Graph Representation Learning for Electronic Health Records (EHR) Data

Zejia You
Department of Computer Science
Tufts University
zyou02@tufts.edu

HyunSu Lee
Department of Computer Science
Tufts University
hyunsu.lee@tufts.edu

Abstract

Electronic Health Records (EHR) are valuable for patient analysis and disease prediction using deep learning. EHR data is high-dimensional with implicit connections among medical concepts (e.g., disease co-occurrence, lab-disease correlations). These connections are informative, especially when data is incomplete. Leveraging these associations can effectively enhance EHR representation learning. Our project explored deep learning methods for graph representation, constructing and implementing several models. Specifically, we developed and applied Graph Representation Learning techniques, including custom Variational Regularization Graph Neural Networks (GNNs) and Relational Graph Convolutional Networks (R-GCNs), to model heterogeneous EHR data. We conducted comparative experiments using the MIMIC-III EHR database, encompassing up to 12 predictive tasks, to evaluate the effectiveness of our approach.

1 Introduction

Electronic Health Records (EHRs) serve as digital repositories of patient information, encompassing a vast amount of clinical data such as medical history, diagnoses, medications, treatment plans, and laboratory test results [13]. Driven by their considerable potential to improve patient care through streamlined processes and comprehensive medical histories, EHR datasets have become a focal point for deep learning-based analytical approaches. Initial explorations applied diverse deep learning techniques to EHR data [3, 14]; however, a pivotal focus has emerged on developing learned representations for medical concepts [4, 5]. Complementing this, recent studies highlight the crucial role of graph-based structures in modeling the complex interrelationships between these medical concepts [18, 6]. By incorporating historical disease and treatment information alongside multifactorial symptoms from EHRs [17], various predictive modeling methods have been introduced, aiming to advance personalized medicine and improve healthcare quality [11].

Effective predictive modeling from EHRs demands capturing the rich, interdependent relationships among patient metadata (e.g., age, gender), historical diagnoses, treatment trajectories, and observed outcomes. These heterogeneous factors not only influence one another but also exhibit high sparsity and frequent missingness. In fact, some conditions are rigorously coded, while others may go undocumented in routine clinical encounters in EHRs. By representing EHR variables as nodes and their interactions as edges, sparse graph structures naturally encode multifactorial dependencies in tabular data. Graph neural networks (GNNs) generalize the localized feature–extraction capabilities of convolutional neural networks (CNNs) to these non-grid domains, allowing models to aggregate information from each node's topological neighborhood, emphasize the most informative signals, and infer missing attributes [2, 7]. As a result, GNNs have emerged as a powerful paradigm for a variety of EHR tasks, including patient representation learning, medical knowledge graph construction, and disease risk prediction.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

Finally, in this project we undertake a comprehensive study of GNN representation learning methods for EHR data by:

- Applying graph representation learning architecture, variationally regularized encoder-decoder graph network and Relational Graph Convolutional Networks (R-GCNs) to model heterogeneous EHR data.
- Exploring multiple attention mechanisms alongside the variational regularization scheme on node embeddings, thereby improving both the focus of attention modules and the quality of learned representations in sparse, heterogeneous clinical graphs.
- Training and fine-tuning these models in the multiple downstream prediction tasks in MIMIC-III dataset.

2 Dataset: MIMIC-III

We did the experiments of our models on the publicly available MIMIC-III database [8], which contains detailed clinical data for 53,423 unique adult ICU admissions between 2001 and 2012. MIMIC-III includes patient demographics, vital signs, laboratory measurements, medication administrations, procedures, diagnostic codes, and free-text clinical notes. From this rich resource, we selected thirteen target outcomes: Advanced Cancer, Advanced Heart Disease, Advanced Lung Disease, Alcohol Abuse, Chronic Neurological Dystrophies, Chronic Pain Fibromyalgia, Dementia, Depression, Non-Adherence, Substance Abuse, and Schizophrenia and Other Psychiatric Disorders to evaluate model performance across a diverse set of clinically relevant prediction tasks.

3 Methods

We built two types of model to finish the prediction task of every labels.

3.1 Variationally regularized Encoder-Decoder Model

After a thorough study and analysis of [18], we followed their underlying principles to construct a Variationally Regularized Formulation of Graph Neural Networks (VGNN). The detailed approach is as follows:

The core of the model is an encoder-decoder graph neural network. We begin by embedding EHR codes $V = 1, 2, \ldots, N$ into high-dimensional vectors h_i $(i \in V)$, where $h_i \in \mathbb{R}^d$. Unlike approaches that rely heavily on external node features (e.g., GAT [15]), this method focuses on learning the representations of each medical concept directly from the data.

For a given patient X, we consider their observed codes $V_{\text{obs}} = x_1, x_2, \ldots, x_n$. These observed nodes are initially fully connected to form a graph. This initial graph structure and the node representations are then refined and updated over L additional graph layers. This sequence of layers constitutes the encoder graph. The encoder's role is to process the medical embeddings and represent the patient's record as a graph structure with learned node representations. Following the encoder, we introduce additional nodes $V_{\text{out}} = y_1, y_2, \ldots, y_m$ corresponding to the prediction tasks. These output nodes are fully connected to the final output nodes of the encoder graph. This part of the network is termed the decoder graph. The decoder takes the learned graph representations from the encoder and uses them to make inferences for the specified prediction tasks.

Within each graph layer l, the node representations are updated through a graph propagation mechanism. The update rule is defined as:

$$H^{(l+1)} = \text{FFN}\left(A^{(l)} H^{(l)} W^{(l)} + b^{(l)}\right) \tag{1}$$

where: $W^{(l)} \in \mathbb{R}^{d \times d}$ and $b^{(l)} \in \mathbb{R}^d$ are the weight matrix and bias vector for a linear transformation applied at layer l. $H^{(l)}$ is a matrix where each row is the representation $h_i^{(l)}$ of an observed node at layer l. $A^{(l)}$ is the adjacency matrix at layer l, representing the connections between nodes. FFN denotes a Feed-Forward Network, which in our case is a multilayer perceptron composed of linear layers, ReLU activations, dropout, and layer normalization.

The dimensions of $H^{(l)}$ and $A^{(l)}$ vary depending on the sample and the layer location (encoder or decoder). In the encoder, the nodes are V_{obs}, so $A^{(l)} \in \mathbb{R}^{n\times n}$ and $H^{(l)} \in \mathbb{R}^{d\times n}$. In the decoder, the nodes include both V_{obs} and V_{out}, resulting in $A^{(l)} \in \mathbb{R}^{(n+m)\times(n+m)}$ and $H^{(l)} \in \mathbb{R}^{d\times(n+m)}$.

The elements of the adjacency matrix $A_{ij}^{(l)}$ on the edge connecting node i to node j are computed using an attention mechanism. This mechanism allows the model to learn the importance of connections between medical concepts. The attention weights are computed using a softmax function:

$$A_{ij} = \text{softmax}(e_{ij}) = \frac{\exp(e_{ij})}{\sum_{p\in\mathcal{N}_i} \exp(e_{ip})} \tag{2}$$

where $e_{ip} \in \mathbb{R}$ are the attention coefficients for node i over its neighborhood $\mathcal{N}_i$. The attention coefficients e_{ij} are computed based on the interaction between the representations of nodes i and j. Utilizing a multi-head attention mechanism to enhance expressive power.

$$e_{ij} = \text{LeakyReLU}\left(a^\top [Wh_i \,\|\, Wh_j]\right)/d_h \tag{3}$$

In this attention mechanism, we need to concatenate two input vectors and apply a linear layer $a \in \mathbb{R}^{2d\times 1}$. The attention coefficients are computed by the upper Eq. 3. W is a linear layer, and d_h is the dimensionaility of h_i. Multi-head attention enables the model to capture diverse relationships by jointly attending to information from different representation subspaces. For K-head attention, the output size of the linear layers $W^{(l)}$ and $b^{(l)}$ in Eq. 3 is adjusted to dK, and the attention heads are computed in parallel. The outputs from multiple heads are concatenated, requiring the input size of the subsequent feed-forward networks to be dK.

We utilized a new attention mechanism inspired by GATv2 [1] in this encoder-decoder framework. For our GATv2 variant, we replace the original "concatenate → linear → LeakyReLU → softmax" scoring with an additive attention formulation that first projects each node's features and then scores edge pairs via a small two-layer MLP.

Concretely, at layer l, each head k first transforms the node feature $\mathbf{x}_i \in \mathbb{R}^d$ into a hidden vector:

$$h_i^{(l,k)} = W_k x_i, \quad W_k \in \mathbb{R}^{d_h\times d} \tag{4}$$

Then, for every edge $u \to v$, we form the pairwise feature vector:

$$z_{uv}^{(k)} = [h_u^{(l,k)} \,\|\, h_v^{(l,k)}] \in \mathbb{R}^{2d_h} \tag{5}$$

An unnormalized attention score is then computed as:

$$e_{uv}^{(l,k)} = \text{LeakyReLU}\left(a_k^\top\, \sigma\left(W_{\text{cat}}\, z_{uv}^{(k)}\right)\right)/\sqrt{d_h}, \quad W_{\text{cat}} \in \mathbb{R}^{d_h\times 2d_h}, \quad a_k \in \mathbb{R}^{d_h\times 1} \tag{6}$$

Here, W_{cat} and a_k together define a twolayer MLP, replacing the single linear projection in standard GAT (e.g., Eq. 3). We then normalize the scores $\alpha_{uv}^{(l,k)}$ across the neighborhood $\mathcal{N}(u)$ using a softmax function like Eq. 2, we just change the e_{ij} to $e_{uv}^{(l,k)}$. Using the attention coefficients, the updated hidden representation for node u at head k is:

$$\tilde{h}_u^{(l,k)} = \sum_{v\in\mathcal{N}(u)} \alpha_{uv}^{(l,k)}\, h_v^{(l,k)} \tag{7}$$

Finally, we concatenate (or average) the representations across all K heads, apply dropout, layer normalization, and a nonlinearity before feeding the result into the feed-forward block, as in Eq. (1). Empirically, this additive GATv2 scoring yields more flexible edge weights and improved downstream AUPRC across most tasks.

Given the often imbalanced nature of predictive tasks in EHR data in Table 3, we employ a weighted binary cross-entropy loss function for training:

$$\mathcal{L}_{\text{bce}} = -\sum_{y_c\in V_{\text{out}}} w_c \cdot \left[Y_c \cdot \log\left(\sigma(\hat{y}_c)\right) + (1-Y_c)\cdot \log\left(1-\sigma(\hat{y}_c)\right)\right] \tag{8}$$

where: σ is the sigmoid function. Y_c is the ground truth label for output node y_c. $\hat{y}_c \in \mathbb{R}$ is the output of the prediction by node y_c. w_c is a weight (specifically, the negative-to-positive ratio) assigned to the minority class to address label imbalance. The total loss for a mini-batch is computed as the mean of the individual losses calculated for each sample.

The next important part of the model is the variationally regularized mechanism inspired by Variational Graph Autoencoders (VGAE) [10], which use a Gaussian prior on node representations to improve link inference. In specific, adding a latent layer between the encoder and decoder. This latent layer serves to regularize the graph representation. Let $Z = \{z_i\}_{i \in V_{\text{obs}}}$, where $z_i \in \mathbb{R}^d$, be the latent variables corresponding to the observed node representations $h_i^{(L)}$ after the encoder layers. We assume a standard normal prior distribution over the latent variables, $p(z_i) \sim \mathcal{N}(0, I)$, and model the generative encoder distribution as a Gaussian, $q(z_i \mid X) \sim \mathcal{N}(\mu_i, \exp(\sigma_i))$. The mean and log-variance are learned from encoder outputs via linear transformations: $\mu_i = W_\mu h_i^{(L)} + b_\mu$ and $\sigma_i = W_\sigma h_i^{(L)} + b_\sigma$. The variance is parameterized exponentially to ensure non-negativity. The sampled latent variables z_i then replace the encoder outputs $h_i^{(L)}$ as inputs to the decoder layer. Assuming conditional independence, the full joint distributions are $p(Z) = \prod_{i \in Z} p(z_i)$ and $q(Z \mid X) = \prod_{i \in Z} q(z_i \mid X)$. The variational formulation of auto-encoders seeks to maximize the log-posterior $p(X \mid Z)$ by optimizing the Evidence Lower Bound (ELBO) [9].

$$\text{ELBO} = \mathbb{E}_q \left[\log p(\hat{X} \mid \mathbf{Z}) \right] - \text{KL}\left[q(\mathbf{Z} \mid X) \,\|\, p(\mathbf{Z}) \right] \tag{9}$$

In our supervised model, we combine this divergence loss with the weighted binary cross-entropy loss $\mathcal{L}_{\text{bce}}$ from Eq. 8 to form the final objective:

$$\mathcal{L}(y, \hat{y}) = \mathcal{L}_{\text{bce}}(y, \hat{y}) + \text{KL}\left[q(\mathbf{Z} \mid X) \,\|\, p(\mathbf{Z}) \right] \tag{10}$$

This combined loss encourages accurate prediction while simultaneously regularizing the learned node representations, mitigating the representation collapse observed in the basic encoder-decoder architecture.

3.2 Relational Graph Convolutional Networks (R-GCN)

To complement the variational approach, we implemented a relational graph convolutional network (R-GCN) as a baseline model for representation learning and prediction on the same EHR graph. R-GCNs are designed for heterogeneous graphs with multiple edge types and are especially suited for settings like EHRs, where relationships between entities are semantically diverse [16] (e.g., diagnosis vs. treatment).

Graph Structure and Data Representation. We define a global heterogeneous graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where each node belongs to one of three types: `patient`, `icd` (procedure code), and `ndc` (diagnosis or medication code). Edges connect patients to their associated codes via typed relations: `process` (patient → icd) and `diagnosis` (patient → ndc), along with their reverse edges `rev_process` and `rev_diagnosis` to enable bidirectional message passing.

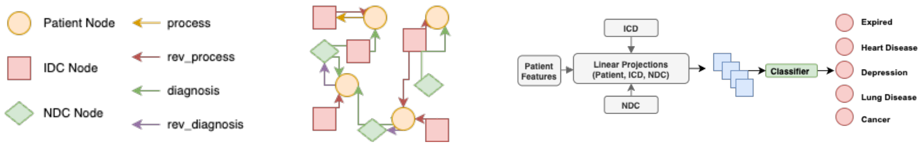
The R-GCN model consists of the following components:

- **Type-specific input projection layers:** A separate linear layer projects the input features of each node type into a shared hidden space $\mathbb{R}^d$.
- **R-GCN layers:** We stack two R-GCN layers to perform message passing over the typed edges. Each layer applies relation-specific transformations and aggregates messages using normalized sums.
- **Classifier:** The output embeddings of patient nodes are passed through a linear classification layer followed by a sigmoid activation to produce probabilities for each binary prediction task.

Formally, each R-GCN layer updates the representation $h_v^{(l)}$ of node v as follows:

$$h_v^{(l+1)} = \sigma \left(\sum_{r \in \mathcal{R}} \sum_{u \in \mathcal{N} * r(v)} \frac{1}{c_{v,r}} W_r^{(l)} h_u^{(l)} + W_0^{(l)} h_v^{(l)} \right) \tag{11}$$

where $\mathcal{N}_r(v)$ is the set of neighbors of v under relation r, $W_r^{(l)}$ are relation-specific weight matrices, $W_0^{(l)}$ is a shared self-loop transformation, $c_{v,r}$ is a normalization constant, and σ is the ReLU activation.



(a) Graph Construction (b) Model Architecture

Figure 1: Introduction to R-GCN structure

4 Experimental Results

Table 1: Model evaluation of mortality prediction in our dataset using precision-recall curves.

Label	VGNN	VGNN(GATv2)	R-GCN
Mortality	0.834	0.823	0.781
Non-Adherence	0.199	0.166	0.133
Advanced Heart Disease	0.366	0.352	0.275
Advanced Lung Disease	0.174	0.231	0.146
Schizophrenia and other Psychiatric Disorders	0.268	0.262	0.288
Alcohol Abuse	0.506	0.524	0.434
Other Substance Abuse	0.352	0.385	0.310
Chronic Pain Fibromyalgia	0.268	0.268	0.290
Chronic Neurological Dystrophies	0.365	0.371	0.259
Advanced Cancer	0.191	0.183	0.134
Depression	0.333	0.330	0.338
Dementia	0.159	0.132	0.042

In this project, we test the methods in the context of 12 different prediction tasks based on the MIMIC-III EHR dataset. Since all of three tasks have imbalances class labels according to Table 3, the precision-recall curve is a more informative evaluation metric on the prediction performance than ROC curve [12]. To quantify PR-curve, we compute the area under PR-curve (AUPRC) to summarize the curve. The results, detailed in Table 1, show that performance of these 3 models in 12 different prediction tasks. The VGNN model and VGNN model (GATv2 version) were trained on NVIDIA L40S with CUDA 12.2 version. Although our modification version VGNN(GATv2) didn't show enough enhancement in performance metric, it has an improvement in training efficiency, achieving at least **20% reduction in training time** compared to the original VGNN. Due to the relatively long training time required for the model, parameter tuning was primarily guided by empirical observations rather than grid search, which may have influced the performance of some tasks.

Compared to the VGNN models, the R-GCN model does not require per-patient subgraph construction or attention mechanisms, making it more scalable and interpretable. However, it lacks the ability to learn personalized attention weights or latent uncertainty, which can be beneficial in modeling sparse and noisy EHR data. Despite its simplicity, the R-GCN achieved competitive performance across several conditions and provides a strong, efficient baseline for heterogeneous graph modeling in EHRs.

5 Conclusion

In this course project, we did a comprehensive survey of graph-based deep learning methods applied to healthcare data and acquired hands-on experience in graph representation learning, attention mechanisms, and large-scale model training on GPU clusters. We implemented three distinct GNN models for EHR dataset prediction tasks and achieved decent AUPRC performance across twelve binary classification challenges. Moreover, by integrating GATv2-style attention into the original variational regularization encoder–decoder framework (VGNN), we enhanced both predictive accuracy and training efficiency. Through this valuable project experience, we have learned a great deal and our interest in machine learning knowledge has become even stronger. We look forward to more interesting and meaningful future projects.

References

[1] Shaked Brody, Uri Alon, and Eran Yahav. How attentive are graph attention networks? In *International Conference on Learning Representations*, 2022.

[2] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Spectral networks and locally connected networks on graphs, 2014. URL https://arxiv.org/abs/1312.6203.

[3] Yu Cheng, Fei Wang, Ping Zhang, and Jianying Hu. Risk prediction with electronic health records: A deep learning approach. In *SDM*, 2016.

[4] Edward Choi, Mohammad Taha Bahadori, Elizabeth Searles, Catherine Coffey, Michael Thompson, James Bost, Javier Tejedor-Sojo, and Jimeng Sun. Multi-layer representation learning for medical concepts. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 1495–1504, New York, NY, USA, 2016. Association for Computing Machinery.

[5] Edward Choi, Cao Xiao, Walter F. Stewart, and Jimeng Sun. Mime: multilevel medical embedding of electronic health records for predictive healthcare. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, page 4552–4562, Red Hook, NY, USA, 2018. Curran Associates Inc.

[6] Edward Choi, Zhen Xu, Yujia Li, Michael Dusenberry, Gerardo Flores, Emily Xue, and Andrew Dai. Learning the graphical structure of electronic health records with graph convolutional transformer. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(01):606–613, Apr. 2020.

[7] Mikael Henaff, Joan Bruna, and Yann LeCun. Deep convolutional networks on graph-structured data, 2015. URL https://arxiv.org/abs/1506.05163.

[8] Alistair Johnson, Tom Pollard, Lu Shen, Li-wei Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Celi, and Roger Mark. Mimic-iii, a freely accessible critical care database. *Scientific Data*, 3:160035, 05 2016.

[9] Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2022. URL https://arxiv.org/abs/1312.6114.

[10] Thomas N. Kipf and Max Welling. Variational Graph Auto-Encoders. *arXiv:1611.07308*, November 2016.

[11] Samson Mataraso, Camilo Espinosa, David Seong, Momsen Reincke, Eloïse Berson, Jonathan Reiss, Yeasul Kim, Marc Ghanem, Chi-Hung Shu, Tomin James, Yuqi Tan, Sayane Shome, Ina Stelzer, Dorien Feyaerts, Ronald Wong, Gary Shaw, Martin Angst, Brice Gaudilliere, David Stevenson, and Nima Aghaeepour. A machine learning approach to leveraging electronic health records for enhanced omics analysis. *Nature Machine Intelligence*, 7:293–306, 01 2025.

[12] Takaya Saito and Marc Rehmsmeier. The Precision-Recall Plot Is More Informative than the ROC Plot When Evaluating Binary Classifiers on Imbalanced Datasets. *PLoS ONE*, 10(3): e0118432, March 2015. ISSN 1932-6203.

[13] Tom Seymour, Dean A. Frantsvog, and Tod Graeber. Electronic health records (ehr). *American Journal of Health Sciences*, 3:201–210, 2012.

[14] Benjamin Shickel, Patrick James Tighe, Azra Bihorac, and Parisa Rashidi. Deep ehr: A survey of recent advances in deep learning techniques for electronic health record (ehr) analysis. *IEEE Journal of Biomedical and Health Informatics*, 22(5):1589–1604, 2018. doi: 10.1109/JBHI.2017.2767063.

[15] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. *6th International Conference on Learning Representations*, 2017.

[16] Tianyu Wang, Ruixue Zhang, Kexin Huang, Rose Yu, and Jimeng Sun. Graph-based deep learning for electronic health records: A survey. *Journal of Biomedical Informatics*, 151:104439, 2024.

[17] Siyue Yang, Paul Varghese, Ellen Stephenson, Karen Tu, and Jessica Gronsbell. Machine learning approaches for electronic health records phenotyping: a methodical review. *Journal of the American Medical Informatics Association*, 30(2):367–381, 11 2022. ISSN 1527-974X.

[18] Weicheng Zhu and Narges Razavian. Variationally regularized graph-based representation learning for electronic health records. pages 1–13, 04 2021.

A Appendix / Supplemental material

A.1 Dataset Pre-processing and Analysis

We begin our data pre-processing step for the raw MIMIC-III CSV tables. From the patient table we extract each subject's core demographics and encounter metadata—Subject ID, HADM ID, ICU stay ID, gender, age and obesity status. From the medication table we capture every administered drug via its National Drug Code (NDC) as well as the CareVue and MetaVision infusion item IDs (Input CV and Input MV). The procedure table supplies all ICD-9 billing codes corresponding to surgeries or interventions performed during the stay. Finally, for laboratory data we record each charted Item ID (e.g. the type of measurement, such as red blood cells) along with its numeric value (e.g. the measured cell count).

Table 2: Summary of feature counts per ICU stay and total unique codes/items.

Feature	Avg. count per ICU stay	Total unique count
Medication	56.52	2041
Procedures	4.91	591
Lab measurements	571.70	1044
Total ICU stays	—	1531
Total patients	—	1034

Table 3: Positive-class prevalence for each binary label in training, validation, test dataset.

Label	Train	Validation	Test
Mortality	0.623	0.630	0.588
Non-Adherence	0.096	0.117	0.072
Advanced Heart Disease	0.165	0.162	0.203
Advanced Lung Disease	0.104	0.097	0.098
Schizophrenia and other Psychiatric Disorders	0.187	0.182	0.170
Alcohol Abuse	0.117	0.156	0.157
Other Substance Abuse	0.096	0.110	0.105
Chronic Pain Fibromyalgia	0.194	0.227	0.222
Chronic Neurological Dystrophies	0.228	0.221	0.242
Advanced Cancer	0.102	0.091	0.085
Depression	0.288	0.305	0.301
Dementia	0.072	0.052	0.072

Specifically, in cases where a patient (identified by Subject ID) made multiple visits to the emergency room (with unique ICU stay IDs) at the same hospital (with unique HADM IDs), multiple ICU stay IDs were assigned. Subject ID, HADM ID, and ICU stay ID serve as unique key values in this con- text. Our focus was on extracting specific details such as prescribed medications, procedures performed, and relevant measurements (e.g., blood pressure) from the dataset. The statistics of the dataset are summarized in Table 2 and Table 3.

A.2 Experiment Setting Details

Since the lab data we use shows a natural temporal attribute, we did not follow the conventional strategy of randomly partitioning the dataset into training, validation, and test sets by unique patient

IDs in an 8:1:1 ratio. To ensure consistency in multi-label task partitioning and comparability across evaluations, we replaced the original random split logic with explicit data partitioning based on predefined index files for training, validation, and test sets. This approach enhances the fairness and robustness of experimental reproducibility and prevents potential data leakage.

For MIMIC-III data, the labels are imbalanced, so the weighted loss cannot effectively improve the performance. Hence, we upsample positive samples of the training set by 3 times to observe more positive samples within a given number of epochs. Validation and test sets retain their original distribution.

For VGNN and VGNN(GATv2), we tune the hyperparameters including the number of attention heads $[3, 4]$, embedding sizes $[256, 1024]$, dropout rates $[0.5, 1]$, and learning rates $[10^{-5}, 10^{-3}]$. Learning rate decay is applied to mitigate overfitting. Due to the relatively long training time and the variety of prediction tasks, parameter tuning was primarily guided by empirical observations rather than exhaustive grid search.

We trained the R-GCN using mini-batches sampled via PyTorch Geometric's `NeighborLoader`, which constructs local subgraphs centered on each patient. We used the binary cross-entropy loss and evaluated performance using AUPRC across all 12 clinical prediction tasks. Node features for patients include normalized lab values, age, and gender, while ICD and NDC nodes use learned embeddings. Edge indices and node type information are preprocessed and stored in `HeteroData` format.