Title
Towards More Generalizable Machine Learning: Improving Model Robustness Against Clinical Event Sequence Shifts

Author
Zhang, Tianran

Publication Date
2022

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA
Los Angeles

Towards More Generalizable Machine Learning: Improving Model Robustness Against Clinical Event Sequence Shifts

A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy in Bioengineering

by

Tianran Zhang

2022

ABSTRACT OF THE DISSERTATION

Towards More Generalizable Machine Learning: Improving Model Robustness Against Clinical Event Sequence Shifts

by

Tianran Zhang
Doctor of Philosophy in Bioengineering
University of California, Los Angeles, 2022
Professor Alex Ahn-Tuan Bui, Co-Chair
Professor William Hsu, Co-Chair

Data-driven models for diagnostic and other clinical prediction tasks have been enabled by the increasing availability of electronic health records (EHRs) and recent developments in machine learning (ML). Notably, the clinical event sequences extracted from EHR data provide important insights into how a patient's illness progresses. However, many of the models developed thus far are trained and validated using data from the same distribution (e.g., a single institutional dataset). When externally validated on distributions other than those used for training, these models exhibit generalizability issues despite their reported improvement. The variation in distributions between the training and deployment environment is called dataset shift, which can be attributed to many factors during the data generation process (e.g., patient demographics, site-specific healthcare delivery patterns, policy changes), and data processing approaches (e.g., concurrent event ordering, feature mapping). This problem and subsequent model generalization is exemplified by current approaches involving EHR data and clinical event sequences.

This dissertation seeks to assess and reduce the impact of dataset shift on the stability of clinical event sequence models, addressing two facets of the problem. First, the research explores a method to learn perturbation-invariant representations of event sequences involving concurrent events by modeling them as a sequence-of-sets, ameliorating the impact of dataset shift caused by inconsistent ordering schemes imposed during pre-processing. With a permutation-sampling-based framework, we enforce perturbation-invariance on a clinical dataset using an additional L1 loss. The proposed framework is tested on a next-visit diagnostic prediction task and shows improved robustness over perturbations in concurrent event ordering shifts. Second, this research develops a domain-invariant representation learning framework using unsupervised adversarial domain adaptation techniques, reducing the impact of dataset shift on a model's target domain performance without requiring any target labels. To improve transfer performance in the unlabelled target domain, the pre-trained Transformer-based framework adversarially learns domain-invariant features that are also beneficial to the discriminative task of next-visit diagnostic prediction. The proposed framework is evaluated for both transfer directions on event sequence datasets from two different healthcare systems and demonstrates superior zero-shot predictive performance on the target data over the non-adversarial baselines.

This dissertation advances our understanding of how dataset shift affects the generalization and stability of clinical event sequence diagnostic prediction models, and offers solutions to reduce its impact in both single-source perturbation and cross-dataset unsupervised transfer learning settings.

The dissertation of Tianran Zhang is approved.

Muhao Chen

Corey Wells Arnold

Ricky Kiyotaka Taira

William Hsu, Committee Co-Chair

Alex Ahn-Tuan Bui, Committee Co-Chair

University of California, Los Angeles

2022

To my family and friends.

When His lamp shone upon my head, And by His light I walked through darkness.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

Five years ago, when asked to make a toast to the senior lab members graduating, I said: 'Your graduation showed me the light at the end of the tunnel.' Now I am getting close to the other side. I hope it shows light to our fellow lab mates too, as they work towards their goals. I am grateful to all the people who walked with me in the tunnel. First, I would like to give thanks to my committee members: Prof. William Hsu, my co-chair, who recruited me through the summer program and taught me from scratch what research is; Prof. Ricky Taira, who inspired me with all the enlightening ideas and witty questions; Prof. Corey Arnold, who kindly served as a consultant when I was pondering on the career prospect; Prof. Muhao Chen, who closely collaborated with me across multiple projects and provided me with great inspirations in both research and life; and last but not least, Prof. Alex Bui, my advisor and co-chair, who trusted me, patiently waited for me when I was walking the bumpy road too slow, and at the same time showed me utmost rigor and diligence as a mature researcher. I would also like to thank other faculty members in our Medical & Imaging Informatics (MII) lab that I have worked with: Prof. Denise Aberle, who taught me the importance of clinical utility in what we do through inquisitive questions during the lab meetings; and Prof. William Speier, who helped me sort out my messy research ideas when I felt lost. It means a lot knowing that you all cared, it really does.

A sincere thank you also goes to staff members who did all the hard work to make my tunnel trip a smoother journey: Ms. Denise Luna, for helping me with the meeting schedules and reimbursements; Ms. Isabel Rippy, for her timely administrative assistance, Mr. Patrick Langdon, for all the technical help and kindly plant-sitting my bamboos during the pandemic; Dr. Cleo Maehara, for helping me with the IRB applications.

Thank you to my fellow labmates who were there throughout the journey, for all the help in research, fun times and foodie trips, and for all the memories we shared that I will cherish forever.

Lastly, I would like to express my gratitude and love to family and friends: to my parents and grandparents, for showing me the world with unconditional love, trust, and support; to Ruowen Hu, Tongxin Guo, Leiya Ma, Hangwen Lu, Yinong Zhao, Simon Han, Hui-hsuan Huang, and Roger Zou, who have stood by me through tears and laughter. I will always treasure the time I spent with you all.

Thank you, thank you, and thank you.

VITA

2011–2015	B.E. in Optical Information, BIT, Beijing, China.
2015–2019	M.S. in Bioengineering, UCLA, Los Angeles, USA.
2016–2019	Teaching assistant/associate, Department of Statistics/Life Sciences, UCLA, Los Angeles, CA.
2019–2021	Graduate Student Researcher at Department of Radiology, UCLA, Los Angeles, USA.
2017	Data Scientist intern at 3M Data Science Lab.
2020	Data Scientist intern at Facebook Inc.

PUBLICATIONS

***T. Zhang**, M. Chen, and A. A. T. Bui*, "Diagnostic Prediction with Sequence-of-sets Representation Learning for Clinical Events", Artificial Intelligence in Medicine (AIME). Conference on Artificial Intelligence in Medicine 2020, vol.12299, p. 348-358

***T. Zhang**, M. Chen, and A. A. T. Bui*, "ADABERT: Adversarial Domain Adaptation for Predictive Modeling of Clinical Event Sequences,", under review, 2022

M Chen, C. J.-T. Ju, G. Zhou, X. Chen, ***T. Zhang****, K. Chang, C. Zaniolo, and W. Wang*, "Multifaceted protein–protein interaction prediction based on Siamese residual RCNN", Bioinformatics 2019, vol.35, p. i305 - i314

CHAPTER 1

Introduction

1.1 Motivation

In recent years, the availability of electronic health records (EHRs) has enabled data-driven machine learning (ML) models that are able to diagnose conditions like skin cancer [34] and to make recommendations based on predicted risk for adverse events, like sepsis and in-hospital mortality [38]. The reliability and robustness of these models is especially crucial given their application in clinical practice. However, when tested and deployed in new environments different from the original development setting, they may perform drastically worse than reported. As an example, Zech et al. [135] demonstrated that pneumonia-screening CNNs trained on images from individual hospital systems failed to generalize consistently to external sites, with 3 of 5 transfer settings showing significantly lower external validation performance than that of the original hospital system. The generalizability issue is caused by shifting conditions between training and testing, which can be attributed to many factors including differences in patient demographics, healthcare delivery patterns, equipment choices, disease prevalence, and underlying data representations [108]. In practice, failing to account for these differences have consequences beyond just suboptimal model performance: a severely ill patient falsely triaged to the floor instead of the intensive care unit (ICU) could be under-treated as a condition worsens, for example. To generalize, a model needs to fulfill certain stability requirements to ensure that its performance is relatively robust against *perturbations* in data distribution. An ideal system will be able to encompass all the relevant variations in demographics, care patterns, and disease states of target patients in real-world

clinical settings.

Classical clinical models, such as the ones developed from clinical trials, have findings that are dependent on strictly controlled environments. When trying to generalize the conclusions to another population, additional procedures are required to either match the corresponding population-level statistics or make statistical adjustments to match the attributes between treatment groups [112]. Developing clinical predictive models (CPM) using EHR data makes controlling population/environmental factors much more difficult, as the data are routinely collected in healthcare practice rather than on an ad-hoc basis. CPM deployment must be safeguarded by better understanding the impact of dataset shift, and finding ways to reduce such impact in the dynamic and complex healthcare system.

These problems are explored throughout this dissertation to develop key ideas around ML generalization in healthcare applications, particularly in relation to the temporal nature of clinical data.

1.2 Contributions

This dissertation addresses the generalizability issue of clinical event sequence models by fulfilling the following two aims:

- **Aim 1**: *Improving the robustness of predictive modeling against perturbation by learning a perturbation-invariant representation of clinical event sequences.* This work is among the first efforts modeling a clinical sequence with concurrent events as sequence-of-sets (SOS) to offset the impact of inconsistent ordering from data management and preprocesssing steps. A permutation sampling based framework, Diagnostic Prediction with Sequence-of-sets (DPSS), is described to improve model robustness against varying ordering schemes. DPSS enforces permutation-invariant representation learning through a jointly trained L1 loss and demonstrated improvement on the next-visit heart failure (HF) diagnosis prediction task (on data with random ordering scheme) over baseline models

with no permutation-sampling mechanism. When a concurrent event ordering scheme different from that of the training data (e.g., random ordering versus alphabetical ordering) is applied as a test-time perturbation, DPSS shows reduced relative performance loss, supporting the conclusion that permutation invariance contributes to the model robustness against ordering scheme perturbation.

- **Aim 2**: *To externally validate a predictive event sequence model and improve its robustness against dataset shift by learning a domain-invariant representation of clinical event sequences.* While the inconsistent event ordering schemes in **Aim 1** is viewed as a type of perturbation that affects the distribution of data from a single source, **Aim 2** further explores the idea of improving model generalizability under perturbation. The perturbation is in the form of dataset shift in event sequences caused by cross-dataset distributional variation. Dataset shift is known for worsening testing/deployment-time performance when extending a model trained on one source (i.e., domain) to another (i.e., target) where the data distribution differs. An external validation is first performed using the state-of-the-art baseline models, self-attentive gated recurrent unit (GRU) and pre-trained Transformer model. Results show that the baseline models perform significantly worse on target data than on source data for HF onset prediction. A novel solution, AdaDiag, is proposed for unsupervised domain adaptation on event sequences, learning from unlabeled target domain sequences under dataset shift. It adversarially learns a domain classifier with the disease classifier through minimax loss optimization. A domain classifier determines a given sequence's domain (i.e., data source) identity and performs similar functions to a discriminator in conventional generative adversarial networks (GANs). This mechanism forces the feature extractor shared by both classifiers to learn a domain-invariant representation, which aids the model in generalizing between domains when used for disease classification. Compared with the non-adversarial baselines, AdaDiag method better utilizes data from both the source and target domains, which leads to improved performance without requiring labels from the target domain.

Motivating scenario. As a testbed for developing Aims 1 and 2, this dissertation looks at the clinical problem of predicting heart failure (HF). A person experiences heart failure when their heart cannot supply enough blood and oxygen to support the rest of their body. As one of the most frequent and serious conditions in the United States, heart failure is associated with difficulty with daily living activities, high costs of care and increased risk of hospitalization and mortality. Nearly 6.2 million Americans are affected by HF [121], costing the country $30.7 billion a year [6]. It has an approximately 50% mortality rate within 5 years of diagnosis [96] and accounts for 13.4% of all deaths [121]. Though early diagnosis and treatment can significantly improve the quality and length of life of patients with this disease, it is hard to detect before officially diagnosed, making it difficult to intervene promptly [15]. The significance of this clinical problem and the underlying temporal nature of observations provides a foundation for exploring and developing the methods in this dissertation.

1.3 Organization

The remaining chapters of this dissertation is organized as follows:

- Chapter 2 describes the technical background of major aspects of this dissertation: contemporary methods for modeling sequential data, types of dataset shift, and domain adaptation methods.
- Chapter 3 presents works on building a disease prediction framework that addresses the inconsistency issue of concurrent event ordering, showing that enforcing permutation-invariant representation improves model robustness against different ordering schemes.
- Chapter 4 discusses difficulties on transferring a predictive event sequence model across institutions due to the effect of dataset shift. An unsupervised adversarial domain adaptation framework is proposed and demonstrated its utility in reducing dataset shifts and improving the transferred HF onset prediction performance on the target dataset.

Finally, Chapter 5 summarizes the contributions and findings from this dissertation, and then

provides possible directions to extend these developments to serve the goal of improving the robustness of clinical predictive models against dataset shift.

CHAPTER 2

Background

This chapter provides an overview of current methods pertinent to the technical developments in this dissertation. We first discuss the challenges of representation learning for EHR data and current approaches to solving this problem in a broader context. Two specific areas are then described, highlighting state-of-the-art methods and approaches: 1) predictive models for sequential data, focusing on contemporary deep learning methods; and 2) domain adaptation methods. Details of comparable clinical models are presented in subsequent chapters.

2.1 Representation Learning for EHR Data

Data-driven approaches have the potential to explore and efficiently solve clinical problems due to the increased availability of electronic health record (EHR) data and rapid development of machine learning techniques. The handling of routinely collected data from various sources, however, can be challenging when preparing the raw data into structured inputs expected by the standard learning algorithms. With different pre-processing steps and encoding methods transforming sparse information into a more compact representation (e.g., embedding), there remains a risk of losing signals that would have been critical for clinical decision-making. Thus, it becomes especially important to learn a representation that preserves information that aligns well with medical knowledge needed for solving specific clinical problems.

A number of challenges have been identified in modeling temporal sequences in the EHR, including representing temporality, sparsity, high-dimensionality, and data heterogeneity. Here, this dissertation focuses on building a more robust representation for clinical event sequences in the face of dataset shift, whether due to bias in data generation or in handling and encoding processes. This chapter lays the foundation for later chapters by explaining the technical background of contemporary modeling techniques relevant to the dissertation. These techniques include temporal modeling techniques such as recurrent neural nets (RNNs) and attention mechanism (involved in both studies in Chapters 3 and 4); and subsequently, more advanced methods modeling sequential data such as Transformers, the basis of the model we describe in Chapter 4. Also at the heart of our innovation in Chapter 4, we introduce existing domain adaptation techniques to motivate its application.

2.2 Predictive Modeling of Sequential Data

2.2.1 Recurrent Neural Networks

Recurrent neural networks (RNN) take sequential data such as text, time series, and audio/video sequences. They are capable of processing variable-length sequence inputs using an internal memory unit structure. However, RNNs are not suitable for learning longer sequences as they may leave out important information from earlier time steps. During backpropagation, it suffers from the vanishing gradient problem: values used to update model weights shrink as optimization continues and do not significantly contribute to further learning. To control such gradient issues and address the limitations of short-term memory, RNN variants such as the long-short-term memory (LSTM)/gated recurrent unit (GRU) cells have been proposed. LSTMs and GRUs regulate the flow of information through internal mechanisms called gates, learning to selectively pass or forget information across time steps to support predictions.

LSTM. LSTM cells maintain an internal cell state and use three types of gates: input gates (I), forget gates (F), and output gates (O) to control information flowing across time steps. As shown in Eq. 2.1, 2.2 and 2.3, for an LSTM cell at time step t, each of its three gates takes the hidden state H_{t-1} from the previous time step and the current input X_t and passes them through a sigmoid function σ that "scales" the gate outputs between 0 and 1, providing the gates with the ability to remove or add information to the cell state. An *input gate* controls the flow of new values into the memory (Eq. 2.1), a *forget gate* controls how long a value remains in memory (Eq. 2.3), and an *output gate* controls how a value in memory is used to compute the cell's output hidden state (Eq. 2.2):

$$\mathbf{I}_t = \sigma\left(W_i \cdot \mathbf{X}_t + U_i \cdot \mathbf{H}_{t-1} + b_i\right) \tag{2.1}$$

$$\mathbf{O}_t = \sigma\left(W_o \cdot \mathbf{X}_t + U_o \cdot \mathbf{H}_{t-1} + b_o\right) \tag{2.2}$$

$$\mathbf{F}_t = \sigma\left(W_f \cdot \mathbf{X}_t + U_f \cdot \mathbf{H}_{t-1} + b_f\right) \tag{2.3}$$

To compute the current cell state, C_t, a candidate value $\tilde{C}_t$ is first calculated (Eq. 2.4) with the past hidden state H_{t-1} and the current input I_t, using a tanh function to scale $\tilde{C}_t$ between -1 and 1:

$$\tilde{\mathbf{C}}_t = \tanh\left(W_c \cdot \mathbf{X}_t + U_c \cdot \mathbf{h}_{t-1} + b_c\right) \tag{2.4}$$

The current cell state C_t is then updated using the previous cell state C_{t-1} and the candidate value $\tilde{C}_t$ weighted by the forget gate and input gate outputs F_t and I_t, respectively (Eq. 2.5):

$$\mathbf{C}_t = \mathbf{F}_t \odot \mathbf{C}_{t-1} + \mathbf{I}_t \odot \tilde{\mathbf{C}}_t \tag{2.5}$$

Finally, the output hidden state h_t is computed by the current cell state c_t passed through a tanh activation function, multiplied by the *output gate* output o_t (Eq. 2.6):

$$\mathbf{H}_t = \mathbf{O}_t \odot \tanh\left(\mathbf{C}_t\right) \tag{2.6}$$

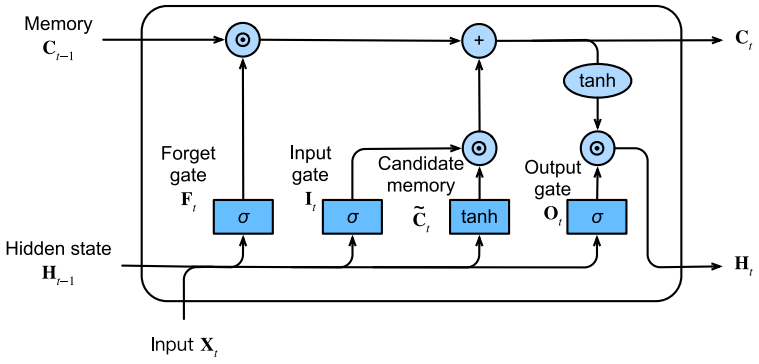


Figure 2.1: Gate operations and data flow in an LSTM cell [136].

In equations above, W_* and U_* are learnable weight matrices and b_* are the bias terms. The gate operations and data flow of an LSTM cell are illustrated in Fig. 2.1.

GRU. As a simpler alternative to LSTMs, GRUs [14] are proven to be more computationally efficient while achieving competitive performance for a variety of machine learning tasks. Instead of a cell state, it simply uses the hidden state to transfer information. Unlike the LSTM cell, a GRU cell has only two gates: the reset gate (R) and the update gate (Z). The update gate Z_t decides what information to add/throw away, while the reset gate R_t controls how much information from the previous time step to remember. These act similar to the input/forget gate in an LSTM cell as they also take the current input X_t and the previous hidden state H_{t-1} and pass through a fully-connected layer Sigmoid activation σ (Eq. 2.7, Eq. 2.8). A key difference between GRUs and LSTMs is that a GRU computes its hidden state, H_t, without maintaining a cell state. A candidate hidden state is first computed in Eq. 2.9 following an RNN's hidden state updating mechanism using tanh activation. Eq. 2.10 then incorporates the effect of the update gate Z_t and decides when constructing the final hidden state how much is directly taken from the previous hidden state H_{t-1}, and to what extent it uses the newly computed candidate hidden state, $\tilde{H}_t$:

$$\mathbf{Z}_t = \sigma\left(W_z \cdot \mathbf{X}_t + U_z \cdot \mathbf{H}_{t-1} + b_z\right) \tag{2.7}$$

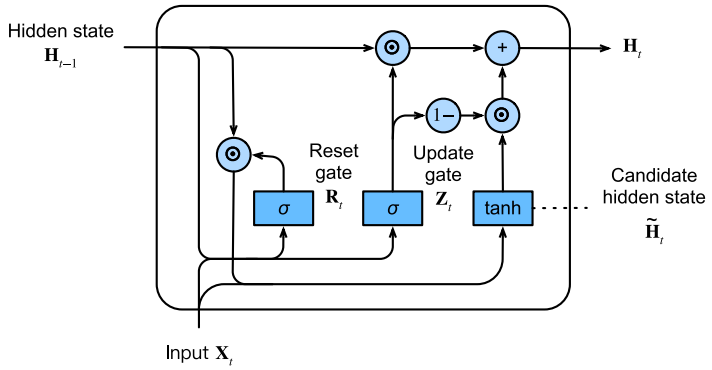


Figure 2.2: Gate operations and data flow in a GRU cell [136].

$$\mathbf{R}_t = \sigma\left(W_r \cdot \mathbf{X}_t + U_r \cdot \mathbf{H}_{t-1} + b_r\right) \tag{2.8}$$

$$\tilde{\mathbf{H}}_t = \tanh\left(W_h \cdot \mathbf{X}_t + U_h \cdot \mathbf{H}_{t-1}\left(\mathbf{R}_t \odot \mathbf{H}_{t-1}\right) + b_h\right) \tag{2.9}$$

$$\mathbf{H}_t = \mathbf{Z}_t \odot \mathbf{H}_{t-1} + (1 - \mathbf{Z}_t) \odot \tilde{\mathbf{H}}_t \tag{2.10}$$

The gate operations and data flow of a GRU cell are illustrated in Fig. 2.2.

Bi-directional RNNs. Bi-directional RNNs (bi-RNNs) were initially proposed in 1997 [102]. Later, bi-LSTMs [50] and bi-GRUs [137] were described with similar ideas. By adding another backward hidden layer that updates in the opposite direction, bi-RNNs look not only into historical sequence segments through left-to-right recurrent updates like RNNs do, they are able to look ahead into future tokens. For a given input X_t, a backward hidden state $\overleftarrow{H_t}$ and a forward hidden state $\overrightarrow{H_t}$ are populated by passing information in the backward and forward direction, and are then concatenated to construct the hidden state H_t and fed into the output layer. Bi-RNNs have the capability of learning more powerful encodings of input sequences than regular RNNs, as the added "look ahead" function provides deeper context: each hidden state at time step t is determined by both tokens before and after the current

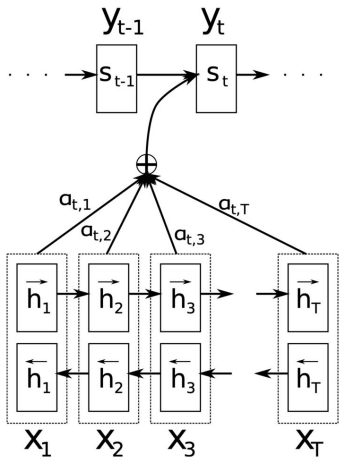


Figure 2.3: The Bahdanau architecture designed for NMT task: given an input sequence X_1, X_2, ..., X_T, the encoder (bi-RNN)-decoder (RNN) model tries to generate the target word y_t. Image is from [3].

step.

2.2.2 Attention Mechanisms

RNN-based methods have become widely adopted in recent years due to the remarkable performance of LSTMs and GRUs in tasks such as translation, speech recognition, and image captioning. A growing number of attempts have been made to enhance RNNs with new mechanisms and functionalities. Several promising directions share the same underlying idea of adding attention to RNNs, allowing them to focus on particular parts of their input (e.g., neural Turing machines [42], adaptive computation time [41]) [81].

In the encoder-decoder architecture of a classic sequence-to-sequence model (e.g., for neural machine translation, NMT), where the encoder/decoder both tend to be RNNs, the encoder first converts the input sequence into a fixed-length "context" vector as an intermediate representation. This context vector is then passed to the decoder to generate the output sequence. In spite of the fact that the tokens in the input sequence are not equally useful for

decoding each target token, the same context vector is used for predicting all tokens in the output sequence. This fixed context vector is unable to efficiently encode information from the entire input sequence (i.e., the bottleneck problem [3]). To improve model performance on NMT tasks, Bahdanau et al. [3] proposed a new mechanism called "attention." As shown in Fig. 2.3, the decoder RNNs process the input to pass information for each token it sees. At each decoding time step i, instead of a fixed context vector, the decoder takes a context vector $\mathbf{c}_i$ that is the weighted sum of all encoder outputs:

$$\mathbf{c}_i = \sum_{j=1}^{T_x} \boldsymbol{\alpha}_{ij} \mathbf{h}_j \tag{2.11}$$

where the attention weight $\boldsymbol{\alpha}_{ij}$ describes how much attention the output word at position i pays to the encoded representation of the input word at position j. This weighting is determined by an *alignment score* between the last decoder hidden state s_{i-1} and the encoder hidden states $\mathbf{h}_j$, normalized by a softmax function:

$$\boldsymbol{\alpha}_{ij} = softmax(score(\mathbf{s}_{i-1}, \mathbf{h}_j)) \tag{2.12}$$

where $score(s_{i-1}, \mathbf{h}_j)$ is the alignment score describing how well the j^{th} input token and the $(i-1)^{th}$ output match. The alignment model is parameterized as a feed-forward neural network, thus we have:

$$score(\mathbf{s}_i, \mathbf{h}_j) = \mathbf{v}_a^\top \tanh\left(\mathbf{W}_a \left[\mathbf{s}_i; \mathbf{h}_j\right]\right) \tag{2.13}$$

where $\mathbf{W}_a$ and $\mathbf{v}_a$ are trainable parameters. As a linear combination of the encoder hidden states $\mathbf{h}_j$ and decoder hidden state $\mathbf{s}_{i-1}$ is used to determine the attention weights, the Bahdanau attention is also known as the *additive attention*. Using Bahdanau attention, the decoder learns where to focus, generating an attention distribution that shows to what extent the model focuses on each segment of the input when generating each token in the output sequence. As a result, the model is advantageous relative to classic encoder-decoder architectures, especially when modeling longer sequences [3]. Given the success achieved by

Table 2.1: A summary of alignment scores for common attention functions.

Attention Mechanism	Alignment Scoring Function
Additive/Bahdanau [3]	$score(\mathbf{s}_i, \mathbf{h}_j) = v_a^\top \tanh\left(\mathbf{W}_a \left[\mathbf{s}_i; \mathbf{h}_j\right]\right)$
Dot-Product [70]	$score(\mathbf{s}_i, \mathbf{h}_j) = \mathbf{s}_i^\top \mathbf{h}_j$
General [70]	$score(\mathbf{s}_i, \mathbf{h}_j) = \mathbf{s}_i^\top \mathbf{W}_a \mathbf{h}_j$
Scaled Dot-Product [117]	$score(\mathbf{s}_i, \mathbf{h}_j) = \frac{\mathbf{s}_i^\top \mathbf{h}_j}{\sqrt{d}}$

attention mechanisms in machine translation, researchers have extended its application to other fields using sequential data, such as image caption generation [130].

Building on top of the Bahdanau attention, instead of using the feed-forward neural network, alternative alignment scoring functions can be used in place of Eq. 2.13 to construct new forms of attention [70, 117]. Table 2.1 provides a summary of alignment scores for commonly used attention mechanisms. Luong et al. proposed several alternative alignment scoring functions [70], where the dot-product score employs a dot product of the current decoder hidden state $\mathbf{s}_i$ and all the encoder hidden states $\mathbf{h}_j$, for $j \in [1, T]$; the general scoring is a parameterized version of the dot scoring with an intermediate matrix multiplication ($\mathbf{W}_a$ contains trainable parameters) step. The Transformer paper [117] later proposed an adapted version of the dot product scoring by adding a scaling factor $\frac{1}{\sqrt{d}}$, where d is the dimension of the source hidden state. Adding this scaling step prevents the gradient of the softmax function from becoming too small when the dot product grows too large as the hidden state dimension increases (as small gradients can impede learning).

Self-attention. Self-attention is a mechanism for learning a context-aware representation of a given sequence $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, ..., \mathbf{x}_T$, by relating tokens at different positions in the sequence. Each token $\mathbf{x}_i$ is mapped to an embedding $\mathbf{A}_i$ that is a weighted sum of all tokens in the sequence, as shown in Eq. 2.14:

$$\mathbf{A}_i = \sum_{j=1}^{T} \boldsymbol{\alpha}_{ij} \mathbf{x}_j \tag{2.14}$$

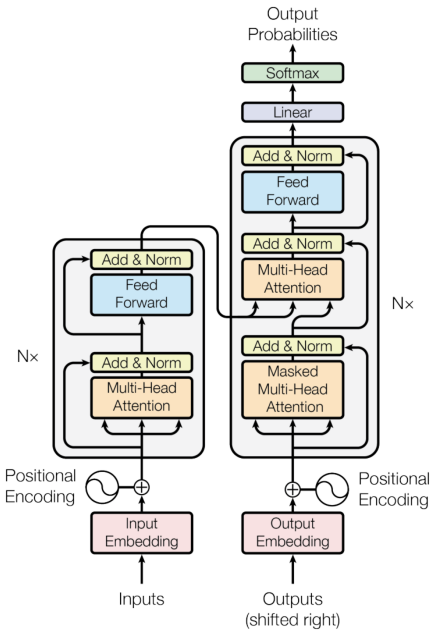


Figure 2.4: The Transformer model architecture. Image from Vaswani et al. [117].

The attention weights $\boldsymbol{\alpha}_{i,j}$ are calculated by:

$$\boldsymbol{\alpha}_{i,j} = \text{softmax}\left(score(\mathbf{x}_i, \mathbf{x}_j)\right) \tag{2.15}$$

where the score describing the association between tokens at position i and position j can be defined as any scoring function listed in Table 2.1. When dot-product scoring is used, the score is calculated as the dot-product of $\mathbf{x}_i$ and $\mathbf{x}_j$:

$$score(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^\top \mathbf{x}_j \tag{2.16}$$

2.2.3 Transformer Model

In comparison with earlier RNN-based models, the Transformer model relies entirely on a self-attention mechanism to compute representations for both input and output sequences. This approach boosts computational efficiency by allowing parallel computation.

Model architecture. As illustrated in Figure 2.4, the Transformer model is composed of an encoder (left) and a decoder (right). The encoder provides an attention-based representation capable of locating a specific piece of information out of the context of the entire sequence. It is comprised of a stack of several identical layers, each containing a multi-head self-attention sub-layer and a position-wise, fully-connected feed-forward network. Each layer also adopts residual connection and layer normalization. Likewise, the decoder consists of a stack of several identical layers: each has two sub-layers of multi-head attention mechanisms and one sub-layer of fully-connected feed-forward network. Residual connection and layer normalization are also used in the decoder sub-layers. Both the source and the target sequence are first passed through the embedding layer to be mapped to initial embeddings of a predefined dimension. To preserve the position information, a sinusoid-wave-based positional encoding is applied and summed with the embedding output. A softmax and linear layer are added to the final decoder output.

Query, key, and value. Taking an analogy of concepts from information retrieval systems, the Transformer paper [117] formulates an attention function as mapping a *query* and a set of *key-value* pairs to an output. To better facilitate language modeling, the self-attention used in [117] is introduced with three trainable matrices, $\mathbf{W}^q$, $\mathbf{W}^k$ and $\mathbf{W}^v$, projecting an input word x_i to its *query*, *key*, and *value*, denoted by row vectors $\mathbf{q_i}$, $\mathbf{k_i}$ and $\mathbf{v_i}$:

$$\begin{aligned} q_i &= \mathbf{W}^q \mathbf{x}_i \\ k_i &= \mathbf{W}^k \mathbf{x}_i \\ v_i &= \mathbf{W}^v \mathbf{x}_i \end{aligned} \tag{2.17}$$

query describes what the model wants to focus on (what the search query is asking for); *value* refers to the features/representation that we are running the queries on (the search results to be retrieved); and *key* stands for the values against which we bias the attention values given a query (corresponds to column/field names in database). Note that each *value* is therefore associated with a *key*.

Scaled dot-product attention. The type of self-attention used in the Transformer model adopts a scaled dot-product alignment function, which was briefly introduced in Table 2.1 from Section 2.2.2. Here, we formulate scaled dot-product attention using terms of *query*, *key*, and *value* we just introduced. The output is calculated as a weighted sum of *value*. For each *value*, a weight is given by a scaled dot-product alignment score assigned to the *query* and the corresponding *key*. For each *query*, the model learns which *key-value* pair it should pay attention to:

$$\mathbf{A}_n = A\left(q_n, \mathbf{K}, \mathbf{V}\right) = \sum_{i=1}^{T} \text{softmax}\left(score(q_n, k_i)\right) \mathbf{v}_i \tag{2.18}$$

$$score(q_n, k_i) = \left(\frac{q_n k_i^T}{\sqrt{d_k}}\right) \tag{2.19}$$

where d_k is the dimension of the *key* vectors. The scaling factor $\sqrt{d_k}$ prevents small gradient values of the softmax function as the dot product value becomes too large, thus facilitating a more efficient learning process.

The following equation shows how computation of attention weights for multiple queries can be performed in parallel through matrix multiplications (Eq. 2.20):

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}}\right)\mathbf{V} \tag{2.20}$$

where $\mathbf{Q}$, $\mathbf{K}$ and $\mathbf{V}$ are the queries, keys, and values in the form of matrices.

Multi-head self-attention. Initially proposed in [117], the idea behind multi-head attention is to run an attention mechanism (e.g., scaled dot-product self-attention) for several times in parallel, which allows handling information from different representation sub-spaces at the same time. Attention outputs from all heads are concatenated and converted into expected dimensions through a linear transformation:

$$\begin{aligned} \text{MultiHead}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) &= \text{Concat}\left(\text{head}_1, \ldots, \text{head}_h\right)\mathbf{W}^O \\ \text{where } head_i &= \text{ Attention } \left(\mathbf{Q}\mathbf{W}_i^Q, \mathbf{K}\mathbf{W}_i^K, \mathbf{V}\mathbf{W}_i^V\right) \end{aligned} \tag{2.21}$$

where $\mathbf{W}_i^Q$, $\mathbf{W}_i^K$, $\mathbf{W}_i^V$ and $\mathbf{W}^O$ are parameters to be learned.

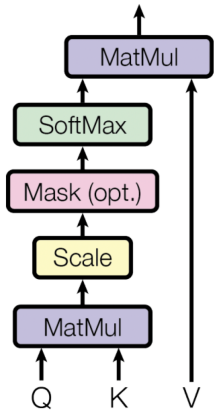


Figure 2.5: Scaled dot-product attention mechanism. Image from Vaswani et al. [117].

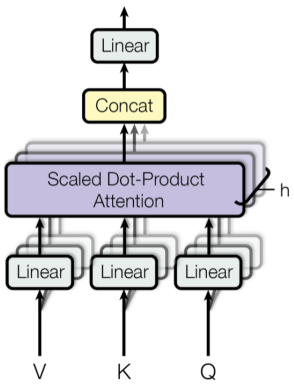


Figure 2.6: Illustration of the multi-head scaled dot-product attention mechanism. Image source: Vaswani et al. [117].

Positional encoding. As described thus far, a representation learned through multi-head self-attention does not account for the order of the words in the input sequence. To address this issue, the Transformer paper [117] uses positional encoding, adding a vector to the input embeddings. These positional encoding vectors follow a specific pattern that reflects the position of each word and distances between different words, providing meaningful distances between the embedding vectors. Specifically, [117] adopts sine and cosine functions of different frequencies for positional encoding, with each dimension of a positional encoding then defined as a sinusoid:

$$\begin{aligned} PE_{(pos,2i)} &= \sin\left(pos/10000^{2i/d_{\text{model}}}\right) \\ PE_{(pos,2i+1)} &= \cos\left(pos/10000^{2i/d_{\text{model}}}\right) \end{aligned} \tag{2.22}$$

where pos denotes the position of the input token; i represents the i^{th} dimension of the positional encoding; d_{model} refers to the dimension of the positional encoding vector, which is the same as the input embedding dimension.

2.2.4 BERT Model

The Transformer architecture's superior computational efficiency over RNNs and the power of learning contextualized representation has given rise to a series of Transformer-based language models [30, 91]. Bidirectional Encoder Representations from Transformers (BERT) is one of the most popular and widely adopted. Using the Transformer architecture as a building block, BERT was proposed in 2018 as a multi-layer bidirectional Transformer encoder based on the original implementation in [117], with each encoder layer referred to as a "Transformer block" [30]. Prior to BERT, OpenAI [91] proposed to build a Transformer-based language model that solves downstream tasks through a two-step pre-training/fine-tuning process. It uses a stacked Transformer decoder architecture, which is still a forward/unidirectional language model. The BERT model, in contrast to the OpenAI model, uses stacked Transformer encoders that can take the entire input sequence at one time. In this way, BERT accounts for context on both sides of a word. Similar to [91], BERT

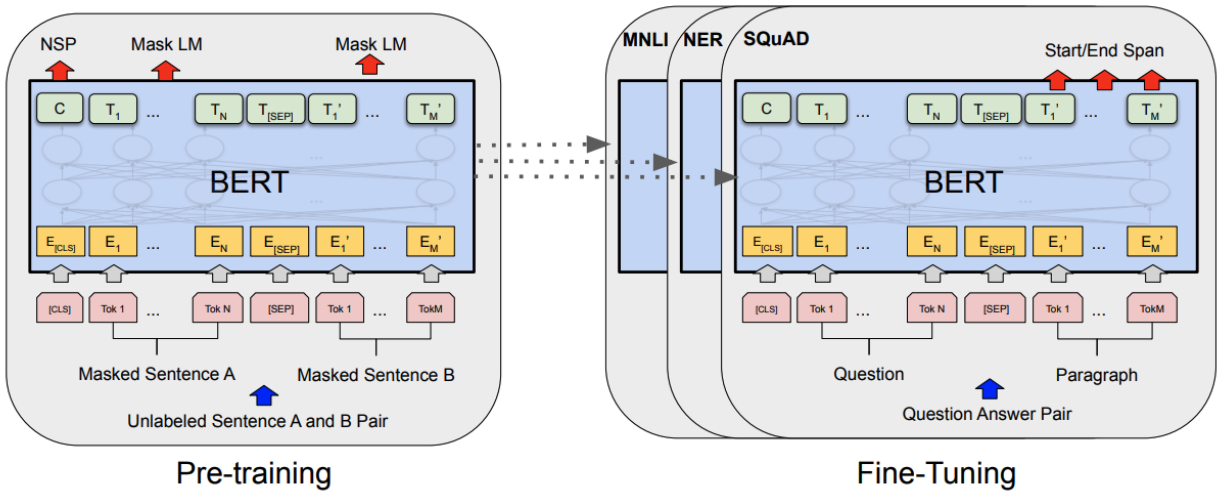


Figure 2.7: Overall pre-training and fine-tuning procedures for BERT. Image source: Devlin et al. [30]

also takes advantage of the two-step training approach that transfers knowledge from pre-training corpora to the downstream fine-tuning language tasks. BERT uses two pre-training tasks: 1) the masked language model task, which masks 15% of words and asks the model to predict the missing word; and 2) the next sentence prediction task, which predicts the likelihood of sentence B following sentence A, given the two sentences. By using parameters initialized from the pre-trained Transformer encoder model, the BERT paper demonstrated building task specific models formulated for a range of language tasks, including sentence classification, question answering, and sentence tagging, etc.. As a powerful representation learning method, the pre-trained BERT model from [30] is shared for public research use. Having the pre-trained BERT weights makes it easier for researchers to make use of robust context-aware representations and transfer them to other tasks without having to invest in expensive pre-training.

2.2.5 Adaptation to Modeling of Clinical Event Sequences

By making an analogy between words and events, these methods can be extended to model temporal events, even though they were originally proposed and widely used in applications involving images and text. In Chapters 3 and 4, we build clinical predictive models for sequences of abnormal lab events based on techniques introduced above.

2.3 Domain Adaptation and Transfer Learning

In Section 2.3 we first discuss common types of data shifts, motivating a discussion of domain adaptation methods, from shallow approaches to those which employ deep neural networks. This topic is further discussed in Chapter 4 due to the adoption of adversarial domain adaptation.

2.3.1 Dataset Shift

In machine learning, a dataset shift is the challenging situation where the shared distribution of input and output varies between training and testing stages [89]. In this section, we introduce three of the most common types of dataset shifts, with examples of these shifts in the clinical context provided: 1) covariate shift; 2) prior shift; and 3) concept shift.

Covariate shift. Covariate shift refers to a special case of dataset shift in which only the input distribution, X changes but the conditional distribution of output Y remains the same: $P^{src}(Y|X) = P^{tgt}(Y|X), P^{src}(X) \neq P^{tgt}(X)$. Note that throughout this dissertation, we use the superscripts src and tgt to indicate elements from source and target domain, respectively. Covariate shift is one of the most studied forms of dataset shift. A frequent cause of covariate shift is sample selection bias. For example, when a clinical dataset contains an underrepresented racial group (relative to the unbiased population), or when a dataset

used to assess disease treatment effect has a disease incidence rate different from the true, population-level prevalence, sample selection bias is considered present [114]. Another cause of covariate shift is data missingness, which is common in clinical practice when there is a consistent cause behind the missingness pattern (i.e., missing-not-at-random, MNAR). Examples are: missing certain measurements due to a sensor failure, a patient dropping out, missing follow-up studies, or skipping responses in surveys. In Chapter 3, we introduce a real-life covariate shift situation in the form of event set permutation, where clinical events recorded with the same timestamp may present with varied ordering in an event sequence. We demonstrate a method against such shift so as to improve model robustness across event ordering schemes by introducing a L2 loss to enforce permutation invariance in event sequence representation.

Prior shift. Opposite to covariate shift, the case of prior shift has different prior probabilities of the class labels in the source and the target domains, while the conditional distribution of the input remains the same: $P^{src}(X|Y) = P^{tgt}(X|Y), P^{src}(Y) \neq P^{tgt}(Y)$. This situation can arise when diagnostic tools are developed and used in different populations (e.g., regions/time periods) with different disease incidence rates, or when the population of interest (i.e., target) cannot be represented by the statistics of the training cohort (i.e., source).

Concept shift. Concept shift refers to the scenario where the relationship between the input and the output changes. The posterior distribution changes while the data distribution remains stable: $P^{src}(Y|X) \neq P^{tgt}(Y|X), P^{src}(X) = P^{tgt}(X)$. This issue is related to dataset drift, where classifiers are deployed in non-stationary environments [60, 125]. Concept shift can happen when the diagnostic criteria of a certain disease shifts over time. For instance, as one of the main causes of morbidity and mortality in critically ill patients, the definition of sepsis has evolved from the initial 1991 consensus definition (Sepsis-1) of, "systemic inflammatory response syndrome (SIRS) with infection" [90], to the 2001 revision (Sepsis-2) when sepsis and septic shock definitions were updated with the threshold for organ dam-

age [63]. The most recent update in 2016 (Sepsis-3) deviated from the previous versions by eliminating the criteria related to the SIRS symptoms. Sepsis-3 defines sepsis as, "a life-threatening organ dysfunction caused by a dysregulated host response to infection," where organ dysfunction is defined as an acute increase in total Sequential Organ Failure Assessment (SOFA) as a result of to the infection [105]. When an algorithm is applied over time for sepsis prediction, as the diagnostic of sepsis is evolving, the relationship of the input and the target (sepsis) is also changing, concept shift may occur. As another example, when a disease predictive model with International Classification of Disease (ICD) code labels applied on electronic health record (EHR) data from different institutions, concept drift may occur given variations in their local coding practices [82, 124].

2.3.2 Domain Adaptation Methods

Given the potential issues related to dataset shift, here, we briefly introduce domain adaptation techniques that help close the gap of generalizing machine learning models from one domain to another.

2.3.2.1 Shallow Domain Adaptation

Earlier domain adaptation efforts correcting the effect of dataset shift utilize hand-crafted features and traditional machine learning algorithms. We first introduce two popular shallow domain adaptation approaches:

1. *Instance weighting.* By re-weighting the source domain training samples to approximate the target domain distribution, instance weighting (i.e., importance sampling) is a statistical method for reducing sample selection bias in supervised machine learning models. By re-weighting the training samples based on density ratio, the sample selection bias can be mitigated. Essentially, the approach learns a higher weight for source instances that is more pertinent to the target instances to minimize the gap between the re-weighted distri-

bution of the source domain and that of the target domain. Instance weighting methods consists of two steps: 1) weight estimation (WE), and 2) weighted classification (WC). The WE step estimates the target-over-source density ratio, while the WC step trains the model using the weighted samples from the source domain. Instance weighting has been applied in natural language processing [53] and medical image analysis tasks [12]. However, despite working well with low capacity models (e.g., linear regression), instance weighting may or may not have any impact on deep neural networks (DNN) models depending on the model specifications (e.g., early stopping, batch normalization) [7].

2. *Feature transformation.* With the same principle of minimizing the distributional gap between the source and target domain, feature transformation methods focus on learning a new shared feature space where the model is less affected by the dataset shift than in the original feature space. Figure 2.8 shows the effect of a low-rank based feature transformation method on synthetic feature distributions from three different domains [123]. Blue squares indicate samples from the target domain, green circles and red triangles represent instances from the two source domains, respectively. Before the feature transformation (Figure 2.8(a)), the distributions of the three domains are distinctively separate from each other. After feature transformation (Figure 2.8(b)), data points from the three domains are well mixed together, showing the effectiveness of the proposed method of reducing distributional variance between domains by converting data from different sources into a shared latent space.

2.3.2.2 Deep Domain Adaptation

By using deep learning pipelines, deep domain adaptation is able to learn more powerful and transferable representations than shallow feature learning methods. Earlier domain adaptation research on supervised and semi-supervised methods requires a small amount of labeled data from the target domain [27, 28]. Over time, the use of deep neural networks to harness

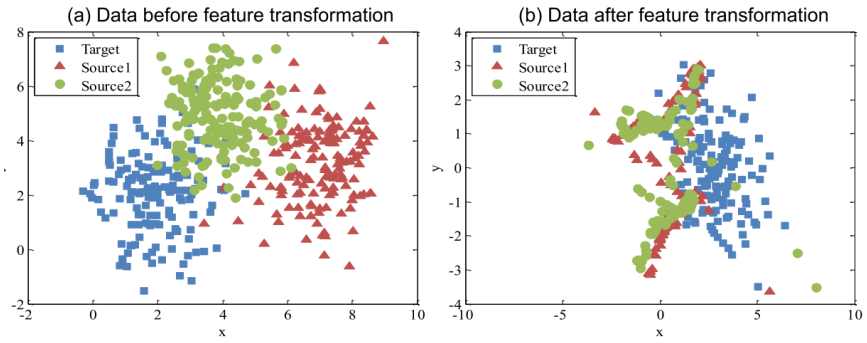


Figure 2.8: Results on synthetic data achieved by low-rank representation method in [123], figure adapted from the original paper.

the vast amounts of unlabeled data resulted in unsupervised methods (e.g., [109, 110]), becoming a popular domain adaptation approach thanks to its wide applicability in label-scarce domains. Here, we specifically introduce unsupervised domain adaptation (UDA) methods through domain adversaries, which is the most commonly used UDA approach [37, 93]. We cover major grounding techniques in this field as a prelude to the dissertation work in Chapter 4.

Domain-Adverserial Neural Network The idea of learning domain-invariant representations as an adversarial game was proposed by Ganin et al. [37] in 2016. The Domain-Adverserial Neural Network (DANN) [37] became a popular adaptive network that aims to make the domains indistinguishable while correctly classifying the samples in each domain. As with the feature-based shallow DA methods, they also aim to reduce the divergence between data distributions in source and target domains. This goal is achieved by using two losses: one for classifying the sample labels, and the other for classifying the sample's domain identity. During optimization, DANN minimizes the loss in label classification and maximizes the loss in domain classification. DANN thus learns features that are discriminative for the label classification task on the source domain while being domain-invariant. Figure 2.9 demonstrates the architecture of the DANN framework. It consists of: 1) a feature

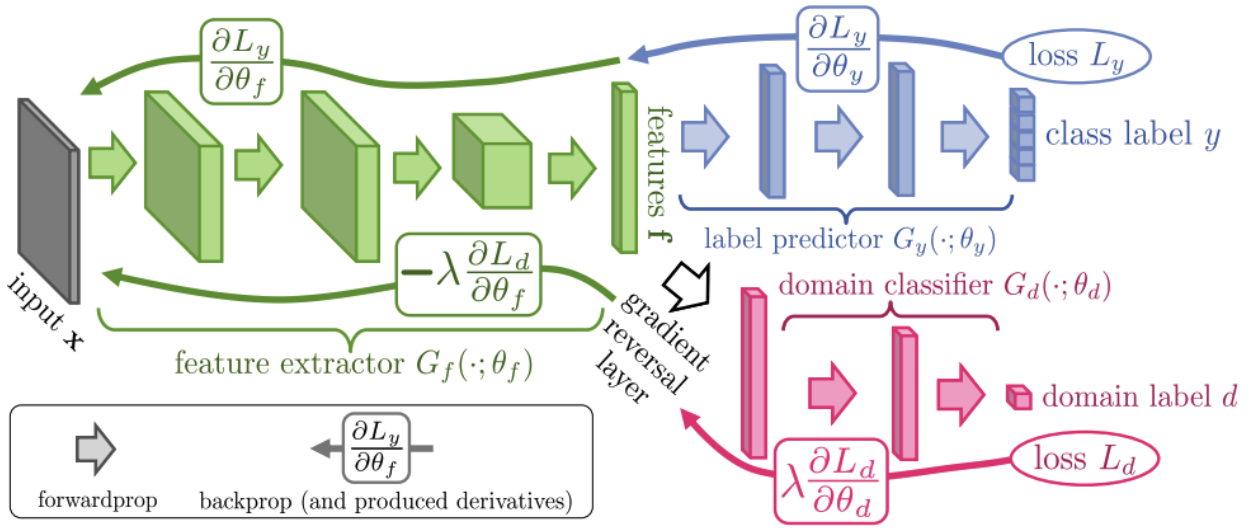


Figure 2.9: Illustration of DANN architecture, figure adapted from the original paper [37].

extractor G_f (in green), 2) a label predictor G_y (in blue) and 3) a domain classifier G_d (in red) that is connected to the feature extractor G_f through a *gradient reversal layer*. The *gradient reversal layer* (GRL) takes the gradient and multiple it with a negative number during back-propagation, to ensure that domain-invariant features are learnt.

The overall objective function is shown as below in Eq. 2.23. A negative sign is added before the domain classification error term to make sure the its loss is maximized when the overall loss function is minimized. A hyperparameter, λ, is introduced as a weighting factor balancing the learning of the label classifier and the domain discriminator.

$$\begin{aligned}\tilde{E}\left(\theta_f,\theta_y,\theta_d\right) &= \frac{1}{n}\sum_{i=1}^{n}\mathcal{L}_y\left(G_y\left(G_f\left(\mathbf{x}_i;\theta_f\right);\theta_y\right),y_i\right)\\ &-\lambda\left(\frac{1}{n}\sum_{i=1}^{n}\mathcal{L}_d\left(G_d\left(\mathcal{R}\left(G_f\left(\mathbf{x}_i;\theta_f\right)\right);\theta_d\right),d_i\right)+\frac{1}{n'}\sum^{N}\mathcal{L}_d\left(G_d\left(\mathcal{R}\left(G_f\left(\mathbf{x}_i;\theta_f\right)\right);\theta_d\right),d_i\right)\right)\end{aligned} \tag{2.23}$$

where θ_f, θ_y and θ_d are trainable parameters for the three model components G_f, G_y, and G_d; $\mathcal{R}(x)$ is the pseudo-function representing the forward/backward-propagation behavior

of GRL defined as:

$$\begin{aligned} \mathcal{R}(\mathbf{x}) &= \mathbf{x} \\ \frac{d\mathcal{R}}{d\mathbf{x}} &= -\mathbf{I} \end{aligned} \tag{2.24}$$

The main learning task loss is optimized over samples from the source domain while the domain discriminator loss is optimized over all N samples available in source and target domains. The first term in Eq. 2.23 is the main learning task prediction loss optimized over samples from the source domain of n samples, y_i denotes the true label for the i^{th} input, x_i. The second term weighted by $-\lambda$ is the sum of two sub-terms, each denoting the domain classification loss for the source domain of n samples and the target domain of n' samples. d_i denotes the true domain label for the i^{th} input, x_i.

DANN was validated on sentiment analysis datasets and achieved superior generalization performance over non-adversarial approaches such as neural networks (NNs) and support vector machines (SVMs).

Adversarial Discriminative Domain Adaptation Subsequent to the success of adversarial methods like DANN in reducing differences between training and test distributions and enhancing generalization performance, Adversarial Discriminative Domain Adaptation (ADDA) was developed with a standard GAN loss with inverted labels. This loss function is used to overcome the vanishing gradient issue when using a GRL layer to optimize for the minimax objective in DANN. As shown in Figure 2.10, the ADDA paper first proposed a generalized architecture of adversarial domain adaptation methods, which reduced the design decisions to three parts: 1) specifying if the base model is generative or discriminative; 2) determining if the source and target domain feature mapping weights tied or untied; and 3) choosing the adversarial loss. ADDA was designed with a unique combination of choices for these three aspects, making it a distinctly different method from other adversarial domain adaptation methods.

Adopting a discriminative base model, ADDA chose to untie the feature mapping weights

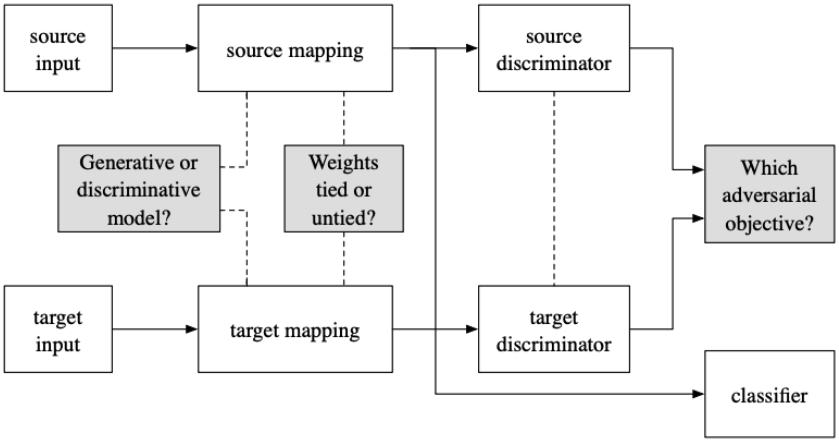


Figure 2.10: Illustration of generalized architecture for adversarial domain adaptation, figure adapted from the ADDA [115] paper.

in source and target domain mapping functions to allow independent feature learning. In this way, domain-specific features can be determined. Its adversarial loss is the inverted-label GAN loss defined as follows, where the adversarial domain discriminator D is optimized according to a standard supervised loss $\mathcal{L}_{\text{adv}_D}$, and the feature mapping M (generator) is trained with $\mathcal{L}_{\text{adv}_M}$, the standard loss function with inverted labels:

$$\mathcal{L}_{\text{adv}_D}\left(\mathbf{X}_s, \mathbf{X}_t, M_s, M_t\right) = -\mathbb{E}_{\mathbf{x}_s \sim \mathbf{X}_s}\left[\log D\left(M_s\left(\mathbf{x}_s\right)\right)\right] - \mathbb{E}_{\mathbf{x}_t \sim \mathbf{X}_t}\left[\log\left(1 - D\left(M_t\left(\mathbf{x}_t\right)\right)\right)\right] \quad (2.25)$$

$$\mathcal{L}_{\text{adv}_M}\left(\mathbf{X}_s, \mathbf{X}_t, D\right) = -\mathbb{E}_{\mathbf{x}_t \sim \mathbf{X}_t}\left[\log D\left(M_t\left(\mathbf{x}_t\right)\right)\right] \quad (2.26)$$

The subscripts s and t indicate if a variable is for the source or the target domain.

The learning objective of ADDA involves minimizing the classification loss for K label categories, where C denotes the classifier:

$$\min_{M_s, C} \mathcal{L}_{\text{cls}}\left(\mathbf{X}_s, Y_s\right) = -\mathbb{E}_{(\mathbf{x}_s, y_s) \sim (\mathbf{X}_s, Y_s)} \sum_{k=1}^{K} \mathbb{1}_{[k=y_s]} \log C\left(M_s\left(\mathbf{x}_s\right)\right) \quad (2.27)$$

and the two adversarial losses $\mathcal{L}_{\text{adv}_D}$ and $\mathcal{L}_{\text{adv}_M}$, which are optimized in an alternating fashion:

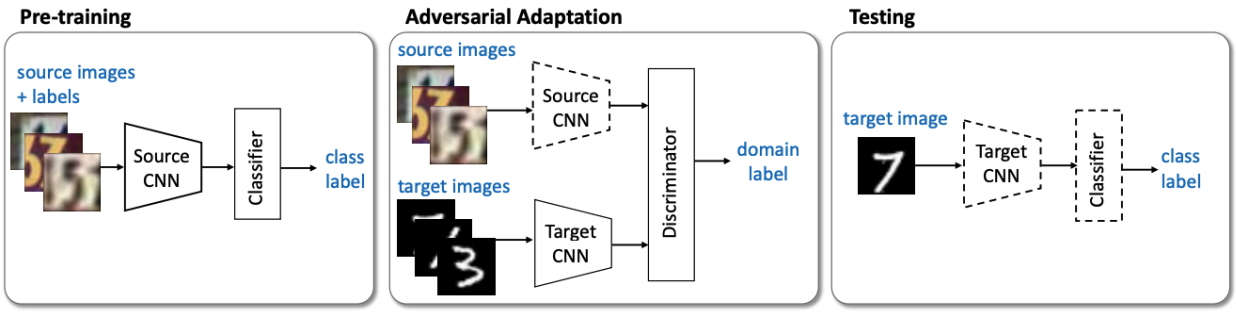


Figure 2.11: Illustration of the ADDA approach proposed in [115], figure adapted from the original paper.

$$\min_{D} \mathcal{L}_{\text{adv}_D}\left(\mathbf{X}_s, \mathbf{X}_t, M_s, M_t\right) = -\mathbb{E}_{\mathbf{x}_s \sim \mathbf{X}_s}\left[\log D\left(M_s\left(\mathbf{x}_s\right)\right)\right] - \mathbb{E}_{\mathbf{x}_t \sim \mathbf{X}_t}\left[\log\left(1 - D\left(M_t\left(\mathbf{x}_t\right)\right)\right)\right] \tag{2.28}$$

$$\min_{M_s, M_t} \mathcal{L}_{\text{adv}_M}\left(\mathbf{X}_s, \mathbf{X}_t, D\right) = -\mathbb{E}_{\mathbf{x}_t \sim \mathbf{X}_t}\left[\log D\left(M_t\left(\mathbf{x}_t\right)\right)\right] \tag{2.29}$$

ADDA's training and testing paradigm is outlined in Figure 2.11. A discriminative base model is first trained in the pre-training stage, then adversarial adaptation performs alternating optimization on the feature mappings and the domain discriminator. Note that the target feature mapping is initialized from the mappings in the pre-trained model, while the source mapping is kept fixed. Finally, at inference stage, the trained model is tested on target data to generate labels for the discriminative base task.

The utility of ADDA was verified by successfully performing two domain adaptation tasks: 1) multi-class classification on digits image datasets, and 2) cross-modality adaptation between RGB and depth image modalities.

We introduce ADDA as part of the background section in order to fit our Chapter 4 work in the generic framework summarized in ADDA and more clearly show the similarities and difference of our work and these foundational adversarial DA methods. Our work is similar

to DANN in the sense that we also had the two-branch design with tied weights. As for how adversarial training was implemented, DANN used gradient reversal for adversarial learning, ADDA used the standard GAN loss, since we used a Wasserstein distance to construct the adversarial loss, which is inspired by WGAN [2], our choice of adversarial loss is closer to ADDA than DANN.

CHAPTER 3

Diagnostic Prediction with Sequence-of-Sets

This chapter is adapted from the paper, "Diagnostic Prediction with Sequence-of-sets Representation Learning for Clinical Events" published in the Artificial Intelligence in Medicine (AIME) Conference in 2020 [138].

Electronic health records (EHRs) contain both ordered and unordered chronologies of clinical events that occur during a patient encounter. However, during data preprocessing steps, many predictive models impose a predefined order on unordered clinical events sets (e.g., alphabetical, natural order from the chart, etc.), which is potentially incompatible with the temporal nature of the sequence and predictive task. To address this issue, we propose Diagnostic Prediction with Sequence-of-Sets (DPSS), which seeks to capture each patient's clinical event records as sequences of event sets. For each clinical event set, we assume that the predictive model should be invariant to the order of concurrent events and thus employ a novel permutation sampling mechanism. This chapter evaluates the use of this permuted sampling method given different data-driven models for predicting a heart failure (HF) diagnosis in subsequent patient visits. Experimental results using the MIMIC-III dataset show that the permutation sampling mechanism offers improved discriminative power based on the area under the receiver operating curve (AUROC) and precision-recall curve (pr-AUC) metrics as HF diagnosis prediction becomes more robust to different data ordering schemes.

3.1 Introduction

Using the growing amounts of electronic health record (EHR) data, increasing attention has been paid to using data-driven machine learning (ML) methods for a range of classification and predictive tasks, including disease phenotyping and risk stratification [16, 49].

Implicit to these ML-based approaches are a data representation that embodies the temporal nature of such data. One challenge of modeling clinical event data is to learn the representation that aligns with medical knowledge [19, 21, 71], where events (i.e., laboratory results, medications, diagnoses, etc.) can be extracted from time-stamped EHRs and other health-related information, such as claims data. However, many studies modeling such data fail to fully capture the nature of clinical events. For instance, studies modeling claim code sequences only consider temporality between visits, absent of within-visit dynamics [80] that contain essential contextual information. While other approaches utilizing time-stamped EHR events incorporate sequential order within-visit [35, 72], they model a patient's medical history as a fully ordered event sequence despite the fact that the sequence may contain unordered event sets when multiple events happen concurrently (i.e., sharing the same timestamp). An arbitrary ordering (e.g., random, alphabetical, etc.) is usually imposed on each event set during data preprocessing to establish a "structured" input (e.g., matrices, vectors or tensors) used in different ML models, including contemporary deep learning methods. Consequently, models trained on the corresponding data can be sensitive to the input sequence order as they assume elements from each input sequence to be strictly ordered [120].

The partially-unordered nature of event sequences in the EHR calls for permutation-invariant models: the prediction based on a patient's medical history should not be affected when the order of concurrent events is changed. In this study, we propose DPSS (Diagnostic Prediction with Sequence-of-Sets), an end-to-end deep learning architecture that incorporates set learning techniques [134] to model event sequences to support downstream diagnostic prediction. DPSS first introduces a permutation sampling technique on each set of concurrent

clinical events. A self-attentive gated recurrent unit (GRU) model is then deployed on top of the permutation samples to characterize multiple sets of concurrent events in a patient visit history and correspondingly estimates the risk of specific diseases. To characterize the contextual features of a clinical event, DPSS also pre-trains an embedding model on a collection of unlabeled event sequences. The key contributions of DPSS are threefold: 1) an end-to-end framework modeling clinical temporal event sequences as *sequences of sets* (SoS) for next-visit disease code prediction, with the ability to capture the temporal patterns within each clinical visit; 2) a permutation-invariant prediction mechanism made possible by introducing a permutation sampling technique on SoS; and 3) a demonstration of the utility of a weighted loss function with additional regularization term enforcing permutation-invariant representation of SoS, which further improves the model predictive performance when using permuted sequences. In this way, DPSS is able to represent clinical event data as sequences of sets that are more consistent with the nature of clinical documentation processes.

We evaluate our proposed framework on a binary prediction task for next-visit diagnostic code prediction of heart failure (HF) using laboratory and diagnostic code data from the MIMIC-III dataset [56]. Our experimental results show that approaching clinical event sequence representation from a set learning perspective with permutation sampling more accurately characterizes the underlying disease dynamics and achieves better disease predictive performance. Techniques such as permutation sampling, sequence Laplacian regularization, and self-attention promote permutation invariance and contribute to robustness against different ordering schemes for concurrent events.

3.2 Related Work

3.2.1 Deep learning on clinical event sequences

Deep learning models, particularly variants of recurrent neural networks (RNN), have achieved some success in modeling sequential data for predictive tasks such as readmission and dis-

ease risk [5, 19, 20, 35, 129]. Early efforts in clinical event sequence representation learning focus on constructing low-dimensional representations of medical concepts through word embedding algorithms proposed for natural language processing (NLP) [24, 129]. Key works improved concept embedding by incorporating EHR structures [18, 19, 21, 23] and medical ontologies [106] to capture the inherent relations of medical concepts. More recent methods seek to utilize temporal information, instead of using the indexed ordering, to better characterize chronologies [8, 72, 85, 92]. Still, these aforementioned models mostly assume a fixed temporal order among sequence elements as they serve as inputs, which can cause discrepancies when modeling inputs containing unordered elements.

3.2.2 Deep set learning

Characterizing heterogeneous feature sets was investigated for applications in point cloud analysis [73, 88, 134, 140] and graph mining [47, 77]. Essentially, a permutation-invariant function is needed for set learning to overcome the limitations of sequence models that are permutation-sensitive [78]. Some of these and other works [78, 88, 134] propose to compress sets of any size into a feature vector using a permutation-invariant pooling operation (e.g., sum/mean/max pooling), although such operations are prone to losing information contained in a feature set [140]. In contrast, permutation sampling-based methods [73, 140] and attention-based methods [62] aim to resolve this issue. For example, Meng et al. [73] specifically use permutation sampling in a hierarchical architecture and concatenation to integrate set element embedding when modeling the structure as a set of sets.

Despite the partially-unordered nature of medical events, only a few studies [80] have been conducted to model clinical event sequences as sequence of sets using a permutation-invariant pooling method. There remains a lack of investigation in the use of permutation sampling strategies on corresponding tasks with EHR-based data, which is the focus of this chapter.

3.3 Method

In this section, we first present the design of the proposed framework, DPSS, for next-visit diagnostic code prediction. Fig. 3.1 illustrates the architecture of DPSS and its three components: 1) a pre-trained lab event embedding layer; 2) an event sequence handler with a permutation sampling mechanism for event sets; and 3) a self-attentive GRU predictor for diagnostic code classification.

3.3.1 Preliminary

We use E to denote the vocabulary of lab events, and P to denote the set of patient visit histories. A patient's visit history in the EHR is defined as a concatenation of lab event sets $S = [s_{t_1} \oplus s_{t_2} \oplus ... \oplus s_{t_n}] \in P$, where each set contains lab events with samples collected at the same time t_k, $s_{t_k} = \{e^1_{t_k}, e^2_{t_k}, ..., e^m_{t_k} \in E\}$. The goal of the diagnostic code prediction task is to provide a regression model to estimate the risk of developing a disease for a patient given the visit history S before the most recent visit. In this case, our goal is to predict codes related to HF.

3.3.2 The DPSS Framework

Our DPSS framework sequentially incorporates three components to characterize and perform prediction on a given patient's visit history. We first pre-train a lab event embedding model on a large collection of unlabeled historical lab event sequences, which seeks to capture the contextual similarity of lab events. Next, with this pre-trained embedding representing the latent features of each lab event, the permutation sampling process then generates permutations for each event set in the visit history. Lastly, a downstream predictor is trained on the permutation-sampled data, learning to predict the risk for a specific disease while preserving the permutation invariance of concurrent events. Details of each model component is described as follows.

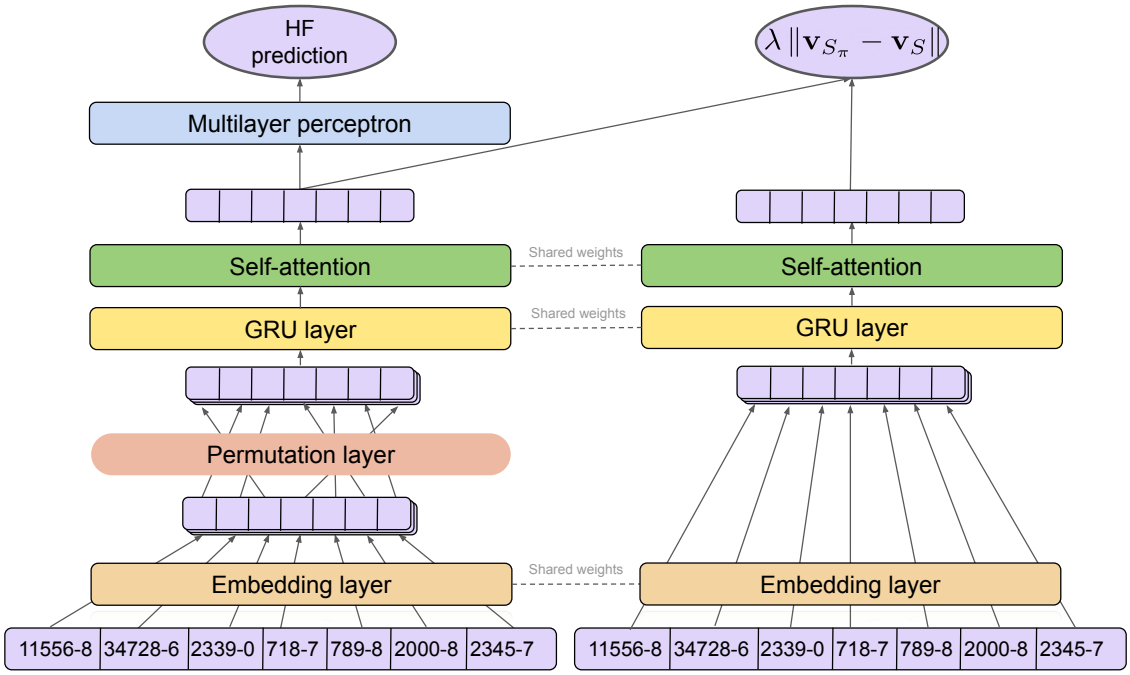


Figure 3.1: Illustration of DPSS architecture

3.3.2.1 Pre-trained lab event embeddings

To encode the non-numerical representations of lab events into numerical representations, we first conduct a pre-training process to obtain an embedding of Logical Observation Identifiers Names and Codes (LOINC) codes. We trained a skip-gram language model [75] on a collection of unlabeled lab event sequences with the objective of representing the contextual similarity of lab events in a continuous vector space (obtained by minimizing log likelihood loss):

$$L_{SG} = -\frac{1}{|P|} \sum_{\text{seq}(S) \in P} \sum_{-C<j<C} \log p(\mathbf{e}_{p+j} | \mathbf{e}_p).$$

such that seq(S) is a temporally-ordered sequence of a visit history S, and where events in each concurrent set are arbitrarily ordered. Specifically, we extract lab event sequences (from MIMIC-III) as partially-unordered sequences to train the embedding model. $\mathbf{e}_p$ is the embedding vector of the t-th event $e_t \in \text{seq}(S)$, $\mathbf{e}_{p+j}$ is that of a neighboring event, and C is

the size of half context.[1]

3.3.2.2 Permutation sampling

Rather than training a decision making model on fixed sequences, the learning objective of DPSS is to make consistent decisions even if such events may be observed in different orders; in our case, this may be dependent on any number of factors as to how an EHR records the data. Inspired by the recent success of deep set learning on point clouds [73, 78, 88, 134], we introduce a permutation sampling strategy for patient visit histories. The principle of this process is to generate event sequences from a given patient's visit history such that events in a concurrent event set will be randomly ordered in each training epoch, while the sequential order across event sets remain unchanged. In detail, given a set of events s, we denote $\pi(s)$ as the set of its permutations. A permutation sample of a visit history S is a sequence $S_\pi \in \pi(S) = \{\bigoplus_{i=1}^{n} \pi(s_{t_i})\}$ that is obtained by sequentially concatenating a permutation of each concurrent event set in S. Specifically, $\pi(S)$ denotes the universal set of permutation samples for S. Based on this sampling strategy, the event sequence encoder introduced next follows an end-to-end learning process for predicting the target diseases, while remaining invariant to the order of concurrent events in a patient visit history.

3.3.2.3 Self-attentive GRU encoder

We use $\mathbf{S}_\pi = [\mathbf{e}_1, \mathbf{e}_2, ..., \mathbf{e}_l]$ to denote an input vector sequence corresponding to an embedded lab event sequence after the permutation sampling process of the visit history, S. The self-attentive gated recurrent unit (GRU) encoder couples two techniques to represent the embedding representation of the permutation sampled visit history $\mathbf{v}_{S_\pi} = A(\mathbf{S}_\pi)$.

The GRU is an alternative to a long-short-term memory network (LSTM) [13], which

[1]The *context* of a skip-gram refers to a subsequence of an ordered event sequence seq(S) such that the subsequence is of $2C + 1$ length.

consecutively characterizes sequential information without using separated memory cells [31]. Each unit consists of two types of gates to track the state of the sequence, a reset gate $\mathbf{r}_p$ and an update gate $\mathbf{z}_p$. Given the embedding vector $\mathbf{e}_p$ of an incoming event, the GRU updates the hidden state $\mathbf{h}_p^{(1)}$ of the sequence as a linear combination of the previous state, $\mathbf{h}_{p-1}^{(1)}$, and the candidate state, $\tilde{\mathbf{h}}_p^{(1)}$ of a new event e_p, calculated as follows:

$$\begin{aligned}
\mathbf{h}_p^{(1)} &= \mathrm{GRU}(\mathbf{v}_p) = \mathbf{z}_p \odot \tilde{\mathbf{h}}_p^{(1)} + (1 - \mathbf{z}_p) \odot \mathbf{h}_{p-1}^{(1)} \\
\mathbf{z}_p &= \sigma\left(\mathbf{M}_z \mathbf{v}_p + \mathbf{N}_z \mathbf{h}_{p-1}^{(1)} + \mathbf{b}_z\right) \\
\tilde{\mathbf{h}}_p^{(1)} &= \tanh\left(\mathbf{M}_s \mathbf{v}_p + \mathbf{r}_p \odot (\mathbf{N}_s \mathbf{h}_{p-1}^{(1)}) + \mathbf{b}_s\right) \\
\mathbf{r}_p &= \sigma\left(\mathbf{M}_r \mathbf{v}_p + \mathbf{N}_r \mathbf{h}_{p-1}^{(1)} + \mathbf{b}_r\right).
\end{aligned}$$

where $\odot$ denotes the element-wise multiplication. The update gate $\mathbf{z}_p$ balances the information of the previous sequence and the new item, where $\mathbf{M}_*$ and $\mathbf{N}_*$ denote different weight matrices, $\mathbf{b}_*$ are bias vectors, and σ is the sigmoid function. The candidate state $\tilde{\mathbf{h}}_p^{(1)}$ is calculated similarly to those in a traditional recurrent unit, and the reset gate $\mathbf{r}_p$ controls how much information of the past sequence contributes to $\tilde{\mathbf{h}}_p^{(1)}$.

Atop the GRU hidden states, the self-attention mechanism seeks to learn attention weights that highlight the clinical events that are important to the overall visit history. This mechanism is added to GRU as below:

$$\mathbf{u}_i = \tanh\left(\mathbf{M}_a \mathbf{h}_i^{(1)} + \mathbf{b}_a\right);\ a_i = \frac{\exp\left(\mathbf{u}_i^\top \mathbf{u}_{S_\pi}\right)}{\sum_{x_i \in S_\pi} \exp\left(\mathbf{u}_i^\top \mathbf{u}_{S_\pi}\right)};\ A(S_\pi) = \mathbf{v}_{S_\pi} = \sum_{i=1}^{l} a_i \mathbf{u}_i$$

where $\mathbf{u}_i$ is the intermediary representation of GRU output $\mathbf{h}_i^{(1)}$. $\mathbf{u}_X = \tanh(\mathbf{M}_a \mathbf{h}_X^{(1)} + \mathbf{b}_a)$ is the intermediary latent representation of the averaged GRU output $\mathbf{h}_X^{(1)}$ and can be interpreted as a high-level representation of the entire input sequence. By measuring the similarity of each $\mathbf{u}_i$ with $\mathbf{u}_X$, the normalized attention weight a_i for $\mathbf{h}_i^{(1)}$ is produced through a softmax function. The final embedding representation $\mathbf{v}_{S_\pi}$ of the visit history is then obtained as the weighted sum of the intermediary representation for each event in the sequence S_π.

3.3.2.4 Learning objective

A multi-layer perceptron (MLP) with sigmoid activation is applied to the previous embedding representation of the visit history, whose output $\hat{c}^{S_\pi}$ is a scalar that indicates the risk of the target disease. The learning objective is to optimize the loss function defined below.

$$L = -\frac{1}{|P|}\sum_{S\in P}\frac{1}{|\pi(S)|}\sum_{S_\pi\in\pi(S)} x^S \log\sigma(\hat{c}^{S_\pi}) + (1-x^S)\log\left(1-\sigma(\hat{c}^{S_\pi})\right) + \lambda\left\|\mathbf{v}_{S_\pi}-\mathbf{v}_S\right\|$$

where $\|\mathbf{v}_{S_\pi}-\mathbf{v}_S\|$ represents the L_1 loss, measuring the distance between the sequence representation before and after permutation.

The main loss function uses binary cross-entropy, where $x^S \in \{0,1\}$ is the training label indicating if the disease code exists in the disease code list from the next patient visit $s_{t_{n+1}}$. Optimizing for the main loss enforces predictions to be invariant to the input within-set order. The last term of the loss function corresponds to a Laplacian regularization term, where λ is a small positive coefficient. Notably, this regularization term teaches the self-attentive GRU encoder to generate similar representations for different permutation samples of the same visit history record, and helps differentiate such representations from those of unrelated records in the embedding space. We show below that this regularization mechanism is able to improve the prediction accuracy of the target disease in various experiments.

3.4 Experiments

We hereby evaluate DPSS on the next-visit HF diagnosis prediction task.

3.4.1 Dataset

We evaluated DPSS using data from MIMIC-III [56], a publicly available clinical dataset associated with patients admitted to critical care units of Beth Israel Deaconess Medical Center between 2001 and 2012. MIMIC-III contains records from different sources including

demographics, lab results, medications, CPT (Current Procedural Terminology) procedures, and ICD-9 (International Classification of Diseases) diagnostic codes. The within-visit temporal information for diagnostic and procedure codes is not available in MIMIC-III as they are only specific to a patient visit; and while medications are tagged with timestamps, they are recorded with a duration (i.e., start and end times), which poses further challenges on determining the relative ordering between medication and lab events. To simplify our task, we choose to model only lab event sequences as they are less vague with respect to temporal ordering when defined as sequence of sets. Specifically, the timestamp recorded for lab events in MIMIC-III indicates sample acquisition time so a set of lab events with shared timestamps inform patient status at a given time point.

To perform next-visit HF diagnosis prediction, we extracted 7,235 sequences of abnormal lab events for adult ($age \geq 18$) patients with at least two hospital admissions from the MIMIC-III dataset by concatenating all abnormal lab events from each visit history. These sequences, each representing a unique patient, are divided into training (75%, 5,426 patients), validation (12.5%, 904 patients) and test (12.5%, 905 patients) datasets. Based on the existence of the level 3 ICD-9 code representing HF, 428, in the diagnostic codes of the most recent visit, we identified a total of 2,495 HF cases.We used LOINC codes as the lab event ontology, with 187 unique codes present in our data. During data preprocessing, all eligible event codes for a patient are extracted by patient ID and admission ID matching, sorted by chart time. Concurrent events during the same patient admission are usually imposed with an arbitrary order (e.g., random or alphabetically ordered event codes) when inputted as part of the sequence.

3.4.2 Experimental Configuration

We set the pre-trained skip-gram embedding model on LOINC codes with a context size of 5 and dimensionality of 256. For all reported models, we use the Adam optimizer [59] with a learning rate of 0.001. For each model variant or baseline, we select hyperparameters that

lead to the lowest validation loss during training for testing, with the maximum number of epochs set to 100. Training may also be terminated before 100 epochs based on early stopping with a patience of 10 epochs on the validation area under the receiver operator characteristic curve (AUROC) metric. The best combination of GRU layer dimension (candidate values from {64, 128, 256, 512}) and sequence length (candidate values: {128, 256, 512}) is selected based on the AUROC score on the validation set.

We compared the proposed method with the following baseline methods: 1) *GRU*, a single-layer GRU, as defined in Section 3.3.2.3; 2) *self-attentive GRU*, a GRU model incorporating the self-attention mechanism; and 3) *Pooling GRU*, following previous work [80,134], we apply a sum-pooling based or a max-pooling based set function on the set element embedding to acquire a permutation invariant feature aggregation. To show the effects of different model components of DPSS, we also evaluate different variants of DPSS, where we remove the sequence Laplacian regularization or self-attention.

3.4.3 Results

Experiments for baseline models and DPSS are each evaluated on the same holdout test set. We repeated the evaluations 10 times to calculate 95% confidence intervals (CIs) for test AUROC and pr-AUC. Table. 3.1 summarizes test performance of the baselines and DPSS.

DPSS significantly outperforms the other models in terms of AUROC and pr-AUC metrics. By comparing all of our permutation sampling based model variants with the baseline, we show that the effectiveness of addressing the partially-unordered nature through a permutation sampling mechanism. Specifically, being able to model within-set element interactions, DPSS is shown to be more suitable for modeling lab events as a sequence of sets compared to other permutation-invariant aggregation methods like sum- and max-pooling, with improvements of 9.8% and 11.7% in AUROC, 16.5% and 18.7% in pr-AUC, respectively and relatively. Comparing DPSS variants, we also see that sequentially adding the self-attention mechanism and the sequence Laplacian for permutation-invariant regularization boosted the

Table 3.1: Model comparison on next-visit HF risk prediction using MIMIC-III data

Method	AUROC (95% CI)	pr-AUC (95% CI)
GRU	0.7421(±0.00331)	0.6133(±0.00564)
Self-attentive GRU	0.7405(±0.0034)	0.6386(±0.0074)
Sum-pooling GRU	0.7070(±0.00101)	0.5839(±0.00173)
Max-pooling GRU	0.6954(±0.00116)	0.5730(±0.00361)
DPSS w/o self-attention&Sequence Laplacian	0.7741(±0.00277)	0.6659(±0.00615)
DPSS w/o Sequence Laplacian	0.7748(±0.00176)	0.6752(±0.00309)
DPSS	**0.7766(±0.00185)**	**0.6801(±0.00453)**

model's discriminative power, with greater improvement observed in pr-AUC, which is a metric that considers the model's ability to cope with imbalanced data [21]. As for the impact of the self-attention mechanism, when added to a basic GRU and DPSS without self-attention and Laplacian loss, the pr-AUC of both models has increased by 4.1% and 1.4%, respectively, while the AUROC metric remained comparable.

We observe that in the raw data of MIMIC-III, concurrent events are ordered randomly in the extracted event sequence. In other data processing scenarios, the event set elements are ordered by the primary key (when applicable) or alphabetically ordered by code strings. The imposed order could lead to bias toward certain data storage methods or a specific coding scheme, which is ultimately irrelevant to the underlying disease. Such inconsistencies may also impair a model's generalizability when the ordering scheme adopted in training differs from that used during inference. We hypothesized that our set learning framework is able to alleviate the aforementioned bias, as the sequence representation is not restricted to any event set ordering scheme. To test this hypothesis, as our previous experiments are trained and tested on data with random within-set order, we further compared DPSS and the best baseline model against a different event set ordering scheme using test sequences with alphabetically-ordered event sets. These evaluation results are presented in Table. 3.2.

Table 3.2: Comparison against the best baseline method on the test data with a different ordering scheme (alphabetical) for concurrent events.

Method	AUROC (95% CI)	pr-AUC (95% CI)
Self-attentive GRU	0.7364(±0.00953)	0.6214(±0.00878)
DPSS	0.7755(±0.00305)	0.6721(±0.00379)

The best baseline model, self-attentive GRU, is trained on set sequences with an imposed arbitrary random order. When tested on alphabetically-ordered set sequences, it suffers from 0.6% decrease in AUROC and 2.7% decrease in pr-AUC. In contrast, DPSS's performance experienced a smaller decline: 0.1% in AUROC and 1.2% in pr-AUC. The results suggest that DPSS benefited from its permutation sampling mechanism and is more robust against different set ordering schemes.

In summary, the experimental results show that DPSS achieved better performance than the non-permutation sampling-based baseline models on the HF prediction task. The proposed techniques are shown to better capture the clinical events in the visit history according to their partially-unordered nature, hence better supports the downstream decision making.

3.5 Conclusion

We introduce DPSS, a permutation-sampling-based RNN architecture that supports diagnostic prediction with sequence-of-set learning on clinical events. Our proposed method uses a permutation-sampling technique, sequence Laplacian regularization, and self-attention to learn a permutation invariant representation that allows for more accurate prediction for a binary disease prediction task. We also demonstrated the robustness of DPSS against arbitrary set orderings by comparing performance on a test set with an altered set order. For future work, we plan to extend DPSS to jointly model lab event sequences with medication and demographic information. We also seek to better support multi-disease prediction

by incorporating structured label representations [48] and leveraging pre-training [141] to improve domain adaptation of DPSS.

CHAPTER 4

AdaDiag: Adversarial Domain Adaptation of Diagnostic Prediction Model with Event Sequences

Early detection of heart failure (HF) can provide patients with the opportunity for more timely intervention and better disease management, as well as efficient use of healthcare resources. Recent machine learning (ML) methods have shown promising performance on diagnostic prediction using temporal sequences from electronic health records (EHRs). In practice, however, these models may not generalize to other populations due to dataset shift. Shifts in datasets can be attributed to a range of factors such as variations in demographics, data management methods, and healthcare delivery patterns. In this work, we use unsupervised adversarial domain adaptation methods to adaptively reduce the impact of dataset shift on cross-institutional transfer performance. The proposed framework is validated on a next-visit HF onset prediction task using a BERT-style Transformer-based language model pre-trained with a masked language modeling (MLM) task. Our model empirically demonstrates superior prediction performance relative to non-adversarial baselines in both transfer directions on two different clinical event sequence data sources.

4.1 Introduction

Recent research has demonstrated the advantages of deep learning (DL) methods for diagnostic prediction using clinical temporal sequences [22, 36, 64].

Despite the reported improvements in predicting outcomes, these models' actual clinical

impact still lags behind their projected potential. A critical reason for this unfilled promise is the inability to generalize findings beyond the development cohort/population [57, 95]. Due to data availability and sharing restrictions, most existing models are only internally validated using same-source, in-distribution data similar to the development data (e.g., from the same institution). Such models tend to fail, if not suffer from lower performance on independent external test cases from other sources and in different distributions [55, 99].

Existing work in transfer learning has thus explored ways to improve clinical model generalizability by utilizing EHRs from multiple sources [33, 45, 111]. One transfer learning method, domain adaptation (DA), leverages knowledge from a different but related domain to train models for decision making in a new target domain, given the same task in each domain both with varying distributions of data. This approach is particularly useful when the target domain lacks labeled data. For example, Desautels et al. [29] decreased the amount of target domain data needed to train a reliable mortality prediction model by training alongside an abundant source domain. Similarly, in a study by Sun et al. [111], performance in the target domain was improved by fine-tuning the source domain mortality prediction model on target domain data. Typically, these DA approaches require target domain ground truth for model fine-tuning, which are often scarce in clinical practice. Markedly, in cross-dataset transfer learning, the representation taken directly from the source domain is not domain-adaptive and may still fail to generalize to new data.

In contrast, more recent works on adversarial domain adaptation (ADA) adaptively learn a domain-invariant representation without requiring labels from the target domain. ADA combines adversarial training with discriminative feature learning to reduce the divergence between the source and target domain distribution, thus improving generalization performance [131]. Despite its successful use in myriad applications including bilingual sentiment classification [11], skin disease image classification [43], biological sequence classification [65] and clinical time series data classification [87, 113], ADA has not yet been investigated for mitigating the domain shift problem in medical event sequence classification.

To handle domain shift in event sequence classification, we propose AdaDiag (Adversarial Domain-Adaptive Diagnostic Prediction), an unsupervised adversarial domain adaptation framework with a pre-trained language model (LM) for clinical event sequences, to reduce the effects of domain shift when adapting a diagnostic prediction model from source to target domain. In this study, we specifically focus on alleviating domain shift across *patient cohorts*, where "domains" stands for *datasets* extracted from different EHR systems. The two datasets used as source and target domains are 1) the Medical Information Mart for Intensive Care IV (MIMIC-IV) dataset [39, 54]; and 2) data extracted from the UCLA Health Systems (hereafter referred to as UCLA data).

To demonstrate the utility of our proposed model, we adapt a heart failure (HF) onset prediction model trained on one patient cohort to another. Heart failure is one of the most frequent and serious conditions in the United States, contributing to one out of nine deaths [121]. For a patient with a developing set of symptoms but as of yet undiagnosed HF, it might take months or years before the next visit prior to HF is uncovered, during which time the disease progresses unchecked. For institutions with limited data availability/quality and/or model development resources – and hence, training a site-specific model is not a viable option – the ability AdaDiag offers in improving testing performance for externally trained models is especially meaningful. It can facilitate earlier detection and intervention by providing accurate predictions of next-visit incidence even when no labeled data is available from the target cohort.

AdaDiag's contributions are twofold. First, we construct a pre-trained Transformer-based LM [30, 117], fine-tuned for next-visit HF prediction on lab event sequences from one EHR dataset, and externally validate it on another dataset from a different institution. Our results show that although pre-trained LMs perform well when fine-tuned for the target task on the single data source, performance drops drastically when deployed to an institution with a shifted data distribution. Second, to address the generalizability issue against dataset shifts across institutions, we present an unsupervised domain adaptation framework for clinical

event sequences that addresses the domain shift problem by learning a domain-invariant representation through an adversarial domain classifier. This approach can adapt to the unseen target domain data distribution without requiring any labels. Notably, when source and target domains are switched, superior performance in adversarial-based methods persists, showing robustness of our proposed framework across different source and domain data quality settings.

4.2 Related Work

4.2.1 Clinical data representation

Medical events cover a wide array of clinical concepts, such as lab orders, medications, procedures, diagnoses, and myriad other observations. The management and storage of clinical event data pose *standardization* and *harmonization* challenges for transferring models between institutions. Events such as labs and medications are recorded under varying established and/or internal coding systems from each institution. Although endeavors are made to adapt events to a unified coding scheme (e.g., International Classification of Diseases (ICD); Logical Observation Identifiers Names and Codes (LOINC)) and/or ontology, manual mapping is often still needed in systems with local terminologies for data standardization. Data structures adopted in different systems create additional barriers to data harmonization. As an effort tackling this problem, researchers developed the Observational Medical Outcomes Partnership (OMOP) Common Data Model (CDM) [122] across multiple observational databases within an organization to facilitate standardized analytics tools when conducting observational research. The OMOP CDM streamlines data extraction process across multiple observational data sources, where different logical organizations and physical formats coexist. It also harmonizes disparate coding systems to an established standard vocabulary to prepare for the integrated analysis with all sources. Although these efforts improve access to multi-source data, they do not resolve any underlying domain shift problem.

As even with two event sequence datasets with the same format/coding system, site-specific characteristics such as patient demographics, disease prevalence and treatment patterns (e.g., procedure/lab/medication ordering habit, site policy), which cannot be explicitly standardized, still cause shifts in data distributions [108].

4.2.2 Diagnostic prediction over time

Modeling numerical clinical time series has been extensively investigated as a means to predict clinical outcomes [9, 58, 87, 133, 139]. There are fewer studies, however, that examine clinical event sequence modeling, which is also a critical part of appreciating the diagnostic prediction problem. A number of earlier works have explored methods to model medical event sequences using word embedding based on the co-occurrence of event codes [18,36,66]. Farhan et al. [36] model clinical abnormal lab sequences to provide next-visit diagnostic prediction using Word2Vec (i.e., skip-gram/CBOW) embeddings [74]. A different representation learned using another word embedding algorithm, GloVe [86], is demonstrated to be effective on next-visit code/risk group prediction [18] and 30-day readmission prediction [66]. However, with each word (event) represented by a fixed vector, these static embedding approaches cannot take into account the varying meanings of a given medical event based on the different patient histories it occurs in.

Pre-trained LMs for EHR data In light of the rapid development of pre-trained deep LMs such as BERT [30] in natural language processing (NLP), recent research has tested LMs for clinical event sequence representation learning by drawing an analogy between word sequences (text) and event sequences [17, 64, 94, 103, 107]. Some works have applied gated recurrent unit (GRU)-based LMs and achieved superior performance over more naive baselines [17,107]. DoctorAI [17] explored representing disease/medication code sequences to predict medical codes appearing in future patient encounters. [107] extends [17] by building clinical event sequences that include labs and procedures, and by evaluating a range of

shallow representation methods (e.g., Word2Vec) with logistic regression (LR) and gradient boosted trees (GBTs) for predicting mortality, long admission and other clinical outcomes. Following BERT's success in natural language, more recent studies utilized Transformer-based LMs trained on clinical event sequences to learn better representations that boost downstream task performance [64, 94, 103]. G-BERT [103] combined the power of graphical neural networks (GNN) and BERT by incorporating a medical ontology on top of a pre-trained LM to represent diagnosis code sequences more accurately for predicting medications. The BEHRT [64] and Med-BERT [94] studies pre-train a Transformer-based model from scratch on disease code sequences combined with structural information specific to the EHRs, achieving good fine-tuning performance on tasks such as prolonged length of stay (LOS) and disease prediction. In contrast to shallow embedding methods and other DL (e.g., recurrent neural network, RNN) methods, these Transformer-based models are able to distinguish and extract *different* semantic meanings of words based on their context, which corresponds with the different indications of a given medical event and observation of a disease trajectory.

Most of the aforementioned methods have largely relied on their capability of learning better representations optimized solely on data from a single population and/or dataset. Such models suffer from lack of robustness under domain shift. Moreover, when using a source domain model on a target population encountered in clinical practice (e.g., testing), target domain labels may not be available for retraining for any number of reasons. Our study thus focuses on solving the challenging problem of unsupervised domain adaptation (UDA) on clinical event sequence data.

Unsupervised domain adaptation in medicine Work has been done on unsupervised domain adaptation for medical image analysis through cross-modality [10, 32, 128], cross-vendor [132], and cross-site [127] adaptations. In other areas, UDA efforts have also been made in clinical NLP for negation detection [76], adapting detection algorithms across four corpora of clinical notes. In the context of EHR data modeling, where domains can be in-

terpreted as patient populations, UDA can be used to improve the performance of machine learning on a target patient group by mitigating the domain shift between one and another, yet related patient population [60,133,139]. Most existing work on clinical domain adaptation using EHR data focus on modeling numerical time series, bridging the gap between patient groups with different age distributions and/or other disparities [4, 69, 87, 133, 139]. Building on earlier ADA works (e.g., domain adversarial neural network [37]) and advancements in generative adversarial networks (GAN) [40], Luo et al. [69] designed a Wasserstein GAN (WGAN [2]) -based framework to improve cross-dataset transfer performance for electroencephalogram (EEG)-based emotion recognition. Purushotham et al. [87] take advantage of adversarial training and variational recurrent neural network (VRNN) [25] to learn latent temporal dependencies underlying EHR time series data adaptive across patient age groups. Similarly, [139] seeks to adversarially learn a domain-invariant representation of clinical time series for septic shock prediction with an LSTM-based framework, where domains are defined as patient groups divided by demographic attributes such as race, age, and gender. With a slightly different adversarial approach, [133] performed clinical time series augmentation by adding adversarial samples for improving the logistic regression (LR) model's generalizability across patient groups. Despite a similar focus on improving transportability across populations, these recent UDA studies are fundamentally different from earlier works that aim to extend the conclusions from randomized controlled trials (RCTs) [51], findings from epidemiology studies and public health decisions [100] to a distinct population with unknown outcomes. These studies [51, 100] use statistical methods to analyze and account for population-level (demographic) changes. In contrast, using EHR-based clinical prediction models with new datasets is more challenging as clinical environments are less controlled than those of classical clinical studies [26]. In view of this, recent UDA methods aim at designing an EHR data representation learning scheme that can not only adjust for differences in cohort demographics, but also distribution shifts inherent to the data generation and collection process (e.g., different lab ordering patterns, policy shifts), which cannot be

easily described and adjusted using classical statistical approaches.

4.3 Methods

We present AdaDiag, an adaptive deep learning framework designed to improve the unsupervised transfer performance on disease prediction tasks using clinical event sequences, moving from a labeled source domain to an unlabeled target domain. We first state the problem to be addressed and define the notations in Section 4.3.1, followed by an introduction to the AdaDiag framework with its main components detailed in Section 4.3.2. Sections 4.3.3 and 4.3.4 describe the two-stage training process of AdaDiag: (1) Transformer-based encoder pre-training, and (2) adversarial training.

4.3.1 Preliminary

Problem statement Predictive models derived from EHR data are often developed and validated on the same population, and yet show a great decline when deployed/tested on external data due to dataset shift [108]. For instance, when a model trained on a national/multi-institutional dataset is used on data from a regional hospital, direct transfer performance may be sub-optimal due to site-specific data generation/storage processes. Here, we addresses the issue of transferring an event sequence diagnostic prediction model from a *source* dataset, where it was developed and trained, to another, *target* dataset, where it could be applied without requiring its disease labels, as an *unsupervised domain adaptation* problem.

The diagnostic prediction task seeks to estimate the likelihood of patients' disease onset based on their visit histories. For instance, the next-visit HF diagnosis prediction task is based on predicting the *first* appearance of HF-related ICD-9/10 codes during the most recent visit, given the combined event history from all past visits of the patient. To differentiate between elements from the two data domains, we use superscripts *src* and *tgt* to indicate

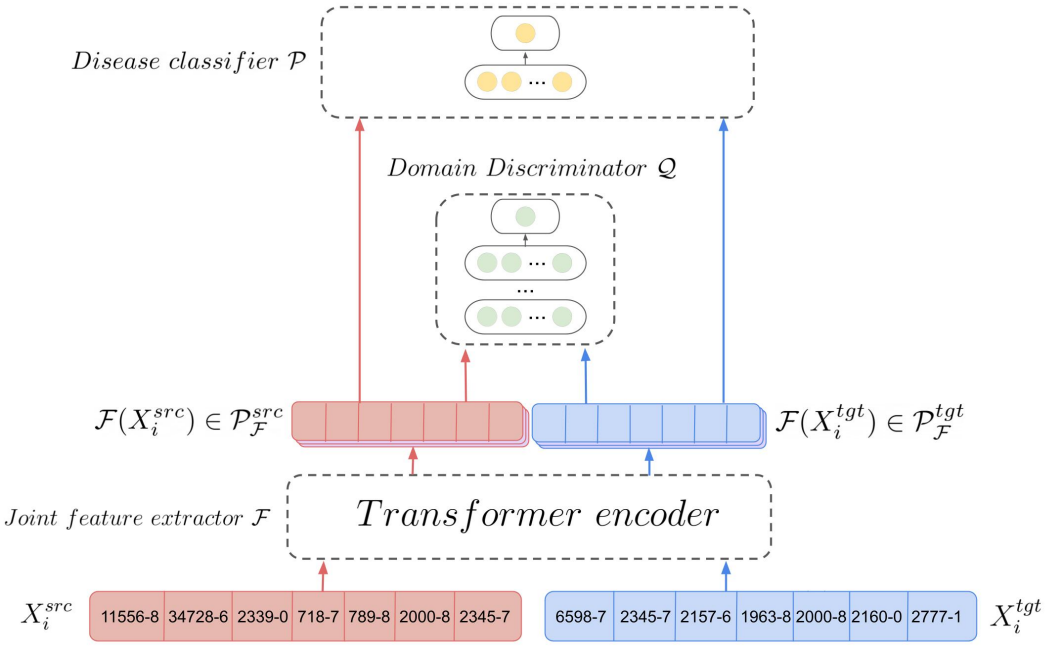


Figure 4.1: Illustration of the proposed AdaDiag framework, consisting of three modules: the joint feature extractor $\mathcal{F}$ that maps sequences from the source and target domain to a shared feature space, the classifier $\mathcal{P}$ that predicts next-visit HF onset and the discriminator $\mathcal{Q}$ for distinguishing source and target domain identity given the features from $\mathcal{F}$.

domain membership. For example, D^{src} and D^{tgt} represent the source and target domain. For a given patient i with a visit history X_i of n encounters $X_i = [x_1 \oplus x_2 \oplus ... \oplus x_n]$, each visit x_j consists of a sequence of events $x_j = [e_1 \oplus e_2 \oplus ... \oplus e_n] \in x_j$, with all events ordered sequentially by time. The next-visit disease label for event sequence X_i is denoted as $y_i \in \{0, 1\}$, which is available during training when $X_i \in D^{src}$. All sequences from D^{src} and D^{tgt} are assigned with domain labels $y_i' \in \{0, 1\}$.

4.3.2 The AdaDiag Framework

As illustrated in Figure. 4.1, AdaDiag is a feed-forward network with two forward branches following the design in [11]. The network consists of three parts: 1) a joint feature extractor $\mathcal{F}$ that maps an input sequence X_i to a shared feature space $\mathcal{F}(X_i)$; 2) a diagnostic classifier

$\mathcal{P}$ that predicts the label for X_i based on the feature representation $\mathcal{F}(X_i)$; and 3) a domain discriminator $\mathcal{Q}$ that also takes $\mathcal{F}(X_i)$ but predicts a label indicating domain identity (source/target) of X.

For improved performance, we pre-train a Transformer encoder as the feature extractor $\mathcal{F}$ to capture the contextualized information in the sequence. $\mathcal{F}$ feeds the sequence representation to $\mathcal{P}$, which is essentially a multi-layer perceptron (MLP) with a sigmoid output for binary diagnostic prediction. While trained with a different optimizer from $\mathcal{P}$'s, the domain discriminator $\mathcal{Q}$ is also an MLP, but ends with a linear layer to output a domain label [11]. During training, the diagnostic predictor $\mathcal{P}$ can only see disease labels from the source-domain dataset, whereas $\mathcal{Q}$ can observe (unlabeled) event sequences from both the source and target domain datasets.

The feature extractor $\mathcal{F}$ tries to learn a domain-invariant representation that aids in the prediction of the diagnostic predictor $\mathcal{P}$ as well as prevents the model from distinguishing features between different domains. The feature learned by $\mathcal{F}$ can be considered domain-invariant if a trained $\mathcal{Q}$ fails to distinguish between sequences from different domains. In this regard, $\mathcal{Q}$ is the adversarial component of the AdaDiag, as its target (distinguishing domains) goes against the overall goal of the AdaDiag framework on learning domain-invariant features. A well-trained $\mathcal{F}$ should be able to learn features that benefit the diagnostic prediction task, while keeping the domain identity as ambiguous as possible. Disease prediction can be performed at inference time by running unlabeled target domain sequences through the feature extractor $\mathcal{F}$ and the diagnostic classifier $\mathcal{P}$. No disease labels from the target domain are required throughout the model development process. At inference time, an input sequence X_i is passed through sufficiently trained $\mathcal{F}$ and $\mathcal{P}$ to predict for the disease label y_i, while keeping the domain discriminator $\mathcal{Q}$ untouched.

4.3.3 Pre-training Transformer Encoder

Following the recent success in Transformer-based pre-trained language models [30, 67, 91] and their adaptations modeling EHR data [64, 94], we construct a BERT-like architecture with six Transformer encoder layers, six attention heads, and an embedding dimension of 768 as the shared feature extractor $\mathcal{F}$ for a contextual representation that accounts for the entire disease progression process. The MLM task is adopted as the pre-training task of the Transformer-based encoder, which seeks to recover randomly masked clinical events in given sequences. All unlabeled event sequences from both source and target domains are used for this process. Unlike language models (e.g. BERT) that processes subword or byte-pair sequences, our encoder treats individual LOINC codes as the minimal units, since a lab event code cannot be further divided into semantically meaningful sub-units. Figure 4.2 illustrates the BERT-like input representation of our pre-trained Transformer-based model. As defined in the BERT paper [30], the input embeddings are the sum of the token embeddings, the segmentation embeddings and the position embeddings. As we do not differentiate segments within each input sequence, all segment embeddings are identical. Similar to [30], `[CLS]` is a special symbol added in front of every input example, whose representation will be used as the final sequence representation in fine-tuning tasks; `[SEP]` is a special separator token, indicating the end of the input sequence.

The MLM pre-training in our study follows a setting similar to the original BERT paper [30]. First, 15% of tokens in the sequences are randomly selected, and these chosen tokens will: 1) be replaced with the `[MASK]` token 80% of the time, 2) be replaced by another random tokens 10% of the time, and 3) stay unchanged the remaining 10% of the time. This mixed masking strategy was chosen to soften the discrepancy between pre-training and fine-tuning, as the `[MASK]` symbol will not appear during the fine-tuning stage [30]. For an input that contains one or more masked tokens, the model will generate the most likely substitution for each. We sampled 25% of all sequences for MLM evaluation, and trained the model for 100 epochs using the remaining sequences for predicting the masked token

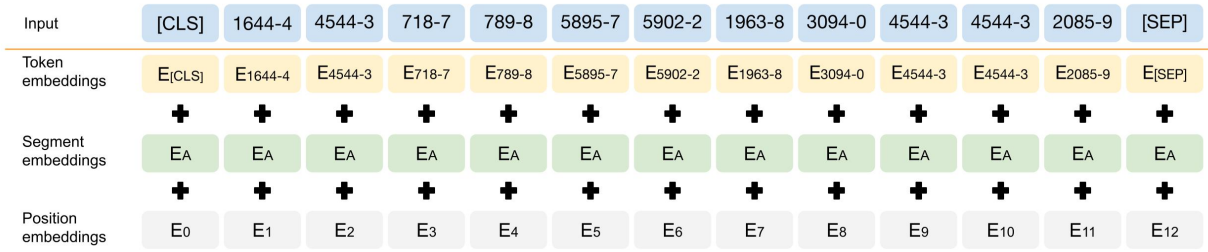


Figure 4.2: BERT-style input representation of the pre-trained Transformer-based model. As defined in the BERT paper [30], the input embeddings are the sum of the token embeddings, the segmentation embeddings and the position embeddings.

with cross-entropy loss. The best model was selected based on the lowest validation loss. In this process, the model captures the bidirectional context of each event in the sequence and accordingly learns a contextualized event representation.

4.3.4 Adversarial Training

AdaDiag aims at learning features from event sequences that are simultaneously beneficial to disease risk discrimination and cross-domain generalization. This goal can be achieved by adversarially optimizing on two discriminative tasks: disease prediction and domain discrimination. Like two-player game training from GANs, the adversarial training scheme of AdaDiag is formed as a minimax problem. Specifically, we need to find a set of parameters that *minimize* the disease prediction loss and at the same time *maximize* the domain discriminator loss.

As a result, adversarial training reduces the disparity between the marginal distributions of the source and target features, $P_{\mathcal{F}}^{src}$ and $P_{\mathcal{F}}^{tgt}$, over the shared feature space $\mathcal{F}(x)$:

$$P_{\mathcal{F}}^{src} \triangleq P(\mathcal{F}(x) \mid x \in D^{src})$$

$$P_{\mathcal{F}}^{tgt} \triangleq P(\mathcal{F}(x) \mid x \in D^{tgt})$$

To learn domain-invariant features, AdaDiag trains $\mathcal{F}$ to make distributions of $P_{\mathcal{F}}^{src}$ and

$P_{\mathcal{F}}^{tgt}$ to be as close as possible to improve cross-domain generalization. Intuitively, if a well-trained $\mathcal{Q}$ cannot determine the domain membership of the extracted features by $\mathcal{F}$ between the source and target domains, the features are domain-invariant.

In earlier works on adversarial domain adaptation (e.g., DANN [37], ADDA [115]), features are learned to confuse a classifier through different adversarial losses. Some [115] use the traditional GAN loss that can be deemed as minimizing the Jensen-Shannon (J-S) divergence between the source and target feature distributions, $P_{\mathcal{F}}^{src}$ and $P_{\mathcal{F}}^{tgt}$. When the learned features fail to mix distributions from both domains, gradient vanishing can occur if traditional probability-based loss measures such as cross-entropy or J-S divergence are used [104]. This situation might be better served by instead minimizing the Wasserstein distance [97], which appears to maintain gradient stability even when two distributions are far apart [2]. Specifically, we minimize the Wasserstein distance W between $P_{\mathcal{F}}^{src}$ and $P_{\mathcal{F}}^{tgt}$ over other alternatives [104] due to its stability on parameter selection as argued in [2, 11], which is defined as follows:

$$W\left(P_{\mathcal{F}}^{src}, P_{\mathcal{F}}^{tgt}\right) = \inf_{\gamma \sim \Pi\left(P_{\mathcal{F}}^{src}, P_{\mathcal{F}}^{tgt}\right)} \mathop{\mathbb{E}}_{(x^{src}, x^{tgt}) \sim \gamma} \left[\left\| x^{src} - x^{tgt} \right\|\right] \tag{4.1}$$

where $\Pi\left(P_{\mathcal{F}}^{src}, P_{\mathcal{F}}^{tgt}\right)$ denotes all possible joint distributions of source and target distributions, $P_{\mathcal{F}}^{src}$ and $P_{\mathcal{F}}^{tgt}$. As Eq. (4.1)'s minimum is computationally intractable, its Kantorovich-Rubinstein duality form is usually used in practice [119]:

$$W\left(P_{\mathcal{F}}^{src}, P_{\mathcal{F}}^{tgt}\right) = \sup_{\|g\|_L \leq 1} \mathop{\mathbb{E}}_{f(x) \sim P_{\mathcal{F}}^{src}} [g\left(f\left(x\right)\right)] - \mathop{\mathbb{E}}_{f(x') \sim P_{\mathcal{F}}^{tgt}} [g\left(f\left(x'\right)\right)] \tag{4.2}$$

The supremum is over functions g where g is 1-Lipschitz continuous. For simplicity, we denote this as $\|g\|_L \leq 1$. Note that the function g is 1-Lipschitz continuous if and only if $|g(x) - g(y)| \leq |x - y|$, for all x and y. In our case, $\mathcal{Q}$ serves as the function g in Eq. (4.2). Following [11], to make $\mathcal{Q}$ a 1-Lipschitz continuous function, all parameters in $\mathcal{Q}$ are clipped to a fixed range, $[-c, c]$, at the end of each $\mathcal{Q}$ optimization step. The minimax optimization process of adversarial training involves two learning objectives: the domain discriminator objective J_q, and the disease classification objective J_p. The model is trained for these two

objectives in an alternating fashion.

First, the discriminator $\mathcal{Q}$ is trained by maximizing the discriminator loss with $\mathcal{F}$ and $\mathcal{P}$ parameters fixed. The domain discriminator objective J_q is an approximation of the Wasserstein distance between the data distributions of the two domains. At the $\mathcal{Q}$ optimization step, it seeks to maximize J_q by updating its parameters in θ_q:

$$J_q\left(\theta_q\right) = W\left(P_{\mathcal{F}}^{src}, P_{\mathcal{F}}^{tgt}\right) = \max_{\theta_q}\left[\mathop{\mathbb{E}}_{\mathcal{F}(x)\sim P_{\mathcal{F}}^{src}}[\mathcal{Q}(\mathcal{F}(x))] - \mathop{\mathbb{E}}_{\mathcal{F}(x')\sim P_{\mathcal{F}}^{\text{tgt}}}\left[\mathcal{Q}\left(\mathcal{F}\left(x'\right)\right)\right]\right] \tag{4.3}$$

Next, the disease classifier is optimized by minimizing the disease classification loss with the discriminator $\mathcal{Q}$ fixed. The disease classification objective J_p, parameterized by θ_p, aims to minimize the binary cross-entropy loss $L_p(\hat{y}, y)$:

$$J_p\left(\theta_p\right) = \min_{\theta_p} \mathop{\mathbb{E}}_{(x,y)}\left[L_p(\mathcal{P}(\mathcal{F}(x)), y)\right] \tag{4.4}$$

$L_p(\hat{y}, y)$ is defined as the negative log-likelihood of correctly predicting the binary disease label:

$$L_p(\hat{y}, y) = -\frac{1}{\mathrm{n}}\sum_{i=1}^{\mathrm{n}} y_i \cdot \log\,\hat{y}_i + (1-y_i)\cdot\log\left(1-\hat{y}_i\right)$$

where $\hat{y}_i$ is the next-visit disease onset prediction for the i-th patient in the $\mathcal{P}$ output, y_i is the corresponding disease label, and output size is the number of predicted values/patients in the $\mathcal{P}$ output.

Lastly, serving for both discriminative tasks, the joint feature extractor $\mathcal{F}$ seeks to minimize the disease classification loss J_p as well as the Wasserstein distance J_q:

$$J_f = \min_{\theta_f}\left[J_p\left(\theta_f\right) + \lambda J_q\left(\theta_f\right)\right] \tag{4.5}$$

where λ is a hyperparameter that balances the losses of $\mathcal{P}$ and $\mathcal{Q}$.

4.4 Experiments

Here, we describe the datasets as well as their pre-processing procedures (Section 4.4.1), introduce implementation details of AdaDiag (Section 4.4.2), define baseline models and AdaDiag variants (Section 4.4.3), and present domain adaptation results on the next-visit HF onset prediction task (Section 4.4.4).

4.4.1 Experimental Setup

Abnormal lab sequences from two data sources, UCLA and MIMIC-IV, are used to predict next-visit HF onset. Compared to disease codes, which are usually unordered within visits, time-stamped lab events capture more fine-grained temporal dynamics within and between visits. The distribution of lab sequences across institutions could differ for a number of reasons, including demographics, mismatched ordering patterns, and policy changes – all of which contribute to domain shift that may limit cross-data model generalization. In this section, experiments are setup to address this challenging case. We briefly introduce the two EHR datasets and describe methods used to extract abnormal lab events with corresponding disease labels. In addition, differences in the two datasets are discussed, which indicate possible domain shifts as a result of disparities in the data generation and curation processes.

UCLA dataset We selected adult ($\geq$ 18 years old at initial admission time) patients who had at least one intensive care unit (ICU) stay at the UCLA Health System between 2013-03-01 and 2021-03-01, extracting all abnormal lab events from all in-patient visits within this time window along with their associated disease codes.

MIMIC-IV dataset MIMIC-IV data (version 1.0) [54] includes de-identified records from Beth Israel Deaconess Medical Center (BIDMC) for over 60,000 patients admitted to an ICU or the emergency department between 2008 and 2019.

Similar pre-processing steps are performed on lab events and diagnosis codes extracted from the two initial cohorts. To maximize clinical utility of our developed models, we focus on predicting *unseen* HF occurrences. Specifically, we excluded all encounters after the initial HF onset (if any), and use only encounters before (not including) the onset encounter as the sequence used for disease prediction. Patients with only one encounter remaining are removed as next-visit diagnosis prediction requires at least two visits. For a given patient i, abnormal lab events from all in-patient visits *before* the most recent visit (post-filtering) x_n are ordered by time and concatenated as the prediction input $X_i = [x_1 \oplus x_2 \oplus ... \oplus x_{n-1}]$, with its HF label defined as y_i, a binary indicator for having at least one HF diagnose (i.e., 3-digit ICD-9 code of *428* or ICD-10 code of *I50*) associated with $\{x_j\}$, $j \in \{1, 2, 3, ..., n-1\}$.

To facilitate a successful transfer, standardization of lab codes is needed so that the sequences from two local systems speak the same "language". We convert the local lab codes to a unified vocabulary, LOINC. UCLA Health has mappings from its local codes to LOINC for almost all available labs. In contrast, MIMIC-IV has no LOINC mappings for lab items in its microbiology events table, so we extracted raw lab sequences only from the *labevents* table using the dictionary file provided to map from local labs to LOINC codes. After removing all labs that are not mapped to LOINC codes, 96.7% [1] of the *labevents* occurrences in the extracted MIMIC-IV sequences remain. The pre-processed UCLA data has sequences for 18,736 patients with 1,218 unique LOINC codes, while the MIMIC-IV LOINC sequences have 27,782 patients with 272 unique codes. There are 139 shared LOINC codes in both vocabularies. The difference in vocabulary coverage is a result of multiple reasons from data generation (e.g., lab availability, ordering patterns, policy changes) to curation processes (e.g., incomplete mapping process), and is one cause of domain shift. There are other differences in the two datasets that may indicate potential shifts in data distribution. As shown in Table 4.1, MIMIC-IV patients have on average fewer visits, shorter

[1] In MIMIC-IV (v1.0), less than 17% (269/1630) of codes in the *d_labitems* mapping file are mapped to LOINC, covering 90.3% of all occurrences. We combined the local to LOINC lab mappings provided in MIMIC-III (v1.4) to map a larger percentage of labs to LOINC.

Table 4.1: Data summary of extracted cohorts from UCLA and MIMIC-IV dataset

	UCLA	MIMIC-IV
Number of patients	18,736	27,782
Number of visits	283,502	145,961
Avg. number of visits per patient	15.1	5.3
Number of unique lab codes	1,218	272
Avg. sequence length per patient	419.5	234.8
Female ratio	43.7%	51.4%
HF incidence rate	16.4%	27.7%

sequence lengths, a higher proportion of females, and a higher HF onset rate compared to the UCLA patients.

We conduct domain adaptation experiments in two directions: from UCLA to MIMIC-IV, and from MIMIC-IV to UCLA, both assuming no labels available in the target domain. The two datasets can each serve as D^{src} or D^{tgt}. 80% of the sequences from D^{src} are randomly selected and used for training, while the rest are used for model selection based on the validation area under the receiver operator characteristic (AUROC) curve metric. We maintain the same splits when UCLA and MIMIC-IV each serves as D^{src} for comparability. The best model is reported with AUROC and precision-recall area under the curve (pr-AUC) on all D^{tgt} sequences. We note here that this is deemed as zero-shot HF prediction, as no D^{tgt} labels are involved during the model development phase.

4.4.2 Implementation Details

We first pre-train AdaDiag with the MLM objective to learn the network parameters in the joint feature encoder $\mathcal{F}$ that can predict the masked lab event tokens, on all unlabeled sequences from D^{src} and D^{tgt}. Similar to the setup reported in Med-BERT [94], the Transformer architecture in the pre-trained model has six layers and six attention heads. Different from Med-BERT, we choose to use a hidden size of 768 (same for all Transformer-based

encoders in this chapter). The pre-trained model is then fine-tuned on the HF onset prediction and domain discrimination tasks with an adversarial objective. $\mathcal{F}$ and $\mathcal{P}$ are optimized jointly using AdamW [68] with a learning rate of 1E-5 and a weight decay of 0.01. A linear learning rate scheduler is used in all experiments. To balance the learning speed of the two alternatingly optimized adversarial objectives, Q is trained using a *separate* AdamW optimizer with a learning rate of 5E-4 and a weight decay of 0.01. To ensure that the discriminator $\mathcal{Q}$ satisfies the 1-Lipchitz constraint of the Wasserstein objective [2], the weights of $\mathcal{Q}$ are clipped within [-0.01,0.01] at the end of each training step of $\mathcal{Q}$, following the values used in [11]; the adversarial objective weight parameter λ from Eq. 4.5 is adjusted to 0.2 for all adversarial experiments. A fixed sequence length is chosen to be 1024 with sequences truncated/padded from the left, considering the fact that recent event history is more relevant to the upcoming disease onset. We train AdaDiag variants and the baselines and select the best model based on the validation AUROC metric from D^{src}. When transferring from UCLA data to MIMIC-IV data, the selected AdaDiag architecture has six layers in the shared feature extractor $\mathcal{F}$ (taken from the pre-trained Transformer model), zero layers in the disease classifier $\mathcal{P}$ ($\mathcal{P}$ is simply an output layer in this case) and two layers in the domain discriminator $\mathcal{Q}$. When the transfer is conducted reversely (from MIMIC-IV to UCLA), the best architecture has zero layers in the domain discriminator, with other settings remaining the same. The domain adaptation performance of AdaDiag is reported through AUROC and pr-AUC metrics on the entire D^{tgt} dataset. AdaDiag was implemented using Huggingface [126] based on PyTorch [84].

4.4.3 Baseline models

GRU/bi-GRU encoder with skip-gram embedding A pre-trained Transformer encoder is used in AdaDiag. While in non-Transformer-based clinical event sequence models [15, 17, 72, 79], the immediate sequence representation is provided using shallow word embedding methods (e.g., skip-gram/CBOW models in Word2Vec), before being fed into

encoders like long short-term memory (LSTM) networks/GRUs or convolutional neural networks (CNN) to learn a final representation. More advanced models apply bidirectional RNNs (i.e., bi-LSTM or bi-GRU) to better capture the temporal dependencies of clinical visits and improve model interpretability. We implemented the first two baseline models as a one-layer GRU encoder and a one-layer bi-GRU encoder. Their initial sequence encoding is provided by a skip-gram algorithm pre-trained on all sequences from D^{src} and D^{tgt}, with a window size of 20 and a embedding dimension of 768, which is the same as the dimension of the pre-trained Transformer encoder. The encoded features are directly passed to a linear output layer with Sigmoid activation to provide HF prediction, for which an Adam optimizer with a learning rate of 1E-3 is used.

Pre-trained Transformer Recent studies have shown the effectiveness of pre-trained Transformer-based encoders on learning better event sequence representation compared to RNN-based methods, achieving improved performance when fine-tuned on downstream tasks [64, 94]. An intuitive baseline is applying the non-adversarial version of AdaDiag with pre-trained Transformer encoder fine-tuned on D^{src} directly to D^{tgt}. For fair comparison, the pre-trained encoder prior to fine-tuning is the same as the one used in AdaDiag, which is pre-trained with the parameters of six layers, six attention heads, and a hidden dimension of 768.

Untrained Transformer model To understand the added value of pre-training to model generalizability, we compare the performance of the fine-tuned Transformer against the fine-tuned Transformer with no pre-training, where the latter is defined with the same architecture as the former but has randomly initialized layers. All the above baseline models discussed thus far are non-adversarial. We also report the results of the adversarial version of the fine-tuned, untrained Transformer model to demonstrate how adversarial training can be beneficial in another model setting, and to further illustrate the utility of pre-training in adversarially trained models.

Table 4.2: MIMIC-IV to UCLA domain adaptation performance comparison. Metrics are reported with 95% CI calculated through bootstrapping.

		MIMIC-IV(D^{src})		UCLA(D^{tgt})	
	Method	AUROC	pr-AUC	AUROC	pr-AUC
Baselines	GRU+Skip-gram	$0.7671 \pm .0143$	$0.5987 \pm .0256$	$0.4628 \pm .0114$	$0.1642 \pm .0083$
	Bi-GRU+Skip-gram	$0.7918 \pm .0139$	$0.6318 \pm .0258$	$0.6623 \pm .0096$	$0.2425 \pm .0112$
	Transformer	$0.7997 \pm .0133$	$0.6525 \pm .0243$	$0.6222 \pm .0114$	$0.2459 \pm .0121$
	Pre-trained Transformer	$0.8000 \pm .0134$	$0.6443 \pm .0246$	$0.6816 \pm .0104$	$0.2828 \pm .0133$
Adversarial	Transformer	$0.7977 \pm .0132$	$0.6468 \pm .0244$	$0.6456 \pm .0111$	$0.2659 \pm .0129$
	Pre-trained Transformer	$0.7985 \pm .0131$	$0.6374 \pm .0251$	$0.7089 \pm .0099$	$0.2944 \pm .0133$

4.4.4 Results

MIMIC-IV to UCLA transfer We first implement AdaDiag and baseline models for adapting from MIMIC-IV to UCLA data, given the fact that the former has a larger population and is publicly available. This is a more realistic scenario considering the data sharing restrictions of institution-specific datasets, as training AdaDiag requires data access from the source domain. In this setting, we emulate the situation where a local hospital system (UCLA) deploys models developed on public data from external institutions (BIDMC) to inform decisions. As shown in Table 4.2, all baseline models performed well on MIMIC-IV validation data. However, when tested on the UCLA sequences, they experienced drastic drops in both metrics. Over all other baselines, the Transformer baseline model performed best on MIMIC-IV and UCLA datasets, while the GRU model with skip-gram embedding performed the worst. Using metrics reported in Table 4.2, we created a graph visualizing relative performance loss for all models in Figure 4.3. In comparison to their non-adversarial counterparts, the two adversarial models had less performance loss from the cross-data transfer. AdaDiag achieved superior predictive performance (highlighted in grey) on UCLA data in comparison to all non-adversarial baselines and its adversarial variant (i.e., AdaDiag without pre-training). Specifically, compared with the best baseline model, non-adversarial

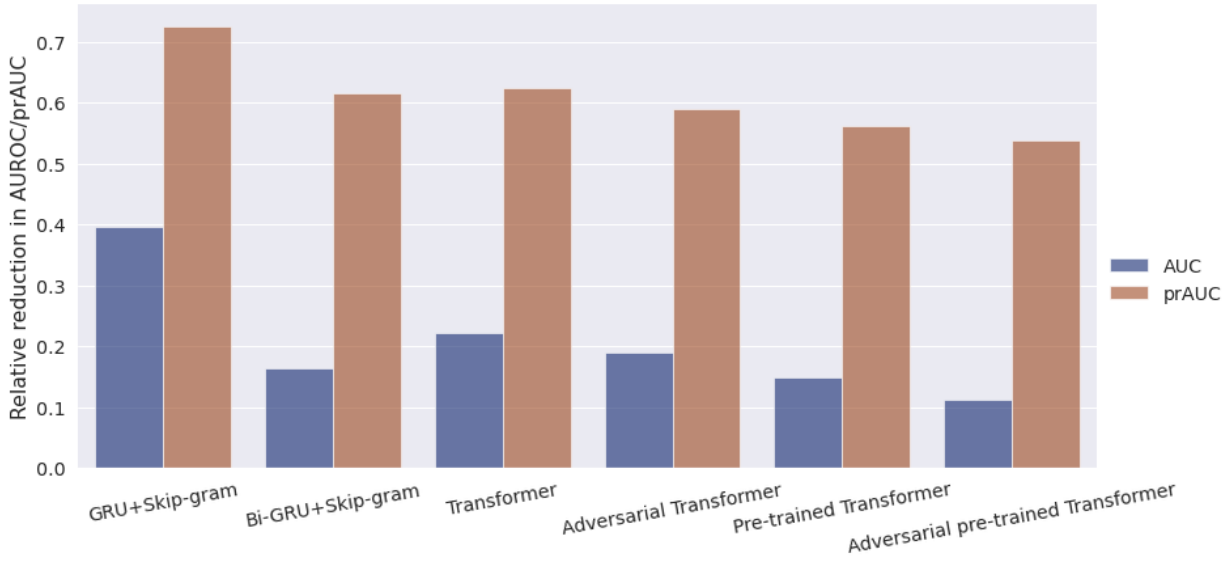


Figure 4.3: Illustration of relative performance losses in all baseline and adversarial models, when adapting from MIMIC-IV to UCLA data, calculated as (source metric-target metric)/source metric $\times$ 100%

pre-trained Transformer, AdaDiag's adversarial training boosted AUROC and pr-AUC on the UCLA data by 4.0% [2] and 4.1%. When no pre-training was performed, adversarial training boosted Transformer model's performance by 3.8% in AUROC and 8.1% in pr-AUC. These observations brought us to the conclusion that adversarial training benefits the Transformer-based models' generalization performance, while not greatly sabotaging their performance on the source domain. The untrained Transformer encoder baseline (fine-tuned on MIMIC-IV data) did not outperform pre-trained Skip-gram embedding with bi-GRU encoder when tested on MIMIC-IV data. Adding pre-training to it significantly improved its AUROC by 9.5% and pr-AUC by 15.0%, achieving superior performance relative to the bi-GRU with Skip-gram embedding baseline. In addition, pre-training was able to improve the AUROC and pr-AUC on target domain by 8.9% and 10.7% when added to the adversarial variant of the untrained Transformer baseline. These improvements show the importance of pre-training on increasing model's generalization performance on new datasets.

[2]This percentage was calculated for *relative improvement*, same as below.

Table 4.3: UCLA to MIMIC-IV domain adaptation performance comparison. Metrics are reported with 95% CI calculated through bootstrapping.

		UCLA(D^{src})		MIMIC-IV(D^{tgt})	
	Method	AUROC	pr-AUC	AUROC	pr-AUC
Baselines	GRU+Skip-gram	$0.7640 \pm .0206$	$0.3971 \pm .0370$	$0.5120 \pm .0076$	$0.3075 \pm .0088$
	Bi-GRU+Skip-gram	$0.8058 \pm .0188$	$0.5064 \pm .0424$	$0.6540 \pm .0071$	$0.4032 \pm .0106$
	Transformer	$0.8004 \pm .0197$	$0.5123 \pm .0402$	$0.6309 \pm .0073$	$0.3851 \pm .0105$
	Pre-trained Transformer	$0.8167 \pm .0182$	$0.5375 \pm .0200$	$0.6727 \pm .0072$	$0.4422 \pm .0114$
Adversarial	Transformer	$0.8018 \pm .0193$	$0.5126 \pm .0418$	$0.6583 \pm .0070$	$0.4110 \pm .0108$
	Pre-trained Transformer	$0.8113 \pm .0188$	$0.5336 \pm .0399$	$0.6959 \pm .0069$	$0.4610 \pm .0115$

UCLA to MIMIC-IV Transfer Given that labels from both datasets are readily available, we can verify if conclusions from MIMIC-IV to UCLA transfer still hold true with a different setup: transferring from a source data (UCLA) with smaller dataset but larger event vocabulary than the target data (MIMIC-IV). Models trained on UCLA data were tested on MIMIC-IV (Table 4.3), showing steep declines in AUROCs and pr-AUCs. AdaDiag had the least performance loss while the GRU+Skip-gram embedding model exhibited the most (Figure 4.4). When comparing the two adversarial models with their non-adversarial counterparts, we found that they experienced smaller relative performance loss. Thus, the same conclusion from our previous experiments (i.e., MIMIC-IV to UCLA transfer) persists: adversarial training helps model generalize when transferring across datasets. Table 4.3 also shows that with 3.4% gain in AUROC and 4.3% gain in pr-AUC, AdaDiag (highlighted in grey) outperformed the best non-adversarial baseline on MIMIC-IV data, while maintaining a comparable source domain validation performance on UCLA data. Both adversarial models outperformed their non-adversarial baselines, indicating that their zero-shot adaptation was enhanced by adversarial training without significant negative impact on their source domain performances. In this transfer setting, pre-training also played a major role, as was the case in UCLA to MIMIC-IV. In baseline models, the untrained Transformer performed worse

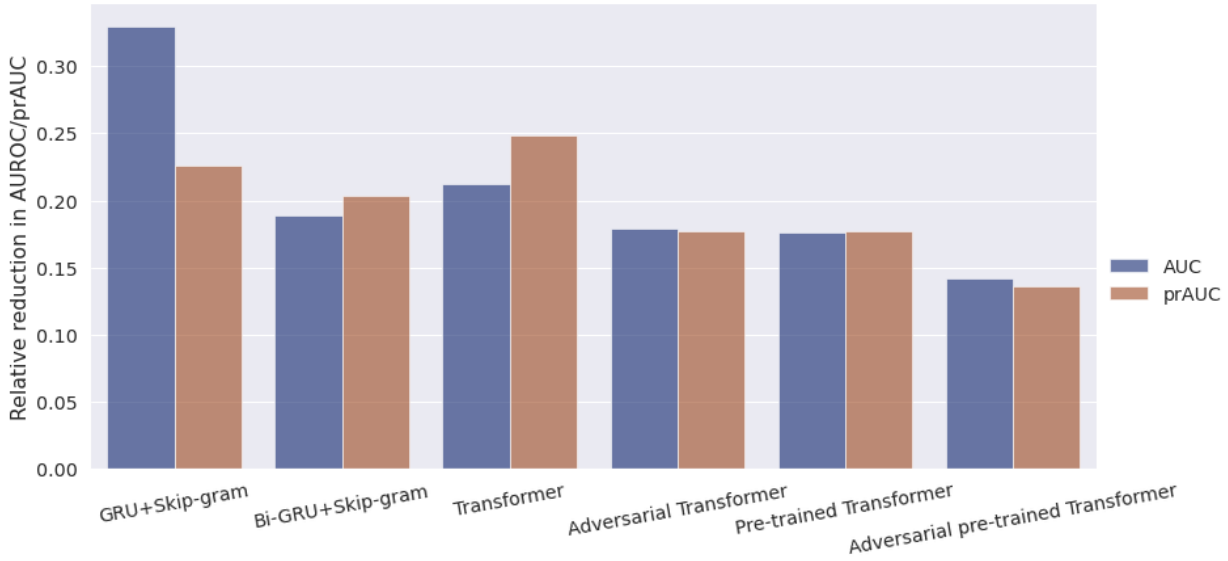


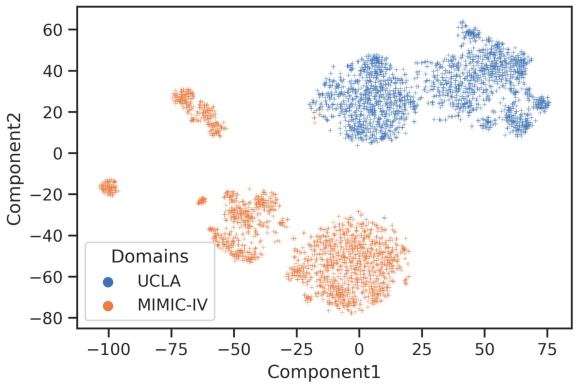
Figure 4.4: Illustration of relative performance losses in all baseline and adversarial models, when adapting from UCLA to MIMIC-IV data. Relative performance loss is calculated as (source metric-target metric)/source metric $\times$ 100%

than the bi-GRU model with skip-gram embedding; pre-training boosted its performance by 6.6% in AUROC and 14.8% in pr-AUC. In adversarial models, pre-training improved the model performance on MIMIC-IV by 5.7% in AUROC and 12.2% in pr-AUC.

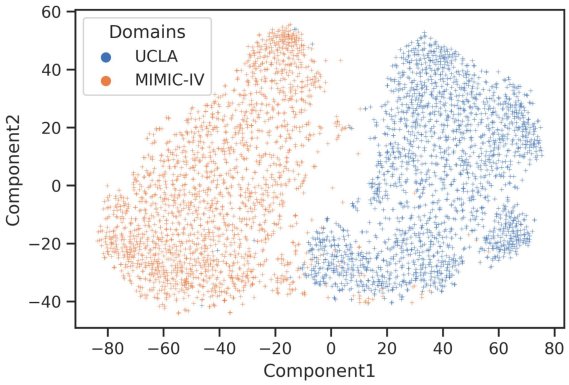
In both adaptation settings, the GRU encoder with skip-gram embedding was significantly less effective on learning features generalizable across datasets than other bi-GRU and Transformer-based methods, which is consistent with results reported in previous studies [94] and is possibly due to its left-to-right recurrent learning scheme and inability of learning bidirectional/contextual representations.

t-SNE visualization of feature distributions To compare and contrast the impact of adversarial training, we use t-SNE [116] for dimensionality reduction and visualize the feature distributions generated by Transformer encoders from different models/training stages in 2D. Figure 4.5 shows distributions of representations of pre-trained Transformer models before (Figure 4.5a) and after fine-tuning (Figures 4.5b and 4.5d); and adversarially trained

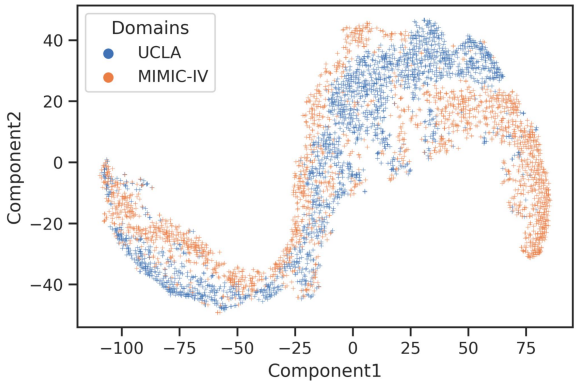
pre-trained Transformer models (i.e., AdaDiag) (Figures 4.5c and 4.5e) in both transfer directions. In Figure 4.5a, sequence representations from the two domains are far away from each other, showing through MLM pre-training alone is not sufficient to bridge the gap between UCLA and MIMIC-IV data. Train-on-source-only models are built on top of the pre-trained Transformer model and fine-tuned on the disease classification task. Their encoders' new mappings (Figures 4.5b and 4.5d) brought features from the two domains slightly closer, while remaining fairly separate from each other.



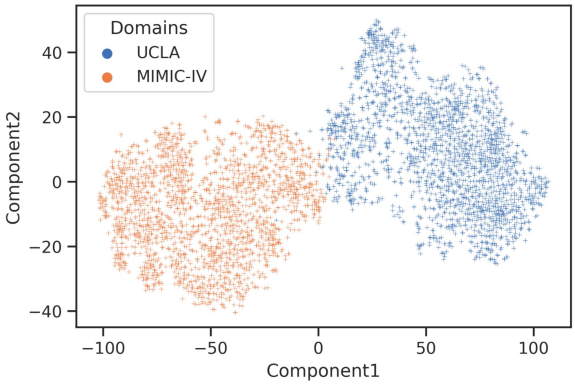
(a) Transformer model pre-trained on unlabeled MIMIC-IV and UCLA sequences.



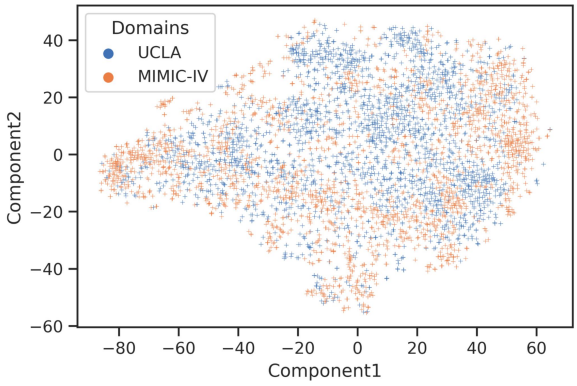
(b) Pre-trained Transformer model fine-tuned on MIMIC-IV data.



(c) Adversarial pre-trained Transformer model for MIMIC-IV to UCLA adaptation.



(d) Pre-trained Transformer model fine-tuned on UCLA data.

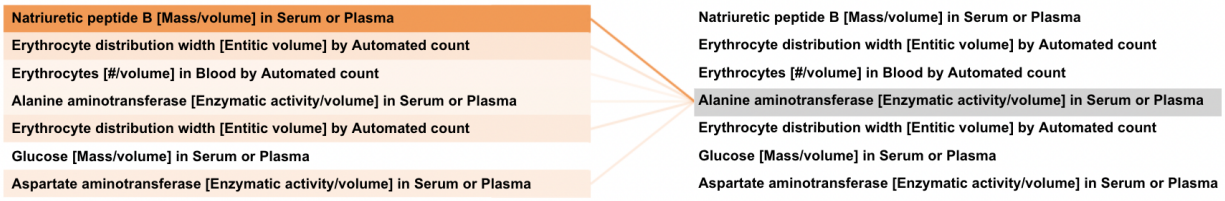


(e) Adversarial pre-trained Transformer model for UCLA to MIMIC-IV adaptation.

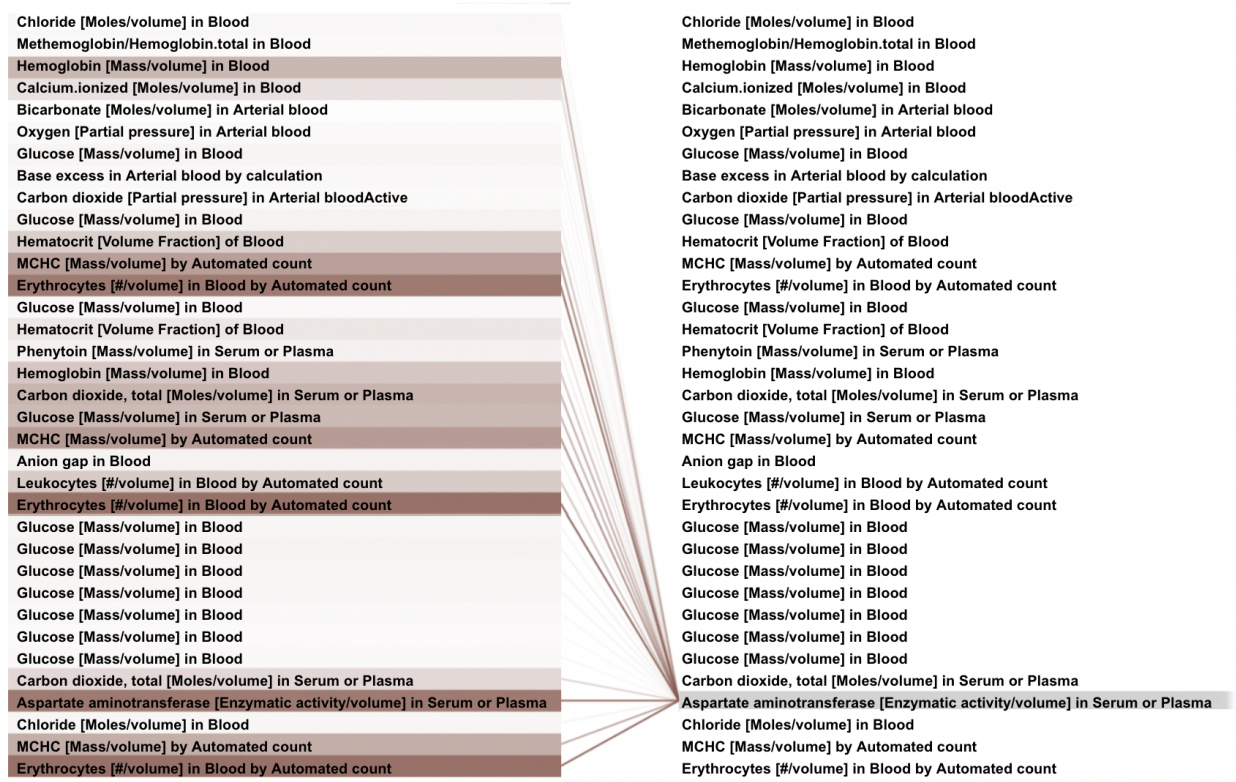
Figure 4.5: t-SNE visualizations of activations at the end of the Transformer feature encoders from different models/training stages. Neither pretraining nor fine-tuning were able to bridge the domain gap, whereas adversarial training mixed the distributions between the two datasets effectively.

Visualization of self-attention in Transformer encoders The self-attention mechanism of the Transformer layers is able to capture complex relationships between lab events, adding interpretability to our model. Figure 4.6 shows an analysis of self-attention in AdaDiag (MIMIC-IV to UCLA)'s Transformer encoder. Based on the approach presented by [118], we analyze attention-based patterns for two patients, referred to as A (Figure 4.6a) and B (Figure 4.6b), from the UCLA cohort. For each patient, the abnormal lab events are presented as two identical columns, with events ordered chronologically from top to bottom. By passing the sequences through the six Transformer encoder layers, each with six attention heads, a 'headview' for each head from each layer is generated depicting attention weights of all events in the sequence, given an event of interest. An abnormal lab (in gray, on the right) is linked with all lab events in the same sequence, with the shades of the color block/edges reflecting the attention weights/degrees of association. For patient A, whose history consists of a short sequence of events from the same encounter, a strong association is found between the B-type natriuretic peptide (BNP) test (elevated value observed in heart failure) and the alanine aminotransferase (ALT) test (used to check for liver damage). The ALT test is also weakly associated with aspartate aminotransferase (AST) (another test for tissue damage in organs like liver/heart) and other red blood cell related tests such as erythrocyte distribution width (EDW) and erythrocyte count. In patient B's history, which spanned across multiple visits, the AST test is closely related to several blood tests that are relevant to red blood cells and their ability to transmit oxygen in blood: hemoglobin/hematocrit concentration, mean corpuscular hemoglobin concentration (MCHC), and erythrocyte count. This example especially shows AdaDiag's ability of extracting long term dependencies in a multi-visit sequence. The associations between ALT and BNP (patient A) could help uncover new patterns when evaluating liver damage as early signs of heart failure due to the complex cardiohepatic interactions [46]. The relationships between ALT or AST and erythrocyte related tests (patient A/B) might indicate the underlying linkage between conditions such as anemia and organ (e.g., liver/heart) tissue damage. Identifying such self-attention pat-

terns has enabled more profound understanding of AdaDiag's functionality and extended its potential into discovering new knowledge that has clinical relevance.



(a) Abnormal lab event sequence for **patient A**. A strong association between abnormalities in alanine aminotransferase (ALT) and niatriuretic peptide B (BNP) is found.



(b) Abnormal lab event sequence for **patient B**. Abnormal values in hemoglobin or erythrocyte concentration in blood are strongly associated with elevated values in aspartate aminotransferase (AST).

Figure 4.6: Analysis of self-attention in AdaDiag's Transformer encoder layer for MIMIC-IV to UCLA adaptation. Colors of the edges corresponds to individual attention heads from the first Transformer layer (e.g, orange: the second head; brown: the sixth head), and shades of the edges/highlighted region indicate attention weights.

4.5 Conclusion

Improved generalizability of clinical predictive models is essential to achieving widespread clinical application under the constraint of low training resources. In this work, we address the dataset shift issue that has prevented successful cross-dataset applications by enforcing domain-invariant representations through unsupervised adversarial training. We introduce a novel Transformer-based adversarial domain adaptation framework that transfers an event sequence diagnostic prediction model from a *source* domain, where it was developed and trained, to another *target* domain where it could be applied without requiring the disease labels. Its utility was demonstrated on next-visit HF onset prediction in two transfer settings, using two large clinical event datasets: from MIMIC-IV to UCLA, and UCLA to MIMIC-IV. While the RNN or Transformer-based non-adversarial baselines suffered greatly when tested on unseen sequences from the target data, adversarial training was found to be effective in improving the Transformer-based model's performance on unseen targets, maintaining similar accuracy on the source data. We also highlighted the importance of pre-training in the ablation studies with an untrained Transformer model, showing that pre-training in conjunction with adversarial training led to an increased generalization power for AdaDiag. A t-SNE plot illustrating the effect of adversarial training on feature distributions is presented for mixing two distributions with originally large differences from two domains. With the help of the Transformer encoder, the interpretability of the self-attention patterns learned within AdaDiag was visualized using the Bertviz [118] tool, which showed clinically meaningful associations between abnormal lab events from a given patient's history. This also allows AdaDiag to explore formerly unknown patterns for medical knowledge discovery.

AdaDiag's application goes beyond HF prediction; other clinical prediction tasks, such as length of stay (LOS) and mortality prediction, also require cross-population generalizability, and are part of our research plans for the future. Future directions we would like to explore include: 1) extending AdaDiag to other types of clinical events such as medica-

tion and diagnostic codes; 2) applying AdaDiag to correct temporal dataset shift; and 3) training AdaDiag without access to source domain data.

CHAPTER 5

Conclusion

This chapter summarizes the results and contributions from this dissertation. Based on the findings presented, we suggest research directions to further improve model generalizability under data perturbation and distribution shifts using electronic health record (EHR) data.

5.1 Summary of Research

Although there remain many challenges to using large-scale observational datasets from the EHR, careful modeling and application of such data can be used for clinical prediction tasks – particularly over time, as data changes. In this dissertation, we provide the following research developments around two aspects of predictive modeling of clinical event sequences under dataset shift:

1. **A new permutation-sampling-based method to mitigate the impact of inconsistent concurrent event ordering in predictive event sequence modeling.** A permutation-invariant prediction mechanism is made possible by introducing a permutation sampling technique on event sequences modeled as sequence-of-sets. We provide a demonstration of the utility of the weighted loss function with additional regularization term enforcing permutation-invariant representation of the input sequence, which further improves the model predictive performance when using permuted sequences.
2. **An adaptive deep learning framework to improve the unsupervised transfer performance on disease prediction tasks using clinical event sequences, mov-**

ing from a labeled source domain to an unlabeled target domain. While the validation results of the existing predictive models for event sequence (e.g., the state-of-the-art pre-trained Transformer model) have shown poor generalizability across data sources, their performance in target domain was significantly enhanced by adversarial training without compromising source validation accuracy.

Chapters 2 and 3 presented key issues related to the modeling of clinical sequential data with algorithms that overcome these challenges. Demonstrations of this research were conducted using public datasets (e.g., MIMIC) and specific institutional datasets (e.g., from UCLA Health System) to illustrate the problems and advances over conventional deep learning methods.

5.2 Future Directions

We identify limitations of works in this dissertation and subsequently suggest several directions for extending the work presented in this dissertation to fit in a broader context of improving model generalizability under dataset shift.

Representing temporal intervals. Efforts in this dissertation directly utilized the indexed sequence order to reflect the temporal order of the clinical events. Additional information can be added to more accurately capture the complex temporal dynamics within a patient's visit history. For example, event intervals could be explicitly modeled using time tokens as special vocabularies motivated by the varying indications of time intervals in different disease trajectories (e.g., acute vs. chronic, short vs. long time spans). With this setup, both between- and within-visit time intervals can be represented in a manner akin to existing clinical events. A new line of research can be conducted to investigate how modeling time intervals would impact predictions for diseases at different levels of acuteness and immediacy.

Simulation studies with single-source dataset shift. Other than concurrent event ordering shift we demonstrated for DPSS in Chapter 3, additional studies could be conducted simulating cases when a range of single-source dataset shift (e.g., age distribution, event code frequencies) is present, investigating their individual impact on the model performance. In this way, we will be able to subsequently evaluate and dissect AdaDiag's applicability to other complex data shift scenarios with a different composite sources of dataset shifts. Furthermore, even though our experiments demonstrated that our framework reduced the discrepancies in model performance between source and target data, the differences are still considerable. This calls for further simulation studies to identify potential improvements needed in our proposed methods, and further close the performance gap.

Event code description embedding. Transferring event sequence models from one system to another can be challenging due to their mismatched vocabularies. For example, in some instances, lab events from one institution may not be available in another, or equivalent lab tests may be coded differently due to factors such as test availability, mapping procedures, repetitive code concepts (e.g., not deprecated in time within the coding system), and code versions not being promptly updated. Contextual embeddings from encoders like Transformers may be able to infer the semantic meaning of unseen words/linking events with relevant concepts through pre-training tasks and sequence context. We can also address this issue more directly by injecting domain knowledge into the learning process. By creating embeddings of the textual event code descriptions, key event concepts can be encoded in a way that preserves semantic relationships. One intuitive way of creating such embeddings is through language models pre-trained on large-scale, domain specific text corpus, such as BioBERT [61] or BioClinicalBERT [1]. Due to the fact that descriptions from coding ontology systems (e.g., LOINC) are inherently different from natural language in pre-training data, such as Wikipedia or biomedical journal articles, a more specifically pre-trained model may be required in order to accurately represent the language style in ontology concept

descriptions.

Dataset shift detection. Taking a step back from searching for solutions to dataset shift, investigating methods for detecting dataset shifts is also a critical aspect of proactively safeguarding model generalizability and deployment-time stability. Depending on the application, it may provide insights during the data curation phase in preparation for building a more reliable model. Recent dataset shift detection methods proposed for medical data include the anomaly-detection-based MedShift framework [44], and Park et al. [83], which translates a distance-metric-based out-of-distribution score to an interpretable confidence score that helps guide user decisions in healthcare ML systems. These shift detection methods are designed to identify out-of-distribution samples given any two sample datasets that are retrospectively collected from different sources/simulated with distinct properties in a controlled environment. One real-world use case of these techniques that can be further explored is to detect temporal shifts within the same data source. When new conditions appear (e.g., COVID-19) and treatment protocols evolve to accommodate the updated medical understanding of diseases, the data generated may shift over time. This scenario requires an additional system to give dynamic assessment and alerts about whether the recommendations from a pre-existing model can be applied to the newly generated samples [52].

Modeling with temporal dataset shift. Dataset shifts other than highlighted in this dissertation may exist in real-life clinical scenarios. As just discussed, dataset shift can occur in the form of temporal shift due to data collected changing over time. Temporal shifts can be present as trends and as abrupt or seasonal changes (e.g., evolving clinical policies [101], upgrades in clinical measuring devices, seasonal epidemics, etc.) [98]. If confirmed exist (i.e., through dataset shift detection), methods can be developed to correct temporal shifts dynamically, with the idea that it is more optimal to proactively update a model when new data is available rather than switching versions only when evidences of model performance deterioration emerge [52]. The modeling techniques discussed in this dissertation could

be applied to the construction of such systems, given their ability to adapt to new data distributions.

REFERENCES

[1] Emily Alsentzer, John R. Murphy, Willie Boag, Wei-Hung Weng, Di Jin, Tristan Naumann, and Matthew B. A. McDermott. Publicly available clinical bert embeddings. *ArXiv*, abs/1904.03323, 2019.

[2] Martín Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. *ArXiv*, abs/1701.07875, 2017.

[3] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473, 2015.

[4] Guangcheng Bao, Ning Zhuang, Li Tong, Bin Yan, Jun Shu, Linyuan Wang, Ying Zeng, and Zhichong Shen. Two-level domain adaptation neural network for eeg-based emotion recognition. *Frontiers in Human Neuroscience*, 14, 2020.

[5] Sebastiano Barbieri, James Kemp, Oscar Perez-Concha, et al. Benchmarking Deep Learning Architectures for Predicting Readmission to the ICU and Describing Patients-at-Risk. *Scientific Reports*, 10(1):1–10, 2020.

[6] Emelia J. Benjamin, Paul M. Muntner, Álvaro Alonso, Márcio Sommer Bittencourt, Clifton Callaway, April P. Carson, Alanna M. Chamberlain, Alexander R. Chang, Susan Cheng, Sandeep R. Das, Francesca N. Delling, Luc Djoussé, Mitchell S. V. Elkind, Jane F. Ferguson, Myriam Fornage, Lori C. Jordan, Sadiya Sana Khan, Brett M. Kissela, Kristen L. Knutson, Tak W Kwan, Daniel T. Lackland, Tené T. Lewis, Judith H. Lichtman, Chris T. Longenecker, Matthew Shane Loop, Pamela L. Lutsey, Seth Shay Martin, Kunihiro Matsushita, Andrew E. Moran, Michael E. Mussolino, Martin O'Flaherty, Ambarish Pandey, Amanda M. Perak, Wayne D Rosamond, Gregory A. Roth, Uchechukwu Sampson, Gary M. Satou, Emily B. Schroeder, Svati H. Shah, Nicole L. Spartano, Andrew C. Stokes, David L. Tirschwell, Connie W. Tsao, Mintu P. Turakhia, Lisa B. VanWagner, John T. Wilkins, Sally S. Wong, and Salim S. Virani. Heart disease and stroke statistics—2019 update: A report from the american heart association. *Circulation*, 139:e56–e528, 2019.

[7] Jonathon Byrd and Zachary Chase Lipton. What is the effect of importance weighting in deep learning? In *ICML*, 2019.

[8] Xiangrui Cai, Jinyang Gao, Kee Yuan Ngiam, Beng Chin Ooi, Ying Zhang, and Xiaojie Yuan. Medical concept embedding with time-aware attention. In *IJCAI*, 2018.

[9] Zhengping Che and Y. Liu. Deep learning solutions to computational phenotyping in health care. *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 1100–1109, 2017.

[10] Cheng Chen, Qi Dou, Hao Chen, Jing Qin, and Pheng-Ann Heng. Synergistic image and feature adaptation: Towards cross-modality domain adaptation for medical image segmentation. In *AAAI*, 2019.

[11] Xilun Chen, Yu Sun, Ben Athiwaratkun, Claire Cardie, and Kilian Q. Weinberger. Adversarial deep averaging networks for cross-lingual sentiment classification. *Transactions of the Association for Computational Linguistics*, 6:557–570, 2018.

[12] V. Cheplygina, Isabel Pino Peña, Jesper Johannes Holst Pedersen, David A. Lynch, Lauge Sørensen, and Marleen de Bruijne. Transfer learning for multicenter classification of chronic obstructive pulmonary disease. *IEEE Journal of Biomedical and Health Informatics*, 22:1486–1496, 2018.

[13] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *EMNLP*, 2014.

[14] Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *EMNLP*, 2014.

[15] E. Choi, Andy Schuetz, Walter F. Stewart, and Jimeng Sun. Using recurrent neural network models for early detection of heart failure onset. *Journal of the American Medical Informatics Association : JAMIA*, 24:361 – 370, 2017.

[16] Edward Choi, Mohammad Taha Bahadori, Andy Schuetz, Walter F Stewart, and Jimeng Sun. Doctor ai: Predicting clinical events via recurrent neural networks. In *MLHC*, pages 301–318, 2016.

[17] Edward Choi, Mohammad Taha Bahadori, Andy Schuetz, Walter F. Stewart, and Jimeng Sun. Doctor ai: Predicting clinical events via recurrent neural networks. *JMLR workshop and conference proceedings*, 56:301–318, 2016.

[18] Edward Choi, Mohammad Taha Bahadori, Elizabeth Searles, Catherine Judith Coffey, Michael Thompson, James Bost, Javier Tejedor-Sojo, and Jimeng Sun. Multi-layer representation learning for medical concepts. In *KDD*, 2016.

[19] Edward Choi, Mohammad Taha Bahadori, Le Song, Walter F. Stewart, and Jimeng Sun. Gram: Graph-based attention model for healthcare representation learning. In *KDD '17*, 2017.

[20] Edward Choi, Andy Schuetz, Walter F Stewart, and Jimeng Sun. Using Recurrent Neural Network Models for Early Detection of Heart Failure Onset. *Journal of the American Medical Informatics Association*, 24(2):361–370, 2016.

[21] Edward Choi, Cao Xiao, Walter F. Stewart, and Jimeng Sun. Mime: Multilevel medical embedding of electronic health records for predictive healthcare. In *NIPS*, 2018.

[22] Edward Choi, Zhen Xu, Y. Li, Michael W. Dusenberry, Gerardo Flores, Y. Xue, and Andrew M. Dai. Learning the graphical structure of electronic health records with graph convolutional transformer. In *AAAI*, 2020.

[23] Edward Choi, Zhen Xu, Yujia Li, et al. Learning the graphical structure of electronic health records with graph convolutional transformer. *AAAI*, 2020.

[24] Youngduck Choi, Chill Yi-I Chiu, and David A Sontag. Learning low-dimensional representations of medical concepts. *AMIA Summits on Translational Science Proceedings*, 2016:41 – 50, 2016.

[25] Junyoung Chung, Kyle Kastner, Laurent Dinh, Kratarth Goel, Aaron C. Courville, and Yoshua Bengio. A recurrent latent variable model for sequential data. In *NIPS*, 2015.

[26] Alicia Curth, Patrick J. Thoral, Wilco van den Wildenberg, Peter Bijlstra, Daan de Bruin, Paul W. G. Elbers, and Mattia Fornasa. Transferring clinical prediction models across hospitals and electronic health record systems. In *PKDD/ECML Workshops*, 2019.

[27] Daumé, Hal Daumé Iii, Abhishek Kumar, and Avishek Saha. Frustratingly easy semi-supervised domain adaptation. 2010.

[28] Hal Daumé. Frustratingly easy domain adaptation. *ArXiv*, abs/0907.1815, 2007.

[29] Thomas Desautels, J. Calvert, J. Hoffman, Q. Mao, Melissa Jay, Grant S. Fletcher, C. Barton, U. Chettipally, Yaniv Kerem, and R. Das. Using transfer learning for improved mortality prediction in a data-scarce hospital setting. *Biomedical Informatics Insights*, 9, 2017.

[30] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2019.

[31] Bhuwan Dhingra, H Liu, et al. Gated-attention readers for text comprehension. In *ACL*, 2016.

[32] Q. Dou, C. Ouyang, C. Chen, Hao Chen, and P. Heng. Unsupervised cross-modality domain adaptation of convnets for biomedical image segmentations with adversarial loss. In *IJCAI*, 2018.

[33] Sébastien Dubois, Nathanael Romano, Kenneth Jung, N. Shah, and David C. Kale. The effectiveness of transfer learning in electronic health records data. In *ICLR*, 2017.

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

[35] Wael Farhan, Zhimu Wang, Yingxiang Huang, et al. A Predictive Model for Medical Events Based on Contextual Embedding of Temporal Sequences. *JMIR Medical Informatics*, 4:e39, 2016.

[36] Wael Farhan, Zhimu Wang, Yingxiang Huang, Shuang Wang, F. Wang, and Xiaoqian Jiang. A predictive model for medical events based on contextual embedding of temporal sequences. *JMIR Medical Informatics*, 4, 2016.

[37] Yaroslav Ganin, E. Ustinova, Hana Ajakan, Pascal Germain, H. Larochelle, François Laviolette, Mario Marchand, and Victor S. Lempitsky. Domain-adversarial training of neural networks. *J. Mach. Learn. Res.*, 17:59:1–59:35, 2016.

[38] Kim Huat Goh, Le Wang, Adrian Yong Kwang Yeow, Hermione Mei Niang Poh, Ke Li, Joannas Jie Lin Yeow, and Gamaliel Yu Heng Tan. Artificial intelligence in sepsis early prediction and diagnosis using unstructured data in healthcare. *Nature Communications*, 12, 2021.

[39] Ary L. Goldberger, Luis A. Nunes Amaral, L Glass, Jeffrey M. Hausdorff, Plamen Ch. Ivanov, Roger G. Mark, Joseph E. Mietus, George B. Moody, Chung-Kang Peng, and Harry Eugene Stanley. Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals. *Circulation*, 101 23:E215–20, 2000.

[40] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, 2014.

[41] Alex Graves. Adaptive computation time for recurrent neural networks. *ArXiv*, abs/1603.08983, 2016.

[42] Alex Graves, Greg Wayne, and Ivo Danihelka. Neural turing machines. *ArXiv*, abs/1410.5401, 2014.

[43] Yanyang Gu, ZongYuan Ge, C. Bonnington, and Jun Zhou. Progressive transfer learning and adversarial domain adaptation for cross-domain skin disease classification. *IEEE Journal of Biomedical and Health Informatics*, 24:1379–1393, 2020.

[44] Xiaoyuan Guo, Judy Wawira Gichoya, Hari Trivedi, Saptarshi Purkayastha, and Imon Banerjee. Medshift: identifying shift data for medical dataset curation. 2021.

[45] Priyanka Gupta, Pankaj Malhotra, Jyoti Narwariya, L. Vig, and Gautam M. Shroff. Transfer learning for clinical time series analysis using deep neural networks. *Journal of Healthcare Informatics Research*, 4:112–137, 2020.

[46] Hamza El Hadi, Angelo Di Vincenzo, Roberto Vettor, and Marco Rossato. Relationship between heart disease and liver disease: A two-way street. *Cells*, 9, 2020.

[47] William L. Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *NIPS*, 2017.

[48] Junheng Hao, Muhao Chen, Wenchao Yu, et al. Universal representation learning of knowledge bases by jointly embedding instances and ontological concepts. In *KDD*, 2019.

[49] Hrayr Harutyunyan, Hrant Khachatrian, David C. Kale, and Aram Galstyan. Multitask learning and benchmarking with clinical time series data. *Scientific Data*, 6, 2019.

[50] Zhiheng Huang, Wei Xu, and Kailiang Yu. Bidirectional lstm-crf models for sequence tagging. *ArXiv*, abs/1508.01991, 2015.

[51] Kosuke Inoue, William Hsu, Onyebuchi A. Arah, Ashley E Prosper, Denise R. Aberle, and Alex A. T. Bui. Generalizability and transportability of the national lung screening trial data: Extending trial results to different populations. *Cancer epidemiology, biomarkers & prevention : a publication of the American Association for Cancer Research, cosponsored by the American Society of Preventive Oncology*, 2021.

[52] David A. Jenkins, Glen Philip Martin, Matthew Sperrin, Richard D. Riley, Thomas P. A. Debray, Gary S. Collins, and Niels Peek. Continual updating and monitoring of clinical prediction models: time for dynamic prediction systems? *Diagnostic and Prognostic Research*, 5, 2021.

[53] Jing Jiang and ChengXiang Zhai. Instance weighting for domain adaptation in nlp. In *ACL*, 2007.

[54] Alistair Johnson, Lucas Bulgarelli, Tom Pollard, Steven Horng, Leo Anthony Celi, and Roger Mark. Mimic-iv (version 1.0). In *PhysioNet*, 2021.

[55] Alistair E. W. Johnson, T. Pollard, and Tristan Naumann. Generalizability of predictive models for intensive care unit patients. *Machine Learning for Health (ML4H) Workshop at NeurIPS 2018*, 2018.

[56] Alistair E. W. Johnson, Tom J. Pollard, et al. Mimic-iii, a freely accessible critical care database. *Scientific Data*, 3, 2016.

[57] Christopher J. Kelly, A. Karthikesalingam, Mustafa Suleyman, Greg C. Corrado, and Dominic King. Key challenges for delivering clinical impact with artificial intelligence. *BMC Medicine*, 17, 2019.

[58] Farzaneh Khoshnevisan and Min Chi. An adversarial domain separation framework for septic shock early prediction across ehr systems. *2020 IEEE International Conference on Big Data (Big Data)*, pages 64–73, 2020.

[59] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2014.

[60] Wouter M. Kouw. An introduction to domain adaptation and transfer learning. *ArXiv*, abs/1812.11806, 2018.

[61] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36:1234 – 1240, 2020.

[62] Juho Lee, Yoonho Lee, Jungtaek Kim, Adam R. Kosiorek, Seungjin Choi, and Yee Whye Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. In *ICML*, 2018.

[63] Mitchell M. Levy, Mitchell P. Fink, John C. Marshall, Edward Abraham, Derek C. Angus, Deborah J. Cook, Jonathan Cohen, Steven M. Opal, Jean Louis Vincent, G. Ramsay, and for the International Sepsis Definitions Conference. 2001 sccm/esicm/accp/ats/sis international sepsis definitions conference. *Intensive Care Medicine*, 29:530–538, 2003.

[64] Yikuan Li, S. Rao, J. R. A. Solares, A. Hassaine, R. Ramakrishnan, D. Canoy, Y. Zhu, K. Rahimi, and G. Salimi-Khorshidi. Behrt: Transformer for electronic health records. *Scientific Reports*, 10, 2020.

[65] Ruogu Lin, Xiangrui Zeng, K. Kitani, and Min Xu. Adversarial domain adaptation for cross data source macromolecule in situ structural classification in cellular electron cryo-tomograms. *Bioinformatics*, 35:i260 – i268, 2019.

[66] Wenshuo Liu, Karandeep Singh, A. Ryan, Devraj Sukul, E. Mahmoudi, A. Waljee, Cooper Stansbury, Ji Zhu, and B. Nallamothu. Predicting 30-day hospital readmissions using artificial neural networks with medical code embedding. *bioRxiv*, 2019.

[67] 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*, abs/1907.11692, 2019.

[68] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019.

[69] Yun Luo, Si-Yang Zhang, Wei-Long Zheng, and Bao-Liang Lu. Wgan domain adaptation for eeg-based emotion recognition. In *ICONIP*, 2018.

[70] Thang Luong, Hieu Pham, and Christopher D. Manning. Effective approaches to attention-based neural machine translation. In *EMNLP*, 2015.

[71] Fenglong Ma, Quanzeng You, Houping Xiao, Radha Chitta, Jing Zhou, and Jing Gao. Kame: Knowledge-based attention model for diagnosis prediction in healthcare. *CIKM*, 2018.

[72] Tengfei Ma, Cao Xiao, and Fei Wang. Health-atm: A deep architecture for multifaceted patient health record representation and risk prediction. In *SDM*, 2018.

[73] Changping Meng, Jiasen Yang, Bruno Ribeiro, and Jennifer Neville. Hats: A hierarchical sequence-attention framework for inductive set-of-sets embeddings. In *KDD*, pages 783–792, 2019.

[74] Tomas Mikolov, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *ICLR*, 2013.

[75] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, pages 3111–3119, 2013.

[76] Timothy Miller, Steven Bethard, Hadi Amiri, and Guergana K. Savova. Unsupervised domain adaptation for clinical negation detection. In *BioNLP*, 2017.

[77] John Moore and Jennifer Neville. Deep collective inference. In *AAAI*, 2017.

[78] Ryan L. Murphy, Balasubramaniam Srinivasan, Vinayak A. Rao, and Bruno Ribeiro. Janossy pooling: Learning deep permutation-invariant functions for variable-size inputs. *ICLR*, 2019.

[79] Phuoc Nguyen, T. Tran, Nilmini Wickramasinghe, and Svetha Venkatesh. Deepr: A convolutional net for medical records. *IEEE Journal of Biomedical and Health Informatics*, 21:22–30, 2017.

[80] Phuoc Nguyen, Truyen Tran, and Svetha Venkatesh. Resset: A recurrent model for sequence of sets with applications to electronic medical records. *IJCNN*, pages 1–9, 2018.

[81] Chris Olah and Shan Carter. Attention and augmented recurrent neural networks. *Distill*, 2016.

[82] Kimberly J O'malley, Karon F. Cook, Matt D. Price, Kimberly Raiford Wildes, John F. Hurdle, and Carol M. Ashton. Measuring diagnoses: Icd code accuracy. *Health services research*, 40 5 Pt 2:1620–39, 2005.

[83] Chunjong Park, Anas Awadalla, Tadayoshi Kohno, and Shwetak Patel. Reliable and trustworthy machine learning for health using dataset shift detection. *ArXiv*, abs/2110.14019, 2021.

[84] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. *ArXiv*, abs/1912.01703, 2019.

[85] Xueping Peng, Guodong Long, Tao Shen, Sen Wang, Jing Jiang, and Michael Blumenstein. Temporal self-attention network for medical concept embedding. *ICDM*, pages 498–507, 2019.

[86] Jeffrey Pennington, R. Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *EMNLP*, 2014.

[87] S. Purushotham, Wilka Carvalho, Tanachat Nilanon, and Y. Liu. Variational recurrent adversarial deep domain adaptation. In *ICLR*, 2017.

[88] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*, pages 652–660, 2017.

[89] Joaquin Quionero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil Lawrence. Dataset shift in machine learning. 2009.

[90] Bone Balk R.A., Cerra F.B., Dellinger R.P., Fein A.M., Knaus W.A., Schein R. M.H., Sibbald W.J., and Members of the ACCP/SCCM Consensus Conference. Definitions for sepsis and organ failure and guidelines for the use of innovative therapies in sepsis. *Chest*, 101:1644–55, 1992.

[91] Alec Radford and Karthik Narasimhan. Improving language understanding by generative pre-training. 2018.

[92] Alvin Rajkomar, Eyal Oren, Kai Chen, et al. Scalable and accurate deep learning with electronic health records. *NPJ Digital Medicine*, 1, 2018.

[93] Alan Ramponi and Barbara Plank. Neural unsupervised domain adaptation in nlp—a survey. *ArXiv*, abs/2006.00632, 2020.

[94] Laila Rasmy, Yang Xiang, Ziqian Xie, Cui Tao, and D. Zhi. Med-bert: pretrained contextualized embeddings on large-scale structured electronic health records for disease prediction. *NPJ Digital Medicine*, 4, 2021.

[95] R. Riley, J. Ensor, Kym I E Snell, T. Debray, D. Altman, K. Moons, and G. Collins. External validation of clinical prediction models using big datasets from e-health records or ipd meta-analysis: opportunities and challenges. *The BMJ*, 353, 2016.

[96] Véronique L. Roger, Susan A. Weston, Margaret May Redfield, Jens P Hellermann-Homan, Jill M. Killian, Barbara P. Yawn, and Steven J. Jacobsen. Trends in heart failure incidence and survival in a community-based population. *JAMA*, 292 3:344–50, 2004.

[97] Ludger Rüschendorf. The wasserstein distance and approximation theorems. *Probability Theory and Related Fields*, 70:117–129, 1985.

[98] Carlos Sáez, Alba Gutiérrez-Sacristán, Isaac S. Kohane, Juan Miguel García-Gómez, and Paul Avillach. Ehrtemporalvariability: delineating temporal data-set shifts in electronic health records. *GigaScience*, 9, 2020.

[99] S. Saria and Adarsh Subbaswamy. Tutorial: Safe and reliable machine learning. *ArXiv*, abs/1904.07204, 2019.

[100] Jennifer L. St. Sauver, Brandon R. Grossardt, Cynthia L. Leibson, Barbara P. Yawn, L. Joseph Iii Melton, and Walter A. Rocca. Generalizability of epidemiological findings and public health decisions: an illustration from the rochester epidemiology project. *Mayo Clinic proceedings*, 87 2:151–60, 2012.

[101] Peter F. Schulam and Suchi Saria. Reliable decision support using counterfactual models. In *NIPS*, 2017.

[102] Mike Schuster and Kuldip K. Paliwal. Bidirectional recurrent neural networks. *IEEE Trans. Signal Process.*, 45:2673–2681, 1997.

[103] Junyuan Shang, Tengfei Ma, Cao Xiao, and Jimeng Sun. Pre-training of graph augmented transformers for medication recommendation. In *IJCAI*, 2019.

[104] Jian Shen, Yanru Qu, Weinan Zhang, and Yong Yu. Adversarial representation learning for domain adaptation. *ArXiv*, abs/1707.01217, 2017.

[105] Mervyn Singer, Clifford S. Deutschman, Christopher W. Seymour, Manu Shankar-Hari, Djillali Annane, Michael Bauer, Rinaldo Bellomo, Gordon R. Bernard, Jean Daniel Chiche, Craig M. Coopersmith, Richard S Hotchkiss, Mitchell M. Levy, John C. Marshall, Greg S. Martin, Steven M. Opal, Gordon David Rubenfeld, Tom van der Poll, Jean Louis Vincent, and Derek C. Angus. The third international consensus definitions for sepsis and septic shock (sepsis-3). *JAMA*, 315 8:801–10, 2016.

[106] Lihong Song, Chin Wang Cheong, Kejing Yin, William Kwok-Wai Cheung, Benjamin C. M. Fung, and Jonathan Poon. Medical concept embedding with multiple ontological representations. In *IJCAI*, 2019.

[107] E. Steinberg, K. Jung, Jason Alan Fries, Conor K. Corbin, S. Pfohl, and N. Shah. Language models are an effective patient representation learning technique for electronic health record data. *ArXiv*, abs/2001.05295, 2020.

[108] Adarsh Subbaswamy and Suchi Saria. From development to deployment: dataset shift, causality, and shift-stable models in health ai. *Biostatistics*, 2019.

[109] Baochen Sun, Jiashi Feng, and Kate Saenko. Return of frustratingly easy domain adaptation. In *AAAI*, 2016.

[110] Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *ECCV Workshops*, 2016.

[111] Zhe Sun, Shaoliang Peng, Yaning Yang, Xiaoqi Wang, and Fei Li. A general fine-tuned transfer learning model for predicting clinical task acrossing diverse ehrs datasets. *2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 490–495, 2019.

[112] Laine E Thomas, Fan Li, and Michael J. Pencina. Overlap weighting: A propensity score method that mimics attributes of a randomized clinical trial. *JAMA*, 2020.

[113] Michele Tonutti, E. Ruffaldi, A. Cattaneo, and C. Avizzano. Robust and subject-independent driving manoeuvre anticipation through domain-adversarial recurrent neural networks. *Robotics Auton. Syst.*, 115:162–173, 2019.

[114] Giovanni Tripepi, Kitty J. Jager, Friedo W Dekker, and Carmine Zoccali. Selection bias and information bias in clinical research. *Nephron Clinical Practice*, 115:c94 – c99, 2010.

[115] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2962–2971, 2017.

[116] Laurens van der Maaten and Geoffrey E. Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[117] Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *ArXiv*, abs/1706.03762, 2017.

[118] Jesse Vig. A multiscale visualization of attention in the transformer model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 37–42, Florence, Italy, July 2019. Association for Computational Linguistics.

[119] Cédric Villani. Optimal transport: Old and new. 2008.

[120] Oriol Vinyals, Samy Bengio, and Manjunath Kudlur. Order matters: Sequence to sequence for sets. In *ICLR*, 2015.

[121] S. Virani, Á. Alonso, E. Benjamin, M. Bittencourt, C. Callaway, A. Carson, A. Chamberlain, Alexander R Chang, Susan Cheng, Francesca N. Delling, L. Djoussé, M. Elkind, J. Ferguson, M. Fornage, S. Khan, B. Kissela, K. Knutson, T. Kwan, D. Lackland, T. Lewis, J. Lichtman, C. Longenecker, M. Loop, P. Lutsey, S. Martin, K. Matsushita, A. Moran, M. Mussolino, A. M. Perak, W. Rosamond, Gregory A. Roth, U. Sampson, G. Satou, E. Schroeder, Svati H. Shah, C. Shay, N. Spartano, A. Stokes, D. Tirschwell, L. VanWagner, and C. Tsao. Heart disease and stroke statistics—2020 update: A report from the american heart association. In *Circulation*, 2020.

[122] Erica A. Voss, Rupa Makadia, Amy Matcho, Qianli Ma, Chris Knoll, Martijn J. Schuemie, Frank Defalco, Ajit A Londhe, Vivienne J. Zhu, and Patrick B. Ryan. Feasibility and utility of applications of the common data model to multiple, disparate observational health databases. *Journal of the American Medical Informatics Association : JAMIA*, 22:553 – 564, 2015.

[123] Mingliang Wang, Daoqiang Zhang, Jiashuang Huang, Pew-Thian Yap, Dinggang Shen, and Mingxia Liu. Identifying autism spectrum disorder with multi-site fmri via low-rank domain adaptation. *IEEE Transactions on Medical Imaging*, 39:644–655, 2020.

[124] Mark G. Weiner. Point: Is icd-10 diagnosis coding important in the era of big data? yes. *Chest*, 153 5:1093–1095, 2018.

[125] Gerhard Widmer and Miroslav Kubát. Learning in the presence of concept drift and hidden contexts. *Machine Learning*, 23:69–101, 2004.

[126] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771, 2019.

[127] T. Wollmann, C. S. Eijkman, and K. Rohr. Adversarial domain adaptation to improve automatic breast cancer grading in lymph nodes. In *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*, pages 582–585, 2018.

[128] Jelmer M. Wolterink, Anna M. Dinkla, Mark Savenije, Peter R. Seevinck, Cornelis A. T. van den Berg, and Ivana Isgum. Deep mr to ct synthesis using unpaired data. In *SASHIMI@MICCAI*, 2017.

[129] Cao Xiao, Tengfei Ma, Adji B. Dieng, David M. Blei, and Fei Wang. Readmission prediction via deep contextual embedding of clinical concepts. *PLoS ONE*, 13, 2018.

[130] Ke Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron C. Courville, Ruslan Salakhutdinov, Richard S. Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, 2015.

[131] Wen Xu, Jing He, and Yanfeng Shu. Transfer learning and deep domain adaptation. In Marco Antonio Aceves-Fernandez, editor, *Advances and Applications in Deep Learning*, chapter 3. IntechOpen, Rijeka, 2020.

[132] Wenjun Yan, Yuanyuan Wang, Menghua Xia, and Qian Tao. Edge-guided output adaptor: Highly efficient adaptation module for cross-vendor medical image segmentation. *IEEE Signal Processing Letters*, 26:1593–1597, 2019.

[133] Yiqin Yu, Pin-Yu Chen, Y. Zhou, and Jing Mei. Adversarial sample enhanced domain adaptation: A case study on predictive modeling with electronic health records. *ArXiv*, abs/2101.04853, 2021.

[134] Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola. Deep sets. In *NIPS*, pages 3391–3401, 2017.

[135] John R. Zech, Marcus A. Badgeley, Manway Liu, Anthony Beardsworth Costa, Joseph J. Titano, and Eric Karl Oermann. Variable generalization performance of a deep learning model to detect pneumonia in chest radiographs: A cross-sectional study. *PLoS Medicine*, 15, 2018.

[136] Aston Zhang, Zachary Chase Lipton, Mu Li, and Alex Smola. Dive into deep learning. *ArXiv*, abs/2106.11342, 2021.

[137] Jianshu Zhang, Jun Du, and Lirong Dai. A gru-based encoder-decoder approach with attention for online handwritten mathematical expression recognition. *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, 01:902–907, 2017.

[138] Tianran Zhang, Muhao Chen, and Alex A. T. Bui. Diagnostic prediction with sequence-of-sets representation learning for clinical events. *Artificial intelligence in medicine. Conference on Artificial Intelligence in Medicine*, 12299:348–358, 2020.

[139] Y. Zhang, Xi Yang, J. Ivy, and Min Chi. Time-aware adversarial networks for adapting disease progression modeling. *2019 IEEE International Conference on Healthcare Informatics (ICHI)*, pages 1–11, 2019.

[140] Yan Zhang, Jonathon S. Hare, and Adam Prügel-Bennett. Fspool: Learning set representations with featurewise sort pooling. *ICLR*, 2020.

[141] Guangyu Zhou, Muhao Chen, Chelsea Ju, et al. Mutation effect estimation on protein-protein interactions using deep contextualized representation learning. *NAR Genom Bioinform*, 2020.