Deep Transfer Learning for Drug Response Prediction

by

Hossein Sharifi Noghabi

B.Eng., Sadjad University of Technology, Mashhad, Iran, 2012
M.Sc., Ferdowsi University of Mashhad, Mashhad, Iran, 2015

Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of
Doctor of Philosophy

in the
School of Computing Science
Faculty of Applied Sciences

SIMON FRASER UNIVERSITY
Summer 2021

Declaration of Committee

Name:	**Hossein Sharifi Noghabi**
Degree:	**Doctor of Philosophy (Computer Science)**
Thesis title:	**Deep Transfer Learning for Drug Response Prediction**
Committee:	**Chair:** Uwe Glässer Professor, Computing Science
	Martin Ester Supervisor Professor, Computing Science
	Colin C. Collins Committee Member Professor, Urologic Sciences University of British Columbia
	Maxwell Libbrecht Examiner Assistant Professor, Computing Science
	Joseph Lehár External Examiner Adjunct Professor Boston University

Abstract

The goal of precision oncology is to make accurate predictions for cancer patients via some omics data types of individual patients. Major challenges of computational methods for drug response prediction are that labeled clinical data is very limited, not publicly available, or has drug response for one or two drugs. These challenges have been addressed by generating large-scale pre-clinical datasets such as cancer cell lines or patient-derived xenografts (PDX). These pre-clinical datasets have multi-omics characterization of samples and are often screened with hundreds of drugs which makes them viable resources for precision oncology. However, they raise new questions: how can we integrate different data types? how can we handle data discrepancy between pre-clinical and clinical datasets that exist due to basic biological differences? and how can we make the best use of unlabeled samples in drug response prediction where labeling is extra challenging? In this thesis, we propose methods based on deep neural networks to answer these questions. First, we propose a method of multi-omics integration. Second, we propose a transfer learning method to address data discrepancy between cell lines, patients, and PDX models in the input and output space. Finally, we proposed a semi-supervised method of out-of-distribution generalization to predict drug response using labeled and unlabeled samples. The proposed methods have promising performance when compared to the state-of-the-art and may guide precision oncology more accurately.

Keywords: Deep Neural Networks, Transfer Learning, Drug Response Prediction, Pharmacogenomics, Multi-Omics Integration, Semi-Supervised Learning, Domain Generalization.

Dedication

For my family, for their unconditional love, support, and inspiration especially my amazing sister Hamide.

Acknowledgements

Finishing a PhD is most certainly not an easy milestone. I feel extremely lucky that I received support – scientifically and emotionally – from many wonderful people during this journey. First of all, I would like to thank my advisory committee Martin Ester and Colin Collins for training me, believing in me, and making me a better person both scientifically and personally.

Martin kept me focused and guided me every single day for the past 5 years which was not easy at all! I tend to jump around and find myself interested in many things, but most of the time PhD requires us to be laser focused on one topic.

Colin had this weird belief in me during my PhD even when I had no faith in myself. I had the opportunity to enjoy many conversations with him about precision medicine, prostate cancer, or The Grateful Dead! The very first thing that Colin said to me was "a stupid question is the one you don't ask" and it changed my PhD life.

I would like to thank my examining committee Maxwell Libbrecht and Joseph Lehár for sharing their insights and providing suggestions on how to extend this thesis in the future.

I also would like to thank my longstanding collaborator, Olga Zolotareva. She was involved in almost all of my PhD research and I learned a lot from her and always trusted her insight and viewpoints in different projects.

I was very fortunate to be in an amazing lab at SFU. I would like to thank all our current and former lab members Oliver Snow, Raquel Aoki, Shuman Peng, Jialin Lu, Sahand Khakabi, Mehrdad Mansouri, Qingyuan Feng, Beidou Wang, Xin Wang, Ali Arab, Atia Hamidizadeh, Arash Khoeini, Parsa Alamzdeh Harjani, Mark Lee, and Lai Wei.

I was also very fortunate to work at the Vancouver Prostate Centre during my PhD and learn from our amazing lab members Anne Haegert, Raunak Shrestha, Yen-Yi Lin, Shawn Anderson, Robert Bell, Stanislav Volik, Funda Sar, and Stephane Le Bihan.

During my PhD I also collaborated with the Princess Margaret Cancer Centre. I would like to sincerely thank Benjamin Haibe-Kains for hosting me. I learned a lot from Benjamin, including how to be a more responsible researcher. I also would like to thank his team Petr Smirnov, Anthony Mammoliti, Sisira Kadambat Nair, Soheil Jahangiri-Tazehkand, and Nikta Feizi.

There were also others who helped me in different ways during my PhD and I am very grateful for the support I received from Leonid Chindelevitch, Amina Zoubeidi, Seagle Liu,

Nicholas Erho, Mohammed Alshalalfa, Elai Davicioni, Baraa Orabi, Ehsan Haghshenas, Salem Malikic, Alexander Gawronski, Nazanin Mehrasa, Saeed Izadi, Reza Mir Asgarshahi, Vahid Reza Asadi, Piotr Bialecki, and Soufiane Mourragui.

Finally, I would like to sincerely thank my dear friends who were always there for me. They are the kindest I can wish for and I am eternally grateful for having them. Thank YOU Hossein Asghari, Ashkan Alinejad, Kiana Mostaghasi, Niloofar Kazemi, Vahid Zahednejad, Mahsa Keramati, Payam Jome Yazdani, Louise Boyl, Neda Kianifar, Anne-Marie Therien-Daniel, and Amir Mohammad Mahmoudian.

I want to THANK my loving family especially my sisters Hamide and Saba, my amazing partner Ellie and her family, and Juju (also known as the tiny dog who slept by my side during most of the writing).

At the end, I did it because of you mom, and you were here with me all the time, helping me to move forward during all of our video calls.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

After the completion of the first draft of a human genome sequence in 2003, clinical fellows have anticipated a data-driven transformation in medicine and healthcare. This transformation, now recognized as precision medicine, provides better diagnoses, more rational treatment, and early prevention of disease. Precision medicine promises improved health outcomes by providing the right treatment for the right patient, at the right time, in the right dosage via taking into account individual variability in genes, environment, and lifestyle. From the scientific and bio/medical point of view, precision medicine has transformed healthcare for the past decades and will continue to have noticeable impacts in the coming decade as it expands through numerous key areas including acquiring huge and interpretable longitudinal cohorts, gathering data and employing artificial intelligence (AI), utilizing routine clinical genomics, phenomics and environment testing, and eventually returning values (e.g., omics data) across diverse populations [1].

From an industrial point of view, in the current era of precision medicine fortified with digital technologies such as AI, drug discovery and development face tremendous opportunities for product and business model innovation. This means fundamentally changing the traditional approaches to drug discovery, development, and marketing. The pharmaceutical industry requires adoption of these new technologies in the drug development process, meaning transition from traditional approaches to a data-driven medicine. Such a paradigm shift needs translation and precision, leading to a modern transnational precision medicine approach to drug discovery and development. Transnational precision medicine consists of key areas such as multi-omics profiling of patients, digital biomarker discovery, and model-based data integration and AI [2]. Therefore, it is not far-fetched to imagine that the advancement of AI (or machine learning) is having a significant impact on precision medicine.

The use of machine learning, in particular the deep neural networks (DNNs) field, has been enabled by the availability of big data, as well as enhanced computing power and cloud storage, across all sectors including industrial and academic. In medicine DNNs have positive impacts at three levels, clinicians via rapid and accurate interpretation of the data; health systems via improving workflow and potentially reducing medical errors; and finally patients, via providing them with the opportunity to process their own data to promote health [3].

For a disease like cancer which is the main cause of death worldwide, the combination of precision medicine by interrogating multi-omics data and computational analysis via machine learning, e.g. DNNs, has emerged the field of precision oncology [4]. Precision oncology is the use of omics data to tailor therapy for an individual cancer patient. However, response to a cancer treatment—chemotherapy or targeted drugs—is a complex phenotype and often depends on multiple factors especially the omics profile of the patient [5]. Presently only 5% of patients benefit from precision oncology [6, 7, 8]. Although there are many reasons underlying this modest success rate, improved drug response prediction will significantly increase the number of patients who benefit from targeted therapy [8] or chemotherapy, and avoid adverse side effects [9, 10].

Various *in vitro* or pre-clinical studies of cancer cell lines and Patient-Derived Xenograft animal models (PDX) [11] have created datasets such as Genomics of Drug Sensitivity in Cancer (GDSCv1 and GDSCv2) [12, 13], The Genentech Cell Line Screening Initiative (gCSI) [14, 15], The Cancer Therapeutics Response Portal (CTRPv2) [16, 17], Cancer Cell Line Encyclopedia (CCLE) [18], and PDX Encyclopedia (PDXE) [11]. These datasets often provide researchers with multi-omics profiles – consisting of genomic (somatic mutation and Copy Number Aberration or CNA), transcriptomic, proteomic, and methylomic data – together with the response to a large number of targeted and chemotherapy drugs compared to clinical trial datasets. This is different from patient datasets, which record the response only to one or a few drugs that have been administered to a patient. These pre-clinical datasets enable researchers to investigate response to a drug at a large scale, in particular for many drugs, and all the way from various types of pre-clinical models to patients [13, 18]. Complementing pre-clinical studies, *in silico* or computational studies have aimed at building computational methods that analyze the cumulative effects of single- or multi-omics data to accurately predict drug response [19, 20]. These studies usually measure the drug response as the drug concentration that reduces viability by 50% (IC50) or the Area Above/Under dose-response Curve (AAC/AUC) [21].

There are three main questions this thesis aims to answer:

1. Multi-omics data promise better characterization of complex biological processes, the question is how can we integrate different omics data types to make more accurate drug response predictions from cell lines to patients?

2. Training a computational model on cell lines and testing it on patients violates the assumption that train and test data are from the same distribution, the question is how can we use both cell line and patient datasets together to build a better model for patients?

3. The Cancer Genome Atlas (TCGA) has provided researchers with a lot of clinical data without the drug response outcome, the question is how can we utilize resources like TCGA along with cell line datasets to alleviate the need for the valuable clinical data with drug response during training?

The goal of this thesis is to answer these questions by employing DNNs [22] which have demonstrated state-of-the-art performance in different problems, ranging from computer vision and natural language processing to genomics [23] and medicine [3].

1.1 Multi-omics integration in drug response prediction

A critical challenge in drug response research is the clinical utility, i.e. whether the outcome of the study is translatable to actual patients [19, 24]. Ideally to achieve translatability, a computational method should be trained on *in vivo* data, however available *in vivo* datasets such as TCGA datasets [25] do not have enough patient records with drug response information and in particular, unlike cell line datasets such as GDSCv1, they do not report responses to multiple drugs. For *in silico* drug response prediction, translatability in the simplest case can be interpreted as a model with good performance (e.g., high prediction accuracy) on *in vitro* data, trained on more samples compared to *in vivo* data, and should also have good performance on *in vivo* data.

The majority of studies suggest that gene expression data is the most effective data type for drug response prediction [13, 19, 26, 27]. Geeleher et al. [19] showed that a ridge regression model trained on GDCSv1 gene expression data is translatable to Docetaxel, Cisplatin, Erlotinib, and Bortezomib clinical trial data. They also showed that, for Docetaxel, including non-breast cancer cell lines in model training increased the predictive power of the final model compared to the model only trained on breast cancer cell lines. This ridge regression-based pipeline on gene expression also imputed the drug response for The Cancer TCGA [25, 24]. Despite the predictive power of gene expression, adding other omics data types can further increase the predictive power especially in pan-cancer models [13, 28].

Multi-omics data provide a machine learning model with different views of the same sample and promise better characterization of biological processes [29, 30]. Multi-omics data have been exploited for different problems such as driver gene identification [31, 32, 33, 34], patient stratification [35], survival prediction [36], subgroup discovery [37], and drug response prediction [20]. For the drug response prediction, Ding et al. [20] proposed a method that concatenates mutation, CNA, and gene expression data and applies autoencoders to learn features for the concatenated multi-omics cell line data. The learned features were used as the input of an elastic net classifier which predicts the binarized IC50 values. We note that the classifier was validated only on CCLE cell lines without studying its translatability to patients or PDX models.

A critical challenge in multi-omics data analysis is how to integrate different data types. There are two major approaches to multi-omics integration:

1. Early integration [38, 39]

2. Late integration [38, 39]

In early integration, all omics data types available for a sample are first concatenated, and then an integrated representation of the sample is created by applying some feature learning method, such as autoencoders [22], to that representation. Early integration has three disadvantages. First, it disregards the unique distribution of each omics data type. Second, it requires proper normalization to avoid giving more weight to the omics data type with more dimensions. Third, it further increases the dimensionality of the input data which often is already a challenge for single-omics input data [38]. In late integration, features are learned separately for each omics data type, and these features are then integrated into one unified representation to be used as the input for a classifier or a regressor. The advantage of this approach is that it works with the unique distribution of each omics data type, it can employ single-omics normalization for each data type, and it does not increase the dimensionality of the input space. However, there is no late integration method based on deep neural networks to predict drug response and a need exists to develop a method for this problem.

1.2 Transfer learning in drug response prediction

In our driving application, drug response prediction [21], the goal is to predict response to a cancer drug given the gene expression data or other omics data types. Since clinical datasets in pharmacogenomics (patients) are small and hard to obtain, many studies have focused on large pre-clinical pharmacogenomics datasets such as cancer cell lines as a proxy to patients [18, 13]. A majority of the current methods are trained on cell line datasets and then tested on other cell line or patient datasets [40, 19]. However, cell lines and patients data, even with the same set of genes, do not have identical distributions due to the lack of an immune system and the tumor microenvironment in cell lines [41]. Moreover, in cell lines, the response is often measured by the IC50 or AAC, whereas in patients, it is often based on changes in the size of the tumor and measured by metrics such as response evaluation criteria in solid tumors (RECIST) [42]. This means that drug response prediction is a regression problem in cell lines but a classification problem in patients. Therefore, discrepancies exist in both the input and output space in pharmacogenomics datasets and a need exists for a computational method to bridge this gap.

DNNs often require a large number of samples for training, which is challenging and sometimes impossible to obtain in the real world applications. Therefore, many studies have employed transfer learning [43] to bridge the gap between *relevant* large and small datasets and use them together to achieve a better performance on the small dataset.

Transfer learning [43, 44, 45] attempts to solve this challenge by leveraging the knowledge in a *source* domain, a large data-rich dataset, to improve the generalization performance on a small *target* domain. Training a model on the source domain and testing it on the target domain violates the Independent and identically distributed (i.i.d) assumption that the train and test data are from the same distribution, which is similar to the cell line and patient datasets challenges. The discrepancy in the input space decreases the prediction accuracy on the test data, which leads to poor generalization [46]. Many methods have been proposed to minimize the discrepancy between the source and the target domains using different metrics such as Jensen Shannon Divergence [47], Maximum Mean Discrepancy [48], and correlation alignment for deep domain adaptation (CORAL) loss [49]. While transductive transfer learning (e.g. domain adaptation) uses a labeled source domain to improve generalization on an unlabeled target domain, inductive transfer learning (e.g. few-shot learning) uses a labeled source domain to improve the generalization on a labeled target domain where label spaces are different in the source and the target domains [50].

Adversarial domain adaptation has shown great performance in addressing the discrepancy in the input space for different applications [51, 52, 53, 54, 55, 56, 57, 58], however, adversarial adaptation to address the discrepancies in both the input and output spaces has not yet been explored which indicates that available methods cannot address the unique challenges in drug response prediction from the transfer learning point of view and a need exists to develop a new method.

1.3 Domain generalization in drug response prediction

Various methods of transfer learning have been proposed in the context of drug response prediction. These methods either address existing discrepancies implicitly [40, 59, 60], or explicitly which means they assume that the model has access to the target domain during training or fine-tuning [61, 41, 62, 63, 64, 65]. However, in the real-world we do not have access to the target domain(s) while training the model on the source domain, e.g., we do not know future patients that may walk into a clinic. Nevertheless, the trained model should generalize to the target domain and be able to make predictions for samples encountered during the deployment time. Since generating large high-quality labeled pre-clinical datasets is an expensive and time-consuming process and we do not know response to a given drug in the target domain (e.g., future patients), there is a need for a computational method that takes not only labeled but also unlabeled source domain data as input and learns a representation that generalizes to a future target domain. This problem is known as out-of-distribution generalization or domain generalization, where the target domain is not accessible during training [66, 67, 68]. Out-of-distribution generalization is particularly important for biomedical applications [69].

Domain generalization aims at learning an invariant representation given input data from a single or multiple domains. However, the main difference is that in domain generalization the target domain is not available during training. This is a much harder scenario compared to domain adaptation or inductive transfer learning for which the target domain is available during training [64, 67]. A domain generalization method should extract invariant representations only using source domains. This is highly important because it is very similar to the real world for which no information is available about unseen data. For example, in medical imaging, different hospitals with different equipment and patients can be separate domains and domain generalization aims at making accurate predictions for unseen hospitals and/or patients [70]. There are two main approaches to out-of-distribution generalization:

1. Generalizing via learning domain-invariant features [67]

2. Generalizing via learning hypothesis-invariant features [71, 72]

In domain-invariant, the most common approach, the goal is to map the input domains to a shared feature space in which the features of all domains are aligned, i.e. look similar to each other. [70, 73, 74, 75, 76]. However, forcing different domains to have very similar features is not always feasible because different domains may have unique characteristics, and completely aligning them ignores these unique characteristics. The second approach does not align the features but rather the predictions across domains. The idea is that if the extracted features of input domains are similar enough for an accurate predictor to make similar predictions, forcing the features to be more similar is not required anymore. [71, 67]. However, a recent benchmark study demonstrated that simple Empirical Risk Minimization (ERM) methods outperform state-of-the-art methods of domain generalization [66]. ERM methods employ simple standard supervised loss functions and are trained on all of the available source domains.

In drug response prediction, given some pre-clinical or clinical datasets from different domains as input, a method of domain generalization should learn an invariant representation capable of making predictions for unseen clinical and pre-clinical datasets as output. The advantage of domain generalization is that it does not need valuable but limited patient data with drug response available to learn such representations. The input can consists of both pre-clinical and clinical resources where for the former labeled cell line datasets are available and for the latter, large unlabeled resources such as TCGA are available. The main advantage of using resources like TCGA is that they are much larger compared to clinical trial datasets and therefore more suitable for representation learning. However, the disadvantage is that there is no drug response information available for the majority of the TCGA patients. This poses an extra challenge on domain generalization because state-of-the-art methods of this area are not designed to take unlabeled data as input. Similarly, ERM methods that demonstrated competitive performance for out-of-distribution generalization cannot take unlabeled samples as input. A recent study aimed at tackling semi-supervised domain generalization [77], however, the proposed method is only applicable for classification problems while drug response prediction can be both regression and classification. Therefore, there is a need for a semi-supervised domain generalization method that takes both labeled and unlabeled samples from different domains and learns an invariant representation with generalization capability to unseen target domains.

Chapter 2

Background and Related Work

The general assumption in traditional machine learning models is that train and test data are from the same distribution. However, this is not a valid assumption in many real-world problems including precision oncology. In precision oncology, pre-clinical resources such as cell lines do not have tumor microenvironment and/or an immune system. Therefore, they are from a different distribution than patients. So, how can we use both large pre-clinical and small clinical, i.e. patient, datasets together to train a more accurate model for patients?

Transfer learning attempts to answer this question by leveraging the knowledge in a large data-rich resource, *source* domain, to improve the prediction performance on a small dataset that we are interested in, *target* domain. For example, in precision oncology pre-clinical data is the source domain and patient data is the target domain. The reason that transfer learning matters in precision oncology is that not only patient datasets are small, but also they are high-dimensional which poses an extra challenge on model development.

There are three questions in transfer learning [50]:

1- When to use transfer learning?
2- What to transfer between source and target domain?
3- How to transfer *knowledge* between source and target domain?

Transfer learning should happen when source and target domain are *relevant*. Although there is no formal definition for two relevant domains, a domain expert knowledge can be utilized to select related source and target domain. In precision oncology, cell lines and PDX resources are related source domains for patient data as a target domain. The goal of the first question is to avoid negative transfer which not only does not improve the generalization performance, but also decreases it.

After figuring out whether or not transfer learning is going to be useful, it is important to decide what to transfer between a source domain and a target domain. The reason is that some knowledge might be domain-specific and some knowledge might be domain-invariant and more suitable to be transferred. There are four types of knowledge that can be transferred between source and target domains:

1. Instance-transfer (sample-transfer)

2. Parameter-transfer

3. Feature-representation-transfer

4. Relational-knowledge-transfer

In instance-transfer certain samples in the source domain which are relevant to the target domain are transferred. In parameter-transfer certain trained parameters from a model trained on the source domain are transferred to another model to be trained on the target domain. In feature-representation-transfer certain knowledge encoded in learned feature representation is transferred between these domains. In relational-knowledge-transfer certain relational knowledge in the source domain is being employed to learn similar relational knowledge for the target domain. Finally, after knowing when to use transfer learning and what to transfer, the question is how to actually perform the transfer which will be the main focus of this report.
Generally, methods of transfer learning can be categorized into three categories:

1. Unsupervised transfer learning

2. Transductive transfer learning

3. Inductive transfer learning.

Before defining these categories, it is important to define transfer learning in a more formal way. Following the notation of [50], a domain like DM is defined by a raw input feature space[1] $\mathbb{X}$, a probability distribution $p(X)$ and a corresponding dataset $X = \{x_1, x_2, ..., x_n\}$ with $x_i \in \mathbb{X}$. A task $\mathbb{T} = \{Y, \mathbb{F}(.)\}$ is associated with $DM = \{X, p(X)\}$ and is defined by a label space $Y \in \mathbb{Y}$ and a predictive function $\mathbb{F}(.)$ which is learned from training data $(X, Y) \in \mathbb{X} \times \mathbb{Y}$. A source domain is defined as $DM_S = \{(x_{s_1}, y_{s_1}), (x_{s_2}, y_{s_2}), ..., (x_{s_{n_S}}, y_{s_{n_S}})\}$ and a target domain is defined as $DM_T = \{(x_{t_1}, y_{t_1}), (x_{t_2}, y_{t_2}), ..., (x_{t_{n_T}}, y_{t_{n_T}})\}$, where $x_s \in X_S$, $x_t \in X_T$, $y_s \in Y_S$, and $y_t \in Y_T$. Since $n_T << n_S$ and it is challenging to train a model only on the target domain, transfer learning aims to improve the generalization on a target task $\mathbb{T}_T$ using the knowledge in DM_S and DM_T and their corresponding tasks $\mathbb{T}_S$ and $\mathbb{T}_T$. In unsupervised transfer learning, there is no label in the source or target domain. In transductive transfer learning, source domain is labeled but target domain is unlabeled, domains can be either the same or different (domain adaptation), but source and target tasks are the same. In inductive transfer learning, target domain is labeled and source domain can be either labeled or unlabeled and domains can be the same or different, but in this category tasks are always different [50].

[1] This is different from features learned by a deep neural network

It is been known that in DNNs, first-layer features are general and last-layer features are specific towards the objective of the network. Yosinski et al. [78] quantified this transition from general to specific features in image classification. They showed that initial layers of DNNs capture more general and invariant features than the last layers and this transition happens in middle layers. To show that, they trained two 8-layers DNNs, denoted by A and B, respectively, on half of ImageNet dataset (1000 classes per sample which was splitted into two 500 class labels). After training A and B, to study how transferable these features are, they copied the first $n \in \{1, 2, .., 7\}$ layers (trained parameters) according to two scenarios: 1) a *Selfer* scenario where n layers are transferred from A to itself (or B to itself), and 2) a *Transfer* scenario, where n layers are transferred from A to B. The transferred layers were fine-tuned or kept frozen in each scenario and the rest of $n - 8$ layers were initialized randomly. Using these scenarios, the authors observed that first, the *Transfer* scenario along with fine-tuning improves generalization while only transferring decreases the performance due to moving from general to specific features as n increases. Second, Performance drops due to fragile co-adaptation between nodes in two consecutive layers (mostly in the last layers), however, fine-tuning decreases these co-adapted interactions. Finally, Performance drops due to representation specificity and this transition occurs in middle layers. Random split of the ImageNet dataset for A and B makes the domains similar. To study the impact of dissimilar domains, ImageNet was splitted into natural and man-made images. As expected, the performance decreased more compared to the previous experiments with similar domains. This indicates the need for at least fine-tuning or a much more sophisticated approach of transfer learning.

Back to the three main questions in transfer learning (when? what? how?), this study used transfer learning in two similar and dissimilar situations, they transferred the trained parameters, and applied fine-tuning as a way to adapt the source and target domains. This study established the foundation of transfer learning in DNNs for image classification and many other papers used its results to employ pre-trained DNNs and then fine-tune the last layers of these networks towards their desired objectives.

While Yosinski et al. [78] showed that fine-tuning on the target domain improves the prediction performance, another recent study [79] suggests that the learned parameters on the source domain should act as both a starting point for the target domain and also as a reference to avoid deviating too much from them in the fine-tuning process. This means that regularization and fine-tuning is better to happen together and this study investigated different regularizations to reduce the gap between the source domain learned parameters and those being fine-tuned on the target domain. Obtained results showed a better prediction performance compared to fine-tuning and regularizing parameters (to shrike their values) without considering the initial learned values on the source domain.

2.1 Transductive transfer learning

Transductive transfer learning aims to improve the performance of the target task using the knowledge in the source domain and source task, where source and target tasks are the same but the target domain is unlabeled [50]. This problem is also known as unsupervised domain adaptation. It is unsupervised because of the lack of labels in the target domain and it needs adaptation because source and target domains are from different distributions. Early works in this area used different metrics such as Maximum Mean Discrepancy (MMD) to minimize the discrepancy between source and target domain. Later these metrics were incorporated into Deep Neural Networks (DNNs) to learn features which are both domain-invariant and predictive of the class labels in the source domain. Recently, domain adaptation methods based on adversarial learning showed better performance in different problems [47]. The minimax objective function of adversarial learning closely resembles domain adaptation because in domain adaptation we want to minimize the discrepancy between domains and at the same time maximizing the performance on the source domain. Most of the available methods adapt a single source domain and a single target domain. However, recent methods have been proposed to adapt multiple domains (multiple source domains or multiple target domains) [80, 58]. This section presents state-of-the-art of single and multiple domain adaptation. Table 2.1 summarizes the methods of this category.

2.1.1 Single domain

Given a source domain $D_s = \{(x_i^s, y_i^s)\}_{i=1}^{n_s}$ with n_s labeled samples and a target domain $D_t = \{(x_i^t)\}_{i=1}^{n_t}$ with n_t unlabeled samples, the goal is to minimize the error on the target domain via jointly minimizing the source domain error and the domain discrepancy between the source and target domain.

Previous work attempt to solve this problem by designing a two part objective function: 1) the first part is a task-specific loss (for example classification loss) on the source domain and the second part is a loss related to the measure of discrepancy between the source and the target. For example, Tzeng et al.[81], proposed to use MMD as the domain loss. In their method, a convolutional DNN, denoted by $f(x)$, is trained on the source domain with a classification loss which shares its parameters with the target domain samples. The features extracted by this backbone for the input domains are further regularized by the MMD which is the difference between means of extracted features for the source and target samples. A similar approach was employed by the Correlation Alignment for Deep Domain Adaptation (CORAL) loss to align the covariance matrices of the extracted features of the source domain and the target domain. CORAL aligns both the mean and the correlation of the extracted features [49].

2.1.2 Adversarial learning

Adversarial domain adaptation has shown great performance in addressing the discrepancy in the input space for different applications [51, 52, 53, 54, 55, 56, 57, 58] and showed better performance compared to discrepancy metrics such as MMD. Adversarial domain adaptation is achieved by recent advances in Generative Adversarial Networks (GANs). GANs [82] attempt to learn the distribution of the input *data* via a minimax framework where two networks are competing: a discriminator D and a generator G. The generator tries to create fake samples from a randomly sampled latent variable that fool the discriminator, while the discriminator tries to catch these fake samples and discriminate them from the real ones. Therefore, the generator wants to minimize its error, while the discriminator wants to maximize its accuracy:

$$\underset{G}{M}in\underset{D}{M}axV(G,D) = \sum_{x \sim data} log[D(x)] + \sum_{z \sim noise} log[1 - D(G(z))] \tag{2.1}$$

Various methods have been proposed for adversarial domain adaptation in different applications such as image segmentation [57, 55], image classification [58, 56], speech recognition [52], domain adaptation under label-shift [83], partial domain adaptation [84]. The idea of these methods is that features extracted from source and target samples should be similar enough to fool a domain discriminator [58] and/or class-wise discriminators [57].

Tzeng et al. [58], proposed Adversarial Discriminative Domain Adaptation (ADDA) for this problem. ADDA has three steps: 1) a feature extractor, denoted by $f_s(x)$, and a classifier, denoted by C, are trained on the source domain as follows:

$$L_{cls} = -\sum_{x_s \sim X_s} \sum_{k=1}^{K} \mathbf{1}(k == \overline{y}) log[C(f_s(x_s))], \tag{2.2}$$

where L_{cls} is the classification loss on the source domain.
2) With a frozen source domain feature extractor backbone, another feature extractor, denoted by $f_t(x)$, is trained on the target domain using adversarial learning. In this step, a domain discriminator is trained to learn domain-invariant features by training the target domain feature extractor to learn features close enough to those of the source domain feature extractor to fool the domain discriminator. The objective function of this step to train a domain discriminator D is as follows:

$$L_{adv_D} = -\sum_{x_s \sim X_s} log[D(f_s(x_s)] - \sum_{x_t \sim X_t} log[1 - D(f_t(x_t))], \tag{2.3}$$

and the objective function to train $f_t(x)$ is as follows:

$$L_{adv_T} = -\sum_{x_t \sim X_t} log[D(f_t(x_t))]. \tag{2.4}$$

Finally, 3) in the last step, the trained target domain feature extractor and the trained source domain classifier are utilized to make prediction for the target domain samples.
In addition to a domain discriminator (also known as global discriminator), other studies employed global and class-wise discriminators to learn domain-invariant features. The goal is to learn these features with respect to specific class labels such that they fool corresponding class-wise discriminators. A class-wise discriminator receives source and target samples from the same class label and should not be able to predict the domain accurately [57, 85]. But the challenge for this approach is that class labels are not available in the target domain to assign target samples to their corresponding discriminators. To tackle this challenge, one solution is to use predicted labels for the target samples provided by the classifier trained on the source domain [57]. This approach sends target samples to their corresponding discriminators but the drawback is that predictions can be uncertain. To address this, Another approach, named Multi-Adversarial Domain Adaptation (MADA), used target samples as the input of all of the class-wise discriminators but weighted their importance by the probability of belonging to that class [85]. This is achievable because the source domain classifier assigns a probability distribution over all of the classes to each target domain sample. The general form of the objective function for these methods is as follows:

$$J = L_{cls} + \lambda \sum_{i} L_{adv_{D_i}}, \tag{2.5}$$

where, $L_{adv_{D_i}}$ is the adversarial loss for the class-wise discriminator i.
Although these methods showed that adaptation based on the output of the feature extractor backbone is a reliable approach, Tsai et al. [55] claimed that one level adaptation may not adapt lower level features particularly in complex tasks such as image segmentation. In their proposed method, the objective function is similar to the previous work: one task-specific loss, in this case image segmentation, and a domain discriminator loss. However, the difference is that they applied this objective function to multiple layers and showed better performance compared to single layer adaptation methods.

Leveraging GANs with cycle consistency constraint, which enforces accurate reconstruction of mapping of source domain samples to the target domain and then back to the source domain and vice versa, Hosseini et al [52] proposed Augmented cycle-GAN (ACAL) which replaces the reconstruction loss with task-specific losses on mapped samples from source to target domain and then back to the source domain and vice versa. This method can work with both labeled and unlabeled target domain. The general idea is that it first pre-trains source classifier on the source domain data. Then, the source model is fine-tuned on the source data and the target data mapped to the source domain. Similarly, the target model is also trained on the target data and the mapped source samples to the target domain. The source and target discriminators should not be able to accurately discriminate a true target/source sample from a mapped one and feature extractors should learn domain-invariant features.

2.1.3 Multiple domains

In this category, we are given multiple domains either as multiple source domains or as multiple target domains. In the first scenario, given N source domains $D_s = \{D_1^s, D_2^s, ..., D_N^s\}$ with collections of *i.i.d* labeled samples for each source domain and a target domain $D_t = \{(x_i^t)\}_{i=1}^{n_t}$ with n_t unlabeled samples, the goal is to minimize the error on the target domain via jointly minimizing the source domain errors and the domain discrepancy between the source domains and target domain. In the second scenario, given a source domain $D_s = \{(x_i^s, y_i^s)\}_{i=1}^{n_s}$ with n_s labeled samples and N target domains $D_t = \{D_1^t, D_2^t, ..., D_N^t\}$ with collections of *i.i.d* unlabeled samples for each target domain, the goal is to minimize the error on the target domains via jointly minimizing the source domain error and the domain discrepancy between the source domain and target domains.

2.1.4 Multiple source domains

Generally the first scenario, i.e, having multiple source domains, is more common. Deep Cocktail Network (DCTN) [86] used the weighted combination of the source domains to achieve a better performance in the target domain. Similar to the previous single domain methods, DCTN employs a feature extractor, denoted by $f(x)$, to learn a representation for the source domains and the target domain. It also has N domain-specific classifiers to be trained on their corresponding source domains. The extracted features of the source and target domains go to N domain discriminators such that $D_j(f(x))$ receives samples from the $j-th$ source domain and the target domain and should not be able to discriminate them. The learned features of the target samples are input to the N classifiers. Since the labels in the target domain are not available, DCTN utilizes a perplexity score, denoted by $S(x_t; f, D_j)$, base on the loss of a target sample x_t in a domain discriminator D_j to weight the classification prediction and determine high confidence predictions as follows:

$$S(x_t; f, D_j) = -log[1 - D_j(f(x_t))] + \alpha_j, \tag{2.6}$$

where α_j is obtained by averaging the performance of D_j on the source domain D_{s_j}. This constant shows how good this discriminator is in general.

The challenge is that source domains can also have different feature distributions among themselves. Therefore, to achieve a better performance on the target domain, a method should minimize the discrepancy between source domains and target domain and also between pairs of source domains. To address this issue, Peng et al. [87], adapted the distribution of the target domain and multiple source domains using the first and the second moment of the extracted features – the first moment is the mean and the second moment is the mean of the square of each samples in the corresponding domain. In addition, they used the same moments to adapt pairs of source domains as well. The Moment Distance (MD) loss is as follows:

$$MD = \sum_{k=1}^{2} \left(1/N \sum_{i=1}^{N} ||\mathbb{E}(D_{s_i}^k) - \mathbb{E}(D_t^k)||_2 + \binom{N}{2}^{-1} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} ||\mathbb{E}(D_{s_i}^k) - \mathbb{E}(D_{s_j}^k)||_2 \right), \tag{2.7}$$

other than the MD loss, the objective function of this method also has N classification loss terms obtained from the source domains. For classifying target samples, they proposed two approaches: 1) based on the average of the predictions from the classifiers, and 2) based on a weighted average obtained by the normalized accuracy of each classifier on the corresponding source domain. This method showed better performance compared to DCTN. But what if the input domains are not all labeled and they have overlapping but distinct labels? meaning that a fraction of unlabeled samples comes from extra classes, i.e. classes with no labeled samples within that domain.

MULAAN (Multi-domain Learning Adversarial Neural Network) [88] addressed this problem by introducing Known Unknown Discriminator (KUD) modules. Given N labeled input domains from C classes where N' of them have unlabeled samples for a subset of the classes, MULAAN uses a feature extractor backbone to learn features for these domains, then it employs a classifier to make prediction for C classes. Furthermore, it also utilizes a global domain discriminator to discriminate the features learned for these N domains. The classifier assigns labels to the N' domains with missing labels. These samples are further ranked based on the entropy of their classification results. A KUD module receives the top $p\%$ of these unlabeled samples with label ”0” meaning they are most likely unknown and also receives labeled samples from that domain with label ”1” meaning they are known and should be able to predict known/unknown status accurately. A KUD module is essentially a binary cross-entropy loss. Therefore, the objective function of MULAAN has three terms: a classification loss, an adversarial loss of the global domain discriminator, and the KUD losses. The driving application of this method is automated microscopy images of cell after being exposed to known and unknown drugs, where each dataset has its own experimental bias.

2.1.5 Multiple target domains

Moving to a single source domain and multiple target domains, the challenge is that the target data has mixed domains and class-irrelevant features lead to negative transfer [50], especially when the target domain is highly mixed. DADA (Deep Adversarial Disentangled Autoencoder) [80] employs adversarial learning and Variational Autoencoder (VAE) [89] to disentangle domain-invariant features from class-irrelevant and domain-specific features. DADA first uses a feature extractor to learn features for the source domain and the target domains. Then a Disentangler module extracts domain-invariant, class-irrelevant, and domain-specific features from the feature generator. The features learned by the Disentangler should be

Table 2.1: Overview of methods of transductive transfer learning

Method	When?	What?	How?	#domain	dataset
MMD [81]	image classification	features	MMD loss	single	Office-31
CORAL [49]	image classification	features	CORAL loss	single	Office-31
GRL [47]	digit classification	features	gradient reversal	single	MNIST, SVHN, GTSRB
ADDA [58]	digit classification	features	adversarial learning	single	MNIST, USPS, SVHN
Chen et al. [57]	image segmentation	features	class-wise adversarial learning	single	Cityscapes, SYNTHIA
MADA [85]	image classification	features	weighted class-wise adversarial learning	single	Office-31, ImageCLEF-DA
Tsai et al. [55]	image segmentation	features	adversarial learning	single	Cityscapes, SYNTHIA, GTA5
ACAL [52]	digit classification	features	augmented cycle GAN	single	MNIST, USPS, SVHN
DCTN [86]	image classification	features	weighted class-wise adversarial learning	multi	Office-31, ImageCLEF-DA
M3SDA [87]	image classification	features	matching first and second moments	multi	DomainNet
MULAAN [88]	cell microscopy image	features	adversarial learning	multi	CELL dataset
DADA [80]	image classification	features	VAE and adversarial learning	multi	DIGIT5, Office-10, DomainNet

rich enough for a Reconstructor to reconstruct the original features learned by the feature extractor. Disentangler-Reconstructor are the encoder-decoder of a VAE. Furthermore, to ensure disentanglement, DADA minimizes the mutual information of domain-invariant and class-irrelevant as well as domain-invariant and domain-specific features. Adversarial learning adapts the source domain and the target domains in the domain-invariant feature space. Moreover, a classifier C is trained on the labeled source domain to predict the class labels and also to adversarially extract class-irrelevant features.

2.1.6 Summary

Back to the original transfer learning questions, the methods of this category employed transfer learning for variety of applications ranging from image classification and segmentation to speech analysis (when?). All of the discussed methods were based on feature-representation-transfer (what?) and most of them utilized different approaches based on adversarial learning (how?). Table 2.1 summarizes these methods. A simple implementation trick in adversarial learning to tackle minimax optimization is employing gradient reversal layer which changes the sign of gradients between discriminator and feature extractor [47]. Based on the label space of source and target domain, domain adaptation can be closed set, partial [90], open set [91], or universal [84]. In the closed set domain adaptation, the label spaces of source and target domain are identical. In partial domain adaptation, source domain has private classes to target domain. In open set domain adaptation, both source and target domains can have private class to each other. Finally, in the universal domain adaptation there is no prior knowledge on the label space of the target domain.

2.2 Inductive transfer learning

Inductive transfer learning aims to improve the performance of the target task using the target domain and its task as well as the knowledge in the source domain and source task, where these tasks are different and the target is labeled [50]. There are three approaches to inductive transfer learning: 1) deep metric learning, 2) few-shot learning, and 3) weight transfer [92]. Deep metric learning methods are independent of the number of samples in each class of the target domain, denoted by k, meaning that they work for small and large k values. Few-shot learning methods focus on small k values ($\leqslant 20$) and weight transfer methods require large k values ($k \geqslant 100$ or $k \geqslant 1000$) [92]. This section does not review the weight transfer category because it follows the same line of work as discussed in the Introduction section [78, 79].

In the inductive transfer learning, the learning process starts by randomly sampling a *support* set and a *query* set from the source domain. The goal is to mimic limited sample size in the target domain and also select different subsets of class labels to mimic different tasks. The model is trained on the support set and evaluated on the query set [93]. The target domain is similarly sampled into support and query sets. Recent evidence indicate that fine-tuning the trained model on the source domain using the support set of the target domain improves the performance [92]. The fine-tuned model is eventually tested on the query set of the target domain. Each of these support and query sets is called an *episode*. In addition to emphasizing on role of fine-tuning, Scott et al. [92] also provided a systematic comparison between these categories and suggested that solely focusing on k may not be the best approach to study inductive transfer learning. For example, they showed that few-shot learning methods can also perform well for larger values of k or metric learning methods which are known to be agnostic to k works well only for a range of k. Nonetheless, they showed that fine-tuning the train model on the support set of the target domain leads to a better performance.

Chen et al. [94] did a similar comparative study but with a specific focus on few-shot learning methods for $k = 1$ and $k = 5$. They observed that having a deeper backbone to extract features reduces the performance gap between state-of-the-art of few-shot learning. Another interesting observation of this paper was that the performance of state-of-the-art drops significantly under domain discrepancy between input domains such that even two simple baselines can achieve a competitive/better performance compared to complex state-of-the-art few-shot learning methods. These baselines had a feature extractor followed by a standard classifier trained on the source domain. Then, the feature extractors of the baselines remained frozen and a new classifier was re-trained on the target domain. For the first baseline the classifier had a linear layer with a softmax activation and for the second baseline it used the cosine distance between the input feature and the learned weight vectors representing each class to find the class label [94]. The next two sections present an overview of deep metric learning methods and few-shot learning methods. Table 2.2 summarizes the methods of this category.

2.2.1 Deep metric learning

In this category, a DNN is trained on the source domain and task with an aim of learning a representation which samples of the same class are closer to one another and those of different classes are far apart. Then, this DNN can further be fine-tuned on the target domain and its task or simply use a nearest neighbor approach based on the obtained representations. Various methods with different objective functions have been proposed to learn such representations [95, 96, 97].

For example, The triplet loss function was introduced in FaceNet [95] for learning the representation from a space of face images to a Euclidean space where the difference between learned features is correlated with the similarity among identities. The idea is that for the image of a given person's face, the distance between the learned features of the given image and those of another image with the same identity should be smaller than the distance between that image's learned features and the learned features of an image with a different identity. For T given triplets in the form of ($Anchor$, $Positive$, $Negative$), where the $Anchor$ and $Positive$ have the same class labels and $Negative$ has a different class label, and a feature extractor backbone, denoted by $f(.)$, the following condition must hold:
$d(f(Anchor_i), f(Positive_i)) \leq d(f(Anchor_i), f(Negative_i))$,
where $d(.)$ is an arbitrary distance function – the Euclidean distance was used in the original study. If we move the right hand-side to the left, we obtain:

$$d(f(Anchor_i), f(Positive_i)) - d(f(Anchor_i), f(Negative_i)) \leq 0 \tag{2.8}$$

In order to avoid the trivial zero solution, a margin $\xi > 0$ is required:

$$d(f(Anchor_i), f(Positive_i)) - d(f(Anchor_i), f(Negative_i)) + \xi \leq 0 \tag{2.9}$$

We want the distance of the $Anchor$ and the $Negative$ to be larger than the distance of the $Anchor$ and the $Positive$. Thus, the value of the triplet loss function for the $i-th$ triplet is:

$$L^i_{Triplet} = max[d(f(Anchor_i), f(Positive_i) - d(f(Anchor_i), f(Negative_i)) + \xi, 0] \tag{2.10}$$

and the total triplet loss for T triplets is:

$$L_{Triplet} = \sum_{i=1}^{T} L^i_{Triplet} \tag{2.11}$$

There are two approaches to select triplets for the triplet loss function, 1) offline selection and 2) online selection. The offline selection builds the triplets based on the class labels before training and the online approach selects the triplets from samples during the training. Triplets also can be built based on all possible combinations of the input samples/mini-batches, called soft selection, or based only on those with highest triplet loss value which is called hard selection. Soft selection provides more triplets for training but the network might rely too much on easy cases, and as a result may be unable to perform well on hard examples [95]. The hard selection approach solves this problem by only relying on the hard cases in the train data to build the triplets, however, it provides fewer triplets for training. In the inductive transfer learning experiment of FaceNet source and target domain were from the same distribution but the identities were disjoint meaning that the tasks in the source and target were different.

2.2.2 Few-shot learning

A human learns new concepts with very few samples of a new object, for example, a child can recognize an apple from a couple of pictures, however, state-of-the-art DNNs need hundreds or even thousands of samples to achieve the same goal. This observation motivates few-shot learning, meaning learning a class from a few labeled examples [98]. Few-shot learning involves training a classifier to recognize new classes, provided only a small number of examples from each of these new classes in the training data [93]. Based on the number of k labeled samples for each C classes, the problem is called C-way k-shot – zero-shot means zero samples of that class are available during training and one-shot means only one sample is available. Various methods have been proposed for few-shot learning [92, 93, 94, 97, 98]. Few-shot learning methods employ simple approaches with few parameters to avoid over-fitting because of limited data for each class in the target domain.

One of the first work in this area is the Matching Networks (MatchingNet) [98]. MatchingNet aims at defining a probability distribution over class labels for a target sample x_t given a small support set of k source domain samples. To achieve this goal, MatchingNet extract features from support and query sets via two separate networks denoted by $g(x)$ and $f(x)$, respectively. MatchingNet computes the predicted output as follows:

$$\hat{y} = \sum_{i=1}^{k} a(x_q, x_i) y_i, \tag{2.12}$$

where x_i and y_i are labeled samples from the support set and a is an attention mechanism as follows:

$$a(x_q, x_i) = \frac{exp((c\,(f(x_q), g(x_i))))}{\sum_{j=1}^{k} exp((c\,(f(x_q), g(x_j))))}, \tag{2.13}$$

where c is the cosine similarity distance. Although MatchingNet showed great performance, it was outperformed by Prototypical Networks (ProtoNet) [93]. ProtoNet constructs prototypical representatives by taking the average of the learned features for samples of the support set in each class. In other words, a prototype for class c is the mean vector of the learned representation of the support set samples belonging to class c. Therefore, a prototype P_c for class c is calculated as follows:

$$P_c = 1/n_S \sum_{x_s \sim X_S \in c} f(x_s) \tag{2.14}$$

After constructing the prototypes, ProtoNet assigns labels to the query set samples based on a softmax over the Euclidean distance, denoted by $d(.)$, between the target domain learned features and the prototypes (class representatives):

$$p(y = c|x_q) = \frac{exp(-d(f(x_q), P_c))}{\sum_{j=1}^{C} exp(-d(f(x_q), P_j))} \tag{2.15}$$

Learning the representation is obtained by minimizing the negative log-probability on the querry set as follows:

$$J(\Theta) = -log_p(y = c|x_{tq}), \tag{2.16}$$

where Θ is the set of parameters of the employed DNN, denoted by $f(x)$. In ProtoNet, training episodes (mini-batches) are constructed by random sampling of a subset of classes from the training data. A subset of the selected samples are used as the source domain to build the prototypes (support set) and the subset of the remainder samples are used as the target domain (query set). ProtoNet showed better performance on image classification [93] compared to the MatchingNet.

RelationNet [99] had a similar idea, but instead of utilizing a fixed distance metric, it employs a learnable mapping that maps the learned features of source and target domains to a relation score. A relation score $r_{i,j}$ between a source domain sample i and a target domain sample j is defined as:

$$r_{i,j} = g[f(x_i) \oplus f(x_j)] \tag{2.17}$$

where x_i is from the support set, x_j is from the query set, $r_{i,j}$ is the relation score between x_i and x_j, $\oplus$ is the concatenation operator, $f(x)$ is a feature extractor obtained by a DNN, and $g(.)$ is another DNN with the Sigmoid activation in the last layer to generate the relation score. The learning proceeds by minimizing the following objective function:

$$J(\Theta, \Phi) = \sum_{i=1}^{m} \sum_{i=j}^{n} (r_{i,j} - \mathbf{1}(y_i == y_j))^2 \tag{2.18}$$

where Θ and Φ are parameters of $f(.)$ and $g(.)$, respectively, m and n are number of samples in the support and query sets, respectively, and $\mathbf{1}$ denotes an indicator. RelationNet showed better performance compared to ProtoNet in image classification.

Table 2.2: Overview of methods of inductive transfer learning

Method	When?	What?	How?	#domain	dataset
FaceNet [95]	image recognition	features	triplet loss function	single	Labeled Faces in the Wild, Youtube Faces DB
HistLoss [96]	image classification	features	histogram loss	single	CUB200-2011, CUHK03, Market-1501, Online Products
Siamese Net [97]	image recognition	features	one-shot learning by the contrastive loss	single	MNIST, Omniglot
MatchingNet [98]	image classification	features	few-shot learning by the cosine distance	single	Omniglot, mini-ImageNet
ProtoNet [93]	image classification	features	few-shot learning by the Euclidean distance	single	Omniglot, mini-ImageNet
RelationNet [99]	image classification	features	few-shot learning by calculating relation score	single	Omniglot, mini-ImageNet
Scott et al. [92]	image classification	features	comparative study of methods of inductive transfer learning	single	MNIST, Omniglot, mini-ImageNet, Isolet
Chen et al. [94]	image classification	features	comparative study of methods of few-shot learning	single	CUB-200-2011, mini-ImageNet

2.2.3 Summary

Back to the original transfer learning questions, the methods of this category employed transfer learning mostly for image classification (when?). All of the discussed methods were based on feature-representation-transfer (what?) and they used metric learning or few-shot learning to learn features (how?). Table 2.2 summarizes these methods. Other areas that are highly entangled with inductive transfer learning are meta-learning, zero-shot, and one-shot learning but this thesis did not study these areas and other applications such as Natural Language Processing.

2.3 Out-of-Distribution Generalization

Out-of-distribution generalization or domain generalization attempts to learn an invariant representation given input data from multiple domains [73]. However, unlike domain adaptation, in domain generalization the target domain is not available during training. This is a much harder scenario compared to domain adaptation where the target domain is available during training [70]. A domain generalization method should extract a domain-invariant representation only using source domains. Learning such a representation is important because it is similar to most real-world tasks for which no information is available about the unseen data (the target domain) [100, 101, 102, 103, 70, 104, 105, 73, 74, 75, 76]. If we focus on the label space of source domain and the unseen target domain, domain generalization can be categorized into two types: homogeneous and heterogeneous. In the homogeneous category, a shared label space exists between source domains and unseen target domains [103, 70, 105, 73, 74, 75, 76], however, in the heterogeneous category these label spaces are disjointed [102, 104].

For example, Dou et al. [70] proposed model-agnostic learning of semantic features (MASF), a method based on meta learning to perform global and local alignment between domains in the homogeneous setting. This method used a class-specific mean and Kullback–Leibler (KL) divergence in the global alignment step and a triplet loss for the local alignment between input domains. The role of meta learning is to utilize episodic training to generalize better under domain shifts. Tseng et al. [102] proposed a model agnostic feature-wise transformation layer to enforce learning more diverse features and to avoid over-fitting to the input domains in the heterogeneous setting. They employed a meta learning approach to optimize the hyper-parameters of the feature-wise transformation layer and showed that such layers can be incorporated to the feature extractor of state-of-the-art meta learning methods and generalize better to unseen domains with discrepancy. Another study named Domain generalization via semi-supervised meta learning (DGSML) [77] employed an entropy-based pseudo-labeling approach to assign labels to unlabeled samples in source domains and then utilized a discrepancy loss to ensure that class centroids before and after labeling unlabeled samples are close to each other. To learn a domain-invariant representation, DGSML utilized an alignment loss to ensure that the distance between pairs of class centroids, computed after adding the unlabeled samples, is preserved across different domains. This method was also trained by a meta learning approach to mimic the distribution shift between the input source domains and unseen target domains. However, this method is only applicable for classification problems because it is based on the class centroids.

Meta-learning is closely related to out-of-distribution generalization because of the episodic training of this approach that can be designed in a way to mimic the distribution shift between source domains and unseen target domains. Meta learning attempts to learn how to train a model when a few labeled examples are available per class (or task) [102, 98, 106, 94, 92, 93]. An episode is a core idea of meta learning where each episode has a support set and a query set [106]. The model is trained on the support set and then evaluated on the query set. Common approaches to meta learning are initialization-based methods and metric-based methods. In initialization-based methods, the idea is to provide a good initialization for the parameters such that the model generalizes to new classes with limited available samples as well as a few gradient steps. Model-agnostic meta learning (MAML) [106] is a well-known example of this category. In metric-based methods, the idea is to employ similarity metrics such as the Euclidean distance to guide the model to learn a representation for which samples of the same class cluster closer to each other and farther from those of the other classes. Prototypical Network (ProtoNet) [93] is a well-known example of this category that uses class centroids and the Euclidean distance to assign class labels. Although methods of meta learning have shown great performance within domain generalization, the performance of these methods drops significantly under domain discrepancy [94].

Out-of-distribution generalization has two common benchmarks for image classification. PACS benchmark [107] includes four domains: Photo, Sketch, Cartoon, and Art. Each domain has seven common categories: dog, elephant, giraffe. guitar, horse, house, and person. The total number of images is 10046. Photo has 1683 images, Sketch has 3942 images, Cartoon has 2357 images, and Art has 2061 images. VLCS benchmark [108] aggregates four domains: Caltech-101 [109], PASCAL VOC 2007 [110], LabelMe [111], and Sun09 [112]. The total number of images is 10765. Each domain has five common categories: bird, car, chair, dog, and person. Caltech has 1424 images, PASCAL has 3385 images, Labelme has 2665 images, and Sun has 3291 images.

From the feature representation point of view, there are two main approaches to out-of-distribution generalization: 1) generalizing via learning domain-invariant features [67], and 2) generalizing via learning hypothesis-invariant features [71, 72]. In domain-invariant, the most common approach, the goal is to map the input domains to a shared feature space in which the features of all domains are aligned, i.e. look similar to each other. [70, 73, 74, 75, 76]. However, forcing different domains to have very similar features is not always feasible because different domains may have unique characteristics, and completely aligning the extracted features from them ignores these unique characteristics. The second approach does not align the features but rather the predictions across domains. The idea is that if the extracted features of input domains are similar enough for an accurate predictor to make similar predictions, forcing the features to be more similar is not required anymore. [71, 67].

A recent study investigated inconsistencies between different out-of-distribution generalization methods in terms of experimental conditions such as datasets, architectures, and model selection approach [66]. They proposed DOMAINBED, a Pytorch testbed for out-of-distribution generalization that includes seven image classification/object recognition datasets with multiple domains, nine baseline methods, and three model selection approaches. One major finding of this study was that under comparable implementation and experimental design, methods of empirical risk minimization (ERM) achieve state-of-the-art performance and outperform the majority of the baseline methods. Methods of ERM are trained in a fully supervised fashion by pooling and aggregating all labeled samples across input source domains. Moreover, the results showed that the CORAL loss which was originally proposed for domain adaptation also achieves a competitive performance for out-of-distribution generalization.

In terms of the model selection approach, this study suggested three approaches including, 1) Training-domain validation set approach by splitting each training domain into training and validation subsets. Then, the validation subsets are pooled together to create an overall validation set. The best model is the one maximizing the accuracy on the overall validation set. 2) Leave-one-domain-out cross-validation approach by using one source domain entirely for validation. Models are evaluated on their held-out domain, and averaged over all held-out domains. The best model is the one maximizing this average accuracy and is retrained using all domains. 3) Test-domain validation set (oracle) approach by maximizing the accuracy on a validation set that has the same distribution as the target domain. This approach allows limited queries per method (one query per choice of hyper-parameters in a random search), meaning that there is no early stopping based on the validation set. Instead, all models are trained for the same fixed number of steps and consider only the final results [66].

In drug response prediction, source domains can be different omics data types from clinical and pre-clinical resources and target domains can be a new patient that the model encounters during deployment. This can be formulated as both homogeneous and heterogeneous because we can assume that measure of drug response is the same for source domains and future target domains or we can consider different measures of drug response for them. To the best of our knowledge, there is no method of out-of-distribution generalization for drug response prediction.

2.4 Drug Response Prediction

2.4.1 Pre-clinical and clinical samples

Broadly categorizing, there are two types of resources available in drug response prediction, clinical resources and pre-clinical resources. Clinical resources contain information about individual patients diagnosed with cancer. These resources can be labeled or unlabeled with respect to a cancer treatment, meaning that an individual patient may have received a specific treatment. Clinical resources especially with drug response available (labeled) are often small or not publicly available for privacy reasons.

Pre-clinical resources are utilized as proxy to clinical data. These resources are often larger and publicly available. Throughout this thesis, we refer to pre-clinical samples and the information they contain as pre-clinical resources. There are three types of pre-clinical resources [113]:

- Cancer cell lines: cancer cell lines are the most common pre-clinical resources. These cell lines are obtained from patient tumor biopsies, which are placed in plastic dishes and treated with different factors to immortalize the cancer cells, meaning to allow them to grow without interruption. Although cell lines have contributed to cancer research tremendously, they have some drawbacks. For example, they need comprehensive adaptation to be able to grow in a dish. Some cell lines may have gone through substantial changes and no longer recapitulate the original tumor. More importantly, they do not have the stroma or tumor microenvironment which is known to be a key player in different cancers.

- Patient-derived Xenograft (PDX) samples: PDX samples are the most common pre-clinical resources based on animals. To create a PDX sample, tissues or cancer cells are obtained from a tumor via biopsy and then engrafted into immunodeficient mice. The same tumor can be passed from mouse to mouse (this is called passaging). PDX samples provide the biological characteristics of human tumor much better than cell lines. However. establishing a PDX sample is a time and resource consuming process and more expensive compared to cell lines. Moreover, an engrafted tumor may undergo mouse related evolution.

- Organoids: The technology to create organoids as a pre-clinical resource has emerged recently (initiated in 2009) based on stem cells that can grow indefinitely and produce other cells. Organoids replicate much of the complexity of human organs and they have demonstrated promising results for drug discovery and drug screening in cancer research. Although they are less resource consuming compared to PDX samples, generating large-scale pharmacogenomics datasets based on organoids is still at the infancy stage.

In cell lines, the lack of an immune system results in downregulation of immune related pathways compared to primary tumors. Modeling the microenvironment of tumors is crucial to investigate the anti-cancer role of immune checkpoint inhibitors. Similar to cell lines, organoids often do not preserve the microenvironment. While PDX samples preserves tumor microenvironment, they are dependent on mice with deficient immune systems. A major concern is when PDX samples do not resemble cancers that usually initiate in an immune-competent host. Moreover, PDX samples in immune-deficient mice may be a poor model to study the effects of immunotherapy on the tumor microenvironment [114].

2.4.2 Omics data types

In 2001, The Human Genome Project sequenced almost 92% of a human genome and paved the way for tremendous number of discoveries [115]. 20 years later in 2021, the remaining 8% was sequenced to have the complete human genome landscape [116]. The Human Genome Project established genomics as the first omics data type which resulted in crucial steps towards the advancement of medicine [117]. However, numerous realizations such as having different phenotype or disease outcomes with the same genomic features demonstrated a need for investigating other omics data types [118]. Some of the most common biological entities that can be characterized using omics data types are as follows [117, 118]:

- Genome: A genome is like a library that contains necessary information of an organism, i.e., the complete set of DNA (including all of its genes). This information is required to build and maintain functions of that organism. Human genome consists of 3 billion DNA base pairs and is stored inside the cellular nucleus, organized in 23 pairs of chromosomes. A small fraction of DNA is also located in mitochondria. Only a very small fraction of human DNA encodes proteins. Just about 1% of the human genome consists of protein-coding genes. Genes consist of coding and non-coding parts known as exons and introns, respectively. Exons are of the most interest because mutations in these regions can change protein sequences and are more likely to be pathogenic. However, some mutations in introns or intergenic regions are also associated with human diseases including different cancers [119, 120] such as prostate cancer [121]. Whole genome sequencing (WGS) and whole exome sequencing (WES) at bulk levels are common approaches to obtain the omics data type associated with the genome. Comparing a sequenced sample (via WGS or WES) to a reference genome provides information on variants of that genome. These variants can be small such as somatic point mutations (one base substitution), insertions or deletions, copy number aberration (CNA), or other larger structural variants.

- Transcriptome: High-throughput transcriptome sequencing (also known as RNA-seq) provides the abundance of RNA from each gene. The abundance of RNA indicates activity of it and is closer to the phenotype compared to the genome. The amount of RNA transcribed from each gene is one way to measure expression value of that gene (also known as gene expression). Microarray is another (older) technology to obtain gene expression data which is based on pre-defined short sequences.

- Proteome: Proteome is the set of proteins that can be expressed by a genome. The omics data type associated with these proteins quantifies the levels of expression of proteins in a given sample. Protein expression is closer to the phenotype compared to gene expression and the concordance between gene expression and protein expression is often low [122]. The most common approach to obtain this omics data type is via mass spectrometry technology that can now quantify thousands of proteins in a single sample. The other common approach proteomics data is the reverse phase protein array (RPPA) which is highly sensitive and can detect low-abundance proteins [123].

- Epigenome: Epigenome characterizes potentially heritable chemical modifications to DNA (DNA methylation) and histone proteins. Histones enhance chromatin structure and regulate genome function which affects gene expression during different developmental stages and progression of different diseases. Epigenetic alterations can be used as markers for cancer detection, diagnosis, and prognosis.

- Metabolome: Metabolome is the collection of small molecules known as metabolites which illustrates the energy status as well as metabolism of a living organism. The omics data type associated with metabolome is another product of mass spectrometry.

- Microbiome: Microbiome provides the genome of the microbes living in individuals. ”They have essential functions in regulating growth and homeostasis and contribute to a significant fraction of our metabolome Emerging evidence suggests that the composition of a person’s microbiome is a combination of innate immunity, introduction to organisms early in life, diet, and exposure to antibiotics and other environmental factors.” [117].

Omics data is often obtained before treatment, but there are also resources and methods that can take omics data before and after treatment [124] also known as drug perturbation studies. The focus of this thesis is only on resources that obtained omics data before treatment.

2.4.3 Cancer treatment

According to the National Cancer Institute (`https://www.cancer.gov/about-cancer/treatment/types`), cancer treatment can be broadly categorized into the following categories:

- Surgery: surgery is a local treatment which is directed at a specific part of the patient's body. Surgery is helpful when tumor is only present in that specific area, or when removing one part of the tumor helps other treatments or ease distress for the patient.

- Radiation: Radiation or radiotherapy is utilizing radiation to destroy cancer cells. This can be done externally using an external device which aims at a specific part of the body (locally), or internally which is done by putting the source of radiation inside patient's body which can be solid or liquid. Radiation can also be helpful to shrink the size of tumor for surgery or other form of treatments.

- Chemotherapy: chemotherapy is a systematic treatment that travels in the body to kill cancer cells. Chemotherapy drugs are toxic and can cure or slow down cancer progression by killing cancer cells. The side effect of chemotherapy is that they also destroy normal cells and cannot distinguish between normal and cancer cells.

- Targeted therapy: Targeted drugs are the foundation of precision oncology. They are designed to kill specific cancer cells with less harm to normal cells compared to chemotherapy drugs. Targeted drugs target specific molecules (such as proteins) on cancer cells or inside them and this is possible due to differences that cancer cells and normal cells have.

- Immunotherapy: Our immune system is designed to fight infections and different diseases. The idea of immunotherapy is to fortify the immune system to fight cancer.

In this thesis, we employed and overviewed resources that utilized chemotherapy and targeted drugs.

2.4.4 Measures of drug response

Pharmacogenomics studies combine omics data of cancer cell lines or PDX samples with high throughput screening for drug response where samples are treated with a given drug (or drugs). For each drug-cell pair investigated in a dataset, cell viability at several increasing doses (concentrations) of the drug is measured and compared to an untreated control, to obtain % viability values. To learn predictors of drug response, it is desirable to obtain a single number summarizing a particular cell line's sensitivity to a drug treatment (which can then be used as a label in training computational models from the omics features).

Two common summary measures are the Area Above the Curve (AAC) and the half maximal inhibitory concentration (IC50). Both of these measures are derived by first fitting a sigmoidal curve to the dose-response data. The AAC is the area above the curve, integrated from the lowest to highest measured concentration, normalized to the concentration range. The IC50 is the concentration at which the curve crosses 50% viability. Some curves estimated in the data never cross this 50% threshold, and therefore the IC50 does not exist for many experiments where the AAC can be calculated [21] (Figure 2.1-A). Other common measures of drug response are Area Under does-response Curve (AUC), Emax which is the maximum response, and EC50 which is the concentration of the drugs that achieves half of the maximum response.

The AAC can be interpreted as measuring an average of potency and maximal efficacy, or as a measure of the mean viability across the concentrations tested [125]. While the IC50 is easily interpretable and is an absolute metric (unlike the AAC, which depends on the concentration range tested), the IC50 has some technical drawbacks which may make it difficult to use in training machine learning models. AAC/AUC is a normalized value between zero and one, but IC50 (the concentration) is not necessarily bounded and can be very small (close to zero) when samples are highly sensitive to a given drug or very large when they are highly resistant to a given drug. These issues make preprocessing of IC50 critical.

The measures like AAC and IC50 are applicable to cell lines, however they are not applicable to patients or animal models because they are obtained by increasing the doses several times. For PDX samples or patients, a common approach is to use change in tumor volume before and after treatment as the measure of drug response. One common approach for this measure is response evaluation criteria in solid tumors (RECIST) [42] that categorizes the response to four categories of complete response, partial response, stable disease, and progressive disease (Figure 2.1-B). The complete response indicates disappearance of all tumor lesions, the partial response indicates reduction of $> 30\%$ in tumor volume after treatment, the stable disease indicates reduction of $< 30\%$ or growth of $< 20\%$, and the progressive disease indicates growth of $> 20\%$ or occurrence of new lesions.

2.4.5 Pharmacogenomics datasets

Generating pharmacogenomics datasets (Table 2.3) began with the NCI-60 cell lines [126, 127] in 1986-1990 and major discoveries have been achieved via these 60 cell lines most notably Bortezomib, the treatment for multiple myeloma [4]. In 2012, the Cancer Genome Project (CGP) and the Cancer Cell Line Encyclopedia (CCLE) screened more cell lines with different drugs. CGP later evolved into the Genomics of Drug Sensitivity in Cancer (GDSCv1-2016) and CCLE evolved into The Cancer Therapeutics Response Portal (CTRPv1–2013 and v2–2015) where more drugs were screened across more cancer types. In 2015, The Genentech Cell Line Screening Initiative (gCSI) was created and at the same time the Patient-Derived Xenograft Encyclopedia (PDXE) introduced the first and only large-scale PDX dataset. Since GDSCv1 utilized the Syto60 drug screening assay but other large-scale datasets

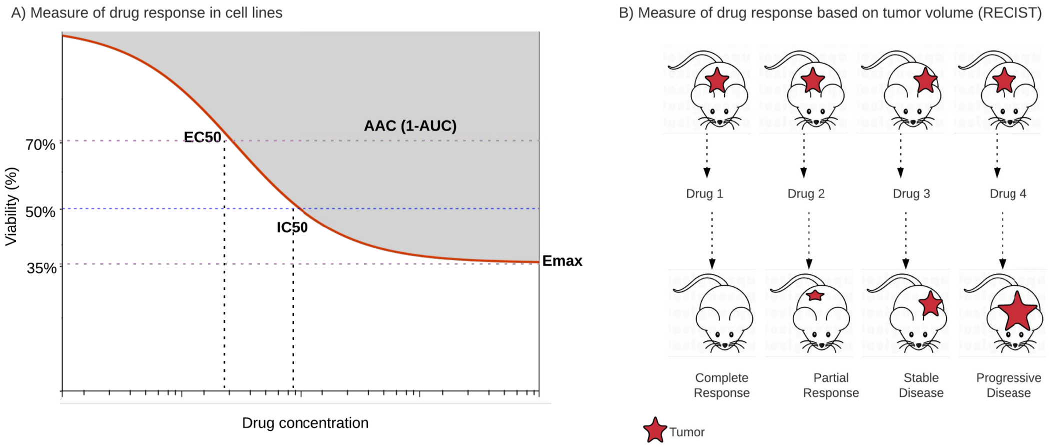

Figure 2.1: Measures of drug response based on drug concentration in cell lines and tumor volume such as RECIST in PDX samples

Table 2.3: Characteristics of pharmacogenomics datasets

Dataset	Type	#Drugs	#Samples	#Tissues	Omics
NCI60	Cell line	22,257	60	7	Multi-omics
GDSCv1	Cell line	250	1109	28	Multi-omics
GDSCv2	Cell line	190	328	27	Multi-omics
CTRPv2	Cell line	544	821	25	Multi-omics
gCSI	Cell line	16	754	22	Multi-omics
CCLE	Cell line	22	1061	25	Multi-omics
PDXE	PDX	36	440	16	Multi-omics

such as CTRPv2 and gCSI utilized the CellTiter Glo assay, in 2020, GDSCv2 dataset was introduced that screened a subset of GDSCv1 cell lines with the CellTiter Glo assay to be more comparable with the CTRPv2 and gCSI.

Due to the complexity of generating pharmacogenomics datasets, discrepancies can even exist across cell line datasets and this has been a source of controversy in this field [14, 128, 129, 130, 131, 132, 133]. Recent efforts such as the PharmacoDB project (`pharmacodb.ca`) [21], the ORCESTRA platform (`orcestra.ca`) [134], and CellMinerCDB [127] aimed at standardizing, and integrating different pre-clinical pharmacogenomics datasets to improve downstream machine learning modeling. These methods often take gene expression as input and predict the area above/under the dose-response curve (AAC/AUC) or half-maximal inhibitory concentration (IC50), the concentration of the drug that reduces the viability of cells by 50%.

2.4.6 Methods

The data-rich nature of pre-clinical pharmacogenomics datasets have paved the way for the development of machine learning approaches to predict drug sensitivity *in vitro* and *in vivo* [4, 135, 136]. These computational approaches range from simple linear regression models [19, 24] Lasso [137], and Elastic Net [26] to Random Forest [138], kernel-based models [28, 139, 41, 62], highly non-linear models based on Deep Neural Networks [20, 124, 140, 40, 61, 59, 60, 64], and most recently, reinforcement learning [141], few-shot learning [63], and multi-task learning [142]. We categorized the state-of-the-art predictors of drug response based on their input, output, and the pharmacogenomics datasets that they used for training and test (Figure 2.2). Gene expression was the most common input data type to predict drug response as it was determined to be the most effective data type in multiple studies [13, 135, 26, 28]. However, some studies based on multi-omics data also demonstrated that adding other omics data types can improve the prediction performance [20, 40]. For drug response, IC50 was the most common measure used. The cross-domain training approach was more common compared to the within-domain approach. Moreover, the majority of these methods were trained on GDSCv1 gene expression data. We also observed that incorporating drug structure, such as the Simplified molecular-input line-entry system (SMILES) representation of the drug molecule, is an emerging trend in the field. Other aspects of employing drugs as input can be drug interaction,or adverse reaction and for simplicity of illustration we labeled all of them under broad categories of “drugs” and similarly for clinical data we labeled it as “patients”.

2.4.7 Transfer learning methods

While machine learning for pharmacogenomics is a promising direction [4], existing guidelines are based on a single pharmacogenomics dataset [143] or based on benchmarking different methods without considering technical differences between molecular profiles or drug screening assays across different datasets [135]. We consider two common machine learning paradigms for drug response prediction (Figure 2.2): within-domain analysis and cross-domain analysis. In within-domain analysis, models are trained and tested on the same dataset via cross-validation which means train and test data are from the same distribution. In cross-domain analysis, models are trained and tested on different cell line datasets to investigate generalization capability. The cross-domain analysis offers investigating translatability of models on clinical datasets but it creates a need for transfer learning to address data discrepancies between pre-clinical and clinical resources.

Various methods of transfer learning have been proposed in the context of drug response prediction [40, 59, 60, 61, 41, 62, 63, 64, 65, 144]. These methods either address existing discrepancies implicitly, or explicitly which means they assume that the model has access to the target domain during training or fine-tuning. Among these methods, Velodrome is the

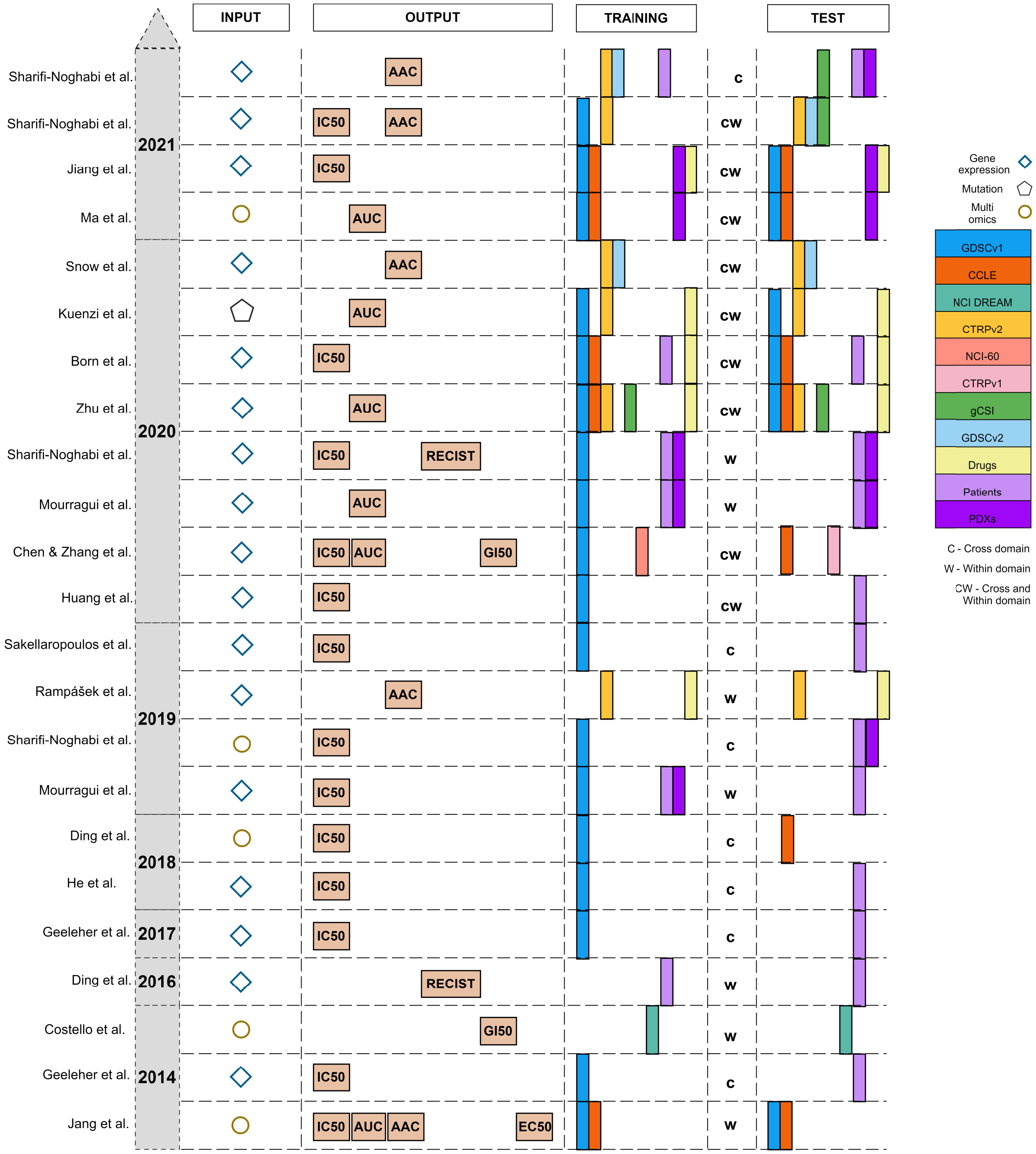

Figure 2.2: Some of the published studies for drug response prediction. Gene expression is the most common molecular profile and IC50 is the most common pharmacological profile, but AAC/AUC has become more common in recent studies. GDSCv1 (originally named CGP) is the most common training dataset and the use of drug information for training has been more frequent in recent years. The cross-domain training approach denoted by “c” was more common compared to the within-domain approach denoted by “w”. When a method employs both of them, we denote it by “cw”.

Table 2.4: Overview of methods of transfer learning in drug response prediction

Method	What?	How?	#domain	pharmacogenomics dataset
MOLI [40]	samples	triplet loss function	single	GDSCv1, PDXE, clinical trials
BDKANN [59]	relational knowledge	standard supervised	single	CTRPv2, GDSCv2
DrugCell [60]	relational knowledge	DNN	single	CCLE, DeepSynergy
AITL [61]	features	adversarial multi-task learning	single	GDSCv1, PDXE, clinical trials
PRECISE [41]	features	kernel learning	single	GDSCv1, PDXE, TCGA
TRANSACT [62]	features	kernel learning	single	GDSCv1, PDXE, TCGA, Hartwig Medical Foundation
Ma et al. [63]	parameters	few-shot learning	single	GDSCv1, CCLE, PDX
Zhou et al. [64]	parameters	fine-tuning	single	GDSCv1, CCLE, CTRPv2, gCSI
Celligner [65]	features	kernel learning	multiple	CCLE, TCGA, TARGET, Treehouse
TUGDA [144]	features	multi-task domain adaptation	single	GDSCv1, TCGA, PDXE
Velodrome	features	out-of-distribution generalization	multiple	CTRPv2, GDSCv2, gCSI, TCGA, PDXE, clinical trials

only method that does not require access to the target domain during training or fine-tuning. MOLI [40] is also the only method that performs transfer learning in the form of transferring relevant samples between source domain and target domain (see section 3.2.4). MOLI is also the only multi-omics methods, however, other studies such as [64] take drug structure (SMILES) as input as well. BDKANN [59] and DrugCell [60] are based on transferring relational knowledge of REACTOME (`https://reactome.org`) and gene ontology terms, respectively to design the architecture of the DNN model. TRANSCACT [62] is non-linear domain adaptation method based on kernel learning and a follow-up version of [41] which is based on linear kernel learning. Table 2.4 summarizes these methods with respect to the three main questions in transfer learning. We did not consider when to transfer in this table because all these methods were proposed for drug response prediction.

2.4.8 Summary

This section described the elements of drug response prediction such as the definition of pre-clinical and clinical resources (patients, cell lines, PDX samples), the definition of different omics data types (the input for computational methods) that we can obtain from these resources. Moreover, this section discussed different measures of drug response (the output of computational methods) obtained from available treatments for cancer patients. Finally this section provided a brief overview of existing datasets, methods, and transfer learning approaches for drug response prediction.

2.5 Semi-supervised learning

Semi-supervised learning attempts to leverage unlabeled data during training. Common approaches to semi-supervised learning are consistency regularization [145] and pseudo-labeling [146]. In consistency regularization, the model predicts labels for the unlabeled samples and these predictions should be consistent for the perturbed version of the same samples. Mean Teacher [147] is a well-known example of consistency regularization methods where a student model and a teacher model are being trained jointly based on a supervised loss on the predictions of the student model and a consistency loss on the predictions of the student and the teacher model. The parameters of the teacher model are being optimized as an exponential moving average of the parameters of the student model but student and teacher apply different noises (augmentations) to the input images to benefit more from the consistency loss. In pseudo-labeling, the idea is to utilize the predicted labels by the model for unlabeled samples with high confidence (e.g. above a certain threshold) and use those samples and their predicted pseudo-labels in retraining the model.

A recent study showed that combining both consistency regularization and pseudo-labeling improves the state-of-the-art performance in semi-supervised learning benchmarks [148]. Moreover, incorporating pseudo-labeling in meta learning in semi-supervised ProtoNet has shown that utilizing both labeled and unlabeled data improves the performance of the models trained on only the labeled data [149]. This method assigns labels based on the Euclidean distance to the class centroids obtained from the labeled data. These centroids are then updated using the pseudo-labels assigned to the unlabeled data.

Another study explored methods of semi-supervised learning from the imbalanced classes point of view [150]. This is particularly important for drug response prediction because pharmacogenomics datasets are often imbalanced. This study demonstrated that if labeled and unlabeled data are relevant to each other (for example most of samples are from the same classes), semi-supervised learning will be beneficial even when the unlabeled samples are very imbalanced themselves. However, if labeled and unlabeled samples are irrelevant (they are mostly from disjoint classes), then self-supervision will be more beneficial compared to semi-supervised learning. Applying self-supervision even when it is only on the labeled examples provides the model with strong initial values for the parameters that can be utilized as a pre-trained model for the ultimate task.

Chapter 3

Multi-omics Integration

This chapter is adapted based on a published article [40] under license CC BY-NC.

3.1 Problem definition

Given a training data with k different modalities $\{X_1,...,X_k\}$, where $X_i \in R^{M \times N}$ and $Y_i^{M \times 1} \in \{0,1\}$ meaning that each modality has M samples, N features, and binary labels, the goal is to predict Y accurately by integrating the input modalities. In the area of pharmacogenomics, the training data has multi modals (also known as multi-omics in the driving application) including the gene expression, mutation, and CNA obtained from the cell lines, and the training label is the drug response in the form of binarized IC50 values.

3.2 MOLI: Multi-Omics Late Integration

MOLI [40] is a deep neural network that predicts the drug response for a given sample, represented by its multi-omics profile, and for a given drug. MOLI assumes that values for the same genes are provided for each omics data type. MOLI's network consists of the following subnetworks. It has multiple feed forward encoding subnetworks, one for each input omics data type. Each encoding subnetwork receives its corresponding omics data and encodes it into a learned feature space. The learned features from the encoding subnetworks are integrated into one representation by concatenation. The concatenated representation serves as input for a classification subnetwork, which predicts the drug response. The entire network is trained in an end-to-end fashion using an objective function combining a classification loss and a triplet loss. Figure 3.1 shows MOLI's components during training and model development, while Figure 3.2-A shows the application of MOLI for external validation.

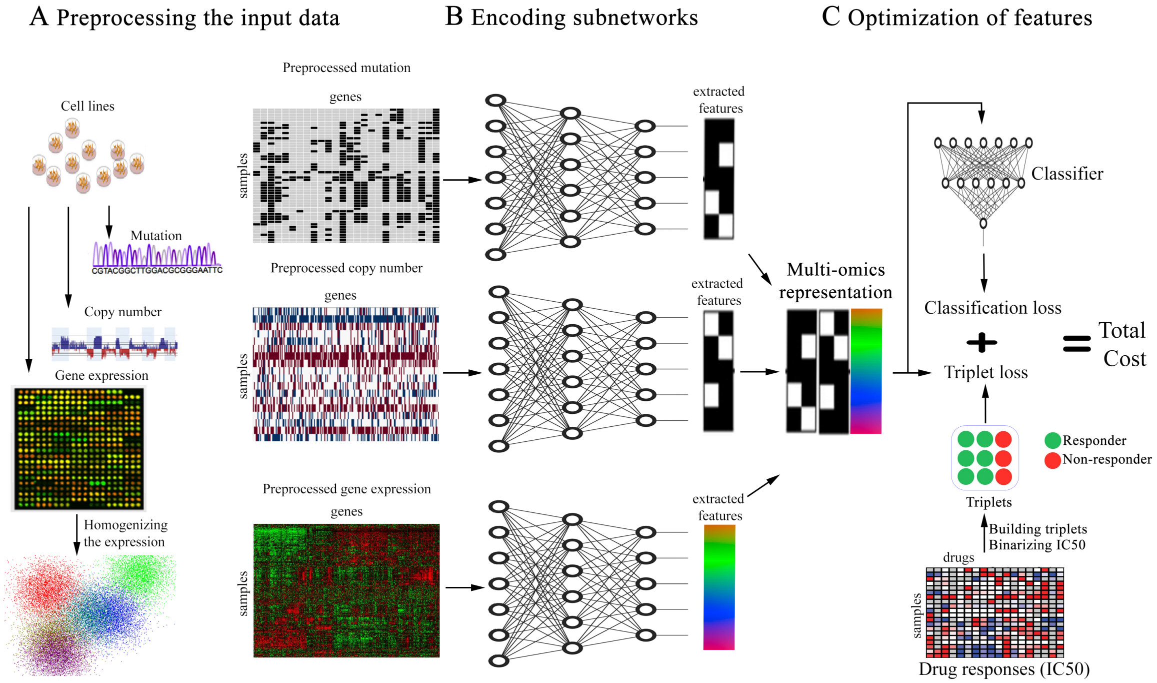

Figure 3.1: **Schematic overview of MOLI** (A) preprocessing mutation, CNA, and gene expression data. (B) Each encoding subnetwork learns features for its omics data type and the learned features are concatenated into one representation. (C) MOLI objective function consists of a triplet loss and a classification loss, obtained from the classifier subnetwork that uses the multi-omics representation to predict drug response.

3.2.1 Learning features by encoding subnetworks

To learn features for each omics data type in the input, we design separate encoding feed forward subnetworks to map the input space to the feature space. We focus on mutation, CNA, and gene expression data. X_M, X_E, and X_C denote mutation, CNA, and gene expression data, respectively, each of which are of dimensionality $N \times D$, where N is the number of samples and D is the number of genes. We note that the proposed approach can be extended for any number of omics data types. Each encoding subnetwork has a fully connected layer with Relu activation functions. In addition, each subnetwork employs dropout to regularize the model and batch normalization to enhance the training process. The input of each encoding subnetwork is one omics data type and the output is the learned features for that omics (Figure 3.1- B). We denote these subnetworks as $f_M(X_M)$, $f_C(X_C)$, and $f_E(X_E)$, respectively.

3.2.2 Integrating learned features by late integration

In the integration step, we utilize a late integration approach and concatenate the learned features of the different single-omics data types to obtain one multi-omics representation. For example, if the outputs of three encoding subnetworks are three $M \times N$ feature matrices,

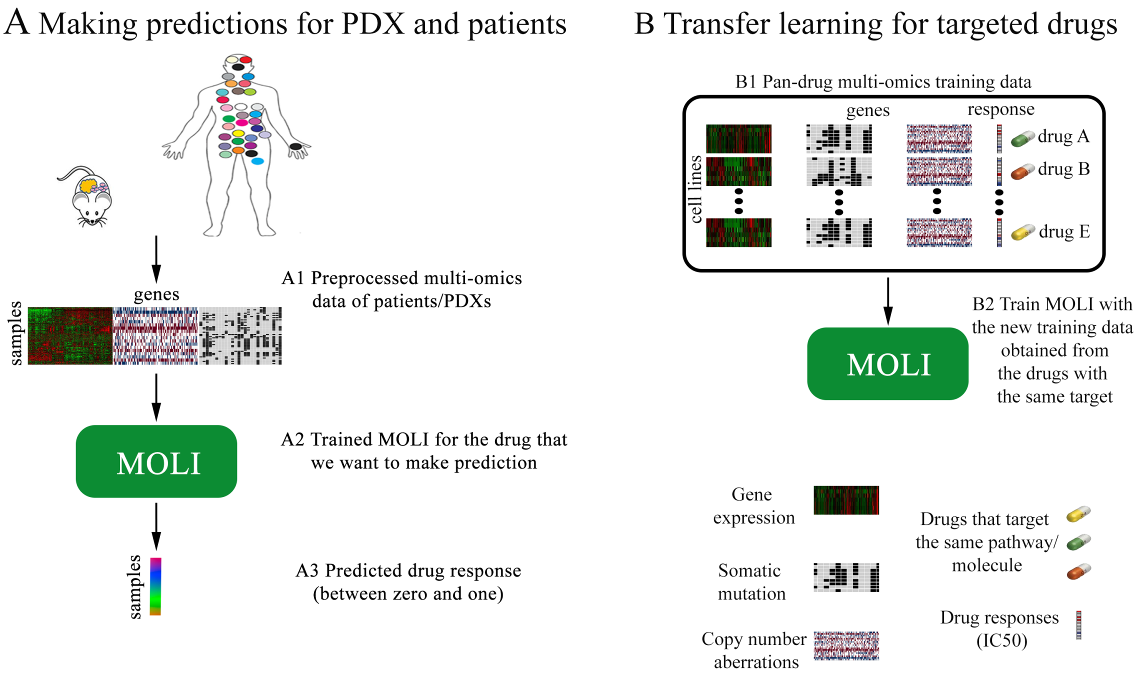
A Making predictions for PDX and patients
genes
samples
A1 Preprocessed multi-omics data of patients/PDXs
MOLI
A2 Trained MOLI for the drug that we want to make prediction
samples
A3 Predicted drug response (between zero and one)
B Transfer learning for targeted drugs
B1 Pan-drug multi-omics training data
genes
response
cell lines
drug A
drug B
drug E
MOLI
B2 Train MOLI with the new training data obtained from the drugs with the same target
Gene expression
Somatic mutation
Copy number aberrations
Drugs that target the same pathway/ molecule
Drug responses (IC50)

Figure 3.2: (A) Using MOLI to make predictions for PDX/patient inputs during external validation. (B) Combining targeted drugs that target the same pathway or molecule to make a pan-drug training dataset for MOLI

after concatenation, the output will be one $M \times 3N$ representation matrix. The integrated representation is further smoothed through a $l2$ normalization layer. We denote MOLI's integration, receiving multi-omics data as input and returning the integrated representation, as follows:

$$F(X_M, X_C, X_E) = f_M(X_M) \oplus f_C(X_C) \oplus f_E(X_E), \tag{3.1}$$

where, $\oplus$ denotes the concatenation operator.

3.2.3 Optimizing the learned features by the combined objective function

The learned features will be used by a classifier that predicts the drug response. Therefore, the last subnetwork of MOLI is a classification layer with the Sigmoid activation function, using dropout and weight decay for regularization (Figure 3.1-C). We denote this classifier as *g(.)*. Since the MOLI network will be used for classification, i.e. drug response prediction, the objective function used for training must include a term that measures the difference between the predicted drug response and the ground truth drug response. We choose the binary cross-entropy classification loss, one of the most common classification losses, defined as follows:

$$L_{Classifier} = -[Y \log g(F(X_E, X_M, X_C)) + (1 - Y) \log(1 - g(F(X_E, X_M, X_C))], \tag{3.2}$$

where, $Y_{N\times 1}$ denotes the binarized IC50 which is used as measure for the drug response.

We add a triplet loss to the objective function to impose a further constraint that is necessary for accurate classification. This constraint forces responders to be more similar to each other than to non-responders. The triplet loss function was introduced in FaceNet [95] for optimizing the mapping from a space of face images to a Euclidean space where the difference between learned features is correlated with the similarity among faces. The idea is that for the image of a given person's face, the distance between that image's learned features and the features of another image of the same person should be smaller than the distance between that image's learned features and the learned features of the image of some other person. In our context, we employ the triplet loss function as follows. For T given triplets in the form of (Anchor, Positive, Negative), where the first two are (the multi-omics data of) responder cell lines to a given anti-cancer drug and the last one is (the multi-omics data of) a non-responder to that drug, we require the following condition:
$d(F(Anchor_i), F(Positive_i)) \leq d(F(Anchor_i), F(Negative_i))$, where *d(.)* is an arbitrary distance function—we used the Euclidean distance.

If we move the right hand-side to the left, we obtain:

$$d(F(Anchor_i), F(Positive_i)) - d(F(Anchor_i), F(Negative_i)) \leq 0 \tag{3.3}$$

In order to avoid the trivial zero solution, a margin $\xi > 0$ is required:

$$d(F(Anchor_i), F(Positive_i)) - d(F(Anchor_i), F(Negative_i)) + \xi \leq 0 \tag{3.4}$$

We want the distance of the Anchor and the Negative to be larger than the distance of the Anchor and the Positive. Thus, the value of the triplet loss function for the i-th triplet is:

$$L^i_{Triplet} = max[d(F(Anchor_i), F(Positive_i) - d(F(Anchor_i), F(Negative_i)) + \xi, 0] \tag{3.5}$$

and the total triplet loss for T triplets is:

$$L_{Triplet} = \sum_{i=1}^{T} L^i_{Triplet} \tag{3.6}$$

Generally, there are two approaches to select triplets for the triplet loss function: offline selection and online selection. The offline selection builds the triplets based on the value of the labels (in this case the drug response) before training the model. The online selection selects the triplets from samples in each mini-batch during the training. We adopted the online approach. Triplets can be built based on all possible combinations of the input samples/mini-batches (soft selection) or based only on those triplets with high triplet loss value (hard selection). Soft selection provides the model with more training triplet examples but the network might rely too much on easy cases, and as a result may be unable to perform well on hard examples [95]. Hard selection solves this problem by only relying on the hard cases in the train data to build the triplets, but this approach may suffer from having fewer training triplets especially in the case of small unbalanced datasets. We adopted the soft selection approach.

Therefore, the combined cost J is defined as follows:

$$J = L_{Classifier} + \gamma L_{Triplet} \tag{3.7}$$

where γ is a regularization term for the triplet loss.

3.2.4 Transfer learning for targeted drugs

For targeted drugs, we use transfer learning and train MOLI with a new pan-drug input. This pan-drug input consists of multi-omics profiles and drug responses for a family of targeted drugs that target the same pathway or molecule. Such drugs are expected to produce highly correlated responses in cell lines. One MOLI model is trained for a family of drugs instead of one separate model for each individual drug. This approach increases the training dataset size, since the set of the screened cell lines and the obtained responses are similar but not identical for the drugs of one family. In our experiments, we evaluate transfer learning for EGFR pathway inhibitors due to the availability of external validation data, but the approach is applicable to any family of targeted drugs. Figure 3.2- B illustrates the idea of transfer learning for targeted drugs.

3.2.5 Predicting drug response for TCGA patients

To study MOLI's performance, similar to [24], we employ the model trained on the pan-drug input for the EGFR inhibitors to predict the drug response for patients in several TCGA datasets for which there was no drug response recorded. Since these drugs target EGFR pathway, we expect the expression status of the genes of this pathway to be strongly correlated with the predicted drug response. We obtain the list of genes in EGFR pathway from REACTOME. To study the correlation, we employ multiple linear regression between the predicted responses and the level of expression. We obtain p-values for each gene and correct them for multiple comparison, using Bonferroni correction ($\alpha = 0.05$).

3.3 Experimental results

3.3.1 Datasets

We use four main resources for multi-omics integration:

- Genomics of Drug Sensitivity in Cancer (GDSCv1) cell lines dataset [13]

- Patient-Derived Xenograft Encyclopedia (PDXE) dataset [11]
- TCGA patients with the drug response available in their records [26]
- TCGA patients without the drug response [25]

The GDSCv1 dataset [151, 13] has created a multi-omics dataset of more than a thousand cell lines from different cancer types, screened with 265 targeted and chemotherapy drugs. We use GDSCv1 as the training dataset due to a high number of screened drugs. Multi-omics profiles and drug responses for GDSCv1 are retrieved from `ftp://ftp.sanger.ac.uk/pub/project/cancerrxgene/releases/release-7.0/`.
We use the other publicly available multi-omics datasets for external validation as follows:

1. We apply PDX Encyclopedia mice models published by [11]. This dataset has more than 300 PDX models for different cancer types, screened with 34 targeted and chemotherapy drugs. Response in terms of RECIST was binarized so that Complete Response and Partial Response were considered as sensitive and Stable Disease and Progressive Disease are considered as resistant, moreover, Unstable Responses were excluded as well as response to combination treatments.

2. TCGA [25] data including profiles of tumor samples collected from more than ten thousand patients with different cancer types, downloaded from Firehose Broad GDAC (`https://doi.org/10.7908/C11G0KM9`, `http://gdac.broadinstitute.org/runs/stddata__2016_01_28/`). For TCGA datasets, we use clinical annotations of the drug response for some patients which were obtained from supplementary material of [26]. Similar to PDX response, Complete Response and Partial Response were considered as sensitive and Stable Disease and Progressive Disease are considered as resistant, moreover, combination treatments were excluded.

3. We also use TCGA patients for breast (BRCA), bladder (BLCA), pancreatic (PAAD), lung (LUAD), kidney (KIRP), and prostate (PRAD) cancers. These patients are without the drug response in their records.

We note that we used only those genes which are in common for all of the omics data types in both training and external validation datasets for each drug.

Table 3.1 provides the characteristics of each dataset such as type of drug, the number of samples, and the number of genes. After the preprocessing, we have the same number of genes for the training and the external validation datasets and for each of the three omics data types. We only consider samples for which all three omics data types are available.

Gene expression profiles

Raw intensities obtained from ArrayExpress (E-MTAB-3610) for GDSCv1 dataset were RMA-normalized (Robust Multi-Array Average) [152], log-transformed and aggregated to

Table 3.1: List of the studied drugs from the used resources with multi-omics profiles available.

Drug	Type	Resource	Number of samples^{+}	Number of genes^{++}	Usage
Afatinib	Targeted	GDSCv1	828 (NR:678, RS:150)	13081	Training
Cetuximab	Targeted	GDSCv1	856 (NR:735, RS:121)	12346*/13081**	Training
Cetuximab	Targeted	PDX	60 (NR:55, RS:5)	12346*/13081**	External validation
Cisplatin	Chemotherapy	GDSCv1	829 (NR:752, RS:77)	15493	Training
Cisplatin	Chemotherapy	TCGA	66 (NR:6, RS:60)	15493	External validation
Docetaxel	Chemotherapy	GDSCv1	829 (NR:764, RS:65)	15016	Training
Docetaxel	Chemotherapy	TCGA	16 (NR:8, RS:8)	15016	External validation
Erlotinib	Targeted	GDSCv1	362 (NR:298, RS:64)	12325*/13081**	Training
Erlotinib	Targeted	PDX	21 (NR:18, RS:3)	12325*/13081**	External validation
Gefitinib	Targeted	GDSCv1	825 (NR:710, RS:115)	13081	Training
Gemcitabine	Chemotherapy	GDSCv1	844 (NR:790, RS:54)	12067*/15381	Training
Gemcitabine	Chemotherapy	PDX	25 (NR:18, RS:7)	12067	External validation
Gemcitabine	Chemotherapy	TCGA	57 (NR:36, RS:21)	15381	External validation
Lapatinib	Targeted	GDSCv1	387 (NR:326, RS:61)	13081	Training
Paclitaxel	Chemotherapy	GDSCv1	389 (NR:363, RS:26)	12482	Training
Paclitaxel	Chemotherapy	PDX	43 (NR:38, RS:5)	12482	External validation
pan-drug	Targeted	GDSCv1	3258 (NR:2747, RS:511)	13081	Training

NR: Non-resonder; RS: Responder; $^{+}$:Number of screened samples with all three omics data types available; $^{++}$: Number of genes in common between the train data and the external validation data for each drug; *: Number of genes for the drug-specific experiments ; **: Number of genes for the pan-drug experiments

the level of genes. Gene expression values of PDX and all TCGA datasets are converted to Transcripts Per Million (TPM) [153] and log-transformed. To make expression profiled by different platforms comparable, we standardize gene expression and perform pairwise homogenization procedure, as described in [154, 19]. Also, in each dataset we exclude the 5% of genes with lowest variance assuming them to be not informative.

Somatic copy number profiles

We remove unreliable segments from genome segmentation files for TCGA datasets and assign every gene a value corresponding to the intensity log-ratio of the segment it overlaps. If the gene overlaps more than one segment, we keep the most extreme log-ratio value. Different from TCGA, the GDSCv1 and PDX datasets provided gene-level estimates of total copy number. In order to make these data comparable with TCGA, we compute for every gene the logarithm of its copy number divided by the ploidy of copy-neutral state in the sample. Finally, for all four datasets we binarize gene-level copy number estimates assigning zeros to copy-neutral genes and ones to all genes overlapping deletions or amplifications.

Somatic point mutations

Similarly with previous work [13, 20], we assign ones to genes carrying somatic point mutations and zeros to all others.

3.3.2 Experimental design

In our experiments, we investigated the following questions:

1. Does MOLI outperform single-omics and early integration baselines in terms of prediction Area Under the Receiver Operating Characteristic curve (AUROC) and the Area Under the Precision-Recall curve (AUPR) on PDX and patient data?

2. Does transfer learning work for targeted drugs, i.e. does MOLI trained on pan-drug data outperform MOLI trained on drug-specific (single drug) data?

3. Finally, for the targeted drugs, does the predicted response by MOLI have associations with the target of that drug?

We trained MOLI on GDSCv1 cell lines screened with Docetaxel, Cisplatin, Gemcitabine, Paclitaxel, Erlotinib, and Cetuximab. We chose these drugs based on availability of PDX/patient multi-omics data for these drugs which is necessary for external validations. We trained all of the baselines for the same drugs and compared them to MOLI in terms of prediction AUROC.

We compared MOLI against early integration via deep neural networks inspired by [20] and early integration via non-negative matrix factorization (NMF) [155, 156], against the single-omics (gene expression) ridge regression method proposed by [19], against an ordinary feed forward network with classification loss trained on the expression data, and against a version of MOLI trained only on the gene expression data. To test whether the triplet loss contributes to improve the performance, we compared MOLI to a late integration feed forward network with an architecture similar to MOLI but using only a classification loss.

Finally, to study transfer learning for the targeted drugs, we focused on drugs that target the EGFR pathway because we have Cetuximab and Erlotinib that target this pathway in the PDX dataset utilized for external validations. In addition, GDSCv1 was screened with numerous drugs that target EGFR including: Afatinib, Cetuximab, Erlotinib, Gefitinib, and Lapatinib. We used multi-omics data of all of these drugs in GDSCv1 and created a large training set (>3,000 samples). We trained MOLI on this pan-drug data and compared the results to MOLI which was trained on the drug-specific inputs.

We used 5-fold cross validation in most of the experiments to tune the hyper-parameters of the deep neural networks based on the AUROC. The hyper-parameters tuned were number of nodes in the hidden layers, learning rates, mini-batch size, weight decay, the dropout rate, number of epochs, and margin and regularization term (only for the triplet loss). The ranges considered for each hyper-parameter are as follows: Mini-batch size $= [8, 16, 32, 64]$.
Number of nodes $= [2048, 1024, 512, 256, 128, 64, 32, 16]$.
Margin $= [0.5, 1, 1.5, 2, 2.5, 3, 3.5]$.
Learning rate $= [0.1, 0.5, 0.01, 0.05, 0.001, 0.005, 0.0001, 0.0005, 0.00001, 0.00005]$.
Number of epochs $= [5, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 150, 200]$.

Dropout rate = $[0.3, 0.4, 0.5, 0.6, 0.7, 0.8]$.
Weight decay = $[0.1, 0.01, 0.001, 0.1, 0.0001]$.
Gamma = $[0.1, 0.2, 0.3, 0.4, 0.5, 0.6]$.

Finally, the network was re-trained on the obtained hyper-parameters on the entire dataset for that drug (train and validation). We used Adagrad for optimizing parameters in all of the deep neural networks [157]. We used the Pytorch framework to implement all deep neural networks codes. For the ridge regression pipeline, we downloaded the implemented pipeline with leave-one-out cross validation provided by the original authors [19] and applied it to our datasets. To make sure that both the downloaded pipeline and the way we preprocessed the gene expression data is correct, we evaluated it on the datasets from the original paper and got AUROCs for Docetaxel and Bortezomib comparable to those of [139]. For early integration via NMF, we first concatenate the omics data types, and then train a NMF on the the resulting matrix to learn the latent factors. Finally we train the [19] method (using the learned factors as features) to predict the drug response. The final selected hyper-parameters for MOLI are as follows:

PDX Paclitaxel:
64 (mini-batch size), 512, 256, 1024 (number of nodes in expression, mutation and CNA subnetworks), 0.0005 (expression learning rate), 0.5 (mutation learning rate), 0.5 (CNA learning rate), 0.5 (classifier learning rate), 0.4 (expression dropout), 0.4 (mutation dropout), 0.5 (CNA dropout), 0.0001 (weight decay), 0.3 (classifier dropout), 0.6 (gamma for regularization), 10 (epochs), 0.5 (margin)

PDX Gemcitabine:
13 (mini-batch size), 256, 32, 64 (number of nodes in expression, mutation and CNA subnetworks), 0.05 (expression learning rate), $1e-5$ (mutation learning rate), 0.0005 (CNA learning rate), 0.001 (classifier learning rate), 0.4 (expression dropout), 0.6 (mutation dropout), 0.3 (CNA dropout), 0.01 (weight decay), 0.6 (classifier dropout), 0.3 (gamma for regularization), 5 (epochs), 1.5 (margin)

PDX Erlotinib:
32 (mini-batch size), 64, 64, 64 (number of nodes in expression, mutation and CNA subnetworks), 0.5 (expression learning rate), 0.5 (mutation learning rate), 0.1 (CNA learning rate), 0.1 (classifier learning rate), 0.5 (expression dropout), 0.5 (mutation dropout), 0.5 (CNA dropout), 0.01 (weight decay), 0.5 (classifier dropout), 0.5 (gamma for regularization), 5 (epochs), 1 (margin)

PDX Cetuximab:
30 (mini-batch size), 256, 512, 128 (number of nodes in expression, mutation and CNA subnetworks), 0.0001 (expression learning rate), 0.0005 (mutation learning rate), 0.0005 (CNA learning rate), 0.0005 (classifier learning rate), 0.3 (expression dropout), 0.8 (mutation dropout), 0.8 (CNA dropout), 0.01 (weight decay), 0.4 (classifier dropout), 0.2 (gamma for regularization), 10 (epochs), 2 (margin)

TCGA Gemcitabine:
13 (mini-batch size), 16, 16, 16 (number of nodes in expression, mutation and CNA subnetworks), 0.001 (expression learning rate), 0.0001 (mutation learning rate), 0.01 (CNA learning rate), 0.05 (classifier learning rate), 0.5 (expression dropout), 0.5 (mutation dropout), 0.5 (CNA dropout), 0.001 (weight decay), 0.5 (classifier dropout), 0.6 (gamma for regularization), 10 (epochs), 2 (margin)
TCGA Docetaxel:
8 (mini-batch size), 16, 16, 16 (number of nodes in expression, mutation and CNA subnetworks), 0.0001 (expression learning rate), 0.0005 (mutation learning rate), 0.0005 (CNA learning rate), 0.001 (classifier learning rate), 0.5 (expression dropout), 0.5 (mutation dropout), 0.5 (CNA dropout), 0.001 (weight decay), 0.5 (classifier dropout), 0.4 (gamma for regularization), 10 (epochs), 0.5 (margin)
TCGA Cisplatin:
15 (mini-batch size), 128, 128, 128 (number of nodes in expression, mutation and CNA subnetworks), 0.05 (expression learning rate), 0.005 (mutation learning rate), 0.005 (CNA learning rate), 0.0005 (classifier learning rate), 0.5 (expression dropout), 0.6 (mutation dropout), 0.8 (CNA dropout), 0.1 (weight decay), 0.6 (classifier dropout), 0.2 (gamma for regularization), 20 (epochs), 0.5 (margin)

3.3.3 Results

Table 3.2 reports the performance of MOLI and the baselines in terms of AUROC.

First, we compared the complete MOLI (MOLI trained on multi-omics data and using its combined objective function) to the early integration baselines. MOLI achieved better performance in six out of seven external validation datasets compared to early integration via deep neural networks. Moreover, MOLI also achieved better performance in all of the external validation datasets compared to early integration via NMF. These results indicate that multi-omics late integration outperforms multi-omics early integration.

Second, we compared MOLI trained on multi-omics data with two deep neural network scenarios trained only on the gene expression data (one of them was MOLI itself). MOLI trained on multi-omics data showed better performance in four out of seven external validation datasets and tied in another dataset. These results indicate that deep neural networks trained on multi-omics data achieve better performance than those trained on single-omics data.

Third, we compared MOLI with MOLI without the triplet loss, both trained on multi-omics input. MOLI with its combined objective function obtained better performance in five out of seven external validation datasets and tied in another one. These results demonstrate the contribution of the triplet loss to improving the prediction performance.

Finally, we compared MOLI against a single-omics non-deep neural networks baseline which is the only published method for drug response prediction [19] that had been tested

Table 3.2: Performance of different versions of MOLI compared to the baselines in terms of prediction AUROC across two targeted therapeutics and five chemotherapy agents.

Method/ Drug	**PDX Paclitaxel**	**PDX Gemcitabine**	**PDX Cetuximab**	**PDX Erlotinib**	**TCGA Docetaxel**	**TCGA Cisplatin**	**TCGA Gemcitabine**	**Input omics**
[19]	0.52	0.59	0.58	0.67	0.59	0.62	0.53	Expression
Early integration (NMF)	0.24	0.56	0.53	0.28	0.39	0.40	0.58	Multi
Early integration (DNNs)	did not converge	**0.66**	did not converge	did not converge	0.52	did not converge	0.59	Multi
Feed Forward Net	0.68	0.48	0.43	0.37	**0.69**	0.44	0.65	Expression
MOLI complete	0.69	0.52	0.51	0.39	0.63	**0.75**	0.64	Expression
MOLI with classifier	did not converge	0.55	0.46	did not converge	0.58	0.6	**0.69**	Multi
MOLI complete	**0.74**	0.64	0.53	0.63	0.58	0.66	0.65	Multi
MOLI pan-drug	not targeted	not targeted	**0.80**	**0.72**	not targeted	not targeted	not targeted	Multi

on patient data. MOLI achieved better performance in four out of seven external validation datasets and tied with this baseline in another one. These experiments show the substantial gain in predictive performance resulting from the combination of using multi-omics data, deep neural networks, and the proposed objective function.

MOLI when trained on the pan-drug input (only applicable for targeted drugs), had significantly better performance compared to itself when it was trained on the drug-specific inputs for Erlotinib and Cetuximab. The majority of the baselines had either poor performance or did not converge for Paclitaxel and Erlotinib. This means that during either cross validation or final re-training with the obtained hyper-parameters the cost and/or AUROC curves were fluctuating. This may be due to the small number of samples, because both of these drugs had the fewest number of cell lines (∼400). Also, we observed the lack of convergence in the early integration baseline for four drugs which may be due to the concatenation at the beginning because it increased the dimensionality substantially, which makes feature learning harder for the autoencoder and later the classifier in this method.

MOLI achieved an AUROC of greater than 0.7 for four drugs (Paclitaxel, Cetuximab, Erlotinib, and Cisplatin) which may be beneficial for precision oncology particularly for the targeted drugs (Cetuximab and Erlotinib).

We also studied the area under precision-recall curve for MOLI and the main baselines including the early integration methods and the [19] method (Figure 3.3). Compared to the early integration methods, MOLI achieved better performance in four out of seven external validation datasets and tied in two other datasets. Compared to the [19] single-omics baseline, MOLI had better performance in six out of seven external validation datasets. We also investigated the area under precision-recall curve for the pan-drug training data. MOLI trained on pan-drug data had better performance in one external validation dataset and had competitive performance in another one compared to MOLI trained on the drug-specific input. All these results again suggest that MOLI may be beneficial for precision oncology.

Transfer learning for targeted drugs improves performance significantly

We observed that for the targeted drugs (in our experiment, EGFR inhibitors), MOLI trained on the pan-drug multi-omics inputs achieved significantly better performance than MOLI trained on drug-specific inputs. Pan-drug MOLI achieved an AUROC of 0.8 for Cetuximab

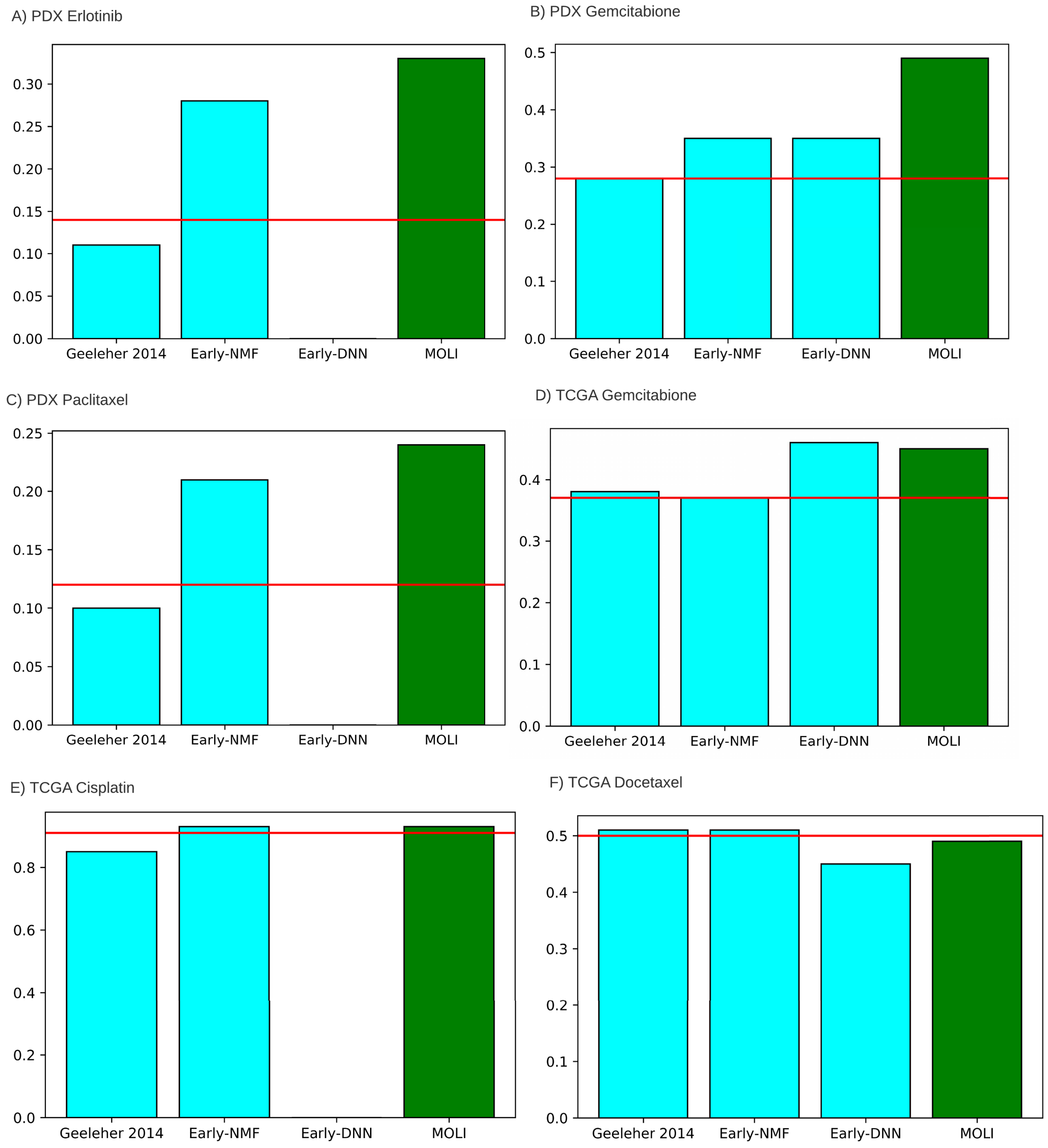

Figure 3.3: Performance of MOLI and the baselines in terms of prediction AUPR, Red line illustrates random predictors performance

and 0.72 for Erlotinib which were significantly higher than the drug-specific performance (Figure 3.4 for AUPR performance). This suggests that transfer learning can improve the prediction performance for the targeted drugs.

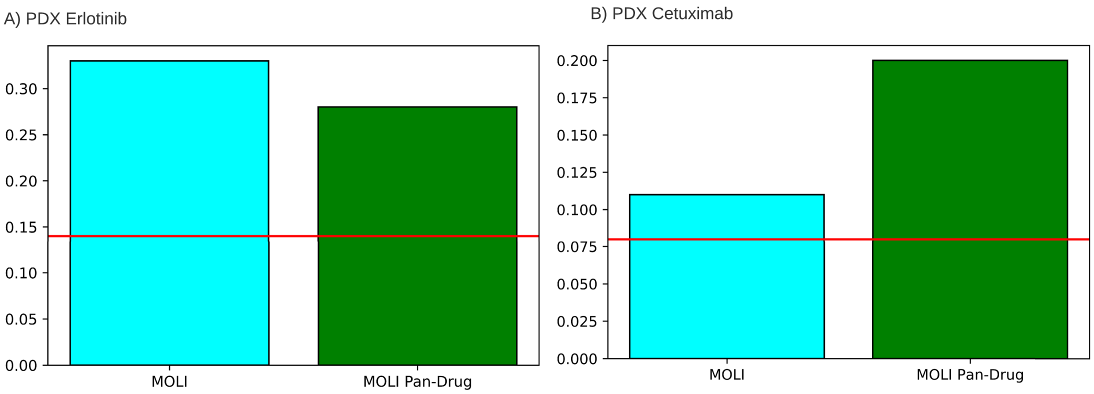

Figure 3.4: Performance of MOLI and the Pan-Drug MOLI in terms of prediction AUPR, Red line illustrates random predictors performance

Predictions for TCGA patients by MOLI have associations with EGFR genes

We applied MOLI (trained on the pan-drug input for EGFR inhibitors) to multi-omics data without drug response downloaded from TCGA (breast, bladder, pancreatic, lung, kidney, and prostate cancers) and predicted the response for these patients. According to the p-values obtained from multiple linear regression, there are a number of strong associations between EGFR genes and the responses predicted by MOLI. For breast cancer, we observed statistically significant associations between the level of expression in AP2A1 ($P = 0.007$), CALM2 ($P = 0.01$), CLTA ($P = 0.0002$), EGFR ($P = 1 \times 10^{-5}$), PIK3CA ($P = 0.007$), and UBA52 ($P = 3 \times 10^{-}6$) genes and the predicted responses. For prostate cancer, we found that the predicted responses have statistically significant associations with the expression of AKT1 ($P = 0.02$), CDK1 ($P = 0.01$), RICTOR ($P = 0.0002$), CREB1 ($P = 0.02$), and CSK ($P = 0.01$). In kidney cancer, expression of EGFR ($P = 0.04$) gene had association with the predicted response. In lung cancer, we observed significant associations for CDC42 ($P = 0.04$), EGFR ($P = 3 \times 10^{-5}$), and PRKAR2A ($P = 0.01$) genes. However, for bladder and pancreatic cancers, we did not observe any significant associations.

3.4 Discussion

We proposed MOLI, a Multi-Omics Late Integration method based on deep neural networks to predict drug response. MOLI integrates somatic mutation, CNA, and gene expression data and predicts the drug responses. To the best of our knowledge, MOLI is the first end-to-end method for multi-omics late integration with deep neural networks that utilizes a combined objective function. Our experiments showed that MOLI with its combined objective function can achieve better performance than single-omics and early integration multi-omics methods based on deep neural networks. We also observed that transfer learning for targeted drugs improves the prediction performance compared to drug-specific inputs. To the best of our knowledge, this is the first method to use transfer learning with a pan-drug approach for targeted drugs. Finally, we analyzed MOLI's predictions for drugs targeting the EGFR pathway on breast, kidney, lung, and prostate cancer patients in TCGA. We showed that MOLI's predictions have statistically significant associations with the level of expression for some of the genes in the EGFR pathway, including the EGFR gene itself, for breast, kidney, and lung cancers. We would like to point out the following directions for future research:

Although we used only somatic mutation, CNA, and gene expression data in our experiments, MOLI can be extended for integrating other omics data types. For example, proteomics data can be a good candidate because it has been shown to be a contributing factor in pan-cancer drug response prediction [158] and is known to be in concordance with the other omics data types [159, 160]. We performed experiments on transfer learning only for the drugs that target EGFR, but this approach is also applicable for other families of targeted drugs if multi-omics data is available for external validation. Another advantage of the pan-drug approach is that there is no need to train separate pan-drug models for each EGFR inhibitor, and one model can be validated on different external datasets. In the drug-specific approach, we trained one model on Cetuximab data and another one on Erlotinib data, and could not validate them on each other's external validation data. However, in the pan-drug approach, we trained one model for all of the EGFR inhibitors and validated it on both Cetuximab and Erlotinib data.

While we studied only the triplet loss for optimizing the concatenated representation, we note that this loss function can be replaced by other similar losses such as the contrastive loss function which was used in the Siamese network [161]. We trained separate MOLI models for different drugs, but it is an interesting direction for future research to utilize multi-task learning [162] and predict the outcome for multiple drugs at the same time. Unlike areas such as medical imaging, transfer learning is yet to be explored in genomics, especially cancer genomics [163]. While in this thesis we explored transferring related samples (also known as instance-transfer), other aspects of transfer learning such as relational-knowledge-transfer [43] should be explored in the future.

In all of the experiments and utilized datasets, we used pan-cancer inputs. The advantage of using pan-cancer multi-omics input is that it can address, to some degree, the challenge of intertumor heterogeneity [164]. However, these datasets are not suitable for addressing intratumor heterogeneity, which would require other resources such as single cell data. Geeleher et al. [19] showed that training on non-breast cancer cell lines in addition to breast cancer cell lines leads to improved prediction accuracy. However, predictions were tested only on breast cancer clinical trial cohort data and only for Docetaxel, a primary treatment for breast cancer. Because some drug-cancer event associations are specific to the tissue of origin and are less detectable in pan-cancer settings [13], we believe that further research in this area is required to study the performance of pan-cancer versus cancer-specific training data for a more diverse range of cancer types and for more drugs. We would like to point out the following limitations of this study:

1. The datasets used were from different resources were not in the same format and required substantial preprocessing and standardization. For example, different studies used different pipelines to detect CNA and reported different estimates of copy number which could not be compared directly. A similar issue was also observed for the drug response. While the GDSCv1 cell lines used IC50 as the response measure, the majority of datasets used other metrics to measure the response. For example, the PDX dataset used tumor volume based on RECIST criteria to define responders and non-responders. Therefore, lack of standardization on both the input and the output side adds extra challenges to the drug response prediction task.

2. In this study we focused on monotherapy and did not explore the effect of the combination of drugs.

3. We did not discriminate between driver and passenger events in the somatic mutation and CNA data and treated all of them similarly. However, in reality, the majority of these genomic alterations seem to have no impact on cancer development [165] and might appear just by chance. Therefore, in future work, we plan to use another format for these data types to distinguish between potential driver and passenger events.

4. All of the datasets used suffered from severely unbalanced class distributions, since the number of responders was much smaller than the number of non-responders. We addressed this problem by oversampling the minority class. However, this approach often causes overfitting particularly for deep neural networks with many parameters. We reduced overfitting with strong regularization such as high dropout rate and weight decay. Moreover, using triplets as input of the network increased the number of samples and led to a more stable network, due to the large number of different combinations for triplets.

5. As a first investigation of late integration, we did not consider interactions between genes in different omics data types in MOLI or the compared baselines. In reality, genes do not function in isolation and work in biological networks and interact with each other. Recently, [166] have shown that incorporating biological domain knowledge from the Gene Ontology leads to more interpretable neural networks with performance

comparable to those of purely data-driven neural networks. Therefore, incorporating domain expert knowledge to multi-omics late integration via deep neural networks is a promising future direction.

Chapter 4

Input and Output Space Adaptation

This chapter is adapted based on a published article [61] under license CC BY-NC.

4.1 Problem definition

Given a labeled source domain DM_S with a learning task $\mathbb{T}_S$ and a labeled target domain DM_T with a learning task $\mathbb{T}_T$, where $\mathbb{T}_T \neq \mathbb{T}_S$, and $p(X_T) \neq p(X_S)$, where $X_S, X_T \in \mathbf{X}$, we assume that the source and the target domains are not the same due to different probability distributions. The goal of Adversarial Inductive Transfer Learning (AITL) is to utilize the source and target domains and their tasks in order to improve the learning of $F_T(.)$ on DM_T.

In the area of pharmacogenomics, the source domain is the gene expression data obtained from the cell lines, and the source task is to predict the drug response in the form of IC50 values. The target domain consists of gene expression data obtained from patients, and the target task is to predict drug response in a different form – often change in the size of tumor after receiving the drug. In this setting, $p(X_T) \neq p(X_S)$ because cell lines are different from patients even with the same set of genes.

Additionally, $Y_T \neq Y_S$ because for the target task $Y_T \in \{0, 1\}$, drug response in patients is a binary outcome, but for the source task $Y_S \in R^+$, drug response in cell lines is a continuous outcome.

4.2 AITL: Adversarial Inductive Transfer Learning

Our proposed AITL method [61] takes input data from the source and target domains, and achieves the following three objectives: first, it makes predictions for the target domain using both of the input domains and their corresponding tasks, second, it addresses the discrepancy in the output space between the source and target tasks, and third, it addresses the discrepancy in the input space. AITL is a neural network consisting of four components:

- The feature extractor receives the input data from the source and target domains and extracts salient features, which are then sent to the multi-task subnetwork component.
- The multi-task subnetwork takes the extracted features of source and target samples and maps them to their corresponding labels and makes predictions for them. This component has a shared layer and two task-specific towers for regression (source task) and classification (target task). Therefore, by training the multi-task subnetwork on the source and target samples, it addresses the small sample size challenge in the target domain. In addition, it also addresses the discrepancy in the output space by assigning cross-domain labels (binary labels in this case) to the source samples (for which only continuous labels are available) using its classification tower.
- The global discriminator receives extracted features of source and target samples and predicts if an input sample is from the source or the target domain. To address the discrepancy in the input space, these features should be domain-invariant so that the global discriminator cannot predict their domain labels accurately. This goal is achieved by adversarial learning.
- The class-wise discriminators further reduce the discrepancy in the input space by adversarial learning at the level of the different classes, i.e., extracted features of source and target samples from the same class go to the discriminator for that class and this discriminator should not be able to predict if an input sample from a given class is from the source or the target domain.

The AITL objective function consists of a classification loss, a regression loss, and global and class-wise discriminator adversarial losses and is optimized end-to-end. An overview of the proposed method is presented in figure 4.1.

4.2.1 Feature extractor

To learn salient features in lower dimensions for the input data, we design a feature extractor component. The feature extractor is a one-layer fully-connected subnetwork with batch normalization and the Rectified Linear Unit (ReLU) activation function that receives both the source and target samples as input. We denote the feature extractor as $f(.)$:

$$Z_i = f(X_i), i \in \{S, T\} \tag{4.1}$$

where Z denotes the extracted features for input X which is from either the source (S) or the target (T) domain. In our driving application, the feature extractor learns features for the cell line and patient data.

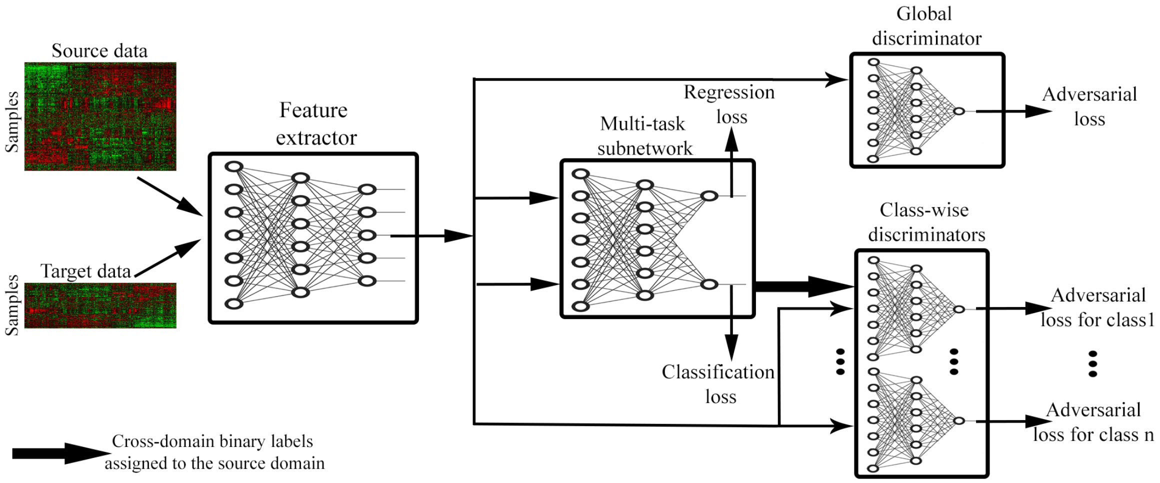

Figure 4.1: **Schematic overview of AITL** First, the feature extractor receives source and target samples and learns feature for them. Then, the multi-task subnetwork uses these features to make predictions for the source and target samples and also assigns cross-domain labels to the source samples. The multi-task subnetwork addresses the discrepancy in the output space. Finally, to address the input space discrepancy, global and class-wise discriminators receive the extracted features and regularize the feature extractor to learn domain-invariant features.

4.2.2 Multi-task subnetwork

After extracting features of the input samples, we want to use these learned features to 1) make predictions for target samples, and 2) address the discrepancy between the source and the target domain in the output space. To achieve these goals, a multi-task subnetwork with a shared layer $g(.)$ and two task-specific towers $M_S(.)$ and $M_T(.)$ is designed, where M_S is for regression (the source task) and M_T is for classification (the target task):

$$\overline{Y_i} = M_i(g(Z_i)), i \in \{S, T\} \tag{4.2}$$

The performance of the multi-task subnetwork component is evaluated based on a binary-cross entropy loss for the classification task on the target samples and a mean squared loss for the regression task on the source samples:

$$L_{BCE}(X_T, Y_T, f, g, M_T) = -\sum_{(x_t,y_t)\sim(X_T,Y_T)} [y_t \log \overline{y_t} + (1 - y_t)\log(1 - \overline{y_t})] \tag{4.3}$$

$$L_{MSE}(X_S, Y_S, f, g, M_S) = 1/n_S \sum_{(x_s,y_s)\sim(X_S,Y_S)} (\overline{y_s} - y_s)^2 \tag{4.4}$$

Where Y_S and Y_T are the true labels of the source and the target samples, respectively, n_S denotes the number of samples in the source domain, and L_{BCE} and L_{MSE} are the corresponding losses for the target and the source domains, respectively. The multi-task subnetwork component outputs 1) the predicted labels for the target samples, and 2) the assigned cross-domain labels for the source samples. The classification tower in the multi-task subnetwork makes predictions for the source samples and assigns binary labels (responder or non-responder) because such labels do not exist for the source samples. Therefore, the multi-task subnetwork adapts the output space of the source and the target domains by assigning cross-domain labels to the source domain. The multi-task subnetwork has a shared fully-connected layer with batch normalization and the ReLU activation function. The regression tower has two layers with batch normalization and the ReLU activation function. The classification tower also has two fully connected layer with batch normalization and the ReLU activation function in the first layer and the Sigmoid activation function in the second layer. In our driving application the multi-task subnetwork predicts IC50 values for the cell lines and the binary response outcome for the patients. Moreover, it also assigns binary labels to the cell lines which is similar to those of the patients.

4.2.3 Global discriminator

The goal of this component is to address the discrepancy in the input space by adversarial learning of domain-invariant features. To achieve this goal, a discriminator receives source and target extracted features from the feature extractor and classifies them into their corresponding domain. The feature extractor should learn domain-invariant features to fool the global discriminator. In our driving application the global discriminator should not be able to recognize if the extracted features of a sample are from a cell line or a patient. This discriminator is a one-layer subnetwork with the Sigmoid activation function denoted by $D_G(.)$. The adversarial loss for $D_G(.)$ is as follows:

$$L_{advD_G}(X_S, X_T, D_G) = -\sum_{x_s \sim X_S} [\log D_G(f(x_s))] - \sum_{x_t \sim X_T} [\log(1 - D_G(f(x_t)))] \tag{4.5}$$

4.2.4 Class-wise discriminators

With cross-domain binary labels available for the source domain, AITL further reduces the discrepancy between the input domains via class-wise discriminators. The goal is to learn domain-invariant features with respect to specific class labels such that they fool corresponding class-wise discriminators. Therefore, extracted features of the target samples in class i, and those of the source domain which the multi-task subnetwork assigned to class i, will go to the discriminator for class i. We denote such a class-wise discriminator as DC_i.

The adversarial loss for DC_i is as follows:

$$L_{advDC_i}(X_S, Y_S, X_T, Y_T, DC_i) = -\sum_{(x_s,y_s)\sim(X_S,Y_S)} [\log DC_i(f(x_s))] - \sum_{(x_t,y_t)\sim(X_T,Y_T)} [\log(1 - DC_i(f(x_t)))] \tag{4.6}$$

In our driving application the class-wise discriminator for the responder samples should not be able to recognize if the extracted features of a responder sample are from a cell line or a patient (similarly for a non-responder sample). Similar to the global discriminator, class-wise discriminators are also one-layer fully-connected subnetworks with the Sigmoid activation function.

4.2.5 Objective function

To optimize the entire network in an end-to-end fashion, we design the objective function as follows:

$$J = L_{BCE} + L_{MSE} + \lambda_G L_{advD_G} + \lambda_{DC} \sum_i L_{advDC_i} \tag{4.7}$$

Where, λ_G and λ_{DC} are adversarial regularization coefficients for the global and class-wise discriminators, respectively.

4.3 Experimental results

4.3.1 Datasets

In our experiments, we used the following datasets (See Table 4.1 for more detail):

- The Genomics of Drug Sensitivity in Cancer (GDSCv1) cell lines dataset, consisting of a thousand cell lines from different cancer types, screened with 265 targeted and chemotherapy drugs. [13]

- The Patient-Derived Xenograft Encyclopedia (PDXE) dataset, consisting of more than 300 PDX samples for different cancer types, screened with 34 targeted and chemotherapy drugs. [11]

- TCGA [25] containing a total number of 117 patients with diverse cancer types, treated with Cisplatin, Docetaxel, or Paclitaxel [26].

- Patient datasets from nine clinical trial cohorts containing a total number of 491 patients with diverse cancer types, treated with Bortezomib [167, 168], Cisplatin [169, 170], Docetaxel [171, 172, 173], or Paclitaxel [171, 174, 175]. For the categorical measures of the drug response such as response evaluation criteria in solid tumors

Table 4.1: Characteristics of the datasets utilized in AITL method

Dataset	**Resource**	**Drug**	**Usage**	**Sample Size**
GSE55145 [167]	clinical trial	Bortezomib	target	67
GSE9782-GPL96 [168]	clinical trial	Bortezomib	target	169
GDSCv1 [13]	cell line	Bortezomib	source	391
GSE18864 [169]	clinical trial	Cisplatin	target	24
GSE23554 [170]	clinical trial	Cisplatin	target	28
TCGA [26]	patient	Cisplatin	target	66
GDSCv1 [13]	cell line	Cisplatin	source	829
GSE25065 [171]	clinical trial	Docetaxel	target	49
GSE28796 [172]	clinical trial	Docetaxel	target	12
GSE6434 [173]	clinical trial	Docetaxel	target	24
TCGA [26]	patient	Docetaxel	target	16
GDSCv1 [13]	cell line	Docetaxel	source	829
GSE15622 [175]	clinical trial	Paclitaxel	target	20
GSE22513 [174]	clinical trial	Paclitaxel	target	14
GSE25065 [171]	clinical trial	Paclitaxel	target	84
PDX [11]	animal (mouse)	Paclitaxel	target	43
TCGA [26]	patient	Paclitaxel	target	35
GDSCv1 [13]	cell line	Paclitaxel	source	389

(RECIST), we consider complete response and partial response as responder (class 1) and consider stable disease and progressive disease as non-responder (class 0).

- the Cancer Genome Atlas (TCGA) cohorts including, breast (BRCA), prostate (PRAD), lung (LUAD), kidney (KIRP), and bladder (BLCA) cancers that do not have the drug response outcome.

The GDSCv1 dataset was used as the source domain, and all the other datasets were used as the target domain. For the GDSCv1 dataset, raw gene expression data were downloaded from ArrayExpress (E-MTAB-3610) and response outcomes from `https:/www.cancerrxgene.org` release 7.0. Gene expression data of TCGA patients were downloaded from the Firehose Broad GDAC (version published on 28.01.2016) and the response outcome was obtained from [26]. Patient datasets from clinical trials were obtained from the Gene Expression Omnibus (GEO), and the PDX dataset was obtained from the supplementary material of [11]. For each drug, we selected those patient datasets that applied a comparable measure of the drug response. For preprocessing, the same procedure was adopted as described in the supplementary material of [40] for the raw gene expression data (normalized and z-score transformed) and the drug response data. After the preprocessing, source and target domains had the same number of genes.

4.3.2 Experimental design

We designed our experiments to answer the following four questions:

1. Does AITL outperform baselines that are trained only on cell lines and then evaluated on patients (without transfer learning)? To answer this question, we compared AITL against [19] and MOLI [40] which are state-of-the-art methods of drug response prediction that do not perform domain adaptation. The [19] is non-deep learning method based on ridge regression and MOLI is a deep learning-based method. Both of them were originally proposed for pharmacogenomics.

2. Does AITL outperform baselines that adopt adversarial transductive transfer learning and non-deep learning adaptation (without adaptation of the output space)? To answer this question, we compared AITL against ADDA [58] and the method of [57], state-of-the-art methods of adversarial transductive transfer learning with global and class-wise discriminators, respectively. For the non-deep learning baseline, we compared AITL to PRECISE [41], a non-deep learning domain adaptation method specifically designed for pharmacogenomics.

3. Does AITL outperform a baseline for inductive transfer learning? To answer this question, we compared AITL against ProtoNet [93] which is a state-of-the-art inductive transfer learning method for small numbers of examples per class.

4. Finally, do the predicted responses by AITL for TCGA patients have associations with the targets of the studied drug?

Based on the availability of patient/PDX datasets for a drug, we experimented with four different drugs: Bortezomib, Cisplatin, Docetaxel, and Paclitaxel. It is important to note that these drugs have different mechanisms and are being prescribed for different cancers. For example, Docetaxel is a chemotherapy drug mostly known for treating breast cancer patients [173], while Bortezomib is a targeted drug mostly used for multiple myeloma patients [167]. Therefore, the datasets we have selected cover different types of anti-cancer drugs.

In addition to the experimental comparison against published methods, we also performed an ablation study to investigate the impact of the different AITL components separately. AITL$-AD$ denotes a version of AITL without the adversarial adaptation components, which means the network only contains the multi-task subnetwork. AITL$-D_G$ denotes a version of AITL without the global discriminator, which means the network only employs the multi-task subnetwork and class-wise discriminators. AITL$-DC$ denotes a version of AITL without the class-wise discriminators, which means the network only contains the multi-task subnetwork and the global discriminator.

All of the baselines were trained on the same data, tested on patients/PDX for these drugs, and eventually compared to AITL in terms of prediction Area Under the Receiver Operating

Characteristic curve (AUROC) and the Area Under the Precision-Recall curve (AUPR). Since the majority of the studied baselines cannot use the continuous log(IC50) values in the source domain, binarized log(IC50) labels provided by [13] using the Waterfall approach [18] were used to train them. Finally, for the minimax optimization, a gradient reversal layer was employed by AITL and the adversarial baselines [176] which is a well-established approach in domain adaptation [46, 56, 84].

We performed 3-fold cross validation in the experiments to tune the hyper-parameters of AITL and the baselines based on the AUROC. Two folds of the source samples were used for training and the third fold for validation, similarly, two folds of the target samples were used for training and validation, and the third one for the test. The reported results refer to the average and standard deviation over the test folds.

The hyper-parameters tuned for AITL were the number of nodes in the hidden layers, learning rates, mini-batch size, the dropout rate, number of epochs, and the regularization coefficients. We considered different ranges for each hyper-parameter. The selected hyper-parameters for AITL are as follows:

Bortezomib:

1024 (number of nodes in the layer of the feature extractor), 1024 (number of nodes in the shared layer of the multi-task subnetwork), 1024 (number of nodes in the hidden layer of the regression tower), 0.0005 (learning rate), 0.2 and 0.4 (regularization for global and class-wise discriminators), 16 and 16 (mini-batch size for the source and target domains), 0.4 (dropout rate), 10 (epoch).

Cisplatin:

512 (number of nodes in the hidden layer of the feature extractor), 16 (number of nodes in the shared layer of the multi-task subnetwork), 16 (number of nodes in the hidden layer of the regression tower), 0.05 (learning rate), 0.3 and 0.3 (regularization for global and class-wise discriminators), 32 and 8 (mini-batch size for the source and target domains), 0.15 (dropout rate), 25 (epoch).

Docetaxel:

256 (number of nodes in the hidden layer of the feature extractor), 512 (number of nodes in the shared layer of the multi-task subnetwork), 512 (number of nodes in the hidden layer of the regression tower), 0.0001 (learning rate), 0.8 and 0.6 (regularization for global and class-wise discriminators), 32 and 32 (mini-batch size for the source and target domains), 0.5 (dropout rate), 35 (epoch).

Paclitaxel:

1024 number of nodes in the layer of the feature extractor), 1024 (number of nodes in the shared layer of the multi-task subnetwork), 1024 (number of nodes in the hidden layer of the regression tower), 0.0001 (learning rate), 0.9 and 0.3 (regularization for global and class-wise discriminators), 32 and 32 (mini-batch size for the source and target domains), 0.5 (dropout rate), 20 (epoch).

Table 4.2: Performance of AITL and the baselines in terms of prediction AUROC

Method/Drug	**Bortezomib**	**Cisplatin**	**Docetaxel**	**Paclitaxel**
Geeleher et al. [19]	0.48	0.58	0.55	0.53
MOLI [40]	0.57	0.54	0.54	0.53
PRECISE [41]	0.54	0.59	0.52	0.56
Chen et al. [57]	0.54±0.07	0.60±0.14	0.52±0.02	0.58±0.04
ADDA [58]	0.51±0.06	0.56±0.06	0.48±0.06	did not converge
ProtoNet [93]	0.49±0.01	0.40±0.003	0.40±0.01	did not converge
AITL$-AD$	0.69±0.03	0.57±0.03	0.57±0.05	0.58±0.01
AITL$-D_G$	0.69±0.04	0.62±0.1	0.48±0.03	**0.62±0.02**
AITL$-D_C$	0.69±0.03	0.54±0.1	0.59±0.07	0.59±0.03
AITL	**0.74±0.02**	**0.66±0.02**	**0.64±0.04**	0.61±0.04

Finally, each network was re-trained on the selected settings using the train and validation data together for each drug. We used Adagrad for optimizing the parameters of AITL and the baselines [157] implemented in the PyTorch framework, except for [19] which was implemented in R. We used the author's implementations for [19], MOLI, PRECISE, and ProtoNet. For ADDA, we used an existing implementation from `https://github.com/jvanvugt/pytorch-domain-adaptation`, and we implemented the method of [57] from scratch.

4.3.3 Results

Table 4.2 and Figure 4.2 report the performance of AITL and the baselines in terms of AUROC and AUPR, respectively. To answer the first experimental question, AITL was compared to the baselines which do not use any adaptation (neither the input nor the output space), i.e. [19] and MOLI [40], and AITL demonstrated a better performance in both AUROC and AUPR for all of the studied drugs. This indicates that addressing the discrepancies in the input and output spaces leads to better performance compared to training a model on the source domain and testing it on the target domain. To answer the second experimental question, AITL was compared to state-of-the-art methods of adversarial and non-deep learning transductive transfer learning, i.e. ADDA [58], the method of [57], and PRECISE [41], which address the discrepancy only in the input space. AITL achieved significantly better performance in AUROC for all of the drugs and for three out of four drugs in AUPR (the results of [57] for Cisplatin were very competitive with AITL). This indicates that addressing the discrepancies in the both spaces outperforms addressing only the input space discrepancy. Finally, to answer the last experimental question, AITL was compared to ProtoNet [93] – a representative of inductive transfer learning with input space adaptation via few-shot learning. AITL outperformed this method in all of the metrics for all of the drugs.

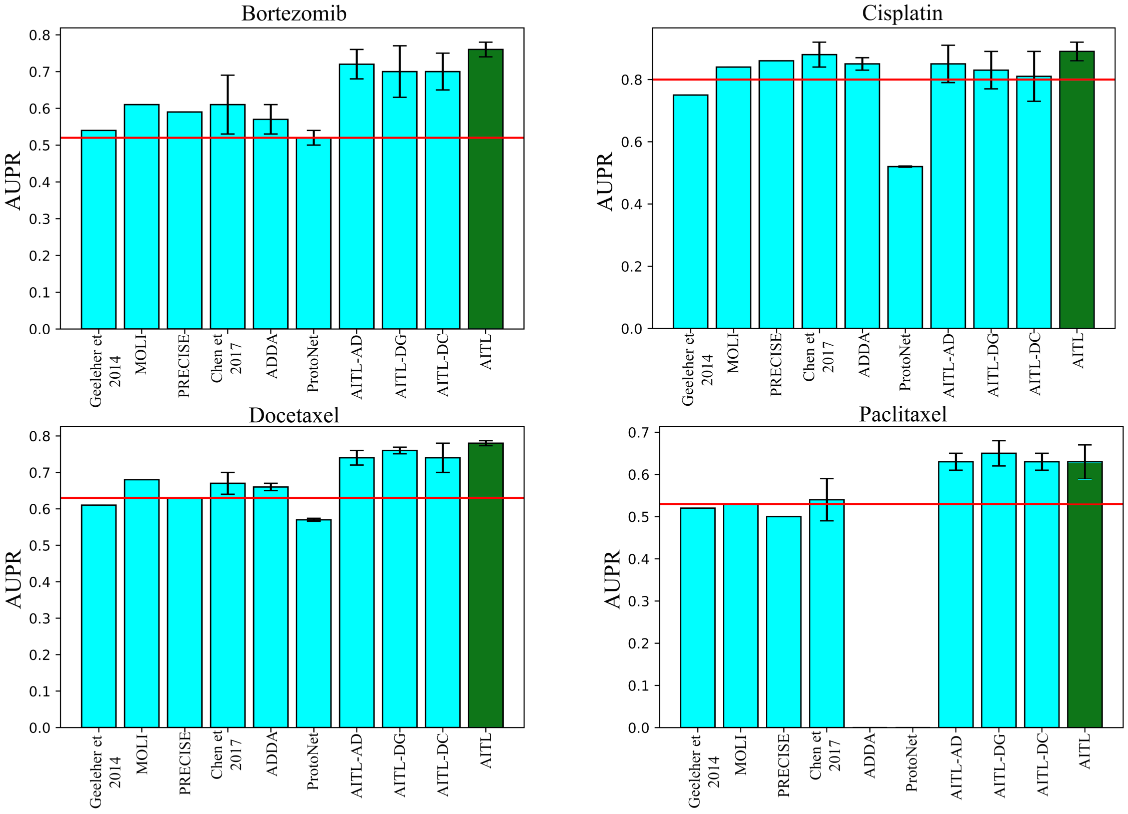

Figure 4.2: Performance of AITL and the baselines in terms of prediction AUPR

We note that the methods of drug response prediction without adaptation, namely [19] and MOLI, outperformed the method of inductive transfer learning based on few-shot learning (ProtoNet). Moreover, these two methods also showed a very competitive performance compared to the methods of transductive transfer learning (ADDA, the method of [57], and PRECISE). For Paclitaxel, ADDA did not converge in the first step (training a classifier on the source domain), which was also observed in another study [40]. ProtoNet also did not converge for this drug.

We observed that AITL, when all of its components are used together, outperforms additional baselines with modified versions of AITL. This indicates the importance of both input and output space adaptation. The only exception was for the drug Paclitaxel, where AITL$-D_G$ outperforms AITL. We believe the reason for this is that this drug has the most heterogeneous target domain (see Table 3.1), and therefore the global discriminator component of AITL causes a minor decrease in the performance. Our ablation study showed that the global discriminator and the class-wise discriminators are not redundant and, in fact, each of them plays a unique constructive role in learning the domain-invariant representation. All these results indicate that addressing the discrepancies in the input and output spaces between the source and target domains, via the AITL method, leads to a better prediction performance.

AITL predictions for TCGA patients have significant associations with target genes

To answer the last experimental question, we applied AITL models (trained on Docetaxel, Bortezomib, and Paclitaxel) to the gene expression data without known drug response from TCGA (breast, prostate, lung, kidney, and bladder cancers) and predicted the response for these patients separately. Based on the corrected p-values obtained from multiple linear regression, there are a number of statistically significant associations between the target genes of the studied drugs and the responses predicted by AITL.

For example, in breast cancer, we observed statistically significant associations in MAP4 ($P < 1 \times 10^{-10}$) for Doxetaxel, BLC2 ($P = 1.7 \times 10^{-4}$) for Paclitaxel, and PSMA4 ($P = 4.7 \times 10^{-6}$) for Bortezomib. In prostate cancer, we observed statistically significant associations in MAP2 ($P < 1 \times 10^{-10}$) for Docetaxel, TUBB ($P < 1 \times 10^{-10}$) for Paclitaxel, and RELA ($P = 2.2 \times 10^{-4}$) for Bortezomib. For bladder cancer, NR1I2 ($P = 0.04$) for Docetaxel, MAP4 ($P < 1 \times 10^{-10}$) for Paclitaxel, and PSMA4 ($P = 0.001$) for Bortezomib were significant. In kidney cancer, BLC2 ($P = 5.4 \times 10^{-8}$) for Docetaxel, MAPT ($P < 1 \times 10^{-10}$) for Paclitaxel, and PSMD2 ($P = 1 \times 10^{-5}$) for Bortezomib were significant. Finally, in lung cancer, MAP4 ($P < 1 \times 10^{-10}$) for Docetaxel, TUBB ($P < 1 \times 10^{-10}$) for Paclitaxel, and RELA ($P < 1 \times 10^{-10}$) for Bortezomib were significant.

4.4 Discussion

The obtained results from the association study are in concordance with previous studies. For example, we observed that Microtubule-Associated Proteins (MAPs) were significant for Docetaxel and Paclitaxel in the studied cancers which aligns with previous research on this family of proteins [177, 178, 179]. For Bortezomib, we observed significant associations for different proteasome subunits such as subunit alpha (PSMA) and beta (PSMB). These subunits have been shown to be key players across different cancers [180, 181, 182]. We also observed significant associations for RELA (also known as Transcription Factor p65) in all of the studied cancers which aligns with its oncogenic role across different cancers [183], and moreover, with its reported associations with Bortezomib in breast cancer [184], prostate cancer [185], and lung cancer [186].

AITL can be quite sensitive to the selection of hyper-parameters, especially to the learning rate, number of training epochs, and the dropout rate. We observe that lower learning rates tend to yield better performance for the AITL models. In addition, a smaller number of training epochs also tends to produce better results, which makes sense because we have limited amounts of training data, and training with higher epochs would overfit the model. Lastly, we observe that dropout rates of around $0.4 - 0.5$ result in the highest performing AITL models.

To our surprise, ProtoNet, and ADDA could not outperform [19], MOLI, and PRECISE. For ProtoNet, this may be due to the depth of the backbone network. A recent study has shown that a deeper backbone improves ProtoNet performance significantly in image classification [94]. However, in pharmacogenomics, employing a deep backbone is not realistic because of the much smaller sample size compared to an image classification application. Another limitation for ProtoNet is the imbalanced number of training examples in different classes in pharmacogenomics datasets. Specifically, the number of examples per class in the training episodes is limited to the number of samples of the minority class as ProtoNet requires the same number of examples from each class. For ADDA, this lower performance may be due to the lack of end-to-end training of the classifier along with the global discriminator of this method. The reason is that end-to-end training of the classifier along with the discriminators improved the performance of the second adversarial baseline [57] in AUROC and AUPR compared to ADDA. Moreover, the method of [57] also showed a relatively better performance in AUPR compared to [19] and MOLI.

In pharmacogenomics, patient datasets with drug response are small or not publicly available due to privacy and/or data sharing issues. We believe including more patient samples and more drugs will increase generalization capability. In addition, recent pharmacogenomics studies have shown that using multi-omics data works better than using only gene expression [40]. In this work, we did not consider genomic data other than gene expression data due to the lack of patient samples with multi-omics data and drug response data publicly available; however, in principle, AITL can be extended to work with such data by adding separate feature extractors for each omics data type. This approach is particularly crucial if the different data types have different dimensionalities. Last but not least, we used pharmacogenomics as our motivating application for this new problem of transfer learning, but we believe that AITL can also be employed in other applications.

For example, in slow progressing cancers such as prostate cancer, large patient datasets with gene expression and short-term clinical data (source domain) are available, however, patient datasets with long-term clinical data (target domain) are small. AITL may be beneficial to learn a model to predict these long-term clinical labels using the source domain and its short-term clinical labels [187]. Finally, although we designed the multi-task subnetwork for a regression task on the source domain and a classification task on the target domain, in principle, AITL can easily be modified to incorporate different types of outputs.

We observed that predictions for TCGA samples tend to have a low variance. We believe the reason for that is first, we created target domains by pooling together samples from different patient datasets treated with the same drug; however, in reality each dataset has its own discrepancies compared to the other datasets within each target domain. Second, we trained the model using pan-cancer cell lines, however, the patient samples were cancer specific due to the lack of pan-cancer patient data with drug response which makes the trained model less applicable for pan-cancer resources such as TCGA.

For future research directions, we believe that the TCGA dataset consisting of gene expression data of more than 12,000 patients (without drug response outcome) can be incorporated in an unsupervised transfer learning setting to learn better features that are domain-invariant between cell lines and cancer patients. The advantage of this approach is that we can keep the valuable patient datasets with drug response as an independent test set and not use it for training/validation. Another possible future direction is to incorporate domain-expert knowledge into the structure of the model. A recent study has shown that such a structure improves the drug response prediction performance on cell line datasets and, more importantly, provides an explainable model as well [59].

Chapter 5

Out-of-distribution Generalization

This chapter is adapted based on a published article [188] under license CC BY-NC.

5.1 Problem definition

A domain DM is defined by a raw input space $\mathbb{X}$, a probability distribution $p(X)$ and a corresponding dataset $X = \{x_1, x_2, ..., x_n\}$ with $x_i \in \mathbb{X}$. A task $\mathbb{T} = \{Y, \mathbb{F}(.)\}$ is associated with $DM = \{X, p(X)\}$ and is defined by a label space $Y \in \mathbb{Y}$ and a predictive function $\mathbb{F}(.)$ which is learned from training data $(X, Y) \in \mathbb{X} \times \mathbb{Y}$. In our case, $Y \in [0, 1]$, which makes drug response prediction a regression problem.

Given multiple labeled and unlabeled source domains denoted by $DM^L = \{DM_i^l\}_{i=1}^{n_l}$ and $DM^U = \{DM_j^u\}_{j=1}^{n_u}$, the goal is to learn the predictive function $\mathbb{F}(.)$ which is implemented through a neural network. $\mathbb{F}(.)$ consists of a shared (across all source domains) feature extractor $F_\theta(X)$ parameterized by θ, which maps X to latent features Z, and domain-specific predictors $G_{\phi_i}^i$ parametrized by ϕ_i, which takes Z_i (the extracted features of DM_i) as input and makes predictions (of the drug response) $\overline{Y_i}$ for this source domain. θ and ${\phi_i}_{i=1}^{n_l}$ are being optimized using an objective function $J(DM^L, DM^U, \theta, \{\phi_i\}_{i=1}^{n_l}) = l(DM^L, \theta, \{\phi_i\}_{i=1}^{n_l}) + \Omega(DM^L, DM^U, \theta, \{\phi_i\}_{i=1}^{n_l})$, with a supervised loss $l(.)$ and some regularization terms $\Omega(.)$.

In drug response prediction, we have access to labeled source domains such as cell line datasets and unlabeled source domains such as cancer patients in the Cancer Genome Atlas (TCGA) and the goal is to learn a model that makes accurate predictions on patients, Patient-derived Xenografts (PDXs), or other cell lines as target domains that it may see during deployment. This is similar to out-of-distribution generalization (also known as domain generalization), where the goal is to optimize parameters of the model (θ and $\{\phi_i\}_{i=1}^{n_l}$) in order to make the model generalizable and predictive of unseen domains. Out-of-distribution generalization assumes that there exists a d-dimensional latent feature space $Z \in \mathbb{R}^d$ that is invariant, predictive, and generalizable to seen and unseen domains in this given space.

5.2 Velodrome

The proposed Velodrome method [188] takes gene expression and AAC of cell line datasets (Genomics of Drug Sensitivity in Cancer– GDSCv2 and The Cancer Therapeutics Response Portal–CTRPv2) as well as gene expression of patients without drug response (TCGA dataset) and learns a predictive and generalizable representation. To achieve this, Velodrome employs a shared feature extractor, which takes the gene expression of CTRPv2 and GDSCv2 samples and maps them to a shared feature space, and domain-specific predictors (e.g. one for CTRPv2 and one for GDSCv2), which take the feature representation of the gene expression and predicts the drug response.

The parameters are optimized using a novel objective function consisting of three loss components. 1) a standard supervised loss to make the representation predictive of drug response, 2) a consistency loss to exploit unlabeled samples in learning the representation, and 3) an alignment loss to make the representation generalizable. The idea of the standard supervised loss is to make the representation predictive of the drug response via a mean-squared loss.

To incorporate unlabeled patient samples, we add a consistency loss. The idea is to first extract features from patient samples using the feature extractor and then assign pseudo-labels to them by utilizing the predictors associated with CTRPv2 and GDSCv2. The consistency loss takes the pseudo-labels (i.e., predictions) from the predictors and regularizes the parameters of the feature extractor and the predictors by the l2distance between the predictions of CTRPv2 predictor and those of the GDSCv2 predictor.

Finally, to make the feature representation generalizable, we add an alignment loss that regularizes the parameters of the feature extractor. This alignment loss takes the extracted features of any two input domains (eg., CTRPv2 and TCGA or CTRPv2 and GDSCv2) and minimizes the difference between the covariance matrices of those domains. Figure 5.1 illustrates the schematic overview of the Velodrome method.

5.2.1 Shared feature extractor

To map the raw input gene expression data to the latent space, Velodrome utilizes a feature extractor which is shared across all labeled and unlabeled source domains:

$$Z_i^j = F_\theta(X_i^j), j \in \{l, u\}, i \in DM_i^j, \tag{5.1}$$

where, Z_i^j denotes the features extracted by the feature extractor $F_\theta(.)$ from X_i^j, the samples obtained from the $i-th$ domain of type j (labeled or unlabeled). These extracted (latent) features will be provided as input to the domain-specific predictors.

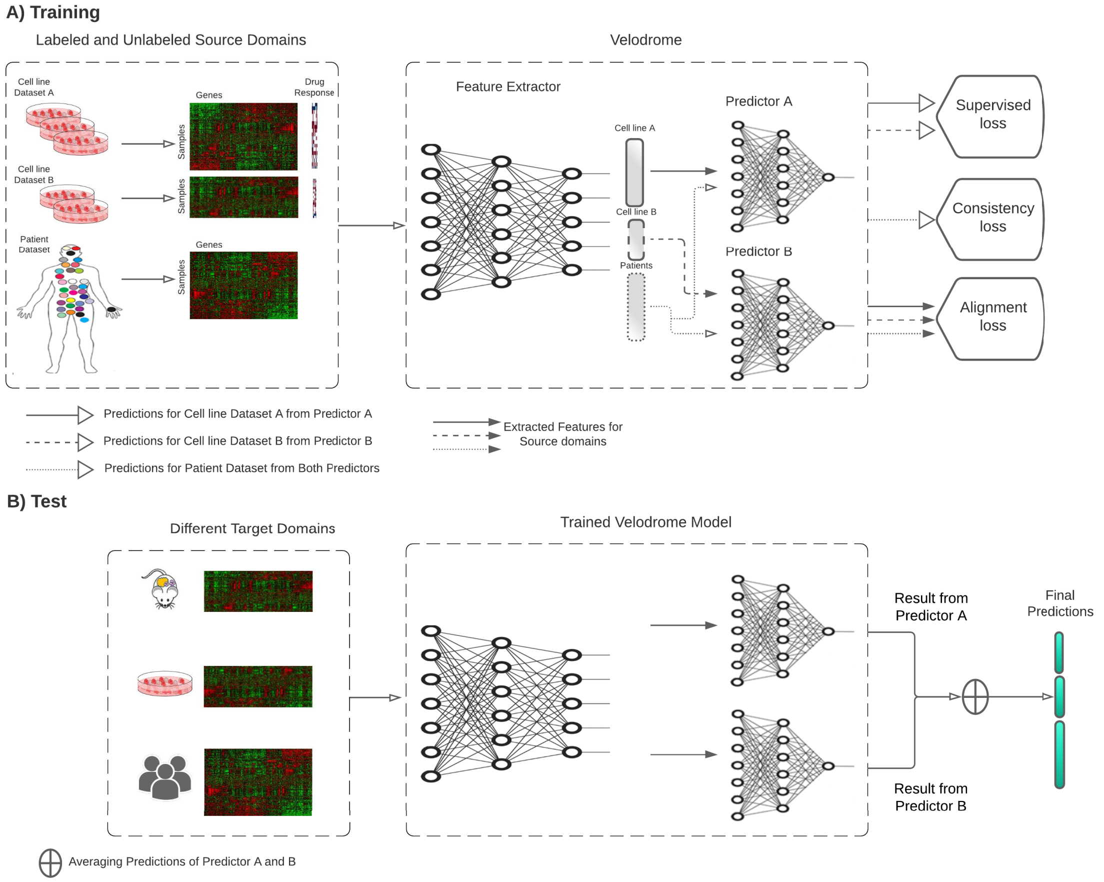

Figure 5.1: **The schematic overview of the Velodrome method with three source domains (two labeled and one unlabeled).** (A) At training time, the feature extractor receives data from different source domains and extracts high-level abstract features. The extracted features of each labeled domain (cell line dataset) are input to the corresponding domain-specific predictor. Predictions are used to optimize the parameters of the predictors and the feature extractor via a standard supervised loss function. The extracted features of the unlabeled domain (patient dataset) are input to both predictors, and the predictions are used to optimize the parameters of predictors and the feature extractor via a consistency loss function. The extracted features of all source domains are used to optimize the parameters of the feature extractor via an alignment loss function. (B) At test time, the trained Velodrome model receives samples from different target domains, extracts features and makes predictions using the trained predictors. The predictions are then averaged to generate the final predictions for each sample.

5.2.2 Domain-specific predictors

To make predictions for the samples in the source domains, Velodrome utilizes n_l domain-specific predictors, meaning the number of domain-specific predictors that Velodrome utilizes is the same as the number of labeled source domains. These predictors are formulated as follows:

$$\overline{Y_i^l} = G_{\phi_i}^l(Z_i^l), \tag{5.2}$$

where, Y_i^l denotes the predictions for the $i-th$ labeled source domain obtained from predictor $G_{\phi_i}^l(.)$ associated with the $i-th$ labeled source domain and parameterized by ϕ_i. These predictions will be utilized to optimize the parameters of the feature extractor and the $i-th$ predictor.

5.2.3 Supervised loss

To make the extracted latent features predictive of the drug response, Velodrome utilizes a standard supervised loss as follows:

$$l(DM^l, \theta, \{\phi_i\}_{i=1}^{n_l}) = 1/n_l \sum_{i=1}^{n_l} ||Y_i^l - \overline{Y_i^l}||_2^2, \tag{5.3}$$

where, $l(.)$ denotes a standard supervised loss function in the form of the mean squared error (MSE). It is important to note that the parameters of the feature extractor are optimized by the total supervised loss but the parameters of the $i-th$ predictor are optimized only by the MSE of predictions of the $i-th$ predictor.

5.2.4 Alignment loss

Optimizing the parameters of the Velodrome model using only the supervised loss is likely to lead to overfitting to the labeled source domains. Therefore, we need an additional loss function to avoid overfitting to the source domains and to make the latent representation generalizable to unseen domains. To achieve this, Velodrome utilizes the CORAL loss function that regularizes the covariance matrices across input domains and has demonstrated state-of-the-art performance for learning invariant representations in computer vision applications [49, 66]. The CORAL loss is defined as follows:

$$CORAL(DM^L, DM^U, \theta, \{\phi_i\}_{i=1}^{n_l}) = \sum_{j=1}^{n_l} \sum_{i=1}^{nu} ||C(Z_j^l) - C(Z_i^u)||_F^2, \tag{5.4}$$

where, $C(.)$ is the covariance operator which receives the extracted features of a source domain and returns the covariance matrix of those features as follows:

$$C(Z) = 1/n \sum_{i=1}^{n} (X_i - \overline{X_i})(X_i - \overline{X_i})^T, \tag{5.5}$$

where, n is the number of samples and $\overline{X}$ denotes the mean vector. Regularizing the covariance matrices across source domains ensures learning invariant feature vectors. It is important to note that the objective function of Velodrome requires a combination of supervised and alignment loss because optimizing only the alignment loss is likely to lead to a trivial “zero” solution where all domains are mapped to the same point [49].

5.2.5 Consistency loss

Aligning the extracted features of the different domains imposes a strict constraint on learning an invariant latent representation because it disregards the unique domain-specific aspects of different source domains. To alleviate this, Velodrome utilizes a consistency loss to ensure that it learns a hypothesis invariant representation, i.e. predictions across source domains are similar when using different predictors. For example, if we have two predictors $G^l_{\phi_i}$ and $G^l_{\phi_j}$, we want them to generate similar predictions for the same unlabeled source domain. This consistency loss is defined as follows:

$$CON(DM^u, \theta, \{\phi_i\}_{i=1}^{n_l}) = MSE[G^l_{\phi_i}(Z^u), G^l_{\phi_j}(Z^u)], \tag{5.6}$$

where, Z^u are extracted features for samples in a given unlabeled source domain.

5.2.6 Objective function

Putting all of the loss functions together, the objective function of Velodrome is as follows:

$$J(DM^L, DM^U, \theta, \{\phi_i\}_{i=1}^{n_l}) = l(DM^L, \theta, \{\phi_i\}_{i=1}^{n_l}) + \Omega(DM^L, DM^U, \theta, \{\phi_i\}_{i=1}^{n_l}), \tag{5.7}$$

where, the regularization function $\Omega(.)$ that we defined in the problem definition is given by:

$$\Omega(DM^L, DM^U, \theta, \{\phi_i\}_{i=1}^{n_l}) = \tag{5.8}$$

$$\lambda CON(DM^U, \theta, \{\phi_i\}_{i=1}^{n_l}) + (1-\lambda) CORAL(DM^L, DM^U, \theta, \{\phi_i\}_{i=1}^{n_l}).$$

where, λ denote the regularization coefficients for the alignment loss and consistency loss. The Velodrome regularization coefficients enables the model to have a trade-off between learning domain-invariant and hypothesis-invariant features because the alignment loss ensures learning domain-invariant features and the consistency loss ensures learning hypothesis-invariant features.

5.2.7 Velodrome at test time

For a target sample x_t,Velodrome makes prediction as follows:

$$\overline{y_t} = \sum_i w_i G_{\phi_i}(F_\theta(x_t)), \tag{5.9}$$

where, $w_i$denotes the average supervised loss for the predictions of G_{ϕ_i}, normalized via a softmax function such that $\sum_i w_i = 1$. This means the final prediction will be a result of a weighted average of all predictors, and more accurate predictors will have higher weights.

5.3 Experimental results

A method of domain generalization (out-of-distribution generalization) for drug response prediction may take pre-clinical or clinical samples during deployment. Therefore, we selected datasets and designed experiments to investigate Velodrome performance on cell lines, PDXs, and patients. Moreover, we evaluate the Velodrome performance by comparison to the state-of-the-art methods of domain generalization, domain adaptation, and semi-supervised learning.

5.3.1 Datasets

We employed the following resources for domain generalization:

- Patients without drug response: more than 1,500 samples obtained from TCGA [25] breast cancer, lung cancer, and pancreatic cancer cohorts with RNA-seq data.

- Cell lines with drug response: The Cancer Therapeutics Response Portal (CTRPv2), The Genomics of Drug Sensitivity in Cancer (GDSCv2), and The Genentech Cell Line Screening Initiative (gCSI) pan-cancer datasets with a total of more than 2000 samples with RNA-seq data and AAC as the measure of the drug response across 11 drugs (in common for the three datasets). These datasets are generated via the same drug screening assay (CellTiter Glo) and are preprocessed using the PharmacoGx package [189]. We chose AAC as the measure of drug response because it is shown to be a better metric compared to IC50 [125, 190]. Similarly, CTRPv2 has been shown to outperform GDSCv1 in training drug response predictors [190]. We focused on the following drugs for this thesis: Erlotinib, Docetaxel, Paclitaxel, and Gemcitabine. All datasets were downloaded from ORCESTRA platform [134].

- PDX samples with drug response: PDXE is a collection of more than 300 PDX samples with RNA-seq data screened with 34 drugs. We use the reported measure of response in response evaluation criteria in solid tumors (RECIST) [42] for Gemcitabine, Erlotinib, and Paclitaxel obtained from supplementary material of [11].

- Patients with drug response: 2 cancer-specific datasets with microarray data and RECIST as the measure of drug response for Docetaxel [171], Paclitaxel [171], and Erlotinib [191]. Plus, a pan-cancer dataset obtained from TCGA patients treated with Gemcitabine [26]. We use clinical annotations of the drug response for some patients which were obtained from supplementary material of [26].

All gene names were mapped to Entrez gene ids. The expression data is obtained before treatment and the response outcome after treatment. We reduced the number of genes to 2128 genes obtained from [192]. After the preprocessing, all the available datasets for each drug had the same number of genes. We focused on these TCGA datasets because they are the most common cancer types across the available cell line datasets and also the drug for which we could find clinical datasets with drug response have been reported effective for them. Table 5.1 provides some characteristics of the employed datasets.

5.3.2 Experimental design

Drug response prediction using multiple labeled and unlabeled domains can be viewed in three approaches: 1) under the assumption that there is no data discrepancy, it can be viewed as a semi-supervised learning problem, 2) under the assumption that unlabeled patient samples are proxies to future patients, it can be viewed as an unsupervised domain adaptation problem, and 3) under the assumption that a generalizable representation can be obtained via only labeled domains, it can be viewed as a supervised domain generalization problem. It is important to note that there is no method of semi-supervised domain generalization for drug response prediction which is the main contribution of the Velodrome method.

To evaluate the performance of Velodrome, we compared it against the state-of-the-art methods of each approach. For the first approach, we compared Velodrome to Mean Teacher [147] which is the state-of-the-art deep neural network for semi-supervised learning [150]. For the second approach, we compared Velodrome to PRECISE [41] as a non-deep learning method based on subspace alignment and [193] as a deep learning method based on adversarial domain adaptation via disagreement between predictors. Finally, for the third approach, we compared Velodrome to Ridge regression as a non-deep learning baseline and DeepAll as a deep learning baseline. Both of them are categorized as methods of empirical risk minimization (ERM). ERM methods achieve state-of-the-art performance for out-of-distribution generalization [66]. They are trained in a supervised fashion by merging all available labeled input domains.

Data Preprocessing

We obtained all cell line datasets from the ORCESTRA platform [134] which stores pharmacogenomics datasets in PharmacoSet (PSet) R objects. Samples with missing values were removed from both the gene expression and drug response data. The cell line datasets were generated via the same drug screening assay (CellTiter Glo) preprocessed using the PharmacoGx package [189]. We also removed all the cell lines originating from non-solid tissue types from the cell line datasets.

We obtained the TCGA dataset via the Firehose (`http://gdac.broadinstitute.org/`) 28.01.2016. Expression values were converted to TPM and log2-transformed. The PDX and clinical trial datasets were preprocessed similar to the approach described in [40]).

Table 5.1: Characteristics of the datasets utilized in the Velodrome method

Dataset	**Drug**	**Type**	**Domain**	**Label**	**Tissue**	**# Samples**	**Genes**
CTRPv2	Docetaxel	Cell line	Source	AAC	Solid	292	1453
GDSCv2	Docetaxel	Cell line	Source	AAC	Solid	234	1453
TCGA-LUAD	Docetaxel	Patient	Source	Unlabeled	Solid	507	1453
TCGA-BRCA	Docetaxel	Patient	Source	Unlabeled	Solid	1051	1453
TCGA-PAAD	Docetaxel	Patient	Source	Unlabeled	Solid	131	1453
gCSI	Docetaxel	Cell line	Target	AAC	Solid	280	1453
GSE25065D	Docetaxel	Patient	Target	RECIST	Solid	51	1453
CTRPv2	Gemcitabine	Cell line	Source	AAC	Solid	514	2080
GDSCv2	Gemcitabine	Cell line	Source	AAC	Solid	226	2080
TCGA-LUAD	Gemcitabine	Patient	Source	Unlabeled	Solid	507	2080
TCGA-BRCA	Gemcitabine	Patient	Source	Unlabeled	Solid	1051	2080
TCGA-PAAD	Gemcitabine	Patient	Source	Unlabeled	Solid	131	2080
gCSI	Gemcitabine	Cell line	Target	AAC	Solid	277	2080
TCGA-Gem	Gemcitabine	Patient	Target	RECIST	Solid	66	2080
PDXE	Gemcitabine	PDX	Target	RECIST	Solid	25	2080
CTRPv2	Erlotinib	Cell line	Source	AAC	Solid	607	2066
GDSCv2	Erlotinib	Cell line	Source	AAC	Solid	230	2066
TCGA-LUAD	Erlotinib	Patient	Source	Unlabeled	Solid	507	2066
TCGA-BRCA	Erlotinib	Patient	Source	Unlabeled	Solid	1051	2066
TCGA-PAAD	Erlotinib	Patient	Source	Unlabeled	Solid	131	2066
gCSI	Erlotinib	Cell line	Target	AAC	Solid	283	2066
GSE33072	Erlotinib	Patient	Target	RECIST	Solid	25	2066
PDXE	Erlotinib	PDX	Target	RECIST	Solid	21	2066
CTRPv2	Paclitaxel	Cell line	Source	AAC	Solid	445	1452
GDSCv2	Paclitaxel	Cell line	Source	AAC	Solid	230	1452
TCGA-LUAD	Paclitaxel	Patient	Source	Unlabeled	Solid	507	1452
TCGA-BRCA	Paclitaxel	Patient	Source	Unlabeled	Solid	1051	1452
TCGA-PAAD	Paclitaxel	Patient	Source	Unlabeled	Solid	131	1452
gCSI	Paclitaxel	Cell line	Target	AAC	Solid	284	1452
GSE25065P	Paclitaxel	Patient	Target	RECIST	Solid	84	1452
PDXE	Paclitaxel	PDX	Target	RECIST	Solid	43	1452
TCGA-PRAD	Docetaxel	Patient	Target	Unlabeled	Solid	498	1453
TCGA-KIRC	Docetaxel	Patient	Target	Unlabeled	Solid	534	1453
TCGA-PRAD	Paclitaxel	Patient	Target	Unlabeled	Solid	498	1452
TCGA-KIRC	Paclitaxel	Patient	Target	Unlabeled	Solid	534	1452
TCGA-PRAD	Gemcitabine	Patient	Target	Unlabeled	Solid	498	2080
TCGA-KIRC	Gemcitabine	Patient	Target	Unlabeled	Solid	534	2080
TCGA-PRAD	Erlotinib	Patient	Target	Unlabeled	Solid	498	2066
TCGA-KIRC	Erlotinib	Patient	Target	Unlabeled	Solid	534	2066
CTRPv2	Docetaxel	Cell line	Source	AAC	Non-solid	62	1453
GDSCv2	Docetaxel	Cell line	Source	AAC	Non-solid	69	1453
gCSI	Docetaxel	Cell line	Target	AAC	Non-solid	50	1453
CTRPv2	Paclitaxel	Cell line	Source	AAC	Non-solid	100	1452
GDSCv2	Paclitaxel	Cell line	Source	AAC	Non-solid	67	1452
gCSI	Paclitaxel	Cell line	Target	AAC	Non-solid	50	1452
CTRPv2	Gemcitabine	Cell line	Source	AAC	Non-solid	129	2080
GDSCv2	Gemcitabine	Cell line	Source	AAC	Non-solid	69	2080
gCSI	Gemcitabine	Cell line	Target	AAC	Non-solid	50	2080
CTRPv2	Erlotinib	Cell line	Source	AAC	Non-solid	135	2066
GDSCv2	Erlotinib	Cell line	Source	AAC	Non-solid	68	2066
gCSI	Erlotinib	Cell line	Target	AAC	Non-solid	49	2066

For all of the employed datasets, all gene names were mapped to Entrez gene ids and the expression data were obtained before treatment and the response outcome after treatment. We reduced the number of genes to 2128 genes obtained from [194]. After the preprocessing, all of the available datasets for each drug had the same number of genes (Table 5.1).

Implementation Detail

We considered a wide range of values for each hyper-parameter of the Velodrome model and optimized these values via a random search separately for each drug. The sets of values considered are as follows: Epoch= $[10, 50, 100, 200]$
Learning rate (LR) = $[0.0001, 0.001, 0.01, 0.0005, 0.005, 0.05]$
Dropout (DR) = $[0.1, 0.3, 0.5, 0.8]$
Weight Decay (WD) = $[0.001, 0.0001, 0.01, 0.05, 0.005, 0.0005]$
$\lambda_1 = [1, 0.1, 0.2, 0.3, 0.4, 0.5, 0.01, 0.05, 0.001, 0.005, 0.0001, 0.0005]$
Minibatch size (MB) = $[17, 33, 65, 129]$
We considered separate learning rates and weight decays for the feature extractor and each predictor, but they all used the same sets of possible values.

We followed the training-domain validation set approach and splitted the labeled cell line datasets (CTRPv2 and GDSCv2) into train and validation and considered 90% for train and 10% for validation. We merged the train splits into one training dataset and similarly, merged the validation splits into one validation set and used the merged validation set to optimize the values of these hyper-parameters.

For Architecture, we followed previous work and designed predefined architectures (denoted by HD) for Velodrome [190, 140]. For the feature extractor, the first architecture has two hidden layers with the size 512×128, the second one has two layers with the size 256×256, the third one has three hidden layers with the size $128 \times 128 \times 128$ and the last architecture has four hidden layers with the size $64 \times 64 \times 64 \times 64$. We considered a batch normalization layer followed by an activation function (which we considered the Relu, the Tanh, and Sigmoid functions) as well as a dropout after the activation function for each hidden layer. The predictors have only one layer $HD \times 1$, where HD denotes the size of the last layer in the feature extractor. The final hyper-parameter and architecture of Velodrome for the studied drugs are as follows: $Drug : Epoch, MB, DR, WD1, WD2, WD3, HD, LR1, LR2, LR3, \lambda_1, \lambda_2$
Docetaxel: $10, 65, 0.1, 0.05, 0.0005, 0.0001, 3, 0.001, 0.005, 0.0005, 0.2, 0.8$
Gemcitabine: $10, 17, 0.1, 0.0001, 0.005, 0.01, 2, 0.01, 0.005, 0.05, 0.005, 0.99$
Erlotinib: $50, 129, 0.1, 0.05, 0.005, 0.0005, 2, 0.001, 0.01, 0.001, 0.01, 0.99$
Paclitaxel: $50, 129, 0.1, 0.005, 0.05, 0.005, 2, 0.05, 0.0005, 0.0001, 0.3, 0.7$
$WD1$, $WD2$, and $WD3$ refers to the values we used for the feature extractor, predictor 1, and predictor 2, respectively (similar for $LR1$, $LR2$, and $LR3$).
For re-running and the ablation study of the trained models, we considered these random values for the random seed: Seed= $[1, 21, 42, 84, 168, 336, 672, 1344, 2688, 5376]$.

We used 42 for the majority of the analyses (because it's the answer to life, the universe and everything!).

We used the same ranges for all of the baseline methods whenever using those values was applicable. For DeepAll-ERM and Ridge-ERM we used the existing implementations here: (`https://github.com/bhklab/PGx_Guidelines`), for PRECISE, we used the existing implementations here: (`https://github.com/NKI-CCB/PRECISE`). For Mean Teacher, we adopted an existing implementation for computer vision and modified it for this problem here: (https://github.com/CuriousAI/mean-teacher).

All of the deep neural network implementations were in the Pytorch framework and we employed the Adagrad optimizer to optimize the parameters of Velodrome as well as the baselines wherever applicable.

For performance evaluation, we employed the Scikit-learn and Scipy Python packages for the evaluation purposes. To be more specific, we utilized scikit-learn to calculate the AUROC and AUPR (for PDX samples and Patients) and we utilized the Scipy to calculate Pearson and Spearman correlations (For cell lines). For the association study, we utilized statsmodels.api Python package to fit the multiple linear regression and obtain the P-values and we obtained the list of known associated target genes for each drug by querying the PharmacoDB resource [21].

5.3.3 Results

Velodrome makes accurate predictions for cell lines

To investigate the generalization of Velodrome to other cell line datasets, we employed the gCSI dataset as the target domain and reported the performance of Velodrome and the baselines in terms of the Pearson and the Spearman correlation on this dataset. On average, DeepAll-ERM achieved the best performance (0.52 ± 0.09 for Pearson correlation coefficient and 0.48 ± 0.09 for Spearman correlation coefficient -Figure 5.2A and D). Velodrome achieved the second best performance (0.48 ± 0.09 for Pearson correlation coefficient and 0.45 ± 0.07 for Spearman correlation coefficient -Figure 5.2A and D). Ridge-ERM (0.46 ± 0.07- Figure 5.2A and D) and Mean Teacher (0.430.07- Figure 5.2A and D) had the third best performance in terms of Pearson and Spearman correlation, respectively. These results indicate that although Velodrome is not the best performing model, it is fairly competitive on cell lines and generalizes well.

Velodrome makes accurate predictions for PDXs samples

To investigate generalization of Velodrome to PDX samples, we employed the PDXE dataset as the target domain and reported the performance of Velodrome and the baselines discussed above in terms of the AUROC and the AUPR. On average, Velodrom achieved the best performance compared to the baselines (for 0.69 ± 0.21 in AUROC and 0.43 ± 0.26 in

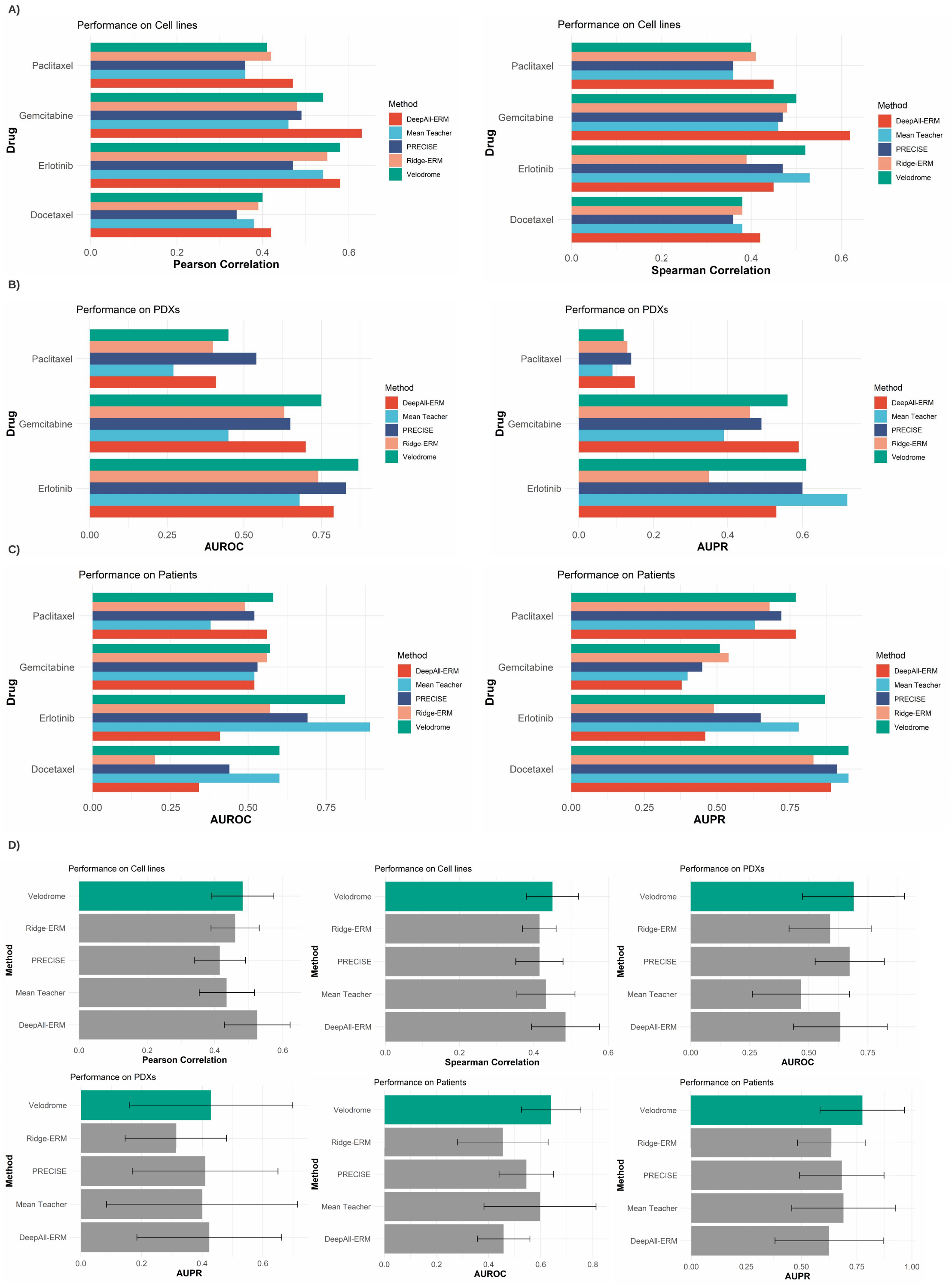

Figure 5.2: Comparison of Velodrome and state-of-the-art of drug response prediction methods on cell lines in terms of Pearson and Spearman correlations (A), PDX models in terms of the Area Under the Receiver Operating Characteristic curve (AUROC) and the Area Under the Precision-Recall curve (AUPR) (B), and patients in terms of AUROC and AUPR (C). On average (D), Velodrome has the best or the second best performance on cell lines, PDX models, and patients compared to the baselines over the studied drugs.

AUPR-Figure 5.2B and D). PRECISE obtained the second best performance in terms of AUROC (0.67 ± 0.14-Figure 5.2B and D) and DeepAll-ERM in terms of AUPR (0.4 ± 20.23-Figure 5.2B and D). Similarly, DeepAll had the third best performance in terms of AUROC (0.63 ± 0.19-Figure 5.2B and D) and PRECISE had the third best performance in terms of AUPR (0.41 ± 0.24-Figure 5.2B and D). These results indicate that utilizing both labeled and unlabeled samples from cell lines and patients improves drug response prediction on PDX samples.

Velodrome makes accurate predictions for patients

To investigate the generalization of Velodrome to patient samples, we employed the patient datasets obtained from clinical trials as target domains and reported the performance of Velodrome and the baselines discussed above in terms of AUROC and AUPR. On average, Velodrome achieved the best performance compared to the baselines and significantly outperformed them (0.64 ± 0.11 in AUROC and 0.77 ± 0.19 in AUPR-Figure 5.2C-D). Mean Teacher obtained the second best performance (0.59 ± 0.21 in AUROC and 0.69 ± 0.23 in AUPR-Figure 5.2C-D) and PRECISE had the third best performance (0.54 ± 0.1in AUROC and 0.68 ± 0.18 in AUPR-Figure 5.2C-D).

These results indicate that incorporating unlabeled patient data as well as labeled data significantly improves the generalization performance on patients because these three methods take unlabeled samples as input as well as labeled samples as opposed to the other baselines which only take labeled samples. However, the results also demonstrate the advantage of learning features that are domain-invariant and hypothesis-invariant for out-of-distribution generalization, because the PRECISE method only ensures a domain-invariant representation.

Velodrome outperforms the baselines over multiple independent runs

To maximize the reproducibility, we utilized a fixed random seed for all methods (Velodrome and the baselines) and found the best settings for the hyper-parameters of each method with that seed. To investigate the performance of the best trained Velodrome model for each drug and those of the baselines, we re-trained all of the models from scratch using the same settings with 10 different random seeds and reported mean±std for each method (Figure 5.3A).

Although we observed that the average performance (over the studied drugs) of all methods decreased, Velodrome still achieved the best performance on patients in terms of both AUROC and AUPR, and also the best performance in terms of both Pearson and Spearman correlation on cell lines. PRECISE and DeepAll-ERM obtained the best performance on PDX samples in terms of AUROC and AUPR, respectively (the performance of these two methods tied on AUPR). Velodrome had the third best performance in terms of AUROC and AUPR on PDX samples. Overall, these results indicate that Velodrome is more accurate and competitive compared to baselines particularly on patients and cell lines.

The complete version of Velodrome demonstrates the best performance

We performed an ablation study to investigate the impact of the different loss components of Velodrome separately. We studied three scenarios as follows: “Velodrome w/o A” represents a version of Velodrome without the alignment loss component, which means the neural network only uses supervised and semi-supervised losses. “Velodrome w/o C” represents a version of Velodrome without the consistency loss, which means the neural network only considers the supervised loss and the alignment loss. Finally, “Velodrome w/o AC” represents a version of Velodrome without both the alignment and the consistency loss, which means the neural network employed only has a standard supervised loss. Our results on patients demonstrate that on average (over 10 independent runs), the complete version of Velodrome outperforms its variants which indicates the added value of both alignment and consistency losses (Figure 5.3-B). Interestingly, removing the consistency loss from the objective function had the biggest impact on the Velodrome performance on patients. This may suggest that hypothesis alignment plays a more critical role than feature alignment for out-of-distribution generalization, which is compatible with recent observations in computer vision applications [71].

Velodrome generalizes to well-represented tissue types

To evaluate the performance of Velodrome on patients, we followed the experimental design of previous pharmacogenomics methods and designed an association study based on the known associated target genes for the investigated drugs [19, 40, 20, 61]. In this analysis, we employed the TCGA Kidney cancer cohort (TCGA-KIRC) as a tissue type well represented in our cell line datasets. In GDSCv2 and CTRPv2 combined, more than 3.3% of the samples originated from this tissue type (Figure 5.4). We trained Velodrome models for each drug (Docetaxel, Erlotinib, Paclitaxel, and Gemcitabine) and applied them to the gene expression data of the patients of this cohort to predict their response. Then, we fit a linear regression model to the level of expression of the known target genes of these drugs and the responses predicted by Velodrome. Based on the corrected p-values (two-tailed t-test) obtained from this multiple linear regression using the bonferroni correction method, there are a number of statistically significant associations between the target genes of the studied drugs and the responses predicted by Velodrome. For Docetaxel, MAP2 had a statistically significant association ($P < 10^{-6}$). For Erlotinib, EGFR and ERBB2 had statistically significant associations (both $P < 10^{-6}$). For Paclitaxel BCL2 and MAP2 had significant associations (both $P < 10^{-6}$). Finally, for Gemcitabine, CMPK1 demonstrated a significant association($P < 10^{-6}$). These results suggest that the responses predicted by Velodrome are not random but capture biological aspects of the drug response.

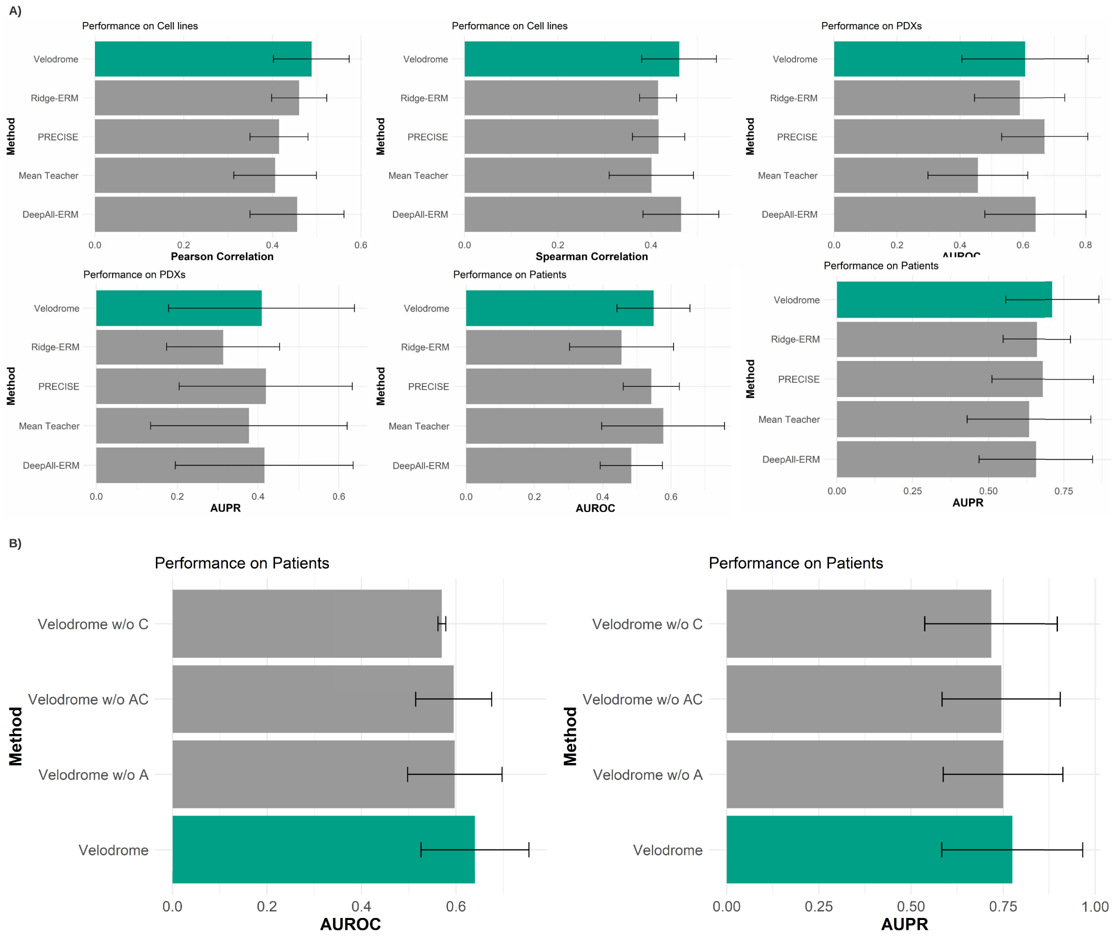

Figure 5.3: (A) Comparison of the average performance of the Velodrome and the state-of-the-art of drug response prediction methods over 10 independent runs . (B) An ablation study of the Velodrome performance on patients.

Velodrome generalizes to under-represented tissue types

To further evaluate the performance of Velodrome, we performed a similar association study on the prostate cancer cohort in TCGA (TCGA-PRAD). We chose prostate because unlike kidney, prostate is a tissue type under-represented in our cell line datasets (only 0.3% of the samples originated from this tissue). Similar to TCGA-KIRC, the Velodrome predictions for TCGA-PRAD patients demonstrated significant associations with known target genes of the studied drugs. For Docetaxel, MAP2 showed a statistically significant association ($P < 10^{-6}$). For Erlotinib, both EGFR and ERBB2 showed statistically significant associations (both $P < 10^{-6}$). For Paclitaxel, BCL2 ($P = 8 \times 10^{-6}$) and MAP2 ($P = 10^{-4}$) had significant associations. Finally, for Gemcitabine, CMPK1 demonstrated significant association ($P < 10^{-6}$). These results confirm again that the responses predicted by Velodrome are not random and they capture biological aspects of the drug response even for a tissue under-represented in the source domain.

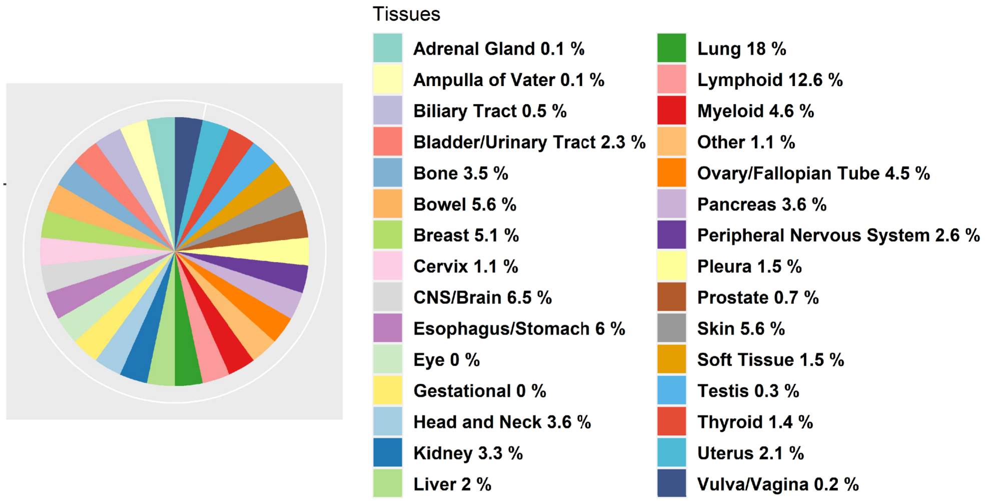

Figure 5.4: The percentage of tissue types in CTRPv2 and GDSCv2 cell line datasets combined.

Velodrome generalizes to new tissue types

Finally, we trained Velodrome and the baselines only on samples (cell lines and patients) that originated from solid tissue types because non-solid tissues such as haematopoietic and lymphoid have different molecular and pharmacological profiles compared to solid ones [190]. Therefore, we wanted to examine the out-of-distribution capability of the Velodrome models on these tissue types that were completely absent during training. For that, we tested the trained Velodrome models for the studied drugs on samples originated from non-solid tissues in the gCSI cell line dataset and evaluated the performance in terms of Pearson correlation between the predictions and the actual AAC values and reported two-tailed p-value as well.

For Erlotinib and Gemcitabine, Velodrome demonstrated significant correlations of 0.4 ($P = 5 \times 10^{-3}$) and 0.39 ($P = 4 \times 10^{-3}$), respectively. For Docetaxel and Paclitaxel, Velodrome did not make accurate predictions and had poor correlations of -0.07 and -0.02, respectively (both $P > 0.05$).

As a baseline to compare the Velodrome performance on non-solid tissues, we trained a Ridge Regression model on samples originated from non-solid tissues in CTRPv2 and GDSCv2 datasets and tested this predictor on non-solid samples of gCSI dataset. Therefore, we built a predictor specifically for non-solid samples and the performance of this model should act as an upper bound for the Velodrome. Similar to the Velodrome results, this predictor also achieved significant correlations of 0.34 ($P = 10^{-2}$) and 0.39 ($P = 5 \times 10^{-3}$) for Erlotinib and Gemcitabine and negative correlations of -0.11 ($P > 0.05$) and -0.4 ($P = 4 \times 10^{-3}$) for Docetaxel and Paclitaxel, respectively.

These results suggest that Velodrome is as accurate (and even more accurate in the case of Erlotinib) as a non-solid predictor on these tissues even though it did not utilize them during training. The poor/negative correlation for Docetaxel and Paclitaxel may be dataset specific, particularly in the case of Paclitaxel where the non-solid predictor had a significant negative correlation, and requires further study.

5.4 Discussion

From the biological point of view, we found interesting connections between the known target genes of the studied drugs and the TCGA cohorts that we investigated (TCGA-PRAD and TCGA-KIRC). For example, BCL2 has known connections to prostate cancer progression [195, 196] and survival [197]. More importantly, the expression of BCL2 may have an antiapoptotic activity against androgen which is a key player in prostate cancer [195]. Similarly, BCL2 can also act as an oncoprotein in kidney cancer [198] and therapeutics roles [199]. As another example, Microtubule-Associated Proteins including MAP2 have also been associated with different cancers including prostate [179] and kidney cancers [200]. Moreover, Microtubule-targeting chemotherapy agents, Docetaxel and Paclitaxel, have been used in combination with anti-androgen therapeutics to increase the survival rate in prostate cancer patients [201]. Prostate cancer progression and lethal outcome have been associated with metabolic signaling pathways and CMPK1 (it mediates the mechanism of action for Gemcitabine) was shown to be highly expressed in prostate cancer patients [202]. A combination of Gemcitabine and other chemotherapy agents has shown to be effective for a subtype of kidney cancer [203]. Finally, EGFR and ERBB2 have been associated with different cancer types including prostate [204, 205] and kidney [206] and they both showed therapeutic opportunities and increase in survival [207]).

From the computational point of view, it has been shown that methods of empirical risk minimization (ERM) are highly competitive for supervised domain generalization [66]. Therefore, it was also expected to see a competitive performance for a semi-supervised method (Mean Teacher) for the semi-supervised domain generalization setting. Moreover, Velodrome, PRECISE, and Mean Teacher were designed to take both labeled and unlabeled samples and, therefore, were expected to achieve better performance on patients than DeepAll-ERM and Ridge-ERM. On the other hand, these two methods achieved better performance on cell lines which makes sense since they were trained on cell lines.

We considered only TCGA-BRCA, TCGA-PAAD, and TCGA-LUAD for training, because these tissue types were well-represented in our cell line datasets (Figure 5.4) and because the four studied drugs were treatment options for these cell lines. This selection increases the relevancy of labeled (cell lines) and unlabeled (TCGA patients) data. Relevancy has been shown to improve semi-supervised learning performance even when both labeled and unlabeled datasets are imbalanced [150], which is the case for drug response prediction. Although methods of adversarial domain adaptation have shown great performance in different applications, especially computer vision [47, 81, 57], we did not consider them as baselines because they were clearly outperformed by PRECISE (which we do use as baseline) in a recent study [61].

Although gene expression data has been shown many times to be the most effective genomic data type for drug response prediction [13, 28], in principle Velodrome can be extended to incorporate other omics data types. Especially promising are proteomics data [208] and germline variants [209], due to their predictive power. The advantage of proteomics is that it is closer to the phenotype and gene expression and protein abundance can be quite discordant. Velodrome can also be extended to incorporate additional information about the drug, such as the chemical representation, to improve the performance [142].

Finally, we did not discuss the explainability of the Velodrome model, but we note that the feature extractor of Velodrome can be replaced by a knowledge-based network [59] to offer explainability and transparency [210]. A major limitation of our work is the output space discrepancy between cell lines, PDX samples, and patients, because on cell lines the drug response is measured based on the concentration of the drug but on PDX samples and patients the response is measured based on the change in the tumor volume after treatment. A recent method adjusts for this output space discrepancy and improves the prediction performance [61], but this method requires access to the target domain during training which violates the assumption of out-of-distribution generalization. In this work, we used AAC as the measure of drug response in cell line datasets and treated it as a score for making predictions for patients and PDX samples. However, measuring AAC is dependent on the tested concentration range which generally differs between different pharmacogenomics studies. Recent efforts have demonstrated that adjusting concentration ranges across different datasets improves the prediction performance [125, 211], however, we did not consider this adjustment because it reduces the sample size substantially.

Chapter 6

Conclusion

6.1 Summary

In this thesis, we proposed three methods of drug response prediction. These methods address some of the major challenges in drug response prediction including, multi-omics integration, input and output discrepancy between cell lines and patients, and out-of-distribution generalization to unseen samples.

We proposed MOLI, a Multi-Omics Late Integration method for drug response prediction based on deep neural networks. We trained MOLI on a pan-cancer cell line dataset and successfully validated it on PDX and patient data for five chemotherapy agents and two targeted therapeutics.

Our results suggest that MOLI outperforms single-omics (gene expression) prediction performance in terms of AUROC and area under precision-recall curve. Also, MOLI outperforms deep neural networks using early integration in terms of AUROC and area under precision-recall curve. Moreover, MOLI with its combined objective function outperforms single- and multi-omics baselines with only the classification loss. Moreover, MOLI trained on the pan-drug inputs, employing transfer learning, outperforms MOLI trained on drug-specific inputs for targeted therapeutics that target EGFR.

Finally, we analyzed the biological significance of MOLI and found substantial evidence that the responses predicted by MOLI have statistically significant associations with the expression level of numerous genes in the EGFR pathway for TCGA patients with breast, kidney, lung, and prostate cancers.

We also proposed AITL, an Adversarial Inductive Transfer Learning method which, to the best of our knowledge, is the first method that addresses the discrepancies in both the input and output spaces. AITL uses a feature extractor to learn features for target and source samples. Then, to address the discrepancy in the output space, AITL utilizes these features as input of a multi-task subnetwork that makes predictions for the target samples and assign cross-domain labels to the source samples. Finally, to address the input space discrepancy, AITL employs global and class-wise discriminators for learning domain-invariant features. In pharmacogenomics, AITL adapts the gene expression data obtained from cell lines and patients in the input space, and adapts different measures of the drug response between cell lines and patients in the output space. In addition, AITL can be employed in other applications such as predicting long-term clinical labels for slow progressing cancers.

We evaluated AITL on four different drugs and compared it against state-of-the-art baselines in terms of AUROC and AUPR. The empirical results indicated that AITL achieved a significantly better performance compared to the baselines showing the benefits of addressing the discrepancies in both the input and output spaces. Moreover, we analyzed AITL's predictions for the studied drugs on breast, prostate, lung, kidney, and bladder cancer patients in TCGA. We showed that AITL's predictions have statistically significant associations with the level of expression of some of the annotated target genes for the studied drugs.

Finally, we proposed Velodrome, a transfer learning method for drug response prediction based on gene expression data. Velodrome is the first semi-supervised method of out-of-distribution generalization. We trained Velodrome on cell line datasets with drug response (measured in AAC) and patient datasets without drug response (i.e., unlabeled data) as source domains and successfully validated it on different target domains such as cell lines, PDX samples, and patient data across three chemotherapy agents and one targeted therapeutic. Our results suggest that Velodrome outperforms state-of-the-art methods of drug response prediction and transfer learning in terms of Pearson and Spearman correlations (on cell lines) and in terms of AUROC and AUPR (on PDX samples and patients).

Moreover, we analyzed the biological significance of the predictions made by Velodrome. We provided substantial evidence that these predictions have statistically significant associations with the expression level of numerous known target genes of the studied drugs. We performed this analysis for a tissue well-represented in our source domains, i.e. kidney cancer, and a tissue under-represented in our source domains, i.e. prostate cancer. Finally, we also demonstrated that Velodrome generalizes to new tissue types that were completely absent in the source domains. All these results demonstrate the superior out-of-distribution generalization capability of the Velodrome model and suggest that Velodrome may guide pharmacogenomics and precision oncology more accurately.

Our experimental results suggest that MOLI, AITL, and Velodrome may have a role in precision oncology where currently only ∼5% of all patients benefit from precision oncology.

Returning to the three main questions in transfer learning (when?, what?, and how?):

1. This thesis explored transfer learning between pre-clinical and clinical resources (when?)
2. MOLI utilized transferring relevant samples and AITL and Velodrome utilized transferring feature representations (what?)
3. We employed deep metric learning, adversarial multi-task learning, and out-of-distribution generalization to achieve accurate predictions (how?).

However, it is also of utmost importance and significance to consider non-technical aspects of this thesis. Key questions that may arise are how these methods can fit into the larger problem of precision oncology? What are the impacts of these methods on precision oncology? Given the described computational methods, what is the right way to administer treatments? i.e., what medically useful information do clinicians need to make treatment decisions?

The answers to these or similar questions lies within the three key areas of data, model, and trust. On the data level, different drugs may have different predictors in terms of omics data types. The success of these methods for having a meaningful impact requires employing the right data type. Part of having the right data type comes from wet lab experimental research to validate biomarkers or omics data types for a specific cancer type or a disease. The other part comes from obtaining omics data as a routine medical procedures. Obviously, this requires affordable assays that with reasonable cost provide reliable and accurate omics data. Finally, the last part is coming from the computational analysis when diverse omics data types are available but the impacts of them on making accurate predictions is unknown. Ideally, a computational model should determine the right data type or data types automatically with respect to the desired output. Moreover, the input data to these models should be free of biases caused by inequality or the lack of inclusion for certain demographic groups. This aspect of the data is particularly important because it can have negative impacts on models in terms of being biased and consequently, diminishing the trust in these models.

On the model level, the models discussed in this thesis followed the notion of black box, meaning given the input what is the desired output without further explanation or interpretation of predictions. This lack of transparency has led to both demands for explainability from lawmakers, such as the European Union's General Data Protection Regulation requirement for transparency and the machine learning community to make that an active area of research to deconvolute these black boxes. This is particularly more important for deploying these models for applications like drug response prediction. Offering an interpretation of the predictions can have direct positive impact on trusting these models. In addition to data and model, it is important to note that gaining trust in these computational models also depends on other factors such as uncertainty estimation of predictions.

Trust is also dependent to other potentially necessary outcomes of a drug response predictor that can influence a treatment decision such as adverse drug reaction, toxicity, drug-target interaction, or drug combination predictions. All of these outcomes can be predicted simultaneously with drug response in a multi-task learning setting to provide clinicians with as much information as possible to make informed decisions. With these three key aspects together, computational models can have huge impacts on determining the most effective treatment options with the reason(s) behind each and the level of uncertainty in them. Moreover, such models can also initiate new hypotheses whenever suggesting novel biological knowledge. Nonetheless such biological knowledge require rigorous wet lab experimental validations. Finally, trust is also related to patients because ultimately the clinician provides different options and recommendations to the patient but the patient has to make the final decision. Therefore, clear communication between computer scientists and patients is necessary in lay language. One way to moderate these communications is via grant agencies, for example, the Prostate Cancer Foundation of British Columbia requires the award winners to present their research to patients as well as the grant panel to inform them about most recent treatment options and the rationale behind them. Moreover, it is also highly important to consider privacy of patients in the entire process especially when it comes to data sharing to assure them that their data cannot be linked to their identities [212, 213].

In principle, the proposed methods in this thesis can be extended to incorporate these key aspects. In addition, these models are not limited to drug response prediction and are applicable to other clinical problems. This chapter introduces some of future directions that make these models more applicable to drug response prediction in the clinic and other applications.

6.2 Future directions

6.2.1 Knowledge-based models

Although current methods for drug response prediction are becoming more accurate, there is still a need to switch from ‘black box’ predictions to methods that offer high accuracy as well as interpretable predictions. This is of particular importance in real-world applications such as drug response prediction in cancer patients [59].

A promising future direction is to incorporate domain expert knowledge into deep neural networks [59, 166]. The advantage of these methods is that having domain expert knowledge offers transparency and explainability of the predictions which is of utmost importance and significance for critical applications such as drug response prediction for cancer patients.

We proposed a method called BDKANN, by employing prior biological knowledge in the form of the hierarchical connections of genes to protein complexes to pathways and finally to drugs as layers of a neural network. The structure of BDKANN is as follows: 1) in the gene layer of this network, a node represents a gene for which the expression data is available. 2) In the protein complex layer, a node represents the complex that genes in the previous layer can form. 3) In the pathway layer, a node represents a pathway that a protein complex (or multiple complexes) in the previous layer is (are) a part of. Lastly, 4) in the drug layer, a node represents a drug that targets a given pathway(s) in the previous layer (Figure 6.1).

We also make an extended version of BDKANN that we call BDKANN+, which has the added ability to discover new connections in the biological domain knowledge through the

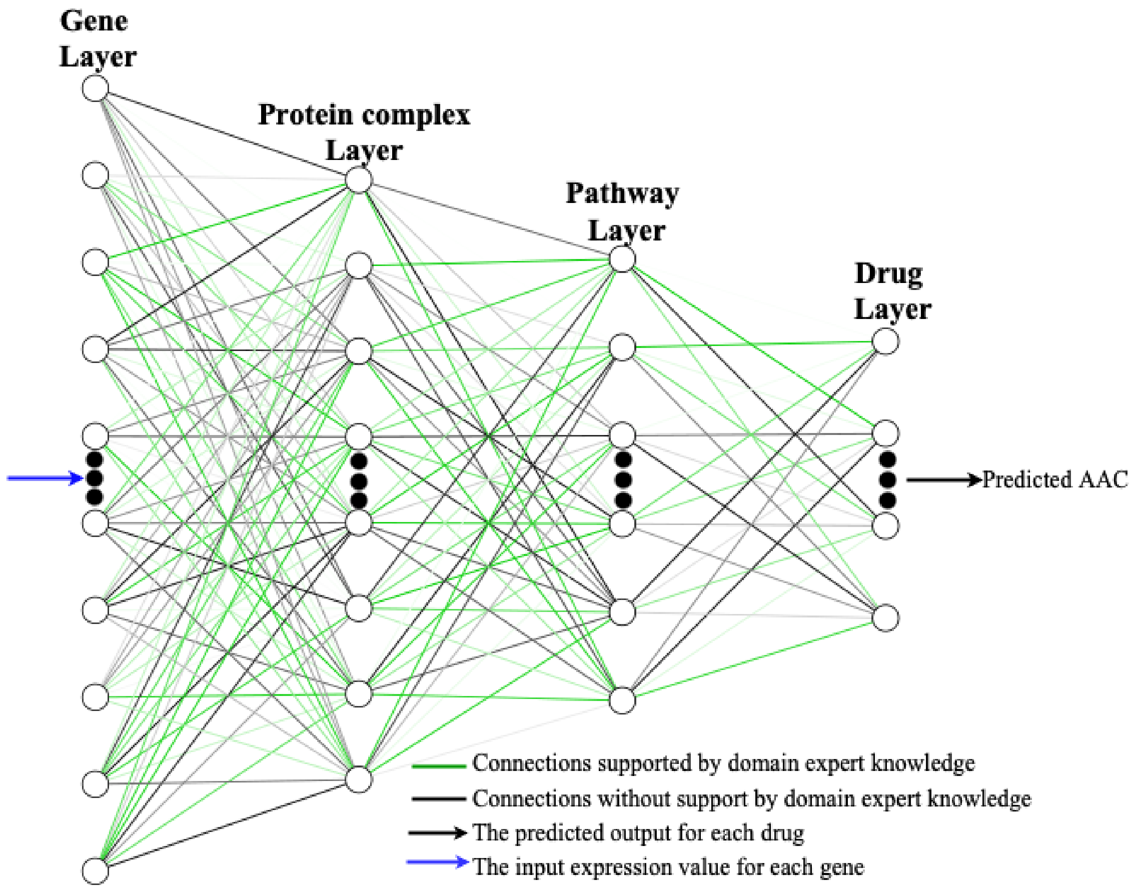
Gene Layer
Protein complex Layer
Pathway Layer
Drug Layer
Predicted AAC
Connections supported by domain expert knowledge
Connections without support by domain expert knowledge
The predicted output for each drug
The input expression value for each gene

Figure 6.1: Overview of the structure of BDKANN and BDKANN+ with black connections that are not supported by domain knowledge but are regularized to give preference to the green connections that are supported by biological knowledge

use of regularization. We compare both versions of our model to both knowledge-based and non knowledge-based DNN baselines and find that not only does BDKANN+ outperform BDKANN and baselines but also allows for meaningful interpretation of those results that can help us better understand the decisions of the model and help generate hypotheses relevant to cancer drug response prediction [59].

BDKANN (and BDKANN+) layers can be utilized as the feature extractor of MOLI, AITL, and Velodrome to offer explainability of the predictions. In MOLI, AITL, and Velodrome, each model was trained to make predictions for one drug only but BDKANN was trained to make predictions for more than one drug per model. In principle, the BDKANN approach is directly applicable for the feature extractors of MOLI, AITL, and Velodrome models, but the prediction layers of them should be extended to include multiple drugs (similar to the BDKANN approach). The advantage of this extension is to make explainable predictions via MOLI, AITL, and Velodrome similar to those of the BDKANN model.

6.2.2 Other input data types

There are also some promising future directions regarding different input data types that are beyond the scope of this thesis. One promising area that has recently received attention is incorporating additional information about the input drug(s) to the model and it has been shown to improve the prediction performance. This information is often in the form of drugs chemical structures [60, 64, 141], drug-target interaction or adverse drug reaction [142].

Another promising area is incorporating other omics data types. Although gene expression has been shown in independent studies to be the most effective data type for drug response prediction [13, 26, 28], and we demonstrated in MOLI experiments the utility of CNA and mutation data, proteomics also has been showing promising results in drug response prediction [214]. Protein expression is closer to the phenotype compared to gene expression and may offer a better viewpoint for drug response prediction [158]. There is a need for large-scale pre-clinical datasets with proteomics data which is currently only available for a few datasets [21, 127]. It is obvious that more proteomics data will be available for both pre-clinical and clinical samples in the future and we need reliable methods to make the best use of them as input. Similarly, germline variants can also offer additional information to the model that somatic mutation and gene expression cannot capture [209]. Currently only one study offers germline mutation information [209]. More, including drug perturbation data, meaning measuring gene expression before and after treatment has been shown to guide drug response prediction more accurately [124].

Finally, recent advancement in single cell technology has provided researchers with biological data with more calibrated resolutions to capture millions of cells within a single tissue. Single cell data has brought numerous computational challenges for which some of them can be addressed via transfer learning [215]. For example, unsupervised domain adaptation can be utilized for cell type identification across different samples because it allows the model to identify cell types that are specific to a source domain and cell types that are specific to a target domain. Similarly, identifying sensitive and resistant cells to a given drug can be formulated as a transfer learning problem because many pharmacogenomics datasets are available based on bulk sequencing but such datasets do not exist for single cell data. Therefore, these resources can be utilized together to identify response to different drugs across different cell types. Similarly, the same transfer learning settings can be defined to study the effect of normal tissues on drug response prediction performance. This is important to study because pharmacogenomics datasets study the impact of drugs on cancer cells but it is equally important to study the impact of these drugs on normal cells. For this goal, resources like The Genotype-Tissue Expression (GTEx) that studied gene expression of healthy individuals can be employed [216].

6.2.3 Clinical utility

Drug response prediction was the driving application of the proposed methods in this thesis. However, in principle, these methods are applicable to other problems, especially in clinical settings. One very relevant problem is metastasis prediction in prostate cancer. For prostate cancer patients, timing and intensity of therapy are adjusted based on their prognosis. Clinical and pathological factors, and recently, gene expression-based signatures have been shown to predict metastatic prostate cancer. Previous studies used labeled datasets, i.e. those with information on the metastasis outcome, to discover gene signatures to predict metastasis. Due to steady progression of prostate cancer, datasets for this cancer have a limited number of labeled samples but more unlabeled samples. In addition to this issue, the high dimensionality of the gene expression data also poses a significant challenge to train a classifier and predict metastasis accurately [187].

To address this challenge, we proposed the Deep Genomic Signature (DGS) method to predict metastasis in prostate cancer patients. DGS is based on Denoising Auto-Encoders (DAEs) and unsupervised transfer learning. We utilized a DAE to extract the most salient features of gene expressions from the samples in a large unlabeled dataset, as the source domain. The learned weights and biases of this DAE were transferred to another DAE, trained on a smaller but labeled (with metastasis outcome) dataset, as the target domain. During training of the second DAE, the transferred parameters were frozen, and only the additional parameters were trained. DGS selects the list of genes with high weights by employing a standard deviation filter on the transferred and learned weights of the second DAE. Finally, DGS made predictions via an elastic net logistic regression model [217] on the labeled target domain, using the expressions of the selected genes as features, to predict metastasis. Due to the elastic net regularization, only a subset of the selected genes have non-zero coefficients, and these genes form the DGS gene signature for metastasis in prostate cancer. We applied DGS to six previously published labeled datasets and one large unlabeled dataset obtained from the Decipher test of GenomeDx Inc. We compared the accuracy of the gene signature discovered by DGS against those of the state-of-the-art signatures for prostate cancer. Figure 6.2 illustrates the schematic overview of the DGS method.

The problem that DGS was developed to address is out-of-distribution generalization from labeled and unlabeled data to generalize to unseen cohorts from different studies/institutions. Therefore, Velodrome is completely applicable to this problem. similarly, if multiple omics data types are available, Velodrome can be extended to multiple omics specific encoding subnetworks to perform late integration similar to MOLI for metastasis prediction. Finally, if different clinical annotations are available across different cohorts (e.g. Gleason score in one cohort and lymph nodes status in another one), the AITL approach will be beneficial to address output space discrepancy. Therefore, one promising future direction is to extend

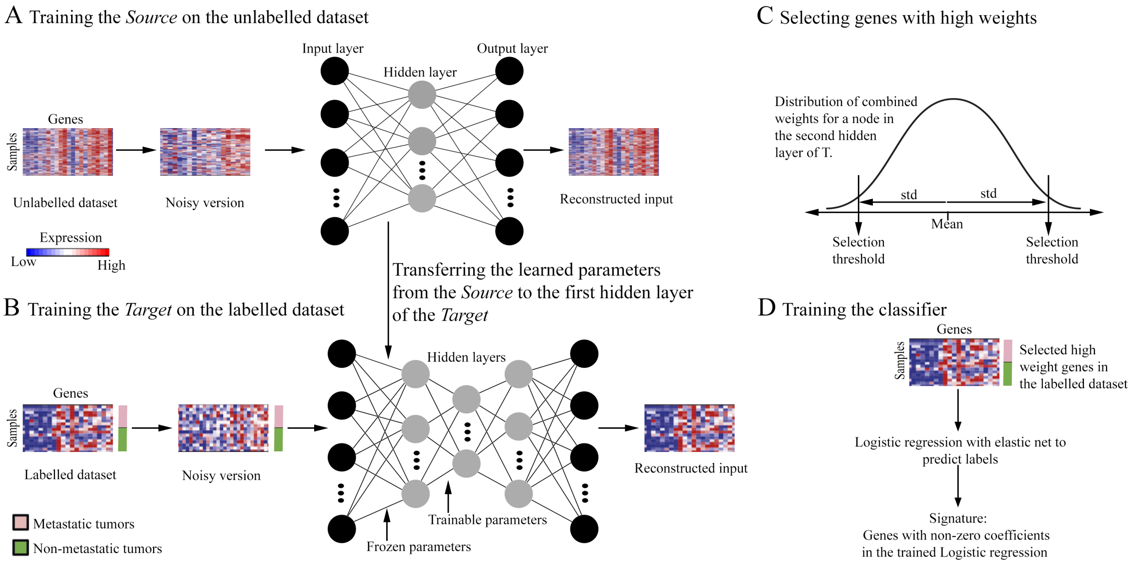
A Training the *Source* on the unlabelled dataset
Input layer
Output layer
Hidden layer
Genes
Samples
Unlabelled dataset
Noisy version
Reconstructed input
Expression
Low
High
C Selecting genes with high weights
Distribution of combined weights for a node in the second hidden layer of T.
std
std
Mean
Selection threshold
Selection threshold
Transferring the learned parameters from the *Source* to the first hidden layer of the *Target*
B Training the *Target* on the labelled dataset
Hidden layers
Genes
Samples
Labelled dataset
Noisy version
Reconstructed input
Trainable parameters
Frozen parameters
Metastatic tumors
Non-metastatic tumors
D Training the classifier
Genes
Samples
Selected high weight genes in the labelled dataset
Logistic regression with elastic net to predict labels
Signature:
Genes with non-zero coefficients in the trained Logistic regression

Figure 6.2: **Schematic overview of Deep Genomic Signature** (A) Training the *Source* (a Denoising Auto-Encoder) on the unlabeled data to extract salient features from this dataset. The *Source* has one hidden layer. (B) Training the *Target* (another Denoising Auto-Encoder) on the labeled data. The *Target* has two hidden layers. Parameters of the first hidden layer are transferred from the *Source* which remain frozen and parameters of the second hidden layer (initialized randomly) are trained. (C) Applying a standard deviation filter to select genes based on their weights in the *Target*. These genes are in the tails of the weight distribution of nodes in the second hidden layer of the *Target*. (D) Training an elastic net logistic regression model (l_1 and l_2 regularization) to predict metastasis. The DGS gene signature consists of all of the genes with non-zero coefficients in this classifier.

and apply the proposed methods in this thesis to clinical problems similar to metastasis prediction in prostate cancer.

6.2.4 Drug combination

Generally, a single drug does not provide adequate treatment for a cancer patient. In this thesis, models were trained to predict the effect of a single drug (also known as monotherapy). However, machine learning methods have demonstrated promising results for drug combination prediction as well [60]. Moreover, integrated and standardized resources are also available for drug combination pharmacogenomics studies [218]. Therefore, a promising future research for this thesis is to expand the proposed methods for drug combination.

6.2.5 Self-supervision learning

Recently, self-supervision learning has received a lot of attention in the machine learning community with the aim of training models without providing supervision from humans [150]. A common approach for self-supervision is to add label-invariant noise to the input data, meaning a type of noise that does not change the semantic of the sample (for example rotating an image) and then training a model to predict the applied noises to each sample. This network has shown to be a reliable pre-trained model for downstream tasks [150]. However, adding noise to omics data types such as gene expression is not as straightforward as images because we do not know the impact of that noise with respect to the labels. For example, it is not clear what type of noise and how much of that noise will not change the drug response outcome. Therefore, a very promising future direction is to propose a method of self-supervision learning applicable to omics data types.

Bibliography

[1] J. C. Denny and F. S. Collins, “Precision medicine in 2030—seven ways to transform healthcare,” *Cell*, vol. 184, no. 6, pp. 1415–1419, 2021.

[2] D. Hartl, V. de Luca, A. Kostikova, J. Laramie, S. Kennedy, E. Ferrero, R. Siegel, M. Fink, S. Ahmed, J. Millholland, *et al.*, “Translational precision medicine: an industry perspective,” *Journal of Translational Medicine*, vol. 19, no. 1, pp. 1–14, 2021.

[3] E. J. Topol, “High-performance medicine: the convergence of human and artificial intelligence,” *Nature medicine*, vol. 25, no. 1, p. 44, 2019.

[4] G. Adam, L. Rampášek, Z. Safikhani, P. Smirnov, B. Haibe-Kains, and A. Goldenberg, “Machine learning approaches to drug response prediction: challenges and recent progress,” *NPJ precision oncology*, vol. 4, no. 1, pp. 1–10, 2020.

[5] J.-K. Lee, Z. Liu, J. K. Sa, S. Shin, J. Wang, M. Bordyuh, H. J. Cho, O. Elliott, T. Chu, S. W. Choi, *et al.*, “Pharmacogenomic landscape of patient-derived tumor cells informs precision oncology therapy,” *Nature genetics*, vol. 50, no. 10, p. 1399, 2018.

[6] A. Zehir, R. Benayed, R. H. Shah, A. Syed, S. Middha, H. R. Kim, P. Srinivasan, J. Gao, D. Chakravarty, S. M. Devlin, *et al.*, “Mutational landscape of metastatic cancer revealed from prospective clinical sequencing of 10,000 patients,” *Nature medicine*, vol. 23, no. 6, p. 703, 2017.

[7] M. L. Cheng, M. F. Berger, D. M. Hyman, and D. B. Solit, “Clinical tumour sequencing for precision oncology: time for a universal strategy,” *Nature Reviews Cancer*, vol. 18, no. 9, p. 527, 2018.

[8] J. Marquart, E. Y. Chen, and V. Prasad, “Estimation of the percentage of us patients with cancer who benefit from genome-driven oncology,” *JAMA oncology*, 2018.

[9] S. P. Gavan, A. J. Thompson, and K. Payne, “The economic case for precision medicine,” *Expert review of precision medicine and drug development*, vol. 3, no. 1, pp. 1–9, 2018.

[10] A. Mishra and M. Verma, “Cancer biomarkers: are we ready for the prime time?,” *Cancers*, vol. 2, no. 1, pp. 190–208, 2010.

[11] H. Gao, J. M. Korn, S. Ferretti, J. E. Monahan, Y. Wang, M. Singh, C. Zhang, C. Schnell, G. Yang, Y. Zhang, *et al.*, “High-throughput screening using patient-derived tumor xenografts to predict clinical trial drug response,” *Nature medicine*, vol. 21, no. 11, p. 1318, 2015.

[12] M. J. Garnett, E. J. Edelman, S. J. Heidorn, C. D. Greenman, A. Dastur, K. W. Lau, P. Greninger, I. R. Thompson, X. Luo, J. Soares, Q. Liu, F. Iorio, D. Surdez, L. Chen, R. J. Milano, G. R. Bignell, A. T. Tam, H. Davies, J. A. Stevenson, S. Barthorpe, S. R. Lutz, F. Kogera, K. Lawrence, A. McLaren-Douglas, X. Mitropoulos, T. Mironenko, H. Thi, L. Richardson, W. Zhou, F. Jewitt, T. Zhang, P. O'Brien, J. L. Boisvert, S. Price, W. Hur, W. Yang, X. Deng, A. Butler, H. G. Choi, J. W. Chang, J. Baselga, I. Stamenkovic, J. A. Engelman, S. V. Sharma, O. Delattre, J. Saez-Rodriguez, N. S. Gray, J. Settleman, P. A. Futreal, D. A. Haber, M. R. Stratton, S. Ramaswamy, U. McDermott, and C. H. Benes, "Systematic identification of genomic markers of drug sensitivity in cancer cells," *Nature*, vol. 483, pp. 570–575, Mar. 2012.

[13] F. Iorio, T. A. Knijnenburg, D. J. Vis, G. R. Bignell, M. P. Menden, M. Schubert, N. Aben, E. Gonçalves, S. Barthorpe, H. Lightfoot, *et al.*, "A landscape of pharmacogenomic interactions in cancer," *Cell*, vol. 166, no. 3, pp. 740–754, 2016.

[14] P. M. Haverty, E. Lin, J. Tan, Y. Yu, B. Lam, S. Lianoglou, R. M. Neve, S. Martin, J. Settleman, R. L. Yauch, and R. Bourgon, "Reproducible pharmacogenomic profiling of cancer cell line panels," *Nature*, vol. 533, pp. 333–337, May 2016.

[15] C. Klijn, S. Durinck, E. W. Stawiski, P. M. Haverty, Z. Jiang, H. Liu, J. Degenhardt, O. Mayba, F. Gnad, J. Liu, G. Pau, J. Reeder, Y. Cao, K. Mukhyala, S. K. Selvaraj, M. Yu, G. J. Zynda, M. J. Brauer, T. D. Wu, R. C. Gentleman, G. Manning, R. L. Yauch, R. Bourgon, D. Stokoe, Z. Modrusan, R. M. Neve, F. J. de Sauvage, J. Settleman, S. Seshagiri, and Z. Zhang, "A comprehensive transcriptional portrait of human cancer cell lines," *Nat. Biotechnol.*, vol. 33, pp. 306–312, Mar. 2015.

[16] A. Basu, N. E. Bodycombe, J. H. Cheah, E. V. Price, K. Liu, G. I. Schaefer, R. Y. Ebright, M. L. Stewart, D. Ito, S. Wang, A. L. Bracha, T. Liefeld, M. Wawer, J. C. Gilbert, A. J. Wilson, N. Stransky, G. V. Kryukov, V. Dancik, J. Barretina, L. A. Garraway, C. S.-Y. Hon, B. Munoz, J. A. Bittker, B. R. Stockwell, D. Khabele, A. M. Stern, P. A. Clemons, A. F. Shamji, and S. L. Schreiber, "An interactive resource to identify cancer genetic and lineage dependencies targeted by small molecules," *Cell*, vol. 154, pp. 1151–1161, Aug. 2013.

[17] B. Seashore-Ludlow, M. G. Rees, J. H. Cheah, M. Cokol, E. V. Price, M. E. Coletti, V. Jones, N. E. Bodycombe, C. K. Soule, J. Gould, B. Alexander, A. Li, P. Montgomery, M. J. Wawer, N. Kuru, J. D. Kotz, C. S.-Y. Hon, B. Munoz, T. Liefeld, V. Dančík, J. A. Bittker, M. Palmer, J. E. Bradner, A. F. Shamji, P. A. Clemons, and S. L. Schreiber, "Harnessing connectivity in a Large-Scale Small-Molecule sensitivity dataset," *Cancer Discov.*, vol. 5, pp. 1210–1223, Nov. 2015.

[18] J. Barretina, G. Caponigro, N. Stransky, K. Venkatesan, A. A. Margolin, S. Kim, C. J. Wilson, J. Lehár, G. V. Kryukov, D. Sonkin, *et al.*, "The cancer cell line encyclopedia enables predictive modelling of anticancer drug sensitivity," *Nature*, vol. 483, no. 7391, p. 603, 2012.

[19] P. Geeleher, N. J. Cox, and R. S. Huang, "Clinical drug response can be predicted using baseline gene expression levels and in vitro drug sensitivity in cell lines," *Genome biology*, vol. 15, no. 3, p. R47, 2014.

[20] M. Q. Ding, L. Chen, G. F. Cooper, J. D. Young, and X. Lu, "Precision oncology beyond targeted therapy: Combining omics data with machine learning matches the majority of cancer cells to effective therapeutics," *Molecular Cancer Research*, vol. 16, no. 2, pp. 269–278, 2018.

[21] P. Smirnov, V. Kofia, A. Maru, M. Freeman, C. Ho, N. El-Hachem, G.-A. Adam, W. Ba-alawi, Z. Safikhani, and B. Haibe-Kains, "Pharmacodb: an integrative database for mining in vitro anticancer drug screening studies," *Nucleic acids research*, vol. 46, no. D1, pp. D994–D1002, 2017.

[22] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep learning*, vol. 1. MIT Press, 2016.

[23] G. Eraslan, Ž. Avsec, J. Gagneur, and F. J. Theis, "Deep learning: new computational modelling techniques for genomics," *Nature Reviews Genetics*, p. 1, 2019.

[24] P. Geeleher, Z. Zhang, F. Wang, R. F. Gruener, A. Nath, G. Morrison, S. Bhutra, R. L. Grossman, and R. S. Huang, "Discovering novel pharmacogenomic biomarkers by imputing drug response in cancer patients from large genomics studies," *Genome research*, 2017.

[25] J. N. Weinstein, E. A. Collisson, G. B. Mills, K. R. M. Shaw, B. A. Ozenberger, K. Ellrott, I. Shmulevich, C. Sander, J. M. Stuart, C. G. A. R. Network, *et al.*, "The cancer genome atlas pan-cancer analysis project," *Nature genetics*, vol. 45, no. 10, p. 1113, 2013.

[26] Z. Ding, S. Zu, and J. Gu, "Evaluating the molecule-based prediction of clinical drug responses in cancer," *Bioinformatics*, vol. 32, no. 19, pp. 2891–2895, 2016.

[27] K. Graim, V. Friedl, K. E. Houlahan, and J. M. Stuart, "Platypus: A multiple–view learning predictive framework for cancer drug sensitivity prediction," 2018.

[28] J. C. Costello, L. M. Heiser, E. Georgii, M. Gönen, M. P. Menden, N. J. Wang, M. Bansal, P. Hintsanen, S. A. Khan, J.-P. Mpindi, *et al.*, "A community effort to assess and improve drug sensitivity prediction algorithms," *Nature biotechnology*, vol. 32, no. 12, pp. 1202–1212, 2014.

[29] R. Argelaguet, B. Velten, D. Arnol, S. Dietrich, T. Zenz, J. C. Marioni, F. Buettner, W. Huber, and O. Stegle, "Multi-omics factor analysis—a framework for unsupervised integration of multi-omics data sets," *Molecular systems biology*, vol. 14, no. 6, p. e8124, 2018.

[30] B. Wang, A. M. Mezlini, F. Demir, M. Fiume, Z. Tu, M. Brudno, B. Haibe-Kains, and A. Goldenberg, "Similarity network fusion for aggregating data types on a genomic scale," *Nature methods*, vol. 11, no. 3, p. 333, 2014.

[31] C. Dimitrakopoulos, S. K. Hindupur, L. Häfliger, J. Behr, H. Montazeri, M. N. Hall, and N. Beerenwinkel, "Network-based integration of multi-omics data for prioritizing cancer genes," *Bioinformatics*, vol. 34, no. 14, pp. 2441–2448, 2018.

[32] Q. Mo, S. Wang, V. E. Seshan, A. B. Olshen, N. Schultz, C. Sander, R. S. Powers, M. Ladanyi, and R. Shen, "Pattern discovery and cancer gene identification in integrated cancer genomic data," *Proceedings of the National Academy of Sciences*, p. 201208949, 2013.

[33] R. Shrestha, E. Hodzic, T. Sauerwald, P. Dao, K. Wang, J. Yeung, S. Anderson, F. Vandin, G. Haffari, C. C. Collins, *et al.*, "Hit'ndrive: patient-specific multidriver gene prioritization for precision oncology," *Genome research*, 2017.

[34] A. Singh, C. P. Shannon, B. Gautier, F. Rohart, M. Vacher, S. J. Tebbutt, and K.-A. Lê Cao, "Diablo: an integrative approach for identifying key molecular drivers from multi-omic assays," *Bioinformatics*, 2019.

[35] S. Khakabimamaghani and M. Ester, "Bayesian biclustering for patient stratification," in *Biocomputing 2016: Proceedings of the Pacific Symposium*, pp. 345–356, World Scientific, 2016.

[36] K. Chaudhary, O. B. Poirion, L. Lu, and L. X. Garmire, "Deep learning–based multi-omics integration robustly predicts survival in liver cancer," *Clinical Cancer Research*, vol. 24, no. 6, pp. 1248–1259, 2018.

[37] M. Liang, Z. Li, T. Chen, and J. Zeng, "Integrative data analysis of multi-platform cancer data with a multimodal deep learning approach," *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, vol. 12, no. 4, pp. 928–937, 2015.

[38] N. Rappoport and R. Shamir, "Multi-omic and multi-view clustering algorithms: review and cancer benchmark," *Nucleic acids research*, vol. 46, no. 20, pp. 10546–10562, 2018.

[39] M. Zitnik, F. Nguyen, B. Wang, J. Leskovec, A. Goldenberg, and M. M. Hoffman, "Machine learning for integrating data in biology and medicine: Principles, practice, and opportunities," *Information Fusion*, vol. 50, pp. 71–91, 2019.

[40] H. Sharifi-Noghabi, O. Zolotareva, C. C. Collins, and M. Ester, "MOLI: multi-omics late integration with deep neural networks for drug response prediction," *Bioinformatics*, vol. 35, no. 14, pp. i501–i509, 2019.

[41] S. Mourragui, M. Loog, M. A. van de Wiel, M. J. Reinders, and L. F. Wessels, "Precise: a domain adaptation approach to transfer predictors of drug response from pre-clinical models to tumors," *Bioinformatics*, vol. 35, no. 14, pp. i510–i519, 2019.

[42] L. H. Schwartz, S. Litière, E. de Vries, R. Ford, S. Gwyther, S. Mandrekar, L. Shankar, J. Bogaerts, A. Chen, J. Dancey, *et al.*, "Recist 1.1—update and clarification: From the recist committee," *European journal of cancer*, vol. 62, pp. 132–137, 2016.

[43] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[44] M. Raghu, C. Zhang, J. Kleinberg, and S. Bengio, "Transfusion: Understanding transfer learning for medical imaging," in *Advances in Neural Information Processing Systems*, vol. 32, pp. 3347–3357, Curran Associates, Inc., 2019.

[45] B. Neyshabur, H. Sedghi, and C. Zhang, “What is being transferred in transfer learning?,” *Adv. Neural Inf. Process. Syst.*, 2020.

[46] Y. Zhang, T. Liu, M. Long, and M. I. Jordan, “Bridging theory and algorithm for domain adaptation,” *arXiv preprint arXiv:1904.05801*, 2019.

[47] Y. Ganin and V. Lempitsky, “Unsupervised domain adaptation by backpropagation,” *arXiv preprint arXiv:1409.7495*, 2014.

[48] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola, “A kernel two-sample test,” *Journal of Machine Learning Research*, vol. 13, no. Mar, pp. 723–773, 2012.

[49] B. Sun and K. Saenko, “Deep CORAL: Correlation alignment for deep domain adaptation,” in *Computer Vision – ECCV 2016 Workshops*, pp. 443–450, Springer International Publishing, 2016.

[50] S. J. Pan and Q. Yang, “A survey on transfer learning,” *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2009.

[51] A. Schoenauer-Sebag, L. Heinrich, M. Schoenauer, M. Sebag, L. F. Wu, and S. J. Altschuler, “Multi-domain adversarial learning,” *arXiv preprint arXiv:1903.09239*, 2019.

[52] E. Hosseini-Asl, Y. Zhou, C. Xiong, and R. Socher, “Augmented cyclic adversarial learning for low resource domain adaptation,” 2018.

[53] P. O. Pinheiro, “Unsupervised domain adaptation with similarity learning,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8004–8013, 2018.

[54] Y. Zou, Z. Yu, B. Vijaya Kumar, and J. Wang, “Unsupervised domain adaptation for semantic segmentation via class-balanced self-training,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 289–305, 2018.

[55] Y.-H. Tsai, W.-C. Hung, S. Schulter, K. Sohn, M.-H. Yang, and M. Chandraker, “Learning to adapt structured output space for semantic segmentation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7472–7481, 2018.

[56] M. Long, Z. Cao, J. Wang, and M. I. Jordan, “Conditional adversarial domain adaptation,” in *Advances in Neural Information Processing Systems*, pp. 1640–1650, 2018.

[57] Y.-H. Chen, W.-Y. Chen, Y.-T. Chen, B.-C. Tsai, Y.-C. Frank Wang, and M. Sun, “No more discrimination: Cross city adaptation of road scene segmenters,” in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1992–2001, 2017.

[58] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, “Adversarial discriminative domain adaptation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7167–7176, 2017.

[59] O. Snow, H. Sharifi, J. Lu, O. Zolotareva, M. Lee, and M. Ester, “Bdkann-biological domain knowledge-based artificial neural network for drug response prediction,” *bioRxiv*, p. 840553, 2019.

[60] B. M. Kuenzi, J. Park, S. H. Fong, K. S. Sanchez, J. Lee, J. F. Kreisberg, J. Ma, and T. Ideker, “Predicting drug response and synergy using a deep learning model of human cancer cells,” *Cancer Cell*, vol. 38, pp. 672–684.e6, Nov. 2020.

[61] H. Sharifi-Noghabi, S. Peng, O. Zolotareva, C. C. Collins, and M. Ester, “Aitl: Adversarial inductive transfer learning with input and output space adaptation for pharmacogenomics,” *Bioinformatics*, vol. 36, no. Supplement_1, pp. i380–i388, 2020.

[62] S. Mourragui, M. Loog, D. J. Vis, K. Moore, A. G. Manjon, M. A. van de Wiel, M. J. T. Reinders, and L. F. A. Wessels, “PRECISE+ predicts drug response in patients by non-linear subspace-based transfer from cell lines and PDX models,” *bioRxiv*, 2020.

[63] J. Ma, S. H. Fong, Y. Luo, C. J. Bakkenist, J. P. Shen, S. Mourragui, L. F. A. Wessels, M. Hafner, R. Sharan, J. Peng, and T. Ideker, “Few-shot learning creates predictive models of drug response that translate from high-throughput screens to individual patients,” *Nature Cancer*, Jan. 2021.

[64] Y. Zhu, T. Brettin, Y. A. Evrard, A. Partin, F. Xia, M. Shukla, H. Yoo, J. H. Doroshow, and R. L. Stevens, “Ensemble transfer learning for the prediction of anti-cancer drug response,” *Sci. Rep.*, vol. 10, p. 18040, Oct. 2020.

[65] A. Warren, A. Jones, T. Shibue, W. C. Hahn, J. S. Boehm, F. Vazquez, A. Tsherniak, and J. M. McFarland, “Global computational alignment of tumor and cell line transcriptional profiles.” Mar. 2020.

[66] I. Gulrajani and D. Lopez-Paz, “In search of lost domain generalization,” July 2020.

[67] J. Wang, C. Lan, C. Liu, Y. Ouyang, and T. Qin, “Generalizing to unseen domains: A survey on domain generalization,” *arXiv preprint arXiv:2103.03097*, 2021.

[68] K. Zhou, Z. Liu, Y. Qiao, T. Xiang, and C. C. Loy, “Domain generalization: A survey,” *arXiv preprint arXiv:2103.02503*, 2021.

[69] H. Zhang, N. Dullerud, L. Seyyed-Kalantari, Q. Morris, S. Joshi, and M. Ghassemi, “An empirical framework for domain generalization in clinical settings,” in *Proceedings of the Conference on Health, Inference, and Learning*, pp. 279–290, 2021.

[70] Q. Dou, D. C. de Castro, K. Kamnitsas, and B. Glocker, “Domain generalization via model-agnostic learning of semantic features,” in *Advances in Neural Information Processing Systems*, pp. 6447–6458, 2019.

[71] S. Zhao, M. Gong, T. Liu, H. Fu, and D. Tao, “Domain generalization via entropy regularization,” *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[72] Z. Wang, M. Loog, and J. van Gemert, “Respecting domain relations: Hypothesis invariance for domain generalization,” in *2020 25th International Conference on Pattern Recognition (ICPR)*, pp. 9756–9763, IEEE, 2021.

[73] M. Ghifary, W. Bastiaan Kleijn, M. Zhang, and D. Balduzzi, "Domain generalization for object recognition with multi-task autoencoders," in *Proceedings of the IEEE international conference on computer vision*, pp. 2551–2559, 2015.

[74] H. Li, S. Jialin Pan, S. Wang, and A. C. Kot, "Domain generalization with adversarial feature learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5400–5409, 2018.

[75] S. Shankar, V. Piratla, S. Chakrabarti, S. Chaudhuri, P. Jyothi, and S. Sarawagi, "Generalizing across domains via cross-gradient training," *arXiv preprint arXiv:1804.10745*, 2018.

[76] F. M. Carlucci, A. D'Innocente, S. Bucci, B. Caputo, and T. Tommasi, "Domain generalization by solving jigsaw puzzles," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2229–2238, 2019.

[77] H. Sharifi-Noghabi, H. Asghari, N. Mehrasa, and M. Ester, "Domain generalization via semi-supervised meta learning," *arXiv preprint arXiv:2009.12658*, 2020.

[78] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?," in *Advances in neural information processing systems*, pp. 3320–3328, 2014.

[79] X. Li, Y. Grandvalet, and F. Davoine, "Explicit inductive bias for transfer learning with convolutional networks," *ICML*, 2018.

[80] X. Peng, Z. Huang, X. Sun, and K. Saenko, "Domain agnostic learning with disentangled representations," *ICML*, 2019.

[81] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell, "Deep domain confusion: Maximizing for domain invariance," *ICCV*, 2014.

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

[83] K. Azizzadenesheli, A. Liu, F. Yang, and A. Anandkumar, "Regularized learning for domain adaptation under label shifts," *arXiv preprint arXiv:1903.09734*, 2019.

[84] K. You, M. Long, Z. Cao, J. Wang, and M. I. Jordan, "Universal domain adaptation," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[85] Z. Pei, Z. Cao, M. Long, and J. Wang, "Multi-adversarial domain adaptation," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[86] R. Xu, Z. Chen, W. Zuo, J. Yan, and L. Lin, "Deep cocktail network: Multi-source unsupervised domain adaptation with category shift," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3964–3973, 2018.

[87] X. Peng, Q. Bai, X. Xia, Z. Huang, K. Saenko, and B. Wang, "Moment matching for multi-source domain adaptation," *arXiv preprint arXiv:1812.01754*, 2018.

[88] A. Schoenauer-Sebag, L. Heinrich, M. Schoenauer, M. Sebag, L. Wu, and S. Altschuler, "Multi-domain adversarial learning," *International Conference on Learning Representations*, 2019.

[89] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *International Conference on Learning Representations*, 2013.

[90] Z. Cao, M. Long, J. Wang, and M. I. Jordan, "Partial transfer learning with selective adversarial networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2724–2732, 2018.

[91] P. Panareda Busto and J. Gall, "Open set domain adaptation," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 754–763, 2017.

[92] T. Scott, K. Ridgeway, and M. C. Mozer, "Adapted deep embeddings: A synthesis of methods for k-shot inductive transfer learning," in *Advances in Neural Information Processing Systems*, pp. 76–85, 2018.

[93] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," in *Advances in Neural Information Processing Systems*, pp. 4077–4087, 2017.

[94] W.-Y. Chen, Y.-C. Liu, Z. Kira, Y.-C. F. Wang, and J.-B. Huang, "A closer look at few-shot classification," in *International Conference on Learning Representations*, 2019.

[95] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 815–823, 2015.

[96] E. Ustinova and V. Lempitsky, "Learning deep embeddings with histogram loss," in *Advances in Neural Information Processing Systems*, pp. 4170–4178, 2016.

[97] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition," in *ICML deep learning workshop*, vol. 2, 2015.

[98] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra, *et al.*, "Matching networks for one shot learning," in *Advances in neural information processing systems*, pp. 3630–3638, 2016.

[99] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. Torr, and T. M. Hospedales, "Learning to compare: Relation network for few-shot learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1199–1208, 2018.

[100] R. Shao, X. Lan, and P. C. Yuen, "Regularized fine-grained meta face anti-spoofing," *arXiv preprint arXiv:1911.10771*, 2019.

[101] T. Matsuura and T. Harada, "Domain generalization using a mixture of multiple latent domains," *arXiv preprint arXiv:1911.07661*, 2019.

[102] H.-Y. Tseng, H.-Y. Lee, J.-B. Huang, and M.-H. Yang, "Cross-domain few-shot classification via learned feature-wise transformation," *arXiv preprint arXiv:2001.08735*, 2020.

[103] D. Li, J. Zhang, Y. Yang, C. Liu, Y.-Z. Song, and T. M. Hospedales, "Episodic training for domain generalization," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1446–1455, 2019.

[104] Y. Li, Y. Yang, W. Zhou, and T. M. Hospedales, "Feature-critic networks for heterogeneous domain generalization," *arXiv preprint arXiv:1901.11448*, 2019.

[105] Y. Balaji, S. Sankaranarayanan, and R. Chellappa, "Metareg: Towards domain generalization using meta-regularization," in *Advances in Neural Information Processing Systems*, pp. 998–1008, 2018.

[106] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 1126–1135, JMLR. org, 2017.

[107] D. Li, Y. Yang, Y.-Z. Song, and T. M. Hospedales, "Deeper, broader and artier domain generalization," in *Proceedings of the IEEE international conference on computer vision*, pp. 5542–5550, 2017.

[108] A. Torralba and A. A. Efros, "Unbiased look at dataset bias," in *CVPR 2011*, pp. 1521–1528, IEEE, 2011.

[109] L. Fei-Fei, R. Fergus, and P. Perona, "Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories," in *2004 conference on computer vision and pattern recognition workshop*, pp. 178–178, IEEE, 2004.

[110] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes challenge 2007 (voc2007) results," 2007.

[111] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman, "Labelme: a database and web-based tool for image annotation," *International journal of computer vision*, vol. 77, no. 1-3, pp. 157–173, 2008.

[112] M. J. Choi, J. J. Lim, A. Torralba, and A. S. Willsky, "Exploiting hierarchical context on a large database of object categories," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 129–136, IEEE, 2010.

[113] J. Drost and H. Clevers, "Organoids in cancer research," *Nature Reviews Cancer*, vol. 18, no. 7, pp. 407–418, 2018.

[114] L. V. Nguyen and C. Caldas, "Functional genomics approaches to improve pre-clinical drug screening and biomarker discovery," *EMBO Molecular Medicine*, p. e13189, 2021.

[115] J. C. Venter, M. D. Adams, E. W. Myers, P. W. Li, R. J. Mural, G. G. Sutton, H. O. Smith, M. Yandell, C. A. Evans, R. A. Holt, *et al.*, "The sequence of the human genome," *science*, vol. 291, no. 5507, pp. 1304–1351, 2001.

[116] S. Nurk, S. Koren, A. Rhie, M. Rautiainen, A. V. Bzikadze, A. Mikheenko, M. R. Vollger, N. Altemose, L. Uralsky, A. Gershman, S. Aganezov, S. J. Hoyt, M. Diekhans, G. A. Logsdon, M. Alonge, S. E. Antonarakis, M. Borchers, G. G. Bouffard, S. Y. Brooks, G. V. Caldas, H. Cheng, C.-S. Chin, W. Chow, L. G. de Lima, P. C. Dishuck,

R. Durbin, T. Dvorkina, I. T. Fiddes, G. Formenti, R. S. Fulton, A. Fungtammasan, E. Garrison, P. G. Grady, T. A. Graves-Lindsay, I. M. Hall, N. F. Hansen, G. A. Hartley, M. Haukness, K. Howe, M. W. Hunkapiller, C. Jain, M. Jain, E. D. Jarvis, P. Kerpedjiev, M. Kirsche, M. Kolmogorov, J. Korlach, M. Kremitzki, H. Li, V. V. Maduro, T. Marschall, A. M. McCartney, J. McDaniel, D. E. Miller, J. C. Mullikin, E. W. Myers, N. D. Olson, B. Paten, P. Peluso, P. A. Pevzner, D. Porubsky, T. Potapova, E. I. Rogaev, J. A. Rosenfeld, S. L. Salzberg, V. A. Schneider, F. J. Sedlazeck, K. Shafin, C. J. Shew, A. Shumate, Y. Sims, A. F. A. Smit, D. C. Soto, I. Sović, J. M. Storer, A. Streets, B. A. Sullivan, F. Thibaud-Nissen, J. Torrance, J. Wagner, B. P. Walenz, A. Wenger, J. M. D. Wood, C. Xiao, S. M. Yan, A. C. Young, S. Zarate, U. Surti, R. C. McCoy, M. Y. Dennis, I. A. Alexandrov, J. L. Gerton, R. J. O'Neill, W. Timp, J. M. Zook, M. C. Schatz, E. E. Eichler, K. H. Miga, and A. M. Phillippy, "The complete sequence of a human genome," *bioRxiv*, 2021.

[117] S. Khakabimamaghani, *Probabilistic graphical models for the analysis of omics heterogeneity.* PhD thesis, Applied Sciences: School of Computing Science, 2019.

[118] Y. Hasin, M. Seldin, and A. Lusis, "Multi-omics approaches to disease," *Genome biology*, vol. 18, no. 1, pp. 1–15, 2017.

[119] H. Ling, K. Vincent, M. Pichler, R. Fodde, I. Berindan-Neagoe, F. J. Slack, and G. A. Calin, "Junk dna and the long non-coding rna twist in cancer genetics," *Oncogene*, vol. 34, no. 39, pp. 5003–5011, 2015.

[120] J. L. Rinn and H. Y. Chang, "Genome regulation by long noncoding rnas," *Annual review of biochemistry*, vol. 81, pp. 145–166, 2012.

[121] V. R. Ramnarine, M. Kobelev, E. A. Gibb, M. Nouri, D. Lin, Y. Wang, R. Buttyan, E. Davicioni, A. Zoubeidi, and C. C. Collins, "The evolution of long noncoding rna acceptance in prostate cancer initiation, progression, and its clinical utility in disease management," *European urology*, vol. 76, no. 5, pp. 546–559, 2019.

[122] X. Wang, Q. Liu, and B. Zhang, "Leveraging the complementary nature of rna-seq and shotgun proteomics data," *Proteomics*, vol. 14, no. 23-24, pp. 2676–2687, 2014.

[123] S. Koplev, K. Lin, A. B. Dohlman, and A. Ma'ayan, "Integration of pan-cancer transcriptomics with rppa proteomics reveals mechanisms of epithelial-mesenchymal transition," *PLoS computational biology*, vol. 14, no. 1, p. e1005911, 2018.

[124] L. Rampášek, D. Hidru, P. Smirnov, B. Haibe-Kains, and A. Goldenberg, "Dr. vae: improving drug response prediction via modeling of drug perturbation effects," *Bioinformatics*, 2019.

[125] N. Pozdeyev, M. Yoo, R. Mackie, R. E. Schweppe, A. C. Tan, and B. R. Haugen, "Integrating heterogeneous drug sensitivity data from cancer pharmacogenomic studies," *Oncotarget*, vol. 7, no. 32, p. 51619, 2016.

[126] O. D. Abaan, E. C. Polley, S. R. Davis, Y. J. Zhu, S. Bilke, R. L. Walker, M. Pineda, Y. Gindin, Y. Jiang, W. C. Reinhold, *et al.*, "The exomes of the nci-60 panel: a genomic resource for cancer biology and systems pharmacology," *Cancer research*, vol. 73, no. 14, pp. 4372–4382, 2013.

[127] A. Luna, F. Elloumi, S. Varma, Y. Wang, V. N. Rajapakse, M. I. Aladjem, J. Robert, C. Sander, Y. Pommier, and W. C. Reinhold, "Cellminer cross-database (cellminercdb) version 1.2: Exploration of patient-derived cancer cell line pharmacogenomics," *Nucleic Acids Research*, vol. 49, no. D1, pp. D1083–D1093, 2021.

[128] B. Haibe-Kains, N. El-Hachem, N. J. Birkbak, A. C. Jin, A. H. Beck, H. J. W. L. Aerts, and J. Quackenbush, "Inconsistency in large pharmacogenomic studies," *Nature*, vol. 504, pp. 389–393, Dec. 2013.

[129] Z. Safikhani, P. Smirnov, M. Freeman, N. El-Hachem, A. She, Q. Rene, A. Goldenberg, N. J. Birkbak, C. Hatzis, L. Shi, A. H. Beck, H. J. W. L. Aerts, J. Quackenbush, and B. Haibe-Kains, "Revisiting inconsistency in large pharmacogenomic studies," *F1000Res.*, vol. 5, p. 2333, Sept. 2016.

[130] P. Geeleher, E. R. Gamazon, C. Seoighe, N. J. Cox, and R. S. Huang, "Consistency in large pharmacogenomic studies," *Nature*, vol. 540, no. 7631, pp. E1–E2, 2016.

[131] J. P. Mpindi, B. Yadav, P. Östling, P. Gautam, D. Malani, A. Murumägi, A. Hirasawa, S. Kangaspeska, K. Wennerberg, O. Kallioniemi, *et al.*, "Consistency in drug response profiling," *Nature*, vol. 540, no. 7631, pp. E5–E6, 2016.

[132] M. Niepel, M. Hafner, C. E. Mills, K. Subramanian, E. H. Williams, M. Chung, B. Gaudio, A. M. Barrette, A. D. Stern, B. Hu, *et al.*, "A multi-center study on the reproducibility of drug-response assays in mammalian cell lines," *Cell systems*, vol. 9, no. 1, pp. 35–48, 2019.

[133] M. Bouhaddou, M. S. DiStefano, E. A. Riesel, E. Carrasco, H. Y. Holzapfel, D. C. Jones, G. R. Smith, A. D. Stern, S. S. Somani, T. V. Thompson, *et al.*, "Drug response consistency in ccle and cgp," *Nature*, vol. 540, no. 7631, pp. E9–E10, 2016.

[134] A. Mammoliti, P. Smirnov, M. Nakano, Z. Safikhani, C. Ho, G. Beri, and B. Haibe-Kains, "ORCESTRA: a platform for orchestrating and sharing high-throughput pharmacogenomic analyses." Sept. 2020.

[135] J. Chen and L. Zhang, "A survey and systematic assessment of computational methods for drug response prediction," *bioRxiv*, p. 697896, 2019.

[136] P. B. Güvenç, H. Mamitsuka, and S. Kaski, "Improving drug response prediction by integrating multiple data sources: matrix factorization, kernel and network-based approaches.," *Briefings in bioinformatics*, 2019.

[137] E. W. Huang, A. Bhope, J. Lim, S. Sinha, and A. Emad, "Tissue-guided lasso for prediction of clinical drug response using preclinical samples," *PLoS computational biology*, vol. 16, no. 1, p. e1007607, 2020.

[138] G. Riddick, H. Song, S. Ahn, J. Walling, D. Borges-Rivera, W. Zhang, and H. A. Fine, "Predicting in vitro drug sensitivity using random forests," *Bioinformatics*, vol. 27, no. 2, pp. 220–224, 2011.

[139] X. He, L. Folkman, K. Borgwardt, and J. Wren, "Kernelized rank learning for personalized drug recommendation," *Bioinformatics*, vol. 1, p. 9, 2018.

[140] T. Sakellaropoulos, K. Vougas, S. Narang, F. Koinis, A. Kotsinas, A. Polyzos, T. J. Moss, S. Piha-Paul, H. Zhou, E. Kardala, *et al.*, "A deep learning framework for predicting response to therapy in cancer," *Cell Reports*, vol. 29, no. 11, pp. 3367–3373, 2019.

[141] J. Born, M. Manica, A. Oskooei, J. Cadow, and M. R. Martínez, "Paccmann rl: Designing anticancer drugs from transcriptomic data via reinforcement learning," in *International Conference on Research in Computational Molecular Biology*, pp. 231–233, 2020.

[142] Y. Jiang, S. Rensi, S. Wang, and R. B. Altman, "DrugOrchestra: Jointly predicting drug response, targets, and side effects via deep multi-task learning." Nov. 2020.

[143] I. S. Jang, E. C. Neto, J. Guinney, S. H. Friend, and A. A. Margolin, "Systematic assessment of analytical methods for drug sensitivity prediction from cancer cell line data," in *Biocomputing 2014*, pp. 63–74, World Scientific, 2014.

[144] R. Peres da Silva, C. Suphavilai, and N. Nagarajan, "Tugda: task uncertainty guided domain adaptation for robust generalization of cancer drug response prediction from in vitro to in vivo settings," *Bioinformatics*, 2021.

[145] M. Sajjadi, M. Javanmardi, and T. Tasdizen, "Regularization with stochastic transformations and perturbations for deep semi-supervised learning," in *Advances in neural information processing systems*, pp. 1163–1171, 2016.

[146] D.-H. Lee, "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks," in *Workshop on challenges in representation learning, ICML*, vol. 3, p. 2, 2013.

[147] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," Mar. 2017.

[148] K. Sohn, D. Berthelot, C.-L. Li, Z. Zhang, N. Carlini, E. D. Cubuk, A. Kurakin, H. Zhang, and C. Raffel, "Fixmatch: Simplifying semi-supervised learning with consistency and confidence," *arXiv preprint arXiv:2001.07685*, 2020.

[149] M. Ren, E. Triantafillou, S. Ravi, J. Snell, K. Swersky, J. B. Tenenbaum, H. Larochelle, and R. S. Zemel, "Meta-learning for semi-supervised few-shot classification," *arXiv preprint arXiv:1803.00676*, 2018.

[150] Y. Yang and Z. Xu, "Rethinking the value of labels for improving Class-Imbalanced learning," June 2020.

[151] W. Yang, J. Soares, P. Greninger, E. J. Edelman, H. Lightfoot, S. Forbes, N. Bindal, D. Beare, J. A. Smith, I. R. Thompson, *et al.*, "Genomics of drug sensitivity in cancer (gdsc): a resource for therapeutic biomarker discovery in cancer cells," *Nucleic acids research*, vol. 41, no. D1, pp. D955–D961, 2012.

[152] R. A. Irizarry, B. M. Bolstad, F. Collin, L. M. Cope, B. Hobbs, and T. P. Speed, "Summaries of affymetrix genechip probe level data," *Nucleic acids research*, vol. 31, no. 4, pp. e15–e15, 2003.

[153] B. Li and C. N. Dewey, “Rsem: accurate transcript quantification from rna-seq data with or without a reference genome,” *BMC bioinformatics*, vol. 12, no. 1, p. 323, 2011.

[154] W. E. Johnson, C. Li, and A. Rabinovic, “Adjusting batch effects in microarray expression data using empirical bayes methods,” *Biostatistics*, vol. 8, no. 1, pp. 118–127, 2007.

[155] A. Cichocki and A.-H. Phan, “Fast local algorithms for large scale nonnegative matrix and tensor factorizations,” *IEICE transactions on fundamentals of electronics, communications and computer sciences*, vol. 92, no. 3, pp. 708–721, 2009.

[156] C. Févotte and J. Idier, “Algorithms for nonnegative matrix factorization with the β-divergence,” *Neural computation*, vol. 23, no. 9, pp. 2421–2456, 2011.

[157] J. Duchi, E. Hazan, and Y. Singer, “Adaptive subgradient methods for online learning and stochastic optimization,” *Journal of Machine Learning Research*, vol. 12, no. Jul, pp. 2121–2159, 2011.

[158] M. Ali, S. A. Khan, K. Wennerberg, and T. Aittokallio, “Global proteomics profiling improves drug sensitivity prediction: results from a multi-omics, pan-cancer modeling approach,” *Bioinformatics*, vol. 34, no. 8, pp. 1353–1362, 2017.

[159] C. J. Ryan, S. Kennedy, I. Bajrami, D. Matallanas, and C. J. Lord, “A compendium of co-regulated protein complexes in breast cancer reveals collateral loss events,” *Cell systems*, vol. 5, no. 4, pp. 399–409, 2017.

[160] E. Gonçalves, A. Fragoulis, L. Garcia-Alonso, T. Cramer, J. Saez-Rodriguez, and P. Beltrao, “Widespread post-transcriptional attenuation of genomic copy-number variation in cancer,” *Cell systems*, vol. 5, no. 4, pp. 386–398, 2017.

[161] R. Hadsell, S. Chopra, and Y. LeCun, “Dimensionality reduction by learning an invariant mapping,” in *null*, pp. 1735–1742, IEEE, 2006.

[162] H. Yuan, I. Paskov, H. Paskov, A. J. González, and C. S. Leslie, “Multitask learning improves prediction of cancer drug sensitivity,” *Scientific reports*, vol. 6, p. 31619, 2016.

[163] T. Ching, D. S. Himmelstein, B. K. Beaulieu-Jones, A. A. Kalinin, B. T. Do, G. P. Way, E. Ferrero, P.-M. Agapow, M. Zietz, M. M. Hoffman, *et al.*, “Opportunities and obstacles for deep learning in biology and medicine,” *Journal of The Royal Society Interface*, vol. 15, no. 141, p. 20170387, 2018.

[164] V. Almendro, A. Marusyk, and K. Polyak, “Cellular heterogeneity and molecular evolution in cancer,” *Annual Review of Pathology: Mechanisms of Disease*, vol. 8, pp. 277–302, 2013.

[165] B. Vogelstein, N. Papadopoulos, V. E. Velculescu, S. Zhou, L. A. Diaz, and K. W. Kinzler, “Cancer genome landscapes,” *science*, vol. 339, no. 6127, pp. 1546–1558, 2013.

[166] J. Ma, M. K. Yu, S. Fong, K. Ono, E. Sage, B. Demchak, R. Sharan, and T. Ideker, “Using deep learning to model the hierarchical structure and function of a cell,” *Nature methods*, vol. 15, no. 4, p. 290, 2018.

[167] S. B. Amin, W.-K. Yip, S. Minvielle, A. Broyl, Y. Li, B. Hanlon, D. Swanson, P. K. Shah, P. Moreau, B. van der Holt, *et al.*, “Gene expression profile alone is inadequate in predicting complete response in multiple myeloma,” *Leukemia*, vol. 28, no. 11, p. 2229, 2014.

[168] G. Mulligan, C. Mitsiades, B. Bryant, F. Zhan, W. J. Chng, S. Roels, E. Koenig, A. Fergus, Y. Huang, P. Richardson, *et al.*, “Gene expression profiling and correlation with outcome in clinical trials of the proteasome inhibitor bortezomib,” *Blood*, vol. 109, no. 8, pp. 3177–3188, 2007.

[169] D. P. Silver, A. L. Richardson, A. C. Eklund, Z. C. Wang, Z. Szallasi, Q. Li, N. Juul, C.-O. Leong, D. Calogrias, A. Buraimoh, *et al.*, “Efficacy of neoadjuvant cisplatin in triple-negative breast cancer,” *Journal of clinical oncology*, vol. 28, no. 7, p. 1145, 2010.

[170] D. C. Marchion, H. M. Cottrill, Y. Xiong, N. Chen, E. Bicaku, W. J. Fulp, N. Bansal, H. S. Chon, X. B. Stickles, S. G. Kamath, *et al.*, “Bad phosphorylation determines ovarian cancer chemosensitivity and patient survival,” *Clinical Cancer Research*, vol. 17, no. 19, pp. 6356–6366, 2011.

[171] C. Hatzis, L. Pusztai, V. Valero, D. J. Booser, L. Esserman, A. Lluch, T. Vidaurre, F. Holmes, E. Souchon, H. Wang, *et al.*, “A genomic predictor of response and survival following taxane-anthracycline chemotherapy for invasive breast cancer,” *Jama*, vol. 305, no. 18, pp. 1873–1881, 2011.

[172] B. D. Lehmann, J. A. Bauer, X. Chen, M. E. Sanders, A. B. Chakravarthy, Y. Shyr, and J. A. Pietenpol, “Identification of human triple-negative breast cancer subtypes and preclinical models for selection of targeted therapies,” *The Journal of clinical investigation*, vol. 121, no. 7, pp. 2750–2767, 2011.

[173] J. C. Chang, E. C. Wooten, A. Tsimelzon, S. G. Hilsenbeck, M. C. Gutierrez, Y.-L. Tham, M. Kalidas, R. Elledge, S. Mohsin, C. K. Osborne, *et al.*, “Patterns of resistance and incomplete response to docetaxel by gene expression profiling in breast cancer patients,” *Journal of Clinical Oncology*, vol. 23, no. 6, pp. 1169–1177, 2005.

[174] J. A. Bauer, A. B. Chakravarthy, J. M. Rosenbluth, D. Mi, E. H. Seeley, N. D. M. Granja-Ingram, M. G. Olivares, M. C. Kelley, I. A. Mayer, I. M. Meszoely, *et al.*, “Identification of markers of taxane sensitivity using proteomic and genomic analyses of breast tumors from patients receiving neoadjuvant paclitaxel and radiation,” *Clinical Cancer Research*, vol. 16, no. 2, pp. 681–690, 2010.

[175] A. A. Ahmed, A. D. Mills, A. E. Ibrahim, J. Temple, C. Blenkiron, M. Vias, C. E. Massie, N. G. Iyer, A. McGeoch, R. Crawford, *et al.*, “The extracellular matrix protein tgfbi induces microtubule stabilization and sensitizes ovarian cancers to paclitaxel,” *Cancer cell*, vol. 12, no. 6, pp. 514–527, 2007.

[176] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, “Domain-adversarial training of neural networks,” *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 2096–2030, 2016.

[177] J. Yang, Y. Yu, W. Liu, Z. Li, Z. Wei, and R. Jiang, “Microtubule-associated protein tau is associated with the resistance to docetaxel in prostate cancer cell lines,” *Research and reports in urology*, vol. 9, p. 71, 2017.

[178] M. Smoter, L. Bodnar, R. Duchnowska, R. Stec, B. Grala, and C. Szczylik, “The role of tau protein in resistance to paclitaxel,” *Cancer chemotherapy and pharmacology*, vol. 68, no. 3, pp. 553–557, 2011.

[179] K. M. Bhat and V. Setaluri, “Microtubule-associated proteins as targets in cancer chemotherapy,” *Clinical Cancer Research*, vol. 13, no. 10, pp. 2849–2854, 2007.

[180] A. Rouette, A. Trofimov, D. Haberl, G. Boucher, V.-P. Lavallée, G. D’Angelo, J. Hébert, G. Sauvageau, S. Lemieux, and C. Perreault, “Expression of immunoproteasome genes is regulated by cell-intrinsic and–extrinsic factors in human cancers,” *Scientific reports*, vol. 6, p. 34019, 2016.

[181] P. Tsvetkov, E. Sokol, D. Jin, Z. Brune, P. Thiru, M. Ghandi, L. A. Garraway, P. B. Gupta, S. Santagata, L. Whitesell, *et al.*, “Suppression of 19s proteasome subunits marks emergence of an altered cell state in diverse cancers,” *Proceedings of the National Academy of Sciences*, vol. 114, no. 2, pp. 382–387, 2017.

[182] Y. Li, J. Huang, J. Sun, S. Xiang, D. Yang, X. Ying, M. Lu, H. Li, and G. Ren, “The transcription levels and prognostic values of seven proteasome alpha subunits in human cancers,” *Oncotarget*, vol. 8, no. 3, p. 4501, 2017.

[183] E. Collignon, A. Canale, C. Al Wardi, M. Bizet, E. Calonne, S. Dedeurwaerder, S. Garaud, C. Naveaux, W. Barham, A. Wilson, *et al.*, “Immunity drives tet1 regulation in cancer through nf-κb,” *Science advances*, vol. 4, no. 6, p. eaap7309, 2018.

[184] H. Hideshima, Y. Yoshida, H. Ikeda, M. Hide, A. Iwasaki, K. C. Anderson, and T. Hideshima, “Ikkβ inhibitor in combination with bortezomib induces cytotoxicity in breast cancer cells,” *International journal of oncology*, vol. 44, no. 4, pp. 1171–1176, 2014.

[185] S. Manna, B. Singha, S. A. Phyo, H. R. Gatla, T.-P. Chang, S. Sanacora, S. Ramaswami, and I. Vancurova, “Proteasome inhibition by bortezomib increases il-8 expression in androgen-independent prostate cancer cells: the role of ikkα,” *The Journal of Immunology*, vol. 191, no. 5, pp. 2837–2846, 2013.

[186] Y. Zhao, N. R. Foster, J. P. Meyers, S. P. Thomas, D. W. Northfelt, K. M. Rowland Jr, B. I. Mattar, D. B. Johnson, J. R. Molina, S. J. Mandrekar, *et al.*, “A phase i/ii study of bortezomib in combination with paclitaxel, carboplatin, and concurrent thoracic radiation therapy for non–small-cell lung cancer: North central cancer treatment group (ncctg)-n0321,” *Journal of Thoracic Oncology*, vol. 10, no. 1, pp. 172–180, 2015.

[187] H. Sharifi-Noghabi, Y. Liu, N. Erho, R. Shrestha, M. Alshalalfa, E. Davicioni, C. C. Collins, and M. Ester, “Deep genomic signature for early metastasis prediction in prostate cancer,” *BioRxiv*, p. 276055, 2019.

[188] H. Sharifi-Noghabi, P. A. Harjandi, O. Zolotareva, C. C. Collins, and M. Ester, “Velodrome: Out-of-distribution generalization from labeled and unlabeled gene expression data for drug response prediction,” *bioRxiv*, 2021.

[189] P. Smirnov, Z. Safikhani, N. El-Hachem, D. Wang, A. She, C. Olsen, M. Freeman, H. Selby, D. M. A. Gendoo, P. Grossmann, A. H. Beck, H. J. W. L. Aerts, M. Lupien, A. Goldenberg, and B. Haibe-Kains, "PharmacoGx: an R package for analysis of large pharmacogenomic datasets," *Bioinformatics*, vol. 32, pp. 1244–1246, Apr. 2016.

[190] H. S. Noghabi, S. Jahangiri-Tazehkand, C. Hon, P. Smirnov, A. Mammoliti, S. K. Nair, A. S. Mer, M. Ester, and B. Haibe-Kains, "Drug sensitivity prediction from cell line-based pharmacogenomics data: Guidelines for developing machine learning models," *bioRxiv*, 2021.

[191] L. A. Byers, L. Diao, J. Wang, P. Saintigny, L. Girard, M. Peyton, L. Shen, Y. Fan, U. Giri, P. K. Tumula, and Others, "An epithelial–mesenchymal transition gene signature predicts resistance to EGFR and PI3K inhibitors and identifies axl as a therapeutic target for overcoming EGFR inhibitor resistance," *Clin. Cancer Res.*, vol. 19, no. 1, pp. 279–290, 2013.

[192] M. Manica, A. Oskooei, J. Born, V. Subramanian, J. Sáez-Rodríguez, and M. R. Martínez, "Towards explainable anticancer compound sensitivity prediction via multimodal attention-based convolutional encoders," *arXiv preprint arXiv:1904.11223*, 2019.

[193] K. Saito, K. Watanabe, Y. Ushiku, and T. Harada, "Maximum classifier discrepancy for unsupervised domain adaptation," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, June 2018.

[194] M. Manica, A. Oskooei, J. Born, V. Subramanian, J. Sáez-Rodríguez, and M. Rodríguez Martínez, "Toward explainable anticancer compound sensitivity prediction via multimodal Attention-Based convolutional encoders," *Mol. Pharm.*, vol. 16, pp. 4797–4806, Dec. 2019.

[195] K. S. Chaudhary, P. D. Abel, and E.-N. Lalani, "Role of the bcl-2 gene family in prostate cancer progression and its implications for therapeutic intervention.," *Environmental Health Perspectives*, vol. 107, no. suppl 1, pp. 49–57, 1999.

[196] S. Catz and J. Johnson, "Bcl-2 in prostate cancer: a minireview," *Apoptosis*, vol. 8, no. 1, pp. 29–37, 2003.

[197] W. Renner, U. Langsenlehner, S. Krenn-Pilko, P. Eder, and T. Langsenlehner, "Bcl2 genotypes and prostate cancer survival," *Strahlentherapie Und Onkologie*, vol. 193, no. 6, pp. 466–471, 2017.

[198] F. Paraf, J. Gogusev, Y. Chrétien, and D. Droz, "Expression of bcl-2 oncoprotein in renal cell tumours," *The Journal of pathology*, vol. 177, no. 3, pp. 247–252, 1995.

[199] J. M. Adams and S. Cory, "The bcl-2 apoptotic switch in cancer development and therapy," *Oncogene*, vol. 26, no. 9, pp. 1324–1337, 2007.

[200] Z. He, H. Liu, H. Moch, and H.-U. Simon, "Machine learning with autophagy-related proteins for discriminating renal cell carcinoma subtypes," *Scientific reports*, vol. 10, no. 1, pp. 1–7, 2020.

[201] S. K. Martin, M. Kamelgarn, and N. Kyprianou, "Cytoskeleton targeting value in prostate cancer treatment," *American journal of clinical and experimental urology*, vol. 2, no. 1, p. 15, 2014.

[202] R. S. Kelly, J. A. Sinnott, J. R. Rider, E. M. Ebot, T. Gerke, M. Bowden, A. Pettersson, M. Loda, H. D. Sesso, P. W. Kantoff, *et al.*, "The role of tumor metabolism as a driver of prostate cancer progression and lethal disease: results from a nested case-control study," *Cancer & metabolism*, vol. 4, no. 1, pp. 1–9, 2016.

[203] K. Numakura, N. Tsuchiya, S. Akihama, T. Inoue, S. Narita, M. Huang, S. Satoh, and T. Habuchi, "Successful mammalian target of rapamycin inhibitor maintenance therapy following induction chemotherapy with gemcitabine and doxorubicin for metastatic sarcomatoid renal cell carcinoma," *Oncology letters*, vol. 8, no. 1, pp. 464–466, 2014.

[204] S. S. El Sheikh, J. Domin, P. Abel, G. Stamp, and E.-N. Lalani, "Phosphorylation of both egfr and erbb2 is a reliable predictor of prostate cancer cell proliferation in response to egf," *Neoplasia*, vol. 6, no. 6, pp. 846–853, 2004.

[205] J.-C. Pignon, B. Koopmansch, G. Nolens, L. Delacroix, D. Waltregny, and R. Winkler, "Androgen receptor controls egfr and erbb2 gene expression at different levels in prostate cancer cell lines," *Cancer research*, vol. 69, no. 7, pp. 2941–2949, 2009.

[206] A. Reid, L. Vidal, H. Shaw, and J. de Bono, "Dual inhibition of erbb1 (egfr/her1) and erbb2 (her2/neu)," *European journal of cancer*, vol. 43, no. 3, pp. 481–489, 2007.

[207] M. S. Gordon, M. Hussey, R. B. Nagle, P. N. Lara Jr, P. C. Mack, J. Dutcher, W. Samlowski, J. I. Clark, D. I. Quinn, C.-X. Pan, *et al.*, "Phase ii study of erlotinib in patients with locally advanced or metastatic papillary histology renal cell cancer: Swog s0317," *Journal of Clinical Oncology*, vol. 27, no. 34, p. 5788, 2009.

[208] M. Ali, S. A. Khan, K. Wennerberg, and T. Aittokallio, "Global proteomics profiling improves drug sensitivity prediction: results from a multi-omics, pan-cancer modeling approach," *Bioinformatics*, vol. 34, no. 8, pp. 1353–1362, 2018.

[209] M. P. Menden, F. P. Casale, J. Stephan, G. R. Bignell, F. Iorio, U. McDermott, M. J. Garnett, J. Saez-Rodriguez, and O. Stegle, "The germline genetic component of drug sensitivity in cancer cell lines," *Nature communications*, vol. 9, no. 1, pp. 1–8, 2018.

[210] K. Y. Michael, J. Ma, J. Fisher, J. F. Kreisberg, B. J. Raphael, and T. Ideker, "Visible machine learning for biomedicine," *Cell*, vol. 173, no. 7, pp. 1562–1565, 2018.

[211] F. Xia, J. Allen, P. Balaprakash, T. Brettin, C. Garcia-Cardona, A. Clyde, J. Cohn, J. Doroshow, X. Duan, V. Dubinkina, *et al.*, "A cross-study analysis of drug response prediction in cancer cell lines," *arXiv preprint arXiv:2104.08961*, 2021.

[212] B. Berger and H. Cho, "Emerging technologies towards enhancing privacy in genomic data sharing," 2019.

[213] H. Cho, S. Simmons, R. Kim, and B. Berger, "Privacy-preserving biomedical database queries with optimal privacy-utility trade-offs," *Cell systems*, vol. 10, no. 5, pp. 408–416, 2020.

[214] W. Zhao, J. Li, M.-J. M. Chen, Y. Luo, Z. Ju, N. K. Nesser, K. Johnson-Camacho, C. T. Boniface, Y. Lawrence, N. T. Pande, *et al.*, “Large-scale characterization of drug responses of clinically relevant proteins in cancer cell lines,” *Cancer Cell*, vol. 38, no. 6, pp. 829–843, 2020.

[215] D. Lähnemann, J. Köster, E. Szczurek, D. J. McCarthy, S. C. Hicks, M. D. Robinson, C. A. Vallejos, K. R. Campbell, N. Beerenwinkel, A. Mahfouz, *et al.*, “Eleven grand challenges in single-cell data science,” *Genome biology*, vol. 21, no. 1, pp. 1–35, 2020.

[216] G. Consortium *et al.*, “The genotype-tissue expression (gtex) pilot analysis: Multitissue gene regulation in humans,” *Science*, vol. 348, no. 6235, pp. 648–660, 2015.

[217] H. Zou and T. Hastie, “Regularization and variable selection via the elastic net,” *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 67, no. 2, pp. 301–320, 2005.

[218] H. Seo, D. Tkachuk, C. Ho, A. Mammoliti, A. Rezaie, S. A. Madani Tonekaboni, and B. Haibe-Kains, “Synergxdb: an integrative pharmacogenomic portal to identify synergistic drug combinations for precision oncology,” *Nucleic acids research*, vol. 48, no. W1, pp. W494–W501, 2020.

Appendix A

A.1 Research reproducibility

All codes, trained models, and preprocessed datasets to reproduce the experimental results for MOLI, AITL, and Velodrome are publicly available:

For MOLI, the code and the models are available here:
`https://github.com/hosseinshn/MOLI`

The preprocessed data is available here:
`https://zenodo.org/record/4036592`

For AITL, all materials (the code, the trained models, and the preprocessed data) are available here:
`https://github.com/hosseinshn/AITL`

For Velodrome, the code is available here:
`https://github.com/hosseinshn/Velodrome`

The preprocessed data and the trained models are available here:
`https://zenodo.org/record/4793442#.YK1HVqhKiUk`