Counterfactual and Factual Reasoning over Hypergraphs for Interpretable Clinical Predictions on EHR

Ran Xu RAN.XU@EMORY.EDU
Department of Computer Science, Emory University, Atlanta, GA 30322

Yue Yu YUEYU@GATECH.EDU
College of Computing, Georgia Institute of Technology, Atlanta, GA 30332

Chao Zhang CHAOZHANG@GATECH.EDU
College of Computing, Georgia Institute of Technology, Atlanta, GA 30332

Mohammed K Ali MKALI@EMORY.EDU
Rollins School of Public Health, Emory University, Atlanta, GA 30322

Joyce C Ho JOYCE.C.HO@EMORY.EDU
Department of Computer Science, Emory University, Atlanta, GA 30322

Carl Yang J.CARLYANG@EMORY.EDU
Department of Computer Science, Emory University, Atlanta, GA 30322

Abstract

Electronic Health Record modeling is crucial for digital medicine. However, existing models ignore higher-order interactions among medical codes and their causal relations towards downstream clinical predictions. To address such limitations, we propose a novel framework CACHE, to provide *effective* and *insightful* clinical predictions based on hypergraph representation learning and counterfactual and factual reasoning techniques. Experiments on two real EHR datasets show the superior performance of CACHE. Case studies with a domain expert illustrate a preferred capability of CACHE in generating clinically meaningful interpretations towards the correct predictions.

Keywords: EHR, Hypergraph, Counterfactual and Factual Reasoning

1. Introduction

Electronic Health Record (EHR) data contain rich information about patients such as diagnosis, medication and lab results, and have been widely used to identify patterns for patients and assist with clinical decisions. In recent years, there has been a strong interest to leverage machine learning techniques to support digital medicine (Fogel and Kvedar, 2018) such as diagnosis prediction (Ma et al., 2017), predictive phenotyping (Fu et al., 2019), and drug recommendation (Yang et al., 2021).

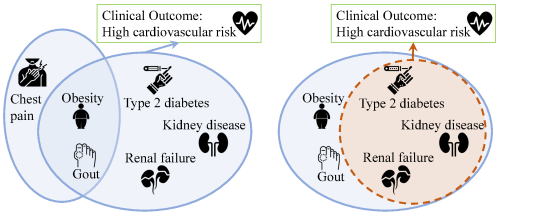
Despite its tremendous importance, it is often non-trivial to model the EHR data for supporting clinical decision-making. While there exist numerous studies in this direction, such as proximity-based embedding techniques (Choi et al., 2016a) and graph neural networks (GNNs) (Choi et al., 2020; Zhu and Razavian, 2021; Ochoa and Mustafa, 2022, *i.a.*) to learn the relations among visits and medical codes, these works are hindered by the following limitations:

Challenge I: Limited expressive power. The co-occurrence relationships between visits and medical codes are often complex. A

visit typically contains a large set of medical codes including diagnosis, medication, and procedure codes with varying sizes. Each medical code can also appear across a set of visits. As a result, it is crucial to represent the set information to effectively capture the relations among these units. Unfortunately, existing models only consider pairwise relations and are not well designed for learning *set* representations. Thus, directly adopting these approaches can yield suboptimal performance for downstream clinical tasks.

Challenge II: Non-interpretable prediction. Previous works mainly focus on improving the predictive performance with deep neural networks and are usually non-transparent. An equally, if not more, critical issue for clinical predictive models is the *interpretability* (Tonekaboni et al., 2019), as understanding how predictions are made by the model is crucial for clinical experts to plan for the treatment. While attention weights have been proposed to fulfill this purpose (Ma et al., 2017; Yu et al., 2020a), their validity have been challenged (Serrano and Smith, 2019) as attention weights can be biased and misleading. Thus, it remains an important challenge to design accurate and interpretable models for EHR modeling.

Motivated by the challenges above, we propose CACHE[1], for predicting patients' clinical outcomes with interpretability. CACHE includes the following two key designs: (1) To effectively learn the representations of visits and medical codes, we propose to leverage *hypergraphs* to model the higher-order relations among them, where medical codes are regarded as nodes and visits are considered as hyperdeges (Cai et al., 2022). An example is shown in Fig. 1, where each hyperedge connects all the medical codes involved in the corresponding visit. With the constructed hypergraph, we harness the powerful set transformer (Lee et al., 2019; Chien et al., 2022) to capture the data permutation invariance property. This enables our model to go beyond pairwise interactions and gives it great expressive power to learn better representations for both nodes and hyperedges. (2) To raise the interpretability of CACHE, we extract an important set of medical codes for each visit that are both *sufficient* and *necessary* for making the correct clinical predictions. The subset in Fig. 1 is an example of such an important set that contains the key factors of the hyperedge, as kidney disease, diabetes and heart disease are highly correlated. To this end, we build a subset generation module that considers *factual* and *counterfactual* reasoning objectives simultaneously (Guidotti et al., 2019), which target sufficiency and necessity respectively. With these two techniques, CACHE makes both *effective* and *insightful* predictions to support clinical decision making.



(a) Hypergraph Construction (b) Interpretable Subset Extraction

Figure 1: An example of hypergraph construction on EHR. Blue circles are hyperedges, and the red circle is the interpretable subset for the hyperedge that leads to the outcome.

We conduct experiments on two datasets, namely MIMIC-III and CRADLE for the important and accessible clinical tasks of phenotypes prediction and cardiovascular disease risk prediction. The results illustrate that CACHE achieves superior performance with the average gain of 3.2% in AUROC

1. short for Counterfactual and Factual Reasoning over Hypergraphs of EHR.

and 7.5% in AUPR. Furthermore, CACHE is able to characterize the most important subsets for each visit on the target tasks. Compared with the attention-based explanation, CACHE generates more reasonable subsets evaluated by a domain expert, justifying its efficacy in providing clinically useful interpretations.

Reproducibility The code for CACHE can be found at https://github.com/ritaranx/CACHE.

2. Related Work

With the development of deep neural networks (DNNs), earlier research has explored learning dense representations for medical concepts (Choi et al., 2016b,a; Fu et al., 2019; Cui et al., 2022b) to support clinical predictions. However, the embeddings are learned in a static way and are unaware of downstream prediction tasks.

To overcome this drawback, graph-based models have been proposed for EHR modeling. They first build a co-occurrence graph from the EHR data, and then leverage graph neural networks (GNNs) to learn the relations among medical codes within each encounter for clinical outcome prediction (Choi et al., 2017, 2020; Wang et al., 2020; Ochoa and Mustafa, 2022). However, their graph structures are usually predefined with domain expertise (Choi et al., 2020), or prior knowledge (Liu et al., 2020), which can be expensive to obtain and are less generalizable. Besides, the GNNs used in their studies are only able to encode pairwise relations, which is not ideal in EHR modeling, given the large set of medical codes involved in each visit.

To the best of our knowledge, HCL (Cai et al., 2022) is the only work that adopts hypergraph learning for EHRs. They generate medical code graphs and patient graphs out of the constructed hypergraph, and leverage contrastive learning to aggregate information from different graphs. However, they focus on combining self-supervised learning techniques with hypergraph learning, while we provide interpretable predictions via counterfactual and factual reasoning.

Compared with learning accurate clinical predictive models, developing *interpretable* models for EHR data has been less studied despite its great significance. Till now, most of techniques focus on harnessing the attention weights (Ma et al., 2017; Mincu et al., 2021; Zhu and Razavian, 2021; Kan et al., 2022) as explainations, while the validity of such explanations are more ambivalent (Jain and Wallace, 2019; Serrano and Smith, 2019) without sufficient human studies. Different from them, we aim to leverage factual and counterfactual explanations to interpret the model's decisions. In the context of graph-based learning, such explanation methods look for a small subset of nodes or edges such that preserving them will retain predictions but removing them would flip the predictions (Ying et al., 2019; Lucic et al., 2022; Tan et al., 2022; Cui et al., 2022a). This has also been applied to EHR data but focusing on the survival analysis (Li et al., 2021; Wang and Sun, 2022; Chapfuwa et al., 2021) or fairness of clinical predictions over demographics (Pfohl et al., 2019), thus are orthogonal to our proposed approach.

3. Method

The overview of CACHE is shown in Figure 2. Notably, there are two key components, namely *hypergraph neural network* and *interpretable subset extraction*. The hypergraph neural network takes the original hypergraph $\mathcal{G}$ as input to learn its node and hyperedge embeddings (Sec. 3.2). Then the subset extraction model learns a weight for each node in a hyperedge from their concatenated embeddings. Finally, the interpretable subset $\mathcal{G}'$ and its complementary set $\mathcal{G}\backslash\mathcal{G}'$ are gen-

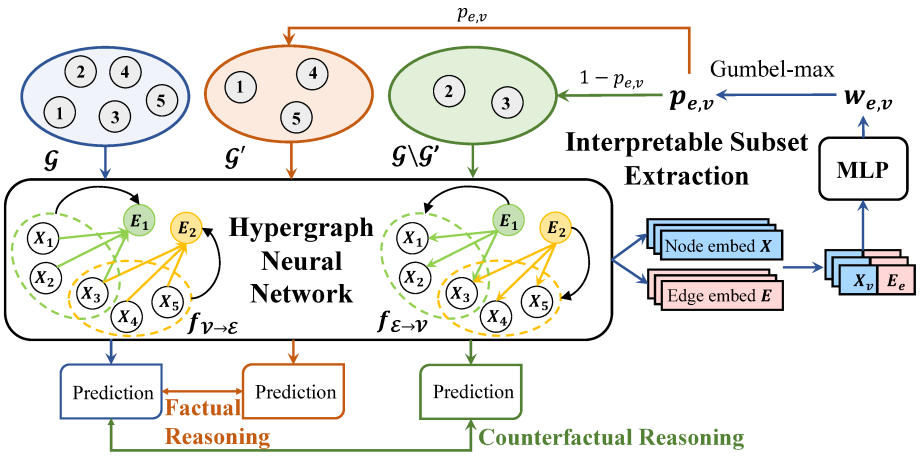


Figure 2: The framework of CACHE. The grey circles with numbers on them denote the nodes within a hyperedge. We describe the details of *hypergraph neural network* and *interpretable subset extraction* in Sec. 3.2 and Sec. 3.3 respectively.

erated (Sec. 3.3). The two key components are trained in an alternate way (Sec. 3.4).

3.1. Notations and Definitions

The EHR data used in this work comprises multiple types of medical codes $\mathcal{C}$, including diseases, medications, procedures and services. For each patient, the *input* of our method is the medical record $\mathcal{X}$ containing a set of medical codes, where $\mathcal{X} \subset \mathcal{C}$.

Our problem is: given the clinical record $\mathcal{X}$, we aim to (1) predict the clinical outcome y of that patient; (2) generate a subset $\widetilde{\mathcal{X}} \subset \mathcal{X}$ of the most important elements in $\mathcal{X}$ that provides interpretations into the predictions.

3.2. Hypergraph Construction and Learning

◇ **Hypergraph Construction.** One characteristic of EHR data is that each visit contains massive medical codes and each medical code appears in multiple visits. Thus, we leverage *hypergraph* structure to model their high-order interactions, and project elements into an unified low-dimension space to facilitate prediction tasks. To transform the EHR into hypergraphs, we follow Cai et al. (2022) to view each clinical visit as a hyperedge and each medical code as a node. We denote the hypergraph as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}, \mathcal{E}$ stand for nodes and hyperedges, respectively.

◇ **Hypergraph Learning Module f_θ.** Compared with vanilla graph neural networks (Kipf and Welling, 2017; Veličković et al., 2018), a particular challenge for learning on hypergraphs is *how to design propagation rules for both nodes and hyperedges.* Directly using average pooling for aggregation (Feng et al., 2019; Yu et al., 2020b) can be suboptimal, as it takes all node information equally and loses structural information. To overcome this issue, we design the hypergraph learning module defined as $f_\theta(\mathcal{G})$ parameterized by θ, to perform message passing on hypergraphs. Specifically, we leverage the *set transformer*, a principled, permutation-invariant model to aggregate the neighborhood information (Lee et al., 2019; Chien et al., 2022).

Denote the embeddings of nodes and hyperedges on l-th layer as $\boldsymbol{X}^{(l)} \in \mathbb{R}^{|\mathcal{V}| \times d}$, $\boldsymbol{E}^{(l)} \in \mathbb{R}^{|\mathcal{E}| \times d'}$ where d and d' are two hyperparameters. In l-th layer, the message passing follows two steps

$$\boldsymbol{E}_e^{(l)} = f_{\mathcal{V} \to \mathcal{E}} \left(\mathcal{V}_{e, \boldsymbol{X}^{(l-1)}} \right), \tag{1}$$

$$\boldsymbol{X}_v^{(l)} = f_{\mathcal{E} \to \mathcal{V}} \left(\mathcal{E}_{v, \boldsymbol{E}^{(l)}} \right). \tag{2}$$

Here $\boldsymbol{E}_e$ and $\boldsymbol{X}_v$ stand for the embeddings of hyperedge e and node v, respectively. $\mathcal{V}_{e,\boldsymbol{X}}$ is the hidden representations of node that contain the hyperedge e, and $\mathcal{E}_{v,\boldsymbol{E}}$ is the hidden representations of hyperedges that contain the node v. To realize the two message passing function $f_{\mathcal{V} \to \mathcal{E}}(\cdot)$ and $f_{\mathcal{E} \to \mathcal{V}}(\cdot)$, we use self-attention (Vaswani et al., 2017) function which has strong expressive power and can identify the most relevant elements within the set for message passing. Thus,

$$f_{\mathcal{V} \to \mathcal{E}}(\boldsymbol{S}) = f_{\mathcal{E} \to \mathcal{V}}(\boldsymbol{S}) = \text{Self-Att}(\boldsymbol{S}), \tag{3}$$

where the mathematical formulation of self-attention is written as

$$\text{Self-Att}(\boldsymbol{S}) = \text{LayerNorm}(\boldsymbol{Y} + \text{FFN}(\boldsymbol{Y})). \tag{4}$$

Note that $\boldsymbol{S} \in \mathbb{R}^{|\boldsymbol{S}| \times d}$ is the embedding of the input set and $\boldsymbol{Y} \in \mathbb{R}^{1 \times d}$ is the representation of $\boldsymbol{S}$ after multi-head self-attention, denoted as

$$\boldsymbol{Y} = \text{LayerNorm}(\boldsymbol{S} + \text{MultiHead}(\boldsymbol{S})),$$

where

$$\text{MultiHead}(\boldsymbol{S}) = \|_{i=1}^{h} \mathbf{O}^{(i)} = \|_{i=1}^{h} \text{SA}_i(\boldsymbol{S}),$$

$$\text{SA}_i(\boldsymbol{S}) = \text{softmax}\left(\frac{\mathbf{W}_i^Q (\boldsymbol{S}\mathbf{W}_i^K)^\top}{\sqrt{\lfloor d/h \rfloor}} \right) \boldsymbol{S}\mathbf{W}_i^V.$$

In the above equations, $\mathbf{W}_i^Q \in \mathbb{R}^{1 \times \lfloor d/h \rfloor}$, $\mathbf{W}_i^K \in \mathbb{R}^{d \times \lfloor d/h \rfloor}$, $\mathbf{W}_i^V \in \mathbb{R}^{d \times \lfloor d/h \rfloor}$, together with the feed-forward neural network (FFN), which is realized with a 2-layer Multi-layer Perceptron (MLP), are trainable parameters, and h is the number of attention heads. It is worth noting that we do not include *position encoding* in the original transformer paper due to the lack of sequential information for our datasets.

By stacking L set transformer layers together, we obtain the embeddings at the last layer for hyperedges as $\boldsymbol{E}^{(L)}$ and nodes as $\boldsymbol{X}^{(L)}$. In experiments, we choose $L = 3$.

However, we find that the model has the over-smoothing issue. As EHR graphs are large and dense, the embeddings after message passing can be less distinguishable from one another but in reality should be quite different (Oono and Suzuki, 2019). To alleviate this issue, we add additional normalization for embeddings in each layer, defined as (we use hyperedges as an example, node embeddings are processed in a similar way):

$$\begin{aligned} \boldsymbol{E}_e^c &= \boldsymbol{E}_e - \frac{1}{|\mathcal{E}|} \sum_{e' \in \mathcal{E}} \boldsymbol{E}'_e \\ \widetilde{\boldsymbol{E}}_e &= \frac{\boldsymbol{E}_e^c}{\sqrt{\frac{1}{|\mathcal{E}|} \sum_{e' \in \mathcal{E}} \left\| \boldsymbol{E}_{e'}^c \right\|_2^2}} = \sqrt{|\mathcal{E}|} \cdot \frac{\boldsymbol{E}_e^c}{\sqrt{\|\boldsymbol{E}_e^c\|_F^2}}. \end{aligned} \tag{5}$$

We remark that this so-called *PairNorm* technique (Zhao and Akoglu, 2019) keeps the total pairwise embedding distances over hyperedges unchanged across layers to prevent them from being identical.

In addition, to support the downstream tasks with the embedding, we stack a classification head on visit embeddings from *all layers* $\widetilde{\boldsymbol{E}}_e^{(l)} (1 \le l \le L)$ as

$$\widehat{y}_e = \sigma \left(\text{MLP}_{\text{CLS}} \left(\|_{l=1}^{L} \widetilde{\boldsymbol{E}}_e^{(l)} \right) \right); \tag{6}$$

where MLP_{CLS} is a 2-layer neural network that converts the vector to a value for binary classification, and $\sigma(x) = \frac{1}{1+\exp(-x)}$ is the sigmoid function. By stacking the embeddings from different layers (a.k.a *jump-*

ing knowledge (Xu et al., 2018)), we further resolve the oversmoothing issue[2].

For the target classification task, we use the binary cross-entropy as the learning objective defined as

$$\ell_{\text{cls}} = -y \log(\widehat{y}_e) - (1-y)\log(1-\widehat{y}_e). \quad (7)$$

3.3. Interpretable Subset Extraction

The above section describes how CACHE supports clinical predictions with the hypergraph neural networks and set transformers. However, it does not provide the capability to explain the clinical predictions. To achieve this, we aim to generate a subset $\mathcal{V}'_e \subset \mathcal{V}_e$ for each hyperedge $e \in \mathcal{E}$ to serve as *local explanations* for model predictions. In particular, we denote the hypergraph with only the subset nodes for each hyperedge as $\mathcal{G}'$, and hypothesize that there should be two key properties for $\mathcal{G}'$: (1) *sufficiency*: the prediction of $f_\theta(\mathcal{G}')$ using the subsets only will be *consistent* based on *factual reasoning*; (2) *necessity*: removing the subset will result in opposite predictions for $f_\theta(\mathcal{G}\backslash\mathcal{G}')$ based on *counterfactual reasoning*.

To fulfill this purpose, a *learnable* interpretable subset extraction model g_ϕ is proposed to dynamically select the most important subsets for hyperedges in $e \in \mathcal{E}$. Specifically, for each hyperedge e with its associated nodes $v \in \mathcal{V}_e$, we assign a random variable $p_{e,v} \sim \text{Bern}(\omega_{e,v})$, where v is preserved in hyperedge e if $p_{e,v} > 0.5$ and is filtered otherwise. We use another 2-layer MLP as a realization of g_ϕ for parameterizing the probability weight $\omega_{e,v}$, with the representation of e and v from f_θ as

$$\omega_{e,v} = \text{MLP}\left([\boldsymbol{E}_e^{(l)}; \boldsymbol{X}_v^{(l)}]\right). \quad (8)$$

2. We note that these additional techniques (PairNorm, Jumping Knowledge) are used by default for both CACHE and baselines. See Sec. 4.5 for more discussions.

To facilitate end-to-end training of g_ϕ, we use the Gumbel-max trick (Jang et al., 2017) to differentiate $p_{e,v}$ based on $\omega_{e,v}$ as

$$\widehat{p}_{e,v} = \sigma\left((\log(\delta/(1-\delta)) + \omega_{e,v})/\tau\right), \quad (9)$$

where $\delta \sim \text{Uniform}(0,1)$ and τ is a temperature hyper-parameter. With the generated $\mathcal{G}' \sim g_\phi(\mathcal{G})$, we define the prediction for factual and counterfactual reasoning for each hyperedge e with label as

$$\widehat{y}_{\text{f}} = f_\theta(\mathcal{G}');\ \widehat{y}_{\text{cf}} = f_\theta(\mathcal{G}\backslash\mathcal{G}'), \quad (10)$$

and the loss can be written as

$$\ell_{\text{f}} = \begin{cases} [\gamma + \widehat{y}_e - \widehat{y}_{\text{f}}]_+, & \text{if } y_e = 1; \\ [\gamma + \widehat{y}_{\text{f}} - \widehat{y}_e]_+, & \text{else.} \end{cases} \quad (11)$$

and

$$\ell_{\text{cf}} = \begin{cases} [\gamma + \widehat{y}_{\text{cf}} - \widehat{y}_e]_+, & \text{if } y_e = 1; \\ [\gamma + \widehat{y}_e - \widehat{y}_{\text{cf}}]_+, & \text{else.} \end{cases} \quad (12)$$

where $[x]_+ = \max(x, 0)$ and $\gamma = 0.5$ is the pre-defined threshold. In this way, we encourage g_ϕ to find a subset to generate $\mathcal{G}'$ which shares the *same* prediction as using the whole graph $\mathcal{G}$, while generating *different* prediction with the graph $\mathcal{G}\backslash\mathcal{G}'$. Besides, to force g_ϕ to generate concise subsets, we add additional regularization on the weight $\omega_{e,v}$. To sum up, the learning objective of g_ϕ is expressed as

$$\mathcal{L}_g = \mathbb{E}_{e\sim p(\mathcal{E})}\mathbb{E}_{v\sim p(\mathcal{V}_e)}[\alpha\ell_f + (1-\alpha)\ell_{\text{cf}} + \lambda_v\omega_{e,v}], \quad (13)$$

where α and λ_v are hyperparameters.

3.4. Alternate Training of f_θ and g_ϕ

To incorporate the factual and counterfactual learning during the training of f_θ, we augment the learning loss with the factual and counterfactual loss as

$$\mathcal{L}_{\text{cls}} = \mathbb{E}_{e\sim p(\mathcal{E})}\{\ell_{\text{cls}} + \lambda_m \mathbb{E}_{v\sim p(\mathcal{V}_e)} [\alpha\ell_{\text{f}} + (1-\alpha)\ell_{\text{cf}}]\}, \quad (14)$$

Table 1: Dataset statistics. For # of hyperedges in MIMIC-III, the first number indicates the hyperedges without labels, while the second one indicates ones with labels.

Stats	MIMIC-III	CRADLE
# of diagnosis	846	7915
# of medication	4525	489
# of procedure	2032	4321
# of service	20	—
# of hyperedges	36875/12353	36611

where ℓ_{cls} is defined in Eq. 7. Joint optimizing f_θ and g_ϕ can be challenging, as directly optimizing them together often cause the model to collapse. For better stability, we use alternate gradient descent (Xu et al., 2019) to train f_θ and g_ϕ. We first train f_θ with Eq. 7 for 10 epochs as the warmup. After that, we train g_ϕ while fixing f_θ as $\phi = \phi - \lambda_{\text{cls}} \nabla_\phi \mathcal{L}_{\text{g}}$. Then, with the generated $\mathcal{G}'$ containing the important subsets, we train f_θ while fixing g_ϕ as $\theta = \theta - \lambda_{\text{g}} \nabla_\theta \mathcal{L}_{\text{cls}}$, where λ_{g} and λ_{cls} are learning rates. Finally, the generated medical code subset for hyperedge e with g_ϕ is regarded as the interpretable elements to support the clinical predictions.

4. Experiments

4.1. Experiment Setup

◇ **Datasets.** We conduct experiments on two clinical prediction datasets: MIMIC-III (Johnson et al., 2016) and a private dataset CRADLE. CRADLE was collected from a large healthcare system in United States. The statistics of two datasets are shown in Table 1. We split them into train/validation/test set by 7:1:2. Their label distributions are shown in Appendix A.

◇ **Tasks.** We perform phenotyping prediction on MIMIC-III. Phenotyping has a wide range of applications such as morbidity detection, repurposing drugs, and diagnosis (Oellrich et al., 2016). In this task, we conduct a multi-label classification (Harutyunyan et al., 2019), that predicts whether the 25 acute care conditions (described in Appendix A) will be present in patients' next visits, given their current ICU records.

We also conduct an outcome prediction task on CRADLE, which predicts whether the patients with type 2 diabetes would experience cardiovascular disease (CVD) endpoints within a year after the initial diagnosis. The CVD endpoint is defined as the presence of coronary heart disease (CHD), congestive heart failure (CHF), myocardial infarction (MI), or Stroke, which are identified by their ICD-9 and ICD-10 clinical codes. As shown in (Einarson et al., 2018), CVD is estimated to affect around 32% of the patients with diabetes, and thus a systematic CVD risk prediction is especially needed. More descriptions are in Appendix A.

◇ **Metrics.** Since the label distribution of both MIMIC-III and CRADLE are imbalanced, we use Accuracy, AUROC, AUPR and Macro-F1 score as the metrics (Choi et al., 2020; Cai et al., 2022). For accuracy and F1 score, we use 0.5 as the threshold after obtaining the predicted results.

4.2. Implementation Details

We implement our model in PyTorch[3]. We use Adam as the optimizer for both the hypergraph learning module and the important subset extraction module, and tune their learning rates in {1e-2, 5e-3, 1e-3, 5e-4}. Other key hyperparameters include α and λ_v in Eq. 13 and λ_m in Eq. 14. We set $\alpha = 0.5$ to balance between factual and counterfactual reasoning. We study the effect of α, λ_m and λ_v in Section 4.5. For our experiments, we set $\lambda_g = 0.01$, $\lambda_{\text{cls}} =$ 1e-3, $\lambda_{\text{m}} = 0.01$,

3. https://pytorch.org/

λ_v = 1e-3, $\alpha = 0.5$, $d = 48$, $h = 4$, dropout to 0 and weight decay to 1e-3. We use 3-layers in hypergraph neural networks.

4.3. Baselines

We compare CACHE with a comprehensive set of baselines:

◇ **Non-graph Baselines.** These baselines model EHR data *without* using graphs to encode relations among items. We select *Logistic Regression (LR)* (Menard, 2002), *Support Vector Machine (SVM)* (Cortes and Vapnik, 1995) and *Multi-layer Perceptron (MLP)* (Naraei et al., 2016) as baselines.

◇ **Graph-based Baselines.** These methods use graph-based approach for modeling the relations. Specifically, for two items, an edge exists only when they co-occur in a visit. We consider two baselines: *Graph Convolutional Transformer (GCT)* (Choi et al., 2020), which learns the hidden EHR structure for predictive tasks, and *Graph Attention Networks (GAT)* (Veličković et al., 2018), which uses attention-based massage passing mechanism for aggregating neighbor features. For these two methods, a task-specific MLP is stacked on the top of the model for prediction.

◇ **Hypergraph-based Baselines.** These baselines uses the same hypergraph structure as CACHE but with different neural architectures for learning on hypergraphs. Specifically, we select several representative methods including *Hypergraph Neural Networks (HGNN)* (Feng et al., 2019), *Hypergraph Convolutional Networks (HyperGCN)* (Yadati et al., 2019), *Hypergraph Convolution and Hypergraph Attention (HCHA)* (Bai et al., 2021), *AllSetTransformer* (Chien et al., 2022) in our experiments. We also consider the contrastive learning technique (denoted as CL) in a recently-proposed hypergraph learning approach for EHR (Cai et al., 2022)[4].

4.4. Experimental Results

Table 2 summarizes the experimental results on the two datasets. Note that accuracy and F1 are influenced by the threshold used for separating predicted scores into different classes, and thus are less comprehensive in demonstrating model performance. From the results, we have the following findings:

◇ CACHE outperforms all the baselines over four different evaluation metrics on both datasets, including our backbone model AllSetTransformer. Compared to the best baselines, CACHE raises the performance by 3.2% in AUROC and 7.5% in AUPR. This indicates that our leverage of counterfactual and factual reasoning contributes to the final performance, as it finds the salient subsets for the downstream predictions.

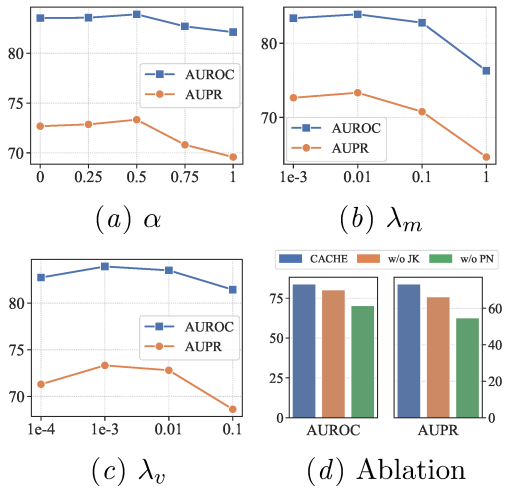
◇ We build additional contrastive learning (CL) on top of AllSetTransformer to study its efficacy, and the result shows that the improvement is marginal. This is because CL focuses on generating more samples for the model to learn the similar attributes among them, which does not necessarily align with the main objective of the tasks.

◇ Graph-based models generally have a better performance than traditional machine learning methods. This phenomenon verifies that considering the interaction between nodes via message passing is beneficial for EHR modeling. In addition, hypergraph-based models can further improve over the graph-based models. Among them, AllSetTransformer has a better performance than others, which illustrates that set function better models the hypergraph structure.

4. Since the code is not publicly available, we only test the contrastive learning technique as their main contribution.

Table 2: Performance on MIMIC-III and CRADLE compared with different baselines. The result is averaged over 5 runs. We use * to indicate statistically significant results ($p < 0.05$).

Model	MIMIC-III				CRADLE			
	ACC	AUROC	AUPR	F1	ACC	AUROC	AUPR	F1
LR	68.66 ± 0.24	64.62 ± 0.25	45.63 ± 0.32	13.74 ± 0.40	76.22 ± 0.30	57.22 ± 0.28	25.99 ± 0.26	42.18 ± 0.35
SVM	72.02 ± 0.12	55.10 ± 0.14	34.19 ± 0.17	32.35 ± 0.21	68.57 ± 0.13	53.57 ± 0.11	23.50 ± 0.15	52.34 ± 0.22
MLP	70.73 ± 0.24	71.20 ± 0.22	52.14 ± 0.23	16.39 ± 0.30	77.02 ± 0.17	63.89 ± 0.18	33.28 ± 0.23	45.16 ± 0.26
GCT	76.58 ± 0.23	78.62 ± 0.21	63.99 ± 0.27	35.48 ± 0.34	77.26 ± 0.22	67.08 ± 0.19	35.90 ± 0.20	56.66 ± 0.25
GAT	76.75 ± 0.26	78.89 ± 0.12	66.22 ± 0.29	34.88 ± 0.33	77.82 ± 0.20	66.55 ± 0.27	36.06 ± 0.18	56.43 ± 0.26
HGNN	77.93 ± 0.41	80.12 ± 0.30	68.38 ± 0.24	40.04 ± 0.35	76.77 ± 0.24	67.21 ± 0.25	37.93 ± 0.18	58.05 ± 0.23
HyperGCN	78.01 ± 0.23	80.34 ± 0.15	67.68 ± 0.16	39.29 ± 0.20	78.18 ± 0.11	67.83 ± 0.18	38.28 ± 0.19	60.24 ± 0.21
HCHA	78.07 ± 0.28	80.42 ± 0.17	68.56 ± 0.15	37.78 ± 0.22	78.60 ± 0.15	68.05 ± 0.17	39.23 ± 0.13	59.26 ± 0.21
AllSetTransformer	79.07 ± 0.31	82.19 ± 0.13	71.08 ± 0.17	41.51 ± 0.25	79.76 ± 0.18	70.07 ± 0.13	40.92 ± 0.12	61.23 ± 0.18
AllSetTransformer+CL	78.98 ± 0.44	82.56 ± 0.32	71.09 ± 0.27	39.33 ± 0.40	79.49 ± 0.21	69.60 ± 0.26	41.48 ± 0.25	56.65 ± 0.31
CACHE	**80.41 ± 0.21***	**83.91 ± 0.17***	**73.33 ± 0.18***	**47.28 ± 0.22***	**80.77 ± 0.19***	**73.34 ± 0.22***	**46.40 ± 0.18***	**63.92 ± 0.24***



(a) α (b) λ_m (c) λ_v (d) Ablation

Figure 3: Effect of different components of CACHE on MIMIC-III. JK denotes Jumping Knowledge and PN denotes PairNorm.

4.5. Parameter and Ablation Studies

We study the effect of different parameters of CACHE on MIMIC-III, shown in Figures 3(*a*), 3(*b*) and 3(*c*). Due to space limitation, results on CRADLE are in Appendix C. α in Eq. 13 balances between the counterfactual and factual reasoning. We observe that both types of reasoning are beneficial, as the performance of leveraging only one type of them (i.e. $\alpha = 0$ or 1) is worse than considering both (i.e. $\alpha = 0.25$, 0.5 or 0.75). The model reaches the best performance when $\alpha = 0.5$ by giving equal weights to both parts, which is consistant with the observations in (Tan et al., 2022). λ_m in Eq. 14 is a weight between counterfactual and factual loss and model classification loss, and the performance achieves the best with λ_m =0.01, indicating a certain ratio between the importance of the two losses. λ_v in Eq. 13 controls the sparsity of remaining nodes. As λ_v get larger, the performance first increases and then decreases. This is because when λ_v is too small or too large, g_ϕ would either keep or remove all the nodes, respectively, and thus hinders the model from learning the causal relationship between the selected subset and the downstream tasks.

We also inspect different components of CACHE in Figure 3(*d*). It is observed that both Jumping Knowledge and PairNorm contribute to model performance as they solve the over-smoothing issue. More ablation studies for baselines are shown in Appendix C to verify that these techniques work well for majority of models on EHR learning.

Subset 1 (generated from CACHE)	Long-term drug therapy
	Restless legs
	Mononeuritis
	Screening for malignant neoplasm of colon
	Albumin; urine (eg, microalbumin), semiquantitative (eg, reagent strip assay)
Subset 2 (generated from attention weights of AllSetTransformer)	Abnormal finding on evaluation procedure
	Nephropathy screening
	Disorder of nervous system due to diabetes mellitus
	Long-term drug therapy
	Mononeuritis
Label	0 (The patient would not experience CVD complications in the next year.)

Figure 4: Case Study: Comparison of the subsets generated from CACHE and from attention weights.

Table 3: Performance on MIMIC-III and CRADLE with different input subgraphs generated by factual and counterfactual reasoning.

Input	MIMIC-III				CRADLE			
	ACC	AUROC	AUPR	F1	ACC	AUROC	AUPR	F1
$\mathcal{G}$	80.41	83.91	73.33	47.28	80.77	73.34	46.40	63.92
$\mathcal{G}'$	77.55	80.42	67.74	35.58	80.24	70.67	43.14	56.61
$\mathcal{G}\backslash\mathcal{G}'$	70.81	66.20	47.81	12.77	32.40	49.99	21.49	32.37

4.6. Interpretability Evaluation

4.6.1. Qualitative Analysis

To justify the advantage of CACHE in generating an interpretable subset within each hyperedge that indicates causal relationships towards the prediction, we randomly select 30 samples in CRADLE, as well as their corresponding generated subsets. For comparison, we use the attention weight from AllSetTransformer to generate subsets for the same 30 samples as the baseline. Specifically, for both models, we rank the weights of all nodes $\omega_{e,v}$ in each hyperedge e and select the top 30% of the nodes as the interpretable subset, in order to force the two models to generate subsets of the same sizes. We put each pair of generated subsets and their corresponding CVD outcome together, and ask a (model-blinded) medical domain expert to select one of the subsets that can better explain the CVD condition. The result from the clinical expert shows that 21 subsets generated from CACHE are selected, which is 70% of the total 30 samples.

To further demonstrate the quality of CACHE's explanations, we present one case study as shown in Figure 4 (more cases are shown in Appendix D). It compares different elements that the two models select as most important from each visit. According to the analysis provided by the domain expert, the subset generated by our model suggests that the patient is engaged in getting preventive screenings like colonoscopy and urine albumin checked. However, from the perspective of the subset generated from the AllSetTransformer with attention weights, it indicates that the patient already has some neurological issue with diabetes, which suggests his/her control of diabetes is poor. Thus, the second subset has a stronger risk factor for a major adverse cardiovascular event. Since the patient did not experience CVD complications in the next year, the first subset generated by CACHE provides better interpretations into the CVD outcome.

4.6.2. Quantitative Analysis

We also provide the *quantitative analysis* to measure the quality of the generated subset. Table 3 provides the average results for the subset $\mathcal{G}'$ (factual reasoning) extracted by CACHE by using the top-30% node in the hyperedge with the highest weight, as well as the graph $\mathcal{G}\backslash\mathcal{G}'$ (counterfactual reasoning). From the results, it is clear that the performance of the factual graph $\mathcal{G}'$ is much better than that of $\mathcal{G}\backslash\mathcal{G}'$, and the performance with $\mathcal{G}'$ is close to the performance of learning with full hypergraphs. Such results justify the advantage of CACHE for find-

Table 4: Interpretability evaluation on MIMIC-III and CRADLE compared with two strong baselines.

Model	MIMIC-III		CRADLE	
	PoS (%)	PoN (%)	PoS (%)	PoN (%)
GNNExplainer	87.63	30.88	83.91	57.40
CF-GNNExplainer	82.37	**86.99**	79.76	89.30
CACHE	**91.62**	36.30	**94.38**	**93.72**

ing the informative subsets, and using the subset only yields comparable performance when compared with using all the nodes in each hyperedge for clinical predictions.

We further consider two additional evaluation metrics, namely *Probability of Sufficiency* (PoS) and *Probability of Necessity* (PoN) as explicit quantitative evaluations of explanations following causal inference theory (Glymour et al., 2016). Specifically, PoS is defined as the percentage of extracted subgraphs that can keep the GNN prediction unchanged to show the *sufficiency* of the explanations. PoN is defined as the percentage of extracted subgraphs that change the GNN prediction if removed, and thus it shows the *necessity* of the explanations.

From the results shown in Table 4, we observe that although CF-GNNExplainer achieves better performance on PoN for MIMIC-III dataset, it sacrifices the performance in terms of PoS. In contrast, CACHE balances between these two terms, and generally achieves good performance especially for CRADLE. We also remark that those post-hoc explanation methods focus on generating explanations only, without improving the performance of the model. Instead, CACHE jointly generates the explanations and uses the explanation as an effective regularizer for the predictive model, which is beneficial to the model performance.

5. Conclusion and Future Works

We develop CACHE, an accurate and interpretable framework for clinical predictions with EHR data. Specifically, we leverage hypergraphs to model the co-occurrence among medical codes and design multiset functions to encode the relations to facilitate *precise* clinical predictions. To produce *insightful* subsets for each visit, we harness counterfactual and factual reasoning techniques to ensure the sufficiency and necessity of the selected medical code. Experiments on two real EHR datasets verifies the superiority of CACHE, and the case study with a domain expert further justifies that CACHE can generate clinically meaningful subsets.

In this work, we mainly focus on providing *local* explanations, which aim to provide interpretability for each patient individually. Apart from the *local* explanations, another aspect of interpretability lies in *global* patterns, which aim to model confounding signals (Lengerich et al., 2022) as well as model biases (Zhang et al., 2020) for more accurate and bias-free clinical predictions. We believe finding global patterns is vital for overcoming the biases from the clinical prediction models and view it as an important future work.

Acknowledgments

We thank the anonymous reviewers for their feedbacks. This research was partially supported by the internal funds and GPU servers provided by the Computer Science Department of Emory University. YY and CZ were partly supported by NSF IIS-2008334, IIS-2106961, and CAREER IIS-2144338. MKA was partially supported by the Georgia Center for Diabetes Translation Research, funded by the National Institute of Diabetes Digestive and Kidney Disorders (P30DK111024). JCH was supported by NSF grants IIS-1838200, IIS-2145411 and NIH grant 5K01LM012924-03.

References

Song Bai, Feihu Zhang, and Philip HS Torr. Hypergraph convolution and hypergraph attention. *Pattern Recognition*, 110: 107637, 2021.

Derun Cai, Chenxi Sun, Moxian Song, Baofeng Zhang, Shenda Hong, and Hongyan Li. Hypergraph contrastive learning for electronic health records. In *Proceedings of the 2022 SIAM International Conference on Data Mining (SDM)*, pages 127–135. SIAM, 2022.

Paidamoyo Chapfuwa, Serge Assaad, Shuxi Zeng, Michael J Pencina, Lawrence Carin, and Ricardo Henao. Enabling counterfactual survival analysis with balanced representations. In *Proceedings of the Conference on Health, Inference, and Learning*, pages 133–145, 2021.

Eli Chien, Chao Pan, Jianhao Peng, and Olgica Milenkovic. You are allset: A multiset function framework for hypergraph neural networks. In *International Conference on Learning Representations*, 2022.

Edward Choi, Mohammad Taha Bahadori, Elizabeth Searles, Catherine Coffey, Michael Thompson, James Bost, Javier Tejedor-Sojo, and Jimeng Sun. Multi-layer representation learning for medical concepts. In *proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1495–1504, 2016a.

Edward Choi, Mohammad Taha Bahadori, Le Song, Walter F Stewart, and Jimeng Sun. Gram: graph-based attention model for healthcare representation learning. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 787–795, 2017.

Edward Choi, Zhen Xu, Yujia Li, Michael Dusenberry, Gerardo Flores, Emily Xue, and Andrew Dai. Learning the graphical structure of electronic health records with graph convolutional transformer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 606–613, 2020.

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

Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

Hejie Cui, Wei Dai, Yanqiao Zhu, Xiaoxiao Li, Lifang He, and Carl Yang. Interpretable graph neural networks for connectome-based brain disorder analysis. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 375–385. Springer, 2022a.

Hejie Cui, Jiaying Lu, Yao Ge, and Carl Yang. How can graph neural networks help document retrieval: A case study on cord19 with concept map generation. In *European Conference on Information Retrieval*, 2022b.

George Dasoulas, Kevin Scaman, and Aladin Virmaux. Lipschitz normalization for self-attention layers with application to graph neural networks. In *International Conference on Machine Learning*, pages 2456–2466. PMLR, 2021.

Thomas R Einarson, Annabel Acs, Craig Ludwig, and Ulrik H Panton. Prevalence of cardiovascular disease in type 2 diabetes: a systematic literature review of scientific evidence from across the world in

2007–2017. *Cardiovascular diabetology*, 17 (1):1–19, 2018.

Yifan Feng, Haoxuan You, Zizhao Zhang, Rongrong Ji, and Yue Gao. Hypergraph neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 3558–3565, 2019.

Alexander L Fogel and Joseph C Kvedar. Artificial intelligence powers digital medicine. *NPJ digital medicine*, 1(1):1–4, 2018.

T Fu, T Hoang, C Xiao, and J Sun. Ddl: Deep dictionary learning for predictive phenotyping. In *Proceedings of the IJCAI*, pages 5857–5863, 2019.

Madelyn Glymour, Judea Pearl, and Nicholas P Jewell. *Causal inference in statistics: A primer*. John Wiley & Sons, 2016.

Riccardo Guidotti, Anna Monreale, Fosca Giannotti, Dino Pedreschi, Salvatore Ruggieri, and Franco Turini. Factual and counterfactual explanations for black box decision making. *IEEE Intelligent Systems*, 34(6):14–23, 2019.

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

Sarthak Jain and Byron C Wallace. Attention is not explanation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3543–3556, 2019.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparametrization with gumble-softmax. In *International Conference on Learning Representations (ICLR 2017)*, 2017.

Alistair EW Johnson, Tom J Pollard, Lu Shen, Li-wei H Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3(1):1–9, 2016.

Xuan Kan, Wei Dai, Hejie Cui, Zilong Zhang, Ying Guo, and Carl Yang. Brain network transformer. *NeurIPS*, 2022.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017.

Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi, and Yee Whye Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. In *International conference on machine learning*, pages 3744–3753. PMLR, 2019.

Benjamin J Lengerich, Rich Caruana, Mark E Nunnally, and Manolis Kellis. Death by round numbers and sharp thresholds: How to avoid dangerous ai ehr recommendations. *medRxiv*, 2022.

Rui Li, Stephanie Hu, Mingyu Lu, Yuria Utsumi, Prithwish Chakraborty, Daby M Sow, Piyush Madan, Jun Li, Mohamed Ghalwash, Zach Shahn, et al. G-net: a recurrent network approach to g-computation for counterfactual prediction under a dynamic treatment regime. In *Machine Learning for Health*, pages 282–299. PMLR, 2021.

Zheng Liu, Xiaohan Li, Hao Peng, Lifang He, and S Yu Philip. Heterogeneous similarity graph neural network on electronic health records. In *2020 IEEE International Conference on Big Data (Big Data)*, pages 1196–1205. IEEE, 2020.

Ana Lucic, Maartje A Ter Hoeve, Gabriele Tolomei, Maarten De Rijke, and Fabrizio Silvestri. Cf-gnnexplainer: Counterfactual explanations for graph neural networks. In *International Conference on Artificial Intelligence and Statistics*, pages 4499–4511. PMLR, 2022.

Fenglong Ma, Radha Chitta, Jing Zhou, Quanzeng You, Tong Sun, and Jing Gao. Dipole: Diagnosis prediction in healthcare via attention-based bidirectional recurrent neural networks. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1903–1911, 2017.

Scott Menard. *Applied logistic regression analysis*. Number 106. Sage, 2002.

Diana Mincu, Eric Loreaux, Shaobo Hou, Sebastien Baur, Ivan Protsyuk, Martin Seneviratne, Anne Mottram, Nenad Tomasev, Alan Karthikesalingam, and Jessica Schrouff. Concept-based model explanations for electronic health records. In *Proceedings of the Conference on Health, Inference, and Learning*, pages 36–46, 2021.

Parisa Naraei, Abdolreza Abhari, and Alireza Sadeghian. Application of multilayer perceptron neural networks and support vector machines in classification of healthcare data. In *2016 Future Technologies Conference (FTC)*, pages 848–852. IEEE, 2016.

Juan G Diaz Ochoa and Faizan E Mustafa. Graph neural network modelling as a potentially effective method for predicting and analyzing procedures based on patients' diagnoses. *Artificial Intelligence in Medicine*, page 102359, 2022.

Anika Oellrich, Nigel Collier, Tudor Groza, Dietrich Rebholz-Schuhmann, Nigam Shah, Olivier Bodenreider, Mary Regina Boland, Ivo Georgiev, Hongfang Liu, Kevin Livingston, et al. The digital revolution in phenotyping. *Briefings in bioinformatics*, 17(5):819–830, 2016.

Kenta Oono and Taiji Suzuki. Graph neural networks exponentially lose expressive power for node classification. *arXiv preprint arXiv:1905.10947*, 2019.

Stephen R Pfohl, Tony Duan, Daisy Yi Ding, and Nigam H Shah. Counterfactual reasoning for fair clinical risk prediction. In *Machine Learning for Healthcare Conference*, pages 325–358. PMLR, 2019.

Sofia Serrano and Noah A. Smith. Is attention interpretable? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2931–2951, Florence, Italy, July 2019. Association for Computational Linguistics.

Juntao Tan, Shijie Geng, Zuohui Fu, Yingqiang Ge, Shuyuan Xu, Yunqi Li, and Yongfeng Zhang. Learning and evaluating graph neural network explanations based on counterfactual and factual reasoning. In *Proceedings of the ACM Web Conference 2022*, pages 1018–1027, 2022.

Sana Tonekaboni, Shalmali Joshi, Melissa D McCradden, and Anna Goldenberg. What clinicians want: contextualizing explainable machine learning for clinical end use. In *Machine learning for healthcare conference*, pages 359–380. PMLR, 2019.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò,

and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018.

Fei Wang, Peng Cui, Jian Pei, Yangqiu Song, and Chengxi Zang. Recent advances on graph analytics and its applications in healthcare. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 3545–3546, 2020.

Zifeng Wang and Jimeng Sun. Survtrace: transformers for survival analysis with competing events. In *Proceedings of the 13th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics*, pages 1–9, 2022.

Haowen Xu, Hao Zhang, Zhiting Hu, Xiaodan Liang, Ruslan Salakhutdinov, and Eric Xing. Autoloss: Learning discrete schedule for alternate optimization. In *International Conference on Learning Representations*, 2019.

Keyulu Xu, Chengtao Li, Yonglong Tian, Tomohiro Sonobe, Ken-ichi Kawarabayashi, and Stefanie Jegelka. Representation learning on graphs with jumping knowledge networks. In *International conference on machine learning*, pages 5453–5462. PMLR, 2018.

Naganand Yadati, Madhav Nimishakavi, Prateek Yadav, Vikram Nitin, Anand Louis, and Partha Talukdar. Hypergcn: A new method for training graph convolutional networks on hypergraphs. *Advances in neural information processing systems*, 32, 2019.

Chaoqi Yang, Cao Xiao, Fenglong Ma, Lucas Glass, and Jimeng Sun. Safedrug: Dual molecular graph encoders for recommending effective and safe drug combinations. In *IJCAI*, pages 3735–3741, 2021.

Zhitao Ying, Dylan Bourgeois, Jiaxuan You, Marinka Zitnik, and Jure Leskovec. Gnnexplainer: Generating explanations for graph neural networks. *Advances in neural information processing systems*, 32, 2019.

Yue Yu, Kexin Huang, Chao Zhang, Lucas M Glass, Jimeng Sun, and Cao Xiao. Sumgnn: Multi-typed drug interaction prediction via efficient knowledge graph summarization. *arXiv preprint arXiv:2010.01450*, 2020a.

Yue Yu, Tong Xia, Huandong Wang, Jie Feng, and Yong Li. Semantic-aware spatio-temporal app usage representation via graph convolutional network. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 4(3):1–24, 2020b.

Haoran Zhang, Amy X Lu, Mohamed Abdalla, Matthew McDermott, and Marzyeh Ghassemi. Hurtful words: quantifying biases in clinical contextual word embeddings. In *proceedings of the ACM Conference on Health, Inference, and Learning*, pages 110–120, 2020.

Lingxiao Zhao and Leman Akoglu. Pairnorm: Tackling oversmoothing in gnns. *arXiv preprint arXiv:1909.12223*, 2019.

Weicheng Zhu and Narges Razavian. Variationally regularized graph-based representation learning for electronic health records. In *Proceedings of the Conference on Health, Inference, and Learning*, pages 1–13, 2021.

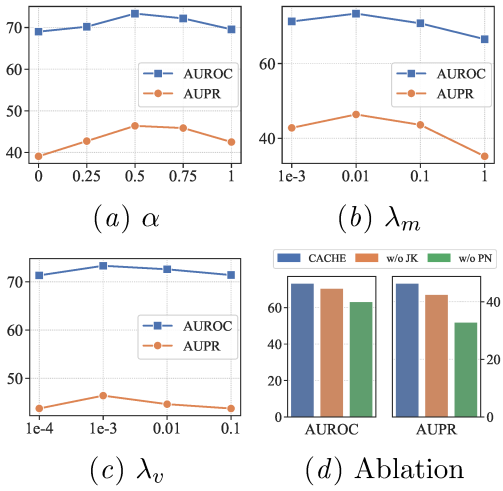
Appendix A. Dataset and Task Description

The descriptions and label distributions of the 25 phenotypes in MIMIC-III are shown in Table 5. The 25 phenotypes are identified using Clinical Classifications Software (CCS) from the Healthcare Cost and Utilization Project (HCUP)[5]. Among the 25 phenotypes, 12 are acute conditions such as respiratory failure and renal failure, 8 are chronic conditions such as kidney disease and hypertension, and the other 5 are mixed conditions since they are recurring acute diseases.

For CRADLE, we let patients to have positive labels (label 1), when they have a CVD complication within a year. For those positive patients, their input encounter is the earliest recorded encounter within a year of the presence of the CVD endpoint. Otherwise, for those with label 0, the input encounter is randomly selected from the encounters that are at least one year before the last recorded encounter. Records that are not sufficient for the modeling are removed, including the patients who only have one diagnosis record, or whose interval between the initial and last record is less than one year.

Appendix B. Additional Experimental Results

We list the results for per-task performances on 25 phenotypes in MIMIC-III in Table 6 for reference. We observe that CACHE outperforms the other two baseline models on 24 phenotypes out of the total 25. The phenomenon demonstrates that our gain is consistent over almost all the phenotyping tasks.



(a) α (b) λ_m (c) λ_v (d) Ablation

Figure 5: Effect of different components of CACHE on CRADLE. JK denotes Jumping Knowledge and PN denotes PairNorm.

Appendix C. Additional Parameter and Ablation Studies

Figures 5(a), 5(b) and 5(c) shows the effects of different parameters on CRADLE. We notice that the results agree with the observation in Sec. 4.5. The performance is the best when $\alpha = 0.5$, $\lambda_m = 0.01$ and λ_v =1e-3.

Figure 5(d) shows the contributions of Jumping Knowledge and PairNorm to the model performance. Results from Figure 6 again justifies their efficacy over different models. In addition, we observe that these techniques benefit CACHE more than the baselines. This is mainly because the self-attention based models, though with stronger expressive power, are even more susceptible to the oversmoothing issue, as has been discussed in the prior works (Dasoulas et al., 2021). When the oversmoothing issue has been mitigated via our selected strate-

5. https://www.hcup-us.ahrq.gov/toolssoftware/ccs/AppendixASingleDX.txt

Table 5: ICU phenotypes used in the benchmark dataset.

Phenotype	**Type**	**Positives' Ratio (%)**
Acute and unspecifed renal failure	acute	29.3
Acute cerebrovascular disease	acute	7.1
Acute myocardial infarction	acute	7.4
Cardiac dysrhythmias	mixed	43.2
Chronic kidney disease	chronic	25.5
Chronic obstructive pulmonary disease	chronic	18.0
Complications of surgical/medical care	acute	84.0
Conduction disorders	mixed	2.4
Congestive heart failure; nonhypertensive	mixed	39.2
Coronary atherosclerosis and related	chronic	29.6
Diabetes mellitus with complications	mixed	36.2
Diabetes mellitus without complication	chronic	42.1
Disorders of lipid metabolism	chronic	27.6
Essential hypertension	chronic	35.4
Fluid and electrolyte disorders	acute	44.4
Gastrointestinal hemorrhage	acute	27.8
Hypertension with complications	chronic	59.5
Other liver diseases	mixed	21.9
Other lower respiratory disease	acute	35.4
Other upper respiratory disease	acute	9.5
Pleurisy; pneumothorax; pulmonary collapse	acute	33.9
Pneumonia	acute	21.7
Respiratory failure; insufciency; arrest	acute	32.8
Septicemia (except in labor)	acute	26.6
Shock	acute	12.2

Table 6: Model performance on 25 phenotypes in MIMIC-III.

Phenotype	**Type**	CACHE		**AllSetTransformer**		**MLP**	
		AUROC	AUPR	AUROC	AUPR	AUROC	AUPR
Acute and unspecifed renal failure	acute	69.53	48.38	67.19	45.31	56.35	35.42
Acute cerebrovascular disease	acute	65.36	13.63	58.11	11.16	49.06	7.22
Acute myocardial infarction	acute	74.51	21.10	73.56	20.52	51.69	8.77
Cardiac dysrhythmias	mixed	77.92	75.03	76.61	72.57	55.88	49.89
Chronic kidney disease	chronic	87.26	78.33	86.42	74.33	61.11	32.99
Chronic obstructive pulmonary disease	chronic	84.69	60.81	82.16	56.15	55.57	20.52
Complications of surgical/medical care	acute	70.76	92.03	70.65	91.96	64.87	89.79
Conduction disorders	mixed	68.82	6.14	63.83	4.08	57.25	3.86
Congestive heart failure; nonhypertensive	mixed	85.07	79.55	82.52	74.95	54.62	44.16
Coronary atherosclerosis and related	chronic	84.02	71.97	82.31	69.56	56.81	39.00
Diabetes mellitus with complications	mixed	93.22	90.54	92.80	88.63	56.16	40.14
Diabetes mellitus without complication	chronic	87.32	87.37	86.80	85.48	56.75	46.57
Disorders of lipid metabolism	chronic	80.36	59.54	76.65	54.52	55.24	34.31
Essential hypertension	chronic	80.41	70.32	76.15	64.26	51.22	38.58
Fluid and electrolyte disorders	acute	68.28	62.00	64.12	59.25	60.39	53.22
Gastrointestinal hemorrhage	acute	64.65	42.90	65.18	43.51	53.42	29.08
Hypertension with complications	chronic	80.70	85.25	76.64	80.99	56.28	63.85
Other liver diseases	mixed	68.58	47.28	68.75	45.90	56.55	24.19
Other lower respiratory disease	acute	68.42	58.43	67.16	55.57	57.18	42.50
Other upper respiratory disease	acute	65.11	27.93	63.81	26.87	54.88	11.22
Pleurisy; pneumothorax; pulmonary collapse	acute	67.28	53.32	65.50	49.20	57.10	42.05
Pneumonia	acute	64.58	32.79	63.85	32.05	57.38	28.21
Respiratory failure; insufciency; arrest	acute	68.12	53.99	66.31	51.30	56.76	41.09
Septicemia (except in labor)	acute	67.45	43.12	64.26	37.85	60.36	35.38
Shock	acute	65.26	22.28	62.74	17.93	60.67	18.04

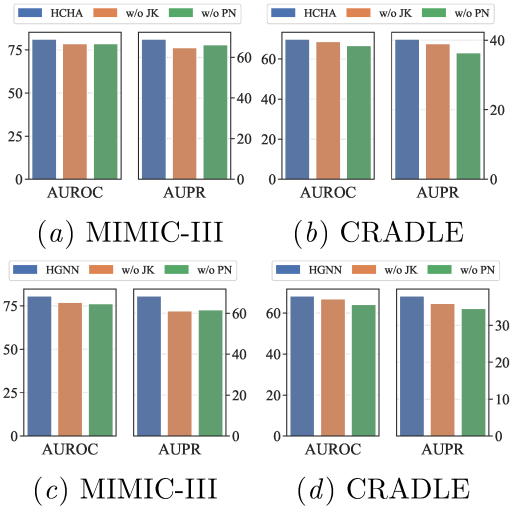


Figure 6: Effect of jumping knowledge and PairNorm on two representative baselines: HCHA and HGNN over MIMIC-III and CRADLE.

Subset 1 (generated from CACHE)	Office or other outpatient visit for the evaluation and management of an established patient, which requires a medically appropriate history and/or examination and moderate level of medical decision making. When using time for code selection, 30-39 minutes of total time is spent on the date of the encounter.
	Benign essential hypertension
	Hyperlipidemia
	Diabetic - poor control
	Needle electromyography, each extremity, with related paraspinal areas, when performed, done with nerve conduction, amplitude and latency/velocity study; complete, five or more muscles studied, innervated by three or more nerves or four or more spinal levels (List separately in addition to code for primary procedure)
Subset 2 (generated from attention weights of AllSetTransformer)	Insulins and analogues for injection, long-acting
	Benzodiazepine derivatives
	Other antidepressants
	Glucagon-like peptide-1 (GLP-1) analogues
	Combinations of oral blood glucose lowering drugs
Label	1 (The patient would experience CVD complications in the next year.)

Figure 7: Additional example I: A comparison of the subsets generated from CACHE and from attention weights of AllSetTransformer.

Subset 1 (generated from CACHE)	Type 2 diabetes mellitus without complication
	Requires influenza virus vaccination
	Sebaceous cyst of skin
	Diabetes mellitus without complication
	Disorder of skin of trunk
	Diabetic - poor control
	Chest pain
	Pure hypercholesterolemia
	Mixed hyperlipidemia
Subset 2 (generated from attention weights of AllSetTransformer)	Echocardiography, transthoracic, real-time with image documentation (2D), includes M-mode recording, when performed, complete, with spectral Doppler echocardiography, and with color flow Doppler echocardiography
	Aortic valve disorder
	Chest pain
	Type 1 diabetes mellitus
	Disorder due to type 1 diabetes mellitus
	Cervical somatic dysfunction
	Adjustment disorder with mixed anxiety and depressed mood
	Essential hypertension
	Gastroesophageal reflux disease
Label	0 (The patient would not experience CVD complications in the next year.)

Figure 8: Additional example II: A comparison of the subsets generated from CACHE and from attention weights of AllSetTransformer.

gies, the performance of such self-attention based multiset functions can lead to stronger empirical performance.

Appendix D. Additional Case Study

In Figure 7 to 9, we offer several additional examples of the subsets extracted from CACHE and attention-based explanations, as well as the domain expert's evaluations.

In Figure 7, from the analysis of the expert, patient 1 appears higher risk due to the need for needle electromyography and poor diabetes control. Such results indicates that the subset 1 generated by CACHE is more *insightful* for clinical predictions. In contrast, the elements generated in subset 2 cannot well indicate the CVD risk of the patient.

In Figure 8, the domain expert points out that the subset generated with attention (subset 2) contains Type 1 diabetes, aortic valve concern, and the hx of an echo. which

Subset 1 (generated from CACHE)	Hyperlipidemia
	Asthma
	Type 2 diabetes mellitus without complication
	Office or other outpatient visit for the evaluation and management of an established patient, which requires a medically appropriate history and/or examination and low level of medical decision making. When using time for code selection, 20-29 minutes of total time is spent on the date of the encounter.
	Third-generation cephalosporins
	Morbid obesity
	Chronic rhinitis
	Spirometry, including graphic record, total and timed vital capacity, expiratory flow rate measurement(s), with or without maximal voluntary ventilation
	Radiologic examination; toe(s), minimum of 2 views
	Insulins and analogues for injection, fast-acting
	Insomnia
Subset 2 (generated from attention weights of AllSetTransformer)	Chest x-ray
	Insulins and analogues for inhalation
	Long-term current use of insulin
	Platelet aggregation inhibitors excl. heparin
	Other agents for local oral treatment
	Salicylic acid and derivatives
	Other dermatologicals
	Insulins and analogues for injection, intermediate-acting
	Insulins and analogues for injection, fast-acting
	Insulins and analogues for injection, intermediate- or long-acting combined with fast-acting
	Dyspnea
Label	0 (The patient would not experience CVD complications in the next year.)

Figure 9: Additional example III: A comparison of the subsets generated from CACHE and from attention weights of AllSetTransformer.

is a sign for higher CVD risks and contradicts with the original label for the patient.

In Figure 9, the domain expert states that the subset generated with attention (subset 2) needs a complex insulin regimen and also uses aspirin and an anti-platelet medication, implying higher risk for atherosclerosis/CVD compared with subset 1.

The above examples provide more concrete examples to support the subset generated via CACHE can provide more important and insightful subsets for clinical prediction tasks.

Appendix E. Ethics Statement

This work has been evaluated by our IRB as Not Human Subject Research.