

HARVARD UNIVERSITY
Graduate School of Arts and Sciences



DISSERTATION ACCEPTANCE CERTIFICATE

The undersigned, appointed by the
Committee on Higher Degrees in Systems Biology
have examined a dissertation entitled
Learning Inductive Representations of Biomedical Data
presented by Samuel G. Finlayson
candidate for the degree of Doctor of Philosophy and hereby
certify that it is worthy of acceptance.

Signature 

Typed name: Prof. Debora Marks

Signature 

Typed name: Prof. Matthew Might

Signature 

Typed name: Prof. Arjun Manrai

Date: June 9, 2020

Learning Inductive Representations of Biomedical Data

A DISSERTATION PRESENTED

BY

SAMUEL G. FINLAYSON

TO

THE COMMITTEE ON HIGHER DEGREES IN SYSTEMS BIOLOGY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

IN THE SUBJECT OF

SYSTEMS, SYNTHETIC, AND QUANTITATIVE BIOLOGY

HARVARD UNIVERSITY

CAMBRIDGE, MASSACHUSETTS

JUNE 2020

©2020 – SAMUEL G. FINLAYSON
ALL RIGHTS RESERVED.

Learning Inductive Representations of Biomedical Data

ABSTRACT

Representation learning with neural networks has catalyzed rapid progress in biomedical pattern recognition. This progress, however, has generally been limited to domains where data are abundant, richly structured, and stable. In contrast, much of biomedicine is marked by limited and poorly structured data and by highly dynamic deployment environments. In particular, many of the most compelling problem areas in biomedicine involve the ‘long tails’ of rare diseases and rare events. In this thesis, I confront the challenge of learning data representations whose utility can extend into dynamic and data-poor biomedical domains. I do so through three primary projects:

First, I present a novel method for representation learning with subgraphs. This method, called Subgraph Neural Networks (Sub-GNN), learns disentangled representations of subgraph structure, neighborhood, and position through property-aware routing channels. The work is motivated by the desire for methods that can better contextualize patient phenotypes (encoded as subgraphs) into the broader context of biomedical knowledge, which could allow for better diagnostic generalization to novel disorders involving previously unseen phenotypes. Subgraph neural networks provide a principled framework for doing just this, by leveraging the relational inductive biases of the underlying knowledge graph while still respecting subgraphs as independent entities.

Next, I present an approach to learning coordinated representations of small molecules and their associated transcriptional signatures. This approach extends a popular paradigm for drug development (known as connectivity mapping) to operate inductively, making predictions involving

drugs that have not previously been experimentally assayed. I benchmark the performance of this approach, studying the circumstances under which it can and cannot achieve strong performance.

Finally, I present an analysis of the clinical challenges posed by dataset shift, the phenomenon in which the input data to a deployed machine learning algorithm become mismatched with its training data. After introducing the problem of general dataset shift, I turn to a special case – adversarial examples – which reflect the worst-case generalization conditions for a machine learning system. I then build and test the representational robustness of three high-accuracy machine learning systems, constructing adversarial examples that cause their accuracy to drop to 0% on data that is imperceptibly different from the training data. I discuss the implications of these findings for clinical machine learning, offering specific regulatory recommendations.

I conclude my thesis with lessons learned from these projects, and provide an extensive appendix with three additional smaller-scale projects that branched off of my research.

Contents

o	INTRODUCTION	i
o.1	Obstacles to Machine Learning Posed by Biomedical Data	3
o.2	Inductive Generalization and Inductive Biases	7
o.3	Learned Representations of Data and Knowledge	14
o.4	Outline of this Thesis	21
i	DISENTANGLED REPRESENTATION LEARNING FOR SUBGRAPHS	23
i.1	Representation Learning on Graphs	24
i.2	Related Representation Learning Research Involving Subgraphs	27
i.3	Formulating the Subgraph Prediction Task	28
i.4	SUBGRAPH NEURAL NETWORKS	31
i.5	Experiments on 8 Novel Subgraph Datasets	38
i.6	Results	39
i.7	Conclusion	42
2	CROSS-MODAL REPRESENTATION ALIGNMENT OF TRANSCRIPTIONAL PROFILES AND SMALL MOLECULE THERAPEUTICS	43
2.1	A New Benchmark Task for Computational Chemistry	44
2.2	Multi-View Representation Alignment	47
2.3	Baseline, Oracle, and Proposed Retrieval Methods	48
2.4	Quantitative IR Experiments, Performance Analysis, Model Introspection	59
2.5	Overall Summary and Future Work	65
2.6	Conclusion	68
3	REPRESENTATIONAL ROBUSTNESS AND WORST-CASE GENERALIZATION IN CLINICAL MACHINE LEARNING	69
3.1	A Note on Terminology	70
3.2	Dataset Shift from a Clinical Perspective	71
3.3	Dataset Shift from a Technical Perspective	73
3.4	Robustness Analysis of Representative Clinical Deep Learning Systems	77
3.5	Regulatory Recommendations	83

3.6	Conclusion	87
4	CONCLUSION	88
4.1	Synthesis	88
4.2	Looking Forward	91
APPENDIX A GENERATIVE ADVERSARIAL NETWORKS FOR HIP FRACTURE SIMU- LATION		93
A.1	Generative Adversarial Networks	94
A.2	Constructing a GAN for Conditional Radiograph Generation	96
A.3	Results and Discussion	99
APPENDIX B ASSESSING AND EXTENDING THE GENERALIZABILITY OF CONNec- TIVITY MAPPING STRATEGIES		106
B.1	Key Assumptions of Connectivity Mapping	107
B.2	Measuring the factors that drive connectivity mapping performance	108
B.3	Data, Preprocessing, and Implementation	116
B.4	Conclusion	118
APPENDIX C ADDITIONAL ANALYSES OF ADVERSARIAL ATTACKS IN THE CLIN- ICAL DOMAIN		119
C.1	Methodological Details and Results of Adversarial Patch Attacks	119
C.2	Existing Adversarial Dynamics in the United States Healthcare System	120
C.3	Distinctive Technical Sources of Vulnerability to Adversarial Attacks on Medical Ma- chine Learning Systems	123
C.4	Strategies for Defending Against Adversarial Attacks	126
APPENDIX D SUPPLEMENTARY MATERIAL		128
D.1	Chapter 1	129
D.2	Chapter 2/Appendix B	138
D.3	Chapter 3	157
REFERENCES		188

List of Figures

1	Paradigms for Incorporating External Knowledge into Learned Representations . .	18
1.1	Topological Properties of Subgraphs	26
1.2	Subgraph Neural Network Architecture	31
2.1	Aligned Chemical-Transcriptomic Representation Learning Method	46
2.2	Cross-Modal Information Retrieval Results - PRC	61
2.3	Predictors of Successful Compound Retrieval	63
2.4	Visualization of Latent Chemical Embedding Space	64
2.5	Visualization of Latent Transcriptional Space	65
3.1	The Anatomy of an Adversarial Example	76
3.2	A Taxonomy of Medical ML Failures in Context of the ML Development Pipeline.	78
3.3	Adversarial Examples Derived from the Natural Images Correctly-Diagnosed with the Highest Confidence.	82
A.1	GAN-Generated Hip Radiographs with Nearest Neighbors in the Training Set. . .	98
A.2	t-SNE of Additional Examples of Real, Non-fractured Images.	101
A.3	t-SNE of Additional Examples of GAN-generated, Non-fractured Images.	102
A.4	t-SNE of Additional Examples of Real, Fractured Images.	103
A.5	t-SNE of Additional Examples of GAN-generated, Fractured Images.	104
A.6	t-SNE of Additional Examples of Real and GAN-generated Images.	105
B.1	Connectivity Mapping Performance Driven by Cell Line Similarity	110
B.2	Connectivity Mapping Performance Driven by Drug Specificity	114
B.3	Connectivity Mapping Performance Driven by Drug Characteristics	115
D.1	Procedures for Generating Synthetic Graphs	132
D.2	Cross-Modal Information Retrieval Results - ECDF	138
D.3	Additional Predictors of IR Performance	139
D.4	Examples of Structurally Similar Molecules	140
D.5	Latent Space Suggests Hidden Confounders	141

D.6	Latent Space of Transcriptional Signatures, Colored by Modal Cell Line	142
D.7	Precision-Recall Curves for All Query-Target Cell Line Pairs.	145
D.8	Empirical Cumulative Distribution Function Curves for All Query-Target Cell Line Pairs.	146
D.9	Cell Line Similarity Versus Retrieval Performance: Model Diagnostics.	147
D.10	Cell Line Similarity Versus Retrieval Performance: Cell Line Specific Visualization	148
D.11	Query-Target Average Performance Reflects Biological Similarity	149
D.12	Cmapping Ensemble Methods Limited to Top N Most Similar Cell Lines.	151
D.13	Cmapping Ensemble Methods with Target Signatures Thresholded by Baseline Correlation.	152
D.14	Drug Specificity Versus Retrieval Performance: Model Diagnostics.	153
D.15	Drug Specificity Versus False Positivity: Model Diagnostics.	154
D.16	Cmapping the LINCS and CMAP datasets Against Each Other.	156
D.17	Additional Examples of Adversarial Perturbations: Retinal Fundoscopy	158
D.18	Additional Examples of Adversarial Perturbations: Chest X-Ray	159
D.19	Additional Examples of Adversarial Perturbations: Dermoscopy	160

List of Tables

1.1	Six Topological Properties of Subgraphs	30
1.2	Sub-GNN Results on Synthetic Datasets	40
1.3	Sub-GNN Results on Real World Datasets	40
1.4	Sub-GNN Channel Ablation Analysis	41
2.1	Dataset Details for Chemical-Transcriptomic Alignment Experiments	49
2.2	Information Retrieval Results for Baseline and Proposed Methods	60
2.3	Information Retrieval Results for Oracle Methods	62
3.1	Results of White and Gray-box Adversarial Attacks	81
A.1	Quantitive Measures of GAN Image Quality.	99
D.1	Properties of Base Graphs in Synthetic and Real-World Datasets	135
D.2	Properties of SubGraphs in Synthetic and Real-World Datasets	135
D.3	Sub-GNN Performance on Synthetic Datasets (AUROC)	136
D.4	Sub-GNN Performance on Real World Datasets (AUROC)	137
D.5	Channel Ablation Analysis (AUROC)	137
D.6	Distance Metrics Versus Connectivity Mapping Performance: Raw Results	143
D.7	Distance Metrics Versus Connectivity Mapping Performance: Statistical Tests.	144
D.8	Cell Line Similarity Versus Retrieval Performance: Additional Details of Model Fit.	145
D.9	Retrieval Performance of Signature Ensembling Methods.	150
D.10	Results of Pairwise Wilcoxon Tests Comparing Raw Ranks from Various Ensembling Methods.	150
D.11	Drug Specificity Versus Retrieval Performance: Additional Details of Model Fit.	150
D.12	Drug Specificity Versus False Positivity: Additional Details of Model Fit.	151
D.13	Results of Pairwise Wilcoxon Tests Comparing Results from Intra- Versus Extracellular Genes.	151
D.14	Cmapping External GEO Datasets against the LINCS and CMAP Datasets.	152
D.15	Cmapping the LINCS and CMAP datasets Against Each Other.	152
D.16	GEO Accession Numbers for Connectivity Mapping Experiments	155
D.17	Results of White and Gray-box Patch Attacks	157

TO KATE.

Acknowledgments

I AM GRATEFUL to have been a part of two fantastic laboratory groups during my PhD studies: the laboratory of Isaac S. Kohane in the Department of Biomedical Informatics (DBMI) at Harvard Medical School, and the Clinical Decision Making Group (MEDG) headed by Peter Szolovits in MIT's Computer Science & Artificial Intelligence Laboratory (CSAIL). My colleagues in each of these groups have been uniformly insightful, entertaining, and generous with their time.

At DBMI, my first two years of research were highly influenced for the better by Andrew Beam, a then-postdoc who took me under his wing as I implemented my very first neural networks. This initially-parasitic relationship may have become more mutualistic over time, but has never ceased to be both instructive and fun. Over the past two academic years, I have also had the pleasure of co-leading a series of new projects with Emily Alsentzer, a fellow Zak/Pete student. Emily has been a dream collaborator – she is brilliant, humble, hard-working, and has been far too patient with me. Michelle Li has also been involved with several of these projects, and has been force of both productivity and positivity. Other excellent collaborators throughout my time in DBMI have included Brett Beaulieu-Jones, Kun-Hsing Yu, Shilpa Kobren, Nathan Palmer, William Yuan, Alex

Pickering, and Inbar Fried. I have learned so much from each of them. I would also like to especially thank Marinka Zitnik, who has become something of a de facto PhD advisor as I've branched into graph neural networks this year. Of course, there would be no Zaklab without Zak Kohane. I feel extremely lucky to be Zak's student; he is as wonderful a man as he is a brilliant thinker. Finally, I'd like to thank Zak's assistant Samantha Lemos, who is both incredibly good at her job and also a saint.

Over the last year and a half at MEDG, I've likewise learned much from MIT colleagues including Tristan Naumann, Anna Bair, Willie Boag, and Marzyeh Ghassemi. In particular, I've had the good fortune of working extensively with Matthew McDermott, who elevates any project in which he engages. As with Zak, I feel very lucky to have been advised by Pete Szolovits, even if only for part of my PhD. Pete puts his students first to an extent I've never before encountered in a PI, and has established a lab culture of complete intellectual purity. I've found it both highly instructive – along numerous axes – and very enjoyable to be a part of his group.

My collaborations outside of Zak and Pete's groups have been very instructive, as well. In particular, John Bowers, Jonathan Zittrain, Adarsh Subbaswamy, Suchi Saria, Joi Ito, and Annabel Kupke have been a pleasure to work with on my dataset shift projects. Aleksander Madry and Alexander Rush provided a lot of great one-off academic mentorship for me, during and after some class projects that ended up making their way into parts of this thesis. My dissertation and defense committees have also been extremely supportive and helpful. These have included Drs Debora Marks, Alexander Rush, Allon Klein, James J. Collins, Matthew Might, and Arjun Manrai. I am very grateful for their input and feedback.

The faculty, students, and staff of the Department of Systems, Synthetic, and Quantitative Biology have provided me with a wonderful home department during my PhD. The atmosphere within SSQBio is friendly but extremely rigorous, which has consistently lifted my scientific sights. In particular, I thank Debora Marks, Allon Klein, Galit Lahav, and John Higgins for all their encourage-

ment, mentorship, and examples. Liz Pomerantz has also been so consistently kind and patient with me, despite all she does to keep the department afloat. My pursuit of this PhD also would have been impossible without the Harvard-MIT MD-PhD program. The students, faculty, and administrators in the MD-PhD program are truly world-class, and I am indebted to many of them, but most notably to Amy Cohen and Loren Walensky, who work so tirelessly to run it.

My family has provided me with ceaseless support, without which I would neither be alive nor a scientist. Any attempt to adequately thank them is futile, so I'll instead acknowledge just a few principles that each member of my childhood family taught me about science: I thank my mother Pam for her boundless curiosity, expansive vision, and tireless work ethic; my father Grant for his love of rigorous argument and sharp prose; my brother Peter for his enthusiasm and continual quest for excellence; and my sister Kate for her undying optimism, kindness, diligence, honesty, and goodness.

My wife Elise has provided me these last eight years with a true partnership, an unshakeable friendship, and unbending support. My daughter Emma, born in my second year of my PhD, has provided me with more joy than I ever could have imagined, and has also supplied me with a multitude of fun insights into human and machine learning. I'm so deeply grateful for all the time I've had with her these last two years. My second daughter, Audrey, was born just a few weeks before this thesis was submitted, which likewise has brought immense happiness to the drafting phase of this work.

Finally, funding for my work came in large part from medical scientist training grant T32GM007753 from the National Institute of General Medical Science (and thus, indirectly, from every American taxpayer – thank you!).

Additional Author Contributions

CHAPTER 1 A variant of this chapter is in submission with author listing Alsentzer EM*, Finlayson SG*, Li M, Zitnik M. {*Equal Contribution}. SGF, EMA, and MZ conceived the study and designed the models and experiments. SGF and EMA implemented the novel methods, and ML implemented baseline methods. All authors participated in writing the manuscript.

CHAPTER 2 A variant of this chapter is in submission with author listing Finlayson SG, McDermott MBA, Pickering AV, Lipnick SL, Kohane IS. SGF conceived the study and wrote all code except as listed below. MBAM assisted with study design and implemented the hyperparameter optimization. AVP ran the differential expression analysis that produced counts of genes affected by each drug. SLL and ISK assisted with study design. The authors thank Connor Coley and Kyle Swanson for graciously providing the pretrained chemical embedder from Yang et al³⁵⁶. MBAM was funded in part by National Institutes of Mental Health grant P50-MH106933 and a Mitacs Globalink Research Award.

CHAPTER 3 AND APPENDIX C These chapters have been adapted from two research works: Finlayson SG, Bowers J, Ito J, Zittrain J, Beam AL, & Kohane IS. Adversarial attacks on medical machine learning. *Science*, 363(6433), 1287–1289^{91,92}, and a manuscript in preparation with author listing Finlayson SG*, Subbaswamy A*, Bowers J, Kupke A, Zittrain J, Kohane IS, Saria S {*Equal Contribution}. SGF conceived analyses and wrote all code, with the exception of the adversarial patch attacks in the Appendix, which were largely implemented by HWC. All other authors contributed to study design and the writing of the respective manuscripts.

APPENDIX A A variant of this chapter has been published as a workshop paper with the author listing Finlayson SG*, Lee H*, Kohane IS, Oakden-Rayner L. Towards generative adversarial networks as a new paradigm for radiology education. *Machine Learning for Healthcare Workshop at NeurIPS 2018*⁹³ {*Equal Contribution}. SGF conceived the study, wrote code, and drafted the manuscript. HL wrote code and drafted the manuscript. LOR and IK helped with study design, and LOR manually annotated and cropped the original dataset.

APPENDIX B A variant of this chapter is in submission with author listing Nakhoul M, Pickering AV, Kohane IS, Finlayson SG. SGF designed all experiments, implemented statistical analyses and figures, and wrote the manuscript. MN implemented retrieval experiments and generated figures. AVP assisted with data processing and all authors assisted with study design.

It is difficult to make predictions, especially about the future.

Karl Kristian Steincke

Don't not try.

Kate

O

Introduction

THE PAST DECADE has witnessed a resurgence of interest in artificial intelligence for biomedicine, catalyzed in large part by prominent successes involving “deep” neural representation learning.

Within clinical medicine, the greatest achievements of deep learning have been reported in image and signal processing tasks, with examples coming from radiology^{101,263}, cardiology¹⁶¹, pathology²³, dermatology⁸¹, and ophthalmology.¹²⁰ In the basic and translational biosciences, machine

learning has showed promise on pattern recognition tasks that include protein modeling¹⁴³, small molecule property prediction²⁹⁵, chemical synthesis²⁸², and genomics.⁷ This has resulted in the creation of billions of dollars in market capitalization, and prompted the United States Food and Drug Administration (FDA) to propose a novel regulatory framework specifically focused on software devices powered by artificial intelligence and machine learning.⁹⁶

For all this excitement, the vast majority of machine learning’s clear successes in biomedicine have taken place within a fairly narrow technical space. Specifically, machine learning has thrived in settings where (i) modelers have access to an abundance of well-curated and thoroughly annotated data, (ii) the data are richly structured and include archetypal motifs that align with the predictive task, and (iii) the testing conditions of the model closely resemble* those of training. As I will argue, these criteria do not currently hold for many of the most important problems in biomedicine. This poses challenges for biomedical machine learning, which I attempt to tackle in my work in this thesis.

The remainder of this introduction will be organized as follows: First, I will further motivate this work by describing major obstacles for machine learning that are posed by biomedical data, specifically focusing on those relevant to the three challenges I just described. I will then introduce and contextualize several technical concepts – inductive generalization, inductive biases, and modern representation learning – that form key technical background for my work addressing these challenges. I will close with an outline of the body of this thesis.

*For an example in which the third of these criteria broke down after development, consider the CC-Cruiser system for cataract diagnosis. Despite extremely accurate prospective performance during initial publication¹⁸⁹, subsequent testing in a new population showed a substantial drop in accuracy¹⁸⁶ – from 98% to 88% – whereas clinician performance was consistent across both populations.

0.1 OBSTACLES TO MACHINE LEARNING POSED BY BIOMEDICAL DATA

0.1.1 LIMITED DATA AVAILABILITY AND QUALITY.

Many of the most compelling challenges in healthcare involve medicine’s ‘long tail’ – rare diseases, rare events, and rare outcomes. For example, in the United States, there are as many as 7,000 individually rare[†] diseases that collectively afflict approximately 30 million Americans. Rare diseases are devastating both medically and economically, and from some perspectives seem to be a natural fit with machine learning’s promise to partially automate care and/or drug development: Patients with rare diseases tend to have highly complicated and poorly managed clinical courses, physicians treating rare diseases self-report limited knowledge of how to do so, and research funding is limited for most rare diseases²⁹⁶. In addition, the cost to the healthcare system of rare diseases is massive: management of some rare disease populations can cost up to billions of dollars per year²⁴⁸, with individual treatments costing an average of \$118,820 per patient per year²⁹⁰ and reaching prices as high \$2 million per dose⁵⁷. On the opposite end of the spectrum from rare diseases, patients with anomalously *positive* clinical courses – the so-called ‘super-responders’ – have attracted attention on the part of researchers, clinicians, and pharmaceutical companies^{309,47}. Unfortunately and unsurprisingly, aggregating large data sets from patients at either of these medical tails is extremely difficult. Furthermore, with approximately 300 entirely novel rare diseases discovered each year⁶⁷, and high-dimensional genomic data increasingly substratifying known disease populations, many patients in the healthcare system are truly ‘n-of-1’ at the time of their workup.

Even common diseases frequently pose small data challenges in a clinical and research ecosystem fraught by poor data infrastructure and reluctant data sharing practices. As of today, no central health data resource exists in the United States, interoperability between electronic medical records (EMR) systems is notoriously poor²⁵⁵, and privacy policies and legal practices create a systemic aver-

[†]Defined by the Orphan Drug act as those affected fewer than 200,000 individuals⁴

sion to sharing healthcare data²²⁴. Several promising efforts seek to change these trends, including data sharing protocols such as Fast Healthcare Interoperability Resources (FHIR)²⁴ and planned central databases such as the All of Us program²⁷⁵. Internationally, efforts such as the UK Biobank have made clinicogenomic data available for fairly large populations, but even these data tend to be limited to fairly healthy populations and to narrow age demographics (typically middle-aged adults)³⁰³. In the basic and translational biosciences, despite fewer legal obstacles, many similar cultural and infrastructural barriers exist to the sharing of data^{311,324}.

Label quality is often just as pressing as data quantity in biomedicine. The majority of clinical data and labels are derived from electronic health records (EHRs) or billing information, each of which have well-documented accuracy concerns^{335,316,266,37}. This is partly due to a fundamental misalignment between the goals of research and a healthcare information ecosystem that is almost exclusively optimized for the purpose of billing^{157,305}. Furthermore, the ground truth in medicine is often ambiguous or difficult to surmise, which can result in high inter-rater variability that necessitates large panels of physicians or gold standard laboratory tests to generate proper labels³⁴². This, in turn, is an expensive process which likely biases researchers and companies towards creating datasets with clear economic value. Data quality concerns in basic and translational datasets are also extremely pressing, with selective reporting, pressure to publish, poor experimental design, and fraud included among the underlying root causes²⁰.

Unsurprisingly, all of the above data concerns are magnified as it pertains to traditionally underrepresented minority populations, for whom machine learning systems have been consistently shown to under-perform^{233,5}.

Taken together, we cannot rely upon the availability of large swaths of well-labeled data for many biomedical problems, and any decision to apply biomedical machine learning only to problems with large datasets would systematically exclude our most physically and socially vulnerable patients. It is therefore necessary to develop methods that extend our success within established domains and tasks

into new domains and tasks. Achieving this goal will require a confluence of methods from many fields of machine learning research, which I outline in Section 0.2.3.

0.1.2 LIMITED DATA STRUCTURE AND DEPENDENCE ON EXTERNAL KNOWLEDGE

Biomedical deep learning has heretofore thrived on imaging, biosignal, and genome/protein sequencing tasks. Such data are well-suited for machine learning for several reasons: First, the data has rich spatial structure that can be exploited by recent advances in deep learning, such as convolutional¹⁷⁶ and self-attention³²¹ mechanisms. Second, the data have comparatively limited contextual dependence on outside information; an x-ray of a leg generally contains all the information necessary for a classifier to determine if that leg is broken, for example.

In contrast, many outstanding biomedical data tasks of interest are centered on high-dimensional[‡] categorical data types – billing codes, sets of mutated gene symbols, etc. – with complicated underlying relationships. When encoded in their raw categorical form, the symbols corresponding to these variables convey no information to machine learning algorithms; again, this can be contrasted with an x-ray, in which the raw value of a given pixel has direct biological meaning (radiodensity). This poses challenges for machine learning systems, which are by default required to learn from scratch the meaning of and background relationships between the symbols, along with the signal related to the actual predictive task. ML models will likewise be unable, by default, to meaningfully generalize to new tokens (codes, words, genes, etc.) that are related to the training data but that the system has never directly seen.

In order to extend the successes of machine learning approaches to data with such properties, it is important to develop methods to incorporate – explicitly or implicitly – pre-existing knowledge about the problem domain into the data representations. This topic is extensively discussed in the

[‡]The Unified Medical Language System (UMLS)³⁴, a government-funded attempt to codify medical terminology, has more than 4 million concepts and 10 million English terms.

second half of this introduction (see the discussion on relational inductive biases in Section 0.2.2 and representation learning in Section 0.3), and plays a central role in this thesis.

0.1.3 HIGHLY DYNAMIC DEPLOYMENT DOMAINS.

Biomedicine is highly dynamic. In clinical settings, it is frequent for new diseases to be discovered or re-categorized, new treatments and diagnostics to be developed, and existing practices to be found dangerous or ineffective and thus ceased²⁵³. More subtly, healthcare data is dynamic due to technical factors and due to operational dynamics at various hospitals^{110,225}. Strikingly, non-biological factors such as time of day, equipment models, and departmental location of various tests have been shown to be preferentially leveraged¹⁷ by ML models and in some cases more predictive than the results of the tests³³⁴.

Shifting technologies and laboratory practices also make the translational biosciences a dynamic domain. For example, consider transcriptional studies, which have shifted over the past several decades from expressed sequence tag, to RNA microarrays, to bulk RNA sequencing, and now single cell RNA sequencing¹⁹². Each of these broad families of technologies, in turn, includes dozens of competing platforms and processing pipelines, each of which update frequently yielding slightly different results⁴⁸.

Given the dynamic nature of biomedical data, and the fact that it is frequently unclear upon which features machine learning algorithms rely, it is imperative that we understand the factors that determine the prospective performance of safety-critical ML systems.

0.1.4 SUMMARY THUS FAR

Many of the most compelling challenges for machine learning in biomedicine are afflicted by limited data and labels (suggesting the need for tools to better extrapolate from the domains for which we

do have data), challenging data structure (suggesting the need to create more informed data representations), and dynamic deployment domains (suggesting the need to better understand the impact of these dynamics on prospective performance). These three challenges provide the guiding motivation for all of the work in my thesis.

In the next two sections of this introduction, I will take another step back and examine the broader technical context for these challenges within the field of machine learning. In doing so, I hope both to add further perspective and motivation for my work, as well as provide a framework for describing the methods I develop.

0.2 INDUCTIVE GENERALIZATION AND INDUCTIVE BIASES

Our goal in building machine learning systems is, with rare exceptions, to create algorithms whose utility extends beyond the dataset in which they are trained. In other words, we desire intelligent systems that are capable of generalizing to future data. The process of leveraging past observations to draw inferences about the unobserved is the principle of *induction*[§].

[§]Terminological note: In a non-technical setting, the term *inductive* – denoting the inference of general laws from particular instances – is typically contrasted with the adjective *deductive*, which denotes the inference of particular instances from general laws. This broad definition of induction is used in machine learning to describe, for example, the model fitting process as the *inductive step* and the deployment on new data as the *deductive step*. By the same token, some AI methods such as automated theorem provers⁹⁵ are described as deductive. In the setting of current ML research, however, it is much more common for the term ‘inductive’ to refer specifically to methods that are structurally capable of operating on new data points without retraining. In contrast, *transductive* methods require a fixed or pre-specified dataset, and are used to make internal predictions about missing features or labels⁴⁶. While many ML methods are assumed to be inductive in both senses of the term, this introduction concerns itself primarily with the broader notion of induction as it relates to learning from observed data. In contrast, Chapters 1 and 2 involve the second use of this term, as I propose new methods that are inductive but whose predecessors were transductive.

0.2.1 PHILOSOPHICAL FOUNDATIONS FOR THE PROBLEM OF INDUCTION

Even ancient philosophers appreciated the tenuity of inductive generalization. As early as the second century, the Greek philosopher Sextus Empiricus argued that the very notion of induction was invalid⁷⁹, a conclusion independently argued by the Charvaka school of philosophy in ancient India²⁴⁶. The so-called “problem of induction,” as it is best known today, was formulated by 18th-century philosopher David Hume in his twin works *A Treatise of Human Nature*¹⁴⁰ and *An Enquiry Concerning Human Understanding*¹⁴¹. In these works, Hume argues that all inductive inference hinges upon the premise that the future will follow the past. This premise has since become known as his “Principle of Uniformity of Nature” (or simply, the “Uniformity Principle”), the “Resemblance Principle,” or his “Principle of Extrapolation.”¹⁰³

In the *Treatise* and the *Inquiry*, Hume examines various arguments – intuitive, demonstrative, sensible, probabilistic – that could be proposed to establish the principle of extrapolation and, having rejected them all, concludes that inductive inference itself is “not determin’d by reason.” Hume thus places induction outside the scope of reason itself, casting it therefore as non-rational if not irrational. In his 1955 work *Fact, Fiction, and Forecast*, Nelson Goodman extended and reframed Hume’s arguments, proposing “a new riddle of induction.”¹¹⁵ For Goodman, the key challenge was not the validity of induction per se, but rather the recognition that for any set of observations, there are multiple contradictory generalizations that could be used to explain them.

At least among scientists, the best known formal response to the problem of induction comes from the philosopher of science Karl Popper. In *Conjectures and Refutations*²⁵², Popper argues that science may sidestep the problem of induction by relying instead upon scientific conjecture followed by criticism. Stated otherwise, according to Popper, the central goal of scientists should be to formulate falsifiable theories which can be provisionally treated as true when they survive repeated

attempts to prove them false[¶]. Popper’s arguments may be helpful as we frame our evaluation of any specific ML system that has already been trained – and thus instantiated, in a sense, as a “conjecture” that can be refuted. However, the training process of ML systems is itself an act of inductive inference and thus relies on a Uniformity Principle in a way that Popper’s conjecture-refutation framework does not address.

This thesis is not a work of philosophy. However, I consider it important to acknowledge that the entire field of machine learning – the branch of AI concerned with constructing computers that learn from experience²¹³ – is predicated upon a core premise that has, for centuries, been recognized as unprovable and arguably non-rational. To boot, even if the inductive framework is accepted as valid, there are an infinite number of contradictory generalizations that are equally consistent with our training data. While these observations may be philosophical in spirit and may appear impractical, they provide a framing for extremely practical questions:

Under which circumstances can we reasonably expect the future to resemble the past, as far as our models are concerned? Given an infinite number of valid generalizations from our data – most of which are presumably useless or even dangerous – what guiding principles do we leverage to choose between them? What are the falsifiable empirical claims that we should be making about our models, and how should we test them? If we are to assume that prospective failure of our systems is the most likely outcome, as Popper would, what reasonable standards can be set to nevertheless trust ML in safety-critical settings such as healthcare?

Each of these questions will be repeatedly considered throughout the course of this thesis.

[¶]Popper’s framing is frequently used to justify the statistical hypothesis testing frameworks proposed by the likes of Neyman, Pearson, and Fisher. However, the compatibility of Popperian falsification and statistical hypothesis testing is a matter of debate.^{130,201,257}

0.2.2 INDUCTIVE BIASES IN MACHINE LEARNING

As outlined above, the paradigm of machine learning presupposes the identification – a la Hume – of some set of tasks and environments for which we expect the future to resemble the past. At this point, we are thus forced to determine guiding principles – a la Goodman – that give our models strong *a priori* preferences for generalizations that we expect to extrapolate well into the future. When such guiding principles are instantiated as design decisions in our models, they are known as *inductive biases*.

In his 1980 report *The Need for Biases in Learning Generalizations*^{2,12}, Tom M. Mitchell argues that inductive biases constitute the heart of generalization and indeed a key basis for learning itself:

If consistency with the training instances is taken as the sole determiner of appropriate generalizations, then a program can never make the inductive leap necessary to classify instances beyond those it has observed. Only if the program has other sources of information, or biases for choosing one generalization over the other, can it non-arbitrarily classify instances beyond those in the training set. ...

The impact of using a biased generalization language is clear: each subset of instances for which there is no expressible generalization is a concept that could be presented to the program, but which the program will be unable to describe and therefore unable to learn. If it is possible to know ahead of time that certain subsets of instances are irrelevant, then it may be useful to leave these out of the generalization language, in order to simplify the learning problem. ...

Although removing all biases from a generalization system may seem to be a desirable goal, in fact the result is nearly useless. An unbiased learning system's ability to classify new instances is no better than if it simply stored all the training instances and performed a lookup when asked to classify a subsequent instance.

A key challenge of machine learning, therefore, is to design systems whose inductive biases align with the structure of the problem at hand. The effect of such efforts is not merely to endow the model with the capacity to learn key patterns, but also – somewhat paradoxically – to deliberately hamper the capacity of the model to learn other (presumably less useful) patterns, or at least to drive the model away from learning them. In other words, inductive biases stipulate the properties that we believe our model should have in order to generalize to future data; they thus encode our key assumptions about the problem itself.

The machine learning toolkit has a wide array of methods to induce inductive biases in learning systems. For example, regularization methods¹⁰⁷ such as L1-/L2-penalties³¹³, dropout²⁹³, or early stopping²⁵⁴ are a simple yet powerful means to impose Occam’s razor onto the training process. By the same token, the maximum margin loss of support vector machines⁶⁰, or model selection based on cross-validation can be described as inductive biases. Bayesian methods of almost any form induce inductive biases by placing explicit prior probabilities over model parameters. Machine learning systems that build on symbolic logic, such as inductive logic programming²¹⁷, encode established knowledge into very strict inductive biases, by forcing algorithms to reason about training examples explicitly in terms of hypotheses derived from pre-specified databases of facts. As nicely synthesized in Battaglia et al²¹, the standard layer types of modern neural networks each have distinct invariances that induce corresponding *relational inductive biases*; for example, convolutional layers have spatial translational invariance and induce a relational inductive bias of locality, whereas recurrent layers have a temporal invariance that induces the inductive bias of sequentiality. Such relational inductive biases are extremely powerful when well-matched to the data on which they are applied.

In section 0.3, I will introduce the neural representation learning framework – the dominant paradigm of machine learning today – and discuss inductive biases in this setting, with a special emphasis on recent tools for infusing external knowledge into the inductive biases of our models.

0.2.3 RELATED AREAS OF MACHINE LEARNING RESEARCH

The following is a brief overview of some areas of machine learning that touch upon different aspects of the challenge of inductive generalization:

*Dataset shift (DS)*²⁵⁸ is a major obstacle to generalization, and refers to systematic changes between training and test distributions. DS can be subclassified into changes due to covariate shift^{304,284}, target shift³⁶⁹, conditional shift^{368,369}, and policy shift²⁸¹. The existence of dataset shift motivates the design of models that are stable²⁹⁸ or distributionally robust²⁸⁶ to expected data instabilities. Alternatively, when properties of and/or samples from the target distribution are available, *domain adaptation* methods such as re-weighting^{297,117} can be used to improve performance in the target domain. Many such methods employ principles from the field of causal inference²⁴². The question of evaluating transportability in light of dataset shift is also intimately connected with the notion of *external validity*^{42,294} developed by statisticians and trialists of the mid-twentieth century. Dataset shift will be discussed in much greater depth in Chapter 3. *Domain Extrapolation* is a term that Tong et al³¹⁴ and others authors¹⁵² have employed a models accuracy outside its training domain; this is a topic of Chapter 2 and the major theme of Chapter B.

Out-of-distribution (OOD) or anomaly detection^{127,173,126,184,121,128,214,302} is a field of study focused on building systems that can recognize the limits of their own generalization. More specifically, the task of OOD detection focuses on identifying individual instances of input data that are significantly different[¶] from the training set. Typically, papers in this field evaluate methods by training a model on one dataset (such as the CIFAR-100 dataset¹⁶⁹) and then evaluating whether proposed methods can predict if input data came from another dataset (such as Imagenet⁶⁹). OOD detection methods may be useful in prospective deployment of mission-critical ML systems, because

[¶]Terminological note: The notion of OOD data is, technically, distinct from that of data in the Universe, which are data from the *same data distribution* but not from any class present in the training set^{338,365}; in practice, this distinction is not always made in the literature.

they could be used to flag anomalous, fabricated, or malformed data. In other words, they may identify one particular class of instances in which inductive generalization is expected a priori to be poor, and thus allow users to discount or ignore the results of the model in this setting.

Adversarial examples^{32,308,114} are inputs to machine learning models that have been crafted to force the model to make an error. Their relationship with generalization is two-fold. First, adversarial examples provide a tangible test-bed for assessing worst-case generalization; they have thus been employed by several papers as benchmarks for both distributional robustness and out of distribution detection^{286,260}. Second, adversarial examples are also studied in the field of computer security, which consider so-called adversarial attacks by end-users. In this second sense, adversarial examples provide a test bed for generalization to new environments with malicious agents. Adversarial examples are discussed (and implemented) in Section 3.3.2 of this thesis.

Fairness^{262,204} in machine learning is a burgeoning field that seeks to ensure that models can generalize with the same efficacy to all populations of patients. This touches upon all aspects of the ML pipeline in healthcare, from data acquisition^{334,129} to model design^{49,247} to deployment^{5,233,33}.

Few-shot learning (FSL)^{172,86,90,209,77} is a research area concerned with designing ML algorithms that can generalize effectively from as few as one or zero labeled training examples. Few-shot learning – especially in the zero-shot case – tends to be strongly dependent** upon representation learning, the topic of the next section of this introduction. The field of FSL is also intimately related with *Meta-learning*^{280,25,94}, the field concerned with training algorithms that operate on collections of learning tasks. Approaches to few shot learning (and meta-learning more broadly) include recurrent models²⁷⁶, metric learning^{164,292,325}, and optimization²⁶⁷. In particular, metric learning methods from this field will appear in Chapter 2 of this thesis, where I present a metric-learning based approach for zero-shot chemical retrieval from transcriptional signatures.

**For an exploration of the connection between few-shot learning and representation learning, see Chapter 15 of Goodfellow et al¹¹¹

0.3 LEARNED REPRESENTATIONS OF DATA AND KNOWLEDGE

The performance of most information processing systems, including machine learning systems, typically depends heavily upon the data representations (or features) they employ. Historically, this meant the devotion of significant labor and expertise to *feature engineering*, the design of data transformations and preprocessing techniques to extract and organize discriminative features from data prior to the application of ML. *Representation learning*^{26,112} is an alternative to feature engineering, and refers to the training of learned representations of data (or knowledge graphs³⁶) that are optimized for utility in downstream tasks such as prediction or information retrieval.

0.3.1 BACKGROUND ON REPRESENTATION LEARNING

Many canonical methods in statistical learning can be considered representation learning methods. For example, low-dimensional data representations with desirable properties are learned by unsupervised methods such as principal components analysis²⁴³, k-means clustering⁹⁷, independent components analysis¹⁵³, and manifold learning methods such as Isomap³¹⁰ and locally-linear embeddings²⁷³. Within the field of machine learning, the most popular paradigm for representation learning are neural networks^{26,112}, which provide an extremely flexible framework that can in theory be used to approximate any continuous function⁶³. Over the past two decades, representation learning with neural networks has steadily outperformed traditional feature engineering methods on a large family of tasks, including speech recognition⁶⁴, image processing¹³¹, and natural language processing²⁰⁷.

A common feature of all the representation learning methods just mentioned is that they are designed to learn data representations that have lower dimensionality than the original data. This basic inductive bias is motivated by the so-called *manifold hypothesis*, which states that most real world data – images, text, genomes, etc. – are captured and stored in high dimensions but actually

consist of some lower-dimensional data manifold embedded in that high-dimensional space.

Another desirable property of learned representations is that they be *distributed representations*^{26,112}, composed of multiple elements that can be set separately from each other. Distributed representations are highly expressive: n learned features with k values can represent k^n different concepts, with each feature element representing a degree of meaning along its own axis. This results in a rich similarity space that improves the generalizability of resultant models. The benefits of distributed representations apply to any data type, but are particularly obvious from a conceptual level when considering settings such as natural language processing²⁰⁸, where the initial data representations are encoded as symbols that lack any relationship with their underlying meaning. For example, the two sentences (or their equivalent triples, in a knowledge graph setting) ‘ibuprofen impairs renal function’ and ‘Advil damages the kidneys’ have zero tokens or ngrams in common. Thus, machine learning programs based only on symbols would be unable to extrapolate from one sentence to the other without relying upon explicit mappings such as ‘ibuprofen has_name Advil’, ‘impairs has_synonym damages’, etc. In contrast, the distributed representations of these sentences should, in principle, be nearly identical, facilitating direct extrapolation.

Over the past decade, neural networks have established themselves as the *de facto* approach to representation learning for essentially every ML problem in which their training has been shown feasible^{26,112}. While some neural architectures – e.g. Word2vec²⁰⁶ – are designed exclusively to produce embeddings that will be utilized in downstream tasks, the primary appeal of neural networks is that *every* deep learning architecture serves as a representation learning system. More specifically, the activations of each layer of neurons serves as a distributed representation of the input that is progressively refined in a hierarchical manner to produce representations of increased abstraction with increasing depth^{††}. In this light, a typical supervised neural network architecture of depth k , for

^{††}While even single-layer neural networks can provably approximate any continuous function, this guarantee is impractical because the proof assumes an infinite number of hidden nodes⁶³. Deep neural networks, in contrast, allow for feature re-use that is exponential in the number of layers, which makes deep networks

example, can arguably be best understood as a representation learning architecture of depth $k - 1$ followed by a simple linear or logistic regression (see Figure 1A).

Representations learned by neural networks have a number of desirable properties. First, neural representations are low-dimensional, distributed, and hierarchically organized, as described above. Neural networks have the ability to learn parameterized mappings that are strongly nonlinear but can still be used to directly compute embeddings for new data points. Yoshuo Bengio and others have extensively argued that neural networks have a higher capacity for generalization versus other well-established ML methods such as kernels^{29,27} and decision trees²⁸, specifically because they avoid an excessively strong inductive bias towards *smoothness*; in other words, when making a new prediction for some new data point x , deep representation learning methods do not exclusively rely upon the training points that are immediately nearby x in the original feature space.

Representation learning using neural networks also benefits from being modular, and therefore flexible^{††} and extensible to design. For example, given two neural architectures that each create a distributed representation of a unique data modality, these can be straightforwardly combined into a single, fused architecture that creates a composite multi-modal representation (e.g. combining audio embeddings and visual embeddings into composite video embeddings²²⁸). Such an approach is leveraged in Chapter 2. Another example of the power afforded by the modularity of neural architectures are *Generative Adversarial Networks* (GANs)¹¹³, which learn to generate richly structured data by pitting a data-simulating ‘generator’ model against a jointly-trained ‘discriminator’ model that is optimized to distinguish real from generated data. In Supplemental Chapter A, I demonstrate this approach using a GAN trained to simulate hip radiographs.

Taken together, neural architectures can be designed to expressively implement a broad array of

more expressive and more statistically efficient to train^{125,26}

^{††}The flexibility of neural networks doesn’t come without a price: In addition to obvious concerns about highly parameterized models and overfitting⁹⁹, for example, the ease of implementing complicated DL architectures has arguably produced a research culture focused on ever-larger – and more costly¹⁷¹ – models that are often poorly characterised and very difficult to reproduce?

inductive biases, while still allowing the network parameters to search over millions of compatible functions.

0.3.2 INFUSING DOMAIN KNOWLEDGE INTO NEURAL REPRESENTATIONS

Neural networks have largely absolved the contemporary researcher of the need to hand-engineer features, but this reality has not eliminated the role of external knowledge in ML design. In this section, I compare and contrast the various methods for explicitly and implicitly infusing domain knowledge into neural representations.

The first paradigm involves the design of layers and architectures that align the representational capacity of the network with our prior knowledge of the problem domain (Figure 1 top left). For instance, if we know that the data we provide have a particular property (e.g. unordered features), we can enforce corresponding constraints in our architecture (e.g. permutation invariance, as in DeepSet³⁷² or self-attention³²¹ without position encodings). This is an example of a relational inductive bias²¹. Relatedly, we can manually wire the network in a manner that corresponds with our prior understanding of relationships between variables. Peng et al²⁴⁵ adopted this approach by building a feed forward neural network for single cell RNA-Seq data in which the input neurons for each gene were wired according to the Gene Ontology¹²; this approach strictly weakens the capacity of the network, but may be useful if we have a strong prior that particular relationships would be confounding. An alternative means to a similar end is to perform graph convolutions over edges that reflect domain knowledge²⁰².

Another explicit paradigm for infusing knowledge into neural networks is to augment the architecture with the ability to query external information (Figure 1 top right). For example, models can be augmented with knowledge graphs in the form of fact triples, which they can query using an attention mechanism^{10,162}. More generally, attention can be used to allow modules to incorporate relevant information from embeddings of any knowledge source or data modality. For example, Xu

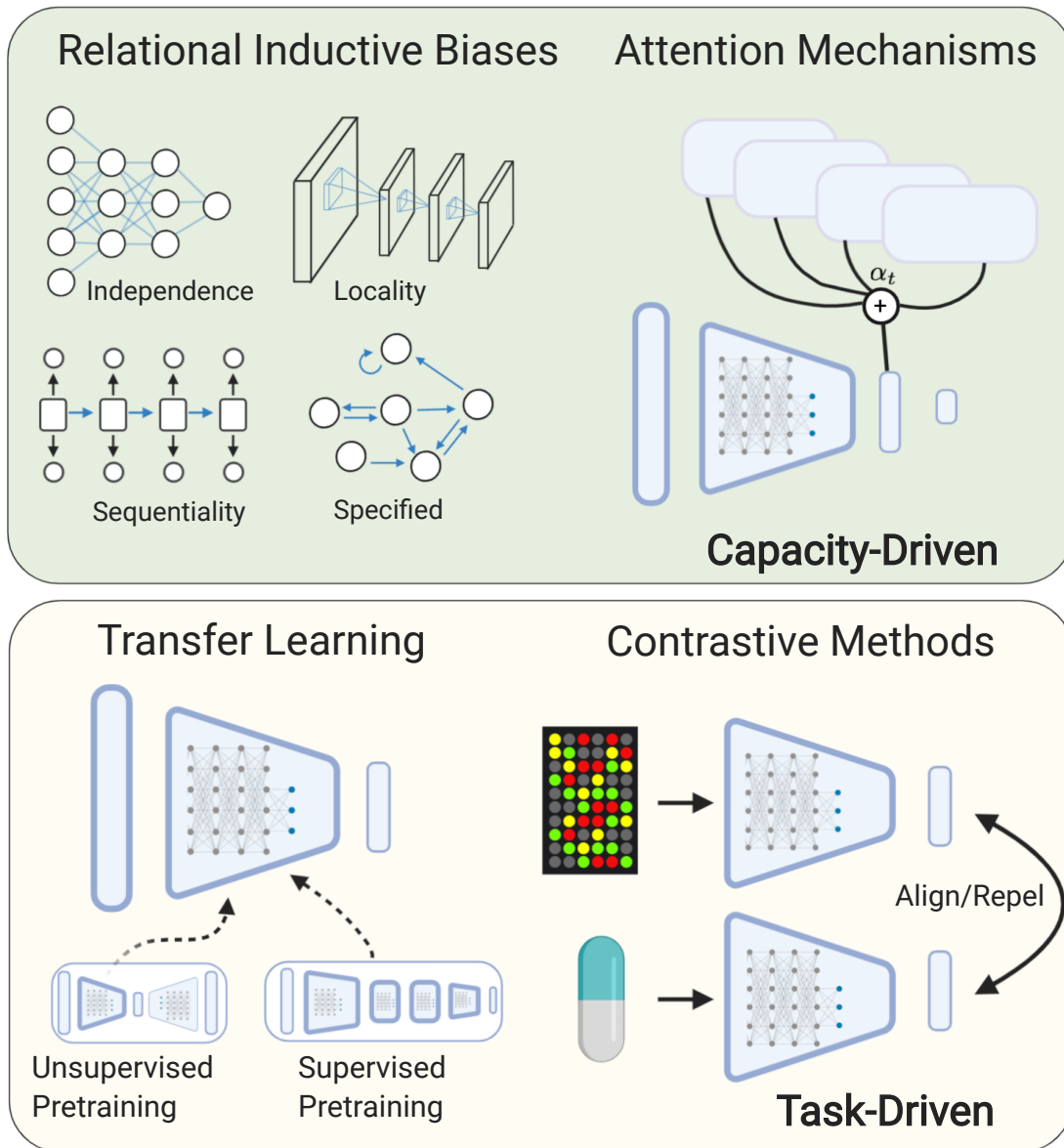


Figure 1: Example paradigms for infusing knowledge into learned representations.

et al.³⁵¹ introduced an architecture in which a language model attends to images to generate image captions. Self-attention, or intra-attention, is an attention mechanism that allows for relating different positions within a single sequence^{52,321}, image²⁴⁰, or other instance of input data; this allows representations to better share and synthesis information across features.

Transfer learning^{357,236} provides a family of methods to infuse knowledge into a learning algorithm that has been gained from a previous learning task. This is related to, but distinct from *multi-task learning*, which seeks to learn several tasks simultaneously under the premise that performance and efficiency can be improved by sharing knowledge between the tasks during learning. While there are many forms of transfer learning³⁷⁵, the canonical form in the setting of deep learning is *pretraining* (Figure 1 bottom left). In pretraining, model weights from a trained neural network are used to initialize some subset of the weights in another network; these parameters can then be either frozen or “fine-tuned” with further training on a the target task. Initial transfer learning experiments were conducted using unsupervised pretraining with autoencoders[¶] before transferring weights to a supervised model for a downstream task; this technique is an example of inductive *semi-supervised learning*³²⁰. In the past decade, supervised pretraining has become very popular, with the quintessential example being the initialization of an image processing architecture with all but the final layer of a model trained on the ImageNet dataset⁶⁹. More recently, *self-supervised* transfer learning has received significant attention, particularly in natural language processing. In self-supervised learning, subsets of a data or feature set are masked, and neural networks are trained to predict them

[¶]Autoencoders¹³³ are an example of a so-called ‘encoder-decoder’ architecture, which learn representations guided by the inductive bias that a good representation should be able to be used to reconstruct its raw input. Encoder architectures consist of an encoder, which take the raw input and use a series of layers to embed it into a low-dimensional space, and a decoder, which takes an embedding from the encoder and tries to construct raw data; this combined architecture is then trained in an end-to-end fashion. When the decoder is trained specifically to reconstruct the exact same input passed into the encoder, this is called an autoencoder. (Alternatively, decoders can be trained to produce related data, a prominent example being Seq2seq models that can, for example, encode a sentence from one language and decode it into another³⁰⁷.) *Variational autoencoders*¹⁶⁰ combine autoencoding with stochastic variational inference to build generative models that can be use for sampling entirely new data.

from remaining features. The resulting representations can then be used directly for downstream tasks, such as information retrieval, or be leveraged for transfer learning. Word embeddings²⁰⁸ are arguably the first widespread instance of self-supervised transfer learning, with more recent methods including language model pretraining^{137,70,321}.

Contrastive learning methods (Figure 1 bottom right) learn representations by taking in small sets of examples and optimize embeddings to bring similar data together while driving dissimilar data apart. This is a form of *metric learning*. Early methods in this field include Siamese^{164,89} and Triplet networks¹³⁵, which were initially developed to learn deep representations of images. Recent analyses suggest that many methods developed in the past several years have failed to advance beyond triplet networks²²². Contrastive methods have been used in the pretraining step of a semi-supervised framework to achieve the current state-of-the-art in limited data image classification⁵¹. In addition, contrastive optimization can be leveraged using multi-modal data to create aligned representations across modalities⁶⁸.

The methods described in this section can be described as a spectrum. Hand-engineered architectures are based on strong and specific prior assumptions about the problem domain, and are used to fundamentally alter the *representational capacity* of the network. In contrast, self-supervised and contrastive architectures make very minimal specific assumptions about the problem domain, and do nothing to alter the representational capacity of the algorithm; instead their innovation lies in devising *training schemes and loss functions* that will guide the network to learn underlying relationships and find a generalizable solution. In between these two extremes, augmented networks make the assumption that specific knowledge will be helpful, but allow the model to determine for itself which knowledge to employ. Transfer learning makes the assumption that other specific learning tasks will provide useful knowledge and experience for the target domain, but makes minimal assumptions about precisely what this knowledge would be. Despite (arguably significant) philosophical differences, these and yet other paradigms are not mutually exclusive, and share the common goal

of improving generalization and data efficiency by introducing richer domain understanding into the neural networks.

Finally, while this section – and indeed several chapters of this thesis – focuses on the design of neural architectures and training curricula, the role of domain knowledge is truly inescapable when it comes to the evaluation of deployable systems. Accordingly, the topic of deployment analysis will be a major theme of this thesis.

0.4 OUTLINE OF THIS THESIS

The remainder of this thesis will be as follows:

In Chapter 1, I formulate the problem of subgraph classification, and present a novel subgraph representation learning architecture. The primary technical goal of this chapter is to improve the *representational capacity* of our methods to encode subgraph data. This work was motivated by desire to have methods that can better contextualize patient phenotypes (subgraphs) into the broader context of biomedical knowledge, which could allow for better diagnostic generalization to novel disorders.

In Chapter 2, I describe a cross-model training scheme to learn coordinated representations of small molecules and their transcriptional profiles in living cells. In contrast with the first chapter, the innovation in here is a not to produce more representational capacity but rather to induce *representation alignment* between chemical embedders and gene expression. The motivation for this work is to make connectivity mapping (an approach to drug discovery) inductive rather than transductive with respect to drugs.

In Chapter 3, I present an analysis of the challenges posed by general dataset shift in machine learning models for healthcare. This includes an extensive evaluation of the *representational robustness* of existing deep learning diagnostic tools, as well as a more qualitative analysis of existing

technology and policy.

I conclude the main body of the text with key lessons learned from these primary projects. In the Appendices, I provide three additional analyses that branched off of my work: In Appendix A, I present generative adversarial networks trained to simulate hip fracture images. In Appendix B, I evaluate the domain extrapolation of connectivity mapping methods, inspired by limitations identified in chapter 2. In Appendix C, I provide a more thorough analysis of practical risks posed by adversarial attacks in clinical medicine. Appendix D contains supplementary figures and tables.

To name an illness is to describe a certain condition of suffering—a literary act before it becomes a medical one. A patient, long before he becomes the subject of medical scrutiny, is, at first, simply a storyteller, a narrator of suffering—a traveler who has visited the kingdom of the ill. To relieve an illness, one must begin, then, by unburdening its story.

Siddhartha Mukherjee

There is no power for change greater than a community discovering what it cares about.

Margaret J. Wheatley

1

Disentangled Representation Learning for Subgraphs

DEEP LEARNING METHODS FOR GRAPHS achieve remarkable performance on many node-level and graph-level prediction tasks. However, despite the proliferation of the methods and their success, prevailing Graph Neural Networks (GNNs) neglect subgraphs, rendering subgraph prediction tasks

challenging to tackle in many impactful applications. Further, subgraph prediction tasks present several unique challenges: subgraphs can have non-trivial internal topology, but also carry a notion of position and external connectivity information relative to the underlying graph in which they exist. Here, we introduce SUB-GNN, a subgraph neural network to learn disentangled subgraph representations. We propose a novel subgraph routing mechanism that propagates neural messages between the subgraph’s components and randomly sampled anchor patches from the underlying graph, yielding highly accurate subgraph representations. SUB-GNN specifies three channels, each designed to capture a distinct aspect of subgraph topology, and we provide empirical evidence that the channels encode their intended properties. We design a series of new synthetic and real-world subgraph datasets. Empirical results for subgraph classification on eight datasets show that SUB-GNN achieves considerable performance gains, outperforming strong baseline methods, including node-level and graph-level GNNs, by 12.4% over the strongest baseline. SUB-GNN performs exceptionally well on challenging biomedical datasets when subgraphs have complex topology and even comprise multiple disconnected components.

1.1 INTRODUCTION

Deep learning on graphs and Graph Neural Networks (GNNs), in particular, have emerged as the dominant paradigm for learning latent representations (*i.e.*, embeddings) on graphs³⁴⁶. These methods condense neighborhood connectivity patterns into low-dimensional embeddings for entities that can be used for downstream prediction tasks. While graph representation learning has made tremendous progress in recent years^{100,376}, prevailing methods focus on learning useful representations for nodes^{122,352}, edges^{102,181} or entire graphs^{19,138}.

Graph-level representations provide an overarching view of the graphs, but at the loss of some finer local structure. In contrast, node-level representations focus instead on the preservation of

the local topological structure (potentially to the detriment of the big picture). It is unclear if these methods can generate powerful representations for subgraphs and effectively capture the unique topology of subgraphs. Despite the popularity and importance of subgraphs for machine learning^{364,54,370}, there is still limited research on subgraph prediction, *i.e.*, to predict if a particular subgraph has a particular property of interest (Figure 1.1a). This can be, in part, because subgraphs are incredibly challenging structures from the topological standpoint (Figure 1.1b). (1) Subgraphs require that we make joint predictions over larger structures of varying sizes—the challenge is how to represent subgraphs that do not correspond to enclosing k -hop neighborhoods of nodes and can even comprise of multiple disparate components that are far apart from each other in the graph. (2) Subgraphs contain rich higher-order connectivity patterns, both internally among member nodes as well as externally through interactions between the member nodes and the rest of the graph—the challenge is how to inject information about border and external subgraph structure into the GNN’s neural message passing. (3) Subgraphs can be localized and reside in our region of the graph or can be distributed across multiple local neighborhoods—the challenge is how to effectively learn about subgraph positions within the underlying graph in which they are defined. (4) Finally, subgraph datasets give rise to unavoidable dependencies that emerge from subgraphs sharing edges and non-edges—the challenge is how to incorporate these dependencies into the model architecture while still being able to take feature information into account and facilitate inductive reasoning.

PRESENT WORK Here, we introduce SUB-GNN* (Figure 1.2), a novel graph neural network for subgraph prediction that addresses all of the above challenges. Unlike current prediction problems that are defined on individual nodes, pairwise relations, or entire graphs, our task here is to make predictions for subgraphs. To the best of our knowledge, SUB-GNN is the only representation learning method designed for subgraph classification. SUB-GNN’s core principle is to propagate messages

*The SUB-GNN implementation and datasets will be provided upon publication.

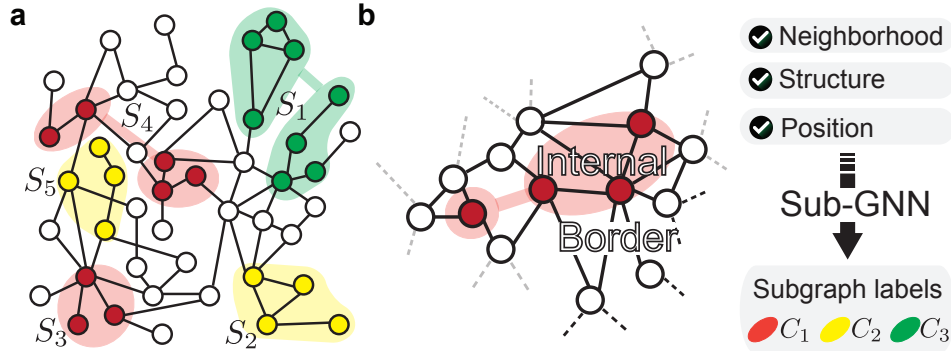


Figure 1.1: **a.** Shown is graph G in which nodes explicitly state their group memberships, resulting in intricate subgraph structures, $\mathcal{S} = \{S_1, \dots, S_5\}$. Subgraphs S_2, S_3 and S_5 comprise single connected components in G whereas subgraphs S_1 and S_4 each form two isolated components. Colors indicate subgraph labels, $\mathcal{C} = \{C_1, C_2, C_3\}$. **b.** We investigate the problem of predicting subgraph properties $\mathcal{C}$ by learning subgraph representations that recognize and disentangle the heterogeneous properties of subgraphs (*i.e.*, neighborhood, structure, and position) and how they relate to underlying G (*i.e.*, internal connectivity and border structure of the edge volume that points outside of the subgraph).

at the subgraph level, via three property-aware channels that capture position, neighborhood, and structure.

Experiments on eight datasets show that SUB-GNN outperforms baselines by an average of 75.4% on synthetic datasets and 100.6% on real-world datasets. Further, SUB-GNN improves the strongest baseline by 12.4%. As an example, on a network of phenotypes where subgraphs encode presenting symptoms associated with a disease, our method is able to distinguish between 10 sub-categories of monogenic neurological disorders. We also find that several naive generalizations of node-level and graph-level GNNs have poor performance. This finding is especially relevant as these generalizations are popular in practical applications, yet they cannot fully realize the potential of neural message-passing for subgraph prediction, as evidenced by our experiments. For example, we find that a popular node-level approach, which represents subgraphs with virtual nodes (*i.e.*, meta-nodes) and then uses an attention-based GNN (*e.g.*, GAT³²²) to train a node classifier, performs poorly in a variety of settings. Finally, we design a suite of synthetic subgraph benchmarks that are uniquely suited for evaluating aspects of subgraph topology and provide four challenging real-world

datasets to practically test the new algorithms.

1.2 RELATED WORK

We proceed by reviewing major threads of research relevant for subgraph representation learning.

NETWORK COMMUNITIES AND SUBGRAPH PREDICTION The topology of border structure has been extensively examined in the context of community detection^{355,226,350} (also known as module detection and graph clustering), highlighting its importance for network science^{30,227}. However, community detection, motif counting, and embedding methods for subgraph extraction and kernel-based similarity estimation^{278,306,361,327,150,61} are fundamentally different from SUB-GNN. These methods search for groups of nodes that are well-connected internally while being relatively well-separated from the rest of the graph and typically limited to individual connected components. In contrast, SUB-GNN is a subgraph prediction frameworks that reason over a given set of subgraphs.

LEARNING REPRESENTATIONS OF HIGHER-ORDER STRUCTURES, EGO NETS, AND ENCLOSING SUBGRAPHS Hypergraph neural networks³⁷³ and their variants^{279,87,371,219} have become a popular approach for learning on hypergraph-structured data. These methods use the spectral theory of hypergraphs³⁵⁴ or clique expansion^{3,148,151} to formulate higher-order message passing, where nodes receive latent representations from their immediate (one-hop) neighbors and from further N -hop neighbors at every message passing step. Recent studies, *e.g.*,^{363,218,337,35} also consider learning of representations for permutation-invariant functions of sequences (or multiset functions²⁰⁵). These methods typically express permutation-invariant functions as the average of permutation-sensitive functions applied to all reorderings of the input sequence. While this approach allows for pooling operators that are invariant to the ordering of nodes, it does not account for rich subgraph dependencies, such as when subgraphs share edges and have many connections to the rest of the graph.

SUBGRAPHS AND PATCHES IN GNNs Subgraph structure is key for numerous graph-related tasks^{323,72,360}. For example, Patchy-San²³¹ uses local receptive fields to extract useful features from graphs. Ego-CNN uses ego graphs to find critical structures³¹⁹. SEAL³⁷⁰ develops a theory showing that enclosing subgraphs are sufficient for link prediction. Building on this evidence, GraIL³¹² extracts local subgraphs induced by nodes occurring on the shortest paths between the target nodes and uses them for inducing logical rules. Cluster-GCN⁵⁴, Ripple walks¹⁸, and GraphSAINT³⁶⁴ use subgraphs to design more efficient and scalable algorithms for training deep and large GNNs. While these methods use substructures to make GNN architectures more efficient or improve performance on node and edge prediction tasks, none of the methods consider prediction on subgraphs.

1.3 FORMULATING SUBGRAPH PREDICTION

Let $G = (V, E)$ denote a graph comprised of a set of edges E and nodes V . While we focus on undirected graphs, it is straightforward to extend this work to directed graphs. $S = (V', E')$ is a subgraph of G if $V' \subseteq V$ and $E' \subseteq E$. Each subgraph S has a label y_S and may consist of multiple connected components, $S^{(C)}$, which are defined as a set of nodes in S such that each pair of nodes is connected by a path (e.g., see Figure 1.1). Importantly, the number and size of the subgraphs is bounded by, but does not directly depend on, the number of nodes in the graph G .

BACKGROUND ON GRAPH NEURAL NETWORKS Many graph neural networks, including ours, can be formulated as message passing networks (MPN)^{21,347,374}. Message-passing networks are defined by three functions, MSG, AGG, UPDATE, which are used to propagate signals between elements of the network and update their respective embeddings. Typically, these functions are defined to operate at the node level, propagating messages to a node v_i from the nodes in its neighborhood $\mathcal{N}_{v_i}$. In this typical context, a message between a pair of nodes (v_i, v_j) at layer l is defined

as a function of the hidden representations of the nodes $\mathbf{h}_i^{l-1}$ and $\mathbf{h}_j^{l-1}$ from the previous layer:
 $m_{ij}^l = \text{MSG}(\mathbf{h}_i^{l-1}, \mathbf{h}_j^{l-1})$. In AGG, messages from $\mathcal{N}_{v_i}$ are aggregated and combined with $\mathbf{h}_i^{l-1}$ to produce v_i 's representation for layer l in UPDATE. Many extensions of the canonical message passing framework have been proposed^{21,322,54,106}, e.g.,³⁶² pass messages from a shared set of anchor nodes rather than a strict neighborhood, to allow node embeddings to capture global position.

1.3.1 SUB-GNN: PROBLEM FORMULATION

Problem (Subgraph Classification). *Given subgraphs $\mathcal{S} = \{S_1, S_2, \dots, S_n\}$, SUB-GNN specifies a neural message passing architecture that generates a d -dimensional subgraph representation $\mathbf{z}_S \in \mathbb{R}^d$ for every subgraph S . SUB-GNN uses subgraph representations to learn a subgraph classifier $f : \mathcal{S} \rightarrow \{1, 2, \dots, C\}$ that predicts subgraph labels $f(S) = \hat{y}_S$.*

Note that while this paper focuses on the subgraph classification task, the methods we propose entail learning an embedding function $E_S : \mathcal{S} \rightarrow \mathbb{R}^d$ that maps each subgraph to a low-dimensional representation that captures the key aspects of subgraph topology necessary for prediction. All the techniques that we introduce to learn E_S can be extended without loss of generality to other supervised, unsupervised, or self-supervised tasks involving subgraphs.

In this work, we define message passing functions that operate at the *subgraph* level. This allows us to explicitly capture aspects of subgraph representation that do not apply to nodes or whole graphs. In particular, messages are propagated to each connected component in the subgraph, which allows us to build meaningful representations of subgraphs with multiple distinct connected components.

SUB-GNN: PROPERTIES OF SUBGRAPH TOPOLOGY Representation learning for subgraphs requires models to encode network properties that are not necessarily defined for either nodes or

graphs. Subgraphs have non-trivial internal structure, border connectivity, and notions of neighborhood and position relative to the rest of the graph. Intuitively, our goal is to learn an representation of each subgraph S_i such that the likelihood of preserving certain properties of the subgraph is maximized in the embedding space. Here we provide a framework of six properties of subgraph structure that are key for learning powerful subgraph representations (Table 1.1).

Table 1.1: Six properties of subgraph topology in Sub-GNN. See also Figure 1.1.

SUB-GNN Channel	SUB-GNN Subchannel	
	Internal (I)	Border (B)
Position (P)	Distance between S_i 's components	Distance between S_i and rest of G
Neighborhood (N)	Identity of S_i 's internal nodes	Identity of S_i 's border nodes
Structure (S)	Internal connectivity of S_i	Border connectivity of S_i

1) **POSITION** Two notions of position may be defined for subgraphs. The first, *border position*, refers to its distance to the rest of G ; this is directly analogous to node position, which can distinguish nodes with isomorphic neighborhoods³⁶². The second, *internal position*, captures the relative distance between components within the subgraph.

2) **NEIGHBORHOOD** Subgraphs extend the notion of a node's neighborhood to include both internal and external elements. *Border neighborhood*, as with nodes, is defined as the set of nodes within k hops of any node in S . Each connected component in a subgraph will have its own border neighborhood. Subgraphs also have a non-trivial *internal neighborhood*, which can vary in size and position.

3) **STRUCTURE** The connectivity of subgraphs is also non-trivial. A subgraph's *internal structure* is defined as the internal connectivity of each component. Simultaneously, subgraphs have a *border structure*, defined by edges connecting internal nodes to the border neighborhood.

We expect that capturing each of these properties will be crucial for learning useful subgraph representations, with the relative importance of these properties determined by the downstream task.

1.4 SUB-GNN: SUBGRAPH NEURAL NETWORK

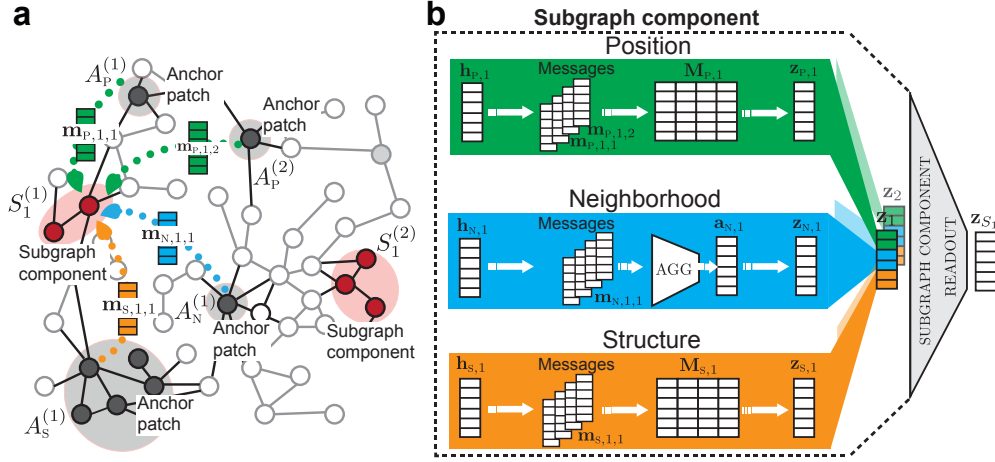


Figure 1.2: Sub-GNN architecture. **a.** Property-specific messages m_x are propagated from anchor patch A_x^q to subgraph component $S^{(c)}$. **b.** Sub-GNN specifies three channels, which are each designed to capture a distinct subgraph property. Channel outputs z_x are concatenated to produce a final subgraph representation z_S .

Next, we describe SUB-GNN, our approach to subgraph classification, which is designed to learn representations that can capture the 6 aspects of subgraph structure from Table 1.1. SUB-GNN learns representations of subgraphs in a hierarchical fashion, by propagating neural messages from anchor patches to subgraph components and aggregating the resultant representations into a final subgraph embedding (Figure 1.2a). Message-passing for each connected component takes place in independent position, neighborhood, and structure channels, each explicitly optimized to encode a distinct subgraph property (Figure 1.2b). The messages are sent from anchor patches sampled from throughout the graph and weighted by a property-specific similarity function. This approach is modular, extensible, and can be customized based on knowledge of the downstream task.

1.4.1 SUBGRAPH-LEVEL MESSAGE PASSING

The core of our approach is a message passing framework, as described in Table 1.1. Our key contribution is a novel aggregation scheme that we define at the level of subgraph components. SUB-GNN specifies how to propagate neural messages from a set of anchor patches, $\mathcal{A} = \{A^{(1)}, \dots, A^{(Q)}\}$, to subgraph components and, eventually, to entire subgraphs, resulting in subgraph representations that capture distinct properties of subgraph topology (Table 1.1).

Anchor patches are subgraphs, which we randomly sample from G in a channel-specific manner, resulting in anchor patches $\mathcal{A}_p$, $\mathcal{A}_n$, and $\mathcal{A}_s$ for each of three SUB-GNN’s channels, position, structure, and neighborhood, respectively. We define the message from anchor patch A_x to subgraph component $S^{(C)}$ as follows:

$$\text{MSG}_x = \gamma_x \left(S^{(C)}, A_x \right) \cdot p_x \quad (1.1)$$

where x is the channel, γ_x a similarity function between the connected component $S^{(C)}$ and the anchor patch A_x , and p_x is the learned feature representation of A_x . Similarity function γ_x is any learned or pre-specified similarity measures that quantifies the relevance of A_x to $S^{(C)}$. These messages are then transformed into order-invariant hidden representation $\mathbf{h}_{x,c}^{(l)}$ at layer l and subgraph component $S^{(C)}$ as follows:

$$\begin{aligned} \mathbf{a}_{x,c} &= \text{AGG}_M \left(\left\{ \text{MSG}_x(S^{(C)}, A_x, p_x), \forall A_x \in \mathcal{A}_x \right\} \right), \\ \mathbf{h}_{x,c}^{(l)} &= \sigma \left(\mathbf{W}_b \cdot [\mathbf{a}_{x,c}; \mathbf{h}_{x,c}^{(l-1)}] \right), \end{aligned} \quad (1.2)$$

where $\mathbf{W}_b$ is a learnable weight matrix, σ is any non-linear activation function, $\mathbf{h}_{x,c}^{(l-1)}$ is the hidden representation for the previous layer, and AGG_M is any aggregation function that operates over messages (*i.e.*, AGG in “Background on GNNs” in Section 1.3.1). Note that c is shorthand notation for $S^{(C)}$. Eq. (1.2) outputs a channel-specific hidden representation $\mathbf{h}_{x,c}^{(l)}$ for component $S^{(C)}$ and

channel x (*i.e.*, UPDATE in Section 1.3.1). This hidden representation is then passed to the next layer of the SUB-GNN architecture.

The order invariance of $\mathbf{h}_{x,c}$ is a necessary property for layer-to-layer message passing, however, it removes the much-needed information on structure and position of subgraphs³⁶². Because of that, we create property-aware output representations, $\mathbf{z}_{x,c}$, by producing a matrix of anchor-set messages $\mathbf{M}_x$, where each row is an anchor-set message computed by MSG_x , and then passing that matrix through a nonlinear activation function (Figure 1.2b and Algorithm 1). In the resulting representation, each dimension of the embedding encodes the structural or positional message from anchor patch $\mathcal{A}$. For the neighborhood channel, we set $\mathbf{z}_{N,c} = \mathbf{h}_{N,c}$. Finally, SUB-GNN routes messages for internal and border properties (*i.e.*, $\{\mathbf{P}_I, \mathbf{P}_B\}$, $\{\mathbf{N}_I, \mathbf{N}_B\}$, $\{\mathbf{S}_I, \mathbf{S}_B\}$) within subchannels for each channel $\mathbf{P}$, $\mathbf{N}$ and $\mathbf{S}$, and concatenates the final outputs.

Property-aware output representations $\mathbf{z}_{x,c}$ are transformed into final subgraph representations through order-invariant functions defined at the channel-, layer-, and component-level. Channel-specific representations $\mathbf{z}_{x,c}$ are first aggregated into a subgraph component representation for each layer, via channel aggregation function AGG_C . We then aggregate component representations across all layers via AGG_L to produce a final subgraph component representation $\mathbf{z}_c$ ³⁵³. Finally, the component representations $\mathbf{z}_c$ for every component $S^{(C)}$ are aggregated into a final subgraph representation $\mathbf{z}_S$ via READOUT. A full description of SUB-GNN is detailed in Appendix D.1.1.

1.4.2 PROPERTY-AWARE ROUTING CHANNELS

We enforce property-awareness through the creation of dedicated routing channels for position, neighborhood, and structure. Each channel has three key elements: (1) a sampling function ϕ to generate anchor patches, (2) an anchor patch encoder ψ , and (3) a similarity function γ to weight messages sent between anchor patches and subgraph components.

Algorithm 1: SUBGRAPH NEURAL NETWORK.

Input: Graph $G = (V, E)$; Node representations $\{\mathbf{x}_u | u \in V\}$; Subgraph S consisting of connected components $S^{(c)}$ for $c = 1, \dots, R$; Channels $\mathbf{N}$, $\mathbf{s}$, and $\mathbf{p}$ corresponding to neighborhood, structure, and position; Subchannels $\mathbf{I}$ and $\mathbf{B}$ corresponding to internal and border subgraph topology; Anchor patch sampling function $\varphi_x : (G, S) \rightarrow A_x$; Anchor patch encoder $\psi_x : A_x \rightarrow \mathbb{R}^d$; Trainable weight matrices $\mathbf{W}_{x,z}^{(l)}$ and $\mathbf{W}_{x,b}^{(l)}$ for each layer $l \in [1, L]$ and each channel x ; Nonlinear activation function σ .

Output: Subgraph representation $\mathbf{h}_S$ for subgraph S

$\mathbf{z}_c^0 = \sum_{u \in S^{(c)}} \mathbf{x}_u$
 $\mathbf{h}_{x,c}^0 = \mathbf{z}_c^0$ for channel $x \in \{\mathbf{N}, \mathbf{s}, \mathbf{p}\}$ // Channel-independent initialization

for layer $l = 1, \dots, L$ **do**

$A_x^{(q)} = \varphi_x(G)$ for $q \in \{1, \dots, Q\}$ and channel $x \in \{\mathbf{s}_I, \mathbf{s}_B, \mathbf{p}_B\}$ // See Section 1.4.2

$A_x^{(q)} = \varphi_x(G, S)$ for $q \in \{1, \dots, Q\}$ and channel $x \in \{\mathbf{p}_I\}$

for connected component $c = 1, \dots, R$ **do**

$A_x^{(q)} = \varphi_x(G, S^{(c)})$ for $q \in \{1, \dots, Q\}$ and channel $x \in \{\mathbf{N}_I, \mathbf{N}_B\}$

for channel $x \in \{\{\mathbf{p}_I, \mathbf{p}_B\}, \{\mathbf{N}_I, \mathbf{N}_B\}, \{\mathbf{s}_I, \mathbf{s}_B\}\}$ **do**

for anchor patch $q = 1, \dots, Q$ **do**

$\mathbf{p}_x^{(q)} = \psi_x(A_x^{(q)})$ // E.g., Algorithm 2

$\mathbf{m}_{x,c,q}^{(l)} = \text{MSG}_x(S^{(c)}, A_x^{(q)}, \mathbf{p}_x^{(q)})$ // (Eq.1.1)

$\mathbf{M}_{x,c}^{(l)}[q] = \mathbf{m}_{x,c,q}^{(l)}$ if $x \in \{\mathbf{s}_*, \mathbf{p}_*\}$

end

$\mathbf{z}_{x,c}^{(l)} = \sigma(\mathbf{W}_{x,z}^{(l)} \cdot \mathbf{M}_{x,c}^{(l)})$ if $x \in \{\mathbf{s}_*, \mathbf{p}_*\}$ // Property-aware output rep.

$\mathbf{a}_{x,c}^{(l)} = \text{AGG}_M(\{\mathbf{m}_{x,c,1}^{(l)}, \dots, \mathbf{m}_{x,c,Q}^{(l)}\})$ // Aggregate messages (Eq.1.2)

$\mathbf{h}_{x,c}^{(l)} = \sigma(\mathbf{W}_{x,b}^{(l)} \cdot [\mathbf{a}_{x,c}^{(l)}; \mathbf{h}_{x,c}^{(l-1)}])$ // Order-invariant hidden rep. (Eq.1.2)

end

$\mathbf{z}_c^{(l)} = \text{AGG}_C(\mathbf{h}_{\mathbf{N},c}^{(l)}, \mathbf{z}_{\mathbf{s},c}^{(l)}, \mathbf{z}_{\mathbf{p},c}^{(l)})$ // Aggregate channels

end

$\mathbf{z}_c \leftarrow \text{AGG}_L(\{\mathbf{z}_c^{(0)}, \dots, \mathbf{z}_c^{(L)}\})$ // Aggregate layers

$\mathbf{h}_S = \text{READOUT}(\{\mathbf{z}_1, \dots, \mathbf{z}_R\})$ // Aggregate components

SAMPLING ANCHOR PATCHES For each subchannel, we define an anchor patch sampling function $\varphi_x : (G, S^{(C)}) \rightarrow A_x$. In the position channel, the internal anchor patch sampler φ_{p_I} returns patch A_{p_I} comprised of a single node $u_p \in S$. The resulting set of anchor patches A_{p_I} are shared across all components in S . This allows representations of S 's components to be positioned relative to each other. In contrast, the border position anchor patch sampler is designed to position subgraphs relative to the rest of the graph G . Accordingly, φ_{p_B} samples nodes $v_p \in G$ such that A_{p_B} is shared across all S in G . In the neighborhood channel, the internal anchor patch sampler φ_{n_I} samples nodes $u_N \in S^{(C)}$ and border patch sampler φ_{n_B} samples nodes v_N from the k -hop neighborhood of the subgraph component. The structure anchor patch sampler φ_s is used for both internal and border structure, and the resulting A_s are shared across all S . φ_s returns a connected component sampled from G via triangular random walks. Triangular random walks are a second-order, biased random walk that extend classical random walks to privilege the traversal of triangles[?]. As shown in ^{118?,38}, this enables more effective recovery of structural properties in graphs. Refer to Appendix D.1.1 for more details.

NEURAL ENCODING OF ANCHOR PATCHES Each channel x specifies an anchor patch encoder $\psi_x : A_x \rightarrow \mathbf{p}$ that encodes an anchor patch A_x into a d -dimensional embedding $\mathbf{p}$. Note that ψ_N and ψ_p simply map to the node embedding of the anchor patch node. In contrast, ψ_s returns representations of structure anchor patches. To generate representations for the structure anchor patches, we first perform w fixed-length triangular random walks with parameter β to produce a sequence of traversed nodes $(u_{\pi_w(1)}, \dots, u_{\pi_w(n)})$ (See Algorithm 2 and Appendix D.1.1). The resulting sequence of node embeddings is then fed into a bidirectional LSTM, whose hidden states are summed to produce a final representation, $\mathbf{p}$. To capture the internal structure of the anchor patch, we perform random walks over $\mathcal{I}$, defined as $\{u | u \in A_s\}$. Separately, to capture the border structure of the anchor patch, we perform random walks over the external neighborhood $\mathcal{E} = \{v | v \notin A_s\}$ and the

border nodes $\mathcal{B} = \{u | u \in \mathcal{I}, v \in \mathcal{E}, e_{uv} \in E\}$. We limit $\mathcal{E}$ to the subset of external nodes within k hops of any node in $\mathcal{I}$.

Algorithm 2: STRUCTURE ANCHOR PATCH EMBEDDER.

Input: Graph G ; Anchor patch A ; node representations $\{\mathbf{x}_u | u \in A\}$; triangle random walk (TRIANGLE_RW) parameter β ; number of walks M
Output: A patch representation $\mathbf{p}$ for anchor patch A
for $m = 1, \dots, M$ **do**
 $(u_{\pi_w(1)}, \dots, u_{\pi_w(n)}) = \text{TRIANGLE_RW}(G, A, \alpha, \beta)$
 $\mathbf{h}_w = \text{BI-LSTM}((\mathbf{x}_{\pi_w(1)}, \dots, \mathbf{x}_{\pi_w(n)}))$ where $\mathbf{x}_{\pi_w(i)}$ is representation for node $u_{\pi_w(i)}$
end
 $\mathbf{p} = \sum_{w=1, \dots, M} \mathbf{h}_w$

ESTIMATING SIMILARITY OF ANCHOR PATCHES The similarity function $\gamma_x : (S^{(C)}, A_x) \rightarrow [0, 1]$ determines the relative weighting of each anchor patch in building the subgraph component representations. In principle, γ can be any learned or predefined similarity function. For the position channel, we define γ_p as a function of the shortest path on the graph G between the connected component $S^{(C)}$ and the anchor patch A_p : $\gamma_p(S^{(C)}, A_p) = 1/(d_{sp}(S^{(C)}, A_p) + 1)$, where $d_{sp}(S^{(C)}, A_p) = \text{SP}(S^{(C)}, A_p)$. $\gamma_n(S^{(C)}, A_n)$ is defined similarly, though note that, due to the sampling schemes, this evaluates to $\gamma_{N_I} = 1$ in the case of internal neighborhood and $\gamma_{N_B} \leq k$ in context of the k -hop border neighborhood sampler. To measure subgraph structure in a permutation-invariant manner, we compute a canonical input ordering²¹⁸, defined by ordered degree sequence, for the component $S^{(C)}$ and anchor patch A_s . These are compared using the normalized Dynamic Time Warping (DTW) metric. Specifically, we define $\gamma_s(S^{(C)}, A_s) = 1/(\text{DTW}(d_{S^{(C)}}, d_{A_s}) + 1)$, where $d_{S^{(C)}}$ and d_A are the ordered degree sequences for the subgraph component and anchor patch, respectively. The degree for each node $u \in S^{(C)}$ or A is defined for internal structure as the number of edges from u to any node $v \in \mathcal{I}$, and for border structure as the number of edges from u to any

node $v \in \mathcal{E}$.

1.4.3 IMPLEMENTATION AND MODEL EXTENSIONS

Below we outline the computational complexity of our implementation of SUB-GNN and describe extensions to the model. Details on hyperparameters and the training setup are found in the appendix.

COMPUTATIONAL COMPLEXITY The complexity of SUB-GNN scales as a function of the number and size of the subgraphs. Notably, a fixed number of anchor patches are sampled, which upper bounds the memory and time complexity of the message passing algorithm. Structural anchor patch representations are generated via fixed-length random walks and are only calculated for a small number of anchor patches shared across all subgraphs. We precomputed similarities to speed up hyperparameter optimization; similarity calculation run time could be improved further through approximation methods such as locality sensitive hashing or k-hop shortest path distances.

MODEL EXTENSIONS Our implementation of SUB-GNN initializes component embedding with pretrained node embeddings, which could alternatively be learned end-to-end. While we specify similarity functions for each channel, we note that γ can be replaced with any function measuring the relevance of anchor patch $\mathcal{A}$ to subgraph component $S^{(C)}$, including learn-able similarity functions. Finally, while this paper focuses on subgraph classification, the representation learning modules we present can be trivially extended to any unsupervised, semi-supervised, or other prediction tasks, simply by modifying the loss functions. SUB-GNN can also be integrated into other frameworks as a sub-module for end-to-end training.

1.5 EXPERIMENTS

We first describe our datasets, the baseline approaches, and the experimental setup. We then present experiments on four synthetic and four real-world datasets for subgraph classification.

SYNTHETIC DATASETS We construct four datasets with subgraphs of specific graph properties such that binned values of the corresponding metrics act as their labels: **DENSITY**, **CUT RATIO**, **CORENESS**, and **COMPONENT**. **DENSITY** challenges the ability of our methods to capture the internal structure of our subgraphs; **CUT RATIO**, border structure; **CORENESS**, defined by the average core number of the subgraph, tests border structure and position; and **COMPONENT**, the number of subgraph components, tests internal and external position. The resulting **DENSITY** and **CUT RATIO** datasets have 250 subgraphs of size 20; **CORENESS**, 221 subgraphs of size 20; and **COMPONENT**, 250 subgraphs, with 15 nodes per component. See Appendix D.1.2 for more information on the methods used to add subgraphs, and basic properties of the datasets.

REAL-WORLD DATASETS We create four real-world benchmark datasets from biology and technology domains. (1) **PPI-BP** is a dataset of human protein-protein interactions (PPI)³⁷⁷, where the 1,591 subgraphs are defined by the Biological Process Ontology from MSigDB, with an average size of 10.2 nodes per subgraph and 265.2 subgraphs across 6 labels³⁰¹. (2) **HPO-METAB** and (3) **HPO-NEURO** are datasets built from a knowledge graph of rare disease characteristics (*e.g.*, symptoms, causal genes) with subgraphs defined by the sets of presenting symptoms—described using Human Phenotype Ontology (HPO) vocabulary[?]—associated with a disease and labeled by the diagnosis category. **HPO-METAB** has 2,400 subgraphs defined by metabolic disorders, with an average size of 14.4 nodes and 400.0 subgraphs across 6 labels^{124?},²¹⁵. **HPO-NEURO** has 4,000 subgraphs defined by neurological disorders, with an average size of 14.8 nodes and 45.5 subgraphs across 10 labels (multilabel classification)^{124?}. (4) **EM-USER** is dataset of workout routines with 1,343 subgraphs

defined by the users’ workout history and labeled by gender, with an average size of 341.8 nodes and 671.5 subgraphs across 2 labels²²⁹. See Appendix D.1.2 for basic properties of the datasets.

ALTERNATIVE BASELINE APPROACHES We consider seven baseline methods that can be applied to training subgraph embeddings. (1) AVG computes the average of the pre-trained node embeddings for nodes in each subgraph. (2) MN-GIN and (3) MN-GAT train meta nodes (also known as virtual nodes or meta-nodes)^{106,144,179} for each subgraph alongside the pre-training of the node embeddings^{352,322}. Both (4) s2v-N and (5) s2v-S compute subgraph embeddings using Sub2Vec⁶, each separately preserving the neighborhood and structural properties, respectively. (6) s2v-NS concatenates the subgraph embeddings from s2v-N and s2v-S. Finally, (7) GC is a graph classification GNN (GIN)^{88,352} that aggregates node features from the pre-trained node embeddings to produce subgraph embeddings; for GC, each subgraph is treated as a standalone graph.

IMPLEMENTATION DETAILS For each dataset, we first train GIN³⁵² on link prediction to obtain node and meta-node embeddings for the base graph. Sub2vec subgraph embeddings are trained using the authors’ implementation⁶. For a fair comparison, the feedforward component of all methods, including our own, were implemented as a 3-layer feed forward network with ReLU nonlinear activation and dropout. The feedforward networks were implemented with and without trainable node and subgraph embeddings, and the best results were reported for each. See Appendix D.1.4 for more training details.

1.6 RESULTS

SYNTHETIC DATASETS Results are shown in Tables 1.2 and D.3. We find that our model, SUB-GNN, significantly outperforms all baselines by 75.4% and the strongest baseline by 17%, on average. Results illustrate relative strengths and weaknesses of baseline models. While the GC method

Table 1.2: Micro-F1 on synthetic datasets. Standard deviations are provided from runs with 10 random seeds. See Appendix D.1.3 for ROC scores.

Method	DENSITY	CUT RATIO	CORENESS	COMPONENT
SUB-GNN (Ours)	0.919 ± 0.016	0.629 ± 0.039	0.659 ± 0.092	0.958 ± 0.098
Node Averaging	0.429 ± 0.041	0.358 ± 0.055	0.530 ± 0.050	0.516 $\pm <0.001$
Meta Node (GIN)	0.442 ± 0.052	0.423 ± 0.057	0.611 ± 0.050	0.784 ± 0.046
Meta Node (GAT)	0.690 ± 0.021	0.284 ± 0.052	0.519 ± 0.076	0.935 $\pm <0.001$
Sub2Vec Neighborhood	0.345 ± 0.066	0.339 ± 0.058	0.381 ± 0.047	0.568 ± 0.039
Sub2Vec Structure	0.339 ± 0.036	0.345 ± 0.121	0.404 ± 0.097	0.510 ± 0.013
Sub2Vec N & S Concat	0.352 ± 0.071	0.303 ± 0.062	0.356 ± 0.050	0.568 ± 0.021
Graph-level GNN	0.816 ± 0.068	0.377 ± 0.058	0.419 ± 0.070	0.526 ± 0.081

Table 1.3: Micro F1 on real-world datasets. Standard deviations are provided from runs with 10 random seeds. See Appendix D.1.3 for ROC scores.

Method	PPI-BP	HPO-NEURO	HPO-METAB	EM-USER
SUB-GNN (Ours)	0.324 ± 0.013	0.632 ± 0.010	0.537 ± 0.023	0.751 ± 0.021
Node Averaging	0.289 ± 0.043	0.490 ± 0.059	0.443 ± 0.063	0.744 ± 0.086
Meta Node (GIN)	0.277 ± 0.040	0.233 ± 0.086	0.151 ± 0.073	0.550 ± 0.025
Meta Node (GAT)	0.308 ± 0.032	0.259 ± 0.063	0.138 ± 0.034	0.536 ± 0.047
Sub2Vec Neighborhood	0.309 ± 0.023	0.211 ± 0.068	0.132 ± 0.047	0.503 ± 0.035
Sub2Vec Structure	0.307 ± 0.013	0.223 ± 0.065	0.124 ± 0.025	0.742 ± 0.023
Sub2Vec N & S Concat	0.295 ± 0.011	0.206 ± 0.073	0.114 ± 0.021	0.536 ± 0.047
Graph-level GNN	0.291 ± 0.026	0.577 ± 0.015	0.480 ± 0.026	0.505 ± 0.041

performs quite well on DENSITY (internal structure), as expected, it performs poorly on datasets requiring a notion of position or border connectivity. The MN baselines, in turn, performed best on COMPONENT, presumably because the meta-node allows for explicit connection across subgraph components. Sub2Vec, the most directly related prior work, notably did not perform well on any task. This is unsurprising, as sub2vec does not explicitly encode position or border structure.

CHANNEL ABLATION To investigate the ability of our channels to encode their intended properties, we performed an ablation analysis over the channels (Table 1.4). Results show that performance

of individual channels aligns closely with their inductive biases; for example, the structure channel performs best on CUT RATIO (a border structure task) DENSITY (an internal structure task), and the position channel performs best on COMPONENTS (an internal position task). The CORENESS task primarily measures internal structure but also incorporates notions of border position; we find that the structure channel and combined channels perform best on this task.

Table 1.4: Channel ablation analyses (Micro F1). Channels that encode properties relevant to each dataset have a ✓ and best performing channels are in **bold**. We see in each case that the channels designed to encode relevant properties yield the best performance on each dataset. This suggests that the Sub-GNN channels successfully encode their desired properties.

SUB-GNN Channel	DENSITY	CUT RATIO	CORENESS	COMPONENT
Position (p)	0.758 $\pm$ 0.046	0.516 $\pm$ 0.083	0.581 $\pm$ 0.044 ✓	0.958 $\pm$ 0.098 ✓
Neighborhood (n)	0.777 $\pm$ 0.057	0.313 $\pm$ 0.087	0.485 $\pm$ 0.075	0.823 $\pm$ 0.089
Structure (s)	0.919 $\pm$ 0.016 ✓	0.629 $\pm$ 0.039 ✓	0.663 $\pm$ 0.058 ✓	0.600 $\pm$ 0.170
All (p+n+s)	0.894 $\pm$ 0.025	0.458 $\pm$ 0.101	0.659 $\pm$ 0.092	0.726 $\pm$ 0.120

REAL-WORLD DATASETS Results are shown in Tables 1.3 and D.4. We find that our model performs strongly on real-world datasets, outperforming baselines by 100.6% on average and performs 6.9% better than the strongest baseline. SUB-GNN performed especially well relative to baselines on the HPO-NEURO and HPO-METAB tasks. These tasks are uniquely challenging, because they require distinguishing subcategories of similar diseases (a challenge for averaging-based methods), exhibit class distributional shift between train and test, and have been designed to require inductive inference to nearby phenotypes using edges in the graph. Strong performance of SUB-GNN on these tasks thus suggests the ability of our model to leverage its relational inductive biases for more robust generalization. Further, our model outperformed even the strongest baselines on EM-USER, even though we speculate that EM-USER is largely a node identity-driven task. Our model outperformed all baselines on the PPI-BP task, however this task was particularly challenging for all

methods. Taken together, all of our new datasets are non-trivial, and present unique challenges that highlight the relative strengths and weakness of existing methods in the field.

1.7 CONCLUSION

We present SUB-GNN, a novel method for subgraph representation learning and classification, along with a framework outlining the key representational challenges facing the field. SUB-GNN performs subgraph level message passing with property-aware channels. We present eight new datasets (4 synthetic, 4 real-world) representing a diverse collection of tasks that we hope will stimulate new subgraph representation learning research. SUB-GNN outperforms baselines by an average of 75.4% on synthetic datasets and 100.6% on real-world datasets.

*Doctors are men who prescribe medicines of which they
know little, to cure diseases of which they know less, in
human beings of whom they know nothing.*

Voltaire

2

Cross-Modal Representational Alignment of Transcriptional Profiles and Small Molecule Therapeutics

2.1 INTRODUCTION

Identifying molecules that are likely to have a specific biological effect is a cornerstone of drug discovery and a key component of efforts to achieve precision medicine.

Classically, computational prioritization of small molecules for new indications has centered on predicting molecular affinities for specific biological targets. To this end, computational chemistry has developed a rich toolkit of methods, ranging from biophysics-driven techniques such as molecular docking¹²³ to literature-mined annotations¹⁶⁸.

More recently, small molecule modeling has also become a major area of interest for the deep learning community, a trend largely catalyzed by the advent of graph neural networks²¹. Graph neural networks allow for end-to-end learning of predictive models from raw molecular graphs^{156,356,188,265}. This has resulted in state-of-the-art performance on certain tasks, even falling below the margin-of-error of gold-standard assays for certain biophysical properties^{83,106}. As is frequently the case, the machine learning community has also been attracted to chemistry by the availability of benchmark datasets, the most prominent of which is the MoleculeNet dataset³⁴⁸. MoleculeNet and other similar datasets are predominantly focused on specific chemical properties, but do include several clinical tasks such as indication or side effect prediction. Missing from the field, however, are benchmark tasks between these two extremes, that test the ability of deep chemical fingerprints to encode rich and more general representations of a molecule’s effect on cellular biology.

In parallel to these developments in machine learning for chemistry, connectivity mapping (cmapping) has emerged as an alternative approach for drug development²²⁰. In connectivity mapping, compounds are foremost characterized not by individual chemical properties or downstream targets, but by the broad transcriptional effects they induce in cells. More specifically, connectivity mapping begins by first developing a large dataset of *perturbational signatures* of various molecules, by physically treated cell lines and measuring the resultant changes in gene expression.

These datasets are then compared to one or more *query signatures*, which are typically differential gene expression (GE) signatures representing disease states that the investigators hope to reverse or mimic. Organizations such as the Broad Institute and the Library of Integrated Network-based Cellular Signatures (LINCS) Consortium have enabled the production of public datasets^{174,300} of more than a million perturbational signatures. Fueled by these data, researchers have sought to match existing drugs in an unbiased manner to new indications and even to specific patients^{221,187,56,73,71,264,53}.

Connectivity mapping is promising because it can be used to search for new indications of drugs without making any specific *a priori* assumptions about their mechanism of action. However, the typical framework for connectivity mapping is limited by the fact that it can only query against drugs that have already been profiled using the transcriptional assay. In other words, connectivity mapping is – in the abstract* – very flexible with respect to the disease signatures they accept as a query, but are transductive rather than inductive with respect to the target small molecule signatures. This is the perfect complement to structure-based computational chemistry, which is typically inductive to new drug structures but can only make predictions for diseases with known targets.

In this chapter, I explore the possibility of combining these two fields, by using deep chemical embedders to learn the transcriptional space encoded by CMAP profiling. More specifically, I train coordinated networks such that learned chemical representations are most similar to the encodings of the transcriptional patterns they induce. I then evaluate these chemical representations by using gene expression signatures as queries into the embedding space and recovering their corresponding compounds. (See Figure 2.1). Crucially, the evaluation is set up such that the compounds in the validation and test sets were never exposed to the model during training, which allows me to test the

*In practice, connectivity mapping relies upon the non-trivial assumption that drug responses *in vitro* (typically in cancer cell lines) are representative of drug responses *in vivo* in the tissue affected by the disease. This is admittedly a highly nontrivial assumption, which I show below appears may be the rate-limiting obstacle to this method's success.

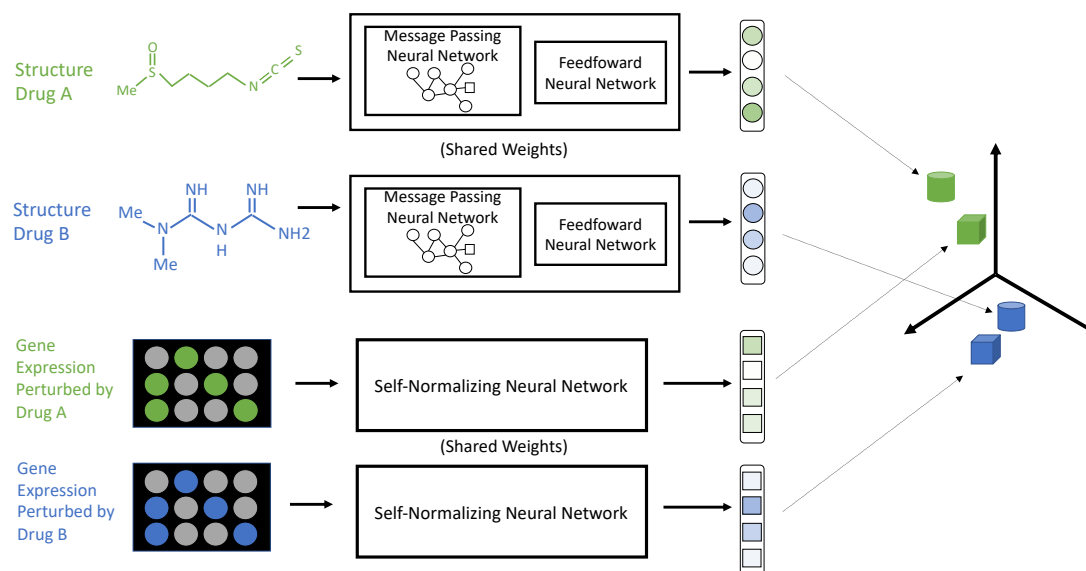


Figure 2.1: Overview of our coordinated representation learning method. Neural networks are jointly trained to embed gene expression profiles and small molecule structures such that transcriptional signatures are brought into close proximity with their corresponding perturbagens. Given a cross-modal alignment, gene expression signatures can then be used as queries to rank chemical structures by their likelihood to induce such a signature.

ability of the model to extrapolate the broad functional properties of drugs it has never seen.

The aim of this project is two-fold: First, to provide a new benchmark task to assess the ability of learned chemical representations to capture broad (here: transcriptome-wide) functional properties of drugs. Second, to provide a potential proof-of-concept for a connectivity mapping strategy that can be applied to unassayed drugs.

In the rest of this work, I first offer some background on joint embedding alignment, then detail the methods used in the creation of this task as well as the experiments we performed. Our experiments include the implementation of baseline models of increasing complexity, oracle models designed to estimate reasonable upper-bound performance, and introspection studies chosen to elucidate the changes in molecular representations induced by the training task. Finally, I walk through

the results and discussion of these experiments, then close with concluding thoughts.

2.2 BACKGROUND: MULTI-VIEW REPRESENTATION ALIGNMENT

Multi-view representation learning seeks to learn representations that relate information from multiple views of the same data, such as an image of a scene and a text description of that same scene. In *multi-view representation alignment*, representations of each data modality are learned separately but in a coordinated manner, such that different views of the same scene are induced to be similar per some metric. In this work, we attempt to learn aligned representations of gene expression perturbational signatures and small molecules’ structures, such that differential gene expressions are embedded in close proximity to the small molecules that induced them.

Multi-view representation alignment can be achieved through a variety of methods, including distance-based, similarity-based, and correlation-based alignment. One classic method for representation alignment, which we leverage as a baseline in this paper, is *canonical correlation analysis (CCA)*¹³⁶. CCA takes a pair of datasets $X = [x_1 \dots x_n]$ and $Y = [y_1 \dots y_m]$ and seeks to identify vectors $a \in \mathcal{R}^n$ and $b \in \mathcal{R}^m$ such that $(a', b') = \arg \max_{a,b} \text{corr}(a^T X, b^T Y)$. The resultant random variables $U = a^T X$ and $V = b^T Y$ are the first pair of canonical variables. This process can be iteratively repeated up to $\min(m, n)$, times with the restriction that each subsequent pair of canonical variables are uncorrelated with the previous pairs. Taken together, the procedure produces a linear mapping from the datasets X and Y that maximizes the element-wise correlation between the representations of paired rows from the two datasets.

A wide range of additional methods for multi-modal alignment have been proposed, many of the most successful of which have been rank-based methods that involve contrastive learning^{183,68}. Such methods are intertwined with a broader literature of metric learning, which pioneered such architectures as Siamese^{164,89} and Triplet networks¹³⁵. These networks optimize representations to

bring similar data together while driving dissimilar data apart. An analysis of best practices working with this architecture can be found in Wu et al ³⁴⁴. Recent analyses suggest that despite the proposals of many sophisticated architectures in the past several years, few if any of these appear to outperform triplet networks when comparisons are fair ²²².

2.3 METHODS

2.3.1 DATASET PREPARATION

DATA ACQUISITION AND SUBSETTING

All raw data in this study comes from the Next-Generation Connectivity Map (CMap) provided by the LINCS Consortium and the NIH ³⁰⁰. Analyses were conducted using the so-called “Level 3” L1000 data as described in Subramanian et al.[†] The raw data comes in the form of gene expression profiles generated using the L1000 technology, a low-cost assay which measures 978 ‘landmark’ genes that have been deemed representative of the full transcriptome. The CMap contains data from a variety of chemical and genetic perturbations applied to a range of human cell lines, along with several control samples per plate treated with buffer. In order to ensure we had sufficient support over possible drugs within our dataset, our primary cut of these data uses the most frequent 8 cell lines, split into a train, validation, and test set split by *both*[‡] drug and cell line (e.g., no cell line in the train set appears in the validation or test sets, nor does any drug in the train set appear in ei-

[†]Available for download at <https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE70138> and <https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE92742>

[‡]Given that this is primarily a chemical embedding task, the rationale behind the drug-level split is obvious. The reason we opted to also include the cell-line split was several-fold: First, we wanted to maintain fidelity to the original cmapping task, which inherently assumes that signatures will generalize outside the incubation conditions. We likewise want to incentivize our chemical embedders to find broadly representative features of each drug, and thus didn’t want to overfit to a specific cell line. Finally, our gene expression embedder is provided with the raw untreated gene expression signatures; given that these are all cell lines and in a dish (and thus pharmacokinetics is not a concern), many of the key differences between conditions should be encoded in the baseline signatures themselves.

	Train	Validation	Test
Cell Lines	HEPG2, HA1E, HCC515, VCAP, A375, PC3, MCF7	A549	HT29
# Samples	97210	1255	2317
# Drugs	3490	436	437

Table 2.1: Dataset description, detailing cell lines, number of samples, and number of drugs per split.

ther the validation or test sets). To avoid non-random missingness, we included only drugs that had been assayed in all cell lines, and limited experiments to those incubated with small molecules for 24 hours at a dose of $10\mu\text{m}$. Final statistics of these data cuts, including which cell lines were used, how many drugs were represented, and the relative sampling rates of various drugs, are shown in Table 2.1. For drug structures, we used the SMILES³³⁶ structures provided by LINCS, which we canonicalized to a consistent format using RDKit¹⁷⁵.

PREPROCESSING AND FEATURE ENGINEERING

Gene expression intensity values from the training, validation, and test sets were centered and scaled at the gene-level based on the mean and standard deviation of each gene intensity across the training set. For each gene expression profile representing a chemical perturbation, we then computed four core sets of features to power our baselines and deep representations: the post-perturbational gene expression itself, corresponding gene expression intensities from a control signature on the same plate, the $\log_2$ fold-change between the perturbation and control signatures, and the difference between these gene expression signatures. A *rank* transform, in which raw expression levels are replaced by their relative rank of over-expression, is commonly used in bioinformatics to make gene expression data more robust—we also experimented with using this transform, but found it to be uniformly unhelpful.

For use in our baseline and oracle models, we also computed two previously established nu-

merical representations of each small molecule. The first of these representations was the Morgan extended-connectivity fingerprints, a standard representation used in many computational chemistry applications²⁷¹. These were computed using RDKit. For a more modern representation, we used the chemical encoder from Yang et al, which recently reported state-of-the-art performance in predicting 128 biophysical properties from the PubChem Bioassay database^{356,331}.

2.3.2 BASELINE AND ORACLE METHODS

NEAREST NEIGHBOR BASELINE

Nearest-neighbor-based methods have been previously shown to be very strong baselines for machine learning tasks on the L1000 data^{134,202}. The standard nearest neighbor algorithm is not defined for this particular task, however, because the train, val, and test sets have been defined as disjoint with respect to both chemical structures and cell line. We thus employ a two-phase nearest neighbor search: Given a gene expression profile as a query, we first identify the nearest gene expression profile in the train set and look up its corresponding small molecule. We then take this small molecule (from the train set) as a query, and return the most structurally similar drug from the *test* set as our final prediction. For the rest of the paper, we call this two-phase approach the “double nearest neighbor” baseline.

We now provide a more formal definition of our double nearest neighbor baseline. We are provided gene expression signatures $\mathcal{G}_{\text{train}}$ and $\mathcal{G}_{\text{test}}$, computed small molecule encodings $\mathcal{M}_{\text{train}}$ and $\mathcal{M}_{\text{test}}$, and a mapping $G2M : \mathcal{G}_{\text{train}} \rightarrow \mathcal{M}_{\text{train}}$ from training gene expression profiles to the small molecule that induced them. We then define transcriptional and chemical similarity functions $Sim_g : \mathcal{G} \times \mathcal{G} \rightarrow \mathcal{R}$ and $Sim_m : \mathcal{M} \times \mathcal{M} \rightarrow \mathcal{R}$ – each typically but not necessarily within the range $[-1, 1]$ or $[0, 1]$, with 1 corresponding to identical molecules or gene expressions. Given some

gene expression query g_{query} , we then rank all molecular structures $m_{test} \in \mathcal{M}_{test}$ according to

$$Score_{NN} = Sim_m \left(G2M \left(\arg \max_{g_{train} \in \mathcal{G}_{train}} Sim_g (g_{query}, g_{train}) \right), m_{test} \right) \quad (2.1)$$

This double-nearest neighbor method is general in the sense that it can be implemented using any numerical representation of molecular structures $\mathcal{M}_*$ and gene expression signatures $\mathcal{G}_*$, and likewise using any number of similarity functions Sim_* . In this paper, we report results encoding molecules as each of: Extended Connectivity Fingerprints, Chemprop Embeddings, or the Aligned chemical embeddings generated by our joint modeling procedure described in section 2.3.3. Gene expression signatures are encoded as either raw transcriptional signatures or from our Aligned gene expression representations. We use Pearson Correlation for both Sim_g and Sim_m in all reported experiments. All computations were implemented using NumPy²³⁵.

CANONICAL CORRELATION ANALYSIS BASELINE

As described in Background, the canonical correlation procedure takes in a pair of data matrices (of potentially different dimensions) and produces a corresponding pair of linear mappings that maximizes the element-wise correlation between the mapped data matrices. This lends itself to a supervised baseline for this task, as follows:

Given training matrices of transcriptional $\mathcal{G}_{train}$ and molecular $\mathcal{G}_{test}$ encodings, and some gene expression query g_{query} , we can rank all molecular structures $m_{test} \in \mathcal{M}_{test}$ according to

$$Score_{CCA} = \text{Corr}(A \cdot g_{query}, B \cdot m_{test}) \quad (2.2)$$

, where $(A, B) = CCA(G_{train}, M_{train})$. Due to the properties of CCA, such a procedure defines an optimal *linear* alignment between drug and transcriptions encodings in the training set. As with the nearest neighbors baseline, the CCA baseline can be run using any numerical representation of

transcriptional signatures and chemical structures. We thus computed CCA baseline results using all the same combinations of input encodings as were employed by the double nearest neighbor approach.

Canonical Correlation Analysis was performed using SciKit Learn²⁴⁴. Default settings from SciKit Learn’s CCA function were used, with two exceptions: first, the number of components was set to 50. The number 50 was chosen because a preliminary benchmark ranging from 5-125 components showed that 50 components had mildly superior performance in retrieval tasks on the validation set. Second, the maximum number of iterations was increased to 1000, since some initial experiments failed to converge at the default setting of 500.

ORACLE MODELS

As described above, the central objective of this task is to learn small molecule representations that can stand in as surrogates for their corresponding gene expression signatures. To provide a rough upper-bound for expected performance on this task, we also implemented “oracle” models that queried GE signatures against the true average gene expression signatures from cells treated with each drug.

Recall that the data was subsetting such that all small molecules had been measured in all cell lines, but also recall that the dataset was split into train, validation, and test sets using *both* cell line and small molecules. This means that for any gene expression signature serving as a query in the test set (comprised of experiments on cell line HT29), there were additional experiments from the same drug that had been run on the train set cell lines but which had been excluded from the train set. This lends itself to two types of oracle models:

First, we simulate an oracle model by querying test set GE signatures against pseudo-“chemical embeddings” that are in reality the average GE signatures from each test set drug when it was measured on *train set cell lines*. This simulates a chemical embedder that perfectly generalizes to new

chemical structures, but not necessarily to the new cell line.

Second, we simulate an oracle model by querying against the average GE signatures for each drug from all experiments in the *test set cell lines* – excluding the query GE signature itself. This oracle model simulates near-perfect (average) generalization of the chemical embedder across cell lines as well as structure.

Importantly, we note that while these oracle models simulate perfect alignment between the chemical embedder and the gene expression signatures, they do not necessarily represent a true upper bound on this task. This is because these oracle models still require some encoding of the gene expression signatures, which may not be the most generalizable representation of the GE data. Accordingly, we compute the performance for each of the above two oracle models using both raw GE signatures and the deep GE signatures learned through our joint models. This was done to identify the extent to which further innovation in GE embedding may be required for improved performance on this task.

2.3.3 DEEP COORDINATED REPRESENTATION LEARNING APPROACH

We designed a cross-modal ranking algorithm to learn representations of chemical structure and gene expression that would each be jointly optimized to facilitate small molecule retrieval from gene expression signatures.

An overview of this algorithm is depicted in Figure 1. In brief, gene expression profiles and drug structures are each embedded through separate neural networks into a shared embedding space. These networks are optimized jointly to produce representations that minimize the distance in the embedding space between gene expression signatures and the drugs that induced them, while maximizing the distance between GE signatures and the representations of other structures.

More formally, given a collection of gene expression signatures $\mathcal{G}_{train}$ and chemical structures $\mathcal{M}_{train}$, and some similarity function $Sim : \mathcal{R}^d \times \mathcal{R}^d \rightarrow \mathcal{R}$, we seek to learn a GE embedder

$E_g : \mathcal{G} \rightarrow \mathcal{R}^d$ and chemical embedder $E_m : \mathcal{M} \rightarrow \mathcal{R}^d$ to maximize $\text{Sim}(E_g(g_i), E_m(m_j))$ where gene expression g_i was induced by molecule m_j . Given our trained embedders and some gene expression query g_{query} , we can then rank all molecular structures $m_{\text{test}} \in \mathcal{M}_{\text{test}}$ according to

$$\text{Score}_{\text{aligned}} = \text{Corr}(E_g(g_{\text{query}}), E_m(m_{\text{test}})). \quad (2.3)$$

We parameterize E_g and E_m using jointly trained neural networks. Below we describe the components of these networks and their training in detail.

GENE EXPRESSION AND DRUG EMBEDDERS

Gene expressions embedders E_g were parameterized using self-normalizing neural networks, which are feedforward neural networks that employ the SeLU activation function¹⁶³. The depth and shape of the hidden layers was tuned with other hyperparameters (see section 2.3.3). In addition, we employed AlphaDropout for regularization on the hidden layers, which was also tuned. No activation function or dropout was used before the final embedding layer. A natural source of data augmentation was employed by randomly sampling only one control-plate signature on each iteration of the algorithm, which meant that every epoch saw a slightly different featurization of every sample. A histogram of the number of controls per plate can be found in the Appendix.

Chemical structure embedders E_m were parameterized using a directed message-passing neural network (D-MPNN) followed by a feedforward neural network. This architecture was initialized using the same D-MPNN from Yang et al³⁵⁶ – ChemProp – which was described above in context of the CCA and nearest-neighbors baselines. The weights of the D-MPNN were fine-tuned in tandem with the training of our feedforward layers. The number and size of the feedforward layers coming out of the D-MPNN were tuned as hyperparameters in our architecture. By contrast, the architecture of the MPN component of the architecture was fixed in accordance with the optimal

hyperparameters found in Yang et al.³⁵⁶. Finally, the dimension of the gene expression and drug structure representations was also set as a hyperparameter.

LOSS FUNCTIONS AND SAMPLING SCHEME

We developed customized margin-based triplet and quadruplet losses for our task. As described in Wu et al.³⁴⁴, an adaptive margin loss for embedding learning can be defined over two data points i and j as

$$\ell^{\text{margin}} := (\alpha + y_{i,j}(D_{ij} - \beta))_+,$$

where D is distance function (here Euclidean distance), α defines a permissible margin of separation, β controls the boundary between positive and negative pairs, and $y_{i,j}$ is an indicator variable equal to 1 if i and j are of the same class and 0 otherwise. We allowed the system to tune α and β with other hyperparameters.

Given this margin loss, we first implemented cross-modal triplet networks^{135,68} to align gene expression representations to molecular structure representations. In these networks, we provide the loss function with an *anchor* embedding of one modality and with one *positive* (matching) and one *negative* (non-matching) embedding of the other modality. We then define the triplet loss as the sum of the margin losses between (a) the anchor and the positive embedding, and (b) the anchor and the negative embedding. The network is thus optimized to bring the positive embedding within the margin of the anchor and negative embedding outside the margin. To give an example: if the anchor is a drug structure, the positive example is a gene expression signature that was induced by that drug structure, and the negative example is a gene expression signature induced by some other molecular structure. The loss function then tries to force the gene expression and its corresponding small molecule structure toward each other in embedding space and away from the other drug embedding. Throughout this paper, we refer to our triplet networks as “Gene Expression-

Anchored” if their anchor embeddings are gene expressions and their positive/negative embeddings are drug structures, and as “Chemical Structure-Anchored” otherwise.

We also implemented a quadruplet design that combined the so-called “Gene expression-Anchored” and “Chemical Structure-Anchored” losses. In the quadruplet architecture, we sample a gene expression signature as anchor, positive and negative small molecule structures, and an additional gene expression signature induced by the negative small molecule structure. We then compute both the “Gene expression-anchored” triplet margin loss using the anchor gene expression signature and two molecular structures, as well as the “Chemical structure-anchored” margin loss using the first structure embedding as an anchor and the two gene expression embeddings as positive and negative examples. These two margin-based triplet losses are then summed together to form the quadruplet margin loss.

For sampling, we adapted the distance-weighted sampling scheme described in Wu et al, which was successful with their margin-based approach³⁴⁴. Since our goal is to learn to rank across modalities, we precomputed the average distance in *gene expression space* between every pair of small molecule structures in the dataset. This was performed by computing the average of the post-perturbational signatures induced by each small molecule and computing their pairwise correlations. These pairwise correlations were then used to guide distance-weighted uniform sampling among potential negative perturbagens.

TRAINING AND HYPERPARAMETER SELECTION

Each model was trained on a single Nvidia GeForce GTX 1080 GPU. Early stopping was used to select the model with the best mean reciprocal rank on the validation set. Hyperparameter tuning via the Bayesian Hyperopt library³¹ was performed over a wide range of possible hyperparameters, including network depth and width (parameterized by the size of the first hidden layer and a growth

rate), learning rate, total number of epochs[§], batch size, margin and β parameters, triplet model orientation (e.g., gene first or compound first), activation function/network type (e.g., SNNN vs. unconstrained fully connected network), and dropout.

2.3.4 EXPERIMENTS

We designed a range of experiments with two primary purposes: First, we evaluate our proposed chemical alignment method on this task, benchmarked below by baseline methods and above by our oracle models. Second, we introspect into the resultant models trained over this task, in order to demonstrate that the task induces meaningful representations that encode clinically-relevant information.

QUANTITATIVE INFORMATION RETRIEVAL RESULTS

We evaluated all scoring strategies using the standard information retrieval metrics *median rank* (MR), *mean reciprocal rank* (MRR), and *Hits* or *Recall @ K*¹⁹⁸. In addition, we plot the empirical cumulative distribution function (E-CDF) of the ranks of the true perturbation among the results using all methods. This E-CDF plot is equivalent to a continuous-valued Hits @ K for all possible K in this dataset, and allows for a visual comparison between experiments.

PERFORMANCE ANALYSIS

In addition to the baseline and oracle analyses, we further examined the role of generalization in retrieval performance by comparing retrieval ranks for a given query to the distance from that query to the nearest neighbor in the training set. We examined these relationships both with respect to GE query signatures (computed both using raw GE profiles as well as embedded GE signatures) and for

[§]Hyperparameter tuning score was determined at the final epoch.

chemical structures to which the query corresponded (distance to the training set computed using Tanimoto distance using Molecular Fingerprints, and Pearson correlation using ChemProp and aligned representations).

We also hypothesized that a compound’s functional specificity may influence the difficulty of learning to encode its effect. To test this hypothesis, we first conducted a standard differential expression analysis for each compound, using the Limma package in R, and tabulating the total genes that are differentially expressed following treatment with each compound. Because the goal of this analysis was to get a simple measure of overall drug specificity and not to identify or report the effect on individual genes, we counted a gene as differentially expressed if Limma assigned it an unadjusted P-value less than 0.05.

EMBEDDING INTROSPECTION EXPERIMENTS

We were also interested in assessing the properties of the representations learned through our training task. In particular, we were most interested in the changes in the ChemProp latent space that were induced by our alignment training scheme. To measure this, we computed UMAP representations of the ChemProp and aligned representations of the small molecules in our training set²⁰³. Using metadata from the LINCS consortium, we then labeled compounds with their known mechanism of action, if any. This enabled us to plot the location in the UMAP latent space of the drugs with the three most common mechanisms of actions, using which we compared the latent spaces of the pretrained Chemprop and aligned representations. Our hypothesis was that even though both models had comparable expressive capacity (and, indeed, both have identical message passing architectures), fine-tuning the representations using our GE-drug alignment task might force the representations to encode more clinically-relevant features than when trained only on the protein-binding task.

Finally, we computed UMAP representations of each compound using the averaged GE signa-

tures from train and test, to examine where test set queries sit in relation to the training data in the latent transcriptional space. While all training set drugs were measured in 6 cell lines, there were a variable number of samples coming from each cell line-drug pair. To correct for cell line imbalances, we thus weighted the GE averages such that each cell line contributed equally to the overall average for each drug. Finally, in order to elucidate clusters in the latent space, additional UMAP projections were generated from the same data, but with raw GE measurements restricted to only those coming from control or perturbed cells.

2.3.5 SOURCE CODE

The source code developed for us in this project is freely available on Github at https://github.com/sgfin/molecule_ge_coordinated_embeddings.

2.4 RESULTS

2.4.1 QUANTITATIVE IR EXPERIMENTS

BASELINE AND PROPOSED METHOD. Information retrieval results from the baseline and proposed methods are reported in Figure 2.2A-C and Table 2.2, as well as in Figure D.2A-C. As can be seen, the numerical representation of the chemical structures has a larger impact on performance than any other factor. Strikingly, the largest increase in performance comes when introducing our alignment training – this is despite the fact that the Chemprop model is the core of the Aligned embedding architecture. In addition, while CCA seems to slightly outperform our nearest neighbors baseline given identical representations, both methods perform their best by far when using our aligned representations as their input. Of note, application of the CCA process to our representations yields a slight improvements in the results among early hits (as evidenced by MRR, H@10, the low recall region of Figure 2.2B). However, simple correlation-based retrieval yields slightly better

Table 2.2: IR metrics across different configurations of the model and baselines. MR=median rank, MRR=mean reciprocal rank, H@K=Hits (Recall) at K. ‘Chemprop’ refers to the unmodified model from Yang et al³⁵⁶. ‘Aligned’ indicates representations learned from our coordinated embedding architectures as described in Section 2.3.3.

GE	Chemical	Method	MR	MRR	H@10	H@100
Raw	Morgan FP	D-NN	206	0.025	0.037	0.240
Raw	Chemprop	D-NN	211	0.025	0.035	0.254
Raw	Aligned	D-NN	189	0.033	0.047	0.290
Aligned	Morgan FP	D-NN	214	0.025	0.041	0.256
Aligned	Chemprop	D-NN	196	0.022	0.037	0.278
Aligned	Aligned	D-NN	137	0.039	0.072	0.402
Raw	Morgan FP	CCA	180	0.027	0.045	0.303
Raw	Chemprop	CCA	184	0.024	0.040	0.294
Raw	Aligned	CCA	134	0.039	0.076	0.412
Aligned	Morgan FP	CCA	177	0.027	0.050	0.319
Aligned	Chemprop	CCA	163	0.028	0.051	0.334
Aligned	Aligned	CCA	130	0.048	0.093	0.425
Aligned	Aligned	Corr	126	0.042	0.085	0.432

results according to metrics that do not actively emphasize earlier results (MR, H@100, etc.). The nearest-neighbors method provides no benefit over simple correlation-based ranking when using our aligned representations.

ORACLE MODEL ANALYSIS. Results from the oracle models are reported in Figure D.2D and Table 2.3. As expected, oracle models using GE signatures from the test cell line greatly outperformed those using signatures from the train cell lines. In addition, aligned embeddings modestly improved the performance of test set oracles, and greatly improved the performance of training set oracles, consistent with the GE embedders learning slightly more generalizable representations. Of note, our proposed model achieved comparable results as the oracle model that leveraged raw GE signatures from the train cell lines. More specifically, our approach yielded slightly worse than the oracle on metrics (MRR, H@10) that emphasize early rankings, and slightly better on metrics (MR,

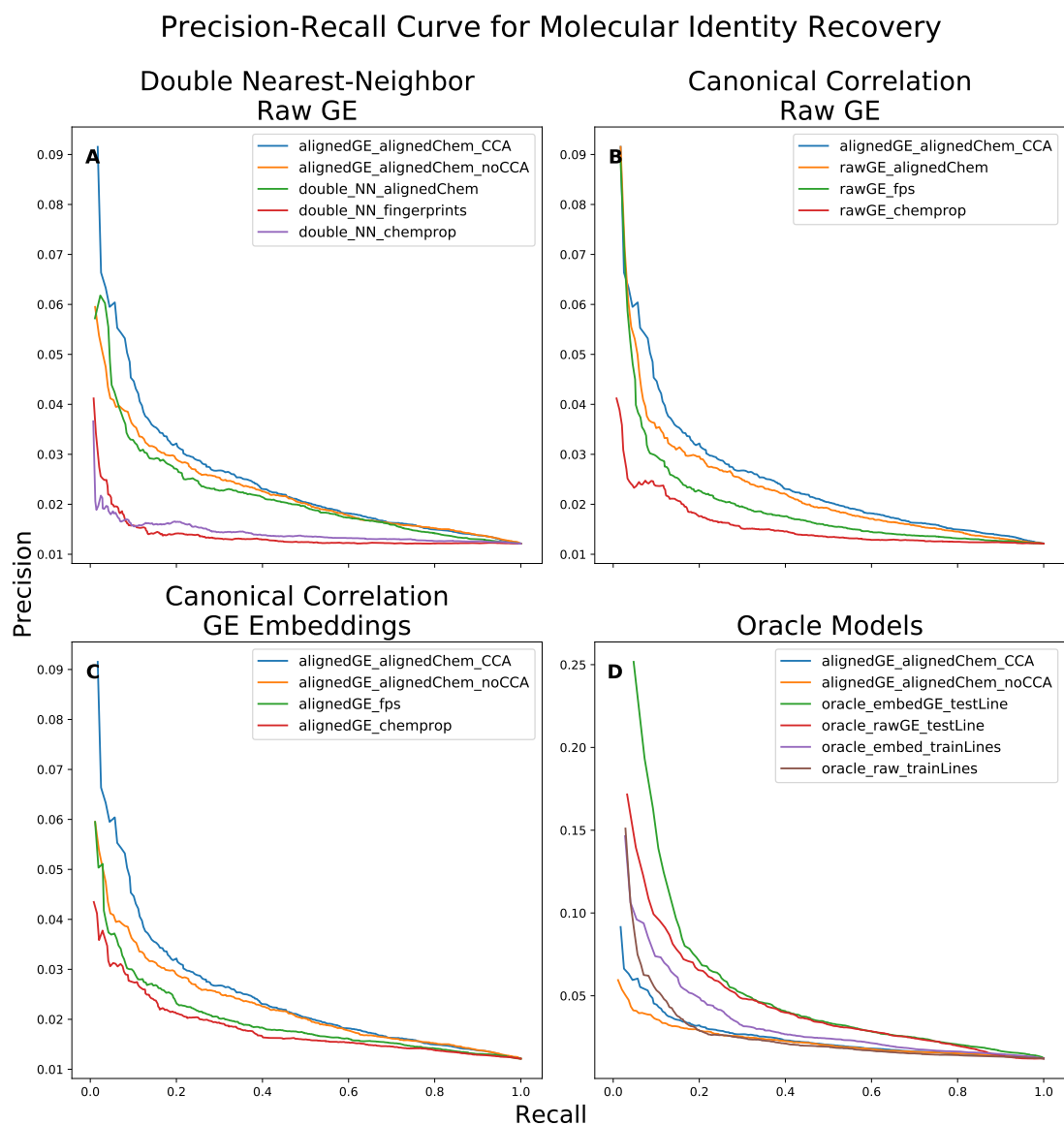


Figure 2.2: Precision Recall curves for drug identification given gene expression signatures. The “alignedGE_alignedChem_CCA” and “alignedGE_alignedChem_noCCA” curves, plotted in blue and orange, respectively, in all plots, represents our approach as described in Section 2.3.3 with and without CCA post-processing. All other methods represent baseline or oracle methods as described in Section 2.3.2. See Figure D.2 for ECDF plots of the same data.

Table 2.3: IR metrics for the various oracle methods. MR=median rank, MRR=mean reciprocal rank, H@K=Hits (Recall) at K.

Oracle Model	MR	MRR	H@10	H@100
Oracle - Train Cell Lines (Raw GE)	138	0.057	0.101	0.400
Oracle - Train Cell Lines (Embed)	110	0.064	0.128	0.466
Oracle - Test Cell Line (Raw GE)	82	0.076	0.147	0.565
Oracle - Test Cell Line (Embed)	79	0.093	0.160	0.568
Our Approach	126	0.042	0.085	0.432

H@100) that focus on more aggregate results. This is also apparent in the eCDF plot, which shows the aligned embeddings curve starting out slightly below that of the raw/train oracle curve but then moving rapidly above it as further results are considered.

2.4.2 PERFORMANCE ANALYSIS

PERFORMANCE AS A FUNCTION OF DISTANCE FROM THE TRAINING SET. Figures 2.3A and D.3 contain the results of our experiments comparing performance to distance from the training set. Regardless of the measure of chemical similarity, compound retrieval performance was inversely correlated with distance from the most similar compound in the training set. As can be seen in the Figure D.3, the same trend held with learned gene expression embeddings, and was present but much weaker for similarities measured using raw gene expression signatures. Finally, as can be seen in the figure in the appendix, we note that some molecules were found to have a distance of zero from the training set according to the molecular fingerprints, but not according to the chemical embedding methods. This is due to the fact that chemical fingerprints fail to encode specific details of molecular structure; examples of structures from the dataset that have identical molecular fingerprints but distinct ChemProp and Aligned Embeddings are shown in Figure D.4.

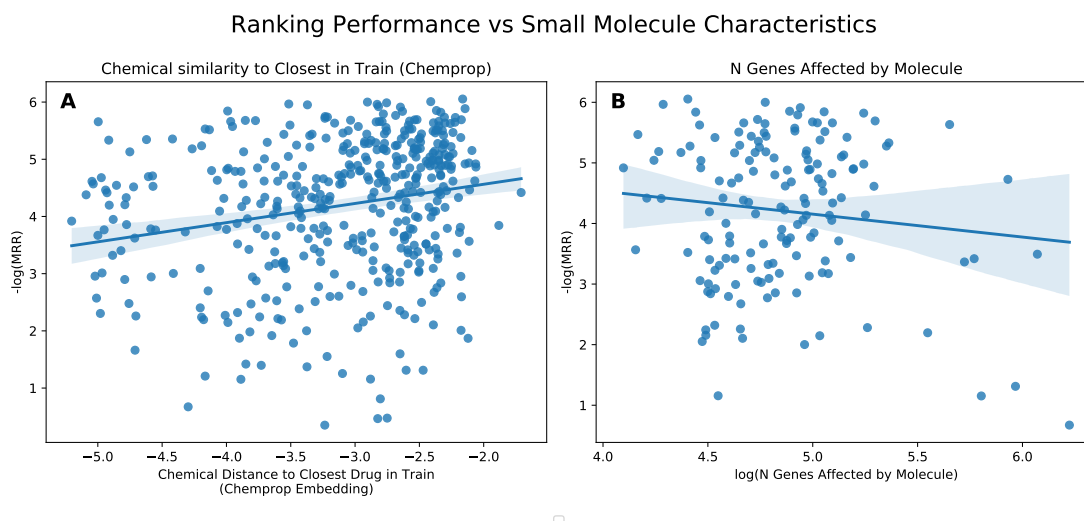


Figure 2.3: Left: Performance (lower is better) vs. Structural distance to the nearest compound in the training set. See Appendix Figure D.3 for analogous plots for four additional measures of distance from the training set. Right: Average Performance vs. # of Genes differentially expressed following treatment with the molecule of interest.

PERFORMANCE AS A FUNCTION OF DRUG SPECIFICITY. Figure 2.3B depicts the relationship between the transcriptional specificity of a compound and its ability to be retrieved using our analysis. As can be seen, there is a mild negative correlation, implying that molecules that affect the expression of many genes are easier to retrieve using this approach.

2.4.3 EMBEDDING INTROSPECTION ANALYSIS

INSPECTING THE CHEMICAL LATENT SPACE. Figure 2.4 depicts the chemical latent space from test set compounds, annotated with the top three mechanisms of action. As hypothesized, fine-tuning the Chemprop model via our alignment task appears to have forced drugs with similar mechanisms of action to cluster closer together. (Note: Future work with more richly clinically annotated datasets will be required to truly validate this hypothesis.)

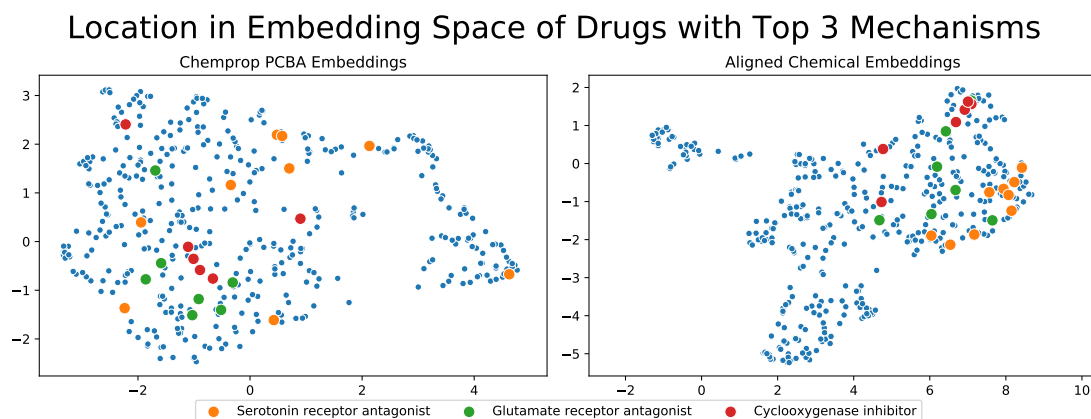


Figure 2.4: This figure shows our induced embeddings (right) compared to established chemical embeddings (left), projected to 2-dimensions using UMAP, colored according to MOA. (Blue dots represent small molecules with different or unknown MOAs.)

INSPECTING THE TRANSCRIPTIONAL LATENT SPACE. Figures 2.5, D.5, and D.6 depict the latent space of the raw gene expression signatures and learned gene expression embeddings, as visualized using UMAP. Data points each represent averages across all samples from one perturbation, weighted to balance across cell lines. As can be seen, well-separated clusters are present in the latent spaces of both raw and embedded gene expression signatures. However, the latent space from the embeddings exhibits clusters that are less spread out and in which the test set signatures appear to be more integrated into the train set signatures. There are three broad categories of potential causes: true clusters of drug function, clusters due to cell-line specific effects, or technical artifact. To try to disentangle these categories of sources, we first tried to tease out drug-specific effects by producing plots of the raw GE signatures from *only the control plates* matched to each perturbation experiment, as well as only the raw values of the perturbed cells. As can be seen in Figure D.5, the clusters in these plots are essentially identical to those of the full raw GE signatures, implying that the clustering in the raw gene expression plots were likely not due to true drug-related signal. Finally, in Figure D.6, we color points based on the cell line in which most of the samples had been originally measured, which aligns partially but not fully with the clusters, and more so in the raw GE space

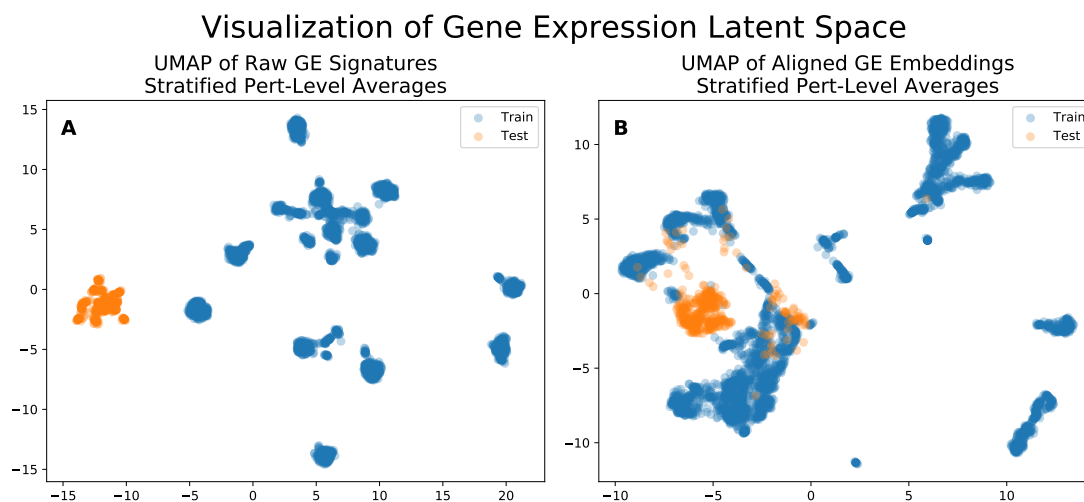


Figure 2.5: UMAP latent space of gene expression signatures, colored by train vs test set. Panel A: Raw gene expression signatures. Panel B: Aligned embedding representations. See also Figures D.5 and D.6, which demonstrate that visible clusters are likely due to non-drug effects, implying that the learned embeddings slightly improve robustness by weakening these effects.

than in the embedded GE space. Given that data points had been weighted to correct for cell line imbalances, it is likely that the remaining correlation with clusters is due to technical confounding (for example, if specific drugs were primarily assayed at specific sites or operators, which also primarily utilized specific cell lines). Taken together, the results indicate that clusters are likely due primarily to non-drug effects, and it is thus a positive sign that the embedding process served to weaken and otherwise shift these clusters.

2.5 DISCUSSION

2.5.1 OVERALL SUMMARY

In this chapter, we have sought to present a new biomedical machine learning task – cross-modal retrieval of molecular structure from perturbational signatures – as well as a set of approaches to benchmark future work on this task.

Our results indicate that the task appears feasible, and that the learned representations of GE and small molecules appear to exhibit stronger retrieval performance on this generalization task than do baselines. In addition, our latent space introspection analyses indicate that the aligned training task may result in more robust and biomedically meaningful representations.

However, even the strongest models studied provide only modest performance on this task overall. This is on one hand promising for the project as a new benchmark task, as it leaves significant room for improvement. On the other hand, our oracle model studies indicate that even direct querying against average test set signatures does not perform extremely well. This indicates that the data may be inherently noisy and/or that more sophisticated representations of gene expression may be necessary to raise the upper-bound performance on this task. Along the same lines, we observe that the gap between our chemical embedding and the train-set oracle was far narrower than the gap between the train-set oracle and the test-set oracle. This indicates that – at least in this context – the problem of generalizing drug activity across *cell types* may be at least as daunting as the generalization problem across new *drug structures*. This raises questions not simply for our task, but also for classical connectivity mapping, which takes as its primary assumption that drug signatures will generalize to not just other cell lines but to humans *in vivo*.

2.5.2 FUTURE WORK

From the perspective of task formulation, one area of future interest is the development of custom metrics to score model ranks. As it stands, the evaluation strategy counts the exact perturbagen as correct, and all other small molecules as equally incorrect. This will potentially under-estimate model performance, as it ignores the possibility of other plausible candidates. It would therefore be useful to consider metrics that incorporate functional or structural similarity to provide non-zero scores to reasonable alternatives. An additional concern is that the evaluation on unseen molecular structures is limited to only 300 molecules, which means that those scores will likely have higher

bias and variance. Finally, expanding the task to include – and ideally translate between – perturbational signatures from multiple cell types and dosages will be essential to assessing and optimizing generalizability to new biological settings. Given sufficiently robust results, we could then assess this method as an *in silico* nominating scheme, either as a drop-in replacement for connectivity mapping techniques or as a method to choose which drugs (from an arbitrarily large library) to profile using transcriptomics readouts.

From the perspective of model formulation, there are several follow-up experiments that we are excited to pursue that would require only additional computational and human resources. These include a more dedicated hyperparameter optimization search and the implementation of test-time augmentation by ensembling predictions over repeat control sampling. In addition, we would like to compute confidence intervals for all experimental values computed by bootstrapping. Such exercises will be indispensable in identifying which if any of the comparisons in performance may be due to statistical aberration rather than substantive difference.

Slightly more sophisticated follow-up work includes the investigation of more intelligent pre-training strategies and negative sampling techniques, which may prove essential for improved generalization. In addition, as described in Li et al, there are a wide variety of additional architectures that could be tried for this task, such as Cross-Modal Factor Analysis, Kernel CCA, and a range of novel deep learning architectures¹⁸³. Anecdotally, one model which we did begin to explore was Deep Canonical Correlation Analysis⁸, which on first implementation has not produced useful embeddings for this task. These results were not included formally in this work, however, due to insufficient time to ensure the results weren't due to poor hyperparameters.

Of greatest interest to me as an author is an even deeper examination of the failure modes that take place when the model tries to generalize to unseen molecules and cell types. In particular, we would like to characterize the properties of *chemical libraries* that do or do not enable mutual generalizability. It is our hope that elucidating these pitfalls could lead both to better algorithms and to

better chemical libraries. By the same token, we hope that in-depth investigation of generalizability of drug signatures across *cell-lines* will not only improve our ability to train better drug embedders in this task, but also yield insights that are beneficial for classic connectivity mapping.

2.6 CONCLUSION

We present a new task, cross-modal multi-view alignment between drug structures and perturbational gene expression profiles. We argue this task fills a notable gap in the current suite of cheminformatics tasks; namely, it links molecular structure to an objective, functional readout of drugs with very broad biomedical relevance. We profile state-of-the-art representation learning methods on this new task, and inspect the learned chemical embeddings. We find that this modeling task induces an embedding space reflective of drug mechanism of action – which is not explicitly included in the training regime – and see modest generalization to both new structures and a new biological environment.

As more scientists consider using connectivity mapping techniques, we hope that this task will prove helpful in enabling a rapid, *in silico* evaluation of which compounds under consideration should be profiled further via gene expression or other assays.

That being said, our oracle experiments highlight potential obstacles to deploying even classical connectivity mapping techniques in the (almost universal) scenario in which the results are intended to generalize to tissues other than those in which the perturbational assays were performed. In Appendix B, I present a thorough evaluation of these obstacles, and provide a suggested approach to mitigating these challenges in the classical connectivity mapping context.

‘Well, in our country,’ said Alice, still panting a little, ‘you’d generally get to somewhere else—if you ran very fast for a long time, as we’ve been doing.’ ‘A slow sort of country!’ said the Queen. ‘Now, here, you see, it takes all the running you can do, to keep in the same place. If you want to get somewhere else, you must run at least twice as fast as that!’

Lewis Carroll

3

Representational Robustness and Worst-Case Generalization in Clinical ML

DATASET SHIFT IS A PHENOMENON in which the data upon which a machine learning algorithm is deployed are mismatched with those on which it was trained. In this chapter, I introduce the concept of dataset shift as it pertains to clinical machine learning systems. I then turn to a special case of

dataset shift – adversarial examples – which reflect the worst-case generalization conditions for a machine learning system. Adversarial examples are particularly intriguing in the setting of deep learning systems, as they highlight the brittleness of learned representations to small changes in their input data. As a proof of concept, I develop three clinical deep learning models and construct adversarial examples that cause their accuracy to drop to 0% on data that is imperceptibly different from the training data. I then discuss the implications of these findings for clinical machine learning, offering specific regulatory recommendations.

3.1 A NOTE ON TERMINOLOGY

The term “dataset shift” has been used with different meanings in various contexts. For the purpose of this chapter, I adopt the framework proposed by Moreno-Torres et al.²¹⁶, who conducted an early survey of related research and described “general dataset shift” as coming from three broad sources: sample selection bias, non-stationary environments, and adversarial environments.

Sample selection bias occurs when the training samples are selected non-randomly from the test population, which produces a systematic distributional shift in predictors and/or labels. Dataset shift from *non-stationary environments* arises from systematic differences in space or time between training and testing. In *adversarial environments*, one or more agents want the model to behave in a specific way, and continually adapt their input data in order to determine the ML model’s output.

Importantly, as one moves along the spectrum from selection bias to adversarial environments, the problem of dataset shift become increasingly difficult to model and correct in simple probabilistic terms. Furthermore, of these three categories, sample selection bias and non-stationarity are far more familiar to biomedical researchers, though all three categories may be relevant to regulation and deployment.

3.2 DATASET SHIFT FROM A CLINICAL PERSPECTIVE

The essential phenomena underlying dataset shift are not new to medicine. Beginning in the mid-20th century, statisticians began to formulate and rigorously analyze the concept of external validity^{42,294}, which has been concisely summarized in the question “to whom do the results of this [study] apply?”²⁷². This question, typically posed in context of clinical trials, is central for evidence-based medicine, and is deceptively complicated due to both selection bias (enrollment criteria, etc.) and non-stationarity (shifting standard of care and background patient behavior over time and in different settings). Almost every obstacle to external validity in the context of clinical trials also applies to machine learning algorithms, because both endeavors presume the comparability of some study population and some downstream deployment population (see Section 0.2.2).

Nevertheless, there is at least one key difference between the external validity of a drug trial and that of a machine learning algorithm: when it comes time to deploy a machine learning algorithm, the algorithm cannot interface directly with biology. Instead, the algorithm depends, even in its very mechanism of action, upon not just the patients (and clinically-meaningful environmental/social factors) but also the information processing ecosystem around them. There is thus a range of dynamic factors in clinical medicine that must be taken into consideration when considering prospective deployment of a clinical machine learning algorithm. These factors include shifting technology, shifting patient populations, and shifting behavior. I here provide a brief summary of how each of these have shown to affect clinical machine learning, with references to related work.

SHIFTING TECHNOLOGY Machine learning systems have been shown, in some cases, to heavily rely upon seemingly mild technological factors, such as the model of scanner being used to acquire an image^{108,17} or the brand of the EHR system used to record medical notes and diagnostic codes^{110,225}. The performance of predictive models has likewise been shown to drop following

changes to the way in which coded data are structured, such as after the transition from ICD9 to ICD10²⁸⁷. Updated laboratory equipment, such as the replacement of low-sensitivity troponin assays with high-sensitivity troponin assays,³¹⁸ has created challenges for human clinicians that are similarly faced by machine learning systems. Of course, the creation or reversal²⁵³ of entirely new devices or drugs will have large effects on clinical care^{165,105} and thus the environment in which relevant ML tools are deployed.

SHIFTING POPULATION The demographic and genetic makeup of a patient population can affect any combination of the bias, variance, and noise of their measured data⁴⁹. As outlined by Chen et al.⁴⁹, this means that effective generalization across populations includes but extends beyond the obvious solution of ensuring equal representation of different demographic groups; different populations may require different models, for example. Even populations of patients with fixed demographics can shift dynamically over time, through seasonal effects such as influenza and sunlight exposure, isolated effects such as epidemic outbreaks, or progressive effects such as the emergence of new chronic diseases or the elimination of environmental exposures.

SHIFTING BEHAVIOR Subtle dynamics of clinician behavior can have a large effect on machine learning systems. For example, the time of day at which laboratory tests are ordered have been shown to be more predictive of patient outcomes than their value, in certain circumstances³³⁴. So long as such signals are fixed, they can be leveraged to improve the predictive performance of machine learning systems¹⁷. However, this creates a potential for failure due to non-stationarity if such practices (or how they are electronically stored) change, and likewise creates a risk of failure due to selection bias for patients who fall outside the support of the dataset. Patient behavior also influences the effective population who presents at the hospital and the workup they receive; for example, celebrity diagnoses³⁵⁹ and new consumer products³²⁶ can prompt healthier populations to

seek evaluation and/or request a specific treatment. Another interesting class of problems involves changes *induced* by machine learning systems; for example, some studies have demonstrated that the performance of machine-augmented human diagnostic performance may worsen over time due to automation bias^{194,158}. Finally, financial incentives result in dataset shift; for example, the rate of sepsis diagnosis as coded in hospital billing records rose nearly three-fold in 2007 when the Centers of Medicare and Medicaid Services (CMS) shifted reimbursement policy to incentivize sepsis workup¹⁰⁹.

3.3 DATASET SHIFT FROM A TECHNICAL PERSPECTIVE

3.3.1 A PROBABILISTIC TAXONOMY OF DATASET SHIFT

In this section, I outline a probabilistic framework for reasoning about dataset shift, which is slightly adapted from Moreno-Torres et al.²¹⁶. This framework is useful for understanding the existing solutions to the problem, and also crucially allows us to discuss sources of general dataset shift that these solutions fail to address and cannot be expected to address in the near future. This, in turn, will motivate the robustness and policy analyses that I introduce in the rest of this chapter.

Consider a supervised learning problem to predict target value y using features x . Dataset shift can then be described as the presence of meaningful differences in the joint distributions $P_A(x, y)$ and $P_B(x, y)$ that exist, respectively, in some training environment A and deployment environment B. If the underlying relationship between the features and the target is assumed to be causal ($x \rightarrow y$), the joint distribution is typically written in the conditional form $P(y|x)P(x)$. If the relationship is anticausal (written $y \rightarrow x$), this joint distribution is typically modeled as $P(x|y)P(y)$. An example of a $x \rightarrow y$ problem is to anatomically segment an image; in this case, the imaging features themselves define and therefore cause the labels. An example of an anticausal $y \rightarrow x$ problem could be screening for a monogenic disease based on phenotypic features; in this case, the disease itself causes the

symptoms and can be established independently using a genomic gold standard.

With the above notation in place, we can define several classes of dataset shift: *Covariate shift*^{304,284} applies exclusively to $x \rightarrow y$ problems, and refers to cases where $P_A(y|x) = P_B(y|x)$, but $P_A(x) \neq P_B(x)$. In other words, the feature space differs between the two environments, but the functional relationship remains unchanged.* *Prior probability shift* (also called *target shift*³⁶⁹) is the analog to covariate shift for $y \rightarrow x$ problems, and represents cases where $P_1(x|y) = P_2(x|y)$, but $P_1(y) \neq P_2(y)$. *Concept shift* (also called *concept drift* or *conditional shift*)^{368,369} applies to both causal and anticausal problems; when $x \rightarrow y$, it represents cases where $P_1(x) = P_2(x)$ but $P_1(y|x) \neq P_2(y|x)$, and when $y \rightarrow x$, concept shift is when $P_1(x|y) \neq P_2(x|y)$. Moreno-Torres et al.²¹⁶ refer to *general dataset shift* as a shift that does not fit one of the above criteria. This framework can be extended by incorporating additional causal variables. Recently, Castro et al. provided a thorough overview of each of these concepts with application to medical imaging, in which they presented the medicine-specific terminology of: Population Shift, Annotation Shift, Prevalence Shift, Manifestation Shift, and Acquisition Shift⁴⁵. *Policy Shift*²⁸¹ is a final class of problems, in which the training data contains relationships caused by actions (such as treatments in the hospital) and these action policies do not generalize to different settings.

Technical solutions to dataset shift typically begin by making specific probabilistic and/or causal²⁴² assumptions in the form just described. Given a set of assumptions about the problem (e.g. $y \rightarrow x$ and $P_1(x|y) = P_2(x|y)$), models can be designed that are either robust/stable^{298,286,299} to the specified form of shift, or alternatively that can be used explicitly to adapt methods to the target population, for example by re-weighting^{297,117}. These approaches can provide provable solutions that are as strong as their assumptions.

*In the case of isolated covariate shift, unlike other forms of dataset shift described above, the model trained on population A is, in principle, *directly transportable* to population B. In practice, however, there will often be regions of the support of the test distribution $P_B(x)$ that were limited or absent in the training data and therefore heavily underfit by the model.

When deployment conditions are misaligned from the training data due to selection bias – and the presence of such bias is known – the above framework provides readily actionable solutions. By the same token, some sources of clinical non-stationarity outlined in Section 3.2 fall into the categories just outlined, meaning that they can also be solved directly. With real-world data, however, non-stationarity in space or time can also produce general dataset shifts that are not known *a priori* and have complicated effects on both $P(x)$ and $P(y|x)$ that do not in general have solutions.

3.3.2 ADVERSARIAL EXAMPLES

Adversarial environments pose the most challenging case of general dataset shift. In this setting, an individual attempts to craft data in order to bypass or otherwise influence the algorithm. This can alter $P(x)$ and $P(y|x)$ arbitrarily, and can only be provably solved under extremely heavy mathematical assumptions, such as bounding the attack to fall within the ε -ball of the training data²⁸⁶. Research that tests the robustness of ML models to adversarial attacks may be useful for two reasons: First, adversarial robustness testing measures the worst-case generalization of ML systems, including in the presence of bad actors. Second, such research provides a tractable test bed for exploring general dataset shift more broadly.

Adversarial robustness research extends back in time at least as far as the spam filter, where systematic modifications to email such as ‘good word attacks’ or spelling modifications have long been employed to try to bypass the filters^{190,191,65}. More recently, adversarial examples were re-discovered and described in the context of deep computer vision systems through the work of Szegedy et al.³⁰⁸ and Goodfellow et al.¹¹⁴. Particularly intriguing in these early examples in deep learning was the fact that adversarial examples could be crafted to be extremely effective despite being imperceptibly different from natural images to human eyes. In the years since, adversarial examples – visible and otherwise – have been shown to exist for a wide variety of classic and modern learning systems^{238,40,78}. By the same token, adversarial examples have been extended to various other domains

such as text and speech processing^{147,43,149}. Figure 3.1 depicts several real and hypothetical adversarial examples that can be constructed for clinical machine learning algorithms. For an interesting history of adversarial examples and methods used to combat them see Biggio & Roli³².

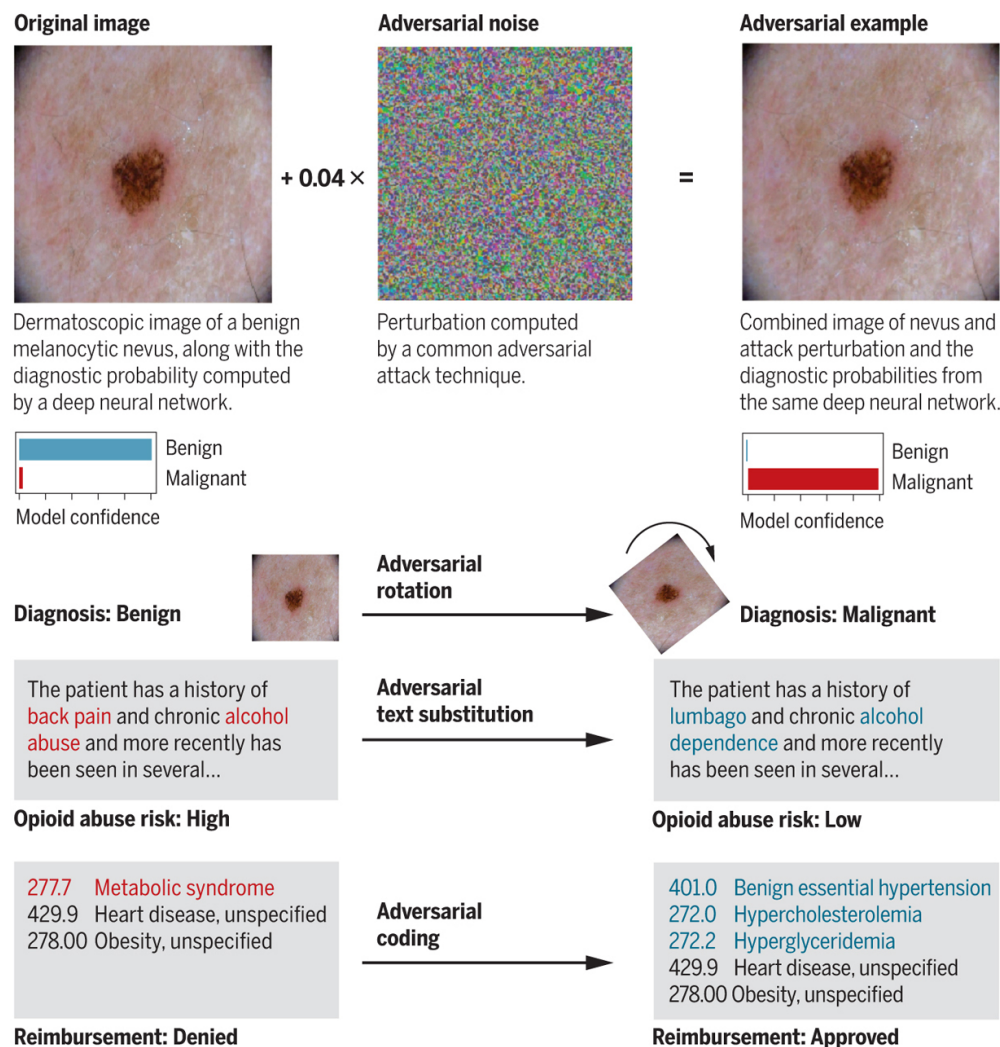


Figure 3.1: The anatomy of an adversarial example. The top row depicts the construction of an adversarial example using modern neural networks; in fact, the images, noise, and model confidences are all real examples from this work. The lower rows depict hypothetical adversarial attacks that could be performed on imaging, text, and billing fraud detectors using existing tools. In particular, the billing codes represented in the bottom panel represent real coding substitutions made to influence reimbursement²⁶⁸, which could be systematically identified using adversarial strategies. See Section C.2 for an extensive discussion of motivations behind such behavior.

Adversarial examples are generally thought to arise from the piecewise linear components of complex nonlinear models¹¹⁴. They are not random, they are not due to overfitting or incomplete model training, they occupy only a comparatively small subspace of the feature landscape, they are robust to random noise, and they have been shown to transfer in many cases from one model to another^{238,315}. Furthermore, in addition to executing many successful attacks in purely virtual settings, researchers in the past several years have also demonstrated that adversarial attacks can generalize to physical world settings^{170,82,15,40}.

Figure 3.2 places adversarial attacks (and dataset shift more broadly) in the context of the machine learning development and deployment process. While there are many possible intentional and inadvertent failures of real-world machine learning systems, adversarial attacks and dataset shift are particularly important to consider from the standpoint of model deployment, because they often lie outside the perceived purview of the typical model developer. In my experience, model developers are typically extremely well-versed in issues concerning dataset construction, model training, and computational efficiency, but are not typically trained or expected to deeply consider distantly downstream performance. Likewise, adversarial attacks fall outside the domain of typical information technology and security teams, as they do not involve any formal breach of the model or private data.

3.4 ROBUSTNESS ANALYSIS OF REPRESENTATIVE CLINICAL DEEP LEARNING SYSTEMS

To assess the worst-case robustness of medical ML to general dataset shift, we developed a series of medical classifiers modeled after state-of-the-art clinical deep learning systems, and launched successful white- and gray-box adversarial attacks against each.

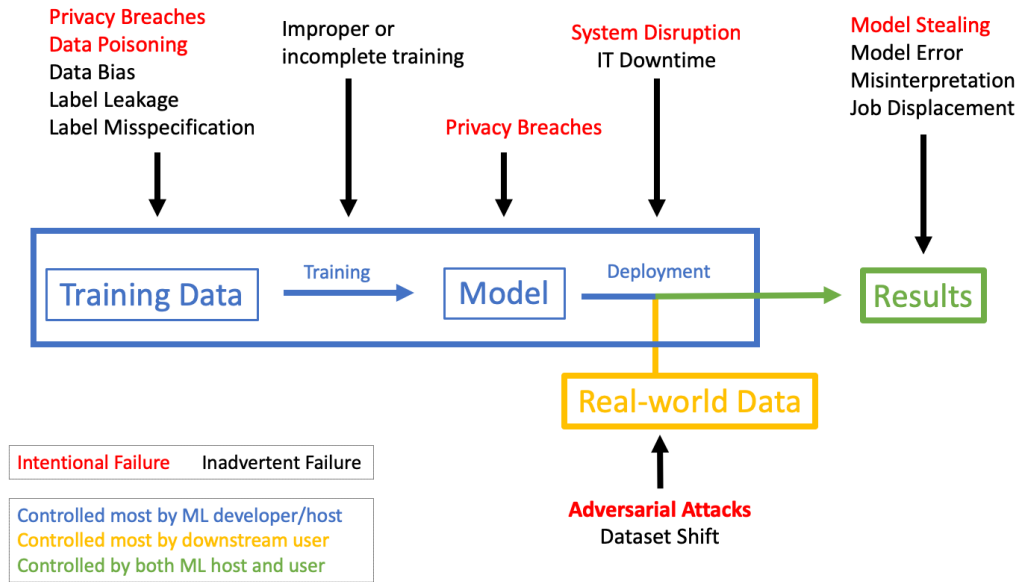


Figure 3.2: Adversarial attacks within the broader taxonomy of risks facing the machine learning pipeline. Adversarial attacks pose just one of many possible risks in the development and deployment of ML systems, but are noteworthy because they enable end-users to manipulate model outputs without ever influencing the training process or gaining access to the deployed model itself.

3.4.1 METHODS

CONSTRUCTION OF MEDICAL CLASSIFICATION MODELS

We developed baseline models to classify referable diabetic retinopathy from retinal fundoscopy (similar to Gulshan et al. ¹²⁰), pneumothorax from chest-xray (similar to Wang et al. ³²⁹ and Rajpurkar et al. ²⁶³), and melanoma from dermoscopic photographs (similar to Esteva et al. ⁸¹). The decision to build models for these particular tasks was made both due to public data availability, as well as the fact that they represent three of the most highly visible successes for medical deep learning.

All of our models were trained on publicly available data. For diabetic retinopathy, this was the Kaggle Diabetic Retinopathy dataset^{kag}. The key distinction with the Kaggle dataset, however, was

that we were seeking to predict *referable* (grade 2 or worse) diabetic retinopathy in accordance with Gulshan et al.¹²⁰ rather than predicting the retinopathy grade itself as was the case in the Kaggle competition. As such, the training and test sets from the kaggle competition were merged, relabeled using their provided grades, and split by patient into training and test sets with probability 0.88/0.12. For the chest x-rays, we used the ChestX-Ray14 dataset described by Wang et al.³²⁹. We identified cases and controls by selecting images whose labeled contained ‘pneumothorax’ or ‘no finding’, respectively, and additionally excluded from our control group any images from patients who had received both labels. We then split by patient into training and test sets with probability 0.85/0.15. For melanoma, we downloaded images labeled as benign or malignant melanocytic lesions from the International Skin Imaging Collaboration website^{isi}, splitting again into training and test sets with probability 0.85/0.15.

As in the case of all three of the original papers that inspired these models, we built our classifiers by fine-tuning a pretrained ImageNet model. For convenience and consistency, we chose to build each of our networks using a pretrained ResNet-50 model, fine-tuned in Keras using stochastic gradient descent with a learning rate of 1E-3 and Momentum of 0.9. Data was augmented using 45° rotation and horizontal flipping chest x-ray images, and with 360° rotation, vertical and horizontal flipping, and mixup³⁶⁶ for fundoscopy and dermoscopy images. In all three cases, these settings provided respectable performance and we therefore didn’t perform dedicated hyperparameter optimization.

CONSTRUCTION OF ADVERSARIAL EXAMPLES

To demonstrate the vulnerability of our models to adversarial attacks under a variety of threat models, we implemented both human-imperceptible and patch attacks.

For our human-imperceptible attacks, we followed the white and black box PGD attack strategies published in Madry et al.¹⁹⁵, which were also described as baselines in Kannan et al.¹⁵⁴. The PGD

attack (see Madry et al. ¹⁹⁵ and also Kurakin et al. ¹⁷⁰) is an iterative extension of the fast gradient sign method (FGSM) attack developed by Goodfellow et al. ¹¹⁴. In the PGD attack, given input $x \in \mathbb{R}^d$, loss function $L(\theta, x, y)$, and a set of allowed perturbations $\mathcal{S} \subseteq \mathbb{R}^d$ (most commonly the ℓ_∞ ball around x), one can perform a projected gradient descent on the negative loss function:

$$x^{t+1} = \Pi_{x+\mathcal{S}}(x^t + \varepsilon \cdot \text{sgn}(\nabla_x L(\theta, x, y))) \quad (3.1)$$

in order to identify an optimal perturbation. We implemented the PGD attack using the library Cleverhans, conducting 20 iterations with hyperparameter $\varepsilon = 0.02$ (ε corresponds to the maximum permitted ℓ_∞ norm of the perturbation)²³⁷. The methods for our patch attacks and controls are extensively described in Section C.1; in brief, we followed the approach of Brown et al. ⁴⁰ to build patches that were universally adversarial for all data in the training set.

White box attacks for each attack were implemented by building attacks directly on the victim model itself. As in Madry et al. ¹⁹⁵ and Kannan et al. ¹⁵⁴, gray box attacks were performed by crafting the attack against an independently-trained model with the same architecture and then transferring the resultant adversarial examples to the victim.

Availability of code: Code to reproduce the analyses and results can be found at

<https://github.com/sgfin/adversarial-medicine>.

3.4.2 ATTACK RESULTS

The results of our experiments are depicted in Tables 3.1 and D.17, and in Figure 3.3.

While discrepancies in data source and train-test partitioning makes direct comparison to state-of-the-art models infeasible, all of our baseline models achieved performance reasonably consistent with the results reported in the original manuscripts on natural images: AUROC of 0.910 for diabetic retinopathy compared with 0.936 reported in Gulshan et al. ¹²⁰, AUROC of 0.936 for pneu-

Inputs	Accuracy	AUC	Avg. Conf.
<i>Fundoscopy</i>			
Clean	91.0%	0.910	90.4%
White Box	0.00%	0.000	100.0%
Black Box	0.01%	0.002	90.9%
<i>Chest X-Ray</i>			
Clean	94.9%	0.937	96.1%
White Box	0.00%	0.000	100.0%
Black Box	15.1%	0.014	92.6%
<i>Dermoscopy</i>			
Clean	87.6%	0.858	94.1%
White Box	0.00%	0.000	100.0%
Black Box	37.9%	0.071	92.0%

Table 3.1: Results of medical deep learning models on clean test set data, white box, and gray-box attacks.

mothorax compared with 0.90 reported by Rajpurkar et al. ²⁶³, and AUROC of 0.86 on melanoma compared with 0.91-0.94 reported in Esteva et al. ⁸¹.

Projected gradient descent attacks, targeting the incorrect answer in every case, produced effective AUROCs of 0.000 and accuracies of 0% for all white box attacks. Black box attacks produced AUROCs of less than 0.10 for all tasks, and accuracies ranging from 0.01% on funduscopy to 37.9% on dermoscopy. Qualitatively, all attacks were human-imperceptible.

Adversarial patch attacks also achieved effective AUROCs of 0.000 and accuracies of <1% for white box attacks on all tasks. Black box adversarial patch attacks achieved AUROCs of less than 0.005 for all tasks and accuracies less than 10%. The "natural patch" controls created by adding patches created from the most-strongly classified image of the desired class resulted in AUROCs ranging from 0.48-0.83 with accuracies ranging from 67.5% to 92.1%.

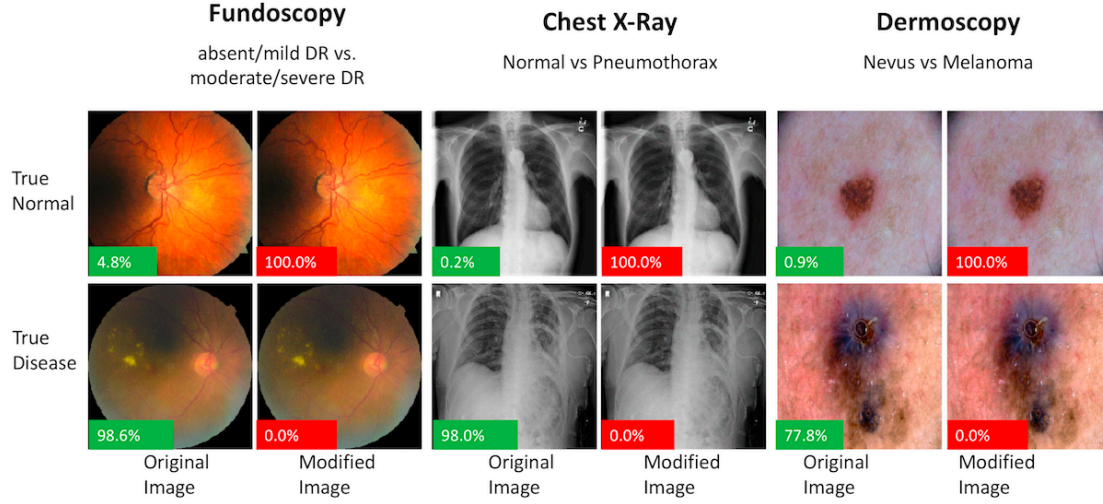


Figure 3.3: Characteristic results of adversarial manipulation. Each clean image represents the natural image to which the model assigns the highest probability for the given diagnosis. The percentage displayed on the bottom left of each image represents the probability that the model assigns that image of being diseased. Green = Model is correct on that image. Red = Model is incorrect.

3.4.3 DISCUSSION OF ATTACK RESULTS

The existence of such potent adversarial examples so close to the natural data manifold has at least two major implications: First, our results indicate that current deep learning systems are not well-aligned with our human perceptual understanding of the classification task. Second, our experiments indicate that from a machine learning perspective, adversarial attacks are likely to be feasible even for extremely accurate medical classifiers, regardless of whether prospective attackers have direct access to the model or require their attacks to be human imperceptible or universal.

The first of these questions is an extremely active area of machine learning research. For example, researchers are seeking to understand if adversarial examples in imaging systems do truly represent mistakes, or whether they in reality hone in on true features of the image classes that are simply human-imperceptible¹⁴². Other work has explored whether adversarial robustness can be used as a proxy task for learning representations that are better-aligned with human understanding of the

associated concepts⁸⁰. While these topics are extremely interesting – and raise pressing questions about whether or not such methods are prepared for safety-critical settings – I defer their discussion to the rich primary literature.

The question of whether there are viable threat models involving adversarial attacks in clinical medicine (in contrast with, for example, self-driving cars^{82,44,193}) had not been thoroughly explored prior to our work. In Section C.2, I present an extensive overview of the patterns of existing adversarial behavior in the United States healthcare system, and outline a process that could naturally yield increasingly sophisticated adversarial attacks. In brief, I make the case that sophisticated adversarial attacks in the form of this thesis are not likely ongoing at present, but that primitive versions of these same patterns – such as the coding example at the bottom of Figure 3.1 – are already rampant in the healthcare system. Thus, the distinction between certain actions against fraud detectors today and the methods presented in this paper is arguably a difference of degree rather than kind. I then outline how ongoing trends could contribute to an increasingly sophisticated information exchange that could give rise to more formal adversarial methods. In Section C.3, I present an overview of the technical features of the healthcare system that could likewise contribute to adversarial attacks. Finally, in Section C.4, I present an overview of possible algorithmic and infrastructural defense strategies against adversarial attacks.

3.5 REGULATORY RECOMMENDATIONS

3.5.1 KEY QUESTIONS RAISED BY EXTREME DATASET SHIFT

The existence of adversarial examples and other extreme instances of dataset shift, as in Nestor et al.²²⁵, raise ethical and regulatory questions.

First, we must ask the question of whether we can trust machine learning prospectively in safety-critical settings at all, given that they are so demonstratively brittle to small changes and we know

healthcare to be such a dynamic deployment environment. While this question is important, it ignores the fact that current processes – and indeed the human brain itself – are also often inaccurate and inexplicable. In this light, evaluations of the trustworthiness of machine learning must be made on a case-by-case basis, and be governed by an assessment of risks and benefits of the machine learning systems in direct comparison with existing systems.

Assuming machine learning systems are to be deployed in at least some settings, a second line of questions involves how to incorporate our understanding of dataset shift into the regulatory and deployment process. As outline above, certifiable algorithmic robustness to specific types of dataset shift (including specific adversarial attacks) is possible; however, such solutions require strong assumptions about the deployment environment, and in many cases also come at the cost of provably lower average accuracy.^{241,317} In this light, when and in what form should regulators require that models are certifiably robust to specific shifts? Again, this requires careful consideration of specific deployment environments. In the final section of this chapter, I provide specific recommendations for how to incorporate the principles discussed in this chapter into the existing regulatory policies of the United States.

3.5.2 PROPOSED ADJUSTMENTS TO THE EXISTING GOVERNMENT FRAMEWORK

The United States Food and Drug Administration (FDA) has recently created a new regulatory category termed “software as medical devices” (SaMDs), and released a novel and non-binding framework specifically for regulating SaMDs that are powered by artificial intelligence and machine learning⁸⁵. Drawing upon the guidance from the International Medical Device Regulators Forum (IMDRF), this regulatory framework includes a risk categorization that classifies each device along two axes based on its intended use: The first, which I refer to as “autonomy,” reflects the significance of information provided by the SaMD to a health care decision, assigning greater risk to more autonomous devices. The second, which I term “clinical urgency,” refers to the seriousness of the

healthcare situation in which the device is applied, ranging from non-serious to critical. Based on these two factors, a risk score is assigned to each SaMD from I to IV, with IV referring to fully autonomous models deployed in critical settings. Additional considerations include whether a model is “locked” (fixed without continuous learning until a manual update is performed) or “adaptive”[†] (continuously learning in a closed loop). Finally, manufacturers are expected to provide information regarding the model’s performance and specific inputs.

The combination of a model’s autonomy, clinical urgency, and pre-clinical performance does indeed provide a sensible window into the harm that a SaMD might confer upon a specific patient. Nevertheless, this framework ignores an equally important factor: the anticipated stability of a model’s performance over time. As we have seen in this chapter, the performance of a model is in fact dynamic, and a function of both the model itself and the environment in which it is deployed. In this light, I recommend the following three-pronged approach to integrate the notion of dataset shift into the existing regulatory approval process:

PRE-APPROVAL: IDENTIFICATION OF ANTICIPATED SOURCES OF DATA/MODEL INSTABILITY

As outlined in section 3.2, there are many sources of clinical dataset shift that are known and well-documented. As a part of the application process, manufacturers could be expected to stipulate dynamic clinical factors that could pose risks to the generalization in their model’s intended. This list could then be adjudicated and refined in a bi-directional process with the FDA, guided, in part, by the principles outline in Section 3.3.1. To facilitate such an adjudication process, the FDA would curate a common list of sources of shifts and instabilities relevant to various indications, such as those listed in Section 3.2 and Castro et al.⁴⁵. Among the considerations would also be the financial

[†]Note: Intuitively, it would seem that adaptive models would be more robust to dataset shift, and for certain classes of shift this would be the case. In other situations, however, adaptive systems could slowly overfit to new sites and/or local prevalence and become *less* robust to distributional shift. As such, continuous learning systems cannot be considered a panacea in this context.

incentives surrounding the model, including whether a mismatch exists between the interests of the model host and the parties submitting data into the algorithm; in this case, the potential for adversarial attacks could be considered if simpler means of fraud are already accounted for.

MID-APPROVAL: EVALUATION OF EVIDENCE OF ROBUSTNESS Once anticipated sources of dataset shift have been established for a specific SaMD, regulators and manufacturers could engage in a principled evaluation of the robustness of a model's performance. More specifically, for each variable of major concern, manufacturers could provide evidence of robustness in one of two forms: algorithmic or empirical. Algorithmic evidence of robustness would apply to sources of dataset shift to which the model is provably invariant. More commonly, empirical evidence of robustness would come in the form of clinical studies that evaluate generalization across a population that varies widely with respect to that variable (e.g. background prevalence of pneumonia for a chest-xray classifier⁷). For cases where neither form of evidence can be provided, labeling restrictions (e.g. the model of a scanner used to acquire an image, the EHR software that provides coded data, etc.) would be required, or specific auditing and reporting requirements might be put in place.

POST-APPROVAL: MONITORING AND LABEL EXPANSION Unanticipated sources of dataset shift will undoubtedly occur, which necessitates the institution of mechanisms to swiftly detect and correct for model deterioration. While this has been recognized by other authors¹⁶ and by the FDA itself,⁸⁵ it would be irresponsible to rely entirely upon monitoring to catch errors that could be anticipated in advance. In addition, monitoring requirements could be made more (or less) aggressive in settings, data types, and model classes that are known to be more (or less) unstable. Taken together, the monitoring process post-approval is complementary to and should be informed by the two steps outlined above.

3.6 CONCLUSION

Dataset shift represents one of the greatest emerging obstacles to the successful deployment of machine learning in healthcare. As we have seen, the essential phenomena that produce dataset shift are readily familiar to the clinicians, statisticians, and regulators who have grappled with similar problems for decades. Despite the many similarities to previous challenges, however, machine learning systems require particular care in safeguarding against dataset shift. In this chapter, I have outlined the problem of dataset shift from a clinical and technical perspective, and conducted an analysis on three state-of-the-art diagnostic systems to assess worst case generalization performance in the presence of dataset shift. Finally, I have proposed additions to the existing regulatory framework that can be leveraged prior to, during, and after the approval process to minimize the risk of catastrophic failure while still allowing for the adoption of this valuable technology.

If biases and initial knowledge are at the heart of the ability to generalize beyond observed data, then efforts to study machine learning must focus on the combined use of prior knowledge, biases, and observation in guiding the learning process. It would be wise to make the biases and their use in controlling learning just as explicit as past research has made the observations and their use.

Tom M. Mitchell

4

Conclusion

4.1 SYNTHESIS

IN THIS WORK, I have sought both to assess and extend the capacity of machine learning algorithms to learn robust representations for important biomedical problems. A major theme of this thesis has been the immense difficulty of reliable generalization: across biological conditions, across chemical

structures, across deployment environments, across time.

In using the phrase ‘inductive representations’ in my title, I have sought to emphasize how this work relates to two key aspects of generalization: First, I emphasize that the methods I’ve developed in Chapters 1 and 2 are inductive rather than *transductive*, which is a major limitation of related prior work. Second, I emphasize that the goal of machine learning is inextricably intertwined with *the problem of induction*, and that this recognition should bring both humility and care to our work in machine learning. Assessing our models ability to be inductive in this broader sense is the primary aim of Chapter 3, but is also a major theme of Chapter 2 and indeed all of my projects.

Robust generalization requires models whose inductive biases are well-aligned with the learning task and the target domain. While deep learning has the reputation for minimizing the need for domain knowledge, due to its obviation of manual feature engineering, my experience is that learned representations instead simply shift the focus of domain knowledge towards other aspects of model design and evaluation. More specifically, given the flexibility of neural networks, domain knowledge is essential to understand the key structures and invariances in the task that should be reflected in model behavior.

Such understanding of the problem can be translated into the representational learning process in three ways: the design of model representational *capacity*, the design of a training regime that induces representation *alignment* with the downstream task, and the interrogation of a model’s *representational robustness*. By way of summary and synthesis, I provide comments on each with examples from the body of this thesis.

MODEL REPRESENTATIONAL CAPACITY Generalization is impossible if the model lacks the representational capacity to robustly encode the signal. In this dissertation, a wide variety of neural modules have been leveraged to match the model capacity with the problem domain: graph neural networks for permutation invariant data (Chapters 1 and 2), recurrent neural networks for se-

quences (Chapter 1), convolutional neural networks for euclidean data (Chapter 3), feedforward neural networks (Chapter 2), attention (Chapter 1), etc. In particular, Chapter 1 was focused on the design of a novel architecture that can encode the key properties of subgraphs from network science in a disentangled fashion. As we have seen throughout, these design decisions have resulted in improvements over pertinent baselines.

REPRESENTATION ALIGNMENT While representational capacity determines what a model is capable of learning, the training regime is what determines what the model actually does learn. In other words, it aligns the learned representation with its intended properties. Much of the innovation in deep learning over the past decade has been focused on designing architectures and losses that drive models to appropriately encode the training data. For example, in Chapter 2, I leveraged contrastive training to align the embeddings from two distinct data modalities. In Chapter A, I leverage a progressive training scheme of dualing imaging architectures to learn to simulate radiographs. While many of these techniques are domain-independent, domain knowledge is essential for guiding judgment about where the signal lies in the data, which tasks or datasets might be synergistic, etc.

REPRESENTATIONAL ROBUSTNESS The most humbling aspect of model development is also its most important: critical evaluation of our models, informed by domain knowledge. As demonstrated by the ease of finding adversarial examples (Chapter 3), *no* machine learning model can be assumed to have fully solved a problem. Indeed, as argued by both computer scientists and philosophers, it is literally impossible to build a model that can generalize into every possible future instance, because many valid generalizations contradict each other. In this light, the question is not whether or not our models are imperfect, but whether or not their narrow window of success has aligned with our best extrapolation of what future data will hold. This is fundamentally an interdis-

ciplinary question, requiring modelers to aggressively interrogate the limitations of their methods and domain experts to provide insight into the invariances and performance measures that truly matter. In this thesis, I have engaged on both sides of this process, and have been thoroughly humbled as a machine learning developer in doing so – particularly in noting the challenge of cell line extrapolation in Chapter 2. This prompted a follow-up study, presented in Appendix B, in which I sought to understand the problem domain in more depth without using machine learning, in hopes of informing both classical approaches to the problem as well as future attempts to use machine learning. On a more positive note, I believe that core principles outlined in Chapter 3 could, if adopted, improve the trustworthiness of medical machine learning as well as the collaboration between technical and domain experts.

4.2 LOOKING FORWARD

I am highly confident that the near future will continue to see heavy investment in the development of models with richer capacities, more clever training regimes, and a diverse body of interesting applications. This will, no doubt, yield productive and exciting fruit. A major gap, however, in the field of applied research, is the incentive to fully embrace the process of critically evaluating one’s own models to ensure that they align with proposed downstream tasks. By the same token, the science of *integrating* academic research into clinical workflows is often relegated to industry, which may be appropriate for mature technology but ignores fundamental research questions about how to do so; even worse, the resultant communication gap stymies the capacity of academic researchers to understand the basic obstacles to real-world use that could be overcome through academic investigation.

Inspired by these and related gaps, I have co-organized four interdisciplinary events during my PhD training: two Machine Learning for Healthcare workshops at the Neural Information Process-

ing Systems conference (a premier machine learning venue), a new Conference on Health, Information, and Learning affiliated with the Association for Computing Machinery, and a new Symposium on Artificial Intelligence in Learning Health Systems (SAIL). In each case, and in particular at SAIL, I have sought not just to foster more sophisticated research but also to bring together diverse stakeholders – machine learning researchers, clinicians, regulators, payers, and publishers – with the goal of identifying and overcoming the often unglamorous obstacles to real-world benefit. This process, which I consider a second half of my graduate education, has been both satisfying and inspiring. As I return to medical training, I hope to identify further opportunities to try to facilitate the integration of academic research into clinical practice.

Taken together, machine learning is still in its relative infancy in healthcare, and the coming maturation process will likely involve both prominent failures and unequivocal successes. It is my sincere hope that with principled approaches for both methods development and performance evaluation, the future of the field will be bright.

End? No, the journey doesn't end here.

GANdalf



Generative Adversarial Networks for Hip Fracture Simulation

Medical trainees must view thousands of images over the course of their training in order to achieve competency in visual diagnosis³²⁸. Nevertheless, medical images pose unique legal and technical challenges that make it difficult to widely disperse images for training between sites or rapidly and simply query data for training purposes within a site⁵⁵.

In this chapter, I propose medical generative adversarial networks (GANs) motivating for developing (human) training tools in the field of radiology. Utilizing the architecture developed by Karras et al. ¹⁵⁵, we train a conditional GAN over fractured and non-fractured hip radiographs from the dataset utilized in ¹⁰¹. We sample conditional images at 1K resolution, and perform in silico experiments to demonstrate that they are both diverse and contain key visual features necessary to distinguish fractured from healthy hips. Finally, I propose human experiments that are under development to more fully elucidate the comparative benefit of GAN-generated images in training medical professionals.

A.1 BACKGROUND

A.1.1 GENERATIVE ADVERSARIAL NETWORKS

Generative Adversarial Networks (GANs) are a framework proposed by Goodfellow et al. ¹¹³ to estimate generative models over rich data distributions. The key insight of the GAN framework is to pit a generator model G against a discriminator model D : The discriminator is optimized to distinguish real images from synthetic images, whereas the generator is optimized to reproduce the true data distribution p_{data} by generating images that are difficult for the discriminator to differentiate.

Taken together, D and G can be said to engage in a two-player min-max game with the following objective function

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_{data}} [\log D(x)] + \mathbb{E}_{z \sim p_z} [\log(1 - D(G(z)))], \quad (\text{A.1})$$

where x is a real image from the true data distribution p_{data} , and z is a noise vector sampled from distribution p_z (e.g., uniform or Gaussian distribution).

Conditional GANs ^{104,211} are an extension of the GAN game, where both the generator and

discriminator receive additional conditioning variables c , yielding $G(z, c)$ and $D(x, c)$. This formulation allows G to generate images conditioned on variables c . Conditional GANs have been used to extend the GAN framework to many applications, including generating images based on specific features such as text descriptions. Karras et al.¹⁵⁵ proposed a GAN architecture that starts with small images and progressively adds layers to grow them to higher resolutions throughout training.

There has been considerable debate in recent years pertaining to the extent to which GANs are truly "distribution learners" and their utility for tasks such as data augmentation^{277,180,11}. This may imply that in the immediate term, some degree of caution should be taken when using GANs in clinical applications.

A.1.2 MOTIVATION AND RELATED WORK

Medical imaging poses a paradoxical pair of challenges from the perspective of information technology: On the one hand, medical images create a *big data problem*: millions of medical images are ordered each year within the United States alone, each of which must be processed, interpreted by medical experts, and stored. Nevertheless, most who work with medical data suffer from a *small data problem*: medical images are generally siloed and incompletely annotated, making it difficult in many cases for individual researchers and medical trainees to compile sufficient examples for human or machine learning tasks⁵⁵.

A number of attempts have been made to apply GANs to medical imaging datasets^{230,66,343,139,119,167,22}, including more recently the use of conditional GANs for data augmentation in training medical machine learning classifiers^{98,345}. Seemingly unexplored, however, is the possibility of utilizing GANs as a source of training images for humans.

There are several technical and practical reasons for which clinical education may be an ideal deployment setting for medical GANs. Educational environments are comparatively lower stakes than clinical settings, and do not need connect with EHR or PACS systems. As the data backbone for

more advanced educational tools, GANS could provide an exceptionally large set of human training examples with comparatively small and entirely self-contained storage requirements. To boot, GANs could in theory allow for human-driven or programmatic conditioning on clinical attributes, which could be used to build rapid and interactive learning experiences.

Educational tools need not cover the full breadth of a clinical domain to provide utility to a learner; for example, consider the frequently visited webpage “Normal Chest X-Ray: Train Your Eye”, which consists entirely of 500 examples of *normal* chest-xrays (see: <http://www.chestx-ray.com/index.php/education/normal-cxr-module-train-your-eye>). This could mean that medical GANs could be useful in an educational setting even if they only learn to sample from a small subset of the data distribution.

A.2 METHODS

A.2.1 DATA AND MODEL

We trained a generative adversarial network to generate high resolution (1024×1024) hip images conditioned on fracture status.

Institutional approval was granted for all stages of this project. Data for this study consisted of 11,734 pelvic X-ray images of which 3,289 images contain fractures. Data were obtained from the clinical radiology archive at The Royal Adelaide Hospital as described in Gale et al.¹⁰¹. In brief, data were preprocessed using a series of convolutional neural networks to identify and extract the region of the pelvic X-rays that contain the hip. Diagnoses were extracted from radiology reports and curated by a practicing radiologist. Right hips were flipped across the vertical axis so that all hips appeared to be left hips.

To implement our model, we leveraged the code accompanying the original progressive GANs paper¹⁵⁵. We conducted several initial experiments (not shown) conditioning on simple features

such as anatomic location and with smaller resolutions. Also not shown are additional experiments using a customization of the StackGAN architecture³⁶⁷, which yielded lower resolution images and worse conditioning. Due to resource constraints, the final GAN was only allowed to train for four days on two NVIDIA Titan GPUs, for a total of 8,800,000 images exposed to the discriminator.

A.2.2 EXPERIMENTS

FRACTURE CLASSIFICATION

To assess whether our conditional GAN was properly conditioning on fractures, we conducted a series of experiments with classification models. To conduct these experiments, we split data our data into training (85%) and test sets (15%) by patient. We then trained our GAN exclusively on the training set, and generated two additional "training sets" of the same number and class composition as the real training set by sampling from the conditional GAN with different seed values. For all classification models, Inception-v3 pretrained on ImageNet was fine-tuned after the last fully-connected layers replaced with a global average pooling, a fully-connected layer, and a sigmoid layer.

Our first test was a conditional variant of the boundary distortion test described in Santurkar et al.²⁷⁷. To execute this test, we first trained and compared classifiers on the real training set and as well as classifiers exclusively trained on each of the two GAN-sampled training sets. Each model was then evaluated on the test set of images held out from the classifier and GAN training.

To further test the characteristics of the GAN-sampled images, we trained classifiers using each of the GAN datasets as augmentation for the real data and for each other, as well as all datasets together. We additionally created one more baseline model by doubling the effective size of the real training set using traditional data augmentation.

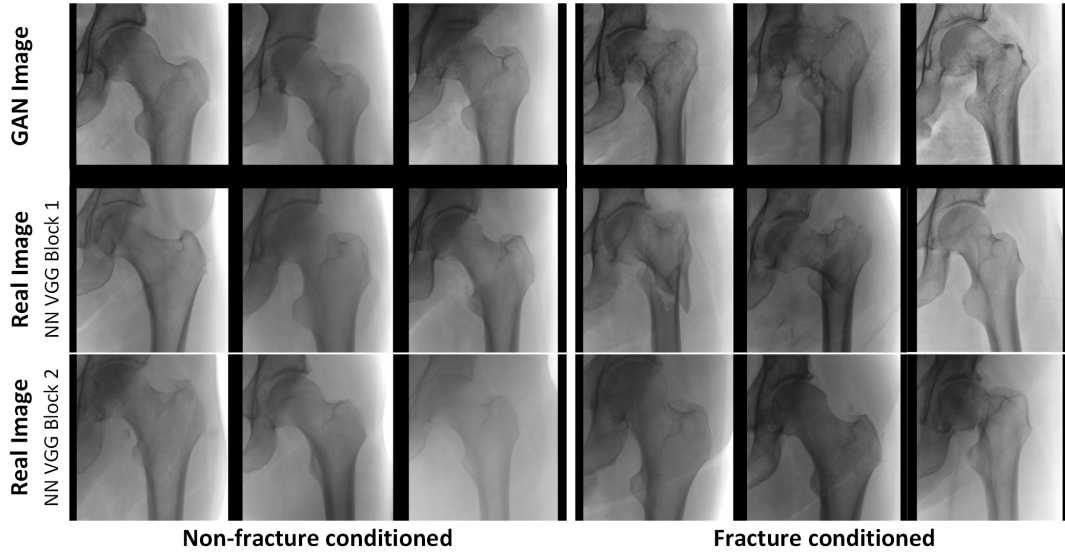


Figure A.1: Randomly sampled images generated by the GAN (3 from each class using a pre-specified random seed), along with nearest-neighbors in the training set in feature space of a VGG16 model trained on radiographs.

NEAREST-NEIGHBORS AND T-SNE ANALYSIS

We next conducted experiments to test whether our model was merely memorizing images from the training set. To do so, we adapted the evaluation of Chen & Koltun⁵⁰ and Karras et al.¹⁵⁵, comparing sampled images to near-neighbors in the training set. To find the nearest-neighbors of generated images, we trained VGG16 on the real images in the training and computing activation embeddings at each intermediate convolution layers. We then identified the nearest neighbor in the training set in L_1 distance for each such layer. Finally, we utilized the same VGG model to generate t-SNE maps of 256 randomly sampled images from each class in each training set¹³².

Training data	AUC	AP
Real	0.992	0.990
GAN-sample1	0.951	0.923
GAN-sample2	0.956	0.935
GAN-sample1 + GAN-sample2	0.945	0.924
Real + Augment	0.991	0.988
Real + GAN-sample1	0.991	0.988
Real + GAN-sample2	0.989	0.983
Real + GAN-sample1 + GAN-sample2	0.989	0.983

Table A.1: Performance of fracture classification models trained with different sets of training data. All Area under the receiver operating curve (AUC) and average precision (AP) scores are reported on a held-out test set of real fracture images that were absent from the training set for the classifiers and GAN.

A.3 RESULTS AND DISCUSSION

Results of our quantitative analysis are presented in Table A.1. Models trained on GAN-sampled images alone achieved an $AUC > 0.95$ on real test images. While this was not as high of test-set accuracy as the models trained on real images (AUC 0.992), it clearly demonstrates that our conditional GAN was able to encode key features that were sufficient for visual diagnosis.

Figure A.1 depicts six randomly sampled (one time, with a pre-specified seed) images from the GAN, along with top-1 near-neighbors in the feature space from the first two convolutional layers of a VGG network trained on fracture images. Top-1 neighbors from all other layers are in the appendix in Figure A.6. No obvious copies are visible in any case. Figures 2-4 in the appendix depict t-SNE plots from 256 images sampled from each class from the real and first GAN-sampled training set. No obvious mode collapse is present, and a range of different types of fractures and annotation markers were learned by the GAN, though the samples appear less diverse than the true images.

Visual inspection and classification results both indicate that our conditional GAN has learned key visual features that distinguish fracture from non-fracture images. Though of slightly lower quality than real images in appearance and performance, the objective success of classifiers trained

solely on GAN samples is still striking and strongly implies that our GAN has learned clinically-relevant features sufficient to aid in machine learning. Based on these results, we are currently designing a simple trial to test the extent to which GAN-sampled images serve as an effective alternative to real images in *human* learning. Of note, while simulating the true data distribution is the goal of GAN training, in the setting of medical education, whether or not sampled images are indistinguishable from or as diverse as real images may or may not prove to be fully correlated with whether they are helpful to students learning to internalize the key distinction between fractures from non-fractures.

From a machine learning perspective, the data augmentation results are intriguing. Prior studies in the medical domain have reported improved classification performance using GANs as a means of data augmentation, although none of these studies were conducted in the setting of nearly perfect performance on real images alone^{277,180,11}. These findings seem to suggest that GAN images may serve as effective augmentation in the low-data regime, but may be actively unhelpful in a high-data regime. We presently intend to conduct experiments simulating the relative impact of GAN augmentation when progressively limiting access to real images.

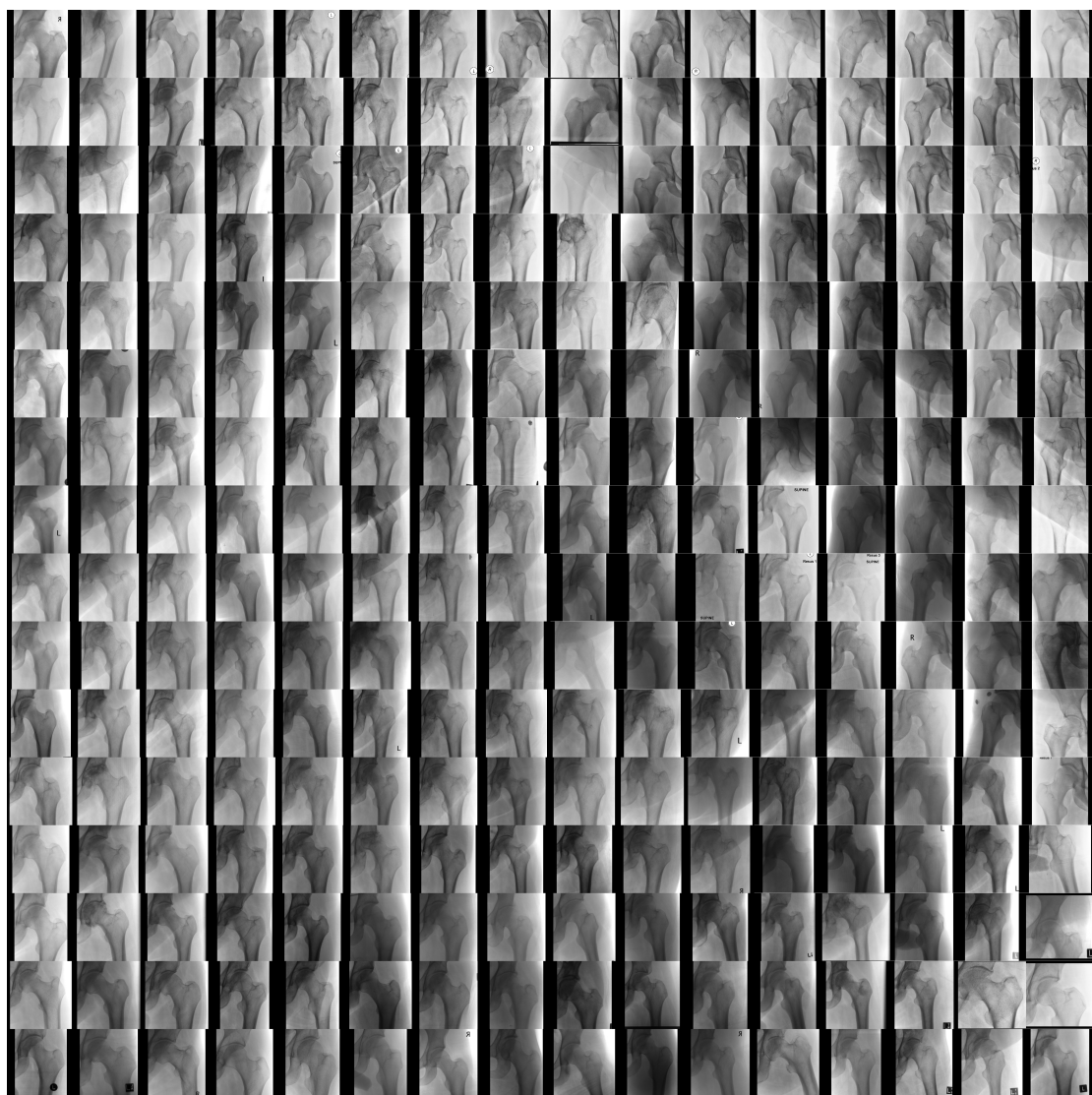


Figure A.2: t-SNE of Additional Examples of Real, Non-fractured Images.

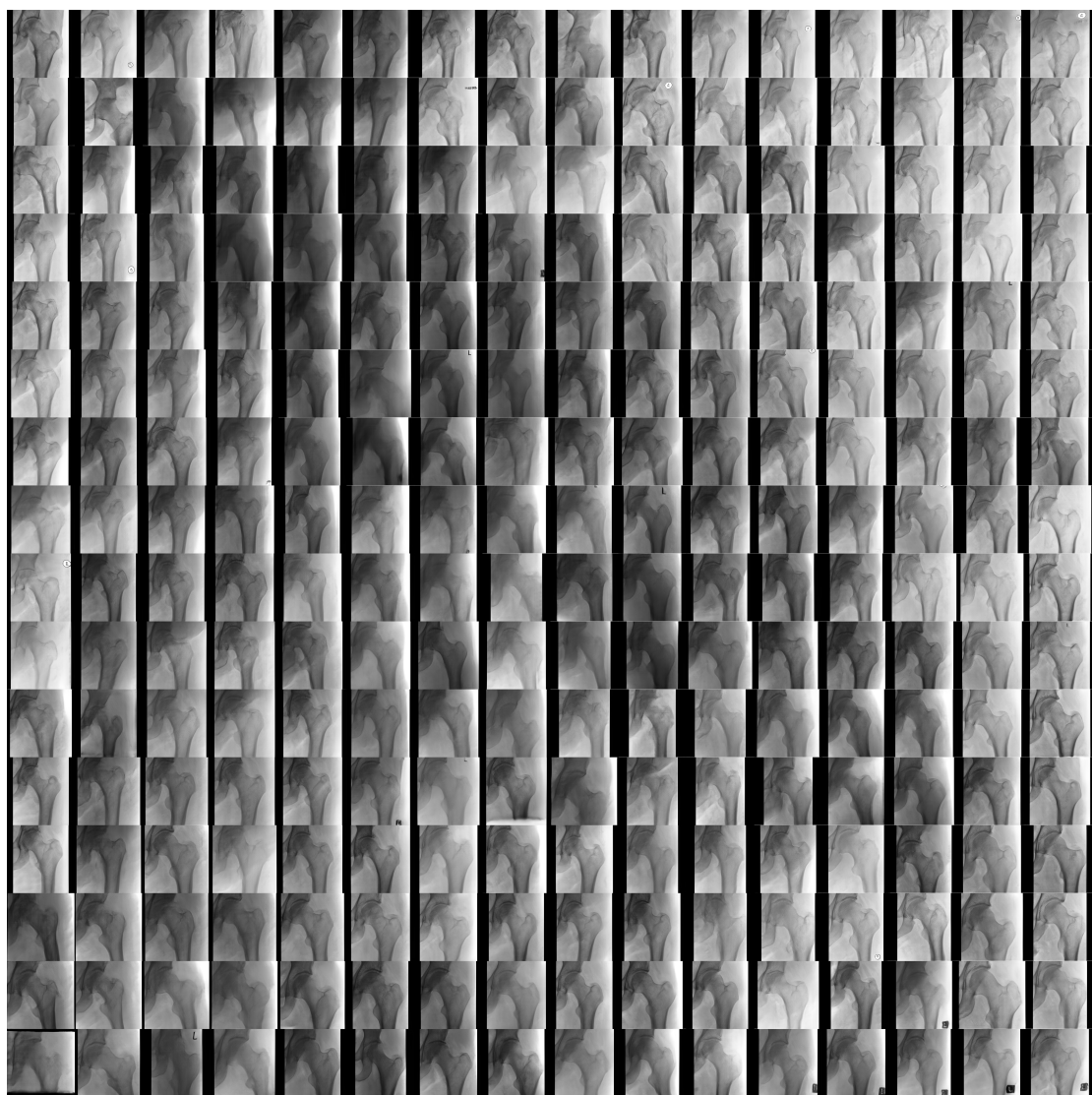


Figure A.3: t-SNE of Additional Examples of GAN-generated, Non-fractured Images.

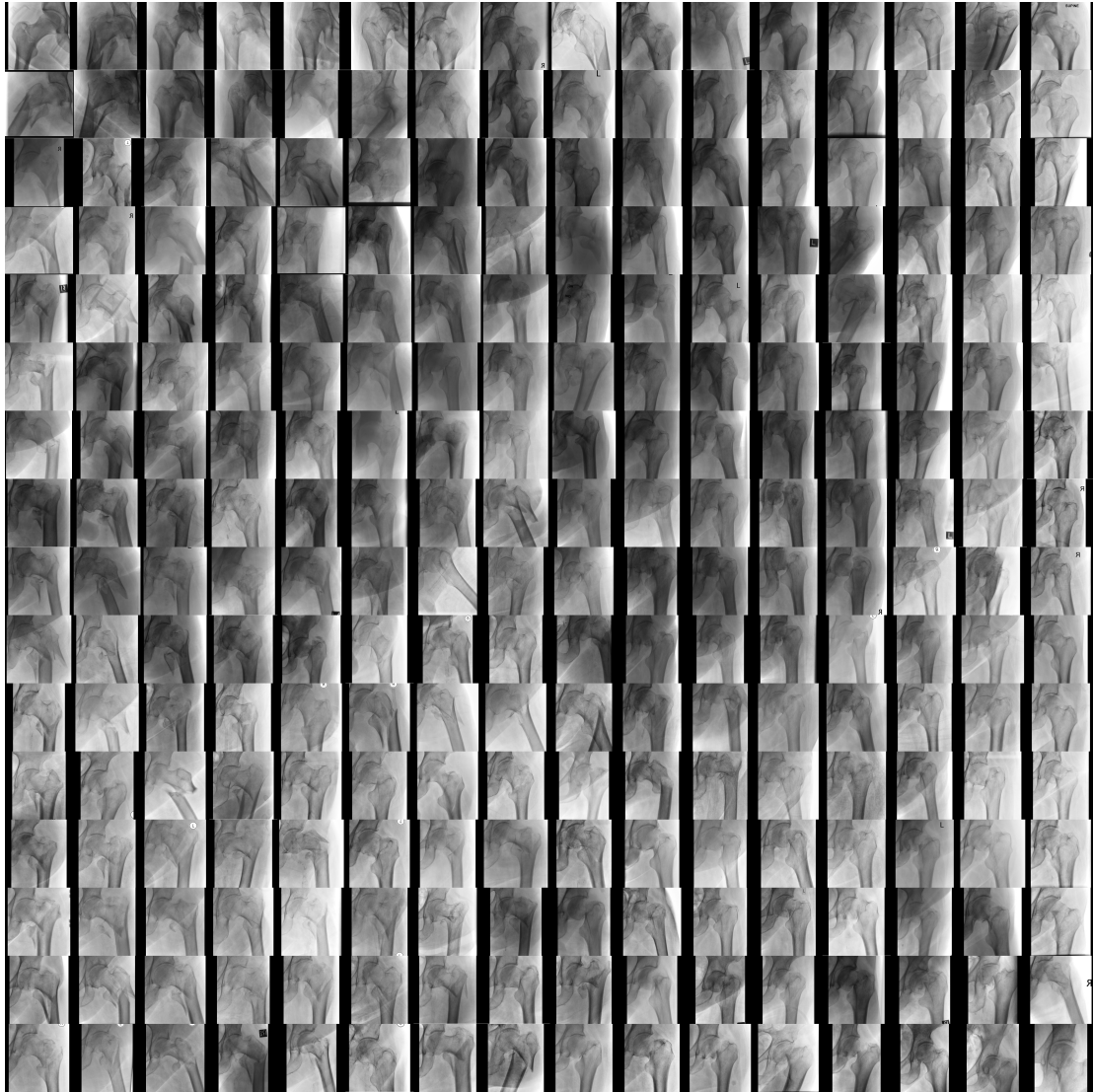


Figure A.4: t-SNE of Additional Examples of Real, Fractured Images.

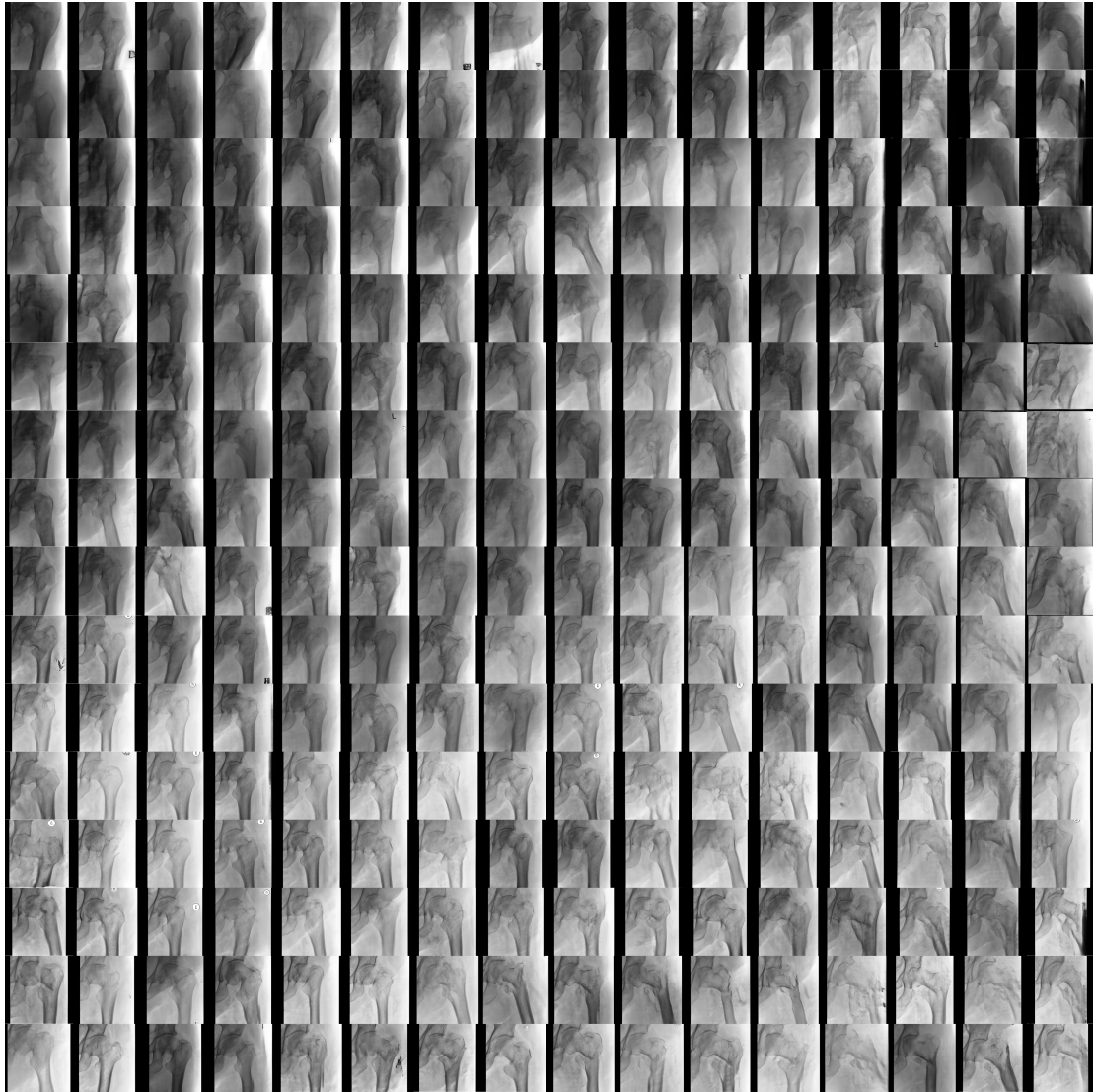


Figure A.5: t-SNE of Additional Examples of GAN-generated, Fractured Images.

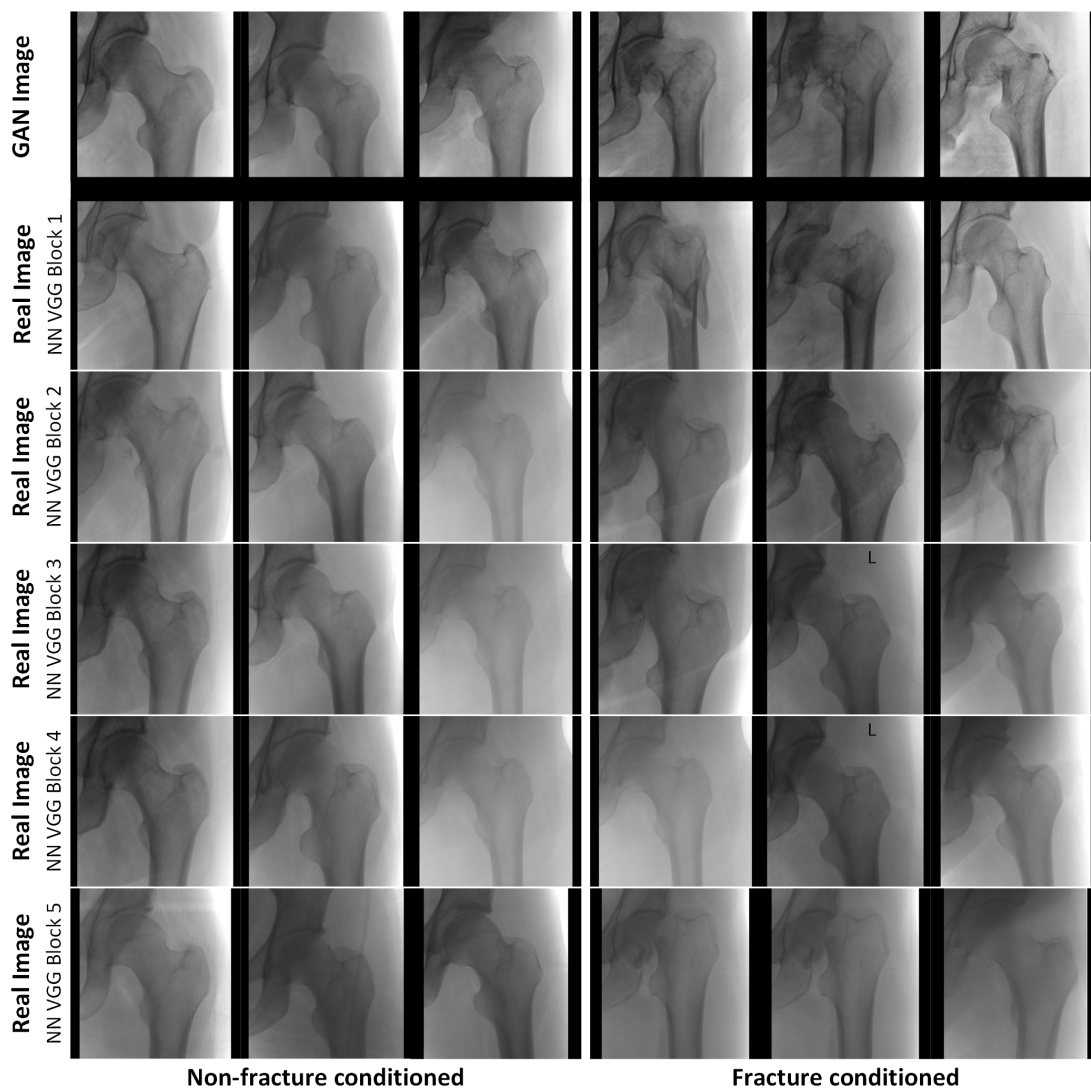


Figure A.6: t-SNE of Additional Examples of Real and GAN-generated Images.

All photographs are accurate; none of them is the truth.

Richard Avedon

B

Assessing and Extending the Generalizability of Connectivity Mapping

In Chapter 2, I began by placing connectivity mapping¹⁷⁴ within the broader context of computational drug modeling, and outlined a series of efforts to generalize connectivity mapping to unseen *chemical structures* in a zero-shot fashion. At the conclusion of the chapter, however, I noted that our analysis implied that even classical connectivity mapping methods might struggle heavily to

generalize across *cell lines*. We therefore decided to explore this question in detail. In this chapter, I describe our efforts to both *assess* and *extend* the generalizability of classical connectivity mapping strategies across cell type, technical configuration, and the functional properties of the target drugs.

B.1 INTRODUCTION

Since the release of the original connectivity map in 2006¹⁷⁴, an expansive body of research has emerged proposing various mapping methods for matching query and target signatures. These methods range in complexity from simple rank-based methods to complicated neural network architectures^{220,56,53,289,73}. In addition, several websites^{75,332,223,74} have been released that allow users to quickly query the CMAP or L1000 data to generate potential matches to biological queries of interest. For all this methodological development, however, the field lacks a systematic understanding of the circumstances under which connectivity mapping is likely to yield actionable results.

Connectivity mapping relies on at least three key assumptions: First, mapping assumes that query signatures are representative of the biological process intended to be reversed or mimicked. Second, mapping assumes that the target signatures are representative of the functional changes induced by their corresponding perturbations. Third, mapping assumes that these query and target signatures are (or can be made to be) sufficiently comparable to allow for functional matchmaking. Each of these assumptions must be adequately assessed before connectivity mapping can be expected to be effective.

The first of the above assumptions is hardly unique to connectivity mapping. Indeed, the core questions raised by this assumption – whether transcriptomics is a valid readout for the disease, whether adequate tissue samples have been acquired, whether the appropriate assays and statistical methods have been used, etc. – apply to any functional genomics study. As such, we do not focus on these questions in this work. The second mapping assumption is a question of chemical biology,

and is of primary concern when large data assets such as the connectivity map are being developed. We therefore do not focus on this question, either, but do note that several papers have published large-scale analyses of the internal consistency of the CMAP and L1000 data, in which they have raised fundamental questions about their reliability^{185,145}.

In this work, we focus on the third of the assumptions described above, which to our knowledge has not been properly addressed in the literature. This assumption can be alternatively stated as the following question: Under what circumstances can we be confident that a given set of targets can be used to cmap a specific query? To begin to answer this question, we designed and implemented a series of information retrieval tasks aimed at elucidating the circumstances under which the assumption of comparable signatures breaks down. We use these experiments to quantify the relationship between cmaping reliability and factors such as baseline cell line similarity and drug specificity. The results of these analyses are used to design simple modifications to standard connectivity mapping methods that improve the robustness of the results.

B.2 RESULTS AND DISCUSSION

We conducted a series of experiments to formally evaluate connectivity mapping using metrics from the field of information retrieval. In brief, we first identified a collection of small molecules from the LINCS dataset that had each been profiled in the same 12 cell lines at the same dose ($10\mu\text{M}$) and duration (6 hours). Details of the data acquisition and preprocessing are described in Section B.3. Perturbational signatures from one cell line were then leveraged as queries into the signatures from all other cell lines. Variations on this basic setup allow for asking specific questions of the connectivity mapping process. From a study design perspective, the primary advantage of our approach is that it provides an objective and scalable readout of performance that can be applied to even preclinical compounds. This is in contrast to evaluations centered on drug indication prediction, which

are clinically interesting but inherently very incompletely labeled and therefore difficult to fully interpret. To our knowledge, one previous study – Lim et al¹⁸⁵ – has adopted a similar setup as this study. However, their analysis was focused on the question of general reproducibility, whereas ours focuses on identifying factors that predict success; in addition, the retrieval experiments in Lim et al were reported using average rankings and ECDF plots, which are not the preferred metrics for retrieval tasks. As such, we utilize more standard information retrieval metrics such as the mean reciprocal rank and precision-recall curves¹⁹⁹. The results of our various experiments are described below.

B.2.1 DISTANCE METRICS

We began our experiments with a limited investigation of distance metrics, to inform the all subsequent experiments. We measured the performance of 8 distinct distance metrics for recovering the drug identity of perturbational signatures. As stated above, many distance metrics have been proposed for connectivity mapping, a full evaluation of which is beyond the focus of this paper. However, we attempted to include a representative collection of computationally tractable mapping methods, including the Kolmogorov-Smirnov statistic of the original paper¹⁷⁴ as well as other more recently popular metrics such as Cosine Similarity⁵³. For each similarity metric, we tested signatures from the A375 line (chosen randomly) against all others and computed the aggregate mean reciprocal rank (MRR) across all compounds in each of the other 11 L1000 cell lines. Visualization of MRR versus distance metric is found in Figure B.1A. In addition, we conducted a Friedman test of significance of differences in ranks across metrics (matched by cell lines, P-value $1.42\text{E-}14$).

Finding these tests to be highly significant, we next performed pairwise Wilcoxon Test comparisons between the each method assigned to the true drug match, and found after Bonferroni correction that the Pearson and Cosine metrics had comparable results to each other and higher performance than all other metrics. Numerical results and statistical tests are available in Tables D.6

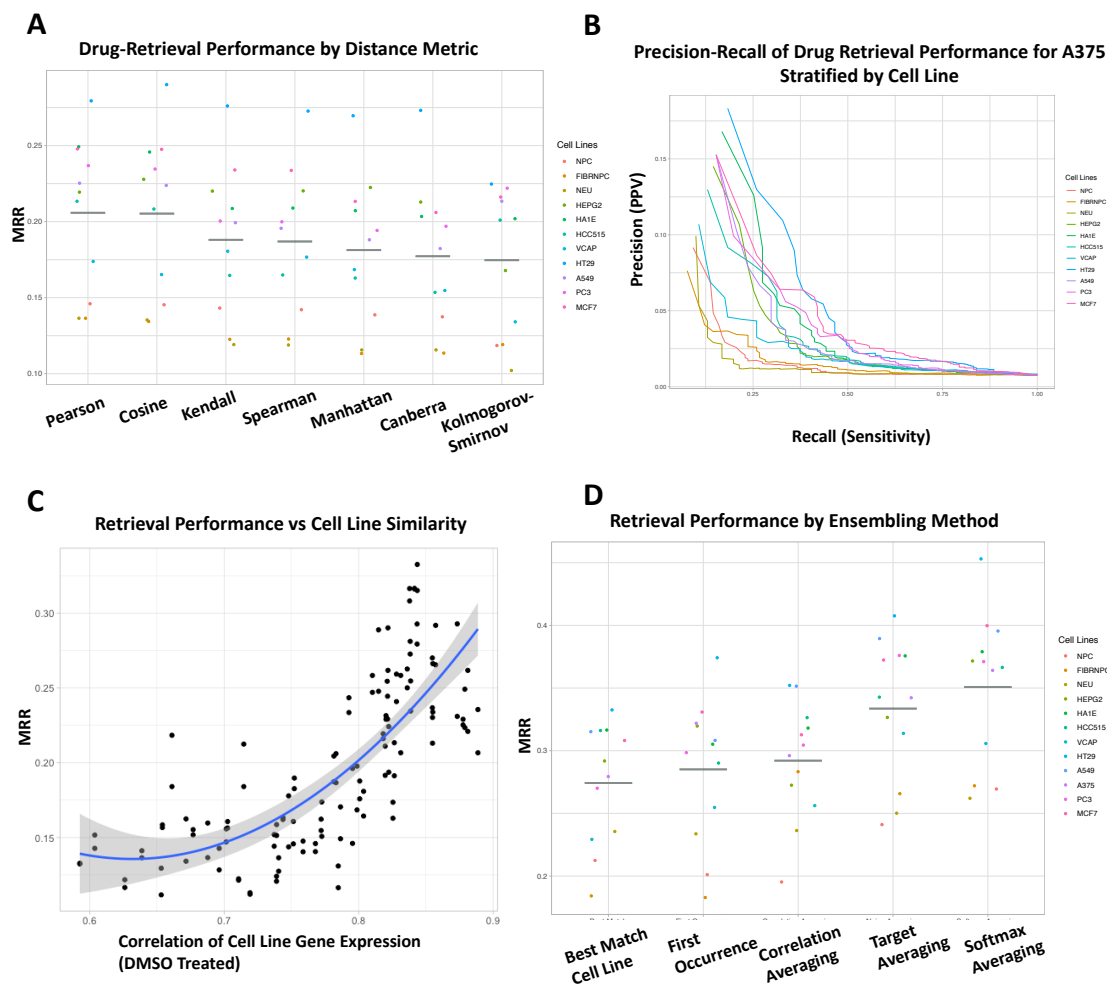


Figure B.1: A: Drug-Retrieval performance (mean reciprocal rank) as a function of distance metric, colored by cell line. B: Precision-Recall curve, stratified by target cell line, for all experiments in which A375 was the query. Equivalent plots for other cell lines in Figure D.7. C: Retrieval performance as a function of similarity (correlation at DMSO treatment) of the query and target cell lines. D: Retrieval performance of the five approaches we tested to combine results from several cell lines. Colored by query cell line.

and D.7. As can be seen in both Table D.6 and Figure B.1A, it is evident that average performance differences between cell lines were larger than average performance differences between distance metrics. To demonstrate this statistically, we compared the variances cell-line-stratified and distance-metric stratified performance averages. The ratio of variances was 16.36, which a 10,000-fold two-sided permutation test found to represent a P-Value of $2e-4$ (similarly, a F-test of difference of variances produced 95% CI of 3.43-64.52, P-Value 0.0012). The effect of cell line on performance is explored in depth in the next section.

Finally, in additional experiments (not shown), the distance metric analyses were rerun using only the top 100, 200, and 500 differentially expressed genes in each query, as executed in some papers such as XCOS⁵³, yielding identical trends but slightly worse MRRs. Taken together, we found that the most reliable retrieval results were produced when utilizing Pearson correlation over all measured genes, and we thus opted to utilize this approach for all subsequent experiments.

B.2.2 CELL-SPECIFIC EFFECTS

To investigate the role of cell lines in driving retrieval performance, we next computed retrieval ranks for all pairwise combinations of cell line query-target pairs. A characteristic precision-recall curve stratified by cell line is shown in Figure B.1B, depicting results for all target cell lines against which A375 was used as a query; equivalent plots for all cell-line query pairs are found in Figures D.7 and D.8. As with distance metrics, we computed a Friedman test of overall rankings, which determined large global differences between cell lines in their relative performances when queried against the same targets (P-Value, $8.208E-52$). In principle, such differences could be due to certain cell lines simply being of poorer experimental quality, and indeed certain cell lines (such as FIBRNP, NPC, and NEU) appeared to have poorer overall performance. However, we also hypothesized that a major driver of performance may have been that specific pairs of cell lines were inherently more similar and thus more mutually compatible for the drug-retrieval task. To test this hypothesis, we examined

whether the correlation between control signatures (DMSO-treated) from various cell lines could elucidate retrieval performance.

Figure B.1C and D.10 depict the relationship between cell line similarity and mapping performance. More specifically, it plots, for each unique pair of cell lines, the average correlation of their DMSO-treated transcriptional signatures versus the mean reciprocal rank of drugs queried from the first cell line onto the second. This relationship was also fit with a simple second order polynomial model of the form $MRR \sim CORR + CORR^2$, which yielded an Adjusted R^2 Statistic of 0.59 and P-Value of $< 2.2e^{-16}$. Table D.8 and Figure D.9 contain the coefficients and various diagnostic plots for the linear model. These results demonstrate a clear and positive relationship between baseline similarity and retrieval performance. Finally, we performed complete linkage hierarchical clustering of the cell lines based on their mutual retrieval performance (using averaged pairwise mean reciprocal ranks); results are visualized in Figure D.11, and appear to reflect biological similarity.

These observations point to a natural question, which can be framed in terms of the Bias-Variance tradeoff⁹⁹. Is it more effective to limit target cell lines to one or more highly compatible cell lines with high baseline similarity (reducing bias), or to query against a wider range of potentially less-relevant target cell lines (reducing variance)? To answer this question, we queried each cell line against the aggregation of all other cell lines using five different ensembling strategies: First, rankings were computed using only the results from the best-matched *individual* cell line, according to baseline gene expression correlation. Second, results from all cell lines were pooled and rankings for each potential drug identity were taken using the first occurrence of that drug in *any* cell line. Third, ranks were assigned after averaging the correlation scores from each drug across *all* target cell lines. Fourth, target perturbational signatures from all cell lines were averaged together equally and then queried against as a single aggregate target cell line (one per drug). Finally, target perturbational signatures were averaged together in a weighted fashion, with weightings determined by taking the temperature-weighted softmax (with a weight of 10) of the correlation between the query cell line

and each target cell line. Results of these experiments are in Figure B.1D and Tables D.9 and D.10. In brief, approaches that favor variance reduction by averaging across multiple cell lines outperform methods that focus on results from individual cell lines. The best-performing method, however, was our hybrid approach that sought to mitigate bias by weighting the average according to baseline cell line similarity. In Figures D.12 and D.13, we report results for each of the above ensemble methods described above, but with the additional modification where results are ensembled only from a subset of the most highly correlated cell lines; in brief, we find that our weighted-average approach performs best with all cell lines, with other methods performing best (but still not as strongly as our weighted approach) when using a subset of 7-8 cell lines.

B.2.3 DRUG-SPECIFIC EFFECTS

We next examined the relationship between functional drug specificity and mapping accuracy. To do so, we computed drug-level averages of the mean reciprocal ranks from all pairwise mapping experiments described in Section B.2.2. We then compared these drug-level ranking metrics with the average number of genes identified as significantly differentially expressed after perturbation with the corresponding drugs (details in Section B.3). The relationship between these two values was modeled as a simple linear model $MRR \sim \log(N_{\text{Genes}})$, which yielded an Adjusted R^2 Statistic of 0.55 and P-Value of $< 2.2e^{-16}$, and is visualized in Figure B.2A. Table D.11 and Figure D.14 contain the coefficients and various diagnostic plots for the linear model.

Given that query signatures affecting many genes were more likely to be successfully retrieved, we wanted to see if *target* signatures affecting many genes were more or less likely to be returned as false positives for *non-matching* signatures. Figure B.2B depicts the average rank for each target signature in response to non-matching query signatures. As can be seen, small molecules affecting many genes tend to be returned at earlier ranks even if they do not match the query signature. This relationship was modeled with a linear model $\text{rank}_{\text{avg}} \sim \log(N_{\text{Genes}})$, which yielded an Adjusted R^2

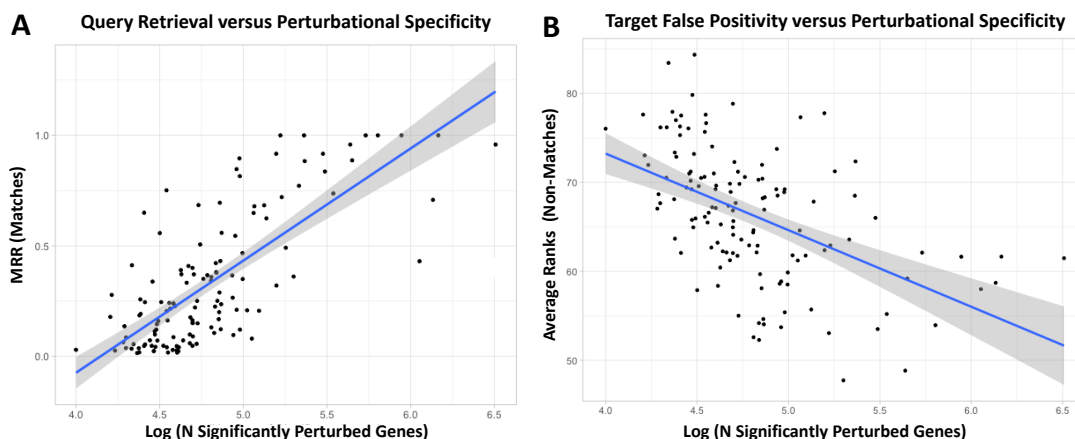


Figure B.2: Left: Drug-Retrieval performance (mean reciprocal rank) as a function of the number of genes significantly perturbed by the drug. Right: Average ranks of target compounds in response to *non-matching* queries versus drug specificity.

Statistic of 0.24 and P-Value of $5.22e^{-10}$. Table D.12 and Figure D.15 contain the coefficients and various diagnostic plots for the linear model. A natural follow-up question to this finding is whether this trend holds even if a subset of genes are used in the query, as recommended in Cheng et al ⁵³; we therefore reran this experiment using only the 200 most significantly altered genes in each query signature, which produced nearly identical results (a Wilcoxon Rank Sum comparing the two sets of rankings produced a P-Value of 0.9675; slope estimate differed by 0.8%).

Taken together, these drug-level results have implications for both the planning and the interpretation of connectivity mapping efforts. First, prior to conducting a connectivity mapping experiment, it should be recognized that query signatures whose transcriptional effects are specifically concentrated in a small subset of genes may be less likely to match to corresponding target signatures using traditional cmapping methods. Second, when interpreting the results of a cmapping experiment, the relative credence afforded to a putatively matching compound should consider the total number of genes that that compound affects. In short: more selective queries are likely harder to match, and more promiscuous targets are less trustworthy.

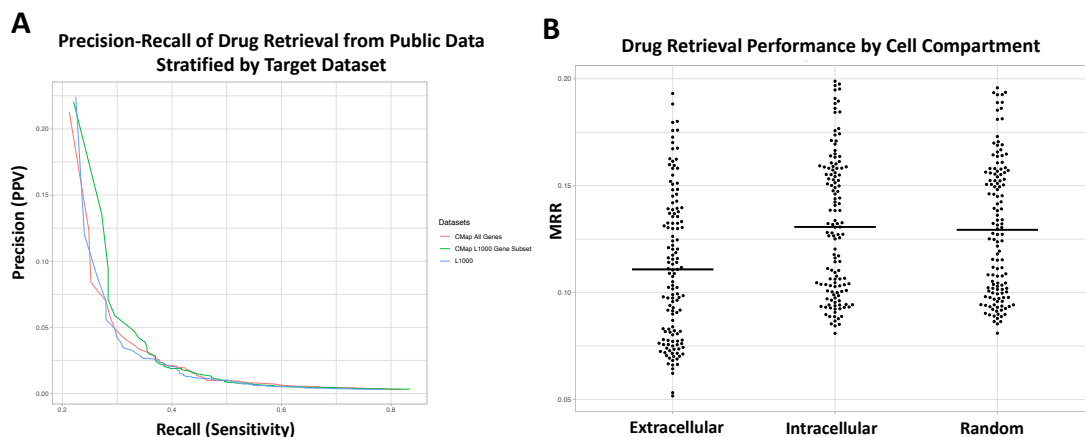


Figure B.3: A: Precision-Recall curve for all drugs taken from non-LINCS/CMAP public datasets. Each curve represents a target dataset (CMap, LINCS, or CMap subset to the L1000 genes). B: Retrieval performance when cmapping is limited to genes whose proteins are located in specific cellular locations.

B.2.4 ASSAY-SPECIFIC EFFECTS

We sought to measure assay-specific connectivity mapping performance, and in particular assess the viability of cmapping when query gene expression signatures were assayed using different technology than the target signatures. To do so, we queried the Cmap (Affymetrix assay) dataset utilizing signatures from the LINCS (L1000 assay) dataset, and vice versa. In addition, we queried both the CMAP and LINCS using signatures from independently conducted, public signatures downloaded from GEO (details in Section B.3). The results, presented in Figures B.3A and D.16 and Tables D.14 and D.15, indicate relatively poor but better-than-random performance generalizing across assays.

Finally, we hypothesized that the selection of genes to include in connectivity mapping assays might influence performance. In particular, we hypothesized that certain genes would be more likely to be stable across cell types and cellular environments and therefore more likely to provide accurate connectivity mapping results. In particular, we hypothesized that intracellular genes would show more consistent responses across cell lines, because they include the cells engaging in universal rather

than cell-specific functions. To test this hypothesis, we randomly sampled 100 intracellular and 100 extracellular genes (with compartmental labels taken from the Gene Ontology⁵⁸), and performed connectivity mapping across all pairs of cell lines in LINCS using only those genes. This process was repeated 100 times and results were averaged across all iterations to produce one average score per drug across all experiments. The results, depicted in Figure B.3B and Table D.13, demonstrate that cmapping using extracellular genes results in worse performance, which is consistent with our hypothesis. In the future, more dedicated efforts to design assays comprised of genes that are stable across cell types – but still responsive to drugs – may help support more robust connectivity mapping.

B.3 DATA, PREPROCESSING, AND IMPLEMENTATION

DATA ACQUISITION AND PREPROCESSING

All experiments leveraged uniformly processed versions of the LINCS and CMap datasets that have been distributed by co-author Alex Pickering via the R package *ccdata*²⁴⁹. In brief, data from both datasets was downloaded in the rawest available format, and processed following best practices including surrogate variable analysis¹⁷⁸ to correct for non-treatment related variables. Final differential expression sequences were computed using the *limma* package²⁹¹. For the drug-specificity analysis, we tabulated the total number of genes differentially expressed by counting the number of genes with unadjusted P-values of less than 0.05.

In order to mitigate the risk of unmeasured confounding effects, we limited the small molecules in each experiment to drugs that had been measured in all pertinent cell lines, at the same dose (10 μ m) and for the same duration (6 hours). 131 drugs met these criteria from the L1000 data, and these drugs were used for all experiments. Included cell lines from the original CMAP data were PC3 (Human Prostate Adenocarcinoma cell line), MCF7 (Human Breast Adenocarcinoma cell

line) and HL60 (Acute Promyelocytic Leukemia cell line). Included cell lines from the L1000 data were NPC (Neuroprogenitor cell line), FIBRNP (Fibroblast Derived NPC cell line), NEU (Neuronal cell lines), HEPG2 (Human Hepatocellular Carcinoma cell line), HA1E (Human kidney epithelial immortalized cell line), HCC515 (Human non-small cell lung adenocarcinoma cell line), VCAP (Human Prostate Metastatic Cancer cell line), HT29 (Human Colorectal Adenocarcinoma cell line), A549 (Human non-small cell lung carcinoma cell line), A375 (Human Malignant Melanoma cell line), PC3 (Human Prostate Adenocarcinoma cell line), and MCF7 (Human Breast Adenocarcinoma cell line).

For additional experiments, we downloaded datasets from the Gene Expression Omnibus (GEO), using the crossmeta R package²⁵⁰. We obtained the GEO studies using 9 drugs present in both L1000 and CMap, which were Tretinoin, Dexamethasone, Doxorubicin, LY-294002, Metformin, Progesterone, Quercetin, Resveratrol, and Vorinostat. Each study had several signature perturbations influenced by the drug used in the study. A full list of GEO accession numbers are available in D.16. In addition to these data, we downloaded the gene expression data from the manual annotation metadata of the CREEDS database³³³. All these data were reprocessed from scratch with the same pipeline as was used to process the CMAP microarray data.

B.3.1 DISTANCE METRICS

The 8 different distance metrics were Pearson correlation, Cosine similarity, Spearman correlation, Kendall correlation, Pearson correlation on PCA data, Manhattan distance, Canberra distance and KS enrichment score for both LINCS and CMap data on cell line pairs and on the whole dataset of chosen cell lines. All metrics were implemented from scratch in the R programming language²⁵⁹, with the exception of the Kolmogorov-Smirnov enrichment score, which was computed using the R package HTSanalyzeR³³⁰, which implements the KS statistic to compute an enrichment score as described by Subramanian et al³⁰¹.

B.3.2 SOFTWARE

In addition to the packages specifically mentioned above, we leveraged the R packages `ggplot2`³³⁹, `ggrepel`²⁸⁸, `dplyr`³⁴¹, `tidyverse`³⁴⁰ and `lindia`¹⁷⁷ for data visualization. All code will be made available on Github.

B.4 CONCLUSION

In this chapter, I have described a series of experiments designed to probe the various factors that drive the performance of connectivity mapping methods. Through these experiments, we identified that the baseline similarity between the query and target cell lines appears to be a powerful *and* actionable predictor of cmapping success. Similarly, drug-specificity and the technical comparability of the assays used to measure transcriptional signatures also have a marked impact on performance. Distance metrics used to rank queries, in contrast, have comparatively less effect on connectivity mapping performance, despite representing a major focus of publications within the field of research.

*The art of war teaches us to rely not on the likelihood
of the enemy's not coming, but on our own readiness
to receive him; not on the chance of his not attacking,
but rather on the fact that we have made our position
unassailable.*

Sun Tzu



Additional Analyses of Adversarial Attacks in the Clinical Domain

C.1 METHODOLOGICAL DETAILS AND RESULTS OF ADVERSARIAL PATCH ATTACKS

The learning process of the adversarial patch $\hat{p}$ uses a variant of the expectation over transformation algorithm (originally proposed by Athalye et al. ¹⁴), which can be expressed as the following maxi-

mization:

$$\hat{p} = \operatorname{argmax}_p \mathbb{E} [\log p_{Y|X}(\hat{y} | \mathcal{A}(p, X, L, T))] \quad (\text{C.1})$$

where $p_{Y|X}(\cdot|\cdot)$ represents the probability output from the classifier given the input image X , L is the location of the patch, T is the transformation (rotation and scaling), $\hat{y}$ is the target label (for our application binary label), and $\mathcal{A}(\cdot, \cdot, \cdot, \cdot)$ is the deterministic mapping of given image, patch, location and transformation into the adversarially patched input image. The expectation is over the locations, transformations and input images, thus allowing the resulting patch to be ‘universal,’ because it was trained over the entire training set and robust since training averaged over various location and transformations. We deployed adversarial patches with scaling parameter 0.4.

Finally, as a control, we implemented a naive patch attack using natural images, as in Brown et al.⁴⁰. To make this as strong a baseline as reasonably possible, we built our natural image patches using the images assigned by the model with the highest probability of the target class. Resultant patches were then applied to the test set images with random rotation and scaling factor 0.4, so match the adversarial patch.

C.2 EXISTING ADVERSARIAL DYNAMICS IN THE UNITED STATES HEALTHCARE SYSTEM

Cutting-edge adversarial attacks have yet to be found in the healthcare context, though the use of less formalized adversarial practices is extremely common. This existing activity is a strong indicator that the incentives for more sophisticated adversarial attacks may be already in place. As a case study to illustrate these existing behaviors, we look to the modern U.S. medical billing industry.

Under the United States’ healthcare model, some of the most direct impacts of machine learning algorithms come in the context of insurance claims approvals. Billions of medical claims are processed each year, with approvals and denials directing trillions of dollars and influencing treatment

decisions for millions of patients. In addition to dictating the availability of patient care, claims approval is vested with competing financial interests, with providers seeking to maximize and payors seeking to minimize reimbursement.¹⁵⁷ Given the volume and value of processing medical claims, it is unsurprising that many providers engage in creative and often fraudulent practices to increase their revenue³⁴⁹. For their part, insurance companies and their contractors have invested in extensive machine learning infrastructure for billing code processing. While much of our discussion highlights financial incentives specific to the fee-for-service model in the US, the implications of algorithmic vulnerabilities have broad relevance.

Medical claims codes attached to a patient visit determine reimbursement for a patient that visit after they have been approved by a payer. To evaluate these claims, payers typically leverage automated fraud detectors, now powered increasingly by machine learning. Health care providers have long exerted influence on payers' decisions (the algorithmic outputs) by shaping their records and accompanying codes of patient visits (the inputs)³⁴⁹.

At the extreme of this tactical shaping of a patient presentation is medical fraud, a \$250 billion industry¹⁴⁶. While some providers may submit overtly fictitious medical claims, misrepresentation of patient data often takes much more subtle forms. Examples of subtle but clearly fraudulent behavior include For example, intentional upcoding is, the practice of systematically submitting billing codes for services related to, but more expensive than, those that were actually performed. This practice is rampant across today's medical system and is just one of many questionable billing practices deployed in clinical practice today. Some physicians, for example, are inclined to report exaggerated anesthesia times to increase revenue^{62,305}. In other circumstances, subtle billing code adjustments fall within a gray zone between fraud and well-intentioned best practices. In one striking example, the website of the Endocrine Society specifically recommends that providers do not bill for the International Classification of Disease (ICD) code 277.77 (Metabolic Syndrome) in patients with obesity, as this combination of code and condition is likely to result in a denial of coverage²⁶⁸. Instead,

the Society recommends billing for codes corresponding to specific diseases that make up metabolic syndrome, such as hypertension. In other words, providers are not encouraged to add fraudulent claims, but are encouraged to avoid adding a true claim that an insurance company would be likely to reject in combination with another. This specific recommendation is arguably motivated to serve the patients seeking coverage, not only the doctors receiving reimbursement. However, it highlights both a moral gray zone and the type of strategy that providers might use to achieve the same end result as upcoding without ever committing overt fraud.

As the machine learning toolkit employed by insurance companies and their contractors continues to expand, the same dynamics that favor creative billing practices in the present may expand to include more sophisticated adversarial attacks. Adversarial methods could allow billing teams to scale-up up-coding practices without getting flagged by fraud detectors. Many insurance companies are beginning to require other data types such as imaging and text to prove that claims are valid. As they do so, other styles of adversarial attacks may be employed as well to try to continue to dodge detection,.

For example, if an insurance company requires that an image from a mole be run through a melanoma classifier before approving reimbursement for an excision, fraudsters may at first be inclined to submit moles from different patients to achieve approval. If insurance companies then begin utilizing human audits or technical tests to try to ensure the images are coming from the correct patient, the next round would be to move to full adversarial attacks with imperceptible alterations such as at the top of Figure 3.1a. Simpler techniques such as the rotation in Figure 3.1b could constitute an ethical gray zone – given that a dermatologist could in theory hold the camera at any angle.

Potential applications of adversarial attacks in the medical context go far beyond insurance fraud, encompassing a spectrum of motivations. For instance, many adversarial attacks could be motivated by a desire to provide high quality care. A hypothetical illustration can be drawn from the opioid crisis. In response to rampant over-prescription of opiates, insurance companies have begun using

predictive models to deny opiate prescription filings based on risk scores computed at the patient or provider level. What if a physician, certain that she had a patient who desperately needed oxycontin but would nonetheless run afoul of the prescription authorization algorithm, could type a special pattern of algorithmically selected billing codes or specific phrases into the record to guarantee approval?

Companies might face temptations in the context of drug and device approvals. Regulatory bodies including the FDA have expressed interest in using algorithmic biomarkers as end-points in clinical trials and other approval processes. If this is realized, adversarial examples could provide a means for companies to bias trial out-comes in their favor. For example, if a regulator requires matched images or wearable readouts from each patient before and after treatment, trialists could inject adversarial noise into post-treatment data, securing the algorithmically-measured results they desired. Motivations could be complex – while some trialists would be motivated by the potential for a big payday, others might turn to adversarial attacks to “adjust” borderline trial results for products that might save lives.

C.3 DISTINCTIVE TECHNICAL SOURCES OF VULNERABILITY TO ADVERSARIAL ATTACKS ON MEDICAL MACHINE LEARNING SYSTEMS

GROUND TRUTH IS OFTEN AMBIGUOUS. Compared to most common image classification tasks, the ground truth in medical imaging is often controversial, with even specialty radiologists disagreeing on well defined tasks^{232,182,39}. As such, if end-users selectively perturb images for which it is difficult to establish the true diagnosis, they can make it extremely difficult to detect their influence through even expert human review. Somewhat ironically, it is these borderline cases are where deep learning is likely to be most valuable.

MEDICAL IMAGING IS HIGHLY STANDARDIZED. Compared to other domains of computer vision, medical imaging is extremely standardized, with images generally captured with pre-defined and well-established positioning and exposure²⁸⁵. As such, medical adversarial attacks do not need to meet the same standards of invariance to lighting and positional changes as attacks on other real-world systems such as self-driving cars. This is potentially important, as some have argued that dynamic viewing conditions imply that ‘there is a good prospect that adversarial examples can be reduced to a curiosity with little practical impact’¹⁹³.

COMMODITY NETWORK ARCHITECTURES ARE OFTEN USED. Nearly all of the most successful published methods in medical computer vision have consisted of the same fundamental architecture: one of a small set of pretrained ImageNet models that was fine-tuned to the specific task^{120,329,81}. This lack of architectural diversity could make it easier for potential attackers to build transferable attacks against medical systems. By the same token, given the importance of peer review and publication in validating and approving medical diagnostics, it is likely the architectures of most medical image classification models will be public for the sake of transparency, allowing for more targeted adversarial attacks.

MEDICAL DATA INTERCHANGE IS LIMITED AND BALKANIZED . Five electronic health record (EHR) vendors constitute about half of the market and hundreds of others serve the other half. Even within an EHR vendor, data sharing is spotty and the terminologies and their semantics vary considerably from one implementation to another. On the one hand, this means that there are no universally shared mechanisms for authentication, verification of message integrity, data quality metrics, nor mechanisms for automated oversight. On the other hand, this allows healthcare providers to customize their EHR’s, billing, and other information technology systems in ways that are opaque to most external auditors using one-size-fits-all tools and methods.

HOSPITAL INFRASTRUCTURE IS VERY HARD TO UPDATE. Medical software is often implemented within monolithic enterprise-wide proprietary software systems, making updates, revisions and fixes expensive and time-consuming. For context, consider the coding dictionaries used to classify patients' diseases, the International Classification of Disease (ICD) system. As recently as 2013, most hospitals were operating using the ninth edition of this coding scheme, published in 1978, despite the fact that a revised version (ICD-10) was published in 1990. All told, the conversion to the ICD-10 coding scheme has been estimated to cost major health centers up to \$20 million *each* and require up to 15 years²⁷⁴. Others decided in the early 2000s that it would be preferable to skip the 1990 schema entirely and wait for ICD-11, despite the fact that its release wasn't scheduled until 2018. Thus, vulnerabilities present in medical software are likely to persist for years due to the difficulty and expense involved with update hospital infrastructure.

MEDICINE CONTAINS A MIX OF TECHNICAL AND NON-TECHNICAL WORKERS. Compared to many other industries, medicine is extremely interdisciplinary and mostly comprised of members who lack a strong computational or statistical training background. For example, in the case of self-driving cars, the teams developing computer vision systems are likely to be led and staffed primarily by engineers, if not computer scientists. In contrast, since the clinical usability of medical imaging systems is also extremely important, hospitals are likely to lean heavily on physician-researchers in developing these systems, who tend to lack robust computational expertise²⁰⁰.

BIOMEDICAL IMAGES CARRY PERSONAL SIGNATURES THAT COULD BE USED TO DEFEND AGAINST MANY SIMPLER ATTACKS, BUT NOT AGAINST ADVERSARIAL EXAMPLES. One potential alternative to adversarial attacks on part of malicious end-users would be to simply substitute in true images of the target class. However, biomedical images – including retinal images, X-rays, and skin photographs – are often as unique to their owners as fingerprints^{256,9,210}. This provides several

advantages against substitution attacks: First, algorithms could be designed to check input images against prior images from the same patient, to ensure that their identities match. Second, algorithms could query against the database of previously submitted images and flag any inputs that are likely from the same patients as previous entries. Both strategies would make it more difficult for fraudsters to continually execute substitution attacks against ML systems, but neither would defend against adversarial attacks, which needn't change the personal identifiers in the image. By analogy, adversarial attacks applied to billing codes or medical text could serve to manipulate reimbursement algorithms by selecting combinations of codes or words that are individually truthful but in combination yield anomalously high reimbursement.

THERE ARE MANY POTENTIAL ADVERSARIES. The medical imaging pipeline has many potential attackers and is thus vulnerable at many different stages. While in theory one could devise elaborate image verification schemes throughout the data processing pipeline to try to guard against attacks, this would be extremely costly and difficult to implement in practice.

C.4 STRATEGIES FOR DEFENDING AGAINST ADVERSARIAL ATTACKS

ALGORITHMIC DEFENSES against adversarial examples remain an extremely open and challenging problem. We defer a full discussion of the extremely rapidly evolving field of adversarial defenses to the primary literature^{154,114,166,195,238,308,315,239,166,76,261,41}. Unfortunately, despite the explosive emergence of defense strategies, there does not appear to be a simple and general algorithmic fix for the adversarial problem available in the short term. For example, one recent analysis investigated a series of promising methods that relied on gradient obfuscation, and demonstrated that they could be quickly broken¹³. Despite this, we also note that principled approaches to adversarial robustness are beginning to show promise. For example, several papers have demonstrated what appears to be both high accuracy and strong adversarial robustness on smaller datasets such as MNIST,^{195,154} and

there have also been several results including theoretical *guarantees* of adversarial robustness, albeit on small datasets and/or with still-insufficient accuracy¹⁶⁶. Generalized attempts at algorithmic robustness are promising, but have yet to provide methods that can demonstrate high levels of both accuracy and adversarial robustness at ImageNet scale. However, **domain-specific algorithmic** defenses such as dataset-specific image preprocessing have been shown to be highly effective on some datasets¹¹⁶. In this light, particularly given the highly standardized image capture procedures in biomedical imaging, we feel that medical-domain-specific algorithmic defenses offer an important and promising area of future research.

INFRASTRUCTURAL DEFENSES against clinical adversarial attacks include methods deployed to prevent potential bad actors from altering medical images – or at least make it easier to confirm image tampering if adversarial examples are suspected. For example, imaging devices could immediately store a hashed version of any image they generate, which could subsequently be used as a reference. Likewise, raw clinical images could be processed and analyzed on a third-party system to prevent any possible systemic manipulation by payers or providers. This family of approaches to standardized best practices is reminiscent of the system of Clinical Laboratory Improvement Amendments (CLIA), a set of federal policies that regulates the process by which clinical laboratory samples are handled and analyzed in the United States²³⁴. Given that algorithmic defenses against adversarial attacks are still very much an area of research, we feel that infrastructural defenses should be strongly considered for all medical classifier systems that could carry incentives for adversarial attacks. However, implementing healthcare system-wide standardization to this end represents an immense challenge that will require buy-in from both the medical and CS communities.

D

Supplementary Material

D.1 CHAPTER 1

D.1.1 FURTHER DETAILS ON SUB-GNN

In this section, we outline additional details and design decisions for the implementation of SUB-GNN. In Section D.1.1, we provide an algorithmic overview of our method, and in Section D.1.1, we provide further motivation and details on triangular random walks and the structural anchor patch embedder algorithm.

MESSAGE PASSING ALGORITHM

For brevity, Algorithm 1 summarizes the forward pass of SUB-GNN for a single subgraph S . While the algorithm demonstrates how to embed subgraph S , we want to emphasize that SUB-GNN learns representations for every subgraph $S \in \{S_1, S_2, \dots, S_n\}$. In practice, we extend Algorithm 1 to multiple subgraphs using mini-batching. Note that any aggregation function that operates over an unordered set of vectors can be used for AGG_M , AGG_C , AGG_L , and/or READOUT . Our implementation leverages the sum operator for AGG_M and READOUT and the concat operator for AGG_C and AGG_L . We empirically found that applying attention³⁵⁸ over the individual subgraph component representations did not yield any performance gains over summation for READOUT .

TRIANGULAR RANDOM WALKS

In Section 1.4.2, we leverage triangular random walks² to (1) sample structure anchor patches via φ_s and (2) embed the structure anchor patches via ψ_s . Refer to Algorithm 2 for a summary of the structure anchor patch embedder.

Triangular random walks are parameterized by $\beta \in [0, 1]$ and yield a sequence of nodes $X_0, X_1, \dots$. β determines whether triangles or non-triangles will be privileged during sampling. A node z is said

to be a triangular successor to nodes x and y if x, y, z would form a triangle. The triangular random walk samples triangular successors with probability β and non-triangular successors with probability $1 - \beta$. When performing an internal random walk, we initialize $P[X_0 = x] = 1/|\mathcal{I}|$ and $P[X_1 = y|X_0 = x] = 1/d_I(x)$, where $d_I(x)$ is number of edges from x to other nodes in $\mathcal{I}$. Similarly, we initialize a border random walk with $P[X_0 = x] = 1/|\mathcal{B}|$ and $P[X_1 = y|X_0 = x] = 1/d_E(x)$ where $d_E(x)$ is number of edges from x to nodes in $\mathcal{E}$.

We then define the transition probability for any subsequent step of the random walk as follows: given current node y and preceding node x , the probability of transition to node z is defined as $P[X_{t+1} = z|X_t = y, X_{t-1} = x] = [\beta(1/|T|)1_{X_{t+1} \in T} + (1 - \beta)(1/|U|)1_{X_{t+1} \in U}]$ where T is the set of triangular successors of nodes x and y and U is the set of non-triangular successors. More precisely, $T(X_t, X_{t-1}) = N(X_t) \cap N(X_{t-1})$ and $U(X_t, X_{t-1}) = N(X_t) \setminus N(X_{t-1})$.

D.1.2 FURTHER DETAILS ON DATASETS

We proceed by describing the construction and processing of synthetic as well as real-world datasets. Note that we provide all datasets in our SUB-GNN code release. Upon publication, we will share SUB-GNN implementation as well as all datasets and baseline algorithms with the community through a public project website, which will be accompanied by a Github repository with relevant data loaders.

SYNTHETIC DATASETS FOR SUBGRAPH CLASSIFICATION

GENERATING BASE GRAPHS For DENSITY, CUT RATIO, and COMPONENT, we start with a base Barabási-Albert graph, where the number of preferentially attached edges m , is 5 for DENSITY and CUT RATIO and $m = 1$ for DIAMETER and COMPONENT. In CORENESS, the base graph is a duplication-divergence graph where the probability of retaining the edge of the replicated node is

0.7. Refer to Table D.1 for properties of the base graphs.

GENERATING SUBGRAPHS We introduce three methods for generating subgraphs: (1) the **PLANT** method, which searches for $n > 1$ common nodes and $e \geq 1$ shared edges between the base graph and the new subgraph, and plants the subgraph on the base graph via the union of the node and edge sets; (2) the **STAPLE** method, which adds an edge between a randomly sampled node in the base graph and a node in the new subgraph; and (3) the **BFS** method, which creates subgraphs in the base graph by performing breadth first search with a specified max depth d from randomly selected start nodes. Figure D.1 describes the **PLANT** and **STAPLE** approaches in more detail.

We construct subgraphs with various desired graph properties: density, cut ratio, k -coreness, and number of components^{355,283}. Density is defined as $D = (2 \cdot |E|) / (|V| \cdot |V - 1|)$ for our undirected graphs. We define cut ratio as the proportion of edges shared between the subgraph S and the rest of the graph G , specifically $CR(S) = |\mathcal{B}_S| / (|S| |G \setminus S|)$, where $\mathcal{B}_S = \{(u, v) \in E | u \in S, v \in G \setminus S\}$. The k -core of a graph is the maximum subgraph with a minimum degree of at least k . A connected component is defined as a set of nodes in a subgraph such that each pair of nodes is connected by a path.

SUBGRAPH CLASSIFICATION TASKS For each subgraph classification task, we generate subgraphs that vary along a specific network property (e.g. density, cut ratio). Labels are then generated by binning the network property values into two or three quantiles: (1) **DENSITY** has 250 subgraphs that are constructed via **BFS** with $d = 3$ to have low, medium, or high density. (2) **CUT RATIO** has 250 subgraphs constructed by **PLANTING** complete graphs onto the base graph such that the subgraphs have low, medium, or high cut ratio. (3) **CORENESS** has 221 subgraphs that are created using the duplication-divergence model and constructed via **PLANTING** onto the base graph. These subgraphs' labels (e.g., low, medium, or high coreness bins) are assigned by calculating the average k -coreness

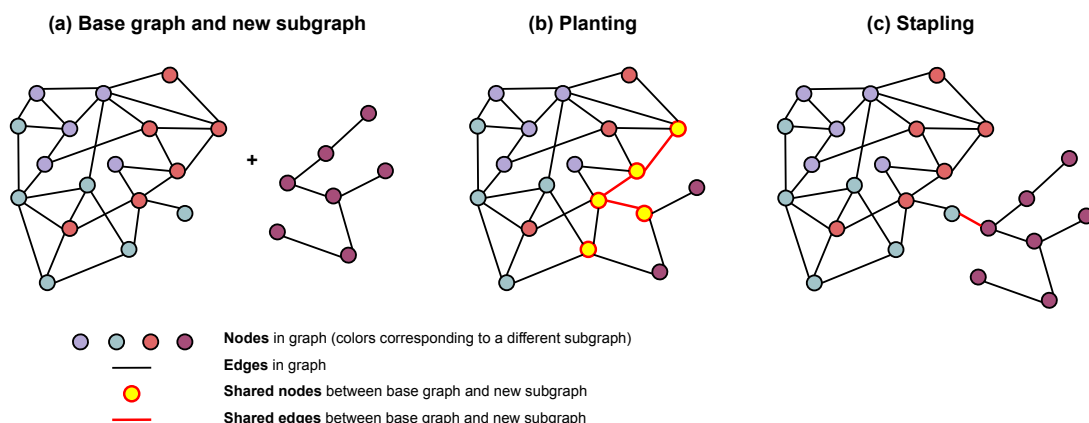


Figure D.1: Procedures for generating synthetic graphs. **(a)** Base graph and new subgraph to be added. **(b)** New synthetic graph with a “planted” subgraph. **(c)** New synthetic graph with a “stapled” subgraph. The base graph **(a)** has 3 subgraphs, and the new graphs **(b)-(c)** each have 4 subgraphs, indicated by the colors red, light blue, light purple, and dark purple. The yellow nodes and bright red edges are those shared between the base graph and the new subgraph.

of all nodes in the subgraph. Finally, (4) **COMPONENT** contains 250 Barabási-Albert subgraphs that are **STAPLED** onto the base graph, and the associated label is the number of connected components in the subgraph (single or multiple components). Subgraphs for synthetic datasets are split into train, validation, and test sets via a 50/25/25 split. See Table D.2 for properties of subgraphs in the datasets.

NOVEL, REAL-WORLD DATASETS FOR SUBGRAPH CLASSIFICATION

PPI-BP DATASET This task is from molecular biology. Given a group of genes known to associate with some common biological process, we predict their collective function (e.g. metabolism, transport, etc.). The base graph of the **PPI-BP** dataset is a human protein-protein interaction (PPI) network³⁷⁷, which contains 17,080 nodes and 630,226 edges. Nodes represent human proteins specified by their Entrez IDs¹⁹⁶, and edges exist between nodes if there is physical interaction between the proteins. Subgraphs are collections of proteins in the PPI network that are involved in the same biological process (e.g., “alcohol biosynthetic process,” “mRNA cleavage,” etc.). These subgraphs

were obtained from the Gene Ontology (GO) gene sets from the Molecular Signatures Database (MSigDB)³⁰¹. PPI-BP subgraphs labels are obtained from the “GO Slim” resource in the GO Biological Process Ontology^{59,12}, which groups narrow processes into broader categories: metabolism, development, signal transduction, stress/death, cell organization, and transport. Gene sets are limited to those containing at least five genes, and we exclude any gene sets that comprised 70% or more of the genes in any other gene set. Subgraphs are split evenly at random into training, validation, and test sets.

HPO-NEURO DATASET This is a clinical diagnostic task from the field of neurology. Given a collection of phenotypes intelligently sampled from a rare disease database, the task is to predict to subcategory of neurological disease most consistent with that combination of phenotypes. The base graph is a knowledge graph containing phenotype and genotype information about rare diseases. Nodes are phenotypes (symptoms), and edges exist between phenotypes if (1) they are caused by a mutation in a shared gene according to DisGeNET²⁵¹, HPO-A², or Orphanet¹⁹⁷ or (2) an edge exists between the phenotypes according to the Human Phenotype Ontology (HPO)²⁷⁰. Each subgraph consists of a set of phenotypes associated with a rare monogenic disease. The subgraphs contain noisy phenotypes unrelated to the disease, distractor phenotypes related to incorrect but similar diseases, and less specific phenotypes generated by walking up the HPO hierarchy (*e.g.*, from “arachnodactyly” to “abnormality of the fingers”). Together, these simulate the imperfect diagnosis process and make the diagnosis task more realistic. Subgraph labels are the diagnosis categories. This is a multi-label dataset consisting of 10 neurological disease categories: neurodegenerative, epilepsy, ataxia, genetic dementia, central nervous system malformation, intellectual, neurometabolic, movement, peripheral neuropathy, and neuromuscular disease. Subgraphs are split by disease into train, validation, and test sets via an 80/10/10 split.

HPO-METAB DATASET This is a clinical diagnostic task for rare metabolic disorders, defined similarly to HPO-NEURO, but for a different collection of diseases and disease categories. The base graph is identical to the base graph in HPO-NEURO, and the subgraphs consist of a set of phenotypes associated with a rare monogenic metabolic disease. The HPO-METAB dataset contains 6 labels corresponding to types of metabolic disease: lysosomal, energy, amino acid, carbohydrate, lipid, and glycosylation. Subgraphs are split by disease into train, validation, and test sets via an 80/10/10 split.

EM-USER DATASET This is a user profiling task. Given the workout history of a user represented by that user’s subgraph, we want to predict characteristics of that user. The base graph for is a social fitness network from Endomondo²²⁹, where the nodes represent workouts, and edges exist between workouts completed by multiple users. Each subgraph is represented as an Endomondo subnetwork that make up the user’s workout history, and the labels are characteristics about the user (here, gender). The Endomondo base graph is a co-occurrence network and as such it contains cliques (*i.e.*, small fully connected networks) of highly popular combinations of workouts. We identify co-occurrence cliques and use random network sampling to break up cliques in the base graph^{159,269}. Examples are split evenly by workout into train, validation, and test sets.

See Table D.1 and D.2 for further properties of all datasets.

D.1.3 DETAILS ON EMPIRICAL EVALUATION

Baseline models and SUB-GNN were evaluated using Micro F1 and AUROC. Both metrics were implemented using Scikit-learn (Version 0.20.2). AUROC scores for synthetic datasets are in Table D.3, and results for real-world datasets are in Table D.4. AUROC scores for the SUB-GNN channel ablation analysis are in Table D.5.

Table D.1: Properties of **base graphs** in synthetic and real-world datasets.

Dataset	# Nodes	# Edges	Density	# Subgraphs	# Labels
DENSITY	5,000	29,521	0.0024	250	3
CUT RATIO	5,000	83,969	0.0067	250	3
CORENESS	5,000	118,785	0.0095	221	3
COMPONENT	19,555	43,701	0.0002	250	2
PPI-BP	17,080	316,951	0.0022	1,591	6
HPO-METAB	14,587	3,238,174	0.0304	2,400	6
HPO-NEURO	14,587	3,238,174	0.0304	4,000	10
EM-USER	57,333	4,573,417	0.0028	324	2

Table D.2: Properties of **subgraphs** in synthetic and real-world datasets.

Dataset	Average # nodes	Average density	Average cut ratio	Average # components
DENSITY	20.0 $\pm$ 0.0	0.232 $\pm$ 0.146	0.0010 $\pm$ 0.0062	3.8 $\pm$ 3.7
CUT RATIO	20.0 $\pm$ 0.0	0.945 $\pm$ 0.028	0.0072 $\pm$ 0.0011	1.0 $\pm$ 0.0
CORENESS	20.0 $\pm$ 0.0	0.219 $\pm$ 0.062	0.0082 $\pm$ 0.0081	1.0 $\pm$ 0.0
COMPONENT	74.2 $\pm$ 52.8	0.150 $\pm$ 0.161	5.1 $\times 10^{-6}$ $\pm$ 3.4 $\times 10^{-6}$	4.9 $\pm$ 3.5
PPI-BP	10.2 $\pm$ 10.5	0.216 $\pm$ 0.188	0.0036 $\pm$ 0.0032	7.0 $\pm$ 5.5
HPO-METAB	14.4 $\pm$ 6.2	0.757 $\pm$ 0.149	0.1844 $\pm$ 0.0396	1.6 $\pm$ 0.7
HPO-NEURO	14.8 $\pm$ 6.5	0.767 $\pm$ 0.141	0.1834 $\pm$ 0.0386	1.5 $\pm$ 0.7
EM-USER	155.4 $\pm$ 100.2	0.010 $\pm$ 0.006	0.0053 $\pm$ 0.0006	52.1 $\pm$ 15.3

D.1.4 IMPLEMENTATION DETAILS

COMPUTING INFRASTRUCTURE We leverage Pytorch Geometric (Version 1.4.3)⁸⁸ and Pytorch Lightning (Version 0.7.1)⁸⁴ for model development. Models were trained on single GPUs from a SLURM cluster containing Tesla V100, Tesla M40, and Tesla K80 GPUs.

PRETRAINING NODE EMBEDDINGS Node embeddings were pretrained using a 2-layer GIN architecture³⁵². Hyperparameters were selected from the following ranges: batch size $\in [256, 4096]$, learning rate $\in [5e-3, 5e-5]$, weight decay $\in [5e-4, 5e-5]$, dropout rate $\in [0.4, 0.5]$, hidden layer

Table D.3: AUROC performance on synthetic datasets. Standard deviations are provided from runs with 10 random seeds.

Method	Datasets			
	DENSITY	CUT RATIO	CORENESS	COMPONENT
SUB-GNN (Ours)	0.971 ± 0.007	0.836 ± 0.021	0.824 ± 0.044	0.997 ± 0.009
Node Averaging	0.619 ± 0.026	0.542 ± 0.069	0.700 ± 0.028	0.623 ± 0.199
Meta Node (GIN)	0.602 ± 0.039	0.602 ± 0.023	0.670 ± 0.047	0.873 ± 0.026
Meta Node (GAT)	0.809 ± 0.024	0.531 ± 0.089	0.682 ± 0.068	0.884 ± 0.014
Sub2Vec Neighborhood	0.580 ± 0.028	0.533 ± 0.046	0.592 ± 0.037	0.629 ± 0.035
Sub2Vec Structure	0.553 ± 0.026	0.504 ± 0.093	0.539 ± 0.084	0.539 ± 0.104
Sub2Vec N & S Concat	0.578 ± 0.040	0.493 ± 0.051	0.562 ± 0.043	0.558 ± 0.042
Graph-level GNN	0.854 ± 0.056	0.514 ± 0.038	0.588 ± 0.052	0.514 ± 0.079

dimension $\in [128, 512]$, and output dimension $\in [32, 128]$. We used NEIGHBORSAMPLER¹²² in Pytorch Geometric⁸⁸ to perform mini-batching with number of hops $k = 1$ and neighborhood size $\in [0.1, 1.0]$. The node features for all graphs were one-hot encodings.

MODEL HYPERPARAMETER TUNING Hyperparameters were selected to optimize micro F1 scores on the validation datasets. In the following paragraph, we describe the hyperparameter ranges we explored, and we will include the best hyperparameters selected for each model with our code release.

Baseline hyperparameters were selected from the following ranges: batch size $\in [8, 128]$, learning rate $\in [1e-5, 0.1]$, weight decay $\in [5e-5, 5e-6]$, and feed forward hidden dimension sizes $\in [8, 256]$. For the MN-GAT baseline method, we use the default parameters for GAT, except for the number of heads, which we set to 4. For the s2v-N and s2v-S, methods, we used all of the default parameters in Sub2Vec⁶ to train subgraph embeddings; s2v-NS concatenates the resulting embeddings from s2v-N and s2v-S. The GC architecture and hyperparameters were adapted from the Pytorch Geometric GIN example on the MUTAG dataset (https://github.com/rustyls/pytorch_geometric/blob/master/examples/mutag_gin.py)⁸⁸.

SUB-GNN hyperparameters were selected from the following ranges: batch size $\in [16, 128]$,

Table D.4: AUROC performance on real-world datasets. Standard deviations are provided from runs with 10 random seeds.

Method	Datasets			
	PPI-BP	HPO-NEURO	HPO-METAB	EM-USER
SUB-GNN (Ours)	0.564 ± 0.013	0.862 ± 0.005	0.843 ± 0.014	0.842 ± 0.013
Node Averaging	0.502 ± 0.010	0.764 ± 0.104	0.814 ± 0.032	0.810 ± 0.096
Meta Node (GIN)	0.533 ± 0.008	0.516 ± 0.044	0.510 ± 0.036	0.575 ± 0.016
Meta Node (GAT)	0.557 ± 0.032	0.502 ± 0.012	0.581 ± 0.017	0.556 ± 0.048
Sub2Vec Neighborhood	0.497 ± 0.008	0.502 ± 0.014	0.504 ± 0.039	0.504 ± 0.055
Sub2Vec Structure	0.490 ± 0.006	0.498 ± 0.010	0.505 ± 0.016	0.810 ± 0.024
Sub2Vec N & S Concat	0.491 ± 0.006	0.504 ± 0.010	0.496 ± 0.015	0.556 ± 0.048
Graph-level GNN	0.528 ± 0.014	0.797 ± 0.008	0.730 ± 0.019	0.505 ± 0.040

Table D.5: Channel ablation analyses (AUROC). Channels that encode properties relevant to each dataset have a ✓ and best performing channels are in **bold**.

SUB-GNN Channel	Datasets			
	DENSITY	CUT RATIO	CORENESS	COMPONENT
Position (P)	0.899 ± 0.016	0.706 ± 0.043	0.712 ± 0.047 ✓	0.997 ± 0.009 ✓
Neighborhood (N)	0.904 ± 0.020	0.528 ± 0.078	0.668 ± 0.066	0.955 ± 0.035
Structure (s)	0.971 ± 0.007 ✓	0.836 ± 0.021 ✓	0.823 ± 0.025 ✓	0.834 ± 0.142
All (P+N+S)	0.968 ± 0.008	0.642 ± 0.100	0.824 ± 0.044	0.968 ± 0.032

learning rate $\in [1e-4, 1e-3]$, gradient clipping $\in [0, 0.5]$, number of layers $l \in [1, 5]$, k-hop neighborhood $\in [1, 2]$, number of internal position anchor patches $|A_{P_I}| \in [25, 75]$, $|A_{P_B}| \in [50, 200]$, $|A_{N_I}| \in [10, 25]$, $|A_{N_B}| \in [25, 75]$, $|A_S| \in [15, 45]$, number of LSTM layers $\in [1, 2]$ with dropout $\in [0.0, 0.4]$, and feed forward hidden dimension sizes $\in [32, 64]$ with dropout $\in [0.0, 0.4]$. We additionally experimented with both sum and max aggregation for the connected component initialization from node embeddings, and we tested the Pytorch Lightning auto learning rate finder.

Empirical Cumulative Distribution for Molecular Identity Recovery

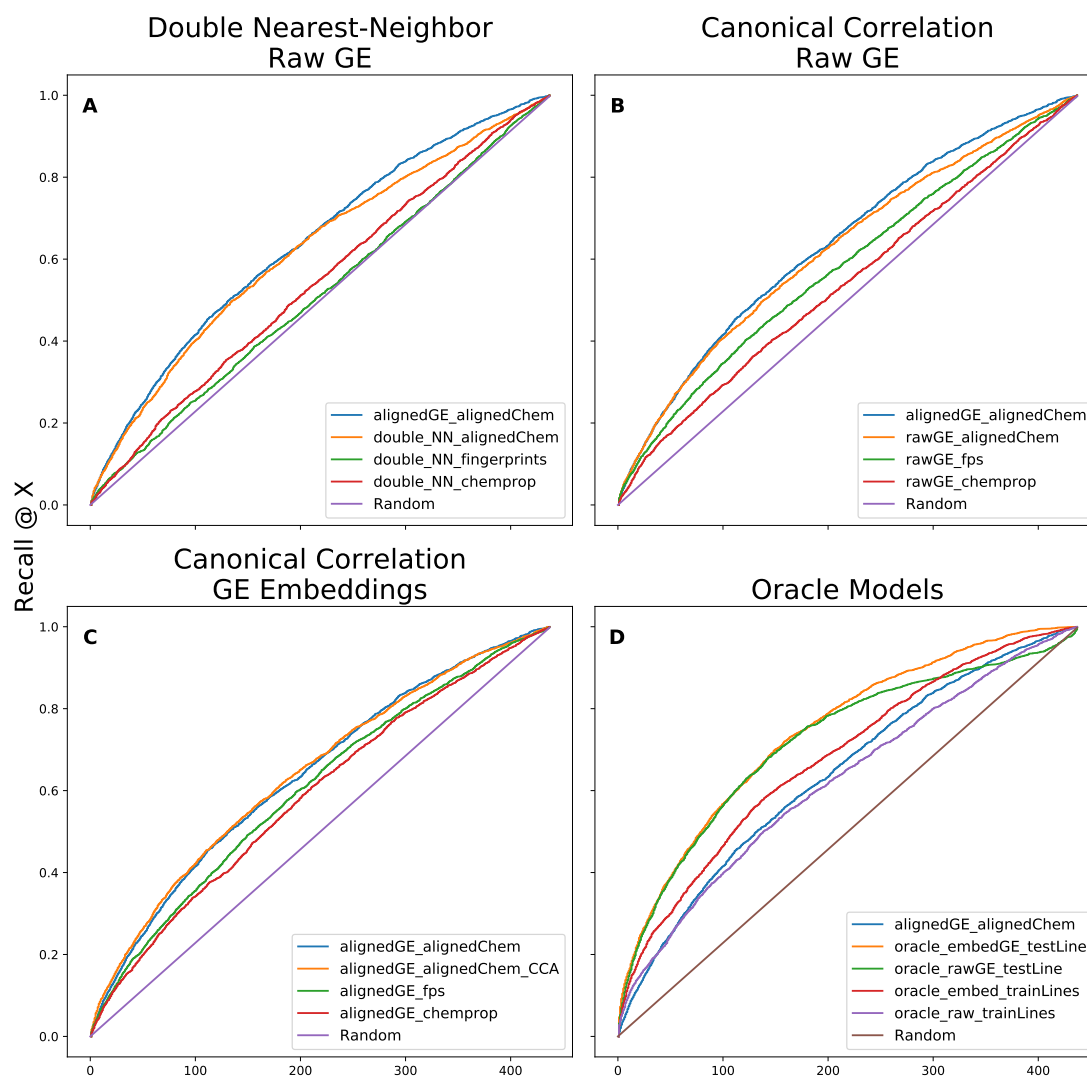


Figure D.2: Empirical CDF representing the Recall @ K for drug identification given gene expression signatures. The "alignedGE_alignedChem" curves, plotted in blue in all plots, represents our approach as described in Section 2.3.3. All other methods represent baseline or oracle methods as described in Section 2.3.2.

Ranking Performance vs Training Set Similarity

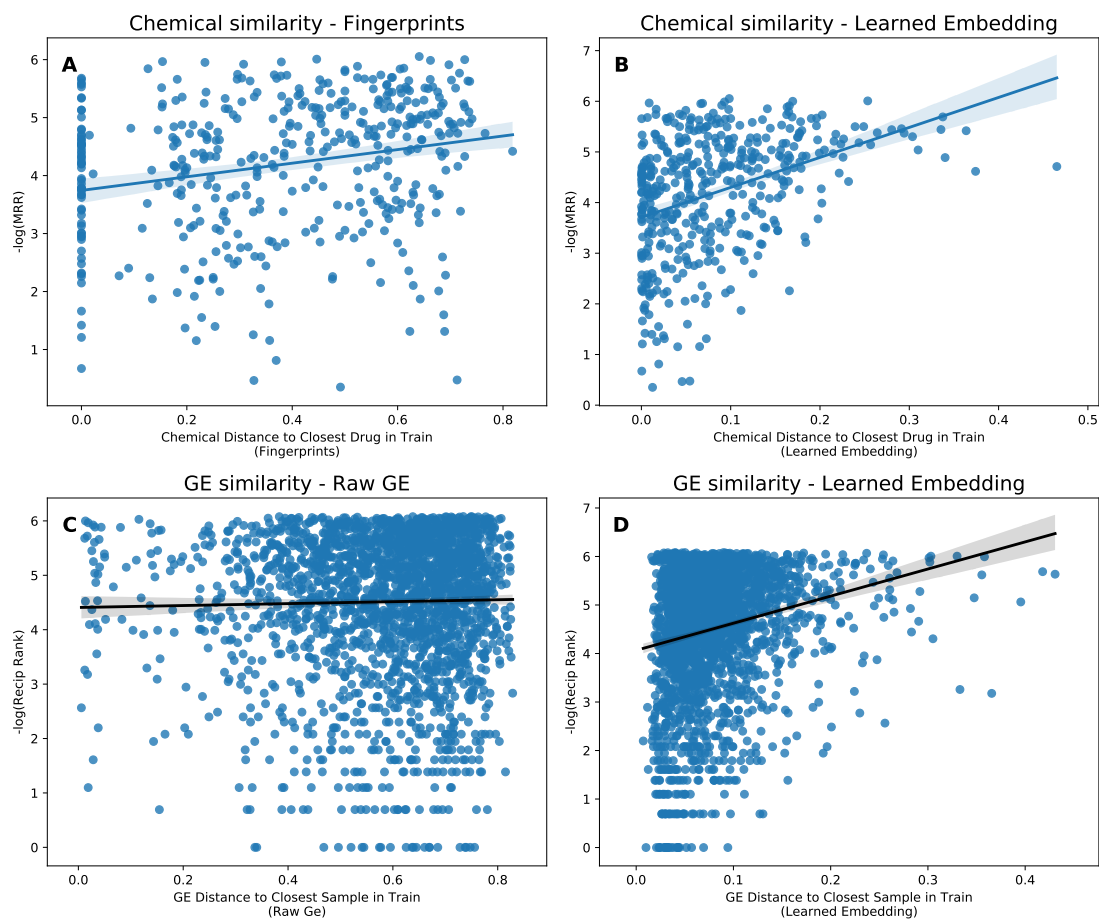


Figure D.3: Alternative measures of performance as a function of distance from the training set. Accompanies Figure 2.3.

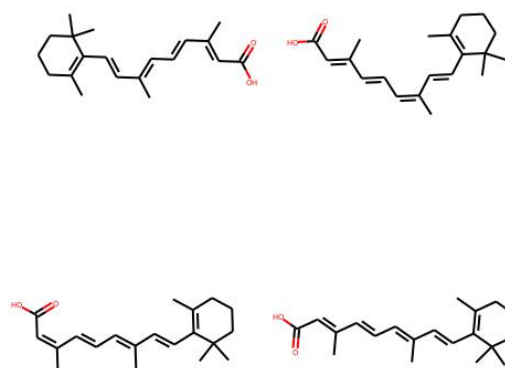


Figure D.4: Examples of four molecules that have slightly distinct structures, but have identical Morgan Circular Fingerprints.

Visualization of Gene Expression Latent Space

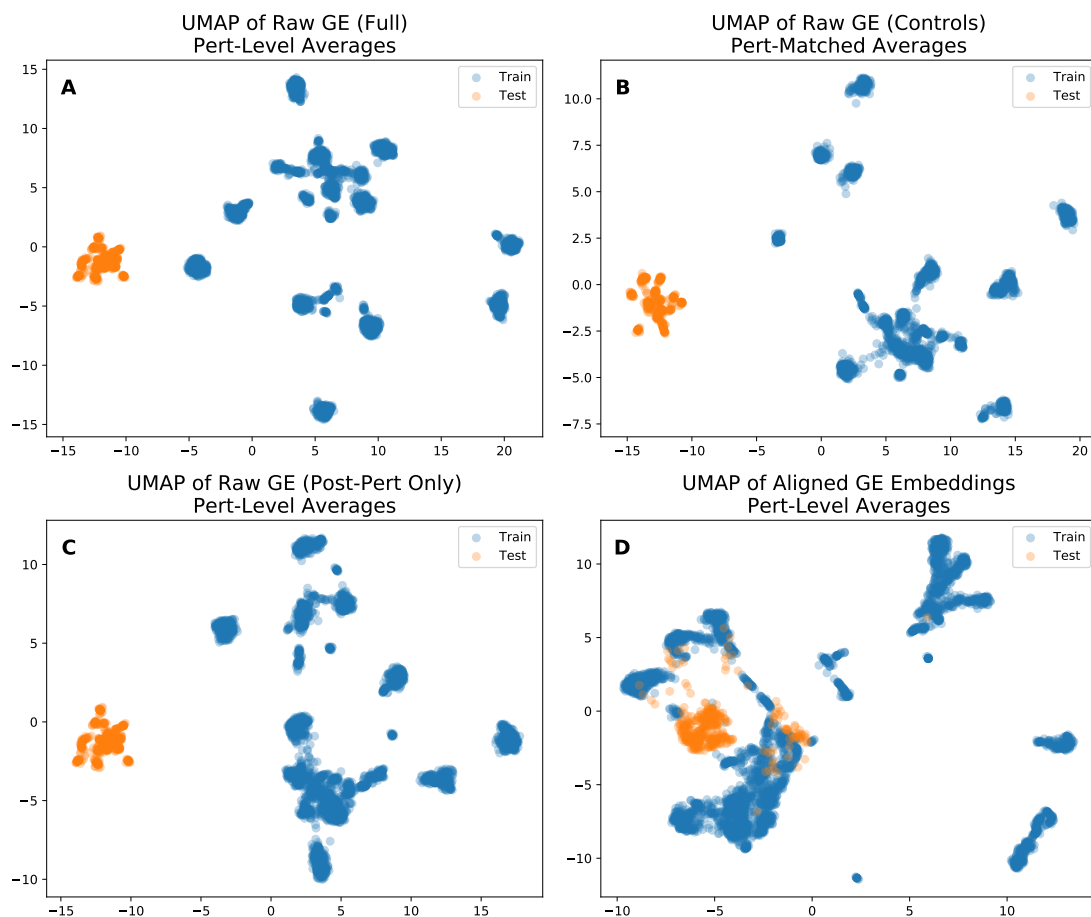


Figure D.5: UMAP latent space of gene expression signatures, colored by train vs test set. A: Full raw gene expression signatures (post-perturbational signature, plate control signature, log-fold change). B: Raw gene expression from control signatures. C: Raw gene expression values, only the post-perturbational measurements. D: Aligned embedding representations. Accompanies Figure 2.5.

Visualization of Gene Expression Latent Space

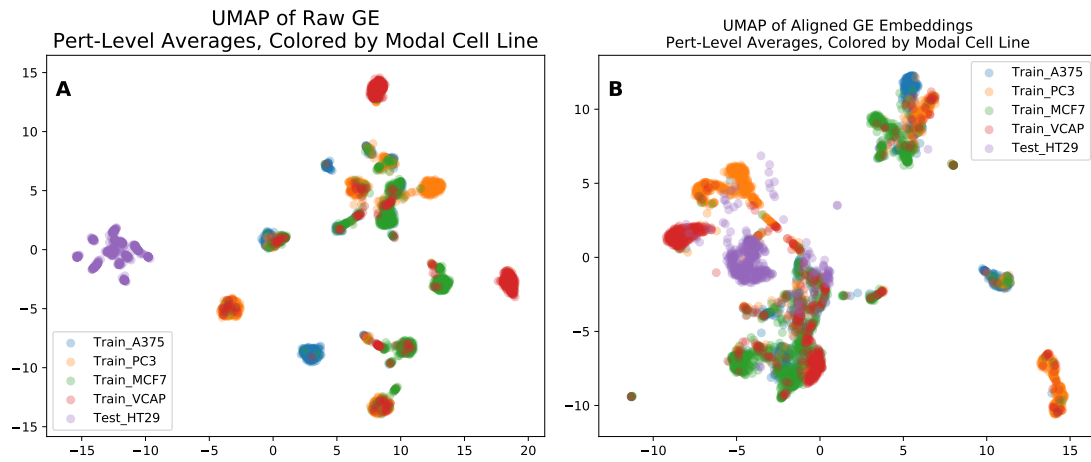


Figure D.6: UMAP latent space of gene expression signatures, colored by the cell line from which the most samples were originally measured. A: Raw gene expression signatures. B: Aligned embedding representation. Accompanies Figure 2.5.

D.2.2 CHAPTER B

	Avg	Pearson	Cosine	Kendall	Spearman	Manhattan	Canberra	KS
NPC	0.140	0.146	0.145	0.143	0.142	0.139	0.137	0.119
FIBRNPC	0.124	0.136	0.134	0.123	0.123	0.113	0.114	0.119
NEU	0.122	0.137	0.135	0.119	0.119	0.116	0.116	0.102
HEPG2	0.214	0.219	0.228	0.220	0.220	0.222	0.213	0.168
HA1E	0.217	0.249	0.246	0.209	0.209	0.207	0.203	0.202
HCC515	0.178	0.213	0.208	0.165	0.165	0.163	0.154	0.201
VCAP	0.165	0.174	0.165	0.181	0.177	0.168	0.155	0.134
HT29	0.270	0.279	0.290	0.276	0.273	0.270	0.273	0.225
A549	0.203	0.225	0.224	0.199	0.196	0.188	0.182	0.213
PC3	0.214	0.237	0.235	0.220	0.200	0.194	0.197	0.222
MCF7	0.230	0.248	0.248	0.234	0.234	0.213	0.206	0.216
Average	0.189	0.206	0.205	0.190	0.187	0.181	0.177	0.175

Table D.6: Mean reciprocal ranks for drug retrieval performance utilizing various distance metrics and target cell lines. Rows indicate distance metrics, columns the query cell lines. For all experiments, the A375 cell line was utilized as the query.

Method 1	Method 2	Statistic	P-Value	P-Adjusted	P-Adjusted Sig
Pearson	Cosine	191491	0.317	1	ns
Pearson	PCA/Pearson	361059.5	0.775	1	ns
Pearson	Kendall	401657	1.78E-06	4.98E-05	****
Pearson	Spearman	290821	1.08E-06	3.02E-05	****
Pearson	Manhattan	420846	5.60E-06	1.57E-04	***
Pearson	Canberra	447833.5	2.31E-07	6.47E-06	****
Pearson	Kolmogorov Smirnov	431586	4.21E-05	0.001	**
Cosine	PCA/Pearson	364649.5	0.538	1	ns
Cosine	Kendall	303019.5	5.32E-06	1.49E-04	***
Cosine	Spearman	301329.5	4.13E-06	1.16E-04	***
Cosine	Manhattan	311007	1.49E-05	4.17E-04	***
Cosine	Canberra	447562.5	4.67E-07	1.31E-05	****
Cosine	Kolmogorov Smirnov	333697.5	9.15E-05	0.003	**
PCA/Pearson	Spearman	313241.5	1.02E-05	2.86E-04	***
PCA/Pearson	Manhattan	436808	1.06E-05	2.97E-04	***
PCA/Pearson	Canberra	448790	2.79E-07	7.81E-06	****
PCA/Pearson	Kolmogorov Smirnov	429905.5	0.008	0.236	ns
Kendall	PCA/Pearson	421815.5	1.40E-05	3.92E-04	***
Kendall	Spearman	50758	0.505	1	ns
Kendall	Manhattan	296146	0.136	1	ns
Kendall	Canberra	398274	0.004	0.125	ns
Kendall	Kolmogorov Smirnov	387815	0.732	1	ns
Spearman	Manhattan	337436	0.14	1	ns
Spearman	Canberra	401072	0.006	0.165	ns
Spearman	Kolmogorov Smirnov	398956.5	0.74	1	ns
Manhattan	Canberra	373929	0.003	0.078	ns
Manhattan	Kolmogorov Smirnov	394614.5	0.956	1	ns
Canberra	Kolmogorov Smirnov	409290.5	0.376	1	ns

Table D.7: Results of Pairwise Wilcoxon tests comparing raw ranks from various distance metrics, matched by the target cell line.

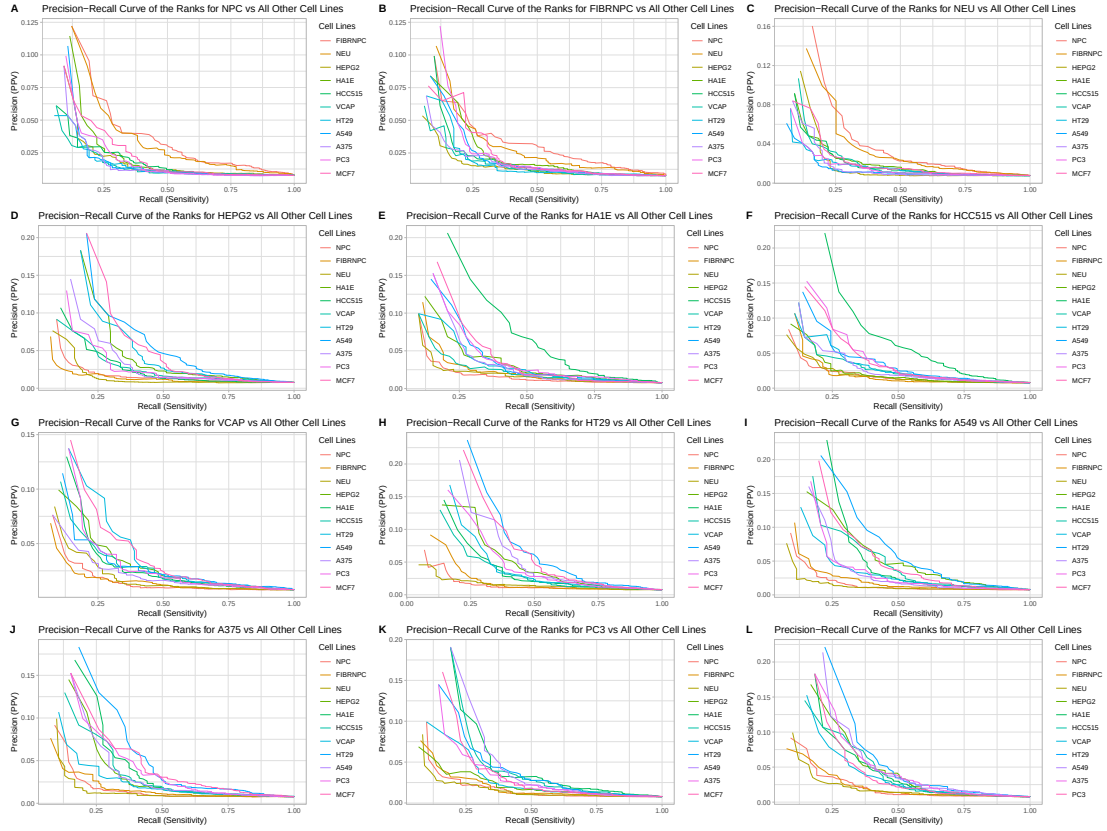


Figure D.7: Each plot represents the Precision-Recall curve, stratified by target cell line, for all possible query cell lines in our dataset.

	Estimate (95% CI)	Std. Error	T Value	P-Value
Intercept	1.0632 (0.4556, 1.671)	0.3071	3.462	7.28e-4
Baseline_Corr	-2.9372 (-4.562, -1.312)	0.8214	-3.576	4.93e-4
Baseline_Corr^2	2.3252 (1.246, 3.404)	0.5453	4.264	3.85e-5

Table D.8: Cell Line Similarity Versus Retrieval Performance: Additional Details of Model Fit.

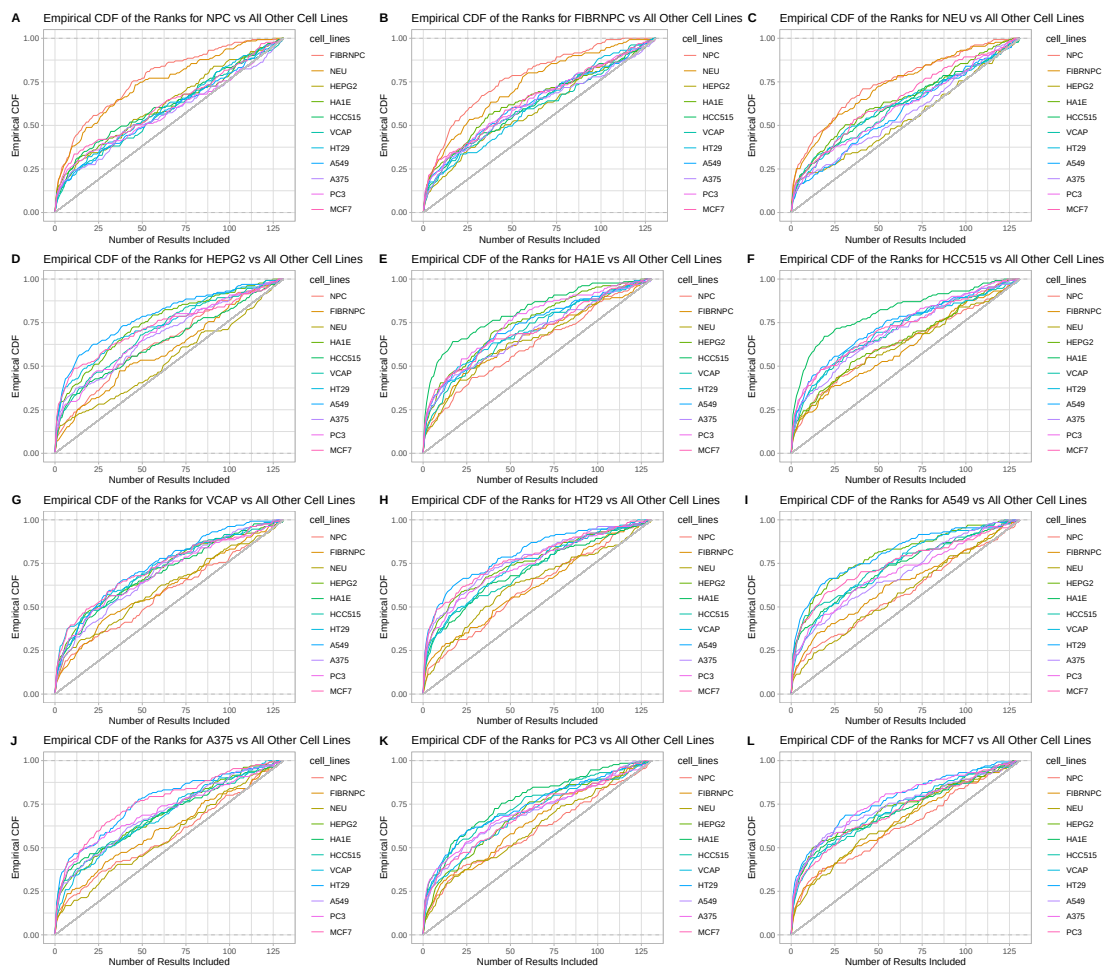


Figure D.8: Each plot represents the Empirical cumulative distribution function (ECDF) curve, stratified by target cell line, for all possible query cell lines in our dataset.

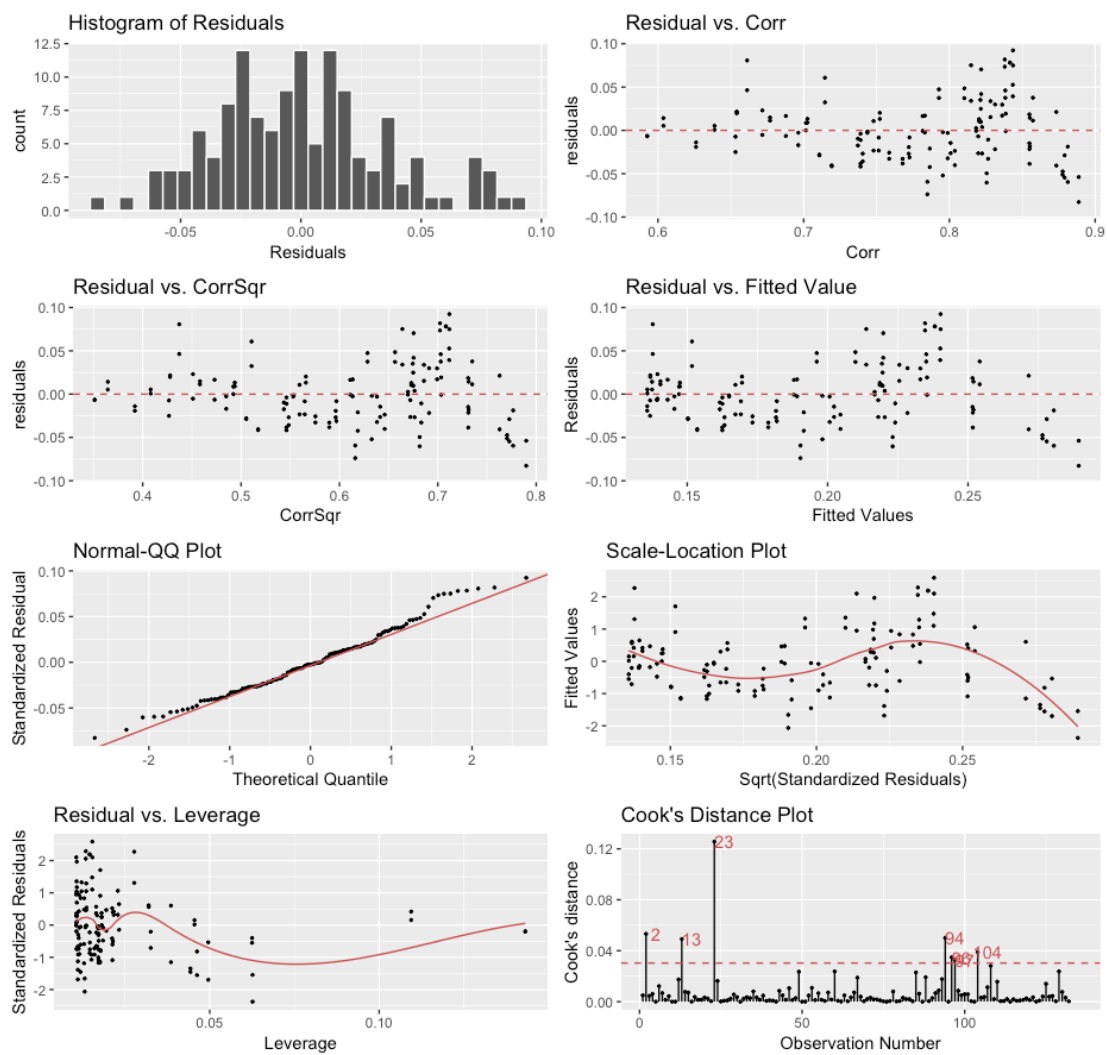


Figure D.9: Cell Line Similarity Versus Retrieval Performance: Model Diagnostics.

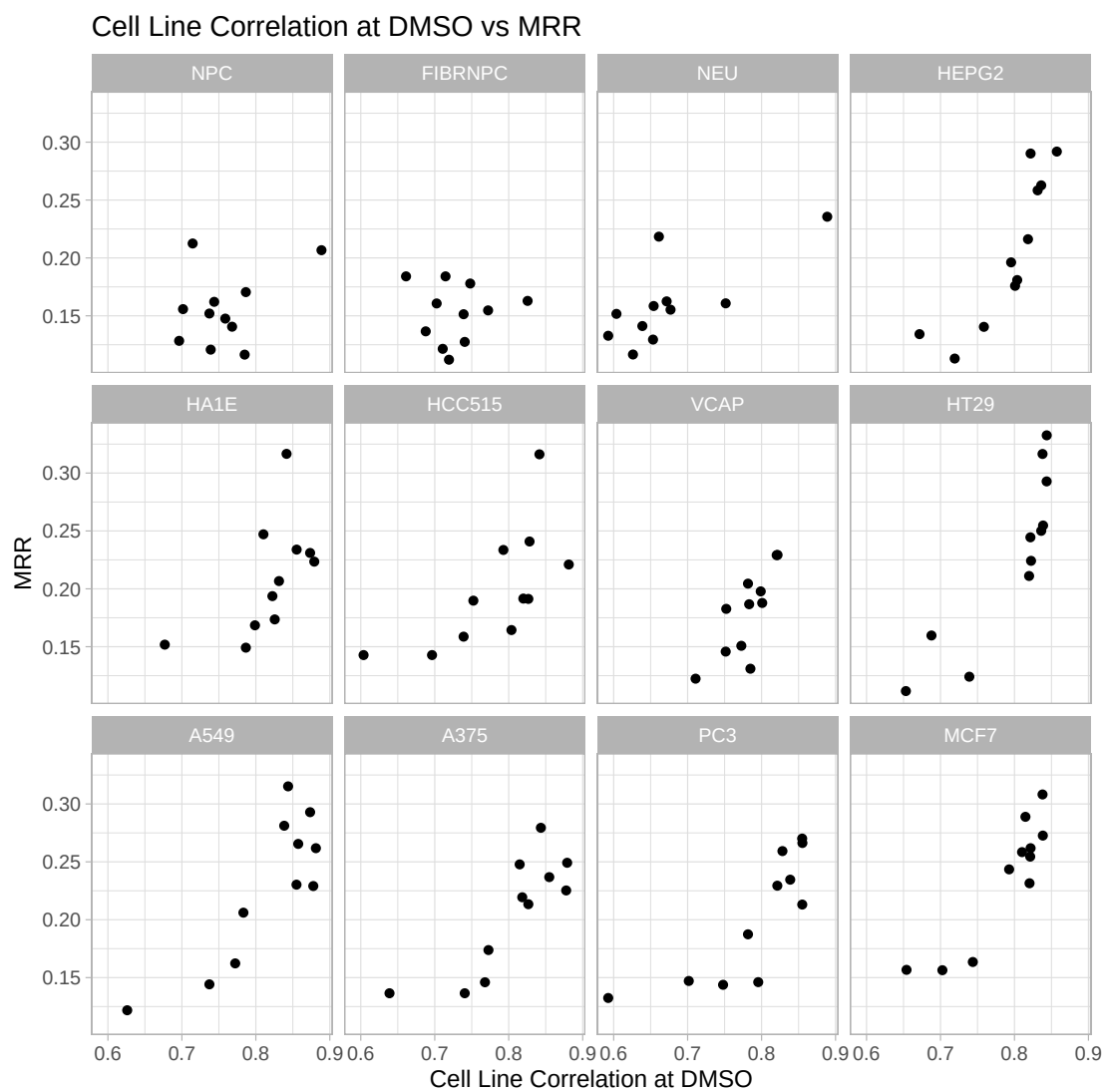


Figure D.10: Cell line similarity versus retrieval performance: cell line specific visualization. Same data as in plot B.1C, but broken down into individual cell lines.

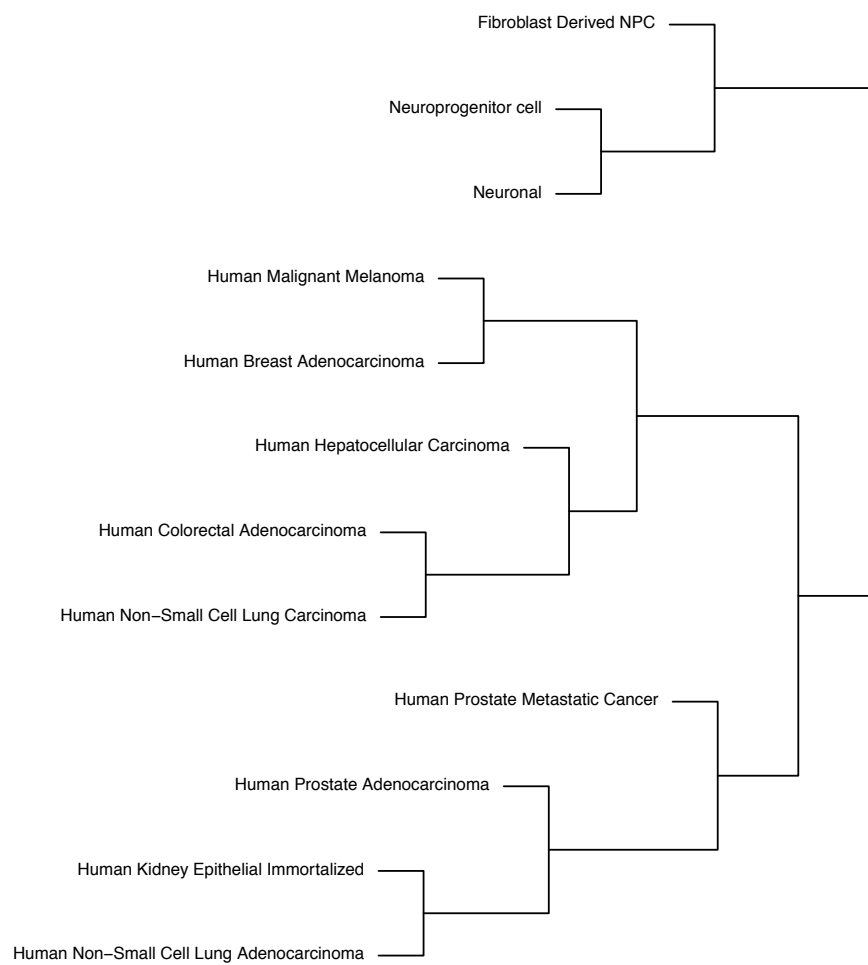


Figure D.11: Query-Target average performance reflects biological similarity. Hierarchical clustering performed on the pairwise MRR scores displayed in B.1C and D.10.

	Best Match	First Occur.	Corr Avg	Target Avg	Softmax Avg
NPC	0.207	0.211	0.195	0.241	0.269
FIBRNPC	0.163	0.193	0.283	0.266	0.272
NEU	0.236	0.237	0.236	0.250	0.262
HEPG2	0.292	0.334	0.273	0.327	0.372
HA1E	0.224	0.319	0.318	0.376	0.379
HCC515	0.221	0.302	0.326	0.343	0.366
VCAP	0.229	0.266	0.256	0.314	0.306
HT29	0.333	0.388	0.352	0.408	0.453
A549	0.262	0.322	0.352	0.389	0.396
A375	0.249	0.333	0.296	0.342	0.364
PC3	0.266	0.311	0.305	0.376	0.371
MCF7	0.273	0.345	0.313	0.372	0.400
Average	0.246	0.297	0.292	0.334	0.351

Table D.9: Retrieval performance of signature ensembling methods. All results reported as mean reciprocal ranks.

Method 1	Method 2	Statistic	P-Value	P-Adjusted	P-Adjusted Sig
Best Match	First Occur.	335663	1.40E-13	1.40E-12	****
Best Match	Corr Avg	514489	1.10E-05	0.00011	***
Best Match	Target Avg	515484	3.40E-07	3.40E-06	****
Best Match	Softmax Avg	534342	5.45E-18	5.45E-17	****
First Occur.	Corr Avg	599482.5	1.82E-36	1.82E-35	****
First Occur.	Target Avg	593130.5	1.98E-44	1.98E-43	****
First Occur.	Softmax Avg	603919	4.51E-59	4.51E-58	****
Corr Avg	Target Avg	389983	0.000844	0.008	**
Corr Avg	Softmax Avg	464904	1.31E-15	1.31E-14	****
Target Avg	Softmax Avg	359648.5	1.94E-09	1.94E-08	****

Table D.10: Results of Pairwise Wilcoxon Tests Comparing Raw Ranks from Various Ensembling Methods.

	Estimate (95% CI)	Std. Error	T Value	P-Value
Intercept	-2.104 (-2.486, -1.722)	0.1930	-10.9	<2e-16
log(NumberOfUniqueGenes)	0.5074 (0.428, 0.5866)	0.0400	12.68	<2e-16

Table D.11: Drug Specificity Versus Retrieval Performance: Additional Details of Model Fit.

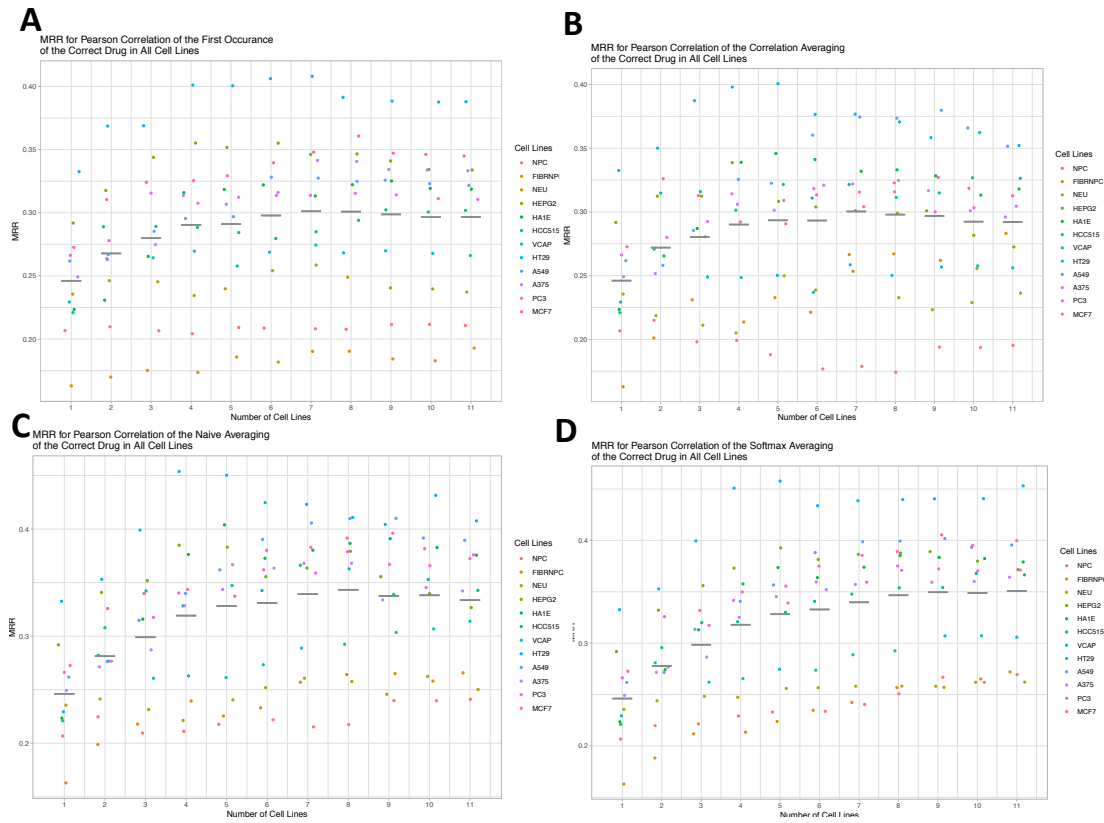


Figure D.12: Mapping Ensemble Methods Limited to Top N Most Similar Cell Lines.

	Estimate (95% CI)	Std. Error	T Value	P-Value
Intercept	107.628 (95.41, 119.84)	6.173	17.436	<2e-16
log(NumberOfUniqueGenes)	-8.603 (-11.135, -6.0708)	1.28	-6.722	5.22e-10

Table D.12: Drug Specificity Versus False Positivity: Additional Details of Model Fit.

Group 1	Group 2	Statistic	P-Value	P-Adjusted	P-Adjusted Sig
Extracellular	Intracellular	362	5.95E-20	1.78E-19	****
Extracellular	Random	464	4.94E-19	1.48E-18	****
Intracellular	Random	6165	5.52E-05	0.000166	***

Table D.13: Results of Pairwise Wilcoxon Tests Comparing Results from Intra- Versus Extracellular Genes.

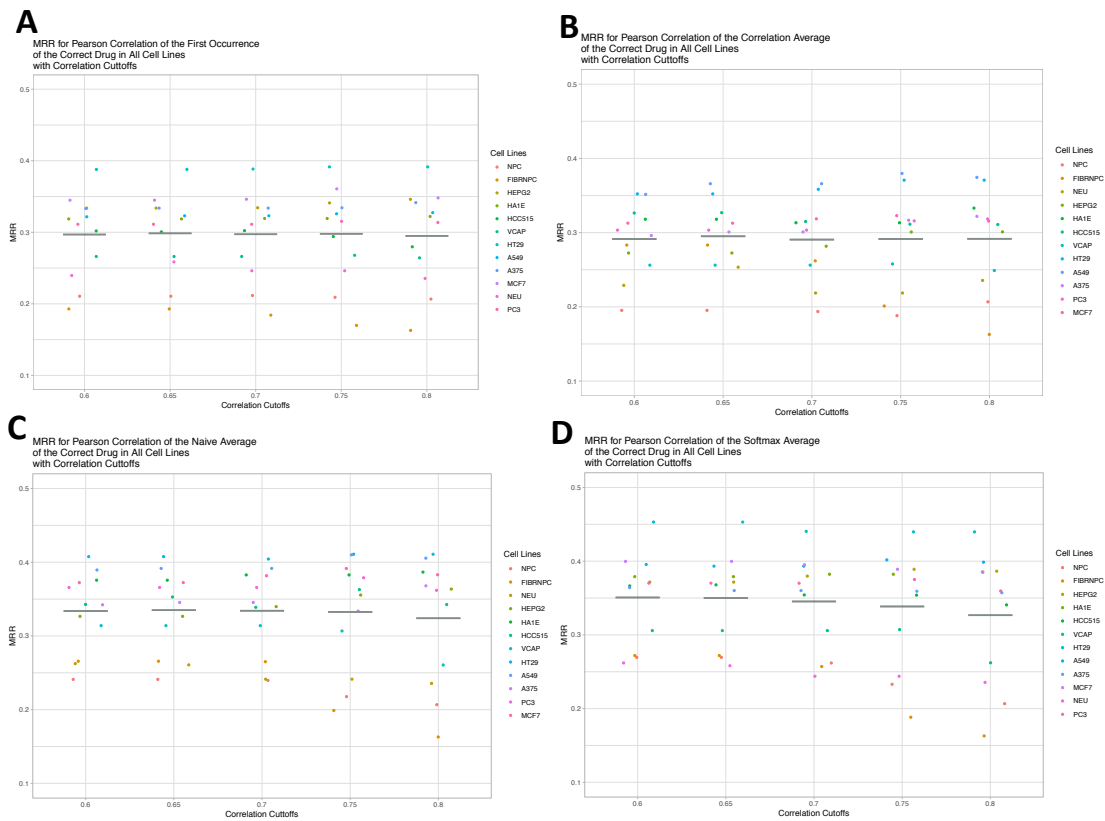


Figure D.13: Mapping Ensemble Methods with Target Signatures Thresholded by Baseline Correlation.

Datasets	MRR
CMap All Genes	0.257691597
CMap L1000 Gene Subset	0.269855226
L1000 Unions	0.261152954

Table D.14: Mapping External GEO Datasets against the LINCS and CMAP Datasets.

Datasets	MRR
L1000 6hr vs CMap	0.143053374
CMap vs L1000 6hr	0.121220802
L1000 Union vs CMap	0.127259894
CMap vs L1000 Union	0.124553181

Table D.15: Mapping the LINCS and CMAP datasets Against Each Other.

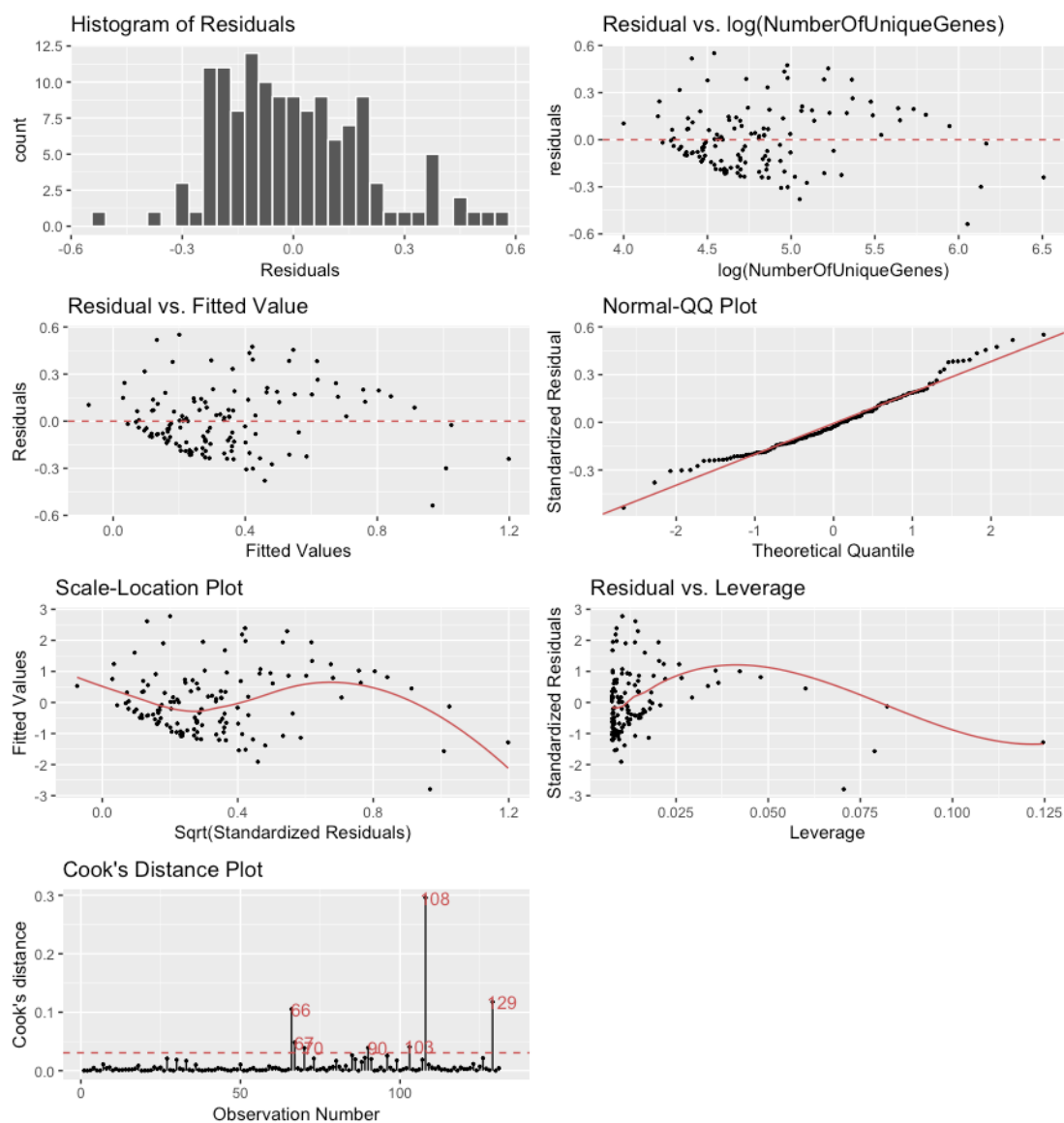


Figure D.14: Drug Specificity Versus Retrieval Performance: Model Diagnostics.

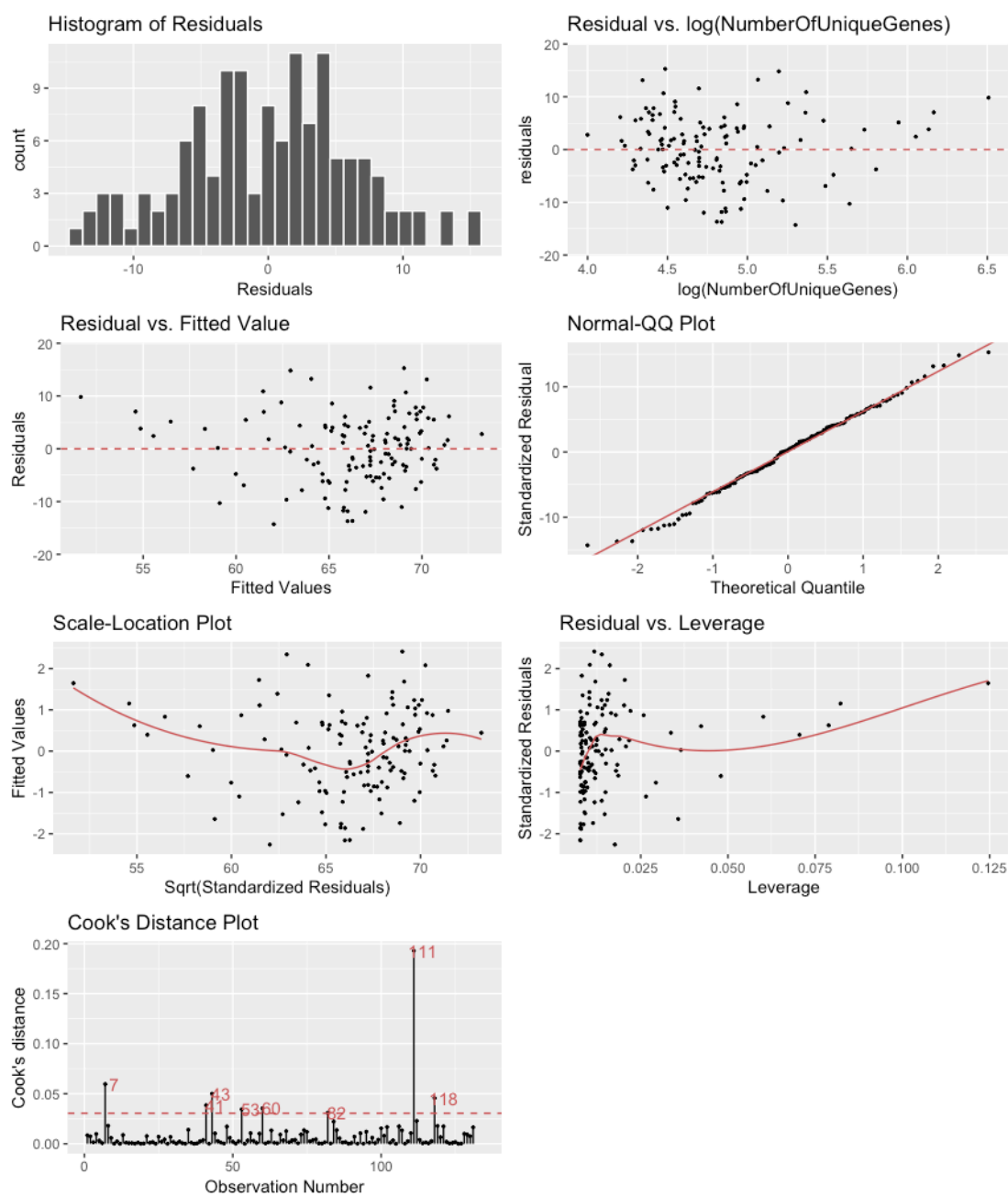


Figure D.15: Drug Specificity Versus False Positivity: Model Diagnostics.

Small Molecule	GEO Accession Numbers
Tretinoin	GSE1588, GSE20500, GSE27200, GSE31280, GSE34324, GSE37060, GSE39883, GSE42619, GSE43221, GSE43405, GSE45587, GSE45714, GSE46268, GSE49516, GSE50028, GSE51549, GSE54464, GSE56683, GSE58749, GSE60956, GSE61002, GSE63351, GSE66043, GSE66837, GSE66854, GSE69851, GSE75138.
Dexamethasone	GSE7683, GSE11056, GSE39272, GSE41062, GSE42902, GSE46116, GSE49591, GSE49804, GSE51669, GSE54608, GSE54778, GSE55877, GSE56831, GSE59080, GSE59805, GSE61879, GSE61880, GSE62635, GSE64826, GSE65240, GSE67290, GSE71615, GSE74624, GSE75682, GSE79761.
Doxorubicin	GSE26599, GSE28416, GSE32301, GSE37543, GSE39870, GSE40289, GSE42368, GSE42531, GSE43541, GSE46275, GSE46493, GSE47013, GSE50650, GSE51068, GSE51709, GSE53474, GSE58074, GSE63351, GSE63410, GSE79413.
LY-294002	GSE8488, GSE9601, GSE15069, GSE29689, GSE34817, GSE50841, GSE60344, GSE62375.
Metformin	GSE14888, GSE17404, GSE33612, GSE35961, GSE36847, GSE46262, GSE46263, GSE67342.
Progesterone	GSE10290, GSE16171, GSE22025, GSE24468, GSE25077, GSE29435, GSE38132, GSE55736, GSE59262, GSE59558, GSE61368, GSE61538, GSE62243, GSE67561, GSE69851.
Quercetin	GSE13899, GSE15162, GSE23523, GSE38067, GSE38136, GSE38141, GSE38325, GSE39140, GSE51343, GSE59368, GSE62805, GSE63227, GSE71367.
Sirolimus	GSE15703, GSE18571, GSE26599, GSE36847, GSE38055, GSE46909, GSE51068, GSE66107, GSE68091.
Resveratrol	GSE6089, GSE7111, GSE11845, GSE25412, GSE32357, GSE38325, GSE39551, GSE41168, GSE45927, GSE50005, GSE62194, GSE72119, GSE79109.
Vorinostat	GSE18544, GSE27226, GSE29419, GSE34880, GSE35242, GSE43010, GSE44621, GSE56188, GSE60408, GSE69851, GSE71127, GSE72688.

Table D.16: GEO Accession numbers for the small molecules not taken from the CREEDS database.

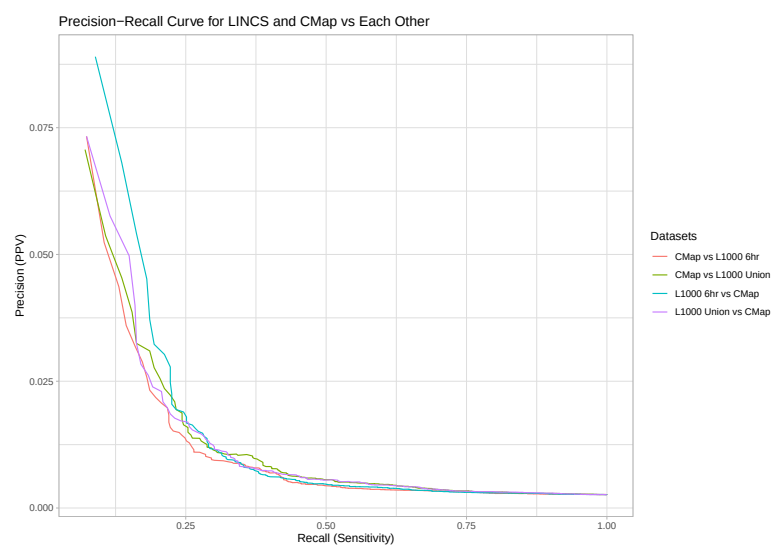
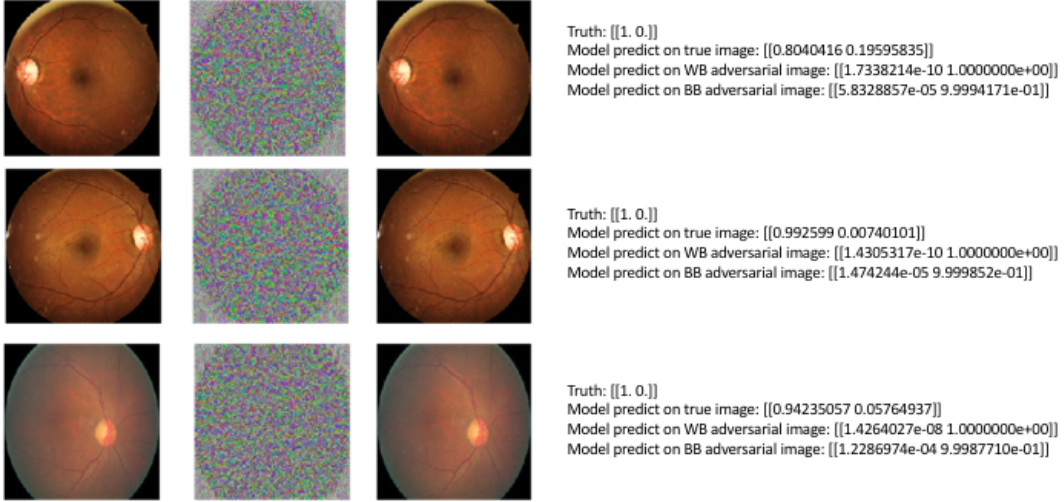


Figure D.16: Mapping the LINCS and CMAP datasets Against Each Other.

Inputs	Accuracy	AUC	Avg. Conf.
<i>Fundoscopy</i>			
Patch - Natural	78.5%	0.828	80.8%
Patch - White Box	0.3%	0.000	99.2%
Patch - Gray Box	3.9%	0.000	97.5%
<i>Chest X-Ray</i>			
Patch - Natural	92.1%	0.539	95.8%
Patch - White Box	0.00%	0.000	98.8%
Patch - Gray Box	9.7%	0.004	83.3%
<i>Dermoscopy</i>			
Patch - Natural	67.5%	0.482	85.6%
Patch - White Box	0.00%	0.000	99.7%
Patch - Gray Box	1.37%	0.000	97.6%

Table D.17: Results of medical deep learning models on clean test set data, white box, and black box patch attacks.

Diagnosis Negatives



Diagnosis Positives

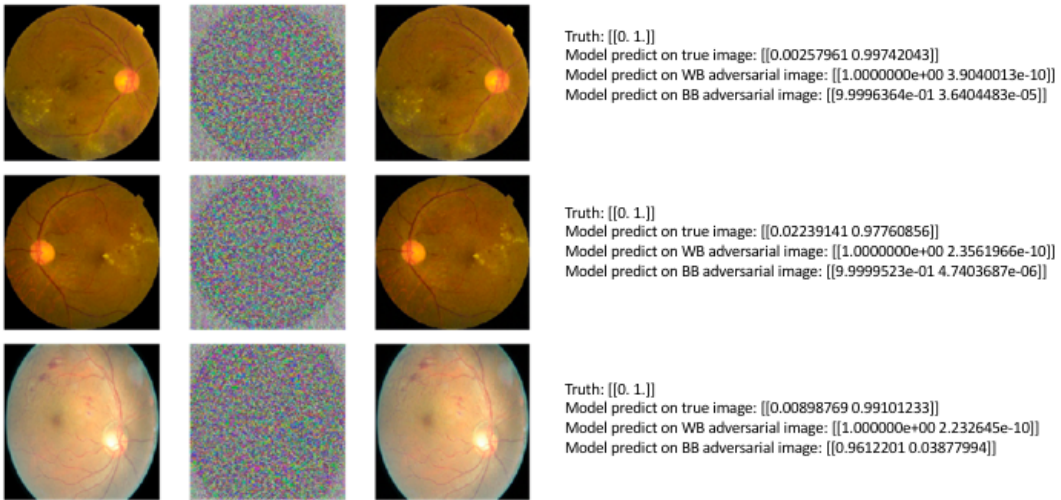
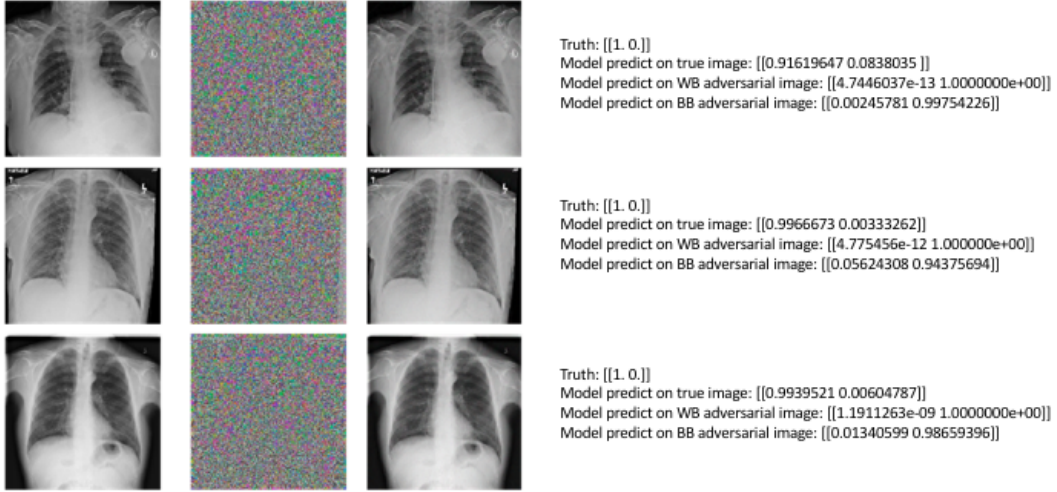


Figure D.17: Additional examples of retinal fundus images, adversarial perturbations, and resultant adversarial examples. Noise displayed is the white box noise.

Diagnosis Negatives



Diagnosis Positives

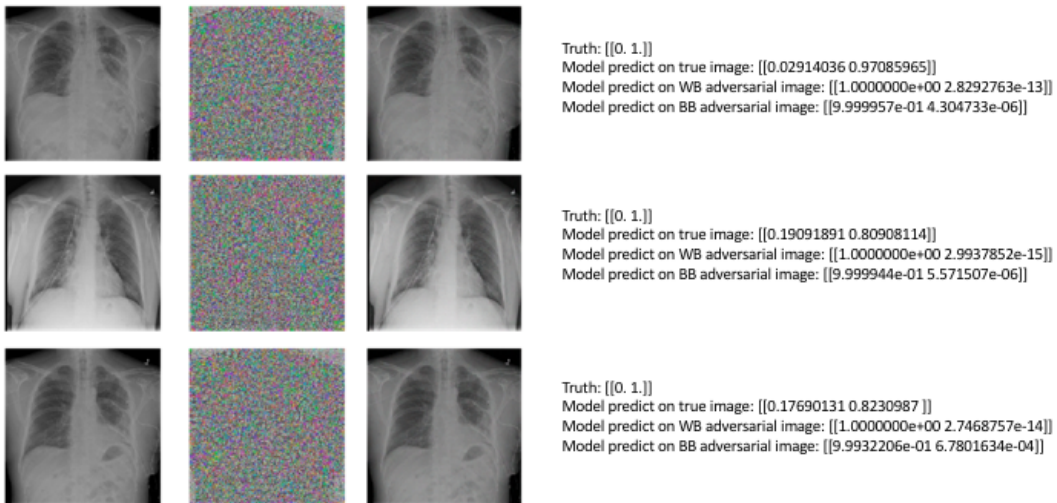
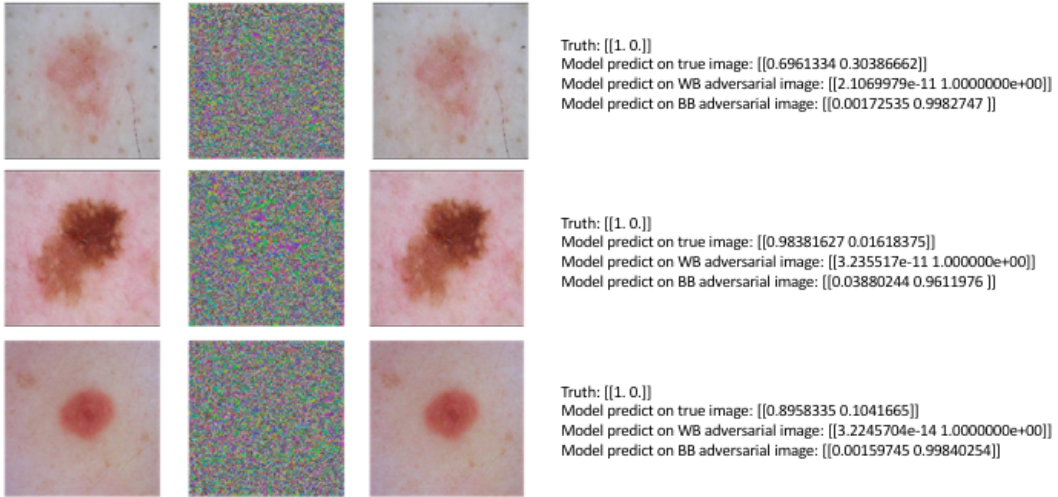


Figure D.18: Additional examples of chest x-ray images, adversarial perturbations, and resultant adversarial examples. Noise displayed is the white box noise.

Diagnosis Negatives



Diagnosis Positives

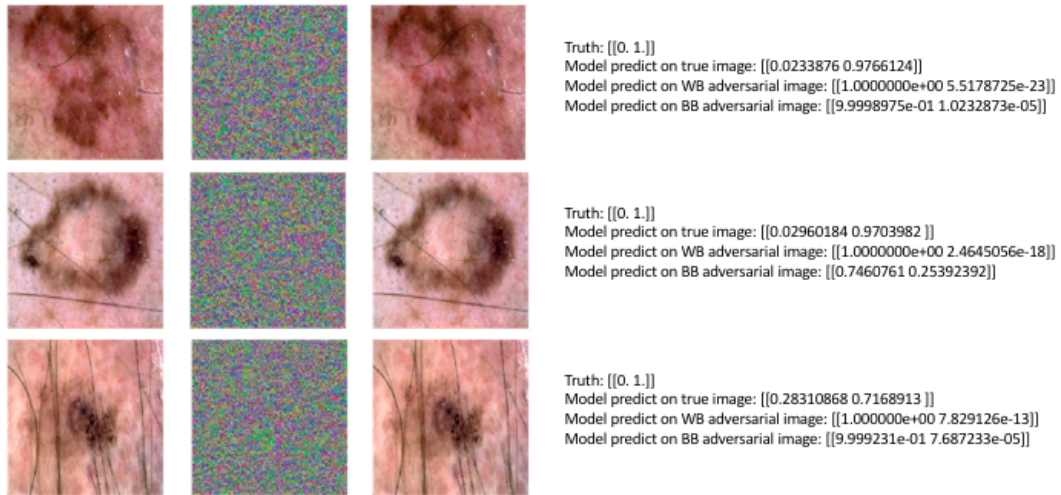


Figure D.19: Additional examples of dermoscopy images, adversarial perturbations, and resultant adversarial examples. Noise displayed is the white box noise.

References

- [isi] International skin imaging collaboration.
- [kag] Kaggle diabetic retinopathy challenge.
- [3] Abu-El-Haija, S., Perozzi, B., Kapoor, A., Harutyunyan, H., Alipourfard, N., Lerman, K., Steeg, G. V., & Galstyan, A. (2019). Mixhop: Higher-order graph convolution architectures via sparsified neighborhood mixing. *ICML*.
- [4] Act, T. O. D. (1983). Public law 97-414. In *97th Congress. January*, volume 4 (pp. 2049–2066).
- [5] Adamson, A. S. & Smith, A. (2018). Machine learning and health care disparities in dermatology. *JAMA dermatology*, 154(11), 1247–1248.
- [6] Adhikari, B., Zhang, Y., Ramakrishnan, N., & Prakash, B. A. (2018). Sub2vec: Feature learning for subgraphs. In *PAKDD*: Springer.
- [7] Alipanahi, B., Delong, A., Weirauch, M. T., & Frey, B. J. (2015). Predicting the sequence specificities of dna-and rna-binding proteins by deep learning. *Nature biotechnology*, 33(8), 831–838.
- [8] Andrew, G., Arora, R., Bilmes, J., & Livescu, K. (2013). Deep canonical correlation analysis. In *International conference on machine learning* (pp. 1247–1255).
- [9] Angyal, M. & Derczy, K. (1998). Personal identification on the basis of antemortem and postmortem radiographs. *Journal of Forensic Science*, 43(5), 1089–1093.
- [10] Annervaz, K., Chowdhury, S. B. R., & Dukkupati, A. (2018). Learning beyond datasets: Knowledge graph augmented neural networks for natural language processing. *arXiv preprint arXiv:1802.05930*.
- [11] Antoniou, A., Storkey, A., & Edwards, H. (2017). Data augmentation generative adversarial networks. *arXiv preprint arXiv:1711.04340*.

- [12] Ashburner, M., Ball, C. A., Blake, J. A., Botstein, D., Butler, H., Cherry, J. M., Davis, A. P., Dolinski, K., Dwight, S. S., Eppig, J. T., et al. (2000). Gene ontology: tool for the unification of biology. *Nature genetics*, 25(1), 25–29.
- [13] Athalye, A., Carlini, N., & Wagner, D. (2018). Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. *arXiv preprint arXiv:1802.00420*.
- [14] Athalye, A., Engstrom, L., Ilyas, A., & Kwok, K. (2017). Synthesizing robust adversarial examples. [abs/1707.07397](https://arxiv.org/abs/1707.07397).
- [15] Athalye, A. & Sutskever, I. (2017). Synthesizing robust adversarial examples. *arXiv preprint arXiv:1707.07397*.
- [16] Babic, B., Gerke, S., Evgeniou, T., & Cohen, I. G. (2019). Algorithms on regulatory lock-down in medicine. *Science*, 366(6470), 1202–1204.
- [17] Badgeley, M. A., Zech, J. R., Oakden-Rayner, L., Glicksberg, B. S., Liu, M., Gale, W., McConnell, M. V., Percha, B., Snyder, T. M., & Dudley, J. T. (2019). Deep learning predicts hip fracture using confounding patient and healthcare variables. *NPJ digital medicine*, 2(1), 1–10.
- [18] Bai, J., Ren, Y., & Zhang, J. (2020). Ripple walk training: A subgraph-based training framework for large and deep graph neural network. *arXiv:2002.07206*.
- [19] Bai, Y., Ding, H., Qiao, Y., Marinovic, A., Gu, K., Chen, T., Sun, Y., & Wang, W. (2019). Unsupervised inductive graph-level representation learning via graph-graph proximity. In *IJCAI*.
- [20] Baker, M. (2016). 1,500 scientists lift the lid on reproducibility. *Nature*, 533(7604), 452–454.
- [21] Battaglia, P. W., Hamrick, J. B., Bapst, V., Sanchez-Gonzalez, A., Zambaldi, V., Malinowski, M., Tacchetti, A., Raposo, D., Santoro, A., Faulkner, R., et al. (2018). Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*.
- [22] Beers, A., Brown, J., Chang, K., Campbell, J. P., Ostmo, S., Chiang, M. F., & Kalpathy-Cramer, J. (2018). High-resolution medical image synthesis using progressively grown generative adversarial networks. *arXiv preprint arXiv:1805.03144*.
- [23] Bejnordi, B. E., Veta, M., van Diest, P. J., van Ginneken, B., Karssemeijer, N., Litjens, G., van der Laak, J. A., Hermesen, M., Manson, Q. F., Balkenhol, M., et al. (2017). Diagnostic assessment of deep learning algorithms for detection of lymph node metastases in women with breast cancer. *Jama*, 318(22), 2199–2210.

- [24] Bender, D. & Sartipi, K. (2013). HL7 fhir: An agile and restful approach to healthcare information exchange. In *Proceedings of the 26th IEEE international symposium on computer-based medical systems* (pp. 326–331).: IEEE.
- [25] Bengio, Y., Bengio, S., & Cloutier, J. (1990). *Learning a synaptic learning rule*. Citeseer.
- [26] Bengio, Y., Courville, A., & Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8), 1798–1828.
- [27] Bengio, Y., Delalleau, O., & Roux, N. L. (2006). The curse of highly variable functions for local kernel machines. In *Advances in neural information processing systems* (pp. 107–114).
- [28] Bengio, Y., Delalleau, O., & Simard, C. (2010). Decision trees do not generalize to new variations. *Computational Intelligence*, 26(4), 449–467.
- [29] Bengio, Y. & Monperrus, M. (2005). Non-local manifold tangent learning. In *Advances in Neural Information Processing Systems* (pp. 129–136).
- [30] Benson, A. R., Gleich, D. F., & Leskovec, J. (2016). Higher-order organization of complex networks. *Science*, 353(6295).
- [31] Bergstra, J., Yamins, D., & Cox, D. D. (2013). Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. In *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28, ICML'13* (pp. 1–115–1–123).: JMLR.org.
- [32] Biggio, B. & Roli, F. (2017). Wild patterns: Ten years after the rise of adversarial machine learning. *arXiv preprint arXiv:1712.03141*.
- [33] Boag, W., Suresh, H., Celi, L. A., Szolovits, P., & Ghassemi, M. (2018). Racial disparities and mistrust in end-of-life care. *arXiv preprint arXiv:1808.03827*.
- [34] Bodenreider, O. (2004). The unified medical language system (umls): integrating biomedical terminology. *Nucleic acids research*, 32(suppl_1), D267–D270.
- [35] Bordes, A., Chopra, S., & Weston, J. (2014). Question answering with subgraph embeddings. *arXiv:1406.3676*.
- [36] Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., & Yakhnenko, O. (2013). Translating embeddings for modeling multi-relational data. In *Advances in neural information processing systems* (pp. 2787–2795).
- [37] Bourgeois, F. C., Fossa, A., Gerard, M., Davis, M. E., Taylor, Y. J., Connor, C. D., Vaden, T., McWilliams, A., Spencer, M. D., Folcarelli, P., et al. (2019). A patient and family reporting

- system for perceived ambulatory note mistakes: experience at 3 us healthcare centers. *Journal of the American Medical Informatics Association*, 26(12), 1566–1573.
- [38] Boyd, J. & Raychowdhury, P. (1989). Discrete dirichlet problems, convex coordinates, and a random walk on a triangle. *The College Mathematics Journal*, 20(5).
 - [39] Brouwer, C. L., Steenbakkers, R. J., van den Heuvel, E., Duppen, J. C., Navran, A., Bijl, H. P., Chouvalova, O., Burlage, F. R., Meertens, H., Langendijk, J. A., et al. (2012). 3d variation in delineation of head and neck organs at risk. *Radiation Oncology*, 7(1), 32.
 - [40] Brown, T. B., Mané, D., Roy, A., Abadi, M., & Gilmer, J. (2017). Adversarial patch. *arXiv preprint arXiv:1712.09665*.
 - [41] Buckman, J., Roy, A., Raffel, C., & Goodfellow, I. (2018). Thermometer encoding: One hot way to resist adversarial examples. In *International Conference on Learning Representations*.
 - [42] Campbell, D. T. & Stanley, J. C. (1966). *Experimental and quasi-experimental designs for research*. Rand McNally.
 - [43] Carlini, N., Mishra, P., Vaidya, T., Zhang, Y., Sherr, M., Shields, C., Wagner, D., & Zhou, W. (2016). Hidden voice commands. In *USENIX Security Symposium* (pp. 513–530).
 - [44] Carlini, N. & Wagner, D. (2017). Towards evaluating the robustness of neural networks. In *Security and Privacy (SP), 2017 IEEE Symposium on* (pp. 39–57).: IEEE.
 - [45] Castro, D. C., Walker, I., & Glocker, B. (2019). Causality matters in medical imaging. *arXiv preprint arXiv:1912.08142*.
 - [46] Chami, I., Abu-El-Haija, S., Perozzi, B., Ré, C., & Murphy, K. (2020). Machine learning on graphs: A model and comprehensive taxonomy.
 - [47] Chang, D. K., Grimmond, S. M., Evans, T. J., & Biankin, A. V. (2014). Mining the genomes of exceptional responders. *Nature Reviews Cancer*, 14(5), 291–292.
 - [48] Chen, G. & Shi, T. (2019). Single-cell rna-seq technologies and related computational data analysis. *Frontiers in genetics*, 10, 317.
 - [49] Chen, I., Johansson, F. D., & Sontag, D. (2018). Why is my classifier discriminatory? In *Advances in Neural Information Processing Systems* (pp. 3539–3550).
 - [50] Chen, Q. & Koltun, V. (2017). Photographic image synthesis with cascaded refinement networks. In *IEEE International Conference on Computer Vision (ICCV)*, volume 1 (pp.3).
 - [51] Chen, T., Kornblith, S., Norouzi, M., & Hinton, G. (2020). A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*.

- [52] Cheng, J., Dong, L., & Lapata, M. (2016). Long short-term memory-networks for machine reading. *arXiv preprint arXiv:1601.06733*.
- [53] Cheng, J., Xie, Q., Kumar, V., Hurle, M., Freudenberg, J. M., Yang, L., & Agarwal, P. (2013). Evaluation of analytical methods for connectivity map data. In *Biocomputing 2013* (pp. 5–16). World Scientific.
- [54] Chiang, W.-L., Liu, X., Si, S., Li, Y., Bengio, S., & Hsieh, C.-J. (2019). Cluster-GCN: An efficient algorithm for training deep and large graph convolutional networks. In *KDD*.
- [55] Ching, T., Himmelstein, D. S., Beaulieu-Jones, B. K., Kalinin, A. A., Do, B. T., Way, G. P., Ferrero, E., Agapow, P.-M., Zietz, M., Hoffman, M. M., et al. (2018). Opportunities and obstacles for deep learning in biology and medicine. *bioRxiv*, (pp. 142760).
- [56] Clark, N. R., Hu, K. S., Feldmann, A. S., Kou, Y., Chen, E. Y., Duan, Q., & Ma’ayan, A. (2014). The characteristic direction: a geometrical approach to identify differentially expressed genes. *BMC bioinformatics*, 15(1), 79.
- [57] Cohen, J. (2019). At over \$2 million zolgensma is the world’s most expensive therapy, yet relatively cost-effective. *Forbes*.
- [58] Consortium, G. O. (2004). The gene ontology (go) database and informatics resource. *Nucleic acids research*, 32(suppl_1), D258–D261.
- [59] Consortium, G. O. (2019). The gene ontology resource: 20 years and still going strong. *Nucleic Acids Research*, 47(D1).
- [60] Cortes, C. & Vapnik, V. (1995). Support-vector networks. *Machine learning*, 20(3), 273–297.
- [61] Costa, F. & De Grave, K. (2010). Fast neighborhood subgraph pairwise distance kernel. In *ICML*.
- [62] Coustasse, A., Frame, M., & Mukherjee, A. (2018). Is upcoding anesthesia time the tip of the iceberg in insurance fraud? *JAMA network open*, 1(7), e184302–e184302.
- [63] Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4), 303–314.
- [64] Dahl, G., Ranzato, M., Mohamed, A.-r., & Hinton, G. E. (2010). Phone recognition with the mean-covariance restricted boltzmann machine. In *Advances in neural information processing systems* (pp. 469–477).
- [65] Dalvi, N., Domingos, P., Sanghai, S., Verma, D., et al. (2004). Adversarial classification. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 99–108).: ACM.

- [66] Dar, S. U. H., Yurt, M., Karacan, L., Erdem, A., Erdem, E., & Çukur, T. (2018). Image synthesis in multi-contrast mri with conditional generative adversarial networks. *arXiv preprint arXiv:1802.01221*.
- [67] Dawkins, H. J., Draghia-Akli, R., Lasko, P., Lau, L. P., Jonker, A. H., Cutillo, C. M., Rath, A., Boycott, K. M., Baynam, G., Lochmüller, H., et al. (2018). Progress in rare diseases research 2010–2016: an irdirc perspective. *Clinical and translational science*, 11(1), 11.
- [68] Deng, C., Chen, Z., Liu, X., Gao, X., & Tao, D. (2018). Triplet-based deep hashing network for cross-modal retrieval. *IEEE Transactions on Image Processing*, 27(8), 3893–3903.
- [69] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition* (pp. 248–255).: Ieee.
- [70] Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
- [71] Dincer, A. B., Celik, S., Hiranuma, N., & Lee, S.-I. (2018). Deepprofile: Deep learning of cancer molecular profiles for precision medicine. *bioRxiv*, (pp. 278739).
- [72] Donnat, C., Zitnik, M., Hallac, D., & Leskovec, J. (2018). Learning structural node embeddings via diffusion wavelets. In *KDD*.
- [73] Donner, Y., Kazmierczak, S., & Fortney, K. (2018). Drug repurposing using deep embeddings of gene expression profiles. *Molecular pharmaceutics*, 15(10), 4314–4325.
- [74] Duan, Q., Flynn, C., Niepel, M., Hafner, M., Muhlich, J. L., Fernandez, N. F., Rouillard, A. D., Tan, C. M., Chen, E. Y., Golub, T. R., et al. (2014). Lincs canvas browser: interactive web app to query, browse and interrogate lincs 11000 gene expression signatures. *Nucleic acids research*, 42(W1), W449–W460.
- [75] Duan, Q., Reid, S. P., Clark, N. R., Wang, Z., Fernandez, N. F., Rouillard, A. D., Readhead, B., Tritsch, S. R., Hodos, R., Hafner, M., Niepel, M., Sorger, P. K., Dudley, J. T., Bavari, S., Panchal, R. G., & Ma’ayan, A. (2016). L100ocds2: LINCS 11000 characteristic direction signatures search engine. *npj Systems Biology and Applications*, 2(1).
- [76] Dvijotham, K., Stanforth, R., Goyal, S., Mann, T., & Kohli, P. (2018). A dual approach to scalable verification of deep networks. *arXiv preprint arXiv:1803.06567*.
- [77] Edwards, H. & Storkey, A. (2016). Towards a neural statistician. *arXiv preprint arXiv:1606.02185*.
- [78] Elsayed, G. F., Shankar, S., Cheung, B., Papernot, N., Kurakin, A., Goodfellow, I., & Sohl-Dickstein, J. (2018). Adversarial examples that fool both human and computer vision. *arXiv preprint arXiv:1802.08195*.

- [79] Empiricus, S. & Bury, R. (1933). Outlines of pyrrhonism. *Eng. trans, by RG Bury (Cambridge Mass.: Harvard UP, 1976), II*, 20, 90–1.
- [80] Engstrom, L., Ilyas, A., Santurkar, S., Tsipras, D., Tran, B., & Madry, A. (2019). Adversarial robustness as a prior for learned representations.
- [81] Esteva, A., Kuprel, B., Novoa, R. A., Ko, J., Swetter, S. M., Blau, H. M., & Thrun, S. (2017). Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, 542(7639), 115.
- [82] Evtimov, I., Eykholt, K., Fernandes, E., Kohno, T., Li, B., Prakash, A., Rahmati, A., & Song, D. (2017). Robust physical-world attacks on machine learning models. *arXiv preprint arXiv:1707.08945*.
- [83] Faber, F. A., Hutchison, L., Huang, B., Gilmer, J., Schoenholz, S. S., Dahl, G. E., Vinyals, O., Kearnes, S., Riley, P. F., & Von Lilienfeld, O. A. (2017). Prediction errors of molecular machine learning models lower than hybrid dft error. *Journal of chemical theory and computation*, 13(11), 5255–5264.
- [84] Falcon, W. (2019). Pytorch lightning. *GitHub. Note: <https://github.com/williamFalcon/pytorch-lightning>* Cited by, 3.
- [85] FDA, U. (2019). Proposed regulatory framework for modifications to artificial intelligence/machine learning (ai/ml)-based software as a medical device (samd).
- [86] Fei-Fei, L., Fergus, R., & Perona, P. (2006). One-shot learning of object categories. *IEEE transactions on pattern analysis and machine intelligence*, 28(4), 594–611.
- [87] Feng, F., He, X., Liu, Y., Nie, L., & Chua, T.-S. (2018). Learning on partial-order hypergraphs. In *WWW*.
- [88] Fey, M. & Lenssen, J. E. (2019). Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*.
- [89] Filzen, T. M., Kutchukian, P. S., Hermes, J. D., Li, J., & Tudor, M. (2017). Representing high throughput expression profiles via perturbation barcodes reveals compound targets. *PLoS computational biology*, 13(2), e1005335.
- [90] Fink, M. (2005). Object classification from a single example utilizing class relevance metrics. In *Advances in neural information processing systems* (pp. 449–456).
- [91] Finlayson, S. G., Bowers, J. D., Ito, J., Zittrain, J. L., Beam, A. L., & Kohane, I. S. (2019). Adversarial attacks on medical machine learning. *Science*, 363(6433), 1287–1289.
- [92] Finlayson, S. G., Chung, H. W., Kohane, I. S., & Beam, A. L. (2018a). Adversarial attacks against medical deep learning systems. *arXiv preprint arXiv:1804.05296*.

- [93] Finlayson, S. G., Lee, H., Kohane, I. S., & Oakden-Rayner, L. (2018b). Towards generative adversarial networks as a new paradigm for radiology education.
- [94] Finn, C., Abbeel, P., & Levine, S. (2017). Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70* (pp. 1126–1135).: JMLR. org.
- [95] Fitting, M. (2012). *First-order logic and automated theorem proving*. Springer Science & Business Media.
- [96] Food, U., Administration, D., et al. (2019). Proposed regulatory framework for modifications to artificial intelligence. *Machine Learning (AI/ML)-based Software as a Medical Device (SaMD). Discussion Paper and Request for Feedback*. Available at: <https://www.fda.gov/media/122535/download>. Accessed June, 12.
- [97] Forgy, E. W. (1965). Cluster analysis of multivariate data: efficiency versus interpretability of classifications. *biometrics*, 21, 768–769.
- [98] Frid-Adar, M., Klang, E., Amitai, M., Goldberger, J., & Greenspan, H. (2018). Synthetic data augmentation using gan for improved liver lesion classification. In *Biomedical Imaging (ISBI 2018), 2018 IEEE 15th International Symposium on* (pp. 289–293).: IEEE.
- [99] Friedman, J., Hastie, T., & Tibshirani, R. (2001). *The elements of statistical learning*, volume 1. Springer series in statistics New York.
- [100] Gainza, P., Sverrisson, F., Monti, F., Rodola, E., Boscaini, D., Bronstein, M., & Correia, B. (2020). Deciphering interaction fingerprints from protein molecular surfaces using geometric deep learning. *Nature Methods*, 17(2).
- [101] Gale, W., Oakden-Rayner, L., Carneiro, G., Bradley, A. P., & Palmer, L. J. (2017). Detecting hip fractures with radiologist-level performance using deep neural networks. *arXiv preprint arXiv:1711.06504*.
- [102] Gao, Z., Fu, G., Ouyang, C., Tsutsui, S., Liu, X., Yang, J., Gessner, C., Foote, B., Wild, D., Ding, Y., et al. (2019). edge2vec: Representation learning using edge semantics for biomedical knowledge discovery. *BMC Bioinformatics*, 20(1).
- [103] Garrett, D. & Millican, P. (2011). *Reason, Induction, and Causation in Hume’s Philosophy*. Institute for Advanced Studies in the Humanities, The University of Edinburgh.
- [104] Gauthier, J. (2014). Conditional generative adversarial nets for convolutional face generation. *Class Project for Stanford CS231N: Convolutional Neural Networks for Visual Recognition, Winter semester, 2014*(5), 2.
- [105] Geelhoed, G. & Drury, E. (1982). Management of the adrenal” incidentaloma”. *Surgery*, 92(5), 866–874.

- [106] Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O., & Dahl, G. E. (2017). Neural message passing for quantum chemistry. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70* (pp. 1263–1272).: JMLR. org.
- [107] Girosi, F., Jones, M., & Poggio, T. (1995). Regularization theory and neural networks architectures. *Neural computation*, 7(2), 219–269.
- [108] Glocker, B., Robinson, R., Castro, D. C., Dou, Q., & Konukoglu, E. (2019). Machine learning with multi-site imaging data: An empirical study on the impact of scanner effects.
- [109] Gohil, S. K., Cao, C., Phelan, M., Tjoa, T., Rhee, C., Platt, R., & Huang, S. S. (2016). Impact of policies on the rise in sepsis incidence, 2000–2010. *Clinical Infectious Diseases*, 62(6), 695–703.
- [110] Gong, J. J., Naumann, T., Szolovits, P., & Guttag, J. V. (2017). Predicting clinical outcomes across changing electronic health record systems. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 1497–1505).
- [111] Goodfellow, I., Bengio, Y., & Courville, A. (2016a). *Deep learning*. MIT press.
- [112] Goodfellow, I., Bengio, Y., & Courville, A. (2016b). Representation learning. *Deep Learning*, (pp. 517–548).
- [113] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014a). Generative adversarial nets. In *Advances in neural information processing systems* (pp. 2672–2680).
- [114] Goodfellow, I. J., Shlens, J., & Szegedy, C. (2014b). Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.
- [115] Goodman, N. (1955). *Fact, fiction, and forecast*. Harvard University Press.
- [116] Graese, A., Rozsa, A., & Boulton, T. E. (2016). Assessing threat of adversarial examples on deep neural networks. In *Machine Learning and Applications (ICMLA), 2016 15th IEEE International Conference on* (pp. 69–74).: IEEE.
- [117] Gretton, A., Smola, A., Huang, J., Schmittfull, M., Borgwardt, K., & Schölkopf, B. (2009). Covariate shift by kernel mean matching. *Dataset shift in machine learning*, 3(4), 5.
- [118] Gu, K. & Sadiku, M. N. (1995). A triangular mesh random walk for dirichlet problems. *Journal of the Franklin Institute*, 332(5).
- [119] Guibas, J. T., Virdi, T. S., & Li, P. S. (2017). Synthetic medical images from dual generative adversarial networks. *arXiv preprint arXiv:1709.01872*.

- [120] Gulshan, V., Peng, L., Coram, M., Stumpe, M. C., Wu, D., Narayanaswamy, A., Venugopalan, S., Widner, K., Madams, T., Cuadros, J., et al. (2016). Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs. *Jama*, 316(22), 2402–2410.
- [121] Guo, C., Pleiss, G., Sun, Y., & Weinberger, K. Q. (2017). On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70* (pp. 1321–1330).: JMLR. org.
- [122] Hamilton, W., Ying, Z., & Leskovec, J. (2017). Inductive representation learning on large graphs. In *NIPS*.
- [123] Hansson, T., Oostenbrink, C., & van Gunsteren, W. (2002). Molecular dynamics simulations. *Current opinion in structural biology*, 12(2), 190–196.
- [124] Hartley, T., Lemire, G., Kernohan, K. D., Howley, H. E., Adams, D. R., & Boycott, K. M. (2020). New diagnostic approaches for undiagnosed rare genetic diseases. *Annual Review of Genomics and Human Genetics*, 21.
- [125] Håstad, J. & Goldmann, M. (1991). On the power of small-depth threshold circuits. *Computational Complexity*, 1(2), 113–129.
- [126] Hein, M., Andriushchenko, M., & Bitterwolf, J. (2019). Why relu networks yield high-confidence predictions far away from the training data and how to mitigate the problem. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 41–50).
- [127] Hendrycks, D. & Gimpel, K. (2016). A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*.
- [128] Hendrycks, D., Mazeika, M., & Dietterich, T. (2018). Deep anomaly detection with outlier exposure. *arXiv preprint arXiv:1812.04606*.
- [129] Hernán, M. A., Hernández-Díaz, S., & Robins, J. M. (2004). A structural approach to selection bias. *Epidemiology*, (pp. 615–625).
- [130] Hilborn, R. & Mangel, M. (1997). *The ecological detective: confronting models with data*. Princeton University Press.
- [131] Hinton, G. E., Osindero, S., & Teh, Y.-W. (2006). A fast learning algorithm for deep belief nets. *Neural computation*, 18(7), 1527–1554.
- [132] Hinton, G. E. & Roweis, S. T. (2003). Stochastic neighbor embedding. In *Advances in neural information processing systems* (pp. 857–864).
- [133] Hinton, G. E. & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *science*, 313(5786), 504–507.

- [134] Hodos, R., Zhang, P., Lee, H.-C., Duan, Q., Wang, Z., Clark, N. R., Ma'ayan, A., Wang, F., Kidd, B., Hu, J., et al. (2017). Cell-specific prediction and application of drug-induced gene expression profiles. In *Pacific Symposium on Biocomputing*, volume 23: World Scientific.
- [135] Hoffer, E. & Ailon, N. (2015). Deep metric learning using triplet network. In *International Workshop on Similarity-Based Pattern Recognition* (pp. 84–92): Springer.
- [136] Hotelling, H. (1992). Relations between two sets of variates. In *Breakthroughs in statistics* (pp. 162–190). Springer.
- [137] Howard, J. & Ruder, S. (2018). Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*.
- [138] Hu, W., Liu, B., Gomes, J., Zitnik, M., Liang, P., Pande, V., & Leskovec, J. (2019). Pre-training graph neural networks. *arXiv:1905.12265*.
- [139] Hu, Y., Gibson, E., Lee, L.-L., Xie, W., Barratt, D. C., Vercauteren, T., & Noble, J. A. (2017). Freehand ultrasound image simulation with spatially-conditioned generative adversarial networks. In *Molecular Imaging, Reconstruction and Analysis of Moving Body Organs, and Stroke Imaging and Treatment* (pp. 105–115). Springer.
- [140] Hume, D. (1739). *A treatise of human nature*. Oxford University Press.
- [141] Hume, D. (1748). *An Enquiry Concerning Human Understanding*. Oxford University Press.
- [142] Ilyas, A., Santurkar, S., Tsipras, D., Engstrom, L., Tran, B., & Madry, A. (2019). Adversarial examples are not bugs, they are features. In *Advances in Neural Information Processing Systems* (pp. 125–136).
- [143] Ingraham, J., Riesselman, A. J., Sander, C., & Marks, D. S. (2019). Learning protein structure with a differentiable simulator. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*: OpenReview.net.
- [144] Ishiguro, K., Maeda, S.-i., & Koyama, M. (2019). Graph warp module: an auxiliary module for boosting the power of graph neural networks. *arXiv:1902.01020*.
- [145] Iwata, M., Sawada, R., Iwata, H., Kotera, M., & Yamanishi, Y. (2017). Elucidating the modes of action for bioactive compounds in a cell-specific manner by large-scale chemically-induced transcriptomics. *Scientific reports*, 7, 40164.
- [146] Jain, A., Nundy, S., & Abbasi, K. (2014). Corruption: medicine’s dirty open secret.
- [147] Jia, R. & Liang, P. (2017). Adversarial examples for evaluating reading comprehension systems. *arXiv preprint arXiv:1707.07328*.

- [148] Jiang, J., Wei, Y., Feng, Y., Cao, J., & Gao, Y. (2019). Dynamic hypergraph neural networks. In *IJCAI*.
- [149] Jin, D., Jin, Z., Zhou, J. T., & Szolovits, P. (2019a). Is bert really robust. *A Strong Baseline for Natural Language Attack on Text Classification and Entailment. arXiv e-prints*, page.
- [150] Jin, D., Rossi, R. A., Koh, E., Kim, S., Rao, A., & Koutra, D. (2019b). Latent network summarization: Bridging network embedding and summarization. In *KDD*.
- [151] Jin, T., Cao, L., Zhang, B., Sun, X., Deng, C., & Ji, R. (2019c). Hypergraph induced convolutional manifold networks. In *IJCAI*.
- [152] Jin, W., Barzilay, R., & Jaakkola, T. (2020). Adaptive invariance for molecule property prediction. *arXiv preprint arXiv:2005.03004*.
- [153] Jutten, C. & Herault, J. (1991). Blind separation of sources, part i: An adaptive algorithm based on neuromimetic architecture. *Signal processing*, 24(1), 1–10.
- [154] Kannan, H., Kurakin, A., & Goodfellow, I. (2018). Adversarial logit pairing. *arXiv preprint arXiv:1803.06373*.
- [155] Karras, T., Aila, T., Laine, S., & Lehtinen, J. (2017). Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*.
- [156] Kearnes, S., McCloskey, K., Berndl, M., Pande, V., & Riley, P. (2016). Molecular graph convolutions: moving beyond fingerprints. *Journal of computer-aided molecular design*, 30(8), 595–608.
- [157] Kesselheim, A. S. & Brennan, T. A. (2005). Overbilling vs. downcoding—the battle between physicians and insurers. *New England Journal of Medicine*, 352(9), 855–857.
- [158] Kiani, A., Uyumazturk, B., Rajpurkar, P., Wang, A., Gao, R., Jones, E., Yu, Y., Langlotz, C. P., Ball, R. L., Montine, T. J., et al. (2020). Impact of a deep learning assistant on the histopathologic classification of liver cancer. *NPJ digital medicine*, 3(1), 1–8.
- [159] Kim, P.-J. & Price, N. D. (2011). Genetic co-occurrence network across sequenced microbes. *PLoS Computational Biology*, 7(12).
- [160] Kingma, D. P. & Welling, M. (2013). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.
- [161] Kiranyaz, S., Ince, T., & Gabbouj, M. (2015). Real-time patient-specific ecg classification by 1-d convolutional neural networks. *IEEE Transactions on Biomedical Engineering*, 63(3), 664–675.

- [162] Kishimoto, Y., Murawaki, Y., & Kurohashi, S. (2018). A knowledge-augmented neural network model for implicit discourse relation classification. In *Proceedings of the 27th International Conference on Computational Linguistics* (pp. 584–595).
- [163] Klambauer, G., Unterthiner, T., Mayr, A., & Hochreiter, S. (2017). Self-normalizing neural networks. In *Advances in neural information processing systems* (pp. 971–980).
- [164] Koch, G., Zemel, R., & Salakhutdinov, R. (2015). Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop*, volume 2.
- [165] Kohane, I. S., Masys, D. R., & Altman, R. B. (2006). The incidentalome: a threat to genomic medicine. *Jama*, 296(2), 212–215.
- [166] Kolter, J. Z. & Wong, E. (2017). Provable defenses against adversarial examples via the convex outer adversarial polytope. *arXiv preprint arXiv:1711.00851*.
- [167] Korkinof, D., Rijken, T., O’Neill, M., Yearsley, J., Harvey, H., & Glocker, B. (2018). High-resolution mammogram synthesis using progressive generative adversarial networks. *arXiv preprint arXiv:1807.03401*.
- [168] Krallinger, M., Rabal, O., Lourenco, A., Oyarzabal, J., & Valencia, A. (2017). Information retrieval and text mining technologies for chemistry. *Chemical reviews*, 117(12), 7673–7761.
- [169] Krizhevsky, A., Hinton, G., et al. (2009). Learning multiple layers of features from tiny images.
- [170] Kurakin, A., Goodfellow, I., & Bengio, S. (2016). Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*.
- [171] Lacoste, A., Luccioni, A., Schmidt, V., & Dandres, T. (2019). Quantifying the carbon emissions of machine learning. *arXiv preprint arXiv:1910.09700*.
- [172] Lake, B. M., Salakhutdinov, R., & Tenenbaum, J. B. (2015). Human-level concept learning through probabilistic program induction. *Science*, 350(6266), 1332–1338.
- [173] Lakshminarayanan, B., Pritzel, A., & Blundell, C. (2017). Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in neural information processing systems* (pp. 6402–6413).
- [174] Lamb, J., Crawford, E. D., Peck, D., Modell, J. W., Blat, I. C., Wrobel, M. J., Lerner, J., Brunet, J.-P., Subramanian, A., Ross, K. N., et al. (2006). The connectivity map: using gene-expression signatures to connect small molecules, genes, and disease. *science*, 313(5795), 1929–1935.
- [175] Landrum, G. et al. (2006). Rdkit: Open-source cheminformatics.

- [176] LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., & Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4), 541–551.
- [177] Lee, Y. Y. & Ventura, S. (2017). *lindia: Automated Linear Regression Diagnostic*. R package version 0.9.
- [178] Leek, J. T. & Storey, J. D. (2007). Capturing heterogeneity in gene expression studies by surrogate variable analysis. *PLoS genetics*, 3(9).
- [179] Li, J., Cai, D., & He, X. (2017). Learning graph-level representation for drug discovery. *arXiv:1709.03741*.
- [180] Li, J., Madry, A., Peebles, J., & Schmidt, L. (2018a). On the limitations of first order approximation in gan dynamics.
- [181] Li, Q., Cao, Z., Zhong, J., & Li, Q. (2019). Graph representation learning with encoding edges. *Neurocomputing*, 361.
- [182] Li, X. A., Tai, A., Arthur, D. W., Buchholz, T. A., Macdonald, S., Marks, L. B., Moran, J. M., Pierce, L. J., Rabinovitch, R., Taghian, A., et al. (2009). Variability of target and normal structure delineation for breast cancer radiotherapy: an rtog multi-institutional and multiobserver study. *International Journal of Radiation Oncology Biology Physics*, 73(3), 944–951.
- [183] Li, Y., Yang, M., & Zhang, Z. M. (2018b). A survey of multi-view representation learning. *IEEE Transactions on Knowledge and Data Engineering*.
- [184] Liang, S., Li, Y., & Srikant, R. (2017). Enhancing the reliability of out-of-distribution image detection in neural networks. *arXiv preprint arXiv:1706.02690*.
- [185] Lim, N. & Pavlidis, P. (2019). Evaluation of connectivity map shows limited reproducibility in drug repositioning. *BioRxiv*, (pp. 845693).
- [186] Lin, H., Li, R., Liu, Z., Chen, J., Yang, Y., Chen, H., Lin, Z., Lai, W., Long, E., Wu, X., et al. (2019). Diagnostic efficacy and therapeutic decision-making capacity of an artificial intelligence platform for childhood cataracts in eye clinics: a multicentre randomized controlled trial. *EClinicalMedicine*, 9, 52–59.
- [187] Liu, T.-P., Hsieh, Y.-Y., Chou, C.-J., & Yang, P.-M. (2018). Systematic polypharmacology and drug repurposing via an integrated 11000-based connectivity map database mining. *Royal Society open science*, 5(11), 181321.
- [188] Lo, Y.-C., Rensi, S. E., Torng, W., & Altman, R. B. (2018). Machine learning in chemoinformatics and drug discovery. *Drug discovery today*, 23(8), 1538–1546.

- [189] Long, E., Lin, H., Liu, Z., Wu, X., Wang, L., Jiang, J., An, Y., Lin, Z., Li, X., Chen, J., et al. (2017). An artificial intelligence platform for the multihospital collaborative management of congenital cataracts. *Nature biomedical engineering*, 1(2), 1–8.
- [190] Lowd, D. & Meek, C. (2005a). Adversarial learning. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining* (pp. 641–647).: ACM.
- [191] Lowd, D. & Meek, C. (2005b). Good word attacks on statistical spam filters. In *CEAS*, volume 2005.
- [192] Lowe, R., Shirley, N., Bleackley, M., Dolan, S., & Shafee, T. (2017). Transcriptomics technologies. *PLoS computational biology*, 13(5).
- [193] Lu, J., Sibai, H., Fabry, E., & Forsyth, D. (2017). No need to worry about adversarial examples in object detection in autonomous vehicles. *arXiv preprint arXiv:1707.03501*.
- [194] Lyell, D. & Coiera, E. (2017). Automation bias and verification complexity: a systematic review. *Journal of the American Medical Informatics Association*, 24(2), 423–431.
- [195] Madry, A., Makelov, A., Schmidt, L., Tsipras, D., & Vladu, A. (2017). Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*.
- [196] Maglott, D., Ostell, J., Pruitt, K. D., & Tatusova, T. (2005). Entrez gene: gene-centered information at ncbi. *Nucleic Acids Research*, 33(suppl_1).
- [197] Maiella, S., Rath, A., Angin, C., Mousson, F., & Kremp, O. (2013). Orphanet and its consortium: where to find expert-validated information on rare diseases. *Revue Neurologique*, 169.
- [198] Manning, C., Raghavan, P., & Schütze, H. (2010). Introduction to information retrieval. *Natural Language Engineering*, 16(1), 100–103.
- [199] Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to information retrieval*. Cambridge university press.
- [200] Manrai, A. K., Bhatia, G., Strymish, J., Kohane, I. S., & Jain, S. H. (2014). Medicine’s uncomfortable relationship with math: calculating positive predictive value. *JAMA internal medicine*, 174(6), 991–993.
- [201] Mayo, D. G. (1996). Ducks, rabbits, and normal science: Recasting the kuhn’s-eye view of popper’s demarcation of science. *The British Journal for the Philosophy of Science*, 47(2), 271–290.

- [202] McDermott, M., Wang, J., Zhao, W. N., Sheridan, S. D., Szolovits, P., Kohane, I., Haggarty, S. J., & Perlis, R. H. (2019). Deep learning benchmarks on 11000 gene expression data. *IEEE/ACM transactions on computational biology and bioinformatics*.
- [203] McInnes, L., Healy, J., & Melville, J. (2018). Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*.
- [204] Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., & Galstyan, A. (2019). A survey on bias and fairness in machine learning. *arXiv preprint arXiv:1908.09635*.
- [205] Meng, C., Yang, J., Ribeiro, B., & Neville, J. (2019). Hats: A hierarchical sequence-attention framework for inductive set-of-sets embeddings. In *KDD*.
- [206] Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013a). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.
- [207] Mikolov, T., Deoras, A., Kombrink, S., Burget, L., & Černocký, J. (2011). Empirical evaluation and combination of advanced language modeling techniques. In *Twelfth Annual Conference of the International Speech Communication Association*.
- [208] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013b). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems* (pp. 3111–3119).
- [209] Miller, E. G., Matsakis, N. E., & Viola, P. A. (2000). Learning from one example through shared densities on transforms. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No. PR00662)*, volume 1 (pp. 464–471).: IEEE.
- [210] Miller, P. E., Rawls, A. W., Pundlik, S. J., & Woodard, D. L. (2010). Personal identification using periocular skin texture. In *Proceedings of the 2010 ACM Symposium on Applied Computing* (pp. 1496–1500).: ACM.
- [211] Mirza, M. & Osindero, S. (2014). Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*.
- [212] Mitchell, T. M. (1980). *The need for biases in learning generalizations*. Department of Computer Science, Laboratory for Computer Science Research
- [213] Mitchell, T. M. et al. (1997). Machine learning.
- [214] Mohseni, S., Pitale, M., Yadawa, J., & Wang, Z. (2020). Self-supervised learning for generalizable out-of-distribution detection.
- [215] Mordaunt, D., Cox, D., & Fuller, M. (2020). Metabolomics to improve the diagnostic efficiency of inborn errors of metabolism. *International Journal of Molecular Sciences*, 21(4).

- [216] Moreno-Torres, J. G., Raeder, T., Alaiz-Rodríguez, R., Chawla, N. V., & Herrera, F. (2012). A unifying view on dataset shift in classification. *Pattern recognition*, 45(1), 521–530.
- [217] Muggleton, S. (1991). Inductive logic programming. *New generation computing*, 8(4), 295–318.
- [218] Murphy, R. L., Srinivasan, B., Rao, V., & Ribeiro, B. (2019a). Janossy pooling: Learning deep permutation-invariant functions for variable-size inputs. *ICLR*.
- [219] Murphy, R. L., Srinivasan, B., Rao, V., & Ribeiro, B. (2019b). Relational pooling for graph representations. *ICML*.
- [220] Musa, A., Ghorai, L. S., Zhang, S.-D., Glazko, G., Yli-Harja, O., Dehmer, M., Haibe-Kains, B., & Emmert-Streib, F. (2017). A review of connectivity map and computational approaches in pharmacogenomics. *Briefings in bioinformatics*, 19(3), 506–523.
- [221] Musa, A., Tripathi, S., Kandhavelu, M., Dehmer, M., & Emmert-Streib, F. (2018). Harnessing the biological complexity of big data from lincs gene expression signatures. *PloS one*, 13(8), e0201937.
- [222] Musgrave, K., Belongie, S., & Lim, S.-N. (2020). A metric learning reality check. *arXiv preprint arXiv:2003.08505*.
- [223] Napolitano, F., Carrella, D., Mandriani, B., Pisonero-Vaquero, S., Sirci, F., Medina, D. L., Brunetti-Pierri, N., & Di Bernardo, D. (2018). gene2drug: a computational tool for pathway-based rational drug repositioning. *Bioinformatics*, 34(9), 1498–1505.
- [224] Ness, R. B., Committee, J. P., et al. (2007). Influence of the hipaa privacy rule on health research. *Jama*, 298(18), 2164–2170.
- [225] Nestor, B., McDermott, M., Boag, W., Berner, G., Naumann, T., Hughes, M. C., Goldenberg, A., & Ghassemi, M. (2019). Feature robustness in non-stationary health records: caveats to deployable model performance in common clinical machine learning tasks. *arXiv preprint arXiv:1908.00690*.
- [226] Newman, M. E. (2004). Detecting community structure in networks. *The European Physical Journal B*, 38(2).
- [227] Newman, M. E. & Clauset, A. (2016). Structure and inference in annotated networks. *Nature Communications*, 7(1).
- [228] Ngiam, J., Khosla, A., Kim, M., Nam, J., Lee, H., & Ng, A. Y. (2011). Multimodal deep learning.
- [229] Ni, J., Muhlstein, L., & McAuley, J. (2019). Modeling heart rate and activity data for personalized fitness recommendation. In *WWW*.

- [230] Nie, D., Trullo, R., Lian, J., Petitjean, C., Ruan, S., Wang, Q., & Shen, D. (2017). Medical image synthesis with context-aware generative adversarial networks. In *International Conference on Medical Image Computing and Computer-Assisted Intervention* (pp. 417–425).: Springer.
- [231] Niepert, M., Ahmed, M., & Kutzkov, K. (2016). Learning convolutional neural networks for graphs. In *ICML*.
- [232] Njeh, C. (2008). Tumor delineation: The weakest link in the search for accuracy in radiotherapy. *Journal of medical physics/Association of Medical Physicists of India*, 33(4), 136.
- [233] Obermeyer, Z., Powers, B., Vogeli, C., & Mullainathan, S. (2019). Dissecting racial bias in an algorithm used to manage the health of populations. *Science*, 366(6464), 447–453.
- [234] of Health, U. D., Services, H., et al. (1992). Medicare, medicaid and clia programs: regulations implementing the clinical laboratory improvement amendments of 1988 (clia). final rule. *Fed Regist*, 57(40), 7002–7186.
- [235] Oliphant, T. E. (2006). *A guide to NumPy*, volume 1. Trelgol Publishing USA.
- [236] Pan, S. J. & Yang, Q. (2009). A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10), 1345–1359.
- [237] Papernot, N., Carlini, N., Goodfellow, I., Feinman, R., Faghri, F., Matyasko, A., Hambardzumyan, K., Juang, Y.-L., Kurakin, A., Sheatsley, R., et al. (2016a). cleverhans v2. 0.0: an adversarial machine learning library. *arXiv preprint arXiv:1610.00768*.
- [238] Papernot, N., McDaniel, P., & Goodfellow, I. (2016b). Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. *arXiv preprint arXiv:1605.07277*.
- [239] Papernot, N., McDaniel, P., Wu, X., Jha, S., & Swami, A. (2016c). Distillation as a defense to adversarial perturbations against deep neural networks. In *Security and Privacy (SP), 2016 IEEE Symposium on* (pp. 582–597).: IEEE.
- [240] Parmar, N., Vaswani, A., Uszkoreit, J., Kaiser, Ł., Shazeer, N., Ku, A., & Tran, D. (2018). Image transformer. *arXiv preprint arXiv:1802.05751*.
- [241] Paschali, M., Conjeti, S., Navarro, F., & Navab, N. (2018). Generalizability vs. robustness: Adversarial examples for medical imaging. *arXiv preprint arXiv:1804.00504*.
- [242] Pearl, J. & Bareinboim, E. (2011). Transportability of causal and statistical relations: A formal approach. In *Twenty-fifth AAAI conference on artificial intelligence*.

- [243] Pearson, K. (1901). Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11), 559–572.
- [244] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. (2011). Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct), 2825–2830.
- [245] Peng, J., Wang, X., & Shang, X. (2019). Combining gene ontology with deep neural networks to enhance the clustering of single cell rna-seq data. *BMC bioinformatics*, 20(8), 284.
- [246] Perrett, R. W. (1984). The problem of induction in indian philosophy. *Philosophy East and West*, 34(2), 161–174.
- [247] Pfohl, S., Marafino, B., Coulet, A., Rodriguez, F., Palaniappan, L., & Shah, N. H. (2019). Creating fair models of atherosclerotic cardiovascular disease risk. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society* (pp. 271–278).
- [248] Picavet, E., Morel, T., Cassiman, D., & Simoens, S. (2014). Shining a light in the black box of orphan drug pricing. *Orphanet journal of rare diseases*, 9(1), 62.
- [249] Pickering, A. (2019a). ccdata: Data for combination connectivity mapping (ccmap) package. R package version 1.12.0.
- [250] Pickering, A. (2019b). crossmeta: Cross platform meta-analysis of microarray data. R package version 1.12.0.
- [251] Piñero, J., Bravo, À., Queralt-Rosinach, N., Gutiérrez-Sacristán, A., Deu-Pons, J., Centeno, E., García-García, J., Sanz, F., & Furlong, L. I. (2016). Disgenet: a comprehensive platform integrating information on human disease-associated genes and variants. *Nucleic Acids Research*.
- [252] Popper, K. (2014). *Conjectures and refutations: The growth of scientific knowledge*. routledge.
- [253] Prasad, V. K. & Cifu, A. S. (2015). *Ending medical reversal: improving outcomes, saving lives*. JHU Press.
- [254] Prechelt, L. (1998). Early stopping-but when? In *Neural Networks: Tricks of the trade* (pp. 55–69). Springer.
- [255] Pronovost, P. J. (2018). *Procuring interoperability: achieving high-quality, connected, and person-centered care*. NAM. EDU.
- [256] Qamber, S., Waheed, Z., & Akram, M. U. (2012). Personal identification system based on vascular pattern of human retina. In *Biomedical Engineering Conference (CIBEC), 2012 Cairo International* (pp. 64–67): IEEE.

- [257] Queen, J. P., Quinn, G. P., & Keough, M. J. (2002). *Experimental design and data analysis for biologists*. Cambridge University Press.
- [258] Quionero-Candela, J., Sugiyama, M., Schwaighofer, A., & Lawrence, N. D. (2009). *Dataset shift in machine learning*. The MIT Press.
- [259] R Core Team (2018). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.
- [260] Rabanser, S., Günnemann, S., & Lipton, Z. (2019). Failing loudly: an empirical study of methods for detecting dataset shift. In *Advances in Neural Information Processing Systems* (pp. 1394–1406).
- [261] Raghunathan, A., Steinhardt, J., & Liang, P. (2018). Certified defenses against adversarial examples. *arXiv preprint arXiv:1801.09344*.
- [262] Rajkomar, A., Hardt, M., Howell, M. D., Corrado, G., & Chin, M. H. (2018). Ensuring fairness in machine learning to advance health equity. *Annals of internal medicine*, 169(12), 866–872.
- [263] Rajpurkar, P., Irvin, J., Zhu, K., Yang, B., Mehta, H., Duan, T., Ding, D., Bagul, A., Langlotz, C., Shpanskaya, K., et al. (2017). Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning. *arXiv preprint arXiv:1711.05225*.
- [264] Rampásek, L., Hidru, D., Smirnov, P., Haibe-Kains, B., & Goldenberg, A. (2019). Dr. vae: improving drug response prediction via modeling of drug perturbation effects. *Bioinformatics*.
- [265] Ramsundar, B., Kearnes, S., Riley, P., Webster, D., Konerding, D., & Pande, V. (2015). Massively multitask networks for drug discovery. *arXiv preprint arXiv:1502.02072*.
- [266] Ratwani, R. M., Savage, E., Will, A., Arnold, R., Khairat, S., Miller, K., Fairbanks, R. J., Hodgkins, M., & Hettinger, A. Z. (2018). A usability and safety analysis of electronic health records: a multi-center study. *Journal of the American Medical Informatics Association*, 25(9), 1197–1201.
- [267] Ravi, S. & Larochelle, H. (2016). Optimization as a model for few-shot learning.
- [268] Reynolds, K., Muntner, P., & Fonseca, V. (2005). Metabolic syndrome: underrated or underdiagnosed?
- [269] Rieck, B., Fugacci, U., Lukasczyk, J., & Leitte, H. (2017). Clique community persistence: A topological visual analysis approach for complex networks. *IEEE Transactions on Visualization and Computer Graphics*, 24(1).

- [270] Robinson, P. N., Köhler, S., Bauer, S., Seelow, D., Horn, D., & Mundlos, S. (2008). The human phenotype ontology: a tool for annotating and analyzing human hereditary disease. *The American Journal of Human Genetics*, 83(5).
- [271] Rogers, D. & Hahn, M. (2010). Extended-connectivity fingerprints. *Journal of chemical information and modeling*, 50(5), 742–754.
- [272] Rothwell, P. M. (2005). External validity of randomised controlled trials: “to whom do the results of this trial apply?”. *The Lancet*, 365(9453), 82–93.
- [273] Roweis, S. T. & Saul, L. K. (2000). Nonlinear dimensionality reduction by locally linear embedding. *science*, 290(5500), 2323–2326.
- [274] Sanders, T. B., Bowens, F. M., Pierce, W., Stasher-Booker, B., Thompson, E. Q., & Jones, W. A. (2012). The road to icd-10-cm/pcs implementation: forecasting the transition for providers, payers, and other healthcare organizations. *Perspectives in health information management/AHIMA, American Health Information Management Association*, 9(Winter).
- [275] Sankar, P. L. & Parker, L. S. (2017). The precision medicine initiative’s all of us research program: an agenda for research on its ethical, legal, and social issues. *Genetics in Medicine*, 19(7), 743–750.
- [276] Santoro, A., Bartunov, S., Botvinick, M., Wierstra, D., & Lillicrap, T. (2016). Meta-learning with memory-augmented neural networks. In *International conference on machine learning* (pp. 1842–1850).
- [277] Santurkar, S., Schmidt, L., & Madry, A. (2017). A classification-based perspective on gan distributions. *arXiv preprint arXiv:1711.00970*.
- [278] Sariyuce, A. E., Seshadhri, C., Pinar, A., & Catalyurek, U. V. (2015). Finding the hierarchy of dense subgraphs using nucleus decompositions. In *WWW*.
- [279] Satchidanand, S. N., Ananthapadmanaban, H., & Ravindran, B. (2015). Extended discriminative random walk: a hypergraph approach to multi-view multi-relational transductive learning. In *IJCAI*.
- [280] Schmidhuber, J. (1987). *Evolutionary principles in self-referential learning, or on learning how to learn: the meta-meta-... book*. PhD thesis, Technische Universität München.
- [281] Schulam, P. & Saria, S. (2017). Reliable decision support using counterfactual models. In *Advances in Neural Information Processing Systems* (pp. 1697–1708).
- [282] Segler, M. H., Preuss, M., & Waller, M. P. (2018). Planning chemical syntheses with deep neural networks and symbolic ai. *Nature*, 555(7698), 604–610.
- [283] Seidman, S. B. (1983). Network structure and minimum degree. *Social Networks*, 5(3).

- [284] Shimodaira, H. (2000). Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of statistical planning and inference*, 90(2), 227–244.
- [285] Simon, G. (1971). *Principles of chest X-ray diagnosis*. Butterworths.
- [286] Sinha, A., Namkoong, H., & Duchi, J. (2017). Certifying some distributional robustness with principled adversarial training. *arXiv preprint arXiv:1710.10571*.
- [287] Slavova, S., Costich, J. F., Luu, H., Fields, J., Gabella, B. A., Tarima, S., & Bunn, T. L. (2018). Interrupted time series design to evaluate the effect of the icd-9-cm to icd-10-cm coding transition on injury hospitalization trends. *Injury epidemiology*, 5(1), 1–12.
- [288] Slowikowski, K. (2018). *ggrepel: Automatically Position Non-Overlapping Text Labels with 'ggplot2'*. R package version 0.8.0.
- [289] Smirnov, P., Safikhani, Z., El-Hachem, N., Wang, D., She, A., Olsen, C., Freeman, M., Selby, H., Gendoo, D. M., Grossmann, P., et al. (2016). Pharmacogx: an r package for analysis of large pharmacogenomic datasets. *Bioinformatics*, 32(8), 1244–1246.
- [290] Smith, A. (2017). The cost of drugs for rare diseases is threatening the us health care system. *Harvard Business Review*, 7.
- [291] Smyth, G. K. (2005). Limma: linear models for microarray data. In *Bioinformatics and computational biology solutions using R and Bioconductor* (pp. 397–420). Springer.
- [292] Snell, J., Swersky, K., & Zemel, R. (2017). Prototypical networks for few-shot learning. In *Advances in neural information processing systems* (pp. 4077–4087).
- [293] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1), 1929–1958.
- [294] Steckler, A. & McLeroy, K. R. (2008). The importance of external validity.
- [295] Stokes, J. M., Yang, K., Swanson, K., Jin, W., Cubillos-Ruiz, A., Donghia, N. M., MacNair, C. R., French, S., Carfrae, L. A., Bloom-Ackerman, Z., et al. (2020). A deep learning approach to antibiotic discovery. *Cell*, 180(4), 688–702.
- [296] Stoller, J. K. (2018). The challenge of rare diseases. *Chest*, 153(6), 1309–1314.
- [297] Storkey, A. (2009). When training and test sets are different: characterizing learning transfer. *Dataset shift in machine learning*, (pp. 3–28).
- [298] Subbaswamy, A. & Saria, S. (2018). Counterfactual normalization: Proactively addressing dataset shift and improving reliability using causal mechanisms. *arXiv preprint arXiv:1808.03253*.

- [299] Subbaswamy, A. & Saria, S. (2020). I-spec: An end-to-end framework for learning trans-portable, shift-stable models. *arXiv preprint arXiv:2002.08948*.
- [300] Subramanian, A., Narayan, R., Corsello, S. M., Peck, D. D., Natoli, T. E., Lu, X., Gould, J., Davis, J. F., Tubelli, A. A., Asiedu, J. K., et al. (2017). A next generation connectivity map: L1000 platform and the first 1,000,000 profiles. *Cell*, 171(6), 1437–1452.
- [301] Subramanian, A., Tamayo, P., Mootha, V. K., Mukherjee, S., Ebert, B. L., Gillette, M. A., Paulovich, A., Pomeroy, S. L., Golub, T. R., Lander, E. S., et al. (2005). Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. *Proceedings of the National Academy of Sciences*, 102(43).
- [302] Subramanya, A., Srinivas, S., & Babu, R. V. (2017). Confidence estimation in deep neural networks via density modelling. *arXiv preprint arXiv:1707.07013*.
- [303] Sudlow, C., Gallacher, J., Allen, N., Beral, V., Burton, P., Danesh, J., Downey, P., Elliott, P., Green, J., Landray, M., et al. (2015). Uk biobank: an open access resource for identifying the causes of a wide range of complex diseases of middle and old age. *PLoS medicine*, 12(3).
- [304] Sugiyama, M., Krauledat, M., & M  ller, K.-R. (2007). Covariate shift adaptation by importance weighted cross validation. *Journal of Machine Learning Research*, 8(May), 985–1005.
- [305] Sun, E. C., Dutton, R. P., & Jena, A. B. (2018). Comparison of anesthesia times and billing patterns by anesthesia practitioners. *JAMA network open*, 1(7), e184288–e184288.
- [306] Sun, F.-Y., Qu, M., Hoffmann, J., Huang, C.-W., & Tang, J. (2019). vGraph: A generative model for joint community detection and node representation learning. In *NeurIPS*.
- [307] Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Advances in neural information processing systems* (pp. 3104–3112).
- [308] Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., & Fergus, R. (2014). Intriguing properties of neural networks. *arXiv*, (pp. 1–10).
- [309] Takebe, N., McShane, L., & Conley, B. (2015). Exceptional responders—discovering predictive biomarkers. *Nature Reviews Clinical Oncology*, 12(3), 132–134.
- [310] Tenenbaum, J. B., De Silva, V., & Langford, J. C. (2000). A global geometric framework for nonlinear dimensionality reduction. *science*, 290(5500), 2319–2323.
- [311] Tenopir, C., Allard, S., Douglass, K., Aydinoglu, A. U., Wu, L., Read, E., Manoff, M., & Frame, M. (2011). Data sharing by scientists: practices and perceptions. *PloS one*, 6(6), e21101.
- [312] Teru, K. K., Denis, E., & Hamilton, W. L. (2020). Inductive relation prediction by subgraph reasoning. In *ICML*.

- [313] Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1), 267–288.
- [314] Tong, W., Xie, Q., Hong, H., Shi, L., Fang, H., & Perkins, R. (2004). Assessment of prediction confidence and domain extrapolation of two structure–activity relationship models for predicting estrogen receptor binding activity. *Environmental health perspectives*, 112(12), 1249–1254.
- [315] Tramèr, F., Papernot, N., Goodfellow, I., Boneh, D., & McDaniel, P. (2017). The Space of Transferable Adversarial Examples. *arXiv*, (pp. 1–15).
- [316] Tse, J. & You, W. (2011). How accurate is the electronic health record?-a pilot study evaluating information accuracy in a primary care setting. *Studies in health technology and informatics*, 168, 158–164.
- [317] Tsipras, D., Santurkar, S., Engstrom, L., Turner, A., & Madry, A. (2018). Robustness may be at odds with accuracy.
- [318] Twerenbold, R., Reichlin, T., Reiter, M., & Muller, C. (2011). High-sensitive cardiac troponin: friend or foe. *Swiss Med Wkly*, 141, w13202.
- [319] Tzeng, R.-C. & Wu, S.-H. (2019). Distributed, egocentric representations of graphs for detecting critical structures. In *ICML*.
- [320] Van Engelen, J. E. & Hoos, H. H. (2020). A survey on semi-supervised learning. *Machine Learning*, 109(2), 373–440.
- [321] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems* (pp. 5998–6008).
- [322] Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., & Bengio, Y. (2017). Graph attention networks. *arXiv:1710.10903*.
- [323] Veličković, P., Fedus, W., Hamilton, W. L., Liò, P., Bengio, Y., & Hjelm, R. D. (2019). Deep graph infomax. In *ICLR*.
- [324] Vines, T. H., Albert, A. Y., Andrew, R. L., Débarre, F., Bock, D. G., Franklin, M. T., Gilbert, K. J., Moore, J.-S., Renaut, S., & Rennison, D. J. (2014). The availability of research data declines rapidly with article age. *Current biology*, 24(1), 94–97.
- [325] Vinyals, O., Blundell, C., Lillicrap, T., Wierstra, D., et al. (2016). Matching networks for one shot learning. In *Advances in neural information processing systems* (pp. 3630–3638).
- [326] Vogt, H., Green, S., Ekstrøm, C. T., & Brodersen, J. (2019). How precision medicine and screening with big data could increase overdiagnosis. *Bmj*, 366, l5270.

- [327] Wang, P., Fu, Y., Xiong, H., & Li, X. (2019). Adversarial substructured representation learning for mobile user profiling. In *KDD*.
- [328] Wang, S.-c. (2012). Competencies and experiential requirements in radiology training. In *Radiology Education* (pp. 55–66). Springer.
- [329] Wang, X., Peng, Y., Lu, L., Lu, Z., Bagheri, M., & Summers, R. M. (2017). Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 3462–3471).: IEEE.
- [330] Wang, X., Terfve, C., Rose, J. C., & Markowetz, F. (2011a). Htsanalyzer: an r/bioconductor package for integrated network analysis of high-throughput screens. *Bioinformatics*, 27(6), 879–880.
- [331] Wang, Y., Xiao, J., Suzek, T. O., Zhang, J., Wang, J., Zhou, Z., Han, L., Karapetyan, K., Dracheva, S., Shoemaker, B. A., et al. (2011b). Pubchem’s bioassay database. *Nucleic acids research*, 40(D1), D400–D412.
- [332] Wang, Z., Lachmann, A., Keenan, A. B., & Ma’ayan, A. (2018). L1000fwd: fireworks visualization of drug-induced transcriptomic signatures. *Bioinformatics*, 34(12), 2150–2152.
- [333] Wang, Z., Monteiro, C. D., Jagodnik, K. M., Fernandez, N. F., Gundersen, G. W., Rouillard, A. D., Jenkins, S. L., Feldmann, A. S., Hu, K. S., McDermott, M. G., et al. (2016). Extraction and analysis of signatures from the gene expression omnibus by the crowd. *Nature communications*, 7(1), 1–11.
- [334] Weber, G. M., Adams, W. G., Bernstam, E. V., Bickel, J. P., Fox, K. P., Marsolo, K., Raghavan, V. A., Turchin, A., Zhou, X., Murphy, S. N., et al. (2017). Biases introduced by filtering electronic health records for patients with “complete data”. *Journal of the American Medical Informatics Association*, 24(6), 1134–1141.
- [335] Weiner, S. J., Wang, S., Kelly, B., Sharma, G., & Schwartz, A. (2020). How accurate is the medical record? a comparison of the physician’s note with a concealed audio recording in unannounced standardized patient encounters. *Journal of the American Medical Informatics Association*, 27(5), 770–775.
- [336] Weininger, D., Weininger, A., & Weininger, J. L. (1989). Smiles. 2. algorithm for generation of unique smiles notation. *Journal of chemical information and computer sciences*, 29(2), 97–101.
- [337] Wendler, C., Püschel, M., & Alistarh, D. (2019). Powerset convolutional neural networks. In *NIPS*.

- [338] Weston, J., Collobert, R., Sinz, F., Bottou, L., & Vapnik, V. (2006). Inference with the universum. In *Proceedings of the 23rd international conference on Machine learning* (pp. 1009–1016).
- [339] Wickham, H. (2016). *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York.
- [340] Wickham, H. (2017). *tidyverse: Easily Install and Load the 'Tidyverse'*. R package version 1.2.1.
- [341] Wickham, H., François, R., Henry, L., & Müller, K. (2019). *dplyr: A Grammar of Data Manipulation*. R package version 0.8.3.
- [342] Willemink, M. J., Koszek, W. A., Hardell, C., Wu, J., Fleischmann, D., Harvey, H., Folio, L. R., Summers, R. M., Rubin, D. L., & Lungren, M. P. (2020). Preparing medical imaging data for machine learning. *Radiology*, 295(1), 4–15.
- [343] Wolterink, J. M., Dinkla, A. M., Savenije, M. H., Seevinck, P. R., van den Berg, C. A., & Išgum, I. (2017). Deep mr to ct synthesis using unpaired data. In *International Workshop on Simulation and Synthesis in Medical Imaging* (pp. 14–23).: Springer.
- [344] Wu, C.-Y., Manmatha, R., Smola, A. J., & Krahenbuhl, P. (2017). Sampling matters in deep embedding learning. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 2840–2848).
- [345] Wu, E., Wu, K., Cox, D., & Lotter, W. (2018a). Conditional infilling gans for data augmentation in mammogram classification. In *Image Analysis for Moving Organ, Breast, and Thoracic Images* (pp. 98–106). Springer.
- [346] Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., & Philip, S. Y. (2020). A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*.
- [347] Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., & Yu, P. S. (2019). A comprehensive survey on graph neural networks. *arXiv:1901.00596*.
- [348] Wu, Z., Ramsundar, B., Feinberg, E. N., Gomes, J., Geniesse, C., Pappu, A. S., Leswing, K., & Pande, V. (2018b). Moleculenet: a benchmark for molecular machine learning. *Chemical science*, 9(2), 513–530.
- [349] Wynia, M. K., Cummins, D. S., VanGeest, J. B., & Wilson, I. B. (2000). Physician manipulation of reimbursement rules for patients: between a rock and a hard place. *Jama*, 283(14), 1858–1865.
- [350] Xie, J., Kelley, S., & Szymanski, B. K. (2013). Overlapping community detection in networks: The state-of-the-art and comparative study. *ACM CSUR*, 45(4).

- [351] Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., Zemel, R., & Bengio, Y. (2015). Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning* (pp. 2048–2057).
- [352] Xu, K., Hu, W., Leskovec, J., & Jegelka, S. (2018a). How powerful are graph neural networks? *arXiv:1810.00826*.
- [353] Xu, K., Li, C., Tian, Y., Sonobe, T., Kawarabayashi, K.-i., & Jegelka, S. (2018b). Representation learning on graphs with jumping knowledge networks. *arXiv:1806.03536*.
- [354] Yadati, N., Nimishakavi, M., Yadav, P., Nitin, V., Louis, A., & Talukdar, P. (2019). HyperGCN: A new method for training graph convolutional networks on hypergraphs. In *NeurIPS*.
- [355] Yang, J. & Leskovec, J. (2015). Defining and evaluating network communities based on ground-truth. *Knowledge and Information Systems*, 42(1).
- [356] Yang, K., Swanson, K., Jin, W., Coley, C. W., Eiden, P., Gao, H., Guzman-Perez, A., Hopper, T., Kelley, B., Mathea, M., et al. (2019). Analyzing learned molecular representations for property prediction. *Journal of chemical information and modeling*.
- [357] Yang, Q. & Wu, X. (2006). 10 challenging problems in data mining research. *International Journal of Information Technology & Decision Making*, 5(04), 597–604.
- [358] Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., & Hovy, E. (2016). Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.
- [359] Yerramilli, D., Charrow, A., & Caplan, A. (2018). How should clinicians respond when patients are influenced by celebrities’ cancer stories? *AMA journal of ethics*, 20(11), 1075–1081.
- [360] Ying, Z., Bourgeois, D., You, J., Zitnik, M., & Leskovec, J. (2019). Gnnexplainer: Generating explanations for graph neural networks. In *NeurIPS*.
- [361] Yoshida, T., Takeuchi, I., & Karasuyama, M. (2019). Learning interpretable metric between graphs: Convex formulation and computation with graph mining. In *KDD*.
- [362] You, J., Ying, R., & Leskovec, J. (2019). Position-aware graph neural networks. In *ICML*.
- [363] Zaheer, M., Kottur, S., Ravanbakhsh, S., Poczos, B., Salakhutdinov, R. R., & Smola, A. J. (2017). Deep sets. In *NIPS*.
- [364] Zeng, H., Zhou, H., Srivastava, A., Kannan, R., & Prasanna, V. (2020). GraphSAINT: Graph sampling based inductive learning method. In *ICLR*.

- [365] Zhang, D., Wang, J., Wang, F., & Zhang, C. (2008). Semi-supervised classification with universum. In *Proceedings of the 2008 SIAM International Conference on Data Mining* (pp. 323–333).: SIAM.
- [366] Zhang, H., Cisse, M., Dauphin, Y. N., & Lopez-Paz, D. (2017a). mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*.
- [367] Zhang, H., Xu, T., Li, H., Zhang, S., Wang, X., Huang, X., & Metaxas, D. (2017b). Stackgan++: Realistic image synthesis with stacked generative adversarial networks. *arXiv:1710.10916*.
- [368] Zhang, K., Gong, M., & Schölkopf, B. (2015). Multi-source domain adaptation: A causal view. In *Twenty-ninth AAAI conference on artificial intelligence*.
- [369] Zhang, K., Schölkopf, B., Muandet, K., & Wang, Z. (2013). Domain adaptation under target and conditional shift. In *International Conference on Machine Learning* (pp. 819–827).
- [370] Zhang, M. & Chen, Y. (2018). Link prediction based on graph neural networks. In *NeurIPS*.
- [371] Zhang, M., Cui, Z., Jiang, S., & Chen, Y. (2018). Beyond link prediction: Predicting hyperlinks in adjacency space. In *AAAI*.
- [372] Zhang, Y., Hare, J., & Prugel-Bennett, A. (2019). Deep set prediction networks. In *Advances in Neural Information Processing Systems* (pp. 3207–3217).
- [373] Zhou, D., Huang, J., & Schölkopf, B. (2007). Learning with hypergraphs: Clustering, classification, and embedding. In *NIPS*.
- [374] Zhou, J., Cui, G., Zhang, Z., Yang, C., Liu, Z., Wang, L., Li, C., & Sun, M. (2018). Graph neural networks: A review of methods and applications. *arXiv:1812.08434*.
- [375] Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., Xiong, H., & He, Q. (2019). A comprehensive survey on transfer learning. *arXiv preprint arXiv:1911.02685*.
- [376] Zitnik, M., Agrawal, M., & Leskovec, J. (2018a). Modeling polypharmacy side effects with graph convolutional networks. *Bioinformatics*, 34(13).
- [377] Zitnik, M., Sosič, R., Maheshwari, S., & Leskovec, J. (2018b). BioSNAP Datasets: Stanford biomedical network dataset collection. <http://snap.stanford.edu/biodata>.



THIS THESIS WAS TYPESET using $\LaTeX$, originally developed by Leslie Lamport and based on Donald Knuth's $\TeX$. The body text is set in 11 point Egenolff-Berner Garamond, a revival of Claude Garamont's humanist typeface. The above illustration, "Science Experiment 02", was created by Ben Schlitter and released under [CC BY-NC-ND 3.0](#). A template that can be used to format a PhD thesis with this look and feel has been released under the permissive MIT (X11) license, and can be found online at github.com/suchow/Dissertate or from its author, Jordan Suchow, at suchow@post.harvard.edu.