Leveraging Structure and Knowledge in Clinical and Biomedical Representation Learning

by

Matthew B. A. McDermott

B.S., Harvey Mudd College (2014)
S.M., Massachusetts Institute of Technology (2019)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 2022

Author . . . .
Department of Electrical Engineering and Computer Science
May 13, 2022

Certified by . . . .
Peter Szolovits
Professor of Electrical Engineering and Computer Science
Thesis Supervisor

Accepted by . . . .
Leslie A. Kolodziejski
Professor of Electrical Engineering and Computer Science
Chair, Department Committee on Graduate Students

Leveraging Structure and Knowledge in Clinical and Biomedical Representation Learning

by

Matthew B. A. McDermott

Submitted to the Department of Electrical Engineering and Computer Science
on May 13, 2022, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

Datasets in the machine learning for health and biomedicine domain are often noisy, irregularly sampled, only sparsely labeled, and small relative to the dimensionality of the both the data and the tasks. These problems motivate the use of *representation learning* in this domain, which encompasses a variety of techniques designed to produce representations of a dataset that are amenable to downstream modelling tasks. Representation learning in this domain can also take advantage of the significant external knowledge in the biomedical domain. In this thesis, I will explore novel pre-training and representation learning strategies for biomedical data which leverage external structure or knowledge to inform learning at both local and global scales. These techniques will be explored in 4 chapters: (1) leveraging unlabeled data to infer distributional constraints in a semi-supervised learning setting; (2) using graph convolutional neural networks over gene-gene co-regulatory networks to improve modelling of gene expression data; (3) adapting pre-training techniques from natural language processing to electronic health record data, and showing that novel methods are needed for electronic health record timeseries data; and (4) asserting global structure in pre-training applications through structure-inducing pre-training.

Thesis Supervisor: Peter Szolovits
Title: Professor of Electrical Engineering and Computer Science

Acknowledgments

Firstly, I need to acknowledge my family, including my parents Shannon and Paul McDermott and my brother Josh and sister Cat for their constant love and support. My Aunt Ann McDermott also needs to be mentioned here, for her constant willingness to answer my many questions about science and research. Beyond my family, I also need to acknowledge my partner, Dina Nash, for her incredible support as I've finished my PhD. She has enriched my life in so many ways, the least of which in finally nudging me to actually learn a bit about the Cambridge/Boston area.

I've also had the privilege of working extensively with my office-mates Emily Alsentzer, Willie Boag, and Sam Finlayson. Whether our conversations were related to ongoing projects, or from our white-board list of "unproductive conversation topics", they have enriched PhD experience. I also need to acknowledge all my other lab-mates within Pete's group, including Geeticka Chauhan, Di Jin, Harry Tsu, Wei-hung Weng, Brendan Yap, Annamarie Bair, Anu Vajapey, Jack Murphy, & Evan Kim. I was also very lucky to have the opportunity to visit Professor Marzyeh Ghassemi's group in Toronto, and have worked extensively with people there, including Bret Nestor, Shirly Wang, Haoran Zhang, Laleh Seyye-Kalantari, Amy Lu, & Kevin Zhang.

Finally, I need to acknowledge several people who have all been incredible mentors to me throughout my time as a PhD student. From the very first day I arrived in Pete's group, Marzyeh Ghassemi has helped push me forward as a researcher—be it through insisting I do indeed need a desk, encouraging me to run the group's reading group, or helping drive my first MIT research project. While I am not, contrary to her opinion, a spy, I certainly am a better researcher in all aspects due to her guidance. Tristan Naumann has also been a central mentor to me; through his insidious influence, I must confess I have become somewhat of an NLP person. Though I only met Professor Marinka Zitnik later in my PhD, she rapidly became a central mentor in my studies of graph neural networks, structure-inducing pre-training, and many other aspects of the academic career path. Last, but far from least, I also need to acknowledge my advisor Professor Pete Szolovits. Pete has been fantastic in all aspects as an advisor, and

I'm incredibly grateful for the opportunities afforded to me in my time in his group. In particular, Pete has allowed me to explore all kinds of research projects within machine learning for healthcare and has been ever patient with my seeming inability to say no to new projects. Simultaneously, his insights have been crucial for both my various research projects directly and for my development as a researcher overall.

Three of these mentors also compose my overall thesis committee: namely, Professors Pete Szolovits, Marzyeh Ghassemi, and Marinka Zitnik. In addition to everything mentioned above, their input has been invaluable in the finalization of this thesis.

Finally, I also need to acknowledge all funding sources that have supported the works contained in this thesis. In particular, these works have been supported in part by a collaborative research agreement with IBM, and grants from the National Institutes of Health (NIH): National Institute of Mental Health (NIMH) grant P50-MH106933, National Human Genome Research Institute (NHGRI) grant U54-HG007963, National Institute of Health (NIH) grant LM013337. Resources used in preparing this research were also provided, in part, by the Province of Ontario, the Government of Canada through CIFAR, and companies sponsoring the Vector Institute (www.vectorinstitute.ai/partners).

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

1.1.1 Challenges in Machine Learning for Health & Biomedicine

Machine learning for health and biomedicine is one of the most exciting areas of machine learning research today, heavily evangelized both among academics [84, 145] and industry leaders [212, 80]. Clinical and biomedical machine learning promises to diagnose diseases [54, 73], forecast patient state [134, 139], build treatment plans [159, 104], reduce healthcare spending by easing clinical operations [241, 142], or augment biomedical research pipelines such as drug discovery or protein folding prediction engines [183, 197].

However, modelling clinical and biomedical data is very challenging. Firstly, the data modality itself is difficult to work with; it is composed of many disparate, sporadically recorded modalities, has very high dimensionality relative to dataset size, is rife with confounders and hidden biases, and suffers from both high rates of noise and missingness [66, 28, 225]. Secondly, nuanced understanding of clinical and biomedical data and tasks also requires significant human expertise. Unlike in an image recognition model, where any human can simply inspect an image and identify what objects are featured therein, in biomedical applications, understanding what a model *should* predict (and *why* it should make that prediction) can be very challenging.

As a result, it can be more difficult to identify and induce appropriate inductive bias in health domains as compared to general domain machine learning—*i.e.*,, it can be more difficult to identify the kinds of relationships we want the latent spaces of models we train to satisfy. Finally, third, there are many instances within clinical and biomedical machine learning where large datasets will *never* be available, making solving few-shot learning problems absolutely necessary. For example, there simply may not be enough patients suffering from certain rare or newly emerging diseases to use highly data-intensive learning techniques.

One modelling strategy that can address all of these challenges is *representation learning*. Representation learning is a general term for the process of automatically learning how to represent (or featurize) data for a particular problem or a particular set of problems. Representation learning approaches encompass traditional, unsupervised methods such as manual featurization, principal component analysis, matrix factorization, or nonlinear autoencoders, as well as more recent approaches using large-scale self- or distantly-supervised *pre-training* regimes, particularly with self-attention or transformer architectures [154, 45, 91, 172, 4, 201]. Under these pre-training algorithms, a model will be first trained to solve an auxiliary task on a pre-training dataset, then transferred to the fine-tuning context and further specialized. Pre-training algorithms have been very effective in a variety of sub-specialties of machine learning, most notably computer vision (CV) and natural language processing (NLP). In part, these strategies can be so effective because they allow us to leverage large, un- or weakly-labeled datasets that are task-independent to induce forms of inductive bias and prior knowledge into our model.

Given their success in other domains, many researchers have explored representation learning and pre-training techniques in biomedical contexts. However, adapting approaches used in other domains naïvely to the biomedical context has not universally achieved the same performance benefits in these new settings as was observed in the original domains. One key reason that these methods have not performed as well in this setting relates to how these methods make use of and impose *structure* and *knowledge* over this data modality.

1.1.2 Structure & Knowledge over Biomedical Data

To see why structure and knowledge are so critical in representation learning and pre-training, we first need to define these terms. To that end, suppose we have a dataset $\boldsymbol{X} \in \mathcal{X}^N$. In this thesis, we use *structure* to refer to both (1) local (*e.g.*, per-sample) relationships that can be assumed to exist between individual features or sets of features within a single sample $\boldsymbol{x} \in \mathcal{X}$ and (2) global (*e.g.*, cross-sample) relationships between whole samples within the data domain $\mathcal{X}$. For example, if we perform a property prediction task over proteins, we may wish to leverage the primary, secondary, or tertiary structure of each individual protein in how we design our neural network model (an example of local structure), or we may wish to regularize the latent space of the model to reflect known relationships from a protein-protein interaction network (global structure).

In this example, both of these forms of structural constraints are induced from our *external knowledge* of the domain and problem. In particular, our knowledge of the secondary and tertiary structure of a protein, and its known relationships in a interaction network, are both based on external scientific research about this problem. While we don't always need to leverage external knowledge to impose structural constraints (*e.g.*,, you can also learn structural constraints to impose from raw data directly), it is particularly useful to leverage knowledge to inform structural constraints in the health and biomedical domains because they allow us to induce meaningful inductive biases into the learning processes, which can greatly aid in learning.

Note that in addition to structure and knowledge being great tools for representation learning in these domains, they also pose unique challenges as compared to other domains of machine learning. For example, we can note that the underlying local structure of biomedical data is often very different than that of data in other domains. For example, whereas images and text sequences are both well explored in machine learning, biomedical data often comes in other, more complex representations, such as graphs (e.g., molecular graphs, protein pathways or co-regulatory networks); multi-dimensional dynamical systems (e.g., protein or molecular conformations in space, epigenetic exposure data, or pharmacodynamic systems); or higher-resolution, multi-scale, or multi-modal data (e.g., ECG waveform data, pathology slides, clinical notes, drug-gene-tissue interaction networks). Similarly, it is also more challenging in the biomedical domain to identify and enforce appropriate forms of global structure during modelling. Not only are the appropriate relationships between samples harder to determine in general, due to the inherent uncertainty and complexity of the biomedical domain, but also they may take on more varied relationships than we find in other domains; for example, protein-protein relationships can be described by interactions, pathway membership, and co-regulatory information. Overall, the effective use of structure and knowledge in clinical and biomedical representation learning presents both tremendous opportunities and significant challenges.

1.2 My Contributions

My thesis focuses on solving these challenges in designing pre-training and representation learning algorithms that leverage structure and knowledge for the clinical and biomedical domains. In particular, in this thesis I will discuss four specific research endeavors, each of which incorporates structure and knowledge into representation learning in different ways. I describe each chapter below, and additionally include in each description a citation for the underlying work driving the chapter, as well as other related works I have co-authored that are not featured in the chapter but are thematically related. At the end of this section, I also include a list of other publications I have completed during my graduate studies that are not featured in this thesis.

First, in Chapter 2, I explore leveraging distributional knowledge learned from unlabeled data via a Cycle Wasserstein Regression Generative Adversarial Network (CWR-GAN) for clinical and biomedical regression problems. We show that this approach significantly outperforms traditional supervised learning alone in predicting individual treatment response estimation in intensive care patients. This work further motivates larger-scale self or semi-supervised pre-training systems, which similarly take advantage of unlabeled data, and in particular pre-training methods that impose global structural constraints on neural network latent spaces. *Main Work* [134], *Other Related Publications* [11]

Second, in Chapter 3, I show that using the graph structure inherent in genetic co-regulatory information can significantly improve the modelling of gene expression data. This shows that the use of local structure can offer higher quality representations in non-traditionally structured biomedical domains, and further motivates my later analyses of how to incorporate structure into pre-training systems at larger scales. *Main Work* [136], *Other Related Publications* [59]

Third, in Chapter 4, I discuss adaptations of traditional pre-training algorithms to electronic health record data, focusing specifically on structured physiological clinical timeseries. This work highlights the limitations of traditional algorithms for this new modality. In particular, we show that multi-class pre-training algorithms significantly outperform imputation based approaches, highlighting that a naïve adaptation of natural language processing methods does not offer success in this modality. This failure thus motivates the development of new kinds of pre-training methods for biomedical modalities. *Main Work* [129], *Other Related Publications* [7]

Finally, fourth, in Chapter 5, I explore a new theoretical framework for pre-training algorithms highlighting the importance of inter-sample inductive biases in pre-training algorithms. We introduce the framework Structure-inducing Pre-training (SIPT), and provide both theoretical and empirical demonstrations that inducing global structure into the pre-training latent space can offer significant benefits, which opens up significant opportunities for developing new pre-training methods for biomedical data specifically. *Main Work* [132], *Other Related Publications* [164, 131]

Overall, throughout this thesis, we will demonstrate that incorporating structure, either directly learned from data or via external knowledge, can significantly improve performance in clinical and biomedical machine learning.

Other Works I have also explored many other works during my academic career to date which are not related to this thesis, including the following:

Open & Effective ML4H Research [19, 133, 219, 147, 146, 135, 12]

Fairness & Bias [184, 185, 243]

Clinical Natural Language Processing [95, 89, 117, 27]

Conference Organization [8, 128, 41, 6, 179, 175, 55]

Other [38, 24, 127, 79, 10, 227]

Chapter 2

Semi-supervised Biomedical Translation with Cycle Wasserstein Regression GANs

Abstract

In this thesis, I investigate clinical and biomedical representation learning. In this work, originally published in [130], we explore these themes by examining how unlabeled data can provide value via semi-supervised learning for biomedical regression tasks. Note that interested readers may also read [11], which features similar themes but is not included in this thesis.

The biomedical field offers many learning tasks that share unique challenges: large amounts of unlabeled data, and a high cost to generate labels. In this work, we develop a method to address these issues with semi-supervised learning in bidirectional regression tasks. Our model uses adversarial signals to learn from unlabeled datapoints, and imposes a cycle-loss reconstruction error penalty to further regularize the learned regressors. We first evaluate our method on synthetic experiments, demonstrating two primary advantages of the system: 1) distribution matching via the adversarial loss and 2) regularization towards invertible mappings via the cycle loss. We then show a regularization effect and improved performance when paired data is supplemented by additional unpaired data on two real biomedical regression tasks: estimating the physiological effect of medical treatments, and extrapolating gene expression (transcriptomics) signals. Our proposed technique is a promising initial step towards more robust use of adversarial signals in semi-supervised regression, and could be useful for other tasks (*e.g.*, causal inference or modality translation) in the biomedical field.

2.1 Introduction

Motivation

Relative to other fields which have seen recent interest in multi-modal translation (*e.g.*, images to text, or audio to video), the biomedical field lacks large datasets that are "paired"—where two sets of data from the same subject are available (*e.g.*, at different times or in different modalities). Additionally, many biomedical translation tasks involve regressions between source and target domains that are either both representations of some shared, underlying state (*e.g.*, modality translation), or driven by real, bio-physical mechanisms. In either case, we expect both directions of translation to have meaningful, approximately invertible solutions. This makes "cycle" consistency—mapping a particular source example to the target domain and back—desirable. It also means we can leverage the inferred cycle map to re-frame the previously independent regression problems as a joint, multi-task learning problem.

Challenge

Learning from "extra" unpaired data is valuable in settings where acquiring a large amount of paired data is not feasible. In many clinical settings, paired dataset collection (*e.g.*, patient data pre- and post-treatment) is impossible, as doctors have an ethical imperative to intervene on patients at times inconvenient for dataset curation. In gene expression tasks, obtaining expression levels for transcripts corresponding to the entire genome is expensive, but there are large corpora of smaller snippets, independently measured for the purposes of individual studies.

Goal

Our goal is a mechanism of semi-supervised learning for regression problems that leverages large amounts of unpaired data to improve performance on tasks with a scarcity of paired data, and provides an approximately invertible solution from source to target domains. For example, in estimating medical treatment effect, a patient may have data from only pre- or post- treatment rather than both. In gene expression tasks, the inherent intracorrelations within gene expression profiles implies that we should be able to translate between different subsets of the transcriptome—the set of all total gene transcription products in a cell—in an invertible, non-lossy manner [52].

Solution

We design a novel joint regression-adversarial model (CWR-GAN) that uses cycle-consistent generative adversarial networks (GANs) for bidirectional regression tasks. We demonstrate our method on synthetic datasets to illustrate its key points and analyze its effects on two real-world biomedical datasets: individual treatment effect (ITE) regression based on electronic health record (EHR) data, and transcriptomics (gene expression) extrapolation.

Contributions

We develop an end-to-end differentiable architecture that uses adversarial signals for semi-supervised bi-directional translation in the biomedical field. In doing so, we make the following specific contributions:

- We design a cycle-consistent regression adversarial network for semi-supervised regression learning.
- We demonstrate the regularization effect and discriminative performance boost of our method on synthetic data in a semi-supervised setting.
- We evaluate our approach in two diverse real-world biomedical datasets: forecasting the individual treatment effect of four ICU interventions on 29 signals in over 2,000 patients, and extrapolating a 978 dimensional subset of the transcriptome to a 5000 dimensional subset.

2.2 Related works

The biomedical field is not alone in that bi-directional, approximately invertible mappings (*i.e.*, translation tasks) are desirable. For example, stacked autoencoders have been used to learn a shared representation between audio and video signals, and multi-modal conditional prediction frameworks have been used to “hallucinate” one modality given another [149, 194].

Generative adversarial networks (GANs) have previously been used for translation tasks. For instance, GANs were used to translate from captions to their associated images, generating images of birds and flowers from text captions [169]. Previous work in the imaging domain has explored using GANs for translation tasks by combining adversarial losses with traditional regression losses [94], but these systems have since been surpassed by adversarial-only systems, such as one which used a bidirectional cycle-consistent adversarial network (Cycle GAN), to translate images from one style to another [251]. No investigations that we know of have applied any adversarial techniques, with or without regression losses, to biomedical translation tasks. GANs have also been explored for semi-supervised learning, but such uses have examined classification tasks in imaging domains, not regression, as we do here [177, 195].

Much prior work in the clinical setting focuses on single domain learning (text, physiological data, etc.) in order to perform supervised prediction or retrieval tasks. For example, predicting mortality given previously observed clinical notes, predicting common billing codes given a portion of a patient’s record, or predicting interventions based on an inferred physiological latent space [65, 116, 68, 229]. Adversarial models have recently been used on clinical data to generate binary and count summarizations of patient records, and to generate clinical time series [33, 53]. In both cases, GANs were used principally for their generative capabilities, rather than modality translation or semi-supervised learning.

2.3 Methods

In the present study, we develop a novel approach to semi-supervised, bi-directional translation shown in Figure 2-1 using a Cycle Wasserstein Regression GAN (CWR-GAN). The CWR-GAN is constructed from several architectures: GANs in general, Wasserstein GANs, and cycle-consistent GANs.

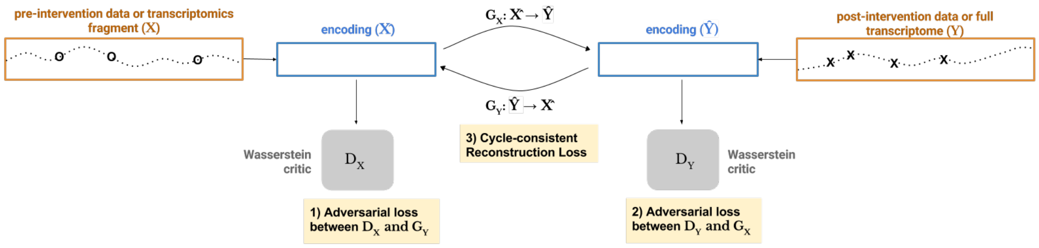

pre-intervention data or transcriptomics fragment (X)
encoding ($\check{X}$)
G_X: $\check{X} \rightarrow \hat{Y}$
encoding ($\hat{Y}$)
post-intervention data or full transcriptome (Y)
G_Y: $\hat{Y} \rightarrow \check{X}$
Wasserstein critic
D_X
3) Cycle-consistent Reconstruction Loss
D_Y
Wasserstein critic
1) Adversarial loss between D_X and G_Y
2) Adversarial loss between D_Y and G_X

Figure 2-1: Overall architecture for the Cycle Wasserstein Regression GAN (CWR-GAN) model.

2.3.1 Generative Adversarial Networks (GANs)

There are two parts to the traditional GAN: a generator $G(\mathbf{z}; \boldsymbol{\theta}_g)$ and a discriminator $D(\mathbf{x}; \boldsymbol{\theta}_d)$ [70]. D and G compete in a two-player minimax game, where D's goal is to discriminate between real and synthetic data, and G's goal is to generate synthetic data that can fool D. In their original formulation, the traditional GAN loss function, at discriminator optimality, measures the Jensen-Shannon divergence between $G(\mathbf{z})$ and p_{data} [69].

Traditionally, GANs are trained in turns: first the discriminator is trained for a number of epochs, then the generator is trained for one epoch, using gradients from the discriminator fixed at its value based on training thus far. This alternating training procedure is repeated until convergence.

Wasserstein GAN (WGAN)

Recent work has proposed use of the Wasserstein (or Earth Mover's/EM) distance to formulate a "critic" in lieu of the traditional GAN discriminator [9]. Compared to traditional discriminators, Wasserstein critics help stabilize GAN training because the EM distance never saturates, and thus provides meaningful gradients to the generator throughout training.

The best known implementation of such a WGAN is via the following loss, which we will use as our adversarial foundation throughout this work [72]:

$$\begin{aligned}\min_G \max_D \mathop{\mathbb{E}}_{\mathbf{x}\sim p_X}\left[D(\boldsymbol{x})\right] - \mathop{\mathbb{E}}_{\mathbf{z}\sim\mathcal{N}(\mathbf{0},\boldsymbol{I})}\left[D(G(\boldsymbol{z}))\right] \\ -\lambda\left(\mathop{\mathbb{E}}_{\bar{\mathbf{x}}\sim p_{\bar{X}}}\left[\|\nabla_{\bar{\boldsymbol{x}}}D(\bar{\boldsymbol{x}})\|_2 - 1\right]\right)^2\end{aligned} \tag{2.1}$$

where $p_{\bar{X}}$ is defined via $\bar{\mathbf{x}} = \epsilon\mathbf{x} + (1-\epsilon)G(\mathbf{z}), \epsilon \sim U([0,1])$. In this loss, λ is a hyperparameter that should be set sufficiently high so as to insist the gradient loss term remains small throughout training.

Cycle-consistent GAN (Cycle GAN)

Cycle GANs learn to translate points between two domains, X and Y, using only unsupervised, adversarial signals. To do this, they learn two "generators" $G_X : X \to Y$ and $G_Y : Y \to X$, and two discriminators (or Wasserstein critics), D_X and D_Y. Both generators are trained not only according to an adversarial loss, but also to minimize a cyclical reconstruction error penalty:

$$\begin{aligned}\mathcal{L}_{\text{Cyc}} = \mathop{\mathbb{E}}_{\mathbf{x}\sim p_X}\left[\|\boldsymbol{x} - G_Y(G_X(\boldsymbol{x}))\|_1\right] \\ + \mathop{\mathbb{E}}_{\mathbf{y}\sim p_Y}\left[\|\boldsymbol{y} - G_X(G_Y(\boldsymbol{y}))\|_1\right].\end{aligned} \tag{2.2}$$

This loss regularizes both learned models towards being each others' inverse, and reframes the two isolated regression tasks as a single multi-task model [251].

2.3.2 Cycle Wasserstein Regression GAN (CWR-GAN)

We present a novel joint regression-adversarial model[1] for biomedical translation problems, where our goal is to learn mapping functions between two encoded domains $\hat{X}$ and $\hat{Y}$ given training samples $\{\hat{x}_i\}_{i=1}^N \in \hat{X}$ and $\{\hat{y}_j\}_{j=1}^M \in \hat{Y}$.

Our model implements a Cycle Wasserstein GAN with the addition of a regression loss on paired samples. Given a source domain X and a target domain Y, with some

[1] Code available at `https://github.com/mmcdermott/CWR-GAN`.

subsets $P \subseteq X \times Y$ consisting of paired observations, our full objective is as follows:

$$\begin{aligned}\mathcal{L}_{\text{CW-Crt}} = &- \mathop{\mathbb{E}}_{\mathbf{x},\mathbf{y}\sim p_{X\times Y}} [C_X(\boldsymbol{x}) + C_Y(\boldsymbol{y})] \\ &+ \mathop{\mathbb{E}}_{\mathbf{x},\mathbf{y}\sim p_{G_X(X)\times G_Y(Y)}} [C_X(\boldsymbol{x}) + C_Y(\boldsymbol{y})] \\ &+ \lambda \mathop{\mathbb{E}}_{\bar{\mathbf{x}}\sim P_{\bar{X}}} [\|\nabla C_X(\bar{\boldsymbol{x}})\|_2 - 1]^2 \\ &+ \lambda \mathop{\mathbb{E}}_{\bar{\mathbf{y}}\sim P_{\bar{Y}}} [\|\nabla C_Y(\bar{\boldsymbol{y}})\|_2 - 1]^2 \qquad (2.3)\end{aligned}$$

$$\begin{aligned}\mathcal{L}_{\text{CW-Gen}} = &-\mathcal{L}_{\text{CW-Crt}} + \nu \mathcal{L}_{\text{Cyc}} \\ &+ \alpha \mathop{\mathbb{E}}_{\mathbf{x},\mathbf{y}\sim p_P} [\|\boldsymbol{x} - G_Y(\boldsymbol{y})\|_2 + \|\boldsymbol{y} - G_X(\boldsymbol{x})\|_2] \qquad (2.4)\end{aligned}$$

Components of this loss offer different training signals:

1. The traditional regression loss term directly trains the generator to perform a low error translation based on the limited paired data available. If all data available is paired, this will be the most direct loss term, and yield the best training signals. This loss term is weighted by hyperparameter α.

2. The cycle loss term regularizes the learned models against those for the opposite direction. This ties the two otherwise independent loss objectives (for G_X and G_Y) together, and is weighted by hyperparameter ν.

3. The adversarial loss term ($-\mathcal{L}_{\text{CW-Crt}}$) helps regularize each model individually by pushing predictions towards regions of high perceived likelihood.

Taken together, these components insist that the two learned maps G_X and G_Y should be approximately invertible, each able to learn well from unpaired data, and able to be refined using paired examples. Traditionally, GANs can suffer from a problem known as *mode collapse*, wherein the generator only generates a very small set of identical examples, each of which is viewed as realistic by the critic. However, our system does not suffer from this problem, as each component of our loss helps penalize this kind of error. Standard regression losses obviously prohibit mode-collapse behavior, as does our cycle consistency penalization, and the Wasserstein formulation of our adversarial

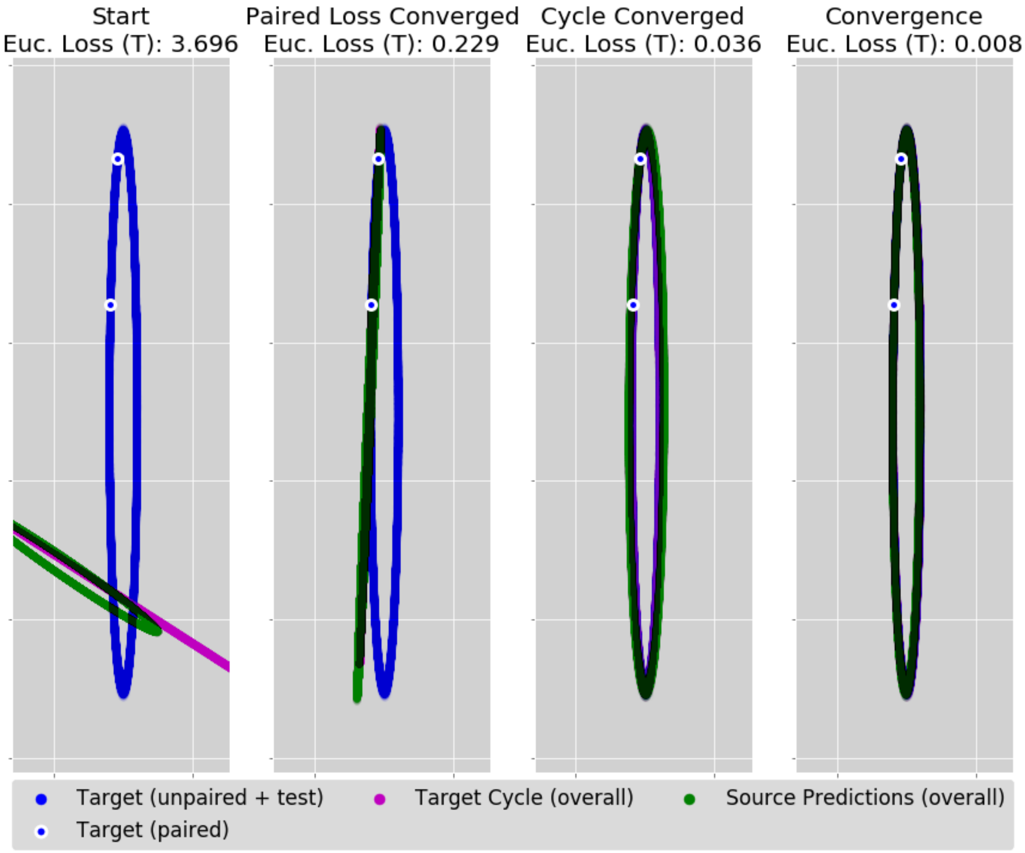


Figure 2-2: The CWR-GAN system on a synthetic domain with only two paired examples. *Far Left:* At initialization, no map is meaningful, and no loss has converged. *Middle Left:* Training first locks both paired points (highlighted in white) to their correct values. *Middle Right:* Training locks both maps into an invertible pair, but has yet to fine tune the output distribution to the exact shape. *Far Right:* The adversarial loss has guided the model to the correct shape. Cycle loss drives the mappings to be invertible before the final distributions are correctly found. After the cycle loss falls to zero, both maps evolve in tandem until convergence.

losses have also been shown independently to suffer much less from mode collapse than a traditional discriminator [9].

2.3.3 Synthetic Experiments

We provide the simplest possible demonstration of the key aspects of our system on synthetic datasets. We generate a source X distributed about the 2D unit circle, and target Y affected by a simple affine transformation, defined as follows:

$$\mathsf{r} \sim \mathcal{N}(1, 0.01) \qquad\qquad \theta \sim U([0, 2\pi])$$

$$X = \begin{bmatrix} \mathsf{r}\cos(\theta) \\ \mathsf{r}\sin(\theta) \end{bmatrix} \qquad\qquad Y = \begin{bmatrix} 0.2 & 0 \\ 0 & 4 \end{bmatrix} X + \begin{bmatrix} 1 \\ 3 \end{bmatrix}$$

We present results on the noiseless domain defined above for simplicity, but note that these results hold with mild, independent Gaussian noise on both X and Y.

We generated 10,000 total samples, with an 80%/20% train/test split, and offered the system either no paired samples or only two paired samples following the train/test split. In this domain, translators are affine transformations, and critics are 3-layer deep, 300-neuron wide leaky rectified linear unit (Leaky ReLu). All networks in this work use a Leaky ReLu activation, with $\alpha = 0.3$ (*e.g.*, $\texttt{LeakyReLu}(\boldsymbol{x}) = \max(0.3\boldsymbol{x}, \boldsymbol{x})$). networks. These depth/width settings were chosen to be similar to the 2D experiments in earlier WGAN works [72]. Regression loss multipliers (α) were set to 1 in this experiment, cycle loss (ν) to 0.2, gradient loss (λ) to 3, and 3 critic epochs were performed for every 1 translator epoch.

We highlight three aspects of the CWR-GAN observed on this synthetic dataset:

- Absent any paired data, the system learns the correct output distributions, and a map consistent with the symmetries of the output distributions, thereby demonstrating the value of adversarial signals in their own right.
- With two paired data points, the system learns not only the correct output distribution, but the correct map, thereby demonstrating that adversarial signals can complement paired examples to learn a map benefiting from both sources of information (Figure 2-2).
- The cycle loss component serves to "snap" the maps together into an invertible pair, thus helping each use the other for regularization.

In this synthetic verification and the two experiments on biomedical datasets that follow, all models were implemented in Tensorflow [1]. We use the Adam optimizer [99], with hyperparameters similar to those recommended in prior work [72] ($\alpha = 0.00005$,

$\beta_1 = 0.5$, $\beta_2 = 0.9$) in the CWR-GAN for critics and generators.

2.4 Experiment 1: Individualized Treatment Effect Prediction with ICU Patient EHRs

In this experiment, we focus on predicting individual patients' responses to interventions (*i.e.*, from pre- to post-treatment). We examine 29 noisy timeseries derived from the electronic health records (EHRs) of intensive care unit (ICU) patients. These signals are recorded hourly; however, it is common for interventions to be applied near the beginning or end of a patient's stay. This limits the availability of paired training examples because few records contain sufficiently large, equally sized windows both pre- and post-intervention. A standard regression system can only use fully paired examples, but our CWR-GAN model can also use those records that only have one such window as an unpaired example of either the source (pre-intervention) domain or the target (post-intervention) domain. This allows us to learn from additional data that is inaccessible to traditional regressors.

Forecasting a patient's response to a treatment—their individualized treatment effect (ITE)— is an important task because the efficacy of clinical interventions can vary drastically among patients. Further, unnecessarily administering an intervention is expensive and potentially harmful. We target two interventions: invasive ventilation and vasopressor use. While ventilation is a commonly used ICU treatment, there are many potential complications and changes in ventilation settings can impact patient outcomes [234, 211]. Similarly, vasopressors are a medication commonly used in the ICU, but have been found to be harmful in certain populations [42].

2.4.1 Data Source & Preprocessing

We use data from the publicly available Multiparameter Intelligent Monitoring in Intensive Care (MIMIC-III v1.4) database [96]. We consider the first ICU stay of patients 15 and older whose stay duration was 12 to 240 hours, yielding 33,287 unique

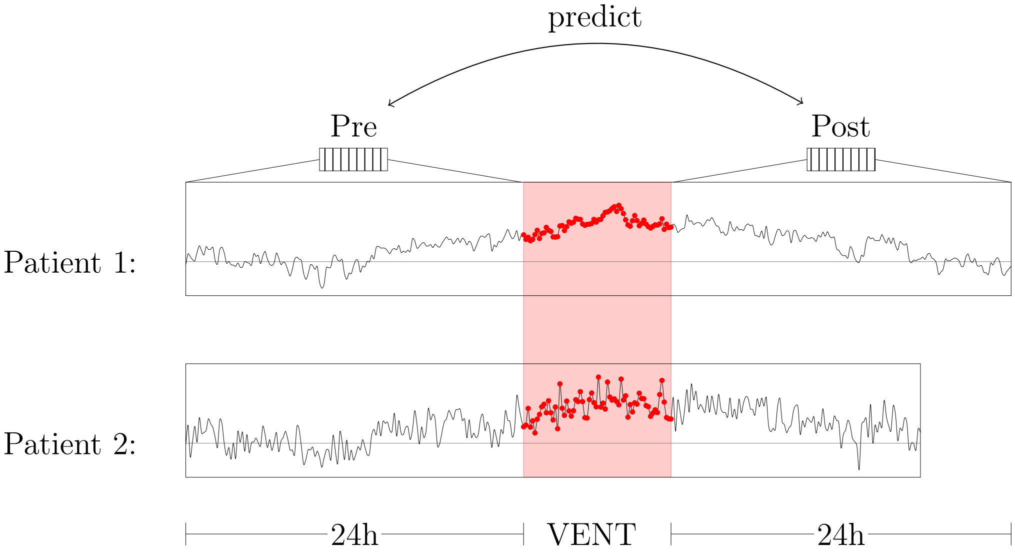


Figure 2-3: ITE regression task setup. Physiological series corresponding to some pre-intervention window (VENT, highlighted in red) are summarized into a fixed-size encoding, then translated to the corresponding post-intervention window. In this example, Patient 1 has sufficient data on both sides the intervention window to form a *paired* training example. However, Patient 2 does have not sufficient time post-intervention, constituting an *unpaired* "pre-" training example from the source domain.

ICU stays. For each patient, we extract the static variables gender and age, as well as 29 time-varying vitals and labs, the same used by other work [209]. Vital and lab measurements are given timestamps rounded to the nearest hour, and multiple measurements within an hour were averaged. Measurements are only recorded for hours in which they are taken, so missing measurements are common.

There have been several proposed encodings for physiological data [28, 68]. Here, we use a fixed sized encoding formed by concatenating measurement counts (*i.e.*, how often a measurement was taken) with the average value observed for each measurement during the time interval. If a measurement was never taken during the interval, we fall back to the patient's average for that measurement; if it was never taken for that patient, we use the population average. Finally, we concatenate the patient's age and gender to this representation to enable our task to use static signals as well as the summarized time series.

	Paired	Pre-	Post-
Ventilation	834	469	7973
Phenylephrine	510	568	3697
Norepinephrine	247	363	1931
Dopamine	159	135	960

Table 2.1: Population sizes for the intervention prediction tasks. Each intervention has three distinct populations: 1) paired patients, 2) "pre-" unpaired records, obtained from patients whose ICU stays did not contain a full 24-hour interval following intervention application, and 3) "post-" unpaired records, obtained from patients whose ICU stays did not contain a full 24-hour interval preceding intervention application.

2.4.2 Experimental Setup

Our primary prediction task is performed over 24-hour windows (Figure 2-3). These yield a range of paired vs. unpaired splits, and our goal is to predict the physiological signals post-treatment (target) given the patient's physiological signals pre-treatment (source). We examine four interventions: invasive ventilation (VENT), and the use of three specific vasopressors—phenylephrine (PHEN), norepinephrine (NOREP), and dopamine (DOP). Final population sizes for each intervention after extraction are illustrated in Table 2.1.

We are primarily interested in the difference between a traditional regression neural network and our semi-supervised system with respect to the natural regression loss for the target domain (*e.g.*, Euclidean distance loss in the post-intervention domain). As such, we train and evaluate two models: 1) a traditional regression multi-layer perceptron (MLP) for either direction of regression, and 2) our semi-supervised system, which augments the traditional network with a Wasserstein critic and the cycle loss penalty.

Models were tuned, then evaluated via nested cross-validation. All regression and critic networks were 3-layer, bidirectional regressors using leaky ReLU activations, dropout of 0.75, and L2 & L1 regularization of 1e$-$3. All hyperparameters were chosen via nested cross-validation search, and results are reported in terms of median Euclidean distance loss on the target domain across the same outer cross-validation split. Loss multipliers were fixed independently of task at a multiplier of 10 for the

	Intervention Type			
Model	VENT	NOREP	DOP	PHEN
MLP	3.780	2.829	2.719	3.186
CWR-GAN	−0.50%	−7.4%	+2.7%	−4.5%

Table 2.2: Comparison of median model performance on four targeted interventions. The traditional MLP regression network performance is reported in Euclidean distance. Our semi-supervised CWR-GAN results report the difference from the MLP's loss, as a percentage of that loss. Thus, a positive percentage is where the CWR-GAN performed *worse* than the MLP (*i.e.*, on the dopamine treatment effect task), and the remaining negative losses on all other ITE tasks are where the CWR-GAN was *better*. All models significantly outperformed a linear baseline.

regression component and 1 for both the adversarial and cycle reconstruction error losses. The gradient loss multiplier was set to 10, but if a critic appeared to suffer from gradient explosion during training, it was increased to 50. Models were trained for up to 9 consecutive critic epochs, stopping after 3 critic epochs that did not improve the adversarial loss, then 1 translator epoch.

2.4.3 Results

Results for the performance of our system are shown in Table 2.2. We see that on three of the four interventions, our joint system yields an improvement over a traditional regression system in terms of the overall loss by fractions ranging from 0.5% to 7.4%. On the dopamine vasopressor prediction task (DOP), we underperform the traditional system by 2.7%. This may be because dopamine has the smallest fraction of $\frac{\text{unpaired}}{\text{paired}}$ data of any of our sources (6.9 for dopamine vs. 8.4, 9.3, and 10.12 for phenylephrine, norepinephrine, and ventilation, respectively), but could also be caused by other, unforeseen complexities. Nevertheless, overall, these results demonstrate that even with as few as 834 paired data points, or as few as ~ 2500 total points, the CWR-GAN can successfully learn from unpaired instances.

We also observed that during the majority of the cross-validation search, the CWR-GAN system would outperform the traditional system across a majority of tasks for a variety of reasonable, though sub-optimal, parameter settings. Upon closer

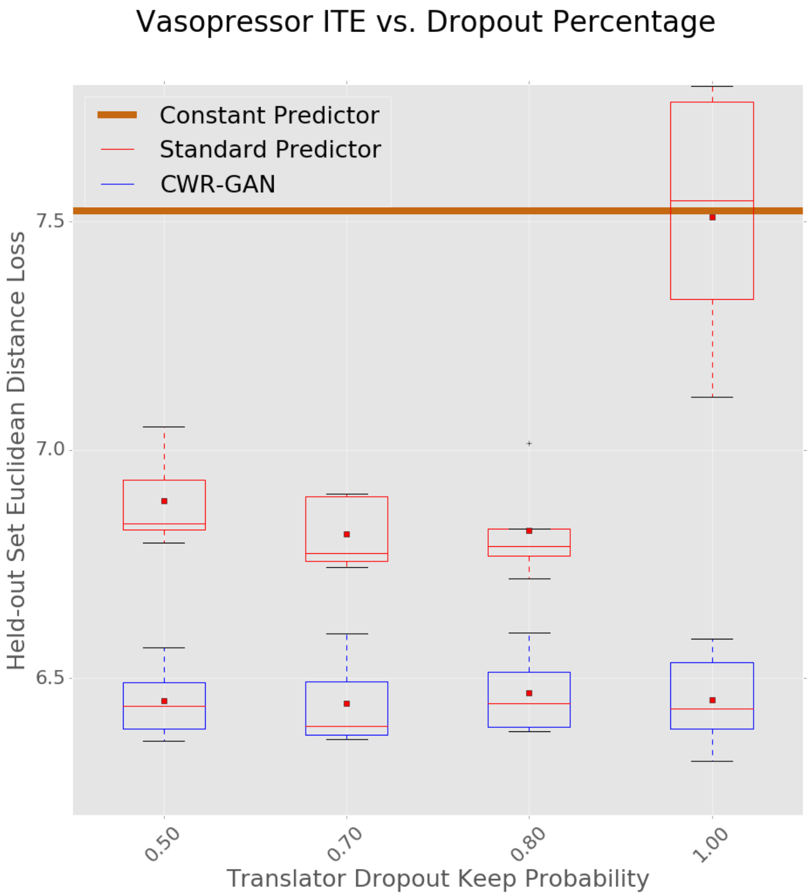


Figure 2-4: Semi-supervised signals offer regularization to ITE regression. The thick, horizontal line is attained by a 'no-change' prediction baseline; *e.g.*, predicting that the intervention does not alter the physiological signals. We demonstrate that appropriate dropout is vital for the MLP (lower is better), but it is not necessary for the improved CWR-GAN results.

inspection, this appeared to be due to additional regularization effects inherent in the adversarial nature of our system's learning, though this warrants additional study. For example, Figure 2-4 shows that dropout is far more influential on the standard predictor than on the CWR-GAN model. This figure is taken from our actual cross validation results, and is thus using sub-optimal parameters; thus, the scale of the losses shown here is not representative of our tuned losses. However, these results do suggest that the adversarial signals and cyclic loss penalty may help to regularize the model in a manner orthogonal to traditional methods of regularization.

2.4.4 Discussion

Analyzing intervention effect is often hampered by the fact that many patients lack sufficient pre-intervention and post-intervention signals to offer a full regression pair. In such cases, the data collected in either their pre- or post- intervention would be ignored by traditional, uni-directional regression approaches. However, our method demonstrates these can still provide valuable signals independently.

Another potential medium for our approach is causal inference, *i.e.*, characterizing how a treatment tested on one cohort would affect a more general population. The ability to use unpaired data that could not be included in standard regression studies in this field would be extremely valuable.

During our experiments, we also considered shorter windows (6 and 12 hours) which contained more paired training data. However, shorter time windows preclude the inclusion of a full diurnal cycle, and it is well-established that circadian rhythm influences most physiological parameters, including metabolic, endocrine and immune functions [207]. This adds a dimension to the learning task which cannot be inferred from the data, as it is not possible to know whether a patient will stop their intervention during the evening or the morning. Additionally, missingness is more prevalent in the 12 and 6 hour periods, as less frequently performed tests are less likely to appear in these shortened windows. As missingness increases, the imputed, average signals will be more common, which induces a non-representative spike in the data distribution. This predominantly penalizes the adversarial system, as it relies on distributional signals. It is thus unsurprising that all model performance decreased on these tasks; the CWR-GAN system in particular failing to ever outperform a traditional model. However, we also note that the CWR-GAN performance consistently suffered more as less data was available as unpaired vs. paired; this finding reinforces the transcriptomics results we discuss next and indicates the model is learning from unpaired samples.

2.5 Experiment 2: Transcriptomics Extrapolation

Transcriptomics data give a view into a cell's internal state by directly measuring the expression levels of genes via their transcriptional byproducts. The full transcriptome is extensive and contains many transcripts, each corresponding to unique proteins with diverse functions. However, it is also redundant, and much can be learned about the cell state from a small subset of the transcriptome. As such, high throughput techniques may measure only a subset of transcripts, thereby saving money and time, and attempt to infer the full gene expression profile [199].

2.5.1 Data Source & Preprocessing

The L1000 technique is commonly used to perform high throughput transcriptomics assays. This technique only directly measures 978 transcripts, and then uses these to infer those remaining [199]. The L1000 developers have released a dataset of 100,000 full transcriptomes, split between the 978 landmark genes and those remaining, to the NCBI GEO database under series number GSE70138[2] [20]. We use this dataset for our task, where the source domain is the 978 landmark genes, and the target domain was restricted to the first 5000 genes for computational efficiency. Data were centered and scale normalized.

2.5.2 Experimental Setup

In the intervention task, our unpaired samples were derived from the same source as the paired examples. Thus, in that context, we could augment our model with those unpaired examples and still fairly evaluate on only a subset of the available paired records. In the context of transcriptomics extrapolation, however, this is not possible; though many scientists have published external transcriptome subsets that we could use as "unpaired" instances in training, we *know* that these are from a different distribution than our main paired dataset because each individual study produces gene expression datasets according to their own individual scientific prerogatives. This

[2] `https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE70138`

means that if we did use these unpaired sources, it would be difficult, if not impossible, to evaluate our model; even if gains in inference occurred generally, they could be negligible on the paired data alone, which is our only testable data source.

Instead, we randomly split our dataset into paired and unpaired datasets according to four variants: $100\%, 50\%, 10\%$, and 3% paired. This allows us to not only effectively evaluate our model in the context of unpaired data, but also to probe how the relationship of our model to a standard regressor varies with the amount of unpaired vs. paired data available.

As before, we are primarily interested in the difference between a traditional regression system and our semi-supervised system in terms of the natural regression loss for the target domain—*i.e.*, the euclidean distance loss on the 5000-dim. transcriptome subset. As such, we train and evaluate a traditional regression network, and our semi-supervised system, and compare both of their regression (*i.e.*, not adversarial) losses in the target domain.

Translator networks were exclusively leaky ReLU networks with varyied hyperparameters and network configurations depending upon the specific variant. On the 100% or 50% paired variants, regression networks had three hidden layers and no regularization. If 90% of the data were paired, regression networks had 2 hidden layers, dropout of 0.6, and L2 regularziation of 1e−3. If 97% of the data were paired, networks had 2 hidden layers, with dropout of 0.5, L2 regularization of 2e−3 and L1 regularization of 2e−5. Hidden layer dimensionality matched input dimensionality in all cases. Critics were 2 hidden layer networks with dropout of 0.9, L2 regularization of 2e−4 and L1 regularization of 1e−6 in all cases. Hyperparameters were chosen according to a grid search with a randomly sampled 15% validation set. Models were tested on a held out, randomly sampled 20% test set.

2.5.3 Results

On this dataset, if all data are paired, our system underperforms a traditional neural network, attaining a loss 3.6% higher. This is expected, as the regression loss is the most direct training objective in the fully paired case. However, as we increase the

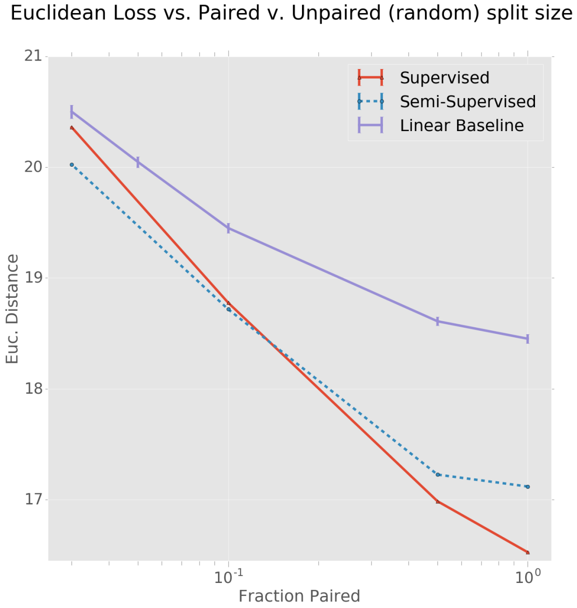


Figure 2-5: Performance (in Euclidean distance regression loss) of a linear model, standard predictor, and our semi-supervised CWR-GAN as a function of the fraction of the data that is paired. Lower is better. As we reduce the amount of paired data available, the semi-supervised model gains ground in performance over both baselines, demonstrating that it learns from the unpaired data sources.

fraction of data that are unpaired, our system gains more and more ground: the CWR-GAN system yielding loss deltas of $1.4\%, -0.3\%, -1.6\%$ at 50%, 90%, and 97% unpaired, respectively. Thus, we see that the amount of unpaired data is perfectly correlated with our model's performance gain over a traditional system, which offers strong support for the argument that the CWR-GAN is learning additional signals from the unpaired samples. Both models significantly outperform a ridge regression ($\lambda = 100$) baseline. The total results in raw units are shown in figure 2-5.

2.5.4 Discussion

Transcriptomics is a promising and rapidly evolving area of computational and clinical biology. It holds tremendous promise for disease subtyping, drug discovery, and diagnostics. However, the expense of collecting a full transcriptome sample and the inherent inter-experimental variability in these data mean there are often not enough data to form generalizable conclusions. The ability to include unpaired or unlabeled instances would aid in circumventing this boundary.

Our results demonstrate that the addition of unpaired data does allow the model to learn additional features from the data, which bodes well for the possibility of using these ideas for real-world use cases, such as augmenting a broader inference algorithm with unpaired data from more recent samples that reflect the full diversity of transcriptomics analyses, as well as for modality translation within multi-omics studies. Further, these results underscore those attained in the intervention task by reiterating that the model can successfully learn from the unpaired data, now on a dataset of total size 100,000, rather than approximately 8,000.

2.6 Conclusion

The ability to incorporate large amounts of unpaired data in a regressive translation is critical in many tasks in the biomedical field. Because these tasks are bio-physically driven, we also desire that any mappings we learn be invertible. These constraints are also applicable in other fields, *e.g.*, mapping pre- and post- trip behaviors from ride-sharing data, or translating social media data pre- and post- an important event. In these settings, the tasks would similarly benefit from the ability to integrate a large amount of unpaired data.

The goal of this work was to create a model that used unpaired data to learn invertible translations more accurately with fewer paired datapoints. We demonstrated our method's applicability to the biomedical field using two different experiments. First, the CWR-GAN was able to use unpaired data pre- and post- treatment to improve ICU patient ITE forecasts in three of four ICU interventions in real-world, noisy physiological signals across more than 2,000 patients. Second, the CWR-GAN was able to successfully extrapolate a 978 dimensional transcriptome fragment to a much larger 5,000 dimensional subset of the transcriptome, and this ability improved relative to traditional methods as the ratio of unpaired to paired data increased.

This approach has two main limitations. First, this method takes much longer than standard predictors. Translator mappings do not typically require more epochs to reach convergence, but because the critic must also be trained, and as more data can be used, they take much longer in terms of wall-clock time. Second, adversarial networks remain a very active area of research, and they are notoriously difficult to train and understand. Their use here, while offering many advantages, also means that this method is inherently more high-maintenance than other strategies. We also note that recent work has shown the EM distance metric (which is central to the Wasserstein GAN critic loss) yields *biased* sample gradients, and thus is perhaps not well suited to stochastic gradient descent [14].

There are many directions for future work building on these methods. Further improvements to training stability, including automatically tuning some of the loss multipliers as training evolves, or with newer variants of GANs, such as the Cramér GAN, would make the training process much smoother. Additionally, further investigations into the contribution of each loss component alone, *e.g.*, cycle loss, could help determine what precise effect they offer. We have previously tried simply adding a cycle loss component to a bi-directional traditional regressor, but this alone did not offer any significant performance improvement. Finally, additional work could be performed attempting to move beyond simple mode-matching via adversarial network theory, and instead perform full distribution matching between the source and target domains.

2.6.1 Relation to this Thesis

In this work, we augment a traditional regression learning setup with additional loss terms designed to push the latent space of these models to conform with structural constraints that are learned from larger, unlabeled datasets. In this way, we can see that this work shows a method for imposing global structural constraints on neural network latent spaces during representation learning. This is a theme that will be revisited throughout the rest of this thesis, and formalized fully in Chapter 5, which demonstrates a framework for inducing global structure during pre-training.

Unique from the rest of the thesis is that in this work the structural constraints imposed are *not* drawn from external domain knowledge, but instead are learned alongside the neural network models from unlabeled data directly. As this work features regression problems, this approach can still provide powerful inductive biases, but as we move into other domains, where we can't so precisely describe the desired geometry of neural network latent spaces, the use of external knowledge to define these structural constraints will become more important, and accordingly external knowledge will feature more heavily in the later chapters of this thesis.

Acknowledgments

This research was funded in part by the Intel Science and Technology Center for Big Data, a Philips-MIT research agreement, the Wistron Corporation, and grants from the National Institutes of Health (NIH): National Library of Medicine (NLM) Biomedical Informatics Research Training grant 2T15 LM007092-22, National Institute of Biomedical Imaging and Bioengineering (NIBIB) grant R01-EB017205, National Institute of Mental Health (NIMH) grant P50-MH106933, National Human Genome Research Institute (NHGRI) grant U54-HG007963.

Chapter 3

Leveraging Structure for Transcriptomic Data Analysis

Abstract

In this thesis, I explore how we can leverage structure and knowledge to improve biomedical pre-training and representation learning algorithms. Central to that goal is the idea that incorporating prior knowledge can help improve the quality of even high-capacity models. In this work, originally published in [136] and featured in my S.M. thesis, we show for the first time (as of its publication) that incorporating prior knowledge of genetic co-regulatory networks can improve representation learning systems over transcriptomics data. Note that for further exploration of the themes of this chapter, interested readers should also examine [59].

In this chapter, we compare the efficacy of traditional classifiers against graph convolutional neural networks (GCNNs), which we augment with prior domain knowledge via co-regulatory networks learned from the scientific literature. Furthermore, we introduce three new classification tasks on multiple L1000 gene expression datasets. In addition, on a private, smaller dataset, we profile per-subject generalizability to provide a novel assessment of performance that is lost in many typical analyses. To thoroughly benchmark the efficacy of this injection of prior knowledge, we compare traditional classifiers, including feed-forward artificial neural networks (FF-ANNs), linear methods, random forests, decision trees, and k-nearest neighbor classifiers against our GCNNs. We find GCNNs offer performance improvements given sufficient data, excelling at all tasks on our largest dataset. On smaller datasets, FF-ANNs offer greatest performance. Linear models significantly under-perform on all dataset scales, but offer the best per-subject generalizability. Ultimately, these results suggest that the incorporation of prior knowledge can help even in high-capacity, large-data regimes, and further motivates future work in the use of structure and domain knowledge in representation learning algorithms.

3.1 Introduction

Transcriptomics data are an increasingly important data modality within biomedicine, and such data are widely used for connectivity mapping, drug development, and other biological research [199]. Appropriately, given its high importance, biologists have accrued significant domain knowledge regarding how gene expression is regulated biologically, providing extensive, if complicated and uncertain, structure around these data. The availability of large-scale, heterogeneous gene expression datasets is also rapidly on the rise, fueled both by falling costs and development of new gene expression profiling technologies [199].

Simultaneous with the increasing availability of gene expression data, deep learning techniques have grown vastly more powerful and popular—showing advances in image processing [73, 54, 98], natural language processing [106, 63, 190], and speech recognition/generation [150, 109], among other fields. In some limited areas, these advances have also translated into the biomedical domain—for example, in analyzing mass spectrometry spectra [213], DNA sequences [248], amino acid sequences [113, 218, 25], or biomedical images [54, 73].

However, among non-sequential, non-imaging modalities, such as gene expression data, "deep" learning methods generally remain limited to simple, unstructured, shallow modeling techniques. In particular, while large-scale benchmarks such as the ImageNet challenge[1] and the existence of clear underlying mathematical structure to constrain network architectures have fueled the development of convolutional neural networks (CNNs) for image processing or recurrent neural network (RNNs) for sequential analysis, bioinformaticians are limited to unstructured feed-forward artificial neural networks (FF-ANNs), which are known to be relatively inefficient learners [102].

In this work, we aim to lay a foundation that will help deep learning succeed for gene expression data as it has in these other domains by providing a fixed definition of success via benchmarks and offering a potential avenue for using structure to

[1] ImageNet is a dataset containing millions of labeled images; its associated challenge tasks computer vision researchers to design algorithms to identify the objects in these images among a fixed set of categories. Many see ImageNet as a critical seed to the current deep learning boom [64, 176].

create more intelligent modeling approaches. In particular, we define three biologically motivated benchmarking tasks over two curated views[2] of the public L1000 LINCS dataset and one privately produced gene expression dataset. On each task, we profile k-nearest neighbor (k-NN) classifiers, decision trees, random forests (RFs), linear classifiers, and two neural network classifiers: feed-forward artificial neural networks (FF-ANNs) and graph convolutional neural networks (GCNNs). GCNNs generalize the notion of convolutional neural networks (CNNs) onto data structured over arbitrary graphs and allow us to use prior biological knowledge, namely regulatory relationships between pairs of genes, to more intelligently model these data.

We find that GCNNs can perform very well, but require large amounts of data; in particular, they offer the strongest performance on all tasks over our largest dataset, but under-perform FF-ANNs on our smaller datasets. Of other methods, FF-ANNs perform best, followed consistently by linear classifiers, then random forests, then decision trees. k-NN classifiers show less consistent relationships, as they perform almost as well as FF-ANNs on our larger datasets, but they underwhelm on our smaller datasets.

Gene expression datasets often contain many samples spanning a very small set of subjects, as a single subject's gene expression profile may be taken many times under varying conditions (*e.g.*, different drugs, etc.). As such, a pronounced risk when modelling gene expression data is that one can learn a model specific to the very limited population expressed in your data. With datasets such as the LINCS data, which often only have the equivalent of one or two subjects (*i.e.*, cell lines) per tissue type, assessing the extent of this overfitting can be difficult, and many works merely report per-sample performance metrics (allowing the model to train and test on the same set of subjects). In this work, we use our private, smaller corpus to assess per-subject generalizability by training on a restricted set of subjects and testing on a held-out subject. We find that all methods struggle to generalize to unseen subjects, showing performance drops ranging from 10 to 18 percent of their per-sample accuracies.

[2] See `https://github.com/mmcdermott/LINCS_Deep_Learning_Benchmarks`

In sum, in this work we make the following contributions:

1. We establish biologically meaningful classification benchmarks at deep learning scale on the largest publicly available gene expression dataset.

2. We profile a number of classifiers on these tasks, including non-neural methods and two variants of neural networks, one of which incorporates prior biological knowledge through the form of genetic co-regulatory networks. We show that incorporating this form of local structure can offer significant benefits for large datasets.

3. We profile these same classifiers on a similar task on a smaller, privately produced gene expression corpus to assess which techniques work well in data-starved environments.

4. We assess how well these techniques transfer to unseen subjects to assess population-level generalizability.

3.1.1 Gene Expression Data

The Biology

The cellular system is governed at the root by the genome: the sequence of DNA base pairs that encode all information necessary for the cell's development and day to day functioning. In order to transmute DNA into useful cellular work, the cell first *transcribes* genes into messenger RNA (mRNA), which is then shuttled towards cellular organelles that *translate* mRNA sequences into proteins: amino-acid built macromolecules that carry out all of the necessary functions of the cell. In this way, we can think of the genome as providing a function library for the cell system, and the proteins present (*i.e.*, the expressed genes) as the actual mechanisms behind cellular functioning. A cell's gene expression profile thus captures a view into the dynamic state of the cell and offers insight far beyond the fixed picture of the DNA alone.

A single cell's gene expression patterns will vary over time and in response to environmental conditions, such as exposure to drugs. The expression of proteins coded

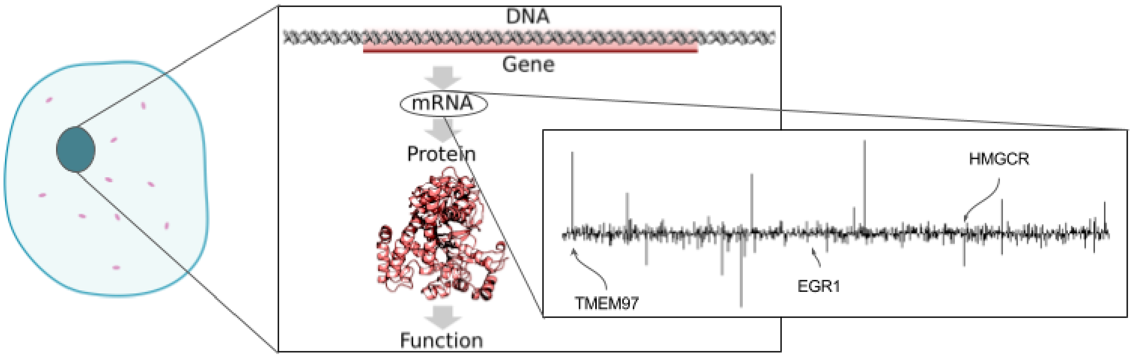


Figure 3-1: Transcriptomics data is measured by quantifying the mRNA produced during transcription. The output of this process is a vector with each dimension quantifying the expression of a particular gene. Both technical (*e.g.*, misplaced reads) and biological (*e.g.*, tissue type) factors add variance to these data. Images: [50, 186].

by DNA is mediated by a host of factors, including other proteins in the cellular environment and external factors, and is critical to cell function. Understanding the genetic regulatory network (*i.e.*, which factors govern what transcription and how) is a topic of intense study.

Measuring Gene Expression/Transcriptomics

Gene expression can be quantified in many ways. Two broad categories of gene expression data are *proteomics*, which directly measures the quantities of produced proteins within the cell, and *transcriptomics*, which measures the quantities of produced mRNA transcripts within the cell (Figure 3-1). Transcriptomic gene expression is far more easily measured and we will focus on this modality in this work.

Note that there is not a direct correspondence between these two measurement techniques. Protein production is heavily regulated post-transcription, and in using transcriptomic data, we ignore these additional layers of biological processing in favor of the increased availability of data.

Measurement Techniques

Transcriptomics data itself can be measured by many techniques, including RNA-Seq, single-cell RNA-Seq (sc RNA-Seq), and the L1000 platform, which we focus on here. The L1000 platform [199] is notably cheaper per-sample than other transcriptomics techniques, which has enabled the creation of large scale public datasets, such as the LINCS dataset, which was produced with the L1000 platform and contains approximately 1.3M samples, available on GEO at accession number GSE92742.[3]

However, this low price point sacrifices some data quality and coverage. Rather than quantifying the full transcriptome, the L1000 platform only directly measures the expression levels of 978 “landmark genes” and requires several additional layers of processing which add their own sources of technical variability. From this directly measured subset, the L1000 technique also uses a linear model to impute the remaining genes’ expression levels, but we ignore those inferred genes in our analyses and use only the landmark genes.

L1000 data is often used at one of two levels of pre-processing:

Level 4 (a.k.a. Roast) Level 4 data is fully normalized and z-scored, and presented at the level of one profile per sample. From a machine learning perspective, this is what you would expect to work with when thinking of “raw L1000 output.”

Level 5 (a.k.a. Brew) Level 5 data takes the Level 4 data and aggregates samples under identical technical conditions into a single averaged view of that profile (see [199] for full details). This process reduces variance, but also dataset size. Typically datasets are reduced to roughly $\frac{1}{3}$ of their original size (L1000 experiments are often performed “in triplicate,” with three identical experimental plates being prepared so that all samples are run under identical conditions at least three times). This variance reduction is useful for traditional bioinformatics, but it is not clear how helpful it should be for machine learning. We would like our classifiers to be able to fully account for the technical variability inherent between repeated measurements, but using Level 5 data would deprive us of that opportunity while costing a significant number of input samples. On the other hand, Level 5 data may be of higher quality.

See Figure 3-2 for a graphical representation of a subset of the L1000 technical pre-processing pipeline.

[3] `https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE92742`

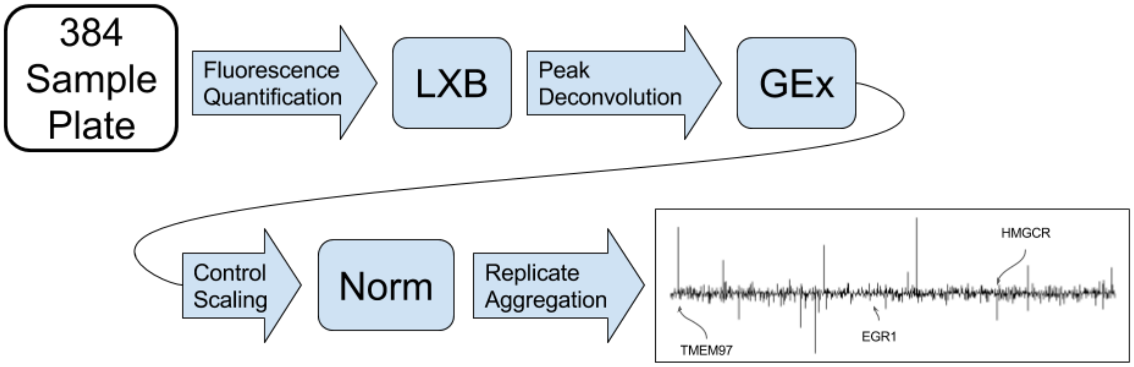


Figure 3-2: The L1000 technique is cheaper than other measurement technologies, but requires novel additional technical pre-processing. Each step in this flow induces more technical variability. In this work, the normalized, pre-aggregation data is referred to as "Level 4" data, whereas the post-aggregation data is referred to as "Level 5" data. This figure omits imputation of the full transcriptome as we never use imputed genes in this work. See [199] for full details.

Experimental Pipelines

In general, experimental pipelines producing large corpora of gene expression data work by acquiring some base cellular sample in pluripotent form, either patient derived or via a stock cell line, cloning that cell line extensively, then perturbing a number of samples and profiling them. In this way, these datasets often have many more samples than cellular sources. This can lead to population-specific over-fitting, where a model specializes only to the population within the corpus and, despite generalizing to unseen samples within the corpus, the model will fail to generalize to unseen cellular sources.

3.1.2 Machine Learning on Gene Expression Data

Traditional Analyses

Traditional analyses on these data focus on statistical or geometric tests for differential gene expression [36], gene set enrichment analyses (GSEA) [200], and (for the L1000 platform specifically) signature based analyses [5, 199]. Some have also used tensor decomposition/completion to disentangle cell-type from perturbagen effects [85, 86], and explored traditional classifiers for adverse drug event prediction [224].

Neural Representation Learning

Other authors have used neural network models to build embeddings of gene expression data. In [58], the authors use a twin network architecture to represent gene expression profiles as 100 dimensional bar-codes. Comparing their representations to the standard representation (raw z-scores) and a representation based on gene set enrichment analysis (GSEA) they find that their perturbation barcodes consistently identify replicates, samples generated from perturbagens with shared targets, and show better clustering overlap with structural properties. Their architecture is a two layer feed-forward neural network, and notably uses inter-replicate variability (a sign of the noise within the technique) as a mechanism to help train their network to learn embeddings that natively differentiate between meaningful biological variation and confounding noise. However, their approach is aimed towards unsupervised representation learning, and thus only can provide soft benchmarking utility, as compared to a supervised task (though this may be more useful in some practical settings).

In [30], the authors use a 4 layer deep sparse autoencoder to analyze binarized yeast differential gene expression microarray data (*e.g.*, their model took as input one feature per gene, with value 1 if that gene was differentially expressed in that sample, and value 0 if not). Their 4 layer architecture is designed to map to known biological levels of processing, and post-training analyses suggested good overlap between transcription-factor mediated regulatory relationships and the connections trained by their network between the first input layer and the first hidden layer.

In [115], the authors explore neural network mediated dimensionality reduction for single cell RNA-Seq data, augmenting traditional networks by adding nodes to the first hidden layer according to known transcription factor or protein-protein interactions, and only connecting input gene nodes to those regulatory or interaction nodes as dictated by prior biological knowledge. These augmentations allowed them to significantly reduce the parameter space relative to a fully connected, dense network of equal width.

Neural Classification & Regression

In [3], authors use a FF-ANN to classify profiles into categories based on the therapeutic effect of the generating perturbagen. Researchers have also explored neural techniques for extrapolating the L1000 set of landmark genes to the full transcriptome. In [31] and [134], the authors explore gene expression extrapolation, albeit in the latter work as a test case for a semi-supervised model architecture rather than a goal in itself. In both cases, FF-ANN models are able to realize significant performance gains over linear models, again demonstrating that nonlinear relationships play a significant role on this modality.

3.1.3 Structured Models via Graph Convolutional Networks

Regulatory Graphs

As stated in Section 3.1.1, gene expression is regulated by complex processes and is a topic of intense study. What we do know of gene expression regulation is often envisioned as a graph, with genes forming the vertices of this graph and edges between genes representing regulatory relationships between those two genes. Regulatory graphs are often represented as directed graphs, with direction representing the direction of the regulatory interaction, but in this work we will simplify them to undirected graphs, having them instead simply flag a regulatory relationship in either direction between two genes. We visualize one such regulatory graph in Figure 3-3.

Many of these relationships are only suspected, and as biologists have yet to study all possible interactions between sets of genes, these graphs are biased towards representing commonly studied proteins. Additionally, regulatory relationships themselves depend on cell type and, even within a single cell, they are dynamic, changing in response to perturbations and environmental conditions, among other factors. Nonetheless, these “regulatory graphs” present at least a partial encoding of the biological understanding of relationships between different genes, and we use them here to augment neural classifiers with domain knowledge via GCNNs.

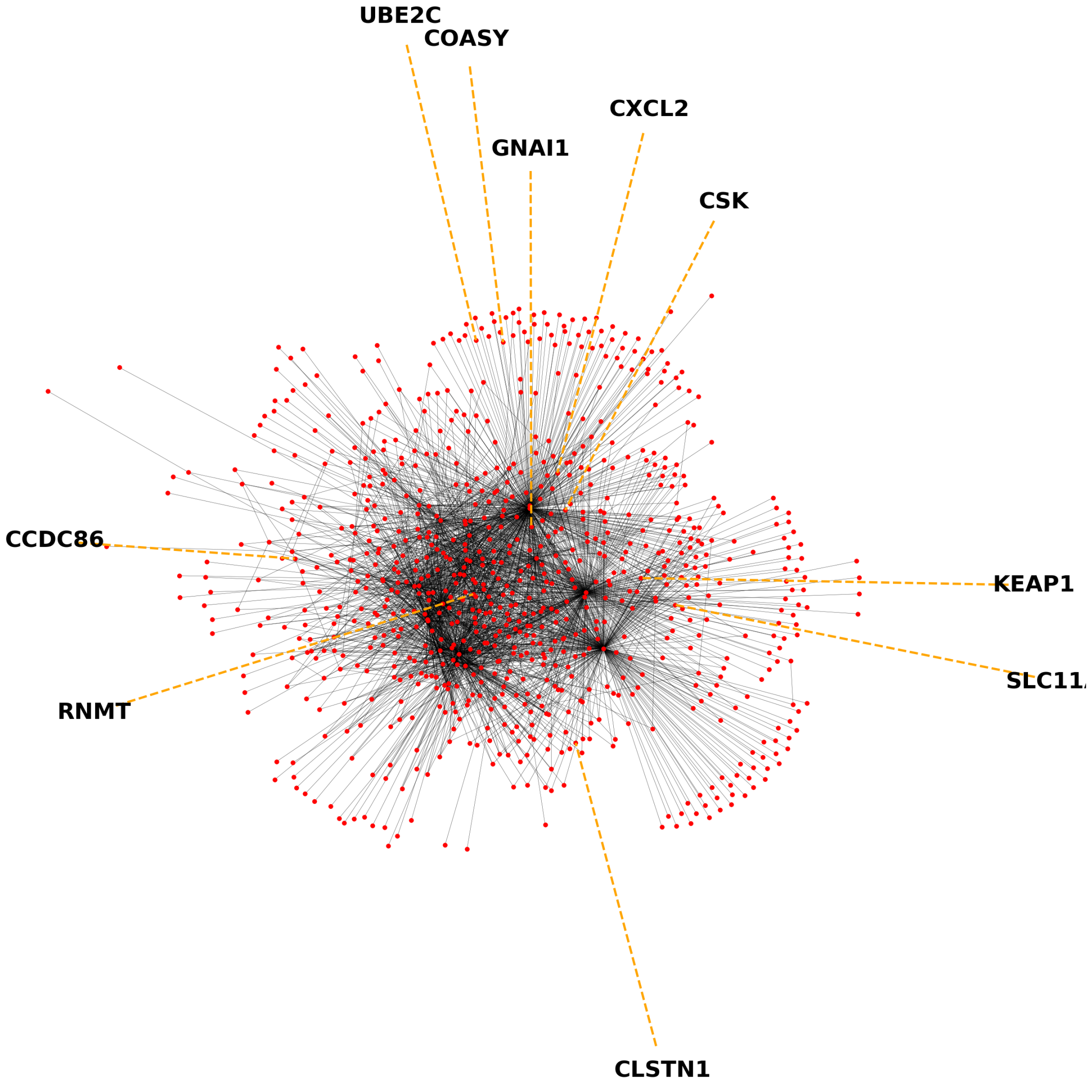


Figure 3-3: The regulatory relationships between L1000 landmark genes, as determined according to [122]. Nodes (red dots) are genes and edges between them represent known or suspected regulatory interactions. Note that many genes only have one known edge connecting them to much denser clusters within the center of the graph. This may reflect biological processes, or that some proteins are studied much more than others.

Graph Convolutional Networks in Theory

GCNNs are extensions of CNNs onto data defined over arbitrary graphs. Qualitatively, we can think of these networks as attempting to analyze data defined over a graph by repeatedly featurizing the data over local neighborhoods within the graph, before aggregating those features into higher level signals spanning larger regions of the graph. This is directly analogous to how convolutional neural networks for image processing learn featurizations of local patches of the image, then pool those signals over larger windows.

There are two main strategies to generalize a CNN to other domains: the spectral approach, which generalizes the notion of a Fourier transform onto a graph via the graph Laplacian, and the locality approach, which uses the idea of processing data defined in local patches via neighborhoods in the graph. GCNNs must also generalize the notion of "pooling" onto graphs, which they generally do via graph clustering algorithms, using the resulting node clusters to determine pooling neighborhoods.

GCNNs promise to bring the normalization obtained via weight sharing over consecutive convolution and pooling operations to features defined over any arbitrary graph, but they present their own challenges. Both local and spectral methods present computational challenges, and efficient graph pooling algorithms run afoul of NP-hard graph clustering algorithms. In practice, many operations are approximated, which affects the power of these models.

Graph Convolutional Networks in Practice

Graph convolutional networks are often used in forming predictions at the node level, or in classifying whole graphs. For example, [100] explored node classification on knowledge and citation graphs. In this vein, GCNNs have also been used in several biological tasks. For example, [77] classifies proteins viewed as nodes in varying tissue-specific protein protein interaction graphs, [51] learns representations of molecular compounds interpreted as unique graphs with vertices determined by atoms and edges by bonds, and [61] learns representations of graphs defined by protein amino acid sequences for protein interface prediction.

Note that in these node classification tasks the authors are making predictions about nodes in a graph, rather than making predictions about a graph (such as a gene expression profile realized as a graph via a regulatory network, *e.g.*, Figure 3-3). In the latter context, spectral methods are enticing; in fact, this picture is so appealing that many papers describing novel GCNN algorithms use this example to frame the impact of their ideas [43, 110, 21, 83]. However, to the best of our knowledge, no work yet has profiled how these ideas actually serve on gene expression data in practice. We fill that lack here.

3.2 Methodology

Our principal goal here is to establish biologically meaningful benchmarking tasks for gene expression data, and to demonstrate the potential utility of structured methods that incorporate prior knowledge, across a variety of dataset sizes and levels of heterogeneity. To that end, in this section we will first profile the two datasets we will use and detail the benchmarking tasks we defined on each. Next, we will walk through the methods we test, paying close attention to the structured method we profile here, and finally we will detail our technical setup and experimental parameters.

3.2.1 Datasets

We will now detail the creation of both our views of the public LINCS corpus and the private MGH NeuroBank corpus. Full details and summary statistics for both corpora can also be found in Table 3.1.

Curated Views of the Public LINCS Corpus

The full Level 4 LINCS dataset contains approximately 1.3 M gene expression profiles over 76 cell lines, ranging in frequency from VCAP, profiled over 200,000 times to NCIH716 with only 43 samples. Each cell line is profiled in diverse conditions—for example, within prostate tissue (the most frequently sampled tissue type) over 40,000 unique perturbagens were tested (including both drugs and genetic knockout or over-expression perturbagens), many sampled only a single time. To be clear, each sample in this dataset is a complete gene expression profile over the landmark genes—*i.e.*, it is a 978 dimensional vector where each number quantifies the expression level of a particular gene in the genome.

On this dataset, we formed three supervised learning tasks:

Primary Site Predicting primary site (*e.g.*, "breast tissue" or "large-intestine") forces the classifier to examine deviations within a gene expression profile indicative of the tissue type, and would have applications to quality control within cell differentiation pipelines. Primary site is cell-line specific.

Subtype Subtype (*e.g.*, "malignant melanoma" or "myoblast") is also cell-line specific and speaks to disease state and provides another way of aggregating the many disparate cell lines within LINCS into useful predictive categories.

MOA Predicting drug mechanism of action (MOA, *e.g.*, "ATPase inhibitor" or "Sodium channel blocker") speaks to drug re-purposing and discovery applications and aggregates many disparate perturbagens into meaningful predictive categories. However, note that though we treat this as a standard multi-class classification problem, in reality many drugs have multiple known MOAs, a distinction we ignore here for simplicity. To ensure this simplifying assumption adds minimal noise to our classification task, we only include compounds with only a single known MOA.

Dataset Curation Procedure

We chose to reduce the LINCS dataset to a single curated view simultaneously suitable for all three of these tasks rather than forming a separate view per task. This causes us to lose some samples which only meet inclusion criteria for a subset of our tasks, but it is much more convenient to work with and disseminate. In that pursuit, we reduced the dataset to only those samples perturbed by compounds (not genetic knock-out or over-expression perturbations), and further only those samples perturbed by compounds with a single known MOA. We further restricted the dataset to only those samples corresponding to MOAs, primary sites, and subtypes that occurred more than 1000 times within the overall dataset, to ensure sufficient training examples for all classes for our classifiers. We performed these filtering steps independently—*i.e.*, we removed all gene expression profiles belonging to a class in any of our three tasks that lacked 1000 full examples at the start. This resulted in some few classes in some of our tasks having fewer than 1000 examples (because, at the beginning of the process, they had over 1000 measurements, but after removing some samples due to their class membership for another task, the class then had fewer than 1000 measurements).

This formed one curated view of our data, and three classification tasks. One qualm some might have with this dataset is that it is very heterogeneous in terms of cell type—perhaps it is better to classify samples only derived from a single tissue type. To that end, we also formed a dataset containing only samples from prostate tissue (chosen as it was the most frequently sampled tissue type). As in our full dataset, here we restrict the samples to only those perturbed by compounds with a single known MOA that occurred at least 1000 times. This formed our "Prostate Only" dataset, on which we predict MOA only.

Full final dataset sizes, heterogeneity (among cell type) statistics and task statistics (*e.g.*, class imbalance, number of classes) are shown in Table 3.1. Note that there is significant class imbalance in this dataset—an unavoidable reflection of the corpus's original makeup—but by filtering to a baseline number of examples per class we assert that there are at least a significant number of samples for every label, ensuring learning power. We have made both of these datasets (though derived from fully public data), along with the cross-validation folds used in all of our experiments, publicly available,[4] so that others can most easily compare novel methodologies against our benchmarks.

We do not claim that these benchmark tasks or views of the data are the best benchmarks available. But these *are* biologically meaningful benchmarks on an important data modality that currently has *none*. We hope that as future methods evolve to better suit this methodology, we can also derive better benchmark tasks.

[4] See `https://github.com/mmcdermott/LINCS_Deep_Learning_Benchmarks`

Note here that we do not mean to claim that no machine learning tasks have been used on this modality previously, but rather that no set of systematized, very large sample size tasks for methodology development currently exist.

Given the very large ratio of samples to cellular sources (*e.g.*, 156k to 36) and the very large skew in perturbagen frequency (*e.g.*, DMSO accounting for approximately 1/6th of all data), as well as the lack of independence between perturbagen and cell type, we measure all accuracies on these datasets as *per-sample* accuracy, not *per-subject*, *per-drug*, or even *per-experimental condition* (as different experimental conditions are repeated to varying degrees). This means that our results on these data should not be interpreted to speak to true generalization outside the LINCS covariate space, but rather should be viewed only in their capacity to enable rigorous methodological comparisons.

A flowchart of our dataset curation process can also be seen in Figure 3-4. Furthermore, a visual description of our benchmark dataset can be found in Figure 3-5.

MGH NeuroBank Corpus

Our private corpus of L1000 data was measured on a collection of subject-derived neural progenitor cells, which were perturbed with one of 60 different small-molecule bioactives at varying doses. Some of these compounds are known to have consistent gene-expression signatures (*e.g.*, HDAC inhibitors), whereas others have known clinical utility but a less well understood transcriptomic profile (*e.g.*, clozapine), and still others were unknown on all counts.

These cells come from a population of five individuals, two healthy control subjects, one with Bipolar Disorder, and two with Schizophrenia (all diagnostic labels are DSM-IV diagnoses confirmed by structured clinical interview). All individuals' cells were treated with the same compounds. On this data, we predict perturbagen identity. Note that each perturbagen was profiled at one of several doses, which we ignore here. We also use this dataset to profile how well classifiers do on Level 4 vs. Level 5 data and make a first attempt at assessing per-subject generalizability, by training a model on only four of the five subjects, then testing on the data for the fifth subject.

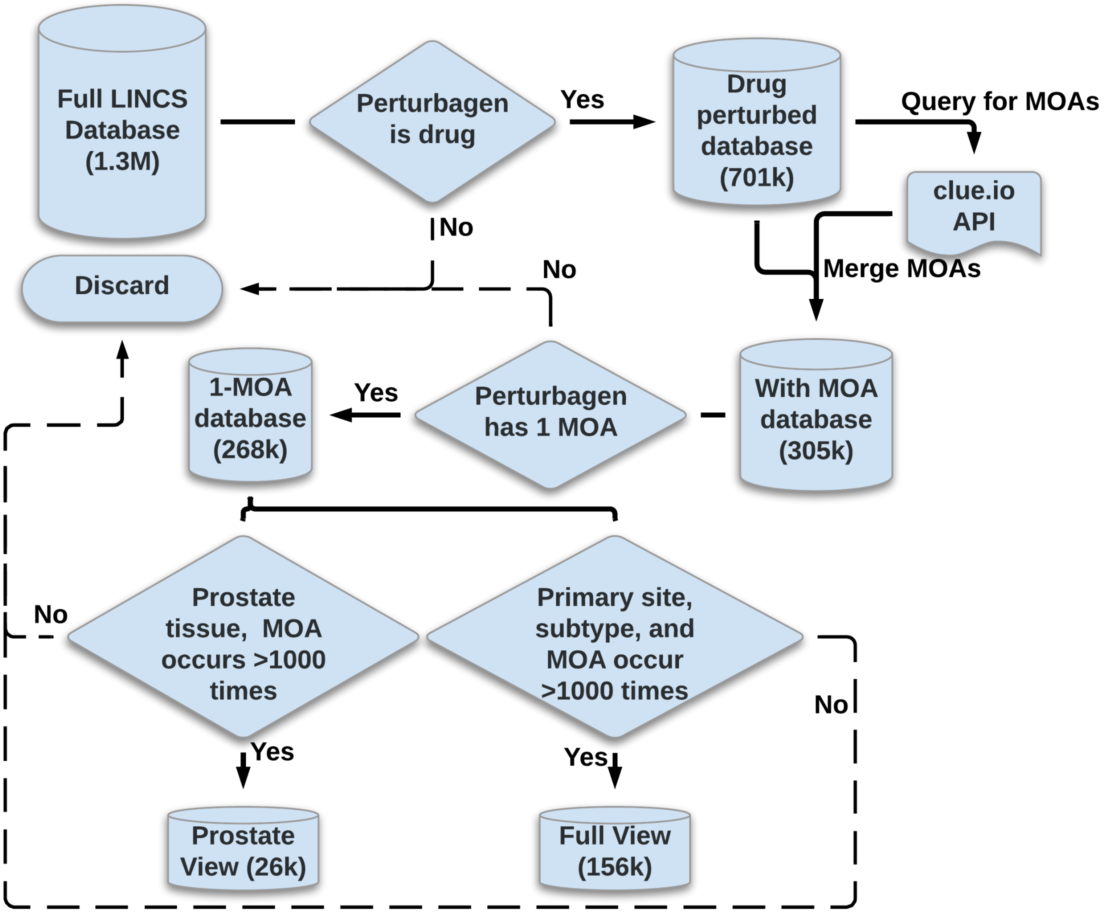
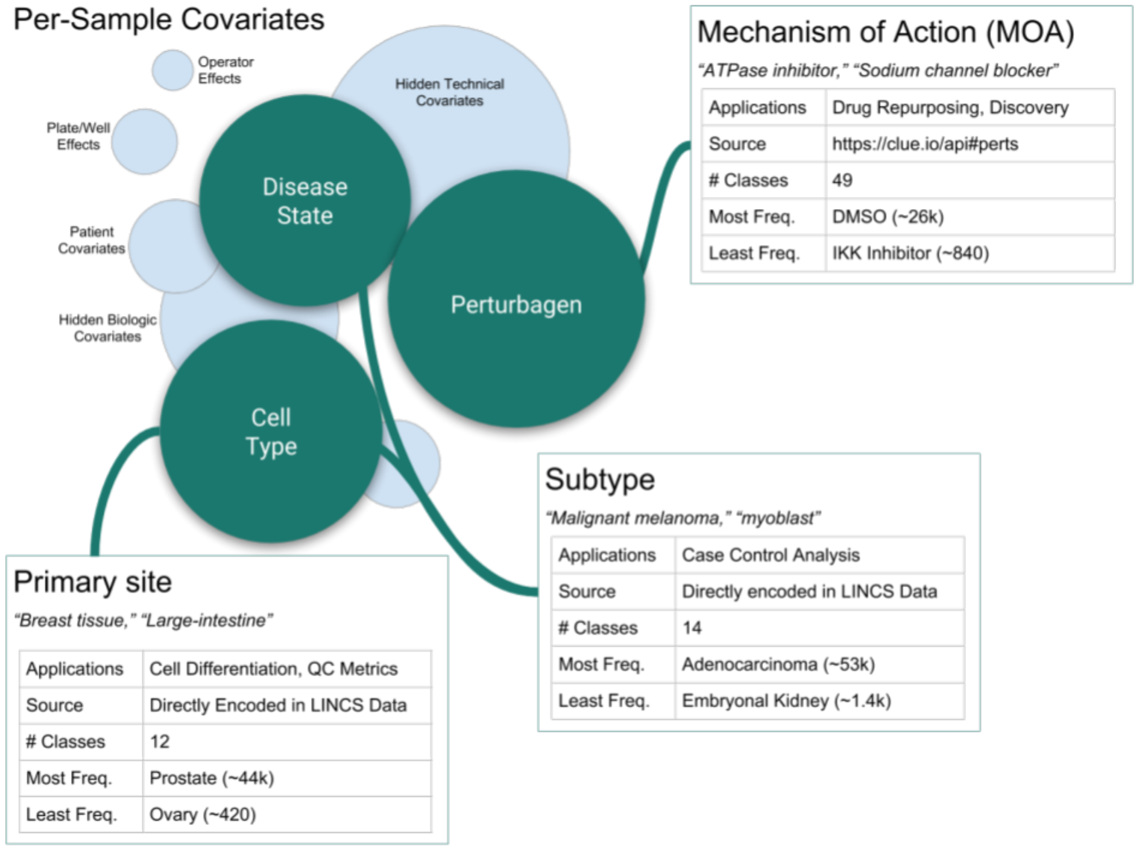


Figure 3-4: A flowchart representing the various decision points when curating our views of the full LINCS corpus.



Mechanism of Action (MOA)

"ATPase inhibitor," "Sodium channel blocker"

Applications	Drug Repurposing, Discovery
Source	https://clue.io/api#perts
# Classes	49
Most Freq.	DMSO (~26k)
Least Freq.	IKK Inhibitor (~840)

Subtype

"Malignant melanoma," "myoblast"

Applications	Case Control Analysis
Source	Directly encoded in LINCS Data
# Classes	14
Most Freq.	Adenocarcinoma (~53k)
Least Freq.	Embryonal Kidney (~1.4k)

Primary site

"Breast tissue," "Large-intestine"

Applications	Cell Differentiation, QC Metrics
Source	Directly Encoded in LINCS Data
# Classes	12
Most Freq.	Prostate (~44k)
Least Freq.	Ovary (~420)

Figure 3-5: A visual description of our three tasks, including upon which aspects of the underlying data covariates they are dependent, sample applications motivating each task, and a description of the relative class imbalance for each.

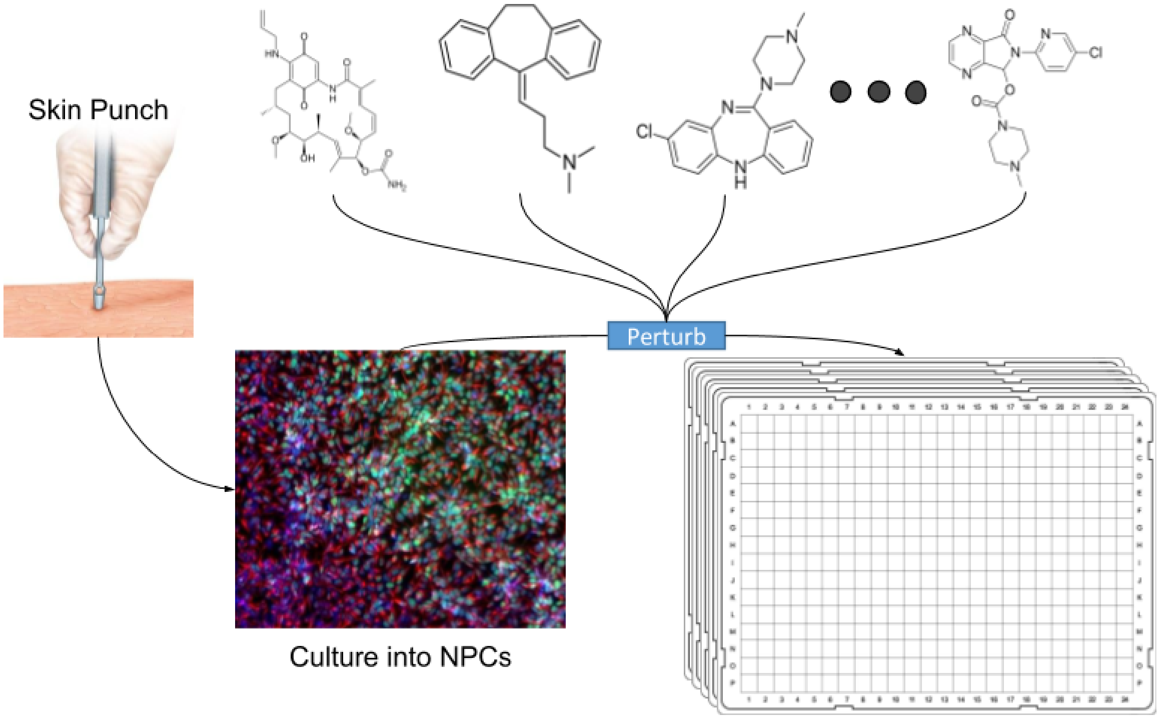


Figure 3-6: The dataset creation pipeline for the MGH NeuroBank Corpus. These data were created by our collaborators at the Center for Quantiative Health (dir. Professor Roy Perlis) and Chemical Neurobiology Laboratory (dir. Professor Stephen Haggarty) at Massachusetts General Hospital. Of particular note are Drs. Jennifer Wang, Wen-Ning Zhao, and Stephen D. Sheridan.

Per-subject generalizability is an important, oft-overlooked element of performance in this domain—many studies which rely only on data from the LINCS public corpora, for example, are often working with only one to two subjects per tissue type, which means their expected generalizability would likely be worse and the magnitude of the problem is difficult to assess. Our experiments here will provide some estimate of the performance delta that should be observed when generalizing to new subjects.

Figure 3-6 shows a graphical representation the data creation pipeline.

3.2.2 Models

We compare a variety of standard classifiers, all (save GCNNs) implemented via `scikit-learn` [152] for maximal reproducibility and ease of use. GCNNs, as previously stated, were implemented via the method of [43].

In the interest of space, we will not provide a primer on each of the standard

Dataset Statistics:

Dataset	Number of Samples	# Cell Lines	Most Frequent Cell Line	Least Frequent Cell Line
Full LINCS	156,461	36	MCF7 (26,546)	NCIH716 (8)
Prostate Only LINCS	25,565	2	PC3 (13,625)	VCAP (11,940)
MGH NeuroBank (Level 4)	5602	5	N/A (1133)	N/A (1109)
MGH NeuroBank (Level 5)	1894	5	N/A (380)	N/A (377)

Task Statistics:

Dataset	Task	# Classes	Most Frequent Class	Least Frequent Class
LINCS (Full)	Primary Site	12	Prostate (43,686)	Ovary (415)
	Subtype	14	Adenocarcinoma (53,245)	Embryonal Kidney (1384)
	MOA	49	DMSO (25,638)	IKK Inhibitor (828)
LINCS (Prostate Only)	MOA	9	DMSO (8833)	Serotonin Receptor Antagonist (1029)
MGH NeuroBank (Level 4)	Perturbagen	60	DMSO (383)	Ruboxistaurin (78)
MGH NeuroBank (Level 5)	Perturbagen	60	DMSO (130)	Ruboxistaurin (27)

Table 3.1: Population Statistics for our Datasets and Tasks.

methods mentioned below in this work, but instead make clear why they were chosen to benchmark for this task and indicate which `scikit-learn` class was used to implement them. For a description of GCNNs see Section 3.1.3.

Classifiers Tested

Feed-forward artificial neural network (FF-ANN) classifiers FF-ANNs are a common, powerful, non-linear modelling technique, and were used in many of the prior works on gene expression data. However, partly because they do not assume any particular structure of their input and are thus least constrained, they are relatively inefficient learners. Some postulate that this inefficiency is simply due to their larger parameter overhead; however, the full reason is not yet known. Implemented via the `MLPClassifier` class.

Linear classifiers Linear classifiers, subsuming both logistic regression (LR) and support vector classifiers (SVCs), are extremely common across all domains, including traditional bioinformatics analyses, and are interpretable. Implemented via the `SGDClassifier` class.

Random forests Random forests are not as commonly used in traditional bioinformatics use cases, but are thought to often provide a compelling non-neural but still non-linear baseline. They are composed of many bagged random decision trees. Implemented via the `RandomforestClassifier` class.

K nearest neighbors classifiers k-NN methods are commonly used in this domain for clustering analyses, and we hope that investigating their performance here can help inform further choices for those and other analyses in these domains. They also shed some light on appropriate distance metrics. Implemented via the `KNeighborsClassifier` class. Index construction, often a computationally intensive task on large datasets, was done via either brute force search, the construction of a KDTree, or the construction of a Ball Tree, as determined by `scikit-learn`'s '`algorithm=auto`' setting.

Decision trees Decision trees are low powered, but extremely mechanistically interpretable. Implemented via the `DecisionTreeClassifier` class.

Graph Convolutional Neural Networks (GCNNs) GCNNs allow us to inject prior biological knowledge in the form of a genetic regulatory network into a neural network, offering structural efficiency improvements and domain appropriate bias.

In this work, our GCNN is built using the spectral approach defined by [43]. We encourage interested readers to refer to the primary source for full details regarding this algorithm, but we provide a brief explanation of the method here. In particular, this method of graph convolutional processing approximates localized filters in the graph Fourier space via polynomials of the graph Laplacian. As follows from the graph theoretical nature of the Laplacian, restricting the order of these polynomials yields a localized radius of effect when impacting on the featurization of each graph node. These polynomials are realized in an efficient manner by relying on the stable recurrence relation of the Chebyshev polynomials, which form an orthogonal basis of a relevant Hilbert space and have been used historically in graph signal analysis for approximate wavelet analysis. Ultimately, this yields a means of producing fast, localized graph convolutional filters. Graph pooling is implemented via the coarsening phase of the Gracus multilevel clustering algorithm [48].

We use the code of [43] with minor modifications to support multi-component graphs. We considered a number of potential regulatory graphs, both tissue specific and tissue independent.

Other Classifiers Considered We also tested Naïve Bayes classifiers, Gaussian Processes Classifiers, Quadratic Discriminant Analysis, Boosted methods via Adaboost, and Kernel Support Vector Classifiers, but these classifiers were removed from our experimental lineup for reasons varying from poor performance, non-insightful new results, computational intensivity, or combinations thereof.

3.2.3 Genetic Regulatory Networks Considered

We considered a number of possible graphs, including those constructed from our data (a gene-gene pairwise correlation graph) and graphs pulled from the literature. For our literature sourced graphs, we will offer brief summaries of the source works here, but interested readers should refer to the primary sources for full details of the graph constructions—for our purposes it suffices to note that they are constructed to capture known or suspected genetic regulatory relationships as in Figure 3-3.

Our tissue-independent regulatory network is a network of transcription-factor and micro-RNA mediated regulatory relationships summarized from 25 literature defined external datasets [122].[5] This graph is unweighted.

Our tissue-dependent regulatory network is built from a probabilistic model of tissue-specific gene-gene correlations [71].[6] We considered a number of possible tissues, profiling both relevant tissues (neuron for MGH NeuroBank and prostate gland for the prostate specific LINCS dataset) and irrelevant tissues (tooth, pancreas, skin fibroblast) to help differentiate whether any performance gains observed with these graphs were due to the appropriate tissue specificity or simply due to this style of graph construction being superior. These graphs were all weighted, with edge weights estimating the confidence in the true existence of that edge, determined via a probabilistic model. Ultimately, no tissue specific graph outperformed the tissue-indepent graph of [122], so the distinction between relevant or irrelevant tissues proved

[5] Networks available for download here: `http://www.regnetworkweb.org/download.jsp`

[6] Networks available for download here: `http://hb.flatironinstitute.org/download`

negligible.

Last but not least, we considered a learned graph, whose (weighted) adjacency matrix was determined via correlation coefficients between genes in our data. This graph is by default fully connected, so to induce effective sparsity, we removed all edges corresponding to correlative relationships with a p value of greater than 0.05. This graph also offered no performance advantage over the tissue-independent network in early experiments, so we removed it from our analyses and focused solely on the literature-defined graphs.

When working with any weighted graphs, we culled all edges with confidence below a cutoff threshold, which was tuned with all other hyperparameters. We treated all graphs as undirected, allowing them to capture merely a notion of regulatory interaction rather than any directed up- or down-regulation. This is certainly a simplification, and exploring more complex representations of regulatory graphs is definitely a promising area of future studies, but using undirected graphs here yields significant technical simplifications for this work enabling these graphs to work natively within our chosen graph convolutional framework.

3.2.4 Experiments

Hyperparameter Search & Technical Setup

Hyperparameters for all classifiers were determined by a random search [16] over all possible parameters and tasks, including over the number and sizes of hidden layers for FF-ANNs and number of graph convolution layers/filter sizes/pooling sizes, loss types, etc. In addition to random search, we also rotated the discovered optimal hyperparameters across tasks during various stages of the search procedure and made certain manual tweaks in pursuit of obtaining strong performance metrics for all models, particularly baseline methods. One notable disparity in the hyperparameter space searched is that the Scikit Learn FF-ANNs do not support dropout (only L2 regularization, which was included in our search), whereas the GCNNs do. To compensate for this potential bias, we took the optimal FF-ANN models found via the hyperparameter search and re-implemented them in Keras, as identically as possible, then performed a miniature grid-search over dropout within these models. This procedure induced a mild performance gain, but not enough to upset the observed model ordering on any tasks where GCNNs performed the best. We also did not hyperparameter optimize over batch size for FF-ANNs, but we did optimize over learning rate, a heavily related parameter, and we also tested several smaller batch sizes with our final models to ensure that we were not biasing the results against this baseline.

For GCNNs, we notably did not hyperparameter search over the number of epochs, but rotated progressively through a very limited fixed set of number of epochs for computational reasons. Additionally, GCNNs only supported a single optimizer, whereas FF-ANNs offered several options. The search process was, however, run over various considered graphs, as well as over the graph edge weight cutoff, which we used to cull irrelevant edges from our graphs.

For our benchmarking tasks, a full list of all hyperparameters tested, the distributions used to back our random search, and the final, chosen hyperparameters are available with our provided code.[7] Additionally, the optimal hyperparameters for all methods across all datasets and tasks can be found in the Appendix.

This random search was performed over 10 fold cross validation on the full LINCS dataset, and 15 fold cross validation on the private L1000 dataset (as that dataset is smaller, it warrants additional folds to improve accuracy). In each case, one fold was held out for testing, one for hyperparameter optimization, and the remaining used for training. The hyperparameter search optimized for mean accuracy over all folds, though we also report macro-F1[8] in our test set results below, as some tasks present significant class imbalance. We chose these two metrics to offer, first, a comparatively understandable metric (accuracy) which allows for a clear baseline measure (majority class performance) but is often overly forgiving for tasks with large class imbalance,

[7] See `https://github.com/mmcdermott/LINCS_Deep_Learning_Benchmarks`

[8] The *F1* score on a binary classifier is the harmonic mean of the classifier's precision and recall. The *macro-F1* score is an unweighted average of the F1 score of each class separately. Generally, *macro-F1* will offer a more conservative measure of performance for tasks with strong class imbalance.

and second, a less overt, but still commonly used, metric which compensates for class imbalance. We chose not to use AUC as it is less immediately understandable than accuracy while also not accounting for class imbalance as directly as macro-F1, and to avoid having too many evaluation metrics and thereby diluting our comparisons. For all results, statistical significance was assessed using paired t-test across all folds, followed by Benjamini-Hochberg multiple tests FDR adjustment within experimental conditions.

As different classifiers required different amounts of computational time to run, we did not run all classifiers for the same number of samples—this induces a mild bias towards the fastest running classifiers, as they will have had the opportunity to test additional hyperparameter settings. We did, however, ensure that we measured at least 60 samples for the standard FF-ANN classifier and linear models to ensure that we did not conclude any model better than those traditionally strong baselines simply due to lack of appropriate sampling. Graph convolutional networks, being highly computationally intensive, in particular on the larger datasets, were under-sampled compared to the other methods—it is possible that with more compute time their performance would improve. Note the direction of this bias: *were more samples to improve the performance of the GCNN methods further, it would only strengthen the performance gap observed on the largest datasets, and potentially cause them to outperform the simpler models on our smaller datasets.* Because this bias is in favor of our baselines, rather than the more exotic, structured GCNN models, we feel comfortable still reporting these results even though they may improve later.

For our data-flush regimes (the tasks over the full and prostate only LINCS datasets), we used only the Level 4 data. This data is less processed, but presents 3 times as much data as the analogous Level 5 data. Note that had we used Level 5 data, our filtering procedure eliminating classes with less than 1000 examples would have eliminated many classes and made the overall task much easier. For our data-sparse tests (the task on our private L1000 corpus), we tested methods on both datasets, wondering whether in this data-sparse regime, the more processed data might prove more valuable than the relatively small increase in dataset size. Additionally, as in neither dataset on the MGH corpus did we filter out infrequent classes (given the dataset size, all classes are infrequent by our standards for the full LINCS data), this change from Level 5 to Level 4 can be done more transparently than on the full LINCS datasets.

Along with our code, the results of these hyperparameter searches are all publicly available.[9]

3.3 Results & Discussion

3.3.1 LINCS Corpus

Full Corpus

Final results are shown in Table 3.2. Accuracies and macro F1s are reported averaged across unseen test folds, using hyperparameters found via a separate validation fold. Included in the results are those obtained using a majority class classifier, which simply predicts the most frequent class with probability equal to that found in the training set. This was tested across the same folds and is reported here to ground all other reported results and variances. Observed differences between mean performance of any pair of classifiers were statistically significant ($p \leq 0.05$).

We note that on all of the tested tasks, GCNNs perform best, by notable margins in accuracy and macro F1 on both primary site and subtype prediction. The margin of accuracy in MOA prediction is smaller, but still statistically significant. k-NNs performed surprisingly well on all three tasks, offering competitive performance even with the FF-ANNs. Investigations of why they performed so well revealed two findings:

1. k-NN classifiers strongly prefer traditional distance metrics (*e.g.*, Euclidean) over correlative based "distance metrics." This is notable because correlation is often used as a signal of biological similarity on these data, which may be contraindicated by these results.

[9] See `https://github.com/mmcdermott/LINCS_Deep_Learning_Benchmarks`

Task	Classifier Name	Accuracy	Macro F1
Primary Site	GCNN	**93.9 ± 0.28**	**90.5 ± 0.82**
	FF-ANN	90.6 ± 0.44	85.6 ± 0.97
	k-NNs	89.6 ± 0.30	87.2 ± 0.61
	Linear Classifier	60.9 ± 0.50	47.6 ± 0.63
	Random Forest	57.2 ± 0.48	40.2 ± 0.77
	Decision Tree	44.4 ± 0.70	24.7 ± 2.22
	Majority Class	27.9 ± 0.16	3.63 ± 0.02
Subtype	GCNN	**93.5 ± 0.34**	**91.7 ± 2.1**
	FF-ANN	90.5 ± 0.30	88.5 ± 0.54
	k-NNs	89.8 ± 0.13	90.2 ± 0.27
	Linear Classifier	62.6 ± 0.62	56.3 ± 1.06
	Random Forest	51.7 ± 0.37	22.3 ± 0.49
	Decision Tree	41.1 ± 0.21	18.4 ± 0.62
	Majority Class	34.0 ± 0.21	3.62 ± 0.02
MOA	GCNN	**46.4 ± 0.35**	**31.6 ± 0.65**
	FF-ANN	45.9 ± 0.43	29.6 ± 0.60
	k-NNs	43.5 ± 0.50	29.5 ± 0.58
	Linear Classifier	39.1 ± 0.29	20.6 ± 0.39
	Random Forest	32.3 ± 0.40	11.5 ± 0.31
	Decision Tree	28.7 ± 0.31	8.5 ± 0.29
	Majority Class	16.4 ± 0.16	0.57 ± 0.005

Table 3.2: Performance (mean ± standard deviation) for the full, tissue-heterogenous LINCS corpus.

2. Our hyperparameter search method also changed the distance metric underlying the k-NN method. Across all tasks and datasets, the optimal distance metric was the "Canberra" distance, defined via

$$d(\boldsymbol{x}, \boldsymbol{y}) = \sum_i \frac{|\boldsymbol{x}_i - \boldsymbol{y}_i|}{|\boldsymbol{x}_i| + |\boldsymbol{y}_i|}.$$

Using this distance metric induced performance gains over correlative and traditional, Euclidean distance measures. The Canberra distance is traditionally used for integer valued vectors and we are unsure why it would be preferred here. The k-NN algorithm could choose uniformly between one of four classes of distance metrics, independently across each of our six dataset/task combinations. This means that if the choice of distance were independent of ultimate performance, the repeated use of the Canberra distance as our highest performing distance metric would be expected with somewhere between 0.02% chance (presuming the ideal choice of parameters is independent from which dataset/task is under consideration) and 25% chance (presuming the ideal choice of parameters shares is totally deterministic across datasets/tasks). We have performed no deeper analyses to determine if this apparent distance metric preference is statistically significant, or to investigate why it might be so.

Linear classifiers robustly performed well. On the MOA task, hyperparameter search selected a logistic regression model (via the `log` loss in `scikit-learn`), whereas on the Subtype and Primary Site tasks, the optimal setting used a `modified_huber` loss, which is a smooth loss that is tolerant to outliers.

Random forests and decision trees both yielded underwhelming results, particularly with respect to Macro F1. One hypothesis as to why this may be is that random forests were less sampled in the hyperparameter search than linear models. Alternatively, these results may suggest that absolute feature values are less meaningful in our data than are relationships between feature values—an idea that meshes well with the fact that this dataset is very heterogenous with respect to cell (*e.g.*, tissue) type, and the same expression level of any individual gene may mean very different things in

Classifier Name	Accuracy	Macro F1
GCNN	67.7 ± 0.76	46.0 ± 0.42
FF-ANN	$\mathbf{68.3 \pm 0.60}$	$\mathbf{50.4 \pm 0.71}$
k-NNs	66.5 ± 0.71	46.2 ± 0.89
Linear Classifier	63.8 ± 0.52	42.6 ± 1.03
Random Forest	60.4 ± 0.48	37.4 ± 0.41
Decision Tree	53.2 ± 1.16	32.6 ± 0.91
Majority Class	34.54 ± 0.05	5.71 ± 0.01

Table 3.3: Performance (mean ± standard deviation) on the prostate LINCS corpus and MOA prediction task.

different tissue types. Some might postulate that this is perhaps due to a poor search space of some critical hyperparameters; we intentionally ensured our hyperparameter search space was very broad, especially over these critical parameters. For number of trees, we searched over an equal mixture of Poisson distributions centered at 50, 200, and 400, respectively, and the optimal hyperparameters (shown in the appendix) showed a mix over this entire range. All regularization parameters were also included in our search space.

Prostate Only Corpus

Final results for prediction of prostate MOA are shown in Table 3.3. All classifier comparisons were statistically significant ($p = 0.05$). Here, FF-ANNs perform best, though GCNNs are quite competitive. Note that GCNNs still preferred tissue non-specific regulatory graphs, rather than prostate specific graphs. Again, k-NNs perform well. Here, RFs and decision trees still under-perform the other methods, but perform better with respect to macro F1 than they do on the more heterogeneous full LINCS corpus, suggesting again that perhaps they may be more appropriate on more homogeneous data sources.

As indicated in Section 3.2.2, we tested both tissue-specific and tissue-independent regulatory graphs. Surprisingly, on the prostate corpus, the GCNN performed better using the tissue independent regulatory network than it did using the prostate specific regulatory graph. This may indicate that our tissue-specific graphs suffer from some unknown problem, or that tissue-independent graphs simply perform better overall.

Similar to the full system MOA task, the optimal linear model here was a logistic regression model.

3.3.2 MGH NeuroBank Corpus

Raw Performance Results

Final results for perturbagen identification on the MGH NeuroBank corpus are shown in Table 3.4. Results were *not* statistically significantly different at $p = 0.05$ between the Level 5 data and Level 4 data for any classifier save the GCNN. All within-level classifier comparisons were statistically significant ($p = 0.05$) save between Level 5 GCNNs and RF, GCNNs and k-NNs, and k-NNs and RFs.

Classifier Name	Level 5		Level 4	
	Accuracy	Macro F1	Accuracy	Macro F1
GCNN	46.0 ± 9.90	44.0 ± 10.8	54.6 ± 3.94	56.4 ± 3.94
FF-ANN	$\mathbf{63.2 \pm 10.3}$	$\mathbf{62.7 \pm 10.8}$	$\mathbf{57.3 \pm 4.12}$	$\mathbf{58.9 \pm 4.00}$
k-NNs	46.9 ± 8.13	44.7 ± 9.15	44.9 ± 3.74	45.7 ± 3.61
Linear Classifier	52.3 ± 9.61	51.4 ± 10.0	49.1 ± 3.98	50.2 ± 3.63
Random Forest	48.0 ± 8.96	44.7 ± 9.15	43.2 ± 4.87	42.7 ± 4.75
Decision Tree	26.7 ± 8.07	25.6 ± 7.45	27.0 ± 2.02	26.4 ± 1.79
Majority Class	7.56 ± 2.37	0.23 ± 0.07	6.88 ± 0.77	0.21 ± 0.02

Table 3.4: Performance (mean ± standard deviation) on the perturbagen identity task on the MGH NeuroBank Corpus.

Here, FF-ANNs lead in performance by a wide margin compared to other methods. We interpret their strong success here relative to GCNNs to be indicative of a strong need for very large datasets for the GCNN models. Recall that this dataset is significantly smaller than our other datasets (see Table 3.1). This intuition is supported by two observations: 1) the apparent slope in GCNN performance relative to dataset size is quite steep, exceeding at all tasks on the largest dataset, nearly matching on the prostate only dataset, and failing by a large margin here, and 2) GCNNs show a statistically significant preference for the larger Level 4 data, whereas no other classifier cares between the two modalities in a statistically significant manner.

It is also possible that GCNNs are less appropriate on this corpus than on the larger corpora due to this dataset's strong neural focus. Or, it may be that GCNNs are most appropriate in heterogeneous datasets spanning many cell types.

Among the other classifiers, linear classifiers perform well, followed by k-NNs and RFs, then, much worse, by decision trees. No classifier save GCNNs shows a statistically significant preference for Level 5 data over Level 4 data, but all save GCNNs do show a (again, statistically *insignificant*) preference for Level 5 data in terms of absolute measure.

Generalization Experiments

We also used the MGH NeuroBank Corpus to assess population level generalizability, by training on four of our subjects and testing on the fifth subject. As the MGH NeuroBank Corpus contains only one subject with Bipolar Disorder, we do not ever test on this subject's data—absent more examples of any subject data in this diagnostic category, we would not expect a classifier to generalize well to this subject. Including their results causes a mild but consistent drop in mean generalization accuracy across almost all classifiers tested. We report all results here using Level 4 data as no classifier statistically significantly preferred Level 5, but the relative drops in performance observed were similar for that modality.

Results for this experiment are shown in Table 3.5. All methods showed a notable drop in accuracy on unseen subjects, ranging from a 10.2% drop for linear classifiers to an 18.5% drop for decision trees (percentages taken of per-sample accuracies, not raw percentage points). This indicates a definite unmet need for either a) more diverse datasets or b) novel methods able to better generalize to unseen subjects. Note, though, that the MGH NeuroBank corpus only contains 5 total subjects to begin with, so it may be the case that these numbers would improve significantly were we to have even a only marginally larger subject pool. Note that on a dataset like LINCS, which is much larger and thus more amenable to higher-capacity learning yet has relatively fewer cellular sources (and with those cellular sources often differing by tissue type or primary diagnosis no less), it is reasonable to imagine that this observed population

Classifier Name	Accuracy	Macro F1
GCNN	47.7 ± 6.78	48.9 ± 7.40
FF-ANN	$\mathbf{48.7 \pm 7.85}$	$\mathbf{50.1 \pm 8.34}$
k-NNs	37.9 ± 5.39	39.0 ± 6.68
Linear Classifier	44.1 ± 4.03	44.7 ± 4.21
Random Forest	38.8 ± 5.37	38.3 ± 6.76
Decision Tree	22.0 ± 3.85	21.8 ± 3.59

Table 3.5: Per (Non-BD) Subject Generalization Accuracy (mean ± standard deviation) on the MGH NeuroBank Corpus.

specific overfitting could forseeably be even worse than what we observe on the MGH dataset—this point is critical given that this dataset has been used historically for many machine learning investigations with clinically generalizable aspirations, unlike our work where the tasks are designed to aid primarily in method development.

3.4 Conclusion

3.4.1 Summary

In this work we aimed to make the following contributions:

Establish biologically meaningful benchmark tasks for gene expression data With the curation of the full and prostate-specific views of the LINCS dataset and specification of the Primary Site, Subtype, and MOA tasks, we meet this goal.

Provide robust benchmarks We provide benchmarks on the tasks defined above for 6 different types of classifiers. We establish that graph convolutional neural networks, which incorporate prior biological knowledge via genetic regulatory graphs, perform very well when dataset size is very large, and feed-forward artificial neural networks offer good performance across all dataset sizes. Additionally, we profile non-neural classifiers, including K nearest neighbor methods, random forests, linear classifiers and decision trees. K nearest neighbor methods provide surprisingly strong performance in data rich environments using the Canberra distance.

Assess how these classifiers function in data-scarce regimes We profile these same classifiers on a similar task on the smaller, privately produced MGH NeuroBank corpus. Here, we find that graph convolutional neural networks no longer offer competitive performance, but feed-forward artificial neural networks continue to perform well, as do linear models.

Assess population level generalizability We demonstrate that subject level generalizability remains an important challenge in this domain. Linear classifiers generalize best, losing only 10.2% of their per-sample accuracy, while decision trees generalize worst, losing 18.5%. It is important to note that we were only able to assess this on our smallest dataset, the MGH NeuroBank Corpus, as differing cell lines represented too divergent demographic conditions in the full LINCS dataset, so this may simply be a reflection of the small dataset size, or indicative of a more chronic problem due to the fact that gene expression corpora contain many samples per subject.

3.4.2 Future Work

There are several notable directions for future work. First, a notable absent classifier is a self-normalizing neural network (SNNN) [102]. Introduced in late 2017, SNNNs have demonstrated improvements in a battery of different tasks and warrant inclusion here. Other types of classifiers capable of using graph structures would also warrant inclusion. Additionally, there are other graph convolutional networks one could use, [110, 77], as well as other sources for our regulatory graphs. One notable contender in that domain is *HuRI: The Human Reference Protein Interactome Mapping Project*[10] which has several large databases of protein-protein interactions found experimentally through yeast two-hybrid screening methods [174, 61]. Additionally, incorporating directional information in our regulatory graphs would also enable significantly more nuanced processing. Finally, we would also like to establish other types of machine learning benchmark tasks, most notably clustering tasks, or other tasks that can

[10] `http://interactome.baderlab.org/about/`

better assess generalizability across subjects, drugs, or even measurement technologies. More investigation into what drove the success of GCNNs here, perhaps by running dataset size ablation experiments, would also help clarify their strengths. Similarly, more investigations into the failings of random forest models or the relative strengths of differing distance metrics would also be informative.

3.4.3 Relation to this Thesis

In Chapter 2, we showed that enforcing global structural constraints that were learned from data directly can offer significant advantages in data-starved settings. Here, we examine the opposite scenario, and show that we can offer significant improvements even in data-rich modelling tasks via the incorporation of prior-knowledge inspired local structure. The importance of local structure can be further seen in the models used later in this work, such as the electronic health record timeseries models explored in Chapter 4, which incorporate local structure in the form of temporal structure and knowledge-driven feature aggregation methods, or in the various settings explored in Chapter 5, where models leverage both local graph, sequential, or textual data structures for optimal modelling.

Unique from the rest of the thesis is that in this work our focus is on offering gains particularly in large-scale, data-rich settings. Whereas in the other chapters, we will often focus more on settings where significantly more unlabeled data exists vs. labeled data, here we show benefits for incorporating knowledge and structure at the largest possible data scales, which shows that these techniques can offer benefits across a variety of modelling scales.

Chapter 4

Pre-training for Electronic Health Record Data

Abstract

In this thesis, I argue that we should develop pre-training and representation learning strategies specifically for the clinical and biomedical domains, and that we should leverage structure and prior knowledge when doing so. In order to show that this is a meaningful problem, we must first show that traditional pre-training strategies don't work sufficiently well on their own in this modality. In this work, originally published in [129] we provide that demonstration, while simultaneously providing valuable benchmarks to the community.

In particular, here we establish a pre-training (PT) benchmark dataset for EHR timeseries data, establishing cohorts, a diverse set of fine-tuning tasks, and PT-focused evaluation regimes across two public EHR datasets: MIMIC-III [96] and eICU [158]. This benchmark fills an essential hole in the field by enabling a robust manner of iterating on PT strategies for this modality. We also profile two simple PT algorithms: a self-supervised, masked imputation system (similar to the successful BERT models explored in NLP) and a weakly-supervised, multi-task system. We find that weakly-supervised PT methods can offer significant gains over both traditional learning and masked imputation methods in few-shot settings, especially on tasks with strong class imbalance. This suggests definitively that we cannot simply adapt successful pre-training strategies from other domains to EHR data, and instead that new method development is needed.

Note that one can also explore pre-training algorithms over unstructured clinical text, rather than structured clinical timeseries. I served as senior author of one such work, ClinicalBERT [7], one of the most prominent language models pre-trained on clinical data available today. However, as that work relies solely on established pre-training methods, and does not contribute to the broader themes of this thesis

beyond exploring pre-training for clinical data, I don't focus on it in this chapter.

4.1 Introduction

Pre-training (PT) methods are instrumental in the success of machine learning in various domains, including examples such as ImageNet [44] PT in computer vision and language-model PT (*e.g.*, ELMO [155] or BERT [46]) in natural language processing. PT has enabled ML researchers to use large, unlabelled or weakly-labeled datasets to learn a representation of a data modality such that specific fine-tuning (FT) tasks can be learned successfully even with minimal task-specific data.

One domain where PT would be particularly impactful is processing electronic health record (EHR) data in machine learning for health (ML4H). ML4H presents a prime use-case for PT in part because there are many clinical applications of ML where the ability to leverage high-capacity models effectively even on relatively small, task-specific datasets would be important. For example, clinicians at smaller health systems could leverage public PT models produced on larger, more diverse populations to produce improved models for their specific institutions via FT. Researchers could also leverage PT models to aid in the study of rare [140] or novel (*e.g.*, COVID-19, in its early days) diseases, where there may not be enough data at *any* institution to train a high-capacity model from scratch. Lastly, researchers can leverage PT models to help reduce the need for annotating large, task-specific gold-standard datasets for specific research cohorts. These examples are also shown visually in Figure 4-1. Simultaneously, PT is eminently feasible in the clinical domain, as the large, EHR datasets collected at the point of care serve as natural sources of PT data. While these datasets are often only weakly labeled and noisy, making them challenging to work with in the context of traditional, fully supervised ML [67], PT algorithms often use self- or weakly-supervised algorithms and thus rely less on label availability and quality.

Despite these important application areas, PT has been only minimally explored in EHR timeseries data. In part, this may be because standardized sets of benchmarks

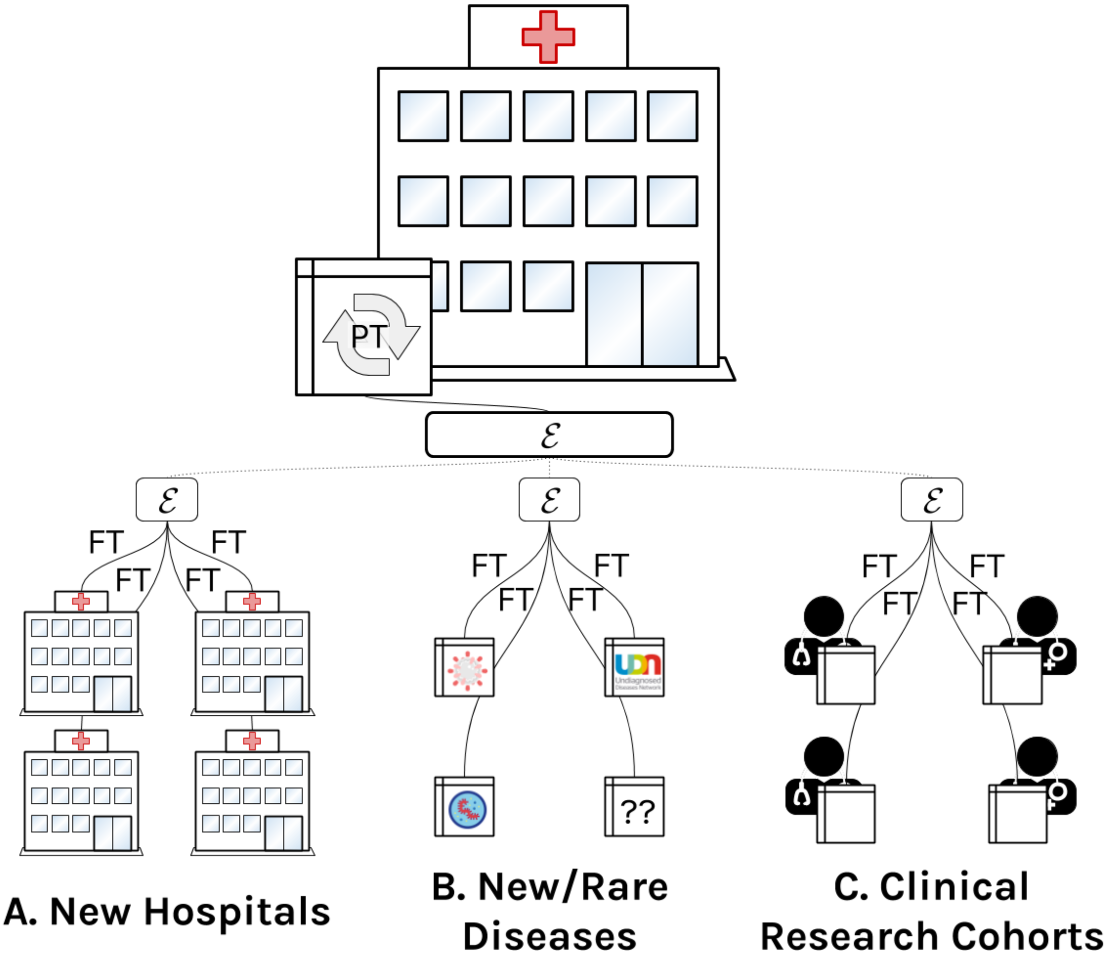


Figure 4-1: Pre-training (PT) an encoder $\mathcal{E}$ on a general domain, then fine-tuning (FT) it on a task specific problem fits naturally into many use-cases within ML4H. Examples include transferring a model from a large health system to smaller, community hospitals (**A**), specializing a model to a rare or novel disease sub-population (**B**), or supporting clinical research efforts which produce fully annotated datasets for select cohorts within a health system (**C**).

for PT/FT paradigms do not exist for EHR data. While benchmarks do exist for ML over clinical tasks in general [219, 81], these are focused on traditional supervised learning, not PT/FT. In contrast to supervised learning, PT benchmarks are concerned primarily with how a system can optimally leverage PT data to improve performance in a disparate, secondary set of FT tasks. As one might not foresee all FT tasks at PT time, any effective benchmark must assess PT algorithms across a broad variety of tasks. Critically, we also cannot simply judge FT performance at a single FT dataset size—PT methods are exciting particularly because they enable models to be leveraged effectively even in few-shot settings, so we must judge PT algorithms over a variety of FT dataset sizes. Additionally, for PT systems in particular within ML4H, where many (though not all) use cases are multi-domain in nature, we should ensure we analyze PT system performance across multiple datasets.

In this work, we introduce the first comprehensive PT benchmark for clinical EHR timeseries data. We define a suite of FT tasks to consider across MIMIC-III [96] and eICU [158], as well as evaluation procedures for model performance across a variety of FT dataset sizes. In contrast with existing clinical benchmarks (*e.g.*, [81]), our system includes multiple datasets, more tasks, and few-shot evaluations, all of which help support its use in analyzing PT algorithms. In addition, we provide two baselines against which the field can compare—first, a weakly-supervised, multi-task PT approach, and second, a masked-imputation based model reminiscent of a continuous analog of the BERT NLP model [46]. Based on these results, we find that while PT does not offer best-in-class performance for FT datasets at the full scale of MIMIC-III or eICU, PT can indeed be very helpful in the small-data regime, showing dramatic improvements in performance in particular on class-imbalanced, time-varying tasks across both datasets. These baseline results suggest that the gains offered by PT in clinical settings warrant future exploration, and we hope that this benchmark will help prompt those gains by enabling iterative development of PT paradigms in the clinical space.

4.2 Related Works

PT over EHR timeseries data has been explored only minimally, but PT on other clinical data modalities has been explored. Learning contextual representations of clinical codes, for example, has been explored via a variety of methods, often leveraging known biomedical hierarchies to improve performance [34, 187]. PT models for clinical text have also been thoroughly explored [7, 192, 249] and are regularly used in the context of clinical NLP.

Three recent examples do study topics closely related to PT over clinical timeseries data, however. In particular, [237] explored PT on tabular data via a masked-imputation based self- and semi-supervised algorithm, [230] explore using meta-learning in a semi-supervised context to specialize PT to a specific downstream task over MIMIC-III, and [196] explores a novel analog of language-modeling on discretized clinical timeseries data. Each of these three cases have slightly different foci, and thus are relevant to our work in different ways.

[237]'s work explores both self- and semi-supervised PT (of which only the self-supervised PT is relevant to us as we do not allow FT data to be leveraged at PT time); however, their primary improvements are demonstrated most soundly in semi-supervised PT, and there is minimal evidence that their algorithm offers consistent improvements in the self-supervised setting. Similarly, [230]'s work is exclusively for semi-supervised learning. As a result, neither of these two works are directly comparable to our benchmark or results. [196]'s work, however, is much more relevant. It focuses squarely on self-supervised PT, using a different analog of language model PT than our masked imputation model, and also studies clinical timeseries (albeit discretized clinical timeseries). However, their approach is tested on a private dataset, and thus is not suitable as a PT benchmark, which is our goal.

Beyond explicit PT systems, more general clinical representation learning has been explored extensively in the literature. Multi-task learning (MTL) has been explored significantly from this perspective [81, 191, 220, 49], as well as those focusing on auto-encoding, imputation, or clustering approaches [208, 60].

Benchmarks for PT paradigms are also growing in use in other domains. [167] examines PT in the context of proteins, for example, and [114] defines a benchmark for cross-lingual PT systems, a topic that is also of interest in clinical contexts such as diagnosing speech pathologies [11].

4.3 Problem Formulation & Notation

Let $\boldsymbol{X}_{\mathrm{PT}} \in \mathbb{R}^{N_{\mathrm{PT}} \times D}$, paired with a collection of auxiliary tasks $\mathcal{T}_{\mathrm{PT}}$ with associated labels $\boldsymbol{Y}_{\mathrm{PT}} \in \mathbb{R}^{N_{\mathrm{PT}} \times |\mathcal{T}_{\mathrm{PT}}|}$ be our "pre-training" (PT) dataset. In addition, let $\mathcal{T}_{\mathrm{FT}}$ denote our set of downstream (fine-tuning/FT) tasks and $\boldsymbol{X}_{\mathrm{FT}} \in \mathbb{R}^{N_{\mathrm{FT}} \times D}, \boldsymbol{Y}_{\mathrm{FT}} \in \mathbb{R}^{N_{\mathrm{FT}} \times |\mathcal{T}_{\mathrm{FT}}|}$ denote the corresponding FT dataset. Note that $\boldsymbol{X}_{\mathrm{FT}}$ may intersect non-trivially with $\boldsymbol{X}_{\mathrm{PT}}$ (*i.e.*, some data may overlap between the PT and the FT settings), *but* no tasks overlap directly between $\mathcal{T}_{\mathrm{PT}}$ and $\mathcal{T}_{\mathrm{FT}}$. Given this lack of overlap in tasks, $\mathcal{T}_{\mathrm{PT}}/\boldsymbol{Y}_{\mathrm{PT}}$ can serve as a form of weak-supervision for the ultimate FT tasks. In most practical scenarios, it will be the case that $N_{\mathrm{FT}} \ll N_{\mathrm{PT}}$.

Let our *pre-training* model be given by $\mathcal{M}(\boldsymbol{x}) = \mathcal{D}_{\mathrm{PT}}(\mathcal{E}(\boldsymbol{x}))$, where $\mathcal{E}$ is an *encoder* (which we will ultimately transfer during FT) and $\mathcal{D}_{\mathrm{PT}}$ is a *PT specific decoder* (which will not be transferred). Then, the goal of PT is to use the dataset $\boldsymbol{X}_{\mathrm{PT}}$ (and possibly $\boldsymbol{Y}_{\mathrm{PT}}$) to learn the parameters of the pre-training model $\mathcal{M}$ such that $\mathcal{E}$ offers strong transfer performance for the tasks $\mathcal{T}_{\mathrm{FT}}$, all without actually leveraging (or even knowing about) the fine-tuning labels $\boldsymbol{Y}_{\mathrm{FT}}$ at any point during the PT process. Note that at fine-tuning time we will also train a freshly initialized decoder $\mathcal{D}_{\mathrm{FT}}$ such that the full FT model makes predictions $\hat{\boldsymbol{y}} = \mathcal{D}_{\mathrm{FT}}(\mathcal{E}(\boldsymbol{x}))$.

4.4 High-level Overview

Here, we will provide an overview of the rest of the chapter, to help provide a high-level grounding for the more detailed content in Section 4.5, which defines the benchmark's data and usage, and Section 4.6, which details our baseline PT experiments.

Our benchmark defines two separate cohorts: one over MIMIC-III and one over eICU (Section 4.5.1). Cohorts consist of timeseries of labs, vitals, & treatments. In addition, we also define a set of 10 clinically meaningful downstream tasks which we use as FT tasks (Section 4.5.2) to judge PT algorithms within our benchmark.

PT systems using our benchmark must fall into one of two categories: self-supervised, in which case they can only leverage the labs, vitals, and treatment dataset during PT, or weakly-supervised, in which case they can also leverage "off-target" tasks during PT as auxiliary labels (Section 4.5.4). After PT, models are fine-tuned under two distinct transfer regimes (Section 4.5.5) and across datasets ranging in size (Section 4.5.6) to simulate extreme $\frac{N_{\text{PT}}}{N_{\text{FT}}}$ ratios.

Ultimately, PT systems are judged on their final FT scores across all tasks, datasets, and $\frac{N_{\text{PT}}}{N_{\text{FT}}}$ ratios. In particular, to profile a PT system on our benchmark, one simply downloads the provided cohorts and the 5 standardized train/validation/test splits, tunes hyperparameter and trains their PT model according to the appropriate procedures (for either self- or weakly-supervised methods), then fine-tunes the model against our 10 downstream tasks at all dataset sizes. This usage procedure is detailed more in Section 4.5.7.

To demonstrate this use in practice, and establish baseline results for further research, we profile one self-supervised and one weakly-supervised PT method against our tasks in the manner described above (Section 4.6). Ultimately, even with simple PT methods, we see important improvements in the few-shot context (Section 4.6.4).

4.5 Pre-training Benchmark

4.5.1 Data Cohorts & Pre-processing

MIMIC-III Cohort Selection Our MIMIC-III [96] cohort is extracted via the MIMIC-Extract pipeline [219], with missingness threshold set to 2% and otherwise default parameters. This pipeline extracts a cohort of ICU stay records corresponding to the first ICU stay of patients over age 15, extracting labs, vitals, and treatments, with labs & vitals aggregated into clinically meaningful buckets to produce a more robust

Dataset	Split	# Stays	# Patients	# Patient-Hrs	Hrs/Patient
MIMIC-III	Train	17.5K	17.5K	1.48M ± 2.22K (1.47M–1.48M)	84.3 ± 47.2 (3–239)
	Tuning	2.19K	2.19K	183K ± 1.5K (182K–186K)	83.9 ± 46.7 (6–239)
	Held-out Test	2.19K	2.19K	184K ± 2.33K (182K–188K)	84.3 ± 47.2 (3–239)
eICU	Train	58.1K ± 34.4 (58.1K–58.2K)	51.5K	4.1M	70.6 ± 45.8 (25–242)
	Tuning	7.27K ± 16.2 (7.26K–7.3K)	6.44K	517K ± 3.99K (511K–522K)	71.1 ± 46.3 (25–242)
	Held-out Test	7.28K ± 34.2 (7.26K–7.34K)	6.44K	518K ± 2.46K (516K–522K)	71.1 ± 46.4 (25–242)

Table 4.1: Dataset Statistics for our MIMIC-III and eICU Cohorts. Values are aggregated across the 5 random splits of our dataset, shown in the format "[mean] ± [standard deviation] ([min]–[max])." If only a single number is shown, the quantity does not vary enough to show any difference at the presented precision. Though the MIMIC-III cohort does include patients with very short stays, in practice we restrict our analyses to only those with sufficient data to encompass a single input window (at least 12 hours).

representation [148]. ICD codes, comfort-measures-only (CMO)/do-not-resuscitate (DNR) codes, and records of death, discharge, and readmission are also extracted via novel extraction code primarily as task labels, not input signals, though CMO/DNR codes that are present or added during an input window are incorporated as features as well. Lastly, we also extract static, demographic data at a per-patient level. Appendix Tables 4.6 & 4.7 reports the set of all labs & vitals we consider in this work along with their relative measurement rate. Treatments studied include various forms of ventilation, vasopressors, or fluid boluses (See Appendix Section 4.8.1 for a full list). Static data includes age, gender, ethnicity, insurance type, admission type, and first care unit. Basic dataset statistics are shown in Table 4.1.

eICU Cohort Selection To extract the eICU [158] data, we attempt to mimic the structure of our MIMIC-III cohort wherever possible. This cohort also extracts labs and vitals, as well as static demographic data (age, gender, ethnicity, and unit type). We also extract records of death and discharge to form our downstream tasks. This cohort contains only patients over age 15 and only labs & vitals measured for at least 5% of all observed time-points are included. In addition, as the eICU dataset is multi-institution, we also restrict our data to correspond only to institutions with at least 500 patients in the dataset. Extraction code for our eICU extraction system will be released publicly after publication. Basic dataset statistics are shown in Table 4.1.

Dataset Post-processing Both datasets are standardized to hourly granularity and represented as numerical timeseries with missingness. Treatment records are also standardized hourly and concatenated to the numerical series via a one-hot encoding. Static data are duplicated and appended to each hour of the series. To form a pre-training or fine-tuning sample, we first sample a random ICU stay from the record, then a random end-time T within that stay, and treat all data for that stay prior to T as the *input window* for this sample, and the task labels corresponding either to the end of the patient's overall stay (for static tasks) or within a prescribed prediction window after T (for time-varying tasks) as the *labels* for this sample (see

Section 4.5.2 for more details on task labels). Users may choose to featurize this input window however they like—in our baselines, for example, rather than processing the entire input window $[0, T]$, we use a fixed size window ranging from 12–96 hours ending at T for computational efficiency. Note that in evaluating rolling or time-varying tasks, whose labels will vary throughout patients' stays, we sample multiple random endpoints and aggregate evaluation results in a per-patient manner across those different endpoints to approximate the expected performance of such tasks at a per-patient level.

Dataset Splits & Release To capture all relevant sources of variance, our benchmark consists of 5 random train/tuning/test splits (split by patient and ICU stay), so that a given PT system can separately undergo hyperparameter tuning, training, and evaluation (including fine-tuning training/hyperparameter tuning) across 5 different data splits. These splits are publicly available with the rest of our benchmark.

4.5.2 Benchmark Fine-tuning Tasks

Our benchmark consists of 10 FT tasks that span a variety of traditional ML4H targets as well as several new tasks. In the interest of ensuring our set of tasks is sufficiently diverse so as to be as generalizable as possible over FT use cases, and in order to capture the variety of task definitions commonly used in the literature, we formulate many of our tasks in a multi-label format, with distinct labels within a single task spanning different possible configurations of the task. For example, our "Imminent Mortality" task encompasses a *multi-label* prediction of both mortality within 24 hours and mortality within 48 hours, both with different gap times. We'll use the term "task" to refer to the overarching learning target (*e.g.*, "Imminent Mortality") and "label" to refer to an individual prediction target (*e.g.*, binary prediction of mortality within 24 hours, or, using an example target from a different task, the presence of a particular ICD code in the record). In addition, we also will use the term "Rolling" to correspond to tasks with time-varying labels (*e.g.*, prediction of mortality within the next 24 hours), "Static" to correspond to tasks that have a single, fixed label corresponding to the end of the patient's stay (*e.g.*, predicting overall ICD codes), or "Autoregressive" to correspond to a task that explicitly is involved with forecasting the future state of the patient *in the feature-space used by our model.* Note that the focus in our task selection is first on utility for an ML benchmark, and second on direct clinical utility. Where the latter is certainly important, we choose to focus here on including a broad variety of tasks, on examining tasks that are well represented in the current ML literature, and on tasks can be defined at scale over MIMIC-III and/or eICU, such that we can easily examine the performance of models across various dataset sizes within MIMIC-III without anchoring ourselves to a particular set of clinical cohorts that already have gold-standard labels.

A full table of the tasks we use, across both datasets, is given in Table 4.2. In the remainder of this section, we will walk through each task in more detail. For each task, we will report a formal definition, over which cohorts the task is defined in this benchmark, over what input windows the task is predicted (*e.g.*, either throughout the patient's stay or only based on the first 24 hours), a brief pointer to any relevant prior literature for the task, and more detailed majority class statistics for the tasks/labels. When reporting task definitions, we will frame our rolling tasks relative to the last measured timepoint in the sample's input window—*e.g.*, if an input sample corresponds to the ICU stay record of patient p up to time T, and the task is defined over a prediction window of 24 hours, with a gap time of 2 hours, then the task will capture instances of a label within the time window $[T+2, T+24]$ for patient p. Note that we include the gap time to ensure both that the learning task isn't biased by any potential temporal leakage in the data and that any superficial signals that would be already known to clinicians during the input window (which is more likely when the task event, *e.g.*, mortality, would take place just after T) don't overwhelm the learning objective. Task statistics will be reported at a *per-patient* level (*i.e.*, rolling tasks will have labels first aggregated within a patient's record, then across patients, so as not to be biased by the behavior of patients with longer overall stays), and aggregated over the train set of all 5 standardized train/tuning/test splits in our benchmark. All statistics (as well as some not reported here) are also available in table form in the appendix, in Supplementary Tables 4.4, 4.5, 4.6, 4.7, and 4.8).

Imminent Mortality: MOR

Definition: We predict whether the patient's recorded time of death is within the subsequent 24/48 hours, with a 2/6 hour gap time.

Cohort: This task is available on both cohorts.

Input Window: Throughout the entire stay.

Prior Art: Prediction of imminent mortality has been studied extensively as a silver learning signal for more general physiological decompensation [81].

Statistics: 24h/48h mortality is false for $97.6 \pm 9.91\%/95.4 \pm 17.36\%$ and $97.9 \pm 10.26\%/96.2 \pm 16.41\%$ of hours per-patient for MIMIC-III and eICU.

Comfort Measures: CMO

Definition: "Comfort Measures Only" (CMO) orders indicate that the (usually terminally ill) patient has requested to receive care *only* designed to provide comfort, not treatment, and otherwise the course of illness should be allowed to progress (typically to mortality). We predict whether a patient will add a CMO flag to their record over the next 24/48 hours, using a 2/6 hour gap time.

Cohort: This task is only available on MIMIC-III.

Input Window: Throughout the entire stay.

Prior Art: In the traditional ML4H community, CMO prediction is somewhat understudied. The only work we know of to study this prediction task is [123], which uses natural language processing over clinical notes and structured data to predict CMO codes and do not resuscitate (DNR) codes.

Statistics: 24h/48h CMO status is false for $99.1 \pm 5.55\%/98.5 \pm 9.59\%$ of hours per-patient.

DNR Ordered: DNR

Definition: "Do Not Resuscitate" (DNR) orders indicate that the patient has requested to not receive resuscitation care (*e.g.*, cardiopulmonary resuscitation a.k.a. CPR). We predict whether a patient will add a DNR flag to their record over the next 24/48

Task	Abbr.	Specific Labels	Temporal	Gap	Pred.	Type	In eICU?	Rel. Work	Majority Class Acc.	
									MIMIC-III	eICU
Imminent Mortality	MOR	Mortality (24h)	Rolling	2h	24h	Bin.	✓	[81]	97%	97%
		Mortality (48h)		6h	48h		✓			
Comfort Measures	CMO	CMO added (24h)	Rolling	2h	24h	Bin.		[123]	98%	
		CMO added (48h)		6h	48h					
DNR Ordered	DNR	DNR added (24h)	Rolling	2h	24h	Bin.		[123]	96%	
		DNR added (48h)		6h	48h					
Imminent Discharge	DIS	Discharge (24h)	Rolling	2h	24h	MC	✓	[18]	43%	44%
		Discharge (48h)		6h	48h		✓			
ICD Code Prediction	ICD	Appendix Table 4.8	Static	12h	N/A	ML		[49, 81]	67%	
Long Length-of-Stay	LOS		Static	12h	N/A	Bin.	✓	[148, 81, 219]	53%	67%
30 Day ICU Readmission	REA		Static	N/A	N/A	Bin.		[81]	95%	
Final Acuity	ACU		Static	12h	N/A	MC	✓	[29, 219, 191]	25%	60%
Next Timepoint	WBM	Appendix Tables 4.6 & 4.7	AR	0h	1h	ML		[26]	92%	88%
Future Treatment Sequence	FTS		AR	N/A	N/A	SMC		[228, 153]	97%	

Table 4.2: The set of tasks defined in our benchmark dataset. These tasks are used both as FT tasks or for signals of weak-supervision during PT. Average (macro) majority class accuracy across train folds is reported for all classification tasks to give an estimate of the relative level of class imbalance of the task. More detailed MCA statistics are reported in Appendix Table 4.4. Both Future Treatment Sequence (FTS), and our more granular Final Acuity (ACU) task are novel tasks. *Abbreviations: AR*: Auto-regressive, *Bin.*: binary classification, *ML:* binary multi-label classification, *MC:* multi-class classification, *SMC*: sequential decoding multi-class classification, *Reg.*: regression.

hours, using a 2/6 hour gap time.

Cohort: This task is only available on MIMIC-III.

Input Window: Throughout the entire stay.

Prior Art: To the best of our knowledge this task has only been studied within the ML4H community in [123].

Statistics: 24h/48h DNR status is false for $96.6 \pm 16.16\%/96.1 \pm 18.03\%$ of hours per-patient.

Imminent Discharge: DIS

Definition: We predict whether the patient will be discharged, *and if so to where* (*e.g.*, discharged home vs. to a skilled nursing facility), within the next 24/48 hours, using a 2/6 hour gap time. Unlike the prior tasks, this task is both multi-label (across prediction/gap windows) and multi-class (across discharge locations).

Cohort: This task is available on both MIMIC-III and eICU.

Input Window: Throughout the entire stay.

Prior Art: Imminent discharge has been primarily predicted in operational contexts, rather than for use as a signal of acuity, *e.g.*, [18].

Statistics: Within 24 hours, the patients are more commonly not discharged than they are discharged to any other possible individual discharge location ($57.7 \pm 24.68\%$ and $48.2 \pm 26.67\%$ of hours per-patient for MIMIC-III and eICU). Within 48 hours, MIMIC-III patients are again most commonly not discharged ($27.6 \pm 26.58\%$), but eICU patients are most commonly discharged home ($40.2 \pm 35.95\%$). A full list of possible discharge locations, and their prevalence per-hour, per-patient, is shown in Appendix Tables 4.9,4.10 for the MIMIC-III and eICU cohorts.

ICD Code Prediction: ICD

Definition: We predict the multi-label presence of each of the 19 major ICD categories under the categorization of [193].

Cohort: ICD codes are available only on the MIMIC-III dataset.

Input Window: The first 24 hours of data.

Prior Art: Prediction of ICD codes is commonly studied in ML4H as a phenotyping task [81, 49].

Statistics: Per-label majority class accuracies are shown in Supplementary Table 4.8. Macro-averaged across all categories, the majority class accuracy of this task is $67.0 \pm 18.07\%$.

Long Length-of-Stay: LOS

Definition: We predict via binary classification whether a patient's total length-of-stay will be longer than 3 days or not.

Cohort: LOS is available on both cohorts.

Input Window: The first 24 hours of data.

Prior Art: Long LOS has been predicted numerous times, both in a classification sense for 3-day LOS [219] and 7-day LOS [81].

Statistics: Patient LOS is longer than 3 days $47.1 \pm 0.11\%$ and $33.3 \pm 0.04\%$ of the time on MIMIC-III and eICU.

30 Day ICU Readmission: REA

Definition: We predict whether or not patients *who are successfully discharged* will be readmitted *to the ICU*[1] within 30 days.

Cohort: This task is defined only on the MIMIC-III cohort. As MIMIC-Extract extracts a cohort only of patients' *first* ICU stays [219], this task also has the bias of only being analyzed on the first ICU visit for a patient.

Input Window: 30-day ICU readmission is predicted over the entirety of the patient's data, up until discharge. In practice this often means that it will be predicted over a fixed size window of, *e.g.*, 48 hours prior to discharge.

Prior Art: Rajkomar et al. [165] examine overall hospital readmission in their work.

Statistics: $95.1 \pm 0.09\%$ of patients aren't readmitted within 30 days.

[1] Hospital readmission would be both a more natural and more actionable task in practice; however, the granularity of our input data only permits ICU readmission, so we use this as a proxy for the more traditional hospital readmission task.

Final Acuity: ACU

Definition: We predict, in a multi-class manner, whether the patient will die in the hospital—and if so, when (*e.g.*, In-ICU v. In-Hospital)—or be discharged—and if so, to where (*e.g.*, Home, a Skilled Nursing Facility)—at the end of their stay.

Cohort: This task is defined over both cohorts.

Input Window: The first 24 hours of data.

Prior Art: Various sub-forms of this task have been explored historically. In-ICU and in-hospital mortality, for example, have been explored as separate, binary classification tasks in numerous ways [81, 219, 28]. Challenging the model to predict death (including location of mortality) and the final discharge location jointly is novel, to the best of our knowledge.

Statistics: Prevalences of all classes for the final acuity task are shown in Supplementary Tables 4.12 and 4.11, for the MIMIC-III and eICU cohorts. The macro averaged majority class accuracy for this task, however, is $25.3 \pm 0.24\%$ of patients being discharged to a home health care system and $59.7 \pm 0.1\%$ being discharged to home for MIMIC-III and eICU.

Next Timepoint Will be Measured: WBM

Definition: We predict which labs & vitals will be measured in the next hour via multi-label binary classification.

Cohort: This task is defined over both cohorts.

Input Window: Throughout the patient's stay.

Prior Art: Imputation and forecasting over clinical data has been explored extensively in the past, both as a necessary technical pre-processing step for large pipelines and as a vehicle for direct use in other clinical tasks, such as anomaly detection [125]. The classification formulation is somewhat less common than the regression formulation, but the analysis of measurement observation patterns in clinical data in general has been explored in a number of contexts beyond just prediction [26].

Statistics: The labs & vitals over which we predict, along with their observed measurement rates, are shown in Appendix Tables 4.6 & 4.7 for both the MIMIC-III and eICU cohorts. Macro averaged majority class accuracy per-hour, per-patient for this task is $92.1 \pm 9.18\%$ on MIMIC-III and $88.0 \pm 19.61\%$ on eICU.

Future Treatment Sequence: FTS

Definition: We predict the sequence of combinations of ventilation, vasopressor, and fluid bolus treatments the patient will receive over the remainder of their stay (bucketed to an hourly granularity), in a duration agnostic manner, meaning this task does not differentiate between a patient who is ventilated for one hour, followed by receiving vasopressors for two hours and a patient who is ventilated for two hours, followed by receiving vasopressors for one hour—in both cases, the task labels would simply be the sequence "ventilation, vasopressors."

As this task is a sequential decoding task, predictions for FTS must use more specialized prediction heads and training regimes than on our other tasks; our baselines, for example, rely on LSTM RNN decoders and teacher forcing [105], but other users may attempt different strategies. We evaluate this task in an autoregressive manner also using teacher forcing [105].

Cohort: This task is defined only on the MIMIC-III cohort.

Input Window: Throughout the patient's stay.

Prior Art: While this task formulation is novel, researchers have investigated learning optimal control policies for applications of treatments, including ventilators or vasopressors [228, 153, 88].

Statistics: We show the relative frequency of the various treatment combinations in Appendix Figure 4-5. The majority class accuracy of this task at a per-patient, per-hour level is $97.4 \pm 2.77\%$.

4.5.3 Pre-training vs. Fine-tuning Data

For both cohorts, we leverage the full dataset (excluding separate hyperparameter tuning and held-out sets) as our PT data $\boldsymbol{X}_{\mathrm{PT}}$. Naturally, this also means that our fine-tuning datasets $\boldsymbol{X}_{\mathrm{FT}}$ will overlap with our PT data. While this renders our benchmark less reflective of cases where one would like to deploy a PT model on a disjoint FT dataset, there are also many use-cases where these two datasets will overlap.

4.5.4 Pre-training Regimes

Our benchmark supports two styles of PT: self-supervised and weakly-supervised. Under self-supervision, only a single PT model is pre-trained, which is then used to assess FT performance directly on each downstream task (through separate FT runs, all transferring from the single PT model). Under weak-supervision, we permit the user to leverage a portion of our provided downstream tasks at pre-training time while still ensuring there is no task leakage from PT to FT via a "leave-one-task-out" (LOTO) framework. If our total set of downstream tasks is given by $\mathcal{T}$, then the LOTO framework requires pre-training a separate encoder $\mathcal{E}_t$ per downstream task $t \in \mathcal{T}$ such that $\mathcal{E}_t$ is transferred only to FT task $\mathcal{T}_{\text{FT}} = \{t\}$ for evaluation and leverages only tasks $\mathcal{T}_{\text{PT}} = \{t' \in \mathcal{T} | t' \neq t\}$ for PT weak supervision signals.

4.5.5 Fine-tuning Regimes

We analyzed two different styles of FT transfer: fine-tuning, decoder-only (FTD), and fine-tuning, full (FTF).

In the fine-tuning decoder-only (FTD) setting, the encoder $\mathcal{E}$ is frozen after PT, and only the decoder $\mathcal{D}$ is allowed to change during the FT stage. In the fine-tuning full (FTF) setting, the entire model, including the PT encoder $\mathcal{E}$ and the FT decoder $\mathcal{D}$ (which is not initialized during PT), can be updated during FT. This setting allows greater capacity, at the expense of a risk of over-fitting during FT. In addition, we naturally also encourage users to profile traditional, non-PT, single-task (ST) models of the same architectures over these tasks, to establish baseline performance levels.

4.5.6 Few-shot Analyses

In addition to comparing FTF vs. FTD performance, we also assess FT systems across various FT dataset sizes to judge models across a wide range of $N_{\text{PT}}/N_{\text{FT}}$ disparities.

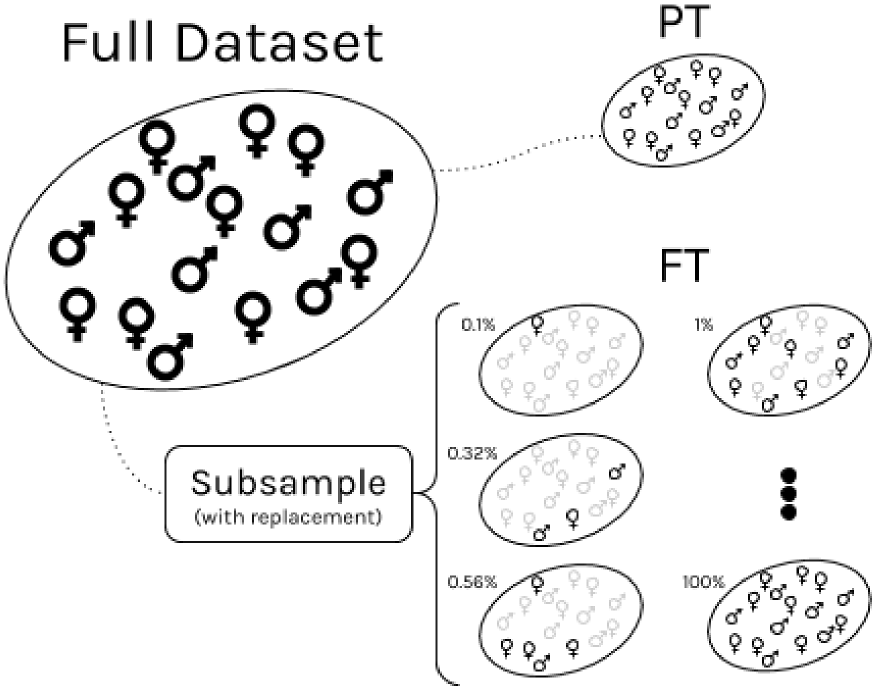


Figure 4-2: We always pre-train on the full available dataset, but additionally assess our models' ability to fine-tune in a few-shot context by randomly subsampling (with replacement) a variety of smaller FT datasets for each experiment.

These few-shot analysis datasets are formed by taking a series of random subsets (with replacement) of our overarching dataset $\boldsymbol{X}_{\mathrm{PT}}$ corresponding to 14 different sampling rates ranging on a logarithmic scale from 0.03% to 100%. This process is shown in Figure 4-2. Note that as all samples are taken randomly, our benchmark currently does not support PT/ FT in a setting with domain shift. This is obviously also an important challenge as well, that we hope to explore in future work.

4.5.7 Benchmark Utilization Protocol

First, the encoder must be pre-trained on the MIMIC-III and eICU cohorts. For a self-supervised PT system, hyperparameter tuning and pre-training are performed once (per random train/test split). For a weakly-supervised PT system, a separate round of pre-training must be performed per task t such that the pre-trained encoder $\mathcal{E}_t$ is trained to optimize task performance on all tasks *except* for task t, which is reserved for fine-tuning evaluation. To ease the hyperparameter tuning burden for weakly-supervised systems, it also is possible to perform a single round of PT hyperparameter tuning using the entire set of tasks, risking a small amount of task leakage at the gain of a significant reduction in compute cost (though of course actual PT must still be repeated for each model $\mathcal{E}_t$ with the proper subdivision of tasks after hyperparameter tuning is complete).

Next, fine-tuning is performed on task-specific models across all cohorts, sub-sampled datasets, and tasks. To assess the self-supervised system, all fine-tuning models will transfer from the same pre-trained source model, whereas for the weakly-supervised system, following LOTO, each encoder $\mathcal{E}_t$ must be fine-tuned on only task t to ensure no overlap between PT and FT tasks. This fine-tuning procedure is repeated across both the FTF and FTD transfer settings defined in Section 4.5.5.

Finally, fully-supervised, single-task (ST) models of the same base architecture are hyperparameter tuned and trained from scratch for each task to provide a baseline score.

The output of this process will yield one score per task, cohort, sub-sampled-dataset, PT algorithm, and FT transfer regime. This process is then repeated across the random splits within the benchmark to assess variance. Based on these results, the user can judge if either of these PT algorithms offer robust benefits across all cohorts and tasks, if one fine-tuning transfer style is preferred over another, or any number of other questions.

4.6 Baseline Experiments

4.6.1 Baseline-specific Data Post-processing

Our baseline models featurize the timeseries into fixed-size input windows of anywhere from 12–96 hours (chosen via hyperparameter search). Within these fixed *input* windows (and not taking into account any data from the prediction windows), any missing features are linearly interpolated between their previous and subsequent measurements. If a measurement is only observed on one side of the value (*e.g.*, there are no future measurements or no previous measurements within the input window), values are carried forward or backward, respectively, and if no measurements are observed, they are imputed to the feature's mean value over the train dataset. In addition, time-since-last-measured ordinal indicators (up to 8 hours) are added to capture how long it has been at any given time-point since a specific feature was last measured. Both to simplify our shared code base, and as a form of data augmentation, all training is done across random time-points throughout the patient's stay, regardless of the specific details of the task's prescribed evaluation input window, though those relationships are, of course, respected during evaluation. For example, though ICD code prediction will only be evaluated using the first 24 hours of a patient's stay, during training we will train this model on inputs throughout the patient's stay.

4.6.2 Models

Encoder Architecture

All models in this work use a GRU model [32] as their encoder $\mathcal{E}$. Early experiments suggested this model outperformed other architectures, including a simpler, linear baseline, a convolutional neural network architecture, and a transformer model, and it is a commonly used model in the literature, so it is a reasonable choice for a baseline architecture here. Input data is projected down to a unified numerical representation, then run through a (potentially) multi-layer, bidirectional GRU (GRU parameters are determined via hyperparameter tuning) to yield a final encoder. This encoded representation is then passed through a task-specific decoder, which is either (1) a LSTM based sequential decoder for the FTS task, or (2) a simple linear layer up to the appropriate dimensionality of the task, followed by an appropriate classification activation (*e.g.*, sigmoid or softmax) for all other tasks. For multi-label tasks, activations and losses are computed in a per-label manner and losses are then averaged.

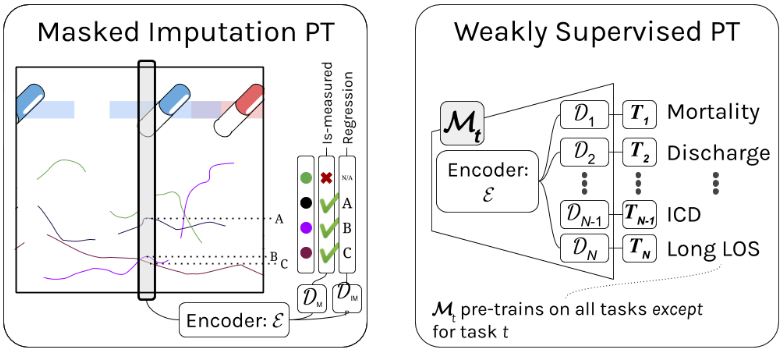


Figure 4-3: We profile both a self-supervised, masked-imputation PT system and a weakly-supervised multi-task PT system.

Supervised, Single-task (ST) Models

We perform fully supervised, single-task (ST) training, with no PT, on each task separately, to provide baselines in comparisons with our PT/FT methods. These runs use the same GRU architecture as our other experiments, and are hyperparameter tuned separately for each downstream task.

Pre-training Algorithms

We profile two distinct PT systems on our benchmark: A weakly-supervised, multi-task (MT) PT model, and a self-supervised, masked-imputation (MI) model. For a visual overview of both of these methods, see Figure 4-3.

Weakly-supervised, Multi-task (MT) Pre-training In multi-task (MT) PT, we use the "leave-one-task-out" method described in Section 4.5.4 to ensure our MT PT approach does not leak task information between FT and PT contexts. Our multi-task approach is very straightforward: all tasks in the learning ensemble (*e.g.*, all tasks save the eventual fine-tuning target) will be jointly trained via a model whose encoder $\mathcal{E}$ is shared across all tasks but with separate decoders $\mathcal{D}_t$ per task. No loss weighting or task-alignment is used.

Masked-Imputation (MI) Pre-training Masked-imputation (MI) PT is inspired by the successes of models such as BERT [46] in NLP. To adapt the ideas of BERT to a continuous domain with missingness, we choose at random approximately 15% of the time-points in the input window to "mask," (replace with all zeros and augment with a bit indicating masking took place). Then, the model is tasked with predicting within this masked time-point which labs/vitals were actually measured via a classification task and what their values were via a continuous regression task. At fine-tuning time, the model is no longer asked to perform masked imputation, and no masking is applied. For this PT task, we limit our GRU models to unidirectional GRUs to avoid leaking information from future time-points[2], and models are hyperparameter tuned to maximize the mean of the classification task's macro AUROC and the regression task's R^2 value.

Fine-tuning For both PT systems, after PT is complete, the model is fine-tuned by initializing an untrained decoder layer and training the system according to the loss criteria appropriate to the type of task at hand (*e.g.*, binary cross-entropy loss or a negative log likelihood loss depending on the task type). Tasks that are multi-label in nature are trained by averaging the losses together for all labels. As described in Section 4.5.5, we profile both FTF and FTD transfer styles in this baseline, and we report across all sub-sampled dataset sizes as described in Section 4.5.6.

4.6.3 Hyperparameter Tuning

Hyperparameter tuning was performed to optimize the underlying architecture via the PT task with random search, via the Bayesian Hyperopt Library [17]. No FT specific hyperparameter tuning was performed, as the majority of the details of the architecture (*e.g.*, the GRU depth and dimensionality) are fixed by the pre-training algorithm. ST models were naturally tuned based on output task performance, as there is no PT stage for these models.

Specific model hyperparameters were chosen to maximize the appropriate score on the validation fold over a random search drawn from a customized hyperparam-

[2] This is especially important in the context of our imputation procedure, which directly encodes how long it has been since any given lab/vital was measured.

eter distribution. For MT PT models, this search was done once across all tasks simultaneously in a MT manner, optimizing for the average AUROC across all tasks in the ensemble, then used for all PT/FT experiments. This represents a possible source of very mild task leakage, but yielded significant computational savings. For MI models, hyperparameter tuning was done once in a task-independent manner (as masked imputation is a self-supervised, rather than weakly-supervised PT method). Final hyperparameters were chosen based on the full dataset, and were not repeated at smaller training set sizes for the few-shot experiments, which may represent another possible source of bias. Additional details on the hyperparameter search can be found in Appendix Section 4.9.

4.6.4 Results

In this section, we will highlight a subset of the most relevant results found in our baseline experiments. Figure 4-4 shows two things of interest: First, it shows a graph cataloging over what fraction of tasks a particular PT/FT model type offers best performance as a function of dataset size. This allows us to see quickly, for example, that for a wide variety of dataset sizes, MT FTF offers significnat improvements over other strategies on a significant portion of the tested tasks. To show these relationships in more detail, Figure 4-4 also shows more complete results for 3 of our 10 tasks across both cohorts and all dataset fractions, comparing specifically both varieties of the FTF models and the ST model. In addition, the results corresponding to the 1%, 10%, and full-data scales for both cohorts are shown in Table 4.3. In this view, we see that at the 1% setting, the Multi-task (MT) FTF setting performs best in 7/10 settings, whereas ST never offers best-in-class performance. At the 10% setting, MT FTF excels 6/10 times, and ST performs best in only one task. Finally, at the 100% (*e.g.*, full) data scale, ST always performs best. This demonstrates a strong trend between the performance benefit offered by MT-FTF PT and the severity of the N_{FT} v. N_{PT} imbalance. Full results can be found in the Appendix, in Figures 4-6,4-7 for MIMIC-III and 4-8,4-9 for eICU.

Dataset Size	Task	MIMIC-III					eICU				
		MI FTD	MI FTF	MT FTD	MT FTF	ST	MI FTD	MI FTF	MT FTD	MT FTF	ST
Few-shot (1%)	MOR	0.55 ± 0.08	0.68 ± 0.08	0.57 ± 0.17	**0.84 ± 0.11**	0.64 ± 0.06	0.57 ± 0.04	0.7 ± 0.06	0.52 ± 0.14	**0.8 ± 0.02**	0.6 ± 0.06
	CMO	0.6 ± 0.09	0.65 ± 0.05	0.52 ± 0.17	**0.76 ± 0.18**	0.58 ± 0.11					
	DNR	0.59 ± 0.04	0.57 ± 0.06	0.55 ± 0.08	**0.63 ± 0.1**	0.55 ± 0.07					
	ICD	0.49 ± 0.03	**0.56 ± 0.03**	0.52 ± 0.02	0.56 ± 0.03	0.56 ± 0.02					
	LOS	0.51 ± 0.09	0.62 ± 0.03	0.6 ± 0.08	**0.67 ± 0.03**	0.58 ± 0.02	0.51 ± 0.02	0.55 ± 0.02	0.53 ± 0.06	**0.59 ± 0.02**	0.54 ± 0.04
	REA	**0.54 ± 0.03**	0.51 ± 0.02	0.5 ± 0.04	0.54 ± 0.03	0.51 ± 0.03					
	DIS	0.52 ± 0.02	0.57 ± 0.05	0.54 ± 0.03	**0.58 ± 0.03**	0.54 ± 0.01	0.51 ± 0.01	0.56 ± 0.01	0.53 ± 0.03	**0.58 ± 0.01**	0.56 ± 0.01
	ACU	0.51 ± 0.03	0.61 ± 0.04	0.56 ± 0.05	**0.61 ± 0.01**	0.6 ± 0.05	0.51 ± 0.01	0.58 ± 0.02	0.55 ± 0.05	**0.62 ± 0.02**	0.58 ± 0.02
	WBM	0.53 ± 0.03	0.61 ± 0.03	0.53 ± 0.03	**0.64 ± 0.08**	0.58 ± 0.02	0.59 ± 0.04	0.77 ± 0.05	0.53 ± 0.02	**0.8 ± 0.05**	0.65 ± 0.09
	FTS	0.61 ± 0.05	0.61 ± 0.04	**0.62 ± 0.05**	0.62 ± 0.06	0.6 ± 0.04					
Few-shot (10%)	MOR	0.61 ± 0.12	0.82 ± 0.02	0.82 ± 0.17	**0.9 ± 0.03**	0.84 ± 0.07	0.62 ± 0.09	0.8 ± 0.03	0.77 ± 0.07	**0.8 ± 0.03**	0.77 ± 0.03
	CMO	0.62 ± 0.08	0.74 ± 0.03	0.76 ± 0.15	**0.85 ± 0.06**	0.77 ± 0.09					
	DNR	0.6 ± 0.04	0.75 ± 0.03	0.76 ± 0.09	**0.82 ± 0.03**	0.71 ± 0.1					
	ICD	0.53 ± 0.05	0.65 ± 0.01	0.6 ± 0.02	0.64 ± 0.01	**0.67 ± 0.03**					
	LOS	0.55 ± 0.09	0.6 ± 0.02	**0.69 ± 0.02**	0.65 ± 0.03	0.66 ± 0.02	0.52 ± 0.03	0.6 ± 0.03	0.61 ± 0.02	**0.61 ± 0.02**	0.61 ± 0.0
	REA	0.54 ± 0.04	0.53 ± 0.03	0.54 ± 0.05	**0.57 ± 0.04**	0.57 ± 0.03					
	DIS	0.56 ± 0.03	0.63 ± 0.02	**0.67 ± 0.04**	0.66 ± 0.04	0.64 ± 0.03	0.53 ± 0.02	0.6 ± 0.01	0.61 ± 0.04	**0.62 ± 0.01**	0.61 ± 0.01
	ACU	0.58 ± 0.06	0.69 ± 0.03	0.68 ± 0.04	**0.7 ± 0.04**	0.69 ± 0.03	0.56 ± 0.03	0.63 ± 0.02	**0.67 ± 0.03**	0.64 ± 0.02	0.64 ± 0.01
	WBM	0.57 ± 0.04	0.76 ± 0.01	0.65 ± 0.06	**0.79 ± 0.05**	0.72 ± 0.04	0.72 ± 0.03	0.87 ± 0.01	0.66 ± 0.05	**0.88 ± 0.02**	0.82 ± 0.05
	FTS	0.74 ± 0.09	0.77 ± 0.08	**0.82 ± 0.05**	0.81 ± 0.05	0.73 ± 0.06					
Full Data	MOR	0.74 ± 0.09	0.8 ± 0.11	0.94 ± 0.01	0.89 ± 0.03	**0.95 ± 0.01**	0.72 ± 0.07	0.83 ± 0.01	**0.86 ± 0.01**	0.82 ± 0.02	0.85 ± 0.01
	CMO	0.72 ± 0.06	0.77 ± 0.05	**0.92 ± 0.01**	0.85 ± 0.04	0.91 ± 0.02					
	DNR	0.72 ± 0.09	0.74 ± 0.01	**0.87 ± 0.02**	0.78 ± 0.06	0.87 ± 0.02					
	ICD	0.65 ± 0.05	0.67 ± 0.01	0.67 ± 0.01	0.68 ± 0.03	**0.74 ± 0.01**					
	LOS	0.61 ± 0.04	0.58 ± 0.03	0.69 ± 0.02	0.64 ± 0.04	**0.71 ± 0.01**	0.54 ± 0.04	0.64 ± 0.02	0.62 ± 0.02	0.63 ± 0.05	**0.65 ± 0.0**
	REA	0.57 ± 0.04	0.56 ± 0.02	0.6 ± 0.04	0.57 ± 0.02	**0.61 ± 0.02**					
	DIS	0.64 ± 0.05	0.68 ± 0.04	**0.75 ± 0.02**	0.72 ± 0.02	0.74 ± 0.03	0.58 ± 0.03	0.64 ± 0.01	**0.66 ± 0.02**	0.64 ± 0.01	0.65 ± 0.0
	ACU	0.7 ± 0.05	0.74 ± 0.02	0.75 ± 0.04	0.74 ± 0.04	**0.78 ± 0.02**	0.62 ± 0.02	0.66 ± 0.01	**0.7 ± 0.02**	0.67 ± 0.02	0.68 ± 0.02
	WBM	0.68 ± 0.07	0.81 ± 0.01	0.77 ± 0.02	0.86 ± 0.02	**0.89 ± 0.01**	0.79 ± 0.04	**0.91 ± 0.01**	0.79 ± 0.02	0.91 ± 0.01	0.9 ± 0.02
	FTS	0.86 ± 0.02	0.89 ± 0.01	0.89 ± 0.01	**0.9 ± 0.0**	0.9 ± 0.01					

Table 4.3: GRU Results (AUROC) subdivided among different PT regimes, under both the full-data fine-tuning setting and few-shot (1%, 10%) settings, on both the MIMIC-III and eICU cohorts. Bolded results indicate top performing result per each task/evaluation setting.

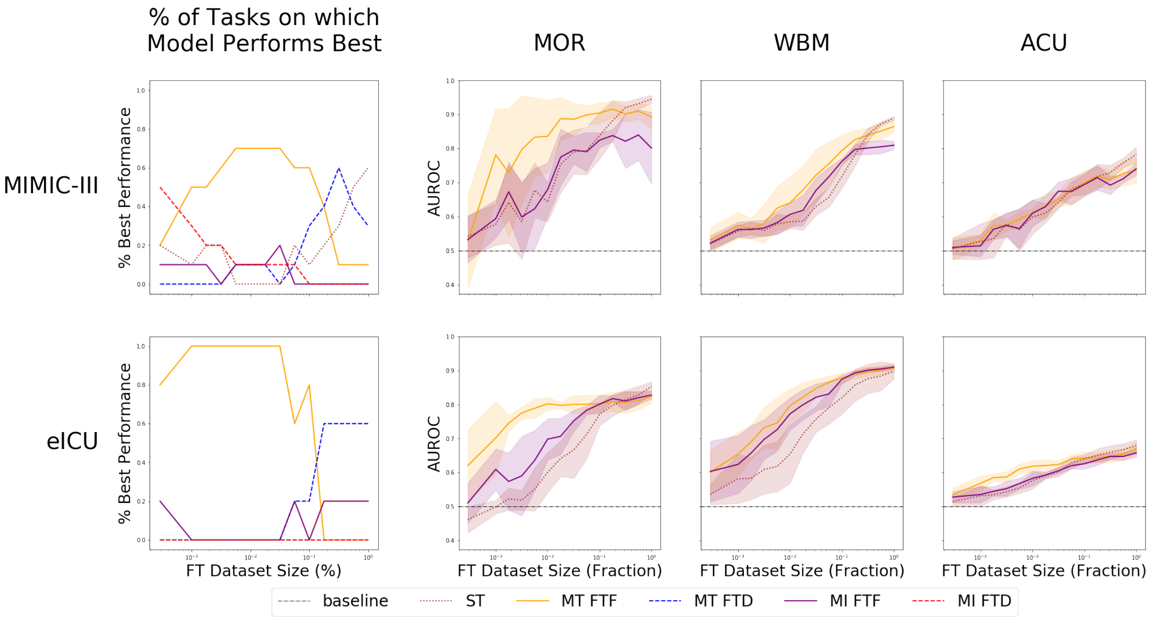


Figure 4-4: *(left column)* For what % of tasks (y-axis) does a given PT/FT regime (linestyle) perform better than all other PT/FT regimes, as a function of dataset fraction (x-axis). *(right 3 columns)* Performance in macro-averaged AUROC (y-axis) of various PT/FT models (linestyle) across various FT dataset sub-sampling rates (x-axis), over 3 sample FT tasks (subplots).

4.6.5 Discussion

Pre-training does not offer benefits at full data scale Table 4.3 shows that PT does not offer any gains over traditional supervised learning at full MIMIC or eICU dataset scales. In some cases, PT actually harms the final results. This is not necessarily surprising; while PT can help compensate for too little data and provide (indirect) access to additional data in the case of larger N_{FT} / N_{PT} discrepancies, when $\boldsymbol{X}_{\text{FT}} = \boldsymbol{X}_{\text{PT}}$ the risks that PT simply serves as a distraction from the (comparatively direct) supervised learning signal is large.

Pre-training can offer significant benefits in few-shot settings Our benchmark reveals that PT in few-shot settings is helpful. Figure 4-4 shows that across a significant fraction of the dataset fractions for both the MIMIC-III and eICU cohorts, MT FTF offers significant benefits over other approaches. In Table 4.3, we see more concretely that in the 1% FT dataset setting, some form of PT/FT offers best in class performance across all tasks in both cohorts except for LOS on the eICU cohort, with performance improvements ranging as high as an AUROC improvement of 0.2/0.24 for mortality prediction in MIMIC-III/eICU. In the 10% dataset size setting, some form of PT/FT still offers best-in-class performance on all tasks save LOS for the eICU cohort and ICD for the MIMIC cohort. The margins of improvement are no longer as high, but offer consistent gains across a variety of other tasks such as with CMO, DNR, MOR, WBM, and FTS all offering AUROC improvements of up to 0.1 on MIMIC-III (improvements are much smaller on eICU at this threshold). These findings provide evidence to affirm that EHR PT/FT strategies could enable more effective modelling even given only very small task-specific datasets, thus potentially offering a vehicle to help train models for novel or rare diseases.

We also observe that the tasks which tend to show the largest improvements with PT (MOR, WBM, CMO, DNR, and FTS) are all *rolling tasks* with substantial class imbalance. Across both cohorts these tasks report majority class accuracies greater than or equal to 88% (with most $\geq$ 95%). No other tasks in our benchmark meet these criteria, suggesting there may be stronger benefits from PT (in particular, from MT PT) on rolling, imbalanced tasks.

Weakly-supervised pre-training out-performs self-supervised pre-training In general, MT PT is superior to MI, even at small data scales, suggesting that simple adoption of the masked language modeling idea is not sufficient for the clinical domain. Despite this, in the few-shot domain, MI FTF training still does outperform traditional ST modelling, just not by as much as MT FTF does. For example, at 1% on MIMIC-III, MI FTF outperforms ST in all but one case, and at 10% it does in all but four cases (though on eICU the situation is murkier).

FTF is in general preferred over FTD Consistent across both MI and MT PT is that FTF models are preferred to FTD models. This is true across datasets and sub-sampling rate, and suggests that despite the increased risk of overfitting offered by fine-tuning the encoder as well as the decoder, this strategy may be integral to obtaining strong PT/FT results in this modality.

4.7 Conclusion

In this work, we present a novel benchmark for PT systems over EHR time-series data. We define a suite of FT task targets, including several novel tasks, over both MIMIC-III and eICU, and establish evaluation procedures for examining a PT system's performance both across various FT dataset sizes. We then establish three baseline systems on this benchmark, including a traditional, non-pre-trained single-task baseline, a weakly-supervised multi-task PT baseline, and a fully self-supervised masked-imputation based PT baseline. These baselines demonstrate that weakly-supervised, multi-task PT can offer substantial improvements in few-shot contexts for tasks suffering from significant class imbalance. In addition, they suggest important findings on the viability of different styles of PT and FT; in particular that masked-imputation based PT currently is not competitive with multi-task PT, and that fine-tuning both model encoders and decoders is necessary for ensuring strong FT performance.

While significant future work remains, including assessing additional PT systems on this benchmark as well as augmenting this benchmark to assess the impact PT has on fine-tuning under domain shift such as pre-training on one hospital and fine-tuning on another, or subpopulation shift in fairness applications, we believe that this benchmark can be an invaluable tool for the ML4H community. By standardizing PT/FT training and evaluation procedures, including few-shot evaluation analyses and the inclusion of a sufficiently diverse set of tasks to assess the utility of PT schemes in general, rather than merely on a isolated, highly specific subset of tasks, this benchmark offers the possibility of greatly increasing the efficiency of PT research on EHR data. This benchmark will help enable iterative analysis and development of PT strategies in this domain and lead to the release of PT encoders that enable easy specialization and deployment in clinical settings.

4.7.1 Relation to this Thesis

In this chapter, we focus our investigations to the question of pre-training (PT) specifically, and show that naïve adaptations of existing PT strategies to clinical data will not yield significant performance improvements over existing methods. In particular, we can surpass these traditional, imputation-based pre-training approaches with even just a simple, multi-task PT baseline. This work thus directly motivates not only the development of new PT methods, but specifically PT methods that allow us to better capture whole-sample signals such as those captured by our multi-task tasks here. Such methodological development is exactly the focus of Chapter 5, next.

4.8 Appendix

4.8.1 Additional Data/Task Information

Additional Dataset Details

MIMIC-III Cohort Treatment Data In the MIMIC-III cohort, we incorporate as inputs treatments including adenosine, colloid bolus, crystalloid bolus, dobutamine, dopamine, epinephrine, isuprel, milrinone, nivdurations, norepinephrine, phenylephrine, vaso (other vasopressor application), vasopressin, and vent (ventilation).

Task Details and Statistics

Imminent Discharge: DIS The below two tables (Table 4.9, 4.10) capture the overall prevalence of all DIS classes observed across both cohorts and all labels.

Final Acuity: ACU The below two tables (Table 4.12, 4.11) capture the overall prevalence of all ACU classes observed across both cohorts.

Next Timepoint: WBM Tables 4.6 & 4.7 shows the majority class accuracy for all labs & vitals used in this work for the WBM task.

Task	MIMIC-III			eICU		
	Train	Tuning	Held-out Test	Train	Tuning	Held-out Test
MOR	96.5 ± 14.14%	96.5 ± 14.23%	96.3 ± 14.67%	97.0 ± 13.67%	97.1 ± 13.58%	97.1 ± 13.60%
CMO	98.8 ± 7.83%	98.8 ± 7.84%	98.7 ± 8.03%			
DNR	96.3 ± 17.12%	96.3 ± 17.13%	96.4 ± 17.13%			
WBM	92.1 ± 9.18%	92.2 ± 9.19%	92.1 ± 9.21%	88.0 ± 19.61%	88.0 ± 19.63%	88.0 ± 19.65%
DIS	42.6 ± 25.65%	42.6 ± 25.53%	42.6 ± 25.71%	44.2 ± 31.65%	44.1 ± 31.72%	44.1 ± 31.69%
ICD	67.0 ± 18.07%	70.2 ± 18.12%	70.2 ± 18.19%			
LOS	52.9 ± 0.11%	53.0 ± 1.26%	52.8 ± 1.02%	66.6 ± 0.04%	66.5 ± 0.26%	66.6 ± 0.32%
REA	95.1 ± 0.09%	95.1 ± 0.62%	95.0 ± 0.61%			
ACU	25.3 ± 0.24%	25.2 ± 1.34%	25.3 ± 1.15%	59.7 ± 0.10%	59.2 ± 0.39%	59.4 ± 0.64%
FTS	97.4 ± 2.77%	97.5 ± 2.91%	97.4 ± 2.96%	97.0 ± 3.75%	97.0 ± 3.75%	97.0 ± 3.75%

Table 4.4: Macro-averaged (train-set) majority class accuracy aggregated across all folds / labels for all tasks.

Task	Label	Majority Class	MIMIC-III			eICU		
			Train	Tuning	Held-out Test	Train	Tuning	Held-out Test
MOR	24H	0	97.6 ± 9.91%	97.6 ± 10.07%	97.5 ± 10.53%	97.9 ± 10.20%	97.9 ± 10.09%	97.9 ± 10.06%
	48H	0	95.4 ± 17.36%	95.4 ± 17.42%	95.2 ± 17.88%	96.2 ± 16.41%	96.2 ± 16.34%	96.2 ± 16.39%
CMO	24H	0	99.1 ± 5.55%	99.1 ± 5.58%	99.1 ± 5.77%			
	48H	0	98.5 ± 9.59%	98.4 ± 9.58%	98.4 ± 9.77%			
DNR	24H	0	96.6 ± 16.16%	96.6 ± 16.18%	96.6 ± 16.22%			
	48H	0	96.1 ± 18.03%	96.1 ± 18.03%	96.1 ± 18.00%			
DIS	24H	No Discharge	57.7 ± 24.68%	57.7 ± 24.57%	57.7 ± 24.79%	48.2 ± 26.67%	48.2 ± 26.73%	48.2 ± 26.71%
	48H	Home				40.2 ± 35.95%	39.9 ± 36.03%	40.0 ± 35.98%
		No Discharge	27.6 ± 26.58%	27.5 ± 26.45%	27.6 ± 26.60%			
LOS		0	52.9 ± 0.11%	53.0 ± 1.26%	52.8 ± 1.02%	66.6 ± 0.04%	66.5 ± 0.26%	66.6 ± 0.32%
REA		0	95.1 ± 0.09%	95.1 ± 0.62%	95.0 ± 0.61%			
ACU		Home				59.7 ± 0.10%	59.2 ± 0.39%	59.4 ± 0.64%
		Home Health Care	25.3 ± 0.24%	25.2 ± 1.34%	25.3 ± 1.15%			
FTS		0	97.4 ± 2.77%	97.5 ± 2.91%	97.4 ± 2.96%	97.0 ± 3.75%	97.0 ± 3.75%	97.0 ± 3.75%

Table 4.5: Per-label majority class accuracies for all tasks aside from WBM and LOS, which are shown separately.

Task	Label	Majority Class	MIMIC-III			eICU		
			Train	Tuning	Held-out Test	Train	Tuning	Held-out Test
WBM	Anion Gap	0	$91.4 \pm 4.95\%$	$91.5 \pm 4.87\%$	$91.5 \pm 4.96\%$			
	Bedside Glucose	0				$86.0 \pm 16.68\%$	$85.9 \pm 16.75\%$	$85.9 \pm 16.77\%$
	Bicarbonate	0	$91.1 \pm 4.88\%$	$91.1 \pm 4.81\%$	$91.1 \pm 4.90\%$			
	Blood Urea Nitrogen	0	$91.0 \pm 4.87\%$	$91.0 \pm 4.78\%$	$91.0 \pm 4.92\%$			
	Bun	0				$94.8 \pm 3.29\%$	$94.8 \pm 3.29\%$	$94.8 \pm 3.32\%$
	Calcium	0	$92.8 \pm 5.05\%$	$92.8 \pm 4.94\%$	$92.8 \pm 5.05\%$			
	Calcium Ionized	0	$95.2 \pm 7.01\%$	$95.3 \pm 6.87\%$	$95.2 \pm 7.00\%$			
	Cardiac Index	0	$96.7 \pm 10.73\%$	$96.6 \pm 11.02\%$	$96.7 \pm 10.84\%$			
	Cardiac Output Thermodilution	0	$97.2 \pm 9.92\%$	$97.1 \pm 10.23\%$	$97.2 \pm 9.99\%$			
	Central Venous Pressure	0	$82.2 \pm 26.70\%$	$82.1 \pm 26.67\%$	$82.0 \pm 26.75\%$			
	Chloride	0	$90.3 \pm 5.68\%$	$90.4 \pm 5.55\%$	$90.3 \pm 5.66\%$			
	Co2	0	$96.5 \pm 3.52\%$	$96.4 \pm 3.54\%$	$96.4 \pm 3.53\%$			
	Co2 (Etco2, Pco2, Etc.)	0	$92.3 \pm 9.05\%$	$92.4 \pm 8.89\%$	$92.3 \pm 8.97\%$			
	Creatinine	0	$90.9 \pm 4.91\%$	$91.0 \pm 4.81\%$	$91.0 \pm 4.95\%$	$94.8 \pm 3.30\%$	$94.8 \pm 3.30\%$	$94.8 \pm 3.33\%$
	Diastolic Blood Pressure	1	$88.0 \pm 12.70\%$	$88.1 \pm 12.61\%$	$88.0 \pm 12.77\%$			
	Fraction Inspired Oxygen	0	$95.9 \pm 8.12\%$	$96.0 \pm 8.12\%$	$96.0 \pm 8.15\%$			
	Fraction Inspired Oxygen Set	0	$94.1 \pm 9.67\%$	$94.1 \pm 9.73\%$	$94.0 \pm 9.79\%$			
	Glascow Coma Scale Total	0	$82.4 \pm 17.85\%$	$82.0 \pm 18.10\%$	$82.1 \pm 17.82\%$			
	Glucose	0	$77.0 \pm 15.35\%$	$77.0 \pm 15.38\%$	$77.2 \pm 15.20\%$	$94.7 \pm 3.62\%$	$94.7 \pm 3.61\%$	$94.7 \pm 3.67\%$
	Hct	0				$94.7 \pm 3.37\%$	$94.7 \pm 3.37\%$	$94.7 \pm 3.41\%$
	Heart Rate	0				$80.3 \pm 29.21\%$	$80.3 \pm 29.29\%$	$80.3 \pm 29.27\%$
		1	$91.1 \pm 11.41\%$	$91.2 \pm 11.44\%$	$91.0 \pm 11.61\%$			
	Hematocrit	0	$88.2 \pm 6.81\%$	$88.3 \pm 6.63\%$	$88.3 \pm 6.66\%$			
	Hemoglobin	0	$90.6 \pm 4.89\%$	$90.7 \pm 4.70\%$	$90.7 \pm 4.80\%$			
	Lactate	0	$97.2 \pm 3.97\%$	$97.3 \pm 3.88\%$	$97.3 \pm 3.92\%$			
	Lactic Acid	0	$97.6 \pm 4.04\%$	$97.7 \pm 4.00\%$	$97.7 \pm 3.99\%$			
	Magnesium	0	$91.5 \pm 4.79\%$	$91.6 \pm 4.67\%$	$91.6 \pm 4.78\%$			
	Mean Blood Pressure	1	$87.5 \pm 13.31\%$	$87.7 \pm 13.21\%$	$87.5 \pm 13.43\%$			
	Mean Corpuscular Hemoglobin	0	$93.5 \pm 2.44\%$	$93.6 \pm 2.40\%$	$93.6 \pm 2.47\%$			
	Mean Corpuscular Hemoglobin Concentration	0	$93.5 \pm 2.44\%$	$93.6 \pm 2.40\%$	$93.6 \pm 2.47\%$			
	Mean Corpuscular Volume	0	$93.5 \pm 2.44\%$	$93.6 \pm 2.40\%$	$93.6 \pm 2.47\%$			
	Noninvasive Diastolic	1				$79.7 \pm 24.12\%$	$79.7 \pm 24.16\%$	$79.6 \pm 24.21\%$
	Noninvasive Mean	1				$79.8 \pm 24.12\%$	$79.8 \pm 24.17\%$	$79.8 \pm 24.22\%$
	Noninvasive Systolic	1				$79.7 \pm 24.12\%$	$79.7 \pm 24.16\%$	$79.6 \pm 24.21\%$
	Oxygen Saturation	1	$86.8 \pm 15.09\%$	$86.9 \pm 14.99\%$	$86.7 \pm 15.18\%$			

Table 4.6: Per-label majority class accuracies for the WBM task (continued in 4.7)

Task	Label	Majority Class	MIMIC-III			eICU		
			Train	Tuning	Held-out Test	Train	Tuning	Held-out Test
	Partial Pressure Of Carbon Dioxide	0	92.3 ± 9.05%	92.4 ± 8.89%	92.3 ± 8.97%			
	Partial Pressure Of Oxygen	0	96.0 ± 6.79%	96.1 ± 6.63%	96.0 ± 6.76%			
	Partial Thromboplastin Time	0	93.7 ± 5.11%	93.8 ± 4.96%	93.8 ± 4.98%			
	Peak Inspiratory Pressure	0	95.4 ± 6.80%	95.3 ± 6.90%	95.3 ± 6.85%			
	Ph	0	91.4 ± 9.74%	91.5 ± 9.62%	91.4 ± 9.67%			
	Phosphate	0	94.4 ± 3.11%	94.4 ± 3.07%	94.4 ± 3.15%			
	Phosphorous	0	94.6 ± 3.30%	94.6 ± 3.29%	94.6 ± 3.30%			
	Plateau Pressure	0	97.2 ± 4.70%	97.1 ± 4.78%	97.1 ± 4.75%			
	Platelets	0	91.4 ± 4.50%	91.5 ± 4.34%	91.5 ± 4.42%			
	Positive End-Expiratory Pressure Set	0	93.5 ± 8.45%	93.4 ± 8.55%	93.4 ± 8.49%			
	Potassium	0	89.4 ± 6.15%	89.4 ± 6.06%	89.4 ± 6.20%	94.8 ± 4.66%	94.8 ± 4.67%	94.8 ± 4.71%
	Potassium Serum	0	96.7 ± 4.12%	96.8 ± 4.08%	96.8 ± 4.13%			
	Prothrombin Time Inr	0	94.0 ± 4.67%	94.1 ± 4.53%	94.1 ± 4.60%			
	Prothrombin Time Pt	0	94.0 ± 4.67%	94.1 ± 4.53%	94.1 ± 4.60%			
	Pulmonary Artery Pressure Mean	0	97.2 ± 12.56%	97.0 ± 13.02%	97.1 ± 12.81%			
	Pulmonary Artery Pressure Systolic	0	91.3 ± 19.70%	91.0 ± 19.93%	91.2 ± 19.77%			
	Red Blood Cell Count	0	93.5 ± 2.44%	93.6 ± 2.41%	93.5 ± 2.47%			
	Respiratory Rate	0				82.1 ± 28.33%	82.0 ± 28.43%	82.1 ± 28.38%
		1	89.6 ± 13.74%	89.7 ± 13.76%	89.5 ± 14.04%			
	Respiratory Rate Set	0	95.8 ± 6.52%	95.8 ± 6.60%	95.7 ± 6.57%			
	Sao2	0				81.6 ± 28.19%	81.6 ± 28.25%	81.6 ± 28.27%
	Sodium	0	89.7 ± 5.89%	89.9 ± 5.74%	89.8 ± 5.87%			
	St1	0				92.2 ± 20.06%	92.3 ± 19.98%	92.2 ± 20.03%
	St2	0				91.8 ± 20.66%	91.9 ± 20.59%	91.8 ± 20.67%
	St3	0				92.4 ± 19.86%	92.4 ± 19.76%	92.4 ± 19.83%
	Systemic Vascular Resistance	0	96.8 ± 10.43%	96.7 ± 10.68%	96.8 ± 10.55%			
	Systolic Blood Pressure	1	88.0 ± 12.69%	88.1 ± 12.60%	88.0 ± 12.77%			
	Temperature	0	70.6 ± 13.32%	70.5 ± 13.47%	70.7 ± 13.33%			
	Tidal Volume Observed	0	94.3 ± 8.13%	94.3 ± 8.18%	94.3 ± 8.15%			
	Tidal Volume Set	0	96.1 ± 6.13%	96.0 ± 6.20%	96.0 ± 6.16%			
	Tidal Volume Spontaneous	0	97.3 ± 4.55%	97.3 ± 4.59%	97.2 ± 4.59%			
	Weight	0	97.3 ± 2.76%	97.3 ± 2.58%	97.3 ± 2.67%			
	White Blood Cell Count	0	91.7 ± 4.27%	91.8 ± 4.12%	91.8 ± 4.24%			

Table 4.7: Per-label majority class accuracies for the WBM task (continued from 4.6)

Task	Label	Class	Train	Tuning	Held-out Test
ICD	Blood	0	$52.5 \pm 0.16\%$	$54.3 \pm 0.77\%$	$54.7 \pm 0.48\%$
	Circulatory	1	$72.0 \pm 0.14\%$	$78.8 \pm 1.45\%$	$78.7 \pm 1.14\%$
	Congenital	0	$91.4 \pm 0.15\%$	$94.8 \pm 0.51\%$	$95.1 \pm 0.56\%$
	Defined	0	$50.8 \pm 0.41\%$	$53.3 \pm 1.30\%$	$53.6 \pm 1.17\%$
	Digestive	0	$51.2 \pm 0.35\%$	$54.0 \pm 1.34\%$	$54.0 \pm 1.34\%$
	Endocrine	1	$63.5 \pm 0.27\%$	$67.6 \pm 1.30\%$	$67.1 \pm 1.18\%$
	Genitourinary	0	$51.9 \pm 0.10\%$	$52.8 \pm 1.77\%$	$52.5 \pm 1.26\%$
	Infection	0	$58.0 \pm 0.24\%$	$64.4 \pm 0.70\%$	$65.5 \pm 1.61\%$
	Injury	0	$50.1 \pm 0.04\%$	$51.8 \pm 0.10\%$	$51.1 \pm 0.63\%$
		1	$50.2 \pm 0.13\%$	$48.8 \pm 1.16\%$	$48.8 \pm 1.05\%$
	Mental	0	$57.0 \pm 0.19\%$	$61.4 \pm 0.88\%$	$61.1 \pm 0.80\%$
	Musculoskeletal	0	$66.7 \pm 0.08\%$	$73.5 \pm 0.28\%$	$73.8 \pm 0.22\%$
	Neoplasms	0	$70.3 \pm 0.32\%$	$76.1 \pm 1.03\%$	$75.7 \pm 0.32\%$
	Nervous	0	$59.3 \pm 0.16\%$	$63.9 \pm 0.62\%$	$64.0 \pm 0.41\%$
	Perinatal	0	$100.0 \pm 0.01\%$	$100.0 \pm 0.03\%$	$100.0 \pm 0.03\%$
	Pregnancy	0	$98.8 \pm 0.04\%$	$99.3 \pm 0.30\%$	$99.5 \pm 0.24\%$
	Respiratory	1	$53.3 \pm 0.15\%$	$53.6 \pm 0.87\%$	$53.3 \pm 0.81\%$
	Skin	0	$76.7 \pm 0.25\%$	$85.1 \pm 1.50\%$	$85.3 \pm 1.13\%$
	Unknown	0	$100.0 \pm 0.01\%$	$100.0 \pm 0.03\%$	$100.0 \pm 0.03\%$

Table 4.8: ICD task per-label majority class accuracies. As the ICD task is defined only on MIMIC-III, this table is specific to that cohort.

Class	24H			48H		
	Train	Tuning	Held-out Test	Train	Tuning	Held-out Test
Long Term Care Hospital	1.0 ± 6.07%	1.0 ± 5.96%	1.0 ± 5.80%	1.9 ± 10.96%	1.9 ± 10.68%	1.8 ± 10.55%
Rehab/Distinct Part Hosp	3.8 ± 11.04%	3.8 ± 11.12%	3.9 ± 11.23%	7.0 ± 20.01%	7.1 ± 20.06%	7.1 ± 20.20%
Home Health Care	8.6 ± 16.23%	8.6 ± 16.23%	8.6 ± 16.09%	15.9 ± 29.41%	15.9 ± 29.40%	15.8 ± 29.22%
Disc-Tran Cancer/Chldrn H	0.5 ± 4.25%	0.5 ± 4.16%	0.5 ± 4.14%	0.9 ± 7.58%	0.8 ± 7.40%	0.8 ± 7.37%
Short Term Hospital	0.3 ± 3.66%	0.4 ± 3.73%	0.4 ± 3.83%	0.6 ± 6.55%	0.6 ± 6.61%	0.7 ± 6.75%
Icf	0.0 ± 1.50%	0.0 ± 1.33%	0.0 ± 1.47%	0.1 ± 2.61%	0.1 ± 2.26%	0.1 ± 2.48%
Disc-Tran To Federal Hc	0.0 ± 0.62%	0.0 ± 0.01%	0.0 ± *nan*%	0.0 ± 1.19%	0.0 ± 0.03%	0.0 ± *nan*%
Disch-Tran To Psych Hosp	0.4 ± 4.20%	0.3 ± 3.86%	0.4 ± 4.13%	0.7 ± 7.29%	0.6 ± 6.80%	0.7 ± 7.22%
Other Facility	0.0 ± 1.30%	0.0 ± 1.19%	0.0 ± 0.01%	0.1 ± 2.36%	0.1 ± 2.13%	0.0 ± 0.03%
Home With Home Iv Providr	0.0 ± 1.42%	0.1 ± 1.60%	0.1 ± 1.72%	0.1 ± 2.50%	0.1 ± 2.90%	0.1 ± 3.01%
Home	9.1 ± 17.60%	9.3 ± 17.76%	9.2 ± 17.66%	16.3 ± 30.90%	16.6 ± 31.15%	16.5 ± 31.05%
Left Against Medical Advi	0.2 ± 2.66%	0.2 ± 2.72%	0.2 ± 2.74%	0.3 ± 4.61%	0.3 ± 4.73%	0.3 ± 4.75%
Hospice-Medical Facility	0.1 ± 1.86%	0.1 ± 1.65%	0.1 ± 1.66%	0.2 ± 3.36%	0.2 ± 3.06%	0.2 ± 3.02%
No Discharge	57.7 ± 24.68%	57.7 ± 24.57%	57.7 ± 24.79%	27.6 ± 26.58%	27.5 ± 26.45%	27.6 ± 26.60%
Snf	5.5 ± 13.24%	5.5 ± 13.32%	5.3 ± 13.09%	10.0 ± 23.95%	10.1 ± 24.02%	9.9 ± 23.79%
Hospice-Home	0.3 ± 3.18%	0.3 ± 3.04%	0.2 ± 2.97%	0.5 ± 5.73%	0.5 ± 5.52%	0.5 ± 5.47%
Snf-Medicaid Only Certif	0.0 ± 0.32%	*nan* ± *nan*%	*nan* ± *nan*%	0.0 ± 0.61%	*nan* ± *nan*%	*nan* ± *nan*%

Table 4.9: All discharge locations we predict on the MIMIC-III cohort, along with the percent of patient-hours in the train set across all 5 splits.

Class	24H			48H		
	Train	Tuning	Held-out Test	Train	Tuning	Held-out Test
	0.3 ± 3.93%	0.3 ± 3.88%	0.3 ± 3.92%	0.5 ± 5.74%	0.5 ± 5.79%	0.5 ± 5.81%
Other	1.4 ± 8.63%	1.5 ± 8.84%	1.4 ± 8.75%	2.1 ± 12.08%	2.2 ± 12.18%	2.1 ± 12.00%
Other External	1.5 ± 8.56%	1.5 ± 8.47%	1.6 ± 8.64%	2.4 ± 12.55%	2.4 ± 12.44%	2.5 ± 12.63%
Other Hospital	0.9 ± 6.68%	0.9 ± 6.56%	0.9 ± 6.53%	1.5 ± 9.91%	1.4 ± 9.68%	1.4 ± 9.67%
Skilled Nursing Facility	5.6 ± 15.51%	5.6 ± 15.52%	5.6 ± 15.56%	8.8 ± 22.67%	8.9 ± 22.79%	8.8 ± 22.71%
No Discharge	48.2 ± 26.67%	48.2 ± 26.73%	48.2 ± 26.71%	21.6 ± 24.60%	21.7 ± 24.74%	21.7 ± 24.68%
Nursing Home	0.4 ± 4.20%	0.4 ± 4.30%	0.4 ± 4.22%	0.6 ± 6.13%	0.6 ± 6.38%	0.6 ± 6.36%
Home	27.8 ± 28.14%	27.6 ± 28.13%	27.6 ± 28.06%	40.2 ± 35.95%	39.9 ± 36.03%	40.0 ± 35.98%
Rehabilitation	2.0 ± 9.46%	1.9 ± 9.30%	2.0 ± 9.30%	3.2 ± 14.15%	3.1 ± 13.96%	3.2 ± 14.00%

Table 4.10: All discharge locations we predict for the eICU cohort, along with the percent of patient-hours in the train set across all 5 splits.

Label	Class	Train	Tuning	Held-out Test
		0.8 ± 0.01%	0.8 ± 0.10%	0.8 ± 0.11%
	Other	3.4 ± 0.04%	3.4 ± 0.14%	3.3 ± 0.33%
	In-Hospital Mortality	3.5 ± 0.05%	3.6 ± 0.29%	3.6 ± 0.32%
	Other External	4.1 ± 0.03%	4.1 ± 0.16%	4.2 ± 0.10%
	Other Hospital	2.6 ± 0.06%	2.5 ± 0.31%	2.5 ± 0.31%
	In-ICU Mortality	4.7 ± 0.01%	4.8 ± 0.15%	4.8 ± 0.17%
	Skilled Nursing Facility	14.7 ± 0.03%	14.9 ± 0.47%	14.7 ± 0.56%
	Home	59.7 ± 0.10%	59.2 ± 0.39%	59.4 ± 0.64%
	Nursing Home	1.0 ± 0.01%	1.1 ± 0.12%	1.1 ± 0.12%
	Rehabilitation	5.5 ± 0.01%	5.5 ± 0.31%	5.6 ± 0.30%

Table 4.11: The prevalence for the various classes for our “Final Acuity” (ACU) task on the eICU cohort, averaged over all 5 rotations.

Label	Class	Train	Tuning	Held-out Test
	Long Term Care Hospital	3.7 ± 0.07%	3.7 ± 0.41%	3.6 ± 0.38%
	Rehab/Distinct Part Hosp	13.2 ± 0.08%	13.4 ± 0.84%	13.4 ± 0.59%
	Disc-Tran Cancer/Chldrn H	1.5 ± 0.05%	1.5 ± 0.31%	1.5 ± 0.28%
	Home Health Care	25.3 ± 0.24%	25.2 ± 1.34%	25.3 ± 1.15%
	Short Term Hospital	1.1 ± 0.05%	1.1 ± 0.22%	1.1 ± 0.18%
	Icf	0.1 ± 0.01%	0.1 ± 0.06%	0.1 ± 0.06%
	Disc-Tran To Federal Hc	0.0 ± 0.00%	0.1 ± 0.03%	0.0 ± *nan*%
	Disch-Tran To Psych Hosp	1.0 ± 0.03%	0.9 ± 0.13%	1.0 ± 0.23%
	In-Hospital Mortality	3.7 ± 0.07%	3.4 ± 0.23%	3.7 ± 0.55%
	In-ICU Mortality	7.4 ± 0.05%	7.5 ± 0.48%	7.5 ± 0.42%
	Home	24.0 ± 0.08%	24.3 ± 0.38%	24.0 ± 0.50%
	Left Against Medical Advi	0.4 ± 0.03%	0.4 ± 0.16%	0.4 ± 0.16%
	Home With Home Iv Providr	0.1 ± 0.01%	0.2 ± 0.11%	0.2 ± 0.08%
	Other Facility	0.1 ± 0.01%	0.1 ± 0.06%	0.1 ± 0.04%
	Hospice-Medical Facility	0.3 ± 0.03%	0.3 ± 0.16%	0.3 ± 0.16%
	Snf	17.3 ± 0.12%	17.2 ± 0.75%	16.9 ± 0.53%
	Hospice-Home	0.9 ± 0.03%	0.8 ± 0.14%	0.8 ± 0.19%
	Snf-Medicaid Only Certif	0.0 ± 0.00%	*nan* ± *nan*%	*nan* ± *nan*%

Table 4.12: The prevalence for the various classes for our “Final Acuity” (ACU) task on the MIMIC-III cohort, averaged over all 5 rotations.

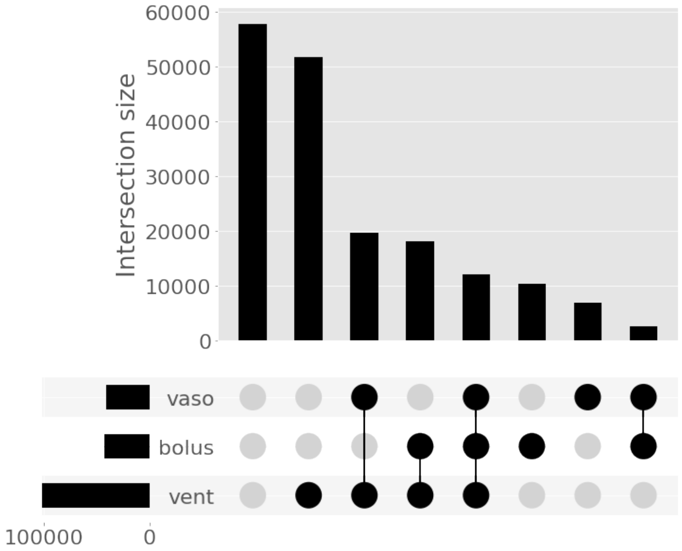


Figure 4-5: A sample Upset plot showing the frequency of relative combinations of our three treatment types: Vasopressors (vaso), Ventilation (vent), and Fluid Bolus administration (bolus) on the MIMIC-III cohort.

Future Treatment Sequence: FTS Figure 4-5 shows which combinations of treatments are most commonly observed over MIMIC-III.

4.9 Hyperparameter Search Analysis

Search Space

For our hyperparameter search procedure, we searched over a wide variety of parameters, including number of epochs, batch size, learning rate, learning rate decay paradigms, L2 regularization penalty, dropout, the maximum length of a patients record included, the size, number, and configuration of various hidden layers, pooling and fully connected stack parameters, and various other model-specific options. All search distributions are shown in Table 4.13. Various numbers of samples were run for each experiment. Universally, at least 100 random samples per search were run. Runs that had more than 100 samples were almost universally single-task runs, not PT/FT runs.

Hyperparameter	Search Space
# Epochs	`Uniform[10, 35]`
Batch Size	`Uniform[8, 512]`
Learning Rate (LR)	`Lognormal[-7, 0.5]`
LR Decay	`Loguniform[-2.3, 0]`
Hidden Dropout	`Uniform[0, 0.5]`
Hidden Size	`Uniform[8, 256]`
Weight Decay	`Uniform[0, 1]`
Input Window Size (h)	`Uniform[12, 168]`
Bidirectional	`Choice[True, False]`
# Hidden Layers	`Uniform[1, 4]`
Encoder Hidden Layer Size	`Uniform[8, 512]`
GRU Pooling Method	`Choice[max, avg, last]`
GRU FC Layer Base Size	`Uniform[8, 512]`
GRU FC Layer Growth	`Loguniform[-1.1, 1.1]`

Table 4.13: The `Hyperopt` search space we used in this work. Distributions are noted in pseudocode, but typically refer directly to the appropriate analog in `Hyperopt` (*e.g.*, a uniform distribution over an integral parameter maps to the quantized uniform distribution that only outputs integers).

4.9.1 Final Results

In Figures 4-6,4-9 we show the absolute performance of all models on the MIMIC-III, eICU cohorts, and Figures 4-7,4-9 show the relative performance of all model types as compared to a ST baseline for the MIMIC-III, eICU cohorts. We note that the eICU results for the ST LOS task appear anomalous—while all runs reported here have gone through internal validation, this oddity warrants further investigation in future work.

4.9.2 Samples Completed

Below are the full experiment counts for all results reported in this work. Note that an extra rotation was also run on the MIMIC-III MT results. This was unintentional, but as all rotations here are separate random train/test splits, we chose to retain the result as it simply improves the quality of our estimates of variance and should add no bias to the results or comparisons.

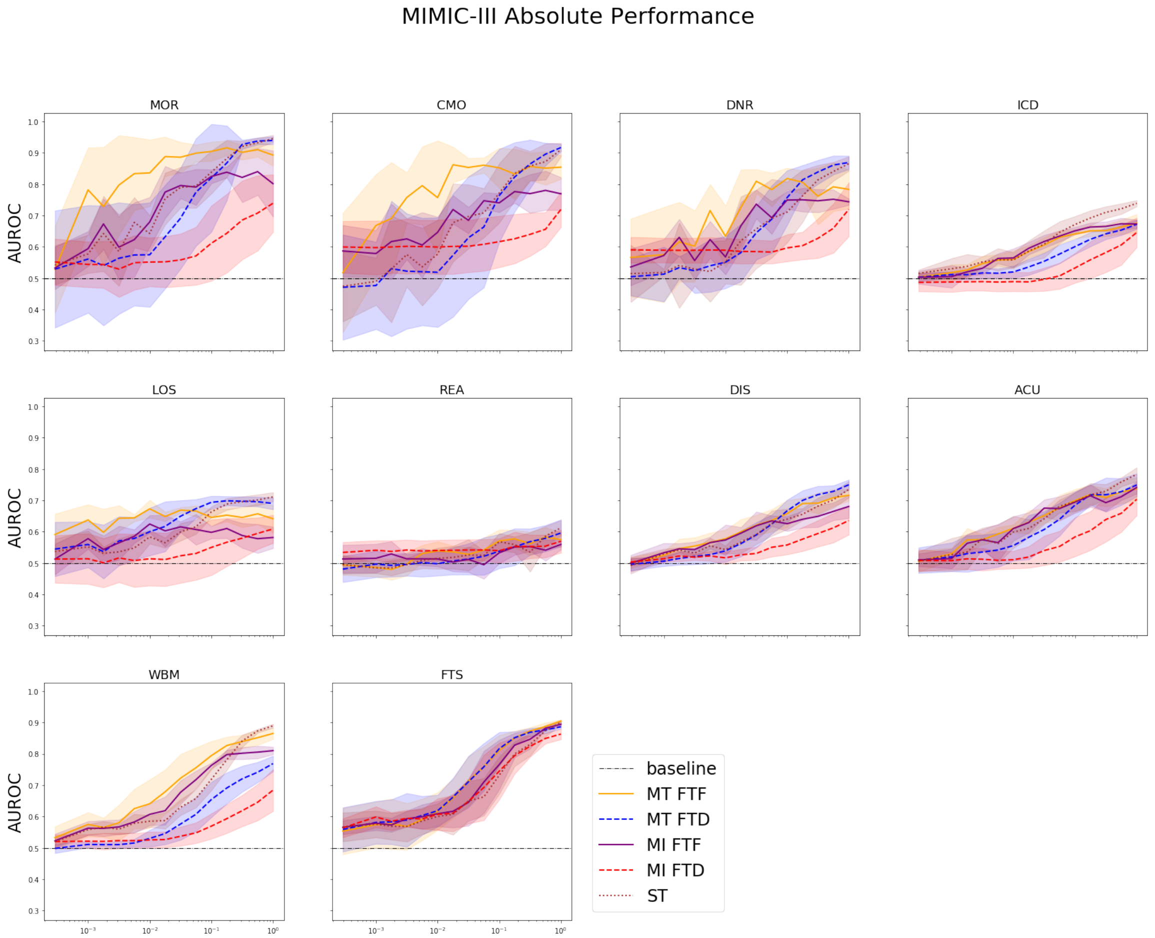


Figure 4-6: Performance in macro-averaged AUROC (y-axis) of various PT/FT models (linestyle) across various FT dataset sub-sampling rates (x-axis), over all FT tasks (subplots) for MIMIC-III.

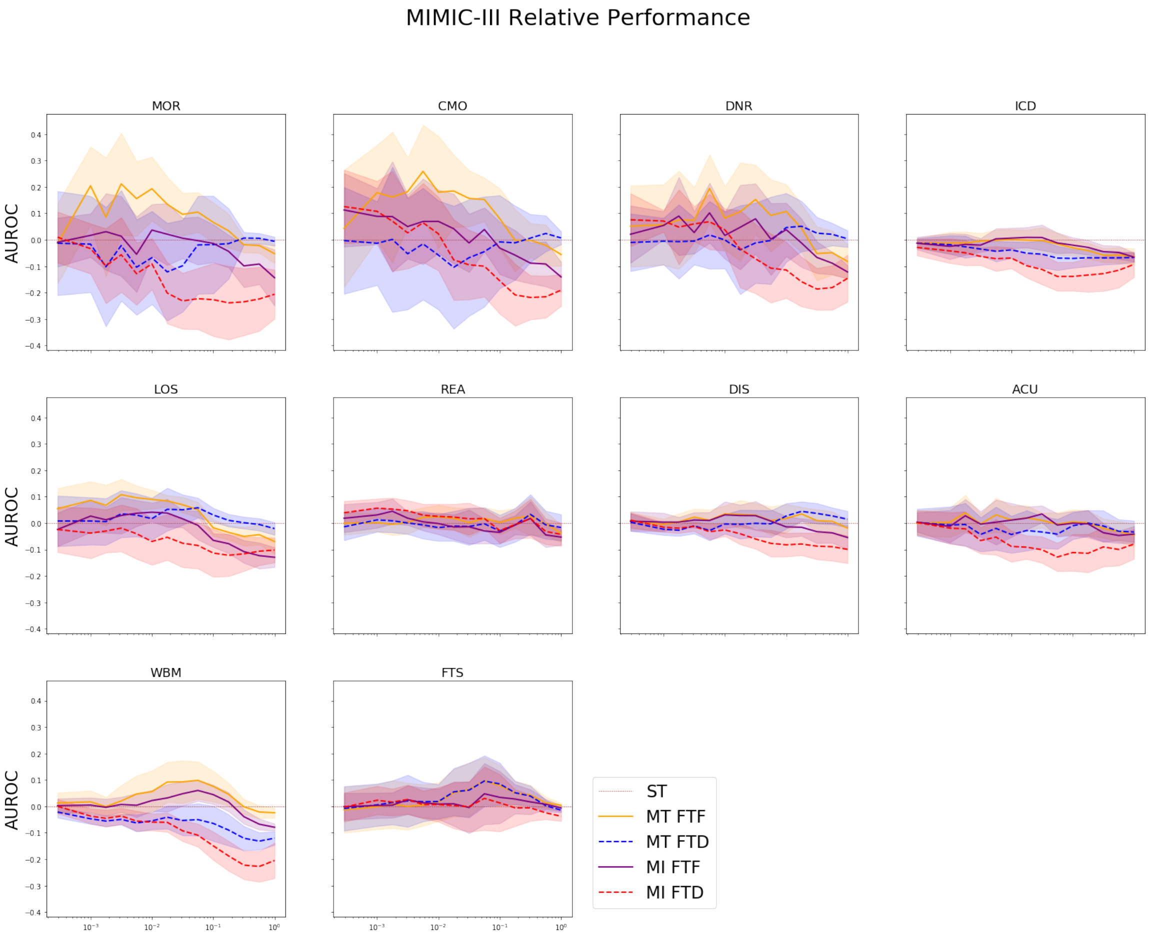


Figure 4-7: The difference between various FT modes and ST results on MIMIC-III.

PT/FT Regime	MI FTD	MI FTF	MT FTD	MT FTF	ST
Task					
ACU	5	5	6	6	5
FTS	5	5	6	6	5
ICD	5	5	6	6	5
DIS	5	5	6	6	5
DNR	5	5	6	6	5
REA	5	5	6	6	5
MOR	5	5	6	6	5
LOS	5	5	6	6	5
CMO	5	5	6	6	5
WBM	5	5	6	6	5

Table 4.14: How many random train/test splits are used to produce each experimental setting shown in this work for MIMIC-III. Unless otherwise stated, the same number of samples are used across all few-shot fractions under a given setting.

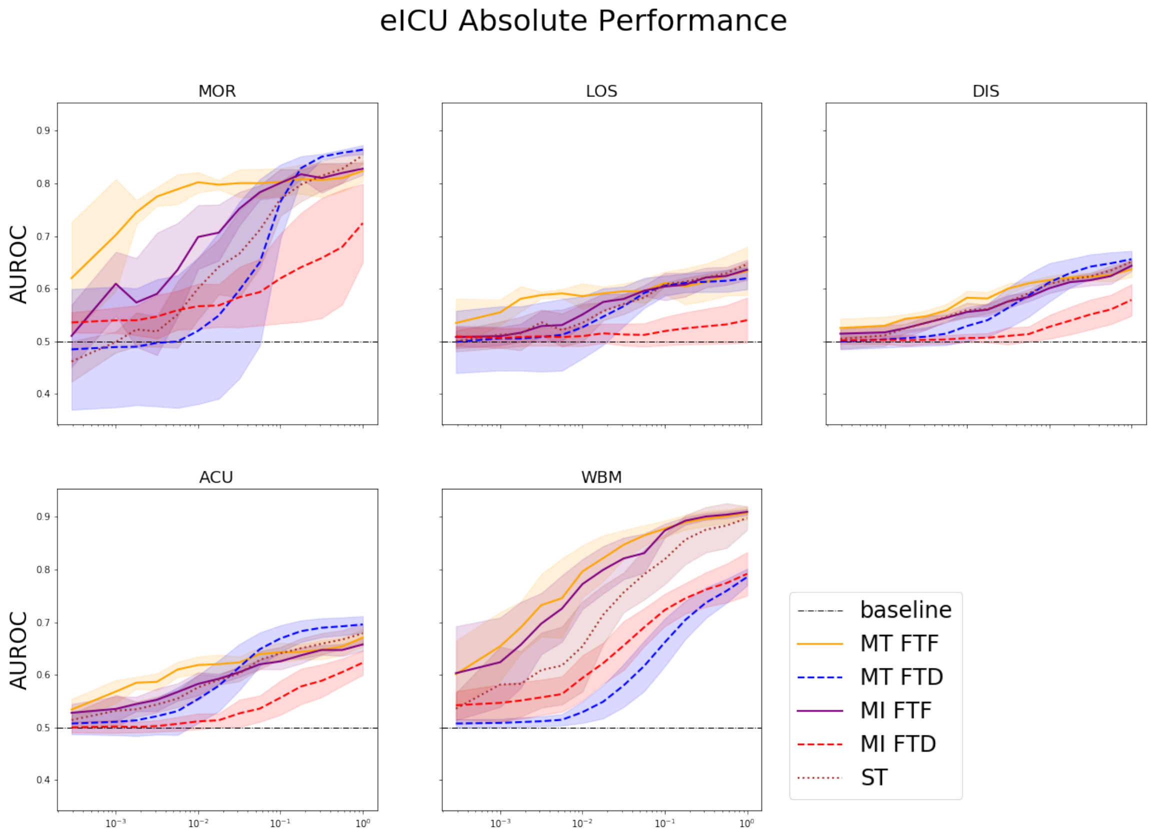


Figure 4-8: Performance in macro-averaged AUROC (y-axis) of various PT/FT models (linestyle) across various FT dataset sub-sampling rates (x-axis), over all FT tasks (subplots) for eICU.

PT/FT Regime	MI FTD	MI FTF	MT FTD	MT FTF	ST
Task					
ACU	5	5	5	5	5
DIS	5	5	5	5	5
MOR	5	5	5	5	5
LOS	5	5	5	5	5
WBM	5	5	5	5	5

Table 4.15: How many random train/test splits are used to produce each experimental setting shown in this work for eICU. Unless otherwise stated, the same number of samples are used across all few-shot fractions under a given setting.

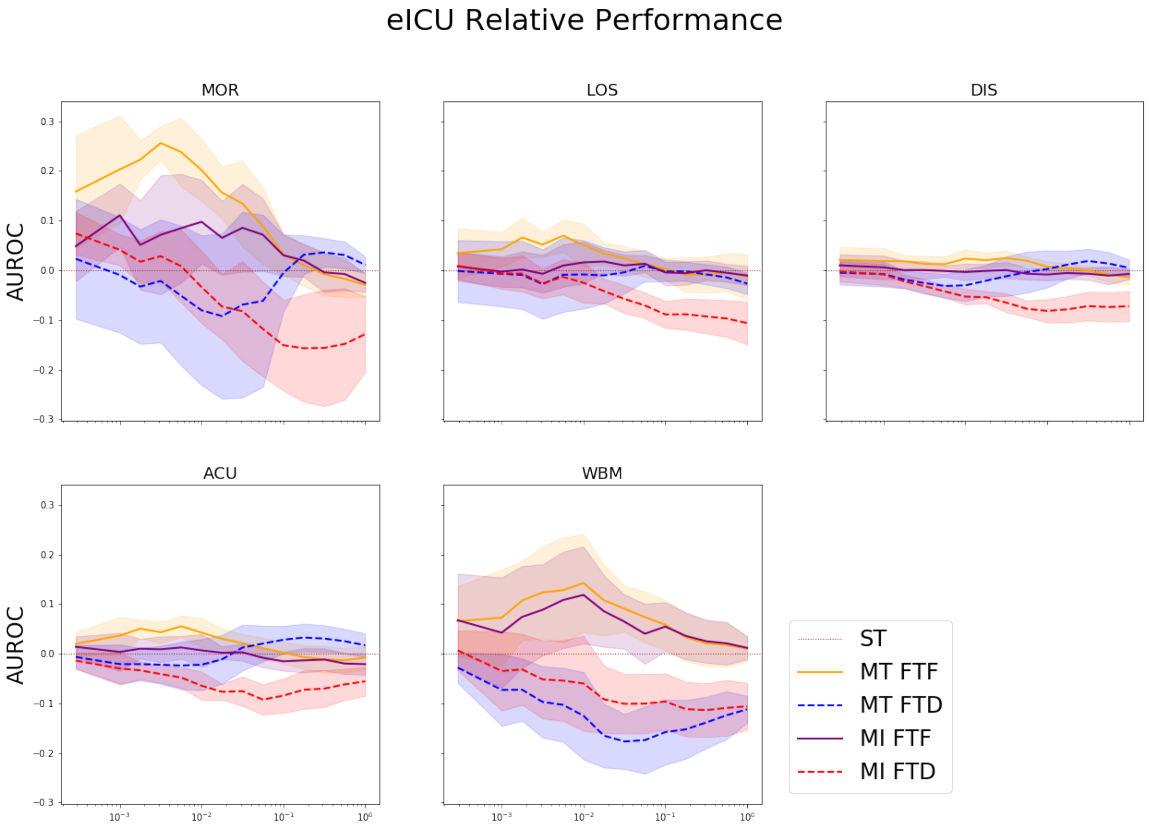


Figure 4-9: The difference between various FT modes and ST results on eICU.

	MIMIC-III	eICU
ACU	-0.02 ± 0.02	0.01 ± 0.02
CMO	0.02 ± 0.02	
DIS	0.0 ± 0.01	0.0 ± 0.01
DNR	0.01 ± 0.03	
FTS	0.0 ± 0.01	
ICD	-0.07 ± 0.04	
LOS	-0.02 ± 0.03	-0.0 ± 0.02
MOR	-0.0 ± 0.01	0.01 ± 0.01
REA	0.01 ± 0.04	
WBM	-0.08 ± 0.03	-0.01 ± 0.02

Table 4.16: Difference between full multi-task hyperparameter search results and single-task results across datasets and tasks. We see no systematic preference towards either multi-task or single-task results.

4.9.3 Negative Transfer Analyses

We can also leverage these experiments to perform a robust analysis of negative transfer within EHR timeseries multi-task learning. In particular, by comparing our multi-task pre-training results, which are trained over all but one task (as one task is withheld for use in fine-tuning) vs. our hyperparameter search results as well as our single-task results vs. our full MT hyperparameter search results. First, at a lgobal scale, we see in Table 4.16 that there is no general apparent preference between ST and MT runs. This suggests that we see no evidence of either global positive or negative transfer.

Examining the transfer utility on a local, per-task level, we can examine how the performance on a particular task is affected by removing a single other task from the full multi-task ensemble or, in a transpose fashion, how including a given task in the learning ensemlbe effects the performance of other downstream tasks. These results are shown visually in Figure 4-10. There, we see that, like our global finding, there is minimal evidence of any universal positive or negative transfer; instead, we see examples of both positive and negative transfer, which, in aggregate, offer no consistent effect. These results suggest that negative transfer is quite likely in a generic MT setting without careful consideration.

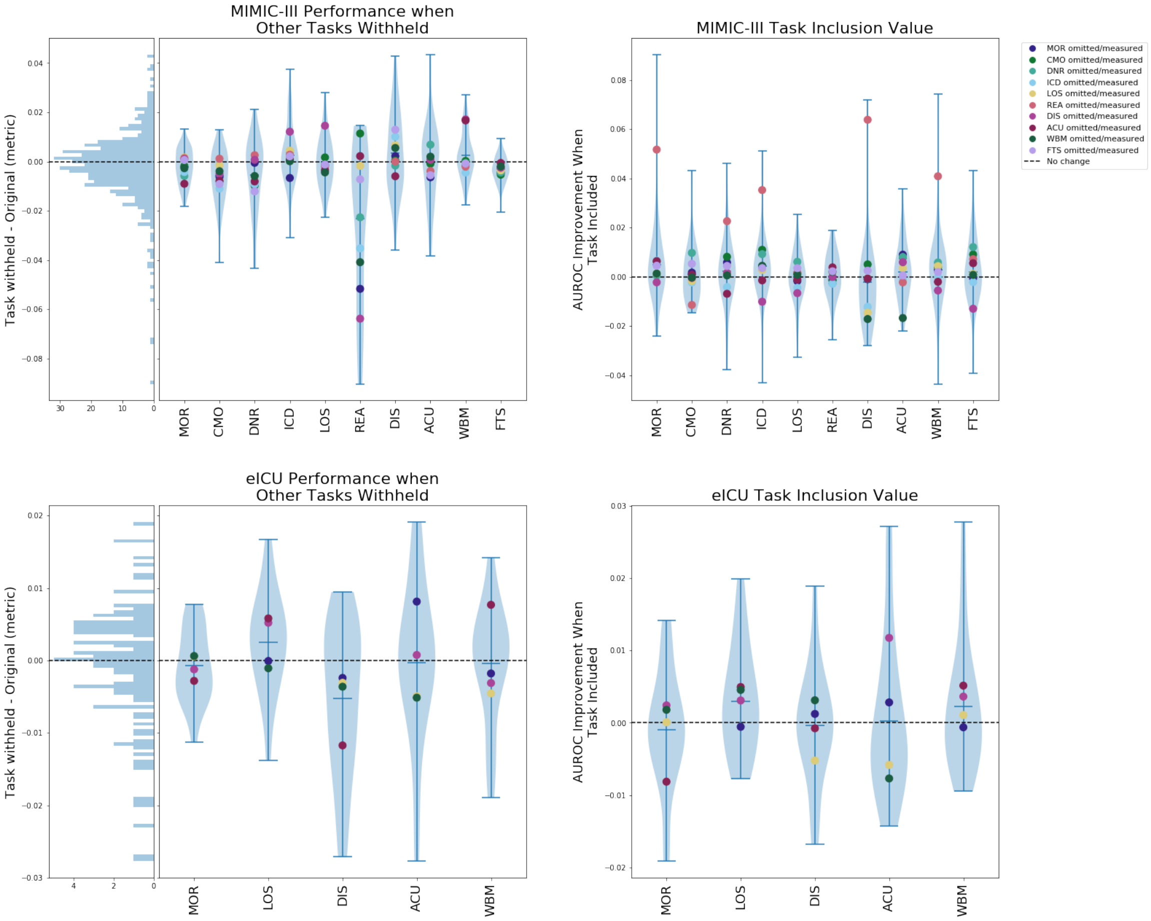


Figure 4-10: We examine the value either for a downstream task or by a downstream task in the context of multi-task ensemble makeup. On the left, we show, for each task on the x-axis, the performance difference *on that task* (y-axis) between a MT learning setting where a single other task (colored dot) is omitted from the ensemble vs. a full MT learning ensemble. This plot also shows an overall histogram of these discrepancies to its left. On the right, we show the transpose view – for any given task (x-axis), we plot how much performance on other tasks (colored dots) is *improved* (y-axis) by *including* the x-axis task in the learning ensemble. The same numbers are summarized in both plots, just from differing perspectives (in particular, the coordinates in the right plot are negated and transposed from those in the left). We see that, like our global finding, there is minimal evidence of any general positive or negative transfer here – instead, any relationships are highly task specific, and on average no transfer is observed one way or another. Note that while these results suggest there is no universal negative transfer, they do suggest that negative transfer is quite likely in a generic MT setting without careful consideration.

Chapter 5

Structure Inducing Pre-training

Abstract

In this thesis, I explore pre-training methods for clinical and biomedical data, and in particular pre-training methods which can incorporate prior domain knowledge and structure. This chapter, currently under review at JMLR, is the capstone of these analyses, as it presents a unified framework for realizing global structure within pre-training systems, and demonstrates through both theory and experiments why that structure is important. Note that interested readers can refer to [164] or [131] for other related works.

Language model derived pre-training has proven incredibly impactful in machine learning. However, considerable uncertainty remains around when pre-training offers improvement for novel fine-tuning tasks and what, if any, meaningful structure is captured during pre-training. Here, we analyze this problem by exploring how existing pre-training methods impose relational inductive biases, finding that the study of how to constrain the per-sample latent space is significantly underdeveloped. Based on these analyses, we introduce a descriptive framework for pre-training methods that allows for a granular description of how global structure can be induced during pre-training. We present a theoretical analysis of this framework from first principles and establish a novel connection between relational inductive bias of pre-training and fine-tuning performance. We build upon these findings with simulations and empirical studies on benchmarks spanning 3 data modalities and 10 fine-tuning tasks. These investigations validate our theoretical analyses and inform the design of novel pre-training methods to incorporate provably richer inductive biases than existing methods in line with user-specified relational graphs.

5.1 Introduction

The pre-training (PT)/fine-tuning (FT) learning paradigm has had tremendous impact on machine learning [47, 44, 22]. In PT/FT, we pre-train an encoder $f_{\boldsymbol{\theta}} : \mathcal{X} \to \mathcal{Z}$ which maps our domain of interest $\mathcal{X}$ into a latent space $\mathcal{Z}$. This encoder $f_{\boldsymbol{\theta}}$ is then transferred for use in various downstream tasks (which are not known at pre-training time). We evaluate PT/FT systems via the transfer performance of $f_{\boldsymbol{\theta}}$. This transfer performance is inherently a question of whether or not the FT task can benefit from the geometry of $\mathcal{Z}$ that is induced by the pre-training process.

Despite the importance of the latent space geometry, we posit that many recent large-scale PT/FT models do not explicitly consider the latent space geometry during pre-training. While we discuss this argument fully in Section 5.2, consider as a motivating example the RoBERTa PT model [121]. This model pre-trains a transformer neural network via a *masked language modelling* (MLM) task, in which the model is fed input sequences of tokens, some of which are masked, and tasked to recover the identity of the masked tokens. While language modelling imposes strong constraints on how the representations of individual *tokens* relate to one another, nothing in this task explicitly constrains how the representations of *samples* (*e.g.*,, sentences) relate to one-another. Concretely, consider trying to use this model for sentiment analysis [126]. This task would be implemented via binary classification leveraging the pre-trained, per-sample (*e.g.*,, `[CLS]`) embeddings; however, we can make no guarantees about whether the PT process will have ensured these sample embeddings meaningfully reflect positive vs. negative sentiment (Figure 5-1).[1]

Present work. In this work, we perform novel theoretical and empirical analyses examining how existing PT methods do or do not constrain the per-sample latent space geometry, and whether new methods that more aggressively constrain this geometry offer any benefits over current models. Our analyses yield a new analytical framework for describing PT methods, under which the PT objective is subdivided

[1] Note that in NLP, we can actually leverage the flexibility of natural language to sidestep this problem; however, the same flexibility is not possible in other domains. See Section 5.2 for further commentary.

into two components: first, a pure language-model inspired imputation/denoising objective that leverages the intra-sample/per-token relationships, and, second, a loss term explicitly driven to regularize the geometry of the per-sample latent space $\mathcal{Z}$ and reflect the connectivity patterns of a user-specified PT graph, G_{PT}. By relying on graphs to capture the structure we wish to induce in $\mathcal{Z}$, this PT paradigm can capture diverse relationships, such as those motivated by external knowledge (*e.g.*,, [252]), self-supervised constraints extracted from the data (*e.g.*,, [215, 90]), or graphs summarizing distances between samples in an alternate modality. Moreover, this new PT framework is simultaneously specific enough to allow us to make theoretical guarantees about how different PT graphs impact the FT performance; general enough to encompass a variety of existing PT systems, including BERT [47], ALBERT [107], RoBERTa [121], MT-DNN [119], and PLUS [138]; and expressive enough to motivate new PT methods that have not been previously studied.

Our work advances PT/FT research through three major contributions. First, we establish theoretical results quantifying how the structure of the graph G_{PT} relates to FT task performance. Crucially, this formalization in our new PT paradigm offers insight into when PT will or won't add value over supervised learning alone. Second, we show that our new framework for PT provides a recipe for pre-training with rich, per-sample latent space geometry constraints. Specifically, we profile three new PT methods leveraging different kinds of PT graphs and demonstrate their utility on three data modalities and 10 FT tasks, showing in all cases that these new methods perform at or above the level of comparable baseline methods. Finally, third, we include a discussion on how our framework prompts new research directions that are of interest to the PT/FT research community.

The rest of our work proceeds as follows. In Section 5.2, we discuss the extent to which existing language-model derived PT methods do or do not regularize per-sample latent space geometry. Next, in Section 5.3, we formally define and examine our new PT paradigm, detailing precisely how we decompose the PT loss and what constraints we place on the different loss components. In Section 5.4, we illustrate theoretically how we can formally relate PT graph structure to FT performance under our PT

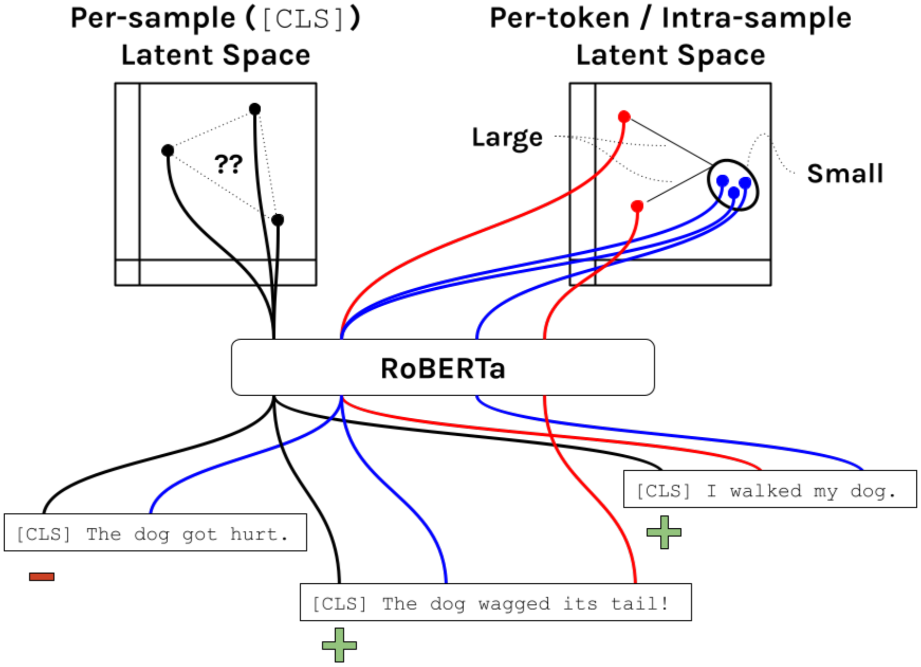


Figure 5-1: Language model pre-training methods typically constrain the *per-token* latent space through the language modelling objective. However, they impose no or only much weaker constraints on the *per-sample* latent space.

paradigm. Then, in Section 5.5 we further validate these theoretical findings in semi-synthetic experiments. In Section 5.6, we empirically examine our theoretical model on real-world datasets spanning 3 modalities and 10 FT tasks. Our analysis shows that across all three data modalities leveraging per-sample latent space constraints yields PT methods that match or outperform baselines. Finally, in Sections 5.7, 5.8, and 5.9 we offer closing discussion, highlight related work, and conclude.

5.2 Pre-Training & Latent-Space Geometry

In this section, we offer more details on our motivating hypothesis introduced in Section 5.1. First, we define both the per-token and per-sample latent spaces, and offer commentary on why this distinction matters particularly outside of the NLP domain. Second we present an analysis of over 27 different language model (LM) or LM-derived PT methods exploring how their objective functions impose structure across both types of latent space. This analysis crystallizes our claim that methods to regularize the per-sample latent space are under-explored, and motivates our new PT paradigm in Section 5.3.

5.2.1 Per-Sample vs. Per-Token Latent Space

Consider a PT model $f_{\boldsymbol{\theta}}$ trained using a masked language modelling objective (MLM), under which certain tokens in the sentence are masked, and the encoder is trained to recover the masked tokens. Given an input sentence $\boldsymbol{s}_j = w_1^{(j)}, w_2^{(j)}, \ldots, w_n^{(j)}$, $f_{\boldsymbol{\theta}}$ will output both per-token embeddings of the individual tokens $w_i^{(j)}$ and a whole-sample embedding of the sentence/sample $\boldsymbol{s}$. At a minor risk of ambiguous notation, we will denote these via $f_{\boldsymbol{\theta}}(w_i^{(j)}|\boldsymbol{s}_j)$ denoting the per-token embedding of token i and $f_{\boldsymbol{\theta}}(\boldsymbol{s}_j)$ denoting the whole-sample embedding of $\boldsymbol{s}_j$. For the BERT model [47], for example, $f_{\boldsymbol{\theta}}(\boldsymbol{s}_j)$ will be given by the output embedding of the `[CLS]` token and $f_{\boldsymbol{\theta}}(w_i^{(j)}|\boldsymbol{s}_j)$ will be given by the output embedding of the i-th token.

There are two ways to interpret the "latent space" $\mathcal{Z}$ of $f_{\boldsymbol{\theta}}$. First, we can take the space induced by the embeddings of free-text tokens, $\mathcal{Z}^{(\mathrm{T})} = \{f_{\boldsymbol{\theta}}(w_i|\boldsymbol{s}_j) \forall i, j\}$. This is the *per-token latent space*. Second, we can take the space induced by the embeddings of the samples/sentences, $\mathcal{Z}^{(\mathrm{S})} = \{f_{\boldsymbol{\theta}}(\boldsymbol{s}_j) \forall j\}$, which is the *per-sample latent space*. In the rest of this paper, we will use $\mathcal{Z}$ in general to refer to $\mathcal{Z}^{(\mathrm{S})}$.

Both of these spaces are very different and are useful in different contexts; for a task like named entity recognition, where the unit of classification is single or short span of tokens, analyzing the per-token latent space will be more informative, whereas for a task like sentiment analysis, where the unit of classification is an entire sample, the per-sample latent space would be preferred [47].

5.2.2 NLP vs. Other Language Domains

In Figure 5-1, we illustrated that even while a PT objective like masked language modelling will directly enrich the per-token space $\mathcal{Z}^{(\mathrm{T})}$, it will not necessarily similarly constrain the per-sample latent space $\mathcal{Z}^{(\mathrm{S})}$, and that this lack of per-sample latent space constraints could yield weaker models at FT time. One seeming contradiction to this is that methods like RoBERTa [121] succeed across both per-token and per-sample tasks.

In fact, this observation does not contradict our hypothesis, but instead reflects a unique advantage of the natural language modality that does not apply in other domains. In particular, in the NLP domain, we can leverage the flexibility of language to sidestep any deficit in $\mathcal{Z}^{(\mathrm{S})}$ by re-framing per-sample tasks as per-token, language modelling tasks. Significant literature exists documenting this phenomenon, through the lenses of both prompting, cloze-filling models, and general theory [22, 181, 178]. For example, [181] examine the efficacy of pre-trained language models on sentiment analysis explicitly, and shows that the language modelling component alone can be used in a per-token manner to indirectly solve a review sentiment analysis task by judging the likelihood of following the review with a ":)" emoji vs. a ":(" emoji [181]. In this way, we shift the *per-sample* task of sentiment analysis to a *per-token* task about an (inserted) final token.

However, language model pre-training has also inspired many derived methods to be used in other, non-NLP domains as well. For example, in modelling graphs, [92] have examined vertex or edge-masking strategies reminiscent of MLM, with vertices and edges analogous to tokens, and entire graphs whole samples; in modelling timeseries data, [129] have examined masked imputation models, with timepoints analogous to tokens and whole timeseries to samples; and in modelling protein sequences [138] have used masked language modelling directly, with individual amino acids representing tokens and entire proteins representing samples. In all three of these domains, we do not have NLP's ability to re-frame per-sample tasks as "per-token" tasks, and accordingly the problem of insufficient per-sample latent space regularization will likely be much more severe on these domains. Accordingly, existing work, including the three works referenced above, all find that augmenting the language model pre-training task with additional, per-sample level supervised tasks can be beneficial, or even instrumental, to improving performance [92, 236, 129, 138].

5.2.3 Categorization of Language Model PT Methods

In Table 5.1 (and extended in Appendix Section 5.10.2), we survey 27 PT methods and categorize how their objective functions constrain their respective latent spaces. We find that a variety of methods for constraining the per-token latent space have been explored, including direct LM, entity and relation masking, knowledge graph (KG) alignment methods, joint token and entity embeddings, attention over external knowledge bases, and embedding aggregation via known relations. Despite this breadth of methods focused on improving modelling at the per-token level, across all methods save for KEPLER [221], only single- or multi-task classification is used to regularize the per-sample latent space. This disparity suggests that study on encoding per-sample relational structure in PT is lacking.

5.3 Re-Framing the Pre-Training Paradigm

Given the limited prior research into regularizing the per-sample latent space for LM-based PT methods (Section 5.2), we next introduce a new, descriptive PT framework to explicitly separate per-token and per-sample latent space regularization. We begin by outlining the PT problem formally, then introduce our new paradigm, clarify how methods within our framework are constrained, and conclude with an overview of existing methods that can be realized within our framing.

5.3.1 Pre-Training Problem Formulation

Given a dataset $\boldsymbol{X}_{\mathrm{PT}} \in \mathcal{X}^{N_{\mathrm{PT}}}$, a PT method aims to learn an encoder $f_\theta : \mathcal{X} \to \mathcal{Z}$ such that f_θ can be transferred to FT tasks that are unknown at pre-training time. While we can leverage additional information at PT time to inform the training of f_θ (*e.g.*,, PT-specific labels $\mathcal{Y}_{\mathrm{PT}}$), the encoder f_θ *must* take only samples from $\mathcal{X}$ as input so it can be effectively used for fine-tuning. Pre-training methods attempt to solve this problem by training f_θ via a pre-training loss $\mathcal{L}_{\mathrm{PT}}$ over the dataset $\boldsymbol{X}_{\mathrm{PT}}$. In language model (LM) or LM derived systems, $\mathcal{L}_{\mathrm{PT}}$ can be a masked imputation

Method	Masked or standard language modelling	Named entity masking	Relation masking†	Per-token knowledge graph alignment	Named entity recognition and linking	(Unconstrained) attention over a KG	Joint token and entity embeddings	Aggregating embedding across relations†	Single-task classification	Multi-task classification	Whole-sample knowledge graph alignment
	Per-token								**Per-sample**		
GPT-series [161, 162, 22]	✓										
RoBERTa [121]	✓										
BERT [47]	✓								✓		
ALBERT [107]	✓								✓		
MT-DNN [119]	✓									✓	
ERICA [160]	✓			✓							
COLAKE [202]	✓	✓	✓								
LUKE [231]	✓	✓									
ERNIE [204]	✓	✓									
ERNIE 2.0 [205]	✓	✓								✓	
ERNIE 3.0 [203]	✓	✓	✓							✓	
ERNIE [247]	✓	✓					✓		✓		
KnowBERT [156]	✓				✓	✓	✓				
KgPLM [82]	✓	✓							✓		
KEPLER [221]	✓										✓
JAKET [238]	✓						✓	✓			
SMedBERT [245]	✓			✓			✓	✓	✓		
KeBioLM [239]	✓				✓	✓	✓				
Coke [198]	✓	✓					✓	✓	✓		
MG-BERT [13]	✓							✓			
EHR PT [129]	✓									✓	
Graph PT [92]	✓									✓	
MG-BERT [246]	✓										
PLUS [138]	✓								✓		
TAPE [166]	✓										

Table 5.1: A subset of existing pre-training methods, broken down by how they constrain per-token and per-sample latent space geometries. This list excludes a number of methods that augment LMPT with additional, fine-tuning time procedures to incorporate knowledge, as they do not change the PT step but only the FT step. † These models rely on entity identification from source text and on relation incorporation. As such, they may struggle to generalize to FT tasks where relations are unknown.

loss, which only explicitly regularizes the per-token latent space, not the per-sample latent space (Section 5.2).

5.3.2 Our New Pre-Training Problem Formulation

To investigate PT methods, we re-cast the above problem as follows. As in the established setting, our input is a dataset $\boldsymbol{X}_{\text{PT}} \in \mathcal{X}^{N_{\text{PT}}}$, we aim to learn an encoder $f_\theta : \mathcal{X} \to \mathcal{Z}$ that enables effective transfer performance to FT tasks, and f_θ will be learned via a task defined with loss $\mathcal{L}_{\text{PT}}$. We differ from the established setting, however, in two regards:

1. We assume that as an input to the PT problem we also have a pre-training graph $G_{\text{PT}} = (V, E)$ where vertices denote pre-training samples within $\boldsymbol{X}_{\text{PT}}$ (*e.g.*, $\{\boldsymbol{x}_{\text{PT}}^{(i)} | 1 \leq i \leq N_{\text{PT}}\} \subseteq V$) and edges represent user-specified relationships.
2. We decompose the loss $\mathcal{L}_{\text{PT}}$ into two components, weighted with hyperparameter $0 < \lambda_{\text{SI}} < 1$ as follows: $\mathcal{L}_{\text{PT}} = (1 - \lambda_{\text{SI}})\mathcal{L}_{\text{M}} + \lambda_{\text{SI}}\mathcal{L}_{\text{SI}}$. These components are a traditional, per-token objective with loss $\mathcal{L}_{\text{M}}$ (*e.g.*,, a language model) and a structure-inducing objective with loss $\mathcal{L}_{\text{SI}}$ designed to regularize the per-sample latent space geometry in accordance with the relationships in G_{PT}.

Beyond typical characteristics, such as continuity and differentiability, under our PT formulation, we further constrain the form of the new loss term $\mathcal{L}_{\text{SI}}$ via the following constraints:

1. There must exist a distance function $d : \mathcal{Z} \times \mathcal{Z} \to \mathbb{R}$, radius $r \in \mathbb{R}$, and loss value $\ell^* \in \mathbb{R}$ such that at any solution $\boldsymbol{\theta}^*$ such that $\mathcal{L}_{\text{SI}}(\boldsymbol{\theta}^*) < \ell^*$ the following property holds. The learned embeddings $\boldsymbol{z}_i = f_{\boldsymbol{\theta}^*}(\boldsymbol{x}_i)$ must recover the graph G_{PT} under a radius nearest neighbor connectivity algorithm via distance function d and radius r. More formally, it must be the case that $(\boldsymbol{x}_i, \boldsymbol{x}_j) \in E$ if and only if $d(f_{\boldsymbol{\theta}^*}(\boldsymbol{x}_i), f_{\boldsymbol{\theta}^*}(\boldsymbol{x}_j)) < r$.
2. It must be possible to find a solution $\boldsymbol{\theta}^*$ satisfying the above constraint. Assuming that all input $\boldsymbol{x}_i$ are distinct, this requirement is about the geometry of the graph

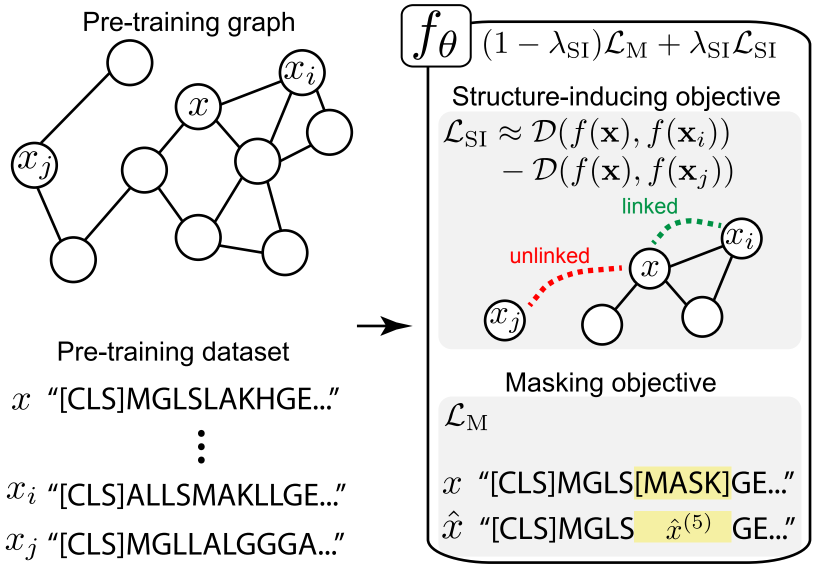

Pre-training graph
x_j
x
x_i
Pre-training dataset
x "[CLS]MGLSLAKHGE..."
x_i "[CLS]ALLSMAKLLGE..."
x_j "[CLS]MGLLALGGGA..."
f_θ
$(1-\lambda_{\mathrm{SI}})\mathcal{L}_{\mathrm{M}} + \lambda_{\mathrm{SI}}\mathcal{L}_{\mathrm{SI}}$
Structure-inducing objective
$\mathcal{L}_{\mathrm{SI}} \approx \mathcal{D}(f(\mathbf{x}), f(\mathbf{x}_i)) - \mathcal{D}(f(\mathbf{x}), f(\mathbf{x}_j))$
linked
unlinked
x
x_i
x_j
Masking objective
$\mathcal{L}_{\mathrm{M}}$
x "[CLS]MGLS[MASK]GE..."
$\hat{x}$ "[CLS]MGLS $\hat{x}^{(5)}$ GE..."

Figure 5-2: We re-cast the PT formulation as taking in a pre-training graph G_{PT} as an auxiliary input. G_{PT} is used to define a new structure-inducing objective $\mathcal{L}_{\mathrm{SI}}$, which pushes a pre-training encoder $f_{\boldsymbol{\theta}}$ to embed samples via a contrastive loss such that closeness in the latent space corresponds to edges in G_{PT}.

G_{PT} and the feasibility of realizing embeddings for G_{PT} within the latent space $\mathcal{Z}$—*e.g.*,, if $\mathcal{Z} = \mathbb{R}^k$, then with a very small k it may be impossible to produce embeddings that fully reflect the geometry of the graph G_{PT} in the reduced dimensionality of $\mathcal{Z}$, and we wish to exclude such cases from our consideration.

Importantly, while we take the graph G_{PT} as an input to the PT problem, *we cannot use it as a direct input to $f_{\boldsymbol{\theta}}$.* Just like in traditional pre-training, $f_{\boldsymbol{\theta}}$ must take as input only samples from $\mathcal{X}$. This is *because otherwise we can not apply $f_{\boldsymbol{\theta}}$ to the same, general class of FT tasks over domain $\mathcal{X}$.* Thus, while there are methods that operate over graphs directly, *e.g.*,, graph neural network methods [101], these are outside our scope and cannot be used in place of the encoder $f_{\boldsymbol{\theta}}$.

The outline of our pre-training problem formulation is shown in Figure 5-2. This formulation does not constrain the architecture of $f_{\boldsymbol{\theta}}$ or the form of the per-token objective $\mathcal{L}_{\mathrm{M}}$. As such, it is sufficiently general to encompass many existing PT methods, which we show next.

5.3.3 Examining Existing Pre-Training Methods via Our Re-Framing

We can realize many existing PT methods within our re-framing. As we can see from Table 5.1, there are two primary vehicles by which existing pre-training methods constrain the per-sample latent space geometry: Single-task and multi-task classification. In each case, we show that there exist pre-training graphs G_{PT} and losses $\mathcal{L}_{\mathrm{SI}}$ that realize these methods under our re-framed PT problem formulation.[2] After outlining single and multi-task classification, we show how this re-framed PT problem allows us to design new PT objectives by discussing two new $\mathcal{L}_{\mathrm{SI}}$ objectives that have not been considering to date.

Masked language modeling with per-sample classification. The simplest manner of augmenting PT to induce per-sample structure is to add a single-task classification task at the level of the whole sample. This is the manner used by a number of existing methods, including BERT [47], PLUS [138], and others. These methods augment pre-training by adding a classification head that ingests the per-sample latent space $\mathcal{Z}$ and uses a single linear layer to predict PT-specific labels $\mathcal{Y}_{\mathrm{PT}}$, training this process via, *e.g.*,, a cross entropy loss. In this setting, we can view the matrix used in the linear layer of the classification head as a set of embeddings for each class, and the cross entropy loss as a function that seeks to maximize the inner product between the embedding of a sample $\boldsymbol{z} \in \mathcal{Z}$ and the embedding of its associated class $y \in \mathcal{Y}$, while minimizing the inner product between non-matching pairs of samples and classes. Approaching optimality, the inner product of a matching pair will be large while those of any non-matching pairs will be low, with a large margin separating the two.

We construct a bipartite pre-training graph G_{PT} where nodes represent samples in $\boldsymbol{X}_{\mathrm{PT}}$ and class labels $y \in \mathcal{Y}_{\mathrm{PT}}$, and edges $(\boldsymbol{x}_i, y_i)$ connect samples with their class labels. We can see then that the cross entropy loss produces embeddings that can

[2] Note this omits the unique case of KEPLER [221], which leverages whole-sample knowledge graph alignment, which can only be realized by our loss subdivision when restricted to graphs with a single edge type.

recover this graph as we approach optimality using the cosine distance function. Thus, a simple classification objective per-sample is a valid example of a method under our restrictions, albeit using a highly constrained graph G_{PT}.

Masked language modeling with multi-task classification. We construct a pre-training graph G_{PT} where nodes represent samples in $\boldsymbol{X}_{\mathrm{PT}}$ and class labels $y \in \mathcal{Y}_{\mathrm{PT}}$, and edges $(\boldsymbol{x}_i, y_i)$ connect samples with their class labels (with each node $\boldsymbol{x}_i$ being connected to potentially many different labels y_j corresponding to each of its tasks). We can see that cross-entropy loss can produce embeddings that recover graph G_{PT} as we approach optimality using the cosine distance function. Thus, a multi-task classification objective is a valid example of a method under our re-framed PT problem, albeit one in which the graph G_{PT} is constrained to reflect only label information in $(\boldsymbol{x}_i, y_i)$.

5.3.4 Defining New Pre-Training Methods via Our Re-Framing

In settings discussed so far, our pre-training graphs have had rigid structures defined directly based on supervised labeled pairs given in $(\mathbf{X}_{\mathrm{PT}}, \mathcal{Y}_{\mathrm{PT}})$. Accordingly, the methods described above do not apply to settings where pre-training graphs G_{PT} are not so constrained. Yet, our loss subdivision $\mathcal{L}$, and the notion that $\mathcal{L}_{\mathrm{SI}}$ must recover the graph G_{PT} as it approaches optimality immediately suggests new possibilities. In particular, if we design a new loss function that pushes embeddings of samples to be close to their G_{PT} neighbors, then this loss function satisfies our constraints and can help to induce the desired structure in $\mathcal{Z}$. To this end, we can build on existing research, including graph embeddings [62], structure-preserving metric learning [76], and topological data analysis [141], among others.

In this example (and for the concrete methods examined in Section 5.6), we investigate two new losses drawn from metric learning: a contrastive loss [76], and a multi-similarity loss [222], both of which we adapt to the setting where relationships are determined via a graph rather than supervised labels.

The multi-similarity loss, parameterized by w_+, w_-, and t, is given below:

$$\mathcal{L}_{\text{SI}} = \frac{1}{Nw_+} \log \left(1 + \sum_{(i,j) \in E} e^{-w_+ (\langle f_{\boldsymbol{\theta}}(\boldsymbol{x}_i), f_{\boldsymbol{\theta}}(\boldsymbol{x}_j) \rangle - t)} \right) + \frac{1}{Nw_-} \log \left(1 + \sum_{(i,j) \notin E} e^{w_- (\langle f_{\boldsymbol{\theta}}(\boldsymbol{x}_i), f_{\boldsymbol{\theta}}(\boldsymbol{x}_j) \rangle - t)} \right),$$

Our contrastive loss is modeled after [76]'s version. For this loss, we assume we are given the following mappings: 'pos', which maps $\boldsymbol{x}$ into a positive node (*i.e.*,, linked to $\boldsymbol{x}$ in G_{PT}), and 'neg', which maps $\boldsymbol{x}$ into a negative node (*i.e.*,, not linked to $\boldsymbol{x}$ in G_{PT}). The union of points $\boldsymbol{x}$ and its images under 'pos' and 'neg' mappings form a full minibatch. This loss is specified by the positive and negative margin parameters μ_+ and μ_-, as well as by a distance function $\mathcal{D}$, as:

$$\mathcal{L}_{\text{SI}}^{(\text{CL})} = \frac{1}{N} \sum_{\boldsymbol{x}_i \in \boldsymbol{X}} \max(\mathcal{D}(\boldsymbol{x}_i, \text{pos}(\boldsymbol{x}_i)) - \mu_+, 0) + \frac{1}{N} \sum_{\boldsymbol{x}_i \in \boldsymbol{X}} \max(\mu_- - \mathcal{D}(\boldsymbol{x}_i, \text{neg}(\boldsymbol{x}_i)), 0).$$

In addition to a loss term, we can also use negative sampling strategies to improve efficiency. Using the full graph G_{PT}, which is not available in many contexts where negative sampling is employed, we can leverage the distance between samples calculated on G_{PT}, which provides a complementary source of information beyond embedding space distance alone. In practice, one could use this, for example, to limit negative samples to occur within the same connected component, but more complex strategies based on graph sampling [240] could also be used.

In our analyses in Sections 5.5 and 5.6, we refer to methods that leverage these losses and more general graphs G_{PT} than the simple graphs used for classification tasks as "Structure-inducing Pre-training" (SIPT) methods, because these losses are designed specifically to induce structure in the per-sample latent space.

5.4 Theoretical Analysis of Pre-Training Methods

Next, we theoretically investigate the formulation outlined in Section 5.3. In particular, we examine how we can relate the structure of PT graph to FT task performance under our paradigm. To this end, the following Theorem 1 provides our central claim.

Informally, the theorem states that as a pre-trained embedder f over graph G_{PT} approaches optimality, it produces an embedding space such that nearest-neighbor performance for any downstream task is lower bounded by the performance that could be obtained via a nearest neighbor algorithm over graph G_{PT}.

Theorem 1. *Let $\boldsymbol{X}_{PT}$ be a PT dataset, G_{PT} be a PT graph, and let $f_{\boldsymbol{\theta}^*}$ be an encoder pre-trained under a PT objective permissible under our framing that realizes a $\mathcal{L}_{\text{SI}}$ value no more than ℓ^*. Then, under embedder f, the nearest-neighbor accuracy for a FT task y converges as dataset size increases to at least the local consistency (see Definition 1) of y over G_{PT}.*

The full proof of Theorem 1 is in the Appendix. Here, we provide an outline of the proof, starting with the definition of local consistency.

Definition 1 (Local Consistency)**.** Let $y : X \to \mathcal{Y}$ be a task over a domain X, and let $G = (V, E)$ be a graph such that $X \subseteq V$. The *local consistency* $\text{LC}_G(y)$ is the probability that a node's label $y(x)$ agrees with the majority of labels of x's neighbors in G:

$$\text{LC}_G(y) = \mathbb{P}\left(y(x) = \operatorname*{argmax}_{c \in \mathcal{Y}} \sum_{x' \in X | (x, x') \in E} \mathbb{1}_{y(x')=c} \right).$$

Note this is closely related to *homophily* [250, 93, 242].

With local consistency defined, proving Theorem 1 takes the following three steps. First, we quantify what conditions yield a valid $\mathcal{L}_{\text{SI}}$ and ℓ^* (*e.g.*,, when approximately-optimal embeddings exist). Second, noting that by definition of a "valid" $\mathcal{L}_{\text{SI}}$ term, we show that G_{PT} is recoverable under a radius-nearest-neighbor algorithm in the latent space as we approach optimality. Third, we conclude that local consistency is reflected in fine-tuned embeddings via nearest-neighbor classifiers. This can be seen by noting that if we use the same distance function d and radius r associated with loss value ℓ^* in our nearest neighbor classifier, our sets of neighbors exactly match the neighbors in G_{PT}. Therefore, local consistency translates directly into realizable accuracy, establishing the guarantee.

5.4.1 Consequences of Theorem 1

We can see the following two consequences of this theoretical result, both of which are stated and proved in Appendix 5.10.5:

1. Pre-training graph G_{PT} defined as a set of disconnected cliques or a star graph makes minimal guarantees about local consistency of the FT task, even as the dataset size increases.

2. Pre-training graph G_{PT} defined as a nearest-neighbor graph on a manifold over which the FT task label is continuous almost everywhere yields a local consistency that approaches unity as the dataset size grows.

The above two properties show why the *richness of pre-training graphs matters.* In particular, as discussed in Section 5.3.3, many existing PT methods can be realized as methods where G_{PT} is given by a set of disconnected cliques or a star graph. Therefore, the extent to which existing PT methods provide theoretical guarantees about the per-sample latent space structure is minimal. As such, moving to new PT methods with richer graphs has the potential to markedly improve performance, as we will demonstrate on synthetic datasets in Section 5.5 and real-world datasets in Section 5.6.

5.5 Empirical Analysis of Theoretical Properties

We can further validate the theoretical analyses of Section 5.4 via semi-synthetic experiments. In particular, we create several datasets of natural language sentences augmented with synthetic graphs with known relationships to certain FT tasks (e.g., low or high local consistency, low or high rates of noise). We then use these datasets to validate three important properties of PT methods: First, do PT methods trained with a $\mathcal{L}_{\text{SI}}$ and G_{PT} yield Nearest-neighbor FT performance in accordance with our theory? In particular, do (a) FT tasks with high local consistency over the PT graph offer better performance, and (b) those with very low local consistency offer worse performance? Second, do PT methods trained with a $\mathcal{L}_{\text{SI}}$ and G_{PT} suffer significantly when pre-training graphs are polluted with noise? Finally, third, do the latent space geometry regularizing properties of $\mathcal{L}_{\text{SI}}$ yield methods whose embeddings more clearly cluster than embeddings produced by traditional pre-training alone?

5.5.1 Datasets and Experimental Setup

While full experimental details are provided in the Appendix, we will first offer a brief commentary on how we create the synthetic datasets in use here.

Pre-training & fine-tuning datasets. Across all experiments, our synthetic datasets consist of free-text sentences[3] labeled with LDA-produced topics used as our FT labels. To test across various kinds of graphs, we produce a number of pre-training graphs per experiment, as detailed below.

Pre-training graphs. We use graphs spanning 3 categories. (1) A graph (CLIQUES) consisting of disconnected cliques, where sentences are linked in the graph if they share the same topic label. (2) Graphs composed of nearest-neighbor graphs defined over simplicial manifolds built using topic probabilities to localize sentences onto simplices. We explore manifolds with a range of topological complexity, including: PLANE, MÖBIUS, SPHERE, and TORUS. Finally, (3) we define several graphs according to a mechanistic process that allows us to control how topic labels relate to graph structure. We build graphs in this setting that have the property that topic labels associate with them in one of three ways: so that topics are maximally conserved within local neighborhoods (NEIGHBORHOOD), by assigning sentences to nodes in the graph such that each graph motif corresponds to a unique topic (MOTIF), and such that node topics are driven by non-local graph structural features, on the basis of graphlet degree vectors (STRUCTURAL). Of all these graphs, we expect that topics will display a low local consistency over the STRUCTURAL graph, a moderately high local consistency over the MOTIF graph (as graph motifs are all connected components), and high local consistency everywhere else.

Experimental setup. To answer our three questions, we will pre-train models under both traditional LM pre-training alone and a new, structure-inducing PT (SIPT) method within our paradigm that augments the loss with a contrastive learning loss

[3] Source: `https://www.kaggle.com/mikeortman/wikipedia-sentences`

over G_{PT}, with $\lambda_{\mathrm{SI}} = 0.1$. Both models use a shallow transformer encoder for $f_{\boldsymbol{\theta}}$ and a character-level tokenization scheme. Final results are reported via the AUROC of 3-nearest-neighbor classifiers over the latent space, per-sample embeddings. In line with our theoretical predictions, we expect to see higher NN FT performance in all settings where the FT task (topic prediction) has high local consistency over the graph G_{PT} (all graphs except STRUCTURAL) and worse performance in the case where the local consistency is very low (STRUCTURAL).

We also assess the stability of our method as the graph G_{PT} is noised using the CLIQUES graph by randomly adding additional edges with varying rates.

5.5.2 Results

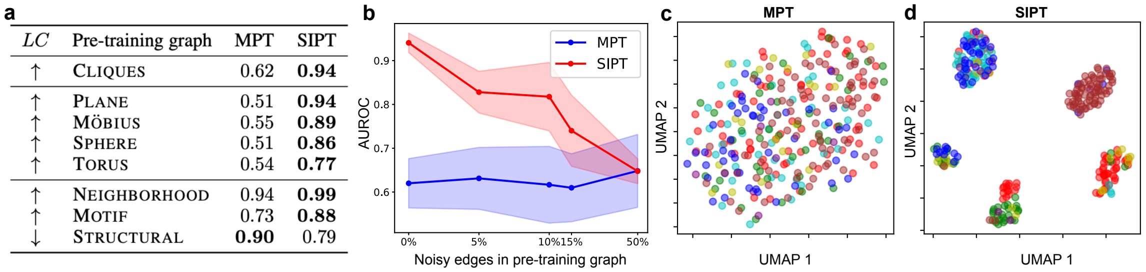
Across all expected settings, we see that our SIPT model, which augments LM PT with the $\mathcal{L}_{\mathrm{SI}}$ objective and graph G_{PT} as motivated by section 5.3.2 offers significant improvements, is robust to noise in the PT graph, and performs in agreement with our theoretical analyses.

SIPT improves performance over LM PT by 0.26 ± 0.13 AUROC on graphs where the topic task has a high local consistency. As can be seen in Figure 5-3a, SIPT offers significant improvements over LM PT in nearest-neighbor FT AUROC across all graph types with strong topic local consistency.

SIPT's empirical results are in agreement with theoretical analyses. In line with our analyses in Section 5.4, SIPT only under-performs LM PT on the STRUCTURAL graph where the topic task (by design) does not have strong local consistency. This validates our theoretical results by showing that local consistency is a strong predictor of Nearest-neighbor FT performance.

SIPT is robust to incomplete and noisy pre-training graphs. Figure 5-3b shows Nearest-neighbor FT AUROC as a function of noise rate on the CLIQUES graph. For up to 15% noise, SIPT shows improvements over LM PT, and even at 50% noise, the two approaches perform comparably.

SIPT pre-trained embeddings show stronger clustering than LM PT embeddings. Figure 5-3c-d shows embeddings produced under the MÖBIUS graph either



LC	Pre-training graph	MPT	SIPT
↑	Cliques	0.62	**0.94**
↑	Plane	0.51	**0.94**
↑	Möbius	0.55	**0.89**
↑	Sphere	0.51	**0.86**
↑	Torus	0.54	**0.77**
↑	Neighborhood	0.94	**0.99**
↑	Motif	0.73	**0.88**
↓	Structural	**0.90**	0.79

Figure 5-3: **(a)** Comparisons between nearest-neighbor FT AUROC (higher is better) of LM PT models and SIPT models over various graphs with various forms of structural alignment. LC indicates the label consistency between FT task and G_{PT} (see Section 5.4). **(b)** nearest-neighbor FT AUROC vs. noise rate. Up to 10% noise SIPT dramatically outperforms LM PT, and at 50% noise, the two approaches are equal. **(c-d)** Embedding space of MPT and SIPT models on the Möbius dataset. Point colors indicate topic labels. SIPT's embedding space reflects the structure of pre-training graph whereas MPT does not.

by LM PT or SIPT, clustered via UMAP into 2 dimensions. It is clear visually from these figures that SIPT embeddings show clear clusters strongly associated with the topic-modelling FT task, whereas LM PT embeddings do not.

5.5.3 Conclusions

From these analyses, we see that augmenting PT with per-sample structure-inducing objectives can both (1) offer significant advantages over existing PT architectures and (2) permit analytical reasoning about over which FT tasks PT will offer improvements. These findings are not surprising; in these semi-synthetic experiments, we designed our graphs explicitly to have either high or low local consistency with respect to our FT task so that we could probe exactly whether SIPT methods would behave in accordance with theory in tightly controlled settings. In this way, the graphs G_{PT} used here may not be reflective of graphs in the real world, which will be chosen more independently of specific FT tasks. To show that the gains observed here persist in more realistic scenarios, we turn next to experiments over diverse real-world datasets with real, FT-task-independent graphs, in Section 5.6.

5.6 Real-World Experiments with SIPT Methods

In Section 5.4, we showed that our new PT framework permits meaningful theoretical analyses of pre-training methods, and that those theoretical findings are validated in semi-synthetic experiments. In this section, we instead demonstrate that we can use our PT framework to design PT methods with rich, per-sample geometric constraints which offer real-world performance benefits on standard benchmarks and meaningful tasks.

In particular, we examine the efficacy of incorporating structural information across three domains: PROTEINS, containing protein sequences; ABSTRACTS containing free-text biomedical abstracts; and NETWORKS, containing sub-graphs of protein-protein interaction (PPI) networks. In each data modality, we use different pre-training datasets, test on different downstream FT tasks, leverage distinct graphs G_{PT}, and compare to baseline per-sample and/or per-token methods. In each case, we show that methods leveraging rich per-sample constraints via a graph G_{PT} as outlined in Section 5.3 will match or exceed the performance of these baselines.

In the remainder of this section, we do the following. First, in Section 5.6.1, we detail how we train models under our framework, and describe the datasets, pre-training graphs, and per-token / per-sample baselines against which we compare. Then, in Section 5.6.2 we will present results of these analyses, including comparisons between new methods and existing per-token and per-sample baselines, qualitative analyses into where structure is most useful, and ablation studies.

5.6.1 Datasets and Experimental Setup

We explore three different domains containing different data modalities in this work. We describe each domain fully in Appendix Section 5.11.2, but all details are also summarized in Table 5.2. Recall that our goal in these experiments is to probe whether or not training PT systems augmented with rich per-sample latent space geometry constraints offers improvements over traditional per-token or per-sample methods. To that end, we perform the following steps.

Baselines. First, for each domain, we identify an appropriate, published per-token and/or per-sample PT baseline. For example, in our PROTEINS domain, we use the TAPE Transformer model [166] (trained via LM) as our per-token baseline, and the PLUS Transformer model [138] (trained via LM plus a supervised, per-sample task) as our per-sample baseline.

PT graphs. Next, we identify a source dataset and graph G_{PT} to train a new structure-inducing pre-training (SIPT) method based on our PT paradigm. We want these graphs to be simultaneously (1) rich sources of per-sample relationships, (2) readily extractable from public data, and (3) of diverse types and forms across data modalities to probe our framework in various circumstances. In our PROTEINS experiments, we use the Tree-of-life dataset [252] and the associated multi-species protein-protein interaction graph as our dataset $\boldsymbol{X}_{\mathrm{PT}}$ and graph G_{PT}, respectively. In this modality, then, our PT system is regularizing the per-sample latent space such that proteins that interact with one another should yield similar embeddings.

Training structure-inducing models. Then, we need to train models leveraging the structure inducing models directly. For our PROTEINS and ABSTRACTS datasets, in order to minimize computational burden, we do not pre-train a model from scratch under our new framework. Instead, we initialize a model from the per-token baseline directly, then perform additional pre-training for only a small number of epochs under the new loss subdivision proposed in Section 5.3. In these settings, we assess both the multi-similarity and contrastive $\mathcal{L}_{\mathrm{SI}}$ variants. On the NETWORKS dataset, we pre-train all models (including baselines) from scratch, and based on early experimental results we only assess the contrastive loss variant. λ_{SI} is chosen on the PROTEINS and ABSTRACTS datasets to maximize the performance on an internal link-retrieval task over G_{PT}. This process suggests that λ_{SI} of 0.1 is a robust setting, and as such 0.1 was used directly for the NETWORKS task. Finally, note that our warm-start procedure allows a powerful ablation study: by additionally training a PT model from the per-token baseline with $\lambda_{\mathrm{SI}} = 0$, we can uniquely assess the impact of the new loss term, rather than simply additional training or the different PT dataset. We perform this ablation study for all applicable datasets.

	PROTEINS	ABSTRACTS	NETWORKS
Data Modality ($\boldsymbol{x}_i$ is a...)	Protein Sequence	Biomedical Paper Abstract	PPI Network Ego-graph
PT Dataset	Tree-of-life [252]	Microsoft Academic Graph [215, 90]	[92]
FT Dataset	TAPE [166]	SciBERT [15]	[92]
Per-token baseline	TAPE [166]	SciBERT [15]	Attribute Masking [92]
Per-sample baseline	PLUS [138]	None	Multi-task learning [92]
G_{PT}: $(\boldsymbol{x}_i, \boldsymbol{x}_j) \in E$ iff	$\boldsymbol{x}_i$ interacts with $\boldsymbol{x}_j$	$\boldsymbol{x}_i$'s paper cites $\boldsymbol{x}_j$'s paper	$\boldsymbol{x}_i$'s central protein agrees on all but 9 Gene Ontology (GO) labels with $\boldsymbol{x}_j$'s central protein.

Table 5.2: A summary of our datasets, tasks, and benchmarks.

FT tasks and evaluation. Lastly, we need to evaluate these pre-training models across FT task benchmarks. In all cases, we use established PT/FT benchmarks, and compare to 10 tasks in total across all 3 modalities. FT models are, naturally, initialized from the pre-trained encoders, and FT training procedures and hyperparameters match those in the published literature, or are fit via minimal hyperparameter searches performed across both baselines and novel methods on validation datasets. In the PROTEINS dataset, for example, we use the TAPE benchmark of 5 FT tasks [166]. All fine-tuning tasks explored across all three of our domains are detailed in Table 5.3. On all FT tasks save TAPE's contact prediction task, which has a pairwise output space and is thus very computationally expensive, FT evaluations were performed over 3 random seeds to assess statistical significance of results.

5.6.2 Results

Incorporating $\mathcal{L}_{\text{SI}}$ performs comparably to or improves over all baselines across all 3 domains and 10 FT tasks. In Table 5.4, we show the relative

FT Dataset	FT Task		Description	Metric
	Name	Abbr.		
TAPE [166]	Remote Homology	RH	A per-sequence classification task to predict protein fold category.	Accuracy
	Secondary structure	SS	A per-token classification task to predict amino acid structural properties.	Accuracy
	Stability	ST	A per-sequence, regression task to predict stability.	Spearman's ρ
	Fluorescence	FL	A per-sequence, regression task to predict fluorescence.	Spearman's ρ
	Contact Prediction	CP	An intra-sequence classification task to predict which pairs of amino acids are in contact in the protein's 3D conformation.	Precision @ $L/5$
SciBERT [15]	Paper Field	PF	A per-sentence classification problem to predict a paper's area of study from its title.	Macro-F1
	SciCite	SC	A per-sentence classification problem to predict citation intent	Macro-F1
	ACL-ARC	AA	A per-sentence classification problem to predict citation intent	Macro-F1
	SciERC	SRE	A per-sentence Relation Extraction task.	Macro-F1
Networks [92]			The multi-label binary classification of 40 gene-ontology term annotations	Macro-AUROC

Table 5.3: Fine-tuning tasks

Domain	Task	Vs. Per-Token PT		vs. Per-Sample	
		RRE	Δ	RRE	Δ
Proteins	RH	**7.0%**$_{\pm 1.2}$	↑	**8.4%**$_{\pm 2.4}$	↑
	FL	-0.8%$_{\pm 1.3}$	∼	**12.8%**$_{\pm 1.1}$	↑
	ST	**13.1%**$_{\pm 2.5}$	↑	2.2%$_{\pm 2.8}$	∼
	SS	**4.5%**$_{\pm 0.2}$	↑	**4.5%**$_{\pm 0.2}$	↑
	CP	**10.5%** *	↑	N/A	
Abstracts	PF	0.3%$_{\pm 0.2}$	∼	N/A	
	SC	2.4%$_{\pm 4.1}$	∼	N/A	
	AA	**17.7%**$_{\pm 6.5}$	↑	N/A	
	SRE	**6.7%**$_{\pm 0.4}$	↑	N/A	
Networks		7.8%$_{\pm 5.2}$	∼	5.1%$_{\pm 2.7}$	↑

Table 5.4: Relative reduction of error (RRE; defined to be $\frac{[\text{baseline error}]-[G_{\text{PT}}\text{ model error}]}{[\text{baseline error}]}$) of models trained under our framework vs. published per-token or per-sample baselines. Higher numbers indicate models under our framework reduce error more, and thus outperform baselines. The Δ column indicates whether the model offers a statistically significant improvement (↑), no significant change (∼), or a statistically significant decrease (↓). Statistical significance is assessed via a t-test at significance level $p < 0.1$. Per-sample analysis and variance estimates for CP were infeasible due to the computational cost of this task.

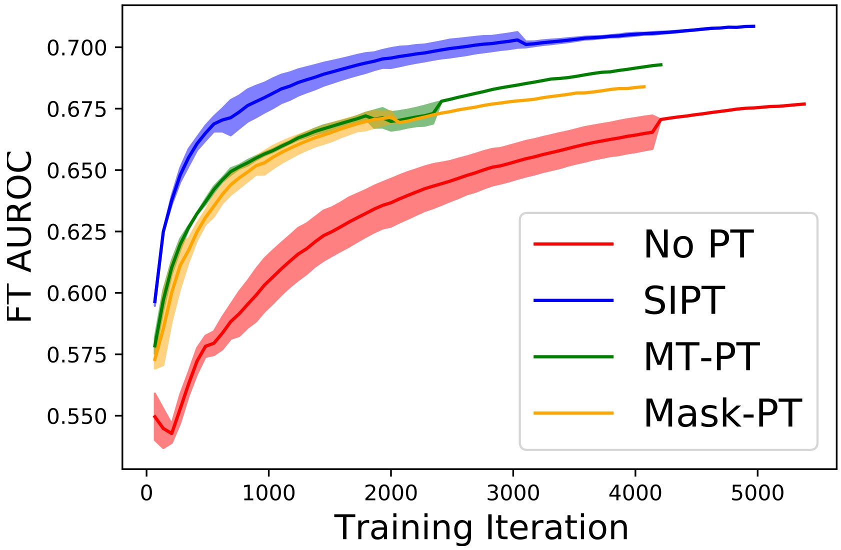


Figure 5-4: FT AUROC as a function of iteration over the NETWORKS dataset. The SIPT method converges faster and obtains better performance than does LM or multi-task PT.

reduction of error[4] of the best performing model leveraging rich structural constraints (*i.e.*,, maximal over the multi-similarity loss or contrastive loss) vs. the per-token or per-sample baselines. We can see that in 10/15 cases, incorporating per-sample latent space regularization improves over existing methods, and in no case does it do worse than either baseline. Furthermore, In some cases, the gains in performance are quite significant, with improvements of approximately 17% (0.05 macro-F1 raw change) on AA, 6% on SRE (0.01 macro-F1 raw change), and 4% on RH (2% accuracy raw change). We further establish a new SOTA on AA and RH, and match SOTA on FL, ST, & PF.

We also see in Figure 5-4 how performance evolves over FT iterations for the NETWORKS dataset, to determine if the improvements observed at the final converged values are present throughout training. We see that methods using per-sample structure inducing losses converge faster to better performance than do other methods.

Note that raw results across all settings are presented in Appendix Section 5.12.1, Tables 5.7,5.8.

[4] $\text{RRE} = \frac{(1-\text{baseline})-(1-\text{SIPT})}{(1-\text{baseline})}$. RRE is unitless and suitable for comparing across tasks.

These performance gains are present across diverse modalities and $G_{\mathbf{PT}}$s. Note that these performance gains persist over all three data modalities and all different G_{PT} types we use here. This shows that explicitly regularizing the per-sample latent space geometry offers value across NLP, non-language sequences, and networks domains, as well as while leveraging graphs including those defined by external knowledge, those inferred from self-supervised signals in the data directly, and those defined by nearest-neighbor methods over multi-task label spaces.

Observed performance improvements are consistent with theory. Hyperparameter tuning finds that far less structure-inducing is necessary on our ABSTRACTS dataset ($\lambda_{\text{SI}} = 0.01$) than on our PROTEINS dataset ($\lambda_{\text{SI}} = 0.1$). This agrees with our guiding hypothesis that per-sample latent space regularization is much more necessary on non-NLP domains than it is on NLP domains. Further details on these analyses are in the Appendix, Section 5.12.2

Observed gains are uniquely attributable to the novel loss $\mathcal{L}_{\text{SI}}$. As outlined in Section 5.6.1, we also perform ablation studies to determine how much of the observed gains in Table 5.4 are due to the novel loss component, as opposed to, for example, continued training, new PT data, or the batch selection procedures used in our method which also indirectly leverage the knowledge inherent in G_{PT}. Unsurprisingly, there are some gains observed due to these other factors, and performance gains shrink as compared to continued training of per-token models alone over the new PT dataset. However, these gains do not change the direction of the observed relationships, and more generally comparing against the maximal performance baseline overall, including the ablation study, does not effect the statistically significant relationships at all at $p < 0.1$. Full ablation study results, along with all other raw results, can be found in Appendix Section 5.12.1, Tables 5.7,5.8.

5.7 Discussion

Here, we provide unifying commentary and emphasize key strengths of our work. In particular, we crystallize the distinction between enforcing per-token and per-sample latent space structure and highlight why the distinction is essential via theory and experiments. We then offer a new framework for understanding and developing PT methods. This framework permits a natural vehicle to associate PT task structure with FT performance, which is especially relevant in non-NLP domains. Finally, we identify the question of effective modeling of per-sample structure as a fruitful direction for future PT method development. This direction is empirically illustrated in Section 5.6, where we show that taking into account richer notions of per-sample structure consistently improves existing methods.

Per-token vs. per-sample latent space structure. Table 5.1 shows how a battery of existing LMPT methods leverage and enforce structure in their learned embeddings. We find an apparent lack of attempts to enrich the latent structure at a *per-sample* level. In NLP, this discrepancy is not so problematic because many downstream tasks can be realized as per-token tasks and implemented using prompting. However, prompting is unlikely to offer such advantages in non-NLP domains, such as protein sequences, graphs, time series, and tabular data. For these reasons, we conclude that leveraging the per-sample structure is critical. Our experiments show that these differences matter in practice. On PROTEINS, ABSTRACTS, and NETWORKS, we show that augmenting LMPT methods with per-sample structural regularization via $\mathcal{L}_{\text{SI}}$ defined on a variety of pre-training graphs can consistently improve performance on downstream tasks.

Alignment of PT task structure to FT performance. Through a simple modification of established PT problem formulation, we can realize a learning framework in which PT task structure is directly related to FT task suitability (as measured by nearest-neighbor accuracy in the PT latent space). This framework allows us to sidestep the traditional theoretical difficulties identifying when and why PT can offer advantages for FT [182]. Our synthetic experiments confirm this theory and establish that metric-learning motivated structure-inducing losses can be resilient to graph noise.

Future research directions. Our work highlights novel directions for research relating to how we can best pre-train models to capture per-sample latent space structure. Are there novel losses better suited than metric learning losses used here for pre-training graphs—*e.g.*,, can we leverage the graph-distance alongside the intra-batch distance to develop improved negative sampling strategies? What adaptations of our restrictions allow a PT system to simultaneously capture multiple graphs (or multiple edge types) through embeddings? Can we (and do we ever need to) reflect forms of structure beyond nearest neighbor relationships used in this work—*e.g.*,, such as by leveraging higher-order topological considerations or by matching a distance function rather than a discrete graph? Can we refine theoretical results to consider better the convergence properties of trained models and understand when and how to effectively converge to solutions that recover G_{PT}? Our PT framework prompts all these questions and more.

5.8 Related Work

This paper builds upon a wealth of previous research at the intersection of PT/FT, neural language models, graph representation learning, and theoretical analysis of metric learning.

5.8.1 Pre-Training Strategies With Explicit Consideration of Geometry

Per-token latent space. As outlined in Section 5.2 and Table 5.1, a significant amount of research in pre-training language models explored ways to incorporate structure into the per-token latent space. For example, prior work considered selectively masking named entities [202, 231, 204, 205, 203, 247, 82, 198], aligning per-token representations to KG embeddings [160, 245], performing named entity recognition [156, 239], and producing joint token-entity embeddings [247, 156, 238, 245, 239, 198].

Per-sample latent space. As outlined in Section 5.2 and Table 5.1, incorporating structure at the per-sample level in language model pre-training remains under-explored or neglected research area. The most common approach by far is to leverage additional classification tasks at the per-sample level [47, 107, 247, 82, 245, 198, 138, 119, 205, 203, 129, 92]. Further, KEPLER [221] aligns per-sample embeddings of free-text descriptions of entities to KG embeddings of those entities to match trans-E [143].

Methods with geometrically motivated auxiliary FT/pre-FT stages. Beyond the methods discussed above and in Section 5.2 and Table 5.1, there are also a number of works that do not explicitly change the PT stage of language-model pre-training. Instead, those methods add additional stages between PT and FT to impose geometric constraints at either a per-token or per-sample level. This includes studies that add "knowledge adapters" to incorporate KG embeddings into a pre-trained language model at a per-token level [216, 137, 124], adapting input text or embeddings to incorporate KG information [118, 56, 232, 157], or adapting sample embeddings via metric learning techniques based on either self-supervised spatial tasks or on FT tasks [170, 74].

5.8.2 Pre-Training Strategies With No Explicit Consideration of Geometry

Pre-training using language models has been extensively studied in NLP. To this end, researchers have explored autoregressive approaches [23, 161, 162, 22], masked language modelling approaches [154, 121], and discriminative modelling approaches [35]. These methods have also been adapted to non-NLP domains where masked imputation can still be employed, including networks [92, 217], tabular data [236], time series [129], and protein sequences [166].

5.8.3 Pre-Training Theory

Prior theoretical analyses found that pre-trained language models reflect properties consistent with domain knowledge [37, 210, 168]. Additionally, [108] suggests that masked-based PT transfers positively to downstream tasks that obey certain conditional independence relationships with PT learning. Further, [182] posits that masked-based PT works because one can re-frame most NLP tasks as language modelling tasks.

5.8.4 Graph Representation Learning

Attributed graph embedding and knowledge graphs. Our work reflects topics in deep attributed graph embedding (DAGE), in which an embedding model f is learned for a graph G equipped with node attributes $\boldsymbol{X}$ such that the connectivity structure of G is reflected in the image of f [62, 40, 112]. DAGE algorithms are typically not adaptable to contexts in which the graph is not known at inference time, and so it does not directly apply to our setting.

Graph neural networks (GNNs). There is also a wide variety of research on graph neural networks [111], including, *e.g.*,, work on semi-supervised graph classification [101], inductive representation learning on graphs [78], and many others. While relevant to our study in principle, these methods are not directly applicable in either the original or our revised PT setting. As we state in Section 5.3, although we take as input the graph G_{PT}, this cannot be processed with a standard GNN. This is because we still want to apply the encoder $f_{\boldsymbol{\theta}}$ at fine-tuning time to contexts with no associated graph, using only the input $\boldsymbol{x}$ itself (*e.g.*,, the sentence or protein sequence).

Structure-preserving metric learning. A sub-field within metric learning that our analyses rely on is structure-preserving metric learning (SPML) [214, 189, 188]. In SPML, rather than using supervised labels, a graph G defines when samples should be close together, such that G can be recovered in the embedding space via a connectivity algorithm [188].

5.9 Conclusion

This work provides a novel perspective on the established pre-training (PT)/fine-tuning (FT) learning framework. Under our formulation, the PT loss is subdivided into two components: one which is designed to capture per-token, *i.e.*,, intra-sample, relationships, and one which is designed to regularize the per-sample latent space to capture relationships between samples given by a user-specified pre-training graph G_{PT}. To show that this new PT perspective offers considerable value, we carry out a theoretical analysis of existing PT methods and perform comprehensive empirical analysis on semi-synthetic and real-world datasets spanning three modalities and 10 FT tasks. Our results provide new insights into the shortcomings of existing methods and establish a theoretical connection between the PT task formulation and FT task performance. Further, we illustrate that new methods designed to follow our formulation can outperform existing per-sample and per-token PT/FT methods. We envision that this work can provide a methodological pathway for future research on pre-training.

5.9.1 Relation to this Thesis

In this chapter, we bring together the various pieces explored in prior chapters into a unified framework (Structure Inducing Pre-training; SIPT) for leveraging external knowledge to induce global structure in the latent space of pre-training models. Much like in Chapter 2, SIPT leverages imposing global structural constraints on model latent spaces during training, and via the pre-training paradigm makes significant use of unlabeled data. Building on Chapter 3, the various PT encoders in SIPT each individually make use of local structure in various ways, be it through sequential transformer models or local graph convolutional neural network models. Finally, SIPT is a direct response to the observations of Chapter 4, where we showed that traditional pre-training algorithms do not yield the same quality of benefits over the clinical domain as they do in natural language processing, and further showed that this deficit is likely due specifically to how these algorithms do (or fail to) encode per-sample structure. Ultimately, we show that the SIPT approach is particularly impactful in clinical and biomedical domains, and provide extensive theoretical and empirical justification for the SIPT framework.

5.10 Appendix

5.10.1 Code and Data Availability

Our code, synthetic datasets, and pointers to real-world datasets are available here: `https://github.com/mmcdermott/structure_inducing_pre-training`

5.10.2 Existing Language Model Pre-training Methods - Free-text Descriptions

In this section, we describe all of the 27 models featured in Table 5.1, highlighting on which domain they operate and other key details of their approach. We will break these models down into categories, and also include some additional models in this section not included in Table 5.1 that do not fit into the paradigm we explore here.

Language modelling alone.

[121] General domain NLP; RoBERTa includes only a masked language modelling objective.

[161, 162, 22] General domain NLP; The GPT series of models use autoregressive language modelling alone, and focus on generative language tasks, not general PT/FT, though GPT-III does show that by reframing many classical NLP fine-tuning tasks as generative language tasks, GPT-III can still offer a compelling zero and few-shot solution to these tasks using only the pre-trained embedder [22].

[166] Protein sequences; the TAPE benchmark profiles various language-model based PT models against a benchmark of PT/FT tasks.

[246] Molecular Graphs; Molecular Graph BERT (MG-BERT; no relation to MG-BERT [13]) uses masked atom prediction to pre-train a GNN over molecular graphs.

Language modelling & Per-token KG Integration.

[204] General domain NLP; ERNIE 1 augments traditional MLM with entity-specific masking (e.g., masking the word "Mozart" from the sentence "Mozart was a musician") to force the model to recover common-sense knowledge about named entities.

[82] General domain NLP; KgPLM adapts the discriminative training ideas of ELECTRA [35] alongside the idea of entity masking explored previously. They perform both entity masking and a discriminative loss identifying which tokens were replaced focused on entity replacements.

[160] General domain NLP; ERICA presents a mechanism for leveraging contrastive learning and distant supervision to incorporate external knowledge into a PLM for improving language understanding. They augment MLM with two per-token tasks aimed at ensuring the per-token representations within a document reflect the structure of the KG; first by ensuring that the pooled representations of head and tail entities are similar conditioned on a relation (which is prepended to the document prior to embedding) and second that relation embeddings (defined as concatenated head, tail per-token entity embeddings) are similar within and across documents. As both tasks are done on per-token embeddings, and never at a per-sample level, this approach induces minimal constraints on the per-sample latent space.

[239] Biomedical domain NLP; KeBioLM integrates a per-token KG into a biomedical language model by augmenting token entity representations with attention look-ups into a biomedical KG (regardless of whether the attended to entities actually match a given entity mention in the source text, though they do only apply this on recognized entities). In order to ensure this attention is meaningful, they further perform named entity linking and recognition auxiliary PT objectives, leveraging the same KG embeddings used during the attention calculation (thereby incentivizing per-token representations to be similar to their associated entities representations thus ensuring that those entities are reflected in the attention over the KG, modulo the fact that the loss is applied after the final layer, rather than the layer directly used to do the search). KG embeddings are initialized using Trans-E [143]. Their usage of automatically attending over entities within their language model (without explicit constraints on those matches) is motivated by [57]'s work in [57] and has similarities to Know-BERT [156].

[231] General domain NLP; LUKE performs pre-training using MLM and an entity-specific masking/recognition scheme that is a slight variation on the traditional entity-specific masking [204] proposed. At FT time, they have other knowledge-specific integrations, including specialized query matrices in KQV attention based on attending to either traditional tokens or entities, but at PT time their only modulation over a ROBERTA [121] baseline is their entity masking task.

[156] General domain NLP; Know-BERT integrates per-token entity information into an MLM pre-training scheme by performing unconstrained attention over a per-entity knowledge graph (only on pre-identified candidate entity spans), alongside any available entity linking supervision information via direct Named Entity Linking. This has similarities with [239] and [57].

[202] General domain NLP; COLAKE performs a priori entity linking on the source text, then replaces per-token mentions with entity embeddings, and appends to the input text sub-graphs from a (relational) knowledge graph, including both neighboring mentions and relations in the augmented input text block. This input is then encoded via a variant of a transformer which limits attention flow between tokens of different types and trains the entire ensemble with masked language, entity, and relation modelling.

Language modelling, Per-token KG Integration, & Supervised Classification.

[205, 203] General domain NLP; ERNIE 2.0 & 3.0 augments traditional MLM with entity-specific masking (e.g., masking the word "Mozart" from the sentence

"Mozart was a musician") as well as a multi-task per-sample task, largely motivated at classifying a block of text on the basis of internal text cohesion (predict the true order of the sentences within an input sample & identify whether the sentences within the input sample are spatial neighbors, come from the same document, or come from different documents). ERNIE 3.0 additionally augments pre-training with a per-token relation-embedding task using cloze-filling as a vehicle to perform relation extraction on pre-specified per-token KGs.

[247] General domain NLP; ERNIE (no relation to [204, 205]) uses both architectural and objective-function changes to inject per-token knowledge into PT. Specifically, they separately embed all named entities in a sample, use a specific architecture to join contextualized entity embeddings alongside the embeddings of tokens that realize that entity in the span, and perform entity-specific masking (through both entity replacement and entity-token relationship masking). In addition, they simultaneously perform standard MLM and next-sentence prediction in the manner of BERT [47].

[13] General domain NLP; MG-BERT introduces a GCNN layer after BERT token which aggregates token embeddings together over a unified graph consisting both of co-occurrence relationships and knowledge graph relationships. Note that as the KG layer is used during pre-training as well, this model may have difficulty adapting to settings at FT time where the KG is unavailable, which limits flexibility.

[238] General domain NLP; JAKET embeds entities by extracting per-token representations of entity texts inside per-entity descriptions, then produces updated KG embeddings via a graph convolutional neural network. Those embeddings are then fed back into a language model alongside per-token embeddings corresponding to those entities in the raw text. The entire system is trained according to an MLM objective, plus entity category prediction and relation prediction (only on the entity embeddings extracted from entity descriptions and fed through the GCNN—*not* on the raw entities within the contextualized text). Notably, unlike many other methods, this approach does leverage relations in the formation of individual per-sample and per-token embeddings, so can't be used at inference time without that level of information as well.

[198] General domain NLP; Coke is similar to ERNIE [247] and JAKET [238] in that it aggregates entity information by leveraging a GCNN over a restricted dynamic context KG based on token-entity mentions then integrates those augmented embeddings into the per-token embeddings of a BERT-style pretrained model (similar to JAKET), but also leverages the denoising entity autoencoder task of ERNIE [247]. By leveraging a GCNN over entity relationships to directly inform per-token embeddings, relations must be known about per-token entities at FT time as well.

[245] Medical domain NLP; SMedBERT leverages a complex, multi-faceted loss including MLM, Sentence-order prediction SOP (as introduced in, e.g., ALBERT [107]), and includes per-token KG information by aggregating token embeddings across KG embeddings (produced via trans-H [223]) corresponding to matching entities and the neighbors of matching entities in the KG. They also include variations on relation and entity masking to ensure the PT model learns per-token information corresponding to the KG. This method bares similarity to Coke [198] and JAKET [238].

Language modelling & Single-task Classification.

[47] General domain NLP; Masked language model plus binary classification of whether text block is sequentially consistent, with samples chosen via true positive pairs vs. randomly joined sentences.

[107] General domain NLP; Masked language model plus binary classification of whether text block is sequentially consistent, with samples chosen via true positive pairs vs. re-ordered positive sentence pairs.

[138] Protein sequences; Masked language model plus multi-class classification of to which protein family an input sequence belongs. Uses non-standard whole-sequence embedding procedure (no `[CLS]` token).

Language modelling & Multi-task Classification.

[119] General domain NLP; Masked language model plus multi-task classification across a variety of NLP tasks.

[92] Graph data; This model uses a masked imputation task similar to a masked langauge model and a highly multi-task supervised whole-graph level prediction. On this non-NLP domain, [92] finds that the multi-task whole-graph level task is essential for performance.

[129] EHR Timeseries data; This model uses a masked imputation task similar to a masked language model over timeseries data and a multi-task supervised whole-sequence prediction task. On this non-NLP domain, [129] finds that the multi-task whole-sequence level task is essential for performance.

Language modelling & Whole-sample KG Constraints via Entity-descriptions.

[221] General domain NLP; KEPLER leverages the most complex inter-sample geometric constraint observed in existing methods. It augments traditional MLM on generic text samples with a constraint ensuring the (per-sample) embeddings of entity descriptions pulled from pre-specified knowledge graphs (KGs) reflect geometric constraints in accordance with the underlying relations in the KG (in particular leveraging the [206] geometric constraints). As we will see in our theoretical analyses, these constraints are much more restrictive on the latent space geometry, and thus imply a greater encoding of domain knowledge in the model. Note that JAKET [238] also leverages entity descriptions in its per-token encoding, but these descriptions are (1) extracted via per-token embeddings, using the first mention of the token, not whole-sample embeddings, and (2) integrated back into the original text in a per-token manner, not optimized over directly via geometric constraints as in KEPLER.

Methods not natively captured in our framework as they leverage relation information during encoding directly.

[238] As outlined above, JAKET isn't natively captured in our framework as it actively leverages the relationships in a KG to produce KG-contextualized per-token entity embeddings during processing. Thus, this system cannot be fine-tuned on KG-free data without significant domain shift, unlike the PT methods we focus on.

[198] CokeBERT leverages both entities and relatonships, summarizing KG-contextualized embeddings at a per-token level. Similar to JAKET and Graphformer, the use of relationships here means that at FT time, if the per-token knowledge cannot be embedded similar to at PT time, the model will face a domain shift burden.

[245] Given SMedBERT leverages (per-token) relational information in producing unified token embeddings, it is not natively realizable within our framework which assumes that the input to the model is the raw data domain.

[203] By leveraging relational data via their relational masking task, ERNIE3 may also have limited generalizability to other contexts.

[202] Given COLAKE leverages (per-token) relational information to construct the augmented inputs to the model, it requires per-token relational data about the source inputs at both PT and FT time.

Methods orthogonal to our framework.

[120] KG-BART is a text-generation model that leverages per-token knowledge after a text-encoder to enrich generated text with information from a textual knowledge graph (in a per-token manner). It is neither used for general pre-training nor does it leverage any additional per-sample constraints.

[233] Text-based Knowledge Graphs; This work produces embeddings of nodes in KGs by combining transformer based text encodings with graph convolutional network KG embedding methods, leveraging link prediction as the pre-training task. The individual nodes are represented by entity descriptions / textual features, and link prediction can be seen as inducing a geometric constraint via the connectivity of the knowledge graph on whole-sample embeddings. However, given that relationships are used in encoding the data as well, GraphFormer can't be used in a context where KG links may not be observed at FT time and should be seen not as a general text PT method but instead simply as an advanced KG embedding mechanism, so it does not directly fall under our framework.

[2] KeLM (unrelated to KELM [124]) is a method for converting a free-text KG into textual nodes so language modelling can be used over that corpus, and is orthogonal to the methods of pre-training.

[171] This paper is a method for populating a KG from free-text via BERT; it has no bearing on incorporating structure or knowledge into PT itself and as such is not relevant to our system.

[244] This paper presents a method to drop redundant triples from a knowledge graph and a regularization technique to limit the impact of added irrelevant knowledge to per-token knowledge-enhanced PT methods such as ERNIE [247].

[235] Text-based Knowledge Grpahs; KG-BERT is a method for which knowledge graph completion in which textual representations of entities and relations in KGs are embedded by fine-tuning a pre-trained BERT style transformer for link prediction over a given KG. As this is only for knowledge graph completion, it is orthogonal to our study of pre-trained models in general.

Methods that only change things at FT time.

[137] Biomedical domain NLP; MOP doesn't change anything at PT time, but further trains sub-KG adapters on entity recognition tasks prior to FT to infuse entity knowledge into the PT system. This is also still per-token. This is similar in spirit to the K-Adapters work which introduced the notion of KG adapters [216].

[118] General domain NLP; K-BERT, at PT time, is actually equivalent to BERT [47]. However, it does do other, interesting things at FT time, including augmenting the sentence flow with injected per-token knowledge graphs and limiting self-attention to only flow along links supported either by the original sentence or the injected knowledge. But, as this is only true at FT time, at PT time it is equivalent to BERT.

[170] General domain NLP; This model, at PT time, is also actually equivalent to BERT [47]. Like [118], however, it specializes a fine-tuning procedure for sentence information retrieval tasks, which is actually similar to how PT is adapted in this framework.

[124] General domain NLP; KELM doesn't modify PT objective explicitly, but insteads enhances a model at FT time by injecting per-token knowledge via a GNN module atop the pre-trained LM embeddings via a unified text-entity graph. It is similar to KBERT [118] in this way, but resolves other issues with that approach relating to knowledge ambiguity and by supporting multi-hop reasoning, again over the per-token embeddings.

[56] General domain NLP; KI-BERT augments BERT with KG specific information via joint token-entity embeddings and information fusion, but does this only at FT time.

[232] General domain NLP; K-XLNet introduces a secondary FT stage in which knowledge injectors throughout an XL-Net architecture are further trained to leverage knowledge (encoded via free-text entity descriptions) that is injected into input sentences alongside matched tokens. It does not modify the XL-Net PT stage at all.

[216] General domain NLP; K-Adapter proposes to also pre-train various knowledge adapters that can be used alongside a pre-trained language model at fine-tuning time. Thus, while there is a pre-training process for the adapters, this process does not modulate the original pre-trained language model at all. In addition, both adapters pre-trained in this work are based on per-token knowledge graphs; one leverages concatenated entity embeddings to perform relation classification, another predicts which token in the sentence is the "head" in a dependency parse tree, so no per-sample constraints are applied.

[157] General domain NLP; E-BERT injects per-token knowledge into BERT by first aligning embeddings of a knowledge grpah with the input wordpiece embedding space of a (fixed, pre-trained) BERT model, then using various strategies to input them alongside their source mentions in FT text. They do no additional pre-training of this system, and so this model only affects the model at FT time.

[74] General domain NLP; [74] augment LMPT systems with an additional, pre-FT procedure in which the model is further trained using a supervised, per-sample metric learning task leveraging FT labels directly to form the classes used for metric learning. They do not materially change the task-independent PT procedure at all, though their FT metric learning procedure does induce some structure at the per-sample level.

5.10.3 Language Model Pre-training on Natural Language vs. General Domains

In this section, we explore the hypothesis introduced in Section 5.2.2 more fully; namely, that language model (LM) Pre-training (PT) methods have limited ability to reflect inter-sample relationships in domains outside of NLP. We break this hypothesis down into 3 claims:

1. LM PT natively captures *intra-sample* relationships.

2. LM PT is *not* guaranteed to capture inter-sample information in general domains.

3. LM PT (in particular, language modelling or masked language modelling at a per-word level) does capture inter-sample information in NLP, via a mechanism that is unlikely to generalize to other domains.

In the remainder of this section, we first define “intra-” vs. “inter-” sample relationships formally, then expand each of these 3 claims to argue our overall hypothesis. Establishing each of these claims will ultimately suggest that just because LM PT methods work well in NLP does not mean they do not need modification to better capture inter-sample associations on more general domains.

Intra- vs. inter-sample relationships. We use the terms *sample* to refer to an individual datapoint in a dataset. *Intra-sample* relationships thus refer to information that helps model one part of a single sample vs. other parts of that same sample. *Inter-sample*, on the other hand, reflects information that models an entire sample vs. other samples in the same dataset. For example, if we have a dataset of sentences such as in a NLP context, then (1) a *sample* refers to a single sentence, (2) masked language modelling, where we occlude a word in the sentence and try to infer it from the rest of the words in the sentence is an *intra-sample* modelling task, as it relates the occluded word to the observed words within a single sample, and (3) sentence classification, natural language inference, or question answering tasks would all refer to *inter-sample* tasks, as they relate entire sentences either to each other either directly or indirectly.

(1) LM PT-derived PT natively captures *intra-sample* relationships. This claim is clear from the definitions above—any form of intra-sample imputation is directly formulating an intra-sample task. While we do not claim that LM PT will necessarily capture every intra-sample association (*e.g.*,, most LM PT methods will have some limit on how much of the input they are allowed to mask at once, which precludes the model from learning to impute given very sparse input features), it is clear that it naturally reflects a rich subset of intra-sample associations natively.

(2) LM PT-derived PT does not need to capture *inter-sample* relationships.. Though it is clear that LM PT-derived PT natively captures intra-sample relationships, this does not preclude it in general from capturing inter-sample relationships. However, for any style of LM PT-derived PT, we can quickly design a synthetic dataset that shows that inter-sample relationships need not be reflected. For example given a dataset $\boldsymbol{X}$, simply appending an extra, categorical feature to each sample in $\boldsymbol{X}$ that has no dependence on the existing features in $\boldsymbol{X}$ induces an inter-sequence relation between elements of $\boldsymbol{X}$ (sharing the same appended feature) that is *not* capture-able by imputation PT (because the appended feature, having no dependence on the other features, is neither impute-able nor informative in imputing other tokens, and thus is not relevant to the imputation task).

LM PT does capture inter-sample information in NLP, but is unlikely to in other real-world domains. The fact that LM PT-derived PT is not guaranteed to capture inter-sequence information in general, yet simultaneously is very effective as a PT strategy on NLP suggests that in either NLP-specifically or real-world domains in general LM PT-derived PT actually will regardless reflect inter-sample information. Here, we argue that this property holds only for NLP-specifically, due to unique properties of NLP that allow one to consistently re-formulate inter-sample tasks as intra-sample tasks, which is not possible in other domains. This reformulation argument has received significant attention in the literature. In particular, [181] shows through both theoretical and empirical analysis how LM PT performance on inter-sample tasks can be directly related to their intra-sample performance through this reformulation argument presented above. More indirectly, [23] has shown that language models are effective few-shot learners, even in inter-sample tasks. Concretely, they show that GPT-3 achieves strong performance on question-answering tasks even in the *zero-shot* setting—thus showing clearly that LM PT (here, in particular, autoregressive language-model pre-training) explicitly captures the inter-sample task of question-answering simply through language modeling alone. [163] has also leveraged this reformulation trick explicitly to form a unified PT objective to great effect.

Critical to this argument is the fact that not only is it possible to perform this reformulation, but that these reformulated examples qualify both as (1) valid NLP samples themselves and (2) sufficiently reasonable NLP samples that they may be observed in large PT datasets. For example, the reformulation trick leveraged for the "review sentiment analysis" task in [182] is to append the tokens ":)" after the end of positive-sentiment reviews and ":(" after negative-sentiment reviews. This addition of an emoji indicating sentence sentiment is very plausible in real-world datasets.

Other domains do not permit this inter- to intra-sample reformulation, at least not while maintaining validity and plausibility. For example, protein sequences do not permit arbitrary modifications to amino acid sequence to still remain valid, plausible proteins, and even if they did there is no way to naturally encode the outcome to all inter-sample tasks of interest. Thus, this argument suggests strongly that the efficacy of LM PT for inter-sample tasks on NLP is not guaranteed, and possibly not even likely, to extend to other domains of interest.

5.10.4 Further Theoretical Analysis of SIPT

We proceed with additional theoretical analysis of SIPT and formal proofs of the claims made in the main paper. Note that throughout all results, we will assume that the PT and FT datasets are iid, that FT tasks, though they may be unobserved over PT samples, are well defined over the entire PT and FT domain and thus true labels (if unknown) do exist for PT samples, and that the sampling distribution of the PT/FT data has full support over the label-space of any considered task.

How do we define "optimal" embeddings?

Let $G = (V, E)$ be an arbitrary graph, $d_e \in \mathbb{N}$ be an embedding dimension, $\mathcal{D}$ be a distance function.

Definition 2 (SI-optimal embedding)**.** We will say that there exists a SI-optimal embedding of G if there exists a positive *radius* $r \in \mathbb{R}^+$ and mapping $f : V \to \mathbb{R}^{d_e}$ such that $\mathcal{D}(f(v_1), f(v_2)) < c$ if and only if $(v_1, v_2) \in E$.

Note that this is a specialization of the idea that we can leverage a connectivity algorithm to recover G from an optimal embedding; in particular, here we take the connectivity algorithm used to be a radius nearest neighbor algorithm with distance $\mathcal{D}$. Note that any SI-optimal embedding also yields a *margin* parameter $m = \min_{(v_i,v_j)\notin E}(D(f(v_i), f(v_j)) - \max_{(v_i,v_j)\in E}(D(f(v_i), f(v_j))$ where $m \in \mathbb{R}^+$ given G is finite and f is an optimal embedding.

SI-optimal embeddings determine SIPT-optimal embeddings

Suppose we have PT dataset $\boldsymbol{X}_{\text{PT}}$, G_{PT}. We will consider some encoder f optimal (and its embeddings optimal as well) provided that f produces optimal embeddings both according to the LM PT objective and the SI objective. Note that this definition is not contradictory—the LM PT objective does not constrain the global embedding space geometry of f at all, and the SI objective is focused exclusively on the overall geometry. Given our primary interest in this work is on the properties of the SI loss, we will operate from the perspective of only considering that loss.

When do optimal embeddings exist?

The study of when optimal embeddings exist for SIPT problems covers many topics of existing interest within graph embedding, metric learning, and other fields of machine learning. Various results are already known about when optimal embeddings will exist for certain kinds of graphs; however, in full generality typically $d_e \in \Theta(|V|)$ may be required. We will highlight several results here that will be of particular relevance for our purposes. Note that, throughout, we will assume that G_{PT} is *fully inferrable* from $\boldsymbol{X}_{\text{PT}}$—in other words, there exists a function $f : \mathcal{X} \times \mathcal{X} \to \{0, 1\}$ that correctly predicts whether or not there will be an edge between two nodes $\boldsymbol{x}$ and $\boldsymbol{x}'$. In the case that this is not possible, then there will not exist optimal embeddings. Our focus in the rest of this section focuses around when, given that relationship exists, we will be able to efficiently (relative to the necessary embedding dimension d_e) produce such embeddings given various graph structures.

Disconnected Cliques. If the graph G_{PT} contains a series of disconnected cliques, then it is trivial to see that optimal embeddings do exist, as we can simply assign each clique a single point and map all elements within that clique to that single point. This embedding schema works in arbitrary dimensions, and enforces minimal geometry over the latent space.

Manifolds. Let $\mathcal{M}$ be a manifold of dimension N embedded in $\mathbb{R}^{2N}$, where such an embedding is guaranteed to exist by the Whitney Embedding Theorem [226, 144].

Note that for any point $\boldsymbol{x} \in \mathcal{M}$, with its corresponding embedding $\pi(\boldsymbol{x}) \in \mathbb{R}^{2N}$, there exists a radius $r_{\boldsymbol{x}} \in \mathbb{R}^+$ such that the ball of radius r around $\pi(\boldsymbol{x})$ in $\mathbb{R}^{2N}$ intersection $B_{r_{\boldsymbol{x}}}(\pi(\boldsymbol{x})) \cap \pi(\mathcal{M})$ is isomorphic both to some ball around $\boldsymbol{x}$ in $\mathcal{M}$ and to a ball of radius r' in the copy of $\mathbb{R}^N$ isomorphic to the neighborhood of $\boldsymbol{x}$ on $\mathcal{M}$. Let us call the manifold $\mathcal{M}$ valid if $\inf_{\boldsymbol{x} \in \mathcal{M}}(r_{\boldsymbol{x}}) > 0$.

Theorem 2. *Let $X_{PT} \in \mathcal{X}^{N_{PT}}$ be our PT dataset as usual, and let $\mathcal{M}$ be a valid manifold. Suppose there exists some (potentially unknown) isomorphism $f : \mathcal{X} \to \mathcal{M}$. If we then set G_{PT} to be a radius nearest-neighbor graph over $\mathcal{M}$, then there exist SIPT-optimal embeddings with as few as $d_e = 2d_m$ dimensions.*

Proof. This a direct consequence of the definition (above) of a *valid* manifold. As $r^* = \inf_{\boldsymbol{x} \in \mathcal{M}} r_{\boldsymbol{x}} > 0$, we know that we can select some strictly positive $c \leq r^*$ such that all neighborhoods within c of any point in the embedding of the manifold only intersect the manifold in a manner that preserves geodesic distance. Then, on the basis of the Whitney embedding theorem, SIPT-optimal embeddings do exist. □

Proofs for Section 5.4

Theorem 3. *Let $\boldsymbol{X}_{PT}$, G_{PT} be a PT dataset and graph, respectively, such that SIPT-optimal embeddings exist and are realized by an embedder f. Then, under embedding f and given sufficient data, the nearest-neighbor accuracy for a FT task y will converge as dataset size increases to* at least *the local consistency of y over G_{PT}.*

Proof. Given f realizes SIPT-optimal embeddings, we know that if we define a r-NN predictor via the same radius r^* at which f achieves optimality, then this predictor will be correct exactly as often as the label of a given node in the graph G_{PT} agrees with the labels of its neighbors—which is precisely $\text{LC}_{G_{\text{PT}}}(y)$. This classifier may not be well defined for small FT dataset sizes, however, as if data is not sufficiently dense there may be no data-points within radius r of a given query. Similarly, without sufficient PT data then the LC computed over the empirical distribution of the graph G_{PT} may be a poor proxy for the true distribution. As PT and FT dataset sizes increase, however, we know that we can always achieve at least this performance. We may be able to achieve even higher performance if other effects motivate stronger performance at radii smaller than r^*, but this is not guaranteed. □

5.10.5 Proofs for Section 5.4.1

In this section, we give formal theorem statements and proofs for the 3 findings reported in Section 5.4.1.

Finding 1. Multi-class or supervised metric learning PT methods correspond to cliques.
Supervised metric learning PT is when we are given a PT dataset $\boldsymbol{X}_{\mathrm{PT}}$, $\boldsymbol{y}_{\mathrm{PT}} \in \mathcal{Y}^N$ and attempt to pre-train an encoder f such that (potentially in addition to other constraints), $f(\boldsymbol{x}_i)$ is similar to $f(\boldsymbol{x}_j)$ if $y_i = y_j$. From that definition, it is clear that this is identical to a SI objective if $G_{\mathrm{PT}} = (\boldsymbol{X}_{\mathrm{PT}}, E)$ such that $(\boldsymbol{x}_i, \boldsymbol{x}_j) \in E$ if and only if $y_i = y_j$. This corresponds to a graph of disconnected cliques, one for each label $\ell \in \mathcal{Y}$.

In supervised classification PT, we have a PT dataset $\boldsymbol{X}_{\mathrm{PT}}$ and labels $\boldsymbol{y}_{\mathrm{PT}} \in \mathcal{Y}^N$, and attempt to pre-train an encoder f and predictor h such that (potentially in addition to other constraints), $h(f(\boldsymbol{x}_i)) = y_i$. We can realize an analogous SI objective with an identical construction as that for supervised metric learning. If G_{PT} has edges $(\boldsymbol{x}_i, \boldsymbol{x}_j)$ such that $y_i = y_j$, then finding SIPT-optimal embeddings under G_{PT} will also solve the supervised classification objective natively, through nearest-neighbor methods. Thus, SIPT with a graph of disconnected cliques encompasses the supervised classification objective.

Finding 2. The local consistency of any downstream task w.r.t. such a graph rapidly tends to a fixed, sub-optimal constant as PT dataset size increases.
Finding 2 is formally described in Theorem 4.

Theorem 4. *Let $\boldsymbol{X}_{PT} \in \mathcal{X}^N$, be a PT dataset with corresponding labels $\boldsymbol{y} \in \mathcal{Y}_{PT}^N$. Define $G_{PT} = (\boldsymbol{X}_{PT}, E)$ such that $(\boldsymbol{x}_i, \boldsymbol{x}_j) \in E$ if and only if $y_i = y_j$. Let $y_{FT} : \mathcal{X} \to \mathcal{Y}_{FT}$ correspond to a FT downstream task. Further, define* $\mathrm{MC}(\boldsymbol{x}_i, y_{FT}) =$

$\mathrm{argmax}_{\ell_{FT} \in \mathcal{Y}_{FT}} \sum_{\boldsymbol{x}_j \in \boldsymbol{X}_{PT} | y_i = y_j} \mathbb{1}_{y_{FT}(\boldsymbol{x}_i) = \ell_{FT}}$ *to be the majority class label for the clique containing* $\boldsymbol{x}_i$ *in* G_{PT}*. Then,*

$$\mathrm{LC}_{G_{PT}}(y_{FT}) = \sum_{\ell_{PT} \in \mathcal{Y}_{PT}} \mathbb{P}(y_i = \ell_{PT}) \mathbb{P}(y_{FT}(\boldsymbol{x}_i) = \mathrm{MC}(\boldsymbol{x}_i, y_{FT}) | y_i = \ell).$$

Proof. This follows directly from the definition of Local Consistency, G_{PT}, and the law of total probability. In particular,

$$\begin{aligned}
\mathrm{LC}_{G_{\mathrm{PT}}}(y_{\mathrm{FT}}) &= \mathbb{P}\left(y_{\mathrm{FT}}(\boldsymbol{x}_i) = \underset{\ell \in \mathcal{Y}_{\mathrm{FT}}}{\mathrm{argmax}} \sum_{\boldsymbol{x}_j \in \boldsymbol{X}_{\mathrm{PT}} | (\boldsymbol{x}_i, \boldsymbol{x}_j) \in E(G_{\mathrm{PT}})} \mathbb{1}_{y_{\mathrm{FT}}(\boldsymbol{x}_i) = \ell} \right) \\
&= \mathbb{P}\left(y_{\mathrm{FT}}(\boldsymbol{x}_i) = \mathrm{MC}(\boldsymbol{x}_i, y_{\mathrm{FT}}) \right) \\
&= \sum_{\ell_{\mathrm{PT}} \in \mathcal{Y}_{\mathrm{PT}}} \mathbb{P}(y_i = \ell_{\mathrm{PT}}) \mathbb{P}(y_{\mathrm{FT}}(\boldsymbol{x}_i) = \mathrm{MC}(\boldsymbol{x}_i, y_{\mathrm{FT}}) | y_i = \ell),
\end{aligned}$$

as desired. □

Note that this has dependence on the PT dataset size as the probabilities $\mathbb{P}$ are taken over the empirical distribution induced by the dataset $\boldsymbol{X}_{\mathrm{PT}}$ and graph G_{PT} inherent in local consistency — if $\boldsymbol{X}_{\mathrm{PT}}$ is too small, these empirical distributions will be poor proxies for the true distribution and this property will not hold as cleanly. However, once saturation is reached it will not improve beyond this fixed upper bound relating to task correlation.

Finding 3. In contrast, manifold nearest neighbor representations yields nearest-neighbor FT performance that converges to optimality.

Finding 3 is formally described in Theorem 5.

Theorem 5. *Let* $\boldsymbol{X}_{PT}$ *be a PT dataset which can be realized over a valid manifold* $\mathcal{M}$*. Assume* $\boldsymbol{X}_{PT}$ *is sampled with full support over* $\mathcal{M}$*. Let* $G_{PT}(\boldsymbol{X}_{PT}, E)$ *be an r-nearest-neighbor graph over* $\mathcal{M}$ *(e.g.,, $(\boldsymbol{x}_1, \boldsymbol{x}_2) \in E$ if and only if the geodesic distance between the two points on $\mathcal{M}$ is less than r: $\mathcal{D}_{\mathcal{M}}(\boldsymbol{x}_1, \boldsymbol{x}_2) < r$).*

Let y_{FT} *be a FT classification task that is almost everywhere smooth on the manifold* *(as y_{FT} is a classification task, this means that almost everywhere $y_{FT}(\boldsymbol{x}_1) = y_{FT}(\boldsymbol{x}_1 + \boldsymbol{\varepsilon})$ where $\boldsymbol{\varepsilon}$ is a sufficiently small transformation along $\mathcal{M}$. Note this requirement is merely stating more formally that y_{FT} is "piecewise constant" along $\mathcal{M}$, such that compact regions of $\mathcal{M}$ correspond to individual labels of y_{FT} and only on points at the boundary between two labels does y_{FT} show non-constant behavior.*

Then, as PT dataset size (and thus size of G_{PT}) tends to ∞ and r tends to zero, the local consistency of y_{FT} over G_{PT} will likewise tend to 1.

Proof. As $r \to 0$, provided PT dataset size increases at a sufficient associated rate so as to maintain constant minimum degree of G, we have the property that the total diameter over $\mathcal{M}$ contained in a node's local neighborhood within G_{PT} likewise decreases. Given some fixed node $\boldsymbol{x} \in \mathcal{M}$ that is within the interior of a set of constant y_{FT} label, this implies that, eventually, it will grow sufficiently small that all of $\boldsymbol{x}$'s neighbors share the same label as $\boldsymbol{x}$ under y_{FT}.

More concretely, it is clear that this point will occur exactly when r is the geodesic distance between $\boldsymbol{x}$ and the boundary of the surrounding constant-label patch containing $\boldsymbol{x}$. But, it is clear that this implies that the only sections of $\mathcal{M}$ will not have the property that neighborhoods around points will be constant w.r.t. y_{FT} labels will almost everywhere be patches within distance r of the points where y_{FT} changes.

This implies that as $r \to 0$, then almost everywhere will the neighborhoods around a node $\boldsymbol{x}$ be constant w.r.t. y_{FT}. However, this implies that almost everywhere would y_{FT} display perfect local consistency, as desired. □

5.11 Full Experimental Details

5.11.1 Full Details for Synthetic Experiments

Details on Dataset & Graph Construction

General Setup. All our synthetic datasets leverage free-text sentences from `https://www.kaggle.com/mikeortman/wikipedia-sentences` (CC BY-SA 4.0 License). Topics were assigned to these sentences by running Latent Dirichlet Allocation via

Scikit-learn [151] over a Bag-of-words representation to 100 topics, with otherwise default parameters.

Given the probabilities over all 100 topics, we treated the prediction of the most probable topic as a 100-class multi-class classification problem for our FT task in these experiments.

Cliques Graph Setup. To construct the Cliques graph setting, we choose a random subset of sentences as $\boldsymbol{X}_{\text{PT}}$ and define $G_{\text{PT}} = (\boldsymbol{X}_{\text{PT}}, E)$ such that $(\boldsymbol{x}_i, \boldsymbol{x}_j) \in E$ if and only if $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ share the same topic label.

Plane, Möbius, Sphere, & Torus Graphs. For these graphs, we take a more involved practice to localize sentences onto specifiable simplicial manifolds, then construct pre-training graphs via radius nearest neighbor graphs on those manifolds. This involves several steps:

Localizing Sentences on Simplices We can localize any sentence in our overall dataset onto a 2-simplex by mapping them onto the (re-normalized) probabilities associated with their top-3 topics. Doing this in this way means that the simplex on which they are localized has vertices that correspond to possible topics among our set of 100 total topics.

Stitching Topic-simplices Into Manifolds Given these topic-simplex localized sentences, we need to construct our manifolds. To do so, we first produce any arbitrary simplicial tiling of a 2-manifold. With this tiling, all that remains to localize sentences onto the manifold is to find a self-consistent mapping of topics to simplex vertices (in the tiling) such that all topic-simplices induced by this mapping have sufficiently many associated samples to enable roughly uniform sampling.

Sampling points After finding a self-consistent mapping of topics to simplicial tiling vertices that satisfies density requirements, we can then sample sentences onto the manifold. To make this process more uniform, we also calculate the relative entropy of each sentence (over the re-normalized probabilities of the top-3 topics), bin those entropies into buckets, then sample first what entropy bucket we wish to draw from such that the induced distribution of sentence entropies is approximately uniform, then sample within that entropy bucket.

Calculating on-Manifold Distances Finally, with sentences sampled and localized onto a simplicial manifold, we then need to compute approximate geodesic distances to enable building radius-nearest-neighbor graphs over these sentences. To do so, we use an approximate algorithm that considers only on-simplex distance (*e.g.*,, it does not consider any curvature penalties) which is equivalent to calculating the distance between any pair of points over the simplices presuming they were flattened onto a plane (this flattening naturally does not preserve manifold topology, but along only the shortest path between any particular set of two points it is always possible to do so with a 2-manifold).

The above process describes how to produce a radius-nearest-neighbor graph for any specifiable manifold using our topic-model outputs. We do this for simplicial manifolds that correspond topologically to a simple plane (PLANE), a möbius strip (MÖBIUS), a sphere (SPHERE), and a torus (TORUS).

STRUCTURAL, NEIGHBORHOOD, & MOTIFS Graphs. In order to form these examples, we must (1) define our overall graphs, (2) featurize these graphs in a manner that is reflective of different forms of graph structure, then (3) use these featurizations to assign sentences to graph nodes to form our pre-training dataset.

Graph Construction We sample graphs as described in the main body, with a base cycle and motifs distributed along that cycle evenly.

Node Featurization Nodes in this graph are then assigned internal features based on three notions of graph topology. For the "Homophily" label, a node n is identified according to an index-vector indicating which nodes in the graph are within shortest-path distance 3 of n. For the "Motif" label, n is identified based on its membership either in the base cycle or any of the attached random subgraphs. For the "Structural" label, n is identified based on its graphlet degree vector (of order 4). For structural and homophily features, categorical labels are then produced by feeding these raw representations through a k-means clustering algorithm.

Sentence Assignment We assign sentences to nodes in multiple ways, so that we can produce datasets that reflect each of the notions of graph structure discussed previously. In particular, for either the homophily, motif, or structural labels, each sentence topic is matched to a node label, then sentences are assigned randomly to nodes in the graph with a matching topic label. Note that this produces a dataset where the graph structure is only partially reflected by the node's features, which is itself another useful test of the SIPT system, as it would not be useful if SIPT could only capture data in contexts where the graph was perfectly reflected by the node features themselves.

Network Architecture & Hyperparameters

The Cliques and Mechanistic experiments in this domain use a shallow Transformer model with 2 layers and 10 hidden units. The Manifold experiments use a 3-layer Transformer model with 256 hidden units. Hyperparameters were not tuned, but were chosen by-hand to produce as small a network as possible while still permitting reasonable learning dynamics.

5.11.2 Full Details for Real-world Experiments

PT Datasets

The Proteins PT data (the Stanford tree-of-life dataset) lists no license on their download page (`https://snap.stanford.edu/tree-of-life/data.html`), though the associated github repository lists an MIT license. The Abstracts PT data (the Microsoft Academic Graph dataset) is licensed with a Open Data Commons Attribution License (ODC-By) v1.0 license. The Networks PT dataset, released in the relevant source article [92] lists no specific data license but releases the code (which contains the dataset files) under an MIT license.

Proteins

PT Dataset. We use a dataset of ∼1.5M protein sequences from the Stanford Tree-of-life dataset [252].

PT Graph. Two proteins are linked in G_{PT} if and only if they are documented in the scientific literature to interact, according to the tree-of-life dataset. This is an external knowledge graph.

FT Dataset/Tasks. We use the TAPE FT benchmark tasks [166], including Remote homology (RH), a per-sequence classification task to predict protein fold category (metric: accuracy); Secondary structure (SS), a per-token classification task to predict amino acid structural properties (metric: accuracy); Stability (ST) & Fluorescence (FL), per-sequence, regression tasks to predict a protein's stability and fluorescence, respectively (metric: Spearman's ρ); and Contact prediction (CP), an intra-sequence classification task to predict which pairs of amino acids are in contact in the protein's 3D conformation (metric: Precision at $L/5$).

Baselines. We compare against the published TAPE model [166], which uses a LM task alone, as our per-token comparison point and the PLUS [138] model, which optimizes for LM and supervised classification jointly, for our per-sample comparison point.

Abstracts

PT Dataset. We use a dataset of ∼650K free-text scientific article abstracts from the Microsoft Academic Graph (MAG) dataset [215, 90].

PT Graph. Two abstracts are linked in G_{PT} if and only if their corresponding papers cite one another. This is a self-supervised graph.

FT Dataset/Tasks. We use the SciBERT benchmark tasks [15], including Paper field (PF), SciCite (SC), ACL-ARC (AA), and SciERC Relation Extraction (SRE), all of which are per-sentence classification problems (metric: Macro-F1). PF tasks models to predict a paper's area of study from its title, SC & AA tasks both are to predict an "intent" label for citations, and SRE is a relation extraction task.

Baselines. We compare against the published SciBERT model [15] as our per-token comparison, and lack an associated per-sample comparison as we don't know of any published per-sample models in the academic papers modality.

Networks

PT Dataset. We use a dataset of ∼70K protein-protein interaction (PPI) ego-networks here, sourced from [92]. Each individual sample here describes a single protein, realized as a biological network (*i.e.*,, an attributed graph) corresponding to the ego-network about that protein (*i.e.*,, a small subgraph containing all nodes within the target protein) in a broader PPI graph. Unlike our other domains, this domain does not contain sequences.

PT Graph. The dataset from [92] is labeled with the presence or absence of any of 4000 protein gene ontology terms associated with each the central protein in each PPI ego-network. Leveraging these labels, two PPI ego-networks are linked in G_{PT} if and only if the hamming distance between their observed label vectors is no more than 9. This is an alternate-representation nearest-neighbor graph.

FT Dataset/Tasks. Our FT task is the multi-label binary classification of the 40 gene-ontology term annotations (metric: macro-AUROC) used in [92]. We use the PT set for FT training, and evaluate the model on a held-out random 10% split.

Baselines. We compare against both attribute-masking [92] and multi-task PT.

Fine-tuning Task Descriptions

Proteins. The tasks in the TAPE benchmark [166] on which we test are described more fully below. All these datasets are publicly available. All datasets can be obtained directly on TAPE's github (`https://github.com/songlab-cal/tape#data`) which lists no licenses for these datasets, though the overall github is released under a BSD 3-Clause "New" or "Revised" License.

Remote Homology This is a per-sequence, multi-class classification problem, evaluated using accuracy, which tasks a model to predict a protein fold category at a per-sequence level. This task's dataset contains 12,312/736/718 train/val/test proteins, and is originally sourced from [87].

Secondary Structure This is a per-token, multi-class classification problem, evaluated using accuracy, which tasks a model to predict the structural properties of each amino acid in the final, folded protein. This task's dataset contains 8,678/2,170/513 train/val/test proteins, and is originally sourced from [103].

Stability This is a per-sequence, continuous regression problem, evaluated using Spearman correlation coefficient, which tasks a model to predict the protein's stability in response to environmental conditions. This task's dataset contains 53,679/2,447/12,839 train/val/test proteins, and is originally sourced from [173].

Fluorescence This is a per-sequence, continuous regression problem, evaluated using Spearman correlation coefficient, which tasks a model to predict how brightly a protein will fluoresce. This task's dataset contains 21,446/5,362/27,217 train/val/test proteins, and is originally sourced from [180].

Abstracts. The tasks in the SciBERT benchmark [15] on which we test are described more fully below. All tasks here are per-sentence, multi-class classification problems (i.e., we do not study any per-token tasks), and all are evaluated in Macro-F1 (out of 1). All FT datasets can be obtained from the SciBERT github (`https://github.com/allenai/scibert`), which lists no dataset specific licenses but is itself released with an Apache-2.0 license.

Paper Field This problem tasks models to predict a paper's area of study given its title. This task's dataset contains 84,000/5,599/22,399 train/val/test sentences. Though original derived from the MAG [215], it was to the best of our knowledge formulated into this task format by SciBERT directly [15].

SciCite This problem tasks models to predict an "intent" label for sentences that cite other scientific works within academic articles. This task's dataset contains 7,320/916/1,861 train/val/test sentences, and is originally sourced from [39].

ACL-ARC This problem tasks models to predict an "intent" label for sentences that cite other scientific works within academic articles. This task's dataset contains 1,688/114/139 train/val/test sentences, and is originally sourced from [97].

Networks. The Networks FT task is a multi-task, binary classification task that is defined as follows. Recall that the dataset here consists of PPI ego-networks, which means more concretely that an individual sample input to the model is an attributed graph $\boldsymbol{x}$ which contains a central node, corresponding to a protein, along with the ego-graph surrounding that node in a larger PPI graph. This ego-graph can thus be seen to correspond to the central protein, and the FT and PT tasks leverage this association, as both of which flag whether or not that central protein is associated with particular gene-ontology (GO) terms (annotations relating to protein properties or function applied in the literature). The PT tasks contain 4000 possible GO annotations, but the FT tasks correspond to a smaller set of only 40 GO terms, chosen as they were of greater interest than the full set. See the original source ([92]) for more information and full details.

Architecture & Hyperparameters

The architectures of our encoders for the Proteins and Abstracts domains are fully determined from our source models in TAPE [166] and SciBERT [15]. In particular, for proteins and scientific articles, we use a 12-layer Transformer with a hidden size of 768, intermediate size of 3072, and 12 attention heads. Provided TAPE and SciBERT tokenizers are also used. A single linear layer to the output dimensionality of each task is used s the prediction head, taking as input the output of the final layer's `[CLS]` token as a whole-sequence embedding. We also tested either pre-training for a single or for four additional epochs, based on validation set performance, and ultimately used a single epoch for proteins and four for scientific articles.

For the Networks domain, we match the architecture used in the original source [92] for the mask model runs, save that for computational efficiency we scale the batch-size up as high as it can go, then proportionally scale up the learning rate to account for the larger batch size. This corresponds to a batch size of 1024, learning rate of 0.01, a GCNN encoder type of GIN, embedding dimensions of 300, 5 layers, 10% dropout, mean pooling, and a JK strategy of "last".

Fine-tuning hyperparameters (learning rate, batch size, and number of epochs) were determined based on a combination of existing results, hyperparameter tuning, and machine limitations. On proteins, most hyperparameters were set to follow those reported for a LM PT model in [131], though additional limited hyperparameter searches were performed to validate that these choices were adequate. As the original source for these hyperparameters was an LM PT model, any bias here should be *against* SIPT, meaning this is a conservative choice. Early stopping (based on the number of epochs without observing improvement in the validation set performance) was employed and batch size was set as large as possible given the limitations of the underlying machine. For the PLUS reproduction, we additionally compared hyperparameters analogous to the reported PLUS hyperparameters for other tasks as well as analogous to our hyperparameters for other tasks and used those that performed best on the validation set. For scientific articles, we performed grid search to optimize downstream task performance on the validation set, with learning rate varying between 5e-6 and 5e-5 and number of epochs between 2 and 5. The same grid search was used in original SciBERT system. We additionally match the SciBERT benchmark by applying dropout of 0.1, using the Adam optimizer with linear warm-up and decay, a batch size of 32, and no early stopping. For the NETWORKS, FT hyperparameters were again chosen to match the original source model [92] save the increase in batch size and learning rate. No additional hyperparameter search was performed.

Final hyperparameters for each downstream task are shown in Tables 5.5 for proteins and 5.6 for scientific articles.

5.11.3 Implementation and Compute Environment

We leverage PyTorch for our codebase. FT Experiments and NETWORKS PT were run over various ubuntu machines (versions ranged from 16.04 to 20.04) with a variety of NVIDIA GPUs. PROTEINS and ABSTRACTS PT runs were performed on a Power 9 system, each run using 4 NVIDIA 32 GB V100 GPUs with InfiniBand at half precision.

Task	Batch Size	LR
Remote Homology	16	1e-5
Fluorescence	128	5e-5
Stability	512	1e-4
Secondary Structure	16	1e-5

Table 5.5: Final hyperparameters for our Proteins domain. All tasks used 200 total epochs and performed early stopping after 25 epochs of no validation set improvement. LR, learning rate.

Task	# Epochs	LR
Paper Field	2	5e-5
ACL-ARC	4/5	5e-5
SciCite	3/2	1e-5

Table 5.6: Final hyperparameters for our Abstracts dataset. All models used a batch size of 32 and no early stopping, to match the original SciBERT paper [15]. LR, learning rate. A / B = [LM PT Hyperparameter] / [SIPT Hyperparameter].

5.12 Further Empirical Results

5.12.1 Raw Results

In Tables 5.7 and 5.8, we show the raw FT results for all tasks in the Proteins and Abstracts domains, respectively.

Model	RH	FL	ST	SS	CP
TAPE	21%	**0.68**	0.73	73%	0.32
PLUS	19.8%±1.7*	0.63	0.76	73%	N/A
LM PT	23.8%±1.1	0.67±0.00	0.76±0.02	73.9%±0.0	0.38
SIPT-C	25.1%±0.6	**0.68±0.00**	**0.77±0.01**	73.9%±0.0	0.38
SIPT-M	**26.6%±1.0**	**0.68±0.00**	0.76±0.01	**74.2%±0.1**	**0.39**

Table 5.7: Results of the TAPE Transformer [166], the PLUS Transformer [138] (*: our measurements), our LM PT baseline, and two SIPT variants ("-C" indicates the contrastive loss, "-M" the multisimilarity loss). Higher is better.

Model	PF	SC	AA	SRE
SciBERT	**0.66**	0.85	0.71	0.80
LM PT	**0.66**$\pm$**0.0**	0.85$\pm$0.01	0.70$\pm$0.05	0.80$\pm$0.01
SIPT-C	**0.66**$\pm$**0.0**	**0.86**$\pm$**0.01**	**0.76**$\pm$**0.02**	**0.81**$\pm$**0.00**
SIPT-M	**0.66**$\pm$**0.0**	0.85$\pm$0.00	0.73$\pm$0.05	N/A

Table 5.8: Results of the original SciBERT [15] model, our own LM PT baseline, and two SIPT variants ("-C" indicates the contrastive loss, "-M" the multisimilarity loss). Higher is better.

5.12.2 A deeper dive into Proteins vs. Abstracts

For the Proteins and Abstracts dataset, to choose the optimal value of λ_{SI} for use at PT time, we pre-trained several models and evaluated their efficacy in a link retrieval task on $G_{\text{PT}} = (V, E)$. In particular, we score a node embedder f by embedding all nodes $n \in V$ as $f(n)$, then rank all other nodes n' by the euclidean distance between $f(n)$ and $f(n')$, and assess this ranked list via IR metrics including label ranking average precision (LRAP), normalized discounted cumulative gain (nDCG), average precision (AP), and mean reciprocal rank (MRR), where a node n' is deemed to be a "successful" retrieval for n if $(n, n') \in E$. In this way, note that we choose λ_{SI} in a manner that is independent of the fine-tuning task, and can be determined solely based on the PT data. Final results for these experiments are shown in Table 5.9 for the proteins dataset and Table 5.10 for scientific articles.

In both settings, we compare the following models.

Random Nodes are embedded with random vectors, to assess chance performance.

Initial Model Nodes are embedded with the base pre-trained model we build on in our experiments without further modifications. This model is TAPE [166] for proteins and SciBERT [15] for scientific articles.

LM PT Nodes are embedded with the final encoder after additional pre-training on our graph-augmented datasets, but without any SIPT (i.e., $\lambda_{\text{SI}} = 0$).

CS RoBERTa *(for scientific articles only)* Nodes are embedded via [75]'s DAPT CS RoBERTa model, which is another LM PT model over scientific abstracts which performed very well on ACL-ARC, the task on which SIPT does best in scientific articles.

SIPT *(for various values of λ_{SI}).* Nodes are represented via SIPT PT models at the specified weighting. For proteins, all SIPT models are initialized from TAPE, but for scientific articles, we test against both initializing from SciBERT and from CS RoBERTa (as both are just different, domain-specific LM PT models).

Note that in addition to the discrepancy in the magnitude of improvement (over scientific articles, average precision goes from 12.9% to 14.2%, vs. 2.4% to 3.5% on proteins, which is proportionally much more significant), we can also see that SIPT improves retrieval performance over the baselines for proteins much more than it does for scientific articles. This is, admittedly, largely due to [75]'s CS RoBERTa model's surprisingly good performance without any modifications, however as we also compare SIPT pre-trained from a CS RoBERTa model and it does not demonstrate significant improvements, we still feel this is a fair comparison. These findings are consistent with our hypothesis that SIPT will offer more significant advantages on non-natural language domains.

Method	λ_{SI}	LRAP	nDCG	AP	MRR
Random Baseline	N/A	0.88%	27.1%	0.88%	0.003
TAPE [166]	N/A	8.50%	34.9%	2.41%	0.226
LM PT Baseline	0	8.92%	38.0%	2.33%	0.238
SIPT (TAPE Initialized)	0.01	9.69%	39.1%	2.56%	0.254
	0.10	10.95%	39.4%	3.46%	0.260
	0.50	10.54%	40.3%	3.43%	0.246
	0.90	10.12%	39.0%	3.16%	0.237
	0.99	14.50%	37.5%	3.13%	0.236

Table 5.9: PT set link-retrieval performance for a random baseline, the raw TAPE model, and SIPT for various weighting parameters λ_{SI} on the dataset of protein sequences. LRAP, label ranking average precision; nDCG, normalized discounted cumulative gain; AP, average precision; MRR, mean reciprocal rank. Higher values indicate better performance. Highlighted in grey are realizations of SIPT framework that yield better results than the strongest baseline, providing evidence that incorporating sequence-level relational information into PT (i.e., $\lambda_{\text{SI}} > 0$) leads to improved performance.

Method	λ_{SI}	LRAP	nDCG	AP	MRR
Random Baseline	N/A	0.89%	26.0%	0.27%	0.016
SciBERT [15]	N/A	17.22%	52.8%	5.16%	0.272
LM PT Baseline (SciBERT initialized)	0	16.79%	35.4%	5.00%	0.271
DAPT CS RoBERTa [75]	N/A	32.56%	50.3%	12.86%	0.459
LM PT Baseline (CS RoBERTa initialized)	0	30.58%	48.3%	12.36%	0.438
SIPT (SciBERT initialized)	0.01	42.26%	58.7%	14.23%	0.536
	0.10	34.73%	52.5%	9.39%	0.457
	0.50	32.85%	50.8%	8.37%	0.438
	0.90	31.61%	49.8%	7.82%	0.426
	0.99	30.72%	49.0%	6.80%	0.415
SIPT (CS RoBERTa initialized)	0.01	33.32%	51.2%	8.61%	0.448
	0.10	25.46%	44.4%	5.88%	0.359
	0.50	25.08%	44.0%	6.08%	0.355
	0.90	22.43%	41.6%	4.27%	0.317
	0.99	22.38%	41.5%	4.68%	0.316

Table 5.10: PT set link-retrieval performance for a random baseline, the raw SciBERT model, and SIPT for various weighting parameters λ_{SI} on the scientific articles dataset. LRAP, label ranking average precision; nDCG, normalized discounted cumulative gain; AP, average precision; MRR, mean reciprocal rank. Higher values indicate better performance. Highlighted in grey are realizations of SIPT framework that yield better results than the strongest baseline, providing evidence that incorporating sequence-level relational information into PT (i.e., $\lambda_{\mathrm{SI}} > 0$) leads to improved performance.

Chapter 6

Conclusion

6.1 Summary

In this thesis, I've explored how we can leverage structure and knowledge in clinical and biomedical representation learning tasks.

In particular, in Chapter 2, we showed that leveraging additional, unlabeled data to learn global distributional statistics can produce richer representations in the context of biomedical regression tasks. These ideas naturally promote the practice of leveraging unlabeled data more generally, which we explore in detail in later chapters through pre-training architectures. In addition, the use of our cycle regularization penalty and the learned distributional signals via adversarial models represents a form of global structural regularization in the latent space of these models, which we will also revisit in our Structure Inducing Pre-Training framework.

Next, in Chapter 3, we explored the use of local structure to refine model frameworks for biomedical data. In particular, we introduce the use of graph convolutional neural networks (GCNNs) to model transcriptomic gene expression data. We find that using GCNNs augmented with *prior knowledge* in the form of genetic co-regulatory networks mined from the scientific literature offers significant advantages over other modelling technologies. This shows that even in the high-capacity, large data regimes that dominate modern machine learning, the use of prior knowledge/structure can be invaluable in high-performance representation learning.

In Chapter 4, we focus more squarely on pre-training techniques, a subset of representation learning methods focused on using additional un- or weakly-labeled datasets to initialize a model prior to separate fine-tuning across diverse downstream techniques. We show that when adapting successful pre-training methods from general machine learning is insufficient for clinical timeseries data. In contrast, a simple approach of multi-task pre-training offers significant improvements in the few-shot regime. The central salient difference between the multi-task pre-training approach and the BERT-inspired masked imputation approach is that the former imposes *greater structural constraints* on the induced pre-training latent space. We believe that this phenomena is exactly why it outperforms the comparatively less restrictive BERT-inspired approach.

Finally, in Chapter 5, we cement this observation by introducing a new pre-training framework that solidifies the role global structure plays in pre-training success. In particular, we introduce *Structure-inducing Pre-training*, a framework which re-casts pre-training tasks as graph embedding problems, and pre-training learning objectives as associated metric learning tasks. Doing this allows us to use the structure of the pre-training graph to infer properties about the geometry of the output latent space, and thus about overall suitability of the pre-training process to eventual downstream tasks. We demonstrate the power of this approach through both theory and experimental analyses, and motivate several new directions for future work.

Ultimately, these works come together to paint a compelling picture for the promise of incorporating structure and external knowledge into clinical and biomedical representation learning, specifically through pre-training algorithms.

6.2 Unifying Perspectives & Future Work

My work identifies several key findings in the area of representation learning for clinical and biomedical data, and promotes several important directions for future work.

Structure and Knowledge Provide Value, but Research on Identifying the Right Structure is Still Needed My research in this thesis consistently shows that leveraging data structure and external knowledge can offer significant advantages over unstructured approaches. Be this in the improvements offered leveraging unlabeled data to build distributional signals (Chapter 2), leveraging external knowledge to inform model architecture at a per-sample level (Chapter 3), by showing that even weak structural regularization through multi-task pre-training offers advantages (Chapter 4), or in providing theoretical guarantees and empirical performance improvements by leveraging global dataset structure in pre-training tasks (Chapter 5). However, it is also clear that such approaches are neither catch-all solutions nor operate fully within expected bounds. For example, when leveraging gene expression co-regulatory networks to inform gene expression modelling in [136], we found that generic, literature-sourced networks were better than data-driven tissue-specific networks, which disagreed with our preliminary hypothesis. Similarly, in [129], while our pre-training approaches offer advantages in the few-shot regime, they fail to outperform single-task learning in the full-data setting, suggesting they are underpowered compared to pre-training solutions in the general domain. These findings suggest that while there is clear value in employing data structure and external knowledge, identifying the right *kind* of structure/knowledge is still a difficult challenge, though the theoretical analyses in [132] do offer some options to intelligently search for identifying such structure.

Biomedical Data Requires Specialized Approaches to Take Full Advantage of its Structure While traditional domains have well-established models to reflect expected mathematical structures (*e.g.*,, convolutional neural networks for images and other spatial domains, recurrent neural networks & Transformers for sequential domains), biomedical data reflect more varied forms of structure and thus may require additional forms of modelling to reflect these other structures (e.g., graph structured data, dynamical systems, causal models, mechanistic models, etc.). My work in this thesis shows that different forms of structure require different kinds of modeling, and generic techniques from traditional ML domains may not work as well in the biomedical domain. In particular, in Chapter 3, we show that graph convolutional neural networks leveraging co-regulatory networks offer significant advantages when modelling gene expression data. Further, Chapters 4 and 5 both show that naïve applications of generic pre-training methods to clinical and biomedical domains don't realize the same performance gains those methods offer in general modalities.

"Structure" is not All-Encompassing, and more Nuanced Notions of *what* we want to capture and *why* we want to capture it are necessary In this thesis, I postulate that we should make use of "structure" to aid in biomedical representation learning, much as others have proposed in the literature previously. However, I (and many others) do so without providing a clear definition of what exactly we mean by "structure" in these contexts. This flexibility can be useful; different notions of structure help models in different ways, and make different theoretical guarantees. However, it also serves to mask what inductive biases are actually being learned. For example, as we show in Chapter 5, traditional pre-training and representation learning works often make minimal guarantees about the per-sample geometry of the induced latent spaces, yet these same models are often used for analogical reasoning tasks and other knowledge discovery processes which rely on assumptions about that latent geometry being meaningful.

While the SIPT framework introduced in that chapter takes steps to alleviate this issue by providing a unified theoretical framing to undestand how structure is being imposed and to realize new methods that can realize richer forms of structure, the notion of "structure" used in that work is still narrow. In particular, SIPT captures only tasks that can be realized as metric learning tasks over a pre-training graph. While this limited form of structure is useful for many tasks, it is insufficient to capture the variety of kinds of structure we encounter within biomedical domains, and further work remains to adapt, refine, and extend this framework to handle these cases.

Bibliography

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mane, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viegas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems. *arXiv:1603.04467 [cs]*, March 2016. arXiv: 1603.04467.

[2] Oshin Agarwal, Heming Ge, Siamak Shakeri, and Rami Al-Rfou. Knowledge graph based synthetic corpus generation for knowledge-enhanced language model pre-training. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3554–3565, Online, June 2021. Association for Computational Linguistics.

[3] Alexander Aliper, Sergey Plis, Artem Artemov, Alvaro Ulloa, Polina Mamoshina, and Alex Zhavoronkov. Deep Learning Applications for Predicting Pharmacological Properties of Drugs and Drug Repurposing Using Transcriptomic Data. *Molecular Pharmaceutics*, 13(7):2524–2530, July 2016.

[4] Ethan C. Alley, Grigory Khimulya, Surojit Biswas, Mohammed AlQuraishi, and George M. Church. Unified rational protein engineering with sequence-based deep representation learning. *Nature Methods*, (12), 2019.

[5] David B Allison, Xiangqin Cui, Grier P Page, and Mahyar Sabripour. Microarray data analysis: from disarray to consolidation and consensus. *Nature Reviews. Genetics*, 7(1):55–65, January 2006.

[6] Emily Alsentzer, Matthew McDermott, Fabian Falck, Suproteem K Sarkar, Subhrajit Roy, and Stephanie L Hyland. Ml4h abstract track 2020. *arXiv e-prints*, pages arXiv–2011, 2020.

[7] Emily Alsentzer, John Murphy, William Boag, Wei-Hung Weng, Di Jindi, Tristan Naumann, and Matthew B.A. McDermott. Publicly Available Clinical BERT

Embeddings. In *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, pages 72–78, Minneapolis, Minnesota, USA, June 2019. Association for Computational Linguistics.

[8] Natalia Antropova, Andrew L Beam, Brett K Beaulieu-Jones, Irene Chen, Corey Chivers, Adrian Dalca, Sam Finlayson, Madalina Fiterau, Jason Alan Fries, Marzyeh Ghassemi, et al. Machine learning for health (ml4h) workshop at neurips 2018. *arXiv preprint arXiv:1811.07216*, 2018.

[9] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein GAN. *arXiv:1701.07875 [cs, stat]*, January 2017. arXiv: 1701.07875.

[10] Annamarie Bair, Matthew McDermott, and Peter Szolovits. Improved modeling and analysis of gene expression.

[11] Aparna Balagopalan, Jekaterina Novikova, Matthew B A Mcdermott, Bret Nestor, Tristan Naumann, and Marzyeh Ghassemi. Cross-Language Aphasia Detection using Optimal Transport Domain Adaptation. In Adrian V. Dalca, Matthew B.A. McDermott, Emily Alsentzer, Samuel G. Finlayson, Michael Oberst, Fabian Falck, and Brett Beaulieu-Jones, editors, *Proceedings of the Machine Learning for Health NeurIPS Workshop*, volume 116 of *Proceedings of Machine Learning Research*, pages 202–219. PMLR, 13 Dec 2020.

[12] Brett Beaulieu-Jones, Samuel G Finlayson, Corey Chivers, Irene Chen, Matthew McDermott, Jaz Kandola, Adrian V Dalca, Andrew Beam, Madalina Fiterau, and Tristan Naumann. Trends and focus of machine learning applications for health research. *JAMA network open*, 2(10):e1914051–e1914051, 2019.

[13] Parishad BehnamGhader, Hossein Zakerinia, and Mahdieh Soleymani Baghshah. Mg-bert: Multi-graph augmented bert for masked language modeling. In *Proceedings of the Fifteenth Workshop on Graph-Based Methods for Natural Language Processing (TextGraphs-15)*, pages 125–131, 2021.

[14] Marc G. Bellemare, Ivo Danihelka, Will Dabney, Shakir Mohamed, Balaji Lakshminarayanan, Stephan Hoyer, and Rémi Munos. The Cramer Distance as a Solution to Biased Wasserstein Gradients. *arXiv:1705.10743 [cs, stat]*, May 2017. arXiv: 1705.10743.

[15] Iz Beltagy, Kyle Lo, and Arman Cohan. SciBERT: A Pretrained Language Model for Scientific Text. In *EMNLP*, 2019.

[16] James Bergstra and Yoshua Bengio. Random Search for Hyper-Parameter Optimization. *Journal of Machine Learning Research*, 13(Feb):281–305, 2012.

[17] James Bergstra, Daniel Yamins, and David Cox. Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. In Sanjoy Dasgupta and David McAllester, editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings*

of Machine Learning Research, pages 115–123, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR.

[18] Dimitris Bertsimas, Jean Pauphilet, Jennifer Stevens, and Manu Tandon. Predicting inpatient flow at a major hospital using interpretable analytics. *medRxiv*, 2020.

[19] William Boag, Tzu-Ming Harry Hsu, Matthew McDermott, Gabriela Berner, Emily Alesentzer, and Peter Szolovits. Baselines for chest x-ray report generation. In *Machine Learning for Health Workshop*, pages 126–140. PMLR, 2020.

[20] Broad Connectivity Map Team. L1000 connectivity map perturbational profiles from broad institute lincs center for transcriptomics lincs phase *II*, 2016.

[21] Michael M. Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. Geometric deep learning: going beyond Euclidean data. *arXiv:1611.08097 [cs]*, November 2016. arXiv: 1611.08097.

[22] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

[23] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language Models are Few-Shot Learners. *arXiv preprint arXiv: 2005.14165*, 2020.

[24] Eliot C Bush, Anne E Clark, Chris M DeBoever, Lillian E Haynes, Sidra Hussain, Singer Ma, Matthew M McDermott, Adam M Novak, and John S Wentworth. Modeling the role of negative cooperativity in metabolic regulation and homeostasis. *PloS one*, 7(11):e48920, 2012.

[25] Renzhi Cao, Colton Freitas, Leong Chan, Miao Sun, Haiqing Jiang, and Zhangxin Chen. ProLanGO: Protein Function Prediction Using Neural Machine Translation Based on a Recurrent Neural Network. *Molecules*, 22(10):1732, October 2017.

[26] Chun-Hao Chang, Mingjie Mai, and Anna Goldenberg. Dynamic measurement scheduling for event forecasting using deep RL. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 951–960, Long Beach, California, USA, 09–15 Jun 2019. PMLR.

[27] Geeticka Chauhan, Matthew McDermott, and Peter Szolovits. A framework for relation extraction across multiple datasets in multiple domains. In *Proceedings of the 2019 Workshop on Widening NLP*, pages 18–20, 2019.

[28] Zhengping Che, Sanjay Purushotham, Kyunghyun Cho, David Sontag, and Yan Liu. Recurrent Neural Networks for Multivariate Time Series with Missing Values. *Scientific Reports*, 8(1):6085, April 2018.

[29] Zhengping Che, Sanjay Purushotham, Kyunghyun Cho, David Sontag, and Yan Liu. Recurrent neural networks for multivariate time series with missing values. *Scientific Reports*, 8(1):6085, Apr 2018.

[30] Lujia Chen, Chunhui Cai, Vicky Chen, and Xinghua Lu. Learning a hierarchical representation of the yeast transcriptomic machinery using an autoencoder model. *BMC Bioinformatics*, 17(1):S9, January 2016.

[31] Yifei Chen, Yi Li, Rajiv Narayan, Aravind Subramanian, and Xiaohui Xie. Gene expression inference with deep learning. *Bioinformatics*, 32(12):1832–1839, June 2016.

[32] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar, October 2014. Association for Computational Linguistics.

[33] Edward Choi, Siddharth Biswal, Bradley Malin, Jon Duke, Walter F. Stewart, and Jimeng Sun. Generating Multi-label Discrete Electronic Health Records using Generative Adversarial Networks. *arXiv:1703.06490 [cs]*, March 2017. arXiv: 1703.06490.

[34] Edward Choi, Cao Xiao, Walter Stewart, and Jimeng Sun. Mime: Multilevel medical embedding of electronic health records for predictive healthcare. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[35] Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators. In *ICLR*, 2019.

[36] Neil R. Clark, Kevin S. Hu, Axel S. Feldmann, Yan Kou, Edward Y. Chen, Qiaonan Duan, and Avi Ma'ayan. The characteristic direction: a geometrical approach to identify differentially expressed genes. *BMC Bioinformatics*, 15:79, March 2014.

[37] Andy Coenen, Emily Reif, Ann Yuan, Been Kim, Adam Pearce, Fernanda Viégas, and Martin Wattenberg. Visualizing and Measuring the Geometry of BERT. In *NeurIPS*, 2019.

[38] Valerie C. Coffman, Matthew B. A. McDermott, Blerta Shtylla, and Adriana T. Dawes. Stronger net posterior cortical forces and asymmetric microtubule arrays produce simultaneous centration and rotation of the pronuclear complex in the early Caenorhabditis elegans embryo. *Molecular Biology of the Cell*, 27(22):3550–3562, October 2016.

[39] Arman Cohan, Waleed Ammar, Madeleine van Zuylen, and Field Cady. Structural Scaffolds for Citation Intent Classification in Scientific Publications. In *Proceedings of the 2019 Conference of the NAACL: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019.

[40] Ganqu Cui, Jie Zhou, Cheng Yang, and Zhiyuan Liu. Adaptive graph encoder for attributed graph embedding. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 976–985, 2020.

[41] Adrian V. Dalca, Matthew B.A. McDermott, Emily Alsentzer, Sam Finlayson, Michael Oberst, Fabian Falck, Corey Chivers, Andrew Beam, Tristan Naumann, and Brett Beaulieu-jones. Machine learning for health (ml4h) 2019 : What makes machine learning in medicine different? In *Proceedings of Machine Learning for Health NeurIPS Workshop*, volume 116, pages 1–9. PMLR, 2020.

[42] Frederick D'Aragon, Emilie P Belley-Cote, Maureen O Meade, Francois Lauzier, Neill KJ Adhikari, Matthias Briel, Manoj Lalu, Salmaan Kanji, Pierre Asfar, Alexis F Turgeon, et al. Blood pressure targets for vasopressor therapy: A systematic review. *Shock*, 43(6):530–539, 2015.

[43] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 3844–3852. Curran Associates, Inc., 2016.

[44] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[45] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[46] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the*

Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[47] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL*, 2019.

[48] Inderjit S Dhillon, Yuqiang Guan, and Brian Kulis. Weighted graph cuts without eigenvectors a multilevel approach. *IEEE transactions on pattern analysis and machine intelligence*, 29(11), 2007.

[49] Daisy Yi Ding, Chloé Simpson, Stephen Pfohl, Dave C Kale, Kenneth Jung, and Nigam H Shah. The effectiveness of multitask learning for phenotyping with electronic health records data. In *Biocomputing 2019*, pages 18–29. World Scientific, 2019.

[50] domdomegg. A simple diagram of an unspecialised animal cell without labels, January 2016.

[51] David K Duvenaud, Dougal Maclaurin, Jorge Iparraguirre, Rafael Bombarell, Timothy Hirzel, Alan Aspuru-Guzik, and Ryan P Adams. Convolutional Networks on Graphs for Learning Molecular Fingerprints. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 2224–2232. Curran Associates, Inc., 2015.

[52] Michael B. Eisen, Paul T. Spellman, Patrick O. Brown, and David Botstein. Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of Sciences*, 95(25):14863–14868, December 1998.

[53] Cristóbal Esteban, Stephanie L. Hyland, and Gunnar Rätsch. Real-valued (Medical) Time Series Generation with Recurrent Conditional GANs. *arXiv:1706.02633 [cs, stat]*, June 2017. arXiv: 1706.02633.

[54] Andre Esteva, Brett Kuprel, Roberto A. Novoa, Justin Ko, Susan M. Swetter, Helen M. Blau, and Sebastian Thrun. Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, 542(7639):115–118, February 2017.

[55] Fabian Falck, Yuyin Zhou, Emma Rocheteau, Liyue Shen, Luis Oala, Girmaw Abebe, Subhrajit Roy, Stephen Pfohl, Emily Alsentzer, and Matthew McDermott. A collection of the accepted abstracts for the machine learning for health (ml4h) symposium 2021. *arXiv e-prints*, pages arXiv–2112, 2021.

[56] Keyur Faldu, Amit Sheth, Prashant Kikani, and Hemang Akabari. Ki-bert: Infusing knowledge context for better language and domain understanding. *arXiv preprint arXiv:2104.08145*, 2021.

[57] Thibault Févry, Livio Baldini Soares, Nicholas FitzGerald, Eunsol Choi, and Tom Kwiatkowski. Entities as experts: Sparse memory access with entity supervision. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4937–4951, Online, November 2020. Association for Computational Linguistics.

[58] Tracey M. Filzen, Peter S. Kutchukian, Jeffrey D. Hermes, Jing Li, and Matthew Tudor. Representing high throughput expression profiles via perturbation barcodes reveals compound targets. *PLOS Computational Biology*, 13(2):e1005335, February 2017.

[59] Samuel G. Finlayson, Matthew B.A. McDermott, Alex V. Pickering, Scott L. Lipnick, and Isaac S. Kohane. Cross-modal representation alignment of molecular structure and perturbation-induced transcriptional profiles. In *Biocomputing 2021*. WORLD SCIENTIFIC, 2020.

[60] Vincent Fortuin, Matthias Hüser, Francesco Locatello, Heiko Strathmann, and Gunnar Rätsch. SOM-VAE: Interpretable Discrete Representation Learning on Time Series. *arXiv:1806.02199 [cs, stat]*, June 2018. arXiv: 1806.02199.

[61] Alex Fout, Jonathon Byrd, Basir Shariat, and Asa Ben-Hur. Protein Interface Prediction using Graph Convolutional Networks. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 6533–6542. Curran Associates, Inc., 2017.

[62] Hongchang Gao and Heng Huang. Deep attributed network embedding. In *Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI))*, 2018.

[63] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. Convolutional Sequence to Sequence Learning. *arXiv:1705.03122 [cs]*, May 2017. arXiv: 1705.03122.

[64] Dave Gershgorn. The data that transformed AI research—and possibly the world, July 2017.

[65] Marzyeh Ghassemi, Tristan Naumann, Finale Doshi-Velez, Nicole Brimmer, Rohit Joshi, Anna Rumshisky, and Peter Szolovits. Unfolding Physiological State: Mortality Modelling in Intensive Care Units. In *Proc. of the 20th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, KDD '14, pages 75–84, New York, NY, USA, 2014. ACM.

[66] Marzyeh Ghassemi, Tristan Naumann, Peter Schulam, Andrew L. Beam, Irene Y. Chen, and Rajesh Ranganath. Practical guidance on artificial intelligence for health-care data. *The Lancet Digital Health*, 1(4):e157–e159, August 2019.

[67] Marzyeh Ghassemi, Tristan Naumann, Peter Schulam, Andrew L Beam, Irene Y Chen, and Rajesh Ranganath. Practical guidance on artificial intelligence for health-care data. *The Lancet Digital Health*, 1(4):e157–e159, 2019.

[68] Marzyeh Ghassemi, Mike Wu, Michael C. Hughes, Peter Szolovits, and Finale Doshi-Velez. Predicting intervention onset in the ICU with switching state space models. *AMIA Summits on Trans. Sci. Proc.*, 2017:82–91, July 2017.

[69] Ian Goodfellow. NIPS 2016 Tutorial: Generative Adversarial Networks. *arXiv:1701.00160 [cs]*, December 2016. arXiv: 1701.00160.

[70] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc., 2014.

[71] Casey S. Greene, Arjun Krishnan, Aaron K. Wong, Emanuela Ricciotti, Rene A. Zelaya, Daniel S. Himmelstein, Ran Zhang, Boris M. Hartmann, Elena Zaslavsky, Stuart C. Sealfon, Daniel I. Chasman, Garret A. FitzGerald, Kara Dolinski, Tilo Grosser, and Olga G. Troyanskaya. Understanding multicellular function and disease with human tissue-specific networks. *Nature Genetics*, 47(6):569–576, June 2015.

[72] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. Improved Training of Wasserstein GANs. *arXiv:1704.00028 [cs, stat]*, March 2017. arXiv: 1704.00028.

[73] Varun Gulshan, Lily Peng, Marc Coram, Martin C. Stumpe, Derek Wu, Arunachalam Narayanaswamy, Subhashini Venugopalan, Kasumi Widner, Tom Madams, Jorge Cuadros, Ramasamy Kim, Rajiv Raman, Philip C. Nelson, Jessica L. Mega, and Dale R. Webster. Development and Validation of a Deep Learning Algorithm for Detection of Diabetic Retinopathy in Retinal Fundus Photographs. *JAMA*, 316(22):2402–2410, December 2016.

[74] Beliz Gunel, Jingfei Du, Alexis Conneau, and Veselin Stoyanov. Supervised contrastive learning for pre-trained language model fine-tuning. In *International Conference on Learning Representations*, 2021.

[75] Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. Don't Stop Pretraining: Adapt Language Models to Domains and Tasks. In *ACL*, 2020.

[76] R. Hadsell, S. Chopra, and Y. LeCun. Dimensionality Reduction by Learning an Invariant Mapping. In *CVPR*, 2006.

[77] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive Representation Learning on Large Graphs. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach,

R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 1025–1035. Curran Associates, Inc., 2017.

[78] William L Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 1025–1035, 2017.

[79] Michael Hansen, Masanori Koyama, Matthew McDermott, Michael E Orrison, and Sarah Wolff. Computational bounds for doing harmonic analysis on permutation modules of finite groups. *Journal of Fourier Analysis and Applications*, 2021.

[80] Simon Harris. AI in Medical Imaging to Top $2 Billion by 2023. Technical report, Signify Research, August 2018.

[81] Hrayr Harutyunyan, Hrant Khachatrian, David C Kale, Greg Ver Steeg, and Aram Galstyan. Multitask learning and benchmarking with clinical time series data. *Scientific data*, 6(1):1–18, 2019.

[82] Bin He, Xin Jiang, Jinghui Xiao, and Qun Liu. Kgplm: Knowledge-guided language model pre-training via generative and discriminative learning. *arXiv preprint arXiv:2012.03551*, 2020.

[83] Mikael Henaff, Joan Bruna, and Yann LeCun. Deep Convolutional Networks on Graph-Structured Data. *arXiv:1506.05163 [cs]*, June 2015. arXiv: 1506.05163.

[84] Geoffrey Hinton. Deep Learning—A Technology With the Potential to Transform Health Care. *JAMA*, 320(11):1101–1102, September 2018.

[85] Rachel Hodos, Ping Zhang, Hao-Chih Lee, Qiaonan Duan, Zichen Wang, Neil R. Clark, Avi Ma'ayan, Fei Wang, Brian Kidd, Jianying Hu, David Sontag, and Joel Dudley. Cell-specific prediction and application of drug-induced gene expression profiles. In *Biocomputing 2018*, pages 32–43. WORLD SCIENTIFIC, October 2017.

[86] Victoria Hore, Ana Viñuela, Alfonso Buil, Julian Knight, Mark I. McCarthy, Kerrin Small, and Jonathan Marchini. Tensor decomposition for multiple-tissue gene expression experiments. *Nature Genetics*, 48(9):1094–1100, September 2016.

[87] Jie Hou, Badri Adhikari, and Jianlin Cheng. DeepSF: deep convolutional neural network for mapping protein sequences to folds. *Bioinformatics*, (8), 2018.

[88] George Hripcsak, David J Albers, and Adler Perotte. Parameterizing time in electronic health record studies. *Journal of the American Medical Informatics Association*, 22(4):794–804, 02 2015.

[89] Tzu-Ming Harry Hsu, Wei-Hung Weng, Willie Boag, Matthew McDermott, and Peter Szolovits. Unsupervised multimodal representation learning across medical images and reports. *arXiv preprint arXiv:1811.08615*, 2018.

[90] Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open Graph Benchmark: Datasets for Machine Learning on Graphs. *arXiv: 2005.00687*, 2020.

[91] Weihua Hu, Bowen Liu, Joseph Gomes, Marinka Zitnik, Percy Liang, Vijay Pande, and Jure Leskovec. Pre-training Graph Neural Networks. *arXiv:1905.12265 [cs, stat]*, May 2019. arXiv: 1905.12265.

[92] Weihua Hu, Bowen Liu, Joseph Gomes, Marinka Zitnik, Percy Liang, Vijay Pande, and Jure Leskovec. Strategies for Pre-training Graph Neural Networks. In *ICLR*, 2020.

[93] Kexin Huang and Marinka Zitnik. Graph meta learning via local subgraphs. In *NeurIPS*, 2020.

[94] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. *CoRR*, abs/1611.07004, 2016.

[95] Di Jin, Franck Dernoncourt, Elena Sergeeva, Matthew McDermott, and Geeticka Chauhan. Mit-medg at semeval-2018 task 7: Semantic relation classification via convolution neural network. In *Proceedings of the 12th international workshop on semantic evaluation*, pages 798–804, 2018.

[96] Alistair E.W. Johnson, Tom J. Pollard, Lu Shen, Li-wei H. Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G. Mark. MIMIC-III, a freely accessible critical care database. *Scientific Data*, 3, May 2016.

[97] David Jurgens, Srijan Kumar, Raine Hoover, Dan McFarland, and Dan Jurafsky. Measuring the Evolution of a Scientific Field through Citation Frames. *Transactions of the ACL*, 6, 2018.

[98] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive Growing of GANs for Improved Quality, Stability, and Variation. *arXiv:1710.10196 [cs, stat]*, October 2017. arXiv: 1710.10196.

[99] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. *arXiv:1412.6980 [cs]*, December 2014. arXiv: 1412.6980.

[100] Thomas N. Kipf and Max Welling. Semi-Supervised Classification with Graph Convolutional Networks. *arXiv:1609.02907 [cs, stat]*, September 2016. arXiv: 1609.02907.

[101] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

[102] Günter Klambauer, Thomas Unterthiner, Andreas Mayr, and Sepp Hochreiter. Self-Normalizing Neural Networks. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 971–980. Curran Associates, Inc., 2017.

[103] Michael Schantz Klausen, Martin Closter Jespersen, Henrik Nielsen, Kamilla Kjaergaard Jensen, Vanessa Isabell Jurtz, Casper Kaae Sønderby, Morten Otto Alexander Sommer, Ole Winther, Morten Nielsen, Bent Petersen, and Paolo Marcatili. NetSurfP-2.0: Improved prediction of protein structural features by integrated deep learning. *Proteins*, (6), 2019.

[104] Matthieu Komorowski, Leo A. Celi, Omar Badawi, Anthony C. Gordon, and A. Aldo Faisal. The Artificial Intelligence Clinician learns optimal treatment strategies for sepsis in intensive care. *Nature Medicine*, 24(11):1716, November 2018.

[105] Alex M Lamb, Anirudh Goyal Alias Parth Goyal, Ying Zhang, Saizheng Zhang, Aaron C Courville, and Yoshua Bengio. Professor forcing: A new algorithm for training recurrent networks. In *Advances In Neural Information Processing Systems*, volume 29, pages 4601–4609, 2016.

[106] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. Neural Architectures for Named Entity Recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270, San Diego, California, June 2016. Association for Computational Linguistics.

[107] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. In *ICLR*, 2019.

[108] Jason D. Lee, Qi Lei, Nikunj Saunshi, and Jiacheng Zhuo. Predicting What You Already Know Helps: Provable Self-Supervised Learning. *arXiv: 2008.01064*, 2020.

[109] Yaniv Leviathan and Yossi Matias. Google Duplex: An AI System for Accomplishing Real-World Tasks Over the Phone, May 2018.

[110] Ron Levie, Federico Monti, Xavier Bresson, and Michael M. Bronstein. CayleyNets: Graph Convolutional Neural Networks with Complex Rational Spectral Filters. *arXiv:1705.07664 [cs]*, May 2017. arXiv: 1705.07664.

[111] Michelle M Li, Kexin Huang, and Marinka Zitnik. Representation learning for networks in biology and medicine: Advancements, challenges, and opportunities. *arXiv:2104.04883*, 2021.

[112] Ye Li, Chaofeng Sha, Xin Huang, and Yanchun Zhang. Community detection in attributed graphs: An embedding approach. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[113] Yu Li, Sheng Wang, Ramzan Umarov, Bingqing Xie, Ming Fan, Lihua Li, Xin Gao, and John Hancock. DEEPre: sequence-based enzyme EC number prediction by deep learning. *Bioinformatics*, 34(5):760–769, March 2018.

[114] Yaobo Liang, Nan Duan, Yeyun Gong, Ning Wu, Fenfei Guo, Weizhen Qi, Ming Gong, Linjun Shou, Daxin Jiang, Guihong Cao, Xiaodong Fan, Ruofei Zhang, Rahul Agrawal, Edward Cui, Sining Wei, Taroon Bharti, Ying Qiao, Jiun-Hung Chen, Winnie Wu, Shuguang Liu, Fan Yang, Daniel Campos, Rangan Majumder, and Ming Zhou. XGLUE: A new benchmark datasetfor cross-lingual pre-training, understanding and generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6008–6018. Association for Computational Linguistics, November 2020.

[115] Chieh Lin, Siddhartha Jain, Hannah Kim, and Ziv Bar-Joseph. Using neural networks for reducing the dimensions of single-cell RNA-Seq data. *Nucleic Acids Research*, 45(17):e156–e156, September 2017.

[116] Zachary C. Lipton, David C. Kale, Charles Elkan, and Randall Wetzel. Learning to Diagnose with LSTM Recurrent Neural Networks. *arXiv:1511.03677 [cs]*, November 2015. arXiv: 1511.03677.

[117] Guanxiong Liu, Tzu-Ming Harry Hsu, Matthew McDermott, Willie Boag, Wei-Hung Weng, Peter Szolovits, and Marzyeh Ghassemi. Clinically accurate chest x-ray report generation. In *Machine Learning for Healthcare Conference*, pages 249–269. PMLR, 2019.

[118] Weijie Liu, Peng Zhou, Zhe Zhao, Zhiruo Wang, Qi Ju, Haotang Deng, and Ping Wang. K-bert: Enabling language representation with knowledge graph. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 2901–2908, 2020.

[119] Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. Multi-Task Deep Neural Networks for Natural Language Understanding. In *ACL*, 2019.

[120] Ye Liu, Yao Wan, Lifang He, Hao Peng, and S Yu Philip. Kg-bart: Knowledge graph-augmented bart for generative commonsense reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 6418–6425, 2021.

[121] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv:1907.11692 [cs]*, 2019. arXiv: 1907.11692.

[122] Zhi-Ping Liu, Canglin Wu, Hongyu Miao, and Hulin Wu. RegNetwork: an integrated database of transcriptional and post-transcriptional regulatory networks in human and mouse. *Database*, 2015, January 2015.

[123] Sharon L Lojun, Christina J Sauper, Mitchell Medow, William J Long, Roger G Mark, and Regina Barzilay. Investigating resuscitation code assignment in the intensive care unit using structured and unstructured data. In *AMIA Annual Symposium Proceedings*, volume 2010, page 467. American Medical Informatics Association, 2010.

[124] Yinquan Lu, Haonan Lu, Guirong Fu, and Qun Liu. Kelm: Knowledge enhanced pre-trained language representations with message passing on hierarchical relational graphs. *arXiv preprint arXiv:2109.04223*, 2021.

[125] Yuan Luo, Peter Szolovits, Anand S Dighe, and Jason M Baron. Using machine learning to predict laboratory test results. *American journal of clinical pathology*, 145(6):778–788, 2016.

[126] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.

[127] Matthew McDermott. Fast Algorithms for Analyzing Partially Ranked Data. *HMC Senior Theses*, January 2014.

[128] Matthew McDermott, Emily Alsentzer, Sam Finlayson, Michael Oberst, Fabian Falck, Tristan Naumann, Brett K Beaulieu-Jones, and Adrian V Dalca. Ml4h abstract track 2019. *arXiv preprint arXiv:2002.01584*, 2020.

[129] Matthew McDermott, Bret Nestor, Evan Kim, Wancong Zhang, Anna Goldenberg, Peter Szolovits, and Marzyeh Ghassemi. A comprehensive ehr timeseries pre-training benchmark. In *Proceedings of the Conference on Health, Inference, and Learning*, pages 257–278, 2021.

[130] Matthew McDermott, Tom Yan, Tristan Naumann, Nathan Hunt, Harini S Suresh, Peter Szolovits, and Marzyeh Ghassemi. Semi-supervised biomedical translation with cycle wasserstein regression gans. In *AAAI Conference on Artificial Intelligence*, volume 32, pages 2363–2370, 2018.

[131] Matthew McDermott, Brendan Yap, Harry Hsu, Di Jin, and Peter Szolovits. Adversarial contrastive pre-training for protein sequences. *arXiv preprint arXiv:2102.00466*, 2021.

[132] Matthew McDermott, Brendan Yap, Peter Szolovits, and Marinka Zitnik. Structure inducing pre-training. *Journal of Machine Learning Research (under review)*, 2021.

[133] Matthew B. A. McDermott, Shirly Wang, Nikki Marinsek, Rajesh Ranganath, Marzyeh Ghassemi, and Luca Foschini. Reproducibility in Machine Learning for Health. In *Reproducibility in Machine Learning ICLR 2019 Workshop*, July 2019. arXiv: 1907.01463.

[134] Matthew B. A. McDermott, Tom Yan, Tristan Naumann, Nathan Hunt, Harini Suresh, Peter Szolovits, and Marzyeh Ghassemi. Semi-Supervised Biomedical Translation With Cycle Wasserstein Regression GANs. In *Thirty-Second AAAI Conference on Artificial Intelligence*, April 2018.

[135] Matthew BA McDermott, Tzu Ming Harry Hsu, Wei-Hung Weng, Marzyeh Ghassemi, and Peter Szolovits. Chexpert++: Approximating the chexpert labeler for speed, differentiability, and probabilistic output. In *Machine Learning for Healthcare Conference*, pages 913–927. PMLR, 2020.

[136] Matthew B.A. McDermott, Jennifer Wang, Wen-Ning Zhao, Steven D. Sheridan, Peter Szolovits, Isaac Kohane, Stephen J. Haggarty, and Roy H. Perlis. Deep Learning Benchmarks on L1000 Gene Expression Data. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2019.

[137] Zaiqiao Meng, Fangyu Liu, Thomas Hikaru Clark, Ehsan Shareghi, and Nigel Collier. Mixture-of-partitions: Infusing large biomedical knowledge graphs into bert. *arXiv preprint arXiv:2109.04810*, 2021.

[138] Seonwoo Min, Seunghyun Park, Siwon Kim, Hyun-Soo Choi, and Sungroh Yoon. Pre-Training of Deep Bidirectional Protein Sequence Representations with Structural Information. *arXiv: 1912.05625*, 2020.

[139] Riccardo Miotto, Li Li, Brian A. Kidd, and Joel T. Dudley. Deep Patient: An Unsupervised Representation to Predict the Future of Patients from the Electronic Health Records. *Scientific Reports*, 6:26094, May 2016.

[140] Aya A Mitani and Sebastien Haneuse. Small data challenges of studying rare diseases. *JAMA Network Open*, 3(3):e201965–e201965, 2020.

[141] Michael Moor, Max Horn, Bastian Rieck, and Karsten Borgwardt. Topological autoencoders. In *International Conference on Machine Learning*, pages 7045–7054. PMLR, 2020.

[142] Mohammad Amin Morid, Kensaku Kawamoto, Travis Ault, Josette Dorius, and Samir Abdelrahman. Supervised Learning Methods for Predicting Healthcare Costs: Systematic Literature Review and Empirical Evaluation. *AMIA Annual Symposium Proceedings*, 2017:1312–1321, April 2018.

[143] Youssef Mroueh, Tomaso Poggio, Lorenzo Rosasco, and Jean-Jacques E Slotine. Multiclass learning with simplex coding. In *Proceedings of the 25th International Conference on Neural Information Processing Systems-Volume 2*, pages 2789–2797, 2012.

[144] Amiya Mukherjee. Approximation Theorems and Whitney's Embedding. In *Differential Topology*, pages 43–67. Springer International Publishing, Cham, 2015.

[145] C. David Naylor. On the Prospects for a (Deep) Learning Health Care System. *JAMA*, 320(11):1099–1100, September 2018.

[146] Bret Nestor, Matthew McDermott, Geeticka Chauhan, Tristan Naumann, Michael C Hughes, Anna Goldenberg, and Marzyeh Ghassemi. Rethinking clinical prediction: why machine learning must consider year of care and feature aggregation. In *Proceedings of the 4th Machine Learning for Healthcare Conference*, volume 106, pages 381–405, 2019.

[147] Bret Nestor, Matthew B. A. McDermott, Willie Boag, Gabriela Berner, Tristan Naumann, Michael C. Hughes, Anna Goldenberg, and Marzyeh Ghassemi. Feature Robustness in Non-stationary Health Records: Caveats to Deployable Model Performance in Common Clinical Machine Learning Tasks. In *Proceedings of Machine Learning for Healthcare 2019 (MLHC '19)*, Ann Arbor, MI, August 2019. arXiv: 1908.00690.

[148] Bret Nestor, Matthew B. A. McDermott, Willie Boag, Gabriela Berner, Tristan Naumann, Michael C. Hughes, Anna Goldenberg, and Marzyeh Ghassemi. Feature robustness in non-stationary health records: Caveats to deployable model performance in common clinical machine learning tasks. In Finale Doshi-Velez, Jim Fackler, Ken Jung, David Kale, Rajesh Ranganath, Byron Wallace, and Jenna Wiens, editors, *Proceedings of the 4th Machine Learning for Healthcare Conference*, volume 106 of *Proceedings of Machine Learning Research*, pages 381–405, Ann Arbor, Michigan, 09–10 Aug 2019. PMLR.

[149] Jiquan Ngiam, Aditya Khosla, Mingyu Kim, Juhan Nam, Honglak Lee, and Andrew Y. Ng. Multimodal deep learning. 2011.

[150] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. WaveNet: A Generative Model for Raw Audio. *arXiv:1609.03499 [cs]*, September 2016. arXiv: 1609.03499.

[151] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[152] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, October 2011.

[153] Xuefeng Peng, Yi Ding, David Wihl, Omer Gottesman, Matthieu Komorowski, Li-wei H Lehman, Andrew Ross, Aldo Faisal, and Finale Doshi-Velez. Improving sepsis treatment strategies by combining deep and kernel-based reinforcement learning. In *AMIA Annual Symposium Proceedings*, volume 2018, page 887. American Medical Informatics Association, 2018.

[154] Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep Contextualized Word Representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[155] Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, 2018.

[156] Matthew E Peters, Mark Neumann, Robert Logan, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A Smith. Knowledge enhanced contextual word representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 43–54, 2019.

[157] Nina Poerner, Ulli Waltinger, and Hinrich Schütze. E-BERT: Efficient-yet-effective entity embeddings for BERT. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 803–818, Online, November 2020. Association for Computational Linguistics.

[158] Tom J Pollard, Alistair EW Johnson, Jesse D Raffa, Leo A Celi, Roger G Mark, and Omar Badawi. The eicu collaborative research database, a freely available multi-center database for critical care research. *Scientific data*, 5:180178, 2018.

[159] Niranjani Prasad, Li-Fang Cheng, Corey Chivers, Michael Draugelis, and Barbara E. Engelhardt. A Reinforcement Learning Approach to Weaning of Mechanical Ventilation in Intensive Care Units. *arXiv:1704.06300 [cs]*, April 2017. arXiv: 1704.06300.

[160] Yujia Qin, Yankai Lin, Ryuichi Takanobu, Zhiyuan Liu, Peng Li, Heng Ji, Minlie Huang, Maosong Sun, and Jie Zhou. Erica: improving entity and relation

understanding for pre-trained language models via contrastive learning. *arXiv preprint arXiv:2012.15022*, 2020.

[161] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2018.

[162] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[163] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *arXiv: 1910.10683*, 2019.

[164] Aniruddh Raghu, Jonathan Lorraine, Simon Kornblith, Matthew McDermott, and David K Duvenaud. Meta-learning to improve pre-training. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 23231–23244. Curran Associates, Inc., 2021.

[165] Alvin Rajkomar, Eyal Oren, Kai Chen, Andrew M Dai, Nissan Hajaj, Michaela Hardt, Peter J Liu, Xiaobing Liu, Jake Marcus, Mimi Sun, et al. Scalable and accurate deep learning with electronic health records. *NPJ Digital Medicine*, 1(1):18, 2018.

[166] Roshan Rao, Nicholas Bhattacharya, Neil Thomas, Yan Duan, Peter Chen, John Canny, Pieter Abbeel, and Yun Song. Evaluating Protein Transfer Learning with TAPE. In *NeurIPS*. Curran Associates, Inc., 2019.

[167] Roshan Rao, Nicholas Bhattacharya, Neil Thomas, Yan Duan, Peter Chen, John Canny, Pieter Abbeel, and Yun Song. Evaluating protein transfer learning with tape. In *Advances in Neural Information Processing Systems*, pages 9689–9701, 2019.

[168] Roshan Rao, Joshua Meier, Tom Sercu, Sergey Ovchinnikov, and Alexander Rives. Transformer protein language models are unsupervised structure learners. *bioRxiv preprint bioRxiv: 2020.12.15.422761*, 2020.

[169] Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative Adversarial Text to Image Synthesis. In *PMLR*, pages 1060–1069, June 2016.

[170] Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China, November 2019. Association for Computational Linguistics.

[171] Danilo Neves Ribeiro and Kenneth Forbus. Combining analogy with language models for knowledge extraction. In *3rd Conference on Automated Knowledge Base Construction*, 2021.

[172] Alexander Rives, Siddharth Goyal, Joshua Meier, Demi Guo, Myle Ott, C. Lawrence Zitnick, Jerry Ma, and Rob Fergus. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *bioRxiv*, 2019.

[173] Gabriel J. Rocklin, Tamuka M. Chidyausiku, Inna Goreshnik, Alex Ford, Scott Houliston, Alexander Lemak, Lauren Carter, Rashmi Ravichandran, Vikram K. Mulligan, Aaron Chevalier, Cheryl H. Arrowsmith, and David Baker. Global analysis of protein folding using massively parallel design, synthesis, and testing. *Science*, 357(6347), 2017.

[174] Thomas Rolland, Murat Taşan, Benoit Charloteaux, Samuel J. Pevzner, Quan Zhong, Nidhi Sahni, Song Yi, Irma Lemmens, Celia Fontanillo, Roberto Mosca, Atanas Kamburov, Susan D. Ghiassian, Xinping Yang, Lila Ghamsari, Dawit Balcha, Bridget E. Begg, Pascal Braun, Marc Brehme, Martin P. Broly, Anne-Ruxandra Carvunis, Dan Convery-Zupan, Roser Corominas, Jasmin Coulombe-Huntington, Elizabeth Dann, Matija Dreze, AmÍie Dricot, Changyu Fan, Eric Franzosa, Fana Gebreab, Bryan J. Gutierrez, Madeleine F. Hardy, Mike Jin, Shuli Kang, Ruth Kiros, Guan Ning Lin, Katja Luck, Andrew MacWilliams, Jörg Menche, Ryan R. Murray, Alexandre Palagi, Matthew M. Poulin, Xavier Rambout, John Rasla, Patrick Reichert, Viviana Romero, Elien Ruyssinck, Julie M. Sahalie, Annemarie Scholz, Akash A. Shah, Amitabh Sharma, Yun Shen, Kerstin Spirohn, Stanley Tam, Alexander O. Tejeda, Shelly A. Trigg, Jean-Claude Twizere, Kerwin Vega, Jennifer Walsh, Michael E. Cusick, Yu Xia, Albert-László Barabási, Lilia M. Iakoucheva, Patrick Aloy, Javier De Las Rivas, Jan Tavernier, Michael A. Calderwood, David E. Hill, Tong Hao, Frederick P. Roth, and Marc Vidal. A Proteome-Scale Map of the Human Interactome Network. *Cell*, 159(5):1212–1226, November 2014.

[175] Subhrajit Roy, Stephen Pfohl, Girmaw Abebe Tadesse, Luis Oala, Fabian Falck, Yuyin Zhou, Liyue Shen, Ghada Zamzmi, Purity Mugambi, Ayah Zirikly, et al. Machine learning for health (ml4h) 2021. In *Machine Learning for Health*, pages 1–12. PMLR, 2021.

[176] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 115(3):211–252, December 2015.

[177] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, Xi Chen, and Xi Chen. Improved Techniques for Training GANs. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in*

Neural Information Processing Systems 29, pages 2234–2242. Curran Associates, Inc., 2016.

[178] Victor Sanh, Albert Webson, Colin Raffel, Stephen H Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, et al. Multitask prompted training enables zero-shot task generalization. *arXiv preprint arXiv:2110.08207*, 2021.

[179] Suproteem K Sarkar, Subhrajit Roy, Emily Alsentzer, Matthew BA McDermott, Fabian Falck, Ioana Bica, Griffin Adams, Stephen Pfohl, and Stephanie L Hyland. Machine learning for health (ml4h) 2020: Advancing healthcare for all. In *Machine Learning for Health*, pages 1–11. PMLR, 2020.

[180] Sarkisyan et al. Local fitness landscape of the green fluorescent protein. *Nature*, 533(7603), 2016.

[181] Nikunj Saunshi, Sadhika Malladi, and Sanjeev Arora. A mathematical exploration of why language models help solve downstream tasks. In *International Conference on Learning Representations*, 2021.

[182] Nikunj Saunshi, Orestis Plevrakis, Sanjeev Arora, Mikhail Khodak, and Hrishikesh Khandeparkar. A Theoretical Analysis of Contrastive Unsupervised Representation Learning. In *ICML*, 2019.

[183] Andrew W Senior, Richard Evans, John Jumper, James Kirkpatrick, Laurent Sifre, Tim Green, Chongli Qin, Augustin Žídek, Alexander WR Nelson, Alex Bridgland, et al. Improved protein structure prediction using potentials from deep learning. *Nature*, 577(7792):706–710, 2020.

[184] Laleh Seyyed-Kalantari, Guanxiong Liu, Matthew McDermott, Irene Y Chen, and Marzyeh Ghassemi. Chexclusion: Fairness gaps in deep chest x-ray classifiers. In *BIOCOMPUTING 2021: Proceedings of the Pacific Symposium*, pages 232–243, 2020.

[185] Laleh Seyyed-Kalantari, Haoran Zhang, Matthew McDermott, Irene Y Chen, and Marzyeh Ghassemi. Underdiagnosis bias of artificial intelligence algorithms applied to chest radiographs in under-served patient populations. *Nature medicine*, 27(12):2176–2182, 2021.

[186] Thomas Shafee. Protein coding genes are transcribed to an mRNA intermediate, then translated to a functional protein. RNA-coding genes are transcribed to a functional non-coding RNA. (PDB: 3bse, 1obb, 3tra) Annotated version of not uploaded yet, April 2015.

[187] Junyuan Shang, Tengfei Ma, Cao Xiao, and Jimeng Sun. Pre-training of graph augmented transformers for medication recommendation. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 5953–5959. International Joint Conferences on Artificial Intelligence Organization, 7 2019.

[188] Blake Shaw, Bert Huang, and Tony Jebara. Learning a Distance Metric from a Network. In *NeurIPS*, 2011.

[189] Blake Shaw and Tony Jebara. Structure preserving embedding. In *ICML*, 2009.

[190] Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi Jaakkola. Style Transfer from Non-Parallel Text by Cross-Alignment. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 6830–6841. Curran Associates, Inc., 2017.

[191] Yuqi Si and Kirk Roberts. Deep patient representation of clinical notes via multi-task learning for mortality prediction. *AMIA Joint Summits on Translational Science proceedings. AMIA Joint Summits on Translational Science*, 2019:779–788, May 2019. 31259035[pmid].

[192] Yuqi Si, Jingqi Wang, Hua Xu, and Kirk Roberts. Enhancing clinical concept extraction with contextual embeddings. *Journal of the American Medical Informatics Association*, 26(11):1297–1304, 2019.

[193] Vergil N Slee. The international classification of diseases: ninth revision (icd-9). *Annals of internal medicine*, 88(3):424–426, 1978.

[194] Kihyuk Sohn, Wenling Shang, and Honglak Lee. Improved Multimodal Deep Learning with Variation of Information. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2141–2149. Curran Associates, Inc., 2014.

[195] Jost Tobias Springenberg. Unsupervised and Semi-supervised Learning with Categorical Generative Adversarial Networks. *arXiv:1511.06390 [cs, stat]*, November 2015. arXiv: 1511.06390.

[196] Ethan Steinberg, Ken Jung, Jason A. Fries, Conor K. Corbin, Stephen R. Pfohl, and Nigam H. Shah. Language models are an effective representation learning technique for electronic health record data. *Journal of Biomedical Informatics*, 113:103637, 2021.

[197] Jonathan M Stokes, Kevin Yang, Kyle Swanson, Wengong Jin, Andres Cubillos-Ruiz, Nina M Donghia, Craig R MacNair, Shawn French, Lindsey A Carfrae, Zohar Bloom-Ackermann, et al. A deep learning approach to antibiotic discovery. *Cell*, 180(4):688–702, 2020.

[198] Yusheng Su, Xu Han, Zhengyan Zhang, Yankai Lin, Peng Li, Zhiyuan Liu, Jie Zhou, and Maosong Sun. Cokebert: Contextual knowledge selection and embedding towards enhanced pre-trained language models. *AI Open*, 2:127–134, 2021.

[199] Aravind Subramanian, Rajiv Narayan, Steven M. Corsello, David D. Peck, Ted E. Natoli, Xiaodong Lu, Joshua Gould, John F. Davis, Andrew A. Tubelli, Jacob K. Asiedu, David L. Lahr, Jodi E. Hirschman, Zihan Liu, Melanie Donahue, Bina Julian, Mariya Khan, David Wadden, Ian Smith, Daniel Lam, Arthur Liberzon, Courtney Toder, Mukta Bagul, Marek Orzechowski, Oana M. Enache, Frederica Piccioni, Alice H. Berger, Alykhan Shamji, Angela N. Brooks, Anita Vrcic, Corey Flynn, Jacqueline Rosains, David Takeda, Desiree Davison, Justin Lamb, Kristin Ardlie, Larson Hogstrom, Nathanael S. Gray, Paul A. Clemons, Serena Silver, Xiaoyun Wu, Wen-Ning Zhao, Willis Read-Button, Xiaohua Wu, Stephen J. Haggarty, Lucienne V. Ronco, Jesse S. Boehm, Stuart L. Schreiber, John G. Doench, Joshua A. Bittker, David E. Root, Bang Wong, and Todd R. Golub. A Next Generation Connectivity Map: L1000 Platform And The First 1,000,000 Profiles. *bioRxiv*, page 136168, May 2017.

[200] Aravind Subramanian, Pablo Tamayo, Vamsi K. Mootha, Sayan Mukherjee, Benjamin L. Ebert, Michael A. Gillette, Amanda Paulovich, Scott L. Pomeroy, Todd R. Golub, Eric S. Lander, and Jill P. Mesirov. Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles. *Proceedings of the National Academy of Sciences*, 102(43):15545–15550, October 2005.

[201] Jimeng Sun, Junyuan Shang, Tengfei Ma, and Cao Xiao. Pre-training of Graph Augmented Transformers for Medication Recommendation. In *IJCAI 2019*. International Joint Conferences on Artificial Intelligence Organization, 2019.

[202] Tianxiang Sun, Yunfan Shao, Xipeng Qiu, Qipeng Guo, Yaru Hu, Xuanjing Huang, and Zheng Zhang. CoLAKE: Contextualized language and knowledge embedding. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3660–3670, Barcelona, Spain (Online), December 2020. International Committee on Computational Linguistics.

[203] Yu Sun, Shuohuan Wang, Shikun Feng, Siyu Ding, Chao Pang, Junyuan Shang, Jiaxiang Liu, Xuyi Chen, Yanbin Zhao, Yuxiang Lu, et al. Ernie 3.0: Large-scale knowledge enhanced pre-training for language understanding and generation. *arXiv preprint arXiv:2107.02137*, 2021.

[204] Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Xuyi Chen, Han Zhang, Xin Tian, Danxiang Zhu, Hao Tian, and Hua Wu. ERNIE: Enhanced Representation through Knowledge Integration. *arXiv:1904.09223 [cs]*, 2019. arXiv: 1904.09223.

[205] Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Hao Tian, Hua Wu, and Haifeng Wang. ERNIE 2.0: A Continual Pre-Training Framework for Language Understanding. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05), 2020. Number: 05.

[206] Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. Rotate: Knowledge graph embedding by relational rotation in complex space. In *International Conference on Learning Representations*, 2018.

[207] Krishnaswamy Sundararajan, Arthas Flabouris, and Campbell Thompson. Diurnal variation in the performance of rapid response systems: the role of critical care services—a review article. *Journal of intensive care*, 4(1):15, 2016.

[208] Harini Suresh, Nathan Hunt, Alistair Johnson, Leo Anthony Celi, Peter Szolovits, and Marzyeh Ghassemi. Clinical Intervention Prediction and Understanding using Deep Networks. In Finale Doshi-Velez, Jim Fackler, David Kale, Rajesh Ranganath, Byron Wallace, and Jenna Wiens, editors, *Proceedings of the 2nd Machine Learning for Healthcare Conference*, volume 68 of *Proceedings of Machine Learning Research*, pages 322–337, Boston, Massachusetts, 18–19 Aug 2017. PMLR.

[209] Harini Suresh, Peter Szolovits, and Marzyeh Ghassemi. The Use of Autoencoders for Discovering Patient Phenotypes. *arXiv:1703.07004 [cs]*, March 2017. arXiv: 1703.07004.

[210] Ian Tenney, Dipanjan Das, and Ellie Pavlick. BERT Rediscovers the Classical NLP Pipeline. In *ACL*, 2019.

[211] Martin J Tobin. *Principles and practice of mechanical ventilation.* LWW, 2006.

[212] Tractica. Artificial Intelligence for Healthcare Applications | Tractica. Technical report, 2018.

[213] Ngoc Hieu Tran, Xianglilan Zhang, Lei Xin, Baozhen Shan, and Ming Li. De novo peptide sequencing by deep learning. *Proceedings of the National Academy of Sciences*, 114(31):8247–8252, August 2017.

[214] Jean-Philipe Vert and Yoshihiro Yamanishi. Supervised graph inference. In *NeurIPS*, 2004.

[215] Kuansan Wang, Zhihong Shen, Chiyuan Huang, Chieh-Han Wu, Darrin Eide, Yuxiao Dong, Junjie Qian, Anshul Kanakia, Alvin Chen, and Richard Rogahn. A Review of Microsoft Academic Services for Science of Science Studies. *Frontiers in Big Data*, 2019.

[216] Ruize Wang, Duyu Tang, Nan Duan, Zhongyu Wei, Xuanjing Huang, Guihong Cao, Daxin Jiang, Ming Zhou, et al. K-adapter: Infusing knowledge into pre-trained models with adapters. *arXiv preprint arXiv:2002.01808*, 2020.

[217] Sheng Wang, Yuzhi Guo, Yuhong Wang, Hongmao Sun, and Junzhou Huang. SMILES-BERT: Large Scale Unsupervised Pre-Training for Molecular Property Prediction. In *ACM International Conference on Bioinformatics, Computational Biology and Health Informatics*, 2019.

[218] Sheng Wang, Jian Peng, Jianzhu Ma, and Jinbo Xu. Protein Secondary Structure Prediction Using Deep Convolutional Neural Fields. *Scientific Reports*, 6:18962, January 2016.

[219] Shirly Wang, Matthew B. A. McDermott, Geeticka Chauhan, Marzyeh Ghassemi, Michael C. Hughes, and Tristan Naumann. Mimic-extract: A data extraction, preprocessing, and representation pipeline for mimic-iii. In *Proceedings of the ACM Conference on Health, Inference, and Learning*, CHIL 2020, page 222–235, New York, NY, USA, July 2020. Association for Computing Machinery.

[220] Xiang Wang, Fei Wang, and Jianying Hu. A multi-task learning framework for joint disease risk prediction and comorbidity discovery. In *2014 22nd International Conference on Pattern Recognition*, pages 220–225. IEEE, 2014.

[221] Xiaozhi Wang, Tianyu Gao, Zhaocheng Zhu, Zhiyuan Liu, Juanzi Li, and Jian Tang. KEPLER: A Unified Model for Knowledge Embedding and Pre-trained Language Representation. *arXiv:1911.06136 [cs]*, 2019. arXiv: 1911.06136.

[222] Xun Wang, Xintong Han, Weilin Huang, Dengke Dong, and Matthew R. Scott. Multi-Similarity Loss With General Pair Weighting for Deep Metric Learning. In *CVPR*, 2019.

[223] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 28, 2014.

[224] Zichen Wang, Neil R. Clark, and Avi Ma'ayan. Drug-induced adverse events prediction with the LINCS L1000 data. *Bioinformatics*, 32(15):2338–2345, August 2016.

[225] Griffin M. Weber, Kenneth D. Mandl, and Isaac S. Kohane. Finding the Missing Link for Big Biomedical Data. *JAMA*, 311(24):2479–2480, June 2014.

[226] Hassler Whitney, James Eells, and Domingo Toledo. *Collected Papers of Hassler Whitney*, volume 1. Nelson Thornes, 1992.

[227] Sarah Wolff, Michael Hansen, Masanori Koyama, and Matthew McDermott. Computational bounds for doing harmonic analysis on permutation modules of finite groups. 2019.

[228] Mike Wu, Marzyeh Ghassemi, Mengling Feng, Leo A Celi, Peter Szolovits, and Finale Doshi-Velez. Understanding vasopressor intervention and weaning: risk prediction in a public heterogeneous clinical time series database. *Journal of the American Medical Informatics Association*, 24(3):488–495, 10 2016.

[229] Mike Wu, Marzyeh Ghassemi, Mengling Feng, Leo A. Celi, Peter Szolovits, and Finale Doshi-Velez. Understanding vasopressor intervention and weaning: risk prediction in a public heterogeneous clinical time series database. *Journal of the American Medical Informatics Association*, 24(3):488–495, May 2017.

[230] Yuan Xue, Nan Du, Anne Mottram, Martin Seneviratne, and Andrew M Dai. Learning to select best forecast tasks for clinical outcome prediction. *Advances in Neural Information Processing Systems*, 33, 2020.

[231] Ikuya Yamada, Akari Asai, Hiroyuki Shindo, Hideaki Takeda, and Yuji Matsumoto. Luke: Deep contextualized entity representations with entity-aware self-attention. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6442–6454, 2020.

[232] Ruiqing Yan, Lanchang Sun, Fang Wang, and Xiaoming Zhang. A general method for transferring explicit knowledge into language model pretraining. *Security and Communication Networks*, 2021, 2021.

[233] Junhan Yang, Zheng Liu, Shitao Xiao, Chaozhuo Li, Guangzhong Sun, and Xing Xie. Graphformers: Gnn-nested language models for linked text representation. *arXiv preprint arXiv:2105.02605*, 2021.

[234] Karl L Yang and Martin J Tobin. A prospective study of indexes predicting the outcome of trials of weaning from mechanical ventilation. *New England Journal of Medicine*, 324, 1991.

[235] Liang Yao, Chengsheng Mao, and Yuan Luo. Kg-bert: Bert for knowledge graph completion. *arXiv preprint arXiv:1909.03193*, 2019.

[236] Jinsung Yoon, Yao Zhang, James Jordon, and Mihaela van der Schaar. VIME: Extending the Success of Self- and Semi-supervised Learning to Tabular Domain. In *NeurIPS*, 2020.

[237] Jinsung Yoon, Yao Zhang, James Jordon, and Mihaela van der Schaar. Vime: Extending the success of self-and semi-supervised learning to tabular domain. *Advances in Neural Information Processing Systems*, 33, 2020.

[238] Donghan Yu, Chenguang Zhu, Yiming Yang, and Michael Zeng. Jaket: Joint pre-training of knowledge graph and language understanding. *arXiv preprint arXiv:2010.00796*, 2020.

[239] Zheng Yuan, Yijia Liu, Chuanqi Tan, Songfang Huang, and Fei Huang. Improving biomedical pretrained language models with knowledge. In *Proceedings of the 20th Workshop on Biomedical Language Processing*, pages 180–190, Online, June 2021. Association for Computational Linguistics.

[240] Hanqing Zeng, Hongkuan Zhou, Ajitesh Srivastava, Rajgopal Kannan, and Viktor Prasanna. Graphsaint: Graph sampling based inductive learning method. In *International Conference on Learning Representations*, 2020.

[241] Ana Cecilia Zenteno, Tim Carnes, Retsef Levi, Bethany J. Daily, and Peter F. Dunn. Systematic Or Block Allocation at a Large Academic Medical Center: Comprehensive Review on a Data-driven Surgical Scheduling Strategy. *Annals of Surgery*, 264(6):973–981, December 2016.

[242] Daokun Zhang, Jie Yin, Xingquan Zhu, and Chengqi Zhang. Homophily, structure, and content augmented network representation learning. In *ICDM*, 2016.

[243] Haoran Zhang, Amy X Lu, Mohamed Abdalla, Matthew McDermott, and Marzyeh Ghassemi. Hurtful words: quantifying biases in clinical contextual word embeddings. In *proceedings of the ACM Conference on Health, Inference, and Learning*, pages 110–120, 2020.

[244] Ningyu Zhang, Shumin Deng, Xu Cheng, Xi Chen, Yichi Zhang, Wei Zhang, Huajun Chen, and Hangzhou Innovation Center. Drop redundant, shrink irrelevant: Selective knowledge injection for language pretraining. In *In IJCAI*, 2021.

[245] Taolin Zhang, Zerui Cai, Chengyu Wang, Minghui Qiu, Bite Yang, and Xiaofeng He. Smedbert: A knowledge-enhanced pre-trained language model with structured semantics for medical text mining. *arXiv preprint arXiv:2108.08983*, 2021.

[246] Xiao-Chen Zhang, Cheng-Kun Wu, Zhi-Jiang Yang, Zhen-Xing Wu, Jia-Cai Yi, Chang-Yu Hsieh, Ting-Jun Hou, and Dong-Sheng Cao. Mg-bert: leveraging unsupervised atomic representation learning for molecular property prediction. *Briefings in Bioinformatics*, 2021.

[247] Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. ERNIE: Enhanced Language Representation with Informative Entities. In *Proceedings of the 57th Annual Meeting of the ACL*. ACL, 2019.

[248] Jian Zhou and Olga G. Troyanskaya. Predicting effects of noncoding variants with deep learning-based sequence model. *Nature Methods*, 12(10):931–934, October 2015.

[249] Henghui Zhu, Ioannis Ch Paschalidis, and Amir Tahmasebi. Clinical concept extraction with contextual word embedding. *arXiv preprint arXiv:1810.10566*, 2018.

[250] Jiong Zhu, Yujun Yan, Lingxiao Zhao, Mark Heimann, Leman Akoglu, and Danai Koutra. Generalizing graph neural networks beyond homophily. In *NeurIPS*, 2020.

[251] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. *arXiv:1703.10593 [cs]*, March 2017. arXiv: 1703.10593.

[252] Marinka Zitnik, Rok Sosič, Marcus W. Feldman, and Jure Leskovec. Evolution of resilience in protein interactomes across the tree of life. *Proceedings of the National Academy of Sciences*, 116(10), 2019.