Mechanistic deep learning for perturbation biology: Application to precision oncology

Nathaniel J. Evans

A dissertation presented in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy in

Bioinformatics & Computational Biomedicine

to the **School of Medicine** at

Oregon Health & Science University.

Department of Medical Informatics & Clinical Epidemiology

School of Medicine

Oregon Health & Science University

Sept, 2024

Bioinformatics & Computational Biomedicine
School of Medicine
Oregon Health & Science University

CERTIFICATE OF APPROVAL

This is to certify that the Ph.D. dissertation of
Nathaniel J. Evans
has been approved.

Shannon McWeeney
Advisor

Xubo Song
Chair

Gordon B. Mills
Committee Member

Guanming Wu
Committee Member

Michael A. Mooney
Committee Member

TABLE OF CONTENTS

To my friends, family, and advisors who have made this work possible and my life wonderful. To the countless researchers on the shoulders of whom I stand.

This work has been supported by grants from the National Cancer Institute (T32 CA106195), National Institute of Allergy and Infectious Diseases / National Library of Medicine (T15LM007088), Oregon Clinical & Translational Research Institute TL1, NCATS (TL1TR002371) and OHSU Dermatology T32 Fellowship, National Cancer Institute (T32CA106195).

List of Figures

List of Tables

Preface

When I embarked on this research journey years ago, I had high aspirations typical of many novice researchers. Over time, the realities of doctoral research required a refinement of these goals. My research, like most, has faced various challenges, each contributing to a foundation of success and achievement. Consequently, if some goals in this work seem ambitious or not fully aligned with the results, consider this dissertation not only a scholarly contribution, but also a testament to personal growth and development.

About the Author. Understanding the motivations behind this research requires some context about my academic and professional background. Initially, I studied at the University of Washington, where I explored various engineering disciplines before ultimately graduating with a B.S. in Biological Physics. Following graduation, I worked as a Mechanical Design Engineer at a bio-technology startup, VisionGate. Here, I contributed to the development of a sophisticated diagnostic device for early detection of lung cancer using 3D imaging of sputum cells. This role involved collaboration with a tight-knit team of engineers, biologists, and computer scientists. It was in this multidisciplinary setting that my passion for research truly ignited.

After several years, I decided to further my education and pursue a Ph.D. in Biomedical Informatics. Although my undergraduate studies introduced me to computer science, it was during my early professional career that I encountered machine learning and its exciting prospects. I had come to appreciate interdisciplinary work and was committed to ensuring that my future research efforts would not be conducted in domain isolation. The OHSU Biomedical Informatics program, with its emphasis on multidisciplinary collaboration and opportunities for translational impact, was an ideal fit.

From the onset of my Ph.D. at OHSU, I was driven to study cancer drug response. I was captivated by the potential of precision medicine and the use of drug combinatorics to tailor patient treatments to each unique disease. I was equally determined to leverage deep learning methods in this field, despite its relative novelty and unproven status at the time. During the next several years, I have gained a deep understanding of precision oncology, deep learning, and healthcare. It is with pride that I present this document, which I believe will contribute to health research.

Please note that in this manuscript, I commonly use "we" in place of "I" as a literary habit; however, all work described here is entirely my work, with of course review and guidance of my dissertation advisory committee.

ABSTRACT

The long-term goals of my research are to develop tools that accelerate precision oncology drug development and produce effective patient drug-response predictive models. The work presented in this dissertation focuses on the development of robust deep learning models to predict in-vitro cancer drug responses, with utility in precision oncology research tasks such as drug repurposing and prioritization of disease-specific drug combinations. This work addresses two critical shortcomings in the current approaches to predicting cancer drug responses with deep learning: 1) data quality issues inherent in high-throughput drug screening datasets by developing algorithms for detection of atypical or low-quality data and 2) improved prediction and utility of perturbation biology models by developing algorithms that operate on mechanistic prior knowledge.

1 Introduction

1.1 Motivations

Cancer is currently the second leading cause of death in the United States, responsible for more than 600,000 deaths per year [cdc]. In addition, the incidence rates of many cancers are increasing around the world, particularly in low-income countries [TSWJ16]. The development of effective, tolerable, and financially effective treatments for this disease is a top priority of healthcare research, highlighted by the National Cancer Institute (NCI) budget of 7.3 billion dollars in 2024 [cana]. Even with this focus and the recent advances in modern understanding of biology and cancer, cancer continues to be a significant challenge in human health research. The development of anti-cancer drugs, in particular, faces significant hurdles during the transition from basic scientific research to clinical application, a task sometimes referred to as crossing "the valley of death" [But08, Ada12]. This term might appear hyperbolic if not for the context of drug development, where drug failure rates can be as high as 90% [Ada12]. Furthermore, there is a serious need for novel drug therapies as a large proportion of cancer mortality is attributable to cancer drug resistance, which fundamentally undermines the current treatment paradigm.

> *"In 2017, about 1.7 million people were diagnosed with cancer and 0.6 million people died from the disease. Drug resistance and the resulting ineffectiveness of the drug treatment are responsible for up to 90% of the cancer related deaths"* [WZC19, Soc08].

To combat this disease, it is imperative that researchers develop methods that improve the generalizability of preclinical research and address the multifaceted nature of cancer and its evasion mechanisms. This requires the development of novel drug therapies that broaden the spectrum of treatment options as well as the strategic application of drugs or drug combinations designed to prevent or mitigate cancer drug resistance. In addition, the integration of precision medicine, which tailors treatments to the unique genetic makeup of individual tumors, has shown remarkable promise and is likely to play an integral role in future cancer treatment. To develop treatments that are both effective and safe requires rigorous preclinical research and subsequent clinical trials to meet FDA standards for approval. A current bottleneck is the alarmingly high rate of drug development failures. Figure 1 highlights that a small fraction of the drugs entering Phase 1 clinical trials are approved for patient use. This stark reality underscores the need for a paradigm shift toward generalizable preclinical drug-response research that will translate into patient treatment and address the on-going challenges of cancer drug resistance.

Notably, drug combinations, which often focus on repurposing previously approved drugs for use in new diseases, have a higher FDA approval rate. Drug combinations are also a potential avenue to address drug resistance by targeting multiple pathways or clonal populations, thus preventing common cancer resistance mechanisms. When combining therapies, however, there is a "combinatorial explosion" of possible therapies, a challenge that we demonstrate in Figure 2. This mathematical quality of combinatorics makes comprehensive in-vitro or in-vivo measurement of drug combinations intractable. For instance, to test all possible pairwise combinations of 500 drugs in a single cell line and

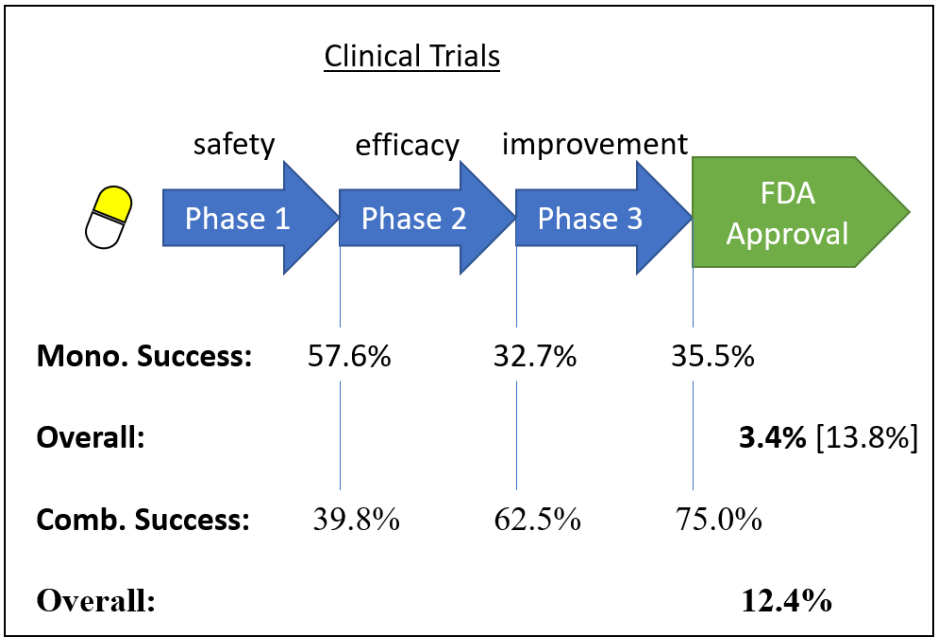


Figure 1: FDA clinical trials drug-development pipeline [WZC19, PHL+19, WSL19].

at a single dose would require approximately 100,000 assays. Given that effective screening is likely to require multiple doses (traditionally 4-7) and cellular contexts (1000+ cancer cell lines available), the number of required assays scales impractically, potentially requiring millions of experiments. If we consider higher-order drug combinations (i.e., combinations of more than two drugs), this problem is further exacerbated.

An adjacent approach that may help improve treatment effectiveness and durability[1] is the stratification of patients based on tumor molecular markers [HVSLJ13]. Functional tumor stratification can be challenging due to the innate complexity of biological systems. The high-dimensionality of potential biomarkers that can be used as predictors of sensitivity can make it challenging to find generalizable predictive models of cancer drug response.

To address the challenges of intractable combinatoric measurement and high-dimensionality of molecular features, many modern research strategies use analytical methods to infer likely drug candidates or effective drug combinations. Figure 3 highlights how predictive models can be implemented to accelerate drug development research pipelines. The development of accurate machine learning models of drug response has enabled comprehensive *in-silico* screening of drug compounds, and are intended to enable researchers to narrow in on a subset of drug candidates that are likely to be effective therapies.

1.2 Goals of this work

At its core, the objective of this dissertation is to develop novel predictive models of cancer drug response to aid and enhance precision oncology research. Successful methods may one day serve as clinical decision aids to match patients with the most effective treatments based on the specific characteristics of their disease or to accelerate pre-clincal health

[1] Durability is a measure of the long-term effectiveness of a drug, often in the context of avoiding drug resistance.

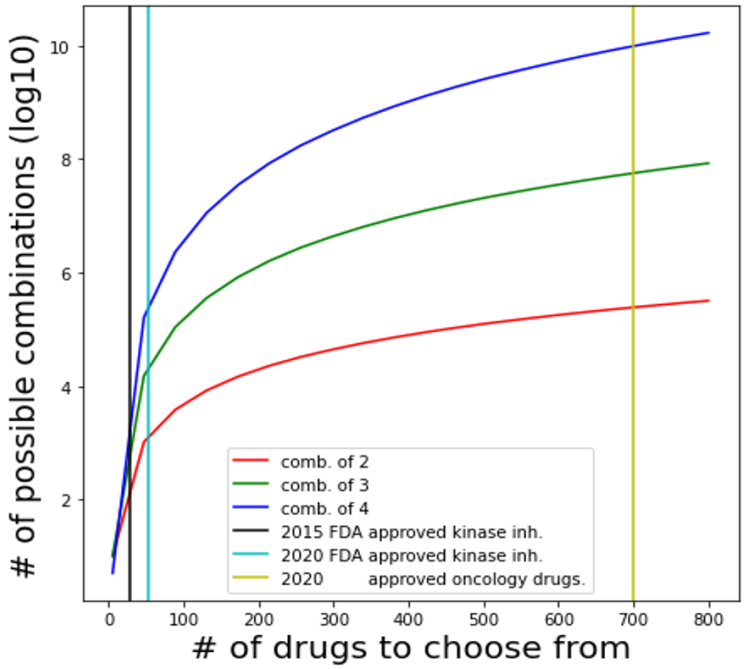


Figure 2: Challenges of drug combinatorics. The number of possible drug combinations (y-axis) given a number of drugs to choose from (x-axis). [Glo, Ros20, WNC15]

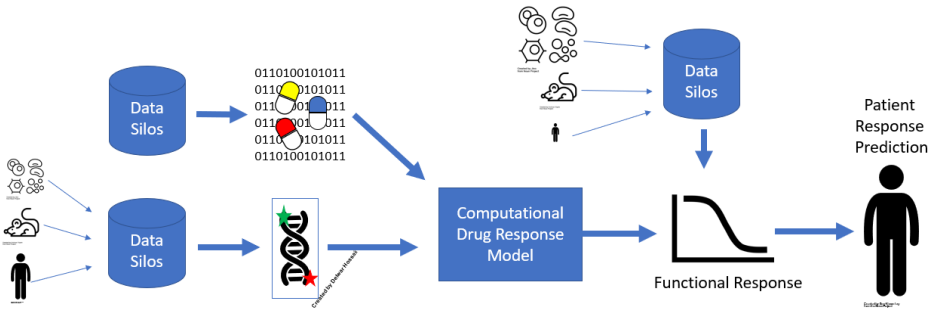


Figure 3: Common research methodologies in precision oncology. Description of the role of computational modeling in research and how it can be used to accelerate drug development and precision oncology research. Predictive models are increasingly being used for prediction of drug response, which can be used to identify molecular markers of drug sensitivity or to help guide drug combinatorial research.

research. Due to existing limitations, which will be explored throughout this dissertation, this work focuses on development of methods with application to pre-clinical precision oncology research. In this work, my aim is to improve healthcare by developing and applying methods to repurpose drugs for nuanced diseases (e.g., disease subsets, specific tumor states, specific expression/mutational patterns) using computational models that learn from large high-throughput in vitro drug screening and perturbation biology datasets. Additionally, in the pursuit of these goals, we highlighted a significant need for methods that improve data quality, in particular methods useful for the identification of atypical, spurious, or erroneous data.

1.3 Current landscape of cancer treatment

Cancer is caused by cellular dysregulation and its presentation can be exceptionally varied. In fact, cancer can be thought of as an umbrella term for many independent diseases that often behave in radically different ways due to the cellular lineage and the specific mechanism of dysregulation. Consequently, cancer treatment strategies depend on the type of cancer, the mechanism of dysregulation, and the location or presentation of the tumor. Due to this heterogeneity, treatment approaches are diverse, although common modalities include surgery, radiation therapy, chemotherapy, targeted therapy, immunotherapy, and hormone therapy. Although a complete description of the cancer treatment landscape is beyond the scope of this work, we will provide a brief overview of these treatment modalities and the importance of each.

Surgical resection is often the first course of action. This can be performed to remove tumors and relieve symptoms caused by the disease. Surgical resection assumes cancer, particularly early stage cancers, are a local disease and therefore removing them removes the disease. While this has exceptions, it has been shown to be particularly effective in some cases. For example, resection of early-stage colorectal carcinomas yields 95% patient survival after five years [WW06]. Although often effective, there are scenarios where resection is not viable due to tumor location, disease stage, or cancer type. For example, hematologic malignancies rarely form tumors and brain tumors can be difficult to remove surgically due to their location and proximity to sensitive organs. Resection of metastasized or late-stage tumors is not always feasible or beneficial for long-term survival [WW06].

Radiation therapy, or radiotherapy, is another common treatment strategy for tumors that cannot be safely resected. This approach can also be used in combination with surgery to target residual disease (i.e., cancer cells "left over" after resection) or prior to surgery to shrink the tumor. This approach uses high-energy radiation to damage the DNA of tumor cells, inhibit tumor growth, and kill tumor cells. Modern improvements in radiation therapy have improved the ability to selectively target tumors, but side effects can be debilitating and can limit its use [WW06].

Chemotherapy uses predominantly cytotoxic chemicals that interfere with the processes of rapidly dividing cells, which is a hallmark of cancer. Although effective in killing some cancers, chemotherapies can also affect normal cells and can lead to tissue toxicity or patient side effects. Chemotherapy can be used as a neoadjuvant therapy to shrink tumors before surgery or as adjuvant therapy after surgery to eliminate residual disease. Its appropriateness and effectiveness depend on the type and stage of cancer. Modern approaches often combine chemotherapy with immunotherapies or targeted therapies to increase effectiveness while minimizing patient toxicities and side effects [WW06].

Immunotherapy uses the patient's own immune system to destroy cancer cells. This strategy usually focuses on restoring immune function or improving the immune system's ability to identify and target cancer. There are many forms of immunotherapy, most of which fall into one of the following categories:

- **Checkpoint Inhibitors** help the immune system identify cancer cells. A common immune escape mechanism is the evolution or up-regulation of protein receptors that communicate with immune cells to evade destruction. Checkpoint inhibitors commonly bind to these protein receptors, inhibiting the cancer's ability to evade immune detection [ERB$^+$20].

- **CAR-T Cell Therapy** involves genetically modifying a patient's T-cells to enhance their ability to identify and destroy cancer. [SS21b]

- **Cancer Vaccines** train the immune system to recognize cancer-specific antigens, enhancing the immune response [WW06].

- **Cytokines** are proteins that modulate the human immune response. Administering certain cytokines, such as interferons and interleukins, can stimulate the patient's immune system and have shown to be effective in certain cancer types [WW06].

- **Monoclonal Antibodies** can be designed to specifically bind to cancer antigens, thereby disrupting cancer cell function. These antibodies can be used alone or be conjugated with chemotherapy drugs to increase the specific delivery of a drug to the cancer cell. Bispecific monoclonal antibodies function by binding two different proteins, which can help recruit immune cells directly to the cancer cell, improving the immune system's capacity to fight the cancer. Additionally, their combinatorial nature can increase the specificity of targeting, which can improve efficacy and reduce toxicity [WW06].

- **Oncolytic Viruses** are an emerging approach using genetically modified viruses that selectively infect and kill cancer cells while leaving normal tissue undamaged. This approach may be particularly effective in certain scenarios because the process of cancer destruction by the oncolytic virus exposes cancer antigens in a way that is easily recognizable by the immune system, potentially stimulating an immune response against the cancer [WW06].

Targeted therapies, which include some immune therapies, are a class of drugs that bind to specific molecular targets, usually proteins, and interfere with one or more cellular processes. Through a detailed understanding of the cancer's precise dysregulation, targeted therapies can be chosen that interfere with processes on which the cancer is dependent. This difference in cellular mechanisms between cancer and normal cells enables selective targeting of the cancer. Effective targeted therapy is integrally dependent on a strong understanding of cancer driver mechanisms [HW11, WW06]. For instance, growth of chronic myeloid leukemia (CML) is driven by a chromosomal translocation of chromosomes 9 and 22, referred to as the Philadelphia chromosome. This translocation results in the fusion of the BCR and ABL genes, creating the BCR-ABL protein, which is known to drive CML proliferation. The drug *Imatinib* selectively binds to BCR-ABL and inhibits kinase activity, thus inhibiting CML proliferation. Since BCR-ABL is not present in normal cells, Imatinib has a minimal effect on them, allowing its long-term use without inducing significant

side effects or organ toxicities. The use of Imatinib has transformed CML from a deadly cancer into a manageable chronic condition and is a first-line approach for CML patients [DTR$^+$01, Sac14].

1.4 Precision Oncology

Precision oncology diverges from the traditional *one size fits all* approach in cancer treatment by tailoring therapy to the specific characteristics of each disease and patient. This field exploits modern technologies for the molecular measurement of tumor features, allowing treatment decisions based on the key molecular features of the tumor itself. It goes beyond traditional cancer subtyping (e.g., primary disease, subtype, lineage, etc.) by focusing on functionally relevant tumor characteristics (for more background, see these reviews [PFB16, SLRZ17]).

Precision oncology uses molecular characteristics of the tumor to suggest the most suitable treatment plan. Unlike traditional treatment approaches, precision oncology sometimes prioritizes treatments independent of the cancer's tissue of origin, which makes some precision oncology treatments applicable to many cancer types. An example of this is the immune therapy Pembrolizumab, which was recently approved for use in microsatellite instability high (MSI-high) tumors, which is a quality that occurs in many cancer types [MLKP19]. Given the high variability in tumor drug response based on factors such as cancer type, genomics, tumor microenvironment, and immune response, precision medicine research is a highly complex and burgeoning research field. That said, there have been notable improvements in patient outcomes using this paradigm. Examples include the treatment of chronic myeloid leukemia (CML) with Imatinib [Sac14, DTR$^+$01] and HER2+ breast cancer treatment [MK17, BPPG19].

Research in precision oncology utilizes functional annotation of tumor genomes, proteomes, metabalomes, microbiomes, transcriptomes, or epigenomes. This is done by comprehensive measurement of molecular features as well as the response to a treatment. Through statistical analysis, researchers can identify molecular features that predict the tumor response. For example, there are several commercial RNA expression panels that can help guide treatment and estimate the prognosis of the patient, including MammaPrint [BPPG19] and OncotypeDX [MK17]. The pre-clinical precision oncology research pipeline often relies on large high-throughput screening assays, which typically use cancer cell lines as a model system for tumor response. Although cell line response does not always generalize to in vivo tumor response [GVG13], it is an effective way to rapidly screen thousands of drugs across many cancer contexts. Collaborative projects like the Broad Institute's Cancer Dependency Map (DepMap), which encompasses the Cancer Cell-Line Encyclopedia (CCLE) [GHJV$^+$19], Gene Dependency datasets [TVM$^+$17], and various inhibitor screenings, exemplify the integration of resources to create comprehensive databases of molecular features and functional annotations. Pooling resources in this way is likely to significantly accelerate precision oncology research.

1.5 Artificial Intelligence, Machine Learning and Deep learning

This section provides a brief introduction to Artificial Intelligence (AI) and touches on some aspects that are critical to understanding the motivations and relevance of the work presented in Chapters 2-6.

AI is a domain of science that seeks to produce machines or algorithms that mimic aspects of human intelligence and is an umbrella term for countless methods. Within AI, there is the field of machine learning (ML), which uses algorithms that can learn from data to make predictions. ML encompasses many classical data analysis methods, such as linear regression or classification, data clustering algorithms, semi-supervised learning algorithms, and reinforcement learning. The relationships of these terms are highlighted in Figure 4. Deep learning (DL) is a subset of machine learning that focuses on the use of deep artificial neural networks to model complex patterns.

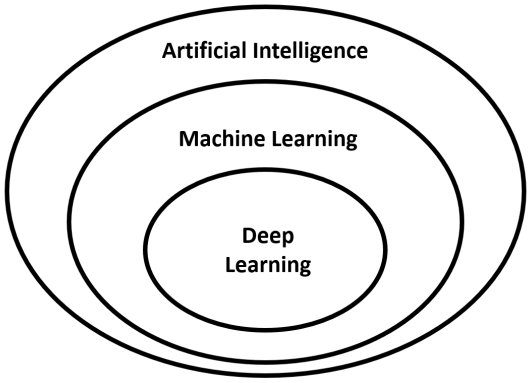
Deep learning has gained rapid use in various fields and shows no signs of waning. Although machine learning as a whole offers a myriad of applications, deep learning excels in many predictive tasks, setting the gold standard in fields such as image classification, natural language processing, and image annotation. However, certain limitations hinder the practical application of these deep learning methodologies, especially to high-consequence decision making. One notable challenge is the interpretability of deep learning predictions.

In simpler models, such as linear regression, the predictions are straightforward: they result from linear combinations of learned coefficients. This simplicity allows for an easy interpretation of why a certain prediction was made. For example, in a linear model, the predictive contribution of a feature can be easily understood as the feature weight multiplied by the feature. Deep learning, in contrast, involves multiple layers of nonlinear multivariate feature combinations and often employs complex architectures, regularization techniques, and training strategies. Although this intricacy can lead to substantial improvements in predictive accuracy, it simultaneously makes the interpretation and validation of predictions significantly more challenging.

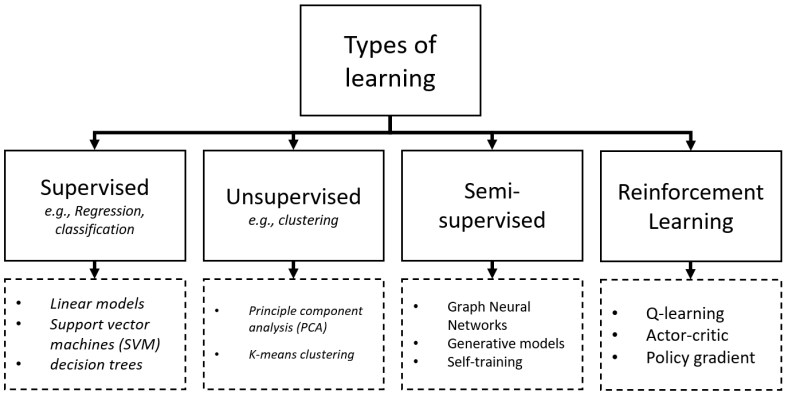
1.5.1 Bias-Variance Trade-off

An essential concept in machine learning is that of the Bias-Variance trade-off, which describes how the complexity of a predictive model can affect the performance on a specific dataset. Given a set of data $\mathcal{D} \sim (X, y)$, which was generated from an underlying function $y \sim f(x) + \epsilon$, where ϵ is noise, then the supervised learning task is to use $\mathcal{D}$ to infer a function $\hat{f}$ that approximates f. No matter what machine learning methodology is used, this process has a fundamental limitation related to the complexity of the function $\hat{f}$. For example, if we use an extremely complex model, such as a neural network with many hidden channels, then we are likely to *overfit*, which is the tendency for complex models to "memorize" the training data, but generalize to new data poorly. Alternatively, if we use a simple model such as a linear model, then it is likely to *underfit* and be unable to predict the training data well. The total error of the model can therefore be attributed to either *variance*[2] (greater in high-complexity models), *bias* (greater in low-complexity models) or irreducible error (aleatoric noise). Choosing a simple model is more likely to result in a low variance error but high bias error, and choosing complex models is more likely to result in a high variance error but low bias error. The optimal choice of model complexity, therefore, depends on the particular dataset and prediction task and should be tailored to each scenario. Figure 5 highlights this bias-variance trade-off.

[2]For instance, if we fit multiple models using unique training data sets, sampled from the same distribution, then their will be greater variance in the estimated parameters. This can be conceptualized as the model that learns the noise in a training dataset, rather than the true underlying signal.



(a) The subsets of Artificial intelligence.



(b) The different learning schemes of AI.

Figure 4: Overview figures describing the elements of AI.

In many machine learning algorithms there are hyperparameters or optimization strategies that influence model complexity, and this lets users conveniently tune the model complexity to minimize the total error. Examples of this include regularization techniques such as weight decay [Tib96, HK70], dropout [SHK+14] or model complexity hyperparameters such as the number of trainable parameters (e.g., neural network hidden channel size). For a specific learning task, the optimal model complexity can be identified using hyperparameter optimization methods where many models with varying hyperparameters are trained and the configuration with the best model is selected based on performance on a hold-out validation set [BB12]. It is also worth noting that model error due to variance has a tendency to decrease as the data volume increases. This trend is often used as justification for complex models, such as deep learning, when large datasets are available. Another interesting aspect is that variance error can also be reduced by using ensembles, for example with the random forest algorithm [Rig17], and can often significantly increase prediction performance. A key takeaway from the bias-variance trade-off is that an algorithm will not necessarily perform well on all problems and that it is critical to tune model complexity to the problem at hand.

1.5.2 No Free Lunch Theorem

The "No Free Lunch" (NFL) theorem was first introduced by Wolpert and Macready and states that:

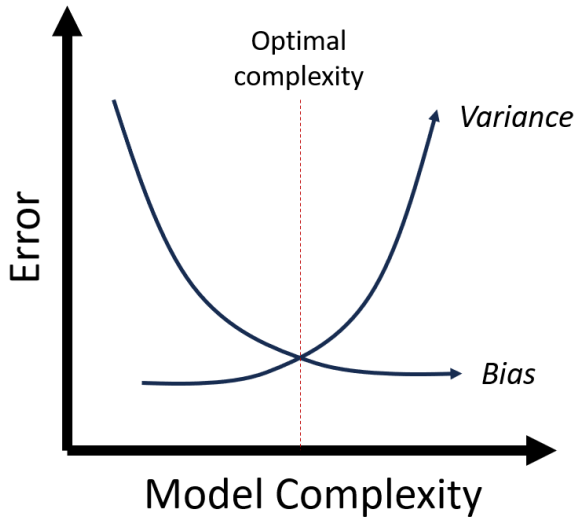


Figure 5: The bias-variance trade-off. As model complexity increases the error due to bias (i.e., under-fitting) reduces while the error due to variance increases (i.e., over-fitting). Due to this common trend in machine learning, there is an optimal model complexity that minimizes the total error.

> "Roughly speaking, we show that for both static and time dependent optimization problems, the average performance of any pair of algorithms across all possible problems is identical. This means in particular that if some algorithm a_1's performance is superior to that of another algorithm a_2 over some set of optimization problems, then the reverse must be true over the set of all other optimization problems." [WM97]

In this work, the authors seek to discuss limitations to the common task of using general optimization algorithms that do not take into account prior knowledge about the task for which they are optimizing. They argue that this is a faulty approach, as the average performance of any optimization algorithm depends on how suitable or "aligned" it is with the task it is applied to. Many researchers have used this work to justify the need to tailor an optimization problem to a given task. Intuitively, some aspects of this are standard practice in machine learning. For example, common machine learning practices include problem-specific feature selection, hyperparameter optimization, and careful choice of model algorithm or architecture.

On the other hand, the success of neural networks and common optimization strategies, notably backpropagation and stochastic gradient descent [Ama93], often leads to competitive predictions in numerous domains. Such behavior would seem to contradict the NFL theorem to some extent. An alternative explanation is that there exist algorithms whose average performance on domain-specific or real-world specific problems is notably higher than others. Such an argument was noted by the original NFL authors and would suggest that real-world problems exist in a subspace not representative of "all possible problems."

The remarkable prediction performance of recent domain-specific optimization algorithms, such as convolutional neural networks (CNNs), transformers, and physics-informed neural networks (PINNs), supports the argument of the NFL

theorem. These methods use prior knowledge of the domain and rational assumptions about the data to design learning mechanisms that improve the prediction performance.

Whether the NFL theorem has real-world utility or not, it is a useful mathematical description of an intuitive concept: that including prior knowledge and tailoring the optimization algorithm to the specific problem is liable to improve performance.

1.5.3 Hypothesis Space and constrained Hypothesis space

In statistical learning theory, the *hypothesis space* ($\mathcal{H}$) is the set of possible functions over which an optimization algorithm can search. In supervised learning tasks, the goal of optimization procedures is to select the function $f_h : X \rightarrow Y$ from the set of functions $\mathcal{H}$ that minimizes dataset loss term:

$$f_h = \text{argmin}_{\{h \in \mathcal{H}\}} \, Loss[f(X), y]$$

For example, in a simple univariate linear regression the hypothesis space is the set all functions parameterized by:

$$f(X) = wX; w \in \mathbb{R}$$

The optimization goal of a linear regression would be to select a value for w that minimizes the loss on the training dataset. The concept of a hypothesis space can be very useful when considering the challenges of over-fitting, application to complex learning tasks, and selection of suitable learning algorithms. When considering the choice between two learning algorithms, each of which having unique hypothesis spaces, it is important to ask: *which hypothesis space is more likely to contain the true hypothesis?* Over- and under-fitting can also be intuitively understood in this context. If one chooses a complex learning algorithm with a consequently large hypothesis space, then the selection of the optimal hypothesis (function) is more challenging (imagine trying to find a needle in a haystack!). On the other hand, if one chooses a simple model with a relatively small hypothesis space, then identifying the optimal function becomes much easier, but may not be accurate.

We can also think about putting constraints on the hypothesis space or by choosing algorithms with hypothesis spaces that contain the true function. For example, if we have a set of data that was generated using the function $Y = X^2 + \epsilon$ then optimizing a linear model of the form $Y = wX$ will never be able to identify the true underlying function, as $Y = X^2$ is not in the hypothesis space.

Feature engineering is another mechanism to change the hypothesis space; selection of appropriate features or removal of spurious features can help narrow the hypothesis space and is likely to improve the results of the learning task.

In deep learning, the choice of model architecture can be thought of as a selection of an appropriate hypothesis space. Convolutional neural networks (CNNs), for instance, use special architectures that apply local functions over an image to extract useful features. The hypothesis space of a CNN is likely to contain an appropriate hypothesis for many vision tasks; however, it is unlikely to be appropriate for many non-vision tasks.

1.6 Role of Artificial Intelligence (AI) in precision oncology

The role of artificial intelligence (AI) in precision oncology is nascent and still has many hurdles to overcome. Modern medicine and health research have an unprecedented ability to rapidly gather information, both in the clinic and in research settings. In many scenarios, humans are not well suited to analyze the sheer volume of data. AI, however, is ideally suited to rapid and useful data analysis. Through the use of robust and trustworthy computational models, large datasets can be effectively translated into useful knowledge that can be used to aid clinician decision-making or to empower research (for more background, see these reviews [Azu19, BSR+19, DFHM20]).

1.6.1 Role of AI in the clinic.

Making diagnoses, assigning treatment plans and predicting the prognosis of the patient are critical components in healthcare. Physicians are experts in this, but with emerging technologies such as sequencing, there are huge volumes of useful information that cannot be easily interpreted or integrated into current clinical practices. AI offers an avenue to aggregate huge volumes of data and use it to make useful predictions to aid clinician effectiveness. There are several AI implementations towards these ends with varying degrees of success [Mes17, ZGE+14, SMWB+16, dee].

These clinical AI implementations aim to aid physician decision-making in some way, and while there have been reported benefits to the use of these tools [HGGW+19], there have also been failures [FD23]. The use of AI is an attractive avenue to improve healthcare; however, the use of predictive models for decision-making is an imposing task, as there is a significant possibility of harm if done incorrectly. It is imperative that AI practitioners and clinicians develop and validate ethical, safe, and beneficial tools before we see widespread adoption and realize the full potential of the domain.

1.6.2 Role of AI in research.

One of the fundamental tasks of research is to transform our observations of the world into useful knowledge, and AI can be exceptionally useful in this role. Importantly, fallible research is usually not as consequential as healthcare decision making, and therefore the potential for harm is much lower. Effective research processes also involve multiple independent and parallel forms of validation, which further mitigates the risk that inaccuracies due to the use of AI can result in serious harm. In research, AI has become an ubiquitous tool for the analysis of large datasets. Specific to precision oncology, AI and machine learning (ML) have been used to develop countless models for drug response, prediction of patient prognosis, and inferring the complex mechanisms of cancer cell functioning.

1.6.3 Limitations of AI.

The development and effective use of AI is a nuanced and multifaceted process and there are several aspects that require careful consideration.

Appropriate data and data quality. The effectiveness and generalizability of AI and ML is critically dependent on the data used to optimize model parameters. To produce robust models, especially for deep learning applications, large diverse datasets are usually required, although some machine learning domains do focus on effective optimization on small datasets [BI21]. Even with large datasets, however, data quality issues can limit the performance and usefulness of models. In supervised learning settings, inaccurate annotations can severely degrade ML performance. Additionally, many data generation mechanisms have measurement noise, which can also hinder the optimization and performance of models. It is also critical that ML practitioners have a keen understanding of how their data is generated and the assumptions of the data so that they can choose the appropriate dataset and algorithm for a given task. Training models with inappropriate data is liable to generate models that do not generalize to actionable predictions on the target task.

Algorithmic Complexity and Interpretability. Depending on the prediction task, algorithms may require greater or less complexity. For instance, predicting a persons weight using their height is a very simple task, and simple linear models are likely to perform well. In contrast, natural language processing (NLP) is an exceptionally complex task and requires multivariate non-linear relationships between words. The appropriate choice of algorithmic complexity is critical for trustworthy performance. There is also commonly a trade-off between algorithmic complexity and model interpretability. For instance, linear models are highly interpretable in that humans can easily understand how a prediction is made and the relationships between variables. Deep learning, on the other hand, is considered a "black-box" in that the prediction logic is often incomprehensible to humans. Therefore, the choice of algorithm complexity should be balanced with the needs for model interpretation. It should also be noted that significant research, including many of the methods presented in this work, focus on producing "Explainable AI," which aims to produce interpretable models without sacrificing model complexity [MWLN22].

Integration into Clinical or Research Workflows. Development of a robust model is only the first step toward useful implementations. A key requirement is that model predictions can be integrated into clinical practice or research workflows. In clinical settings, this means seamless interactions with the Electronic Health Record (EHR), practical visualizations, useful presentation of information, and appropriate explanations or presentation of supporting evidence. Effective use also requires timely processing of real-time data, which may necessitate special computation requirements. Clinician training and the ability to customize workflows are also integral for AI to be useful in a range of settings or to different clinicians.

Regulatory and Ethical Issues. Ethical and safety concerns significantly impede the clinical adoption of AI, leading to rigorous regulatory processes overseeing its use in healthcare. Besides the risks associated with AI-driven decision-making, there are substantial security and privacy issues. AI often involves the processing of sensitive data, necessitating strict adherence to regulations to ensure patient information protection. In addition, when AI is integrated into decision-making processes, liability and accountability issues arise, which must be addressed for effective implementation. In research environments, although some of these concerns may be less pronounced, other factors come into play. For example, AI models in research can potentially misrepresent data, thus necessitating cautious

application. This concern is intimately linked to the interpretability of AI models, as conclusions might be drawn from correlative rather than causal or logical reasoning.

1.7 Cancer drug response

Cancer drug response (CDR) is an extremely active domain and one in which machine learning and artificial intelligence researchers play a significant role. The goal of CDR is to understand and ultimately predict the response of a cell line, tumor, or patient to a given drug or set of drugs. Effective CDR models have application as:

- **Basic research tools** to aid in the identification of drug-targets, elucidation of signaling patterns, identification of bio-markers for sensitivity, identification of genetic vulnerabilities in cancer and to answer many other research questions.
- **Clinical decision aids:** There are several simple drug response predictive models currently in practice as decision aids today, including MammaPrint [BPPG19] and Oncotype [MK17]. The future development of more complex models is likely to be critical to the full realization of precision medicine.

The development of effective CDR models often relies on large high-throughput cell line screening datasets to train machine learning models to predict response metrics. Traditionally, CDR has focused on the prediction of functional annotations measured by dose-response assays and quantified using IC_{50} or area under the curve (AUC) metrics. Although functional annotation is a useful summary metric for cancer cell line growth rates, it does not capture more complex aspects of the mechanism of drug response. A more holistic and information-rich drug response target is perturbed expression and is utilized in perturbation biology methods. These approaches predict how the molecular state of a cancer cell line, tumor, or patient will modulate the response to a perturbagen and can help answer research questions focused on the mechanism of response or toxicity. In particular, hybrid drug response models can predict both functional annotation and perturbed expression, and are often synergistic tasks, as perturbed expression has been shown to predict cell death and functional annotations [ZLC$^+$20b, SSH$^+$19, LCQ21].

1.8 Functional genomics datasets for drug response studies

Functional annotation of cancer cell lines or patient tumors is a common research approach in precision oncology. This is done by measuring some form of cell line or tumor functional response to a drug or genetic perturbation. In patients, the functional response of a treatment can be measured by quantifying tumor volume before and after treatment. In vivo models, such as patient-derived xenograph mice (PDX), can be measured by reduction in tumor size in response to a treatment. Experimentation in patients and mouse models, however, are not feasible for high-throughput assays due to either ethical considerations (in patients) or assay requirements (e.g., mouse models are expensive and slow). Due to this limitation of in vivo and patient models, the vast majority of functional annotation data is derived from cancer cell line models. Cancer cell lines were originally derived from patient tumors, but have since been immortalized and offer a reusable model of in vitro cancer drug response.

The cancer cell line encyclopedia (CCLE) offers aggregated resources characterizing the genomics, proteomics, transcriptomics, metabolomics, and epigenomics (jointly referred to as 'omics) of over a thousand cancer cell lines [GHJV$^+$19]. There are also several resources that measure the functional response of these cancer lines to various perturbations [CBS$^+$]. The most common form of functional annotation for cell lines is derived from cell viability, typically measured from zero (all cells dead) to one (all cells alive, relative to controls). The response of a cell line to a drug can be measured over multiple doses to attain a dose-response curve. From the dose-response curve summary metrics, such as area under-the-curve (AUC) or 50% inhibitory concentration (IC_{50}), are then used to summarize the response of cell lines to a drug or perturbation. Several large, impactful datasets of this form include:

- **Genomics of drug sensitivity in cancer (GDSC)**: This resource characterizes the cell line response in more than half a million single agent dose-response curves, which translates to sparse measurement of approximately 600 drugs in over a thousand cell lines. [YSG$^+$12]
- **Cancer therapeutics response portal (CTRP)**: This dataset characterizes the response of single agent cell line dose-response assays for almost 500 drugs in 860 different cell lines. [SLRC$^+$15, CBS$^+$]
- **National Institute of Cancer (NCI)-ALMANAC (A Large Matrix of Anti-Neoplastic Agent Combinations)**: This dataset characterizes cancer cell line response to drug combinations, offering the ability to investigate potential drug synergies [HCC$^+$17]
- **Profiling relative inhibition simultaneously in mixtures (PRISM)**: Although technically not a dose-response assay, this resource is capable of accurately estimating cell viability and producing summary metrics of functional response analogous to traditional dose-response methods. This resource characterizes the response of more than 8,000 drugs in several hundred cancer cell lines. [YMY$^+$16, CNS$^+$20]
- **DrugCombDB:** This is a resource that aggregates cancer cell line functional response to drug combinations from many resources. Currently, it characterizes almost half a million dose-response assays in numerous drugs and cell lines [LZZ$^+$20].

A recently released dataset characterized the proteomics (447 proteins) of approximately 8,000 tumor samples and around 900 cell lines with significant relevance to precision oncology research [LLM$^+$24]. This dataset expands the available molecular markers that can be analyzed for relevance in cancer mechanisms and therapeutic strategy designs.

1.9 Utility of perturbation biology datasets for elucidating drug response

Perturbation biology is a discipline within systems biology that studies how minor alterations, such as the introduction of a ligand, drug, or genetic lesion, can lead to significant functional changes in an organism [Jan03, MKW$^+$13, LCP$^+$06]. This field investigates the broad effects of such perturbations, often by comparing molecular features before and after a change to a biological system. The emergence of high-throughput sequencing technologies has been pivotal in this area, enabling the extensive measurement of various 'omics, including proteomics,

transcriptomics, metabolomics, and epigenomics. Key technologies in this domain include RNA sequencing [KM15], the L1000 assay [SNC+17a], reverse-phase protein array (RPPA), and many others [CGR+21]. These tools are crucial for measuring the systemic response of a cell line, tumor, or organism to perturbations and providing a comprehensive view of the intricate network of biological interactions. The molecular changes caused by a perturbation may be post-translational (e.g., phosphorylation or ubiquitation of a protein), epigenetic (e.g., methylation changes or chromatin remodeling) or expression changes (RNA or protein).

The response of the biological system to a perturbation begins in an unperturbed state (time=0) and changes over time, usually progressing to a new steady state. Typically, measurement assays capture these responses at a single time-point, which can make it difficult to quantify the full temporal dynamics. In some cases, multiple assays can be performed to provide measurements at multiple time points; however, this can be prohibitively expensive. Due to this limitation, it is critical to ensure that an assay's perturbation measurement time captures the appropriate temporal dynamics of the system and of the process being analyzed. The choice of measurement time can be challenging, as biological processes progress at different rates [SBOPM16]. For example, developmental biology research may require measurements over multiple days or weeks to characterize cell differentiation, whereas drug perturbation assays are commonly measured after 24 hours [SNC+17b].

1.9.1 Library of Integrated Network-Based Cellular Signatures (LINCS)

Few, if any, datasets are devoid of data quality issues, and addressing these challenges can improve the results of downstream analytics. A foundational dataset that has been highly impactful in modern research, especially in the cancer and drug development domain, is the Library of Integrated Network-Based Cellular Signatures (LINCS) project. The LINCS program has generated high-dimension transcriptomic profiles (L1000 assay; 978 *landmark* genes) characterizing the effect of chemical and genetic perturbations in a variety of cellular contexts, time points, and dosages [SNC+17b]. This data has been used successfully in many applications; however, a continued challenge with high-throughput data pipelines is the identification of low-quality samples. In 2016, a systematic quality control analysis of the LINCS L1000 data showed that differentially expressed genes (DEGs) inferred from the L1000 platform were often unreliable. For example, on average only 30% of DEGs overlapped between any two selected control viral vectors in short-hairpin RNA (shRNA) perturbations [CL]. To address these issues, many researchers have proposed methods to improve the L1000 data analysis pipeline, including alternative approaches to peak deconvolution [QLLX20, LLP17], and a novel method of aggregating bio-replicates in order to improve the noise-to-signal ratio [CHF+14, DRC+16].

A recent paper, which sought to use the LINCS L1000 dataset for the repurposing of COVID-19 drugs, proposed a simple but effective method of quantifying sample-level data quality by computing the average Pearson correlation (APC) between replicates of a perturbation. Intuitively, if replicates are discordant, and therefore have low or negative pairwise correlations, then the resulting APC value is low; however, if the replicates are concordant and have high

pairwise correlations, then the APC value is high. The authors went on to show that filtering L1000 data based on APC values could significantly improve the predictive accuracy of machine learning models [PQZ+].

Improvement of data quality in large publicly available datasets, such as the LINCS project, has the potential to significantly improve the usefulness and impact of these datasets. In addition, effective data quality metrics could be used to inform the selection of new conditions that will be most beneficial to select prediction tasks or to avoid conditions that are unlikely to be useful.

1.9.2 Cancer Perturbed Proteomics Atlas

The cancer perturbed proteomics atlas (CPPA) is a resource that characterizes the protein expression of cancer cell lines before and after chemical perturbation. This resource consists of reverse-phase protein arrays (RPPA), an antibody-based protein measurement assay. This resource characterizes $\tilde{1}$2,000 assays of cancer cell lines perturbation response to $\tilde{1}$50 drugs. Additionally, while not included in the CPPA dataset, the same authors have characterized $\tilde{8}$000 patient samples from the cancer genome atlas (TCGA) [ZLC+20a]. To our knowledge and at the time of writing this, CPPA is the only perturbation data set that characterizes protein expression (versus RNA expression). Compared to RNA-seq, RPPA is relatively low-dimensional and most of these samples have between 200-400 protein features, which vary from experiment to experiment depending on the choice of antibodies. This dataset was shown to be useful in training models predictive of drug response and capable of inferring drug-protein interactions [ZLC+20b].

1.9.3 Pan-cancer Analysis of Chemical Entity Activity

The Columbia Cancer Target Discover and Development (CTD2) center is developing a dataset that they term Pan-cancer Analysis of Chemical Entity Activity (PANACEA). This resource measures RNA-seq profiles of twenty five cancer cell lines perturbed with $\tilde{4}$00 oncology drugs. In addition, a functional annotation is provided for each of the observations in the form of dose-response assays. This dataset was released as part of a DREAM challenge, whose goal was to infer drug binding affinity [DAS+22]. This dataset characterizes $\tilde{2}$3,000 genes RNA expression, which is markedly larger than alternative bulk-RNA perturbation datasets (L1000: 978 measured genes, $\tilde{1}$2,000 inferred genes).

1.9.4 Harmonized single-cell perturbation data

Released in 2023, the **scPertub** resource aggregates >44 single cell perturbation datasets. These datasets include both scRNA-seq and scATAC-seq. Peidli et. al apply uniform pre-processing, quality control pipeline, and harmonized feature annotations to provide a broadly applicable resource for single cell perturbation biology [PGS+23].

1.10 Generalizability of cell line models to patient response

There is a known gap between basic research and clinical impact and the National Institute of Health (NIH) is working to bridge this translational gap they have called the "valley of death" [But08]. Representative of this gap is that of oncology drugs to enter clinical trials, less than 4% are approved for clinical use [WZC19]. Cell line models are

ubiquitous in precision oncology research as they allow comprehensive and reproducible measurements; however, their ubiquity may be partially responsible for the translational gap. A critical question to ask is:

Does cancer cell line drug response generalize to patient tumor response?

There are several reasons why cancer cell lines may not be representative of patient disease and treatment:

- **Appropriate choice of cell line for disease.** Intuitively, cell lines derived from a given disease are more likely to be representative of that disease. For instance, the A549 cell line, which was originally derived from non-small cell lung cancer (NSCLC), is a commonly used model for NSCLC. However, A549s are unlikely to be representative of breast cancer. There is added complexity as many of the commonly studied cancer cell lines were derived long ago and passaged many times over. This can lead to significant change in the cell line due to stochastic genetic mutations introduced at each passage. As such, cancer cell lines that were once representative of the disease from which they derived may lose validity after many passages. Some researchers have proposed that ex-vivo patient tumor cell line screening is a more effective model of individual patient response and may address many of these issues of patient generalizability [McD18].

- **Tumor microenvironment.** In a tumor, cancer cells do not exist in isolation, rather they are heterogeneous populations of normal and cancerous cells. These cells invariably have cell-cell interactions that change cellular behavior and signaling properties. Moreover, cells exist in three-dimensional structures, often adhering to the extracellular matrix. These interactions are known to play a significant role in the pathogenesis of cancer [dVJ23]. The presence of extracellular enzymes and hormones in the tumor microenvironment can change the behavior of cancer cells by binding to receptors and causing distinct signaling patterns. In the absence of these hormones, the response of a cell to a given drug may change. Recognizably, these effects could potentially be modeled by combinatorial perturbations that include both drug and hormones that may be more representative of the tumor microenvironment, however, this requires a strong understanding of the state of the microenvironment and is rarely attempted in high-throughput screens. Three-dimensional (3D) organoid cultures have been proposed to address some of these limitations of cell lines by providing elements of cell-cell interaction and 3D cell organization [GHVLJ21].

- **Immune interactions.** Immune cells and cytokines are often ubiquitous throughout the tumor microenvironment and can play a huge role in pathogenesis, tumor evolution, and cancer cell signaling. Cancer cells are likely to behave differently in the absence of an active immune system.

- **Tissue or organ toxicity.** Drug therapies are canonically considered *systemic* in that a drug therapy is expected to be present in many tissues throughout the body rather than localized in the tumor itself. This aspect means that tumor response must be considered in the holistic context of the other tissues and biological systems. If a drug causes damage to normal tissue, or to organs involved in drug metabolism, then the drug is unlikely to have clinical utility. Unfortunately, cancer cell line drug response is traditionally evaluated

independently of the normal cell response and therefore does not quantify patient or normal tissue toxicities. There has been limited work using select normal cell lines as a surrogate for tissue toxicity, however, the ability to predict toxicities from cell line response has notable challenges [SBA+18, HSZH20].

- **Absorption, distribution, metabolism, excretion, and toxicity (ADMET).** The pharmacodynamics and pharmcokinetics of a drug can be extremely complex mediated by multiple organs and systems in the body. In general, these factors describe how a drug will be distributed throughout an organism, as well as how it will be removed. In a cell line model, drug concentration is precisely controlled in an isolated system; however, in vivo, drug concentration can vary greatly depending on ADMET variables and may prevent clinical utility if effective drug dosages cannot be maintained in the tumor.

Given the limitations presented above, it may seem that cancer cell lines are a poor surrogate for tumor behavior or patient response. There are, however, aspects in which cell lines are effective cancer models. Specifically, researchers often describe aspects of cancer behavior as either:

- **Intrinsic Factors:** Characteristics that are inherent to cancer cells. These qualities are primarily determined by the genetic makeup, the epigenetic state, and the metabolic state of the cell. Intrinsic factors can be used to identify cancer vulnerabilities and cellular signaling dynamics. Notable research that has capitalized on intrinsic cancer factors includes the development of Imatinib, which is an effective treatment of chronic myleoid leukemia (CML) by targeting the BCR-ABL protein fusion resulting from the abnormality of CML called the Philadelphia chromosome [Sac14].

- **Extrinsic Factors:** Influences that are external to the cell. These factors include microenvironment interactions, immune response, and aspects of toxicity. Most of these factors are poorly modeled with in vitro cancer cell line models.

It is critical that the use of cancer cell line models is matched with appropriate research questions and that an in-depth discussion of the potential limitations is noted in any research strategy that utilizes these methods.

There has been some work toward development of machine learning models that learn from cancer cell line models with improved generalizability to patient response. These include non-linear transfer learning [MLV+21], few-shot learning [MFL+21], and careful feature selection approaches [LUK+21].

1.11 Drug-target premise of drug response

In Chapters 3-6, we adopt the premise of *drug-target perturbation response*. Within this premise, a drug molecule binds to one or more proteins and causes a conformational change that alters the protein function and leads to a cascade of physically interacting molecular elements (proteins, RNA, ligands, etc.). These signaling cascades often culminate in the activation of transcription factors (TFs). TFs regulate gene expression by binding to sequence-specific DNA segments and increasing or inhibiting transcription of the respective genes. Activation of a TF can also initiate more

complex transcriptional programs via gene regulatory networks (GRN), which comprise a web of interacting transcriptional regulators and may include microRNAs (miRNA), long non-coding RNA (lncRNA) and transcription factors. Additionally, GRNs do not necessarily require regulator-regulator interactions and may have intermediary post-translational signaling (e.g., regulator→protein-protein signaling→regulator). In summary, our *drug-target* premise envisions a complex network of physically interacting molecular entities that characterize the systemic response to a perturbation.

Notably, this premise is appropriate only for targeted therapy, and some chemotherapies, such as DNA damage drugs, do not follow this mechanism of action. In the work presented here, we focus primarily on targeted drug therapies. We discuss the limitations of this premise in more detail in Chapter 6.

1.12 Landscape of drug response prior knowledge

Drug-protein, protein-protein and gene-regulatory interactions have been studied, categorized, and made available through many public knowledge bases including STRING [SGN+20], Reactome [GJS+21], Omnipath [TKSR16], Targetome [BCKM+17] and STITCH [KvMC+07]. Cellular context, which may depend on cell type, disease, species, or genetic background, will define the subset of active interactions in a given biological system. Most database resources do not specify in which cellular context an interaction is active, and therefore reported molecular interactions should be considered as a set of *possible* interactions across all cellular contexts. The unique activity of molecular interactions within a given context, sometimes called the "edgotype" [SYT+15], can be attributed to many mechanisms. The binding affinity between a drug and a protein can vary due to gene mutations [HCM+18] and the functional effect of a drug will depend on the concentration of the target protein. Differences in protein expression between cellular contexts can mediate different responses to a drug. Precision oncology often takes advantage of these differences by developing drugs that preferentially target proteins that are overexpressed in certain cancers, such as the use of HER2 targeting tyrosine kinase inhibitors (TKI) in HER2+ breast cancer [SS21a]. The contextual "edgotype" can also be mediated by the expression of key molecular entities, and gene mutations may result in non-functional protein products. Some gene mutations can prevent or encourage specific protein-protein interactions (PPI) [SYT+15]. Differences in expression and genetic background affect cell signaling and can lead to divergent contextual responses to the same drug. The ability of a TF to regulate downstream genes (referred to as the "regulon" of a TF) depends on the expression and state of the TF and other co-regulatory proteins. Additionally, chromatin organization, methylation, and gene mutations can affect the ability of a TF to bind to its DNA targets [HB19, LJC+18, SYT+15]. Although there has been considerable research characterizing molecular interactions, many knowledge bases (including those used in this study) are not complete, and many important interactions may be missing. Drug-target interactions, for example, are notoriously sparse and many bioactive compounds do not have a known protein target: An estimated 7- 18% of FDA-approved drugs do not have known molecular targets [MVL+17]. Furthermore, there is concern of drug promiscuity, as many drugs designed for specific targets may also bind to secondary targets and cause unexpected and potentially harmful effects [GGE+21].

1.13 Landscape of cancer drug response and perturbation modeling

There have been countless models developed for the prediction of cancer drug response and perturbation biology. In this section, we provide a brief summary that describes several methods that we believe to be particularly relevant to the field in the context of our work. By no means should this be considered a comprehensive review of available CDR methods, which is unfortunately outside the scope of this introduction.

1.13.1 Deep learning

DeepCE was proposed by Pham et. al as a model for the prediction of perturbed expression in cell lines. It was trained and evaluated on the LINCS L1000 dataset. While this method was not explicitly developed for use on cancer tasks, it was trained predominately on cancer cell lines. This method embeds the STRING protein-protein interaction network using a pre-trained node2vec [GL16] model and uses a graph convolution network to embed drug fingerprints. These embeddings are then concatenated to predict expression perturbation patterns. Notably, this method uses literature curated knowledge of molecular interactions to improve the prediction of drug perturbed expression patterns. This method does not, however, model the signal transduction in the biological network itself but rather embeds STRING network and drug finger prints independently. Using latent biological network embeddings have shown to be effective strategies for improving performance, however, these approaches are not inherently interpretable and not necessarily mechanistic with respect to biological signal transduction.

Dr.VAE was a method to improve cancer drug response modeling using joint learning on perturbed expression and functional response screens [RHS$^+$19]. This method used a Variational Autoencoder (VAE) [KW22] architecture to embed and decode expression profiles (pre-treatment and perturbed). This method showed that the use of perturbed expression data could improve the predictive accuracy of functional drug response.

DrugCell is a visible neural network (VNN), which we describe in more detail in Section 1.14. DrugCell predicts drug response based on genetic background and incorporates prior knowledge constraints using the gene ontology (GO). This method showed that using prior knowledge can aid interpretation and benefit modeling performance [KPF$^+$20].

DeepCDR predicts cancer drug sensitivity using 'omic profiles and drug structure. Drug structure is embedded using a graph convolution network and 'omics are encoded as contextual biological networks. These features are then concatenated and a one-dimensional convolutional neural network predicts the response. This method showed admirable predictive performance, significantly outperforming simpler models [LHJZ20].

DeepDSC is a method that uses gene expression and the Morgan footprint of a drug (a binary encoding of a chemical's atom groups) to predict drug sensitivity using a fully connected deep neural network. A notable aspect of their model is the use of a stacked autoencoder to automatically learn relevant gene expression features [LWZ$^+$19].

MOLI: multi-omics late integration is a method that uses mutation, copy-number variation and expression to predict drug response. This method uses a novel deep learning architecture that individually encodes omic representations before concatenation and prediction of IC50. This approach out performed simpler architectures for a subset of drugs

[SNZCE19]. Notably, this method one-hot encodes drugs rather than using drug structure like other methods described in this section.

CDRScan is a deep learning method that uses cancer genomics to predict drug response (IC50). This approach markedly outperformed support vector machines and random forest models [CPY$^+$18].

Compositional Perturbation Autoencoder (CPA) is a generative model designed to predict gene expression changes under complex perturbations by modeling compositional interventions. CPA operates by learning disentangled representations of both genetic and drug perturbations, allowing the model to generalize to unseen combinations of perturbations. By leveraging variational autoencoders, CPA captures the underlying biological processes and interactions, offering a flexible framework for predicting cellular responses to perturbations. This method has shown strong performance on tasks involving multiple perturbations and demonstrated robustness in scenarios where data on combinatorial treatments is limited [LKSDD$^+$23] .

1.13.2 Network based models

DYNamics-Agnostic Network MOdels (DYNAMO). Proposed by Barbasi and Santolini in 2018, this is an ensemble of simple topology based models for the prediction of biological perturbation patterns. This work showed that knowledge of the interactome network topology offers 65–80% accuracy in predicting the impact of perturbation patterns evaluated using various biochemical reaction models. Importantly, this work showed that perturbation responses could be estimated with moderate accuracy without knowledge kinetic rate parameter [SB18]. More recently, Li et. al proposed a method that improved on the DYNAMO method by using graph convolutional networks (GCN) to improve predictions of perturbation response using biological network topology [LG19].

1.13.3 Mechanistic Models

BioChemical Reaction Networks (BRN). A mathematical representation of biological systems that comprise interactions between biochemical species and molecular entities. BRNs can be used to model the behavior of biological systems at various levels of granularity, and modeling strategies include ordinary differential equations (ODE), Boolean Networks, Markov processes, Kinetic rate laws, and agent-based systems. Many BRN methods focus on modeling detailed kinetic systems, and while some of these methods are capable of scaling to a large number of entities, many of them focus on systems with fewer than a thousand parameters [LAM19]. Some general limitations of mechanistic models, such as ODEs, are the difficulty of fitting parameters, especially as the system size scales.

In the domain of cancer drug response, several **mechanistic models** of cell signaling have been proposed to address the scalability issues of BRNs. In 2018, a mechanistic pan-cancer pathway model was introduced, which could learn the ODE parameters for a system of $\sim$100 proteins/genes and <$10e4$ reactions in approximately a week of training on a system of 400 CPUs [FKW$^+$18]. More recently, a machine learning approach termed **CellBox** was proposed to model perturbation biology using an ODE ($\sim$10,000 interactions/w_{ij}) and solved with gradient descent optimization. These methods address a limitation of many machine learning algorithms that lack the interpretability of predictions. ODE

models are both transparent (simple, known mathematical model) and traceable (outcomes can be traced along the inferred biological network) [YSL+21], and are likely to be more trustworthy and actionable than traditional machine learning methods.

PAthway Recognition Algorithm using Data Integration on Genomic Models (PARADIGM), uses factor graphs to model cell signaling [VBS+10]. More recently, **Boolean networks** have been used to create logic models of cellular signaling [TTE+17]. Some pathway knowledge bases have included modeling techniques directly into their platform, such as **ReactomeFIViz** [BMSW19] and the breadth and ubiquity of cell signaling modeling highlight the integral role that it plays in biological research.

1.14 Methodological landscape for inclusion of prior knowledge in deep learning

Geometric learning algorithms such as **graph neural networks** (GNNs) have seen remarkable development in the last few years and have applications in a myriad of fields [LXTG20, KW16, VCC+17b, PMP+21, ZSH+19, XHLJ18]. In general, these methods operate on graphs to recognize complex patterns and are commonly applied to problems in node classification, graph classification, or graph regression. Importantly, graphs can be used to encode many forms of heterogeneous prior knowledge and therefore are an attractive avenue for incorporating prior knowledge constraints into deep learning algorithms.

Structural equation models (SEM). A class of methods that focus on modeling processes with known or postulated dependencies between variables or latent variables. These methods are commonly used in the social sciences to evaluate the appropriateness of casual structure to explain or predict data [BV10]. Many SEMs assume simple linear relationships between variables and struggle to scale to a large number of variables, and this limits their application to many tasks [DYM18].

A foundational method that incorporates prior knowledge into deep learning is the **Visible Neural Network (VNN)**. VNNs use prior knowledge to constrain variable interactions and encourage the model to mimic the behavior of true biological systems [MYF+18]. The VNN does this by enforcing a parent-child relationship of Gene Ontology latent states and mapping genotype behavior onto them to predict cellular phenotype. This work showed that including prior knowledge constraints could improve the interpretability and usefulness of a model while maintaining the predictive performance of traditional deep learning. More recent work developed *DrugCell*, which uses a VNN to predict drug response and synergy [KPF+20]. A limitation of current VNN methods is that prior knowledge and interpretation are restricted to the ontology used. Specifically, DrugCell does not provide molecular-level prior knowledge and assumes hierarchical prior knowledge (i.e., requires directed acyclic graphs).

Another method that aims to incorporate prior knowledge into deep learning is **physics-informed neural networks (PINN)**. These methods use a unique loss function to enforce a set of prior knowledge constraints encoded as partial differential equations (PDEs) [RPK19]. To our knowledge, PINNs have not been applied to systems biology or drug

response modeling and tend to focus on domains where the constraints are well known and can be encoded as a system of PDEs.

Knowledge distillation was proposed to distill the knowledge and high performance of a large model, or an ensemble of models, into a smaller or single model, thus reducing inference overhead [HVD15]. A recent paper used a similar method to distill the knowledge from a slow but highly mechanistic cellular model into neural networks. By doing so, they were able to obtain accurate mechanistic-trained deep learning model, which was orders of magnitude faster at inference than the alternative mechanistic model [WFL+19]. This method may represent an alternative means of training robust mechanistic-like deep learning models while maintaining fast inference. Importantly, this approach is likely to suffer from issues of interpretability and explanability because even though the deep learning model is trained on data from a mechanistic model, it is still a "black box," which inhibits clear understanding of the internal prediction logic.

1.15 Ideal Modeling Requirements of Perturbation Biology

In this section, we highlight several (but not necessarily all) important aspects of signaling that an effective modeling strategy should be able to capture.

Molecular state. Many molecular entities, such as proteins, RNA, and DNA, have a contextual *state*, which describes its unique behavior and interactions within the system; the contextual state is likely to vary by disease, patient, and cell type. For example, a protein's state may include aspects such as expression, post-translational modifications, mutational status, or complex co-factors. An ideal algorithm should be able to infer molecular entity state from contextual features (e.g., 'omics) and appropriately mediate the prediction logic to align with true contextual signaling patterns.

Source awareness. The signaling behavior of molecular entities may vary based on the source of upstream signaling. For example, G protein-coupled receptors (GPCR) and receptor tyrosine kinases (RTK) are known to exhibit *ligand bias*, which describes unique signaling patterns dependent on the specific ligand or drug that binds to the receptor [SLR18, KPPH20]. The behavior of a protein can also depend on the source of the upstream signal and may have stimulatory or inhibitory regulators that cause distinct downstream behavior. An ideal algorithm should be able to delineate specific input signals to a molecular entity and simulate different downstream signaling based on the source.

Signal latency. Perturbation response evolves over time, and molecular relationships have different rates at which they progress. For example, we expect most post-translation modifications to be much faster than transcription and translation [SBOPM16]. The rate of PPI signaling is characterized by the association of proteins and is mediated by molecular diffusion rates and conformational kinetics [SHZ09]. The rates of specific molecular interactions are, to our knowledge, not available in any resource and therefore an ideal algorithm should be able to infer the rate of signaling processes from the training data.

Nonlinear multivariate input-output relationships. Many molecular interactions in cellular signaling are known to exhibit *amplification*, where a small input signal may lead to a large output signal, and are likely to be non-linear based

on computational models [HNR02, SC06]. Additionally, protein-protein signaling often exhibits unique behavior dependent on multiple upstream signals and can be modeled as logic gates. For example, proteins may require two or more upstream signals (AND gate) to be activated, can be activated by multiple signals (OR gate), or a protein may need the absence of an inhibitory signal (NOT gate). Temporal attenuation due to feedback loops or autoregulation can lead to nonlinear temporal relationships in signaling. These characteristics suggest that an ideal model should be able to learn multivariate and nonlinear relationships between input and output signals.

1.16 Current limitations of traditional Neural Networks applied to cellular signaling

An artificial neural network (NN) [SB58, Saz06] is a machine learning algorithm that has been applied in many domains with great success. NNs have been shown to be *universal approximators*, meaning that they can learn any continuous function if given sufficient data and resources [HSW89]. In most applications, however, machine learning has limited training data and NNs suffer from *overfitting*, which is the tendency of models to accurately predict within training data but generalize to new data poorly. There are many ways to address overfitting, such as curating larger datasets, performing data augmentation [MM22] or using regularization methods such as weight decay or dropout [SHK+14]. Despite extensive research in this area, overfitting remains a challenge in many prediction tasks. Processes that involve high-dimensional inputs or outputs, or characterize complex systems, are likely to have an immense *hypothesis space* and finding the appropriate hypothesis can be exceptionally challenging, especially in limited or noisy data settings.

The *no free lunch theorem* (NFL) states that the performance of all optimization algorithms, when averaged over all possible problems, will be equivalent and suggests that developing unique optimization algorithms tailored to specific problems is key to improving performance [WM97]. A strategy to create algorithms tailored to specific problems involves integrating inductive biases. This approach enables the model to favor certain solutions over others, regardless of the data presented. [BHB+18]. For example, when working with image data, a rational assumption is that nearby pixels are more relevant to each other, and convolutional neural networks (CNN) [AAS20] incorporate this inductive bias by applying a shared function[3] to local regions of the image, which can reduce the number of trainable parameters and improve performance. Similarly, in natural language processing (NLP), the Transformer model [VSP+23] is designed to take advantage of a shared dictionary of tokens and sequence order. Both CNNs and Transformers have been shown to significantly outperform traditional neural networks in their respective problems [VSP+23, AAS20]. Researchers can use these domain-specific characteristics to incorporate relevant assumptions about the data into learning algorithms, which markedly narrows the hypothesis space, often leading to improved performance and usefulness.

The scientific community has put great effort into developing well-supported theories of drug response and perturbation biology. Including these assumptions into a perturbation biology learning algorithm is likely to lead to far more

[3] Referred to as a kernel

mechanistic, accurate, and useful predictive models; unfortunately, traditional neural networks do not have a convenient means to incorporate this prior knowledge.

1.17 Current limitations of Graph Neural Networks applied to cellular signaling

Graph Neural Networks (GNNs) are designed to learn the relationships between graph structures and an outcome variable. For instance, the CORA citation network characterizes academic manuscripts (nodes) that cite each other (edges) and GNNs can be used to accurately infer manuscript research domains [KW16]. Foundational GNNs include the Graph Convolutional Network (GCN) [KW16], the Graph Isomorphism Network (GIN) [XHLJ18] and the Graph Attention Network (GAT) [VCC$^+$17b].

Representing a system of entities as a graph is a powerful way to include complex and heterogeneous prior knowledge into a form amenable to machine learning. Common GNN architectures, however, make nuanced and often unspoken assumptions about the behavior of their input graphs, which are not valid in some learning tasks. We summarize the common GNN assumptions as follows:

- **Edge Uniformity**: Almost all GNNs use a permutation-invariant aggregation scheme to pass information between nodes (e.g., max, mean, or sum functions), which allows for GNN convolutions to be applied to all nodes regardless of number of edges; however, these methods implicitly assume that any two edges (with the same features if applicable) are equivalent. One notable exception to this assumption is the Graph Attention Network (GAT), which uses edge-specific attention to weight edges during message aggregation [VCC$^+$17b].
- **Homophily**: Like nodes are more likely to be connected [MLST23]. Graphs in which unlike nodes are often connected are considered *heterophilous*.
- **Locality and Relational Equivalence**: The properties of a node are influenced by their K-th local neighborhood [KW16], where K is the number of layers in the GNN. A consequence of locality is *relational equivalence*, which posits that any two nodes with the same K-th order relations will have equal GNN representations.

These GNN assumptions are valid in many domains, however, some of these assumptions conflict with the perturbation biology premise that we have described above. The molecular relationships described in the scientific literature are often unannotated and highly contextual in behavior, and therefore the resulting biological graphs are likely to represent relationships that have markedly different and unknown behaviors (e.g., stimulatory vs. inhibitory edges). Given this, it is likely that GNNs that make strong assumptions of *edge uniformity* and *relational equivalence* will not be able to learn distinct edge behaviors due to the dearth of molecular interaction annotations. That said, continued scientific discovery and more detailed categorization of molecular relationships may overcome this limitation in the future and enable the construction of biological graphs suitable for GNNs. Alternatively, some GNN architectures with a weaker assumption of *edge uniformity* and *relational equivalence*, such as graph attention networks [VCC$^+$17b], may be more

appropriate for cellular signaling tasks. The development of novel GNN mechanisms, such as joint learning of edge-specific features during training, may also overcome these limitations.

Recent work has shown that traditional GNN architectures perform poorly on heterophilous graphs [MLST23]. Given our premise of cellular signaling and, assuming that we are using gene expression as the target variable, then biological graphs for this learning task are likely to be largely heterophilous. For example, many molecular interactions directly impact the gene expression (transcription factor regulation, ubiquitination/degradation, miRNA regulation, etc.) while others will indirectly influence the expression (phosphorylation, protein complexes, etc.) via latent (i.e., unmeasured) cell signaling. Regions of the biological network describing protein-protein signaling cascades are likely to be largely heterophilous (signaling cascades), while local regions of gene regulation will be highly homophilous (e.g., transcription factors, GRNs). Given these conditions, it is likely that GNN architectures that make strong assumptions of homophily are likely to perform poorly on cell signaling graphs. While there has been work to develop suitable GNNs for heterophilous graphs [ZLP$^+$22, ZSH$^+$19, ZYZ$^+$20], it remains an open challenge.

The depth of the GNN, or the number of layers in the network can also cause challenges in this domain. A GNN operates by subsequent layers of *message passing* between nodes. As a result, the information from any given node can only propagate within the K-th neighborhood defined by the number of layers (K) in the network[4]. For example, a one-layer GNN can only pass information to its immediate neighbors. Deep GNNs (many layers) tend to generate node representations that are very similar to each other, a phenomenon called *oversmoothing*, which can significantly degrade performance [CLL$^+$19]. Although many methods have been proposed to address oversmoothing [CLL$^+$19, ZA20], there is no one-size-fits-all approach and remains a limitation in many prediction tasks. In the context of cellular signaling, biological pathways often involve deep signaling cascades and complex gene regulatory networks, and this suggests that deep networks are essential to accurately model the process, and GNNs may therefore suffer from oversmoothing.

1.18 Trustworthy Artificial Intelligence

Machine learning (ML) methods are often evaluated for their ability to accurately predict the outcome variable. Even accurate models, however, may be insufficient for many use-cases if they are not trustworthy. The National Institute of Standards and Technology (NIST) defines the characteristics of *Trustworthy* Artificial Intelligence (AI) as [noab]:

- Validity and Reliability
- Safety
- Security and Resilience
- Accountability and Transparency
- Explainability and Interpretability

[4]Note: this gives rise to the assumption of *locality*

- Privacy
- Fairness with Mitigation of Harmful Bias

These characteristics are intended to provide guidelines for the development of artificial intelligence models that can be trusted to perform in a beneficial manner. Complex and powerful modeling strategies, such as deep learning, often lack aspects of trustworthiness. In particular, most deep learning algorithms are considered a "black box," which refers to the inability of humans to explain or interpret predictions. Not knowing how a prediction is made can lead to poor generalization or unexpected behavior in new settings, and making decisions based on "black box" model predictions may result in harm.

The development of reliable and high-performance drug response algorithms will pave the way for highly impactful applications in translational and research settings. Pre-clinical precision oncology research can use perturbation biology models to prioritize therapeutic drug candidates for further study. Basic science can use interpretable and explainable models to understand the complex behavior of biological systems. Future clinicians may be able to use trustworthy models of drug response to choose optimal patient treatment options based on tumor or patient characteristics.

1.19 Importance of Interpretable machine learning

The predictive prowess of deep learning (DL) has led to a rise in its popularity and use, and while DL is state-of-the-art in many prediction tasks, the actionability of predictions remains limited in risk-involved scenarios like clinical decision making. When using machine learning (ML) or DL predictions as decision aids, interpretability is an important requirement. Interpretation of DL or ML models can help build patient and physician trust in a model and ensure that predictions are robust and reliable by testing the rationality of the prediction logic. Interpretability can be used to test alternative criteria that users expect in a robust and ethical decision-making process, such as fairness or causality.

A major challenge is that deep learning, and many machine learning models, are considered "black box" models, which means the internal prediction mechanisms are exceptionally challenging to understand by human users. In this section, I review the literature for methods of interpretation of deep learning models and explore the significance of these methods toward clinical and research actionability of predictions in precision oncology. There have been many important advances toward ML interpretation; however, we will focus on a subset of these including Partial Dependence Plots (PDP), attention, local Interpretable Model-agnostic Explanation (LIME), Shapley Additive exPlanations (SHAP), and Visible Neural Networks (VNN). These methods significantly extend the DL toolkit and allow effective, but in some ways limited, interpretation of blackbox models. We find that PDPs, attention, and SHAP are useful for interpreting complex black-box DL models; however, they are unlikely to be sufficient for risk-involved scenarios such as precision oncology. Rather, structured models, such as VNNs, can greatly extend the interpretability of predictions but will require rigorous methods to validate interpretations. Lastly, I will explore an additional need for uncertainty quantification in model interpretation, which is particularly critical in risk-involved decision-making processes.

Why do we care about deep learning interpretation as long as it predicts the correct value? To answer this question, we will explore a number of predictive tasks and the proposed application or use case. In each setting, we highlight how interpretation is valuable.

Basic Research. A fundamental goal of research, especially in the modern arena of big data, is to use it to transform data into knowledge. In many scenarios, predictive models are trained on datasets without the express intent of predicting new data but rather to understand the relationships between variables in the data. This approach can be used to produce hypotheses about causal relationships. An example of this application is a recent deep-generative model that was used to infer drug-drug similarity and to infer drug protein targets, although the researchers trained their model as a perturbed expression prediction task [XDL20]. The ability to interpret a model prediction can be useful to understanding variable interactions, suggest causal mechanisms, and ultimately transforming raw data into useful knowledge.

Complex decision frameworks. Healthcare involves notoriously complex decision processes that integrate heterogeneous data sources, involves multiple decision-making agents, and has significant risk of harm . For instance, physicians routinely use many diagnostic inputs to make a decision and relies on specialized expertise to make a decision. The prevalence of the Socratic method in healthcare training exemplifies another important characteristic: rational and evidence-based decision making is often developed through discussion, which can be viewed as an interpretation or explanation of *why* a diagnosis or action. Once a decision or action has been made, the consequences can be quantified along many axes including financial, quality of care, pain, ethics, survival, etc. These consequence measures can sometimes have conflicting goals, for example, financial burden, ease of use, and quality of care. For instance, a qualitative evaluation of the integration of a machine learning system into clinical workflows found three dominant themes: perceived utility and trust, implementation of process, and workforce considerations [SLB$^+$20]. In any decision process, the usefulness of a predictive model will be highly dependent on its ability to integrate seamlessly with the current system. For this to happen, a necessary model attribute is the ability to explain *why* it made a prediction, so that the decision can be involved in physician's discourse. Additionally, in this framework, interpretation will be essential to gain the trust of the physician and the patient.

Stakeholder trust. For a predictive model in a healthcare decision-making process, there are a myriad of stakeholders who must trust the model. These stakeholders include regulatory bodies, hospital administration, physicians and patients. If just one of these stakeholders does not trust the model predictions, then adoption may be stymied. There are a number of papers that have explored the relationship with human trust in a model and the model's interpretability [TPB$^+$20]. In general, interpretability can be helpful to build human trust in the predictions and ability of a machine learning model.

Model Validation. Any clinically actionable model will require extensive verification and validation before it can be used to influence patient care. This process can be quite difficult, but model interpretation can be a very useful component of it. For example, imagine a hypothetical task to build an image classifier that distinguishes dogs from cats.

We have collected a large set of images and accurately annotated each one. We can then use this dataset to train a machine learning model. When evaluating performance on a hold-out test set we find that we can predict images near perfectly. In the classical machine learning development pipeline we may be tempted to say that we have produced an accurate and useful model that can tell the difference between dogs and cats, however, a critical evaluation is to understand the *why* of the predictions. For instance, if the model is using background colors or setting to distinguish dogs from cats (maybe dogs are usually outdoors and cats are usually indoors), then we actually have an unreliable model that is unlikely to generalize well or may fail in important edge cases. In this example, interpretation of model predictions is critical to identifying which features are most important for a prediction, and these results should align with the expectations of human experts and be critical for model validation. This concept also extends to the identification of data sources or label leakage [GNS+20]. This concept is particularly important to ensure that machine learning models use ethical, causal or rational prediction logic. In decision making settings, the absence of this trait has significant danger of harm or abuse. For instance, risk-assessment machine learning models are increasingly used for criminal justice applications; however, early implementations of these models were shown to unjustly discriminate against black defendants and provide inaccurate risk assessments in these cases [ALMK22]. This is an example of how interpretation of a model prediction is critical for ensuring ethical and rational logic. In effect, machine learning predictions for decision-making models must be held to the same standards as human logic, which cannot be evaluated without effective interpretation.

1.19.1 Interpretability vs. Explainability

While there has been significant effort toward developing methods that help humans understand machine learning models, there is some debate over the proper nomenclature to use. The use of terms *interpretable* and *explainable* are often used interchangeably when referring to the ability to understand a model's behavior; however, some researchers delineate separate meanings between these two terms, while others do not. For instance, in Christopher Molnars work on Interpretable AI he describes interpretability as the ability to "explain how [a machine learning model] came to the prediction" and uses explainable and interpretable interchangeably [Mol20]. This is in contrast to work by Cynthia Rudin, who describes *explainable* machine learning as the process of post-hoc explanation of a black box model versus development of interpretable models, which are usually "constrained in model form so that it is either useful to someone, or obeys the structural knowledge of the domain" [Rud19]. Another perspective, from the work of Doshi-Velez et al., explores the challenge of describing interpretability and simplifies the definition to "Interpretability is used to confirm other important desiderata of ML systems," for example, ensuring that model prediction logic is ethical, causal, or aligns with the human worldview [DVK17]. Amazon web services describes interpretability as the transparency of a model, or the ability of human users to understand how internal weights or prediction mechanisms produce an outcome, while explainability is the ability to explain a model (potentially a "black box") behavior in human terms using post-hoc methods [aws]. Unfortunately, these delineations are somewhat contradicted by the recent definition of "Explainable AI" which is currently used by the National Institute of Standards and Technology (NIST)

and does not explicitly delineate between interpretable and explainable. The NIST presents Explainable AI as requiring the following aspects [PHF+20]:

- deliver accompanying evidence or reasons for outcomes and processes
- provide explanations that are understandable to individual users
- provide explanations that correctly reflect the system's process for generating the output
- that a system only operates under conditions for which it was designed and when it reaches sufficient confidence in its output

To further muddy the waters, "XAI" (i.e., eXplainable AI) has also been presented as a criteria for development of "algorithms that generate inherently interpretable models versus deep learning algorithms that are complicated in structure and learning mechanisms and generate models that are inherently uninterpretable to human users" [Rai20]. This perspective aligns with Cynthia Rudin's definitions and often describes XAI as "glass boxes," or "transparent" in that they improve the ability of users to understand the prediction mechanisms of a model.

At this point, I must apologize to my readers, as I have taken what was probably an intuitive concept and made it rather unclear and nuanced. In the absence of a single clear definition for these terms, we will define our definition of these terms for use in this dissertation.

> **Interpretable** tends to emphasize the development of AI models such that the internal prediction mechanisms can be easily understood by human users. Examples of interpretable machine learning models include general linear models, CORELS [ALSA+18] and decision trees [Loh11].
>
> **Explainable** tends to represent the ability to explain complex or black-box functions in a way that is comprehensible and useful to human users. This term is commonly used with methods such as partial dependence plots, Local interpretable model-agnostic explanations, and Shapley additive values [Mol20].
>
> **Explainable AI** (XAI), not to be confused with simply **Explainable**, is the use of algorithms that produce inherently interpretable models. Examples of this XAI is Visible Neural Networks [MYF+18] and CellBox [YSL+21].

Even in the presence of these definitions, it is important to recognize that these terms have significant overlap, and therefore their nuanced meaning will depend on the context in which they are used. Independent of nomenclature, we expect both terms to adhere to the pillars presented by the NIST (above) and should be useful to human users in some way.

1.19.2 Model-agnostic interpretation methods

In these next sections, I have curated what I believe to be the most promising methods for deep learning model interpretation, particularly for use in explaining black-box deep learning models. I will discuss methodology, strengths, and weaknesses and discuss the usefulness in various precision oncology prediction tasks. This section will cover partial dependence plots, local interpretable model-agnostic explanation, and SHAP.

Partial Dependence Plots (PDP)

PDPs are intended to show the relationship of one or two characteristics with the predicted outcome of a model [Fri01, Mol20]. These plots can be useful to examine the relationship between variables and to answer questions such as: is there any relationship? is the relationship linear, monotonic, or complex? To calculate the partial dependence plots of a variable, one can marginalize all variables except the variable of interest. PDP functions can be estimated efficiently using the Monte Carlo method (right-most equation below).

$$\hat{f}_S(x_S) = E_{X_C}[\hat{f}(x_s, X_C)] = \int \hat{f}(x_S, X_C) dP(X_C) \approx \frac{1}{n} \sum_{i=1}^{n} \hat{f}(x_S, X_C^{(i)})$$

Where,

- $\hat{f}$ is the PDP function
- x_s is the feature(s) to measure partial dependence for
- X_C are other features in the model

The Monte Carlo method assumes that the variables of X_C are not correlated [Mol20]. When this assumption is violated, the approximation can become inaccurate. This method excels in showing intuitive relationships between variables; however, it is limited in that one can only visualize at most two variables at a time. Additionally, the assumption of variable independence is not always valid and can lead to inaccurate interpretations.

1.19.3 Local Interpretable Model-agnostic Explanations (LIME)

LIME was first introduced by Riberio et al. in 2016 [RSG16]. Given a black box model, f(x), and a select observation, this method attempts to explain the impact of each variable on prediction by fitting an interpretable local model to the results of f(x). This procedure first creates a synthetic dataset by generating small perturbations of a selected observation (x_i). This can be done by making small changes in x_i and then inferring $\hat{y}_i$ from $f(x_i)$. By repeating this procedure many times, one creates a synthetic dataset that represents the local space around the original observation. Using this dataset, one can train an interpretable model (linear regression, decision tree, etc.). This interpretable model can then be used to explain how each feature contributes to prediction of the observation in the black box model [Mol20].

This method is useful to explain the contribution of a variable to a prediction; however, there are a number of limitations. Notably, this is an observation specific explanation and must be re-computed for each observation that we have interest in. Additionally, several hyperparameters need to be tuned and can significantly affect the accuracy of the local model. A critical hyperparameter is the magnitude to perturb x_i and defines the size of the neighborhood of synthetic data. If the neighborhood is too large, then the local model may be incorrect due to inaccurate approximations of non-linear functions.

This method can be quite convenient, as each researcher can work with their favorite interpretable model (for instance, linear models, decision trees, or support vector machines). Doing so removes the burden of trying to use new methods or metrics for interpretation. The other side of this approach is that many of these interpretable models have their own limitations, and this stack-up of fallibility. For example, fallibility in black box model, fallibility in LIME, fallibility in the local model, or fallibility in interpretation.

Shapley Additive exPlanations (SHAP)

SHAP is a method built on Shapley values, which was introduced by Lloyd Shapley as a metric for coalition game theory in 1953 [S$^+$53, Mol20, LL17]. This algorithm is a clever way to think of model interpretation and attempts to explain a prediction for a given observation. In this game-theory framework, we treat each input variable as a player and attribute the payout (impact on the outcome) to each of the players. The payout in this system is the difference between the predicted value and the prediction mean.

$$payout = y - \hat{y}$$

The goal of this game is to explain the payout by attributing portions of it to each variable (player). Succinctly stated, "The Shapley value is the average marginal contribution of a feature value across all possible coalitions" [Mol20].

Intuitively, the Shapley value of a given variable can be thought of as the contribution of that variable to the difference in prediction value from the prediction mean. SHAP extends this, by characteristic of the additivity of Shapley values. For a specific instance, if we compute the shap values for three variables x,y, and z as $S_{x,i}, S_{y,i}$ and $S_{z,i}$ then we know that they will obey the relationship:

$$\hat{y} = y_i - S_x - S_y - S_z$$

This additive nature of Shapley values lets one attribute model predictions to a set of variables. The SHAP method has many similarities to LIME; however, SHAP has stronger mathematical foundation. Also, like LIME, this method assumes variable independence and violation can lead to invalid interpretations. SHAP values can be computationally expensive to compute, and may be intractable if there are a large number of input variables in the model.

1.19.4 Deep learning mechanisms to improve interpretability

In the following section, we will discuss several approaches that can be used to develop deep learning models that improve the interpretability. These aspects are key examples of Explainable AI (XAI).

Attention mechanisms

Attention mechanisms are specific to neural networks. This technique is a simple but clever mechanism, and has three parts:

1. A process that "reads" raw data (such as words in a sentence) and converts them into distributed representations, with a feature vector associated with each word position. 2. A list of feature vectors that store the reader's output. This step can be understood as a "memory" containing a sequence of facts, which can be retrieved later, not necessarily in the same order, without having to visit all of them. 3. A process that "exploits" the content of the memory to perform a task and having the ability put attention on the content of one memory element (or a few, with a different weight) [GBC16].

Attention has been used in numerous deep learning architectures including RNNs [BCB14], transformers [VSP+17], convolutional networks [BZV+19], and graph convolutional networks [VCC+17a] to great effect. Relevant to this topic, this technique also has application in interpretation. Attention can act as an observation-specific weighting on input or latent variables, and as such, can be used to interpret variable importance. A recent paper built a model to predict readmissions for heart failure patients using EHR data, in their model architecture they used an attention weighted encoding of the input variables and showed high performance compared to naïve models. In his model, the attention weights could be used as surrogates for observation-specific measures of variable importance.

Attention is a useful contribution to the field of interpretable machine learning, however, there are several ways that attention can misrepresent explanations. For instance, deep learning models often have complex non-linear latent variable interactions, and thus, a small attention weight does not explicitly indicate small prediction contribution. Additionally, unscaled input data may bias attention weight magnitudes, leading to misinterpretation of variable of importance. It is critical, therefore, that the use of attention mechanisms be carefully engineered for applications in interpretation.

A recent extension to attention mechanisms is *uncertainty-aware attention* [HLK+18]. This method uses attention mechanisms in a framework such that input uncertainty results in wide attention variance. This attention variance has the role of a) letting the model say "I don't know," which is a critical need in high-risk task that allow deferred prediction and b) implying uncertainty in prediction interpretation.

Visible Neural Networks (VNNs)

VNNs were introduced by Trey Ideker's lab to model cell biology and function [MYF+18]. The novelty of their method is that they use prior information, in their case Gene Ontology (GO) [ABB+00], to constrain the connections within the artificial neural network. These constraints create specific network substructures that mimic the GO

hierarchies, and model weights are optimized within these structured constraints. In their paper, they show that this method can achieve performance comparable to traditional methods, while including far fewer parameters. The authors show how using GO hierarchies to constrain a deep learning drug response model provides useful interpretations of latent activation. For a given prediction, these latent spaces can be used to explain which biological mechanisms of action are involved in a prediction. By querying the latent space activations within the constrained hierarchies, one can map the relative activation magnitude back to GO terms. For instance, a given prediction may implicate the involvement of one or multiple GO terms.

This is a classic example of XAI, where the specific deep learning architecture is tailored to useful interpretations during the inference stage. Additionally, this is an excellent example that showcases how including latent variables (GO terms) can provide more useful interpretations.

1.19.5 Closing comments on Interpretable Machine Learning

Interpretation is a critical aspect of model development and is particularly relevant for improving the actionability of predictions for use in decision-making tasks by evaluating additional desiderata (e.g., causal, rational, or ethical logic). While black box models can use post-hoc methods such as PDPs, LIME and SHAP to produce explanations, these explanations are likely to fall short if they do not produce useful or accurate explanations.

Accurate Explanations can still be useless. Useful explanation requires understanding by humans, however, many machine learning models make predictions involving hundreds, or thousands of input variables, which even an accurate explanation of the ensuing prediction logic is unlikely to be useful to human understanding. Another challenge is that explanations, even if accurate, may not properly convey model behavior or may only explain a limited aspect of model behavior. An example of this is the use of saliency maps [Mol20], which are commonly used to image models to explain where in an image (i.e., which pixels) were used to make a prediction. It has been pointed out, however, that even knowing which regions of an image is used for a prediction does not explain *how* that region is being used [Rud19].

Useful Explanations may require abstraction or sparsity. Sparse explanations, which involve only a few features, tend to be more useful to humans, as we can handle a limited number of cognitive entities at a time [Rud19]. Sparse explanations can be attained by using sparse models, but these models are not always appropriate or feasible. When dealing with high-dimensional or complex functions, sparse explanations (i.e., focusing on only the top most important features) are likely to be inaccurate in that they may no longer represent the true model behavior. Alternatively, humans often build knowledge and understanding by generating abstractions that are useful and tend to inherently induce sparsity. The Gene Ontology is a perfect example of this, where the many genes are grouped by function, location, or process. Machine learning methods that can provide explanations in terms of abstractions, such as the Visible Neural Network are likely to be more useful to human comprehension and knowledge generation.

Useful Explanations may require explicit modeling of latent variables. Many machine learning models may involve complex latent variables that are not explicitly measured in a dataset. Modeling processes without explicitly involving

latent variables may prevent useful interpretations by over-simplifying or inaccurately representing interactions. For example, models of perturbation biology often involve molecular features such as RNA expression and a perturbagen, such as the introduction of a drug or genetic change. In this process, there are a myriad of often unmeasured variables such as DNA, protein, ligands, or additional molecular entities. The absence of these attributes prevents model explanations from involving many critical biological elements and, therefore, is unlikely to provide useful or representative biological explanations. This suggests that effective model interpretations are highly contextual in that they depend on the problem or domain and that proper inclusion of latent variables or prior knowledge is likely to be a requirement of effective explainable AI.

Explanations can be inaccurate. Post-hoc explanation methods for black-box models such as SHAP, LIME, PDPs have significant limitations and in some cases may misrepresent the behavior of a model [Mol20, Rud19]. For instance, correlated features (a common occurrence in many datasets) can violate assumptions of PDPs and SHAP. Using explanations generated from local models (i.e., LIME), while useful in some cases, do not ensure accurate representation of the global model and have may cause harm if used for decision-making or model validation [Rud19]. As we noted previously, sparsity is also a useful explanation attribute, however, accurate and sparse explanations can be unattainable in many settings. Using these methodologies may force ML practitioners to compromise accuracy for sparse explanations, which devalues the explanation.

1.20 Relevance of Data Quality to machine learning

Modern research and "big data" have led to remarkable discoveries and spurred many fields toward high-throughput data collection to capitalize on emerging methods in data science, machine learning, and artificial intelligence. Scientists involved in data collection go to great efforts to generate accurate and reproducible data, however, unavoidable measurement noise, batch effects, and natural stochasticity often lead to varying data quality. Many foundational high-throughput datasets are affected by reproducibility and data quality issues, which often limit the actionable results of these studies [NHM$^+$19b, BE12, PSA11, CL].

Data quality relates to the capacity of the data to represent the underlying process. For instance, the ability of an image to capture information about a three-dimensional scene or the fidelity of the measured temperature to represent the kinetic energy of an object. Images can be distorted, for instance, by chromatic aberration or lens imperfection. The miscalibration of a thermometer may inaccurately report temperature. Data inaccuracies can occur in a myriad of ways in almost every domain. In machine learning, data quality issues in the data on which a model is optimized can lead to poor performance [CHWY14, BFI$^+$22, CZ15]. Even a small subset of inaccurate samples can significantly damage modeling performance, even if most samples are high-quality. Curating high-quality datasets (i.e., datasets devoid of inaccurate samples) can be challenging and usually requires expert knowledge of both the data generation process and the underlying process being measured.

1.20.1 Data Valuation

One approach to quantifying data quality is a class of algorithms called *data valuation*, which assigns a numerical value to each sample in a dataset that quantifies its usefulness toward a predictive task. In the right context, data valuation can effectively capture many aspects of data quality. While there are a number of published data valuation algorithms, almost all of them follow the same overarching approach, in which the user must define the:

- **Source** dataset (sometimes called the *Training* dataset): This is the dataset on which samples will be valued, i.e., quality values assigned[1].
- **Target** dataset (often called the *Validation* dataset in other data valuation methods): This dataset characterizes the task or goal of the data valuation, and the choice of alternative target datasets are liable to result in different data values[2].
- **Learning algorithm**: The choice of predictive model, e.g., Logistic regression, random forest, neural network, etc.
- **Performance metric**: The evaluation metric used to compare the learning algorithms predictions against the ground truth, e.g., Accuracy, area-under-the-receiver-operator-curve (for classification), mean-squared-error, r^2 (for regression), etc.

Provided these four user-defined elements, a Data Valuation algorithm then assigns a numerical value to each sample in the source dataset that quantifies the sample contribution to the predictive performance of the learning algorithm as evaluated on the target dataset. This method can be used in a number of ways, such as:

- **Model Enhancement**: To improve the predictive performance of a model by filtering low-quality data or identifying mis-labeled samples.
- **Attribution**: To quantify data value for monetary recompense or to quantify fair contribution, e.g., credit.
- **Domain Adaptation**: To identify samples from an alternative domain that are relevant to a given target task.
- **Efficiency**: Reduce the compute resources (run-time or memory) required to train machine learning models by selecting a subset of the most useful or contributing samples. This application is also referred to as *Instance Selection*.

Existing methods for data valuation include Leave-One-Out (LOO) [Coo], Data Shapley [GZ], and Data Valuation using Reinforcement Learning (DVRL) [YAP]. Under some conditions, DVRL has been shown to outperform both Data Shapley and LOO and has been applied to large datasets (more than 500k samples). In noisy or corrupted datasets,

[1]We use this naming convention to avoid confusion later since DVGS updates model parameters based on gradient from the "Target Dataset" rather than the "Source Dataset." The Data Shapley [GZ] and Data Valuation with Reinforcement Learning (DVRL) [YAP] would refer to this as the "Training" dataset.

[2]The Data Shapley [GZ] and Data Valuation with Reinforcement Learning (DVRL) [YAP] would refer to this as the "Validation" dataset.

these methods can be used to significantly improve machine learning prediction performance by filtering low data values prior to model training. Additionally, data values were shown to effectively quantify data quality aspects such as the amount of noise in an image or incorrect class labels [GZ] (low values are expected to correlate with high-noise or mislabeled observations). As a demonstration of these methods, a recent paper used Data Shapley to value an x-ray image dataset for the prediction of pneumonia. By removing approximately 20% of their training data with the lowest data values, the authors were able to improve the test set prediction accuracy by more than 15%. Furthermore, when the authors inspected a subset of images with the lowest data values, they found that it was significantly enriched for mislabeled images [TGY$^+$].

A key aspect of Data Shapley is the definition of *equitable data conditions* [GZ], which we summarize as:

- **Nullity**: If a sample does not affect model performance, it should have a value of zero.
- **Equivalency**: Two samples with equal contribution should have equal values.
- **Additivity**: The sum of samples data values should be equal to the data value of the grouped samples.

While these conditions are convenient descriptors of data in many settings, they are not required for most of the pragmatic tasks of data valuation. Furthermore, Data Shapley is the only data valuation method to our knowledge with theoretical justifications fulfilling these conditions. Other methods, such as DVRL, perform comparably or better in many data valuation applications, such as corrupted label identification [YAP].

1.21 Contributions and road-map of this dissertation

In this dissertation, we focus on two overarching topics: 1) Methods for identification of atypical or low-quality data, with application in automated data cleaning and ML enhancement (Chapters 2-3) and 2) Methods to incorporate prior knowledge into deep learning to encourage mechanistic prediction logic, specifically for application to cancer perturbation biology (Chapters 4-6).

Chapter 2 presents a method for identification of atypical dose-response curves, evaluated in synthetic data, and applied to high-throughput drug screens in the BeatAML dataset. **In Chapter 3**, we propose a data valuation algorithm that can effectively and scalably quantify many aspects of data quality. In **Chapter 4**, we extend previous methods for generation of synthetic data with utility in perturbation biology modeling. We then develop and evaluate a graph neural network (GNN) that uses prior knowledge of synthetic gene regulatory networks to improve predictive accuracy. In **Chapter 5**, we apply GNNs to incorporate prior knowledge of cell signaling to perturbation biology datasets. Specifically, we develop a custom GNN that uses functional interactions between molecular entities to improve the prediction of drug perturbed expression. In **Chapter 6**, we develop a novel algorithm, termed Graph Structured Neural Networks (GSNN) that also uses prior knowledge of molecular interactions to predict drug perturbed expression changes, evaluated on the LINCS L1000 dataset. We also develop an accompanying algorithm, called *GSNNExplainer*, that can be used to provide useful explanations of GSNN predictions. We apply GSNN for disease-specific drug

prioritization and evaluate using FDA drug indications. **Chapter 7** provides concluding remarks and discussion of our work.

The work in this dissertation provides several novel algorithms with utility in drug response prediction, perturbation biology, drug repurposing, and precision oncology research. We show that development of mechanistic deep learning that emulates the biological premise of cellular signaling can be an effective way to improve prediction performance and utility on a range of tasks.

2 Bayesian modeling of uncertainty in dose-response assays for specific detection of atypical data

2.1 Abstract

Motivation: High-throughput functional drug screening pipelines have driven significant progress in precision oncology, pharmacogenetics, and drug development. The large volume and velocity of data that these pipelines generate often preclude manual review and, therefore, automated quality control and filtering steps need to be in place to ensure high-quality data for downstream analytics. A key area in quality control is the identification and handling of atypical data, such as dose-response curves that display unusual behavior. Filtering atypical data can improve the utility of the data for downstream analytics.

Results: We present a flexible and specific Bayesian algorithm to detect dose-response curves that display uncharacteristic hormetic (U shaped) behavior in functional drug data at dose resolution. We evaluate the algorithm on synthetic data and apply it to a subset of the BeatAML patient ex-vivo drug screening dataset. We identify several inhibitors with a particularly high proportion of atypical curves and provide recommendations for incorporation of this work into drug screening pipelines.

2.2 Introduction

Dose-response data is a fundamental component of precision oncology research and has been paramount in the identification of genomic variants predictive of drug sensitivity. Resources and datasets such as the Genomics of Drug Sensitivity in Cancer (GDSC) provides dose-response results for many drugs tested in a range of cancer cell lines [YSG+12]. Although cell lines are useful models for preclinical research pipelines, the cell line response is not always predictive of patient response. In an attempt to develop more generalizable models of tumor sensitivity to drugs, recent research pipelines, such as the BeatAML study, apply ex vivo patient tumor samples to an inhibitor panel. The BeatAML dataset measures drug effect as cell viability at various concentrations to generate a dose-response curve [TTB+18].

Precision oncology assumes that variation in genetic backgrounds and cancer genomes leads to variation in drug response. Due to this, there exists an optimal drug therapy that can be chosen on the basis of tumor genomics or molecular state. Drug response research is often approached by measuring molecular features, such as cancer genomics, in parallel to functional assays such as dose-response screening. This combination allows researchers to identify correlations between molecular features and drug sensitivity.

Dose-response assays measure cell viability, a metric of cell growth rate or death compared to controls, over increasing concentrations of a drug. A regression is then fit to the cell-viability measurements and typically make the pharmcological assumption of monotonicity.

After fitting the dose-response regression, summary metrics such as Area Under the Curve (AUC) or half maximal inhibitor concentration (IC_{50}) are computed to quantify a dose-independent response metric. These summary metrics provide a scalar value that characterizes the response of an individual sample to a drug and can be used as a surrogate for drug sensitivity. Often times, these metrics are stratified based on the distribution of responses across tumors or cell lines. Assays in the bottom quantiles can be designated as "sensitive" compared to the rest of the cell lines, and samples with sensitivity metrics in the top quantiles are designated as "resistant". In the BeatAML study, the top 20% and bottom 20% of AUC values were used to characterize the samples as "resistant" and "sensitive" for each inhibitor [TTB+18].

During this sensitivity feature processing, quality control methods work to "clean" the data by removing any atypical dose response data, or detrimental artifacts introduced my pre-processing steps. A quality control step that has not received widespread adoption is the detection of dose response curves displaying atypical behavior, such as detection of hormetic data, which exhibit a nonmonotonic or "U" shaped response. Inhibitor dose-response relationships assume that cell viability monotonically decreases with concentration, and this assumption is vital when selecting the dose-response regression model. Although the AUC summary metric performs well under normal dose-response assumptions, if the underlying data is non-monotonic, the common pharmacological regression models will fit poorly, resulting in AUC values that do not accurately characterize the response. This is problematic, as there are often a subset of dose-response curves that violate the assumption of monotonicity, and are commonly observed in drugs exhibiting colloids, multiple targets or binding sites, and other sources of drug promiscuity [ODG+14]. Drug colloids can form above a critical aggregation concentration (CAC), causing the concentration of the effective drug (non-aggregate) to decrease, reducing cell inhibition. One literature review estimated that 37% of the measured toxicological dose-response relationships met the criteria for hormesis [CB01b, CB01a]. An added challenge is that hormesis can be attributed to both biological causes (like drug aggregates) or the result of measurement error. Distinguishing between these can be difficult, as dose-response assays are notoriously noisy and often have poor reproducibility [NHM+19b]. Hormesis can be exhibited as U or J shaped dose-response curves, which are often trivial to detect; however, more nuanced hormetic behavior such as W shaped curves or a slight increasing-tail can be more difficult to distinguish. Employing AUC for sensitivity designations in assays where normal dose-response assumptions break down has the potential to confound downstream analysis. Practically, in the presence of such atypical dose-response curves, we may make incorrect assumptions about cell line or patient tumor response to a particular drug.

Several models have previously been developed to detect hormetic data, including Bayesian methods [KBG16]. However, most of these models use concentration-dependent models and focus on J-shaped curves. Furthermore, to our knowledge, these methods have not been applied to patient-specific drug screening data.

In this chapter, we propose a flexible Bayesian approach for identifying dose-response curves displaying non-characteristic, or atypical behavior. We evaluate this approach on a synthetic dose-response dataset and a subset of the BeatAML study [TTB+18]. We identify several inhibitors that display disproportionately high atypical behavior. This work is applicable to ex-vivo and in-vitro drug screens across disease domains.

The Beat AML dataset has publicly released a high-quality dataset, which can be accessed via `http://vizome.org/aml/` or the Genomic Data Commons (GDC) [canb]. This dataset represents one of the largest available functionally annotated patient cancer datasets and has been carefully curated with numerous quality control steps. The prevalence and effect of hormesis in this dataset, however, has not been explored. Since the Tyner et al. paper was published in 2018, there have already been 1005 citations. Considering the high-frequency use of this data, and given the novelty of patient dose-response assays, we believe that exploring the prevalence of non-monotonicity in the dataset is an important task and can inform QC efforts in future studies.

Also, note that this work was performed prior to the final release of the BeatAML dataset [BLS+22], and therefore may not be up-to-date with more recent quality control pipelines or data subsets.

2.3 Methods

We propose a Bayesian method to estimate the probability of cell viability at each dose and use this to compare subsequent dose points to test for hormetic behavior. This approach assumes that typical dose-response data is monotonic decreasing and that any non-monotonic or increasing response between subsequent doses is atypical. During the development of this model, we focused on several key characteristics that guided our design:

1. Atypical dose-response data does not always take the form of U or J shaped curves, and in some cases have complex behavior (For instance W shaped). To detect this form of data as well, we wanted a flexible model with dose-level resolution.

2. In dose-response data, we are often quite data poor, such that we may have only a single technical replicate and commonly 3-7 dose measurements. This, coupled with significant assay-to-assay measurement variability indicates that we often do not have enough information to accurately classify a curve as atypical or not. Because of this, we focus on modeling the cell-viability uncertainty at each dose, and make atypical classifications only when we have high confidence. This translates to identifying curves as atypical only when we are very confident that they break monotonicity. In this way, we prioritize algorithm specificity over sensitivity. This is of utmost importance as filtering data inappropriately is detrimental.

3. For wide-spread adoption and effective implementation, we believe that simple, easily explained methods are critical. Therefore, we prioritized model simplicity and attempt to explain this process succinctly in a minimally-technical fashion. We believe that subsequent dose-dose comparison is fundamentally the simplest monotonicity test and provides flexibility in detecting atypical behavior.

2.3.1 Frequentist Perspective.

It is useful to first explain the frequentist analog of this task, and then build upon this with the Bayesian perspective. In dose-response data, each dose may have one or multiple technical replicates. If we assume that cell viability noise follows a Gaussian distribution, then we can use this to characterize the mean and variance of each dose's effect on cell

viability. This provides a Gaussian distribution that defines cell viability at each dose. Subsequent dose distributions can be compared to answer the question:

> *What is the probability that the current dose cell-viability distribution is greater than the previous dose cell-viability distribution?*

The decision criteria we will use to compare doses can be formulated by the difference of two Gaussian distributions. The current dose (Y1) subtracted by the previous dose (Y2) will result in a normal distribution that characterizes the difference in probabilities.

$$Z = Y_2 - Y_1 = N\left(\mu_z, \sigma_z^2\right)$$

Where,

$$\mu_z = E\left(Y_2\right) - E\left(Y_1\right) = \mu_2 - \mu_1$$

$$\operatorname{Var}(Z) = \operatorname{Var}\left(Y_1\right) + \operatorname{Var}\left(Y_2\right) - 2\operatorname{Cov}\left(Y_1, Y_2\right)$$

For this model, we assume that each dose is independent from the previous dose, which we believe is a valid approximation considering each dose is cultured in separate wells. In this way, the covariance term goes to zero and we get the scale of Z as:

$$\sigma_z^2 = \sigma_2^2 + \sigma_1^2$$

Using Z, we can assign a probability of typical behavior (decreasing or unchanged) by:

$$P\left(H_0\right) = P(Z \leq 0) = \int_{-\infty}^{0} N\left(\mu_z, \sigma_z\right) dx = cdf(Z, 0)$$

Which we then use in our decision criteria such that:

typical if $P($ Typical $) > 0.5$ atypical if $P($ Typical $) < 0.5$

The frequentist method of estimating mean and variance from replicates provides a simple and easily interpretable method of modeling the uncertainty of cell viability. However, there are a number of ways in which this method is inadequate. First, in the case where an assay is made up of only one technical replicate (a common occurrence in high-throughput screens) the variance would take on an infinite value, making any subsequent dose's positive difference in mean, no matter how small, appear as hormetic with high probability. Additionally, even in the case of two to five replicates, it can be difficult to model variance from such a small number of points and often leads to inaccurate estimations of variance. With these issues in mind, we use the frequentist method as a base-line approach and show how incorporating domain knowledge and patient controls variance as priors in a Bayesian hierarchical model can improve parameter inference and thereby improve detection of atypical data.

2.3.2 Bayesian Hierarchical Model.

To improve the modeling of cell viability, we employ a Bayesian hierarchical model, with three avenues of improvement. First, while many dose-response assays have very few technical replicates, each patient or cell line tested

commonly has many control replicates (replicates tested with no drug). Under the assumption that the biological and technical variance is comparable between the control and the drug replicates, we can use the controls data to improve the estimation of the variance of an assay. Second, within an assay, we expect each dose to have similar variance, and therefore, by modeling variance hierarchically, variance information is shared across doses. Lastly, by incorporating priors into the model, domain knowledge can be leveraged to further improve our predictions.

To model this, we consider a single dose-response plot, which characterizes a patient or cell line response to a given drug. We make the assumption that the underlying cell viability (Y) distribution at each dose is Gaussian. Our model assumes seven doses (t) and between one and five replicates (k) for each dose, with the goal to infer:

$$P(\theta_t|Y_t, Y_0)$$

Where,

$$t \in [1:7] \text{ and } \mathrm{k} \in [1:(1-5)]$$

We can then use these models to compare subsequent dose distributions to test for increasing doses by the method described above. We assume that cell viability is distributed according to a normal distribution with unknown mean μ and standard deviation, σ^2. Our goal is to model:

$$Y_t \sim N\left(\mu_t, \sigma^2\right)$$

We choose the prior distribution on μ to be uniform between 0 and 1 because we do not have prior information and therefore want an uninformative prior.

$$\mu_t \sim U(0,1)$$

To incorporate information from the control replicates, we use:

$$\sigma_t^2 \mid \sigma_c^2 \sim \mathrm{Inv-Gamma}\left(\xi+1, \sigma_c^2\xi\right)$$

The controls are modeled with a normal-inverse-gamma prior and normal likelihood. Such that,

$$\sigma_c^2 \sim \mathrm{Gamma}\left(\frac{n_0}{2}+\alpha_0^2, \quad \beta_0+\frac{1}{2}\left(n_0 s_0+\frac{\lambda_0 n_0\left(\overline{y_0}-\mu_0\right)^2}{\lambda_0+n}\right)\right)$$

Where [default values],

$n_0 \sim$ number of controls for a given patient [21]

$\mu_0 \sim$ controls mean prior [1]

$\alpha_0 \sim$ controls precision shape parameter, [1]

$\beta_0 \sim$ controls precision rate parameter [0.2]

$\xi \sim$ dose precision shape parameter; weighted importance of controls information in dose precision [10]

$\lambda_0 \sim$ controls mean prior sample size; weighted importance of $\mu_0[1]$

To obtain our posterior probabilities, we use Markov Chain Monte Carlo (MCMC) to fit model parameters. 150 steps are used to fit each model, which was determined as sufficient for all parameters to converge, estimated by the potential scale reduction factor $\hat{r} = 10.02$. [GR92].

Last, in order to determine the probability that a dose is hormetic:

For S times:

1. Sample $\boldsymbol{\mu_t}, \boldsymbol{\tau_t}$ and $\boldsymbol{\mu_{t-1}}, \boldsymbol{\tau_{t-1}}$ from the posterior probability
2. Calculate and record $\boldsymbol{P_s}\left(\boldsymbol{H_0}\right)$
3. $P(\text{ Hermetic }) = \frac{1}{s}\sum_{s=0}^{S}\left(1 \text{ if } P_s\left(H_0\right) > 0.5\right)$
4. Classification Hermetic $= P(\text{ Hermetic }) > 0.7$

This method is slightly different from our frequentist method of classification, since our Bayesian method allows us to propagate uncertainty and only make decisions which the model has high confidence in. The threshold for classifying probabilities above 0.7 as hormetic was chosen manually based on the specificity and sensitivity optimizations.

2.3.3 Synthetic Dose-Response Data

Models that seek to evaluate dose-response curves are inherently challenging to validate for several reasons. Ex vivo drug screens serve as an approximation of how a patient tumor may respond to a particular inhibitor. Notably, there does not exist a ground-truth dose response curve for each patient sample-drug pair. However, we traditionally expect that typical dose-response data will exhibit decreasing monotonicity.

To test our model, we first create a synthetic dose-response data set. Our generator assumes that the dose-response curves start high, with cell viability at one, and obeys a logistic curve during the inhibitory period, after which there is a refractory or hormetic region wherein the curve shape is assumed to be linearly increasing. The parameters for the inhibitory and hormetic models are uniformly sampled from a user-defined range.

Table 1: Synthetic dose-response data parameter ranges. This table describes the parameter values for the 18,427 synthetic dose-response assays generated.

Parameter	Description	Value Range
nrepl	number of replicates	(1,5)
std	noise scale	[0, 0.7)
β_0	(inhibitory) Logistic slope	(-10,-6)
β_1	(inhibitory) Logistic intercept	(-10, -5)
m	(Refractory) slope	(0, 0.5)
b	(Refractory) intercept	(-0.49, 1.4)
t	Transition point	(0.1, 10)

Inhibitory region model: Logistic decreasing:

$$f_i(c) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 c)}}$$

where c is the drug concentration, β_0 is the intercept parameter, and β_1 is the slope parameter.

Hermetic region model: linear increasing

$$f_h(c) = \begin{cases} mc + b, & f_h(c) < 1 \\ 1, & f_h \geq 1 \end{cases}$$

where,

- m is slope (hermetic: $m > 0$)
- b is y-intercept

We expect there to be both biological and measurement noise, which we will model as a Gaussian distribution. The overall piecewise probability function is:

$$y_{\text{synthetic}} = N(\mu, \sigma)$$

where σ is user defined and

$$\mu = \begin{cases} f_i(c), & c \leq t \\ f_h(c), & c > t \end{cases}$$

Where *t* is the transition point. We also expect our underlying model to be continuous and therefore:

$$f_i(t) = f_h(t)$$

At this point, we can draw any number of replicates from our model at discrete concentrations. Here, we select concentrations according to those used in the ex vivo drug screen in the Beat AML study.

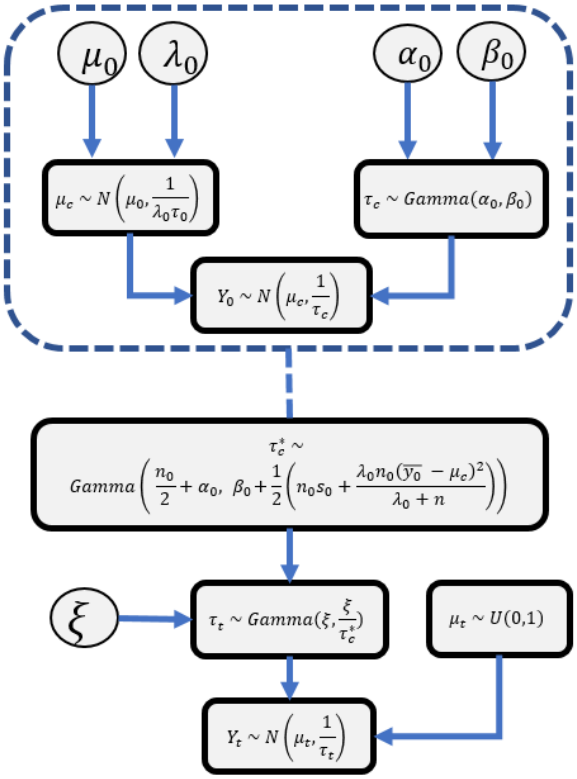


Figure 6: Graphical representation of our Bayesian cell-viability model. This graph shows the information flow in our model. The dashed box represents the model structure from which the controls posterior precision is derived.

2.4 Results

We validate this algorithm by applying it to classification of non-characteristic dose-response data in a synthetic dataset, and show that dose-response parameters are accurately modeled. To show the value of our Bayesian method, we compare our algorithm performance with the analogous frequentist method and show that our method has a greater area under the receiver operating curve (AUROC). Additionally, we show that our algorithm is highly specific and robust to synthetic assay parameters. Next, we apply our algorithm to patient oncology data, using data from the BeatAML study, and show the results of several drugs previously reported to have critical aggregation concentrations (CAC). Finally, we report several drugs with comparatively high atypical behavior that may suggest unreported CACs, or potential drug promiscuity.

2.4.1 Synthetic Data.

A synthetic dataset was created such that the underlying model followed a decreasing logistic inhibitory region and an increasing linear hormetic region. Replicates were sampled from Gaussian distributions at each dose, where the underlying model value at dose concentration is mean and a user-defined Gaussian noise. Model parameters were sampled from ranges commonly seen in dose-response data. To emulate control data, 21 replicates were sampled from

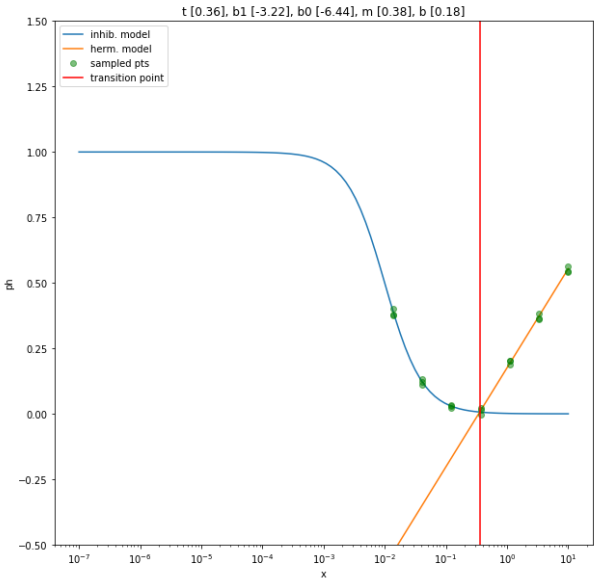


Figure 7: Synthetic dose-response data was generated to test algorithm performance. Model parameters are randomly sampled from a user defined range. For our dataset, we chose parameter ranges common in inhibitor dose-response datasets. The underlying model is defined piecewise by an inhibitory (decreasing) logistic model followed by a hermetic (increasing) linear model. The transition point, t, ranged between the min and max allowed dose, which mirrored the concentration range most commonly used in the beatAML study. Variable number of replicates were sampled at each concentration with Gaussian noise, which ranged from standard deviation of 0-0.7. In this assay, the transition point occurred at 0.44 μM, (red line) and noise scale of 0.2.

a Gaussian centered near 1, with variance equal to variance used to sample dose-replicates. It should be noted that due to sampling noise and certain combinations of synthetic model parameters, not all synthetic dose-response curves can be accurately classified using the sampled replicates, even by manual review. Rather, general trends and comparative performance are of importance in this validation.

2.4.2 Dose-response parameter modeling.

This algorithm is fundamentally based on accurately modeling each underlying dose mean (μ) and precision ($\tau = \frac{1}{\sigma^2}$). From the results in Figure 8, it is clear that the Bayesian method more accurately infers the mean and precision of the synthetic cell viability data, compared to the frequentist method. In particular, the Bayesian method has an increased ability to infer the variance given fewer technical replicates.

2.4.3 Classification performance.

To compare the performance of the algorithm for the detection of atypical dose-responses, we evaluate using the area under the receiver operating curve (AUROC) in both the frequentist and Bayesian methods. Compared across the entire simulated dataset, the Bayesian hormetic classifier had an AUROC of 0.75, and the frequentist hormetic model had an AUROC of 0.72, indicating the Bayesian approach had a small gain in performance over the frequentist approach. In

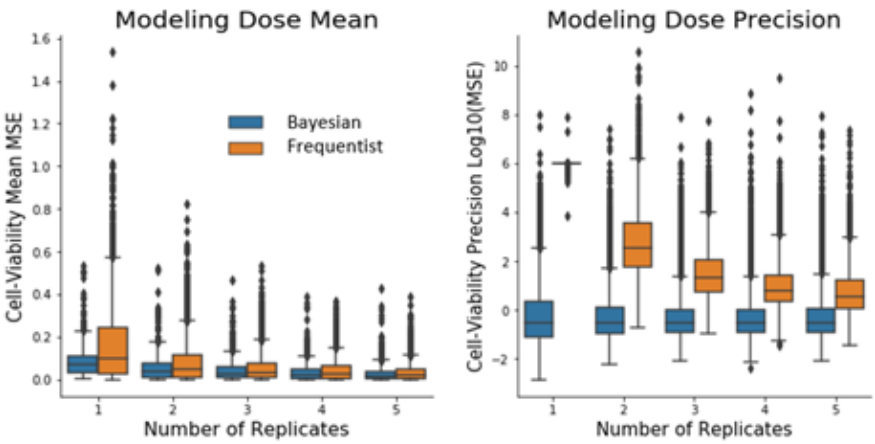


Figure 8: Algorithm performance compared to the frequentist analog on synthetic data. Mean squared error (MSE) is plotted against number of sampled replicates. Bayesian modeled dose-mean is more robust to noise, independent of the number of replicates. Bayesian modeled dose precision is far more accurate than the frequentist method. Note that the precision of the frequentist method when given one replicate was assigned a value of 1000, as a stand-in to infinity

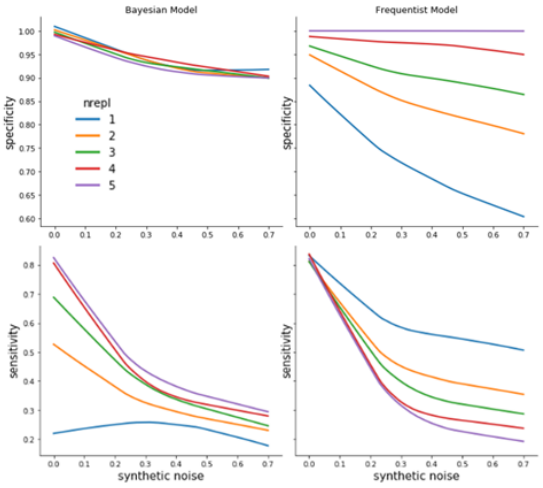
Table 2: Prevalence of assays with different number of technical replicates. This table describes the proportion of the beatAML dataset has 1,2 or 3 replicates per assay. An assay is defined here as a patient-inhibitor unique pair and does not include control data.

# Replicates	BeatAML counts	Proportion
1	253,656	0.86
2	31,117	0.11
3	9,614	0.03

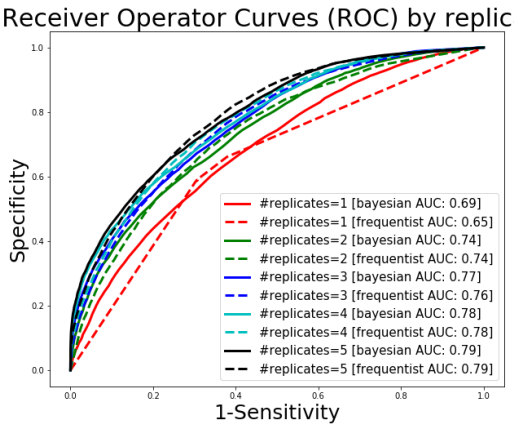
Figure 9 we plot the performance comparisons annotated by number of technical replicates. When we break the AUROC out by number of replicates in a given assay, we see that most of the difference in AUROC between the two models is attributable to observations with one replicate. The improved performance of the Bayesian approach at 1 replicate is highly desirable given the limited number of replicates often found in real world drug screening datasets. The synthetic dataset we are testing on uses a wide range of parameters, and, depending on these parameters, there may not be enough signal to classify a point as hormetic or not, even by humans. This is an issue of simply not enough, or too variable, of data at each dose and is a problem seen in real data as well; it can be exceedingly difficult, even by manual annotation, to distinguish measurement noise from hormetic behavior in dose-response plots. Therefore, the ultimate goal when designing this hormetic classifier was to model this uncertainty so as to be confident in positive classifications. These assays are expensive and analytically valuable; therefore, only assays that we are confident are atypical and problematic should be flagged. Furthermore, it should be noted that much of the available dose-response data have only one technical replicate, so an effective model should perform well with only a single replicate. To illustrate this point, we quantify the proportions of the beatAML dataset by the number of technical replicates in Table 2.

We chose to use a threshold of 0.7 to assign atypical classifications. When applied to the full synthetic dataset, this resulted in the Bayesian algorithm sensitivity of 0.32 and specificity of 0.94. We chose this threshold to prioritize the

specificity of our model, since we know that not all atypical observations can be rationally detected; In both synthetic and real dose-response assays, there is often not enough information to make an informed decision.



(a) Precision and Sensitivity curves of atypical identification in synthetic data For each synthetic dose-response assay, the hermetic classification specificity was plotted against the Gaussian noise used when sampling replicates from the assay. The Bayesian algorithm maintains similar specificity independent of the number of replicates in an assay, indicating that it accurately models the hermetic uncertainty.



(b) Area under the receiver operator curves (AUROC). The receiver operator curves for the Bayesian and frequentist algorithms were grouped by number of replicates and plotted here. We can see that when there are fewer replicates available, the Bayesian algorithm has a superior area under the receiver operating curve (AUROC).

Figure 9: Atypical dose-response algorithm performance evaluated on synthetic data.

2.4.4 Atypical oncology inhibitors.

To explore the value of our algorithm when applied to real dose-response data, we evaluate atypical responses in the BeatAML dataset. Using our hormetic detection algorithm, we tested 34,874 assays, defined as a unique patient-inhibitor pair, from 62 inhibitors.

Dose-response assay inspection. Manual inspection of a subset of the tested assays agreed with our interpretation of hermetic and non-hermetic behavior. In Figure 10 we plot four randomly sampled dose-response curves and visualize the inferred atypical probability of our method. Notably, our method does well to identify large atypical differences between doses but may miss small trends in atypical behavior, as shown in the upper right plot of Figure 10.

Inhibitor atypical trends. There is no feasible way to annotate dose-response assays with ground-truth hormetic behavior; however, many inhibitors have been found to exhibit aggregations or colloids over a threshold concentration, termed critical aggregation concentration (CAC). Drug colloids can modify the drug-receptor interaction or the drug-delivery mechanism. This behavior often causes a refractory response in dose-response assays. To explore whether our algorithm can identify the signal indicative of this colloidal behavior, we grouped all the assays tested by inhibitor, then calculated the proportion of atypical data at each dose. We chose three drugs (sorafenib, crizotinib and

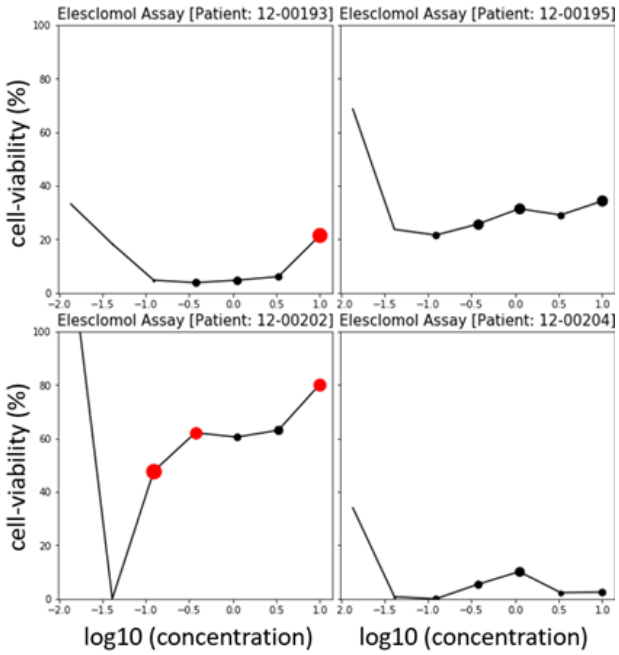


Figure 10: Randomly chosen Elesclomol dose-response plots of atypical identification predictions. Marker size correlates with algorithm predicted probability atypical; red markers indicate the probability is greater than 0.7, and therefore deemed atypical, black indicates typical.

lapatinib) that have previously reported CACs [ODG+14] and were present in BeatAML and compared our predicted atypical results with the expected refractory doses. The expected refractory dose information is reported in Table 3.

Table 3: This table lists the inhibitors from the beatAML study that have previously reported critical aggregation concentrations. It also lists the dose in beatAML assay at which we expect to first see hermetic behavior based on the given CACs. The max dose measured in beatAML assays was 10uM, therefore Crizotinib is expected not to have any hermetic behavior.

Inhibitor	CAC (uM)	Expected Refractory Dose (uM)
Sorafenib	3.5	7 [10]
Crizotinib	19.3	None
Lapatinib	0.6	5 [1.11]

Figure 11 reports the proportion of atypical doses for the three drugs with known CAC values as well as for Elesclomol, a drug with notably high atypical behavior. Our results for these three CAC annotated inhibitors subjectively match the expected refractory dose; for example, Lapatinib has an increase in the proportion of atypical behavior at dose 5, which is the first dose greater than Lapatinib's CAC value of 0.6uM. Sorafenib has an increased proportion of atypical behavior at dose 7, which is the first dose greater than Sorafenibs CAC value of 3.5 uM. Recognizably, Crizotinib's CAC value is larger than the maximum concentration tested in this dataset and therefore we cannot confirm an increase in atypical behavior.

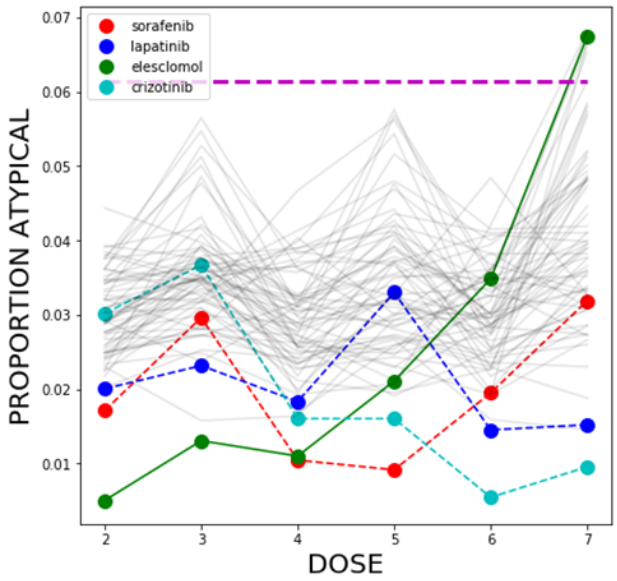


Figure 11: Inhibitor grouped atypical dose-response trends. Each line corresponds to a unique inhibitor; Model predicted probability that a given dose is atypical was threshold of 0.7 to classify as atypical and then summed at each dose, which provides a proportion of atypical doses for a given inhibitor at each dose. The dashed lines are three drugs with previously reported critical aggregation concentrations (CAC). Sorafenib CAC of 3.5uM, which is between dose's 6 [1 uM] and 7 [10 uM]. Lapatinib CAC of 0.6, which is between dose 4 [0.37uM] and dose 5 [1.11uM]. Crizotinib CAC of 19.3 uM is after dose 7 [10 uM]. The solid line is an example of an extremely atypical inhibitor, elesclomol, that exhibits atypical behavior between doses 5-7. The dashed purple line indicates three standard deviations greater than the mean atypical proportion across all inhibitors.

Table 4: Inhibitors from the Beat AML dataset with comparatively high atypical dose-response curve trends. This table lists the inhibitors from the beatAML study that exhibit atypical doses three standard deviations greater than the average proportion of atypical occurrence at each dose.

Inhibitor	Dose	[prev.] Conc. (uM)	atypical (%) [n]
CX5461	7	[3.3] 10	6.67 [104]
X-367	7	[3.3] 10	6.53 [78]
Tandutinib (MLN518)	7	[3.3] 10	6.6 [311]
JAK Inhibitor I	7	[3.3] 10	6.68 [830]
Elesclomol	7	[3.3] 10	6.67 [294]
Quizartinib (AC220)	7	[3.3] 10	6.68 [1168]

To identify outliers that exhibit a very high prevalence of atypical behavior, we compare all inhibitors and subset by three standard deviations above the average atypical proportion, which falls just above 6%. This means that for a given inhibitor, at least 6% of the assays were predicted to be atypical at a given dose. Using this method, we identify six inhibitors, reported in Table 4 all of which display a high proportion of hormetic behavior at dose 7. This may be indicative of an unreported CAC between dose 6 [3.33uM] and dose 7 [10uM]. Alternatively, there may be a more complex mechanism of inhibitor promiscuity that would explain this result, such as allosteric receptor interactions, or multiple receptor binding sites.

Atypical impact on drug response sensitivity metrics. To show that atypical data may confound downstream analytics we examine the relationship between atypical data and AUC values. Fitting a regression with the exogenous variable as the number of doses predicted atypical [1-7] and the endogenous variable as AUC showed a statistically significant trend (p-value of 9.7e-12) that AUC increases (m=2.7 AUC/number-atypical-doses). This is further visualized in Figure 12 comparing mean AUC increases with number of atypical doses in an assay. These results suggest that assays with atypical or hormetic behavior are interpreted more commonly as resistant. In downstream analytics to predict assay sensitivity (based on AUC) from clinical or genomic characteristics, results can be biased so that features associated with atypical data are used to predict inhibitor resistance.

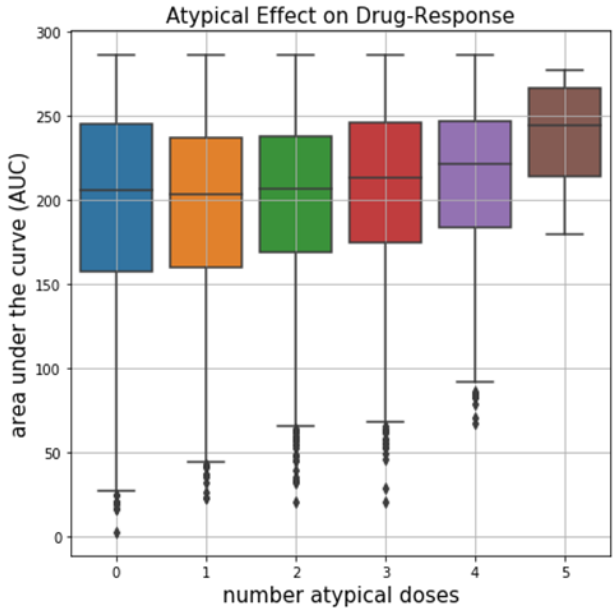


Figure 12: Dose-response assays classified with atypical doses have higher AUC values. For a subset of the beatAML dataset, all the assays which were tested were aggregated by number of atypical dose points and compared to each assays area under the dose-response curve, a metric of drug response. To further test if number of atypical doses in an assay correlated with higher AUC values, we ran a linear regression and found a slope coefficient of 2.7 AUC/atyp-doses and p-value of 9.7e-12. This suggests that atypical data does affect sensitivity metrics and may be confounding down-stream analytics.

The BeatAML drug screening pipeline already includes rigorous quality analysis and quality control steps. For assessment of drug-response curves, model fit for each curve is assessed, and any dose-response curves deviating from this fit are flagged, manually reviewed, and removed as necessary. To check if our algorithm overlaps with this already in-place QC step, we pulled all patient ID's from assays that had an AIC > 12 or deviance > 2. We then compared to see if any of the assays flagged by the BeatAML QC were also classified as atypical by our method. In this BeatAML data release, there were 73,191 unique patient-inhibitor assays, of these, 5,173 of them had either an AIC > 12 or a deviance > 2. Out of these 5,173 flagged assays, 1345 of them had been tested using our algorithm. Of this subset that we analyzed here, 81% of the assays previously flagged by the goodness-of-fit criteria (Beat AML QC) were also identified as having at least one atypical dose by our algorithm. When you compare this to the subset of assays not flagged by the

beatAML QC criteria, we see that only 62% of the assays had at least one atypical dose. This result suggests that the samples flagged by the BeatAML QC are enriched for atypical "hormetic" dose-response curves.

2.5 Discussion

In this work, we have presented a Bayesian approach to identify dose-response curves that display atypical behavior, specifically hormetic or "U" shaped responses. This work is intended to be used as an automated quality control step that can identify potentially erroneous dose-response curves, which can then be flagged for manual review, removed as "low-quality" or separated for alternative analysis. We anticipate the use of this work within larger precision oncology pipelines, such as high-throughput drug screening pipelines.

In the past, drug screening resources have suffered from reproducibility concerns, and efforts to address this will improve the utility of research resources. This can be done by careful experimental design, such as an appropriate number of technical replicates, quality control methods, and documented protocols. Employing methods, including the method presented in this paper, to detect atypical data may be useful as a quality control step, and future work should be devoted to exploring its application to improve data quality, reproducibility, and between-dataset agreement.

In an attempt to advance the field of precision medicine by improving QC algorithms, we have made this modeling framework open-source and available on GitHub (`https://github.com/nathanieljevans/bayesian-hormesis-detection`). This provides the code to reproduce the paper figures and apply our algorithm to novel data.

There are several future directions for this work. First, further algorithm validation and comparisons should be performed to benchmark performance against previously developed hormetic detection algorithms, such as those presented by Kim, Bartell, and Gillen in 2016 [KBG16]. Second, we would like to apply our algorithm to the full BeatAML dataset to comprehensively report the prevalence of hormetic data.

Future research questions should also investigate how drug promiscuity (drugs with many protein targets) may cause an increase in atypical dose-response behavior. The results in this paper suggest that CAC values affect dose-response curves, and therefore we hypothesize that atypical behavior may correlate with other characteristics of absorption, distribution, metabolism, excretion and toxicity (ADMET) features or by drug promiscuity. If this is the case, the detection of atypical behavior in dose-response curves may be useful to predict adverse drug events (ADE). Finally, we are interested in comparing the proportion of atypical curves in combination therapies compared to single-agent assays. Combination therapies with allosteric, drug-drug interactions, or competitive binding may be detectable as atypical data in functional drug pipelines, which may be informative of the mechanism of action, ADE likelihood, or analytic biases in drug development (such as appearing resistant due to hormetic tails). At the very least, atypical prevalence in combination therapies should be explored due to the recency of their use in functional drug pipelines.

The computational complexity of our algorithm poses a limitation to feasible and widespread adoption. To predict atypical data, we fit a model using the Markov chain Monte Carlo (MCMC), which takes approximately 1-3 minutes

per assay on common hardware. When applied to data sets of many thousands of assays, this represents a significant computational requirement. While parallelization offers a brute-force solution, further work can also be done to derive a Monte Carlo (MC) sampling method by making minor changes to our model architecture. This may improve mixing and allow for faster parameter convergence, decreasing the necessary compute requirements.

Last, we would like to comment on one issue in developing and applying these methods to publicly available datasets such as the Cancer Cell Line Encyclopedia (CCLE), Genomics of Drug Sensitivity in Cancer (GDSC), Cancer Therapeutics Response Portal (CTRP), and the Beat AML study. Although these data sets have made many of the high-level summary metrics (AUC/IC50) publicly available, it is often difficult to obtain raw dose-response data. This precludes the option to rerun quality control or preprocessing steps and hampers the development of reproducibility-improving algorithms. Additionally, providing raw dose-response data from these datasets can facilitate large dataset aggregations, as each dataset could be reprocessed with identical protocols and quality control steps.

2.6 Data and Software Availability

The code for the analysis presented here is available on GitHub: `https://github.com/nathanieljevans/bayesian-hormesis-detection`https://github.com/nathanieljevans/bayesian-hormesis-detection. This implementation is built in python using the *pyro* package, a probabilistic programming language (PPL).

The code responsible for synthetic dose-response generation can be found at: `https://github.com/nathanieljevans/synthetic_doseresponse_generator`.

The beatAML dataset is publicly available through the `http://www.vizome.org/` website and GDC [canb].

2.7 Acknowledgements

We thank Dr. Ted Laderas for helpful discussion of code development and reproducibility, the OHSU Head and Neck Squamous Cell Carcinoma Research Group for related discussions on patient-specific functional drug screens and feedback, and Gareth Harman for many helpful discussions.

3 Data Valuation with Gradient Similarity

3.1 Abstract

High-quality data is crucial for accurate machine learning and actionable analytics; however, mislabeled or noisy data is a common problem in many domains, despite the best efforts of researchers. Distinguishing low- from high-quality data can be challenging, often requiring detailed understanding of how the data were generated, expert knowledge, and considerable manual review. *Data Valuation* algorithms are a class of methods that seek to quantify the *value* of each sample in a dataset based on its contribution or importance to a given predictive task. Previous Data Valuation methods have shown an impressive ability to identify mislabeled or noisy data and that filtering low-value data can improve the predictive accuracy of machine learning algorithms. In this work, I present a simple alternative to existing methods, termed *Data Valuation with Gradient Similarity* (DVGS). This approach can be easily applied to any gradient descent learning algorithm, scales well to large datasets, and performs comparably or better to state-of-the-art methods for tasks such as corrupted label discovery and noise quantification. We evaluate our method on tabular, image, and 'omic datasets to show the generalizability and effectiveness of the method across data types. Our approach has the ability to rapidly and accurately identify low-quality data, thereby reducing the need for expert knowledge and manual intervention in data cleaning tasks and improving machine learning performance.

3.2 Introduction

As discussed in Chapter 1, data valuation is a field focused on algorithms that assign scalar values to each observation in a dataset that characterize the "value" or "usefulness" of that observation toward a given predictive task. Previous work has shown that this approach can be effectively used to identify corrupted labels and that machine learning performance can often be improved by filtering low-valued data [YAP, GZ]. A limitation of current methods, such as Data Shapley [GZ] and data valuation with reinforcement learning (DVRL) [YAP], is the scalability of the algorithm. In this work, we present a novel data valuation algorithm that can be feasibly applied to extremely large datasets and run with time complexity appropriate for use with commonly available hardware.

3.2.1 Prior Art

Dataset Distillation is a related field, which attempts to distill knowledge from a large dataset into a small one by synthesizing a new dataset that is representative of the original dataset but much smaller [WZTE18, YLW23]. Adjacent to this domain is *core-set* or *instance selection* that focus on selecting a subset of a dataset that leads to comparable or better machine learning performance. In many pragmatic applications, *data valuation* can be seen as coreset or instance-selection method; For instance, data valuation produces a ranked list of the samples in a given dataset, based on their value or usefulness towards a predictive task. From the lens of instance selection or coreset, the only distinction between a ranked dataset and a corset is the choice of threshold. Selection of a data value threshold, either by post-hoc analysis or manual choice, re-frames data valuation methods as a *core-set or instance selection* approach. Additionally,

many of the evaluation techniques of common data valuation methods are analogous to instance selection (e.g., corrupted label identification). It should be noted that there is no analog for the equitable data value conditions described by Ghorbani et al. [GZ] in core-set or instance selection. Several notable methods of core-set or instance selection includes *herding* [Wel09, CWS12], distribution-matching [BLK15, FFK11] and incremental-gradient matching approaches [MBL19]. There have also been instance selection approaches for large language models, which require large amounts of data to train, and the choice of prompting can have drastic impacts on model performance [LQC+22, XSML23].

Anomaly detection or *outlier detection* attempts to separate data instances that deviate from the majority of samples [PSCH21]. Data valuation, especially when used to identify corrupted labels or characterizing exogenous feature noise, can be examined from the lens of anomaly detection. For instance, the DVRL *Estimator* model tries to learn a joint probability distribution of exogenous and endogenous features that maximizes predictive performance of a given learning algorithm. If we make the assumption that identifying in-distribution training data will lead to test performance generalization, then DVRL can be thought of as a method for separating anomalous (out-of-distribution) from normal samples (in-distribution). There have been countless methods introduced for anomaly detection, however, of particular relevance to this paper is a gradient-based anomaly representation for autoencoders proposed by Kwon et. al, which defines an anomaly score based on both reconstruction error and the gradient. [KPTA20].

Learning with noise. There has also been significant research on how to train machine learning models in the presence of noisy or corrupted data. These methods range broadly, and include meta-learning sample re-weighting schemes [RZYU18, JZL+17], noise-robust loss functions [ZS18] and loss correction algorithms [HMWG18]. These methods predominately focus on training high-performing models without explicitly removing corrupted or spurious observations; however, several of these methods use re-weighting schemes that rely on interim observation-specific weights and could be considered analogous to data values.

3.2.2 Contributions

Data valuation is an automated approach for characterizing sample informativeness, particularly in data cleaning tasks such as identifying incorrectly labeled or noisy samples. Existing data valuation methods, however, have limitations that hinder their widespread application. Data Shapley does not scale well to large datasets and under performs in certain tasks like corrupted label identification compared to DVRL. DVRL often exhibits high performance in the identification of mislabeled data but is sensitive to hyperparameters, choice of dataset, and model architectures. It can be inconvenient and time-consuming to tune DVRL hyperparameters and is ineffective in certain predictive tasks. Furthermore, while DVRL is significantly faster than Data Shapley, this method still requires sequentially training models to accurately estimate data values, which requires significant computational resources.

In this paper, we introduce a novel data valuation method and compare it against baselines in two key tasks: 1) identifying corrupted labels and 2) identifying samples with high exogenous feature noise. We also explore the application of data valuation in unsupervised learning settings, which to our knowledge is the first method to evaluate

this. Unsupervised data valuation is ideal for quantifying sample noise in biological data types such as 'omics sequencing data (RNA expression, DNA mutation, methylation, etc.). Finally, we apply our method to quantify the data values in the LINCS L1000 level 5 dataset, which contains more than 700,000 high-dimensional samples. Our method demonstrates performance comparable to that of state-of-the-art approaches while being significantly faster than the baseline models. The speed and scalability of our method make it applicable to many large datasets, even with small compute budgets. Moreover, our method is robust to hyperparameters, making it user-friendly and convenient.

Although data quality metrics have been proposed for the LINCS L1000 dataset, such as average Pearson correlation (APC), our data valuation results offer an alternative data quality metric derived directly from the utility of each sample. We show that filtering data based on our data values results in models that perform equivalent or better than data filtering based on APC. Furthermore, we show that our method is more effective in capturing high-valued samples than the APC metric, which could be used to inform future data acquisition decisions.

3.3 Methods

3.3.1 Data Valuation with Gradient Similarity

We propose a method of Data Valuation with Gradient Similarity (DVGS), based on the premise that **source samples with a loss surface similar to the target loss surface will be more useful to a shared predictive task than source samples with dissimilar loss surfaces**. For instance, a training sample loss surface with a similar minima to the validation set is likely to positively contribute to the validation predictive task. This premise is visualized by a toy example in Figure 13. Analytically computing the loss criteria for all parameter values (i.e., the loss surface) is intractable for most problems, and therefore comprehensive comparison of loss surfaces is challenging; however, we can approximate the comparison of loss surfaces by comparing gradient similarities at select parameter values. The comparison of gradients is also advantageous as it factors out the absolute loss value, and instead compares the shape of the loss surface, which is more important for optimization problems.

Similarly to other data valuation methods, DVGS requires a *target* dataset that characterizes the target distribution and to which the *source* dataset is compared. The target dataset may be of high quality, specific prediction domain, or a randomly sampled holdout set. Additionally, the user must define a differentiable predictive model that can be trained using gradient descent. Finally, the *source* dataset is the input on which data valuation will occur, with the goal of characterizing useful or detrimental samples. To perform DVGS, we optimize model parameters using stochastic gradient descent (SGD) on the target dataset and at each iteration compute the target gradient similarity to each source sample gradient. We posit that this approach is liable to accurately estimate data values if the gradient similarities are measured in critical regions of the weight space, such as regions near the validation minima. This procedure is documented in Algorithm 1. There is not justification that this approach satisfies the equitable data value conditions proposed by Ghorbani et al.; however, we empirically demonstrate that this approach effectively characterizes data

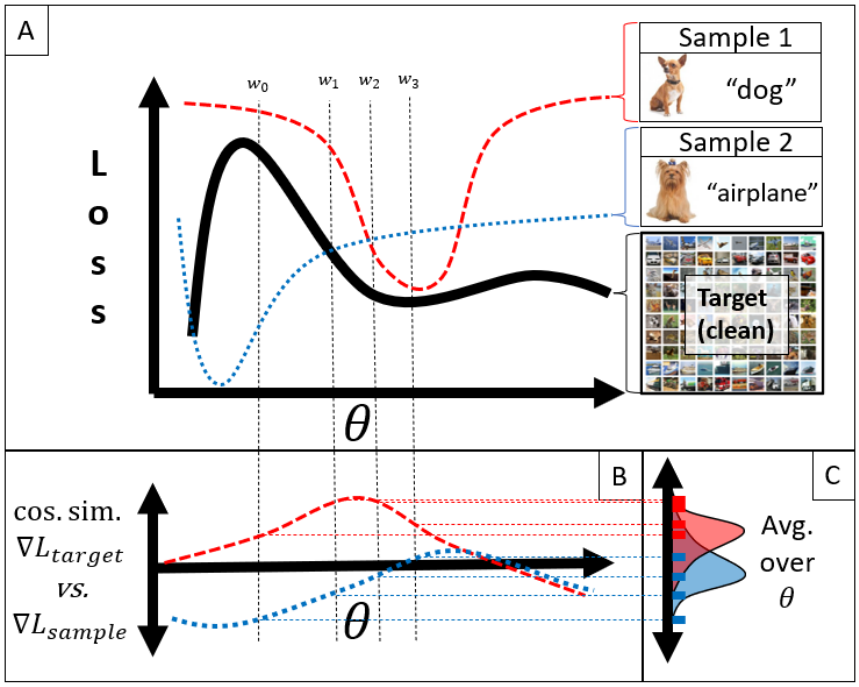


Figure 13: Data valuation with gradient similarity overview. We propose a method of data valuation that compares source samples to the target samples by computing the similarity of gradients during stochastic gradient descent. In panel A, we depict a 1-d loss landscape toy-example; Sample 1 is high-quality (accurately labeled), whereas sample 2 is low quality (incorrectly labeled). In Panel B, we plot the source samples (red and blue dashed lines) gradient similarity to the target samples (black solid line in panel A) gradient at different model parameters values (θ). Panel C shows the marginal distribution of the source samples gradient similarity along the gradient flow (θ). The final sample-specific data value is a mean over the gradient similarity marginal distribution. To make this process tractable, gradient similarities are computed over a limited number of model parameter values during traditional stochastic gradient descent. The computed gradients are visualized by dotted lines in panels A,B and C (w_0, w_1,...,w_3). To choose the relevant values of θ, we use stochastic gradient descent (SGD), with gradients calculated from the target set.

quality in many real-world prediction tasks while being simple, scalable, and easily extensible to a wide range of model architectures and predictive tasks.

To compute the similarity between the gradients of source samples and the target dataset, we must define a similarity function that takes as input two high-dimensional gradient vectors and returns a single scalar characterizing similarity. Theoretically, any distance metric is applicable here; however, we chose to use cosine similarity because it produces easily interpreted values between [-1,1] and neglects the magnitude of the vector, which we rationalize may skew gradient comparisons. Gradient magnitudes are likely to vary drastically between early- and late-stage training, and to avoid biasing data values by training-stage-dependent magnitudes, we rationalize that gradient magnitude should be ignored.

In classification problems, each class is likely to have a distinct gradient (particularly in early training), and therefore target sets with a class imbalance are likely to introduce class-specific biases to data values. For instance, if the target set is majority positive class, then source samples with the negative class will be particularly dissimilar, even if they are

Algorithm 1 Data Valuation with Gradient Similarity

Require: differentiable model (f_θ), learning rate (α), source dataset $\mathcal{D}_s$, target dataset $\mathcal{D}_t$, number of training iterations (N_{iter}), target batch size (R), loss criteria ($\mathcal{L}$), and similarity criteria (C).

for $i = 0, 1, \ldots, N_{iter}$ **do**
  $B_i \sim \mathcal{D}_t$ ▷ sample mini-batch from target set
  for $j = 0, 1, \ldots, R$ **do**
    $x_j, y_j \sim B_i$
    $\hat{y_j} \leftarrow f_\theta(x_j)$ ▷ predict outcome for target batch
  end for
  $\nabla\mathcal{L}_i^{target} \leftarrow \frac{\partial}{\partial\theta}(\frac{1}{R}\sum_{j=0}^{R}\mathcal{L}(\hat{y}_j, y_j))$ ▷ compute target batch gradient
  for $k = 0, 1, \ldots, N_{source}$ **do**
    $x_k, y_k \sim \mathcal{D}_s$
    $\hat{y_k} \leftarrow f_\theta(x_k)$ ▷ predict outcome for source sample
    $\nabla\mathcal{L}_k^{source} \leftarrow \frac{\partial}{\partial\theta}(\mathcal{L}(\hat{y}_k, y_k))$ ▷ compute the gradient for the source sample
    $\nu_k^i \leftarrow C(\nabla\mathcal{L}_k^{source}, \nabla\mathcal{L}_i^{target})$ ▷ compute similarity of source sample gradient to the target batch gradient
  end for
  $\theta_{i+1} \leftarrow \theta_i - \alpha\nabla\mathcal{L}_i^{target}$ ▷ update model parameters using the target batch gradient
end for
for $k = 0, 1, \ldots, N_{source}$ **do**
  $\nu_k \leftarrow \frac{1}{N_{iter}}\sum_{i=0}^{N_{iter}}\nu_k^i$ ▷ compute the average gradient similarity for each source sample
end for

valuable in training. To avoid inadvertently biasing data values based on class, we suggest balancing class weights as described by [Zen99] when computing target gradients.

Intuitively, the choice of initialization weights is likely to produce different data values, especially if the target set has a multimodal loss surface. To prevent variance in DVGS data values due to weight initialization or stochastic mini-batch sampling, we add the option to run the DVGS algorithm multiple times, each with unique weight initialization and random sampling seeds. Using this approach enables DVGS to explore multiple minima and compute similarity values on a wider range of parameter values. To aggregate a final data value, gradient similarities are then averaged across all iterations and runs.

3.3.2 Time Complexity

In most applications, it is reasonable to assume that the target dataset is much smaller than the source dataset, and therefore most of the runtime is spent looping through the source samples to compute gradient similarities. This can be partially mitigated by only computing gradient similarities every T iterations or by pre-training the model. We estimate the computational complexity[5] in big O notation as:

$$\mathcal{O}(\frac{N_{iter}N_{source}}{T})$$

DVGS should scale approximately linearly with the number of source samples and training iterations. A particular advantage of the DVGS methods is that only a single model need be trained, whereas Data Shapley and DVRL require

[5] See supplementary for experimental evaluation of time complexity.

training many models sequentially. This time complexity makes it suitable for application to large datasets, such as the LINCS dataset with more than 700k high-dimensional (978 features) samples. Additionally, DVGS can be run in parallel and the results averaged to compute more accurate data values; this ensemble approach is ideal for large datasets and complex loss surfaces. In many tasks, such as image classification, it is recommended to pre-train the convolutional layers using the source dataset prior to performing DVGS, which can be done in a supervised or unsupervised manner.

3.3.3 Data

In this paper, we apply our data valuation algorithm to four datasets under various conditions:

- The **ADULT** dataset, also known as the "census income" dataset, consists of 14 categorical or integer features representative of an adult individual and labeled based on whether they make more than 50k dollars per year [DG].
- The **BLOG** dataset consists of internet blog characteristics parsed from the raw html file and the output is the average number of comments received; We then binarize the endogenous variable with threshold of 0 [Búz12].
- The **CIFAR10** dataset, which consists of tiny images labeled as one of 10 possible objects [Kri09]; we transform the images into an informative feature representations using a pre-trained InceptionNet prior to data valuation [SLJ+14].
- The **LINCS L1000** dataset measures RNA expression in cell lines some time after a chemical or genetic perturbation [SNC+17b] We further break the LINCS L1000 into two data partitions: 1) all data and 2) high-APC (>0.5) data (see supp. note 9.2).

We chose the first three datasets (ADULT, BLOG, and CIFAR10) to match the evaluations performed in previous work [YAP, GZ]. Similarly, we try to match the respective dataset size (target, source, test) choices made in previous work to provide similar evaluations.

The LINCS L1000 is a widely used dataset that suffers from known data quality issues; Removing inaccurate or noisy samples from this dataset could benefit the cancer drug response domain and lead to more accurate and actionable results from downstream analytics.

3.3.4 Dataset Corruption

To simulate poor data quality, we artificially corrupt datasets in two ways:

- **Label Corruption**; Endogenous variable (y)
- **Feature Corruption**; Exogenous variable (x)

Labels are corrupted by randomly shuffling a proportion of the source dataset class labels; for instance, an image of a dog may be re-labeled as "cat". The corrupted sample indices are then used as ground truth of data quality, which can be compared with the data values. The expectation is that corrupted labels will have lower data values, indicating that they are less valuable to model performance. To summarize the ability of data values to identify corrupted samples, we use the area under the receiver operator curve (AUROC) metric:

$$AUROC(c, -\nu)$$

Here c is the corrupted label mask $(0 = uncorrupted;\ 1 = corrupted)$ and ν are the data values. Notably, we flip the data value sign as we expect large data values to indicate high quality data, and small data values to indicate low quality or mislabeled observations.

To explore the ability of data valuation to capture exogenous feature sample quality, we add Gaussian noise to each observation:

$$x^*_{i,j} = \mathcal{N}(0, \phi_i) + x_{i,j}$$

where x^*_i is the feature j of the corrupted sample i, and ϕ_i is an observation-specific noise rate sampled from a uniform distribution. Thus, samples with larger noise rates (ϕ_i), will have noise with greater variance. The primary evaluation task is to apply the data valuation and compare the data values against sample-specific noise rates. We expect that samples with large noise rates will have small data values, indicating that they are less valuable to model performance. To evaluate performance on this task, we use Spearman correlation [Spe87]:

$$\rho = Spearman(\phi, -\nu)$$

3.4 Results

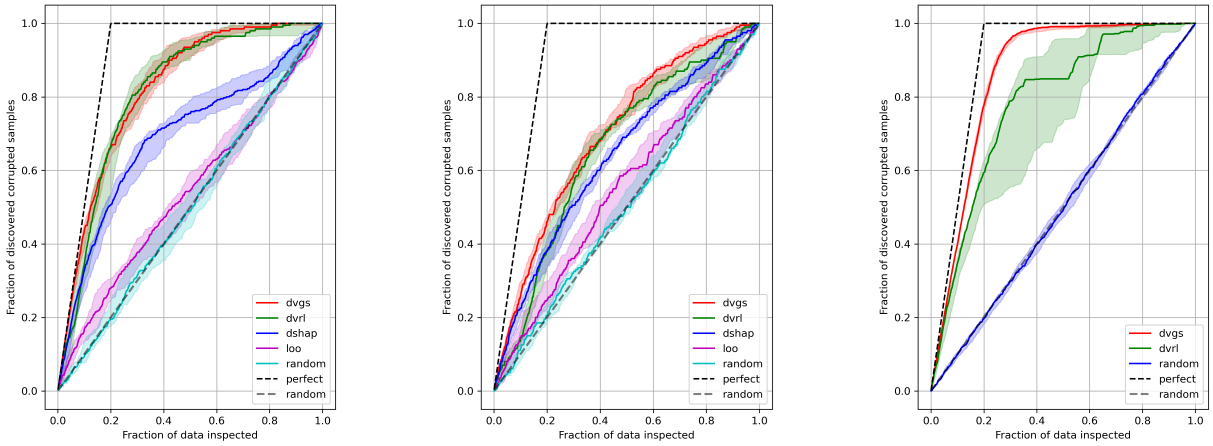
3.4.1 Label Corruption

To evaluate the ability of data values to capture mislabeled samples, we artificially corrupt labels in three classification datasets: ADULT, BLOG and CIFAR10. We compare DVGS to several baseline methods:

- Randomly assigned data values (null model)
- Leave-one-out (LOO)
- Data Shapley (dshap)
- Data Valuation with Reinforcement learning (DVRL)

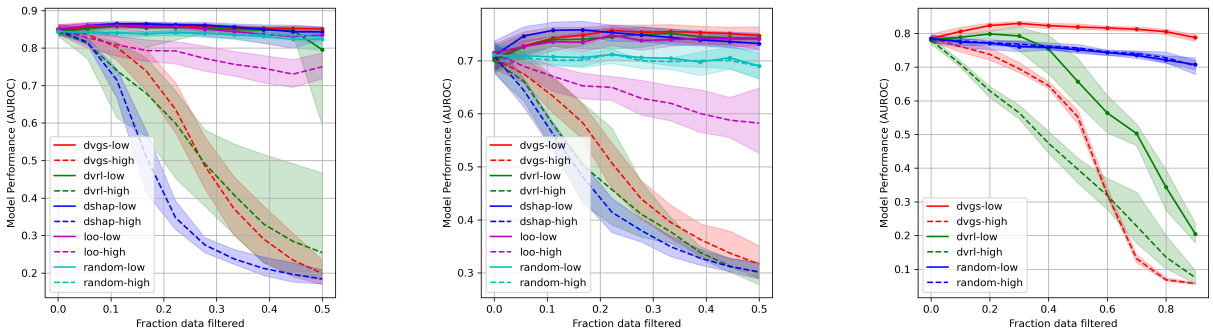
Due to the time complexity of Leave-one-out and Data Shapley, we only apply these to the ADULT and BLOG datasets.

In all three datasets, we corrupt 20% of the labels. For the ADULT and BLOG datasets, we use 1000 source observations and 400 target observations. For the CIFAR10 dataset, we use 5000 source observations and 2000 target observations. We expect accurate data valuation to produce values such that corrupted samples data values will be smaller than uncorrupted samples, indicating that they are less valuable or useful toward our target predictive task. Additionally, we expect that filtering corrupted labels should improve model performance. In each experiment, we evaluate the ability of data values to 1) identify corrupted labels and 2) modify model performance as measured on a hold-out test set when we filter a proportion of the dataset. In this second task, we evaluate the performance changes when we filter high-values (expectation that performance will decrease) versus low-values (expectation that performance will improve or be unaffected).



(a) ADULT Dataset **(b)** BLOG Dataset **(c)** CIFAR10

Figure 14: Evaluation of respective data valuation methods ability to identify corrupted labels.



(a) ADULT Dataset **(b)** BLOG Dataset **(c)** CIFAR10

Figure 15: The evaluation of respective data valuation methods ability to impact model performance when filtering either high value (dashed lines) or low values (solid lines).

DATASET	DVGS	DSHAP	DVRL	LOO	RANDOM
adult	**0.896 ± 0.030**	0.731 ± 0.049	0.887 ± 0.042	0.542 ± 0.056	0.503 ± 0.050
blog	**0.750 ± 0.028**	0.671 ± 0.021	0.697 ± 0.033	0.558 ± 0.063	0.509 ± 0.028
cifar10	**0.954 ± 0.009**	NA	0.835 ± 0.110	NA	0.499 ± 0.019

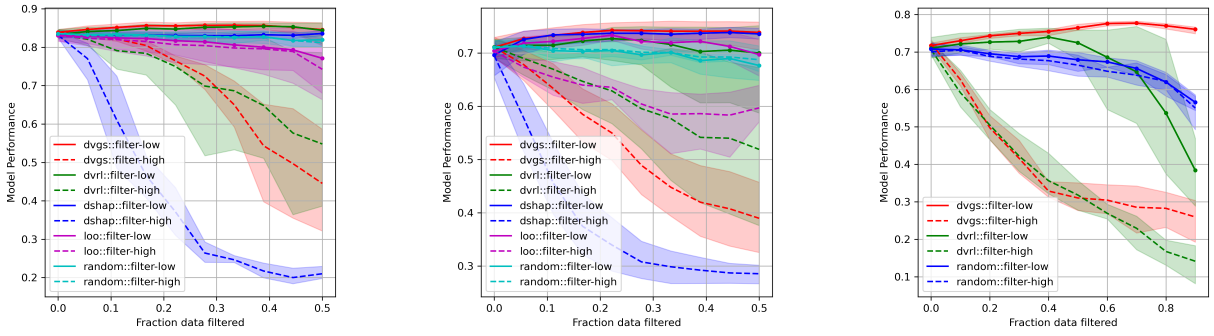
Table 5: The Area under the receiver operator curve (AUROC) scores if the data values are used to predict corrupted labels (e.g., $score = AUROC(noise\_labels, -data\_values)$); mean ± std.

For all three datasets, we use a 2-layer neural network as the learning algorithm and the area under the receiver operator curve (AUROC) as the performance metric [HM82]. Each experiment is run at least five times with randomly sampled data subsets and unique neural network weight initialization. Experiments are repeated to ensure stable results across diverse subsets of data and weight initializations.

We visualize the results of five data valuation methods in Figure 14, and show that DVGS performs comparably or better to the baseline methods. Of note, DVGS performs particularly well on the CIFAR10 dataset; Which may be due to the highly informative exogenous features extracted from a pre-trained InceptionNet model [SLJ+14], which were used as input to data valuation.

These results are shown in Table 5. DVGS is better able to quantify the corrupted labels in all three datasets, as measured by the AUROC score. Notably, DVRL often performed comparably to DVGS; however, DVRL convergence was inconsistent and occasionally resulted in a suboptimal policy, which is evidenced by the wide confidence intervals of DVRL in Figure 14 and large standard deviations of CIFAR10 in Table 5. Additionally, as evidenced in Figure 15, DVGS underperforms compared to Data Shapley in characterizing high data values.

3.4.2 Characterization of Sample Noise



(a) ADULT Dataset **(b)** BLOG Dataset **(c)** CIFAR10

Figure 16: The evaluation of respective data valuation methods ability to impact model performance when filtering either high value (dashed lines) or low values (solid lines). The y-axis measures model performance using the AUROC metric.

In many domains, input features may be noisy due to measurement error, natural stochasticity, or batch effects, leading to inaccurate sample informativeness. To explore the ability of data valuation to quantify input feature noise, we

Dataset	Learning	DVGS	DSHAP	DVRL	LOO	RANDOM
adult	supervised	**-0.225 ± 0.061**	-0.130 ± 0.091	-0.159 ± 0.074	-0.022 ± 0.076	0.007 ± 0.026
blog	supervised	**-0.106 ± 0.077**	-0.086 ± 0.074	-0.100 ± 0.344	-0.045 ± 0.078	-0.011 ± 0.054
cifar10	supervised	**-0.402 ± 0.081**	NA	-0.358 ± 0.103	NA	-0.000 ± 0.018
cifar10	unsupervised	**-0.757 ± 0.131**	NA	NA	NA	-0.003 ± 0.014
lincs (APC>0.5)	unsupervised	**-0.505 ± 0.018**	NA	NA	NA	NA

Table 6: The Spearman correlation of predicted data values and artificial sample noise rates. The top performing method for each row is bolded; mean ± std.

artificially corrupt the exogenous features as described in Section 3.3. For this task, we evaluate data valuation in supervised (ADULT, BLOG and CIFAR10) and unsupervised learning (CIFAR10 and LINCS) settings. In the supervised setting, we use architectures and hyper-parameters identical to those described in Section 3.4.1. In unsupervised settings, we use an autoencoder architecture [RM87, Bal12] to create a low-dimensional representation and train using reconstruction error. We justify that noisy samples will be more difficult to reconstruct and are likely to be detrimental to the performance. For the unsupervised setting, we apply our methods to two datasets: the CIFAR10 dataset and a high-quality subset of the LINCS L1000[6]. The results are shown in Table 6. As in section 3.4.1, we also evaluate the performance impact of filtering data based on data values, and these results are shown in figure 16. We find that DVGS can effectively characterize noise rates across all datasets. Compared to baseline methods, DVGS produces data values that most strongly correlate (negatively; as expected) with ground-truth noise rates. Additionally, when we compare model performance improvements when low-valued data are removed, shown by the solid lines in figure 16, in we find that DVGS model performance is equivalent or better than baseline methods. As in the supervised setting results, we find that Data Shapley outperforms DVGS at quantifying high-quality data values, measured by model performance decrease when filtering high-valued data in both the ADULT and BLOG datasets, shown in figure 16 (a,b). In some of the learning tasks listed in Table 6 only one or none of the baseline methods are calculated due to compute limitations.

3.4.3 Computational Complexity

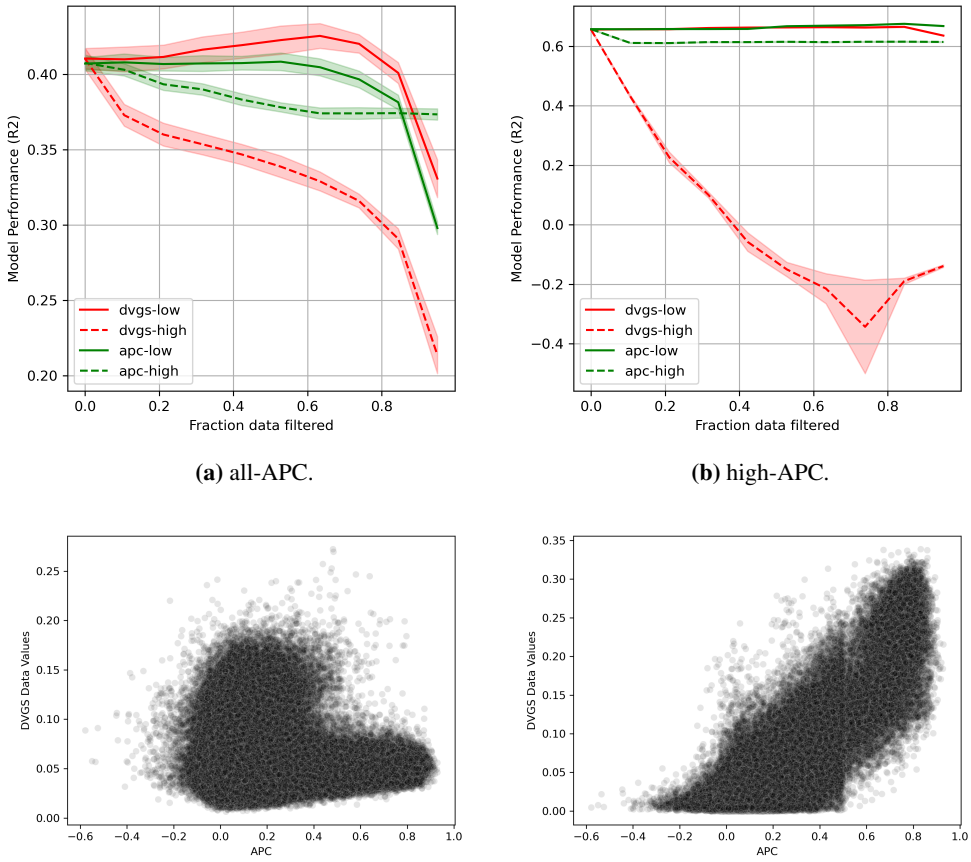
DVGS can be applied to large datasets and complex tasks with markedly lower computational costs than previous data valuation methods and enables application to new domains and data types. In Table 7, we show the runtime of four data valuation algorithms. On average, DVGS is roughly five times faster than DVRL and more than 100 times faster than truncated Monte-Carlo (TMC) Data Shapley. In comparison to DVRL and Data Shapley, which require sequential training of models on different subsets of data, the DVGS method requires training only one model. Furthermore, by computing the gradient similarities every T batches, the DVGS runtime can be reduced by a factor of T. In practice, we find that using values of t between 2 and 5 has a marginal impact on the performance of data values for corrupted label discovery. Further DVGS time complexity experiments can be found in Supplementary 9.3.

[6]Observations with an average Pearson correlation between replicates greater than 0.5

method	exp1	exp2	exp3	exp4	exp5	exp6	exp7	exp8
DShap	515.2	774.9	NaN	404.5	631.0	NaN	NaN	NaN
DVGS	**1.3**	**1.2**	**5.3**	**1.4**	**1.3**	**5.1**	154.0	41.7
DVRL	9.9	9.5	13.2	9.8	9.8	11.7	NaN	NaN
LOO	33.0	34.0	NaN	35.1	34.7	NaN	NaN	NaN

Table 7: Average runtime (in minutes) of 8 experiments. Experiments 1-3 were for label corruption; Experiments 4-6 were for noise characterization; Experiments 7 and 8 were unsupervised characterization of noise.

3.4.4 Data Valuation of the LINCS dataset



(a) all-APC. **(b)** high-APC.

(c) all-APC. **(d)** high-APC.

Figure 17: (a-b) The reconstruction error of autoencoders applied to the LINCS L1000 data when filtering low- and high- value data. (c-d) DVGS data values compared to APC values.

In this section, we apply our DVGS method to quantify LINCS L1000 sample quality across all chemical perturbations (chemical and genetic). In each experiment, we randomly sampled a target set and hold-out test set (5000 observations) in two conditions:

- **Noisy Target set** (all-APC). Target dataset sampled from all available observations.

- **Clean Target set** (high-APC). Target dataset sampled from high-APC observations (APC > 0.5).

In both configurations, we adjust the target/test set sampling probability so that each sampled target/test set should have roughly the same proportion of each perturbation type. The source set consists of all samples that are not in the target or test sets. See Supplementary 9.2 for more information on APC calculation.

Although data valuation of LINCS could be done in a supervised or unsupervised setting, we chose to use an unsupervised prediction task for the following reasons:

- **Simplicity**: Encoding drug and cell lines requires additional overhead and may bias the results toward the method chosen; e.g., encoded by drug targets, cell line expression, etc.
- **Imbalanced Dataset**: drug perturbations and cell lines are not equally represented in the LINCS dataset, and this may cause bias toward the over represented drugs or cell lines. While this is a concern in an unsupervised setting, we rationalize that removing exogenous variables may help mitigate the issue. Additionally, to further mitigate this concern we select a target set with more balanced proportions of drug perturbations.
- **Noise Quantification**: We consider measurement noise to be the primary data quality issue in the LINCS L1000 dataset and would like our data values to characterize sample noise rates. The results from Table 6 suggests that DVGS can effectively quantify sample noise using an unsupervised learning task.

For this task, we use an autoencoder with 2-layers in the encoder and decoder networks and 32 latent channels (embedding dimension). To avoid dependence on a specific target set, we run this experiment several times ($n \geq 3$) using different source, target, and test sets, as well as unique neural network weight initialization. We compare the DVGS data values with the APC metric, proposed by Pham et al., to compare the generated data values with alternative LINCS L1000 sample quality metrics. We evaluate the performance of LINCS data values by their ability to modify model performance when filtering high- and low-value data. Figure 17 shows the performance comparison between the APC and DVGS data values. In the high-APC and all-APC conditions, we see that DVGS captures low data quality much better than the APC metric. In the all-APC condition, DVGS outperforms APC in capturing high-quality data, however, the DVGS data values and APC perform comparably in the high-APC condition. Additionally, we find that DVGS values and APC values correlate in the high-APC condition (Pearson Correlation $\sim$ 0.84) but not in the all-APC condition (Pearson Correlation $\sim$ -0.05). More specifically, in Figure 17c we see that high APC values are depleted for high data values, suggesting that DVGS data values in the all-APC condition characterize a different aspect of data quality or usefulness than APC.

3.5 Discussion

In this work, we address the limitations of current data valuation methods by proposing a fast and robust method to estimate the value of the data. We show that this method performs comparably or better than baseline methods in several tasks, including 1) identifying corrupted labels and 2) characterizing exogenous feature noise. Additionally, we

have shown that our method works well to modify model performance when filtering data based on data values, and performs comparably or better than baselines when filtering low-value data. While Data Shapley and DVRL tend to lead to larger decreases in model performances when filtering high-value data, DVGS performs exceptionally well at identifying corrupted labels and noisy samples, especially in vision tasks using pre-trained models. DVGS is also, on average, 100 times faster than Data Shapley (TMC) and 5 times faster than DVRL. This improvement in time complexity makes DVGS applicable to a wide range of datasets and domains. Additionally, in the reported experiments, DVGS was stable across hyperparameters (see Supplementary note 9.1), data partition, and weight initialization. These characteristics make DVGS convenient and robust for application in data cleaning and machine learning.

To show the value of our DVGS method in a real-world scenario and to address data quality issues in a foundational dataset, we apply DVGS to the LINCS L1000 level 5 dataset that has more than 700k high-dimensional samples. We compare our method with a previous LINCS quality metric, the Average Pearson Correlation (APC), and show that our DVGS-produced data values are better able to modify model performance when filtering based on value. Interestingly, using a target dataset drawn randomly from the dataset (i.e. not necessarily high-quality) leads to data values that 1) do not correlate well with APC, and 2) significantly outperform APC as measured on a hold-out test set drawn from the full dataset.

3.5.1 Limitations and Future Directions

Similarly to DVRL, our DVGS method lack the equitable data value properties proposed by Ghorbani et al. and therefore should not be interpreted in the same way; DVGS data values do not have a convenient or articulable interpretation like Data Shapley values. Rather, DVGS data values should be considered latent variables characterizing data informativeness, and we make no assumption about the linearity or magnitude of DVGS data values. These traits suggest that the DVGS data values should be treated contextually, e.g., as a ranked list of sample values. Pragmatically, ranked samples fulfil the requirements of many of the evaluation techniques used by previous data valuation methods [GZ, YAP] including identifying corrupted labels and noise quantification. Future directions may consider learning a task-specific function to estimate Data Shapley values from DVGS data values, which would allow users to interpret the DVGS data values in a comparable way to Data Shapley. This could be done by performing the DVGS data valuation and calculating a limited number of Data Shapley values, which could then be used as a training set ($values_{DVGS} \sim f_{nn}(values_{DataShapley})$. Such an approach could merge the scalability advantages of DVGS with the interpretability of Data Shapley.

Through the lens of anomaly detection, DVGS can be viewed as a meta-learning algorithm that quantifies the similarity of the source samples to the target dataset and could potentially be used for anomaly detection. Additionally, this perspective may help explain why the DVGS method under performs compared to baselines at identifying high-value data. For instance, if DVGS data values are considered a metric of similarity to the target set, then it may be that the most "similar" samples are not necessarily the most useful, whereas the most "dissimilar" data are likely erroneous or detrimental. It is therefore important that large data values be treated with caution. Additionally, it raises the question:

How does DVGS handle redundant (or highly-similar) data in either the target or source datasets? Future work should address these concerns and characterize how redundancy can skew or alter DVGS data values.

While DVGS works remarkably well on the evaluations listed in this paper, we do recognize that it is rare for gradient-based learning algorithms to be trained on gradient from single samples (e.g., on-line learning) and that more commonly they are trained with mini-batches, thus implying that any sample's value or usefulness toward a predictive task cannot be considered independent of the other samples. Future work may wish to address this by looking at gradient similarity within mini-batches, or by selecting samples that align mini-batch gradients to the target dataset. One can imagine bi- or multimodal sample gradients, all of which may align poorly to a target mini-batch gradient, but when source samples are averaged in a mini-batch, may align far more closely. In the Appendix section 9.4, we present an alternative approach to DVGS, termed Data Valuation with Gradient Alignment (DVGA). In this method, we propose learning sample selection weights that align the batch gradients of the source sample with the target gradient. This allows for the selection of the source sample in the context of the full dataset.

Code and Data Availability

The Adult, Blog and Cifar10 datasets can be accessed from the UCI machine learning repository [DG]. The LINCS data can be accessed from the CLUE data library.

All code used for production of the DVGS figures and the methods described can be found here `https://github.com/nathanieljevans/DVGS`.

4 Graph Neural Networks for prediction of synthetic perturbation biology

4.1 Abstract

Modeling perturbation biology, with its vast array of molecular entities, presents a formidable challenge. In this chapter, we explore the potential of Graph Neural Network (GNN) algorithms to leverage mechanistic prior knowledge to accurately predict expression response to synthetic perturbations. First, we extend the GeneNetWeaver (GNW) package, an ODE based synthetic data generator, enabling it to generate drug-like perturbations. Next, we introduce a novel GNN architecture specifically designed to model perturbation biology synthetic data generated by GNW. The GNN model parameters are optimized using synthetic single-agent perturbed expression time series and evaluated on holdout test sets of single and combination agents. Our GNN algorithm achieved an R^2 value of 0.895 in the single agent test set and a R^2 of 0.765 in the 2-drug combinations. Additionally, we show that biologically tailored GNNs can be structured to infer adjacent aspects of drug response, such as drug binding affinity. These results provide nascent evidence that GNNs are a suitable algorithm to incorporate mechanistic prior knowledge of molecular interactions into deep learning.

4.2 Introduction

In the next three chapters, we explore the question:

> *Can the inclusion of mechanistic prior knowledge of molecular interactions 1) improve machine learning performance and 2) aid interpretation of deep learning models for perturbation biology prediction tasks?*

In this chapter, we develop early GNN prototypes to investigate if GNNs are suitable algorithms for the inclusion of mechanistic prior knowledge of molecular interactions. Specifically, we will focus on the use of GNNs in synthetic perturbation biology prediction tasks.

To accurately predict perturbation biology requires an understanding of the interactions between countless molecular entities involved in a biological system. Given the current landscape of perturbation biology, there is limited data available to infer these processes accurately. This task is further challenged by the measurement noise inherent in current high-throughput perturbation assays.

An attractive approach to address these limitations, which could significantly improve the ability of deep learning models to predict perturbation biology, is to leverage the current biological knowledge base to constrain feature interactions and build custom algorithms tailored specifically to the perturbation biology. This approach is supported by the *no free lunch theorem* and by the success of alternative deep learning tasks such as computer vision and natural language processing.

For those familiar with the field, one might ask *why not just use ordinary differential equations (ODE)?* ODEs are commonly used to model mechanistic processes, including gene regulatory networks (GRNs), and have the advantage of being highly interpretable. Notably, the GNW package that we extend in this chapter utilizes ODEs to generate

expression time-series, which further highlights their utility in this field. The performance of ODEs, however, can degrade severely if given inaccurate constraints (e.g., inaccurate gene-gene/protein-protein interactions) or in the presence of measurement noise. ODE algorithms also scale poorly to a large number of interactions and can result in "stiff" systems that are challenging to effectively optimize. Additionally, common ODE models tend to assume linear interactions between variables, which may not be appropriate for the complexities of biology. We seek to overcome some of these limitations by using deep learning algorithms that may be more resilient to noise in the data, scale efficiently, and perform well in the presence of inaccurate or missing knowledge.

Contributions. In this work, we seek to 1) produce a tool that can be used to generate synthetic perturbation biology datasets and 2) investigate if graph neural networks are a suitable algorithm for the development of accurate models of synthetic perturbation biology through the inclusion of knowledge about the underlying gene regulatory network.

4.3 Proposed Methods

4.3.1 Synthetic Data Generator

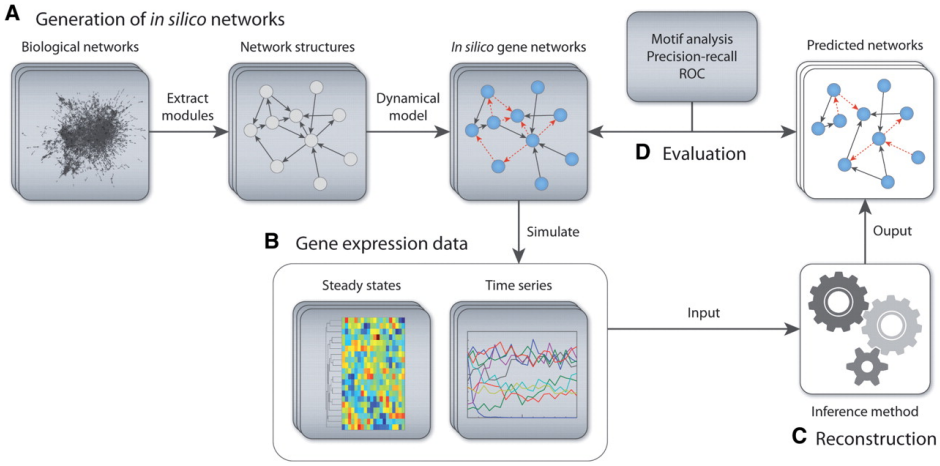


Figure 18: Reproduction of GeneNetWeaver (GNW) Overview figure. The GNW software is released under MIT license and this publication Figure is covered by the Creative Commons 4.0 copy right. "Benchmarking and performance assessment of network inference methods using GNW. (A) In silico gene networks are obtained by extracting subnetwork structures from known transcriptional networks (Escherichia coli, Saccharomyces cerevisiae, etc.) before being endowed with detailed dynamical models of gene regulation accounting for both transcription and translation, independent and synergistic interactions, as well as molecular and measurement noise. (B) In silico gene networks are simulated to produce steady-state and time-series expression data for a variety of experiments such as wild-type, knockout, knockdown and multifactorial perturbation experiments. (C) Inference methods are asked to predict structures of in silico benchmark networks from gene expression data. (D) From network prediction files, GNW performs a network motif analysis which often reveals systematic prediction errors, thereby indicating potential ways of network reconstruction improvements. It also automatically generates comprehensive reports including standard metrics such as PR and ROC curves." [SMF11]

To produce synthetic perturbation data, we have created a pipeline that extends the GeneNetWeaver (GNW) software package [SMF11]. GNW uses ordinary differential equations (ODEs) with feature constraints from real-world gene regulatory network (GRN) (yeast, Ecoli, etc.) to simulate bulk mRNA gene expression time series under a variety of perturbation conditions. GNW outputs include:

1. Cell line steady-state gene expression in the absence of any perturbation and,
2. Gene expression time-series after the introduction of genetic or chemical perturbation.

GNW uses ODEs of the form:

$$\frac{dx_i}{dt} = m_i f_i(y) - \lambda_i^{RNA} x_i$$
$$\frac{dy_i}{dt} = r_i x_i - \lambda_i^{PROT} y_i$$

Where,

- x_i is RNA gene expression.
- y_i is protein abundance.
- m_i and r_i are production rates.
- $f_i(y)$ is a function of transcription factor activation
- λ represents respective degradation rates.
- The subscript i represents the respective gene index.

We extend the GNW method by simulating a knockout (KO) by setting m_i to zero, a knockdown (KD) by setting m_i to half its original value and over-expression (OE) by setting m_i to twice its original value. Chemical perturbations are simulated by randomly sampling a set of gene targets and dissociation constants and then modifying the target gene m_i values according to the equation:

$$m_i^{perturbed} = m_i \pm \frac{m_i}{1 + \frac{k_d}{c}}$$

Here k_d represents the dissociation constant and c is the concentration in micro molars. We simulate multi-drug combinations by modifying the union of drug targets. Drug target collisions are handled by either:

Same drug types ("antagonistic combination"):

$$m_i^{perturberd} = m \pm m_i * \frac{c_1}{c_1 + k_{d,1}(1 + \frac{c_2}{k_{d,2}})}$$

Different drug types ("simple average"):

$$m_i^{perturberd} = \frac{m_i^{agonist} + m_i^{inhibitor}}{2}$$

We note that the accepted scientific premise of pharmacological binding is that a drug binds to a protein and stabilizes an active or inactive conformation, and therefore drug binding does not necessarily change the protein expression directly. However, in this ODE approximation of a biological system, modeling drug binding as a modification in the protein product has an analogous functional effect of changing the system dynamics, and therefore we believe it to be a rational approximate model for drug binding. In practice, we find that our approach generates reasonable simulations that capture many aspects of real-world perturbation behavior.

Different cell contexts (such as cell type, patient, disease, etc.) are likely to have different gene regulatory networks (GRNs), which will result in unique expression steady states and perturbation response. To emulate this behavior, each synthetic cell line is simulated using a unique GRN. Each cell line starts with a "parent" or "shared" GRN and from which a random subset of edges are removed to create a cell-line specific GRN.

There are a variety of configuration parameters that can be manually adjusted, allowing the generation of smaller or larger datasets and with different levels of resolution. Table 8 describes the available settings and their impact on synthetic outputs.

Table 8: Synthetic data generator configuration parameters.

Parameter	Description	Default Value
maxtTimeSeries	max time	500
dt	Time step	50
n_cell_lines	Number of cell lines	10
prob_remove	Probability of edge removal	0.1
graph_name	Graph name to use as base PPI network	InSilicoSize100-Ecoli1.tsv
num_agonists	Number of agonist drugs to create	5
num_inhibitors	Number of inhibitor drugs to create	5
min_log_conc	minimum concentration to simulate (log10(uM))	-5
max_log_conc	maximum concentration to simulate (log10(uM))	1
num_dose_pts	number of concentrations to simulate	5
expected_targs	The mean of the number of drug targets	2
min_kd	minimum dissociation constant (kd) (log(uM))	-3
max_kd	maximum dissociation constant (kd) (log(uM))	0

In Figure 23a we use the principal component analysis (PCA) to visualize the perturbation to ensure rationality of the generated data. Our expectation is that primary sources of variance will be mediated by:

- cell line
- perturbation (drug, or genetic change)

- time since perturbation was introduced

The Synthetic GNNCDR data object

Now that we have generated synthetic data, we need to organize them in a format conducive to ‘graph neural networks‘. We use PyTorch geometric [PGM+19] to build our deep learning models, and use the "HeteroData" object to structure the training data.

We will format our synthetic data as a heterogeneous graph with 6 node types:

- protein
- agonist
- inhibitor
- KO (knockout)
- KD (knockdown)
- OE (overexpression)

and 7 edge types:

- (agonist, targets, protein)
- (inhibitor, targets, protein)
- (KO, targets, protein)
- (KD, targets, protein)
- (OE, targets, protein)
- (protein, activates, protein)
- (protein, inhibits, protein)

Genetic nodes do not have node attributes and only target a single protein. Chemical perturbations have concentration node attributes (‘conc‘) and can target multiple protein nodes. Protein-protein edges can be of two types: "activate" and "inhibit". A toy graph is demonstrated in Figure 19.

For programmatic simplicity, we have opted to include all perturbation nodes in each observation graph, and specify "active" perturbations with non-zero node attributes. For chemical perturbations (agonist, inhibitor), this node attribute represents concentration. For genetic perturbations (KO, KD, OE), we use an arbitrary nonzero value of 1 to specify an active perturbation (zero otherwise).

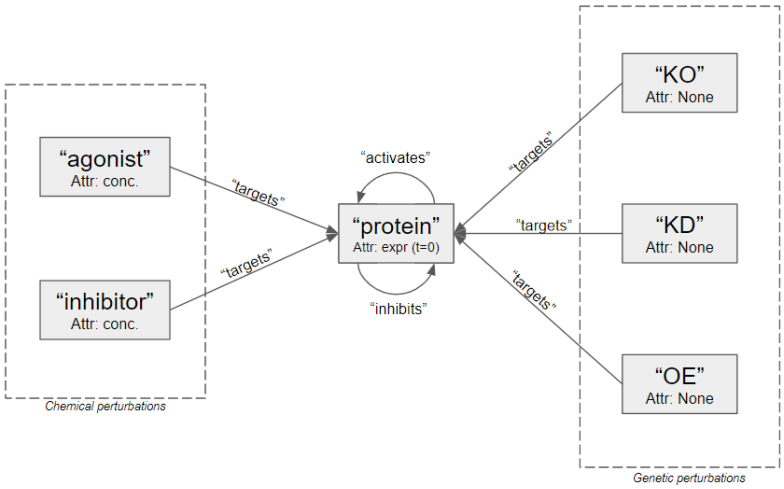


Figure 19: Synthetic data graph object overview.

The 'SynthHeteroDataset' function also includes options to specify additional forms of Gaussian noise, sparsity (feature dropout), introduce false protein-protein edges, introduce false drug-target edges, to remove protein-protein edges, or to remove drug-target edges. This allows the user to specify different forms of measurement noise or structural inaccuracies.

Training partition scheme. Training partitions will only include single-agent perturbations while test sets will include both single-agents and all combination agents. Since our model attempts to learn drug-specific parameters (learning architectures designed to capture binding affinity information), we cannot exclude a drug entirely for the training dataset. Instead, we will group observations by cell-line and drug pairs and separate them into *either* train or test datasets. This allows us to train drug-specific parameters while still testing on unseen cell line responses. This procedure is illustrated in Figure 20.

We will create three partitions:

- **train**: ALL genetic perturbations and a proportion of single agents
- **validation**: A proportion of single agent data
- **test**: Combination drug responses

Figure 21 describes our analysis pipeline. First, we use our synthetic generator method to produce synthetic data. Next, we randomly assign data to the train, validation, and test partitions as described above. Lastly, we use the training set to optimize the GNN parameters and evaluate performance on holdout sets of combination agents (test) and single agents (validation). Note that while we use the common ML nomenclature of "test" and "validation," we do not perform any hyperparameter optimization using the validation set and are therefore not concerned about data leak during evaluation.

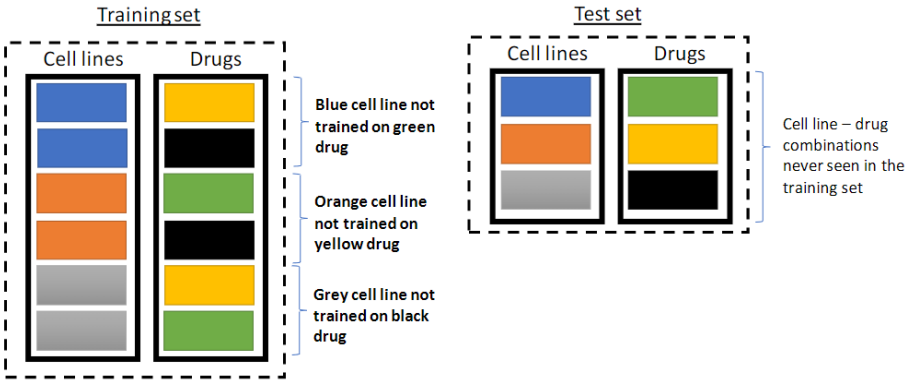


Figure 20: The synthetic train-test partitioning scheme.

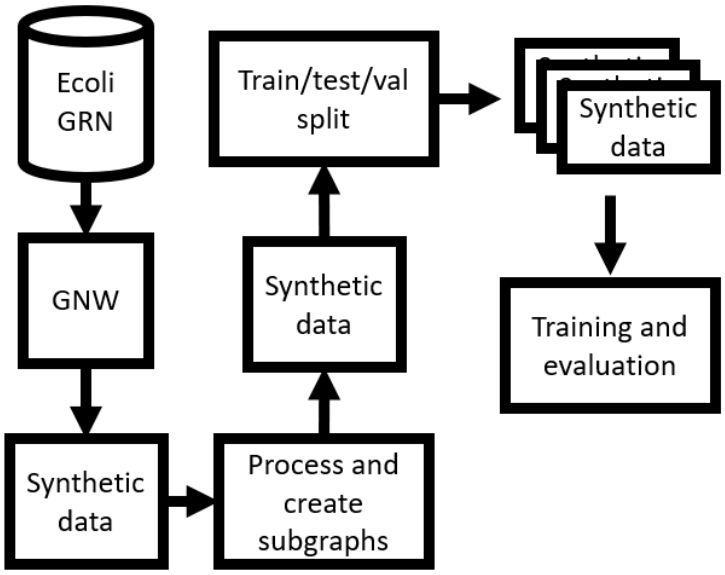


Figure 21: Synthetic data and model training pipeline.

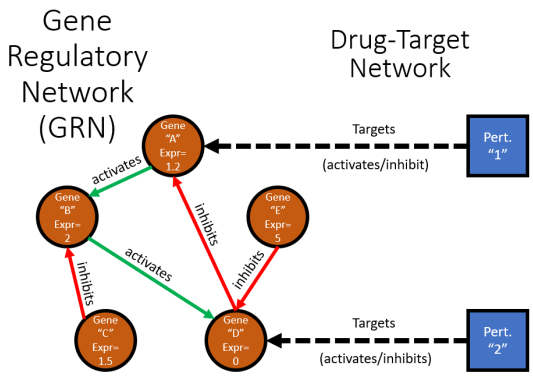
4.3.2 Graph Neural Network architecture

A novel GNN architecture was developed, arbitrarily named *waddle3*, tailored to this synthetic perturbation biology task. An example of the graph structure we use is shown in Figure 22a. This method operates on a bipartite graph of drugs and genes ("gene" and "protein" are used interchangeably in this section), where drugs can have directed edges to genes specifying the known protein target or perturbation action. Genes can have directed cyclic interactions with each other, but not with drug nodes. Gene-gene edge features ("activate" or "inhibit") are one-hot encoded. Unperturbed (i.e., baseline steady-state) expression features of each cell line are included as gene node attributes. Perturbation actions are encoded as a node attribute of the respective drug node such that nonzero values indicate an active or present perturbation. Genetic perturbations are assigned a value of 1 (indicating active genetic perturbation), and drug perturbations are assigned a scalar value representing the drug concentration. A major challenge to infer cell line

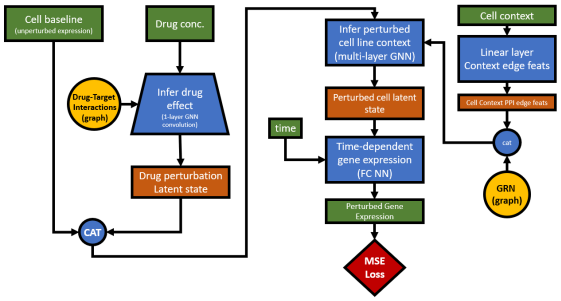
perturbation response is the contextual differences in gene-gene edge behaviors; For instance, two cell lines may have varying degrees of "flux" (i.e., activity) across a gene-gene interaction and that these contextual edge behaviors are rarely annotated in current literature. It is critical, therefore, for our algorithm to be capable of inferring cell-line specific edge behaviors. To address this, an interpretable learning mechanism learns cell-line specific edge scalars indicating the level of "activity" or "weight" a specific gene-gene edge has. This contextual edge weight is then concatenated to the edge-type attribute ("activate" or "inhibit" features). The *NNConv*, proposed by Gilmer et al. [GSR+17], is used for drug-gene convolution that passes information from drug nodes to drug-targeted gene node. The *TransformerConv*, proposed by Shi et al. [SHF+21], is used for gene-gene convolutions. To address potential limitations due to *oversmoothing*, we use the *PairNorm* normalization method [ZA20] at each layer of the Gene-Gene message passing. A graphical representation of our complete GNN algorithm is shown in Figure 22b. A pseudocode summary of our GNN algorithm is described in Algorithm 22b.

Algorithm 2 Perturbation Biology GNN (waddle3)

Input: Gene names, perturbation names, node and edge features, number of edges, network parameters
Output: Expression mean and variance predictions
procedure INFER_TARGET_PERTURBATION(x_pert, edge_index, edge_attr, batch)
    Calculate perturbation parameters and modify edge attributes
    Apply convolution and activation function
    return perturbation output
end procedure
procedure INFER_PERTURBED_CELL_LINE_CONTEXT(x, edge_index, edge_attr, edge_weight, batch)
    for each layer in GNN **do**
        Apply convolution and activation function
        if pair normalization is used **then**
            Apply pair normalization
        end if
    end for
    return perturbed cell line context
end procedure
procedure GET_CONTEXT_FEAT(context, edge_attr_sz0)
    Compute and reshape contextual edge features
    Apply sigmoid function and offset adjustment
    return contextual edge features
end procedure
procedure FORWARD(ppi, pti)
    $x\_pert \leftarrow$ INFER_TARGET_PERTURBATION$(pti.x, pti.edge_index, pti.edge_attr, pti.batch)$
    $x\_cell \leftarrow$ concatenate$(ppi.x, x_pert)$
    $cfeats \leftarrow$ GET_CONTEXT_FEAT$(ppi.context, ppi.edge_attr.size(0))$
    $x\_cell \leftarrow$ INFER_PERTURBED_CELL_LINE_CONTEXT$(x_cell, ppi.edge_index, ppi.edge_attr, cfeats, ppi.batch)$
    $time \leftarrow$ concatenate time feature to x_cell
    if use x_resid **then**
        $x\_cell \leftarrow$ concatenate original features to x_cell
    end if
    $\mu \leftarrow$ predict expression mean from x_cell
    $\sigma \leftarrow$ predict expression variance from x_cell
    return μ, σ
end procedure



(a) Graph example used in our GNN algorithm.



(b) Custom GNN architecture designed for synthetic perturbation biology.

Figure 22: GNN model architecture and graph information.

4.4 Results

To evaluate the proposed methods, we begin by demonstrating the rationality and utility of our synthetic data, and then evaluate the performance of our algorithm on the synthetic data. Finally, we will examine cell line specific predictions and show-case the ability of our model to learn auxiliary biological information such as drug binding affinity.

Synthetic Data

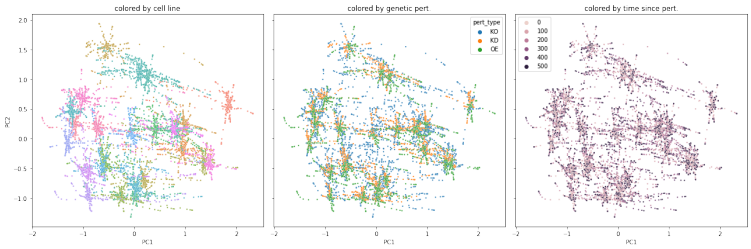
Synthetic data, produced using the process described above, is visualized in Figure 23 using Principal Component Analysis (PCA) [Pea01]. The results subjectively align with typical expectation of perturbation biology in that the main sources of variation are attributable to the cell line, perturbation, and time. Of particular note, cell states that diverge from the unperturbed state over time, however, cell lines are still the primary source of variance and cell-line clusters are present. Note that cell lines were created randomly and, therefore, we have no expectation of cell line similarity based on analogs of lineage or disease.

4.5 GNN prediction of synthetic expression time-series

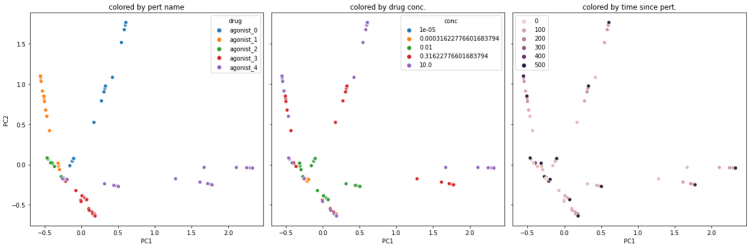
We apply our synthetic GNN to the task of predicting synthetic expression time series in response to various perturbations and report that our algorithm performs well, with R^2 of 0.895 when evaluated on heldout single agents and an R^2 of 0.765 on combination agents. Notably, the MSE evaluated on combinations is more than three times that of the MSE on single agents; this may suggest that our model performs worse on combination agents; however, combinatorial perturbations are also likely to induce more expression changes than single agents, which could explain this difference in performance.

Table 9: Results of GNN prediction of synthetic perturbation biology

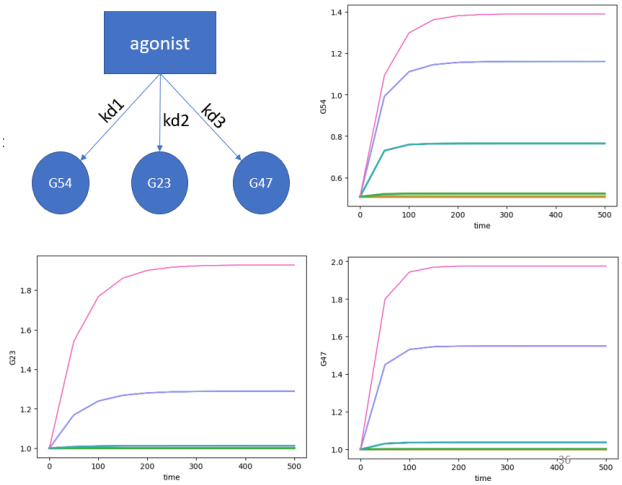
model	single agents		combinations	
	R2	MSE	R2	MSE
GNN (waddle3)	0.895	0.000481	0.765	0.00141



(a) PCA visualizations of synthetic data for multiple cell lines. Colored by cell line (left), perturbation type (center) and time since perturbation was introduced (right).



(b) PCA visualizations of synthetic data for a single cell line. Colored by drug ID (left), drug concentration (center) and time since perturbation was introduced (right).



(c) Graphical representation of an agonist and its targets and the resulting expression effect of the agonist perturbation.

Figure 23: Visualization of synthetic data generator outputs.

We also investigate the ability of our custom learning architectures to infer alternative aspects of perturbation biology, such as the binding affinity of drugs to specific genes. We compare learned drug-gene edge features to the true synthetic binding affinity parameters in Figure 24. From these figures, we can see a strong positive correlation between edge features and binding affinity in agonists and a strong negative correlation in inhibitors. These results suggest that our model encodes drug-gene *activation* using negative drug-gene edge weights, and encodes drug-gene *inhibition* using positive drug-gene edge weights. Interestingly, there are two agonist edges with positive edge weights (which we would expect to indicate inhibition) and three inhibitor edges with negative edge weights (which we would expect to indicate activation). Notably, all these edges have a large (i.e., weak binding) dissociation constant with values greater than or

equal to 0.1. This aspect may suggest that they have minimal perturbation effects in the model and would therefore explain the inaccuracies as having trivial impact on the system response.

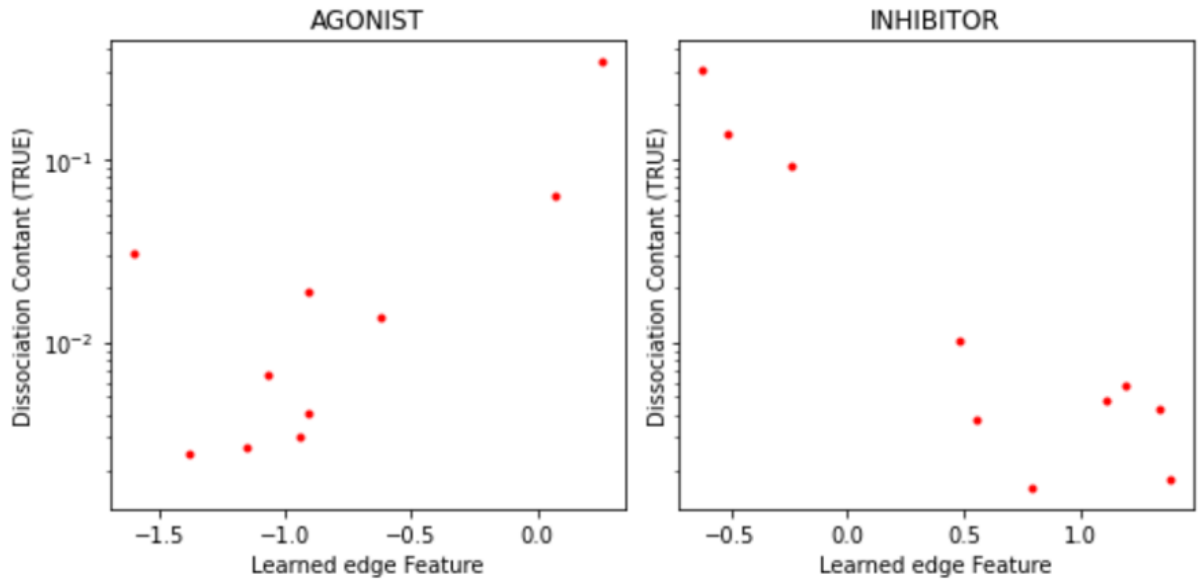


Figure 24: Learned synthetic binding affinity parameters.

This prediction task is formulated as an expression time series for multiple genes and is therefore a high-dimensional multioutput prediction problem. To inspect the performance at the level of individual genes, we randomly sample genes from the test set and plot the predicted vs. true response in Figure 25. From these results, it is clear that many genes are well predicted by our GNN algorithm; however, there are some genes that are not well predicted. In particular, there are many genes that have false positive response predictions, meaning that our GNN algorithm predicts a response when there should not be one. With that said, most of these false response predictions are relatively low-magnitude responses and may suggest a need for continued or refined optimization, for example, using learning rate decay during training. Another explanation is that this erroneous behavior may be due to *oversmoothing*, which is common in GNNs, and causes the nodes to have very similar latent representations to each other. Further research would benefit from investigating if these false positive predictions are more likely when there are neighbors with true response predictions.

We also sought to evaluate the rationality of our GNN time-series predictions. To do this, we randomly sample genes and plot the expression change over time. To further distinguish if our GNN algorithm predicts accurate cell-line specific responses, we plot the response of a specific cell line as well as the response of other cell lines. These examples are presented in Figure 26. From these results, it is clear that our GNN algorithm can effectively learn cell-specific responses and rational predicted behavior. For example, the response starts (time=0) in an unperturbed state and progresses to a perturbed steady state. It should be noted, however, that predictions are not perfect and there are several examples in which the predicted response correlates well with the true response but does not accurately capture the magnitude of response (e.g., 2nd row, 2nd column of Figure 26).

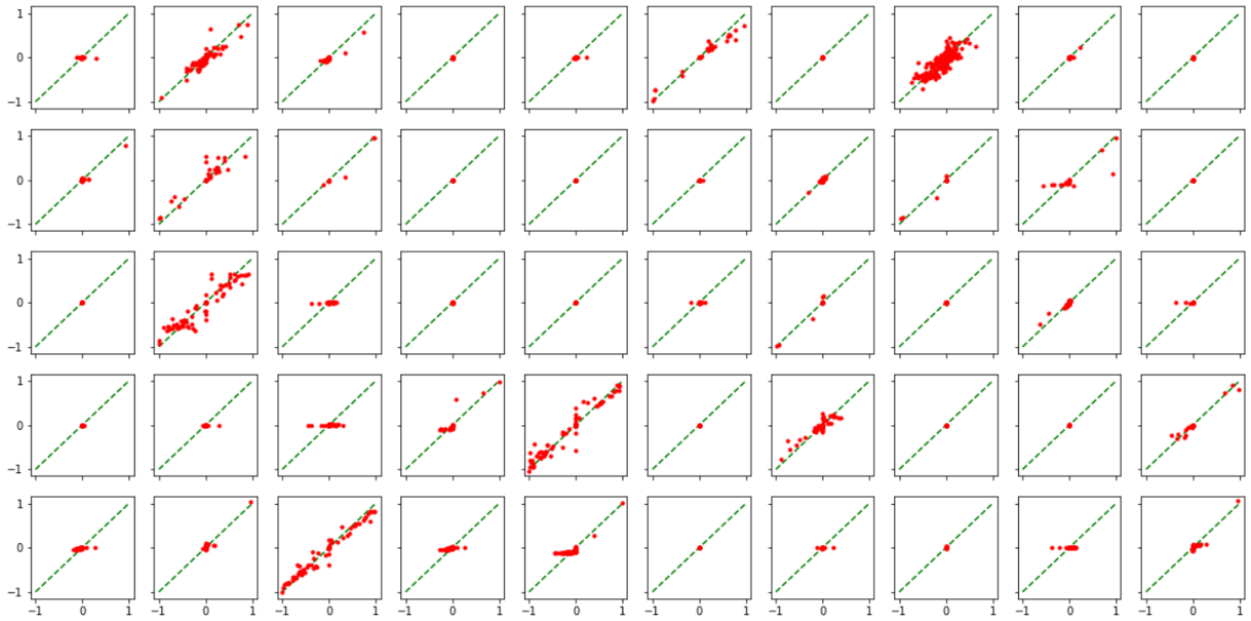
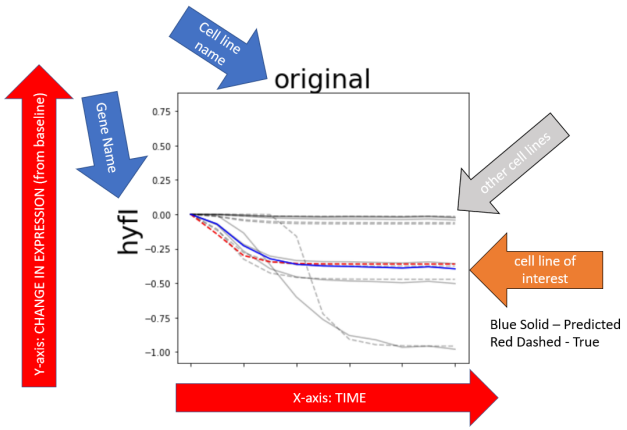


Figure 25: Randomly sampled gene nodes performance on single-agent drug perturbations. Predicted (x-axis) and True (y-axis) expression values.

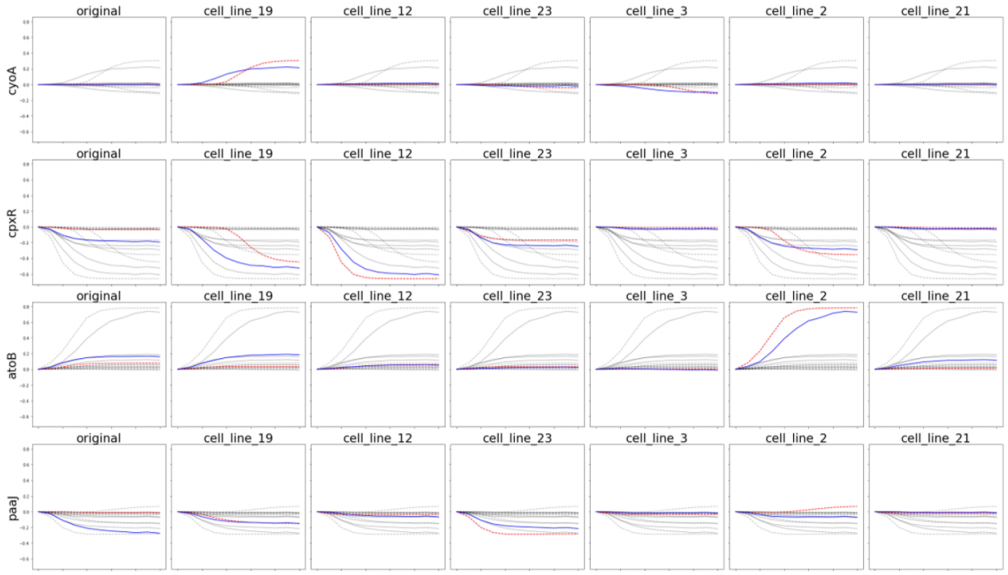
4.6 Discussion

In this work, we have extended the GeneNetWeaver package to produce a method to generate synthetic data that approximate perturbation biology. We then developed a custom GNN algorithm capable of predicting perturbed expression time series that operates on the gene regulatory network. We have shown that our GNN algorithm can accurately predict time-series expression and produces rational and smooth time-series predictions. Additionally, by creating custom learning architectures customized to this task, we can infer additional aspects, such as drug binding affinity parameters.

There are many limitations with this approach; most notably, we recognize that using ODE models derived from real-world GRNs have limited relevance to the true learning tasks in perturbation biology. GRNs assume that interactions between genes or proteins define expression changes; however, in many other real-world molecular interactions, such as protein-protein interactions, an interaction may not necessarily cause expression changes. For instance, the majority of post-translation modifications have no expectation of direct expression changes. This is a key difference as many biological graph prediction tasks utilize functional interactions or protein-protein interaction graphs. GRNs, therefore, can be expected to generate homophilous graphs, where perturbed genes are expected to be in a connected component. PPI or functional interaction (FI) graphs, because of the limited assumptions of interaction behavior, cannot be assumed to be homophilous. Due to this distinction, we believe that our synthetic data assumptions and the GNN algorithm may have limited utility in experimental data utilizing common biological knowledge graphs (PPIs, FIs, etc.).



(a) Figure annotation and descriptions.



(b) Randomly sampled gene expression time series (single-agent prediction).

Figure 26: GNN performance as node time-series prediction.

Another major challenge is that real-world experimental data are exceptionally sparse and noisy. In this work, we generated and trained our model on expression time series, available for all genes. In experimental data, time-series expression is particularly challenging to obtain, and the majority of large perturbation datasets (such as the LINCS L1000) usually only characterize the expression response at one or a few time points and for a limited subset of genes. Additionally, data sets such as the LINCS L1000 have sparse measurement across cell lines and drugs, which can make inference of accurate responses challenging. Moreover, the L1000 assay has many known data quality issues, which can limit the utility of experimental data.

Due to these limitations, it is likely that these methods have limited utility to real-world experimental data and learning tasks. Due to this, we made only limited efforts in the development and analysis of the *waddle3* method. Future research would benefit from a synthetic data generator that simulates a mix of direct expression regulatory interactions (such as transcription factors, miRNA interactions, etc.) and indirect expression regulation interactions (such as

post-translational modifications, complexes, etc.). In particular, such a method should focus on producing heterophilous perturbed expression graphs and focus on using sparse training data such as a single perturbed expression time point and sparse gene measurements.

5 Graph Neural Networks for prediction of experimental perturbation biology

5.1 Abstract

Computational models of drug response seek to identify relationships between patient characteristics and therapeutic response. Modern modeling methods, such as deep learning, accurately predict drug response in many settings; however, these methods lack interpretability and are not guaranteed or even expected to make predictions based on causal or rational biological logic. In this work, we investigate whether graph neural networks can be developed to 1) leverage heterogeneous forms of prior biological knowledge for improved prediction of perturbation biology and 2) develop interpretable deep learning models useful for precision oncology research. To explore these questions, we propose a novel drug response model to improve the transparency (e.g., interpretability or inspectability of internal latent states) and the rationality (e.g., encourage model logic to emulate true biology) of predictions. To achieve this, we use a graph neural network (GNN) that operates on a curated biological network that represents drug binding, protein signaling transduction, and gene regulation. Model parameters are trained using experimental measurements of drug-perturbed RNA expression changes (LINCS L1000). We evaluate the ability of learned embeddings, which are designed to represent aspects of drug action, protein behavior, or cell line state, to understand the respective entities' function or importance. Drug response predictions can be traced backward through the biological network to understand the prediction logic in terms of drug- and cell-specific gene regulation and protein signaling. We show that this algorithm enables novel forms of interpretability. While the performance of this model is comparable to that of traditional neural networks, the performance is not always affected by randomization of prior knowledge, suggesting that some experiments do not effectively leverage prior knowledge to improve performance. In its current state, our developed GNN models are not conclusively useful for application to perturbation biology tasks, however, we highlight future research directions that may overcome current limitations. Future research in this work has application to precision cancer, mechanistic deep learning, and drug repurposing.

5.2 Introduction

In this chapter, we seek to build on the lessons learned from our work in Chapter 4, where we applied GNNs to synthetic perturbation biology.

> The goal of this chapter is to develop graph neural networks (GNNs) suitable for application to **experimental** perturbation biology and drug response prediction tasks. We hypothesize that including prior knowledge of molecular interactions in deep learning will improve model performance, aid interpretability, and encourage mechanistic prediction logic.

There are several notable distinctions between the assumptions and data types used in Chapter 4 to produce synthetic data and those appropriate for real-world cancer perturbation biology. In the discussion of Chapter 4, we noted the limitation that the synthetic data is produced based on gene regulatory networks (GRNs), which can be assumed to be

homophilous with respect to gene expression changes. In this chapter, we plan to use functional interactions (FI) as the source of prior knowledge, which comprise a wide range of relationships between molecular entities. These relationships could be expression regulation or protein-protein interactions (PPIs) that do not directly impact expression changes, such as protein-protein post-translational modifications or complex associations. FI prior knowledge, therefore, cannot be assumed to be homophilous with respect to expression changes. Additionally, in Chapter 4 we operated on largely accurate and complete prior knowledge; however, real-world data is sparsely annotated and false or missing molecular interactions are likely to introduce challenges. In particular, we are concerned with the potential sparse limitations of transcription factor regulons, which are likely to be critical for the prediction of expression changes. To mitigate this issue of sparse prior knowledge annotations, in this chapter, we will shift from a node-prediction GNN task (as implemented in Chapter 4) to a graph-prediction task such that we use a GNN encoder in combination with a fully connected linear decoder to predict individual gene responses. In Chapter 4, the model was trained on time series expression from all genes in the GRN network. In real-world perturbation datasets, such as the LINCS L1000, we have limited gene annotation (978 genes) with perturbation measured at only a few time points. To highlight the sparsity of molecular measurements, if we assume $\tilde{2}$0k human genes, than we had $\tilde{5}\%$ measurement annotation. Furthermore, the majority of chemical perturbations in the LINCS dataset are measured at 24 hours, which leads to a sparsity of measurements in the temporal domain. These real-world limitations necessitate a shift in our approach compared to Chapter 4: Rather than modeling experimental data as a time series, which is inherently compute intensive and requires large data volumes, in this chapter we will predict only a single time point and focus on local functionally relevant biological subnetworks.

For a review of related work, refer to Sections 1.13 and 1.14. To our knowledge, GNNs have not previously been applied to mechanistic prior knowledge for use in perturbation biology or cell signaling.

5.3 Methods

We present a deep learning architecture called Graph Neural Networks for Cancer Drug Response (GNNCDR) that incorporates molecular interactions curated from the literature into deep learning algorithms, thus encouraging prediction logic to emulate the causal biological behavior of the cancer drug response. We do this using a GNN that operates on a heterogeneous input graph. Our input graph consists of drug and gene nodes, where a subset of genes are also transcription factors. For simplicity, we collapse proteins, RNA transcripts, and genes into a single node, which we refer to as a "gene" node. The *gene-space* is the subset of gene nodes that we include in our model, the *drug-space* is the subset of drug nodes included, and the *cell-space* is the subset of cell lines that are modeled. Drug nodes have two primary features: a trainable node embedding (z_{drug}) and a concentration parameter that encodes the presence or absence of a drug. Gene nodes similarly have a trainable node embedding (z_{gene}) and cell-line specific features consisting of gene expression, mutation, methylation and copy number variation.Drug-target interaction (DTI) edges have a univariate trainable feature (z_{DTI}), scaled between zero and one, and are intended to learn the binding affinity of DTIs. Gene-gene edges have a multivariate embedding (z_{FI}).

Figure 27 provides an overview of our drug response prediction algorithm and evaluation strategies. Each sample prediction is performed by constructing a cell-line- and perturbation-specific graph. Cell line 'omic features, drug concentration, and trainable node and edge features are overlaid onto a literature curated biological network that describes the interactions between drugs, genes, and transcription factors (TF). We then use a custom GNN architecture to predict the expression changes caused by a given perturbation on a particular cell line. Following model training, we evaluate whether the model mimics true biology by inspecting several elements of the GNNCDR model. Respective to the numerical labels in Figure 27:

1. The drug embedding is compared against drug mechanism-of-action annotations collected from literature, as described by the CLUE drug repurposing resource [CBL$^+$17].

2. Gene embedding is compared to the gene ontology (GO) molecular function terms [ABB$^+$00].

3. The gene-gene functional interaction embedding (FI edges) are compared to literature curated FI interactions from the Reactome FI resource [GJS$^+$21, WH17]

4. The drug-gene features are compared to binding affinity datatypes in the Cancer Targetome [BCKM$^+$17] including dissociation constant , inhibitory constants , and half-maximal inhibitory constant

5. The cell embedding is compared to cancer cell line encyclopedia (CCLE) annotations [GHJV$^+$19] including disease, subtype, metastasis status, and cell lineage.

6. The prediction behavior of our Graph Neural Network (GNN) is inspected using Shapley values [S$^+$53], and we compare the results with the expected biological behavior.

7. We compare the cell-line specific gene-regulation weight matrices to literature curated transcription factor (TF) regulons as described in the Dorothea resource [GAITSR18]. These weights describe the linear relationship between the inferred TF perturbation and the gene expression perturbations.

Importantly, all these datatypes, except for (4) can also be used to pretrain the respective embedding prior to model training on LINCS. This approach allows users to incorporate heterogeneous forms of prior knowledge that may aid model performance.

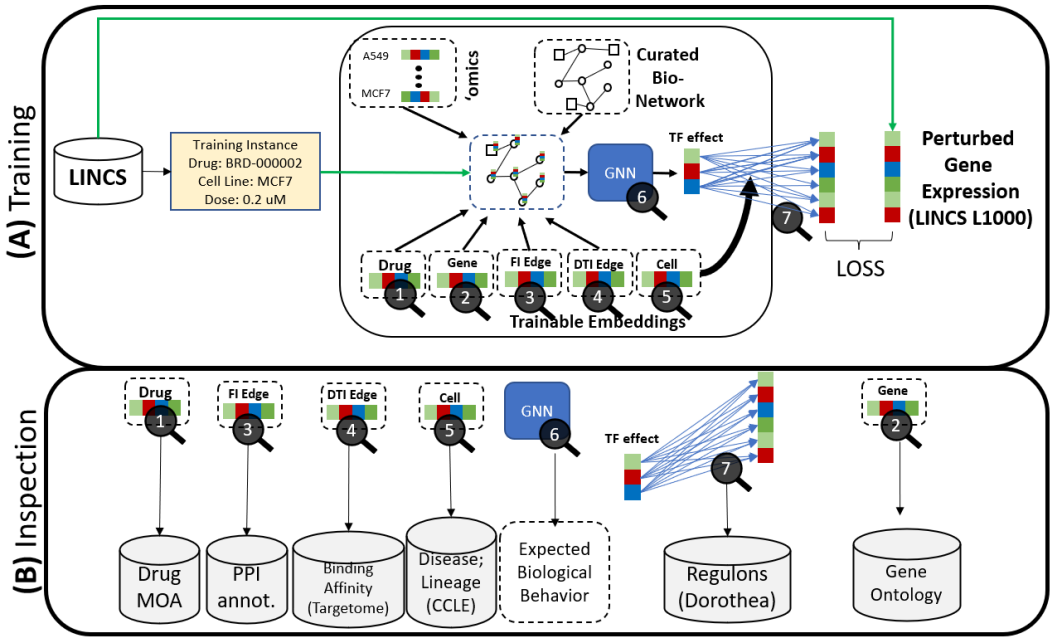


Figure 27: Overview describing the training (upper box) and inspection or pre-training (lower box) of our proposed GNNCDR deep learning model.

We have introduced two custom graph convolutions, *DrugConv* and *CellConv*, to model direct drug effects and the subsequent downstream signaling cascade. The final layer of our model is linear, mapping the perturbation of transcription factors (constrained to values between -1 and 1) to changes in gene expression. To model contextual differences in transcription factor activity, this linear layer is tailored to individual cell lines through a cell line embedding (z_{cell}). These trainable embeddings can be used in two ways:

- Inspect embeddings after optimization to investigate their information content relevant to specific aspects of the underlying biology.
- Pretrain the embeddings to provide information that we expect will improve model prediction performance. This is analogous to a transfer learning approach.

The sources used for embedding inspection and pretraining are described in Table 10.

5.3.1 Data.

To construct a gene-gene graph, we use the Reactome functional interaction network [GJS+21]. The Reactome FI network is a scale-free network, with several high-degree outliers. The degree distribution is presented in Figure 28. There are more than 12,000 genes in the Reactome FI network. To avoid the inclusion of irrelevant genes and to minimize compute requirements, we chose to focus on a specific set of pathway genes that are likely to be involved in drug response. We also filter out all 'predicted' edges from the Reactome FI graph, thus focusing on high-confidence, literature-curated functional interactions.

The choice of gene-space has three main considerations:

- Relevant drug-target inclusion
- Relevant inclusion of genes involved in drug signal transduction
- Relevant transcription factor inclusion

We use the Reactome pathways to choose the gene-space (i.e., the biological sub-graph). The Reactome pathways are gene sets curated by experts and reviewed in the literature relevant to specific biological processes. Therefore, we believe that pathways are likely to include the relevant molecular entities for a given process, and therefore careful manual selection of processes related to drug response should produce a functionally relevant biological sub-network. There are computational limitations of our modeling approach: large input graphs take significantly longer to train and require more memory. We limit our graph size to less than 3000 gene nodes to ensure hardware sufficiency. The cell-space was chosen based on the overlap of cell lines in LINCS and those with 'omics annotation in the cancer cell line encyclopedia (CCLE) [GHJV+19]. For a drug to be included in our drug-space there must be at least one drug-target annotation, and the target must be included in our gene-space. For drug target annotation, we use the CLUE repurposing hub dataset [CBL+17] and the Cancer Targetome resource [BCKM+17].

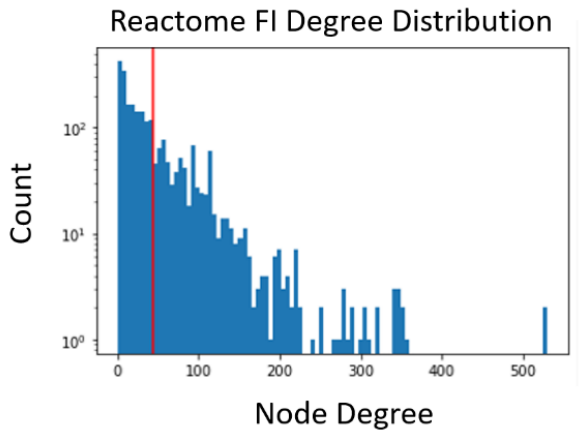


Figure 28: Reactome FI network degree distribution. The linear relationship of node degrees in log-space indicates a scale-free network. The red line indicates average node degree (44.)

5.3.2 Graph Neural Network for Cancer Drug Response (GNNCDR)

During the process of developing GNN algorithms suitable for perturbation biology and drug response, a key design trade-off was to incorporate useful prior knowledge (known molecular interactions, drug MOA, gene function, etc.) while not over-constraining the algorithm. Current biological knowledge bases have many sources of bias and sparse annotation, meaning that if we force the model to rely *only* on the available prior knowledge, it would likely be over-constrained. On the other hand, building mechanisms that allow for more flexible learning may under-constrain

the system and result in inaccurate prediction logic. In this work, we chose a two-stage prediction mechanism; First, we use a GNN-encoder to infer the perturbation of known transcription factors. In this GNN-encoder, we apply strong constraints such that the GNN cannot infer new interactions and therefore must rely on only the known prior knowledge of molecular interactions. In the second stage, we implement a flexible cell-line specific linear decoder module such that each transcription factor can predict any gene expression response. We rationalize this design decision because gene-regulatory interactions are likely to be highly contextual and sparsely annotated in the available knowledge bases.

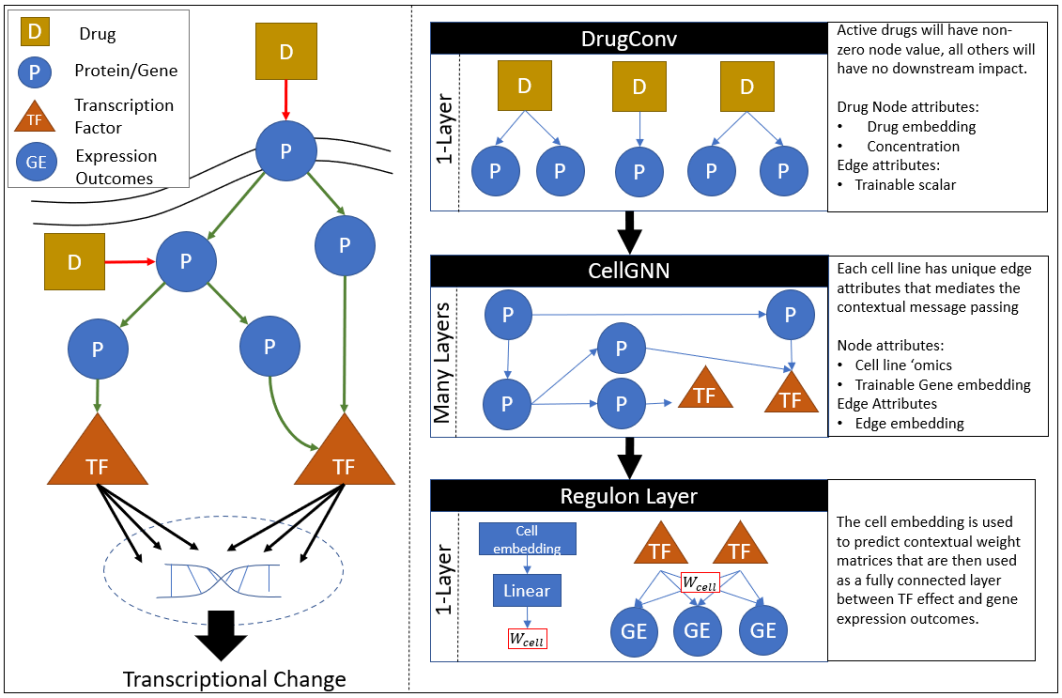


Figure 29: (Left) Graphical representation of the molecular biology of targeted drug-response. (Right) The three primary modules in our model, which emulates our biological premise of biological signal transduction and allows for explainable drug response predictions. (Right, Top) DrugConv module is a bi-partite graph with edges from drug nodes to protein nodes. Only drug nodes with non-zero concentration attributes will lead to non-zero protein perturbations. The drug effect is parameterized by a trainable drug embedding, intended to capture drug mechanism-of-action (e.g., agonist vs inhibitor) and a trainable edge scalar, which captures the strength of drug effect and analogous to binding affinity. (Right, Middle) CellGNN module is responsible for propagating the drug-perturbed protein signal (from DrugConv), between proteins and transcription factors. (Right, Bottom) A fully connected linear layer maps transcription factor perturbations to gene expression changes (endogenous variable).

5.3.3 Drug-Target Interaction convolution (DrugConv).

To model the effect of a drug on its protein target, we propose a graph convolution termed *DrugConv*, which is described in Figure 30. This layer operates on a bipartite graph of drugs and genes, with directed edges from drug to gene. The drug effect is modeled by a drug mechanism-of-action (MOA) vector (z_{moa}), which is retrieved from a training drug embedding and scaled by two inputs, z_{ij}^{aff} and the concentration c_i. z_{ij}^{aff} is a trainable parameter between [0,1] specific to each edge of drug i to gene j, and intended to capture binding affinity information. We scale

the concentration parameter such that it's log-linear with respect to concentration in the relevant concentration range, and yet maintain that a value of zero indicates no drug. In this convolution, all drugs with concentrations of zero will not have an effect on downstream genes.

$$c_i = \frac{log_{10}(uM) - log_{10}(\epsilon)}{log_{10}(\epsilon)}$$

$$x_j = z_{ij}^{aff} * c_i * softmax(z_i^{moa})$$

Note, we apply the Softmax transformation on the drug embedding to ensure that all drugs are scaled similarly, which is intended to discourage the drug embedding from capturing binding affinity information.

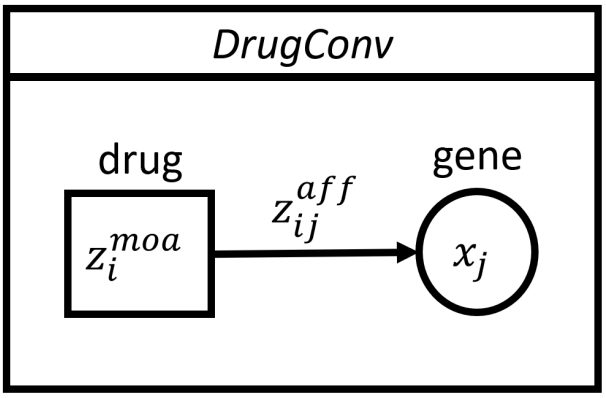


Figure 30: The DrugConv graphical depiction. z_{ij}^{aff} is a trainable scalar between 0 and 1 representing effect strength. z_i^{moa} is a multivariate trainable embedding intended to capture drug mechanism-of-action, such as inhibitors vs agonists.

5.3.4 Functional Interaction convolution (CellConv)

To model signal transduction, we propose a graph convolution termed *CellConv*, which is described in Figure 31. The CellConv takes the output of DrugConv and propagates information from gene-to-gene over many layers. This convolution has a unique first step that calculates and assigns cell contextual edge features that are used in all subsequent layers. Each layer passes information between neighboring nodes in a manner consistent with our biological premise. In the first step, the model assigns the following:

- An edge function vector (W_{ij}^{func}) which is a matrix R^{lxl}, where l is the number of channels. This matrix is intended to capture the function or mechanism of each edge (e.g., activation, inhibition, complex, etc.).
- A cell-line specific univariate edge importance value ($\beta_{ij} \in [0, 1]$). This feature is used to describe the contextual importance of an edge. A β_{ij} value of 0 will prevent information from passing from gene node i to gene node j.

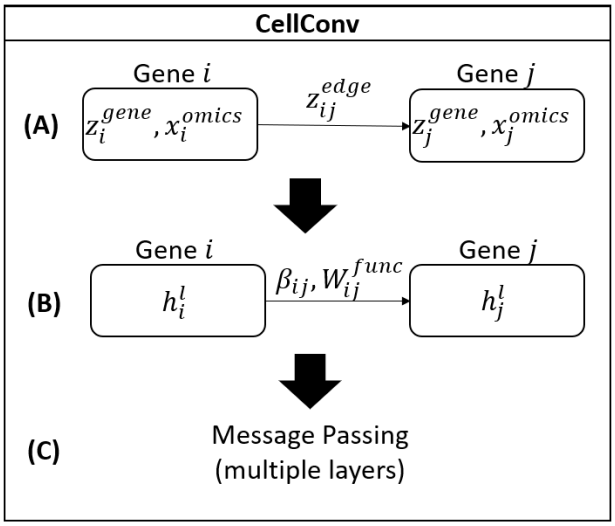


Figure 31: The CellConv graphical depiction. (A) depicts the cell context prediction convolution, which predicts edge features B_{ij} and W_{ij}^{func}, which are then used in all subsequent message passing layers. This convolution operates in two steps: In the first step, two latent edge features are generated: (β_{ij}) a univariate cell contextual feature, scaled [0,1], which is used to model edge importance or edge flux, and (W_{ij}^{func}) a non-contextual weight matrix intended to capture edge function or type. These latent edge features are then used in all subsequent layer's message passing.

The features β_{ij} and W_{ij}^{func} are then used to update the latent state of the gene node in each successive layer, as described in panel B of Figure 31. We rationalize that this approach emulates our biological premise by defining a graph capable of capturing distinct molecular functions, specific to each set of interacting genes. Furthermore, we allow for contextually unique graphs, thus allowing differences in signal transduction based on cellular background. The terms β_{ij} and W_{ij}^{func} are described below, where f_θ, f_ψ are parameterized by a neural network. z^{gene} is a trainable embedding, unique to each gene node. z^{edge} is a trainable embedding specific to each gene-gene edge. x^{omics} are cell line 'omic features, specific to each gene.

$$\beta_{ij} = sigmoid(f_\theta(z_{ij}^*))$$

$$z_{ij}^* = cat(z_i^{gene}, x_i^{omics}, z_j^{gene}, x_j^{omics}, z_{ij}^{edge})$$

$$W_{ij}^{func} = f_\psi(concat(z_i^{gene}, z_j^{gene}, z_{ij}^{edge}))$$

$$h_j^{l+1} = tanh(\sum_{\mathcal{N}(j)}^{n} \beta_{ij} w_{ij}^{func} h_i) + h_j^l$$

5.3.5 Regulon Module

The previous convolutions were intended to infer the perturbation of transcription factors (TFs) due to predominately post-translational cell signaling; in this section, we define a *regulon module*, which is intended to learn which genes a given TF regulates (often referred to as the "regulon" of a TF). A key motivation for this module is that public

knowledge bases have limited regulon annotations and therefore we were concerned that using strong prior knowledge constraints may over-constrain the model and degrade performance. Additionally, it is well known that a TF regulon varies from cell to cell based on cellular state (chromatin organization, TF cofactor expression, methylation of DNA, etc.). Therefore, we aim to design a learning mechanism that can i) infer novel TF-DNA targets and ii) learn cell-specific TF-DNA targets. To achieve this, we define two key elements illustrated in Figure 32:

1. cell embedding (z_k^{cell}) that aims to capture cell-specific regulon information
2. TF-LINCS linear layer that models TF impact on expression as a linear effect based on the perturbation of a given TF

The cell embedding is used as input into a linear layer (f_ϵ) to predict a set of regulon weights (w_{lm}^{reg}). The latent state of the transcription factor (h_l; scalar; [-1,1]) is then treated as input to a linear model to predict the output of gene expression change ($\hat{y}_m$), described by:

$$\hat{y}_m = (w_{lm}^{reg})^T h_l$$

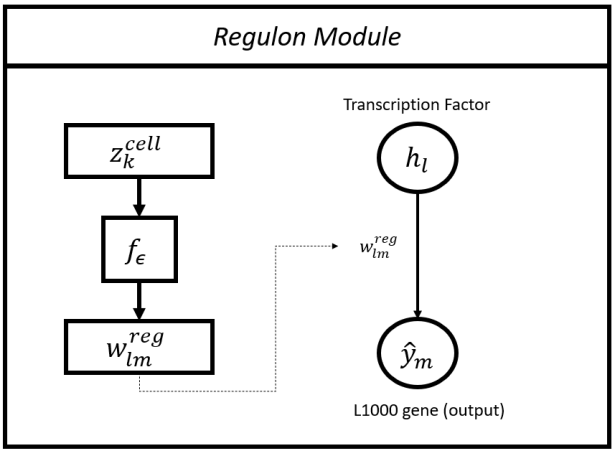


Figure 32: The GNNCDR regulon module.

5.3.6 Trainable Embeddings

In this model, we use a trainable embedding to encapsulate specific types of information. The gene embedding is used as feature of the gene node and is intended to capture information related to gene type, function, and importance to our predictive task. The drug embedding is used as drug node features and is intended to capture drug mechanism-of-action such as agonism or inhibition. There is also a univariate drug-to-gene edge embedding, which we use to capture the magnitude of drug effect (i.e., binding affinity information). The gene-to-gene edge embedding is used to delineate the importance, function, and direction of an edge. Lastly, a cell embedding is used to capture contextual regulons, or more accurately: the cell embedding is used to predict a linear layer that maps transcription factor perturbations to gene expression changes. The use of trainable embeddings allows the model to learn information related to our overall prediction task, but effectively encapsulates different types of information to specific embeddings, which can then be

investigated post-hoc to understand model behavior, explain predictions, and distinguish biological characteristics. Additionally, by choosing relevant datatypes, each embedding can be pre-trained to encode information we expect to be relevant to the prediction task. Table 10 outlines the various datatypes we use for pretraining or comparison (note: we never do both pretraining and comparison as this would be self-fulfilling). Pretraining allows us to incorporate alternative forms of information, while still allowing learning through modification of the embedding during training. To ensure that the pre-trained embeddings can be utilized effectively during target task training, we pre-train using a variational approach that encourages the embeddings to be normally distributed and uncorrelated. We rationalize that this is likely to improve downstream training.

Table 10: The datatypes and information sources used for either a) post-training embedding inspection or b) pre-training of embeddings.

Feature	Embedding	Evaluation/Pretraining	Example	Pre-trainable	Source
Drug Node	z_{drug}	Mechanism-of-action	Agonist, inhibitor	Yes	CLUE [CBL$^+$17]
Gene Node	z_{gene}	Molecular Function	Regulation of binding	Yes	Gene Ontology [ABB$^+$00]
Gene-Gene Edge	z_{FI}	Functional edge annotation	activates, inhibits	Yes	Reactome FI [WH17, GJS$^+$21]
Drug-Gene Edge	z_{DTI}	Binding Affinity	K_d = 100nM	No	Cancer Targetome [BCKM$^+$17]
Cell Context	$z_{cell}	Lineage, Disease, Subtype	LUNG, AML	Yes	DepMap [SNC$^+$17b, STK$^+$20]

5.3.7 Baseline Models.

We consider several baseline algorithms:

- NaïveNN: As shown in Figure 33, this is a simple multi-layer perceptron which learns a drug and cell embedding and takes as input the drug dosage. This is intended to capture the performance of a model that is naïve to drug target, protein functional interactions and regulon information.

- Null Model: this is a NaiveNN model trained on samples with shuffled (X,y) labels and intended to capture the performance of a random model.

- GNNCDR Input graph permutation

  - The “random” GNNCDR model, the edges are randomly chosen. This maintains the number of edges but does not maintain node degree. This may introduce isolates based on random sampling.

  - The "Rewired" GNNCDR model, is identical to the GNNCDR except that the gene-gene edges and the drug-gene edges have been shuffled, to create a permuted network. This shuffling maintains the number of edges and the degree of the node. Table 11 outlines the relevant characteristics of each graph permutation approach.

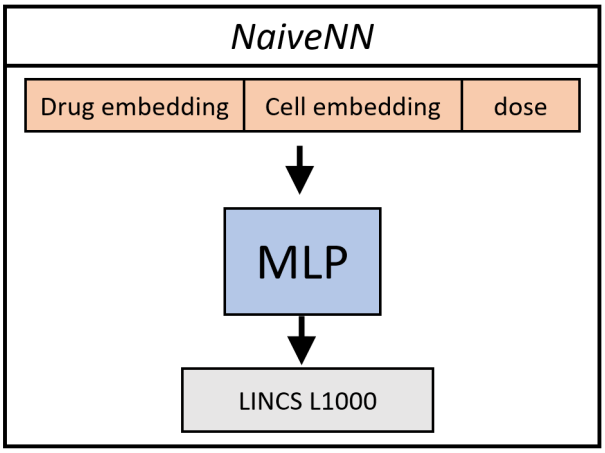


Figure 33: *NaiveNN* architecture. The drug and cell embeddings are trainable parameters intended to capture representative aspects of drug and cell interactions. MLP represents a two-layer neural network. The LINCS L1000 represents multioutput prediction of gene expression changes at 24 hours after introduction of the perturbagen. Dose represents the drug concentration.

Table 11: Comparison of graph randomization strategies. The average proportion of true edges was obtained by permuting the graph 100 times and computing the proportion of true edges that were maintained after permutation.

Graph permutation	Subgraph	Prob. True Edge (gene-gene) [95% CI]	Prob True Edge (drug-gene) [95% CI]
"Random"	R-HSA-9006934	0.0690 [0.065, 0.07]	0.0002 [0,0.01]
"Rewire"	R-HSA-9006934	0.1942 [0.19, 0.20]	0.0458 [0.03, 0.07]
"Random"	R-HSA-162582	0.0182 [0.018, 0.019]	0.0008 [0, 0.003]
"Rewire"	R-HSA-162582	0.089 [0.088, 0.091]	0.0287 [0.023, 0.035]

In this work, we explore two experiments that operate with different hyper-parameters and on unique subsets of the available data. Table 12 describes the network characteristics and training partitions used in each experiment. Additionally, we tailor the algorithm hyper-parameters to each experiment, and report the experiment hyper-parameters in Appendix section 9.5.

Table 12: GNNCDR Experimental information.

Experiment	Subgraph	Subgraph Name	#Genes	#Drugs	#gene-gene edges	#drug-gene edges	#TFs	#obs [train/test]
1	R-HSA-9006934	Signaling by Receptor Tyrosine Kinases	551	330	21304	556	30	29119/7951
2	R-HSA-162582	Signal Transduction	2392	907	105026	1820	129	62994/16047

5.4 Results

This section will evaluate model performance against several key baselines including traditional neural networks and randomized prior knowledge. Additionally, this section showcases a number of ways that the described GNN

architecture can be used for interpretation and evaluation, and compare specific learned weights to expected behaviors to evaluate if the model has utility and rational behavior.

5.4.1 Model Performance

To evaluate the performance of the model, we report the mean squared error (MSE) and the R^2 (variance weighted multi-output) score in Table 13 measured in a holdout test set (unique drug, cell line pairs not present in the training set). In Exp. 01, the GNNCDR model with no pretraining performed best, out-performing the NaiveNN model, the GNNCDR models trained with inaccurate biological networks ("rewire" and "randomized") and the average response of each drug across cell lines. This result suggests that, for Exp. 01, the GNNCDR model effectively leverages the prior knowledge in our literature curated biological network to improve predictive performance on perturbation biology. Additionally, the GNNCDR model markedly outperforms the average drug response across cell lines, which suggests that we are effectively predicting cell line contextual response. Interestingly, pretraining the GNNCDR model embeddings with alternative data types did not improve the performance as we had expected. In Exp. 02, the NaiveNN and the "rewire" GNNCDR models out-performed the GNNCDR models (pretrained and no pretraining) trained on the true biological network. These results suggest that the prior knowledge encoded in our biological network was not useful to GNNCDR prediction. Unexpectedly, the GNNCDR model trained on inaccurate prior knowledge ("rewire" network) out performed the NaiveNN model, and this may suggest prediction advantage of the GNNCDR model may be attributable to a mechanism other than the prior knowledge. Alternatively, this result could also indicate that the biological network characteristics, namely the degree distribution (which was maintained by the "rewire" permutation method), are still "useful" to prediction of perturbation biology.

It was also surprising to see that pretraining the GNNCDR embeddings improved prediction in models operating on inaccurate biological networks ("rewire" and "random") in both experiments. We had expected that randomizing the biological network would destroy the relational significance of gene-gene interactions and thus prevent the usefulness of gene and edge pretraining. However, the cell line embedding pretraining is likely to be useful to model predictions independent of the network structure. The cell line pretraining may be particularly useful in a permuted network where the node 'omic features are functionally randomized.

We acknowledge that we have performed limited model comparisons and that further evaluations, including hyperparameter tuning, should be performed to further understand the performance of the respective algorithms. In the Appendix section 9.5 we describe the hyper-parameters used for these experiments. Additionally, we note that the choice of prior knowledge, particularly the choice of biological network subgraph, is likely to be critical to GNNCDR performance and further work should be invested in understanding the key molecular entities and gene relationships that may be useful for prediction of perturbation biology.

Table 13: The performance of the GNNCDR compared to baseline algorithms evaluated on a hold-out test set of the LINCS L1000 dataset.

(a) EXP 01 "Signaling by Receptor Tyrosine Kinases" (R-HSA-9006934) Model performance.

Model	Network	Pretrained	R2 (var. weighted)	MSE
GNNCDR	**True**	**False**	**0.467**	**1.954**
GNNCDR	True	True	0.429	2.095
GNNCDR	"rewire"	False	0.344	2.407
GNNCDR	"rewire"	True	0.380	2.274
GNNCDR	"random"	False	0.033	3.547
GNNCDR	"random"	True	0.279	2.646
NaiveNN	NA	NA	0.398	2.209
NaiveNN (x,y shuffled)	NA	NA	-0.018	3.734
Avg. Response by drug	NA	NA	0.165	3.089

(b) EXP 02 "Signal Transduction" (R-HSA-162582) Model performance.

Model	Network	Pretrained	R2 (var. weighted)	MSE
GNNCDR	True	False	0.474	2.055
GNNCDR	True	True	0.477	2.047
GNNCDR	"rewire"	False	0.452	2.142
GNNCDR	**"rewire"**	**True**	**0.485**	**2.012**
GNNCDR	"random"	False	0.377	2.436
GNNCDR	"random"	True	0.439	2.194
NaiveNN	NA	NA	0.479	2.039
NaiveNN (x,y shuffled)	NA	NA	-0.053	4.117
Avg. Response by drug	NA	NA	0.180	3.243

5.4.2 Performance by drug, cell-line, and quality metrics

In Figure 34 we explore the Exp. 02 GNNCDR (no pretraining) when grouped by performance and performance trends compared to quality metrics. Somewhat unexpectedly, the majority of drug performances were quite low, with 50% of the drugs having a R^2 of less than 0.04. Furthermore, 50% of the cell line grouped performances had an R^2 less than 0.15. R^2 performance grouped by gene ranged from 0 to $\tilde{0}$.8. with a median of 0.39. These results suggest that, while

many drug, cell-lines and genes have fairly strong performance, many of these groupings perform marginally and suggest that utility of the model will depend on the specific drug, cell-line or genes for which we are predicting. In the middle row of Figure 34 we plot the performance against the number of training observations, and the results of which suggest a moderate trend between number of observations and the performance of a given grouping. In the bottom row of Figure 34 we plot performance against quality metrics. The Average Pearson Correlation (APC) metric was first proposed by Pham et. al [PQZ$^+$] and is calculated by measuring the pairwise Pearson correlation of LINCS L1000 level 4 replicates. Intuitively, low APC values indicate discordance in response between performance. As seen in the left and middle plots of the bottom row, there is a strong correlation between GNNCDR performance and average APC metric, particularly in the drug-grouped performances (middle left). In the right plot of the bottom row, we use the standard deviation of genes, computed within the LINCS L1000 level 4 control replicates, and compare to GNNCDR performance. Interestingly, there appears to be a trend between performance and gene standard deviation within the standard deviation range of 0 and $\tilde{1}.5$, however, the trend disappears after $\tilde{1}.5$. These results suggest that drug, gene and cell-line specific data quality is significantly variable and critical to effective prediction performance.

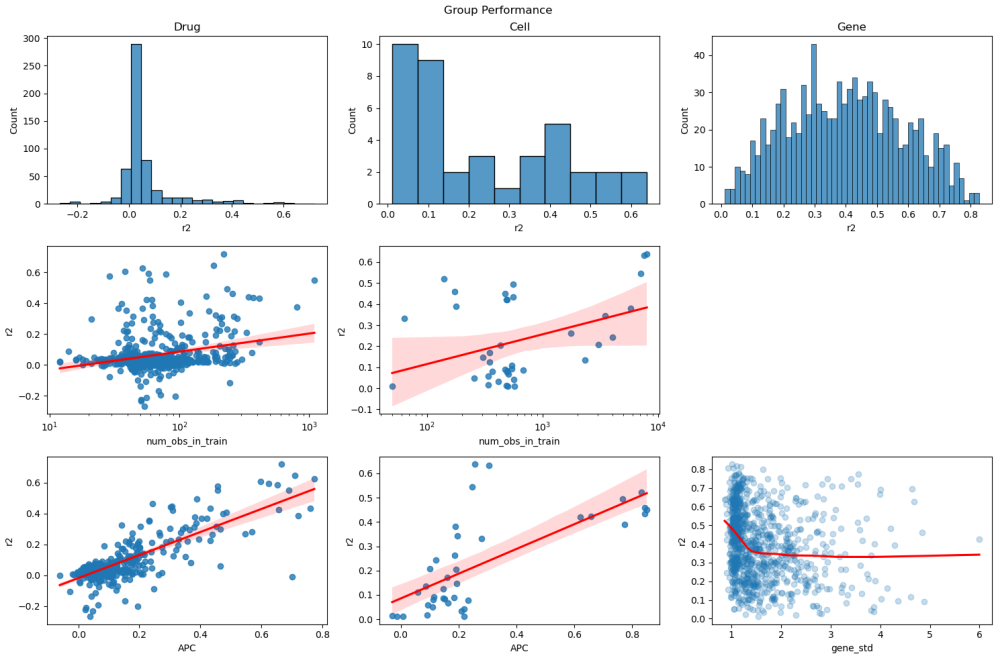
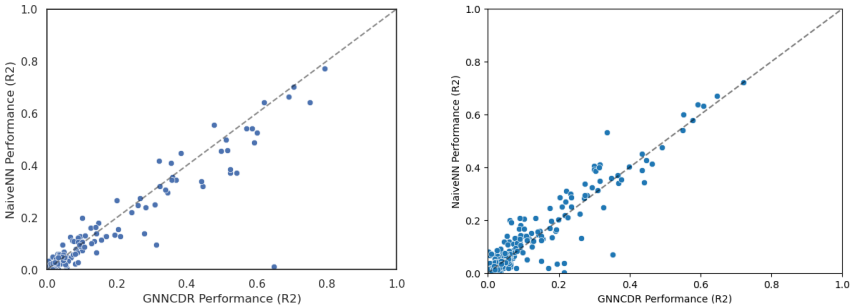


Figure 34: Exp. 02 Performance grouped by Drug (left column), cell line (middle column) and gene (right column). The top row characterizes the distribution of performances reported by R^2. The middle row shows performance trends by the number of training observations for each grouping. The bottom row reports the performance trends by quality metrics. Average Pearson Correlation (APC) of LINCS level 4 replicates are used for drug and cell line groupings. Standard deviation within LINCS level 4 control replicates are used for the gene grouping.

To evaluate the performance of individual drugs, compared between the GNNCDR model and the NaiveNN model, we group test observations by drug and calculate the R^2 score and plot the relationship in Figure 35. In Exp. 01, the GNNCDR model performed better than the NaiveNN model for 111/183 (61%) drugs in the test set (note: drugs with

fewer than 10 test observations were not evaluated). In Exp. 02, the GNNCDR model performed better than the NaiveNN for 197/422 drugs (47%). These results suggest that the GNNCDR algorithm may have some performance advantage in Exp. 01 but a limited advantage in Exp. 02. We note that a more comprehensive comparison should be made to quantify performance variance due to train/test partitioning, weight initialization, and training stochasticity. In lieu of this, we recognize that these results are suggestive but not conclusive and should be interpreted with some caution, particularly because some drug-groups will have as few as 10 observations and therefore may be strongly influenced by slight changes in model behavior.



(a) Exp. 01 Drug Performance by model **(b)** Exp. 02 Drug Performance by model

Figure 35: Comparison of performance (R^2) grouped by drug between GNNCDR (no pretraining) model and the NaiveNN model. Each point represents a drug and points lying under the diagonal are better predicted by the GNNCDR model.

5.4.3 Inspection of GNNCDR embeddings

To evaluate if the GNNCDR model behavior emulates our biological premise, we first train a GNNCDR model (without pre-training any embeddings) and then inspect the embeddings for biologically relevant information content. This inspection is done by comparing the embeddings with the resources described in Table 10. For gene, drug, and FI edge embeddings, the comparative datatypes are in the form of binary groupings; for example, the gene embedding is compared to Gene Ontology (GO) molecular function terms, in which each gene is either within the GO term or not. We evaluate these binary groupings using visualization and limited quantitative metrics. The significance of the clustering is then computed by comparing it with random embedding permutations.

Inferring Binding Affinity from GNNCDR embeddings.

The *DrugConv* uses an edge-specific embedding (z_{ij}^{DTI}) as a scalar [0,1] that acts somewhat analogously to concentration and scales the magnitude of a given perturbation. We expect that this embedding should capture drug-target specific binding affinity. To evaluate this, we use the Cancer Targetome [BCKM$^+$17] that characterizes a limited number of drug-target binding affinity metrics. Notably, there is sparse annotation of binding affinity in this dataset, and therefore we can only evaluate a subset of the DTI embeddings. Additionally, the Cancer Targetome often reports multiple binding affinity metrics for each DTI (which metric is reported depends on the drug and all metrics are rarely annotated) and therefore we average the metrics prior to comparison with the DTI embedding. Note that, while

different metrics are reported for different experiments, we expect the metrics to be strongly correlated. It is also important to recognize that since not all metrics are reported for each drug, we are evaluating on largely disjoint subsets of drugs for each metric. We transform all binding affinity metrics (x) using the function: $-log_{10}(x)$. We expect that embeddings that accurately capture binding affinity information should positively correlate with our transformed literature curated binding affinity values. In Table 14 we report the Spearman correlation and p-value significance between the DTI embedding and the literature curated binding affinity metrics. Of the four binding affinity metrics, the inhibitory constant (Ki) had the highest correlation and statistical significance, which may be due to the prevalence of inhibitors used in the LINCS L1000 dataset. We plot the inhibitory constant relationship to Exp. 01 DTI embedding in Figure 36. The other three had somewhat marginal correlations, although we note that all four metrics in Exp. 01 had positive correlations. Notably, permuting the biological network (methods: "rewire" and "random") create false edges and therefore do not have reported binding affinity metrics, and because of this, we could not evaluate the performance on the permuted network GNNCDR models.

Model	Spearman Correlation [p-value]			
	EC50	IC50	Kd	Ki
EXP01	0.118 [0.275]	0.049 [0.167]	0.222 [1e-4]	0.681 [0.00]
EXP02	0.177 [0.041]	-0.232 [1.00]	-0.182 [1.00]	0.442 [0.00]

Table 14: Using the GNNCDR (no pretraining, true network) model drug-gene embedding compared to literature curated drug binding affintiy metrics from the Cancer Targetome [BCKM$^+$17]. DTI edges with multiple metrics are averaged. Each metric was transformed using $-log_{10}(metric)$ and compared to the untransformed embedding weight $[-\inf, \inf]$. The p-value was calculated by permuting (n=10000) the GNNCDR DTI weights and computing the proportion of permutations that had a greater spearman correlation than the true embedding values.

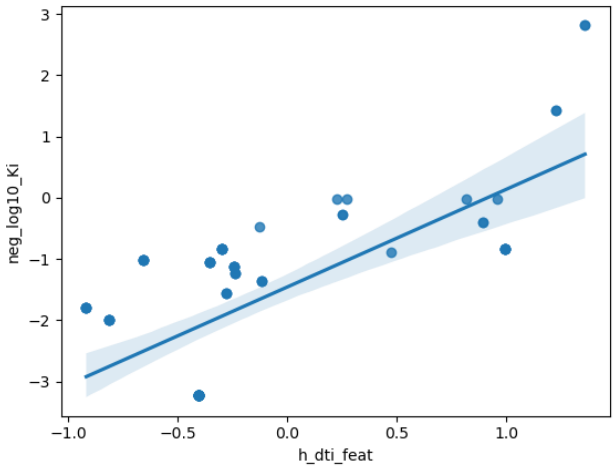
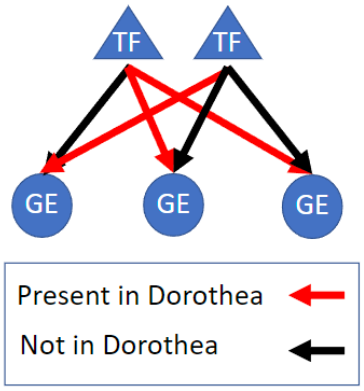


Figure 36: EXP01 (No pretraining) GNNCDR model. The learned DTI edge embedding feature compared to true binding affinity (Inhibitor constant; K_i) information from the cancer Targetome. Strong correlation indicates our model is learning accurate binding affinity information during training.

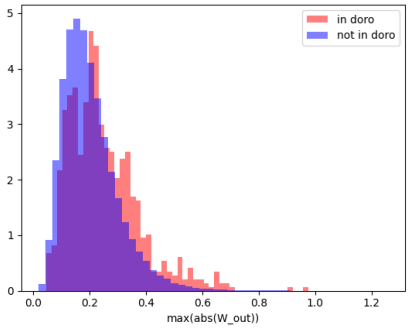
Inferring TF regulons using GNNCDR learned weights.

The *Regulon Module* was designed to be interpretable and capture transcription factor effect on gene expression. To evaluate whether our learned regulon weights are accurate, we compare the learned weights to literature curated TF-DNA interactions aggregated in the Dorothea resource [GAITSR18]. Importantly, this resource aggregates non-contextual interactions, whereas our *Regulon Module* learns cell-specific edge weights. To address this discrepancy, we aggregate the learned TF-gene edge weights across cell lines be selecting the maximum absolute edge weight value. This results in a non-contextual TF-gene weight and we rationalize that the greatest magnitude edge weights are likely to be represented in known TF-DNA resources.

The GNNCDR TF-gene edge weights are then divided into two groups depending on whether the TF-gene edge was present in Dorothea (confidence: "A"). We plot the distribution of the groups in Figure 37b. The distribution of Exp. 02 edge weights reported in Dorothea had significantly higher values than the distribution of edge weights not reported in Dorothea. We report additional metrics for each experiment model in Table 15. The area under the receiver operator score (AUROC) is computed by using the edge weights to predict the inclusion in Dorothea. Statistical significance is calculated using the Mann-Whitney U test [MW47]. Both experiments GNNCDR models, trained on true biological networks, show evidence of inferring true TF regulons although with marginal predictive value (AUROC values: 0.582, 0.611). Somewhat surprisingly, we found that the GNNCDR models, trained on inaccurate biological networks also show evidence of inferring true TF regulons. We expected that permuting the biological network would destroy the relational structure that should be critical for inferring accurate regulons. This result may suggest an alternative learning mechanism that predisposes to these results. For instance, given a known TF-DNA interaction between transcription factor A and DNA target X, it may be that our GNNCDR algorithm learns large edge weights from all TFs to DNA target B (A->X, B->X, C->X). Such a mechanism may learn a non-specific TF inference but accurate DNA target inference. To investigate this, we computed the average pairwise cosine similarity (ACS) between the learned weights of the GNNCDR TFs. This metric was intended to elucidate if the model is learning similar TF-gene weights for each TF. The ACS scores for each experiment model are listed in Table 15. In general, we do not see a marked difference in ACS scores between models, and the similarity scores are fairly small, suggesting that each TF learns a unique set of TF-gene weights. Future research should investigate this result in more detail.



(a) Regulon evaluation methodology.



(b) EXP 02 GNNCDR (No pretraining) regulon distribution of known TF-gene edges compare to learned edge weights.

Figure 37: EXP02 learned TF-target weights distributions of known and unknown TF-targets.

Table 15: Comparison of known TF regulons (Dorothea "A" confidence) to GNNCDR learned weights, aggregated over all cell lines by: max(abs(weights)). We evaluate GNNCDR models that were not pretrained. P-values were computed using the Mann-Whitney U test. Average Cosine Similarity (ACS) was computed by average of pairwise transcription factor learned weights.

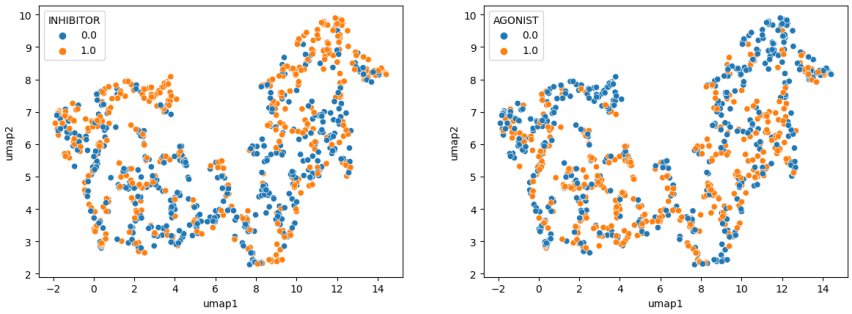
EXP ID	Network status	In-Doro. mean weight (95% CI)	Out-of-Doro. mean weight (95% CI)	AUROC	p-value	ACS
01	True	0.84 (0.27, 1.81)	0.74 (0.24, 1.7)	0.582	1.61e-6	0.112
01	Rewired	1.06 (0.33, 2.54)	0.83 (0.26, 1.81)	0.635	5.24e-21	0.111
01	Randomized	0.98 (0.32, 2.25)	0.87 (0.29, 1.87)	0.562	2.00e-4	0.113
02	True	0.25 (0.08, 0.57)	0.20 (0.07, 0.44)	0.611	4.856e-19	0.082
02	Rewired	0.16 (0.05, 0.36)	0.15 (0.05, 0.34)	0.534	5.59e-4	0.111
02	Randomized	0.24 (0.07, 0.54)	0.20 (0.07, 0.44)	0.583	2.90e-16	0.113

Inferring Drug MOA from GNNCDR embeddings.

The *DrugConv* employs a drug-specific embedding, intended to capture information related to drug mechanism-of-action (MOA) such as agonism or inhibition. To determine whether our drug embedding learns this information, we use UMAP dimension reduction [MHM18] to visualize the embedding and color the points based on the MOA of known drugs, as described in the CLUE drug repurposing dataset [CNS$^+$20]. In Figure 38, we evaluate the information content of Exp. 02 GNNCDR (no pretraining) relevant to "agonist" or "inhibitor" MOA annotations. Subjectively, there appears to be vertical separation between MOA annotations; however, there also appears to be local clustering. To investigate whether embeddings with similar MOAs have significant local clustering, we used the Precision@K metric (K=3), which measures the proportion of K nearest neighbors that share the same label. The Precision@3 results are presented in Table 16. Significance is computed by randomly permuting the embedding and calculating the proportion of precision@3 values greater than the true embedding value (computed in the original embedding space, not UMAP embeddings). Interestingly, Exp. 01 does not show evidence of local clustering. Exp. 02,

however, does show evidence of local clustering across all three models tested (network status: true, rewire, random). Notably, permuting the biological network does not change the effect of a drug on the network, and therefore we have no expectation that the permuted networks should *not* capture drug MOA information.

It should also be noted that we make the assumption that the GNNCDR model learns underlying biological signaling, which requires an understanding of gene-gene edge functions (i.e., activation vs inhibition). The drug embedding is therefore a mechanism for distinguishing differences in the impact of the drug on a given protein or process. However, this premise also requires that drugs with different MOAs share the same drug targets or biological processes. In the absence of shared drug targets, the model may be able to learn unique signaling based on the target identity, independent of the drug MOA. Due to this trait, we expect the drug MOA embedding to become more important and distinguishable as we increase the number of drugs that are modeled (assuming more drugs m̃ore shared targets/processes). This expectation may explain why we do not see evidence that Exp. 01 embeddings capture MOA information (Exp. 01 models s̃300 drugs while Exp. 02 models s̃900).



(a) Label: "Inhibitor" **(b)** Label: "Agonist"

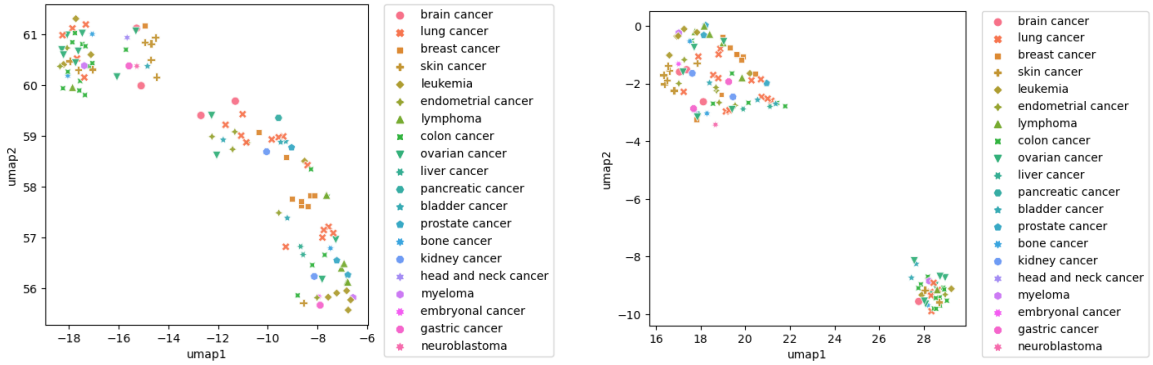
Figure 38: EXP02 UMAP visualizations of drug embedding colored by MOA annotations.

Table 16: Precision@3 [permutation test p-value]

Exp. ID	Graph status	"Inhibitor"	"Agonist"	"Antagonist"
01	True	0.689 [1.0]	0.719 [0.63]	0.853 [0.45]
01	Rewired	0.725 [0.545]	0.725 [0.363]	0.838 [1.0]
01	Randomized	0.687 [0.727]	0.700 [0.909]	0.771 [1.0]
02	True	0.591 [0.0]	0.593 [0.0]	0.64 [0.0]
02	Rewired	0.563 [0.0]	0.574 [0.0]	0.627 [0.0]
02	Randomized	0.565 [0.0]	0.573 [0.0]	0.634 [0.0]

Inspecting GNNCDR cell embeddings.

True biological TF regulons are known to vary by cellular context depending on chromatin accessibility, TF co-factors and DNA methylation. The GNNCDR *regulon module* uses a cell embedding to learn cell-line specific TF regulons (TF-to-LINCS edge weights). Based on these design decisions, we expect cell embeddings to cluster by cell lineage and primary disease. To investigate this, we use UMAP dimensionality reduction to visualize the cell embedding in Figure 39. Although there is subjective evidence that some cell types cluster (most notably: breast cancer, lung cancer, leukemia), the embeddings show marginal clustering by primary disease. This result could be attributed to an over-expectation of cell-specific regulons (i.e., cell regulons are largely similar) or due to poor performance of the GNNCDR algorithm. It should be noted that each prediction also has access to cell line 'omics as node features, which may be used for cell-specific response predictions and therefore detract from the necessity of a cell-specific embedding.



(a) EXP01 GNNCDR. **(b)** EXP02 GNNCDR.

Figure 39: UMAP visualizations of the GNNCDR (no pretraining) cell embeddings annotated by primary disease.

Inferring Gene Ontology annotations from GNNCDR embeddings.

The *CellConv* utilizes a gene node embedding to infer the edge functions and edge weights. Our design goals were that this embedding would capture gene-specific features representative of molecular function, biological process, or gene relevance. To investigate this, we evaluated the separability of annotations when using the gene embedding as input. For each GO annotation, we train a logistic regression to predict the inclusion of the GO term [0,1] and then evaluate using the AUROC. Note that we do not train and evaluate on separate data partitions, as this is intended to be a metric of simple separability, not predictive quality. We report a select subset of GO term separability scores in Table 17. Note that the five GO terms reported here were the only five terms that had at least 50 gene members in Exp. 01. Exp. 01 shows limited separability of the evaluated GO terms, and are comparable to the scores obtained from the Rewired and Randomized GNNCDR models, which suggest the Exp. 01 gene embedding does not effectively learn molecular functions of the gene. Exp. However, 02 shows an improved separability of GO terms compared to the Rewired and Randomized GNNCDR gene embeddings. To further investigate this performance, we plot a subset of UMAP embeddings molecular function GO ontology in Figure 40 with annotated GO terms colored by the separability score. We note that there are local regions of the Gene Ontology that have notably strong separability.

Table 17: The separability of select GO terms using learned GNNCDR EXP01 gene embeddings (protein kinase binding, metal ion binding, identical protein binding, ATP binding, protein binding).

		Area under the receiver operator curve (AUROC)				
Exp. ID	Graph Status	GO:0019901	GO:0046872	GO:0042802	GO:0005524	GO:0005515
01	True	0.565	**0.557**	0.557	0.579	0.565
01	Rewired	0.564	0.538	0.487	0.564	0.517
01	Randomized	0.555	0.531	0.559	0.532	0.574
02	True	**0.590**	0.520	**0.577**	**0.596**	**0.584**
02	Rewired	0.527	0.521	0.516	0.543	0.510
02	Randomized	0.510	0.518	0.521	0.530	0.513

In particular, the GO terms annotated in red seem to be locally separable, which we plot in Figure 41. Interestingly, the six GO terms visualized in Figure 41, which have strong separability scores, are related to DNA binding activity (AUROC scores > 0.68):

- GO:1990837 - sequence-specific double-stranded DNA binding
- GO:0000976 - transcription cis-regulatory region binding
- GO:0003700 - DNA-binding transcription factor activity
- GO:0000981 - DNA-binding transcription factor activity, RNA polymerase II-specific
- GO:0000978 - RNA polymerase II cis-regulatory region sequence-specific DNA binding
- GO:0001228 - DNA-binding transcription activator activity, RNA polymerase II-specific

Several additional GO terms with the strongest separability in Exp. 02 include:

- GO:0003713 (AUROC: 0.680) - transcription coactivator activity
- GO:0008083 (AUROC: 0.663) - growth factor activity
- GO:0140297 (AUROC: 0.663) - DNA-binding transcription factor binding
- GO:0019899 (AUROC: 0.655) - enzyme binding
- GO:0003677 (AUROC: 0.653) - DNA binding
- GO:0003682 (AUROC: 0.635) - chromatin binding

- GO:0008013 (AUROC: 0.631) - beta-catenin binding
- GO:0051092 (AUROC: 0.629) - positive regulation of NF-kappaB transcription factor activity

GO terms with the weakest separability include (AUROC < 0.51):

- GO:0003779 - actin binding
- GO:0005096 - GTPase activator activity
- GO:0031267 - small GTPase binding
- GO:0004930 - G protein-coupled receptor activity
- GO:0043547 - positive regulation of GTPase activity
- GO:0005509 - calcium ion binding
- GO:0005178 - integrin binding
- GO:0008201 - heparin binding
- GO:0005525 - GTP binding
- GO:0003924 - GTPase activity

If we adopt separability as a surrogate for importance (weak assumption since separability is necessary for importance but not necessarily sufficient), then we can infer that DNA binding is particularly important to model predictions. It should be noted, however, that due to the GNNCDR model architecture, this may be caused by a data leak since we select transcription factors as input to the *Regulon Module*. Indeed, this explanation is supported by a strong correlation between AUROC scores obtained from the True GNNCDR and the Rewired GNNCDR embeddings (r=0.77). To elucidate which GO terms have higher separability in the True GNNCDR embeddings as compared to the Rewired GNNCDR embeddings, we compute the difference in AUROC scores and report the top five greatest differences:

- GO:0008083 ($\Delta AUROC$: 0.111) - growth factor activity
- GO:0005085 ($\Delta AUROC$: 0.100) - guanyl-nucleotide exchange factor activity
- GO:0003713 ($\Delta AUROC$: 0.091) - transcription coactivator activity
- GO:0007189 ($\Delta AUROC$: 0.088) - adenylate cyclase-activating G protein-coupled receptor signaling pathway
- GO:0046982 ($\Delta AUROC$: 0.077) - protein heterodimerization activity

These results suggest that there are several groups of genes that capture molecular function from the true biological network topology and the LINCS L1000 response data and are important for the GNNCDR prediction. The results of

"transcription coactivator activity" are particularly interesting, as we rationalize that effective transcriptional response prediction is expected to require an understanding of TF coactivators.

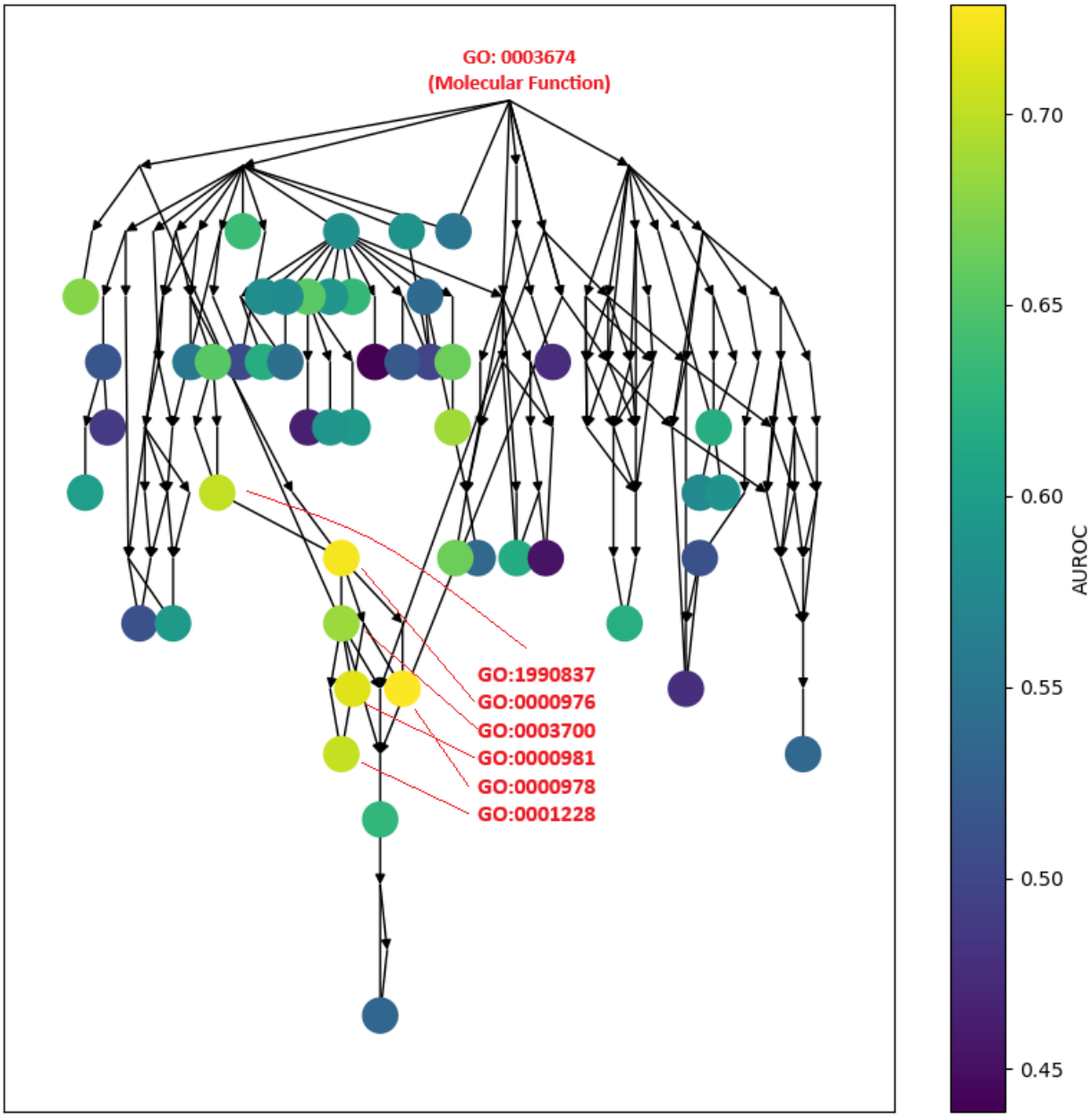


Figure 40: The Gene Ontology (GO) Molecular Function subgraph relevant to the EXP02 GO terms that annotated gene nodes had annotations for (colored nodes). Nodes not represented with a colored node were intermediate GO terms that we did not include annotation for. The node color is colored by the AUROC score, indicating separability of the given GO term. We included GO term annotations for a subset of highly separable terms.

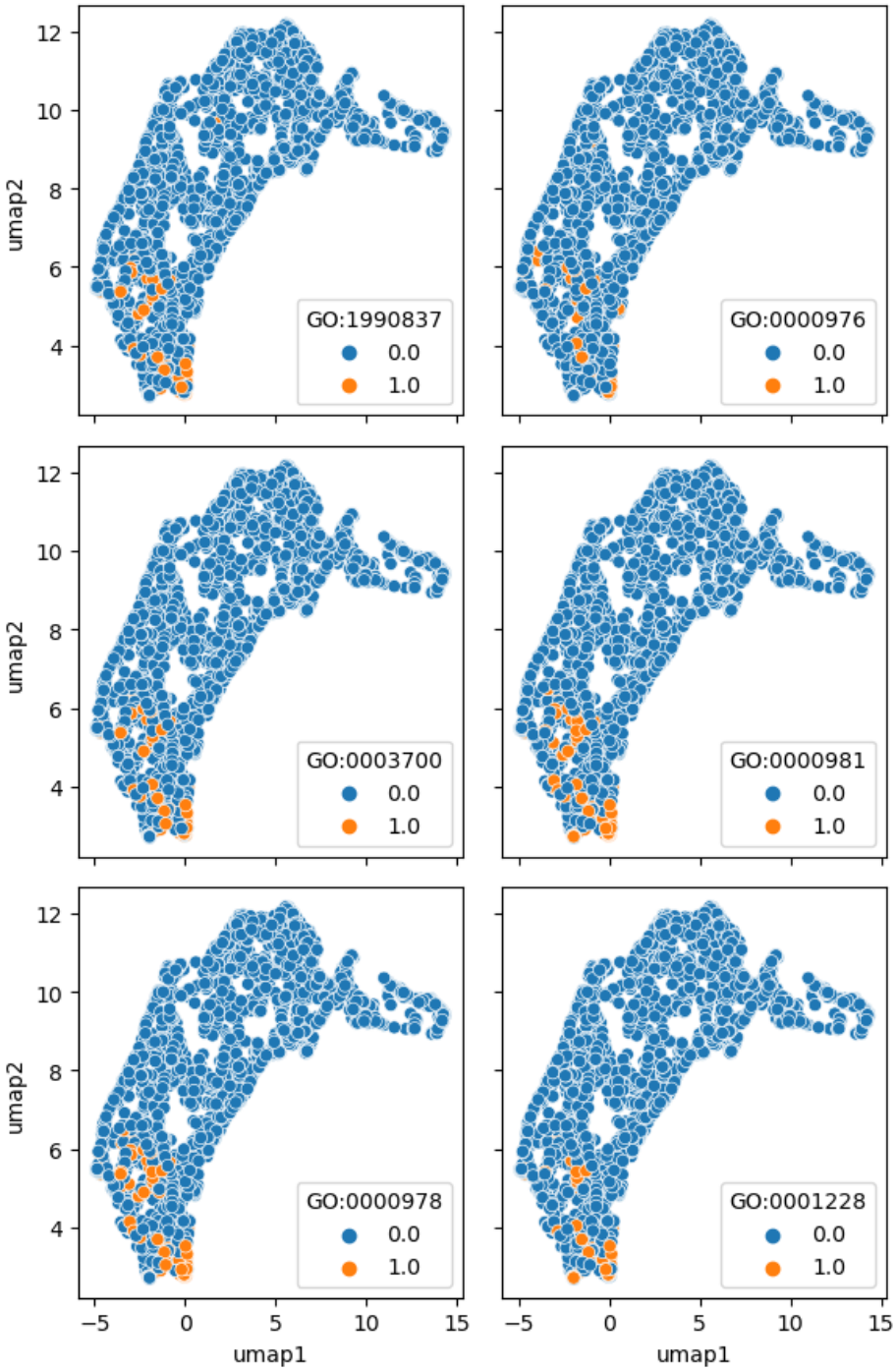


Figure 41: Gene embedding UMAP visualizations. Select GO term annotations (from Figure40) highlighted in each plot.

Inferring gene-gene interaction annotations from GNNCDR embeddings.

To evaluate whether the GNNCDR gene-gene edge embeddings captured biologically relevant information, we ask the question: *Are the edge embeddings predictive of the Reactome FI edge annotations?* To answer this, we chose to test

the linear separability of the FI edge annotations in the gene-gene edge embedding space. We formulate this as a simple logistic regression where the exogenous features (X) are the GNNCDR embedding features and the endogenous features (y) [0,1] are the respective edge annotation label. Since our only desire is to evaluate if the embeddings are separable, we train and evaluate the full dataset (Note: no hold-out test set). We then performed permutation testing by shuffling the edge embeddings and retesting separability (N = 250). We quantify separability using the area under the receiver operator curve (AUROC). There are many edge annotations, and therefore we quantify performance by the number of FI annotations that were "predictable" (i.e., p-value <= 0.05). We also report the maximum AUROC value that we achieved across all the FI annotations. We report the results in Table 18

We expect that the GNNCDR model that operates on a true biological graph should learn edge embeddings that are predictive of FI annotations, and that GNNCDR models that operate on inaccurate or random biological graphs should not be predictive of FI annotations.

Interestingly, in Exp. 01 (True network) 3/8 most separable FI annotations were related to inhibition: "inhibit", "inhibition", and "-|". Somewhat unexpectedly, the most predictable annotation was "predicted."

Table 18: GNNCDR edge embedding separability performance. Exp. 02 not evaluated.

EXP ID	Graph status	Num. Predictable	max AUROC
01	True	8	0.5526
01	Rewired	2	0.5347
01	Random	6	0.5595

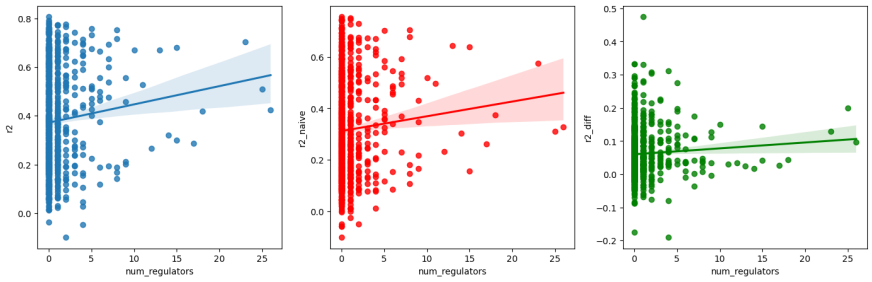
5.4.4 Gene Performance compared to known Transcriptional Factors

A limitation of this approach is that the selection of a gene space implicitly biases the genes for which we can expect to have good predictive performance. For our model to mechanistically predict the transcriptional activity of a gene then we must:

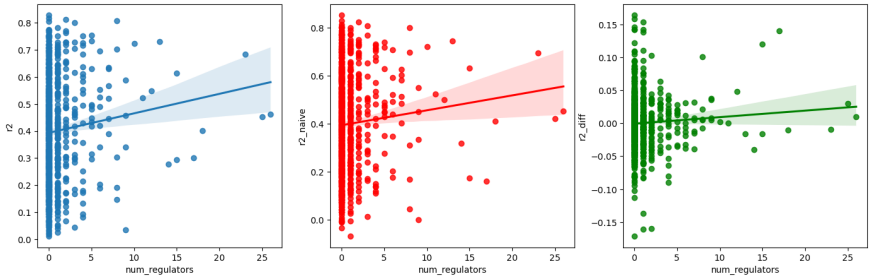
1. Our gene-space must include relevant upstream genes/proteins
2. Our gene-space must include the transcriptional regulator (TF) of each gene

Recognizably, we do not know all the upstream proteins or transcriptional regulators, and this is further confounded by contextually active regulators, so we cannot filter our endogenous genes. As a simple approach to evaluate the effect of this bias, we use Dorothea regulons to compare gene-level prediction performance (Pearson correlation) against the number of regulators in our gene space, shown in Figure 42. If our model accurately mimics the described biological premise and if the Dorothea resource is relatively complete, we would expect endogenous genes with fewer (or none) gene-space regulators to perform significantly worse than endogenous genes with gene-space regulators.

Our results show that genes with more gene-space regulators have modest correlation with the number of transcriptional regulators, however, when adjusted for general prediction trends (baseline prediction by the NaiveNN) this trend is markedly decreased.



(a) Exp. 01



(b) Exp. 02

Figure 42: Gene-grouped performance compared to the number of transcriptional regulators for the GNNCDR models (no pretraining) and the NaiveNN model. The number of transcriptional regulators was calculated using the Dorothea "A" confidence transcription factor regulons. (right) We use the difference in gene performance between the GNNCDR and NaiveNN model to remove performance bias attributable to data quality or prediction trends shared by both algorithms. The Left column is performance of the GSNN (R^2), the center column is performance of the NaiveNN (R^2), and the right column is the difference between GSNN performance and NaiveNN performance.

5.5 Discussion

In this work, we sought to answer several questions:

- *Can we use GNNs to build more robust and accurate models of drug response and perturbation biology?*
- *Can we use heterogenous sources of prior knowledge to improve prediction performance?*
- *Can we design GNN architectures that are inherently interpretable and useful for inferring key aspects of drug response?*

We explore these questions by developing a novel GNN architecture that utilizes prior knowledge about known molecular interactions, gene ontologies, drug mechanism-of-action, and annotation of molecular interaction type or behavior. We evaluate the performance of this approach in several ways:

1. **Prediction performance compared to traditional neural networks and GNNs using inaccurate knowledge.** We show that our method performs comparably to traditional neural networks (NaiveNN) but does not notably outperform neural networks across experiments. While Exp. 01 performances indicate that the true biological network resulted in the greatest performance compared to the permuted networks, Exp. 02 did not exhibit this, and therefore we cannot conclusively state that the GNNCDR model is aided by prior knowledge.

2. **Use heterogenous sources of prior knowledge, including gene ontologies, drug MOA and molecular edge function annotations to pre-train relevant GNN learning mechanisms.** Counter to our expectations, our results suggest that pretraining embeddings designed to capture key aspects of drug response does not improve prediction performance of the GNNCDR algorithm. Interestingly, pretraining the permuted networks did improve performance.

3. **Inspect knowledge-naive (i.e., models not pre-trained with prior knowledge) embeddings to show that our custom learning mechanisms capture key aspects of drug response.** We inspect the learned GNN embeddings and find weak evidence that our method can learn aspects of drug response including binding affinity, gene function, drug MOA, molecular interaction function, and transcription factor regulons. Unexpectedly, we find that the permuted networks also learn many of these same aspects, which suggests that there may be sources of data leak due to the GNNCDR architecture and may suggest that our interpretation of embeddings may be invalid.

These results suggest that the GNNCDR has limited utility for use in predicting perturbation biology and that further research is warranted to effectively leverage prior knowledge into predictive models of drug response while still maintaining the deep learning advantages of flexible and accurate prediction performance.

5.5.1 Limitations

Perhaps the most limiting feature of this approach is that we can only encode and predict drugs that have known molecular targets. This reduces the volume of data on which we can train and prevents the application of our method to novel drugs with no known prior knowledge. Additionally, it is likely that our GNN algorithm is limited by sparse annotation of molecular interactions and that missing or false molecular interactions are likely to degrade the performance of our GNNCDR algorithm. A key motivation for future research would be to develop mechanisms to infer drug-gene interactions, as well as novel gene-gene interactions, thereby providing a means to "fill in the gaps" of current prior knowledge.

Another challenge is that our results, while suggestive that the GNN embeddings do capture aspects of drug response, have only a marginal capacity to predict these aspects (e.g., binding affinity, gene ontologies, drug MOA, edge functions). This marginal performance may be attributable to data quality issues, ineffective learning mechanisms, or poor data volume. Future research would benefit from ablation studies and careful hyperparameter tuning to investigate

which aspects of the GNNCDR architecture are useful or limiting and how performance might be improved. Additionally, application of these methods to other datasets would help to explore if these marginal results are due to data quality issues, and to explore if this method is applicable to a wide range of drug response prediction tasks.

In particular, we note that our GNNCDR algorithm may suffer from known GNN challenges such as *oversmoothing* [ZA20] and *oversquashing* [Aka23]. These aspects are particularly relevant, as effective GNN modeling of perturbation biology is likely to require deep GNNs, which are particularly affected by these challenges. Future work should seek to quantify the effects of oversmoothing and oversquashing in the GNNCDR model and propose solutions to mitigate any issues. Additionally, we note that functional interaction networks are likely to include many interactions that do not *directly* regulate gene expression (e.g., PPIs between complexes, post-translational protein modifications, etc.) and therefore the resulting graph structure is likely to be *heterophilous* (meaning that unlike nodes are commonly connected) with respect to gene expression changes. The development of GNN algorithms that are applicable to heterophilous graphs will be beneficial to this research. Additionally, in this work, we formulated our gene-gene graph such that we treat protein, RNA and DNA as a single "gene" and that this may be poorly representative of the true biology. More complicated approaches that encode these molecules as separate entities may improve the rationality of our prediction logic and the validity or usefulness of prior knowledge for drug response prediction.

We also note that the selection and preprocessing of prior knowledge is likely to have a significant impact on the performance of our model. In this work, we used the *Reactome Functional Interaction* dataset [GJS$^+$21], however, comprehensive comparison of performance of our algorithm should also be performed when using alternative resources. Alternative molecular interaction resources that could be used include Omnipath [TKSR16], StringDB [SGN$^+$20], Kyoto Encyclopedia of genes and genome (KEGG) [KG00] and Pathway Commons [CGD$^+$10]. Additional aspects of pre-processing include the pre-processing of 'omic features' and the selection of confidence thresholds for edge inclusion.

In this work, we manually selected subsets of molecular interactions that we believe to be relevant to drug response, and this pathway selection step likely impacts the performance and usefulness of our models. Including key molecular entities, such as transcription factors or major molecular regulators, is critical for accurate prediction logic of the GNNCDR model and therefore should be chosen carefully. However, inclusion of spurious or uninvolved molecular entities, for example, ribosomal complexes that are unlikely to be directly involved in drug response, may be detrimental to performance and increase the memory and time complexity of our algorithm.

In this work, we conducted only a rudimentary analysis that evaluated our method against baselines. Further validation of these methods should be performed on a wider range of baselines and with a more rigorous evaluation scheme to assess significant performance differences. For example, we do not perform hyperparameter optimization, and comprehensive comparisons of this method to baselines should be evaluated on a larger set of possible hyperparameters to ensure accurate comparisons. Additionally, in this work, we tested on a single train/test partition, and future work

should use Monte Carlo cross-validation [XL01] to ensure that the performance gains were not attributable to the random sampling of partitions.

In summary, this work represents an early approach to development of interpretable and useful GNN models for drug-response and perturbation biology prediction tasks; however, there are many limitations that need to be addressed before the full potential of this method can be realized.

5.5.2 Comments on the methodological design of GNNs suitable to perturbation biology

The methodological design of a suitable GNN for perturbation biology is a challenging task. As discussed in Section 1.17, there are several major challenges with traditional GNNs applied to this task. We sought to overcome these limitations with custom learning mechanisms. In the design phase, we envisioned our learning task similar to a diffusion process (note we are not referencing the mathematical definition of diffusion here), where information flows from one or a few drug nodes, to all other accessible nodes analogous to water flow through a network of pipes or heat diffusion through a wire web. Cellular context, we imagined, should be a mediator of how information would flow throughout the network, which we compare to changing pipe diameters in the fluid flow analogy. For instance, on a static (i.e., fixed graph shared across cellular contexts) graph structure, the cell context would mediate the flux of information across edges and change the final perturbation response. Developing GNNs that fulfill this vision while still maintaining effective learning (i.e., dealing with gradient vanishing/exploding, oversmoothing, oversquashing, etc.) is difficult. Our first task was to infer a message passing methodology that would treat drugs as the primary "source" of information and behave as diffusion from the active drug node(s). To do this, we learn edge and gene parameters which are then used to infer an "edge function" (W_{ij}) that allows for a static function (same across all layers) that propagates information from one node to another ($x_j = Tanh(\sum_{i \in \mathcal{N}_|} \alpha_{ij}^{cell} * W_{ij}^T x_i) + x_j$). Notably, this edge function does not have a bias term, and therefore if all upstream nodes have a value of zero (no perturbation), then x_j will similarly be zero. This ensures that information can only be propagated from active (non-zero concentration) drug nodes. The variable α_{ij}^{cell} is a cell line specific, and inferred based on the local 'omic features, and is the primary mediator of differences in cell signaling by the cell line. In our water flow analogy, α_{ij}^{cell} was intended to mediate the diameter of the pipes. Small values will prevent information flow, while large values will encourage it. While this mathematical approach fulfills our desired requirement of propagating information from a drug "source" to all accessible transcription factor regulators, it has a number of notable limitations. In particular, this procedure is not amenable to most normalization procedures. For example, batch normalization [IS15] would allow data leaks between nodes. For instance, a resulting transcription factor with a latent value of zero (no perturbation) could become non-zero after batch normalization, which would allow prediction of those TF targets, even though there may be no path from the active drug to that TF. Similarly, the layer norm [BKH16] and Instance norm [UVL16] could also allow data leakage between nodes. A potential normalization mechanism that we did not attempt but could address these issues is MessageNorm [LXTG20]. To our current knowledge, MessageNorm should not cause data-leak between nodes, and may help address common limitations of deep GNNs, such as gradient vanishing/exploding and oversmoothing.

We spent significant thought on whether to formulate this as a node-prediction problem or as a graph-prediction problem. The node prediction problem would be ideal in many respects as the response logic would be based on prior knowledge interactions and easily traceable on the basis of the graph structure. We reason, however, that as a node prediction problem, the prior knowledge encoded as a graph is likely to be over-constraining due to missing prior knowledge. We were particularly concerned about the sparse annotation of transcription factor regulons. As a node prediction problem, we did not have a convenient method to infer new edges during the learning process, and therefore, missing edges were likely to degrade performance. Instead, we chose to formulate this task more analogous to a graph-prediction problem, however, we added constraints in an attempt to aid interpretability. After GNN inference of transcription factor perturbation, we infer a scalar [-1,1] value for each TF and then use a simple linear model to predict the LINCS L1000 expression changes. This allows any TF to predict the response of any gene and, therefore, we rationalize that sparse annotations of regulons in available resources would be mitigated. Using a linear layer rather than a fully connected neural network allows human interpretation of transcription factor roles. With this method, it is also fairly easy to trace the prediction response to i) the responsible transcription factor and, because of this, to ii) the genes responsible for TF perturbation. Additionally, we envision that future transcription factor regulon constraints could be added by pre-training the TF linear layer.

A limitation to the TF-gene linear layer approach is that it cannot model higher-order interactions of TF-TF gene expression changes. For example, GRNs often recognize the regulatory impact that one TF can have on another. Future directions of work may benefit from novel approaches to modeling the TF expression effect more mechanistically.

Future directions in this research will also benefit from different architectures or methodologies; in particular, we believe Graph Neural Ordinary Differential Equations (GNODE) may be appropriate for perturbation biology prediction [PMP+19]. Such an approach would allow for inclusion of prior knowledge constraints in the form of a biological network (as described in this work) but would model the response as a time series evolving from an unperturbed state to a perturbed state. This methodology aligns well with our goal of information diffusion design. Additionally, recent work in neural ordinary differential equations allows these models to be trained by gradient descent [CRBD18]. A current limitation of this approach is the high time complexity of numerical integration, which is a necessary step to compute gradients.

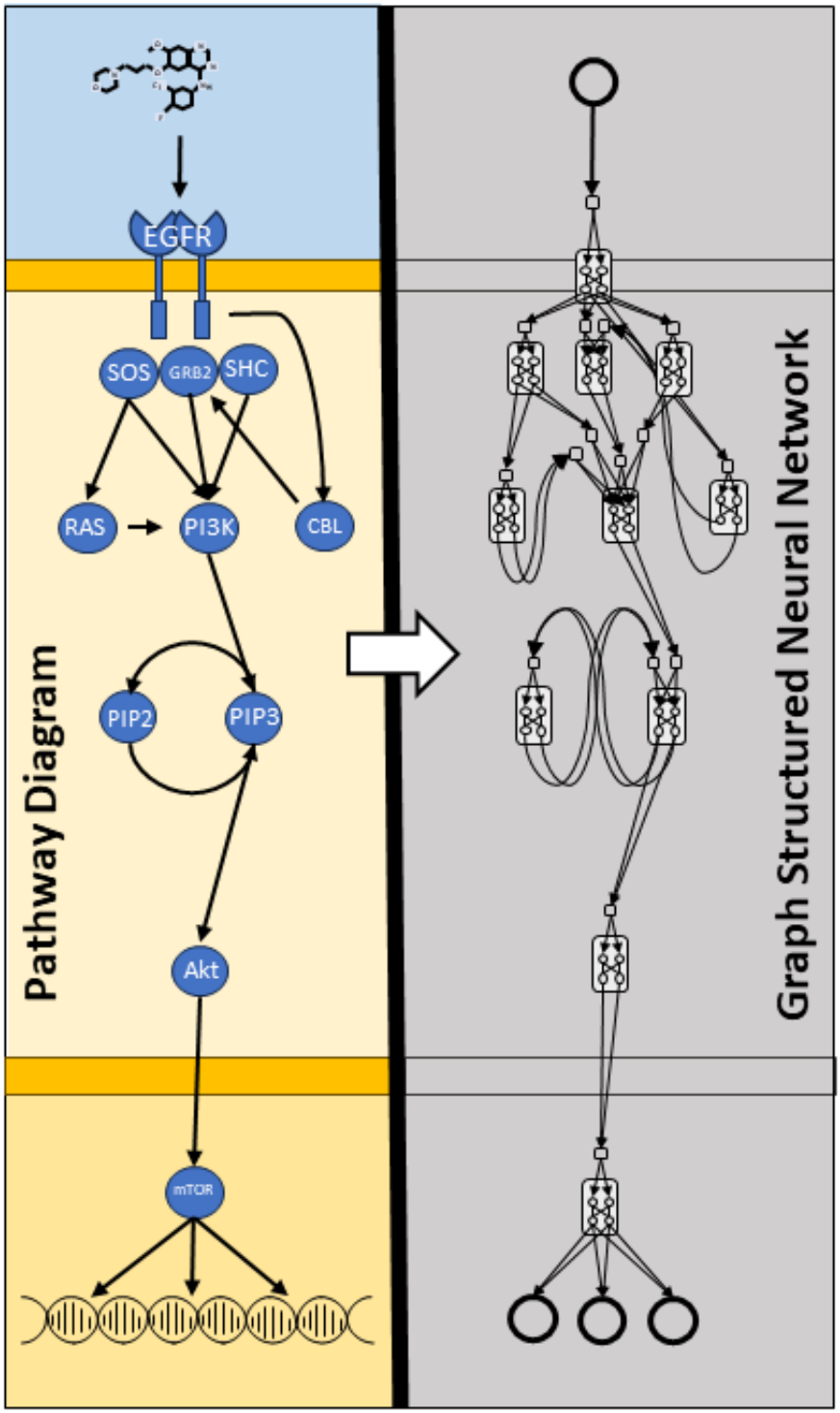


Figure 43: Graph structured neural network (GSNN) summary figure. An example pathway diagram used to describe cellular signaling knowledge (left). A GSNN model built from the pathway diagram. Although not visualized, entity-specific features (e.g., gene mutations, expression, or other 'omics) can be included as additional input to each entity (right).

6.1 Abstract

Computational modeling of perturbation biology identifies relationships between molecular elements and cellular response, and an accurate understanding of these systems will support the full realization of precision medicine. Traditional neural networks, while often accurate in predicting response, are unlikely to capture the true sequence of molecular interactions that mediate the response to a perturbation. Our work is motivated by two assumptions: Methods that encourage mechanistic prediction logic are likely to be more trustworthy and that problem-specific[7] algorithms are likely to outperform generic algorithms. We present a novel deep learning method called *Graph Structured Neural Networks* (GSNN), which uses cell signaling knowledge, encoded as a graph data structure, to add inductive biases to deep learning. We apply our method to the LINCS L1000 perturbation biology dataset and use literature-curated molecular interactions to incorporate mechanistic constraints. We demonstrate that GSNNs outperform baseline algorithms in several prediction tasks, including 1) perturbed expression, 2) cell viability of drug combinations, and 3) disease-specific drug prioritization. We also present a method called *GSNNExplainer* to explain GSNN predictions in a biologically interpretable form[8]. This work has broad application in basic sciences and in preclinical drug repurposing. We believe the GSNN method will lay a foundation for future trustworthy models of drug response, and future refinement of these methods may lead to impactful clinical decision aids.

6.2 Introduction

In Chapters 4 and 5, we focused on using Graph Neural Networks (GNNs) to model perturbation biology, however, the results of GNN algorithms suggest limited utility and performance gains compared to traditional methods. In this chapter, we propose a novel algorithm, which we term Graph Structured Neural Networks (GSNNs) to distinguish them from GNNs.

Our goal for this work is to develop mechanistic deep learning algorithms by incorporating knowledge constraints into the prediction logic, similar to our approach with GNNs in Chapters 4 and 5. We sought to develop the GSNN algorithm with the characteristic of cell signal transduction and gene regulation described in Chapter 1 section 1.15. To reiterate, we seek to fulfill the design requirements of perturbation biology:

- Ability to infer **molecular state** of a cell line based on the local biological network neighborhood.
- A **source awareness**, meaning an ability to delineate different functional effects of upstream signals.
- Ability to model **signal latency**; some molecular interactions progress rapidly (e.g., post-translational modifications) while others progress slowly (e.g., gene-regulation).

[7] Algorithms designed specifically for a certain task, datatype or problem. For example, convolutional neural networks for vision tasks.

[8] We use subgraphs to identify the network of molecular entities that are critical for prediction, this aligns well with biological interpretation and goes beyond simple feature importance by identifying latent variables (molecular entities) that are involved in the prediction logic.

- Ability to learn **non-linear multivariate input-output relationships** between molecular entities in a signal transduction process.

Additionally, we aim to develop an algorithm that overcomes the limitations inherent to many GNN methodologies such as oversmoothing, oversquashing, and gradient vanishing/explosion. In particular, these elements become particularly problematic as the depth of the network increases.

6.2.1 Contributions

This work is inspired by SEMs[9] ability to inject causal structure into modeling strategies and by methods such as GNNs and Visible Neural Networks (VNN) [KPF+20, MYF+18] that combine heterogeneous prior knowledge with powerful data-driven deep learning methods. To overcome the limitations that prevent the application of current methods to cellular signaling, we present a method termed *Graph Structured Neural Networks* (GSNN), which enables the inclusion of prior knowledge encoded as directed graphs with allowable cycles (e.g., feed-back loops, auto-regulation, etc.). We also introduce the *GSNNExplainer* method, which can be used to inspect the GSNN prediction logic in a biologically relevant way. We show that the GSNN algorithm can be used to effectively predict perturbed expression (LINCS L1000) and cell viability for single (PRISM) and combination (NCI Almanac) drug perturbations. Finally, we demonstrate how these methods can be used to prioritize drugs that induce a disease-specific response and to evaluate using FDA drug indications.

6.3 Methods

6.3.1 Problem Description

In this project, we present a learning problem in which we are given a graph, $\mathcal{G}$, which has E edges and N nodes. The nodes can be characterized by their in- and out-degree: *input nodes* have an in degree of zero and a non-zero out degree, *output nodes* have an out degree of zero and an in-degree of one, and *function nodes* have a non-zero in- and out-degree. Respectively, edges can be characterized similarly, where *input egdes* are edges from an input node, *function edges* are edges from a function node and *output edges* are edges to an output node and have precedence over function edges. The training data, $\mathcal{D}$, have node inputs x and outputs y. Only the input nodes will have nonzero x values, while only the output nodes will have nonzero y values. Each observation i will have unique values for x_i, y_i. We propose that the graph learning task is to predict y_i using x_i based on the constraints provided by $\mathcal{G}$.

Like GNNs, this problem assumes *locality*, that is, the properties of a node should be influenced by its neighborhood; however, there are several key distinctions from the common GNN inspired tasks, specifically:

- This problem requires a single fixed graph, where different observations will have different input/output node features. This can be viewed as a transductive learning task, and does not need to generalize to unseen nodes or novel graph structures.

[9]The name *Graph **Structured** Neural Networks* is intended to recognize the role that SEMs played in the motivation of this work.

Table 19: Alternative domains that the GSNN method could applied to.

Description	Inputs	Outputs	Inductive Bias
Fluid flow	Initial flow rate	Flow at a different location or time	Geometric constraints (pipes, landscape)
Heat conduction	Initial temp.	Temp. at different location or time	Material geometries (shapes, contacts)
Electrical circuits	Initial voltages	Voltage at different point or time	Circuit components and connections
Causal modeling	Input variables	Output variables	Latent variable interactions

- The graph structure does not necessarily connect like nodes, and edges can represent a variety of behaviors (no assumption of *homophily* or *relational equivalence*).
- Relations (edges) will have different behaviors and must be inferred from the data (no assumption of *edge uniformity*).

In this work, we focus on graphs that represent cellular signaling networks; however, this problem description is applicable to model a number of other real-world scenarios, which may benefit from the inclusion of heterogeneous forms of inductive bias. Table 19 describes several alternative domains in which the GSNN method could be applied.

6.3.2 Graph Structured Neural Network

We present a deep learning method called Graph Structured Neural Networks (GSNN), suitable for modeling biological signaling networks. The GSNN method is initialized using a structural graph ($\mathcal{G}$) that describes the molecular entities (nodes) and the interactions between them (edges); $\mathcal{G}$ defines a set of constraints in the GSNN algorithm by indicating the allowable interactions and expected latent variables (i.e., molecular entities). *Function nodes* (f_n) are parameterized by a neural network, where the number of inputs is equal to the in-degree of node n in $\mathcal{G}$ and the number of outputs of the neural network is equal to the out-degree of node n in $\mathcal{G}$. The number of hidden channels and network layers[10] of f_n are user-defined hyperparameters, which allow for variable model capacity. We rationalize that proteins with fewer inputs or outputs can be better described by more simple functions, and therefore we provide the option to scale the number of hidden channels in f_n based on the in- or out-degree of node n .

GSNN layer updates can be performed by masked linear operations, an example of which is shown in panel A of Figure 44. Parameters are not shared between *function nodes*, therefore, each function node will learn different relationships between the input and output edges. The GSNN method operates by evolving the *edge* latent representation via sequential GSNN layers[11]. Each layer will update the latent edge values so that the output edges of a *function node* are predicted from the input edge values of the previous layer. Iterative edge updates allow the information to propagate through the structural graph a path length of L, where L is the number of layers in the GSNN. The latent features are representative of *edge* state. Note that this aspect is divergent from GNNs, where latent representations typically characterize the state of a *node*. This allows the GSNN method to learn nonlinear multivariate relationships between input edges and output edges. As we noted in the Introduction, there is a temporal aspect to cellular signaling such that

[10]To avoid ambiguity: there are two "layer" parameters we reference. The function node number of layers ($k \sim 1, 2$), which describe the number of layers in a function node neural network, and the GSNN's number of layers ($L \sim [10 - 20]$) that represent the number of sequential masked linear layers that are used in a GSNN model.

[11]Layer here refers to the number of edge-updates, not the number of layers in a function node

many molecular entities will have a latency between input and output signals. As a means of modeling this edge latency, we include residual connections at each consecutive layer, which allows for an "accumulation" of signal and provides a mechanism to learn edge latency. Residual connections have also been shown to mitigate gradient vanishing issues that are common in deep networks [HZRS15a].

To efficiently implement the GSNN method, we conceptualize the edge-updates as a series of masked linear layers. The weight matrices have dimensions $(E, N * C)$, where E is the number of edges in $\mathcal{G}$, N is the number of function nodes in $\mathcal{G}$, and C is the number of hidden channels in each *function node*. Implementing dense matrix multiplications of these linear layers would require undesirable memory and compute resources, making this method applicable only to relatively small graphs. Fortunately, we can use sparse matrices, which massively reduces the required memory.

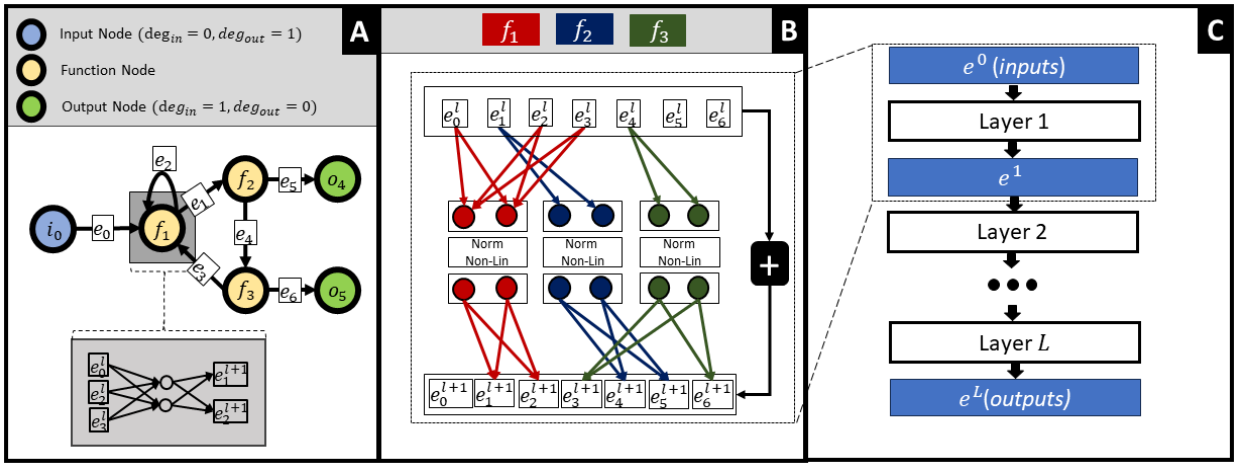


Figure 44: GSNN method overview. A toy example demonstrating how any given graph structure can be formulated as a feed forward neural network with masked weight matrices. Each yellow node in the left graph represents a fully-connected 1-layer neural network with two hidden channels (Note: function node neural networks can optionally be multi-layer). Panel A describes the structural graph ($\mathcal{G}$) which imposes constraints on the GSNN model. Panel B depicts how the edge latent values (e_i) can be updated in a single forward pass. Note that panel B shows sparse weight matrices, where the missing edge connections are equal to zero. The plus sign in panel B indicates a skip connection from the previous layer.

The GSNN method can be considered a residual network [HZRS15a], such that $x_{l+1} = F(x_l) + x_l$ with the added constraints imposed by the structural graph $\mathcal{G}$. The function node parameters may be optionally shared across layers, which prevents the model parameters from scaling with depth; however, in practice we find that the GSNN model performs better when the parameters are not shared across layers. We also optionally add self-edges to all function nodes to allow them to incorporate self-information from the previous layer. Taking lessons learned from traditional residual networks, we also include normalization layers. The ResNet model uses batch normalization [HZRS15a], however, due to the memory requirements of the GSNN method, we are required to use small batch sizes for training, and therefore batch normalization is unlikely to perform well. Instead, we use layer normalization [BKH16] *within* each function node to prevent data leak between function-nodes.

We include options to use Kaiming/He or Xavier/Glorot weight initialization [HZRS15b, Kum17]; however, since the function nodes can only access a subset of inputs and outputs, we use the in-degree (D_i^{in}) of node i in the input graph $\mathcal{G}$ as the "fan in" value and out-degree (D_i^{out}) as the "fan out" value. With this modification, our weight initialization methods are described as follows.

$$w_i^{kaiming} \sim \mathcal{N}(0, \frac{2}{D_i^{in}}) \qquad w_i^{xavier} \sim \mathcal{N}(0, \frac{2}{D_i^{in} + D_i^{out}})$$

We implement the GSNN model in PyTorch [PGM+19] and, at the time of writing this, PyTorch's native sparse matrix multiplication is not well optimized for batched operations. To improve on this, we use the PyTorch Geometric package [FL19] to perform mini-batching and formulate our sparse matrix multiplication as a PyTorch Geometric graph convolution. This approach is markedly faster, particularly when operating on a GPU, than using PyTorch's native sparse matrix multiplication.

6.3.3 Model Evaluation

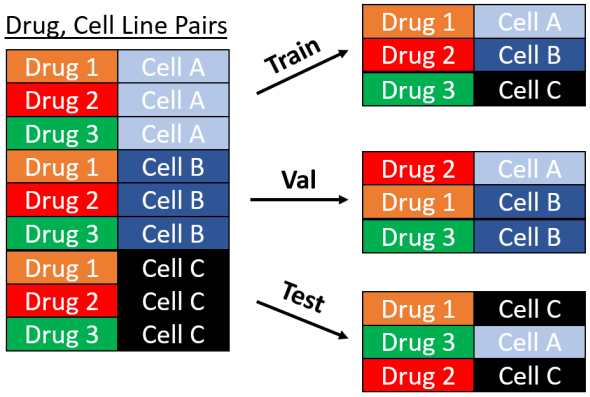


Figure 45: The data partitioning scheme used to train and evaluate the GSNN algorithm.

To evaluate performance, we use Monte Carlo Cross-Validation (MCCV) [XL01] to randomly sub-sample train (60%), validation (20%) and test (20%) data subsets. We run multiple folds ($n \geq 3$) and choose the model that performs best within each fold using the validation set. We report the performance as the average Pearson correlation ($\bar{\rho}$) evaluated in the test set across all folds. Each data partition is a disjoint set of *(drug, cell-line)* pairs (e.g., Imatinib, A549). This evaluation approach allows the model to train on all drugs and cell lines while evaluating on unseen drugs and cell line pairs. Note that this partitioning scheme evaluates the performance of the model within the measured drugs and cell lines, but generalization to unseen cell lines is not evaluated. A model evaluated with this approach can be effectively used to impute missing drug and cell line combinations.

To measure the performance of the model, we use the mean Pearson correlation ($\bar{\rho}$) [FPP07] for all predicted LINCS gene outputs (y_i), where N is the number of LINCS gene outputs.

$$\bar{\rho} = \frac{1}{N} \sum_{i=0}^{N} pearson(\hat{y}_i, y_i)$$

To determine significant performance differences between models, we perform two-sided paired t-test, adjusted for the number of tests (n=3) with the Bonferroni correction [Hay13].

6.3.4 GSNN performance on random networks

As we discussed in Chapter 1, the *no free lunch theorem* suggests that including prior knowledge in deep learning algorithms has the potential to improve prediction performance compared to models without prior knowledge. We have developed the GSNN method to do precisely this and expect that if the GSNN algorithm outperforms the baseline, then the prediction advantage is due to inclusion of prior knowledge; however, it is possible (albeit unlikely) that the GSNN performance could be independent of the prior knowledge (i.e., due to a different aspect of the GSNN algorithm). To test this assumption, we compare the performance of matched hyperparameter GSNN models that are initialized using either 1) a randomized biological network or 2) the true biological network. If the GSNN performance is due to the inclusion of accurate prior knowledge, then we should see significant performance differences between models trained on true and random prior knowledge.

Randomization of the biological network is performed by sampling new edges of the input graph ($\mathcal{G}$). This approach maintains the total number of edges, but not the in- or out-degree of function nodes. *Input edges*, *function edges* and *output edges* are randomized independently to maintain the same number of input and output edges. For all *input edges*, only the destination of each edge is randomized, and similarly for *output edges* only the source of each edge is randomized.

6.3.5 Biological Graph Construction

To create a biological network suitable for modeling cellular signal transduction, we used the resources listed in Table 20. In our biological network, allowable input nodes include: DRUG, EXPR (rna expression), CNV (copy number variation), MUT (mutation) and METHYL (methylation input). We may refer to EXPR, CNV, MUT or METHYL as OMIC nodes, and these inputs represent cellular context (e.g., cell line encoding). All PROTEIN and RNA nodes will be *function* nodes. In this project, all output nodes refer to a LINCS L1000 gene and will have a single edge connection from the respective RNA node (e.g., RNA__P53 $\rightarrow$ LINCS__P53). Note that we do not include DNA molecules as separate entities; rather, we collapse this representation within the RNA nodes. For example, a transcription factor that targets a DNA gene will be encoded as targeting the respective RNA gene. In the current formulation, including DNA nodes offers little advantage and would increase the computational complexity of the model during training and inference.

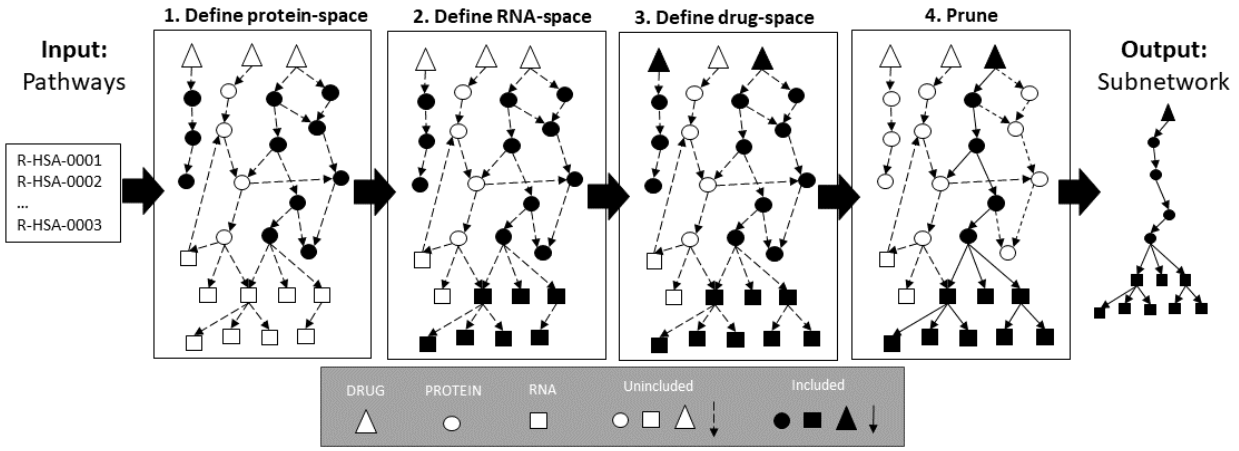


Figure 46: Network construction diagram. Given a full biological network relating drugs, proteins and RNA (mRNA + miRNA) we construct a subnetwork focused on user-provided pathways. Each panel describes a sequential step in the construction. First, we use a user-provided set of pathways to define a *protein-space* that includes all proteins from the full network that have membership in at least one input pathway. The second step defines the *RNA-space* as all RNAs that are regulated by at least one protein in the protein-space; entities from multi-step regulation (e.g., TF->miRNA->mRNA) are also included. The third step is to define the *drug-space* as all drugs that target at least one included protein. Finally, there is a pruning step that removes any nodes that do not regulate a given proportion of downstream RNA outputs (user-defined; typically $\sim 25\%$). The output of this process is a biological subnetwork that comprise the molecular entities and relationships that describe the input pathways.

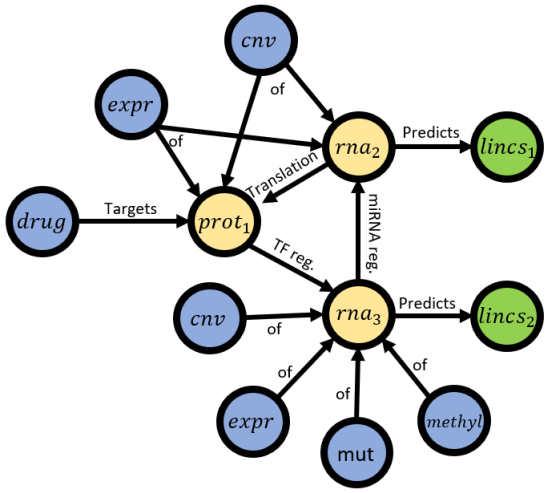


Figure 47: A toy example demonstrating how a biological network for perturbation modeling can be created. Input nodes include both drug and 'omics features (blue), output nodes represent LINCS measurements of a respective RNA molecule (green), and proteins and RNA molecules are function nodes (yellow). Note that while we have added edge labels in order to demonstrate what a biological network can represent, we do not include edge-specific features in the GSNN modeling.

The resources listed in Table 20 span many pathways and processes, of which many are likely to be irrelevant to gene regulation or common chemical perturbation pathways. Furthermore, the number of trainable parameters in the GSNN model scales with the input graph size. For these two reasons, we developed a process to select a subset of proteins that are likely to be predictive of chemical perturbation signaling processes. This graph construction process is based on a set of user-defined Reactome pathways. We then define the intersection of pathway proteins as the *protein-space* of $\mathcal{G}$. Next, we define the *rna-space* as all RNAs that are targeted by a member of the *protein-space*; Optionally, the user can provide a depth and to which RNA descendents are included, which allows for construction of more complex gene regulation networks (e.g., protein->miRNA->mRNA). Any drugs that have a reported protein target in the *protein-space* are then added to the *drug-space*. The *LINCS-space* is the intersection of available LINCS genes and *rna-space*. We then include all edges from the resources listed in 20 if both source and destination are in one of the node-spaces (the user can optionally choose a subset of resources to include). Next, there is a drug pruning step based on how many LINCS nodes are descendants (the default is 25% of the number of LINCS nodes). This pruning step is used to remove drugs that are overconstrained by the available prior knowledge and therefore cannot impact the prediction of most outputs. PROTEIN and RNA nodes are also pruned if they do not have downstream LINCS genes. OMICS nodes are added based on the intersection of available 'omic features' and protein-space or rna-space. In this construction process, an OMIC node (e.g., EXPR__P53) can have edges to a PROTEIN, RNA, or both (example: see $prot_1$ and rna_2 in Figure 44). For instance, we rationalize that RNA expression is relevant to both RNA and protein behavior, and therefore we make the respective OMICS easily accessible to both.

Table 20: Literature curated resources documenting molecular interactions.

Resource Type	Resource	Expert reviewed	Source Node	Target Node	Refs
Drug-Target	The Cancer Targetome	Some	Drug	Protein	[BCKM$^+$17]
Drug-Target	CLUE Repurposing	No	Drug	Protein	[CBL$^+$17]
Drug-Target	STITCH	No	Drug	Protein	[KvMC$^+$07]
Protein-protein interaction	Omnipath	Some	Protein	Protein	[TKSR16]
Transcription Factor Gene Regulation	Omnipath (Dorothea)	Some	Protein	RNA	[TKSR16, GAITSR18]
miRNA Gene Regulation	Omnipath (MiRNA)	Some	RNA	RNA	[TKSR16]
Translation	NA	NA	RNA	Protein	NA

Protein inclusion based on pathways makes the assumption that all output nodes (LINCS expression) in the graph can be well predicted via the entities we include; however, this assumption is likely violated in many cases, especially for small pathways. Including larger or multiple pathways could mitigate this issue, allowing for cross-talk between pathways. The exclusion of a drug's targets is likely to prevent the GSNN method from effectively predicting the response to that drug. Similarly, exclusion of critical transcriptional regulators of an RNA molecule may lead to worse performance on the respective LINCS output. Therefore, we manually select pathways to balance the graph size (so that the GSNN method is memory- and compute-efficient), while including molecular entities relevant to drug-response.

6.3.6 Baseline Models

We compare the performance of the GSNN algorithm against several baseline algorithms: Artificial neural network (NN) [MP43], Graph Convolutional Network (GCN) [KW16], Graph Attention Network (GAT) [VCC+17b] and Graph Isomorphism Network (GIN) [XHLJ18]. We also compare performance against a "cell-agnostic" Neural Network, in which all cell-context specific features are removed (e.g., expression, mutation, methylation, and copy-number variation features are removed from the input). This provides a baseline that characterizes the prediction of the average response of each drug, across all cell lines.

We implement the NN baseline using two layers with the Exponential Linear Unit (ELU) activation function [CUH15] and batch normalization [IS15]. All Graph Neural Networks (GCN, GAT, GIN) are implemented with batch normalization layers, jumping knowledge [XLT+18], and use the ELU activation function.

6.3.7 Hyper-parameter Tuning

For all three models, we performed a limited hyperparameter tuning within each MCCV fold, and the parameters tested are reported in Table 21.

Table 21: Hyper-parameter grid search performed in each experiment for the five tested algorithms. The "Other" line indicates additional Boolean flags that describe specific algorithm behavior.

Hyper-Param.	GSNN	NN	GIN	GAT	GCN
Channels	10, 20	100, 500, 1000	64, 128	64, 128	64,128
Layers	10, 20	2	5, 10	5, 10	5,10
Dropout	0, 0.25	0, 0.25, 0.5	0	0	0
Learning Rate	1e-2, 1e-3	1e-2, 1e-3	1e-2, 1e-3	1e-2, 1e-3	1e-2, 1e-3
Other	add_self_edges, scale_channels_by_degree				

6.3.8 Drug Dose Transformation

To encode the presence of a drug, each drug node is assigned a scalar value that represents the concentration of that drug in an observation. We transform the drug concentration (μM) using the following function:

$$x_{dose} = -\frac{log_{10}(\mu M + \epsilon) - log_{10}(\epsilon)}{log_{10}(\epsilon)}$$

This transformation ensures that the drug effect is log-linear in the relevant therapeutic concentration range, and which can be shifted by choice of ϵ. It also maintains that a concentration of $0\mu M$ (i.e., no drug) will still be equal to zero after transformation and a concentration of 1μM (a typically large dose) will be equal to one after transformation. Most

of the LINCS L1000 observations were measured with concentrations between -2.5 and 2.5 ($log_{10}(\mu M)$), and choosing an epsilon of $1e-6$ ensures a logarithmic linear relationship in the concentration range relevant to the LINCS L1000 dataset.

6.3.9 GSNN Explanation: Edge Importance Scores

An advantage of the GSNN algorithm is that we can use the network structure to explain predictions in a form that is amenable to biological interpretation. Previous work has presented a method for explaining GNN predictions, termed *GNNExplainer*, which identifies a subset of nodes and features that are most involved in the prediction of a given observation [YBY+19]. We implement a similar method to explain a given observation predicted by the GSNN network; however, rather than identifying a subset of nodes or features, we identify a subset of *edges*. Given an observation (x), baseline observation (x_b), and a trained model (f_{GSNN}), we initialize an edge mask (M) and use gradient descent to identify a subset of edges that result in comparable model predictions. Further details are available in Algorithm 3. The output of the *GSNNExplainer* method is a subgraph, which includes the most critical or involved edges to predict a given observation compared to a baseline prediction. A key distinction between the *GSNNExplainer* and the *GNNExplainer* is our use of a baseline prediction. GNNs traditionally use a permutation invariant aggregation scheme such as *mean, max* or *sum*, and as such, removing edges or nodes in the GNN setting is analogous to setting the respective latent state to zero; however, in the GSNN algorithm, edges may have some level of endogenous latent activation (perhaps related to 'omic inputs), even in the absence of drug inputs, and this means that setting an edge latent state to zero is not necessarily representative of "removing" an edge. In the GSNN algorithm, setting an edge latent value to zero is likely to be more analogous to setting the edge to the "average" signaling behavior of the training dataset, which is unlikely to represent any specific observation. To mitigate this challenge, we use a *baseline* observation to characterize the baseline latent edge values. For example, to explain the prediction of a given drug $Drug = A$, concentration $Conc = c$ in a cell line $Line = l$, we can use a baseline observation without any drug ($Drug = A, Conc = 0, Line = l$). Using such a baseline allows the *GSNNExplainer* to ignore any endogenous latent activation that may be due to cell type and instead focuses on the edges involved in drug response prediction. Alternatively, one could ask the question *What are the key edges involved in the prediction differences between two different cell lines?* To do this, we can use one cell line as the baseline observation and the other as the input observation, while keeping the drug concentrations the same for both cell lines.

There are a few additional divergences from the methods presented in *GNNExplainer*: First, to induce discrete sampling of edges, we use the softmax-gumbel distribution and anneal the temperature parameter during optimization [MMT17, JGP17]. Second, we use the mean-squared error (MSE) loss function, which is convenient for multi-output regression. In practice, we initialize the weight mask parameters such that each edge is likely to be selected, thus the optimization starts with almost all edges included in the mask. A strong mask penalty term (β) encourages the removal of uninvolved edges during optimization. We also include an optional weight decay coefficient (γ), intended to encourage exploration by preventing large (confident) edge parameters. In summary, *GSNNExplainer* produces a

Algorithm 3 GSNNExplainer

INPUT: A trained GSNN model f_{GSNN}, an observation to explain x, an observation x_b to use as *baseline*, number of optimization iterations N, mask size penalty coefficient β, weight decay term γ, number of "free" edges E, learning rate α, and initialization prior P.
OUTPUT: Edge importance scores that characterize which edges are critically involved in the prediction of a given observation compared to a baseline (x_b).
$\theta \leftarrow init(P)$*
$target = f_{GSNN}(x) - f_{GSNN}(x_b)$
$a_b \leftarrow layerwise\ \ activations\ \ of\ \ f_{GSNN}(x_b)$
$g_{GSNN}(X, X_b, M, A) \leftarrow masked\ \ forward\ \ operation$**
for $i = 1$ to N **do**
    $m_{ij} \leftarrow GumbelSoftmax(\theta)$
    $output_{ij} \leftarrow g_{GSNN}(X = x, X_b = x_b, M = m_{ij}, A = a_b) - f_{GSNN}(x_b)$
    $\mathcal{L}_{ij} = MSE(target, output_{ij}) + \beta((\sum M) - E) + \gamma||\theta||_2$
    $\mathcal{L}_i \leftarrow \frac{1}{B} \sum_{j=0}^{B} \mathcal{L}_{ij}$
    $\theta \leftarrow \theta - \alpha \nabla \mathcal{L}_i$
end for

**The prior P sets the initial probability of a given edge. In practice, setting a high probability of edge inclusion (i.e., the subgraph $\sim$ full graph) paired with a large β term leads to robust subgraphs.*

***The **masked forward operation** (g_{GSNN}) modifies the GSNN forward operation such that edges not included in the mask ($M = 0$) will be set to the baseline latent activation values. The edges included in the mask ($M = 1$) will be unchanged.*

biologically relevant explanation of an observation. Using this method, we can interrogate which molecular entities and entity interactions are important for the prediction of an observation.

6.3.10 Drug Prioritization

One of our primary research goals is to use the GSNN method for effective and interpretable prioritization of drugs for nuanced research goals. We base our prioritization procedure on the premise that drugs that induce the same response in all cell lines are unlikely to be good therapeutic candidates due to the detrimental effects on normal cell types. Rather, we seek to identify drugs that induce a selective response in a subset of user-designated cell lines representative of their research goals. For example, researchers may seek a selective response in cell lines derived from a certain primary disease (e.g., breast cancer), or cell lines with shared mutations (e.g., TP53) or an expression pattern (e.g., HER2+).

First, the user must define the contextual response of an ideal drug candidate. This can be done by designating in which cell lines a candidate drug should cause a *desirable* response, termed the "target" lines (T), and the cell lines that should have a less *desirable* response, termed the "background" lines (B). For example, to prioritize drugs that have a selectively desirable response in breast cancer cell lines, we can assign all breast cancer derived cell lines to the "target" set and all other cell lines to the "background" set:

$$T = \{SKBR3, BT20, MCF7, etc.,\}$$

$$B = \{all\ other\ lines\}$$

Next, we must choose a metric to quantify the *desirable* response. A desirable response is often measured in terms of cell viability[12](the proportion of cells alive after the treatment of a drug), where a low cell viability indicates a desirable response. Although cell viability is a convenient measurement for cancer drug response, there are alternative metrics that can be tailored to more specific research questions. For instance, metrics derived from gene expression signatures or single gene expression values may be more appropriate for certain research goals, such as quantifying DNA damage or the activation of apoptotic pathways. For this work, we chose to quantify the "sensitivity" of a response using cell viability such that low cell viability indicates sensitivity to a drug. We train a probabilistic cell viability predictor network (f_{viab}) using the output of a trained GSNN, such that:

$$p(\hat{y}_{viab}) = f_{viab}(f_{gsnn}(x))$$

The GSNN parameters are frozen so that they do not change during the optimization of f_{viab}. We use deep ensembles as described by Blundell et al. [LPB17] to quantify the uncertainty in the outcome variable and assume that cell viability is generated from a Beta distribution. Cell viability values are clipped between 0 and 1. Individual viability networks predict two concentration parameters (a, b), which characterize the predicted cell viability probability distribution: $P(\hat{y}) = Beta(a, b)$. We parameterize f_{viab} using a 1-layer neural network and optimize the parameters using the negative log-likelihood and the PRISM dataset [CNS$^+$20], which characterizes cell viability after drug treatment in a range of doses. We include all (drug, cell) PRISM observations for which the drug is included in the GSNN model and the cell is included in our GSNN training data. We maintain the same validation and test partitions as used for GSNN training.

To prioritize drugs that create a selective cytotoxic response, we calculate the probability that sensitive lines have lower cell viability than resistant lines. To do this, we define the "target" cell viability probability ($P_T(\hat{y}_{viab})$) as a mixture over the "target" lines. Respectively, the cell viability probability of the "background" cell lines is defined as a mixture over all members within the background set ($P_B(\hat{y}_{viab})$). We then compute the difference in cell viability between the "target" and "background" lines. Finally, we compute the probability that the target cell lines have an average cell viability lower than the average cell viability of the resistant lines (p_{sens}).

$$P_T(\hat{y}_{viab}) = \frac{1}{|T|} \sum_{i \in T} p_i(\hat{y}_{viab})$$

$$P_B(\hat{y}_{viab}) = \frac{1}{|B|} \sum_{j \in B} p_j(\hat{y}_{viab})$$

$$p_{sens} = P(\bar{y}_T < \bar{y}_B | drug = d) = \int_{-\infty}^{0} (P_T(\hat{y}_{viab}) - P_B(\hat{y}_{viab}))dy$$

[12]or metrics derived from cell viability measured over a range of concentrations such as the area under the dose response curve (AUC) or the half maximal inhibitory concentration (IC50)

The drugs are then ranked by p_{sens} to produce a prioritized list such that the top candidates are the most likely to be selectively responsive as defined by the "target" and "background" cell lines. Recognizably, an analytical solution for a mixture of Beta distributions is not convenient, so instead we use Monte Carlo simulations [RC05] to approximate p_{sens}.

To evaluate the effectiveness of our drug prioritization method, we use disease indications provided by the Drug Repurposing Hub dataset [CBL$^+$17]. This dataset provides limited drug annotations that specify the disease(s) for which a drug has FDA approval. To evaluate the rationality of our prioritization algorithm, we generate drug rankings (as described above) where the "target" set includes all cell lines derived from a certain primary disease, termed *target disease*, and the "background" is a set of all cell lines derived from *background disease*. Our expectation is that drugs with FDA indications for the *target disease* will be prioritized over drugs with indication *background disease*. Drugs with indication for multiple diseases are not considered. We then quantify the prioritization results using the area under the receiver operator curve (AUROC) such that drugs with an indication for *target disease* are assigned a label of 1 and drugs with an indication for *background disease* are assigned a label of 0. A perfect ranking (AUROC = 1) would be achieved if all drugs with indication for *target disease* are ranked higher than drugs with indication for *background disease*. Notably, using disease-specific target and background sets is not an ideal comparison for drug repurposing applications, and we would, if possible, use normal (non-cancerous) cell lines as the background set. Unfortunately, there are very few normal cell line models available in the LINCS dataset and the available drug indications are not well formulated to compare to normal cell response. On the other hand, using a cancer background can be envisioned as a method for identifying drug sensitivities that are unique to a specific cancer type, and this is especially important when exploring drug combinatorics as there is ample risk of the drug combinations being toxic to all cell lines. From this perspective, we could potentially use all non-target lines or even random cancer lines, as a surrogate for general cellular response, or a "background" toxicity readout.

6.4 Results

We apply our method to model cellular signal transduction using literature-curated prior knowledge to create a biological network and optimize model parameters using the LINCS L1000 dataset. The LINCS dataset characterizes cell line RNA expression changes in response to chemical perturbations. The L1000 assay measures 978 genes directly and then uses these *landmark* genes to infer the RNA expression of approximately 12 thousand more genes (the "best inferred" and "inferred" features spaces) [SNC$^+$17a]. We construct a biological network relating proteins, drugs and RNA using the Omnipath resource [TKSR16], CLUE compound information [CBL$^+$17], the Cancer Targetome [BCKM$^+$17] and the STITCH database [KvMC$^+$07].

To limit memory requirements and focus on relevant signaling entities, we generate biological subgraphs that pertain to specific biological pathways. These subgraphs allow us to evaluate the performance of our method on several distinct pathways and data subsets. We define an "experiment" as a choice of biological pathway and network construction hyperparameters, and list the evaluated experiments in Table 22a and additional experiment details are described in

Appendix 9.6. The experiment parameter choices will define which biological entities are included in the biological network, as well as the cell lines, drugs, and observations that are applicable to the respective experiment. In addition, each experiment will randomly assign train, test, and validation partitions. To mitigate the variance in performance due to partition sampling, we run each experiment in replicate ($n = 3$), such that each replicate will share all hyperparameters, but will have unique data splits and weight initialization. Within each replicated experiment, the validation set is used to select the best performing model from a hyperparameter grid search, and we report the average test performance in Table 22b. Table 22a describes the experiment hyperparameters and lists the "primary pathways" that were used in network construction; for further details regarding the experiment construction parameters, see Supplemental section 9.6. Table 22c reports the results of a paired t-test comparing the performance of GSNN and NN (the two algorithms with the highest performance). The GSNN algorithm obtains the highest mean Pearson correlation in all three experiments and significantly outperforms (Family-wise error rate (FWER) < 0.05) the NN in experiment 1. Of the GNN algorithms, the GIN obtains the highest mean Pearson correlation but still underperforms compared to the NN and GSNN algorithms.

The resulting GSNN and NN models, for which performance is described in Table 22b had roughly the same number of parameters and are described in more detail in the Appendix section 9.7. Additionally, the GSNN algorithm is slower to train, owing to the added complexity of the enforced latent variables (graph structure), and is discussed in more detail in the Appendix section 9.8. From the three experiments that were performed, we find that training a GSNN model requires 10-15x more time than a traditional feedforward neural network and 3-4x more time than a graph neural network that operates on the same graph structure. That said, as illustrated in Appendix figure 57, we find that the GSNN commonly converges more rapidly than traditional neural networks, and early stopping during model optimization is likely to narrow the difference in training times between the GSNN and NN. Additionally, the runtime and memory consumption scales with the number of GSNN layers, and due to the available compute resources, we were required to train GSNNs with fewer than 20 layers. The results of Appendix Figure 9.9, suggest that deeper GSNNs tend to perform better.

6.4.1 Local Performance Advantages

Perturbation biology requires the prediction of many outputs based on numerous inputs. Primary sources of variance in response to a perturbation include drug, concentration, and cellular context. In such a complex system, it can be useful to inspect the *local* performance of a predictive model, which we define as the performance grouped by attribute or performance within a subset of outputs. *Local performance* can help investigate questions such as: *Which drug(s) perform best? Which genes (outputs) are well predicted?* Investigating such model behavior can help highlight where the GSNN method works well and where it falls short.

The GSNN model is constrained by the biological network ($\mathcal{G}$) constructed from literature-curated datasets. It is therefore likely that the biological network will have quality biases toward well-studied pathways or relevance to certain cellular contexts (i.e., the contexts that are most commonly studied). Additionally, the choice of network

Table 22: GSNN performance results and comparisons.

EXP.	Primary Pathway(s)	LINCS Feature-space	time	Nodes (in, func, out)	# Edges	# drugs	# cell lines	# obs
1	Signaling by EGFR, Signaling by ERBB2	landmark	24H	5369 (3505, 1411, 453)	16203	516	80	40527
2	Death Receptor Signaling	landmark	24H	6466 (4210, 1789, 467)	21830	554	80	41768
3	Signaling by ALK	landmark	24H	5871 (3816, 1581, 474)	18897	569	80	43682

(a) Experiment input network characteristics and hyper-parameters.

EXP.	GSNN	GSNN (randomized)	NN	NN (cell agnostic)	GIN	GAT	GCN
1	**0.54 (0.51,0.56)**	0.44 (0.39,0.47)	*0.52 (0.48,0.53)*	0.39 (0.38,0.4)	0.26 (0.16,0.36)	0.1 (0.07,0.12)	0.12 (0.06,0.19)
2	**0.52 (0.45,0.6)**	0.41 (0.3,0.5)	*0.5 (0.41,0.58)*	0.38 (0.36,0.4)	0.26 (0.07,0.45)	0.11 (0.04,0.22)	0.11 (0.04,0.18)
3	**0.51 (0.49,0.52)**	0.38 (0.35,0.41)	*0.47 (0.44,0.48)*	0.36 (0.34,0.38)	0.1 (0.08,0.13)	0.06 (0.04,0.07)	0.06 (0.04,0.08)

(b) The mean (95% confidence interval) Pearson correlation of the best performing models of experiments 1-3 measured on a hold-out test set, aggregated over all MCCV folds (n=3). **Bold** font indicates the greatest average Pearson correlation, *italics* indicates second best performance.

EXP.	H0	p-value [adj.]	GSNN scores [mean]	NN scores [mean]
1	NN > GSNN	0.005 [0.014]	0.56,0.51,0.56 [0.54]	0.53,0.48,0.53 [0.52]
2	NN > GSNN	0.034 [0.102]	0.44,0.52,0.61 [0.52]	0.41,0.49,0.59 [0.50]
3	NN > GSNN	0.022 [0.065]	0.51,0.52,0.49 [0.51]	0.48,0.48,0.44 [0.47]

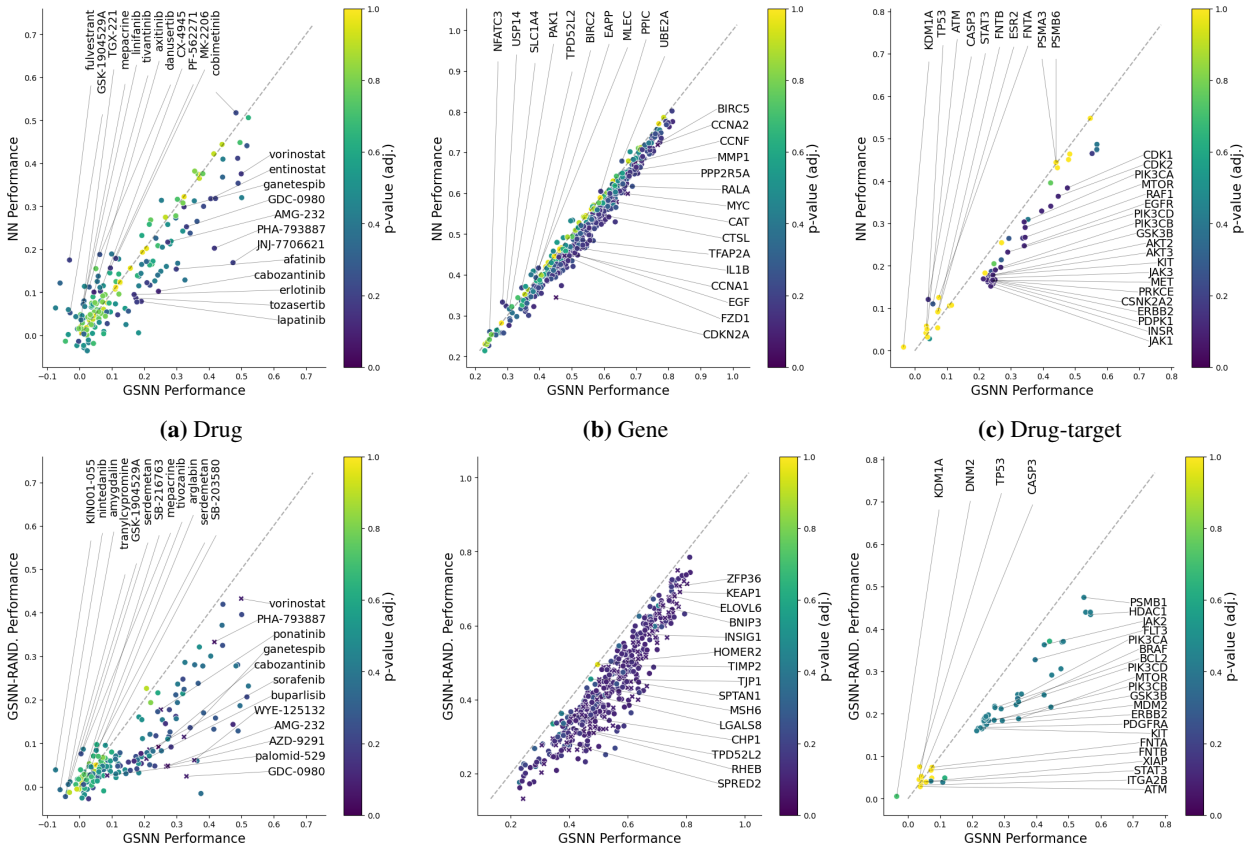
(c) Comparison of GSNN vs NN pearson correlation scores with a two-sided paired t-test. GSNN and NN scores are ordered by replicate fold (1-3).

construction hyperparameters (particularly the choice of pathways) may benefit the prediction of a subset of drugs or genes. A plausible outcome is that such biases will translate to local regions of the biological network that are more useful or accurate in predicting the expression change of particular genes. Alternatively, certain drugs, cell lines, or observations may be particularly well predicted because of the GSNN inductive bias. To investigate this, we examine the *local* performance advantages of the GSNN model compared to the NN (top row of Figure 48) and the GSNN model initialized with random prior knowledge ("GSNN-rand"; bottom row of Figure 48). We used the best model from each fold (N=3) to evaluate performance grouped by various attributes. For example, 48a shows the results of the observations grouped by drug; all observations with a given non-zero drug concentration are grouped, and the performance is computed as an average across all outputs and experiment replicates. We evaluate group-performance in the test set and tested the statistical significance of each group comparison using a paired two-sided t-test and adjusted for multiple tests using the Benjamini-Yekutieli or Benjamini-Hochberg[13] false discovery rate (FDR) method [BY01, BH95] and an FDR threshold of 0.1.

The drug targets with the most significant prediction advantage are proteins that are integrally involved in the pathways chosen for Exp. 1 including EGFR, ERBB2 and CDKs. The two most significant drug-targets with prediction advantage by the NN are CASP3 and ATM. Furthermore, there are many drugs and drug-targets that are poorly predicted by both the GSNN and NN algorithms ($r < 0.2$), which may suggest poor data quality (e.g., noisy L1000

[13]The Benjamini-Hochberg method is used for local performance groupings by cell line, gene, drug and disease where the groups are disjoint sets and therefore the p-values are independent. The Benjamini-Yekutieli method is used for local performance groups by drug-target where an observation may be assigned to two or more groups and therefore p-values are not independent

measurements) or low data volume. There are also several drugs that are predicted quite well (r>0.5) by both GSNN and NN, and this may suggest that these drugs induce a simple response (i.e., "easy" to predict) or have high data volume (many measurements for the given drug).



(a) Drug **(b)** Gene **(c)** Drug-target

(d) Drug **(e)** Gene **(f)** Drug-target

Figure 48: Test performance comparison between the GSNN model (x-axis) and either the NN (top row; y-axis) or Randomized-GSNN (bottom row; y-axis) when grouped by several attributes (Drug (left), gene/output (middle) or drug-target (right)). Performance is reported as the Pearson correlation averaged over predictions from the best model from each MCCV fold of Exp-1 (EGFR + ERBB2 signaling). The gray dashed line on all plots represents equivalent average performance across folds; any groups lying under the dashed line are better predicted by the GSNN algorithm (i.e., "GSNN performance advantage"). Significance was determined using a paired t-test (n=3) and groups with a p-value less than 0.1 are denoted with an 'x'. Groups with fewer than five observations in each fold (n=3) are not included. Note that we have limited power to detect significant performance differences because we only ran three replicates.

6.4.2 GSNN Performance on random networks

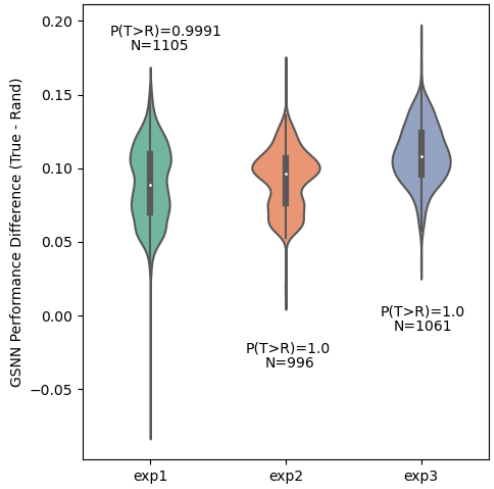
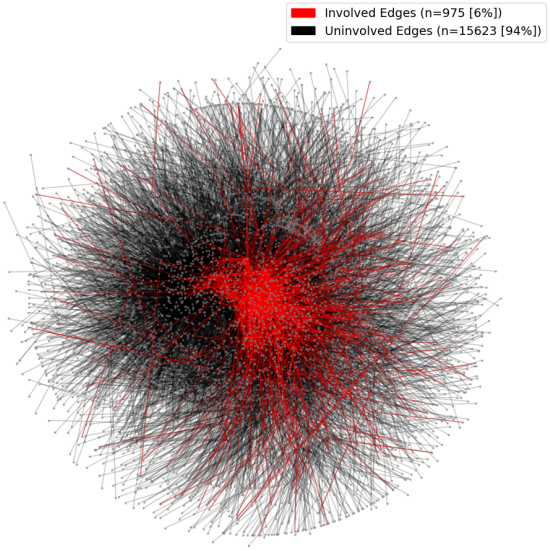


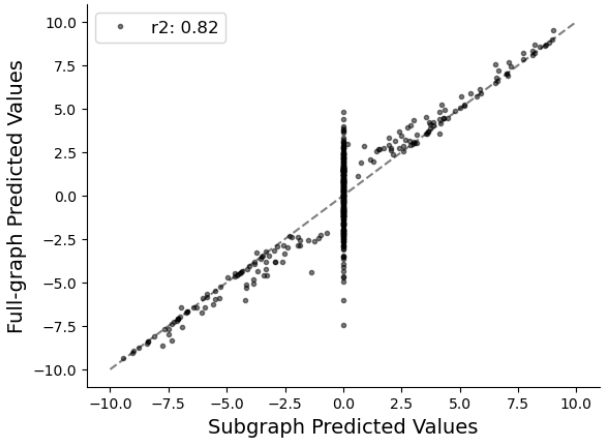
Figure 49: The performance differences between hyper-parameter and replicate matched GSNN-true (true bio. network) and GSNN-rand (random bio. network). Performance evaluated with Pearson correlation and differences computed by $\Delta r = r_{GSSN-true} - r_{GSNN-rand}$. P(T>R) is calculated as the proportion of positive differences (i.e., proportion of GSNN-true performances that are greater than GSNN-rand performances).

The premise of our research assumes that prior knowledge encoded in the biological graph will be useful for the prediction of the endogenous variable. To test this assumption, we compare the performance of the GSNN algorithm when using true or randomized biological graphs. We find that GSNN models that use the true biological graph result in an average performance improvement of 24% (95% CI: 11.4% - 40.1%) compared to hyperparameter-matched GSNN models that use a randomized biological graph. Figure 49 shows the performance advantage when using the true biological network. Randomization of the biological network decreases performance in all three experiments. Interestingly, the randomized GSNN model outperforms the cell-agnostic NN and all three GNN algorithms, suggesting that the GSNN model is capable of accurate predictions despite random inductive bias. Of note, the biological graph prior knowledge that the GSNN utilizes has two forms of inductive bias, 1) the interactions between molecular entities and 2) the molecular entities themselves. Our randomization scheme can only remove inductive bias from molecular interactions and there may still be predictive value in the knowledge of molecular entities themselves.

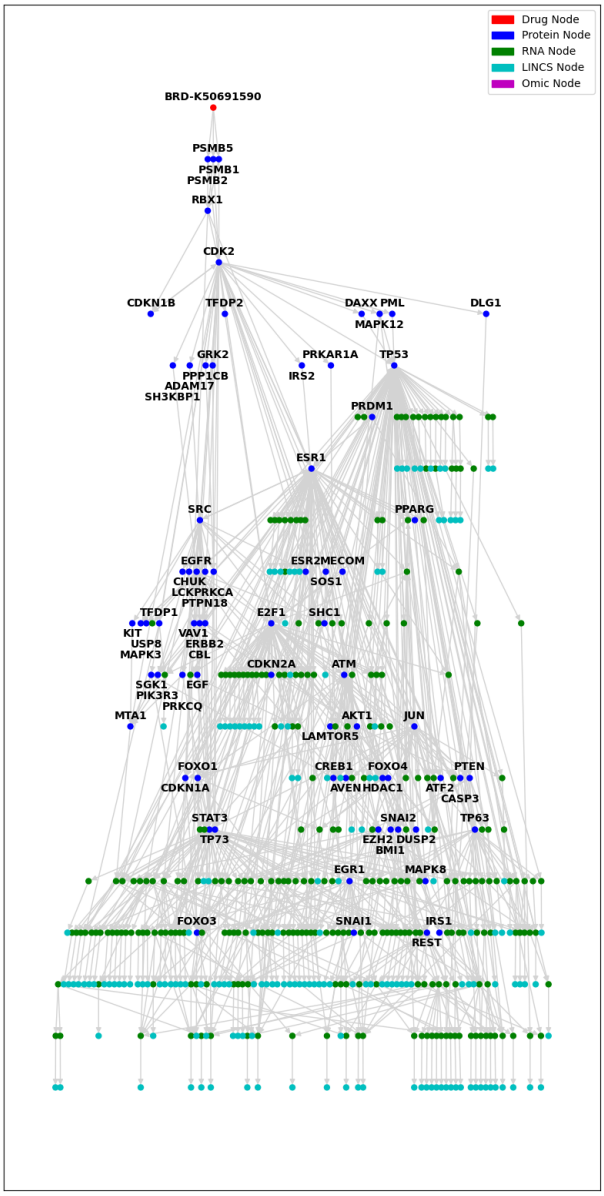
6.4.3 Explaining Predictions using Edge Importance Scores



(a) An uninterpretable example of the full GSNN Exp. 1 structural graph $\mathcal{G}$ (black) overlaid with the *involved* edges (red).



(b) The predicted values when using the subgraph displayed in (c) versus predicted values with the full graph.



(c) The red *involved* edges from (a) with nodes organized hierarchically and annotated.

Figure 50: The *GSNNExplainer* method can be used to identify a subgraph that maintains comparable prediction outputs. This can help delineate which edges are involved in the prediction of a given outcome. In this example, we explain the prediction of gene expression response to a proteasome inhibitor (BRD-K50691590) in the PC3 cell line (prostate cancer). For this observation, predictions using the subgraph shown in (b) maintains 82% of the variance of the full-graph prediction.

Significant progress has been made toward the interpretation of traditional neural networks, including attribution methods [LL17, STY17] and black-box explanation methods [RSG16]; however, applying these methods to cellular signaling models does not express model prediction logic in a biologically relevant form. This limitation is exacerbated

Table 23: The Pearson correlation between importance scores generated by subsequent GSNNExplainer replicates (n=5). Performed on Dabrafenib (BRD-K09951645) in A375 cell line at a 10uM dose. The average pairwise correlation between replicates is 0.92.

	repl_0	**repl_1**	**repl_2**	**repl_3**	**repl_4**
repl_0	1.000000	0.920050	0.922028	0.924731	0.921699
repl_1		1.000000	0.919183	0.920203	0.917493
repl_2			1.000000	0.922614	0.919730
repl_3				1.000000	0.921099
repl_4					1.000000

in traditional neural networks, since the prediction logic is unlikely to accurately capture the sequence of molecular interactions, and therefore even accurate explanations of traditional neural network logic are not necessarily useful to understand the underlying biology. The network-based architecture of the GSNN presents new approaches to interpret explanations in a way that may be more useful to biologists. To do this, we implement a method called *GSNNExplainer*, inspired by previous work [YBY+19], which explains an observation by identifying a subset of edges that are "important" to the prediction of the model.

Subgraph explanations produced by the *GSNNExplainer* can be conceptualized as a testable hypothesis of the true underlying drug response. For example, the subgraph shown in Figure 50c highlights the key molecular entities that the Exp. 1 GSNN model uses for the prediction of outcome variables. Subsequent work could validate these explanations by comparison with the literature or by performing wet-lab bench testing to confirm the involvement of molecular entities. For example, Figure 50c clearly shows ESR1 (Estrogen Receptor) has an important role in the prediction of the response to BRD-K50691590 (proteasome inhibitor). To confirm or deny the involvement of this transcription factor, researchers could use experimental assays, such as chromatin immunoprecipitation sequencing (ChIP-seq) [Par09], to investigate the activity of ESR1 (or other transcription factors implicated) in response to BRD-K50691590. Representing drug response explanations as a testable hypothesis can be used to build knowledge (e.g., identify important molecular entities) or to encourage user trust in the model (through validation of explanations). Although traditional neural networks can explain predictions using methods such as SHAP [LL17], explanations can only relate inputs and outputs and lack a representation of intermediate molecular entities and therefore can be challenging to use as a testable hypothesis of the underlying biology.

Since the GSNNExplainer utilizes the stochastic softmax-gumbel distribution to induce discrete edge selection, one concern is the repeatability of the edge importance scores generated by subsequent GSNNExplainer replicates. We investigated this concern by running replicates (n=5) and computing the Pearson correlation of subsequent replicates; the results of a representative use case are shown in Table 23. On average, the pairwise correlation between replicates for this example is greater than 0.9, suggesting repeatable results. The small amount of discordance between replicates is likely due to a tendency for the GSNNExplainer to focus on outputs with larger prediction values, thus ignoring low-value predictions (which are more likely to be dominated by measurement noise). Considering the abundance of

predicted outputs, it is likely that there is a stochastic inclusion of low-value predicted outputs. The repeatability between replicates of the GSNNExplainer can be further improved by averaging the edge importance scores between replicates; for example, the pairwise averaging of replicates in Table 23 improves the average pairwise correlation to 0.96. This result suggests that, for applications where repeatability is critical, multiple GSNNExplainer replicates should be computed and the results averaged.

6.4.4 Evaluation of predicted cell viability

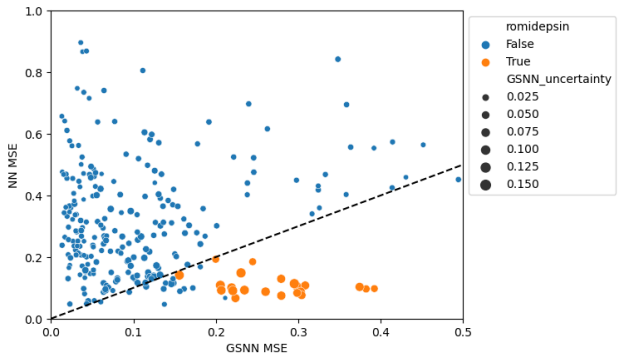


Figure 51: Cell viability prediction error (MSE) computed within drug combinations. X-axis characterizes GSNN error and Y-axis characterizes NN performance; all drug combinations above the diagonal (black dashed line) are better predicted by the GSNN. Notably, all combinations with Romidepsin (orange) were poorly predicted by the GSNN. The size of the points indicates the predicted uncertainty of the GSNN (mean variance computed within combination), which should correlate with GSNN error if the predictions are well calibrated.

Perturbed expression can be useful to characterize the multifaceted response of a biological system; however, it is sometimes convenient to measure the response of a perturbation by a simple phenotypic outcome. For example, the field of *cancer drug response* (CDR) predominantly uses cell viability (or summary metrics) as the outcome variable. Past research has shown that perturbed expression can be used to identify cell death signatures and that perturbed expression can be used to accurately predict cell viability [SSH+19, LCQ21]. In this section, we train a deep ensemble of probabilistic 1-layer neural networks (f_{viab}) to predict cell viability from predicted perturbed expression.

$$f_{cdr} \sim f_{viab}(f_{expr}(x))$$

Where f_{expr} is a frozen[14] GSNN or NN model from Exp. 1. We optimize f_{viab} using single-agent cell viability data from the PRISM dataset, and evaluate on a hold-out test dataset (62 cell lines, 305 drugs, ∼115k observations). We then used the NCI ALMANAC dataset [HCC+17] to assess performance when applied to unseen two-drug combinations (13 cell lines, 24 drugs, 264 combinations, ∼30k observations). The primary purpose of this evaluation is to compare the relative usefulness of the expression predictions made by the GSNN or NN models. If the GSNN

[14] *Frozen* indicates that the parameters of f_{expr} are not updated during optimization of f_{viab}

predictions are more robust and mechanistically grounded, then we expect them to perform better in downstream applications such as prediction of cell viability (single and combination agents).

Table 24: Predicted cell viability performance. Values in bold indicate the best performance. The column acronyms are Mean Squared Error (MSE), Expected Calibration Error (ECE), Error-Variance Correlation (EVC), Accuracy (Acc), Area under the receiver operator curve (AUROC). The "Null/Rand." row refers to metrics computed using random predictions (uniform(0,1)).

f_{expr} arch.	f_{viab} arch.	MSE	r (pearson)	ECE
GSNN	NN	**0.017**	0.87	0.11
NN	NN	0.022	**0.90**	**0.11**
Null/Rand.	Null/Rand.	0.263	-0.02	NA

(a) Performance evaluated on a hold-out test set of single agent drugs (PRISM dataset).

f_{expr} arch.	f_{viab} arch.	MSE	r (pearson)	ECE	EVC	Acc. (y,$\hat{y} > 0.5$)	AUROC ($y > 0.5$)
GSNN	NN	**0.16**	**0.30**	**0.28**	**0.22**	**0.75**	**0.7**
NN	NN	0.34	0.22	0.35	-0.24	0.46	0.63
Null/Rand.	Null/Rand.	0.27	0.00	NA	NA	0.5	0.5

(b) Performance evaluated on unseen two-agent drug combinations (NCI Almanac dataset).

The results of cell viability predictions for single-agent and two-drug combination are shown in Table 24. We find that single-agent cell viability predictions from a GSNN model perform comparably to predictions made with a NN model; however, the GSNN model markedly outperforms the NN model when predicting cell viability for two-drug combinations with a mean squared error (MSE) of less than half the NN model. To evaluate the calibration of cell viability predictions (i.e., the quality of uncertainty quantification), we use the expected calibration error (ECE) [GPSW17] and the correlation between predicted variance and error (EVC). In Figure 51 we compare the performance GSNN vs NN predictions when grouped within two-drug combinations (i.e., performance calculated over all doses and cell lines tested with a given drug combination). The GSNN has lower MSE for almost all drug combinations tested, with the notable exception of combinations involving the prodrug[15] Romidepsin, which inhibits histone deacetylases (HDACs) [WKG+06].

It is important to note that training and evaluating a model on different datasets can introduce several limitations. For example, a seemingly marginal performance of $\sim$0.3 Pearson correlation should be interpreted in the context of potential data shift issues. This consideration is particularly relevant for cell viability measurements, which are known to suffer from limited reproducibility and measurement noise [NHM+19a]. Additionally, the datasets used in our study,

[15] a drug that is activated once inside the cell.

namely the single agent data from PRISM [CNS+20] and the combination agent data from the NCI Almanac [noaa, HCC+17], were generated using different cell viability assays (PRISM and NCI-60 protocol, respectively). This disparity raises the possibility of a covariate shift [QCSSL09], which may affect our results. To address these concerns, we refined our evaluation by binarizing cell viability (y > 0.5) and employing classification metrics, specifically accuracy (Acc.) and area under the receiver operator curve (AUROC). The results in Table 24 show that the GSNN model demonstrated superior performance across all classification metrics when predicting the cell viability of two-drug combinations.

6.4.5 Disease-specific drug prioritization

In this section, we use the GSNN model (same hyperparameters as Exp. 1) to rank drugs by their selective response in cell lines derived from specific cancer types. We then evaluate the rationality of these prioritizations by comparing them with a limited number of FDA-approved drug indications. We manually select target and background diseases in Table 25 by choosing diseases that have cell lines in the GSNN cell-space and indications in the PRISM drug repurposing dataset [CNS+20]. We use Monte Carlo simulations (N=1000) to estimate the probability that target cell lines have lower average cell viability than background cell lines (p_{sens}). We then rank drugs by p_{sens} and report the performance of our ranking in Table 25. The reported p-values are calculated as the proportion of null-AUROC values (e.g., random drug lists; N=1000) that are greater than our AUROC value. The GSNN prioritization performs well across disease types with perfect AUROC values in 8/10 disease-specific prioritizations; however, this approach would benefit from a greater number of drug indications to accurately measure significance. For example, Acute Myeloid Leukemia (AML) and Prostate Cancer have only a single drug with indication for these diseases, and prioritizations involving these diseases do not have significant p-values (alpha=0.05) even though they achieved a perfect AUROC score (i.e., all drugs with the target disease indication are prioritized before drugs with the background disease indication). These results should also be interpreted with caution, as the drug indications in the PRISM drug repurposing dataset may not be up-to-date or may not capture ongoing research. For instance, when prioritizing drugs that are preferentially cytotoxic in breast cancer compared to non-small cell lung cancer (NSCLC) our algorithm ranks Afatinib and Gefitinib (EGFR inhibitors with indications for NSCLC) above drugs with breast cancer indications. Although this initially appears to be an inaccurate ranking, a literature search reveals evidence supporting both Afatinib and Gefitinib as a potential treatment option for breast cancer [HSH14, LWW+12, BAR+05]. In prioritization for selective response in NSCLC over Kidney Cancer, Axitinib (with indication for renal cell carcinoma) was ranked above drugs with indication for NSCLC (alectinib, certitinib), but there is research suggesting that axitinib may also be useful for the treatment of NSCLC [SvPL+07]. In addition, a systematic review of ALK inhibitors (including ceritinib, alectinib; indications for NSCLC) has shown early promising results for use in renal cancer [IRS+22]. It may be that similar diseases will have missing or overlapping disease indications that can confound the evaluation of our drug prioritization.

Table 25: The results of disease-specific drug prioritizations evaluated on FDA approved drug indications. Bold font indicates the greatest AUROC value. The last two columns characterize the number of drugs with indications for target or background diseases, respectively. Note that we define the background cell lines as those from a specific disease rather than all non-target cell lines to ensure distinct cellular contexts and clear prioritization goals.

Target Dis. (# lines)	Background Dis. (# lines)	GSNN AUROC (FDR)	NN AUROC (FDR)	# target indications	# background indications
NSCLC (8)	AML (2)	**1.00** (0.17)	0.60 (0.50)	5	1
breast (9)	prostate (3)	1.00 (0.20)	1.00 (0.19)	4	1
breast (9)	AML (2)	1.00 (0.20)	1.00 (0.20)	4	1
NSCLC (8)	prostate (3)	**1.00 (0.17)**	0.60 (0.47)	5	1
breast (9)	NSCLC (8)	**0.80 (0.10)**	0.75 (0.15)	4	5
melanoma (7)	breast (9)	1.00 (0.02)	1.00 (0.02)	4	4
breast (9)	kidney (2)	1.00 (0.07)	1.00 (0.07)	4	2
melanoma (7)	NSCLC (8)	1.00 (0.01)	1.00 (0.01)	4	5
melanoma (7)	kidney (2)	1.00 (0.07)	1.00 (0.06)	4	2
NSCLC (8)	kidney (2)	0.70 (0.28)	**0.90 (0.10)**	5	2

6.5 Discussion

As suggested by the NFL theorem and further supported by the success of problem-specific models in vision (CNNs) and NLP (Transformers) tasks, including inductive biases into deep learning algorithms is a potential way to improve performance. In this work, we have shown that *graph structured neural networks* can use prior knowledge, encoded as a graph, to impose useful inductive biases. When applied to a perturbation biology task, the GSNN algorithm outperforms traditional neural networks, graph neural networks (GCN, GAT, and GIN) and GSNN models operating on randomized biological networks. We further investigate *local* performance and show that the GSNN model performs particularly well in a subset of drugs, genes, doses and diseases. Variations in local performance may suggest that the GSNN algorithm is better suited to model certain types of perturbation, biological processes, or cellular contexts. An in-depth understanding of these advantages or pitfalls could be used to refine the GSNN algorithm or the choice of prior knowledge to build even more robust perturbation models.

The proposed *GSNNExplainer* procedure enables a biologically relevant explanation of GSNN predictions, which can be used as a testable hypothesis to build knowledge of cell signaling processes or to develop trust in the GSNN algorithm. Furthermore, the predictions made by GSNNs are both traceable[16] and inspectable[17]. Future work should also investigate how model-agnostic interpretability methods, such as SHAP [LL17] , can be used to understand the behavior of specific function nodes and could be used to identify the specific role a molecular entity plays in signaling. These aspects are important improvements over traditional deep learning and are a necessary step toward the

[16] *Traceable* refers to the ability to trace a prediction to intermediary entities, states or input features.
[17] *Inspectable* refers to the ability to inspect the behavior of specific intermediate entities

development of trustworthy models of perturbation biology. Well-validated GSNN models have the potential for application in basic research, pre-clinical drug prioritization, and as a clinical decision aid.

We have shown that the GSNN method can be used to effectively predict disease-specific drug prioritizations, and this suggests that these methods could be used for a wide range of prioritization goals that are uniquely tailored to specific research goals. Furthermore, we have shown that the GSNN can accurately predict the cell viability of drug combinations when trained on single agents and outperforms equivalent models that use traditional NNs. These results suggest that the GSNN algorithm could also be used for prioritizing drug combinations.

6.5.1 Limitations

Appropriate inclusion of molecular entities and interactions

The introduction of prior knowledge in the GSNN can be both a boon and a curse: choosing the right set of molecular entities and interactions is liable to create interpretable high-performing models; but choosing the wrong set and we may overconstrain the model, resulting in poor models and inaccurate prediction logic. Identification of the correct subset of molecular entities required to model a system is a challenging task. In the methods we presented, we used Reactome pathways to select molecular entities that we believe are critical to certain processes and which we can tailor to the type of drug or disease we wish to model. While we believe this is justified as a nascent first approach, we do note that there are many ways this approach is lacking. Many knowledgebases, including Reactome, partition healthy signaling pathways separate from diseased signaling pathways with varying maturity in curation, and therefore the inclusion of relevant healthy pathways may miss molecular entities critical to disease. In future work, we would like to address this by expert curation or methodological development of entity and interaction selection algorithms. An attractive research direction is to perform hyper-optimization of the biological network during model training. For example, reinforcement learning could be used to select the best set of molecular entities and interactions that maximize the performance of a set of observations.

In this work, we used the *Omnipath* resource for the construction of our prior knowledge graph. A limitation that may arise from this is that the Omnipath resource focuses primarily on protein-protein, protein-DNA, and protein-RNA interactions and does not include some relevant molecular entities, such as ions. Ions, such as calcium or potassium, are well known to play an important role in many signaling pathways [GJS+21]. Further work may wish to explicitly model ion-like molecular entities; however, due to the role in many different interactions and pathways, encoding them as a distinct entity (i.e., function node) is likely to result in a high centrality[18] that may lead to inappropriate modeling of pathway cross-talk. For example, two unrelated pathways (e.g., different cellular compartments) that both involve calcium ions would have an unintended means of signaling. This challenge may suggest the need for additional learning mechanisms to model ion-like entities, or any entity that is independently involved in many pathways.

[18] centrality is a network science metric characterizing the importance or connectedness to other nodes

In this work, we assumed available a priori knowledge of the relevant molecular interactions, however, it should be noted that the GSNN method cannot learn new edges or denovo molecular entity interactions, and a consequence of this is that understudied pathways may lack appropriate prior knowledge to effectively model with the GSNN algorithm. Additionally, this characteristic makes the GSNN method critically dependent on user selection of accurate interactions and may be susceptible to over-constraints if important interactions are omitted. Although out of scope for this project, future work should consider how to infer new relationships between entities during model training. For example, a simple method would be the incorporation of a "global" node, to which all entities are connected. This addition would enable inference of new relationships between entities *through* the global node. Regularization might be applied to balance exploration of new interactions and exploitation of high-fidelity known interactions.

The drug-target perturbation premise

As described in section 1, we adopt a drug-target premise of chemical perturbation, which requires knowledge and presence of the proteins that a drug binds to; however, some drug mechanisms indirectly affect protein signaling. For instance, some molecular mechanisms change cellular conditions (which we term "condition-response" perturbations), which then activate protein sensors that measure the changing environment and initiate a response; examples of this include hypoxia, heat shock response, oxidative stress, osmotive stress, and DNA damage response. Hypoxia-inducible factors (HIFs) measure oxygen levels and activate transcriptional responses to adapt [WSB+20], PARP and DNA-PK proteins initiate the DNA damage response (DDR) signaling pathway [LA12]. DNA damaging drugs common in cancer therapies, such as Cisplatin, bind directly to DNA and cause DNA adducts [Ald19]. As such, Cisplatin does not have any primary protein targets and yet it is likely to lead to a DDR or apoptotic response. Future work should identify alternative ways to encode these condition-response perturbations. A simple approach may be to include intermittent *process* nodes, for example, include a "DNA damage" node and structure the network so that $Drug \rightarrow DNA\ damage \rightarrow Protein\ Signaling\ Cascade$; however, this approach will require significant expert knowledge and manual curation.

Limitations of the LINCS L1000 dataset

The LINCS L1000 dataset [SNC+17a] is a large high-throughput data set that characterizes thousands of drugs in hundreds of cell lines; however, it has known data quality issues [CL, QLLX20, LLP17, CHF+14, DRC+]. Future work in this field would benefit from the application of the GSNN algorithm to alternative datasets such as the Cancer Perturbed Proteomics Atlas (CPPA) [ZLC+20b]. Ensuring that the GSNN performs well on a range of datasets would provide additional evidence that this algorithm can be applied effectively to model cellular signaling.

Scalability and re-usability

The current formulation of the GSNN algorithm is an order of magnitude slower than traditional neural networks (see supp. 9.8), and therefore it can be computationally expensive to apply to large biological networks or datasets. There are several approaches that could improve the GSNN training and inference speeds. In Section 6.3.9 we describe how

the GSNNExplainer can be used to identify a subset of edges that are required to predict an observation. Using analogous methods to prune unimportant edges during the training process or at inference time could significantly reduce the compute requirements. Moreover, each drug perturbation can affect only downstream nodes, and many nodes are likely to be inaccessible to a given drug. This constraint suggests that we could obtain equivalent performance with drug-specific forward passes, which operate on a subset of the full biological graph and would markedly reduce the compute requirements. This concept of drug- or observation- specific forward passes also suggests the premise of *reusuability*. For example, a function node could be trained using one datatype, network, pathway, or drug set, and then *re-used* in a new datatype or pathway. This approach could enable efficient *localized* training of much larger cellular or microenvironment models.

Appropriate parameter sharing in GSNNs

The GSNN algorithm overcomes many limitations of GNNs by eschewing the parameter-sharing paradigm of GNN message aggregation. Although we have shown that our approach has notable advantages in modeling cell signaling, there are also limitations to our approach. Foremost, parameter sharing can significantly reduce the number of trainable weights and, consequently, may improve generalizability in some prediction tasks. This work is likely to benefit from future research that identifies aspects of cellular signaling where parameter sharing may be appropriate and useful, as well as reinforces aspects where it is not. For instance, the relationship between 'omic features and molecular state of the respective entities is likely to be similar across much of the interactome (e.g., deleterious mutations are likely to prevent the involvement of the respective proteins). Given this consideration, we believe that using parameter sharing to infer the state of molecular entities (i.e., cellular context) could be particularly useful in reducing the number of trainable weights and improving the performance of the GSNN algorithm.

Pathway isolation via protein localization

Cell signaling is typically characterized by biological pathways, which can function independently or interact with each other (pathway *cross-talk*). For a protein-protein interaction (PPI) to be active, the involved proteins must not only be structurally compatible but also be co-localized. Proteins from different pathways may be found in separate sub-cellular compartments, which can prevent interaction and pathway cross-talk. However, interaction between these isolated pathways can occur by translocation of a protein between compartments.

Our current method for constructing biological networks does not consider the specific pathway or subcellular location of proteins. This oversight may lead to inaccuracies in understanding inter-pathway interactions. For instance, if our model merges two pathways that share a protein but are otherwise isolated, it might incorrectly suggest inter-pathway interactions that do not actually occur.

The GSNN algorithm was designed to identify and respond to different input signals (i.e., *source awareness*), suggesting that it could learn specific pathway responses with the appropriate training data. However, given the current

constraints of perturbation biology[19], the GSNN algorithm may improve with additional inductive biases that promote signaling within specific pathways or subcellular areas. One way to do this is by using pathway or location annotations to clarify the role and position of each protein. In this approach, a protein could be represented by several function nodes, each modeling the protein's role in a different pathway or cellular location. Including edges between the same molecular entity in different compartments would allow for pathway cross-talk via translocation (e.g., NFkB_cytosol <-> NFkB_nucleus), and inter- vs intra- compartment signaling could be mediated by regularization (e.g., weight decay) on between-compartment edge weights.

Modeling time with Graph Structured Neural Ordinary Differential Equations

In its current state, the GSNN algorithm predicts a single-time point (24H). Future work may pursue methods to adapt the GSNN for temporal data (e.g., time-series or multiple time points), which better aligns with the true behavior of transcriptional response. These changes could lead to improved utility by allowing prediction of multiple time points or improve performance by the inclusion of temporal information. Neural Ordinary Differential Equations (ODE) have been proposed to solve ODEs using neural networks [CRBD19, PMP$^+$21], however, parameterizing the neural ODE is likely to fall victim to many of the limitations we discussed for traditional neural networks. The GSNN architecture is well suited to parameterize the neural ODE, which would allow the inclusion of prior knowledge constraints and could be applied to temporal modeling of transcriptional response. In practice, the GSNN layers would be replaced by sequential ODE solver steps to compute the change in state over time.

GSNNs for multi-cellular models

Another attractive future direction is the inclusion of multicellular or microenvironment models to study more complex behaviors of an organism. In simplicity, two cells could be encoded as a single GSNN model, with known cell-cell interactions connecting the two distinct cellular models. This approach could be used to study the impact of immune cells or the tumor microenvironment on drug response. Ignoring the current pragmatic constraints (e.g., memory and compute requirements), the GSNN algorithm may one day be used to model drug response at the tissue or multi-tissue scale; modeling the involvement of hundreds or thousands of cells.

Inclusion of experimental confounders

In-vitro cellular models of drug response require a variety of experimental conditions particular to the assay and cellular model, and these conditions are liable to introduce complexities and bias to the measured response. For example, many drug assays including the LINCS L1000 assay use dimethylsulfoxide (DMSO) to solubilize the various drugs that are tested. There is evidence that DMSO causes statistically significant expression changes in many pathways and cellular models [BSA$^+$21]. Furthermore, the Drugbank database lists three known protein targets for DMSO (Accession Number: DB01093), including the MYC transcription factor [WKG$^+$06]. To adjust for these experimental conditions, LINCS data processing uses zero-drug DMSO replicates as control when calculating gene perturbations. Even with

[19]Perturbation biology data is limited in volume and often measured via noisy assays.

controls, the various experimental conditions may introduce unexpected biases. Including experimental conditions as inputs to the model could better account for these factors and may help improve model performance and reliability. The presence of DMSO, for instance, could be encoded in the drug-target premise, and therefore each observation would be a combination of DMSO + drug. Including experimental conditions in this way may help delineate the expression changes induced by DMSO from the changes induced by the drug.

Encoding differential RNA splicing

In the current GSNN formulation, we chose to construct a simple biological network in which DNA and RNA are modeled as a joint entity. This approach is memory efficient and captures much of the prior knowledge; however, future work may benefit from the development of biological networks where DNA and RNA are modeled separately. Such an approach would allow for the inclusion of multiple RNA transcripts from the same DNA and would enable a more detailed characterization of the molecular landscape. A current limitation of this approach is that most molecular interaction databases are characterized at the gene level, which prevents a detailed understanding of how these interactions change with differential splicing.

6.5.2 Data and Code Availability

Availability and implementation: Our implementation of the GSNN method is available at `https://github.com/nathanieljevans/GSNN` (DOI: `https://doi.org/10.1101/2024.02.28.582164`). All data used in this work is publicly available.

7 Concluding remarks and discussion

In Chapter 1, we highlight the importance and limitations of AI in precision oncology, providing background and motivations for the work presented in this dissertation. In Chapters 2 and 3, we present methods for the detection of atypical or low-quality data with broad application in data cleaning tasks, critical for machine learning. In Chapters 4-6, we present deep learning algorithms designed to model perturbation biology and to incorporate alternative forms of prior knowledge into neural prediction logic. More specifically, in Chapter 4, we show that GNNs can accurately predict the majority of variance in synthetic perturbation biology datasets; however, we highlight that many of the assumptions of the proposed synthetic data and prediction task are not generalizable to experimental or real-world perturbation biology. In Chapter 5, we develop a novel GNN architecture and apply it to an experimental perturbation biology dataset (LINCS L1000). We show that our GNN algorithm can perform comparably with traditional neural networks but has a number of limitations that suggest prevent utility to precision oncology. In Chapter 6 we present a novel deep learning algorithm, called graph structured neural networks (GSNN), to distinguish it from the GNN methodology. The GSNN algorithm uses literature-curated biological networks to form a mechanistically constrained deep learning algorithm suitable for perturbation biology. In particular, the GSNN algorithm constrains the deep learning topology to emulate mechanistic biological networks. We show that the GSNN method outperforms traditional neural networks evaluated on perturbation biology prediction tasks and that performance is degraded when the biological network is permuted, which suggests performance gains are reliant on accurate mechanistic knowledge. We demonstrate that GSNN predictions can be explained in a biologically useful form, that predictions that are useful to downstream tasks such as functional annotation prediction, and present an approach to accurately predict disease-specific drug prioritizations.

Although Chapters 2, 4, and 5 investigate important questions relevant to precision oncology, we believe that our main contribution to the field is presented in Chapter 3 (DVGS) and Chapter 6 (GSNN). The DVGS method provides a particularly useful data cleaning tool that can quickly and accurately quantify the usefulness of the data toward a predictive task. Additionally, we showed that the DVGS method can capture general aspects of data quality in many scenarios, which suggests that this method can be used for general data cleaning or instance selection tasks. For example, large high-throughput datasets, which often suffer from data quality issues, are likely to benefit from high-quality data selection via the DVGS method. This application can remove inaccurate or noisy samples and create resource-efficient datasets (i.e., reduced storage requirements) while maintaining or even improving the usefulness to machine learning and data analytic tasks. A major advantage of the DVGS method is that it operates far more rapidly than alternative methods (such as Data Shapley or Data valuation with reinforcement learning) while performing equivalently or better. This means that it can be applied to much larger datasets or performed on common personal computers in convenient time spans. In many ways, we see this as a tool that could become as ubiquitous and expected as exploratory data analysis (EDA).

The GSNN method offers a unique and effective way to include prior knowledge into deep learning prediction logic. Compared to alternative methods, such as physics informed neural networks (PINNS) which require knowledge of partial derivative equations to specify prior knowledge, the GSNN method requires only topology characterizing latent variable interactions, which is readily available in many domains. Compared to methods like visible neural networks, which also use prior knowledge to constrain deep learning topology, our method can be used with mechanistic constraints and allows for cyclic graphs. This difference encourages mechanistic prediction logic rather than using abstract ontology forms that may not translate into mechanistic behavior. We envision that the GSNN method can be applied in many domains to improve prediction performance, aid interpretability, and generally increase the utility of predictive models. Obviously, the GSNN method is in a nascent phase of development, and there are still a number of limitations that need to be addressed. In particular, the curation of useful prior knowledge is critical for use with the GSNN method. Inaccurate data can lead to spurious or inaccurate predictions, while sparse prior knowledge may result in an over-constrained system. Future research in this area would benefit from careful evaluations of this trade-off and would benefit from the development of additional mechanisms to mitigate the challenges of limited or spurious prior knowledge.

Precision oncology, while already used in clinical practice, has untapped potential to revolutionize medicine and improve patient outcomes, however, as we have discussed in Chapter 1, there are many challenges that have yet to be overcome. In this work, we have sought to address several of these challenges by exploring how to improve cancer drug response modeling both in terms of improving predictive accuracy and ways to aid in the interpretation and utility of predictions. Developing methods like ours that incorporate prior knowledge and constrain algorithm logic based on well-validated and evidence-based latent variable interactions is an attractive avenue to improve model performance and garner trust from clinicians, patients, and the research community. Moreover, the custom architectures we have designed provide insight into the prediction logic and enable interpretation of model behavior. Again, this behavior is critical to build user trust, but also has utility in biological research as we can investigate not just predicted outcomes, but also the role of latent variables to guide mechanistic insight. The full realization of precision oncology will undoubtedly rely heavily on artificial intelligence, given the complexity of biological systems, and the work we have presented in this manuscript is an important, if nascent, step toward actionable precision oncology computational modeling.

8 References

[AAS20] Arohan Ajit, Koustav Acharya, and Abhishek Samanta. A review of convolutional neural networks. *2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE)*, pages 1–5, 2020.

[ABB+00] Michael Ashburner, Catherine A Ball, Judith A Blake, David Botstein, Heather Butler, J Michael Cherry, Allan P Davis, Kara Dolinski, Selina S Dwight, Janan T Eppig, et al. Gene ontology: tool for the unification of biology. *Nature genetics*, 25(1):25–29, 2000.

[Ada12] David J Adams. The valley of death in anticancer drug development: a reassessment. *Trends in pharmacological sciences*, 33(4):173–180, 2012.

[Aka23] Singh Akansha. Over-squashing in graph neural networks: A comprehensive survey. *arXiv preprint arXiv:2308.15568*, 2023.

[Ald19] Sara A Aldossary. Review on pharmacology of cisplatin: clinical use, toxicity and mechanism of resistance of cisplatin. *Biomedical and Pharmacology Journal*, 12(1):7–15, 2019.

[ALMK22] Julia Angwin, Jeff Larson, Surya Mattu, and Lauren Kirchner. Machine bias. In *Ethics of data and analytics*, pages 254–264. Auerbach Publications, 2022.

[ALSA+18] Elaine Angelino, Nicholas Larus-Stone, Daniel Alabi, Margo Seltzer, and Cynthia Rudin. Learning certifiably optimal rule lists for categorical data. *Journal of Machine Learning Research*, 18(234):1–78, 2018.

[Ama93] Shun-ichi Amari. Backpropagation and stochastic gradient descent method. *Neurocomputing*, 5(4-5):185–196, 1993.

[aws] Interpretability versus explainability - Model Explainability with AWS Artificial Intelligence and Machine Learning Solutions.

[Azu19] Francisco Azuaje. Artificial intelligence for precision oncology: beyond patient stratification. *NPJ precision oncology*, 3(1):6, 2019.

[Bal12] Pierre Baldi. Autoencoders, unsupervised learning, and deep architectures. In Isabelle Guyon, Gideon Dror, Vincent Lemaire, Graham Taylor, and Daniel Silver, editors, *Proceedings of ICML Workshop on Unsupervised and Transfer Learning*, volume 27 of *Proceedings of Machine Learning Research*, pages 37–49, Bellevue, Washington, USA, 02 Jul 2012. PMLR.

[BAR+05] José Baselga, Joan Albanell, Amparo Ruiz, Ana Lluch, Pere Gascón, Vicente Guillém, Sonia González, Silvia Sauleda, Irene Marimón, Josep M Tabernero, et al. Phase ii and tumor pharmacodynamic study of gefitinib in patients with advanced breast cancer. *J Clin Oncol*, 23(23):5323–5333, 2005.

[BB12] James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *Journal of machine learning research*, 13(2), 2012.

[BCB14] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[BCKM+17] Aurora S. Blucher, Gabrielle Choonoo, Molly F. Kulesz-Martin, Guanming Wu, and Shannon K. McWeeney. Evidence-based precision oncology with the cancer targetome. *Trends in pharmacological sciences*, 38 12:1085–1099, 2017.

[BE12] C Glenn Begley and Lee M Ellis. Drug development: Raise standards for preclinical cancer research. *Nature*, 483(7391):531—533, March 2012.

[BFI+22] Lukas Budach, Moritz Feuerpfeil, Nina Ihde, Andrea Nathansen, Nele Noack, Hendrik Patzlaff, Felix Naumann, and Hazar Harmouch. The effects of data quality on machine learning performance, 2022.

[BH95] Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the royal statistical society series b-methodological*, 57:289–300, 1995.

[BHB+18] Peter W. Battaglia, Jessica B. Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, Caglar Gulcehre, Francis Song, Andrew Ballard, Justin Gilmer, George Dahl, Ashish Vaswani, Kelsey Allen, Charles Nash, Victoria Langston, Chris Dyer, Nicolas Heess, Daan Wierstra, Pushmeet Kohli, Matt Botvinick, Oriol Vinyals, Yujia Li, and Razvan Pascanu. Relational inductive biases, deep learning, and graph networks, 2018.

[BI21] Lorenzo Brigato and Luca Iocchi. A close look at deep learning with small data. In *2020 25th international conference on pattern recognition (ICPR)*, pages 2490–2497. IEEE, 2021.

[BKH16] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization, 2016.

[BLK15] Olivier Bachem, Mario Lucic, and Andreas Krause. Coresets for nonparametric estimation - the case of dp-means. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 209–217, Lille, France, 07–09 Jul 2015. PMLR.

[BLS+22] Daniel Bottomly, Nicola Long, Anna Reister Schultz, Stephen E Kurtz, Cristina E Tognon, Kara Johnson, Melissa Abel, Anupriya Agarwal, Sammantha Avaylon, Erik Benton, et al. Integrative analysis of drug response and clinical outcome in acute myeloid leukemia. *Cancer Cell*, 40(8):850–864, 2022.

[BMSW19] Aurora S Blucher, Shannon K McWeeney, Lincoln Stein, and Guanming Wu. Visualization of drug

target interactions in the contexts of pathways and networks with reactomefiviz. *F1000Research*, 8, 2019.

[BPPG19] Mariana Brandão, Noam Pondé, and Martine Piccart-Gebhart. Mammaprint™: a comprehensive review. *Future oncology*, 15(2):207–224, 2019.

[BSA+21] Elisa Baldelli, Mahalakshmi Subramanian, Abduljalil M Alsubaie, Guy Oldaker, Maria Emelianenko, Emna El Gazzah, Sara Baglivo, Kimberley A Hodge, Fortunato Bianconi, Vienna Ludovini, et al. Heterogeneous off-target effects of ultra-low dose dimethyl sulfoxide (dmso) on targetable signaling events in lung cancer in vitro models. *International Journal of Molecular Sciences*, 22(6):2819, 2021.

[BSR+19] Kaustav Bera, Kurt A Schalper, David L Rimm, Vamsidhar Velcheti, and Anant Madabhushi. Artificial intelligence in digital pathology—new tools for diagnosis and precision oncology. *Nature reviews Clinical oncology*, 16(11):703–715, 2019.

[But08] D Buter. Translational research: crossing the valley of death. *Nature*, 453:840–2, 2008.

[Búz12] Krisztián Búza. Feedback prediction for blogs. In *Annual Conference of the Gesellschaft für Klassifikation*, 2012.

[BV10] Tanya N Beran and Claudio Violato. Structural equation modeling in medical research: a primer. *BMC research notes*, 3(1):1–10, 2010.

[BY01] Yoav Benjamini and Daniel Yekutieli. The control of the false discovery rate in multiple testing under dependency. *Annals of Statistics*, 29:1165–1188, 2001.

[BZV+19] Irwan Bello, Barret Zoph, Ashish Vaswani, Jonathon Shlens, and Quoc V Le. Attention augmented convolutional networks. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3286–3295, 2019.

[cana] Budget and Appropriations — cancer.gov. `https://www.cancer.gov/about-nci/budget#:~:text=NCI%20is%20operating%20under%20a,of%20the%20Cancer%20Moonshot%E2%84%A0.` [Accessed 22-02-2024].

[canb] Functional genomic landscape of acute myeloid leukaemia | NCI Genomic Data Commons — gdc.cancer.gov. `https://gdc.cancer.gov/about-data/publications/BEATAML1-0-COHORT-2018.` [Accessed 22-02-2024].

[CB01a] Edward J Calabrese and Linda A Baldwin. The frequency of u-shaped dose responses in the toxicological literature. *Toxicological Sciences*, 62(2):330–338, 2001.

[CB01b] Edward J Calabrese and Linda A Baldwin. Hormesis: U-shaped dose responses and their centrality in toxicology. *Trends in pharmacological sciences*, 22(6):285–291, 2001.

[CBL+17] Steven M. Corsello, Joshua A. Bittker, Zihan Liu, Joshua Gould, Patrick McCarren, Jodi E. Hirschman, Stephen E. Johnston, Anita Vrcic, Bang Wong, Mariya Khan, Jacob K. Asiedu, Rajiv Narayan, C. C. Mader, Aravind Subramanian, and Todd R. Golub. The drug repurposing hub: a next-generation drug library and information resource. *Nature Medicine*, 23:405–408, 2017.

[CBS+] Jaime H Cheah, Haley S Bridger, Alykhan F Shamji, Stuart L Schreiber, and Paul A Clemons. Cancer therapeutics response portal: A ctd^2 network resource for mining candidate cancer dependencies.

[cdc]

[CGD+10] Ethan G Cerami, Benjamin E Gross, Emek Demir, Igor Rodchenkov, Özgün Babur, Nadia Anwar, Nikolaus Schultz, Gary D Bader, and Chris Sander. Pathway commons, a web resource for biological pathway data. *Nucleic acids research*, 39(suppl_1):D685–D690, 2010.

[CGR+21] Cristian Coarfa, Sandra L Grimm, Kimal Rajapakshe, Dimuthu Perera, Hsin-Yi Lu, Xuan Wang, Kurt R Christensen, Qianxing Mo, Dean P Edwards, and Shixia Huang. Reverse-phase protein array: Technology, application, data processing, and integration. *Journal of biomolecular techniques: JBT*, 32(1):15, 2021.

[CHF+14] Neil R. Clark, Kevin S. Hu, Axel S. Feldmann, Yan Kou, Edward Y. Chen, Qiaonan Duan, and Avi Ma'ayan. The characteristic direction: a geometrical approach to identify differentially expressed genes. *BMC Bioinformatics*, 15, 2014.

[CHWY14] Hong Chen, David Hailey, Ning Wang, and Ping Yu. A review of data quality assessment methods for public health information systems. *International journal of environmental research and public health*, 11(5):5170–5207, 2014.

[CL] L Cheng and L Li. Systematic quality control analysis of LINCS data. 5(11):588–598.

[CLL+19] Deli Chen, Yankai Lin, Wei Li, Peng Li, Jie Zhou, and Xu Sun. Measuring and relieving the over-smoothing problem for graph neural networks from the topological view, 2019.

[CNS+20] Steven M Corsello, Rohith T Nagari, Ryan D Spangler, Jordan Rossen, Mustafa Kocak, Jordan G Bryan, Ranad Humeidi, David Peck, Xiaoyun Wu, Andrew A Tang, et al. Discovering the anticancer potential of non-oncology drugs by systematic viability profiling. *Nature cancer*, 1(2):235–248, 2020.

[Coo] R. Dennis Cook. Detection of influential observation in linear regression. 19(1):15–18. Publisher: [Taylor & Francis, Ltd., American Statistical Association, American Society for Quality].

[CPY+18] Yoosup Chang, Hyejin Park, Hyun-Jin Yang, Seungju Lee, Kwee-Yum Lee, Tae Soon Kim, Jongsun Jung, and Jae-Min Shin. Cancer drug response profile scan (cdrscan): a deep learning model that predicts drug effectiveness from cancer genomic signature. *Scientific reports*, 8(1):8857, 2018.

[CRBD18] Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.

[CRBD19] Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. Neural ordinary differential equations, 2019.

[CUH15] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). *arXiv: Learning*, 2015.

[CWS12] Yutian Chen, Max Welling, and Alex Smola. Super-samples from kernel herding, 2012.

[CZ15] Li Cai and Yangyong Zhu. The challenges of data quality and data quality assessment in the big data era. *Data Sci. J.*, 14:2, 2015.

[DAS+22] Eugene F Douglass, Robert J Allaway, Bence Szalai, Wenyu Wang, Tingzhong Tian, Adrià Fernández-Torras, Ron Realubit, Charles Karan, Shuyu Zheng, Alberto Pessia, et al. A community challenge for a pancancer drug mechanism of action inference from perturbational profile data. *Cell Reports Medicine*, 3(1), 2022.

[dee]

[DFHM20] Zodwa Dlamini, Flavia Zita Francies, Rodney Hull, and Rahaba Marima. Artificial intelligence (ai) and big data in cancer and precision oncology. *Computational and structural biotechnology journal*, 18:2300–2311, 2020.

[DG] D. Dua and C. Graff. UCI machine learning repository.

[DRC+] Qiaonan Duan, St Patrick Reid, Neil R. Clark, Zichen Wang, Nicolas F. Fernandez, Andrew D. Rouillard, Ben Readhead, Sarah R. Tritsch, Rachel Hodos, Marc Hafner, Mario Niepel, Peter K. Sorger, Joel T. Dudley, Sina Bavari, Rekha G. Panchal, and Avi Ma'ayan. L1000cds2: LINCS l1000 characteristic direction signatures search engine. 2(1):1–12. Number: 1 Publisher: Nature Publishing Group.

[DRC+16] Qiaonan Duan, St. Patrick Reid, Neil R. Clark, Zichen Wang, Nicolas F. Fernandez, Andrew D. Rouillard, Ben Readhead, Sarah R. Tritsch, Rachel Hodos, Marc Hafner, Mario Niepel, Peter K. Sorger, Joel T. Dudley, Sina Bavari, Rekha Panchal, and Avi Ma'ayan. L1000cds2: Lincs l1000 characteristic direction signatures search engine. *NPJ Systems Biology and Applications*, 2, 2016.

[DTR+01] Brian J Druker, Moshe Talpaz, Debra J Resta, Bin Peng, Elisabeth Buchdunger, John M Ford, Nicholas B Lydon, Hagop Kantarjian, Renaud Capdeville, Sayuri Ohno-Jones, et al. Efficacy and safety of a specific inhibitor of the bcr-abl tyrosine kinase in chronic myeloid leukemia. *New England Journal of Medicine*, 344(14):1031–1037, 2001.

[dVJ23] Karin E de Visser and Johanna A Joyce. The evolving tumor microenvironment: From cancer initiation to metastatic outgrowth. *Cancer Cell*, 41(3):374–403, 2023.

[DVK17] Finale Doshi-Velez and Been Kim. Towards a rigorous science of interpretable machine learning, 2017.

[DYM18] Lifang Deng, Miao Yang, and Katerina M Marcoulides. Structural equation modeling with many variables: A systematic review of issues and developments. *Frontiers in psychology*, 9:580, 2018.

[ERB+20] K Esfahani, L Roudaia, NA Buhlaiga, SV Del Rincon, N Papneja, and WH Miller. A review of cancer immunotherapy: from the past, to the present, to the future. *Current Oncology*, 27(s2):87–97, 2020.

[FD23] Hadiya Faheem and Sanjib Dutta. Artificial intelligence failure at ibm'watson for oncology'. *IUP Journal of Knowledge Management*, 21(3):47–75, 2023.

[FFK11] Dan Feldman, Matthew Faulkner, and Andreas Krause. Scalable training of mixture models via coresets. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc., 2011.

[FKW+18] Fabian Fröhlich, Thomas Kessler, Daniel Weindl, Alexey Shadrin, Leonard Schmiester, Hendrik Hache, Artur Muradyan, Moritz Schütte, Ji-Hyun Lim, Matthias Heinig, et al. Efficient parameter estimation enables the prediction of drug response using a mechanistic pan-cancer pathway model. *Cell systems*, 7(6):567–579, 2018.

[FL19] Matthias Fey and Jan Eric Lenssen. Fast graph representation learning with pytorch geometric, 2019.

[FPP07] David Freedman, Robert Pisani, and Roger Purves. Statistics (international student edition). *Pisani, R. Purves, 4th edn. WW Norton & Company, New York*, 2007.

[Fri01] Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.

[GAITSR18] Luz Garcia-Alonso, Mahmoud M. Ibrahim, Denes Turei, and Julio Sáez-Rodríguez. Benchmark and integration of resources for the estimation of human transcription factor activities. *Genome Research*, 29:1363 – 1375, 2018.

[GBC16] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[GGE+21] Kathleen Gallo, Andrean Goede, Andreas Eckert, Barbara Moahamed, Robert Preissner, and Björn-Oliver Gohlke. Promiscuous 2.0: a resource for drug-repositioning. *Nucleic Acids Research*, 49(D1):D1373–D1380, 2021.

[GHJV+19] Mahmoud Ghandi, Franklin W Huang, Judit Jané-Valbuena, Gregory V Kryukov, Christopher C Lo, E Robert McDonald III, Jordi Barretina, Ellen T Gelfand, Craig M Bielski, Haoxin Li, et al. Next-generation characterization of the cancer cell line encyclopedia. *Nature*, 569(7757):503–508, 2019.

[GHVLJ21] Sreenivasulu Gunti, Austin TK Hoke, Kenny P Vu, and Nyall R London Jr. Organoid and spheroid tumor models: Techniques and applications. *Cancers*, 13(4):874, 2021.

[GJS+21] Marc Gillespie, Bijay Jassal, Ralf Stephan, Marija Milacic, Karen Rothfels, Andrea Senff-Ribeiro, Johannes Griss, Cristoffer Sevilla, Lisa Matthews, Chuqiao Gong, Chuan Deng, Thawfeek Varusai, Eliot Ragueneau, Yusra Haider, Bruce May, Veronica Shamovsky, Joel Weiser, Timothy Brunson, Nasim Sanati, Liam Beckman, Xiang Shao, Antonio Fabregat, Konstantinos Sidiropoulos, Julieth Murillo, Guilherme Viteri, Justin Cook, Solomon Shorser, Gary Bader, Emek Demir, Chris Sander, Robin Haw, Guanming Wu, Lincoln Stein, Henning Hermjakob, and Peter D'Eustachio. The reactome pathway knowledgebase 2022. *Nucleic Acids Research*, 50(D1):D687–D692, 11 2021.

[GL16] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864, 2016.

[Glo]

[GNS+20] Marzyeh Ghassemi, Tristan Naumann, Peter Schulam, Andrew L Beam, Irene Y Chen, and Rajesh Ranganath. A review of challenges and opportunities in machine learning for health. *AMIA Summits on Translational Science Proceedings*, 2020:191, 2020.

[GPSW17] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks, 2017.

[GR92] Andrew Gelman and Donald B Rubin. Inference from iterative simulation using multiple sequences. *Statistical science*, 7(4):457–472, 1992.

[GSR+17] Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural message passing for quantum chemistry, 2017.

[GVG13] Jean-Pierre Gillet, Sudhir Varma, and Michael M Gottesman. The clinical relevance of cancer cell lines. *Journal of the National Cancer Institute*, 105(7):452–458, 2013.

[GZ] Amirata Ghorbani and James Zou. Data shapley: Equitable valuation of data for machine learning.

[Hay13] Winston Haynes. Bonferroni correction. *Encyclopedia of systems biology*, pages 154–154, 2013.

[HB19] Éléa Héberlé and Anaïs Flore Bardet. Sensitivity of transcription factors to dna methylation. *Essays in biochemistry*, 63(6):727–741, 2019.

[HCC+17] Susan L Holbeck, Richard Camalier, James A Crowell, Jeevan Prasaad Govindharajulu, Melinda Hollingshead, Lawrence W Anderson, Eric Polley, Larry Rubinstein, Apurva Srivastava, Deborah Wilsker, et al. The national cancer institute almanac: a comprehensive screening resource for the detection of anticancer drug pairs with enhanced therapeutic activity. *Cancer research*, 77(13):3564–3576, 2017.

[HCM+18] Alexander Sebastian Hauser, Sreenivas Chavali, Ikuo Masuho, Leonie Johanna Jahn, Kirill A. Martemyanov, David E. Gloriam, and M. Madan Babu. Pharmacogenomics of gpcr drug targets. *Cell*, 172:41 – 54.e19, 2018.

[HGGW+19] Jada G Hamilton, Margaux Genoff Garzon, Joy S Westerman, Elyse Shuk, Jennifer L Hay, Chasity Walters, Elena Elkin, Corinna Bertelsen, Jessica Cho, Bobby Daly, et al. "a tool, not a crutch": patient perspectives about ibm watson for oncology trained by memorial sloan kettering. *Journal of Oncology Practice*, 15(4):e277–e288, 2019.

[HK70] Arthur E Hoerl and Robert W Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.

[HLK+18] Jay Heo, Hae Beom Lee, Saehoon Kim, Juho Lee, Kwang Joon Kim, Eunho Yang, and Sung Ju Hwang. Uncertainty-aware attention for reliable interpretation and prediction. *Advances in neural information processing systems*, 31, 2018.

[HM82] James A Hanley and Barbara J McNeil. The meaning and use of the area under a receiver operating characteristic (roc) curve. *Radiology*, 143(1):29–36, 1982.

[HMWG18] Dan Hendrycks, Mantas Mazeika, Duncan Wilson, and Kevin Gimpel. Using trusted data to train deep networks on labels corrupted by severe noise. *ArXiv*, abs/1802.05300, 2018.

[HNR02] Reinhart Heinrich, Benjamin G Neel, and Tom A Rapoport. Mathematical models of protein kinase signal transduction. *Molecular cell*, 9(5):957–970, 2002.

[HSH14] Sara A Hurvitz, Rebecca Shatsky, and Nadia Harbeck. Afatinib in the treatment of breast cancer. *Expert opinion on investigational drugs*, 23(7):1039–1047, 2014.

[HSW89] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989.

[HSZH20] Jan G Hengstler, Anna-Karin Sjögren, Daniele Zink, and Jorrit J Hornberg. In vitro prediction of organ toxicity: the challenges of scaling and secondary mechanisms of toxicity, 2020.

[HVD15] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015.

[HVSLJ13] Caitriona Holohan, Sandra Van Schaeybroeck, Daniel B Longley, and Patrick G Johnston. Cancer drug resistance: an evolving paradigm. *Nature Reviews Cancer*, 13(10):714–726, 2013.

[HW11] Douglas Hanahan and Robert A Weinberg. Hallmarks of cancer: the next generation. *cell*, 144(5):646–674, 2011.

[HZRS15a] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[HZRS15b] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification, 2015.

[IRS+22] Giovanni Maria Iannantuono, Silvia Riondino, Stefano Sganga, Mario Roselli, and Francesco Torino. Activity of alk inhibitors in renal cancer with alk alterations: A systematic review. *International Journal of Molecular Sciences*, 23(7):3995, 2022.

[IS15] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, 2015.

[Jan03] Ritsert C Jansen. Studying complex biological systems using multifactorial perturbation. *Nature Reviews Genetics*, 4(2):145–151, 2003.

[JGP17] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax, 2017.

[JZL+17] Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. 2017.

[KBG16] Steven B Kim, Scott M Bartell, and Daniel L Gillen. Inference for the existence of hormetic dose–response relationships in toxicology studies. *Biostatistics*, 17(3):523–536, 2016.

[KG00] Minoru Kanehisa and Susumu Goto. Kegg: kyoto encyclopedia of genes and genomes. *Nucleic acids research*, 28(1):27–30, 2000.

[KM15] Kimberly R Kukurba and Stephen B Montgomery. Rna sequencing and analysis. *Cold Spring Harbor Protocols*, 2015(11):pdb–top084970, 2015.

[KPF+20] Brent M Kuenzi, Jisoo Park, Samson H Fong, Kyle S Sanchez, John Lee, Jason F Kreisberg, Jianzhu Ma, and Trey Ideker. Predicting drug response and synergy using a deep learning model of human cancer cells. *Cancer cell*, 38(5):672–684, 2020.

[KPPH20] Kelly Karl, Michael D Paul, Elena B Pasquale, and Kalina Hristova. Ligand bias in receptor tyrosine kinase signaling. *Journal of Biological Chemistry*, 295(52):18494–18507, 2020.

[KPTA20] Gukyeong Kwon, Mohit Prabhushankar, Dogancan Temel, and Ghassan AlRegib. Backpropagated gradient representations for anomaly detection, 2020.

[Kri09] Alex Krizhevsky. Learning multiple layers of features from tiny images. pages 32–33, 2009.

[Kum17] Siddharth Krishna Kumar. On weight initialization in deep neural networks, 2017.

[KvMC+07] Michael Kuhn, Christian von Mering, Monica Campillos, Lars Juhl Jensen, and Peer Bork. Stitch: interaction networks of chemicals and proteins. *Nucleic Acids Research*, 36:D684 – D688, 2007.

[KW16] Thomas Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *ArXiv*, abs/1609.02907, 2016.

[KW22] Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2022.

[LA12] Christopher J Lord and Alan Ashworth. The dna damage response and cancer therapy. *Nature*, 481(7381):287–294, 2012.

[LAM19] Pavel Loskot, Komlan Atitey, and Lyudmila Mihaylova. Comprehensive review of models and methods for inferences in bio-chemical reaction networks. *Frontiers in genetics*, page 549, 2019.

[LCP+06] Justin Lamb, Emily D Crawford, David Peck, Joshua W Modell, Irene C Blat, Matthew J Wrobel, Jim Lerner, Jean-Philippe Brunet, Aravind Subramanian, Kenneth N Ross, et al. The connectivity map: using gene-expression signatures to connect small molecules, genes, and disease. *science*, 313(5795):1929–1935, 2006.

[LCQ21] Jiaxing Lu, Ming Chen, and Yufang Qin. Drug-induced cell viability prediction from lincs-l1000 through wrfen-xgboost algorithm. *BMC bioinformatics*, 22:1–18, 2021.

[LG19] Diya Li and Jianxi Gao. Towards perturbation prediction of biological networks using deep learning. *Scientific reports*, 9(1):11941, 2019.

[LHJZ20] Qiao Liu, Zhiqiang Hu, Rui Jiang, and Mu Zhou. Deepcdr: a hybrid graph convolutional network for predicting cancer drug response. *Bioinformatics*, 36(Supplement_2):i911–i918, 2020.

[LJC+18] Samuel A. Lambert, Arttu Jolma, Laura F. Campitelli, Pratyush Kumar Das, Yimeng Yin, Mihai Albu, Xiaoting Chen, Jussi Taipale, Timothy R. Hughes, and Matthew T. Weirauch. The human transcription factors. *Cell*, 172:650–665, 2018.

[LKSDD+23] Mohammad Lotfollahi, Anna Klimovskaia Susmelj, Carlo De Donno, Leon Hetzel, Yuge Ji, Ignacio L Ibarra, Sanjay R Srivatsan, Mohsen Naghipourfar, Riza M Daza, Beth Martin, et al. Predicting cellular

responses to complex perturbations in high-throughput screens. *Molecular Systems Biology*, page e11517, 2023.

[LL17] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017.

[LLM+24] Jun Li, Wei Liu, Kamalika Mojumdar, Hong Kim, Zhicheng Zhou, Zhenlin Ju, Shwetha V Kumar, Patrick Kwok-Shing Ng, Han Chen, Michael A Davies, et al. A protein expression atlas on tissue samples and cell lines from cancer patients provides insights into tumor heterogeneity and dependencies. *Nature Cancer*, pages 1–17, 2024.

[LLP17] Zhaoyang Li, Jin Li, and YU Peng. l1kdeconv: an r package for peak calling analysis with lincs l1000 data. *BMC Bioinformatics*, 18, 2017.

[Loh11] Wei-Yin Loh. Classification and regression trees. *Wiley interdisciplinary reviews: data mining and knowledge discovery*, 1(1):14–23, 2011.

[LPB17] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles, 2017.

[LQC+22] Pan Lu, Liang Qiu, Kai-Wei Chang, Ying Nian Wu, Song-Chun Zhu, Tanmay Rajpurohit, Peter Clark, and Ashwin Kalyan. Dynamic prompt learning via policy gradient for semi-structured mathematical reasoning, 2022.

[LUK+21] Yuanyuan Li, David M Umbach, Juno M Krahn, Igor Shats, Xiaoling Li, and Leping Li. Predicting tumor response to drugs based on gene-expression biomarkers of sensitivity learned from cancer cell lines. *BMC genomics*, 22:1–18, 2021.

[LWW+12] Nancy U Lin, Eric P Winer, Duncan Wheatley, Lisa A Carey, Stephen Houston, David Mendelson, Pamela Munster, Laurie Frakes, Steve Kelly, Agustin A Garcia, et al. A phase ii study of afatinib (bibw 2992), an irreversible erbb family blocker, in patients with her2-positive metastatic breast cancer progressing after trastuzumab. *Breast cancer research and treatment*, 133:1057–1065, 2012.

[LWZ+19] Min Li, Yake Wang, Ruiqing Zheng, Xinghua Shi, Yaohang Li, Fang-Xiang Wu, and Jianxin Wang. Deepdsc: a deep learning method to predict drug sensitivity of cancer cell lines. *IEEE/ACM transactions on computational biology and bioinformatics*, 18(2):575–582, 2019.

[LXTG20] Guohao Li, Chenxin Xiong, Ali Thabet, and Bernard Ghanem. Deepergcn: All you need to train deeper gcns, 2020.

[LZZ+20] Hui Liu, Wenhao Zhang, Bo Zou, Jinxian Wang, Yuanyuan Deng, and Lei Deng. Drugcombdb: a comprehensive database of drug combinations toward the discovery of combinatorial therapy. *Nucleic*

acids research, 48(D1):D871–D881, 2020.

[MBL19] Baharan Mirzasoleiman, Jeff Bilmes, and Jure Leskovec. Coresets for data-efficient training of machine learning models. 2019.

[McD18] Ultan McDermott. Cancer cell lines as patient avatars for drug response prediction. *Nature genetics*, 50(10):1350–1351, 2018.

[Mes17] Bertalan Mesko. The role of artificial intelligence in precision medicine, 2017.

[MFL+21] Jianzhu Ma, Samson H Fong, Yunan Luo, Christopher J Bakkenist, John Paul Shen, Soufiane Mourragui, Lodewyk FA Wessels, Marc Hafner, Roded Sharan, Jian Peng, et al. Few-shot learning creates predictive models of drug response that translate from high-throughput screens to individual patients. *Nature Cancer*, 2(2):233–244, 2021.

[MHM18] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

[MK17] Terri P McVeigh and Michael J Kerin. Clinical use of the oncotype dx genomic test to guide treatment decisions for patients with invasive breast cancer. *Breast Cancer: Targets and Therapy*, pages 393–400, 2017.

[MKW+13] Evan J Molinelli, Anil Korkut, Weiqing Wang, Martin L Miller, Nicholas P Gauthier, Xiaohong Jing, Poorvi Kaushik, Qin He, Gordon Mills, David B Solit, et al. Perturbation biology: inferring signaling networks in cellular systems. *PLoS computational biology*, 9(12):e1003290, 2013.

[MLKP19] Leigh Marcus, Steven J Lemery, Patricia Keegan, and Richard Pazdur. Fda approval summary: pembrolizumab for the treatment of microsatellite instability-high solid tumors. *Clinical Cancer Research*, 25(13):3753–3758, 2019.

[MLST23] Yao Ma, Xiaorui Liu, Neil Shah, and Jiliang Tang. Is homophily a necessity for graph neural networks?, 2023.

[MLV+21] Soufiane MC Mourragui, Marco Loog, Daniel J Vis, Kat Moore, Anna G Manjon, Mark A van de Wiel, Marcel JT Reinders, and Lodewyk FA Wessels. Predicting patient response with models trained on cell lines and patient-derived xenografts by nonlinear transfer learning. *Proceedings of the National Academy of Sciences*, 118(49):e2106682118, 2021.

[MM22] Alhassan Mumuni and Fuseini Mumuni. Data augmentation: A comprehensive survey of modern approaches. *Array*, page 100258, 2022.

[MMT17] Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables, 2017.

[Mol20] Christoph Molnar. *Interpretable machine learning*. Lulu. com, 2020.

[MP43] Warren S McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, 1943.

[MVL+17] John G Moffat, Fabien Vincent, Jonathan A Lee, Jörg Eder, and Marco Prunotto. Opportunities and challenges in phenotypic drug discovery: an industry perspective. *Nature reviews Drug discovery*, 16(8):531–543, 2017.

[MW47] Henry B Mann and Donald R Whitney. On a test of whether one of two random variables is stochastically larger than the other. *The annals of mathematical statistics*, pages 50–60, 1947.

[MWLN22] Dang Minh, H Xiang Wang, Y Fen Li, and Tan N Nguyen. Explainable artificial intelligence: a comprehensive review. *Artificial Intelligence Review*, pages 1–66, 2022.

[MYF+18] Jianzhu Ma, Michael Ku Yu, Samson Fong, Keiichiro Ono, Eric Sage, Barry Demchak, Roded Sharan, and Trey Ideker. Using deep learning to model the hierarchical structure and function of a cell. *Nature methods*, 15(4):290–298, 2018.

[NHM+19a] Mario Niepel, Marc Hafner, Caitlin E Mills, Kartik Subramanian, Elizabeth H Williams, Mirra Chung, Benjamin Gaudio, Anne Marie Barrette, Alan D Stern, Bin Hu, et al. A multi-center study on the reproducibility of drug-response assays in mammalian cell lines. *Cell systems*, 9(1):35–48, 2019.

[NHM+19b] Mario Niepel, Marc Hafner, Caitlin E. Mills, Kartik Subramanian, Elizabeth H. Williams, Mirra Chung, Benjamin Gaudio, Anne Marie Barrette, Alan D. Stern, Bin Hu, James E. Korkola, Caroline E. Shamu, Gomathi Jayaraman, Evren U. Azeloglu, Ravi Iyengar, Eric A. Sobie, Gordon B. Mills, Tiera Liby, Jacob D. Jaffe, Maria Alimova, Desiree Davison, Xiaodong Lu, Todd R. Golub, Aravind Subramanian, Brandon Shelley, Clive N. Svendsen, Avi Ma'ayan, Mario Medvedovic, Heidi S. Feiler, Rebecca Smith, Kaylyn Devlin, Joe W. Gray, Marc R. Birtwistle, Laura M. Heiser, and Peter K. Sorger. A multi-center study on the reproducibility of drug-response assays in mammalian cell lines. *Cell Systems*, 9(1):35–48.e5, 2019.

[noaa] NCI-60 Screening Methodology | NCI-60 Human Tumor Cell Lines Screen | Discovery & Development Services | Developmental Therapeutics Program (DTP).

[noab] Trustworthy and responsible AI. Last Modified: 2023-05-04T13:20-04:00.

[ODG+14] Shawn C Owen, Allison K Doak, Ahil N Ganesh, Lyudmila Nedyalkova, Christopher K McLaughlin, Brian K Shoichet, and Molly S Shoichet. Colloidal drug formulations can explain "bell-shaped" concentration–response curves. *ACS chemical biology*, 9(3):777–784, 2014.

[Par09] Peter J Park. Chip–seq: advantages and challenges of a maturing technology. *Nature reviews genetics*, 10(10):669–680, 2009.

[Pea01] Karl Pearson. 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, 1901.

[PFB16] Vinay Prasad, Tito Fojo, and Michael Brada. Precision oncology: origins, optimism, and potential. *The Lancet Oncology*, 17(2):e81–e86, 2016.

[PGM+19] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library, 2019.

[PGS+23] Stefan Peidli, Tessa D. Green, Ciyue Shen, Torsten Gross, Joseph Min, Samuele Garda, Bo Yuan, Linus J. Schumacher, Jake P. Taylor-King, Debora S. Marks, Augustin Luna, Nils Blüthgen, and Chris Sander. scperturb: Harmonized single-cell perturbation data. *bioRxiv*, 2023.

[PHF+20] P Jonathon Phillips, Carina A Hahn, Peter C Fontana, David A Broniatowski, and Mark A Przybocki. Four principles of explainable artificial intelligence. *Gaithersburg, Maryland*, 18, 2020.

[PHL+19] Channing J Paller, Erich P Huang, Thomas Luechtefeld, Holly A Massett, Christopher C Williams, Jinxiu Zhao, Amy E Gravell, Tami Tamashiro, Steven A Reeves, Gary L Rosner, et al. Factors affecting combination trial success (facts): investigator survey results on early-phase combination trials. *Frontiers in Medicine*, 6:122, 2019.

[PMP+19] Michael Poli, Stefano Massaroli, Junyoung Park, Atsushi Yamashita, Hajime Asama, and Jinkyoo Park. Graph neural ordinary differential equations. *arXiv preprint arXiv:1911.07532*, 2019.

[PMP+21] Michael Poli, Stefano Massaroli, Junyoung Park, Atsushi Yamashita, Hajime Asama, and Jinkyoo Park. Graph neural ordinary differential equations, 2021.

[PQZ+] Thai-Hoang Pham, Yue Qiu, Jucheng Zeng, Lei Xie, and Ping Zhang. A deep learning framework for high-throughput mechanism-driven phenotype compound screening and its application to COVID-19 drug repurposing. 3(3):247–257. Number: 3 Publisher: Nature Publishing Group.

[PSA11] Florian Prinz, Thomas Schlange, and Khusru Asadullah. Believe it or not: how much can we rely on published data on potential drug targets? *Nature Reviews Drug Discovery*, 10:712–712, 2011.

[PSCH21] Guansong Pang, Chunhua Shen, Longbing Cao, and Anton Van Den Hengel. Deep learning for anomaly detection. *ACM Computing Surveys*, 54(2):1–38, mar 2021.

[QCSSL09] Joaquin Quionero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D. Lawrence. *Dataset Shift in Machine Learning*. MIT, 2009.

[QLLX20] Yue Qiu, Tianhuan Lu, Hansaim Lim, and Lei Xie. A Bayesian approach to accurate and robust signature detection on LINCS L1000 data. *Bioinformatics*, 36(9):2787–2795, 01 2020.

[Rai20] Arun Rai. Explainable ai: From black box to glass box. *Journal of the Academy of Marketing Science*, 48:137–141, 2020.

[RC05] Christian P. Robert and George Casella. Monte carlo statistical methods. *Technometrics*, 47:243 – 243, 2005.

[RHS+19] Ladislav Rampášek, Daniel Hidru, Petr Smirnov, Benjamin Haibe-Kains, and Anna Goldenberg. Dr. vae: improving drug response prediction via modeling of drug perturbation effects. *Bioinformatics*, 35(19):3743–3751, 2019.

[Rig17] Steven J Rigatti. Random forest. *Journal of Insurance Medicine*, 47(1):31–39, 2017.

[RM87] David E. Rumelhart and James L. McClelland. *Learning Internal Representations by Error Propagation*, pages 318–362. 1987.

[Ros20] Robert Roskoski. Properties of fda-approved small molecule protein kinase inhibitors: A 2020 update. *Pharmacological Research*, 152:104609, 2020.

[RPK19] Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics*, 378:686–707, 2019.

[RSG16] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should i trust you?": Explaining the predictions of any classifier, 2016.

[Rud19] Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature machine intelligence*, 1(5):206–215, 2019.

[RZYU18] Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. Learning to reweight examples for robust deep learning, 2018.

[S+53] Lloyd S Shapley et al. A value for n-person games. 1953.

[Sac14] Tomasz Sacha. Imatinib in chronic myeloid leukemia: an overview. *Mediterranean journal of hematology and infectious diseases*, 6(1), 2014.

[Saz06] Murat Hüsnü Sazli. A brief review of feed-forward neural networks. 2006.

[SB58] Terence Sanger and Pallavi N. Baljekar. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65 6:386–408, 1958.

[SB18] Marc Santolini and Albert-László Barabási. Predicting perturbation patterns from the topology of biological networks. *Proceedings of the National Academy of Sciences*, 115(27):E6375–E6383, 2018.

[SBA+18] Anna-Karin Sjögren, Katarina Breitholtz, Ernst Ahlberg, Lucas Milton, Malin Forsgard, Mikael Persson, Simone H Stahl, Martijn J Wilmer, and Jorrit J Hornberg. A novel multi-parametric high content screening assay in ciptec-oat1 to predict drug-induced nephrotoxicity during drug discovery. *Archives of Toxicology*, 92:3175–3190, 2018.

[SBOPM16] Maya Shamir, Y. Bar-On, Rob Phillips, and Ron Milo. Snapshot: Timescales in cell biology. *Cell*, 164:1302–1302.e1, 2016.

[SC06] Eduardo D Sontag and Madalena Chaves. Exact computation of amplification for a class of nonlinear systems arising from cellular signaling pathways. *Automatica*, 42(11):1987–1992, 2006.

[SGN+20] Damian Szklarczyk, Annika L. Gable, Katerina C. Nastou, David Lyon, Rebecca Kirsch, Sampo Pyysalo, Nadezhda T. Doncheva, Marc Legeay, Tao Fang, Peer Bork, Lars Juhl Jensen, and Christian von Mering. The string database in 2021: customizable protein–protein networks, and functional characterization of user-uploaded gene/measurement sets. *Nucleic Acids Research*, 49:D605 – D612, 2020.

[SHF+21] Yunsheng Shi, Zhengjie Huang, Shikun Feng, Hui Zhong, Wenjin Wang, and Yu Sun. Masked label prediction: Unified message passing model for semi-supervised classification, 2021.

[SHK+14] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

[SHZ09] Gideon Schreiber, Gilad Haran, and Huan-Xiang Zhou. Fundamental aspects of protein-protein association kinetics. *Chemical reviews*, 109:839–60, 03 2009.

[SLB+20] Sahil Sandhu, Anthony L Lin, Nathan Brajer, Jessica Sperling, William Ratliff, Armando D Bedoya, Suresh Balu, Cara O'Brien, and Mark P Sendak. Integrating a machine learning system into clinical workflows: qualitative study. *Journal of Medical Internet Research*, 22(11):e22421, 2020.

[SLJ+14] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions, 2014.

[SLR18] Jeffrey S Smith, Robert J Lefkowitz, and Sudarshan Rajagopal. Biased signalling: from simple switches to allosteric microprocessors. *Nature reviews Drug discovery*, 17(4):243–260, 2018.

[SLRC+15] Brinton Seashore-Ludlow, Matthew G Rees, Jaime H Cheah, Murat Cokol, Edmund V Price, Matthew E Coletti, Victor Jones, Nicole E Bodycombe, Christian K Soule, Joshua Gould, et al. Harnessing connectivity in a large-scale small-molecule sensitivity dataset. *Cancer discovery*, 5(11):1210–1223, 2015.

[SLRZ17] Daniela Senft, Mark DM Leiserson, Eytan Ruppin, and A Ronai Ze'ev. Precision oncology: the road ahead. *Trends in molecular medicine*, 23(10):874–898, 2017.

[SMF11] Thomas Schaffter, Daniel Marbach, and Dario Floreano. Genenetweaver: in silico benchmark generation and performance profiling of network inference methods. *Bioinformatics*, 27(16):2263–2270, 2011.

[SMWB+16] Tanveer Syeda-Mahmood, Eugene Walach, David Beymer, Flora Gilboa-Solomon, Mehdi Moradi, Pavel Kisilev, Deepika Kakrania, C Compas, Hongzhi Wang, R Negahdar, et al. Medical sieve: a cognitive assistant for radiologists and cardiologists. In *Medical Imaging 2016: Computer-Aided Diagnosis*, volume 9785, pages 58–63. SPIE, 2016.

[SNC+17a] Aravind Subramanian, Rajiv Narayan, Steven M. Corsello, David Peck, Ted E. Natoli, Xiaodong Lu, Joshua Gould, John F. Davis, Andrew A. Tubelli, Jacob K. Asiedu, David L. Lahr, Jodi E. Hirschman, Zihan Liu, Melanie K. Donahue, Bina Julian, Mariya Khan, David Wadden, Ian Smith, Daniel Lam, Arthur Liberzon, Courtney Toder, Mukta Bagul, Marek Orzechowski, Oana M. Enache, Federica Piccioni, Sarah Johnson, Nicholas J. Lyons, Alice H. Berger, Alykhan F. Shamji, Angela N. Brooks, Anita Vrcic, Corey Flynn, Jacqueline Rosains, David Y. Takeda, Roger Hu, Desiree Davison, Justin Lamb, Kristin G. Ardlie, Larson J Hogstrom, Peyton Greenside, Nathanael S. Gray, Paul A. Clemons, Serena J. Silver, Xiaoyun Wu, Wen-Ning Zhao, Willis Read-Button, Xiaohua Wu, Stephen J. Haggarty, Lucienne V. Ronco, Jesse S. Boehm, Stuart L. Schreiber, John G. Doench, Joshua A. Bittker, David E. Root, Bang Wong, and Todd R. Golub. A next generation connectivity map: L1000 platform and the first 1,000,000 profiles. *Cell*, 171:1437–1452.e17, 2017.

[SNC+17b] Aravind Subramanian, Rajiv Narayan, Steven M. Corsello, David Peck, Ted E. Natoli, Xiaodong Lu, Joshua Gould, John F. Davis, Andrew A. Tubelli, Jacob K. Asiedu, David L. Lahr, Jodi E. Hirschman, Zihan Liu, Melanie K. Donahue, Bina Julian, Mariya Khan, David Wadden, Ian Smith, Daniel Lam, Arthur Liberzon, Courtney Toder, Mukta Bagul, Marek Orzechowski, Oana M. Enache, Federica Piccioni, Sarah Johnson, Nicholas J. Lyons, Alice H. Berger, Alykhan F. Shamji, Angela N. Brooks, Anita Vrcic, Corey Flynn, Jacqueline Rosains, David Y. Takeda, Roger Hu, Desiree Davison, Justin Lamb, Kristin G. Ardlie, Larson J Hogstrom, Peyton Greenside, Nathanael S. Gray, Paul A. Clemons, Serena J. Silver, Xiaoyun Wu, Wen-Ning Zhao, Willis Read-Button, Xiaohua Wu, Stephen J. Haggarty, Lucienne V. Ronco, Jesse S. Boehm, Stuart L. Schreiber, John G. Doench, Joshua A. Bittker, David E. Root, Bang Wong, and Todd R. Golub. A next generation connectivity map: L1000 platform and the

first 1,000,000 profiles. *Cell*, 171:1437–1452.e17, 2017.

[SNZCE19] Hossein Sharifi-Noghabi, Olga Zolotareva, Colin C Collins, and Martin Ester. Moli: multi-omics late integration with deep neural networks for drug response prediction. *Bioinformatics*, 35(14):i501–i509, 2019.

[Soc08] American Cancer Society. *Cancer facts & figures*. The Society, 2008.

[Spe87] C. Spearman. The proof and measurement of association between two things. by c. spearman, 1904. *The American journal of psychology*, 100 3-4:441–71, 1987.

[SS21a] Ilana Schlam and Sandra M Swain. Her2-positive breast cancer and tyrosine kinase inhibitors: the time is now. *NPJ breast cancer*, 7(1):56, 2021.

[SS21b] Robert C Sterner and Rosalie M Sterner. Car-t cell therapy: current limitations and potential strategies. *Blood cancer journal*, 11(4):69, 2021.

[SSH+19] Bence Szalai, Vigneshwari Subramanian, Christian H Holland, Róbert Alföldi, László G Puskás, and Julio Saez-Rodriguez. Signatures of cell death and proliferation in perturbation transcriptomics data—from confounding factor to effective prediction. *Nucleic Acids Research*, 47(19):10010–10026, 2019.

[STK+20] Vasileios Stathias, John Turner, Amar Koleti, Dusica Vidovic, Daniel Cooper, Mehdi Fazel-Najafabadi, Marcin Pilarczyk, Raymond Terryn, Caty Chung, Afoma Umeano, et al. Lincs data portal 2.0: next generation access point for perturbation-response signatures. *Nucleic acids research*, 48(D1):D431–D439, 2020.

[STY17] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks, 2017.

[SvPL+07] JH Schiller, J von Pawel, T Larson, SI Ou, SA Limentani, AB Sandler, EE Vokes, S Kim, KF Liau, and PW Bycott. Efficacy and safety of single-agent axitinib (ag-013736; ag) in patients with advanced non–small-cell lung cancer (nsclc): A phase ii trial. *Clinical Lung Cancer*, 8(7):452, 2007.

[SYT+15] Nidhi Sahni, S. Stephen Yi, Mikko Taipale, Juan Fuxman Bass, Jasmin Coulombe-Huntington, Fan Yang, Jian Peng, Jochen Weile, Georgios Ioannis Karras, Yang Wang, István A. Kovács, Atanas Kamburov, Irina Krykbaeva, Mandy Hiu Yi Lam, George Tucker, Vikram Khurana, Amitabh Sharma, Yang-Yu Liu, Nozomu Yachie, Quan Zhong, Yun Shen, Alexandre Palagi, Adriana San-Miguel, Changyu Fan, Dawit Balcha, Amélie Dricot, Daniel M. Jordan, Jennifer M. Walsh, Akash A. Shah, Xinping Yang, Ani K Stoyanova, Alexander T. Leighton, Michael A. Calderwood, Yves Jacob, Michael E. Cusick, Kourosh Salehi-Ashtiani, Luke Whitesell, Shamil R. Sunyaev, Bonnie Berger, Albert-László Barabási, Benoît Charloteaux, David E. Hill, Tong Hao, Frederick P. Roth, Yu Xia,

Albertha J. M. Walhout, Susan Lindquist, and Marc Vidal. Widespread macromolecular interaction perturbations in human genetic disorders. *Cell*, 161:647–660, 2015.

[TGY+] Siyi Tang, Amirata Ghorbani, Rikiya Yamashita, Sameer Rehman, Jared A. Dunnmon, James Zou, and Daniel L. Rubin. Data valuation for medical imaging using shapley value and application to a large-scale chest x-ray dataset. 11(1):8366. Number: 1 Publisher: Nature Publishing Group.

[Tib96] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 58(1):267–288, 1996.

[TKSR16] Dénes Türei, Tamás Korcsmáros, and J. Saez-Rodriguez. Omnipath: guidelines and gateway for literature-curated signaling pathway resources. *Nature Methods*, 13:966–967, 2016.

[TPB+20] Richard Tomsett, Alun Preece, Dave Braines, Federico Cerutti, Supriyo Chakraborty, Mani Srivastava, Gavin Pearson, and Lance Kaplan. Rapid trust calibration through interpretable and uncertainty-aware ai. *Patterns*, 1(4), 2020.

[TSWJ16] Lindsey A Torre, Rebecca L Siegel, Elizabeth M Ward, and Ahmedin Jemal. Global cancer incidence and mortality rates and trends—an update. *Cancer epidemiology, biomarkers & prevention*, 25(1):16–27, 2016.

[TTB+18] Jeffrey W Tyner, Cristina E Tognon, Daniel Bottomly, Beth Wilmot, Stephen E Kurtz, Samantha L Savage, Nicola Long, Anna Reister Schultz, Elie Traer, Melissa Abel, et al. Functional genomic landscape of acute myeloid leukaemia. *Nature*, 562(7728):526–531, 2018.

[TTE+17] Pauline Traynard, Luis Tobalina, Federica Eduati, Laurence Calzone, and Julio Saez-Rodriguez. Logic modeling in quantitative systems pharmacology. *CPT: pharmacometrics & systems pharmacology*, 6(8):499–511, 2017.

[TVM+17] Aviad Tsherniak, Francisca Vazquez, Phil G Montgomery, Barbara A Weir, Gregory Kryukov, Glenn S Cowley, Stanley Gill, William F Harrington, Sasha Pantel, John M Krill-Burger, et al. Defining a cancer dependency map. *Cell*, 170(3):564–576, 2017.

[UVL16] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016.

[VBS+10] Charles J Vaske, Stephen C Benz, J Zachary Sanborn, Dent Earl, Christopher Szeto, Jingchun Zhu, David Haussler, and Joshua M Stuart. Inference of patient-specific pathway activities from multi-dimensional cancer genomics data using paradigm. *Bioinformatics*, 26(12):i237–i245, 2010.

[VCC+17a] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

[VCC+17b] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio', and Yoshua Bengio. Graph attention networks. *ArXiv*, abs/1710.10903, 2017.

[VSP+17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[VSP+23] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023.

[Wel09] Max Welling. Herding dynamical weights to learn. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, page 1121–1128, New York, NY, USA, 2009. Association for Computing Machinery.

[WFL+19] Shangying Wang, Kai Fan, Nan Luo, Yangxiaolu Cao, Feilun Wu, Carolyn Zhang, Katherine A Heller, and Lingchong You. Massive computational acceleration by using neural networks to emulate mechanism-based biological models. *Nature communications*, 10(1):4354, 2019.

[WH17] Guanming Wu and Robin Haw. Functional interaction network construction and analysis for disease discovery. *protein bioinformatics: from protein modifications and networks to proteomics*, pages 235–253, 2017.

[WKG+06] David S Wishart, Craig Knox, An Chi Guo, Savita Shrivastava, Murtaza Hassanali, Paul Stothard, Zhan Chang, and Jennifer Woolsey. Drugbank: a comprehensive resource for in silico drug discovery and exploration. *Nucleic acids research*, 34(suppl_1):D668–D672, 2006.

[WM97] David H. Wolpert and William G. Macready. No free lunch theorems for optimization. *IEEE Trans. Evol. Comput.*, 1:67–82, 1997.

[WNC15] Peng Wu, Thomas E Nielsen, and Mads H Clausen. Fda-approved small-molecule kinase inhibitors. *Trends in pharmacological sciences*, 36(7):422–439, 2015.

[WSB+20] James W. Wilson, Dilem Shakir, Michael Batie, Mark Frost, and Sonia Rocha. Oxygen-sensing mechanisms in cells. *The FEBS Journal*, 287(18):3888–3906, 2020.

[WSL19] Chi Heem Wong, Kien Wei Siah, and Andrew W Lo. Estimation of clinical trial success rates and related parameters. *Biostatistics*, 20(2):273–286, 2019.

[WW06] Robert A Weinberg and Robert A Weinberg. *The biology of cancer*. WW Norton & Company, 2006.

[WZC19] Xuan Wang, Haiyun Zhang, and Xiaozhuo Chen. Drug resistance and combating drug resistance in cancer. *Cancer Drug Resistance*, 2(2):141, 2019.

[WZTE18] Tongzhou Wang, Jun-Yan Zhu, Antonio Torralba, and Alexei A. Efros. Dataset distillation, 2018.

[XDL20] Yifan Xue, Michael Q Ding, and Xinghua Lu. Learning to encode cellular responses to systematic perturbations with deep generative models. *NPJ systems biology and applications*, 6(1):35, 2020.

[XHLJ18] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? *ArXiv*, abs/1810.00826, 2018.

[XL01] Qing-Song Xu and Yi-Zeng Liang. Monte carlo cross validation. *Chemometrics and Intelligent Laboratory Systems*, 56(1):1–11, 2001.

[XLT+18] Keyulu Xu, Chengtao Li, Yonglong Tian, Tomohiro Sonobe, Ken ichi Kawarabayashi, and Stefanie Jegelka. Representation learning on graphs with jumping knowledge networks. *ArXiv*, abs/1806.03536, 2018.

[XSML23] Sang Michael Xie, Shibani Santurkar, Tengyu Ma, and Percy Liang. Data selection for language models via importance resampling, 2023.

[YAP] Jinsung Yoon, Sercan O. Arik, and Tomas Pfister. Data valuation using reinforcement learning. Number: arXiv:1909.11671.

[YBY+19] Rex Ying, Dylan Bourgeois, Jiaxuan You, Marinka Zitnik, and Jure Leskovec. Gnnexplainer: Generating explanations for graph neural networks, 2019.

[YLW23] Ruonan Yu, Songhua Liu, and Xinchao Wang. Dataset distillation: A comprehensive review, 2023.

[YMY+16] Channing Yu, Aristotle M Mannan, Griselda Metta Yvone, Kenneth N Ross, Yan-Ling Zhang, Melissa A Marton, Bradley R Taylor, Andrew Crenshaw, Joshua Z Gould, Pablo Tamayo, et al. High-throughput identification of genotype-specific cancer vulnerabilities in mixtures of barcoded tumor cell lines. *Nature biotechnology*, 34(4):419–423, 2016.

[YSG+12] Wanjuan Yang, Jorge Soares, Patricia Greninger, Elena J Edelman, Howard Lightfoot, Simon Forbes, Nidhi Bindal, Dave Beare, James A Smith, I Richard Thompson, et al. Genomics of drug sensitivity in cancer (gdsc): a resource for therapeutic biomarker discovery in cancer cells. *Nucleic acids research*, 41(D1):D955–D961, 2012.

[YSL+21] Bo Yuan, Ciyue Shen, Augustin Luna, Anil Korkut, Debora S Marks, John Ingraham, and Chris Sander. Cellbox: interpretable machine learning for perturbation biology with application to the design of cancer combination therapy. *Cell systems*, 12(2):128–140, 2021.

[ZA20] Lingxiao Zhao and Leman Akoglu. Pairnorm: Tackling oversmoothing in gnns, 2020.

[Zen99] Langche Zeng. Logistic regression in rare events data 1. 1999.

[ZGE+14] Marjorie Glass Zauderer, Ayca Gucalp, Andrew S Epstein, Andrew David Seidman, Aryeh Caroline, Svetlana Granovsky, Julia Fu, Jeffrey Keesing, Scott Lewis, Heather Co, et al. Piloting ibm watson oncology within memorial sloan kettering's regional network., 2014.

[ZLC+20a] Wei Zhao, Jun Li, Mei-Ju Chen, Rehan Akbani, Yiling Lu, Gordon Mills, and Han Liang. An atlas of perturbed functional proteomics profiles of cancer cell lines. *Cancer Research*, 80(16_Supplement):5139–5139, 2020.

[ZLC+20b] Wei Zhao, Jun Li, Mei-Ju M Chen, Yikai Luo, Zhenlin Ju, Nicole K Nesser, Katie Johnson-Camacho, Christopher T Boniface, Yancey Lawrence, Nupur T Pande, et al. Large-scale characterization of drug responses of clinically relevant proteins in cancer cell lines. *Cancer cell*, 38(6):829–843, 2020.

[ZLP+22] Xin Zheng, Yixin Liu, Shirui Pan, Miao Zhang, Di Jin, and Philip S. Yu. Graph neural networks for graphs with heterophily: A survey, 2022.

[ZS18] Zhilu Zhang and Mert Rory Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. *ArXiv*, abs/1805.07836, 2018.

[ZSH+19] Chuxu Zhang, Dongjin Song, Chao Huang, Ananthram Swami, and Nitesh V. Chawla. Heterogeneous graph neural network. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, page 793–803, New York, NY, USA, 2019. Association for Computing Machinery.

[ZYZ+20] Jiong Zhu, Yujun Yan, Lingxiao Zhao, Mark Heimann, Leman Akoglu, and Danai Koutra. Beyond homophily in graph neural networks: Current limitations and effective designs, 2020.

9 Appendix

9.1 DVGS Robustness to Hyperparameters

To test the robustness of the DVGS algorithm to the hyperparameters used, we performed a limited grid search on the ADULT dataset with 20% corrupted endogenous labels. We record the ability of DVGS to identify the corrupted labels across all tested hyperparameters. Figure 52 shows the cumulative distribution function of the resulting AUROC values (characterizing the ability to classify corrupted labels), across all tested hyperparameters. Of note, we find that almost 85% of the tested hyperparameter configurations resulted in performance within 25% of the maximum performance, and more that 50% of the tested hyperparameters resulted in performance within 10% of maximum performance, indicating that the DVGS method is robust to choice of hyperparameters. The hyperparameter configurations tested are shown in Table 26.

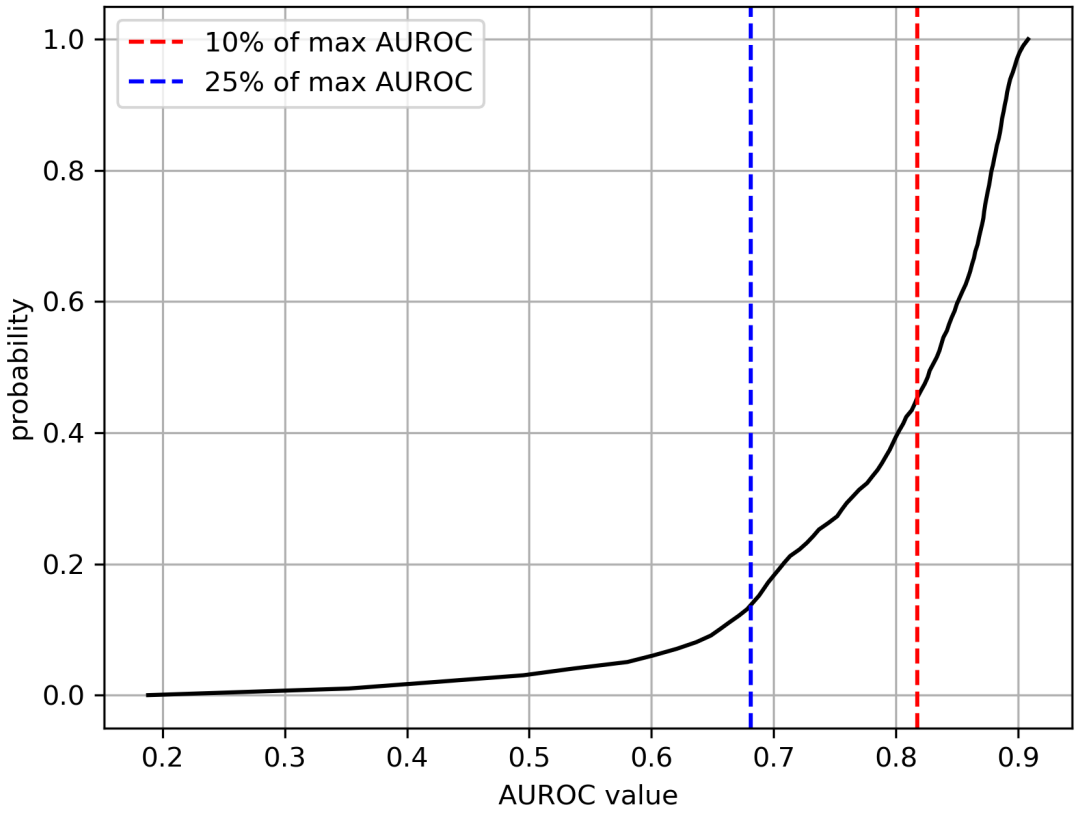


Figure 52: The cumulative distribution function (CDF) of $AUROC(-\nu_i, c_i)$ across all tested hyperparameters, where ν_i are data values generated by DVGS and c_i are the corrupted labels label. The red dashed line demarkates all AUROC values larger than this are within 10% of the max AUROC value (e.g., roughly 55% of all tested hyperparameters resulted in an AUROC value within 10% of the max AUROC).

Table 26: The DVGS hyperparameter configurations tested in a grid search with 2 replicates per configuration.

Hyperparameter	Values	Best Performing Value (mean)
balanced class weights	True, False	False
dropout	0, 0.25, 0.5, 0.75	0.25
target batch size	100, 200, 400	200
similarity	Euclidean, Cosine Similarity, Dot Product, Scalar Projection	Euclidean
lr	1e-2, 1e-3, 1e-4	1e-3
instance norm	True, False	True
number layers	1,2	1
activation fn.	Mish, ReLU	Mish

9.2 Average Pearson Correlation (APC) metric

We compute the previously proposed Average Pearson Correlation (APC) [PQZ$^+$] of LINCS level 4 replicates by:

For a given level 5 LINCS sample:

- Identify the level 4 bio-replicate *sample id*s that were used to generate the level 5 aggregate sample.
- Load the level 4 sample id expression profile into memory
- Filter to select only landmark genes (978)
- Compute the average pairwise Pearson correlation of level 4 bio-replicates

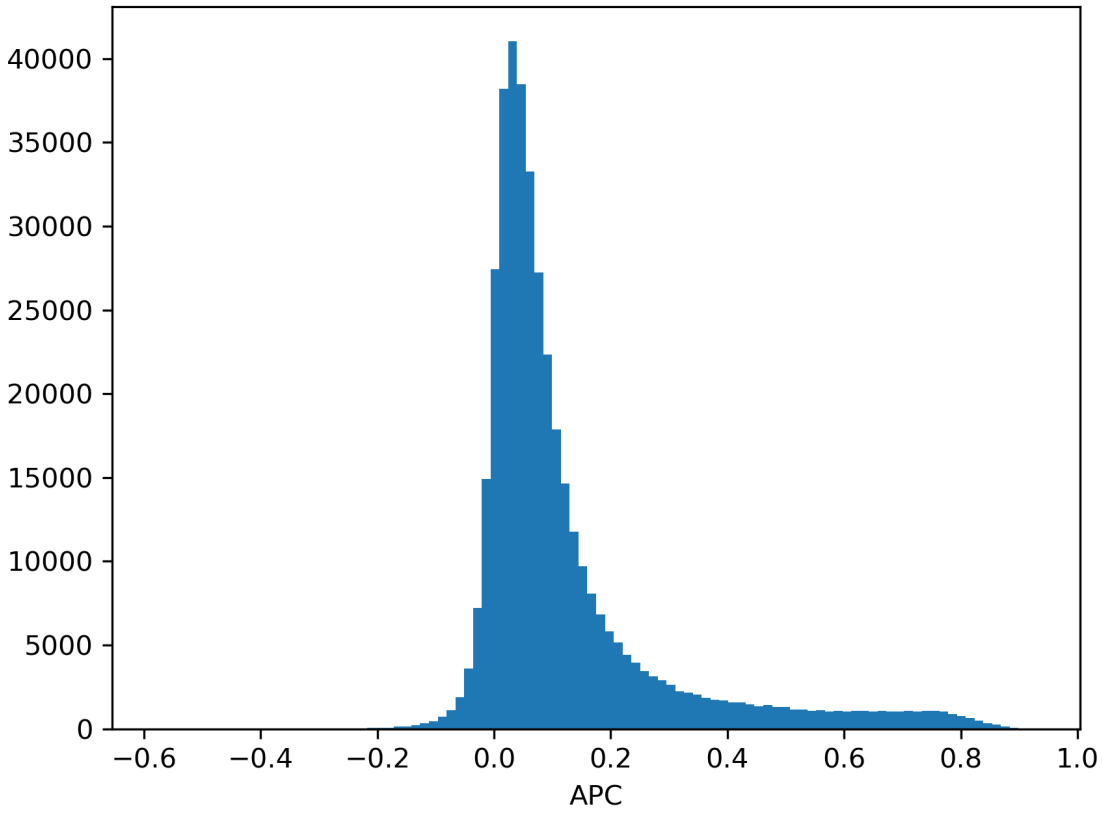


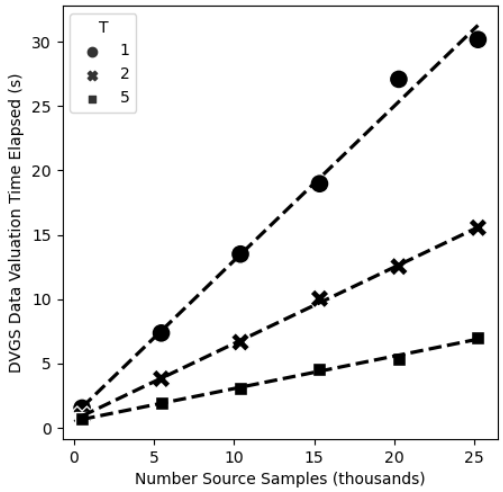
Figure 53: The Average Pearson Correlation of level 5 LINCS samples distribution.

9.3 Additional DVGS Runtime Experiments

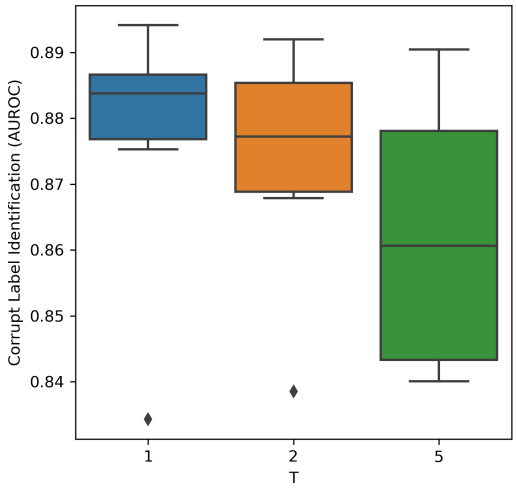
In figure 54 we show the experimental results of DVGS as the number of source samples increases. As expected, DVGS scales linearly with the number of source samples, divided by the period of gradient computations (T). In figure 54b we show the ability of DVGS to classify corrupted labels, when we increase the value of T, as one would expect, the AUROC value decreases with larger T; however, the marginal decreases in performance may be worthwhile for improvements in runtime, especially on large datasets. When applying our method to the LINCS dataset, we were able to run 500 epochs of DVGS on 710,216 source samples using a multilayer autoencoder neural network (Number parameters > 650k) in roughly 8 hours on a NVIDIA 3090 GPU.

The memory requirements of most problems are comparable to classical SGD optimization problems; however, the computation of high-dimensional sample-wise gradients can increase the memory requirements. Therefore, as the number of model parameters increases, the memory footprint of the sample gradients will also increase. To mitigate this issue, we chose to compute sample gradients in mini-batches, which can be manually specified to fit a given task.

Reducing the source batch size will therefore reduce the memory footprint, but lead to a small increase in computation time.



(a) DVGS runtime on the ADULT dataset, compared to the number of source samples.



(b) Ability of DVGS to identify corrupted labels, with different values of T (period of source gradient computations.

Figure 54: The scalability and performance of the DVGS method dependant on number of source samples and the period of source similarity computations (T).

9.4 Data Valuation with Gradient Alignment

We propose a method of Data Valuation with Gradient Alignment (DVGA), which takes two datasets (source and target), and attempts to select the source samples that most closely align the source loss surface with the target loss surface; this premise is visualized in Figure 55. We posit that as the source loss surface becomes more similar to the target loss surface, the loss minimas will also align; therefore, models trained using gradient descent on the source dataset are liable to generalize to the target dataset. Additionally, we posit that mislabeled samples or noisy exogenous features are likely to encourage misalignment, and therefore, we expect the selected samples that align the loss surfaces to be depleted for noisy data.

Notably, the calculations of closed-form loss surfaces are intractable for most complex functions; however, the comparison of gradients during classical stochastic gradient descent (SGD) optimization is tractable and simple to implement. We rationalize that by iterative alignment of the local source gradient with the target gradient during SGD, we can approximate the global loss surface alignment.

Identical to other data valuation methods, DVGA requires a target dataset, which characterizes our desired predictive task. The target dataset may be of high quality, a specific prediction domain, or a randomly sampled holdout set. Our approach additionally requires two predictive models, and we follow the naming conventions used by Yoon, et. al [YAP]:

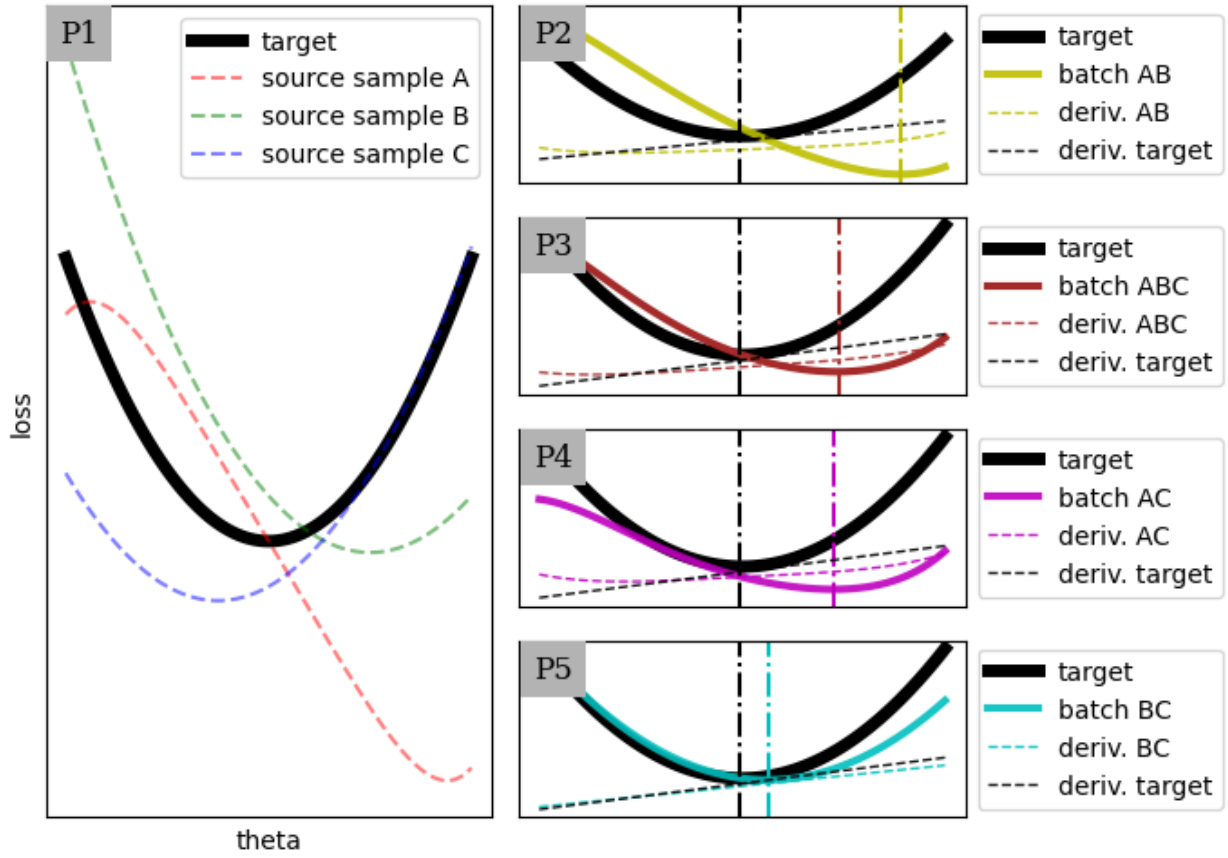


Figure 55: Toy example of the possible alignments of three source samples (A,B,C). Panel 1 (P1) shows the target loss surface (black) and three source sample loss surfaces (r,g,b); Note that in all figures the x-axis is an arbitrary 1-d model parameter θ and y-axis is the loss at that parameter value. In Panels 2-5 we show the possible combinations of source samples and resulting loss surfaces and gradients. Vertical dashed lines in panels 2-5 indicate the loss minimas. Our DVGA method attempts to select a subset of source samples which most closely align the source loss surface to the target loss surface; In our toy example panel 5 illustrates the ideal source subset (BC).

- a **predictor model** ($\hat{y}_i = f_\theta(X_i)$): The parameters are optimzed on a subset of the source dataset.

- an **estimator model** ($w_i = f_\phi(X_i, y_i, i)$): The parameters are optimized by minimizing the difference between source and target gradients through the selection of source samples. The estimator model can predict the selection weight (w_i) from X,y, X & y or the sample index (e.g., sample assigned weight).

The source dataset will be used to train the predictor model and on which data values will be assigned, with the goal of characterizing useful or detrimental samples. Additionally, DVGA requires a differentiable predictive model that can be trained using gradient descent. Algorithm 1 describes the DVGA method. Our goal is to train an estimator model that selects source samples that will most closely align the target and source gradients along the entire gradient descent path; however, our algorithm is liable to select different source samples in different regions of θ, and thus overfit to local

Algorithm 4 Data Valuation with Gradient Alignment

Require: Given learning rates (α, σ), predictor model (θ), estimator model (ϕ), number of restarts (R), number of epochs (N), number of inner iterations (I), source batch size (K), target batch size (J), target dataset ($\mathcal{D}_t$), source dataset ($\mathcal{D}_s$), predictor loss function ($\mathcal{L}$) and estimator loss function ($\mathcal{C}$)

$\phi \leftarrow initialize$
for $r = 1, 2, \ldots, R$ **do**
  $\theta \leftarrow initialize$
  for $n = 1, 2, \ldots, N$ **do**
    $X_t, y_t \sim \mathcal{D}_t$
    $\hat{y}_t \leftarrow f_\theta(X_t)$
    $l_t \leftarrow \mathcal{L}(y_t, \hat{y}_t)$
    $\nabla\mathcal{L}_t \leftarrow \frac{d}{d\theta}(l_t)$
    for $X_s, y_s \sim \mathcal{D}_s$ **do**
      for $k = 1, 2, \ldots, K$ **do**
        $\hat{y}_s^k \leftarrow f_\theta(X_s^k)$
        $l_s^k \leftarrow \mathcal{L}(y_s^k, \hat{y}_s^k)$
        $\nabla\mathcal{L}_s^k \leftarrow \frac{d}{d\theta}(l_s^k)$
      end for
      for $i = 1, 2, \ldots, I$ **do**
        for $k = 1, 2, \ldots, K$ **do**
          $w_s^k \leftarrow f_\phi(X_s, y_s, k)$
        end for
        $\nabla\mathcal{L}_s \leftarrow \frac{1}{\sum_{k=0}^{K} w_s^k} \sum_{k=0}^{K} w_s^k \nabla\mathcal{L}_s^k$
        $\kappa_{reg} \leftarrow \beta \sum_{k=0}^{K} (w_s^k - 0.5)^2 - \gamma \sum_{k=0}^{K} w_s^k$
        $c \leftarrow \mathcal{C}(\nabla\mathcal{L}_t, \nabla\mathcal{L}_s) + \kappa_{reg}$
        $\phi \leftarrow \phi - \sigma\nabla c$
      end for
      $\theta \leftarrow \theta - \alpha\nabla\mathcal{L}_s$
    end for
  end for
end for

regions of the loss surface. In practice, we have found that using a small estimator learning rate (β) and a single inner iteration (I) prevents the estimator from overfitting to any local region of θ.

The estimator model can take x,y or the sample index as input and predicts a sample weight between 0,1, which is used for selection of sample gradients. Although this sample weight can be a continuous value, we have found that using categorical reparameterization of the softmax-gumbel distribution to discretize selection weights works well in practice [JGP17].

Additionally, we introduce two hyperparameters intended to regularize the estimator learning: γ encourages the estimator to include as many samples as possible by applying a negative L1 regularization of selection weights, and β prevents confidence by an L2 regularization on the selection probabilities centered around 0.5.

In essence, this approach can be conceptualized as stochastic gradient descent on a subset of the source data set, where the selection of the source subset is guided by similarity to the target gradients.

Although we do not have mathematical justification that this approach satisfies the three equitable data value conditions proposed by Ghorbani et al. [GZ], we experimentally show that this approach:

- effectively characterizes data quality in many real-world prediction
- duplicate samples have equitable data values
- is scalable to large datasets
- extensible to a wide range of model architectures and predictive tasks

To train the estimator model, we must define a loss function to characterize the similarity of two high-dimensional gradient vectors ($\nabla\mathcal{L}_s$, $\nabla\mathcal{L}_t$). Although any multi-output regression cost function is applicable, we have primarily focused our attention on cosine similarity and mean squared error, both of which work well in practice. The notable distinctions of the two being that cosine similarity does not take magnitude into account, which can be conceptualized as matching the loss topology general shape (e.g., location of peaks and valleys) but not necessarily the magnitude of the topology (e.g., different depths of valleys or heights of peaks).

A challenge with this approach is sample selection may vary depending on the region of the loss surface the DVGA algorithm is exploring, if the learning rates are too large, it may preferentially select unique sample sets during early- and late-stage training. This problem can be partially mitigated by choosing a small learning rate that prevents rapid changes in sample selection. Even with this mitigation strategy, this problem raises the question: *Should gradient alignment be global (based on early and late stage training) or local (based on early or late)?* Future directions should address how to answer this question and mitigate issues that arise from local gradient alignment.

We provide an early implementation of the DVGA algorithm in: `https://github.com/nathanieljevans/DVGA`.

9.5 GNNCDR hyper-parameter description

For results described in Chapter 5, we describe here the hyperparameters used. We selected LINCS L1000 observations that were measured 24 hours after the introduction of the perturbagen. We filter all predicted Reactome FI edges and include only edges with a "score" of 1 (high confidence). For inclusion in the test set, we ensure that:

- Each drug included has at least 50 observations.
- Cell lines in the test set must have at least 10 drugs.
- Drugs in the test set must be measured in at least 3 cell lines.
- Cell lines in the test set must have at least 50 observations.

  We randomly assign 70% of the keys (drug, cell line) to the train partition and 30% of the keys to the test partition, resulting in 29119 training observations and 7951 test observations. We include unperturbed 'omics as exogenous node features:

- RNA expression: Scaled by 14 to ensure that most expression values are between 0 and 1 [GHJV$^+$19].
- Copy number variation (CNV): Scaled by 7 to ensure that most values are between 0 and 1 [GHJV$^+$19].

Table 27: The GNNCDR parameters used in Experiment 01.

Parameter	Exp. 01	Exp. 02	Description
Reactome Pathway	R-HSA-9006934	R-HSA-162582	The reactome pathway ID used to create the biological subgraph
GNN_HIDDEN_CHANNELS_LAYER1	32	8	The hidden channels used for the cell-context convolution
WEIGHT_DECAY	1e-6	1e-5	L2 weight decay on all model parameters
DRUG_EMBEDDING_DIM	3	3	Number of features used for the drug embedding
GENE_EMBEDDING_DIM	5	4	Number of features used for the gene embedding
CELL_EMBEDDING_DIM	5	5	Number of features used for the cell embedding
EDGE_EMBEDDING_DIM	2	1	Number of features used for the gene-gene edge embedding
REGULON_DROPOUT	0.3	0.5	Dropout probability used on the regulon prediction layer (TF->LINCS) linear weights.
NUM_LAYERS	10	10	Number of GNN layers used (not counting first cell context convolution)
HIDDEN_CHANNELS	8	8	Number of hidden channels used for consecutive GNN layers.
DROPOUT	0	0	Latent node dropout probability.
EPOCHS	25	25	Number of training epochs to perform.
GAMMA	0	0	L2 weight decay applied to only latent gene-gene edge weights.
KAPPA	0	0	L2 weight decay applied to only drug-gene edge weights. Parameter value of drug-gene weights are normalized -1,1.
WARMUP	0	0	Number of epochs to freeze the drug, gene, edge and cell embeddings. For use with pre-training.
LR	1e-2	5e-3	Optimization learning rate.
BATCH_SIZE	512	75	Number of observations in each training batch.
optim	Adam	Adam	Optimization algorithm used.

Table 28: The NaiveNN parameters used in experiments 01 and 02.

Parameter	Value	Description
EMBEDDING_DIM	250	Number of embedding channels used for the drug and cell embeddings.
HIDDEN_CHANNELS	256	Number of hidden channels.
LR	1e-3	Optimization learning rate
DO	0.1	Dropout probability
BATCH_SIZE	1000	Training batch size.
EPOCHS	25	Number of training epochs.
optim	Adam	Optimization algorithm.
nonlin	ELU	The nonlinearity function.

- Mutation (5 features): Functional effect annotations are one-hot encoded based on terms, 'damaging', 'other' and 'silent.' The allele count (AC) is encoded as a ratio. The number of mutations in a gene is scaled by 24 to ensure that the values are between 0 and 1 [GHJV$^+$19].
- Methylation is encoded as provided without any pre-processing (values between 0,1) [GHJV$^+$19].

To pre-train GNN embeddings, or alternatively, to inspect knowledge-naive embeddings, we use:

- The top 20 most-common PPI edge annotation words [GJS$^+$21, WH17]
- The top 50 most common Gene Ontology terms [ABB$^+$00]
- The top 10 most common drug mechanism-of-action annotation words [CBL$^+$17]
- Binding affinity information (K_i, K_d, IC_{50} or EC_{50}) from the Cancer Targetome [BCKM$^+$17]

All model configurations were trained and evaluated using the same training and test partitions.

9.6 GSNN Experiment Details

Each experiment is described by a set of hyperparameters that specify the nodes (drugs, proteins, RNA & LINCS genes), cell lines, and observations that will be included. One of the key parameters is the choice of proteins to be

included in the biological graph on which the GSNN will operate. Ideally, we would select a subset of proteins that are relevant to a certain drug or set of drugs; however, inferring which proteins are involved in a given drug response is a challenging task. In lieu of identifying proteins that are relevant to the drug response, we select proteins based on their involvement with specific biological processes or pathways and manually select the subset of pathways that we believe are likely to be involved in the drug response. In experiments 1-3, we first select a set *primary* Reactome pathways and then manually search for each primary pathway to identify *linked* pathways[20]. To avoid exceptionally large protein sets, we use our discretion to select *linked* pathways which we believe are relevant to the *primary* pathway. For instance, we generally avoided including pathways that are unlikely to be well modeled by the GSNN premise of cellular signaling such as the DNA damage response[21]. We also did not include disease pathways, which Reactome has designated as separate pathways; therefore, all specified pathways should be considered canonical and healthy signaling processes. The details of the experiment pathway are shown in Table 29. Although each experiment has a unique set of *primary* pathways, there are many included *linked* pathways which are shared by all three experiments. Additionally, many proteins have multiple roles in several pathways. This overlap in pathways and protein roles means that even distinct experiment pathway parameters may result in relatively similar biological networks. Figure 56 shows the overlapping entities between experiments 1-3. Of note, while most elements have substantial overlap between each experiment, the protein-space[22] is relatively distinct between each experiment.

[20]Pathways that are not subpathways but are referenced within a pathway, e.g., pathway A -> activates -> pathway B but pathway B is not a subpathway of A)

[21]We are concerned that DNA damage drugs will not be well represented by the drug-target premise

[22]all proteins included in the biological network

Table 29: (a-c) The pathways that were used in each experiment to specify the proteins included in the GSNN input graph. Bold text indicates the initial pathway choice from which all other pathways were "linked." Pathway size refers to the number of proteins in each Reactome pathway and may not reflect the exact number of proteins included in the resulting biological network.

Reactome ID	Description	Size
R-HSA-177929	**Signaling by EGFR**	**52**
R-HSA-1489509	DAG and IP3 signaling	41
R-HSA-1257604	PIP3 activates AKT signaling	282
R-HSA-5673001	RAF/MAP kinase cascade	292
R-HSA-1227986	**Signaling by ERBB2**	**56**
R-HSA-109606	Intrinsic Pathway for Apoptosis	55
R-HSA-6806003	Regulation of TP53 Expression and Degradation	37
R-HSA-202131	Metabolism of nitric oxide: NOS3 activation and regulation	26
R-HSA-6807070	PTEN Regulation	139

(a) Exp. 1

Reactome ID	Description	Size
R-HSA-73887	**Death Receptor Signaling**	**161**
R-HSA-75157	FasL/ CD95L signaling	5
R-HSA-140534	Caspase activation via Death Receptors in the presence of ligand	19
R-HSA-75158	TRAIL signaling	8
R-HSA-75893	TNF signaling	61
R-HSA-5218859	Regulated Necrosis	62
R-HSA-5213460	RIPK1-mediated regulated necrosis	35
R-HSA-5620971	Pyroptosis	27
R-HSA-109606	Intrinsic Pathway for Apoptosis	55
R-HSA-446652	Interleukin-1 family signaling	155
R-HSA-5686938	Regulation of TLR by endogenous ligand	21
R-HSA-193704	p75 NTR receptor-mediated signalling	99
R-HSA-187037	Signaling by NTRK1 (TRKA)	117
R-HSA-5673001	RAF/MAP kinase cascade	292
R-HSA-1257604	PIP3 activates AKT signaling	282
R-HSA-9031628	NGF-stimulated transcription	39
R-HSA-1489509	DAG and IP3 signaling	41

(b) Exp. 2

Reactome ID	Description	Size
R-HSA-201556	**Signaling by ALK**	**28**
R-HSA-1257604	PIP3 activates AKT signaling	282
R-HSA-165159	MTOR signalling	41
R-HSA-380972	Energy dependent regulation of mTOR by LKB1-AMPK	29
R-HSA-6807070	PTEN Regulation	139
R-HSA-109606	Intrinsic Pathway for Apoptosis	55
R-HSA-202131	Metabolism of nitric oxide: NOS3 activation and regulation	14
R-HSA-6806003	Regulation of TP53 Expression and Degradation	37
R-HSA-6804756	Regulation of TP53 Activity through Phosphorylation	92
R-HSA-5693606	DNA Double Strand Break Response	61
R-HSA-5673001	RAF/MAP kinase cascade	292
R-HSA-1489509	DAG and IP3 signaling	41

(c) Exp. 3

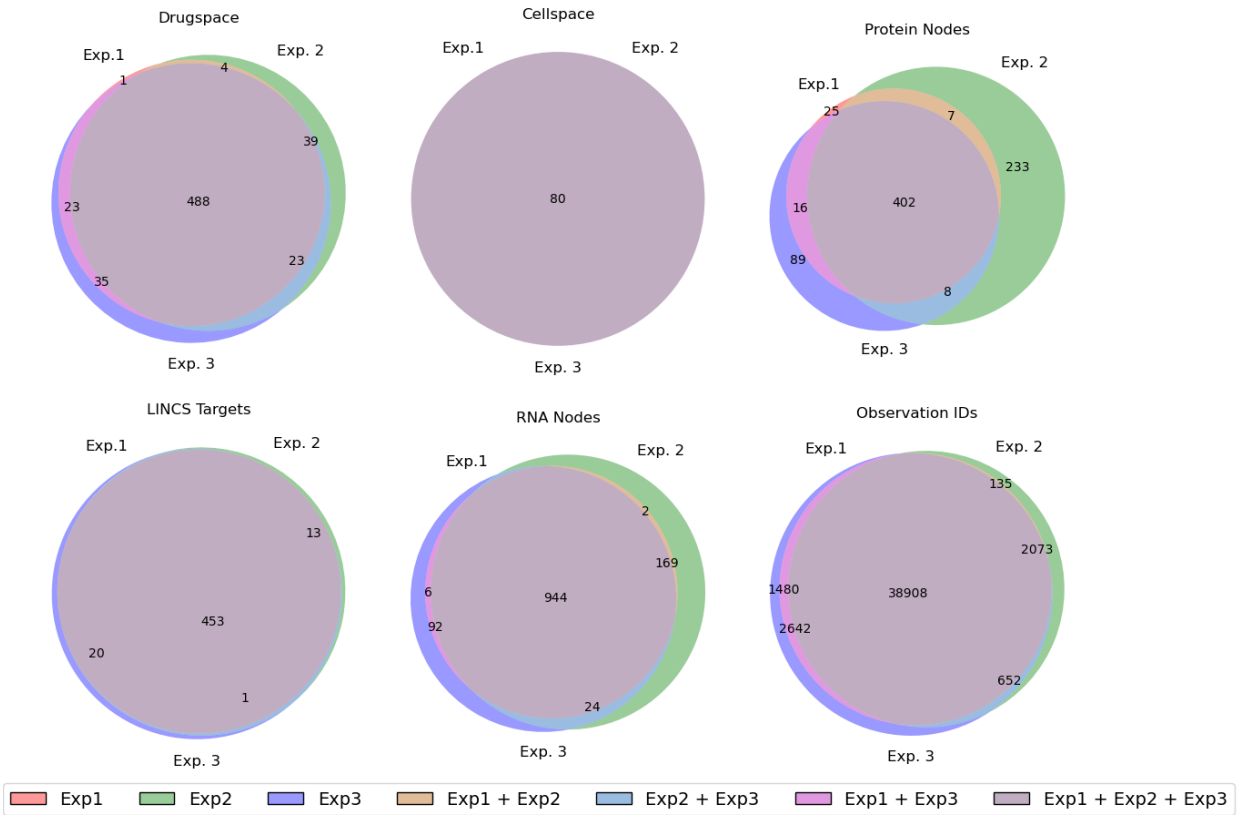


Figure 56: The overlapping elements of each experiment. Most of the targets, nodes and drugs are shared across all three experiments; however, there are distinct protein subsets for each of the three experiments.

9.7 Number of model parameters: GSNN vs. NN

In table 30 we report the number of GSNN and NN parameters in the best-performing models of each experiment. Across all three experiments, the best GSNN models of each fold had more trainable parameters than the best NN model from that fold. This may be indicative of the GSNN model being less prone to overfitting. Another explanation is that, due to the GSNN biological graph structure, there are likely to be many function nodes that are rarely involved in prediction logic or impact only a few targets (i.e., only a few LINCS nodes are descendants) and therefore the functional set of parameters may not be well represented by the total number of trainable parameters. In other words, prior knowledge may lead to some function nodes being effectively spurious or underutilized, and therefore the direct parameter comparison should be interpreted with caution.

Table 30: Number of trainable parameters of the GSNN and NN algorithms used in experiments 1-3 (median of best models from each MCCV fold). Percent change is calculated as $\frac{N_{gsnn}-N_{NN}}{N_{nn}}$, where N is the median number of algorithm parameters

EXP. ID	Num. GSNN params	Num. NN params	Percent Change
exp1	6.56e+06	2.23e+06	193.8 %
exp2	7.85e+06	5.68e+06	38.2 %
exp3	6.70e+06	5.3e+06	26.5 %
AVG.			86.2 %

9.8 Computational Complexity of the GSNN method

The GSNN algorithm takes significantly longer to train due to being a particularly deep architecture and due to it's use of sparse matrix operations. Table 31 reports the average training times for each algorithm. Specifically, the GSNN algorithm requires between 3-15 times as much training time as the alternative algorithms tested (NN, GNN). However, it should be noted that the training curves shown in Figure 57 compare the validation performance by epoch for representative GSNN and NN models; The GSNN validation performance increases markedly faster, achieving approximately the maximum NN performance in the first 20 epochs. This aspect of the training dynamics may suggest that the GSNN algorithm can be trained with fewer epochs, which would markedly reduce the compute requirements.

Table 31: Average training time of each algorithm (reported in minutes). Note: GSNN and GNN were trained on GPUs whereas the NNs were trained on CPU only.

EXP.	GSNN	GNN	NN	GSNN / NN	GSNN / GNN
exp1	419.8	105.0	29.0	14.5	4.0
exp2	493.3	138.4	32.4	15.2	3.6
exp3	460.8	133.9	35.9	12.9	3.4

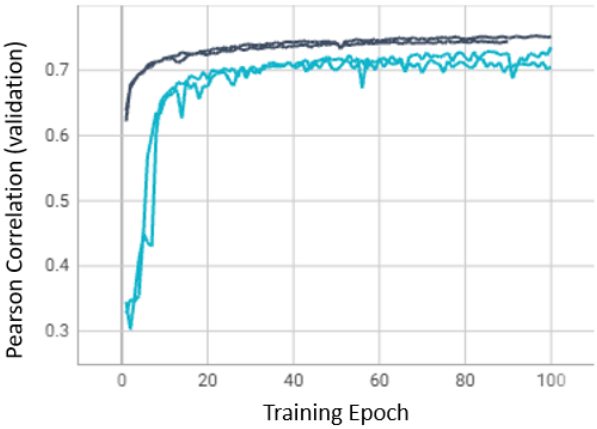


Figure 57: Representative training curves from experiment 1 (EGFR + ERBB2 signaling). Dark Gray/Blue indicates the GSNN training curves, light blue are NN training curves.

9.9 Effect of Layer Depth on GSNN performance

The GSNN algorithm passes information during sequential *layers* allowing information to diffuse through the network up to the number of layers L in the model. Cell signaling often involves many entities in many sequential interactions, as well as feedback loops that may alter behavior. Due to this trait, deeper networks may be more representative of the underlying biology and therefore more accurate. To test this, we compare the performance of the GSNN algorithms with a different number of layers (L=10,20). Figure 58 shows the results and suggests that 20-layer GSNNs have a small improvement in performance compared to 10-layer GSNNs. In particular, training deeper neural networks also introduces more parameters, greater memory complexity, and longer training times. It is therefore critical that the choice of GSNN layers be tailored to the available hardware and training budget. Improvements in the time and memory complexity of the GSNN algorithm may enable deeper and more accurate models.

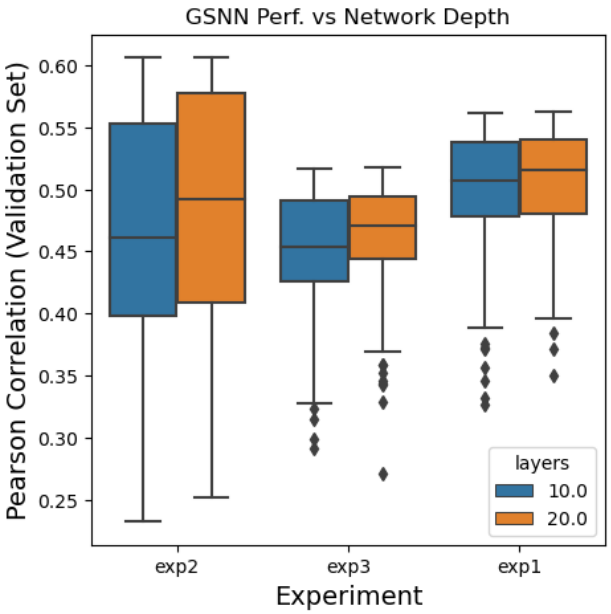


Figure 58: The performance of GSNN algorithms in experiments 1-3 compared by the number of layers (L) hyper-parameter.