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

Ran Xu  
Department of Computer Science, Emory University, Atlanta, GA 30322

Yue Yu  
College of Computing, Georgia Institute of Technology, Atlanta, GA 30332

Chao Zhang  
College of Computing, Georgia Institute of Technology, Atlanta, GA 30332

Mohammed K Ali  
Rollins School of Public Health, Emory University, Atlanta, GA 30322

Joyce C Ho  
Department of Computer Science, Emory University, Atlanta, GA 30322

Carl Yang  
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, 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 hyperedges (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.

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](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 explanations, 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 $G$ 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 $G'$ and its complementary set $G \setminus G'$ are gen-
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 $C$, including diseases, medications, procedures and services. For each patient, the input of our method is the medical record $X$ containing a set of medical codes, where $X \subset C$.

Our problem is: given the clinical record $X$, we aim to (1) predict the clinical outcome $y$ of that patient; (2) generate a subset $e \subset X$ of the most important elements in $X$ 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 $G = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ and $\mathcal{E}$ stand for nodes and hyperedges, respectively.

○ Hypergraph Learning Module $f_\theta$. 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 $f_\theta(G)$ parameterized by $\theta$, 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).

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

\[
\mathbf{E}_e^{(l)} = f_{V \rightarrow E} \left( \mathcal{V}_{e, \mathbf{X}}^{(l-1)} \right),
\]
\[
\mathbf{X}_v^{(l)} = f_{E \rightarrow V} \left( \mathcal{E}_{v, \mathbf{E}}^{(l)} \right).
\]

Here \( \mathbf{E}_e \) and \( \mathbf{X}_v \) stand for the embeddings of hyperedge \( e \) and node \( v \), respectively. \( \mathcal{V}_{e, \mathbf{X}} \) is the hidden representations of node that contain the hyperedge \( e \), and \( \mathcal{E}_{v, \mathbf{E}} \) is the hidden representations of hyperedges that contain the node \( v \). To realize the two message passing function \( f_{V \rightarrow E} (\cdot) \) and \( f_{E \rightarrow V} (\cdot) \), 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,

\[
f_{V \rightarrow E} (\mathbf{S}) = f_{E \rightarrow V} (\mathbf{S}) = \text{Self-Att}(\mathbf{S}),
\]

where the mathematical formulation of self-attention is written as

\[
\text{Self-Att}(\mathbf{S}) = \text{LayerNorm}(\mathbf{Y} + \text{FFN}(\mathbf{Y})).
\]

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

\[
\mathbf{Y} = \text{LayerNorm}(\mathbf{S} + \text{MultiHead}(\mathbf{S})),
\]

where

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

\[
\text{SA}_i(\mathbf{S}) = \text{softmax} \left( \mathbf{W}^Q_i (\mathbf{S} \mathbf{W}^K_i)^\top \sqrt{\frac{1}{d/h}} \right) \mathbf{S} \mathbf{W}^V_i.
\]

In the above equations, \( \mathbf{W}^Q_i \in \mathbb{R}^{1 \times [d/h]} \), \( \mathbf{W}^K_i \in \mathbb{R}^{[d/h] \times [d/h]} \), \( \mathbf{W}^V_i \in \mathbb{R}^{[d/h] \times [d/h]} \), 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 \( \mathbf{E}^{(L)} \) and nodes as \( \mathbf{X}^{(L)} \). 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):

\[
\mathbf{E}_e^c = \mathbf{E}_e - \frac{1}{|\mathcal{E}|} \sum_{\mathbf{e}' \in \mathcal{E}} \mathbf{E}'_{e'}
\]
\[
\tilde{\mathbf{E}}_e = \frac{\mathbf{E}_e^c}{\sqrt{\frac{1}{|\mathcal{E}|} \sum_{\mathbf{e}' \in \mathcal{E}} \| \mathbf{E}'_{e'} \|^2_2}} = \sqrt{|\mathcal{E}|} \cdot \frac{\mathbf{E}_e^c}{\sqrt{\| \mathbf{E}_e \|^2_F}}.
\]

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 \( \tilde{\mathbf{E}}_e^{(l)} (1 \leq l \leq L) \) as

\[
\hat{y}_e = \sigma \left( \text{MLP}_{\text{CLS}} \left( \|_{l=1}^L \tilde{\mathbf{E}}_e^{(l)} \right) \right);
\]

where MLP_{\text{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. \textit{jump-}
ing knowledge (Xu et al., 2018)), we further resolve the oversmoothing issue.

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

\[ \ell_{\text{cls}} = -y \log(\hat{y}_e) - (1 - y) \log(1 - \hat{y}_e). \]  (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 \( f_\theta(\mathcal{G}\setminus\mathcal{G}') \) based on counterfactual reasoning.

To fulfill this purpose, a learnable interpretable subset extraction model \( g_\phi \) 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 \( g_\phi \) for parameterizing the probability weight \( \omega_{e,v} \), with the representation of \( e \) and \( v \) from \( f_\theta \) as

\[ \omega_{e,v} = \text{MLP}\left(\left[ E_e^{(l)}; X_v^{(l)} \right] \right). \]  (8)

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

\[ \tilde{p}_{e,v} = \sigma\left( (\log(\delta/(1 - \delta)) + \omega_{e,v})/\tau \right), \]  (9)

where \( \delta \sim \text{Uniform}(0,1) \) and \( \tau \) 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

\[ \hat{y}_e = f_\theta(\mathcal{G}'); \hat{y}_{\text{cf}} = f_\theta(\mathcal{G}\setminus\mathcal{G}'), \]  (10)

and the loss can be written as

\[ \ell_f = \begin{cases} [\gamma + \hat{y}_e - \hat{y}_f]_+, & \text{if } y_e = 1; \\ [\gamma + \hat{y}_f - \hat{y}_e]_+, & \text{else}. \end{cases} \]  (11)

and

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

where \( [x]_+ = \max(x, 0) \) and \( \gamma = 0.5 \) is the pre-defined threshold. In this way, we encourage \( g_\phi \) 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 \( \mathcal{G}\setminus\mathcal{G}' \). Besides, to force \( g_\phi \) to generate concise subsets, we add additional regularization on the weight \( \omega_{e,v} \). To sum up, the learning objective of \( g_\phi \) is expressed as

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

where \( \alpha \) and \( \lambda_v \) are hyperparameters.

### 3.4. Alternate Training of \( f_\theta \) and \( g_\phi \)

To incorporate the factual and counterfactual learning during the training of \( f_\theta \), we augment the learning loss with the factual and counterfactual loss as

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

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.
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.

<table>
<thead>
<tr>
<th>Stats</th>
<th>MIMIC-III</th>
<th>CRADLE</th>
</tr>
</thead>
<tbody>
<tr>
<td># of diagnosis</td>
<td>846</td>
<td>7915</td>
</tr>
<tr>
<td># of medication</td>
<td>4525</td>
<td>489</td>
</tr>
<tr>
<td># of procedure</td>
<td>2032</td>
<td>4321</td>
</tr>
<tr>
<td># of service</td>
<td>20</td>
<td>—</td>
</tr>
<tr>
<td># of hyperedges</td>
<td>36875/12353</td>
<td>36611</td>
</tr>
</tbody>
</table>

where $\ell_{\text{cls}}$ is defined in Eq. 7. Joint optimizing $f_{\theta}$ and $g_{\phi}$ 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 $f_{\theta}$ and $g_{\phi}$. We first train $f_{\theta}$ with Eq. 7 for 10 epochs as the warmup. After that, we train $g_{\phi}$ while fixing $f_{\theta}$ as

$$
\phi = \phi - \lambda_{\text{cls}} \nabla_{\phi} L_{\text{g}}.
$$

Then, with the generated $G'$ containing the important subsets, we train $f_{\theta}$ while fixing $g_{\phi}$ as

$$
\theta = \theta - \lambda_{\text{g}} \nabla_{\theta} L_{\text{cls}},
$$

where $\lambda_{\text{g}}$ and $\lambda_{\text{cls}}$ are learning rates. Finally, the generated medical code subset for hyperedge $e$ with $g_{\phi}$ 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 $\alpha$ and $\lambda_{\text{v}}$ in Eq. 13 and $\lambda_{\text{m}}$ in Eq. 14. We set $\alpha = 0.5$ to balance between factual and counterfactual reasoning. We study the effect of $\alpha$, $\lambda_{\text{m}}$ and $\lambda_{\text{v}}$ in Section 4.5. For our experiments, we set $\lambda_{\text{g}} = 0.01$, $\lambda_{\text{cls}} = 1e-3$, $\lambda_{\text{m}} = 0.01$, $\lambda_{\text{v}} = 5e-3$. \(^3\)https://pytorch.org/

265
\[ \lambda_v = 1e-3, \alpha = 0.5, d = 48, h = 4, \text{dropout to 0 and weight decay to 1e-3}. \] 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 use 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 (HGN) (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. 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 ($p < 0.05$).

<table>
<thead>
<tr>
<th>Model</th>
<th>ACC</th>
<th>AUROC</th>
<th>AUPR</th>
<th>F1</th>
<th>ACC</th>
<th>AUROC</th>
<th>AUPR</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>LR</td>
<td>68.66 ± 0.24</td>
<td>64.82 ± 0.25</td>
<td>45.63 ± 0.32</td>
<td>13.74 ± 0.40</td>
<td>76.22 ± 0.30</td>
<td>57.22 ± 0.28</td>
<td>23.59 ± 0.26</td>
<td>42.18 ± 0.36</td>
</tr>
<tr>
<td>SVM</td>
<td>72.02 ± 0.12</td>
<td>55.10 ± 0.14</td>
<td>34.29 ± 0.17</td>
<td>32.35 ± 0.21</td>
<td>68.57 ± 0.13</td>
<td>53.57 ± 0.11</td>
<td>23.50 ± 0.15</td>
<td>52.34 ± 0.22</td>
</tr>
<tr>
<td>MLP</td>
<td>70.74 ± 0.24</td>
<td>71.20 ± 0.22</td>
<td>52.14 ± 0.21</td>
<td>16.39 ± 0.30</td>
<td>79.02 ± 0.17</td>
<td>63.89 ± 0.18</td>
<td>31.28 ± 0.21</td>
<td>45.16 ± 0.26</td>
</tr>
<tr>
<td>GCT</td>
<td>76.58 ± 0.23</td>
<td>78.62 ± 0.21</td>
<td>63.99 ± 0.27</td>
<td>35.48 ± 0.34</td>
<td>77.26 ± 0.22</td>
<td>67.08 ± 0.19</td>
<td>35.90 ± 0.28</td>
<td>56.66 ± 0.25</td>
</tr>
<tr>
<td>GAT</td>
<td>76.75 ± 0.26</td>
<td>78.89 ± 0.12</td>
<td>66.22 ± 0.29</td>
<td>34.88 ± 0.33</td>
<td>77.82 ± 0.20</td>
<td>66.55 ± 0.27</td>
<td>36.06 ± 0.18</td>
<td>56.43 ± 0.26</td>
</tr>
<tr>
<td>HGCNN</td>
<td>77.58 ± 0.41</td>
<td>80.12 ± 0.30</td>
<td>68.25 ± 0.24</td>
<td>40.04 ± 0.35</td>
<td>78.77 ± 0.24</td>
<td>67.21 ± 0.25</td>
<td>37.93 ± 0.18</td>
<td>56.95 ± 0.23</td>
</tr>
<tr>
<td>HyperGCN</td>
<td>78.01 ± 0.23</td>
<td>80.34 ± 0.15</td>
<td>67.68 ± 0.16</td>
<td>39.29 ± 0.20</td>
<td>78.18 ± 0.11</td>
<td>67.83 ± 0.18</td>
<td>38.29 ± 0.19</td>
<td>60.24 ± 0.21</td>
</tr>
<tr>
<td>HCHN</td>
<td>78.07 ± 0.28</td>
<td>80.42 ± 0.17</td>
<td>68.56 ± 0.15</td>
<td>37.78 ± 0.22</td>
<td>78.60 ± 0.15</td>
<td>68.05 ± 0.17</td>
<td>39.23 ± 0.13</td>
<td>59.26 ± 0.21</td>
</tr>
<tr>
<td>AllSetTransformer</td>
<td>79.07 ± 0.31</td>
<td>82.19 ± 0.13</td>
<td>71.09 ± 0.17</td>
<td>41.51 ± 0.25</td>
<td>79.76 ± 0.18</td>
<td>70.07 ± 0.13</td>
<td>40.92 ± 0.12</td>
<td>61.21 ± 0.18</td>
</tr>
<tr>
<td>AllSetTransformer+CL</td>
<td>79.58 ± 0.45</td>
<td>82.56 ± 0.32</td>
<td>71.09 ± 0.27</td>
<td>39.33 ± 0.40</td>
<td>79.49 ± 0.21</td>
<td>69.60 ± 0.26</td>
<td>41.48 ± 0.25</td>
<td>56.65 ± 0.31</td>
</tr>
<tr>
<td>CACHE</td>
<td>80.41 ± 0.21*</td>
<td>83.91 ± 0.17*</td>
<td>73.33 ± 0.18*</td>
<td>47.28 ± 0.22*</td>
<td>80.77 ± 0.19*</td>
<td>73.34 ± 0.22*</td>
<td>46.40 ± 0.18*</td>
<td>63.92 ± 0.24*</td>
</tr>
</tbody>
</table>

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. $\alpha$ 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 consistent with the observations in (Tan et al., 2022). $\lambda_m$ in Eq. 14 is a weight between counterfactual and factual loss and model classification loss, and the performance achieves the best with $\lambda_m = 0.01$, indicating a certain ratio between the importance of the two losses. $\lambda_v$ in Eq. 13 controls the sparsity of remaining nodes. As $\lambda_v$ get larger, the performance first increases and then decreases. This is because when $\lambda_v$ is too small or too large, $g_\phi$ 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.
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 $G'$ (factual reasoning) extracted by CACHE by using the top-30% node in the hyperedge with the highest weight, as well as the graph $G \setminus G'$ (counterfactual reasoning). From the results, it is clear that the performance of the factual graph $G'$ is much better than that of $G \setminus G'$, and the performance with $G'$ 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.

<table>
<thead>
<tr>
<th>Model</th>
<th>MIMIC-III</th>
<th>CRADLE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PoS (%)</td>
<td>PoN (%)</td>
</tr>
<tr>
<td>GNNExplainer</td>
<td>87.63</td>
<td>30.88</td>
</tr>
<tr>
<td>CF-GNNExplainer</td>
<td>82.37</td>
<td><strong>86.99</strong></td>
</tr>
<tr>
<td>CACHE</td>
<td>91.62</td>
<td>36.30</td>
</tr>
</tbody>
</table>

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 verify 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


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


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.


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)\textsuperscript{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.

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 $\lambda_v = 1e^{-3}$.

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 strategy, the performance is better.

\textsuperscript{5} https://www.hcup-us.ahrq.gov/toolssoftware/ccs/AppendixASingleDX.txt
Table 5: ICU phenotypes used in the benchmark dataset.

<table>
<thead>
<tr>
<th>Phenotype</th>
<th>Type</th>
<th>Positives’ Ratio (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acute and unspecified renal failure</td>
<td>acute</td>
<td>29.3</td>
</tr>
<tr>
<td>Acute cerebrovascular disease</td>
<td>acute</td>
<td>7.1</td>
</tr>
<tr>
<td>Acute myocardial infarction</td>
<td>acute</td>
<td>7.4</td>
</tr>
<tr>
<td>Cardiac dysrhythmias</td>
<td>mixed</td>
<td>43.2</td>
</tr>
<tr>
<td>Chronic kidney disease</td>
<td>chronic</td>
<td>25.5</td>
</tr>
<tr>
<td>Chronic obstructive pulmonary disease</td>
<td>chronic</td>
<td>18.0</td>
</tr>
<tr>
<td>Complications of surgical/medical care</td>
<td>acute</td>
<td>84.0</td>
</tr>
<tr>
<td>Conduction disorders</td>
<td>mixed</td>
<td>2.4</td>
</tr>
<tr>
<td>Congestive heart failure; nonhypertensive</td>
<td>mixed</td>
<td>39.2</td>
</tr>
<tr>
<td>Coronary atherosclerosis and related</td>
<td>chronic</td>
<td>29.6</td>
</tr>
<tr>
<td>Diabetes mellitus with complications</td>
<td>mixed</td>
<td>36.2</td>
</tr>
<tr>
<td>Diabetes mellitus without complication</td>
<td>chronic</td>
<td>42.1</td>
</tr>
<tr>
<td>Disorders of lipid metabolism</td>
<td>chronic</td>
<td>27.6</td>
</tr>
<tr>
<td>Essential hypertension</td>
<td>chronic</td>
<td>35.4</td>
</tr>
<tr>
<td>Fluid and electrolyte disorders</td>
<td>acute</td>
<td>44.4</td>
</tr>
<tr>
<td>Gastrointestinal hemorrhage</td>
<td>acute</td>
<td>27.8</td>
</tr>
<tr>
<td>Hypertension with complications</td>
<td>chronic</td>
<td>59.5</td>
</tr>
<tr>
<td>Other liver diseases</td>
<td>mixed</td>
<td>21.9</td>
</tr>
<tr>
<td>Other lower respiratory disease</td>
<td>acute</td>
<td>35.4</td>
</tr>
<tr>
<td>Other upper respiratory disease</td>
<td>acute</td>
<td>9.5</td>
</tr>
<tr>
<td>Pleurisy; pneumothorax; pulmonary collapse</td>
<td>acute</td>
<td>33.9</td>
</tr>
<tr>
<td>Pneumonia</td>
<td>acute</td>
<td>21.7</td>
</tr>
<tr>
<td>Respiratory failure; insufficiency; arrest</td>
<td>acute</td>
<td>32.8</td>
</tr>
<tr>
<td>Septicemia (except in labor)</td>
<td>acute</td>
<td>26.6</td>
</tr>
<tr>
<td>Shock</td>
<td>acute</td>
<td>12.2</td>
</tr>
</tbody>
</table>
Table 6: Model performance on 25 phenotypes in MIMIC-III.

<table>
<thead>
<tr>
<th>Phenotype</th>
<th>Type</th>
<th>CACHE</th>
<th>AllSetTransformer</th>
<th>MLP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acute and unspecified renal failure</td>
<td>acute</td>
<td>69.53</td>
<td>48.38</td>
<td>67.19</td>
</tr>
<tr>
<td>Acute cerebrovascular disease</td>
<td>acute</td>
<td>65.36</td>
<td>13.63</td>
<td>58.11</td>
</tr>
<tr>
<td>Acute myocardial infarction</td>
<td>acute</td>
<td>74.51</td>
<td>21.10</td>
<td>73.56</td>
</tr>
<tr>
<td>Cardiac dysrhythmias</td>
<td>mixed</td>
<td>77.92</td>
<td>75.03</td>
<td>76.61</td>
</tr>
<tr>
<td>Chronic kidney disease</td>
<td>chronic</td>
<td>87.26</td>
<td>78.33</td>
<td>86.42</td>
</tr>
<tr>
<td>Chronic obstructive pulmonary disease</td>
<td>chronic</td>
<td>84.69</td>
<td>60.81</td>
<td>82.16</td>
</tr>
<tr>
<td>Complications of surgical/medical care</td>
<td>acute</td>
<td>70.76</td>
<td>92.03</td>
<td>70.65</td>
</tr>
<tr>
<td>Conduction disorders</td>
<td>mixed</td>
<td>68.82</td>
<td>6.14</td>
<td>63.83</td>
</tr>
<tr>
<td>Congestive heart failure; nonhypertensive</td>
<td>mixed</td>
<td>85.07</td>
<td>79.55</td>
<td>82.52</td>
</tr>
<tr>
<td>Coronary atherosclerosis and related</td>
<td>chronic</td>
<td>84.02</td>
<td>71.97</td>
<td>82.31</td>
</tr>
<tr>
<td>Diabetes mellitus with complications</td>
<td>mixed</td>
<td>93.22</td>
<td>90.54</td>
<td>92.80</td>
</tr>
<tr>
<td>Diabetes mellitus without complication</td>
<td>chronic</td>
<td>87.32</td>
<td>87.37</td>
<td>86.80</td>
</tr>
<tr>
<td>Disorders of lipid metabolism</td>
<td>chronic</td>
<td>80.36</td>
<td>59.54</td>
<td>76.65</td>
</tr>
<tr>
<td>Essential hypertension</td>
<td>chronic</td>
<td>80.41</td>
<td>70.32</td>
<td>76.15</td>
</tr>
<tr>
<td>Fluid and electrolyte disorders</td>
<td>acute</td>
<td>68.28</td>
<td>62.00</td>
<td>64.12</td>
</tr>
<tr>
<td>Gastrointestinal hemorrhage</td>
<td>acute</td>
<td>64.65</td>
<td>42.90</td>
<td>65.18</td>
</tr>
<tr>
<td>Hypertension with complications</td>
<td>chronic</td>
<td>80.70</td>
<td>85.25</td>
<td>76.64</td>
</tr>
<tr>
<td>Other liver diseases</td>
<td>mixed</td>
<td>68.58</td>
<td>47.28</td>
<td>68.75</td>
</tr>
<tr>
<td>Other lower respiratory disease</td>
<td>acute</td>
<td>68.42</td>
<td>58.43</td>
<td>67.16</td>
</tr>
<tr>
<td>Other upper respiratory disease</td>
<td>acute</td>
<td>65.11</td>
<td>27.93</td>
<td>63.81</td>
</tr>
<tr>
<td>Pleurisy; pneumothorax; pulmonary collapse</td>
<td>acute</td>
<td>67.28</td>
<td>53.32</td>
<td>65.50</td>
</tr>
<tr>
<td>Pneumonia</td>
<td>acute</td>
<td>64.58</td>
<td>32.79</td>
<td>63.85</td>
</tr>
<tr>
<td>Respiratory failure; insufficiency; arrest</td>
<td>acute</td>
<td>68.12</td>
<td>53.99</td>
<td>66.31</td>
</tr>
<tr>
<td>Septicemia (except in labor)</td>
<td>acute</td>
<td>67.45</td>
<td>43.12</td>
<td>64.26</td>
</tr>
<tr>
<td>Shock</td>
<td>acute</td>
<td>65.26</td>
<td>22.28</td>
<td>62.74</td>
</tr>
</tbody>
</table>
In Figure 6, we compare the performance of HCHA and HGNN over MIMIC-III and CRADLE.

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

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 indicate 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

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

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

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.

---

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