layers with 512 and 256 neurons in its encoder, and also has two FC hidden layers with 256 and 512 neurons in its decoder. Each hidden layer is followed by a batch normalization layer, a Leaky Rectified Linear Unit (ReLU) activation and a dropout regularization with a dropout probability of 0.2. Two other layers with 10 neurons of the encoder generate means and standard deviations of the latent variable. An additional output layer and a sigmoid activation in the decoder output the 15,988-dimensional one-hot annotation vector. We adjusted different learning rates, batch size and epochs. We finally trained GenotypeVAE on the annotation vectors of single and double target genes from the GO annotation dataset with batch size of 128 for 300 epochs at a learning rate of 10^{-4}.

We utilized the pre-trained ESM model specialized for prediction of variant effects (*Meier et al.*, 2021) to obtain the perturbation representations of coding sequence mutations, and trained KNN and PerturbNet models. We utilized the same cINN

architecture and implementation in Chapter III for training PerturbNet.

For cellular VAE models, we utilized scVI version 0.7.1 with 10-dimensional latent space and its default setting of learning rate of 10^{-3} and batch size of 128. We trained scVI on the whole GI and Ursu data and their subsets with observed perturbations for 700 epochs. We trained scVI adjusted for the batch key of 'gemgroups' on the whole GSPS dataset and its subset with observed perturbations for 400 epochs. We trained a regular VAE model with multilayer perceptron (MLP) units in Chapter III on the LINCS-Gene data with observed perturbations with batch size of 128 and a learning rate of 10^{-4} for 150 epochs.

4.5.3 Supplementary Figures

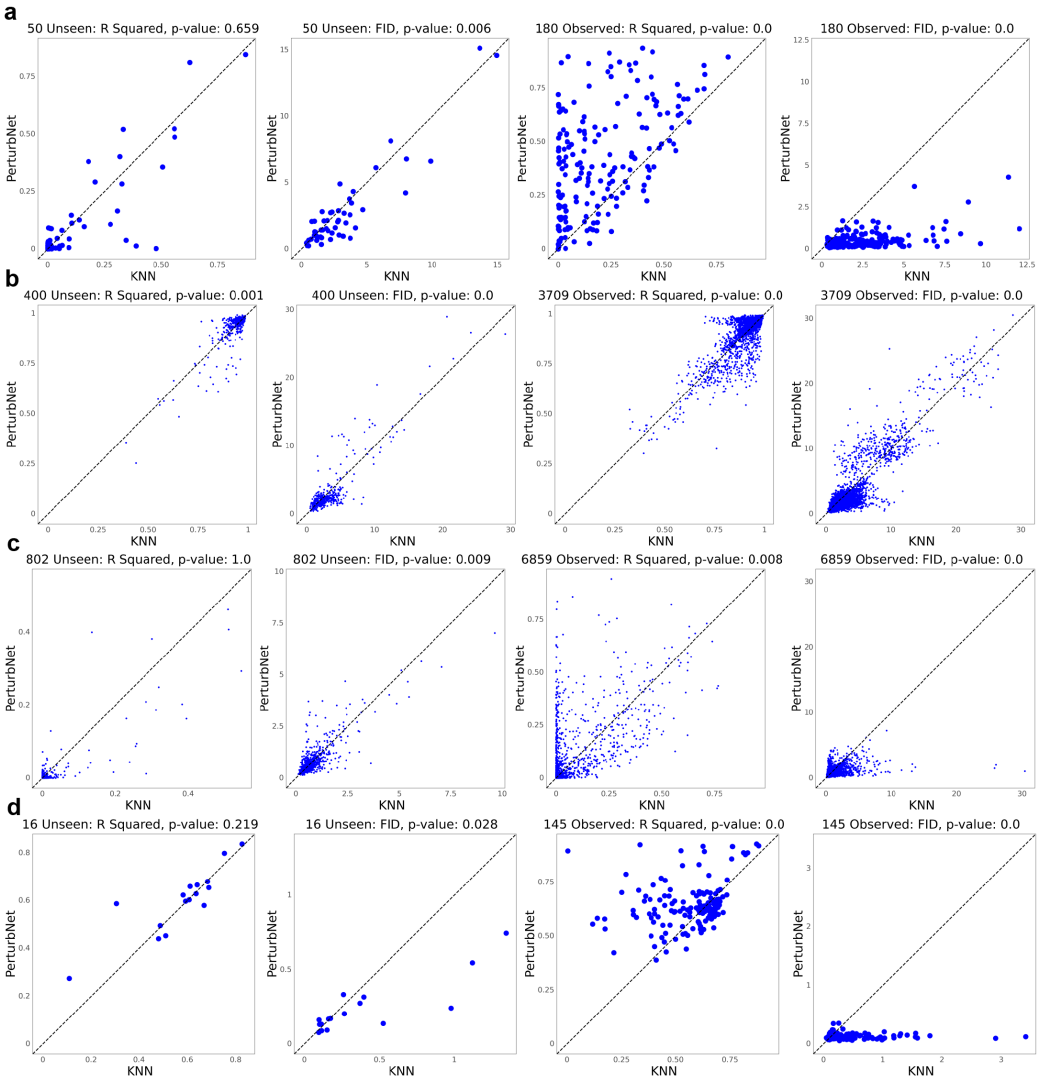

Figure 4.12: R squared and FID of PerturbNet over KNN for unseen and observed genetic perturbations of the GI (**a**), LINCS-Gene (**b**), GSPS (**c**) and Ursu (**d**) data.

CHAPTER V

Perturbation Design and Biological Discovery with PerturbNet

5.1 Introduction

In previous chapters, we have showed that PerturbNet can successfully predict the effects of unseen perturbations. In this chapter, we use the predictive modeling capabilities of PerturbNet to (1) design perturbations with desired effects on cell state and (2) discover the features of perturbations that predict their effects.

To summarize previous chapters: The PerturbNet framework has been shown to effectively model cellular responses from perturbations. It first learns perturbation and cellular representations independently through two powerful VAE-based models, avoiding potential interference of unbalanced joint distribution between perturbation and cell state of the method by *Lotfollahi et al.* (2020). Then, conditional invertible neural networks (cINN) connect the perturbation representation and cellular representation from individual cells. Therefore, the PerturbNet framework comprises a flexible multi-stage modeling process to learn representations and their relationships. The learned relationships among representations can further predict out-of-distribution or counterfactual cellular representations and cellular responses under various perturbations.

The flexibility of PerturbNet to predict out-of-distribution cellular responses provides pragmatic guidance on designing perturbations. For instance, characterizing counterfactual profiles of a group of diseased cells to several drug treatments may bring a potential cure for the disease or more understanding about an optimal drug to achieve the desired responses. Additionally, CRISPR genetic perturbations usually focus on a limited set of target genes due to the time-consuming and expensive experiments with potential side effects. The counterfactual profiles predicted by PerturbNet enhance the understanding of a desired genetic perturbation to shift the cell state of a group of cells, providing potential therapeutics for genetic diseases such as HIV. Thus, designing perturbations efficiently advances biomedical development, and can be guided from counterfactual cell responses predicted by PerturbNet.

These counterfactual responses not only enable perturbation design, but also reveal key components or functions in a chemical or genetic perturbation that influence the cell state. Estimating counterfactual responses is a fundamental problem and has numerous implications for understanding the heterogeneous effects of drugs (*Shalit et al.*, 2017; *Alaa and van der Schaar*, 2017). In these studies, some measurable quantitative trait serves as the potential outcome variable, and the heterogeneous effects are quantified and interpreted by individual treatment effect (ITE), which is the difference between the mean outcomes of a perturbation of interest and a baseline perturbation. High-dimensional single-cell responses, however, possess complex cellular information, and ITE does not uncover the underlying perturbation effects on the cell state. As a single-cell expression profile defines its cell state, understanding single-cell profiles of perturbations directly enables interpreting perturbation effects. Furthermore, PerturbNet's ability to predict cell states from perturbation features (atoms or gene ontology terms) provides an opportunity to hone in on mechanisms by implicating specific perturbation features. Recent developments in explainable artificial intelligence (XAI, *Gilpin et al.*, 2018) have improved the transparency of

model details and reasoning. Many model interpretability methods for neural networks can attribute classification decisions to certain key features of the input data (*Shrikumar et al.*, 2016, 2017; *Selvaraju et al.*, 2017; *Mudrakarta et al.*, 2018). These model interpretability methods usually involve interpreting the features of an input tensor compared to a baseline tensor in terms of their effects on the outcome of a neural network classifcation model (*Kokhlikyan et al.*, 2020). In the context of our application, these interpretability methods enable us to determine which input features make cells more likely to be in a particular cell state.

In this chapter, we use the counterfactual prediction capability of PerturbNet to design optimal perturbations that achieve desired effects. We consider a group of real cells treated by some perturbation, and we aim to learn an alternative perturbation that can translate these cells to approximate a desired cell state. To achieve the cell state translation, we utilize PerturbNet to extract the residual representation from pairs of perturbation and cellular representations, and to predict the counterfactual cellular representation under another perturbation. We propose two algorithms to design perturbations that optimally translate cells from the starting cell state to the desired cell state.

In addition, we interpret our predictive model to implicate key perturbation features that influence cell state distributions. We employ the method of integrated gradients to determine, for each input feature, whether the presence of the feature increases or decreases the probability of cells being in a particular state. We also interpret the attributions of the optimal chemical perturbation in the optimal translations. Finally, we identify GO terms that contribute greatly to the formation of different cell state distributions between two genetic perturbations.

5.2 Methods

5.2.1 Optimal Perturbation Design

Consider a starting cell state with latent space values $\tau_1(\boldsymbol{Z})$, and a target cell state with latent space values $\tau_2(\boldsymbol{Z})$. We want to find a perturbation that changes the cells in the starting cell state to the target cell state. From PerturbNet trained with single-cell perturbation responses, we can obtain the encoded representations for m cells in the starting cell state with the latent values $\{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_m\} \sim \tau_1(\boldsymbol{Z})$. Each starting cell is originally treated with a perturbation. For simplicity, we assume that these cells are treated with the same perturbation g_1. The target cell state can be represented by the latent values of n cells $\{\boldsymbol{z}_{m+1}, \ldots, \boldsymbol{z}_{m+n}\} \sim \tau_2(\boldsymbol{Z})$. The optimal translation task thus aims to find an alternative perturbation g^* for the starting cells to change their cell state to be close to $\tau_2(\boldsymbol{Z})$.

As PerturbNet translates perturbation representation $\boldsymbol{Y}$ and residual representation $\boldsymbol{V}$ to cellular representation $\boldsymbol{Z}$, we can predict the counterfactual cell state under a new perturbation for each cell with two translation procedures. Denote cINN forward translation as $f(\cdot)$, $\boldsymbol{B}_1$ as the perturbation matrix of the starting perturbation g_1 and the perturbation encoder as $h(\cdot)$. First, we obtain residual values $\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_m\}$ with the inverse translation function $\boldsymbol{v}_i = f^{-1}(\boldsymbol{z}_i \mid \boldsymbol{y}_i)$ with perturbation representation $\boldsymbol{y}_i = h(\boldsymbol{B}_1)$. The translation function then gives each cell's counterfactual cellular representation $\boldsymbol{z}_{i,*} = f(\boldsymbol{v}_i \mid \boldsymbol{y}^*)$ under an alternative perturbation's representation value $\boldsymbol{y}^*$. We therefore seek the translated counterfactual cell state $\{\boldsymbol{z}_{1,*}, \ldots, \boldsymbol{z}_{m,*}\} \sim \tau_*(\boldsymbol{Z})$ to have a similar distribution to $\{\boldsymbol{z}_{m+1}, \ldots, \boldsymbol{z}_{m+n}\} \sim \tau_2(\boldsymbol{Z})$.

5.2.2 Continuous Optimal Translation

We devise a method to design a perturbation representation $\boldsymbol{y}^*$ that shifts the cells in the starting cell state to approximate the target cell state. To quantify the differ-

ence between the cell state distributions, we use Wasserstein distance, which has been widely used to quantify cell populations' distance (*Schiebinger et al.*, 2019; *Crowley et al.*, 2020; *Demetci et al.*, 2020). We use Wasserstein-2 (W2) distance (*Vaserstein*, 1969), which is also known as Fréchet distance, to quantify the dissimilarity between the cell state distributions of $\tau_2(\boldsymbol{Z})$ and $\tau_*(\boldsymbol{Z})$. The W2 distance is defined as

$$d\{\tau_2(\boldsymbol{Z}), \tau_*(\boldsymbol{Z})\} = \left\{ \inf_{\gamma \in \Pi(\tau_2, \tau_*)} \mathbb{E}_{(\boldsymbol{Z}_2, \boldsymbol{Z}_*) \sim \gamma} \|\boldsymbol{Z}_2 - \boldsymbol{Z}_*\|^2 \right\}^{1/2},$$

where $\Pi(\tau_2, \tau_*)$ is the set of all joint distributions $\gamma(\boldsymbol{Z}_2, \boldsymbol{Z}_*)$ whose marginal distributions are $\tau_2(\boldsymbol{Z})$ and $\tau_*(\boldsymbol{Z})$, respectively.

Evaluating the W2 distance is extremely difficult for general distributions. To simplify the calculations of the W2 distance, we assume that latent spaces follow multivariate Gaussian distributions (*Dowson and Landau*, 1982; *Heusel et al.*, 2017), which is also commonly assumed in calculating Fréchet inception distance (FID) in image data (*Heusel et al.*, 2017). Assuming that the latent space $\tau_i(\boldsymbol{Z})$ has a multivariate Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$ for $i \in \{2, *\}$, the squared W2 distance has a closed form:

$$d^2\{\tau_2(\boldsymbol{Z}), \tau_*(\boldsymbol{Z})\} = \|\boldsymbol{\mu}_2 - \boldsymbol{\mu}_*\|_2^2 + \text{trace}\{\boldsymbol{\Sigma}_2 + \boldsymbol{\Sigma}_* - 2(\boldsymbol{\Sigma}_2\boldsymbol{\Sigma}_*)^{1/2}\}. \tag{5.1}$$

Therefore, we can evaluate the squared W2 distance between the translated counterfactual cell state and the target cell state as $d^2[\{\boldsymbol{z}_{m+j}\}_{j=1}^n, \{\boldsymbol{z}_{i,*}\}_{i=1}^m]$. The problem of designing a desired perturbation is then to find the optimal $\boldsymbol{y}_{\text{opt}}^*$ that minimizes the squared W2 distance:

$$\boldsymbol{y}_{\text{opt}}^* = \arg\min_{\boldsymbol{y}^*} d^2 \left[\{\boldsymbol{z}_{m+j}\}_{j=1}^n, \{\boldsymbol{z}_{i,*}\}_{i=1}^m\right].$$

We can further infer the optimal perturbation from representation $\boldsymbol{y}_{\text{opt}}^*$. Figure 5.1

summarizes the procedure to design the optimal perturbation using PerturbNet.

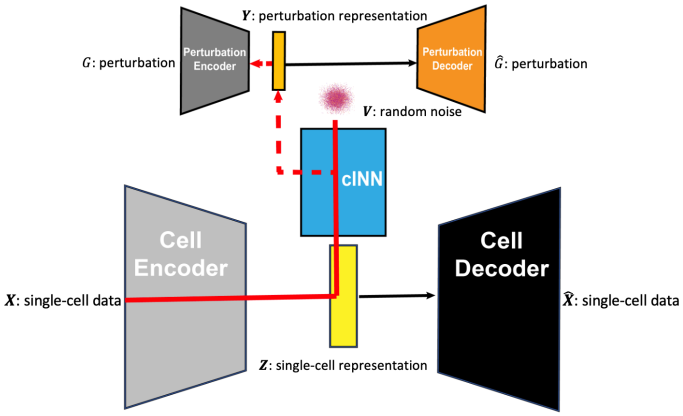

Figure 5.1: Overview of the translation optimization.

Based on the objective above, we propose what we refer to as "continuous optimal translation." We first initialize a value for $\boldsymbol{y}^*_{\text{opt}}$ from the standard multivariate Gaussian distribution and then we perform stochastic gradient descent with momentum (*Kingma and Ba*, 2014) to minimize the squared W2 loss over $\boldsymbol{y}^*_{\text{opt}}$. One important implementation detail concerns the calculation of the W2 distance. The distance formula includes the term $(\boldsymbol{\Sigma}_2\boldsymbol{\Sigma}_*)^{1/2}$, which is difficult to calculate and can become ill-conditioned or approximately singular. We thus rewrite the term as

$$\boldsymbol{C}_{2,*} = \boldsymbol{\Sigma}_2^{1/2}\left(\boldsymbol{\Sigma}_2^{1/2}\boldsymbol{\Sigma}_*\boldsymbol{\Sigma}_2^{1/2}\right)^{1/2}\boldsymbol{\Sigma}_2^{-1/2},$$

which allows us to replace the difficult term with $\boldsymbol{C}^2_{2,*}$ as $\boldsymbol{C}^2_{2,*} = \boldsymbol{\Sigma}_2\boldsymbol{\Sigma}_*$.

We use the Adam optimizer to perform stochastic gradient descent with momentum. For the matrix square root terms in $\boldsymbol{C}_{2,*}$, $\boldsymbol{\Sigma}_2^{1/2}$ keeps a fixed value during training, and $\boldsymbol{\Sigma}_2^{1/2}\boldsymbol{\Sigma}_*\boldsymbol{\Sigma}_2^{1/2}$ is much more likely than the original term $\boldsymbol{\Sigma}_2\boldsymbol{\Sigma}_*$ to be symmetric positive semi-definite and have a square root matrix. After the continuous optimization, we obtain an optimal perturbation representation $\boldsymbol{y}^*_{\text{opt}}$ that represents a potential perturbation that achieves the desired shift in cell state distribution.

5.2.3 Discrete Optimal Translation

The continuous optimal translation model can give an optimal perturbation representation $\boldsymbol{y}^*_{\text{opt}}$ that translates the starting cells to have a similar cell state to $\tau_2(\boldsymbol{Z})$. If the real cells in the target cell state $\{\boldsymbol{z}_{m+1}, \ldots, \boldsymbol{z}_{m+n}\} \sim \tau_2(\boldsymbol{Z})$ are treated by a perturbation g_2, we can compare it with the fitted optimal perturbation representation $\boldsymbol{y}^*_{\text{opt}}$ to evaluate if the optimal perturbation representation can achieve the desired cell state shift like the perturbation g_2.

However, the chemical or genetic perturbation from the optimal perturbation representation of a continuous optimal translation is not immediately clear, as an inference model needs to be processed on the perturbation representation. Although it is possible to employ the perturbation generative model to generate chemical or genetic perturbations, doing so brings a host of additional challenges related to molecular structure optimization (*Gómez-Bombarelli et al.*, 2018), which is not the focus of this dissertation.

To design the optimal perturbation to achieve the desired cell state shift, we propose another perturbation design strategy that uses discrete optimization. Rather than optimizing the squared W2 loss in the continuous space, the discrete optimal translation searches through a constrained set $\mathcal{G}$ of perturbations, and calculates the squared W2 distance $d^2[\{\boldsymbol{z}_{m+j}\}_{j=1}^n, \{\boldsymbol{z}_{i,*}\}_{i=1}^m]$ for each perturbation $g \in \mathcal{G}$ with $\boldsymbol{y}^* = h(\boldsymbol{B}_g)$. Then the optimal perturbation is selected as the one giving the smallest distance so that

$$g^*_{\text{opt}} = \arg\min_{g \in \mathcal{G}} d^2[\{\boldsymbol{z}_{m+j}\}_{j=1}^n, \{\boldsymbol{z}_{i,*}\}_{i=1}^m].$$

This discrete optimal translation strategy gives both the optimal perturbation representation $\boldsymbol{y}^*_{\text{opt}}$ to achieve the desired translation, and also the optimal perturbation g^*_{opt}. If the cells in the target latent space are treated by a perturbation, we can evaluate if the optimal perturbation g^*_{opt} matches the one for the target latent space.

5.2.4 Model Interpretation Using Integrated Gradients

As we connect perturbation and cell state in PerturbNet, we can interpret how a perturbation changes the cell state distribution by predicting cellular representations using PerturbNet. We can further interpret the effects of features and components of the perturbation with the state-of-the-art XAI methods. Denote $F(\cdot)$ as a function taking input feature vector $\boldsymbol{T} = (T_1, \ldots, T_n)^T \in \mathbb{R}^n$ to generate output in $[0, 1]$. Then its attribution is a vector $\boldsymbol{A} = (a_1, \ldots, a_n)^T$ and each value a_i is the contribution of T_i to the prediction of $F(\boldsymbol{T})$.

Previous attempts to interpret neural network models have focused on gradients (*Baehrens et al.*, 2010; *Simonyan et al.*, 2013) and back-propagation (*Shrikumar et al.*, 2016, 2017). We use the method of integrated gradients (*Sundararajan et al.*, 2017), which has been applied to interpret deep learning models across a range of domains, including computational chemistry (*McCloskey et al.*, 2019). The attribution score of the integrated gradients method for the ith dimension of input $\boldsymbol{T}$ is defined as

$$a_i = (T_i - T_{0,i}) \int_{\alpha=0}^{1} \frac{\partial F\{\boldsymbol{T}_0 + \alpha(\boldsymbol{T} - \boldsymbol{T}_0)\}}{\partial T_i} d\alpha,$$

where $\boldsymbol{T}_0 = (T_{0,0}, \ldots, T_{0,n})^T$ is a baseline input.

A prediction neural network model on cellular representation can be formulated from PerturbNet as $\boldsymbol{Z} = f(\boldsymbol{V} \mid \boldsymbol{Y})$ and $\boldsymbol{Y} = h(\boldsymbol{B})$. The input $\boldsymbol{T}$ can be formulated as $(\boldsymbol{V}^T, \boldsymbol{Y}^T)^T$ or $(\boldsymbol{V}^T, \boldsymbol{B}^T)^T$. In addition, a classification neural network model on $\boldsymbol{Z}$ provides a classification score within $[0, 1]$. We can then find input features that increase the probability of generating cells in a particular cell state.

5.3 Experiments

5.3.1 Continuous Optimal Translation for Perturbation Representations

We performed continuous optimal translation on the chemical perturbations of the sci-Plex and LINCS-Drug datasets. For PerturbNet trained on sci-Plex without adjusting for cell state covariates, we used the latent values of cells treated by S1172 as the starting latent space, and considered the target latent space as the latent values of each of the 158 observed drug treatments. For the LINCS-Drug dataset, we also fixed a drug treatment as the starting perturbation and used each of the 200 selected observed drug treatments as the target perturbation. For each translation on the two datasets, we trained the continuous optimization algorithm for 600 epochs to obtain the optimal perturbation representation. As the cells of the target latent space are also treated with a perturbation, we also evaluated the translation using the perturbation representation of the target perturbation.

The target perturbation, however, does not necessarily translate the cells in the starting latent space to overlap with the target latent space. Figure 5.2a shows a translation example to translate the cells treated by S1007 to a target latent space using the target perturbation S1628. As can be seen, the translated counterfactual latent values from S1007 to S1628 have a distinct distribution from the latent values of real cells treated by S1628. This distinction is due to the fact that the residual representation $\boldsymbol{V}$ of the real cells treated by S1007 is different from that of S1628, and therefore a translation of S1007 cells using S1628 potentially gives a different latent distribution. Thus, the target perturbation might not perform well in translating the starting latent space to approximate the target latent space, and might even enlarge the original distance between the two latent spaces.

Figure 5.3 shows the continuous optimal translations for the sci-Plex and LINCS-Drug datasets. For simplicity, we named squared W2 distance and W2 distance in-

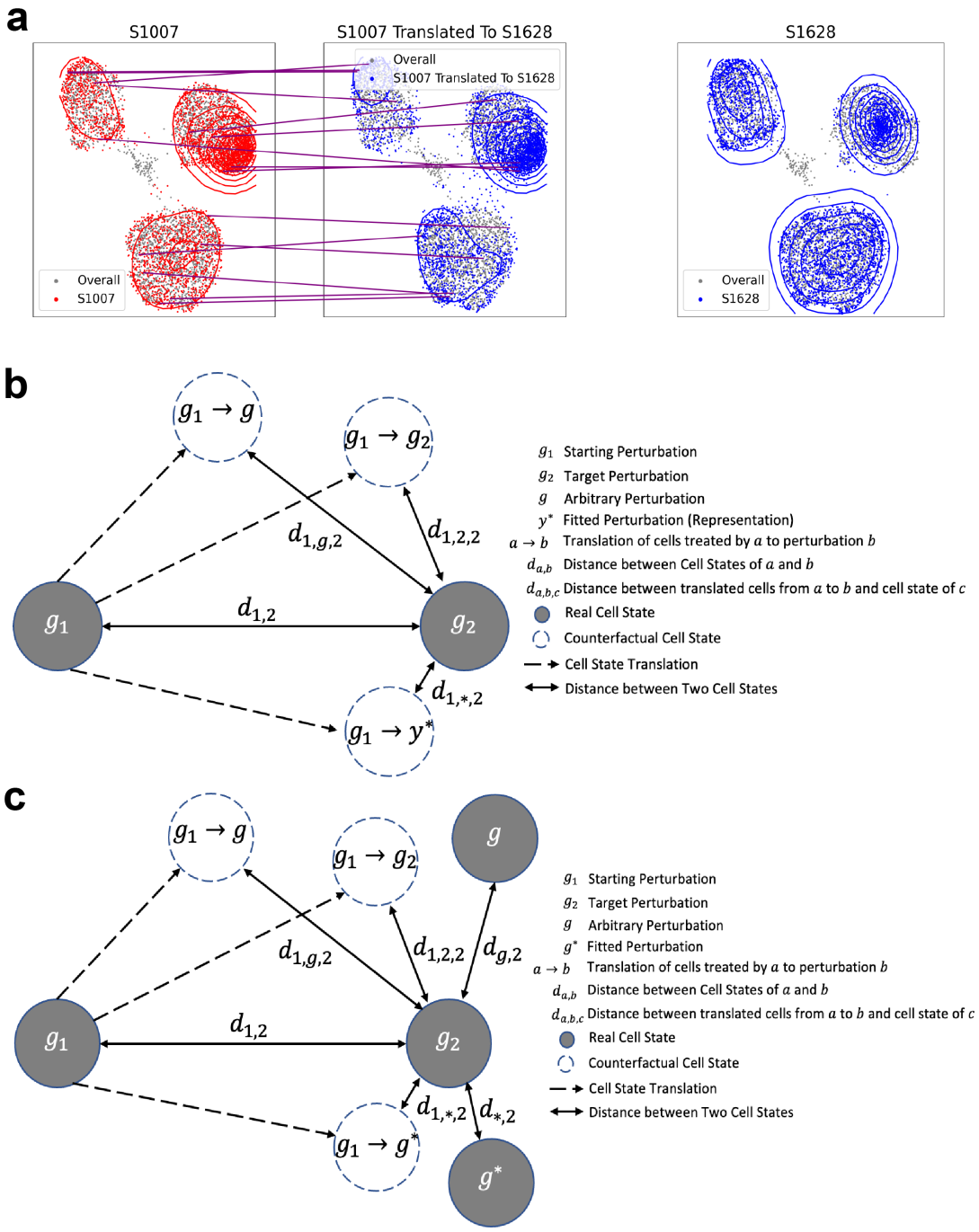

Figure 5.2: **a** UMAP plots of latent values of cells treated by S1007, their translated latent values to treatment S1628, and latent values of real cells treated by S1628 in the sci-Plex data. **b** Diagram of evaluation measures for a continuous optimal translation experiment. **c** Diagram of evaluation measures for a discrete optimal translation experiment.

terchangeably, both corresponding to the form in Equation (5.1). Figure 5.2b shows the evaluation measures for each continuous optimal translation experiment. We calculated the W2 distances between the target latent space and the translated latent space, using either the fitted perturbation representation ($d_{1,*,2}$) or the target perturbation representation ($d_{1,2,2}$). We then normalized the two distances by the original W2 distance between the starting latent space and target latent space ($d_{1,2}$), and called them normalized fitted W2 distance and normalized target W2 distances. A normalized W2 distance smaller than 1 means that the translation reduces the original distance between the two latent spaces. We show the scatter plots of the normalized fitted W2 and normalized target W2 for the sci-Plex (Figure 5.3a) and LINCS-Drug (Figure 5.3b) data. We found that 99.4% of the translations using either fitted or target perturbation effectively reduced the original latent distances for sci-Plex data. The fitted perturbation representation has an overall close performance to the target perturbation. For the LINCS-Drug data, the target perturbation can shrink the original latent distances in 67.5% of the 200 translations, while the fitted perturbation representations have a better performance to shrink the distance in 88% of the translations. For translations shrinking the original latent distance from both target representation and fitted representation, the fitted representation can sometimes perform better than the target perturbation (Figure 5.3b). This means that the target perturbation is possibly not optimal to reduce the latent distances, and the continuous optimal translation algorithm provides a better perturbation representation to achieve the latent approximation.

We also plot the percentiles of the fitted W2 ($d_{1,*,2}$ in Figure 5.2b) and target W2 ($d_{1,2,2}$ in Figure 5.2b) in the distribution of W2 distances between the target latent space and the 2000 translated latent spaces across the 200 translations ($d_{1,g,2}$ in Figure 5.2b) for the sci-Plex (Figure 5.3c) and LINCS-Drug (Figure 5.3d) data. Both the fitted and target perturbations tend to give W2 percentiles of 0's for the sci-Plex

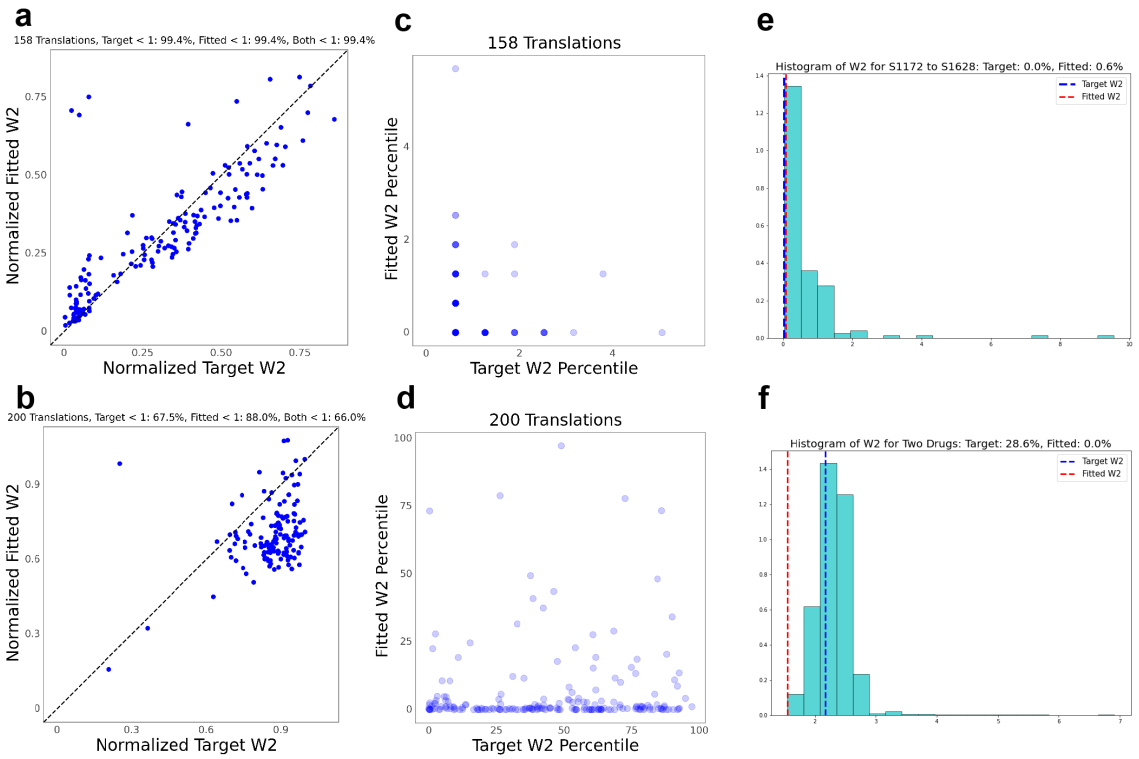

Figure 5.3: Continuous optimal translations of the sci-Plex and LINCS-Drug data. **a** Scatter plot of normalized fitted W2 and normalized target W2 for 158 continuous optimal translations of the sci-Plex data. **b** Scatter plot of normalized fitted W2 and normalized target W2 for 200 continuous optimal translations of the LINCS-Drug data. **c** Scatter plot of fitted W2 percentile and target W2 percentile for 158 continuous optimal translations of the sci-Plex data. **d** Scatter plot of fitted W2 percentile and target W2 percentile for 200 continuous optimal translations of the LINCS-Drug data. **e** Histogram of the W2 distances between the target latent space of S1628 and the translated latent space from cells treated by S1172 to each of the 158 observed drug treatments, along with fitted W2, target W2 and their percentiles in the histogram. **f** Histogram of the W2 distances between the target latent space of a target drug treatment and the translated latent space from the cells treated by a starting drug treatment to each of the 2000 sampled observed drug treatments, along with fitted W2, target W2 and their percentiles in the histogram.

translations, while LINCS-Drug has small percentile values for fitted W2 but varying target W2 percentiles. The LINCS-Drug dataset has many more perturbations than the sci-Plex and most of the perturbations have small numbers of cells. As a result, the starting and target latent spaces of LINCS-Drug might possess an overall

larger difference in their residual representations, and the translation using the target perturbation possibly does not perfectly approximate the target latent space.

To evaluate how well the fitted perturbation representation translates the starting latent space to approximate the target latent space in a translation, we computed the W2 distances between the target latent space and a translated latent space using each of the available perturbations ($d_{1,g,2}$ in Figure 5.2b). Figure 5.3e shows a translation example with the histogram of W2 distances for the translation from the starting perturbation of S1172 to the target perturbation of S1628, where the translations using both the fitted perturbation representation ($d_{1,*,2}$ in Figure 5.2b) and the target drug treatment S1628 ($d_{1,2,2}$ in Figure 5.2b) have W2 distances smaller than almost all of the W2 distances between the target latent space and the translated latent space via the 158 observed drug treatments. For the LINCS-Drug data, we also randomly sampled 2000 observed drug treatments to compute the distribution of W2 distances between the target latent space and the translated latent space, and show a translation example in Figure 5.3f. Of the 2000 translations, 28.6% have W2 distances smaller than the target W2, and none of the 2000 translations has a W2 distance smaller than the fitted W2.

5.3.2 Discrete Optimal Translation for Optimal Perturbation Selections

We performed discrete optimal translations on the sci-Plex and LINCS-Drug data to select the optimal drug treatment to translate a starting latent space to approximate a target latent space. For the sci-Plex, we randomly selected 10 observed drug treatments as the set of starting perturbations and considered the 158 observed drug treatments as the set of target perturbations. We implemented the discrete optimal translation algorithm on each pair of a starting and target perturbations, with evaluation measures for each discrete optimal translation experiment shown in Figure 5.2c. Figure 5.4 shows the 1580 discrete optimal translations of the sci-Plex

data. The normalized fitted W2 ($d_{1,*,2}/d_{1,2}$) and normalized target W2 ($d_{1,2,2}/d_{1,2}$) are overall the same, both shrinking the original latent distance in around 99.5% of the translations. Because the starting latent space and target latent space in each translation might possess different distributions of residual representation $\boldsymbol{V}$, we also computed the W2 distances of residual $\boldsymbol{V}$'s between the two latent spaces across the 1580 translations, and categorized the W2 distances to three tertiles (Figure 5.4b). Most of the translations are with the smallest tertile of residual distance (V Distance 1) and do not show an obvious difference from the translations with other tertiles in their normalized fitted and target W2 distances. We also trained a KNN algorithm on the observed treatments and evaluated the nearest neighbor index of the target perturbation g_2 to the fitted perturbation g^* in each translation. We show KNN indices of the target perturbation to the fitted perturbation in Figure 5.4c and also computed the percentile of the W2 distance between the latent spaces of real cells treated by the target and the fitted perturbations ($d_{*,2}$) in the distribution of $d_{g,2}$, the W2 distances between the target perturbation and other perturbations (Figure 5.4d). Of the 1580 translations, 82.5% were selected with the target perturbation as the fitted perturbation, whose KNN indices and W2 distances were 0's. Figure 5.4e shows a discrete optimal translation example with the W2 distances between the target latent space of cells treated by S1703 and the translated latent space from a starting perturbation of S1515, using each of the 158 observed drug treatments ($d_{1,g,2}$). The fitted W2 ($d_{1,*,2}$) has the smallest W2 among all the translations and 0.6% of the translations have smaller W2 distances than the target W2 ($d_{1,2,2}$).

We also considered translation optimizations of the sci-Plex data in each cell type by dose stratum, and employed the PerturbNet adjusting for cell state covariates to perform the discrete optimal translations between each pair of the 10 starting perturbations and the 158 target perturbations. Figure 5.5a-c show the normalized fitted W2 and normalized target W2 of the 18,960 translations by cell type, dose and

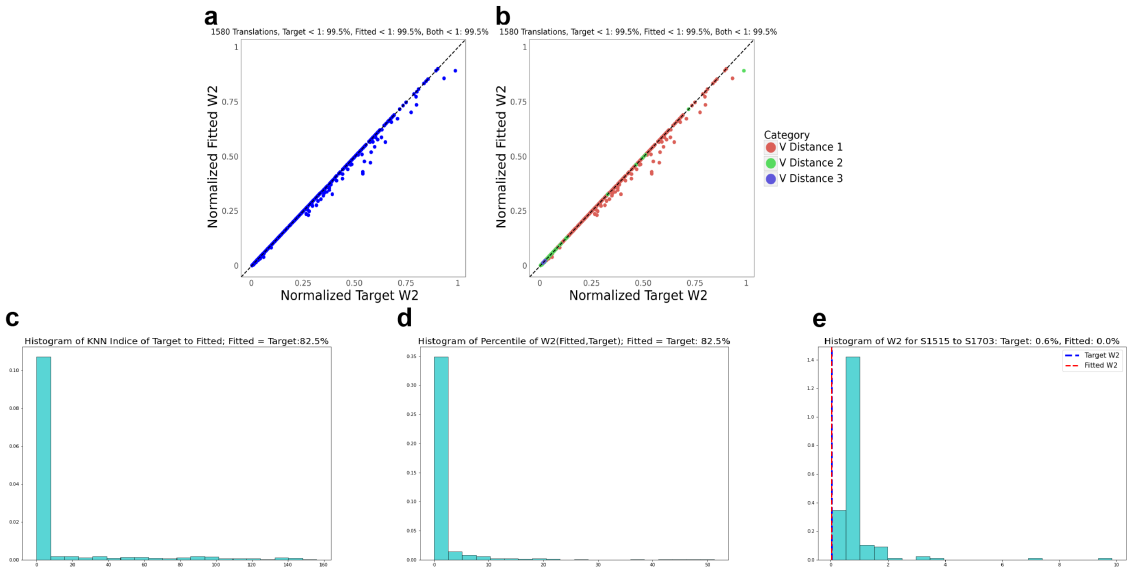

Figure 5.4: Discrete optimal translations of the sci-Plex data. **a** Scatter plot of normalized fitted W2 and normalized target W2 for 1580 discrete optimal translations. **b** Scatter plot of normalized fitted W2 and normalized target W2 for the 1580 discrete optimal translations by residual distance tertile. **c** Histogram of KNN indices of target perturbation to fitted perturbation for 1580 discrete optimal translations. **d** Histogram of percentiles of W2 distances between the latent spaces of the real cells treated by fitted perturbation and target perturbation in the distribution of the W2 distances between the latent spaces of the real cells treated by the target perturbation and other perturbations across the 1580 discrete optimal translations. **e** Histogram of the W2 distances between the target latent space of S1703 and the translated latent space from the cells treated by S1515 to each of the 158 observed drug treatments, along with fitted W2, target W2 and their percentiles in the histogram.

residual W2 distance tertile. The comparison between the normalized fitted W2 and normalized target W2 remains similar across different cell types or doses. Most of the translations have residual distances in the smallest tertile (V Distance 1), while the translations with residual distances in the second tertile tend to have smaller fitted W2 than target W2. Around 29.3% of the translations select the target perturbation as the fitted perturbation (Figure 5.5d-e). Finally, as an example, Figure 5.5f shows the histogram of the W2 distances between the target latent space of S1315 and the translated latent space from a starting drug treatment of S1122 to each of the 158

observed drug treatments with cell type K562 and dose 10. The fitted W2 has a much better performance than the target W2 that is larger than around half of the translations.

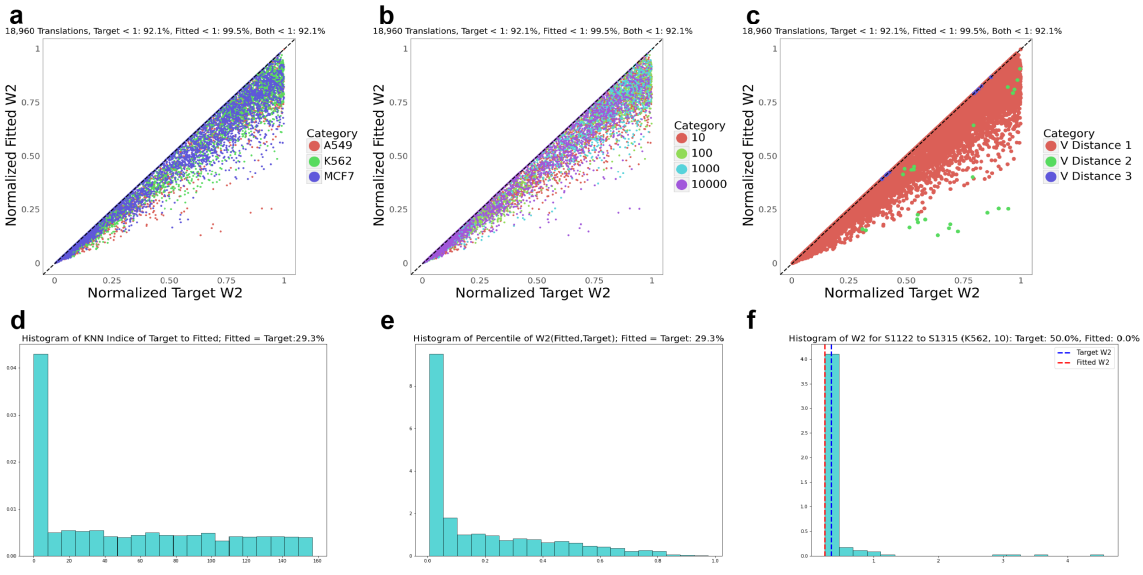

Figure 5.5: Stratified discrete optimal translations of the sci-Plex data. **a-c** Scatter plot of normalized fitted W2 and normalized target W2 for 18,960 discrete optimal translations by cell type, dose and residual distance tertile. **d** Histogram of KNN indices of target perturbation to fitted perturbation for 18,960 discrete optimal translations. **e** Histogram of percentiles of W2 distances between the latent spaces of the real cells treated by fitted perturbation and target perturbation in the distribution of the W2 distances between the latent spaces of the real cells treated by the target perturbation and other perturbations across the 18,960 discrete optimal translations. **f** Histogram of the W2 distances between the target latent space of S1315 and the translated latent space from the cells treated by S1122 to each of the 158 observed drug treatments with cell type K562 and dose 10, along with fitted W2, target W2 and their percentiles in the histogram.

We previously sampled 200 observed drug treatments from the LINCS-Drug data for continuous optimal translations, and these perturbations had small numbers of cells (< 200). Thus, we selected another five drug treatments with relatively large numbers of cells (> 1000). We utilized the five drug treatments and another two drug treatments in the previous 200 treatments as the set of starting perturbations. We

considered the combined set of 205 drug treatments as the set of target perturbations. Figure 5.6a shows the normalized fitted W2 and normalized target W2 by whether both the starting and target perturbations have large numbers of cells (> 1000). The translations with large numbers of cells tend to give close normalized fitted W2 and normalized target W2. Figure 5.6b shows the normalized fitted W2 and normalized target W2 by residual distance tertile, and most of the translations have small to medium residual distances. Only 5.3% of the 1435 translations select the target perturbation as the fitted perturbation (Figure 5.6c), and the latent W2 distance from target perturbation to the fitted perturbation is likely to be smaller than distances from the target perturbation to other perturbations (Figure 5.6d). Figure 5.6e shows a discrete optimal translation example between two drugs in LINCS-Drug, where the fitted perturbation is different from the target perturbation, and the fitted perturbation gives a percentile close to zero in the histogram of W2 distances between the target latent space and translated latent space using each of the 205 drug treatments, while the target W2 is larger than around 60% of the W2 distances.

Figure 5.7 shows the UMAP plots of starting latent space, target latent space and translated latent spaces of a stratified discrete optimal translation example of the sci-Plex data. The starting latent space of K562 cells treated by S1122 with dose 100 has a different latent distribution from that of the K562 cells treated by S2692 with dose 100. The starting latent space changes slightly after being translated to latent spaces using the target treatment S2692 or the fitted treatment S2736. Based on the W2 distance, the translated latent space using the fitted perturbation S2736 is closer to the target latent space than the one using the target perturbation. Figure 5.8 shows an discrete optimal translation example for the LINCS-Drug data. The cells treated by the starting perturbation G1 are translated to different latent distributions using the fitted perturbation G3 and the target perturbation G2. The translated latent distribution using the fitted perturbation G3 is closer to the target latent space, and

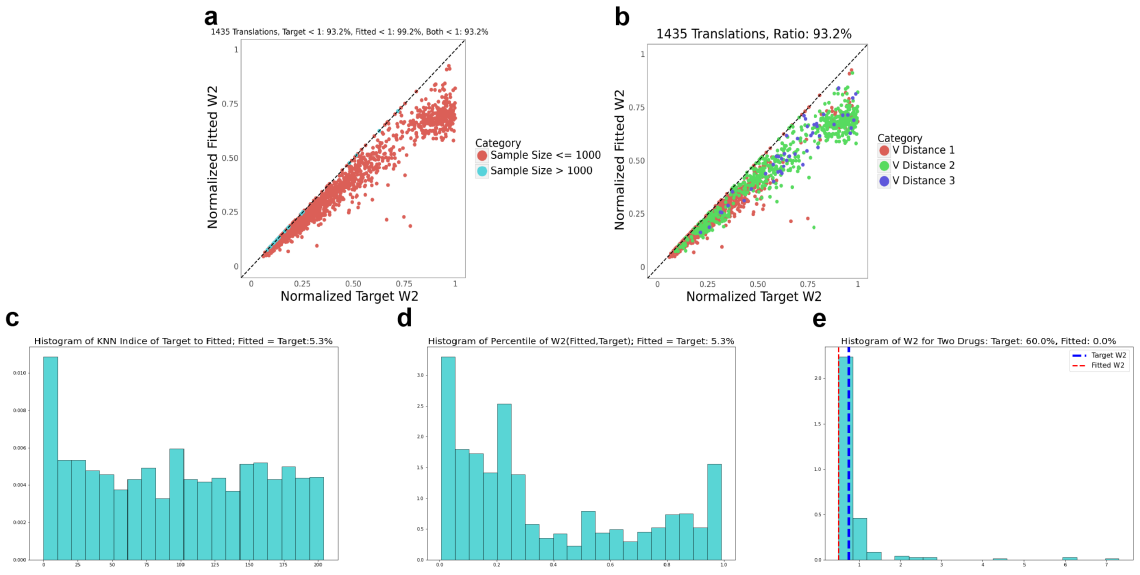

Figure 5.6: Discrete optimal translations of the LINCS-Drug data. **a** Scatter plot of normalized fitted W2 and normalized target W2 for 1435 discrete optimal translations. **b** Scatter plot of normalized fitted W2 and normalized target W2 for 1435 discrete optimal translations by residual distance tertile. **c** Histogram of KNN indices of target perturbation to fitted perturbation for 1435 discrete optimal translations. **d** Histogram of percentiles of W2 distances between the latent spaces of the real cells treated by fitted perturbation and target perturbation in the distribution of the W2 distances between the latent spaces of the real cells treated by target perturbation and other perturbations across the 1435 discrete optimal translations. **e** Histogram of the W2 distances between the target latent space of a drug treatment and the translated latent space from the cells treated by a starting drug treatment to each of the 205 observed drug treatments, along with fitted W2, target W2 and their percentiles in the histogram.

differs from the real latent distribution treated by the fitted perturbation G3.

5.3.3 Perturbation Attributions of Cell States for Atomic Scores

We utilized the model interpretability method of integrated gradients to interpret how a perturbation shapes a cell state. We performed k-means clustering on the latent values of VAE trained on the LINCS-Drug data. We calculated the average silhouette widths for a range of cluster numbers, and identified 20 as the optimal number of clusters (Figure 5.9a-b). We trained a neural network model to classify

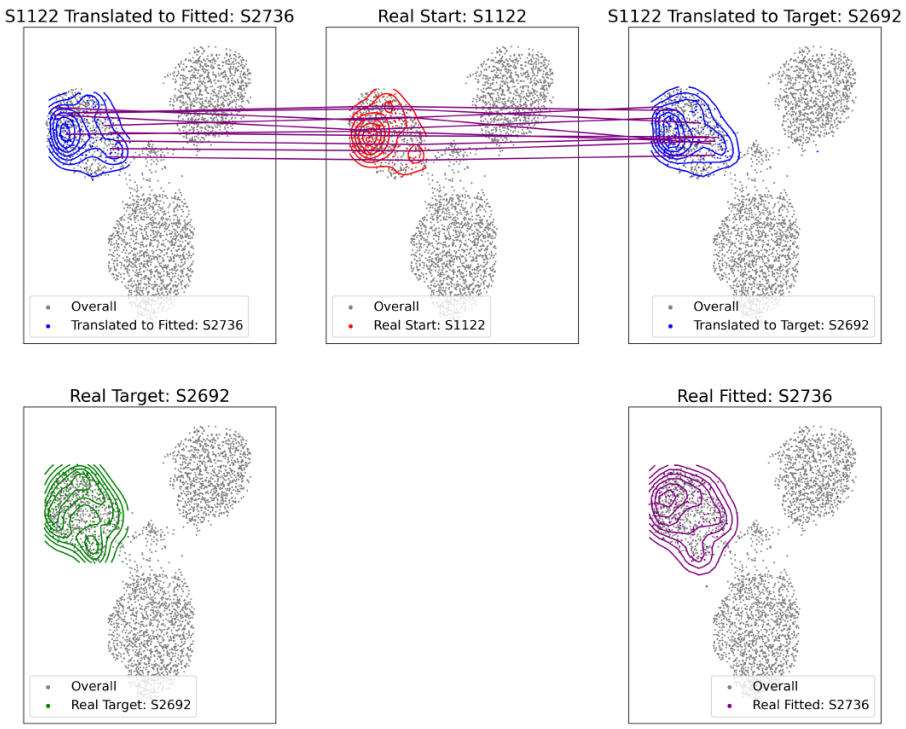

Figure 5.7: UMAP plots of latent values of K562 cells treated by the starting perturbation S1122 with dose 100, target perturbation S2692 with dose 100, fitted perturbation S2736 with dose 100, as well as translated latent values from K562 cells treated by S1122 with dose 100 to S2736 and S2692.

latent values into each of these 20 clusters. We then interpreted the features of a chemical perturbation in contributing to the probability of generating cells in a particular cluster.

We utilized an observed drug treatment G1 in the LINCS-Drug dataset and also selected a random observed drug treatment G0. We employed both G1 and G0 with the same sampled $\boldsymbol{V}$ from the standard normal prior distribution to translate to cellular representations, which we further treated as inputs to the latent classifier model to obtain their probabilities of being classified to a cluster. We used the integrated gradients method to interpret the attributions of features of G1 to contribute to the probability of generating cells in a latent cluster based on the baseline input G0. Figure 5.10 summarizes the model interpretation procedure on a perturbation to impact

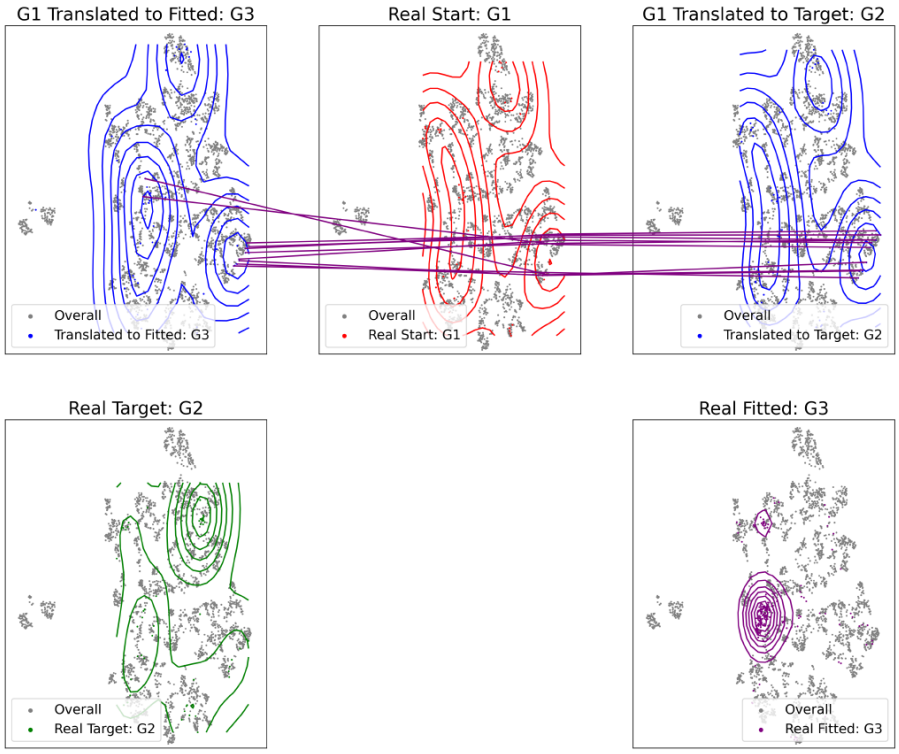

Figure 5.8: UMAP plots of latent values of cells treated by the starting perturbation G1, target perturbation G2, fitted perturbation G3, as well as translated latent values from cells treated by G1 to G3 and G2.

the formation of a latent cluster. We replicated this model interpretation procedure 3000 times with a fixed G1 and a random G0 in each replication, and then averaged the attributions of G1 across the replications.

After we obtained the average feature attributions of G1 for each cluster, we utilized the SimilarityMaps package (*Riniker and Landrum*, 2013) to visualize the molecular structure of G1 with its atoms colored by their attributions. We provide details of implementing SimilarityMaps for plotting molecular structures in Supplementary Materials Section 5.5.1. We utilized four clusters of the LINCS-Drug data (Figure 5.9c) with test-set accuracy values of {99.78%, 99.86%, 99.93%, 99.97%} from the classification model. Figure 5.9d shows the annotated molecular structures of G1 for the four latent clusters. The atomic attributions show atoms that increase the probability of generating a particular cell state cluster (green) or decrease the

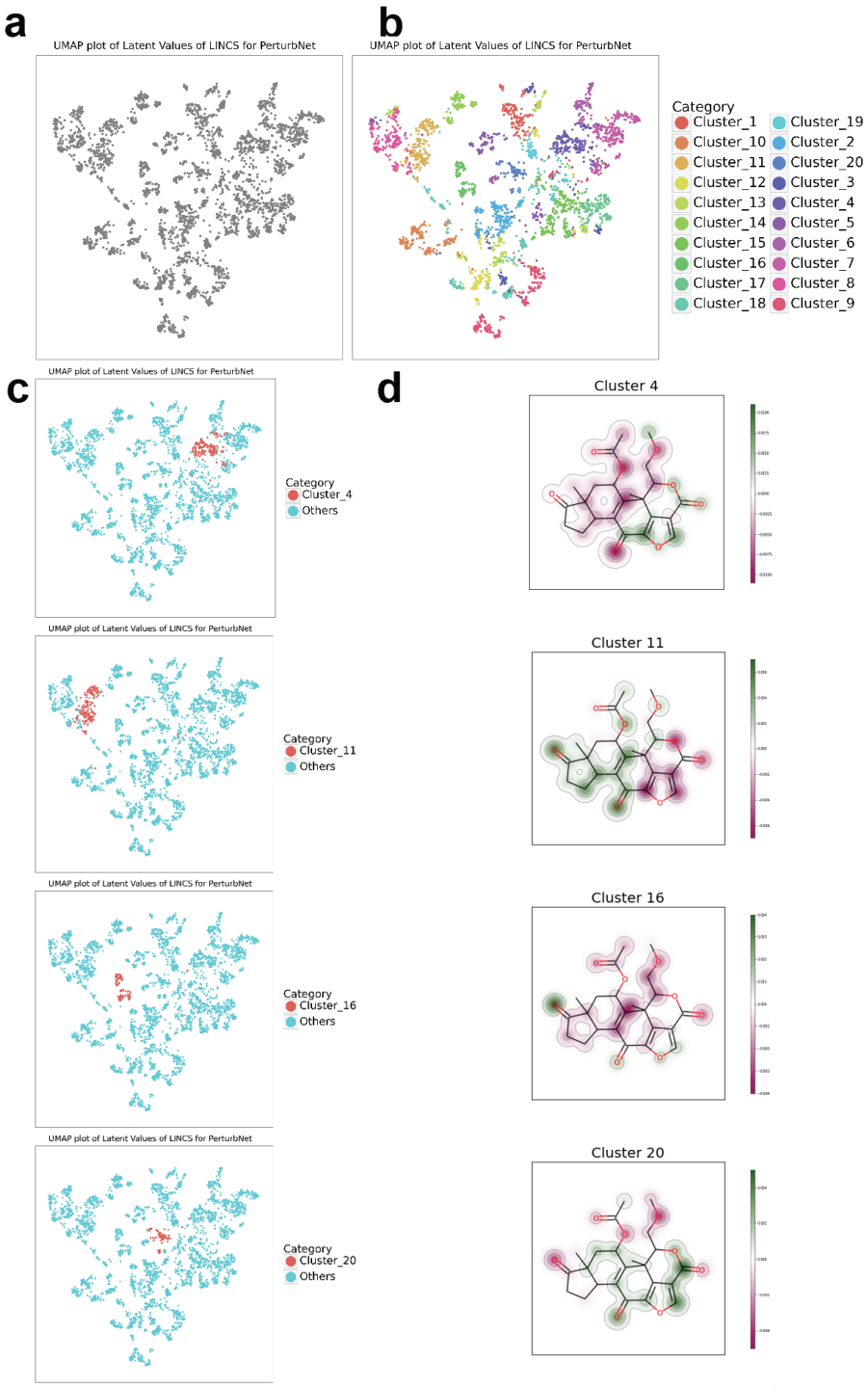

Figure 5.9: Model interpretation of the chemical perturbation G1 for latent clustering of the LINCS-Drug data. **a** UMAP plot of latent values. **b** UMAP plot of latent values by cluster label assigned by k-means clustering with $k = 20$. **c** UMAP plots of latent clusters 4, 11, 16 and 20. **d** Molecular structures of G1 colored by atomic attributions to the formations of latent clusters 4, 11, 16 and 20.

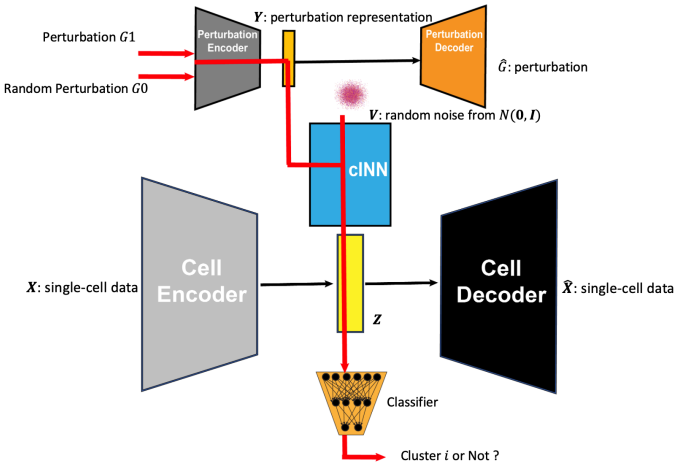

Figure 5.10: Overview of Interpreting Perturbations for Latent Clustering.

probability (red). The atoms of G1 have varying contributions to the probabilities of generating cells in different clusters.

5.3.4 Perturbation Attributions of Cell States for Gene Ontology Scores

We also performed a similar analysis with genetic perturbations. We utilized the integrated gradients method to determine the gene ontology terms that contribute to the probability of generating a particular cell state cluster. As with the drug data, we performed k-means clustering on the latent values of VAE trained on the LINCS-Gene data, calculated the average silhouette widths for a range of cluster numbers, and identified 20 as the optimal number of clusters (Figure 5.11a-b). We also trained a neural network model to classify latent states into each of these clusters. Following the model interpretation procedure in Figure 5.10, we determined the features of a genetic perturbation that increased or decreased the probability of generating latent cell states classified as a particular cluster.

We selected an observed genetic perturbation (knockdown of the gene 'ERG') and a random observed genetic perturbation as input and baseline. We also replicated the interpretation 3000 times and averaged the attributions across these replications. We then mapped the attributions to the specific GO terms with which ERG is annotated.

We show the UMAP plots of two latent clusters 9 and 17 whose test-set accuracy values are {99.91%, 99.89%} from the classification model (Figure 5.11c), and the GO terms with the 10 highest attributions for the two clusters (Figure 5.11d). The two annotations for DNA binding ('GO:0001228' and 'GO:0003677') of the ERG gene have the highest attributions to generate latent values in cluster 9 and 17. These two annotations are present in 5.1% and 6.2% of the baseline genetic perturbations, respectively.

We further visualized the attributions of the GO annotations of ERG in biological process, molecular function and cellular component following a similar procedure to the GO enrichment analysis conducted in *Lu and Welch* (2022). We performed multidimensional scaling of the GO terms and plot GO terms as circles with both color and size indicating the attribution values (Figure 5.12). Some GO terms are related to the transcription factor activity of ERG, as shown by the DNA-binding annotations with high attributions in Figure 5.11d.

We also evaluated the attributions of GO terms of ERG for the latent clusters 1 and 5 whose test-set accuracy values are {99.96%, 99.97%} from the classification model in Supplementary Figure 5.18a. The protein serine kinase/threonine activity annotation of 'GO:0004674' is not an annotation of ERG but is present in 8.9% of the baseline perturbations. This annotation gives a negative contribution to generate latent values in cluster 1. The DNA-binding transcription activator activity annotation of 'GO:0001228' is only in 5.1% of the baseline perturbations and is the most important annotation for ERG to the formation of cluster 5 (Supplementary Figure 5.18b). The plots of GO terms also reflect the transcription factor activity of ERG for cluster 1 and 5 (Figure 5.18c).

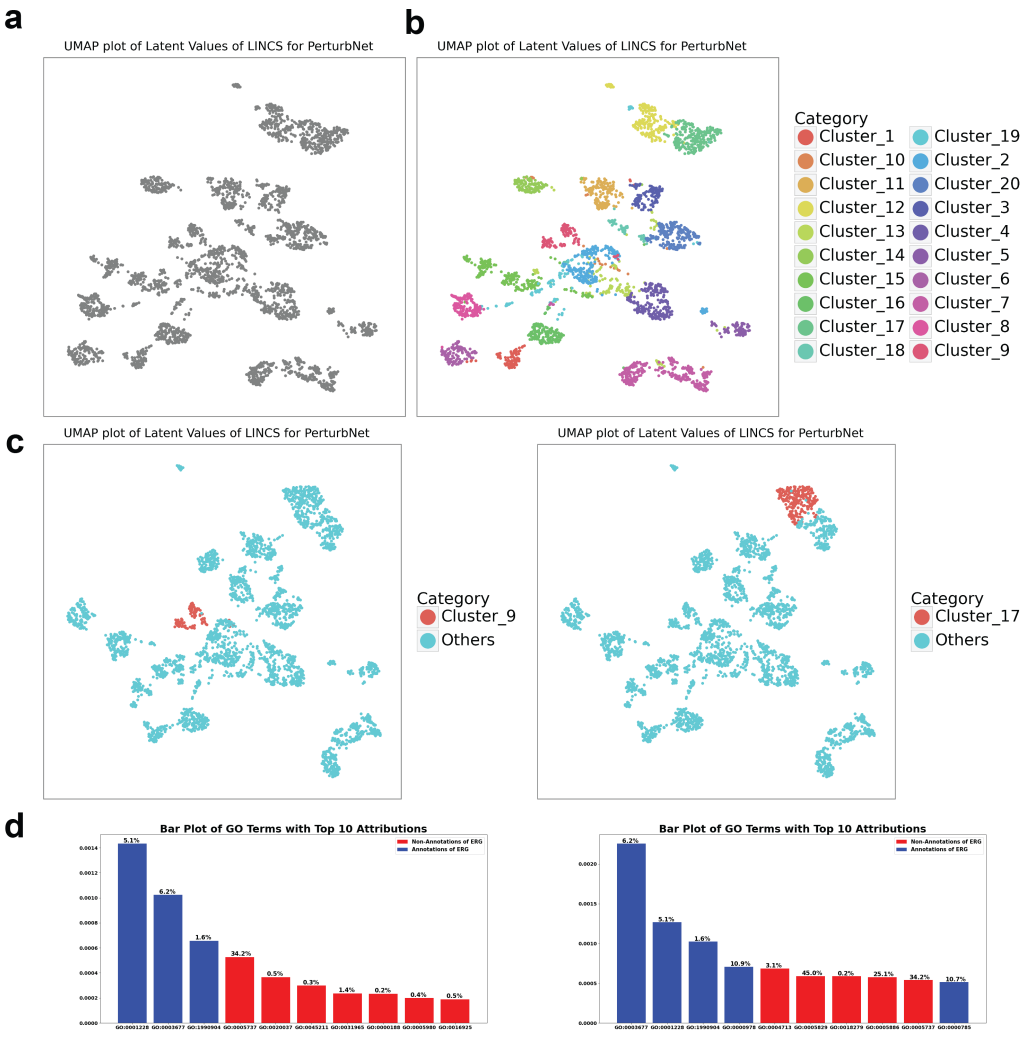

Figure 5.11: Model interpretation of the genetic perturbation 'ERG' for latent clustering of the LINCS-Gene for clusters 9 and 17. **a** UMAP plot of latent values. **b** UMAP plot of latent values by cluster label assigned by k-means clustering with $k = 20$. **c** UMAP plots of latent clusters 9 and 17. **d** Bar plots of the 10 highest attributions of GO annotations colored by being in ERG or not, with percentages in baseline perturbations for clusters 9 and 17.

5.3.5 Perturbation Attributions for Optimal Translations

We used the integrated gradients method to interpret the discrete optimal translations we performed in Section 5.3.2 above. We calculated the attributions of fitted and target perturbations, compared to a baseline perturbation. We selected three scenarios of discrete optimal translations of sci-Plex with different fitted and target

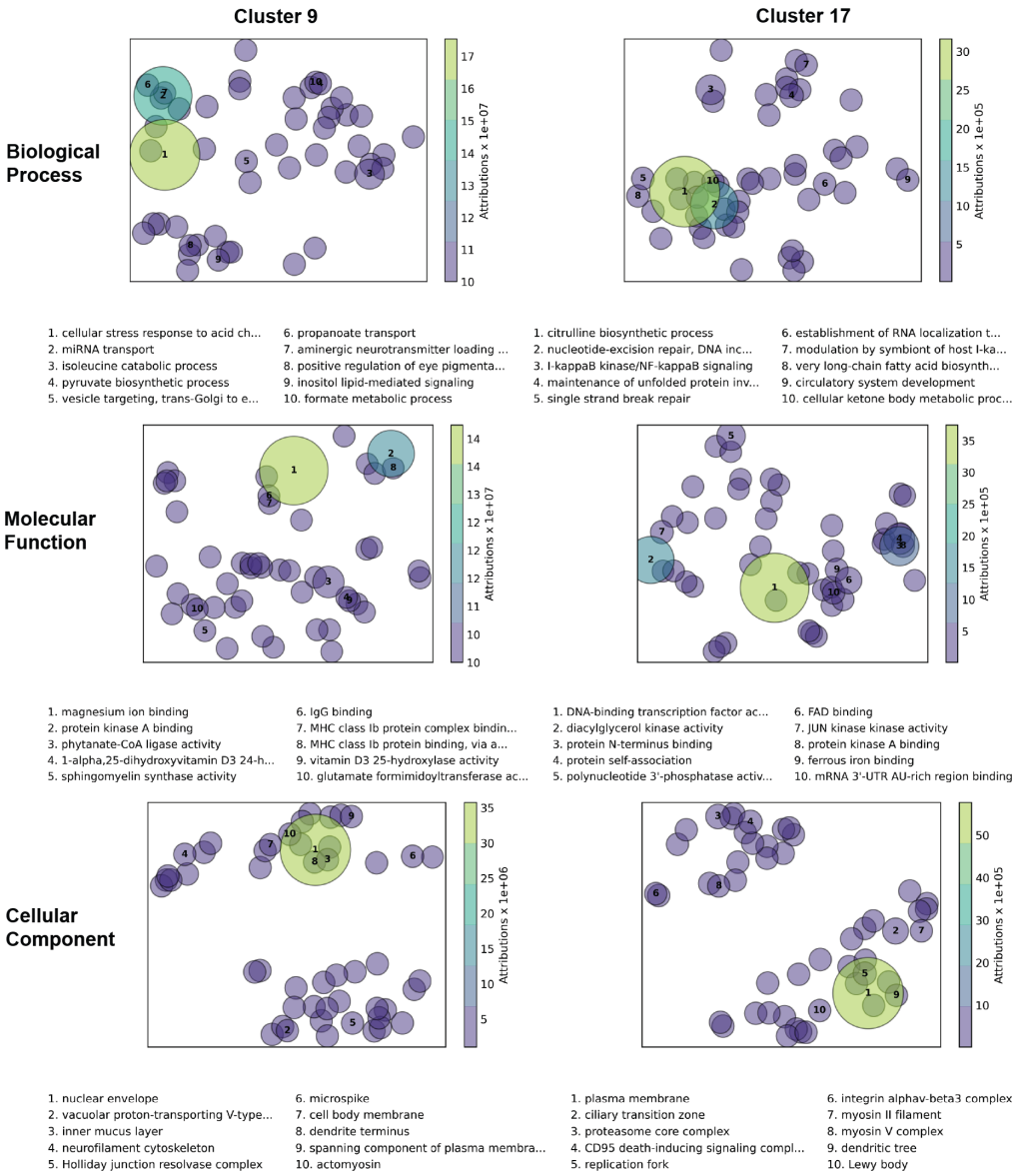

Figure 5.12: Plots of GO terms from the attributions of the genetic perturbation with target gene ERG for forming latent clusters 9 and 17, showing biological process, molecular function and cellular component.

perturbations in Table 5.1. For each selected optimal translation, we have a starting latent space with cells of a certain cell type treated by a starting perturbation at a certain dose, as well as a target latent space with cells of the same cell type and dose but a different perturbation. The fitted perturbation is the optimal perturbation obtained in discrete optimal translations, and Figure 5.14a shows the normalized fitted

W2 and normalized target W2 of the three selected scenarios of the sci-Plex. We trained a neural network model on the latent values in the starting and target latent spaces to classify cell state to the target latent space with test-set accuracy values being {56.76%, 69.23%, 98.81%}. We then interpreted a perturbation to translate to the target cell state using the $\boldsymbol{V}$'s from the starting latent space. Figure 5.13 shows the model interpretation procedure for discrete optimal translations. This model interpretation procedure evaluates the performance of a perturbation to translate the cells in the starting latent space to approximate the target latent space in a translation. We interpreted the perturbation features of the one-hot matrix $\boldsymbol{B}$ and of the perturbation representation $\boldsymbol{Y}$.

Table 5.1: Selected Scenarios of Discrete Optimal Translations of the sci-Plex and LINCS-Drug data.

Dataset	Notation	Start	Cell Type	Dose	Target	Fitted	Accuracy
sci-Plex	S1	S1180	A549	10	S2219	S7634	56.76%
	S2	S1122	K562	100	S2692	S2736	69.23%
	S3	S1515	MCF7	10000	S7605	2806	98.81%
LINCS-Drug	Scenario	G1	—	—	G2	G3	86.36%

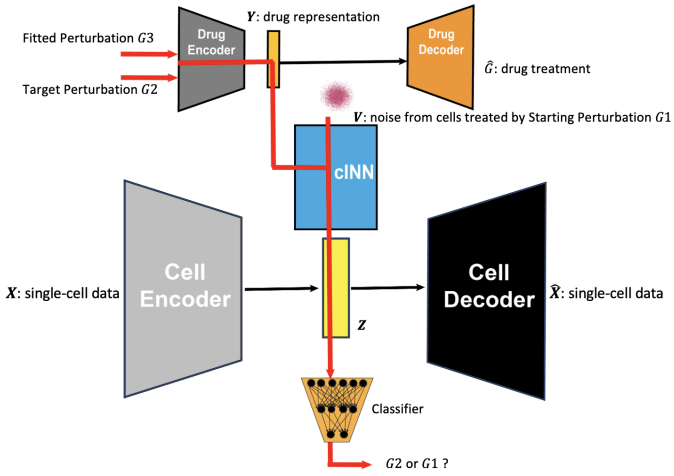

Figure 5.13: Overview of Interpreting Perturbations for Optimal Translations.

For each of the three selected sci-Plex scenarios {S1, S2, S3}, we interpreted the input and baseline pairs of {Target, Start}, {Fitted, Start} and {Fitted, Target}.

We obtained the average feature attributions of one-hot matrix $\boldsymbol{B}$ or perturbation representation $\boldsymbol{Y}$ of fitted and target perturbations. Figure 5.14b shows molecular structures of fitted and target perturbations colored by their atomic attributions based on the starting perturbation, along with their structures colored by fitted-target atomic attributions.

For each scenario, the fitted perturbation has some similar components as the target perturbation, and they possess different atomic scores to translate the starting cell state to the target cell state. Both S2 and S3 show that the fitted and target perturbations have some atoms with positive attribution scores based on the starting perturbations and have their atomic scores attenuated or diminished when compared with each other. This phenomenon reflects the fact that the target and fitted perturbation have similar performances in shrinking the original latent distances for S2 and S3 in Figure 5.14a. The attenuation of atomic scores of fitted and target perturbations of S1 is less obvious, as its fitted perturbation outperforms the target perturbation in shrinking the latent distance. Figure 5.14c shows the interpretation of the feature attributions for optimal translations on perturbation representations $\boldsymbol{Y}$. In each scenario, the target and fitted perturbations have distinctive attributions of perturbation representations based on the starting perturbation. However, the attributions of the fitted perturbation compared with the target perturbation in each scenario show that each dimension diminishes in magnitude, especially for S3, whose translations using the fitted and target perturbation both effectively approximate the target latent space. The attributions comparing the fitted and target perturbation of S2 have also been slightly shrunk, while those of S1 do not change significantly.

We also interpreted one scenario of the discrete optimal translations for the LINCS-Drug data, as shown in Table 5.1. The classification model gave the test-set accuracy of 86.36%. The fitted perturbation G3 has a slightly better performance in shrinking the original latent distance than the target perturbation G2 (Figure

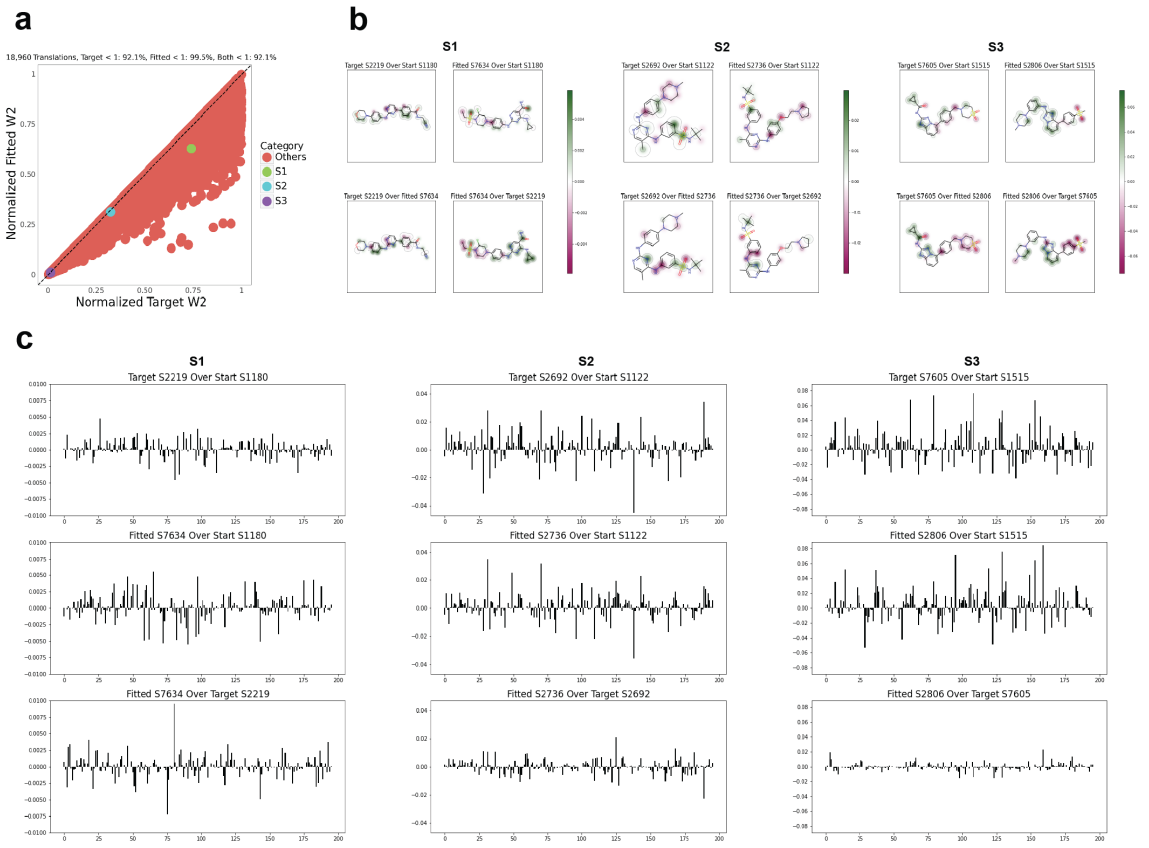

Figure 5.14: Model interpretation for three discrete optimal translations of the sci-Plex. **a** Scatter plot of normalized fitted W2 and normalized target W2 for 18,960 discrete optimal translations and three selected scenarios. **b** Molecular structures of fitted and target perturbations colored by atomic attributions to translate the starting latent space to the target latent space for each of the three scenarios. **c** Bar plots of attributions of perturbation representations of fitted and target perturbations to translate the starting latent space to the target latent space for each of the three scenarios.

5.15a). Figure 5.15b shows that when G1 serves as the baseline perturbation, the fitted perturbation G3 has overall higher attributions of perturbation representation than target perturbation G2. It also shows that G2 attenuates the attributions of G3. The molecular structure of G3 has more atoms than G2 to show positive attributions to translate the starting latent space to the target latent space, with G1 as the baseline perturbation (Figure 5.15c).

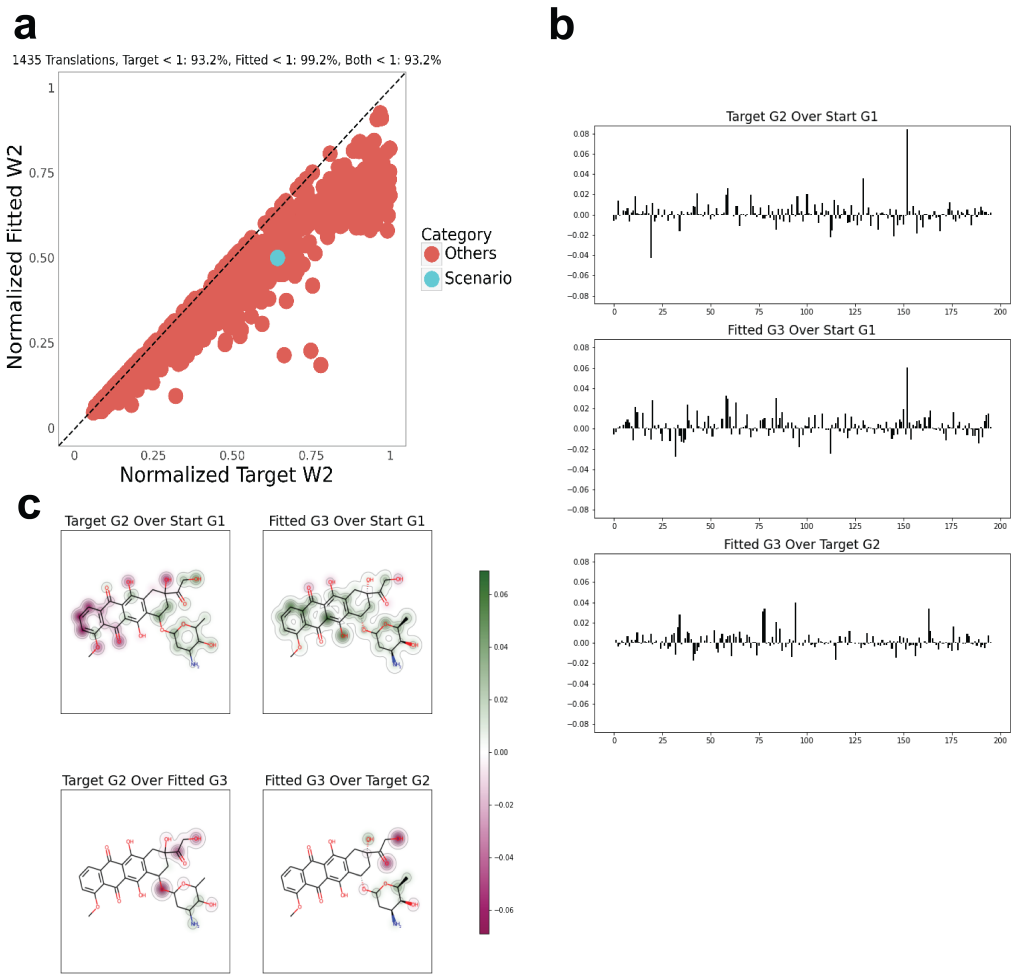

Figure 5.15: Model interpretation for a discrete optimal translation of the LINCS-Drug. **a** Scatter plot of normalized fitted W2 and normalized target W2 for 1435 discrete optimal translations and the selected scenario. **b** Bar plots of attributions of perturbation representations of fitted and target perturbations to translate the starting latent space to the target latent space for the scenario. **c** Molecular structures of fitted and target perturbations colored by atomic attributions to translate the starting latent space to the target latent space for the scenario.

5.3.6 Perturbation Attributions of Genetic Perturbations for Shifting Cell State Distributions

We also interpreted the attributions of the features of a genetic perturbation to forming its cell state distribution, compared to the cell state distribution of another perturbation. We selected three pairs of genetic perturbations including {(ERG,

ERBB3), (ERBB3, KRAS), (KRAS, ERG)} in Figure 5.17a. We denoted each pair as (G0, G1), and trained a neural network model on the latent values of cells treated by G0 or G1 to classify their cell state to latent space of G1. We utilized G1 and G0 to translate the cells treated by G0 to new cell state, and then obtained probabilities of being classified as G1 latent space. Figure 5.16 summarizes the model interpretation procedure of pairs of genetic perturbations for shifting cell state distributions. The classification models for the three pairs gave the test-set accuracy values of {92.51%, 62.81%, 93.7%}.

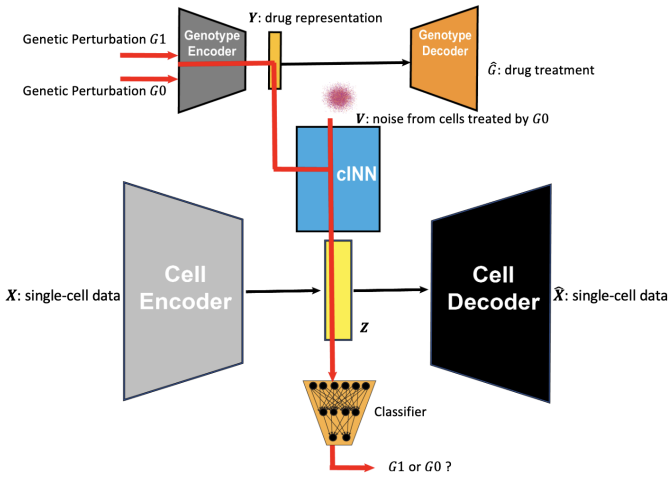

Figure 5.16: Overview of Interpreting Perturbations for Shifting Cell State Distributions.

We obtained the feature attributions of GO annotation vector of G1 compared to that of G0 in shifting the cells treated by G0 to cells treated by G1. The common genetic annotations between G1 and G0 do not provide different feature input and thus have attributions of 0's. Figure 5.17b shows the uncommon annotations between G1 and G0 with the 10 highest attributions. Not having ERG annotations of nucleus ('GO:0005634'), protein phosphorylation ('GO:0006468'), DNA-binding transcription factor activity ('GO:0000981') and regulation of transcription by RNA polymerase II ('GO:0006357') attributes the most for ERBB3 to shift the original ERG cell state to the ERBB3 cell state. Not having transmembrane receptor protein tyrosine kinase

signaling pathway annotation ('GO:0007169') and having protein phosphorylation annotation ('GO:0006468') for ERG gives the highest attributions to change the cell states of ERBB3 and KRAS to those of KRAS and ERG, respectively.

Figure 5.17c shows the GO terms using the attributions in shifting a cell latent space to another cell latent space. Several GO terms give meaningful interpretations of shifting the cell states. For example, the GO term for ERBB2 signaling pathway in biological process has a high attribution to shift ERG cells to ERBB3. In addition, the 'ERBB3:ERBB2' annotation in cellular component has a strong signal of distinguishing two genetic perturbations in the pairs of (ERBB3, ERG) and (KRAS, ERBB3).

5.4 Discussion

In this chapter, we consider designing perturbations from predicted single-cell responses using PerturbNet, and also discovering key components within a chemical or genetic perturbation to the formation of a cell state. We propose two algorithms: continuous optimal translation of perturbation representation, and discrete optimal translation to search for the optimal perturbation to translate a starting cell state to a desired target cell state. We also employ the integrated gradients method to interpret the feature attributions of a chemical or genetic perturbation in increasing the probability of generating a specific cell state, with its molecular structures colored by atomic scores or its plot of GO scores. We also interpret the optimal translation experiments, and the attributions of a genetic perturbation to shift another perturbation's cell state.

Our proposed optimal translation algorithms aim to design an optimal perturbation for individual cells, and can be further utilized to perform research in specific biomedical scenarios. For example, the optimal translation algorithms can discover a suitable drug treatment to change some diseased cells to approximate a healthy cell

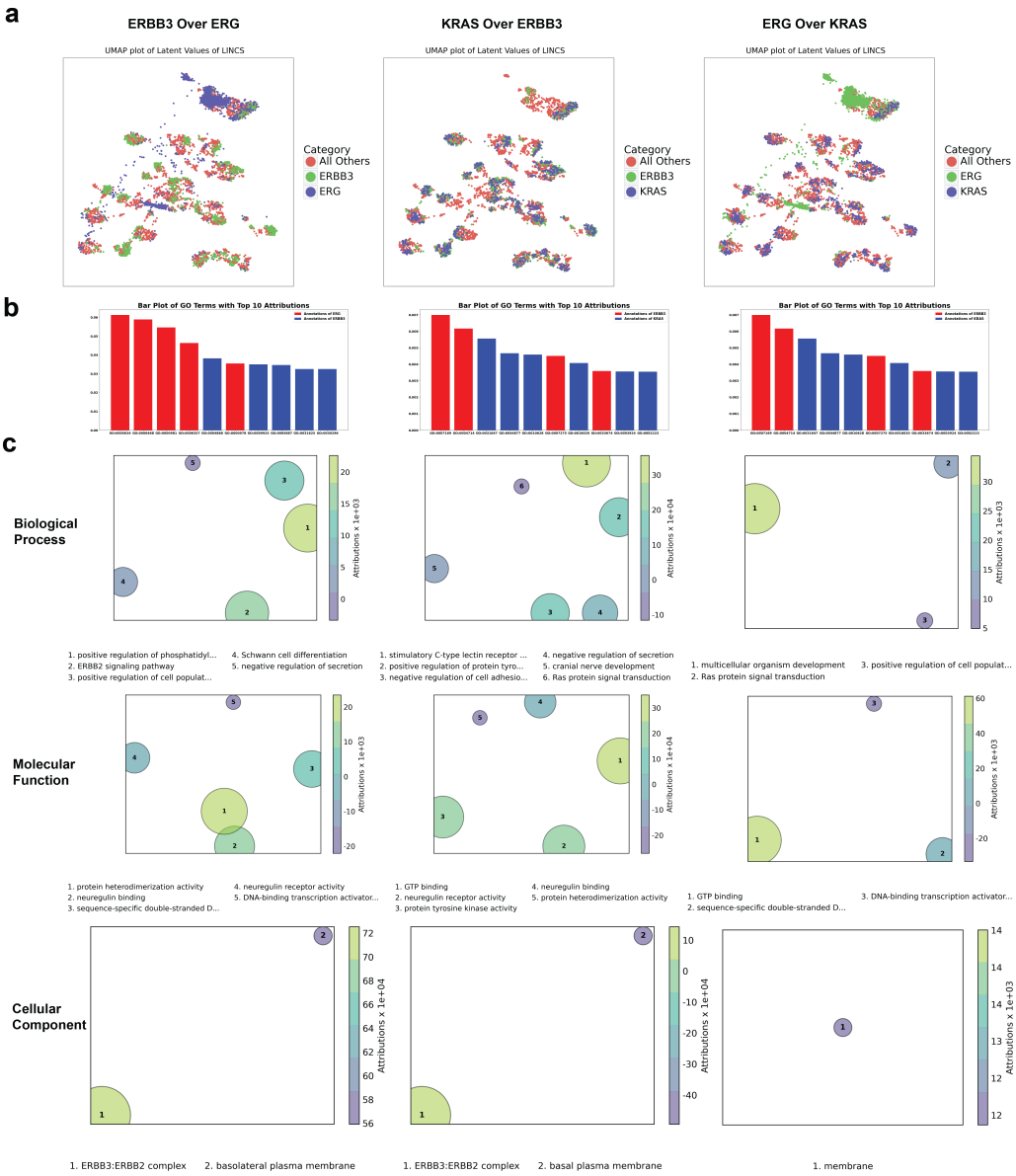

Figure 5.17: Model interpretation of pairs of genetic perturbations for shifting cell state distributions. **a** UMAP plot of latent values of cells treated by three pairs of genetic perturbations. **b** Bar plots of the 10 highest attributions of GO annotations colored by being in the input perturbation or not, for the three pairs of perturbations. **c** Plots of GO terms from the attributions of a genetic perturbation for shifting the cell state of a baseline perturbation to its cell state for the three pairs of perturbations, showing biological process, molecular function and cellular component.

state. In addition to drug treatments, the optimal translation algorithms can also be utilized to find an optimal genetic perturbation to change genetic functions.

Our experiments perform discrete optimal translations with limited numbers of drugs due to the computational intensity to search through the discrete drug set and to demonstrate the properties of optimal perturbation. In practice with a well-defined perturbation discovery goal, the discrete optimal translations can be implemented in large chemical databases such as PubChem (*Kim et al.*, 2016). In contrast, the continuous optimal translation is more computationally efficient. However, the optimal perturbation representation obtained from the continuous optimal translation needs to be further engineered to design an optimal chemical or genetic perturbation. For example, a possible direction for chemical perturbations is to utilize the ChemicalVAE to generate drug treatments from the perturbation representation (*Gómez-Bombarelli et al.*, 2018). This is an exciting direction for future work.

A possible limitation of our experiments lies in the fact that we do not have the counterfactual responses for validation. We find the latent values of starting cells translated by fitted perturbation approximate the cells of the target perturbation, but can differ from the real fitted cells. This is due to the difference of individual cellular residual representation that is invariant from the condition variable between the real starting cells and real fitted cells. In future work, we hope to design experiments, such as Perturb-seq, to validate the predictions of our designed perturbations.

5.5 Supplementary Materials

5.5.1 Atomic Attributions Visualizations

We employ the SimilarityMaps package (*Riniker and Landrum*, 2013) to draw molecules from canonical SMILES strings. We extract the atomic attributions in the one-hot perturbation vector using the edited smiview Python package and use them

as atomic weights for similarity maps.

5.5.2 Classification Models

The neural network classification models use multilayer perceptron (MLP) units and have a fully-connected (FC) hidden layers with 32 neurons with the Rectified Linear Unit (ReLU) activation, batch normalization, dropout regularization with a dropout probability of 0.1. The output layer has one neuron with the sigmoid activation. We train the classification models in Sections 5.3.3 and 5.3.4 for latent clustering with batch Adam optimization, and train the classification models in Sections 5.3.5 and 5.3.6 for pairs of perturbations with minibatch Adam optimization with a batch size of 128. We fixed the training of these classification models with a learning rate of 10^{-3}, batch size of 128, training size of 0.8. We utilized different numbers of epochs ranging from 30 to 250 for each classification model to achieve its convergence of loss curves.

5.5.3 Supplementary Figures

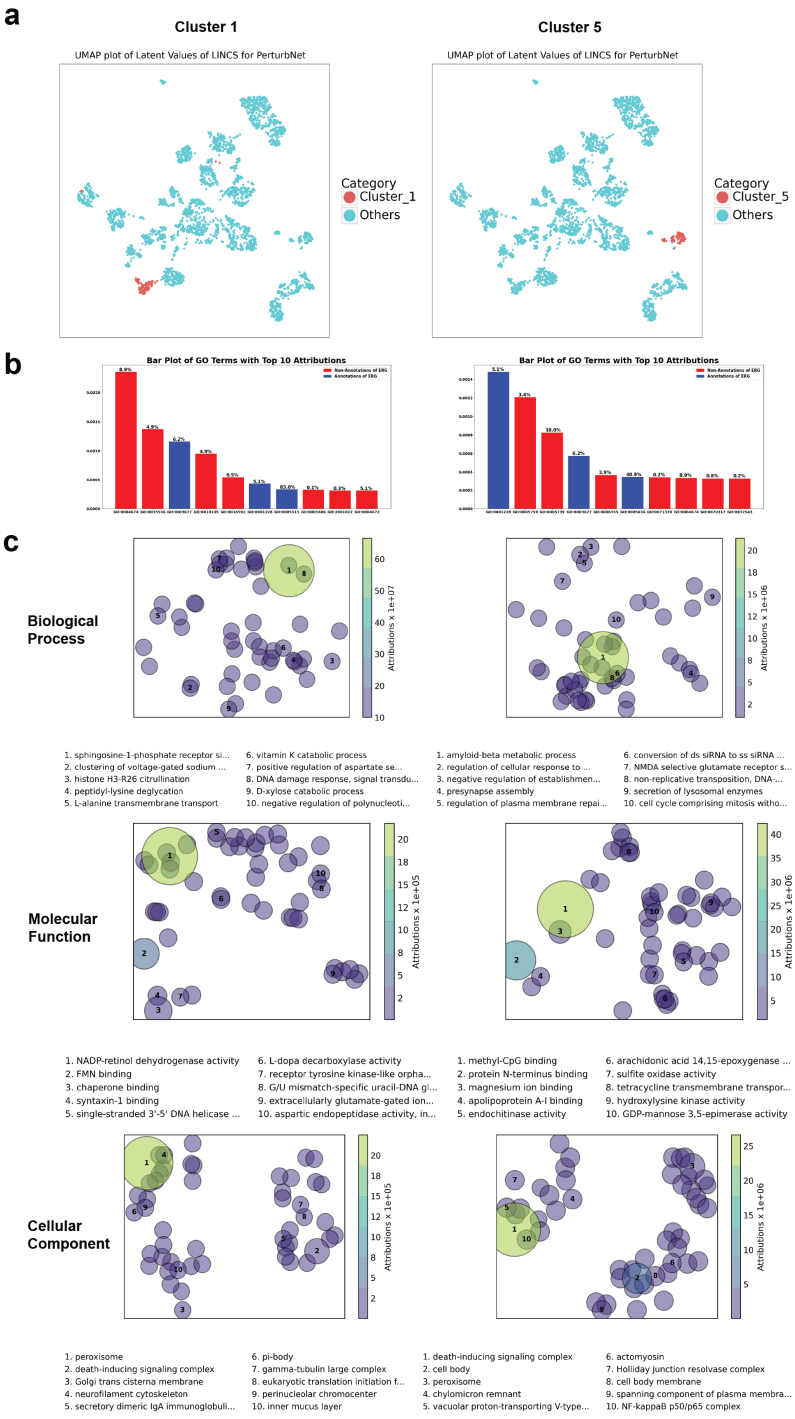

Figure 5.18: Model interpretation of the genetic perturbation ERG for latent clustering of LINCS-Gene for clusters 1 and 5. **a** UMAP plots of latent clusters 1 and 5. **b** Bar plots of the 10 highest attributions of GO annotations colored by being in ERG or not, with percentages in baseline perturbations for clusters 1 and 5. **c** Plots of GO terms from the attributions of the genetic perturbation with target gene ERG for generating latent clusters 1 and 5, showing biological process, molecular function and cellular component.

CHAPTER VI

Summary and Future Work

6.1 Summary

The first project in Chapter II is motivated by state-of-the-art disentangled representation learning techniques (*Chen et al.*, 2016). A disentangled representation captures semantic factors of variation in separate dimensions (*Higgins et al.*, 2018), and can potentially uncover factors of cellular identity in single-cell data. Most disentangled representation learning techniques strive to attain high disentanglement without losing the generation performance through either variational autoencoders (VAEs) methods or generative adversarial networks (GANs) methods (*Chen et al.*, 2018; *Karras et al.*, 2019), but rarely via an integrative framework. The trade-off between disentanglement and generation performance of VAE-based and GAN-based deep generative models motivate an integrative framework to learn disentangled representations and achieve high-quality data generation. Our MichiGAN framework leverages VAE-based and GAN-based deep generating models and provides better flexibility of deep learning methods on the scRNA-seq data. For scRNA-seq data with unknown underlying factors of variation, the MichiGAN method efficiently extracts the disentangled representations through a VAE-based method and subsequently improves the data generation quality using GANs. Our use of PCWGAN-GP, an improved Wasserstein GAN based on PCGAN (*Miyato and Koyama*, 2018), gives sub-

stantially improved data generation performance. The MichiGAN framework can preserve disentanglement performance of VAE-based methods, and is also applicable to various disentanglement methods on scRNA-seq data such as PCA, ICA and NMF without the constraint of an inference model (*Khemakhem et al.*, 2020; *Lee et al.*, 2020). Unlike the interpretable variables in images or natural languages, the ground-truth variables in real single-cell data captured by disentangled representations are usually not immediately known. We therefore perform latent traversals of fibroblast samples, and find several dimensions with semantically meaningful transitions. We also predict single-cell data of drug treatments using the latent space vector arithmetic algorithm. MichiGAN learns disentangled representations from the single-cell data and gives significantly higher data generation than its VAEs counterpart for unseen cell type/drug treatment combinations.

The projects in the previous chapters were motivated by the sci-Plex and Perturb-seq datasets with multiple chemical and genetic perturbations (*Srivatsan et al.*, 2020; *Dixit et al.*, 2016). As there are limited numbers of perturbations explored in such data, we wanted to develop deep generative models that predict single-cell responses to multiple perturbations and also predict unseen perturbations. These out-of-distribution predictions facilitate exploring distributions of single-cell responses to perturbations and designing optimal perturbations for cell states.

In Chapter III, we develop PerturbNet, a multi-stage deep generative model to predict single-cell responses to drug treatments. The PerturbNet framework consists of a three-stage modeling process with ChemicalVAE on drug treatments, a single-cell VAE model on single-cell samples, as well as a conditional invertible neural network (cINN) that connects perturbation representation and cellular representation. We show that PerturbNet effectively connects perturbation information and cell state. We demonstrate the excellent prediction performance of PerturbNet, with another proposed KNN model, to predict single-cell responses to either observed or unseen

perturbations. By regularizing through properties of cellular representation, we develop an algorithm to fine-tune the learned perturbation representation of ChemicalVAE, and to further enhance the performance of single-cell predictions for unseen perturbations. We also show the importance of adjusting for cell state covariates to better learn the condition-invariant or residual representations for individual cells and to more accurately generate individual cellular responses through the PerturbNet framework.

In Chapter IV, we extend PerturbNet to predict single-cell responses to genetic perturbations. We consider two types of genetic perturbations in CRIPSR gene-editing experiments. For the genetic perturbations with target gene identification, we develop a deep generative model called GenotypeVAE to encode the gene ontology (GO) annotation vector of target genes to dense representation, and then decode it to reconstructed vector. For the genetic perturbations with protein-coding sequence variants, we employ a pre-trained start-of-the-art protein transformer model (*Rives et al.*, 2021) to encode coding variants. We evaluate KNN and PerturbNet on several high-throughput genetic screen datasets, and show their high prediction performances. We also fine-tune GenotypeVAE to improve the prediction performance of PerturbNet.

In Chapter V, we focus on designing perturbations to the formation of desired cell states and discovering components of chemical and genetic perturbations that have important biological effects. The project was motivated by the problem of finding perturbations to shift some diseased cells to a healthy cell state. We formulate the objective to design an optimal perturbation to translate a group of cells to approximate a target cell state. We propose two optimal translation algorithms based on PerturbNet. We show that both continuous optimal translation and discrete optimal translation can find an optimal perturbation to translate a group of observed cells to approximate a group of target cells in their latent space. We also utilize a

model interpretability method to interpret atomic scores of a drug treatment and GO scores in forming a cell state. We also employ the model interpretability method to atomic scores of fitted optimal perturbations from the discrete optimal translation algorithm. In addition, we interpret and identify GO terms that have high attributions to distinguish a pair of genetic perturbations in their cell states.

6.2 Future Directions

There are several interesting directions for future research. One may consider our studies for a broader field of drug or intervention discovery. Both of our MichiGAN and PerturbNet frameworks predict single-cell responses by learning the perturbation effects on cellular representations. The MichiGAN framework explores the disentanglement of cellular representations with respect to the semantic cellular information including drug treatments. We show the applications of latent traversals and latent space vector arithmetic using the disentangled representations. Future research is needed to learn the semantic cell identity captured by each dimension of a disentangled representation. In addition, methods of boosting the fidelity of predicted out-of-distribution samples (*Berthelot et al.*, 2018) can be utilized to further improve the generation performance from disentangled representations.

The PerturbNet framework, on the other hand, connects the perturbation representations from perturbation encoders to cellular representations through normalizing flows. Future improvement can employ other state-of-the-art methods for chemical and genetic perturbations to obtain better perturbation representations. We can also consider training these frameworks on larger chemical databases such as PubChem (*Kim et al.*, 2016) or larger GO annotation sets by incorporating genetic perturbations with more than two target genes as well. From our experiments, we find that the prediction performance of PerturbNet on cellular responses to unseen perturbations is likely to be impacted by the number of observed perturbations for training

the cINN. The LINCS dataset has large numbers of chemical (LINCS-Drug) and genetic (LINCS-Gene) perturbations to train the cINN translations, and generally have better performance in unseen chemical or genetic perturbations. To improve the cINN translations for single-cell data with a small number of observed perturbations, one may consider transfer learning (*Lotfollahi et al.*, 2022) to utilize a cINN model trained on a dataset with a large number of perturbations such as LINCS-Drug and LINCS-Gene. The transfer learning between the two datasets might need to integrate their cellular representations in a comprehensive latent space before employing the pre-trained cINN model. More future research is needed for this direction.

A possible future direction can combine the disentanglement of deep generative models studied in Chapter II and the PerturbNet framework. In addition to obtaining disentangled perturbation and cellular representations, one can also consider improving the disentanglement performance of the residual representation in the cINN translation. The improved disentanglement performance of these condition-invariant representations further captures the cellular identity in its separate dimensions for individual cells. The disentangled residual representations might uncover cell states to a perturbation with semantically meaningful interpretations. In addition, we can integrate MichiGAN in Chapter II with PerturbNet by having an extra stage to train a conditional GAN-based model to replace the single-cell VAE model. The extra stage possibly further improves data generation quality from cellular representations learned by the single-cell VAE model.

Another direction related to PerturbNet is to learn the relationship between cell state covariates and a perturbation. Although we find that adjusting for the covariates improves the prediction performance of PerturbNet for individual cells, it is possible that the covariates' values of an unseen perturbation are not immediately known in practice. A two-stage process might be performed to model the cellular response distribution to a perturbation. The first stage learns the covariates' values

from the perturbation and the second stage can take the perturbation and predicted covariates to generate cellular representations. In addition, other methods of causal representation learning (*Schölkopf et al.*, 2021) can also be utilized to address the covariates' relationship with a perturbation such as instrument variables (*Hartford et al.*, 2017).

Both the MichiGAN and PerturbNet frameworks presented in this dissertation focus on the prediction accuracy for single-cell perturbation responses, and an additional fascinating direction is to quantify the uncertainty of predicted single-cell perturbation responses. As both frameworks utilize variational inference to obtain cellular representations, the posterior probabilities of cellular representations can therefore be employed for hypothesis testing. For example, the disentangled representations learned from MichiGAN can be used to test if cells possess certain values for factors of variation, and the cellular representations of PerturbNet might give the quantified uncertainty for each cell under a perturbation. In addition, researchers can integrate the inference models of the perturbation encoder, cINN and cellular VAE in PerturbNet to derive the posterior distribution of single-cell response to each perturbation. This measures uncertainty for single-cell data, and can be utilized to perform hypothesis testing and downstream analyses.

The dissertation focuses on modeling single-cell data in perturbation experiments with each cell measured after a chemical or genetic perturbation. A future direction is to predict the trajectories of single-cell responses after a sequence of perturbations (*Bergen et al.*, 2020). The PerturbNet might be improved to sequentially model new cellular representation on the new perturbation representation and the previous cellular representation. This direction also relates to reinforcement learning to find the optimal perturbations at each stage within a dynamic decision process (*Sutton and Barto*, 2018). As the reinforcement learning process usually requires the ability to observe the effects of actions and the single-cell measurement usually only provides

one static snapshot of a cell, future research is needed.

6.3 Closing Remarks and Perspectives

Research in deep generative models is fast-moving, and new methods have enabled many applications in computer vision and natural language processing. Over the last decade, various deep generative models have been developed to unravel and understand different aspects of single-cell data such as cell identity, cell reprogramming and perturbation responses. In this dissertation, we evaluate existing methods and also construct novel frameworks for single-cell data. A principal modeling motivation of our frameworks is to integrate distinct paradigms of deep generative models. As with the classical no free lunch theorem in learning theory (*Wolpert*, 1996; *Wolpert and Macready*, 1997), which indicates that different optimization algorithms are equivalent across evaluations over all possible problems, we also find that different deep generative models have varying strengths and none of them is able to outperform others in every aspect. Therefore, an integrative framework can inherit the strength of each algorithm in specific aspects of a problem. For example, MichiGAN integrates VAEs and GANs for their complementary strengths in disentanglement and generation. We expect our frameworks to also achieve remarkable performances in vision and natural languages. Integrative deep generative models are interesting and active research topics with promising applications in molecular biology, vision and natural languages.

We employ various deep generative models including VAEs, GANs and normalizing flows. We find each paradigm has its own advantages in modeling single-cell data. VAEs can learn disentangled representations, GANs generate highly realistic data samples and normalizing flows give stable domain-to-domain translations. Future developments of deep generative models for single-cell data can further employ these advantages and explore myriad applications in molecular biology.

Finally, we expect more studies on causal inference for modeling single-cell perturbations responses. The dissertation explores disentangled representation, counterfactual responses and deep learning model interpretability for single-cell data. Numerous applications using causal representation learning techniques are forthcoming to infer complex cellular components and dynamics under perturbations.

BIBLIOGRAPHY

BIBLIOGRAPHY

Achille, A., and S. Soatto (2018), Emergence of invariance and disentanglement in deep representations, *The Journal of Machine Learning Research*, *19*(1), 1947–1980.

Adamson, B., et al. (2016), A multiplexed single-cell crispr screening platform enables systematic dissection of the unfolded protein response, *Cell*, *167*(7), 1867–1882.

Alaa, A. M., and M. van der Schaar (2017), Bayesian inference of individualized treatment effects using multi-task gaussian processes, *Advances in Neural Information Processing Systems*, *30*.

Alberts, B., A. Johnson, J. Lewis, M. Raff, K. Roberts, and P. Walter (2002), *Molecular Biology of the Cell*, Garland Science, New York.

Altschuler, S. J., and L. F. Wu (2010), Cellular heterogeneity: do differences make a difference?, *Cell*, *141*(4), 559–563.

Amann, R. I., L. Krumholz, and D. A. Stahl (1990), Fluorescent-oligonucleotide probing of whole cells for determinative, phylogenetic, and environmental studies in microbiology, *Journal of bacteriology*, *172*(2), 762–770.

Anonymous (2014), Method of the year 2013, *Nature Methods*, *11*(1), doi:10.1038/nmeth.2801.

Anonymous (2020), Method of the year 2019, *Nature Methods*, *17*(1), doi:10.1038/s41592-019-0703-5.

Ardizzone, L., C. Lüth, J. Kruse, C. Rother, and U. Köthe (2019), Guided image generation with conditional invertible neural networks, *arXiv preprint arXiv:1907.02392*.

Arjovsky, M., S. Chintala, and L. Bottou (2017), Wasserstein gan, *arXiv preprint arXiv:1701.07875*.

Asp, M., J. Bergenstråhle, and J. Lundeberg (2020), Spatially resolved transcriptomes—next generation tools for tissue exploration, *BioEssays*, *42*(10), 1900,221.

Aubry, M., D. Maturana, A. A. Efros, B. C. Russell, and J. Sivic (2014), Seeing 3d chairs: exemplar part-based 2d-3d alignment using a large dataset of cad models, in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3762–3769.

Baehrens, D., T. Schroeter, S. Harmeling, M. Kawanabe, K. Hansen, and K.-R. Müller (2010), How to explain individual classification decisions, *The Journal of Machine Learning Research*, *11*, 1803–1831.

Bai, Y., and L. L. Duan (2019), Tuning-free disentanglement via projection, *arXiv preprint arXiv:1906.11732*.

Baltrušaitis, T., C. Ahuja, and L.-P. Morency (2018), Multimodal machine learning: A survey and taxonomy, *IEEE transactions on pattern analysis and machine intelligence*, *41*(2), 423–443.

Banker, M. J., T. H. Clark, and J. A. Williams (2003), Development and validation of a 96-well equilibrium dialysis apparatus for measuring plasma protein binding, *Journal of pharmaceutical sciences*, *92*(5), 967–974.

Barratt, S., and R. Sharma (2018), A note on the inception score, *arXiv preprint arXiv:1801.01973*.

Bastidas-Ponce, A., et al. (2019), Comprehensive single cell mrna profiling reveals a detailed roadmap for pancreatic endocrinogenesis, *Development*, *146*(12).

Bengio, Y., A. Courville, and P. Vincent (2013), Representation learning: A review and new perspectives, *IEEE transactions on pattern analysis and machine intelligence*, *35*(8), 1798–1828.

Bergen, V., M. Lange, S. Peidli, F. A. Wolf, and F. J. Theis (2020), Generalizing rna velocity to transient cell states through dynamical modeling, *Nature biotechnology*, *38*(12), 1408–1414.

Berthelot, D., C. Raffel, A. Roy, and I. Goodfellow (2018), Understanding and improving interpolation in autoencoders via an adversarial regularizer, *arXiv preprint arXiv:1807.07543*.

Bialk, P., N. Rivera-Torres, B. Strouse, and E. B. Kmiec (2015), Regulation of gene editing activity directed by single-stranded oligonucleotides and crispr/cas9 systems, *PloS one*, *10*(6), e0129,308.

Bickle, M. (2010), The beautiful cell: high-content screening in drug discovery, *Analytical and bioanalytical chemistry*, *398*(1), 219–226.

Bishop, C. M. (2006), *Pattern recognition and machine learning*, springer.

Bowman, S. R., L. Vilnis, O. Vinyals, A. M. Dai, R. Jozefowicz, and S. Bengio (2015), Generating sentences from a continuous space, *arXiv preprint arXiv:1511.06349*.

Bray, M.-A., et al. (2016), Cell painting, a high-content image-based assay for morphological profiling using multiplexed fluorescent dyes, *Nature protocols*, *11*(9), 1757–1774.

Brock, A., J. Donahue, and K. Simonyan (2018), Large scale gan training for high fidelity natural image synthesis, *arXiv preprint arXiv:1809.11096*.

Brown, T., et al. (2020), Language models are few-shot learners, *Advances in neural information processing systems*, *33*, 1877–1901.

Brown, T. A. (2018), *Genomes 4*, Garland science.

Burgess, C. P., I. Higgins, A. Pal, L. Matthey, N. Watters, G. Desjardins, and A. Lerchner (2018), Understanding disentangling in \β-vae, *arXiv preprint arXiv:1804.03599*.

Burgess, D. J. (2019), Spatial transcriptomics coming of age, *Nature Reviews Genetics*, *20*(6), 317–317.

Burkhardt, D. B., et al. (2021), Quantifying the effect of experimental perturbations at single-cell resolution, *Nature biotechnology*, *39*(5), 619–629.

Cai, D., X. He, J. Han, and T. S. Huang (2010), Graph regularized nonnegative matrix factorization for data representation, *IEEE transactions on pattern analysis and machine intelligence*, *33*(8), 1548–1560.

Chandrasekaran, S. N., H. Ceulemans, J. D. Boyd, and A. E. Carpenter (2021), Image-based profiling for drug discovery: due for a machine-learning upgrade?, *Nature Reviews Drug Discovery*, *20*(2), 145–159.

Chen, K. H., A. N. Boettiger, J. R. Moffitt, S. Wang, and X. Zhuang (2015), Spatially resolved, highly multiplexed rna profiling in single cells, *Science*, *348*(6233), aaa6090.

Chen, T. Q., X. Li, R. B. Grosse, and D. K. Duvenaud (2018), Isolating sources of disentanglement in variational autoencoders, in *Advances in Neural Information Processing Systems*, pp. 2610–2620.

Chen, X., Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel (2016), Infogan: Interpretable representation learning by information maximizing generative adversarial nets, in *Advances in neural information processing systems*, pp. 2172–2180.

Cheng, A. W., et al. (2013), Multiplexed activation of endogenous genes by crispr-on, an rna-guided transcriptional activator system, *Cell research*, *23*(10), 1163–1171.

Cheng, F., I. A. Kovács, and A.-L. Barabási (2019), Network-based prediction of drug combinations, *Nature communications*, *10*(1), 1–11.

Chicco, D., P. Sadowski, and P. Baldi (2014), Deep autoencoder neural networks for gene ontology annotation predictions, in *Proceedings of the 5th ACM conference on bioinformatics, computational biology, and health informatics*, pp. 533–540.

Chithrananda, S., G. Grand, and B. Ramsundar (2020), Chemberta: Large-scale self-supervised pretraining for molecular property prediction, *arXiv preprint arXiv:2010.09885*.

Cong, L., et al. (2013), Multiplex genome engineering using crispr/cas systems, *Science*, *339*(6121), 819–823.

Consortium, T. M., et al. (2018), Single-cell transcriptomics of 20 mouse organs creates a tabula muris., *Nature*, *562*(7727), 367.

Crowley, L., et al. (2020), A single-cell atlas of the mouse and human prostate reveals heterogeneity and conservation of epithelial progenitors, *Elife*, *9*, e59,465.

Cui, H., C. Zhou, X. Dai, Y. Liang, R. Paffenroth, and D. Korkin (2020), Boosting gene expression clustering with system-wide biological information: a robust autoencoder approach, *International Journal of Computational Biology and Drug Design*, *13*(1), 98–123.

Danuser, G. (2011), Computer vision in cell biology, *Cell*, *147*(5), 973–978.

Datlinger, P., et al. (2017), Pooled crispr screening with single-cell transcriptome readout, *Nature methods*, *14*(3), 297–301.

Demetci, P., R. Santorella, B. Sandstede, W. S. Noble, and R. Singh (2020), Gromov-wasserstein optimal transport to align single-cell multi-omics data, *BioRxiv*.

Deng, Y., F. Bao, Q. Dai, L. F. Wu, and S. J. Altschuler (2018), Massive single-cell rna-seq analysis and imputation via deep learning, *bioRxiv*, p. 315556.

Denton, E. L., et al. (2017), Unsupervised learning of disentangled representations from video, in *Advances in neural information processing systems*, pp. 4414–4423.

Desjardins, G., A. Courville, and Y. Bengio (2012), Disentangling factors of variation via generative entangling, *arXiv preprint arXiv:1210.5474*.

Devlin, J., M.-W. Chang, K. Lee, and K. Toutanova (2018), Bert: Pre-training of deep bidirectional transformers for language understanding, *arXiv preprint arXiv:1810.04805*.

Devlin, J. P. (1997), *High throughput screening: the discovery of bioactive substances*, CRC Press.

Ding, J., A. Condon, and S. P. Shah (2018), Interpretable dimensionality reduction of single cell transcriptome data with deep generative models, *Nature communications*, *9*(1), 1–13.

Dinh, L., D. Krueger, and Y. Bengio (2014), Nice: Non-linear independent components estimation, *arXiv preprint arXiv:1410.8516*.

Dinh, L., J. Sohl-Dickstein, and S. Bengio (2016), Density estimation using real nvp, *arXiv preprint arXiv:1605.08803*.

Dixit, A., et al. (2016), Perturb-seq: dissecting molecular circuits with scalable single-cell rna profiling of pooled genetic screens, *cell*, *167*(7), 1853–1866.

Dosovitskiy, A., J. Tobias Springenberg, and T. Brox (2015), Learning to generate chairs with convolutional neural networks, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1538–1546.

Doudna, J. A., and E. Charpentier (2014), The new frontier of genome engineering with crispr-cas9, *Science*, *346*(6213).

Dowson, D., and B. Landau (1982), The fréchet distance between multivariate normal distributions, *Journal of multivariate analysis*, *12*(3), 450–455.

Dupont, E. (2018), Learning disentangled joint continuous and discrete representations, in *Advances in Neural Information Processing Systems*, pp. 710–720.

Eastwood, C., and C. K. Williams (2018), A framework for the quantitative evaluation of disentangled representations, in *International Conference on Learning Representations*.

Efremova, M., and S. A. Teichmann (2020), Computational methods for single-cell omics across modalities., *Nature Methods*, *17*(1), 14–17.

Eng, C.-H. L., et al. (2019), Transcriptome-scale super-resolved imaging in tissues by rna seqfish+, *Nature*, *568*(7751), 235–239.

Esmaeili, B., H. Wu, S. Jain, A. Bozkurt, N. Siddharth, B. Paige, D. H. Brooks, J. Dy, and J.-W. Meent (2019), Structured disentangled representations, in *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 2525–2534, PMLR.

Feldman, D., A. Singh, J. L. Schmid-Burgk, R. J. Carlson, A. Mezger, A. J. Garrity, F. Zhang, and P. C. Blainey (2019), Optical pooled screens in human cells, *Cell*, *179*(3), 787–799.

Frangieh, C. J., et al. (2021), Multimodal pooled perturb-cite-seq screens in patient models define mechanisms of cancer immune evasion, *Nature genetics*, *53*(3), 332–341.

Futamura, Y., M. Kawatani, S. Kazami, K. Tanaka, M. Muroi, T. Shimizu, K. Tomita, N. Watanabe, and H. Osada (2012), Morphobase, an encyclopedic cell morphology database, and its use for drug target identification, *Chemistry & biology*, *19*(12), 1620–1630.

Gao, S., R. Brekelmans, G. Ver Steeg, and A. Galstyan (2019), Auto-encoding total correlation explanation, in *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 1157–1166.

Gaulton, A., et al. (2012), Chembl: a large-scale bioactivity database for drug discovery, *Nucleic acids research*, *40*(D1), D1100–D1107.

Gehring, J., J. Hwee Park, S. Chen, M. Thomson, and L. Pachter (2020), Highly multiplexed single-cell rna-seq by dna oligonucleotide tagging of cellular proteins, *Nature Biotechnology*, *38*(1), 35–38.

Gilbert, L. A., et al. (2014), Genome-scale crispr-mediated control of gene repression and activation, *Cell*, *159*(3), 647–661.

Gilpin, L. H., D. Bau, B. Z. Yuan, A. Bajwa, M. Specter, and L. Kagal (2018), Explaining explanations: An overview of interpretability of machine learning, in *2018 IEEE 5th International Conference on data science and advanced analytics (DSAA)*, pp. 80–89, IEEE.

Gómez-Bombarelli, R., et al. (2018), Automatic chemical design using a data-driven continuous representation of molecules, *ACS central science*, *4*(2), 268–276.

Goodfellow, I., J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio (2014), Generative adversarial nets, in *Advances in neural information processing systems*, pp. 2672–2680.

Goodfellow, I., Y. Bengio, and A. Courville (2016), *Deep learning*, MIT press.

Goodwin, S., J. D. McPherson, and W. R. McCombie (2016), Coming of age: ten years of next-generation sequencing technologies, *Nature Reviews Genetics*, *17*(6), 333–351.

Grahn, H., and P. Geladi (2007), *Techniques and applications of hyperspectral image analysis*, John Wiley & Sons.

Grønbech, C. H., M. F. Vording, P. N. Timshel, C. K. Sønderby, T. H. Pers, and O. Winther (2018), scvae: Variational auto-encoders for single-cell gene expression datas, *bioRxiv*, p. 318295.

Grover, A., M. Dhar, and S. Ermon (2018), Flow-gan: Combining maximum likelihood and adversarial learning in generative models, in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32.

Gulrajani, I., F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville (2017), Improved training of wasserstein gans, in *Advances in neural information processing systems*, pp. 5767–5777.

Gupta, A., H. Wang, and M. Ganapathiraju (2015), Learning structure in gene expression data using deep architectures, with an application to gene clustering, in *2015 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pp. 1328–1335, IEEE.

Hamilton, N. (2009), Quantification and its applications in fluorescent microscopy imaging, *Traffic*, *10*(8), 951–961.

Hartford, J., G. Lewis, K. Leyton-Brown, and M. Taddy (2017), Deep iv: A flexible approach for counterfactual prediction, in *International Conference on Machine Learning*, pp. 1414–1423, PMLR.

Heusel, M., H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter (2017), Gans trained by a two time-scale update rule converge to a local nash equilibrium, in *Advances in neural information processing systems*, pp. 6626–6637.

Higgins, I., L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner (2017), beta-vae: Learning basic visual concepts with a constrained variational framework., *Iclr*, *2*(5), 6.

Higgins, I., D. Amos, D. Pfau, S. Racaniere, L. Matthey, D. Rezende, and A. Lerchner (2018), Towards a definition of disentangled representations, *arXiv preprint arXiv:1812.02230*.

Hinton, G. E. (2002), Training products of experts by minimizing contrastive divergence, *Neural computation*, *14*(8), 1771–1800.

Hinton, G. E., and R. R. Salakhutdinov (2009), Replicated softmax: an undirected topic model, *Advances in neural information processing systems*, *22*.

Hinton, G. E., S. Osindero, and Y.-W. Teh (2006), A fast learning algorithm for deep belief nets, *Neural computation*, *18*(7), 1527–1554.

Horlbeck, M. A., et al. (2016), Compact and highly active next-generation libraries for crispr-mediated gene repression and activation, *elife*, *5*, e19,760.

Hsu, W.-N., Y. Zhang, and J. Glass (2017), Unsupervised learning of disentangled and interpretable representations from sequential data, in *Advances in neural information processing systems*, pp. 1878–1889.

Hu, Q., and C. S. Greene (2019), Parameter tuning is a key part of dimensionality reduction via deep variational autoencoders for single cell rna transcriptomics., in *PSB*, pp. 362–373, World Scientific.

Hutchison III, C. A. (2007), Dna sequencing: bench to bedside and beyond, *Nucleic acids research*, *35*(18), 6227–6237.

Im, K., S. Mareninov, M. Diaz, and W. H. Yong (2019), An introduction to performing immunofluorescence staining, *Biobanking*, pp. 299–311.

Irwin, J. J., and B. K. Shoichet (2005), Zinc- a free database of commercially available compounds for virtual screening, *Journal of chemical information and modeling*, *45*(1), 177–182.

Islam, S., U. Kjällquist, A. Moliner, P. Zajac, J.-B. Fan, P. Lönnerberg, and S. Linnarsson (2011), Characterization of the single-cell transcriptional landscape by highly multiplex rna-seq, *Genome research*, *21*(7), 1160–1167.

Isola, P., J.-Y. Zhu, T. Zhou, and A. A. Efros (2017), Image-to-image translation with conditional adversarial networks, in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1125–1134.

Jain, I. H., et al. (2016), Hypoxia as a therapy for mitochondrial disease, *Science*, *352*(6281), 54–61.

Jaitin, D. A., et al. (2016), Dissecting immune circuits by linking crispr-pooled screens with single-cell rna-seq, *Cell*, *167*(7), 1883–1896.

Janzen, W. P. (2001), *High throughput screening: methods and protocols*, 190, Springer Science & Business Media.

Jeon, I., W. Lee, M. Pyeon, and G. Kim (2021), Ib-gan: Disengangled representation learning with information bottleneck generative adversarial networks, in *35th AAAI Conference on Artificial Intelligence/33rd Conference on Innovative Applications of Artificial Intelligence/11th Symposium on Educational Advances in Artificial Intelligence*, pp. 7926–7934, ASSOC ADVANCEMENT ARTIFICIAL INTELLIGENCE.

Jin, W., R. Barzilay, and T. Jaakkola (2020a), Hierarchical generation of molecular graphs using structural motifs, in *International Conference on Machine Learning*, pp. 4839–4848, PMLR.

Jin, X., et al. (2020b), In vivo perturb-seq reveals neuronal and glial abnormalities associated with autism risk genes, *Science*, *370*(6520).

Jinek, M., K. Chylinski, I. Fonfara, M. Hauer, J. A. Doudna, and E. Charpentier (2012), A programmable dual-rna–guided dna endonuclease in adaptive bacterial immunity, *science*, *337*(6096), 816–821.

Johansson, F., U. Shalit, and D. Sontag (2016), Learning representations for counterfactual inference, in *International conference on machine learning*, pp. 3020–3029, PMLR.

John, H., M. Birnstiel, and K. Jones (1969), Rna-dna hybrids at the cytological level, *Nature*, *223*(5206), 582–587.

Kamimoto, K., C. M. Hoffmann, and S. A. Morris (2020), Celloracle: Dissecting cell identity via network inference and in silico gene perturbation, *bioRxiv*.

Kaneko, T., K. Hiramatsu, and K. Kashino (2018), Generative adversarial image synthesis with decision tree latent controller, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6606–6615.

Kang, J., C.-H. Hsu, Q. Wu, S. Liu, A. D. Coster, B. A. Posner, S. J. Altschuler, and L. F. Wu (2016), Improving drug discovery with high-content phenotypic screens by systematic selection of reporter cell lines, *Nature biotechnology*, *34*(1), 70–77.

Karras, T., T. Aila, S. Laine, and J. Lehtinen (2017), Progressive growing of gans for improved quality, stability, and variation, *arXiv preprint arXiv:1710.10196*.

Karras, T., S. Laine, and T. Aila (2019), A style-based generator architecture for generative adversarial networks, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4401–4410.

Kazemi, H., S. M. Iranmanesh, and N. Nasrabadi (2019), Style and content disentanglement in generative adversarial networks, in *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 848–856, IEEE.

Khemakhem, I., D. Kingma, R. Monti, and A. Hyvarinen (2020), Variational autoencoders and nonlinear ica: A unifying framework, in *International Conference on Artificial Intelligence and Statistics*, pp. 2207–2217.

Kim, H., and A. Mnih (2018), Disentangling by factorising, *arXiv preprint arXiv:1802.05983*.

Kim, S., et al. (2016), Pubchem substance and compound databases, *Nucleic acids research*, *44*(D1), D1202–D1213.

Kingma, D. P., and J. Ba (2014), Adam: A method for stochastic optimization, *arXiv preprint arXiv:1412.6980*.

Kingma, D. P., and P. Dhariwal (2018), Glow: Generative flow with invertible 1x1 convolutions, *Advances in neural information processing systems*, *31*.

Kingma, D. P., and M. Welling (2013), Auto-encoding variational bayes, *arXiv preprint arXiv:1312.6114*.

Kingma, D. P., T. Salimans, R. Jozefowicz, X. Chen, I. Sutskever, and M. Welling (2016), Improved variational inference with inverse autoregressive flow, *Advances in neural information processing systems*, *29*.

Kokhlikyan, N., et al. (2020), Captum: A unified and generic model interpretability library for pytorch, *arXiv preprint arXiv:2009.07896*.

Konermann, S., et al. (2015), Genome-scale transcriptional activation by an engineered crispr-cas9 complex, *Nature*, *517*(7536), 583–588.

Kuenzi, B. M., J. Park, S. H. Fong, K. S. Sanchez, J. Lee, J. F. Kreisberg, J. Ma, and T. Ideker (2020), Predicting drug response and synergy using a deep learning model of human cancer cells, *Cancer cell*, *38*(5), 672–684.

Kusner, M. J., B. Paige, and J. M. Hernández-Lobato (2017), Grammar variational autoencoder, in *International Conference on Machine Learning*, pp. 1945–1954, PMLR.

Laine, S. (2018), Feature-based metrics for exploring the latent space of generative models.

Landrum, G. (2016), Rdkit: open-source cheminformatics http://www. rdkit. org, *Google Scholar There is no corresponding record for this reference*.

Larsen, A. B. L., S. K. Sønderby, H. Larochelle, and O. Winther (2016), Autoencoding beyond pixels using a learned similarity metric, in *International conference on machine learning*, pp. 1558–1566, PMLR.

LeCun, Y., Y. Bengio, and G. Hinton (2015), Deep learning, *nature*, *521*(7553), 436–444.

Lee, H., H. Yu, and J. Welch (2019), A beginner's guide to single-cell transcriptomics, *The Biochemist*, *41*(5), 34–38.

Lee, W., D. Kim, S. Hong, and H. Lee (2020), High-fidelity synthesis with disentangled representation, *arXiv preprint arXiv:2001.04296*.

Li, G., Y. Liu, Y. Zhang, N. Kubo, M. Yu, R. Fang, M. Kellis, and B. Ren (2019), Joint profiling of dna methylation and chromatin architecture in single cells, *Nature methods*, *16*(10), 991–993.

Lin, Z., K. K. Thekumparampil, G. Fanti, and S. Oh (2019), Infogan-cr: Disentangling generative adversarial networks with contrastive regularizers, *arXiv preprint arXiv:1906.06034*.

Liu, B., Y. Zhu, Z. Fu, G. de Melo, and A. Elgammal (2020), Oogan: Disentangling gan with one-hot sampling and orthogonal regularization., in *AAAI*, pp. 4836–4843.

Liu, Z., P. Luo, X. Wang, and X. Tang (2015), Deep learning face attributes in the wild, in *Proceedings of the IEEE international conference on computer vision*, pp. 3730–3738.

Locatello, F., S. Bauer, M. Lucic, G. Raetsch, S. Gelly, B. Schölkopf, and O. Bachem (2019), Challenging common assumptions in the unsupervised learning of disentangled representations, in *international conference on machine learning*, pp. 4114–4124.

Lopez, R., J. Regier, M. B. Cole, M. I. Jordan, and N. Yosef (2018), Deep generative modeling for single-cell transcriptomics, *Nature methods*, *15*(12), 1053–1058.

Lotfollahi, M., F. A. Wolf, and F. J. Theis (2019), scgen predicts single-cell perturbation responses., *Nature methods*, *16*(8), 715–721.

Lotfollahi, M., M. Naghipourfar, F. J. Theis, and F. A. Wolf (2020), Conditional out-of-distribution generation for unpaired data using transfer vae, *Bioinformatics*, *36*(Supplement_2), i610–i617.

Lotfollahi, M., A. K. Susmelj, C. De Donno, Y. Ji, I. L. Ibarra, F. A. Wolf, N. Yakubova, F. J. Theis, and D. Lopez-Paz (2021), Compositional perturbation autoencoder for single-cell response modeling, *bioRxiv*.

Lotfollahi, M., et al. (2022), Mapping single-cell data to reference atlases by transfer learning, *Nature Biotechnology*, *40*(1), 121–130.

Lu, G., and B. Fei (2014), Medical hyperspectral imaging: a review, *Journal of biomedical optics*, *19*(1), 010,901.

Lu, L., and J. D. Welch (2022), PyLiger: scalable single-cell multi-omic data integration in Python, *Bioinformatics*, doi:10.1093/bioinformatics/btac190, btac190.

Ma, J., M. K. Yu, S. Fong, K. Ono, E. Sage, B. Demchak, R. Sharan, and T. Ideker (2018), Using deep learning to model the hierarchical structure and function of a cell, *Nature methods*, *15*(4), 290–298.

Macosko, E. Z., et al. (2015), Highly parallel genome-wide expression profiling of individual cells using nanoliter droplets, *Cell*, *161*(5), 1202–1214.

Marouf, M., P. Machart, V. Bansal, C. Kilian, D. S. Magruder, C. F. Krebs, and S. Bonn (2020), Realistic in silico generation and augmentation of single-cell rna-seq data using generative adversarial networks, *Nature Communications*, *11*(1), 1–12.

Matthey, L., I. Higgins, D. Hassabis, and A. Lerchner (2017), dsprites: Disentanglement testing sprites dataset, *URL https://github. com/deepmind/dsprites-dataset/.[Accessed on: 2018-05-08]*.

McCloskey, K., A. Taly, F. Monti, M. P. Brenner, and L. J. Colwell (2019), Using attribution to decode binding mechanism in neural network models for chemistry, *Proceedings of the National Academy of Sciences*, *116*(24), 11,624–11,629.

Meier, J., R. Rao, R. Verkuil, J. Liu, T. Sercu, and A. Rives (2021), Language models enable zero-shot prediction of the effects of mutations on protein function, *Advances in Neural Information Processing Systems*, *34*.

Meissner, F., J. Geddes-McAlister, M. Mann, and M. Bantscheff (2022), The emerging role of mass spectrometry-based proteomics in drug discovery, *Nature Reviews Drug Discovery*, pp. 1–18.

Mescheder, L., S. Nowozin, and A. Geiger (2017), Adversarial variational bayes: Unifying variational autoencoders and generative adversarial networks, *arXiv preprint arXiv:1701.04722*.

Metzker, M. L. (2005), Emerging technologies in dna sequencing, *Genome research*, *15*(12), 1767–1776.

Metzker, M. L. (2010), Sequencing technologies—the next generation, *Nature reviews genetics*, *11*(1), 31–46.

Mimitou, E. P., et al. (2019), Multiplexed detection of proteins, transcriptomes, clonotypes and crispr perturbations in single cells, *Nature methods*, *16*(5), 409–412.

Minor, L. K. (2006), *Handbook of assay development in drug discovery*, CRC Press.

Mirza, M., and S. Osindero (2014), Conditional generative adversarial nets, *arXiv preprint arXiv:1411.1784*.

Miyato, T., and M. Koyama (2018), cgans with projection discriminator, *arXiv preprint arXiv:1802.05637*.

Mnih, A., and K. Gregor (2014), Neural variational inference and learning in belief networks, in *International Conference on Machine Learning*, pp. 1791–1799, PMLR.

Mudrakarta, P. K., A. Taly, M. Sundararajan, and K. Dhamdhere (2018), Did the model understand the question?, *arXiv preprint arXiv:1805.05492*.

Neal, R. M. (1992), Connectionist learning of belief networks, *Artificial intelligence*, *56*(1), 71–113.

Norman, T. M., M. A. Horlbeck, J. M. Replogle, Y. G. Alex, A. Xu, M. Jost, L. A. Gilbert, and J. S. Weissman (2019), Exploring genetic interaction manifolds constructed from rich single-cell phenotypes, *Science*, *365*(6455), 786–793.

Odell, I. D., and D. Cook (2013), Immunofluorescence techniques, *The Journal of investigative dermatology*, *133*(1), e4.

Odena, A., C. Olah, and J. Shlens (2017), Conditional image synthesis with auxiliary classifier gans, in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 2642–2651, JMLR. org.

Oord, A. v. d., N. Kalchbrenner, L. Espeholt, O. Vinyals, A. Graves, et al. (2016a), Conditional image generation with pixelcnn decoders, *Advances in neural information processing systems*, *29*.

Oord, A. v. d., N. Kalchbrenner, and K. Kavukcuoglu (2016b), Pixel recurrent neural networks, in *International conference on machine learning*, pp. 1747–1756, PMLR.

Oord, A. v. d., S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu (2016c), Wavenet: A generative model for raw audio, *arXiv preprint arXiv:1609.03499*.

Papadopoulos, N., P. R. Gonzalo, and J. Söding (2019), Prosstt: probabilistic simulation of single-cell rna-seq data for complex differentiation processes, *Bioinformatics*, *35*(18), 3517–3519.

Papalexi, E., et al. (2021), Characterizing the molecular regulation of inhibitory immune checkpoints with multimodal single-cell screens, *Nature genetics*, *53*(3), 322–331.

Papamakarios, G., E. Nalisnick, D. J. Rezende, S. Mohamed, and B. Lakshminarayanan (2021), Normalizing flows for probabilistic modeling and inference, *Journal of Machine Learning Research*, *22*(57), 1–64.

Pardue, M. L., and J. G. Gall (1969), Molecular hybridization of radioactive dna to the dna of cytological preparations, *Proceedings of the National Academy of Sciences*, *64*(2), 600–604.

Paysan, P., R. Knothe, B. Amberg, S. Romdhani, and T. Vetter (2009), A 3d face model for pose and illumination invariant face recognition, in *2009 Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance*, pp. 296–301, Ieee.

Pei, D., X. Shu, A. Gassama-Diagne, and J. P. Thiery (2019), Mesenchymal–epithelial transition in development and reprogramming, *Nature Cell Biology*, *21*(1), 44–53.

Pereira, D., and J. Williams (2007), Origin and evolution of high throughput screening, *British journal of pharmacology*, *152*(1), 53–61.

Perlman, Z. E., M. D. Slack, Y. Feng, T. J. Mitchison, L. F. Wu, and S. J. Altschuler (2004), Multidimensional drug profiling by automated microscopy, *Science*, *306*(5699), 1194–1198.

Pinkel, D., T. Straume, and J. W. Gray (1986), Cytogenetic analysis using quantitative, high-sensitivity, fluorescence hybridization, *Proceedings of the National Academy of Sciences*, *83*(9), 2934–2938.

Pinkel, D., J. Landegent, C. Collins, J. Fuscoe, R. Segraves, J. Lucas, and J. Gray (1988), Fluorescence in situ hybridization with human chromosome-specific libraries: detection of trisomy 21 and translocations of chromosome 4, *Proceedings of the National Academy of Sciences*, *85*(23), 9138–9142.

Pu, Y., W. Wang, R. Henao, L. Chen, Z. Gan, C. Li, and L. Carin (2017), Adversarial symmetric variational autoencoder, in *Advances in neural information processing systems*, pp. 4330–4339.

Qi, L. S., M. H. Larson, L. A. Gilbert, J. A. Doudna, J. S. Weissman, A. P. Arkin, and W. A. Lim (2013), Repurposing crispr as an rna-guided platform for sequence-specific control of gene expression, *Cell*, *152*(5), 1173–1183.

Raj, A., P. Van Den Bogaard, S. A. Rifkin, A. Van Oudenaarden, and S. Tyagi (2008), Imaging individual mrna molecules using multiple singly labeled probes, *Nature methods*, *5*(10), 877–879.

Ramesh, A., Y. Choi, and Y. LeCun (2018), A spectral regularizer for unsupervised disentanglement, *arXiv preprint arXiv:1812.01161*.

Rampášek, L., D. Hidru, P. Smirnov, B. Haibe-Kains, and A. Goldenberg (2019), Dr. vae: improving drug response prediction via modeling of drug perturbation effects, *Bioinformatics*, *35*(19), 3743–3751.

Ran, F., P. D. Hsu, J. Wright, V. Agarwala, D. A. Scott, and F. Zhang (2013), Genome engineering using the crispr-cas9 system, *Nature protocols*, *8*(11), 2281–2308.

Rao, R. M., J. Liu, R. Verkuil, J. Meier, J. Canny, P. Abbeel, T. Sercu, and A. Rives (2021), Msa transformer, in *International Conference on Machine Learning*, pp. 8844–8856, PMLR.

Reed, S., Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee (2016), Generative adversarial text to image synthesis, *arXiv preprint arXiv:1605.05396*.

Replogle, J. M., et al. (2022), Mapping information-rich genotype-phenotype landscapes with genome-scale perturb-seq, *Cell*.

Rezende, D., and S. Mohamed (2015), Variational inference with normalizing flows, in *International conference on machine learning*, pp. 1530–1538, PMLR.

Ridgeway, K. (2016), A survey of inductive biases for factorial representation-learning, *arXiv preprint arXiv:1612.05299*.

Riniker, S., and G. A. Landrum (2013), Similarity maps-a visualization strategy for molecular fingerprints and machine-learning methods, *Journal of cheminformatics*, *5*(1), 1–7.

Rives, A., et al. (2021), Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences, *Proceedings of the National Academy of Sciences*, *118*(15).

Rodriques, S. G., et al. (2019), Slide-seq: A scalable technology for measuring genome-wide expression at high spatial resolution, *Science*, *363*(6434), 1463–1467.

Rogers, D., and M. Hahn (2010), Extended-connectivity fingerprints, *Journal of chemical information and modeling*, *50*(5), 742–754.

Rolinek, M., D. Zietlow, and G. Martius (2019), Variational autoencoders pursue pca directions (by accident), in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 12,406–12,415.

Rombach, R., P. Esser, and B. Ommer (2020), Network-to-network translation with conditional invertible neural networks, *arXiv preprint arXiv:2005.13580*.

Rubin, A. J., et al. (2019), Coupled single-cell crispr screening and epigenomic profiling reveals causal gene regulatory networks, *Cell*, *176*(1-2), 361–376.

Sainburg, T., M. Thielk, B. Theilman, B. Migliori, and T. Gentner (2018), Generative adversarial interpolative autoencoding: adversarial training on latent space interpolations encourage convex latent distributions, *arXiv preprint arXiv:1807.06650*.

Salakhutdinov, R., and H. Larochelle (2010), Efficient learning of deep boltzmann machines, in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 693–700, JMLR Workshop and Conference Proceedings.

Schiebinger, G., et al. (2019), Optimal-transport analysis of single-cell gene expression identifies developmental trajectories in reprogramming, *Cell*, *176*(4), 928–943.

Schölkopf, B., F. Locatello, S. Bauer, N. R. Ke, N. Kalchbrenner, A. Goyal, and Y. Bengio (2021), Toward causal representation learning, *Proceedings of the IEEE*, *109*(5), 612–634.

Selvaraju, R. R., M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra (2017), Grad-cam: Visual explanations from deep networks via gradient-based localization, in *Proceedings of the IEEE international conference on computer vision*, pp. 618–626.

Shalem, O., et al. (2014), Genome-scale crispr-cas9 knockout screening in human cells, *Science*, *343*(6166), 84–87.

Shalit, U., F. D. Johansson, and D. Sontag (2017), Estimating individual treatment effect: generalization bounds and algorithms, in *International Conference on Machine Learning*, pp. 3076–3085, PMLR.

Shapiro, E., T. Biezuner, and S. Linnarsson (2013), Single-cell sequencing-based technologies will revolutionize whole-organism science, *Nature Reviews Genetics*, *14*(9), 618–630.

Shen, Y., J. Gu, X. Tang, and B. Zhou (2020), Interpreting the latent space of gans for semantic face editing, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9243–9252.

Shrikumar, A., P. Greenside, A. Shcherbina, and A. Kundaje (2016), Not just a black box: Learning important features through propagating activation differences, *arXiv preprint arXiv:1605.01713*.

Shrikumar, A., P. Greenside, and A. Kundaje (2017), Learning important features through propagating activation differences, in *International conference on machine learning*, pp. 3145–3153, PMLR.

Simonyan, K., A. Vedaldi, and A. Zisserman (2013), Deep inside convolutional networks: Visualising image classification models and saliency maps, *arXiv preprint arXiv:1312.6034*.

Singh, S., A. E. Carpenter, and A. Genovesio (2014), Increasing the content of high-content screening: an overview, *Journal of biomolecular screening*, *19*(5), 640–650.

Snijder, B., and L. Pelkmans (2011), Origins of regulated cell-to-cell variability, *Nature reviews Molecular cell biology*, *12*(2), 119–125.

Spurr, A., E. Aksan, and O. Hilliges (2017), Guiding infogan with semi-supervision, in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 119–134, Springer.

Srivatsan, S. R., et al. (2020), Massively multiplex chemical transcriptomics at single-cell resolution, *Science*, *367*(6473), 45–51.

Statello, L., C.-J. Guo, L.-L. Chen, and M. Huarte (2021), Gene regulation by long non-coding rnas and its biological functions, *Nature Reviews Molecular Cell Biology*, *22*(2), 96–118.

Subramanian, A., et al. (2017), A next generation connectivity map: L1000 platform and the first 1,000,000 profiles, *Cell*, *171*(6), 1437–1452.

Sundararajan, M., A. Taly, and Q. Yan (2017), Axiomatic attribution for deep networks, in *International conference on machine learning*, pp. 3319–3328, PMLR.

Sutton, R. S., and A. G. Barto (2018), *Reinforcement learning: An introduction*, MIT press.

Swanson, E., et al. (2021), Simultaneous trimodal single-cell measurement of transcripts, epitopes, and chromatin accessibility using tea-seq, *Elife*, *10*, e63,632.

Tan, J., M. Ung, C. Cheng, and C. S. Greene (2014), Unsupervised feature construction and knowledge extraction from genome-wide assays of breast cancer with denoising autoencoders, in *Pacific Symposium on Biocomputing Co-Chairs*, pp. 132–143, World Scientific.

Tang, F., C. Barbacioru, E. Nordman, B. Li, N. Xu, V. I. Bashkirov, K. Lao, and M. A. Surani (2010), Rna-seq analysis to capture the transcriptome landscape of a single cell, *Nature protocols*, *5*(3), 516–535.

Theis, L., A. v. d. Oord, and M. Bethge (2015), A note on the evaluation of generative models, *arXiv preprint arXiv:1511.01844*.

Tipping, M. E., and C. M. Bishop (1999), Probabilistic principal component analysis, *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *61*(3), 611–622.

Ursu, O., et al. (2020), Massively parallel phenotyping of variant impact in cancer with perturb-seq reveals a shift in the spectrum of cell states induced by somatic mutations, *bioRxiv*.

Van Den Oord, A., O. Vinyals, et al. (2017), Neural discrete representation learning, *Advances in neural information processing systems*, *30*.

Vaserstein, L. N. (1969), Markov processes over denumerable products of spaces, describing large systems of automata, *Problemy Peredachi Informatsii*, *5*(3), 64–72.

Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin (2017), Attention is all you need, *Advances in neural information processing systems*, *30*.

Wang, D., and J. Gu (2018), Vasc: dimension reduction and visualization of single-cell rna-seq data by deep variational autoencoder, *Genomics, proteomics & bioinformatics*, *16*(5), 320–331.

Wang, T., K. Birsoy, N. W. Hughes, K. M. Krupczak, Y. Post, J. J. Wei, E. S. Lander, and D. M. Sabatini (2015), Identification and characterization of essential genes in the human genome, *Science*, *350*(6264), 1096–1101.

Wang, Z., M. Gerstein, and M. Snyder (2009), Rna-seq: a revolutionary tool for transcriptomics, *Nature reviews genetics*, *10*(1), 57–63.

Way, G. P., and C. S. Greene (2017), Extracting a biologically relevant latent space from cancer transcriptomes with variational autoencoders, *BioRxiv*, p. 174474.

Welling, M., M. Rosen-Zvi, and G. E. Hinton (2004), Exponential family harmoniums with an application to information retrieval, *Advances in neural information processing systems*, *17*.

Wessels, H.-H., et al. (2022), Efficient combinatorial targeting of rna transcripts in single cells with cas13 rna perturb-seq, *bioRxiv*.

White, T. (2016), Sampling generative networks, *arXiv preprint arXiv:1609.04468*.

Wolf, F. A., P. Angerer, and F. J. Theis (2018), Scanpy: large-scale single-cell gene expression data analysis, *Genome biology*, *19*(1), 15.

Wolpert, D. H. (1996), The lack of a priori distinctions between learning algorithms, *Neural computation*, *8*(7), 1341–1390.

Wolpert, D. H., and W. G. Macready (1997), No free lunch theorems for optimization, *IEEE transactions on evolutionary computation*, *1*(1), 67–82.

Wu, Y., J. Donahue, D. Balduzzi, K. Simonyan, and T. Lillicrap (2019), Logan: Latent optimisation for generative adversarial networks, *arXiv preprint arXiv:1912.00953*.

Xu, Z., S. Wang, F. Zhu, and J. Huang (2017), Seq2seq fingerprint: An unsupervised deep molecular embedding for drug discovery, in *Proceedings of the 8th ACM international conference on bioinformatics, computational biology, and health informatics*, pp. 285–294.

Yadav, A., S. Shah, Z. Xu, D. Jacobs, and T. Goldstein (2017), Stabilizing adversarial nets with prediction methods, *arXiv preprint arXiv:1705.07364*.

Yan, X., J. Yang, K. Sohn, and H. Lee (2016), Attribute2image: Conditional image generation from visual attributes, in *European conference on computer vision*, pp. 776–791, Springer.

Yeo, G. H. T., S. D. Saksena, and D. K. Gifford (2021), Generative modeling of single-cell time series with prescient enables prediction of cell trajectories with interventions, *Nature communications*, *12*(1), 1–12.

Yu, C., et al. (2016), High-throughput identification of genotype-specific cancer vulnerabilities in mixtures of barcoded tumor cell lines, *Nature biotechnology*, *34*(4), 419–423.

Yu, H., and J. D. Welch (2021), Michigan: sampling from disentangled representations of single-cell data using generative adversarial networks, *Genome biology*, *22*(1), 1–26.

Yuan, B., C. Shen, A. Luna, A. Korkut, D. S. Marks, J. Ingraham, and C. Sander (2021), Cellbox: interpretable machine learning for perturbation biology with application to the design of cancer combination therapy, *Cell systems*, *12*(2), 128–140.

Zappia, L., B. Phipson, and A. Oshlack (2017), Splatter: simulation of single-cell rna sequencing data, *Genome biology*, *18*(1), 174.

Zhu, J., et al. (2021), Prediction of drug efficacy from transcriptional profiles with deep learning, *Nature Biotechnology*, pp. 1–9.

Zhu, J.-Y., T. Park, P. Isola, and A. A. Efros (2017), Unpaired image-to-image translation using cycle-consistent adversarial networks, in *Proceedings of the IEEE international conference on computer vision*, pp. 2223–2232.