ProtoEHR: Hierarchical Prototype Learning for EHR-based Healthcare Predictions

Zi Cai*
University of Cambridge
Cambridge, United Kingdom
caizicharlesofficial@gmail.com

Yu Liu†
University of Oxford
Oxford, United Kingdom
yu.liu@eng.ox.ac.uk

Zhiyao Luo
University of Oxford
Oxford, United Kingdom
zhiyao.luo@eng.ox.ac.uk

Tingting Zhu
University of Oxford
Oxford, United Kingdom
tingting.zhu@eng.ox.ac.uk

Abstract

Digital healthcare systems have enabled the collection of mass healthcare data in electronic healthcare records (EHRs), allowing artificial intelligence solutions for various healthcare prediction tasks. However, existing studies often focus on isolated components of EHR data, limiting their predictive performance and interpretability. To address this gap, we propose ProtoEHR, an interpretable hierarchical prototype learning framework that fully exploits the rich, multi-level structure of EHR data to enhance healthcare predictions. More specifically, ProtoEHR models relationships within and across three hierarchical levels of EHRs: medical codes, hospital visits, and patients. We first leverage large language models to extract semantic relationships among medical codes and construct a medical knowledge graph as the knowledge source. Building on this, we design a hierarchical representation learning framework that captures contextualized representations across three levels, while incorporating prototype information within each level to capture intrinsic similarities and improve generalization. To perform a comprehensive assessment, we evaluate ProtoEHR in two public datasets on five clinically significant tasks, including prediction of mortality, prediction of readmission, prediction of length of stay, drug recommendation, and prediction of phenotype. The results demonstrate the ability of ProtoEHR to make accurate, robust, and interpretable predictions compared to baselines in the literature. Furthermore, ProtoEHR offers interpretable insights on code, visit, and patient levels to aid in healthcare prediction.

CCS Concepts

• **Applied computing → Health informatics**; • **Information systems → Data mining**.

Keywords

AI for Healthcare; Healthcare Prediction; Hierarchical Representation Learning; Prototype Learning

ACM Reference Format:
Zi Cai, Yu Liu, Zhiyao Luo, and Tingting Zhu. 2025. ProtoEHR: Hierarchical Prototype Learning for EHR-based Healthcare Predictions. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management (CIKM '25), November 10–14, 2025, Seoul, Republic of Korea.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3746252.3761381

1 Introduction

The digitization of healthcare systems in recent years has led to the accumulation of substantial electronic health records (EHRs) [25, 28]. The patient's EHR data contain demographics, as well as detailed information about the hospital visit, including medical procedures performed, clinical diagnoses, and prescribed medications. Numerous studies [12, 30] have focused on developing models to facilitate the prediction of patient outcomes, such as mortality risk [7] and readmission possibilities [31], as well as personalized treatment strategies, including classification of phenotypes [11] and drug recommendation [27]. Accomplishing these tasks assists clinical decision-making, improves treatment quality, and allows better allocation of resources.

Existing frameworks for healthcare prediction capitalize on the various characteristics of EHR data [25]. Each patient can have multiple hospital visits at different times. Each hospital visit is for specific reasons, and with doctors reaching varying diagnoses, ordering the required procedures, and prescribing the corresponding medications (typically recorded using a standardized coding system, for example, ICD codes[1]). Therefore, EHR data naturally form a *hierarchical structure* that spans the levels of the patient, the visit, and the code [6, 7]. This structure provides a rich and interpretable representation of a patient's clinical journey, enabling models to capture both temporal and semantic dependencies across levels for more effective healthcare prediction.

Beyond hierarchical connections across different levels of EHR data, it is equally important to consider the *intrinsic similarities* among entities within each level. At the patient level, individuals may share lifestyle habits that influence their risk factors, increasing the propensity towards certain illnesses, e.g., smoking and poor diet increase the likelihood of coronary heart disease [16]. At the visit

*Work done while Zi Cai was a research intern at The University of Oxford.
†Yu Liu is the corresponding author.

[1] https://www.who.int/standards/classifications/classification-of-diseases

CIKM '25, Seoul, Republic of Korea
© 2025 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 979-8-4007-2040-6/2025/11
https://doi.org/10.1145/3746252.3761381

level, patients may visit hospitals for similar purposes. It could be a routine visit to pick up prescriptions or it could be an emergency. Thus, we argue that visits also exhibit similarities with predictive value. At the code level, medical codes are typically categorized into diagnoses, procedures, and medications [15], suggesting that similarity can be inferred within each category. In addition, codes can also be examined from a pathological or toxicological perspective, further enriching their contextual interpretation. While prior works have leveraged patient-level similarities through prototype learning [36, 38], intrinsic similarities within the visit and code levels remain largely underexplored. We argue that modeling both within-level similarities and cross-level hierarchies offers significant potential to improve healthcare predictions by fully capturing the complex semantics in EHR data.

To effectively capture both the hierarchical structure and the intrinsic similarities within EHR data, we propose ProtoEHR, a hierarchical prototype learning framework for healthcare predictions. We first apply large language models (LLMs) to construct a medical knowledge graph (KG) that captures the rich semantic relationships among medical codes, thereby enhancing the code-level modeling. Subsequently, building on the medical KG, we develop three specialized local encoders to model information at the code, visit, and patient levels, in alignment with the natural structure of EHR data. At each level, we introduce a novel prototype-based encoder to capture intrinsic similarities among objects and enhance the quality of representation through shared information. The patient-level representation, together with the learned prototypes from all three levels, is then fed into a customized hierarchical fusion module that integrates information across levels. The output of this module is finally passed through a linear projection layer to generate task-specific predictions. This architecture not only enhances predictive performance across clinical tasks, but also improves interpretability by highlighting the relative contributions of code-, visit-, and patient-level information to corresponding tasks. In summary, our contributions are as follows:

- We propose ProtoEHR, a novel hierarchical prototype learning framework that integrates prototype learning and hierarchical representation learning to model both within-level similarities and cross-level hierarchies in EHR data.
- We develop an interpretable mechanism based on hierarchical prototypes and conduct both qualitative and quantitative analyses to provide valuable insights into how different levels of the hierarchy contribute to clinically meaningful outcomes.
- We conduct extensive experiments on two real-world datasets across five prediction tasks, where ProtoEHR consistently achieves strong performance, outperforming state-of-the-art baselines.

2 Preliminaries

Here we introduce the mathematical definition of the EHR dataset and medical KG, as well as the research problem.

Definition 1 (EHR Dataset). An EHR dataset $\mathcal{D} = (C, \mathcal{V})$ is composed of a set of medical codes $C = \{c_i\}_{i=1}^{N_{\text{code}}}$ and hospital visit information $\mathcal{V} = \{\mathcal{V}_i\}_{i=1}^{N}$, where N_{code} and N are the total numbers of unique medical codes and patients in the dataset, respectively. For each patient i, $\mathcal{V}_i$ is an ordered sequence of hospital visits, defined as $\mathcal{V}_i = (\mathcal{V}_{i,1}, \cdots, \mathcal{V}_{i,|\mathcal{V}_i|})$, where $|\mathcal{V}_i|$ denotes the number of visits. Each visit $\mathcal{V}_{i,j}$ contains multiple medical codes recorded during that visit, denoted as $\mathcal{V}_{i,j} = \{c_{i,j,k} \in C\}_{k=1}^{|\mathcal{V}_{i,j}|}$, where $|\mathcal{V}_{i,j}|$ denotes the number of medical codes associated with the visit, and $c_{i,j,k}$ denotes the k-th medical code recorded in the j-th visit of patient i.

Definition 2 (Medical KG). A medical KG is defined as $\mathcal{G} = (C, \mathcal{R}, \mathcal{F})$, where C and $\mathcal{R}$ are the sets of entities and relations, respectively. The fact set is given by $\mathcal{F} = \{(c_h, r, c_t) | c_h, c_t \in C, r \in \mathcal{R}\}$, where each triplet (c_h, r, c_t) indicates that a medical code c_h is related to another one c_t through relation r.

Problem 1 (EHR-based Healthcare Prediction). Given an EHR dataset $\mathcal{D} = (C, \mathcal{V})$, the objective of EHR-based healthcare prediction is to learn a patient-specific representation and predict the corresponding clinical outcome. For each patient i with historical records $\mathcal{V}_i$, the prediction is performed using a function/model f, formulated as $\hat{y}_i = f(\mathcal{D}, i, \mathcal{V}_i)$, where $\hat{y}_i$ represents the predicted outcome. The nature of $\hat{y}_i$ varies depending on the specific prediction task, e.g., $\hat{y} \in \{0, 1\}$ for mortality prediction, and $\hat{y} \in \{1, \cdots, K\}$ for length-of-stay prediction with K discrete classes.

3 Proposed Method

Figure 1 presents an overview of our proposed ProtoEHR framework. It consists of three main stages: building the medical KG to exploit the relationships between medical codes in the EHR data; learning hierarchical representations and incorporating shared information with prototypes; and fusing the extracted patient representations with the learnt hierarchical prototypes for the prediction of healthcare outcomes. The structure and details of each of these stages are discussed below.

3.1 Medical Knowledge Graph Construction

To take advantage of the rich medical knowledge stored within the relationships between the medical codes, we constructed a medical KG with the diagnosis, procedure, and prescription codes in the dataset, as shown in Figure 1a. Since the numerous meaningful relationships therein are overwhelming for human experts to identify, we leverage powerful LLMs for automatic construction [12, 37].

Existing biomedical KGs, such as the UMLS-KG [3], could be used to incorporate medical knowledge but there is a possible mismatch between its ontologies and the medical code set C, thus requiring further processing. Closed-source LLMs like GPT-4 [1] demonstrate strong capabilities in automatically retrieving triplets for KG construction but are prohibitively expensive when dealing with large entity sets, such as the medical code set in our case. In contrast, open-source LLMs, such as the Llama series [8], offer a more cost-effective alternative but have weaker retrieval capabilities. To balance quality and efficiency, we employ an open-source LLM to retrieve triplets, a closed-source LLM to train a classifier for filtering out false triplets, and a clustering-based approach to refine the relations. This hybrid strategy ensures that the constructed medical KG remains both expressive and robust. The three steps for KG construction are detailed as follows:

Retrieval. To begin with, we use an open-source LLM (Llama3-70B [8]) denoted as LLM_1 to discover the semantic relationships within the medical codes. Taking each pair of the medical codes in the medical code set C as input, LLM_1 identifies the set of plausible

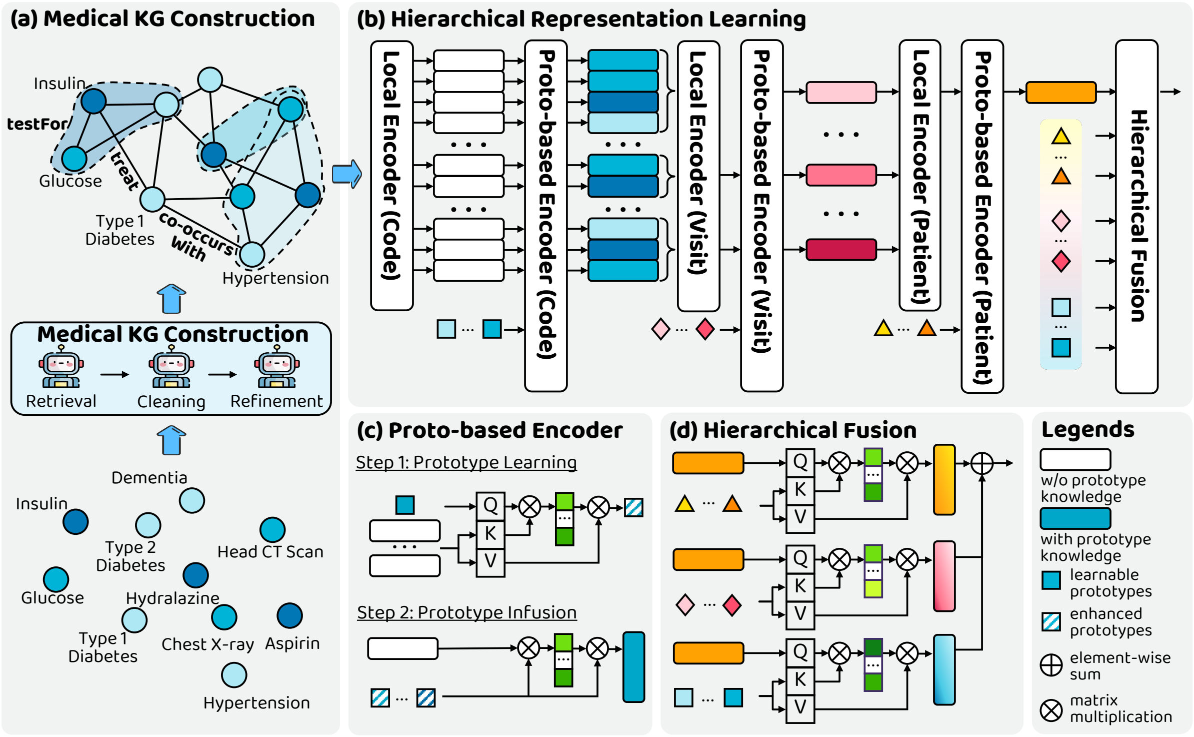


Figure 1: The architecture of our proposed ProtoEHR framework. (a) A set of medical codes in EHR data is used to construct a medical KG with LLMs via three stages: retrieval, cleaning, and refinement. (b) Taking the constructed medical KG as well as patient information as input, hierarchical representations at the code, visit, and patient levels are learned. The local encoders take the representation of the previous level and learns a prototype-free representation. This is then used as input to the proto-based encoder to infuse prototype knowledge into the representation. (c) Prototype-based encoder consists of prototype learning and prototype infusion, where enhanced prototypes are first obtained and the representation is updated with these enhanced prototypes. (d) Hierarchical fusion aggregates the learned patient representation with hierarchical prototype knowledge. "Proto-based" is the abbreviation for "prototype-based".

triplets, $\mathcal{F}_1$. Denoting the pair of codes as $\texttt{ENTITY}_1$ and $\texttt{ENTITY}_2$, the extraction process can be expressed as:

$$\text{Prompt}_1(\texttt{ENTITY}_1, \texttt{ENTITY}_2) \xrightarrow{LLM_1} (\texttt{ENTITY}_1, \underline{\texttt{Relation}}, \texttt{ENTITY}_2)$$

Cleaning. To improve the quality of the retrieved triplets in $\mathcal{F}_1$, we further employ a closed-source LLM (GPT-4 [1]) via the OpenAI API, denoted as LLM_2. This model curates a subset of triplets, $\mathcal{F}_1' \subset \mathcal{F}_1$, by assessing their validity. While $\mathcal{F}_1$ is designed to be comprehensive, ensuring that no meaningful relations are overlooked, $\mathcal{F}_1'$ prioritizes accuracy, filtering out false or misleading relations to maintain the integrity of the constructed KG. The subset curation process can be expressed as:

$$\text{Prompt}_2(\texttt{ENTITY}_1, \underline{\texttt{Relation}}, \texttt{ENTITY}_2) \xrightarrow{LLM_2} \texttt{True/False}$$

We take the triplet subset $\mathcal{F}_1'$ and the associated labels from LLM_2 to train a classifier f_1 to clean the remaining triplets $\mathcal{F}_1 \setminus \mathcal{F}_1'$. This identifies the plausible triplets in a time- and cost-effective manner, producing the cleaned triplet set $\mathcal{F}_2$.

Refinement. The triplets retrieved and filtered through LLMs often contain lexically similar but distinct relations, such as `is treated with` and `is treated using`, which should be unified for effective representation learning in the medical KG. Following [12], we first extract word embeddings for all relations using a pre-trained language model (BERT [17]) and apply agglomerative clustering [24] to group lexically similar relations. However, lexical similarity does not always equate to semantic similarity—relations with opposing meanings may differ by only a few words, e.g., `be not typically associated with` and `be often associated with`. To address this, we leverage an open-source LLM to detect and refine clusters containing semantically contradictory relations. Finally, we obtain the high-quality triplet set $\mathcal{F}$ for medical KG.

The three-step process ensures the construction of a more expressive and robust medical KG, which serves as the foundation for subsequent model stages. Further details on the reliability of the generated KG are provided in Appendix B.

3.2 Hierarchical Representation Learning

Building on the medical KG, we design a hierarchical representation learning process, which can be seen in Figure 1b. To obtain **code-level representations**, we use the medical KG $\mathcal{G} = (\mathcal{C}, \mathcal{R}, \mathcal{F})$ along with patient data $\mathcal{V}_i$ as input to the local encoder at the code level. Specifically, we employ a multi-relational graph convolutional network, CompGCN [29], to iteratively update the attributes of entities and relations with global medical knowledge. The representations are updated as follows:

$$\boldsymbol{c}_v^l = \sigma\left(\sum_{(u,r_i)\in\mathcal{N}_v} \boldsymbol{W}_{\text{ent}}^l \phi(\boldsymbol{c}_u^{l-1}, \boldsymbol{r}_i^{l-1})\right), \quad \boldsymbol{r}_i^l = \boldsymbol{W}_{\text{rel}}^l \boldsymbol{r}_i^{l-1}, \tag{1}$$

where $\boldsymbol{c}_v^l$ and $\boldsymbol{r}_i^l$ denote the layer-l representations of entity v and relation i, respectively. $\boldsymbol{W}_{\text{ent}}^l$ and $\boldsymbol{W}_{\text{rel}}^l$ are the learnable weight matrices for updating entities and relations at layer l, while σ denotes an activation function. The set $\mathcal{N}_v$ consists of the neighbors of entity v connected via the associated relations r_i. The function ϕ represents the circular correlation.

After global message passing, the updated representations of medical codes are processed by the code-level prototype-based encoder, $PEnc^c(\cdot)$, which incorporates shared information across codes through prototype learning. The details of this prototype-based encoder are provided in Section 3.3. Following this step, we obtain the updated code representation, denoted as $\boldsymbol{c}_{i,j,k} \in \mathbb{R}^d$, where d is the embedding dimension, for the k-th medical code recorded in the j-th visit of the i-th patient.

The **visit-level representations** are obtained by passing the code representations into the visit-level local encoder. Here we simply use average pooling, and the representation for j-th visit of i-th patient is calculated based on the representations of medical codes recorded in this visit:

$$\boldsymbol{v}'_{i,j} = \frac{1}{|\mathcal{V}_{i,j}|}\sum_{k=1}^{|\mathcal{V}_{i,j}|} \boldsymbol{c}_{i,j,k}. \tag{2}$$

Similarly, $\boldsymbol{v}'_{i,j}$ is fed into the visit-level prototype-based encoder $PEnc^v(\cdot)$ with the updated visit representation $\boldsymbol{v}_{i,j}$ obtained.

To obtain the **patient-level representations**, we account for the temporal sequence of multiple visits and employ a Transformer-based encoder to capture interactions among them. Given that recent visits have a stronger impact on future healthcare predictions, we use the representation of the last visit after the Transformer encoder as the patient-level representation $\boldsymbol{p}'_i$. This representation is further enhanced through the patient-level prototype-based encoder, $PEnc^p(\cdot)$. This enables us to obtain the updated patient representation $\boldsymbol{p}_i$.

3.3 Prototype-based Encoder

We now provide a detailed explanation of the prototype-based encoder $PEnc(\cdot)$, which is applied at all three levels of the hierarchy. This module (Figure 1c) consists of two key steps: prototype learning and prototype infusion.

Prototype Learning. The first step takes the representations of objects (codes, visits, or patients) within the same level and learnable prototypes as input. For each level of the hierarchy, the prototypes are randomly initialized and the number of prototypes is a hyperparameter. Without loss of generality, let the input representations be denoted as $\boldsymbol{X}' \in \mathbb{R}^{n\times d}$ and the learnable prototypes as $\boldsymbol{H} \in \mathbb{R}^{m\times d}$, where n is the number of objects, m is the number of prototypes, and d is the embedding dimension. To enable prototypes to absorb intrinsic similarities from the objects, we employ a cross-attention mechanism, formulated as the equation below:

$$\hat{\boldsymbol{H}} = \text{Softmax}(\frac{(\boldsymbol{H}\boldsymbol{W}_Q)(\boldsymbol{X}'\boldsymbol{W}_K)^\top}{\sqrt{d}})\boldsymbol{X}'\boldsymbol{W}_V, \tag{3}$$

where $\boldsymbol{W}_Q$, $\boldsymbol{W}_K$, and $\boldsymbol{W}_V$ are learnable parameters. This mechanism enhances the prototypes by integrating information from the objects at the corresponding level, enabling a more structured and meaningful representation.

Prototype Infusion. With the enhanced prototypes obtained in the previous step, the second step infuses the intrinsic similarities captured by the prototypes into the object representations. For each input object representation $\boldsymbol{x}'_i$, we apply the similarity-weighted summation idea for infusion [19], formulated as follows:

$$\boldsymbol{x}_i = \boldsymbol{x}'_i + \frac{1}{m}\sum_{j=1}^{m} \alpha_{i,j}\boldsymbol{W}_I\hat{\boldsymbol{h}}_j, \text{ with } \alpha_{i,j} = \text{Softmax}(\frac{\boldsymbol{x}_i'^\top \hat{\boldsymbol{h}}_j}{\|\boldsymbol{x}'_i\|\|\hat{\boldsymbol{h}}_j\|}), \tag{4}$$

where $\boldsymbol{W}_I$ represents learnable parameters, and $\alpha_{i,j}$ denotes the normalized similarity between object $\boldsymbol{x}_i$ and enhanced prototype $\hat{\boldsymbol{h}}_j$. The resulting prototype-infused object representations, $\boldsymbol{X}$, are subsequently used to compute representations at the next level in the code-visit-patient hierarchy.

3.4 Hierarchical Fusion

We now detail the hierarchical fusion module used to generate the final patient representation. The structure of this module is outlined in Figure 1d. Specifically, for each patient i, the patient-level representation $\boldsymbol{p}_i$ is fused with prototypes from all three hierarchical levels—code, visit, and patient—denoted as $\boldsymbol{H}^c$, $\boldsymbol{H}^v$, and $\boldsymbol{H}^p$, respectively. This fusion is achieved using a cross-attention mechanism, formulated as follows:

$$\boldsymbol{p}_i^t = \text{Softmax}(\frac{(\boldsymbol{p}_i\boldsymbol{W}_Q^t)(\boldsymbol{H}^t\boldsymbol{W}_K^t)^\top}{\sqrt{d}})\boldsymbol{H}^t\boldsymbol{W}_V^t, \; t \in \{c, v, p\} \tag{5}$$

The resulting level-specific representations $\{\boldsymbol{p}_i^t | t \in \{c, v, p\}\}$ are then combined to form the final patient representation:

$$\boldsymbol{p}_i^{\text{final}} = \sum_{t\in\{c,v,p\}} \beta_t \boldsymbol{p}_i^t, \text{ with } \beta_t = \frac{\exp(\boldsymbol{p}_i^{t\top}\boldsymbol{w}_F/\tau)}{\sum_{t\in\{c,v,p\}}\exp(\boldsymbol{p}_i^{t\top}\boldsymbol{w}_F/\tau)}. \tag{6}$$

Here, τ is the softmax temperature, and β_t represents the contribution weight of each level, computed based on the normalized similarity between the learnable vector $\boldsymbol{w}_F \in \mathbb{R}^d$ and $\boldsymbol{p}_i^t$, thereby determining the relative importance of information from each hierarchical level to the prediction, enabling a more interpretable and effective modeling approach.

For healthcare prediction, the final patient representation is passed through a linear projection layer to generate the predicted outcome, i.e., $\hat{y}_i = \text{Linear}(\boldsymbol{p}_i^{\text{final}})$. The whole framework is trained with a task-specific loss, which depends on the prediction task [12]: binary cross-entropy loss is used for binary classification and multi-label classification tasks, while cross-entropy loss is applied for multi-class classification tasks.

Table 1: Basic information about MIMIC-III and MIMIC-IV after preprocessing. c, v, and p are the abbreviations for code, visit, and patient respectively. #v/p and #c/p separately denotes the average number of visits per patient and the average number of codes per visit.

Dataset	#Patients	#Visits	#Codes	#v/p	#c/v
MIMIC-III	5,453	14,330	657	2.63	39.7
MIMIC-IV	51,473	167,042	708	3.25	21.6

4 Experiments

4.1 Experimental Setup

Datasets. We use two real-world medical datasets, MIMIC-III [15] and MIMIC-IV [14] for experiments. The basic statistics of the datasets are presented in Table 1. For MIMIC-III, a sliding window is employed to augment the sample size for all tasks other than mortality prediction. We split both datasets into train/validation/test sets with a ratio of 6:2:2.

Tasks and Metrics. Five tasks were used for evaluation:

- **Mortality Prediction.** An imbalanced binary classification task with label $y_i \in \{0, 1\}$ where $y_i = 1$ indicates mortality. For a patient with $|\mathcal{V}_i|$ visits, the model takes data $(\mathcal{V}_{i,1}, \ldots, \mathcal{V}_{i,|\mathcal{V}_i|-1})$ to predict if the patient is deceased in the visit $\mathcal{V}_{i,|\mathcal{V}_i|}$ and if it occurs within 30 days from the visit $\mathcal{V}_{i,|\mathcal{V}_i|-1}$.
- **Readmission Prediction.** A binary classification task with label $y_i \in \{0, 1\}$ where $y_i = 1$ indicates readmission. For a patient with $|\mathcal{V}_i|$ visits, the model takes data $(\mathcal{V}_{i,1}, \ldots, \mathcal{V}_{i,|\mathcal{V}_i|-1})$ to predict if visit $\mathcal{V}_{i,|\mathcal{V}_i|}$ occurs within 30 days from the visit $\mathcal{V}_{i,|\mathcal{V}_i|-1}$.
- **Length-of-stay Prediction** A multi-class classification task with 10 labels indicating the duration of the patient's hospital visit $y_i \in \{0, 1, \ldots, 9\}$. For a patient with $|\mathcal{V}_i|$ visits, the model takes data $(\mathcal{V}_{i,1}, \ldots, \mathcal{V}_{i,|\mathcal{V}_i|})$ to predict the duration of the visit $\mathcal{V}_{i,|\mathcal{V}_i|}$. The label $y_i = 0$ represents discharge within one day, the labels $y_i \in \{1, \ldots, 7\}$ indicate that the duration of stay is $\{y_i \leq t < y_i + 1\}_{y_i=1}^{7}$ days, the label $y_i = 8$ means the patient's stay is $7 < t \leq 14$ days, and the label $y_i = 9$ indicates that the stay is longer than 14 days.
- **Drug Recommendation** A multi-label classification task where the model takes data $(\mathcal{V}_{i,1}, \ldots, \mathcal{V}_{i,|\mathcal{V}_i|-1}) \cup \mathcal{V}'_{i,|\mathcal{V}_i|}$ to predict the prescriptions on visit $\mathcal{V}_{i,|\mathcal{V}_i|}$. Notation $\mathcal{V}'_{i,|\mathcal{V}_i|}$ denotes the data of the last visit excluding prescription codes. The output is a multi-hot vector $\hat{\boldsymbol{y}}_i \in \mathbb{R}^{202}$ where the set of prescriptions contain 201 elements, the last dimension is used to indicate no prescriptions on the last visit.
- **Phenotype Prediction** A multi-label classification tasks where the model takes data $(\mathcal{V}_{i,1}, \ldots, \mathcal{V}_{i,|\mathcal{V}_i|-1}) \cup \mathcal{V}''_{i,|\mathcal{V}_i|}$ to predict the phenotypes on visit $\mathcal{V}_{i,|\mathcal{V}_i|}$. Notation $\mathcal{V}''_{i,|\mathcal{V}_i|}$ denotes the data of the last visit excluding diagnosis codes. The output is a multi-hot vector $\hat{\boldsymbol{y}}_i \in \mathbb{R}^{26}$ where the set of phenotypes contains 25 diagnoses [11], and the last dimension is used to indicate no phenotype is present on the last visit.

Three different metrics were used across tasks for performance evaluation. The area under the receiver operating characteristic curve (**AUROC**) quantifies the model's ability to differentiate between classes; the area under the precision-recall curve (**AUPRC**) is similar to AUROC but emphasizes measuring performance on separating imbalanced classes; the **F1** score is a balance between precision and recall, capturing the importance of false positives and false negatives. For these metrics, we report both the performance mean and the standard deviation (std) of bootstrapping 100 times.

Baselines. We compare the performance for different tasks against several representative baselines: Deepr [22], AdaCare [21], GRASP [36], StageNet [9], GraphCare [12], and KerPrint [33]. For each baseline, we perform hyperparameter tuning based on the recommended search ranges specified in their original papers to ensure a fair and competitive comparison. For methods that incorporate a KG in their modeling, we use the same medical KG constructed in our framework to ensure consistency across evaluations.

Implementation. In our implementation, the Adam gradient optimizer and ExponentialLR scheduler are used. To optimize model performance, hyperparameter search is performed using grid search. It should also be noted that this can be implemented more efficiently with Bayesian-based search methods such as Optuna [2]. The parameters considered include CompGCN layers {1,2,3,4}, transformer encoder depth {1,2,4}, number of code-level prototypes {32,64}, number of visit-level prototypes {4,8,16,32}, number of patient-level prototypes {2,4,8,16}, dropout probability {0.1, 0.3, 0.5}, and learning rate {0.0001, 0.0005, 0.001}. We use early stopping to prevent overfitting for all models, and AUPRC on the validation set is used as the early stopping metric for all tasks apart from the length-of-stay prediction, for which we use AUROC. The early stopping threshold is set to 20 epochs. The implementation codes are available here[2].

4.2 Performance Evaluation

Table 2 displays the results for ProtoEHR and the baselines on both datasets across five tasks. Our approach outperforms all baselines on almost all of the tasks, demonstrating the effectiveness of modeling within and across hierarchical levels for prediction. We can observe that on MIMIC-III, ProtoEHR outperforms the baselines most noticeably in predicting mortality, achieving a 20.7% AUPRC improvement. On MIMIC-IV, our approach achieves the best results in predicting mortality and length of stay, achieving a 17.2% AUPRC improvement and a 7.4% F1-score improvement, respectively. Since mortality prediction is a patient-level task (as explained in Section 4.4.1), achieving superior performance on this task shows that explicitly modeling the hierarchical structure and incorporating patient-level similarity are crucial in helping the model distinguish patients of imbalanced classes.

Among the baselines, GRASP is better at predicting mortality due to its use of prototypes at the patient's level, further demonstrating the importance of capturing intrinsic similarity in the EHR data. In comparison, KerPrint achieves competitive performance on drug recommendation and phenotype prediction because it models the complex relations within the code level and integrates it directly with the learned patient representation, enabling the model to apply code-level knowledge to code-level tasks. Regarding GraphCare, its original implementation relies on GPT-4 to retrieve all possible knowledge triplets associated with each entity, resulting in a highly enriched knowledge graph that includes many entities not present in the original EHR dataset. This external augmentation introduces additional medical knowledge but is computationally expensive and not feasible in our setting. To ensure fairness and reproducibility, we use GPT-4 to generate a smaller set of triplets, limiting the entities to medical codes only. The relatively lower performance of GraphCare under our KG construction suggests that its effectiveness is heavily dependent on the richness of external knowledge. In contrast, our proposed ProtoEHR achieves robust

[2] https://github.com/caizicharles/ProtoEHR.git

Table 2: Results comparison for five tasks on MIMIC-III and MIMIC-IV. Best results are in bold and the second best results are underlined. Performances are reported in the form of mean(std). ProtoEHR achieves best or second-best performance on 24/24 metrics across all five tasks on two datasets, demonstrating robust generalization across datasets and clinical prediction targets.

Model	Task 1: Mortality Prediction				Task 2: Readmission Prediction				Task 3: Length of Stay Prediction			
	MIMIC-III		MIMIC-IV		MIMIC-III		MIMIC-IV		MIMIC-III		MIMIC-IV	
	AUPRC	AUROC	AUPRC	AUROC	AUPRC	AUROC	AUPRC	AUROC	AUROC	F1	AUROC	F1
Deepr	$8.2_{\pm 2.4}$	$67.4_{\pm 4.4}$	$\underline{14.5}_{\pm 2.9}$	$88.2_{\pm 1.6}$	$30.9_{\pm 2.1}$	$59.1_{\pm 2.0}$	$54.3_{\pm 0.9}$	$\underline{70.6}_{\pm 0.5}$	$72.3_{\pm 0.9}$	$22.2_{\pm 1.2}$	$80.8_{\pm 0.2}$	$28.0_{\pm 0.5}$
AdaCare	$6.0_{\pm 1.6}$	$61.5_{\pm 5.5}$	$12.7_{\pm 2.5}$	$88.4_{\pm 1.5}$	$31.4_{\pm 2.2}$	$\underline{60.1}_{\pm 2.0}$	$\underline{54.4}_{\pm 0.9}$	$\underline{70.6}_{\pm 0.5}$	$\underline{73.0}_{\pm 0.8}$	$\underline{22.7}_{\pm 1.2}$	$\underline{82.4}_{\pm 0.2}$	$\underline{29.7}_{\pm 0.4}$
GRASP	$\underline{9.2}_{\pm 3.1}$	$\underline{68.5}_{\pm 4.2}$	$13.6_{\pm 2.7}$	$\underline{88.8}_{\pm 1.4}$	$29.5_{\pm 2.1}$	$57.0_{\pm 2.1}$	$53.0_{\pm 0.9}$	$69.6_{\pm 0.6}$	$71.2_{\pm 0.9}$	$18.1_{\pm 1.0}$	$81.1_{\pm 0.2}$	$26.0_{\pm 0.4}$
StageNet	$5.5_{\pm 1.8}$	$59.4_{\pm 4.5}$	$13.2_{\pm 2.9}$	$87.3_{\pm 1.7}$	$\mathbf{35.2}_{\pm 2.7}$	$60.0_{\pm 2.0}$	$50.2_{\pm 1.0}$	$68.7_{\pm 0.6}$	$71.9_{\pm 0.9}$	$22.3_{\pm 1.1}$	$81.3_{\pm 0.2}$	$27.4_{\pm 0.5}$
GraphCare	$5.6_{\pm 1.5}$	$58.4_{\pm 4.3}$	$9.2_{\pm 2.2}$	$80.8_{\pm 1.9}$	$31.7_{\pm 2.5}$	$58.5_{\pm 2.2}$	$52.3_{\pm 0.9}$	$69.1_{\pm 0.6}$	$65.0_{\pm 0.8}$	$13.4_{\pm 0.7}$	$69.2_{\pm 0.4}$	$19.6_{\pm 0.4}$
KerPrint	$5.0_{\pm 1.3}$	$58.8_{\pm 4.7}$	$10.6_{\pm 2.5}$	$83.7_{\pm 2.2}$	$29.4_{\pm 2.0}$	$57.8_{\pm 1.9}$	$53.4_{\pm 0.9}$	$70.2_{\pm 0.5}$	$70.9_{\pm 0.9}$	$21.6_{\pm 1.2}$	$77.9_{\pm 0.3}$	$26.7_{\pm 0.4}$
ProtoEHR	$\mathbf{11.1}_{\pm 3.7}$	$\mathbf{71.4}_{\pm 3.1}$	$\mathbf{17.0}_{\pm 3.3}$	$\mathbf{89.2}_{\pm 1.3}$	$\underline{33.0}_{\pm 2.5}$	$\mathbf{61.7}_{\pm 2.2}$	$\mathbf{54.5}_{\pm 0.9}$	$\mathbf{70.9}_{\pm 0.5}$	$\mathbf{75.3}_{\pm 0.9}$	$\mathbf{23.6}_{\pm 1.2}$	$\mathbf{83.2}_{\pm 0.2}$	$\mathbf{31.9}_{\pm 0.5}$

Model	Task 4: Drug Recommendation						Task 5: Phenotype Prediction					
	MIMIC-III			MIMIC-IV			MIMIC-III			MIMIC-IV		
	AUPRC	AUROC	F1	AUPRC	AUROC	F1	AUPRC	AUROC	F1	AUPRC	AUROC	F1
Deepr	$67.0_{\pm 0.6}$	$\underline{91.0}_{\pm 0.3}$	$50.9_{\pm 0.7}$	$70.6_{\pm 0.2}$	$95.6_{\pm 0.0}$	$56.6_{\pm 0.3}$	$63.4_{\pm 0.8}$	$83.6_{\pm 0.4}$	$42.9_{\pm 0.9}$	$77.5_{\pm 0.3}$	$92.6_{\pm 0.1}$	$60.9_{\pm 0.4}$
AdaCare	$62.6_{\pm 0.7}$	$88.0_{\pm 0.4}$	$44.5_{\pm 0.5}$	$69.3_{\pm 0.2}$	$94.7_{\pm 0.1}$	$56.0_{\pm 0.3}$	$57.6_{\pm 0.8}$	$80.6_{\pm 0.4}$	$31.5_{\pm 0.9}$	$75.9_{\pm 0.3}$	$91.9_{\pm 0.1}$	$59.9_{\pm 0.4}$
GRASP	$63.0_{\pm 0.7}$	$89.7_{\pm 0.3}$	$45.8_{\pm 0.5}$	$68.9_{\pm 0.2}$	$95.2_{\pm 0.0}$	$55.4_{\pm 0.3}$	$57.6_{\pm 0.7}$	$80.1_{\pm 0.5}$	$31.9_{\pm 0.9}$	$76.2_{\pm 0.3}$	$91.9_{\pm 0.1}$	$60.7_{\pm 0.4}$
StageNet	$66.0_{\pm 0.6}$	$90.4_{\pm 0.3}$	$50.7_{\pm 0.7}$	$70.5_{\pm 0.2}$	$95.4_{\pm 0.0}$	$57.4_{\pm 0.3}$	$62.6_{\pm 0.7}$	$83.1_{\pm 0.4}$	$42.3_{\pm 0.9}$	$77.5_{\pm 0.3}$	$92.6_{\pm 0.1}$	$60.4_{\pm 0.4}$
GraphCare	$65.9_{\pm 0.6}$	$84.3_{\pm 0.4}$	$43.7_{\pm 0.9}$	$64.7_{\pm 0.2}$	$94.4_{\pm 0.1}$	$46.6_{\pm 0.2}$	$62.4_{\pm 0.6}$	$82.5_{\pm 0.4}$	$43.8_{\pm 0.7}$	$73.0_{\pm 0.3}$	$90.0_{\pm 0.1}$	$51.3_{\pm 0.4}$
KerPrint	$\underline{67.9}_{\pm 0.7}$	$\underline{91.0}_{\pm 0.3}$	$\underline{52.3}_{\pm 0.7}$	$\underline{72.1}_{\pm 0.2}$	$\mathbf{95.9}_{\pm 0.0}$	$\underline{58.7}_{\pm 0.2}$	$\underline{67.8}_{\pm 0.8}$	$\underline{85.4}_{\pm 0.4}$	$\mathbf{50.2}_{\pm 1.0}$	$\mathbf{78.7}_{\pm 0.3}$	$\mathbf{93.3}_{\pm 0.1}$	$\underline{63.6}_{\pm 0.4}$
ProtoEHR	$\mathbf{70.6}_{\pm 0.6}$	$\mathbf{91.7}_{\pm 0.3}$	$\mathbf{54.4}_{\pm 0.8}$	$\mathbf{72.4}_{\pm 0.2}$	$\underline{95.8}_{\pm 0.1}$	$\mathbf{60.0}_{\pm 0.2}$	$\mathbf{68.2}_{\pm 0.7}$	$\mathbf{86.1}_{\pm 0.4}$	$\underline{45.6}_{\pm 1.0}$	$\underline{78.5}_{\pm 0.3}$	$\underline{93.2}_{\pm 0.1}$	$\mathbf{64.2}_{\pm 0.4}$

performance by fully leveraging both within-level similarities and cross-level hierarchical structures in the EHR data, demonstrating effectiveness even with a lightweight and clinically grounded KG.

4.3 Ablation Study

Detailed ablations are performed to analyze the contribution of the medical KG constructed, as well as the prototype learning modules. Five sets of ablations are performed, including ProtoEHR without medical KG, without code-, visit-, and patient-level learnable prototypes, and without the hierarchical fusion (HF) module. As shown in Table 3, removing any of the components from the model results in poorer performance. From the results, we can see that removing the learned medical KG causes a 47.6% decrease in AUPRC for the prediction of mortality. This is clear evidence that using medical KG to enhance the modeling within the code-level is critical for the prediction of the EHR and that our KG construction process is effective. In addition, removing code-, visit-, and patient-level prototypes results in a decrease of 5.3%, 20.6%, and 20.0% AUPRC for the task, providing evidence that capturing intrinsic similarities within each level also benefits predictions.

As the importance of the medical KG is clear, we further perform ablations of the different types of edges in the KG to analyze individual contributions from the medical relations. Figures 2a to 2e visualize the results of edge removal in MIMIC-IV. We can observe that removing P ↔ P edges has the most noticeable effect compared to removing the edges of a single type of code. This is particularly significant for the prediction of mortality, resulting in a decrease in AUPRC 30%. Procedures often indicate the severity of the condition, this knowledge is strengthened by linking between procedures, since procedures are generally applied simultaneously. Among the experiments that remove the edges between different types of codes, the removal of D ↔ P connections has the greatest impact on three of the five tasks. Connections between diagnosis and procedure codes reveal the motivations behind the procedures. Tasks such as phenotype prediction require this knowledge to infer the diagnoses behind the visit, it is therefore reasonable for the model's predictive ability to decrease after detaching said edges.

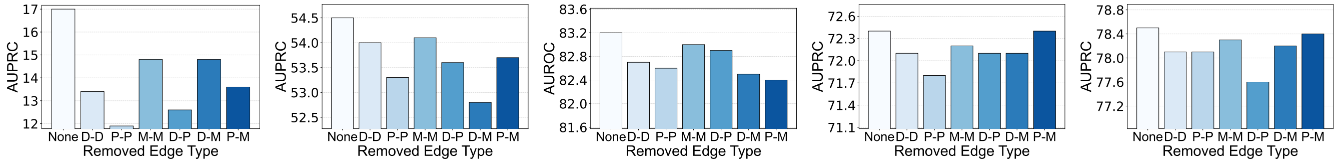
4.4 Interpretability Study

The interpretability granted by hierarchical prototype learning in our framework is two-fold: understanding which level of the hierarchy contributes the most to the predictions and the patterns that the hierarchical prototypes capture. We conducted these experiments in MIMIC-IV because it contains more patients, thereby producing more stable and robust results.

4.4.1 ***Modeling Level Importance v.s. Tasks.*** As explained in Section 3.4, the fusion weights $\{\beta_t | t \in \{c, v, p\}\}$ represent the extent to which prototypes at each level contribute to the final representation and evaluation task. Hence, by recording the fusion weights for each sample in the test set, we can plot the average weights for each task, as shown in Figure 3. To begin with, we can see that patient-level prototypes are the most important for mortality prediction. This is intuitive as mortality prediction is a patient-level task, it requires analyzing the health status of the patient as a whole to determine whether the patient will be deceased within the time frame,

Table 3: Ablation study of our proposed method for five tasks on MIMIC-IV. Notations w/o, Proto., and HF abbreviates without, prototype, and hierarchical fusion respectively.

	Task 1: Mortality		Task 2: Readmission		Task 3: LOS		Task 4: Drug Rec.			Task 5: Phenotype		
Model	AUPRC	AUROC	AUPRC	AUROC	AUROC	F1	AUPRC	AUROC	F1	AUPRC	AUROC	F1
w/ ALL	$17.0_{\pm 3.3}$	$89.2_{\pm 1.3}$	$54.5_{\pm 0.9}$	$70.9_{\pm 0.5}$	$83.2_{\pm 0.2}$	$31.9_{\pm 0.5}$	$72.4_{\pm 0.2}$	$95.8_{\pm 0.1}$	$60.0_{\pm 0.2}$	$78.5_{\pm 0.3}$	$93.2_{\pm 0.1}$	$64.2_{\pm 0.4}$
w/o Medical KG	$8.9_{\pm 1.5}$	$86.7_{\pm 1.8}$	$52.1_{\pm 0.9}$	$68.8_{\pm 0.6}$	$81.6_{\pm 0.2}$	$29.2_{\pm 0.4}$	$71.4_{\pm 0.2}$	$95.4_{\pm 0.1}$	$59.3_{\pm 0.3}$	$77.4_{\pm 0.3}$	$92.6_{\pm 0.1}$	$61.7_{\pm 0.4}$
w/o Code Proto.	$16.1_{\pm 3.6}$	$88.8_{\pm 1.5}$	$52.6_{\pm 0.9}$	$69.9_{\pm 0.5}$	$82.5_{\pm 0.2}$	$30.2_{\pm 0.5}$	$71.7_{\pm 0.2}$	$95.5_{\pm 0.1}$	$60.0_{\pm 0.2}$	$78.4_{\pm 0.3}$	$93.2_{\pm 0.1}$	$63.3_{\pm 0.3}$
w/o Visit Proto.	$13.5_{\pm 2.7}$	$86.8_{\pm 1.6}$	$52.8_{\pm 0.9}$	$69.3_{\pm 0.6}$	$82.8_{\pm 0.2}$	$30.3_{\pm 0.5}$	$71.9_{\pm 0.2}$	$95.6_{\pm 0.1}$	$59.4_{\pm 0.2}$	$78.3_{\pm 0.3}$	$93.1_{\pm 0.1}$	$63.6_{\pm 0.4}$
w/o Patient Proto.	$13.6_{\pm 2.8}$	$86.3_{\pm 1.7}$	$53.3_{\pm 0.9}$	$70.2_{\pm 0.5}$	$83.0_{\pm 0.2}$	$30.6_{\pm 0.5}$	$71.3_{\pm 0.2}$	$95.5_{\pm 0.1}$	$59.8_{\pm 0.2}$	$78.1_{\pm 0.3}$	$93.0_{\pm 0.1}$	$63.6_{\pm 0.4}$
w/o HF	$14.0_{\pm 2.6}$	$88.3_{\pm 1.5}$	$54.2_{\pm 0.9}$	$70.8_{\pm 0.5}$	$83.0_{\pm 0.2}$	$30.9_{\pm 0.5}$	$72.4_{\pm 0.2}$	$95.7_{\pm 0.1}$	$59.6_{\pm 0.3}$	$78.3_{\pm 0.2}$	$93.1_{\pm 0.1}$	$62.3_{\pm 0.3}$



(a) Mortality **(b) Readmission** **(c) LoS** **(d) Drug** **(e) Phenotype**

Figure 2: Bar plots of model performance for five tasks on MIMIC-IV when ablation is performed on the six different types of edges in the medical KG. D, P, and M are the abbreviations for diagnosis, procedure, and medication/prescription respectively.

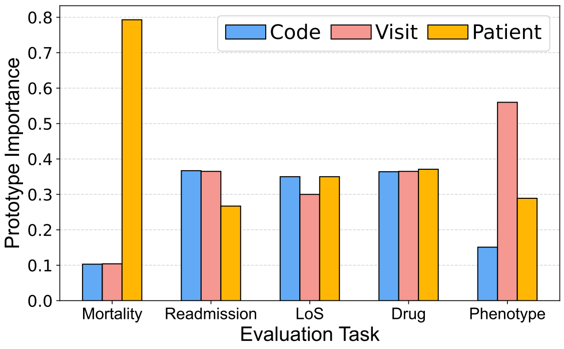


Figure 3: Bar plots showing the contribution of code-, visit-, and patient-level prototype knowledge in the hierarchical fusion module for all five tasks on MIMIC-IV.

and the inclusion of patient-level cohort information improves this assessment. Moreover, despite drug recommendation and phenotype prediction being code-level tasks, the level each task deems important is different, most noticeably the contributions from the visit-level for phenotype prediction. We hypothesize that this is because diagnoses between visits are more correlated than drugs between visits. To verify this statement, we calculate the Jaccard score between the diagnoses/drugs of each patient's visit and the task label, and subsequently average the scores across patients in the test set. The values obtained are 0.287 and 0.439 for drug recommendation and phenotype prediction, respectively, supporting the fact that visit-level information is more important for phenotype prediction and therefore contributes more to predictions.

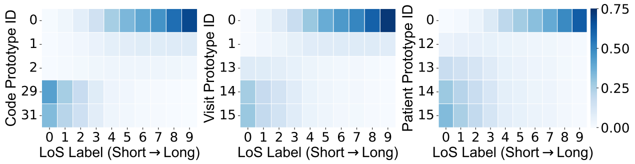


Figure 4: Heat maps of code-, visit, and patient-level prototype importance against the length of stay task label on MIMIC-IV. The five most important prototypes are displayed.

4.4.2 ***Prototype Importance Visualization***. By analyzing the information prototypes encode, we gain further insights into how prototypes encode intrinsic similarity within each level to enhance prediction. Figure 4 visualizes the importance of each prototype in cross-attention for the prediction of the length of stay, where we selected the top five most important prototypes for analysis. For all three levels, as the task label increases (longer stay), some prototypes increase, whilst others decrease in significance in a monotonic manner. This suggests that different cohort knowledge is encoded by different prototypes, and for different patients, different cohort knowledge is used for prediction.

4.4.3 ***Prototype Effectiveness***. Looking at the top 5 most frequent diagnosis and procedure codes present in the last visit of the top 300 patients of each patient-level prototype, the information that these prototypes capture could be inferred. The full results can be seen in Table 4. The importance of patient prototype ID 0 increases from 0.0034 to 0.62 as the patient's length of stay becomes longer, which can be validated by looking at the recorded medical codes. `Complications of Surgical Procedures or Medical Care`, `Other Vascular Catheterization; Not Heart`, and `Extracorporeal Circulation Auxiliary to Open Heart Procedures` are examples of these critical medical codes, generally requiring longer stays as the condition is severe and recovery is gradual. In contrast, since the importance of patient prototype ID 13 decreases from 0.18 to 0.0097 with increasing stay duration, the associated codes should represent less severe medical codes that typically require shorter hospital stays. As observed in the results, `Nonspecific Chest Pain`, `Alcohol-related Disorders`, and

Table 4: Top-5 diagnosis and top-5 procedure codes associated with the top-300 patients sorted by patient-level prototype importance for length of stay prediction on MIMIC-IV. First row: ID 0; Second row: ID 13.

Diagnosis Codes	Procedure Codes
• Residual Codes; Unclassified • Essential Hypertension • Disorders of Lipid Metabolism • Complications of Surgical Procedures or Medical Care • Other Aftercare	• Other Vascular Catheterization; Not Heart • Extracorporeal Circulation Auxiliary to Open Heart Procedures • Enteral and Parenteral Nutrition • Coronary Artery Bypass Graft (CABG) • Other OR Procedures on Vessels Other than Head and Neck
• Essential Hypertension • Nonspecific Chest Pain • Residual Codes; Unclassified • Alcohol-related Disorders • Cardiac Dysrhythmias	• Other Diagnostic Procedures • Routine Chest X-ray • Electrocardiogram • Other Therapeutic Procedures • Other CT Scan

`Routine Chest X-ray` are related to this prototype. Conditions without complications are generally less complex, whilst results from routine checkups are often examined later without the need for the patient to stay, suggesting that this prototype captures medical events with earlier discharge. In a clinical setting, this interpretable characteristic of the model allows clinicians to pinpoint the medical codes/conditions that may have resulted in the deterioration of the patient's health. From another point of view, if a certain prototype repeatedly occurs for different patients, hospital management can inspect the associated codes to prevent the occurrence of any device malfunction or oversight.

To further quantify the effectiveness of the learned prototypes, we perform a clustering experiment to test whether the learned prototypes can be used to discover shared patterns in patients. For each patient in the test set, we retrieve three importance vectors of prototypes from the hierarchical fusion module—one corresponding to each hierarchy level—with each vector encoding the relative importance of that level's prototypes (e.g., a visit-level vector of length four reflects the attention weights of the four learned visit-level prototypes). Subsequently, we apply K-Means clustering to each set of prototype vectors. The silhouette scores are computed for each cluster to measure the compactness and separation of the resulting clusters. Knowing that our learned prototypes are positively correlated with the task labels, higher silhouette scores would indicate that the prototypes effectively capture intrinsic similarities among patients and form a clinical pattern that is beneficial for prediction, thereby demonstrating their quantitative efficacy. The results obtained for mortality and length-of-stay prediction on MIMIC-IV are detailed in Table 5. In accord with the previous analysis in Figure 4, patient-level prototypes play an important role in mortality prediction, and three levels of prototypes contribute equally to length-of-stay prediction.

Table 5: Silhouette scores between prototypes of each level for mortality and length-of-stay prediction tasks.

	Code	Visit	Patient
Mortality	0.49	0.73	0.81
Length-of-Stay	0.56	0.56	0.54

5 Related Work

EHR-based Healthcare Predictions. Existing studies develop various models for EHR-based healthcare predictions. Specifically, GRAM [5] uses recurrent neural networks (RNNs) to capture the temporal dynamics of patient representations, while Deepr [22] and AdaCare [21] apply convolutional neural networks (CNNs) for visit information aggregation. StageNet [9] and MiME [7] further leverage both RNN and CNN to encode and combine visit embeddings for final patient representations.

Recent studies leverage graph structures to capture the relations between medical codes and visits for patient representation learning. For example, GT-BEHRT [23] designs a graph transformer to better capture implicit information in long visit sequences, with more robust patient representations obtained. G-BERT [26] constructs a medical ontology tree to account for the relations between medical codes. Moreover, GraphCare [12], KerPrint [33] and SeqCare [32] either construct personalized KGs or introduce external KGs as the foundation for hierarchical representation learning, where learned representations are subsequently used for healthcare predictions. To further enhance the patient representations, GRASP [36] and PRISM [38] introduce the prototype learning at the patient level only and learn patient-specific prototypes for sharing information.

Due to the high-stakes nature of healthcare, offering interpretability for EHR-based healthcare predictions endows trust in both patients and clinicians [20]. MedPath [34] offers interpretability by encapsulating disease progression paths on the KG that connect symptoms to diagnoses. KARE [13] first augments the patient's EHR context using a KG, and then leverages an LLM to analyze the augmented context for interpretable reasoning chains. GraphCare [23] computes attention weights for nodes in a KG to capture their respective importance. Instead of using KGs, MedRetriever [35] extracts patient-relevant segments directly from the unstructured text, which serve as the rationale for model predictions.

The aforementioned studies underscore the importance of explicitly modeling the hierarchical structure of EHR data, while also highlighting the need to capture intrinsic similarities therein. In contrast to approaches that focus solely on patient-level similarity, our method systematically incorporates intrinsic similarities at all levels of the hierarchy. We construct patient representations by progressively aggregating information from the code level, through the visit level, and ultimately to the patient level. Additionally, the learned prototypes capture clinically meaningful patterns, enhancing both predictive performance and interpretability.

Prototype Learning. Prototype learning refers to a set of machine learning techniques that identifies or extracts a representative "prototype" reflecting the overall information of data within a specific group. This approach has been widely adopted in computer vision. For instance, ClusterFormer [19] iteratively learns prototypes of image features using cross-attention to resemble expectation maximization. ProtoPNet [4] uses prototypes as weights for a CNN to highlight different properties of the images. ProtoGAN [18] acquires class prototypes by training a network to map class features to a lower-dimensional space. Another method [10] applies the moving average to update class prototypes to use as anchors for contrastive learning. Motivated by its advantage of capturing shared information, we introduce hierarchical prototype learning to capture the intrinsic similarity at all three levels, as well as directly enhancing the patient representation at the final fusion stage.

6 Conclusion

In this work, we proposed a novel and interpretable EHR prediction framework that takes advantage of prototype learning and hierarchical learning to explore the within-level similarities and cross-level hierarchy for healthcare prediction. Comprehensive experiments on mortality prediction, readmission prediction, length of stay prediction, drug recommendation, and phenotype prediction were conducted across two datasets to determine the effectiveness of the model. Our interpretability study supports the efficacy of using prototype learning to unravel the reasons behind model predictions, which includes examining the contributions of each level and the importance of hierarchical prototypes toward the output.

In future work, we aim to enhance the construction and learning of medical KGs to better capture the correlations between medical codes and task labels, and to incorporate additional clinical information such as laboratory test results to further improve predictive performance while validating on a more diverse set of datasets.

7 Acknowledgements

Tingting Zhu was supported by the Royal Academy of Engineering under the Research Fellowship scheme.

A Details of Medical KG Construction

During the **retrieval** stage, we use an open-source LLM (Llama 3 70B [8]) to extract candidate relations, yielding 398,555 triplets. As these include spurious links, a **cleaning** stage follows: we ask GPT-4 to label a subset of 30,000 triplets with binary validity, of which 14,378 are judged *true*. We then train a classifier on this labeled subset by first embedding each triplet with BERT [17] and feeding the embeddings to a multilayer perceptron (MLP) to predict validity. Applying the trained classifier to the remaining unlabeled triplets marks 122,422 as likely true at a probability threshold of ≥ 0.5. For the final KG, we combine all 14,378 GPT-verified true triplets with the top 71,890 classifier-scored triplets (the highest-probability subset, sized at 5× the GPT-verified positives), resulting in a KG with 86,268 triplets and 713 entities (including one for padding).

The aim of the final **refinement** stage is to reduce the number of unique edges to obtain generalized representations of medical knowledge. We apply agglomerative clustering with ward linkage to merge edges based on their word embedding distance. This method is effective in recognizing edges with similar names. However, this also introduces the problem that semantically opposing edges could differ by a single word, but could still be grouped as similar. Hence, we use the same LLM as in the second stage to determine whether any cluster contains semantically opposite edges. Triplet refinement reduces the number of unique edges from 2,330 to merely 269, significantly improving the robustness of the constructed medical KG. Note that the directed edges are subsequently reversed for the CompGCN layer. All the prompt designs are provided in the implementation codes.

B LLM Reliability for KG Construction

We assess the reliability of our medical KG constructed using LLMs, focusing on its efficiency, stability, and robustness.

Our approach exhaustively retrieves candidate relations by querying each code pair using an LLM. This offline procedure ensures comprehensive coverage without incurring repeated computational costs. The resulting KG centers on general semantic relations (e.g., "causes"), forming a strong foundation for downstream healthcare tasks. To address concerns about the reliability of LLM-generated facts, we implemented a multi-stage filtering and refinement process to eliminate low-confidence or implausible triplets. These steps contribute to a high-quality, consistent, and non-redundant KG.

In our framework, Llama-70B is used to extract relations between medical code pairs. We evaluated its stability by randomly selecting 100 codes from the set of medical codes C to generate all possible code pairs and extract the linking relation five times for each pair. We then embedded these relations using BERT and computed cosine similarities between the five extractions for each pair. A mean and a standard deviation were obtained for cosine similarities within the five extractions. The overall mean cosine similarity across all code pairs was 0.93 with a standard deviation of only 0.05. These results demonstrate that Llama produces highly consistent outputs, and any sampling randomness is negligible. Compared to traditional KG construction, which relies heavily on manual curation by domain experts, our LLM-based strategy offers a scalable and cost-effective alternative for clinically meaningful graphs.

C LLM for Direct Healthcare Predictions

In addition to the compared baselines, one might question the effectiveness of using LLMs directly for healthcare prediction tasks. Here, we use prompts directly with Llama models to perform mortality prediction on MIMIC-III, which are shown in Table 6. Both Llama-8B and Llama-70B are unsuitable for directly performing mortality prediction when compared to the F1-score achieved by ProtoEHR of 49.1. The smaller 8B model produces invalid outputs at a much higher rate compared to the 70B model, highlighting that prompt engineering alone is insufficient to mitigate these issues in smaller models. Even though the 70B model performs slightly better, its prediction reliability remains unsatisfactory, and the associated deployment costs further discourage its practical application. From the Llama output, we also observe that both models exhibit a bias that frequently predicts patient death, resulting in a high false positive rate. These results reinforce the necessity of our tailored, data-driven approach.

Table 6: Performance of Llama for mortality prediction on MIMIC-III. T denotes temperature.

Model (T)	F1 Score	Error Rate
Llama-8B (T=0)	0.06	14.56%
Llama-8B (T=0.7)	0.05±0.01	25.61%
Llama-70B (T=0)	0.08	0.09%
Llama-70B (T=0.7)	0.07±0.00	3.91%

GenAI Usage Disclosure

LLMs are used during this research for two main purposes: (1) medical KG construction by proposing triplets between medical codes and determining whether the triplets are true or false, and (2) performing grammar and clarity checks during the preparation of the paper. The LLM-generated KG has been validated from multiple aspects, as shown in the experiment section and appendix. No text was generated without human review, and all research contributions are the result of the authors' original work.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. GPT-4 technical report. *arXiv preprint arXiv:2303.08774* (2023).

[2] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A Next-Generation Hyperparameter Optimization Framework. In *The 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2623–2631.

[3] Olivier Bodenreider. 2004. The unified medical language system (UMLS): integrating biomedical terminology. *Nucleic acids research* 32, suppl_1 (2004), D267–D270.

[4] Chaofan Chen, Oscar Li, Daniel Tao, Alina Barnett, Cynthia Rudin, and Jonathan K Su. 2019. This looks like that: Deep learning for interpretable image recognition. *NeurIPS* 32 (2019).

[5] Edward Choi, Mohammad Taha Bahadori, Le Song, Walter F Stewart, and Jimeng Sun. 2017. GRAM: graph-based attention model for healthcare representation learning. In *KDD*. 787–795.

[6] Edward Choi, Mohammad Taha Bahadori, Jimeng Sun, Joshua Kulas, Andy Schuetz, and Walter Stewart. 2016. RETAIN: An interpretable predictive model for healthcare using reverse time attention mechanism. *NIPS* 29 (2016).

[7] Edward Choi, Cao Xiao, Walter Stewart, and Jimeng Sun. 2018. MiME: Multilevel medical embedding of electronic health records for predictive healthcare. *NeurIPS* 31 (2018).

[8] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783* (2024).

[9] Junyi Gao, Cao Xiao, Yasha Wang, Wen Tang, Lucas M Glass, and Jimeng Sun. 2020. StageNet: Stage-aware neural networks for health risk prediction. In *WWW*. 530–540.

[10] Rohit Gupta, Anirban Roy, Claire Christensen, Sujeong Kim, Sarah Gerard, Madeline Cincebeaux, Ajay Divakaran, Todd Grindal, and Mubarak Shah. 2023. Class prototypes based contrastive learning for classifying multi-label and fine-grained educational videos. In *CVPR*. 19923–19933.

[11] Hrayr Harutyunyan, Hrant Khachatrian, David C Kale, Greg Ver Steeg, and Aram Galstyan. 2019. Multitask learning and benchmarking with clinical time series data. *Scientific Data* 6, 1 (2019), 96.

[12] Pengcheng Jiang, Cao Xiao, Adam Cross, and Jimeng Sun. 2024. GraphCare: Enhancing healthcare predictions with personalized knowledge graphs. *ICLR* (2024).

[13] Pengcheng Jiang, Cao Xiao, Minhao Jiang, Parminder Bhatia, Taha Kass-Hout, Jimeng Sun, and Jiawei Han. 2024. Reasoning-enhanced healthcare predictions with knowledge graph community retrieval. *arXiv preprint arXiv:2410.04585* (2024).

[14] Alistair EW Johnson, Lucas Bulgarelli, Lu Shen, Alvin Gayles, Ayad Shammout, Steven Horng, Tom J Pollard, Sicheng Hao, Benjamin Moody, Brian Gow, et al. 2023. MIMIC-IV, a freely accessible electronic health record dataset. *Scientific Data* 10, 1 (2023), 1.

[15] 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. 2016. MIMIC-III, a freely accessible critical care database. *Scientific Data* 3, 1 (2016), 1–9.

[16] William B Kannel, Thomas R Dawber, Abraham Kagan, Nicholas Revotskie, and Joseph Stokes III. 1961. Factors of risk in the development of coronary heart disease—six-year follow-up experience: the Framingham Study. *Annals of Internal Medicine* 55, 1 (1961), 33–50.

[17] Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, Vol. 1. Minneapolis, Minnesota.

[18] Sai Kumar Dwivedi, Vikram Gupta, Rahul Mitra, Shuaib Ahmed, and Arjun Jain. 2019. ProtoGAN: Towards few shot learning for action recognition. In *ICCV Workshop*. 0–0.

[19] James Liang, Yiming Cui, Qifan Wang, Tong Geng, Wenguan Wang, and Dongfang Liu. 2024. ClusterFomer: clustering as a universal visual learner. *NeurIPS* 36 (2024).

[20] Yu Liu, Weiyao Tao, Tong Xia, Simon Knight, and Tingting Zhu. 2025. SurvUnc: A meta-model based uncertainty quantification framework for survival analysis. In *KDD*. 1903–1914.

[21] Liantao Ma, Junyi Gao, Yasha Wang, Chaohe Zhang, Jiangtao Wang, Wenjie Ruan, Wen Tang, Xin Gao, and Xinyu Ma. 2020. AdaCare: Explainable clinical health status representation learning via scale-adaptive feature extraction and recalibration. In *AAAI*, Vol. 34. 825–832.

[22] Phuoc Nguyen, Truyen Tran, Nilmini Wickramasinghe, and Svetha Venkatesh. 2016. Deepr: A convolutional net for medical records. *JBHI* 21, 1 (2016), 22–30.

[23] Raphael Poulain and Rahmatollah Beheshti. 2024. Graph transformers on EHRs: Better representation improves downstream performance. In *ICLR*.

[24] Lior Rokach and Oded Maimon. 2005. Clustering methods. *Data Mining and Knowledge Discovery Handbook* (2005), 321–352.

[25] Tabinda Sarwar, Sattar Seifollahi, Jeffrey Chan, Xiuzhen Zhang, Vural Aksakalli, Irene Hudson, Karin Verspoor, and Lawrence Cavedon. 2022. The secondary use of electronic health records for data mining: Data characteristics and challenges. *CSUR* 55, 2 (2022), 1–40.

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

[27] Junyuan Shang, Cao Xiao, Tengfei Ma, Hongyan Li, and Jimeng Sun. 2019. Gamenet: Graph augmented memory networks for recommending medication combination. In *AAAI*, Vol. 33. 1126–1133.

[28] Benjamin Shickel, Patrick James Tighe, Azra Bihorac, and Parisa Rashidi. 2017. Deep EHR: a survey of recent advances in deep learning techniques for electronic health record (EHR) analysis. *JBHI* 22, 5 (2017), 1589–1604.

[29] Shikhar Vashishth, Soumya Sanyal, Vikram Nitin, and Partha Talukdar. 2020. Composition-based multi-relational graph convolutional networks. In *ICLR*.

[30] Shirly Wang, Matthew BA McDermott, Geeticka Chauhan, Marzyeh Ghassemi, Michael C Hughes, and Tristan Naumann. 2020. MIMIC-Extract: A data extraction, preprocessing, and representation pipeline for MIMIC-III. In *CHIL*. 222–235.

[31] Shuwen Wang and Xingquan Zhu. 2021. Predictive modeling of hospital readmission: challenges and solutions. *TCCB* 19, 5 (2021), 2975–2995.

[32] Yongxin Xu, Xu Chu, Kai Yang, Zhiyuan Wang, Peinie Zou, Hongxin Ding, Junfeng Zhao, Yasha Wang, and Bing Xie. 2023. SeqCare: Sequential training with external medical knowledge graph for diagnosis prediction in healthcare data. In *WWW*. 2819–2830.

[33] Kai Yang, Yongxin Xu, Peinie Zou, Hongxin Ding, Junfeng Zhao, Yasha Wang, and Bing Xie. 2023. KerPrint: Local-global knowledge graph enhanced diagnosis prediction for retrospective and prospective interpretations. In *AAAI*, Vol. 37. 5357–5365.

[34] Muchao Ye, Suhan Cui, Yaqing Wang, Junyu Luo, Cao Xiao, and Fenglong Ma. 2021. Medpath: Augmenting health risk prediction via medical knowledge paths. In *WWW*. 1397–1409.

[35] Muchao Ye, Suhan Cui, Yaqing Wang, Junyu Luo, Cao Xiao, and Fenglong Ma. 2021. Medretriever: Target-driven interpretable health risk prediction via retrieving unstructured medical text. In *CIKM*. 2414–2423.

[36] Chaohe Zhang, Xin Gao, Liantao Ma, Yasha Wang, Jiangtao Wang, and Wen Tang. 2021. GRASP: Generic framework for health status representation learning based on incorporating knowledge from similar patients. In *AAAI*, Vol. 35. 715–723.

[37] Yuqi Zhu, Xiaohan Wang, Jing Chen, Shuofei Qiao, Yixin Ou, Yunzhi Yao, Shumin Deng, Huajun Chen, and Ningyu Zhang. 2024. LLMs for knowledge graph construction and reasoning: Recent capabilities and future opportunities. *WWW* 27, 5 (2024), 58.

[38] Yinghao Zhu, Zixiang Wang, Long He, Shiyun Xie, Xiaochen Zheng, Liantao Ma, and Chengwei Pan. 2024. PRISM: Mitigating EHR Data sparsity via learning from missing feature calibrated prototype patient representations. In *CIKM*. 3560–3569.