# Collaborative Graph Learning with Auxiliary Text for Temporal Event Prediction in Healthcare

**Chang Lu**[1], **Chandan K. Reddy**[2], **Prithwish Chakraborty**[3], **Samantha Kleinberg**[1], **Yue Ning**[1]
[1]Department of Computer Science, Stevens Institute of Technology
[2]Department of Computer Science, Virginia Tech
[3]IBM Research
{clu13, samantha.kleinberg, yue.ning}@stevens.edu, reddy@cs.vt.edu, prithwish.chakraborty@ibm.com

## Abstract

Accurate and explainable health event predictions are becoming crucial for healthcare providers to develop care plans for patients. The availability of electronic health records (EHR) has enabled machine learning advances in providing these predictions. However, many deep learning based methods are not satisfactory in solving several key challenges: 1) effectively utilizing disease domain knowledge; 2) collaboratively learning representations of patients and diseases; and 3) incorporating unstructured text. To address these issues, we propose a collaborative graph learning model to explore patient-disease interactions and medical domain knowledge. Our solution is able to capture structural features of both patients and diseases. The proposed model also utilizes unstructured text data by employing an attention regulation strategy and then integrates attentive text features into a sequential learning process. We conduct extensive experiments on two important healthcare problems to show the competitive prediction performance of the proposed method compared with various state-of-the-art models. We also confirm the effectiveness of learned representations and model interpretability by a set of ablation and case studies.

## 1 Introduction

Electronic health records (EHR) consist of patients' temporal visit information in health facilities, such as medical history and doctors' diagnoses. The usage and analysis of EHR not only improves the quality and efficiency of in-hospital patient care but also provides valuable data sources for researchers to predict health events, including diagnoses, medications, and mortality rates, etc. A key research problem is improving prediction performance by learning better representations of patients and diseases so that improved risk control and treatments can be provided. There have been many works on this problem using deep learning models, such as recurrent neural networks (RNN) [Choi *et al.*, 2016a], convolutional neural networks (CNN) [Nguyen *et al.*, 2017], and attention-based mechanisms [Ma *et al.*, 2017]. However, several challenges remain in utilizing EHR data and interpreting models:

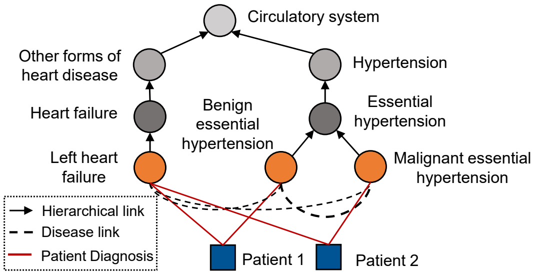


Figure 1: An example of the hierarchical structure of the ICD-9-CM system, disease interaction, and patient diagnosis.

1. **Effectively utilizing the domain knowledge of diseases**. Recently, graph structures are being adopted [Choi *et al.*, 2017] using disease hierarchies, where diseases are classified into various types at different levels. For example, Figure 1 shows a classification of two forms of hypertension and one form of heart failure. One problem is that existing works [Choi *et al.*, 2017; Shang *et al.*, 2019] only consider the vertical relationship between a disease and its ancestors (hierarchical link). However, they ignore horizontal disease links that can reflect disease complications and help to predict future diagnoses.
2. **Collaboratively learning patient-disease interactions.** Patients with the same diagnoses may have other similar diseases (patient diagnosis in Figure 1). Existing approaches [Choi *et al.*, 2017; Ma *et al.*, 2017] treat patients as independent samples by using diagnoses to represent patients, but they fail to capture patient similarities, which help in predicting new-onset diseases from other patients.
3. **Incorporating unstructured text.** Unstructured data in EHR including clinical notes contain indicative features such as physical conditions and medical history. For example, a note: "*The patient was intubated for respiratory distress and increased work of breathing. He was also hypertensive with systolic in the 70s*" indicates that this patient has a history of respiratory problems and hypertension. Most models [Choi *et al.*, 2016b; Bai *et al.*, 2018] do not fully utilize such data. This often leads to unsatisfactory prediction performance and lack of interpretability.

To address these problems, we first present a hierarchical embedding method for diseases to utilize medical domain knowledge. Then, we design a collaborative graph neural network to learn hidden representations from two graphs: a *patient-disease observation graph* and a *disease ontology graph*. In the observation graph, if a patient is diagnosed with a disease, we create an edge between this patient and the disease. The ontology graph uses weighted ontology edges to describe horizontal disease interactions. Moreover, to learn the contributions of keywords for predictions, we design a TF-IDF-rectified attention mechanism for clinical notes which takes visit temporal features as context information. Finally, combining disease and text features, the proposed model is evaluated on two tasks: predicting patients' future diagnoses and heart failure events. The main contributions of this work are summarized as follows:

- We propose to collaboratively learn the representations of patients and diseases on the observation and ontology graphs. We also utilize the hierarchical structure of medical domain knowledge and introduce an ontology weight to capture hidden disease correlations.
- We integrate structured information of patients' previous diagnoses and unstructured information of clinical notes with a TF-IDF-rectified attention method. It allows us to regulate attention scores without any manual intervention and alleviates the issue of using attention as a tool to audit a model [Jain and Wallace, 2019].
- We conduct extensive experiments and illustrate that the proposed model outperforms state-of-the-art models for prediction tasks on MIMIC-III dataset. We also provide detailed analysis for model predictions.

## 2 Related Work

Deep learning models, especially RNN models, have been applied to predict health events and learn representations of medical concepts. DoctorAI [Choi *et al.*, 2016a] uses RNN to predict diagnoses in patients' next visits and the time duration between patients' current and next visits. RETAIN [Choi *et al.*, 2016b] improves the prediction accuracy through a sophisticated attention process on RNN. Dipole [Ma *et al.*, 2017] uses a bi-directional RNN and attention to predict diagnoses of patients' next visits. Both Timeline [Bai *et al.*, 2018] and ConCare [Ma *et al.*, 2020b] utilize time-aware attention mechanisms in RNN for health event predictions. However, RNN-based models regard patients as independent samples and ignore relationships between diseases and patients which help to predict diagnoses for similar patients.

Recently, graph structures are adopted to explore medical domain knowledge and relations of medical concepts. GRAM [Choi *et al.*, 2017] constructs a disease graph from medical knowledge. MiME [Choi *et al.*, 2018] utilizes connections between diagnoses and treatments in each visit to construct a graph. GBERT [Shang *et al.*, 2019] jointly learns two graph structures of diseases and medications to recommend medications. It uses a bi-directional transformer to learn visit embeddings. MedGCN [Mao *et al.*, 2019] combines patients, visits, lab results, and medicines to construct a heterogeneous graph for medication recommendations. GCT [Choi *et al.*, 2020] also builds graph structures of diagnoses, treatments, and lab results. However, these models only consider disease hierarchical structures, while neglecting disease horizontal links that reflect hidden disease complications. As a result, prediction performance is limited.

In addition, CNN and Autoencoder are also adopted to predict health events. DeepPatient [Miotto *et al.*, 2016] uses an MLP as an autoencoder to rebuild features in EHR. Deepr [Nguyen *et al.*, 2017] treats diagnoses in a visit as words to predict future risks such as readmissions in three months. AdaCare [Ma *et al.*, 2020a] uses multi-scale dilated convolution to capture dynamic variations of biomarkers over time. However, these models neither consider medical domain knowledge nor explore patient similarities as discussed.

In this paper, we explore disease horizontal connections using a disease ontology graph. We collaboratively learn representations of both patients and diseases in their associated networks. We also design an attention regulation strategy on unstructured text features to provide quantified contributions of clinical notes and interpretations of prediction results.

## 3 Methodology

### 3.1 Problem Formulation

An EHR dataset is a collection of patient visit records. Let $\mathcal{C} = \{c_1, c_2, \dots, c_{|\mathcal{C}|}\}$ be the entire set of diseases represented by medical codes in an EHR dataset, where $|\mathcal{C}|$ is the medical code number. Let $\mathcal{N} = \{\omega_1, \omega_2, \dots, \omega_{|\mathcal{N}|}\}$ be the dictionary of clinical notes, where $|\mathcal{N}|$ is the word number.

**EHR dataset.** An EHR dataset is given by $\mathcal{D} = \{\gamma_u | u \in \mathcal{U}\}$ where $\mathcal{U}$ is the collection of patients in $\mathcal{D}$ and $\gamma_u = (V_1^u, V_2^u, \dots, V_T^u)$ is a visit sequence of patient $u$. Each visit $V_t^u = \{C_t^u, N_t^u\}$ is recorded with a subset of medical codes $C_t^u \subset \mathcal{C}$, and a paragraph of clinical notes $N_t^u \subset \mathcal{N}$ containing a sequence of $|N_t^u|$ words.

**Diagnosis prediction.** Given a patient $u$'s previous $T$ visits, this task predicts a binary vector $\hat{\mathbf{y}} \in \{0, 1\}^{|\mathcal{C}|}$ which represents the possible diagnoses in $(T+1)$-th visit. $\hat{\mathbf{y}}_i = 1$ denotes $c_i$ is predicted in $C_{T+1}^u$.

**Heart failure prediction.** Given a patient $u$'s previous $T$ visits, this task predicts a binary value $\hat{y} \in \{0, 1\}$. $\hat{y} = 1$ denotes that $u$ is predicted with heart failure[1] in $(T+1)$-th visit.

In the rest of this paper, we drop the superscript $u$ in $V_t^u$, $C_t^u$, and $N_t^u$ for convenience unless otherwise stated.

### 3.2 The Proposed Model

In this section, we propose a **C**ollaborative **G**raph **L**earning model, **CGL**. An overview of CGL is shown in Figure 2.

#### Hierarchical Embedding for Medical Codes

ICD-9-CM is an official system of assigning codes to diseases. It hierarchically classifies medical codes into different types of diseases in $K$ levels. This forms a tree structure where each node has only one parent. Note that most medical codes in patients' visits from EHR data are leaf nodes. However, a patient can be diagnosed with a higher level disease, i.e., non-leaf node. Therefore, we recursively create virtual

[1] The medical codes of heart failure start with 428 in ICD-9-CM.

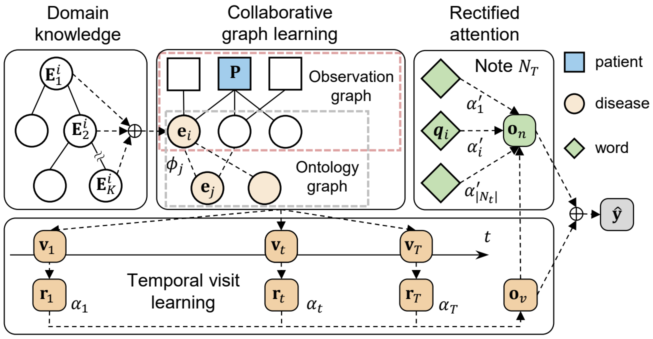


Figure 2: An overview of the proposed model. The graph learning first learns disease hidden features with two collaborative graphs: an observation graph and an ontology graph, based on the hierarchical embedding from medical domain knowledge. Then an RNN is designed to learn temporal information of visit sequences. Rectified Attention mechanism encodes clinical notes with the guide of TF-IDF and uses the visit representation as an attention context vector to integrate structured visit records and unstructured clinical notes.

child nodes for each non-leaf node to pad them into virtual leaf nodes. We assume there are $n_k$ nodes at each level $k$ (smaller $k$ means higher level in the hierarchical structure).

We create an embedding tensor $\{\mathbf{E}_k\}_{k\in[1,2,\ldots,K]}$ for nodes in the tree. $\mathbf{E}_k \in \mathbb{R}^{n_k \times d_c}$ is the embedding matrix for nodes in level $k$, and $d_c$ is the embedding size. For a medical code $c_i$ as a leaf node, we first identify its ancestors in each level $k = [1, 2, \ldots, K-1]$ in the tree and select corresponding embedding vectors from $\{\mathbf{E}_k\}$. Then, the hierarchical embedding $\mathbf{e}_i \in \mathbb{R}^{Kd_c}$ of $c_i$ is calculated by concatenating the embeddings in each level: $\mathbf{e}_i = \mathbf{E}_1^i \oplus \mathbf{E}_2^i \ \oplus \ ,\ldots, \oplus\, \mathbf{E}_K^i$, where $\oplus$ denotes the concatenation. We use $\mathbf{E} \in \mathbb{R}^{|\mathcal{C}|\times Kd_c}$ to represent medical codes after hierarchical embedding.

## Graph Representation

In visit records, specific diagnosis co-occurrences could reveal hidden similarities of patients and diseases. We explore such relationship by making the following hypotheses:

1. **Diagnostic similarity of patients**. If two patients get diagnosed with the same diseases, they tend to have diagnostic similarities and get similar diagnoses in the future.
2. **Medical similarity of diseases**. If two diseases belong to the same higher-level disease, they might have medical similarities such as symptoms, causes, and complications.

Based on these hypotheses, we construct a collaborative graph $\mathcal{G} = \{\mathcal{G}_{\mathcal{UC}}, \mathcal{G}_{\mathcal{CC}}\}$ for patients and medical codes. $\mathcal{G}_{\mathcal{UC}}$ is the patient-disease ***observation graph*** built from EHR data. Its nodes are patients and medical codes. We use a patient-code adjacency matrix $\mathbf{A}_{\mathcal{UC}} \in \{0,1\}^{|\mathcal{U}|\times|\mathcal{C}|}$ to represent $\mathcal{G}_{\mathcal{UC}}$. Given patient $u$, if $u$ is diagnosed with a code $c_i$ in a previous visit, we add an edge $(u, c_i)$ and set $\mathbf{A}_{\mathcal{UC}}[u][i] = 1$. $\mathcal{G}_{\mathcal{CC}}$ is the ***ontology graph***. Its nodes are medical codes. To model horizontal links of two medical codes (leaf nodes), we create a code-code adjacency matrix $\mathbf{A}'_{\mathcal{CC}} \in \mathbb{N}^{|\mathcal{C}|\times|\mathcal{C}|}$. If two medical codes $c_i$ and $c_j$ have their lowest common ancestor in level $k$, we add an ontology edge $(c_i, c_j)_k$ and set $\mathbf{A}'_{\mathcal{CC}}[i][j] = k$. This process is based on the idea that two medical codes with a common ancestor in lower levels of the hierarchial graph of ICD-9-CM should be similar diseases. Finally, we set $\mathbf{A}'_{\mathcal{CC}}[i][i] = 0$ for all diagonal elements. Although $\mathbf{A}'_{\mathcal{CC}}$ can reflect the hierarchical structure of medical codes, it is a dense matrix and generates a nearly complete ontology graph, which will cause a high complexity for graph learning. We further propose a disease co-occurrence indicator matrix $\mathbf{B}_{\mathcal{CC}}$ initialized with all zeros. If two medical codes $c_i$ and $c_j$ appear in a patient's visit record, we set $\mathbf{B}_{\mathcal{CC}}[i][j]$ and $\mathbf{B}_{\mathcal{CC}}[j][i]$ as 1. Then, we let $\mathbf{A}_{\mathcal{CC}} = \mathbf{A}'_{\mathcal{CC}} \odot \mathbf{B}_{\mathcal{CC}}$ be a new adjacency matrix for $\mathcal{G}_{\mathcal{CC}}$ to neglect disease pairs in $\mathbf{A}'_{\mathcal{CC}}$ which never co-occur in EHR data. Here $\odot$ denotes element-wise multiplication. Finally, we not only create a sparse ontology graph for computational efficiency, but also focus on more common and reasonable disease connections in the ontology graph.

## Collaborative Graph Learning

To learn hidden features of medical codes and patients, we design a collaborative graph learning method on the fact and ontology graphs. Instead of calculating patient embeddings with medical codes like DeepPatient [Miotto *et al.*, 2016], we assign each patient an initial embedding. $\mathbf{P} \in \mathbb{R}^{|\mathcal{U}|\times d_p}$ is the embedding matrix of all patients with the size of $d_p$. Let $\mathbf{H}_p^{(0)} = \mathbf{P}, \mathbf{H}_c^{(0)} = \mathbf{E}$ and $\mathbf{H}_p^{(l)} \in \mathbb{R}^{|\mathcal{U}|\times d_p^{(l)}}, \mathbf{H}_c^{(l)} \in \mathbb{R}^{|\mathcal{C}|\times d_c^{(l)}}$ be the hidden features of patients and medical codes (i.e., inputs of $l$-th graph layer). We design a graph aggregation method to calculate the hidden features of patients and medical codes in the next layer. First, we map the medical code features $\mathbf{H}_c^{(l)}$ into the patient dimension and aggregate adjacent medical codes from the observation graph ($\mathbf{A}_{\mathcal{UC}}$) for each patient:

$$\mathbf{Z}_p^{(l)} = \mathbf{H}_p^{(l)} + \mathbf{A}_{\mathcal{UC}}\mathbf{H}_c^{(l)}\mathbf{W}_{\mathcal{CU}}{}^{(l)} \in \mathbb{R}^{|\mathcal{U}|\times d_p^{(l)}}. \tag{1}$$

Here $\mathbf{W}_{\mathcal{CU}}{}^{(l)} \in \mathbb{R}^{d_c^{(l)}\times d_p^{(l)}}$ maps code embeddings to patient embeddings. For the ontology graph, if $c_i$, $c_j$ are connected in level $k$, we assign an ontology weight $\phi_j$ to $c_j$ when aggregating $c_j$ into $c_i$:

$$\phi_j(k) = \sigma(\mu_j \times k + \theta_j). \tag{2}$$

Here $\sigma$ is the sigmoid function. $\mu_j, \theta_j \in \mathbb{R}$ are trainable variables for $c_j$. $\phi_j(k)$ is a monotonic function w.r.t. level $k$. This function enables the model to describe the horizontal influence of a disease on other diseases via assigning increasing or decreasing weights by levels. Let $\Phi \in \mathbb{R}^{|\mathcal{C}|\times|\mathcal{C}|}$ be the ontology weight matrix and $\mathbf{M}, \mathbf{\Theta} \in \mathbb{R}^{|\mathcal{C}|}$ be the collection of $\mu, \theta$. $\mathbf{H}_p^{(l)}$ is mapped into the medical code dimension and aggregated with adjacent patients from the observation graph:

$$\Phi = \sigma(\mathbf{M} \odot \mathbf{A}_{\mathcal{CC}} + \mathbf{\Theta}) \in \mathbb{R}^{|\mathcal{C}|\times|\mathcal{C}|}, \tag{3}$$

$$\mathbf{Z}_c^{(l)} = \mathbf{H}_c^{(l)} + \mathbf{A}_{\mathcal{UC}}^{\top}\mathbf{H}_p^{(l)}\mathbf{W}_{\mathcal{UC}}{}^{(l)} + \Phi\mathbf{H}_c^{(l)} \in \mathbb{R}^{|\mathcal{C}|\times d_c^{(l)}}. \tag{4}$$

Here $\mathbf{W}_{\mathcal{UC}} \in \mathbb{R}^{d_p^{(l)}\times d_c^{(l)}}$ maps patient embeddings to code embeddings. Given that $\mathbf{A}_{\mathcal{CC}}$ stores the level where two diseases are connected, we use $\mathbf{A}_{\mathcal{CC}}$ to compute $\Phi$. Finally, $\mathbf{H}_p^{(l)}$ and $\mathbf{H}_c^{(l)}$ of the next layer are calculated as follows:

$$\mathbf{H}_{\{p,c\}}^{(l+1)} = \mathrm{ReLU}\left(\mathrm{BatchNorm}\left(\mathbf{Z}_{\{p,c\}}^{(l)}\mathbf{W}_{\{p,c\}}^{(l)}\right)\right), \tag{5}$$

where $\mathbf{W}^{(l)}_{\{p,c\}}$ maps $\mathbf{Z}^{(l)}_{\{p,c\}}$ to the $(l+1)$-th layer, and we use batch normalization to normalize features. In the $L$-th graph layers, we do not calculate $\mathbf{H}_p^{(L)}$ and only calculate $\mathbf{H}_c^{(L)}$ as the graph output, since the medical codes are required for further calculation. We let $\mathbf{H}_c = \mathbf{H}_c^{(L)} \in \mathbb{R}^{|\mathcal{C}|\times d_c^{(L)}}$ be the final embedding for medical codes.

### Temporal Learning for Visits

Given a patient $u$, we first compute a embedding $\mathbf{v}_t$ for visit $t$:

$$\mathbf{v}_t = \frac{1}{\mid C_t \mid}\sum_{c_i\in C_t}\mathbf{H}_c^i \in \mathbb{R}^{d_c^{(L)}}. \tag{6}$$

After the collaborative graph learning, $\mathbf{H}_c^i$ contains the information of its multi-hop neighbor diseases by the connection of patient nodes. Hence, different from GRAM, it enables the model to effectively predict diseases that have never been diagnosed on a patient before. We then employ GRU on $\mathbf{v}_t$ to learn visit temporal features and get a hidden representation $\mathbf{R} = \{\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_T\}$ where the size of the RNN cell is $h$:

$$\mathbf{R} = \mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_T = \text{GRU}(\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_T) \in \mathbb{R}^{T\times h}, \tag{7}$$

Then we apply a location-based attention [Luong *et al.*, 2015] to calculate the final hidden representation $\mathbf{o}_v$ of all visits:

$$\boldsymbol{\alpha} = \text{softmax}\left(\mathbf{R}\mathbf{w}_\alpha\right) \in \mathbb{R}^T, \tag{8}$$

$$\mathbf{o}_v = \boldsymbol{\alpha}\mathbf{R} \in \mathbb{R}^h, \tag{9}$$

where $\mathbf{w}_\alpha \in \mathbb{R}^h$ is a context vector for attention and $\boldsymbol{\alpha}$ is the attention weight for each visit.

### Guiding Attention on Clinical Notes

We incorporate the clinical notes $N_T$ from the latest visit $V_T$, since $N_T$ generally contains the medical history and future plan for a patient. We propose an attention regulation strategy that automatically highlights key words, considering traditional attention mechanisms in NLP have raised concerns as a tool to audit a model [Jain and Wallace, 2019; Serrano and Smith, 2019]. Pruthi *et al.* [Pruthi *et al.*, 2020] present a manipulating strategy using a set of pre-defined impermissible tokens and penalizing the attention weights on these impermissible tokens. To implement the regulation strategy, we propose a TF-IDF-rectified attention method on clinical notes. Regarding all patients' notes as a corpus and each patient's note as a document, for a patient $u$, we first calculate the TF-IDF weight $\beta_i$ for each word $\omega_i$ in $u$'s note $N_T$ and normalize the weights into [0, 1]. Then, we select the embedding $\mathbf{q}_i \in \mathbb{R}^{d_w}$ from a randomly initialized word embedding matrix $\mathbf{Q} \in \mathbb{R}^{|\mathcal{N}|\times d_w}$. For attention in Eq. (8), the context vector $\mathbf{w}_\alpha$ is randomly initialized, while clinical notes are correlated with diagnoses. Therefore, we adopt $\mathbf{o}_v$ as the context vector. Firstly, we project word embeddings $\mathbf{Q}$ into the dimension of visits to multiply the context vector $\mathbf{o}_v$:

$$\mathbf{Q}' = \mathbf{Q}\mathbf{W}_q \in \mathbb{R}^{|\mathcal{N}|\times h} \tag{10}$$

Then, let $\mathbf{N}$ be the embedding matrix selected from $\mathbf{Q}'$ for words in $N_T$, we calculate the attention weight $\boldsymbol{\alpha}'$ as well as the output $\mathbf{o}_n$ for clinical notes:

$$\boldsymbol{\alpha}' = \text{softmax}\left(\mathbf{N}\mathbf{o}_v\right) \in \mathbb{R}^{|N_T|}, \tag{11}$$

$$\mathbf{o}_n = \boldsymbol{\alpha}'\mathbf{N} \in \mathbb{R}^h. \tag{12}$$

| | |
|---|---|
| Patient number | 7,125 |
| Avg. visit number per patient | 2.66 |
| Patient number with heart failure | 2,604 |
| Medical code (disease) number | 4,795 |
| Avg. code number per visit | 13.27 |
| Dictionary size in notes | 67,913 |
| Avg. word number per note | 4,732.28 |

Table 1: Statistics of the MIMIC-III dataset.

For a word with a high TF-IDF weight in a clinical note, we expect the model to focus on this word with a high attention weight. Therefore, we introduce a TF-IDF-rectified attention penalty $\mathcal{L}_0$ for the attention weights of words:

$$\mathcal{L}_0 = -\sum_{\omega_i\in N_T}\left(\alpha_i'\log\beta_i + (1-\alpha_i')\log\left(1-\beta_i\right)\right). \tag{13}$$

The attention weights that mismatch the TF-IDF weights will be penalized. We believe that irrelevant (impermissible) words such as "patient" and "doctor" tend to have low TF-IDF weights. Finally, we concatenate $\mathbf{o}_n$ and $\mathbf{o}_v$ as the output $\mathbf{O} \in \mathbb{R}^{2h}$ for patient $u$: $\mathbf{O} = \mathbf{o}_v \oplus \mathbf{o}_n$.

### Prediction and Inference

Diagnosis prediction is a multi-label classification task, while heart failure prediction is a binary classification task. We both use a dense layer with a sigmoid activation function on the model output $\mathbf{O}$ to calculate the predicted probability $\hat{\mathbf{y}}$. The loss function of classification for both tasks is cross-entropy loss $\mathcal{L}_c$. Then, we combine the TF-IDF-rectified penalty $\mathcal{L}_0$ and cross-entropy loss as the final loss $\mathcal{L}$ to train the model:

$$\mathcal{L} = \lambda\mathcal{L}_0 + \text{CrossEntropy}(\hat{\mathbf{y}}, \mathbf{y}). \tag{14}$$

Here, $\mathbf{y}$ is the ground-truth label of medical codes or heart failure, and $\lambda$ is a coefficient to adjust $\mathcal{L}_0$. In the inference phase, we freeze the trained model and retrieve the embeddings $\mathbf{H}_c$ of medical codes at the output of heterogeneous graph learning. Then, given a new patient for inference, we continue from Eq. (6) and make predictions.

# 4 Experiments

## 4.1 Experimental Setup

### Dataset Description

We use the MIMIC-III dataset [Johnson *et al.*, 2016] to evaluate CGL. Table 1 shows the basic statistics of MIMIC-III. We select patients with multiple visits (# of visits $\geq 2$) and select clinical notes except the type of "Discharge summary", since it has a strong indication to predictions and is unfair to be used as features. For each note, we use the first 50,000 words, while the rest are cut off for computational efficiency, given the average word number per note is less than 5,000. We split MIMIC-III randomly according to patients into training/validation/test sets with patient numbers as 6000/125/1000. We use the codes in patients' last visit as labels and other visits as features. For heart failure prediction, we set labels as 1 if patients are diagnosed with heart failure in the last visit. Finally, the observation graph is built based on the training set. A 5-level hierarchical structure and the ontology graph are built according to ICD-9-CM.

| **Models** | **w-$F_1$** (%) | **R@20** (%) | **R@40** (%) | **Param.** |
|---|---|---|---|---|
| RETAIN | 19.66 (0.58) | 33.90 (0.47) | 42.93 (0.39) | 2.90M |
| Deepr | 12.38 (0.01) | 28.15 (0.08) | 37.26 (0.14) | 0.80M |
| GRAM | 21.06 (0.19) | 36.37 (0.16) | 45.61 (0.27) | 1.38M |
| Dipole | 11.24 (0.19) | 26.96 (0.15) | 36.83 (0.26) | 2.08M |
| Timeline | 16.83 (0.62) | 32.08 (0.66) | 41.97 (0.74) | 1.23M |
| MedGCN | 20.93 (0.25) | 35.69 (0.50) | 43.36 (0.46) | 4.59M |
| LR$_{\text{notes}}$ | 17.56 (0.41) | 36.71 (0.28) | 46.02 (0.38) | 325.65M |
| CGL | **22.97 (0.19)** | **38.19 (0.16)** | **48.26 (0.15)** | 3.55M |

Table 2: Diagnosis prediction results in w-$F_1$ and R@$k$.

| **Models** | **AUC** (%) | **$F_1$** (%) | **Param.** |
|---|---|---|---|
| RETAIN | 82.73 (0.21) | 71.12 (0.37) | 1.67M |
| Deepr | 81.29 (0.01) | 68.42 (0.01) | 0.49M |
| GRAM | 82.82 (0.06) | 71.43 (0.05) | 0.76M |
| Dipole | 81.66 (0.07) | 70.01 (0.04) | 1.45M |
| Timeline | 80.75 (0.46) | 69.81 (0.34) | 0.95M |
| MedGCN | 81.25 (0.15) | 70.86 (0.18) | 3.98M |
| LR$_{\text{notes}}$ | 80.33 (0.12) | 69.18 (0.27) | 0.07M |
| CGL | **85.66 (0.19)** | **72.68 (0.22)** | 1.62M |

Table 3: Heart failure prediction results in AUC and $F_1$.

| **Models** | **Diagnosis** | | | **Heart failure** | | |
|---|---|---|---|---|---|---|
| | w-$F_1$ | R@20 | Param. | AUC | $F_1$ | Param. |
| CGL$_{h\text{-}}$ | 20.87 | 35.66 | 3.98M | 82.58 | 71.02 | 2.04M |
| CGL$_{n\text{-}}$ | 22.10 | 37.59 | 1.50M | 84.53 | 71.96 | 0.53M |
| CGL$_{w\text{-}}$ | 22.06 | 37.31 | 3.54M | 83.91 | 71.59 | 1.60M |
| CGL | **22.97** | **38.19** | 3.55M | **85.66** | **72.68** | 1.62M |

Table 4: w-$F_1$, R@20 of diagnosis prediction and AUC, $F_1$ of heart failure prediction for CGL variants. CGL$_{h\text{-}}$: no hierarchical embedding; CGL$_{n\text{-}}$: no clinical notes; CGL$_{w\text{-}}$: no ontology weights.

### Evaluation Metrics

We adopt weighted $F_1$ score (w-$F_1$ [Bai *et al.*, 2018]) and top $k$ recall (R@$k$ [Choi *et al.*, 2016a]) for diagnosis predictions. w-$F_1$ is a weighted sum of $F_1$ for each class. R@$k$ is the ratio of true positive numbers in top $k$ predictions by the total number of positive samples, which measures the prediction performance on a subset of classes. For heart failure predictions, we use $F_1$ and the area under the ROC curve (AUC), since it is a binary classification on imbalanced test data.

### Baselines

To compare CGL with state-of-the-art models, we select the following models as baselines: 1) *RNN-based models*: RETAIN [Choi *et al.*, 2016b], Dipole [Ma *et al.*, 2017], Timeline [Bai *et al.*, 2018]; 2) *CNN-based models*: Deepr [Nguyen *et al.*, 2017]; 3) Graph-based models: GRAM [Choi *et al.*, 2017], MedGCN [Mao *et al.*, 2019]; and 4) *A logistic regression model*, LR$_{\text{notes}}$, on clinical notes using only TF-IDF features of each note (whose dimension is the dictionary size).

Deepr, GRAM, and Timeline use medical code embeddings as inputs, while others use multi-hot vectors of medical codes. We do not consider SMR [Wang *et al.*, 2017] because 1) it does not compare with the above state-of-the-art models and 2) it focuses on medication recommendation which is different from our tasks. We also do not compare with MiME [Choi *et al.*, 2018] and GCT [Choi *et al.*, 2020] because we do not use treatments and lab results in our data.

### Parameters

We randomly initialize embeddings for diseases, patients, and clinical notes and select the paramters by a grid search. The embedding sizes $d_c$, $d_p$, and $d_w$ are 32, 16, and 16. The graph layer number $L$ is 2. The hidden dimensions $d_p^{(1)}$, $d_c^{(1)}$, and $d_c^{(2)}$ are 32, 64, and 128, and the GRU unit $h$ is set to 200. The coefficient $\lambda$ in $\mathcal{L}_0$ for diagnosis and heart failure prediction is 0.3 and 0.1. We set the learning rate as $10^{-3}$, optimizer as Adam, and use 200 epochs for training. The source code of CGL is released at https://github.com/LuChang-CS/CGL.

## 4.2 Experimental Results

### Diagnosis and Heart Failure Prediction

Table 2 shows the results of baselines and CGL on diagnosis prediction. We use $k = [20, 40]$ for R@$k$. Each model is trained for 5 times with different variable initializations. The mean and standard deviation are reported. The proposed CGL model outperforms all the baselines. We think this is mostly because CGL captures hidden connections of patients and diseases and utilizes clinical notes. In addition, the results of LR$_{\text{notes}}$ indicate that only using clinical notes does not improve performance in predicting diagnosis. Table 3 shows the heart failure prediction results. We observe that CGL also achieves the best performance in terms of AUC and $F_1$.

### Ablation Study

To study the effectiveness of components, we also compare 3 CGL variants: CGL without hierarchical embedding (CGL$_{h\text{-}}$), CGL without clinical notes as inputs (CGL$_{n\text{-}}$), and CGL without ontology weights (CGL$_{w\text{-}}$). The results are shown in Table 4. We observe that even without clinical notes, CGL$_{n\text{-}}$ with hierarchical embeddings and ontology weights still achieves the best performance among all other baselines. This indicates that domain knowledge including hierarchical embeddings and ontology weights also help to learn better representations of medical codes. In addition, from Table 4 we can infer that the complexity of CGL mostly comes from modeling clinical notes, i.e., word embeddings. Therefore, CGL is scalable and can be generalized to other tasks when clinical notes are not accessible.

### Prediction Analysis

**New-onset diseases.** For a patient, new-onset diseases denote new diseases in future visits which have not occurred in previous visits of this patient. We use the ability of predicting new-onset diseases to measure learned diagnostic similarity of patients. It is natural for a model to predict diseases that have occurred in previous visits. With the help of other similar patients' records, the model should be able to predict new diseases for a patient. The idea is similar to collaborative filtering in recommender systems. If two patients are similar, one of them may be diagnosed with new-onset diseases which have occurred in the other patient. We also use R@$k$ ($k = [20, 40]$) to evaluate the ability of predicting occurred and new-onset diseases. Here, R@$k$ denotes the ratio between the number of correctly predicted occurred (or new)

| Models | Occurred | | New-onset | |
|---|---|---|---|---|
| | R@20 | R@40 | R@20 | R@40 |
| GRAM | 21.05 | 23.11 | 15.32 | 22.50 |
| MedGCN | 20.51 | 21.89 | 15.38 | 21.53 |
| CGL$_{n\text{-}}$ | **21.26** | **23.85** | **16.33** | **23.58** |

Table 5: R@$k$ of predicting occurred/new-onset diseases.

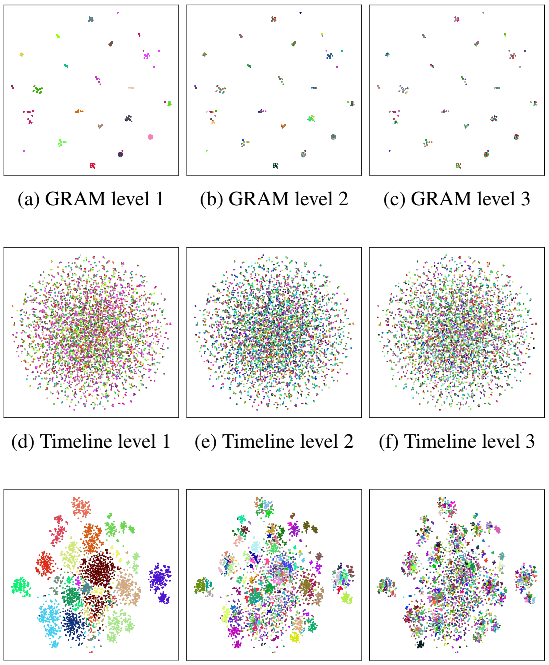

(a) GRAM level 1
(b) GRAM level 2
(c) GRAM level 3
(d) Timeline level 1
(e) Timeline level 2
(f) Timeline level 3
(g) CGL level 1
(h) CGL level 2
(i) CGL level 3

Figure 3: Code embeddings in 3 levels learned by GRAM, Timeline, and CGL. Colors correspond to disease types in each level.

diseases and the number of ground-truth diseases. We select GRAM and MedGCN which have good performance in diagnosis prediction, and CGL$_{n\text{-}}$ without clinical notes, because we want to explore the effectiveness of the proposed observation and ontology graphs. Table 5 shows the results of R@$k$ on test data. We can see that CGL$_{n\text{-}}$ has similar results to GRAM on occurred diseases while achieving superior performance on new-onset diseases. This verifies that our proposed collaborative graph learning is able to learn from similar patients and predict new-onset diseases in the future.

**Disease embeddings.** To show the similarity of diseases, we plot the learned 4795 code embeddings $\mathbf{H}_c$ using t-SNE [Maaten and Hinton, 2008]. Figure 3 shows the embeddings learned by GRAM, Timeline, and CGL in 3 levels. Colors denotes different disease types in each level. In Figure 3, disease embeddings learned by GRAM and CGL are basically clustered according to their real categories, while Timeline seems like a random distribution. In the plot of GRAM, we observe the clusters are far away from each other given large inter-cluster distances, while nodes in a cluster are close to each other due to small intra-cluster distances. We can observe that the embeddings learned by GRAM do not capture distinguishable features of low-level diseases as well as the relationships across clusters. Therefore, we can infer that learning proper representations that reflect disease hierarchical structures and correlations is helpful for predictions.

| Without penalty | With penalty | Correct Predictions |
|---|---|---|
| ... Patient had fairly acute decompensation of respiratory status today with hypoxia and hypercarbia associated with hypertension ... Differential diagnosis includes flash pulmonary edema and acute exacerbation of CHF vs aspiration vs infection (HCAP) ... Acuity suggests possible flash pulmonary edema vs aspiration ... | ... Patient had fairly acute decompensation of respiratory status today with hypoxia and hypercarbia associated with hypertension ... Differential diagnosis includes flash pulmonary edema and acute exacerbation of CHF vs aspiration vs infection (HCAP) ... Acuity suggests possible flash pulmonary edema vs aspiration ... | • Hypertensive chronic kidney disease<br>• Acute respiratory failure<br>• Congestive heart failure<br>• Diabetes<br>• ... |

Table 6: An example of word contributions without/with the TF-IDF rectified penalty. The pink/gray color denotes high/low attention weights.

**Contribution of notes.** We compare the proposed TF-IDF rectified attention weights with regular attention weights to verify if the model focuses on important words. Table 6 demonstrates an example with a part of a note and predicted diagnoses. In this example, the patient is diagnosed with 33 diseases, and CGL predicts 10 of them correctly in top 20 predicted codes. Important words with high $\alpha'$ values are highlighted in pink. We first observe that pink words are relevant to diagnoses. In addition, we notice the rectified attention weights are more semantically interpretable. For example, "acute" and "HCAP" (Health care-associated pneumonia) get higher weights with the rectified attention loss. Meanwhile, we show the unimportant words with low $\alpha'$ values in gray. We observe that our model detects unimportant words which have less contributions. For example, "patient" and "diagnosis" are regarded as an unimportant word but not captured in the regular attention mechanism. Therefore, we may conclude that the TF-IDF-rectified attention method improves the accuracy of interpretations using clinical notes.

## 5 Conclusion

In this paper, we propose CGL, a collaborative graph learning model to jointly learn the representations of patients and diseases, and effectively utilize clinical notes in EHR data. We conducted experiments on real-world EHR data to demonstrate the effectiveness of the learned representations and performance improvements of CGL over state-of-the-art models. We also provide analysis of CGL on multiple aspects, including new onset diseases, disease embeddings, and contribution of clinical notes. In the future, we plan to explore methods to quantify the contributions of certain admissions to each predicted medical code. Usage of single admission records in EHR data will also be considered for further investigation.

## Acknowledgments

This work was supported in part by US National Science Foundation under grants 1838730 and 1948432. SK was supported in part by the NLM of the NIH under Award Number R01LM013308.

## References

[Bai *et al.*, 2018] Tian Bai, Shanshan Zhang, Brian L Egleston, and Slobodan Vucetic. Interpretable representation learning for healthcare via capturing disease progression through time. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 43–51. ACM, 2018.

[Choi *et al.*, 2016a] Edward Choi, Mohammad Taha Bahadori, Andy Schuetz, Walter F Stewart, and Jimeng Sun. Doctor ai: Predicting clinical events via recurrent neural networks. In *Machine Learning for Healthcare Conference*, pages 301–318, 2016.

[Choi *et al.*, 2016b] Edward Choi, Mohammad Taha Bahadori, Jimeng Sun, Joshua Kulas, Andy Schuetz, and Walter Stewart. Retain: An interpretable predictive model for healthcare using reverse time attention mechanism. In *Advances in Neural Information Processing Systems*, pages 3504–3512, 2016.

[Choi *et al.*, 2017] Edward Choi, Mohammad Taha Bahadori, Le Song, Walter F Stewart, and Jimeng Sun. Gram: graph-based attention model for healthcare representation learning. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 787–795, 2017.

[Choi *et al.*, 2018] Edward Choi, Cao Xiao, Walter Stewart, and Jimeng Sun. Mime: Multilevel medical embedding of electronic health records for predictive healthcare. In *Advances in neural information processing systems*, pages 4547–4557, 2018.

[Choi *et al.*, 2020] Edward Choi, Zhen Xu, Yujia Li, Michael W. Dusenberry, Gerardo Flores, Yuan Xue, and Andrew M. Dai. Learning the graphical structure of electronic health records with graph convolutional transformer. In *Proceedings of the 34th Conference on Association for the Advancement of Artificial Intelligence*, 2020.

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

[Johnson *et al.*, 2016] Alistair EW Johnson, Tom J Pollard, Lu Shen, H Lehman Li-wei, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3:160035, 2016.

[Luong *et al.*, 2015] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, 2015.

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

[Ma *et al.*, 2020a] Liantao Ma, Junyi Gao, Yasha Wang, Chaohe Zhang, Jiangtao Wang, Wenjie Ruan, Wen Tang, Xin Gao, and Xinyu Ma. Adacare: Explainable clinical health status representation learning via scale-adaptive feature extraction and recalibration. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 825–832, 2020.

[Ma *et al.*, 2020b] Liantao Ma, Chaohe Zhang, Yasha Wang, Wenjie Ruan, Jiangtao Wang, Wen Tang, Xinyu Ma, Xin Gao, and Junyi Gao. Concare: Personalized clinical feature embedding via capturing the healthcare context. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence*, AAAI, pages 833–840. AAAI Press, 2020.

[Maaten and Hinton, 2008] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[Mao *et al.*, 2019] Chengsheng Mao, Liang Yao, and Yuan Luo. Medgcn: Graph convolutional networks for multiple medical tasks. arXiv Preprint https://arxiv.org/abs/1904.00326, 2019.

[Miotto *et al.*, 2016] Riccardo Miotto, Li Li, Brian A Kidd, and Joel T Dudley. Deep patient: an unsupervised representation to predict the future of patients from the electronic health records. *Scientific reports*, 6:26094, 2016.

[Nguyen *et al.*, 2017] Phuoc Nguyen, Truyen Tran, Nilmini Wickramasinghe, and Svetha Venkatesh. `Deepr`: A convolutional net for medical records. *IEEE Journal of Biomedical and Health Informatics*, 21(1):22–30, 2017.

[Pruthi *et al.*, 2020] Danish Pruthi, Mansi Gupta, Bhuwan Dhingra, Graham Neubig, and Zachary C. Lipton. Learning to deceive with attention-based explanations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4782–4793, Online, July 2020. Association for Computational Linguistics.

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

[Shang *et al.*, 2019] Junyuan Shang, Tengfei Ma, Cao Xiao, and Jimeng Sun. Pre-training of graph augmented transformers for medication recommendation. *arXiv preprint arXiv:1906.00346*, 2019.

[Wang *et al.*, 2017] Meng Wang, Mengyue Liu, Jun Liu, Sen Wang, Guodong Long, and Buyue Qian. Safe medicine recommendation via medical knowledge graph embedding. *arXiv preprint arXiv:1710.05980*, 2017.

```
Algorithm 1: Embedding($\mathcal{C}, K$)
Input  : The medical code set $\mathcal{C}$; level number $K$ of
         the ICD-9-CM structure
Output: Medical code embedding $\mathbf{E}$
for $k \leftarrow 1$ to $K$ do
    $\mathbf{E}_k \leftarrow$ initialize embedding for codes in level $k$
end
$\mathbf{E} \leftarrow$ blank embedding matrix for leaf nodes
for $i \leftarrow 1$ to $|\mathcal{C}|$ do
    $\mathbf{e}_i \leftarrow \mathbf{e}_i = \mathbf{E}_1^i \oplus \mathbf{E}_2^i \ \oplus \ , \ldots, \oplus \mathbf{E}_K^i$
    $\mathbf{E}_i \leftarrow \mathbf{e}_i$
end
return $\mathbf{E}$
```

```
Algorithm 2: GraphLearning($\mathbf{E}, \mathbf{A}_{\mathcal{UC}}, \mathbf{A}_{\mathcal{CC}}, L$)
Input  : The medical code embedding $\mathbf{E}$; adjacency
         matrices $\mathbf{A}_{\mathcal{UC}}, \mathbf{A}_{\mathcal{CC}}$ of the observation and
         ontology graph; layer number $L$
Output: Final medical code embedding $\mathbf{H}_c$
$\mathbf{H}_p^{(0)} \leftarrow \mathbf{P}$
$\mathbf{H}_c^{(0)} \leftarrow \mathbf{E}$
for $l \leftarrow 0$ to $L-1$ do
    $\mathbf{Z}_p^{(l)} \leftarrow$ aggregate neighbors ($\mathbf{H}_c^{(l)}$) of patient
      nodes on the observation graph
    $\Phi \leftarrow$ calculate the ontology weight using Eq. (3)
    $\mathbf{Z}_c^{(l)} \leftarrow$ aggregate neighbors ($\mathbf{H}_p^{(l)}, \Phi\mathbf{H}_c^{(l)}$) of code
      nodes on the observation and ontology graph
    $\mathbf{H}_{\{p,c\}}^{(l+1)} \leftarrow$ message passing from $\mathbf{Z}_{\{p,c\}}^{(l)}$
end
$\mathbf{Z}_c^{(L)} \leftarrow$ aggregate neighbors ($\mathbf{H}_p^{(L-1)}, \Phi\mathbf{H}_c^{(L-1)}$) of
  code nodes on the observation and ontology graph
$\mathbf{H}_c \leftarrow$ message passing from $\mathbf{Z}_c^{(L-1)}$
return $\mathbf{H}_c$
```

```
Algorithm 3: CGL-Training($\mathcal{D}, K, L$)
Input  : An EHR dataset $\mathcal{D}$; level number $K$ of the
         ICD-9-CM structure; graph layer number $L$
$\mathcal{C} \leftarrow$ medical code set of $\mathcal{D}$
$\mathcal{U} \leftarrow$ patient set of $\mathcal{D}$
$\mathbf{A}_{\mathcal{UC}}, \mathbf{A}_{\mathcal{CC}} \leftarrow$ build observation and ontology graphs
  from $\mathcal{D}$
$E \leftarrow$ Embedding($\mathcal{C}, K$)
$\mathbf{H}_c \leftarrow$ GraphLearning($\mathbf{E}, \mathbf{A}_{\mathcal{UC}}, \mathbf{A}_{\mathcal{CC}}, L$)
$\mathbf{Q} \leftarrow$ initialize word embeddings
repeat
    $u \leftarrow$ a patient in $\mathcal{U}$ with multiple admissions
    $\mathbf{y} \leftarrow$ ground-truth of $u$
    $C_1, \ldots, C_T \leftarrow$ diagnoses of $u$'s all $T$ visits
    $N_T \leftarrow$ notes of $u$'s $T$-th visit
    for $t \leftarrow 1$ to $T$ do
        $\mathbf{v}_t \leftarrow$ average embeddings of code in $C_t$
    end
    $\mathbf{R} \leftarrow \mathrm{GRU}(\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_T)$
    $\mathbf{o}_v \leftarrow$ visit output by attention on $\mathbf{R}$
    $\mathbf{Q}' \leftarrow$ project $\mathbf{Q}$ to the dimension of visit
    $\mathbf{N} \leftarrow$ select word embeddings from $\mathbf{Q}'$ in $N_T$
    $\mathbf{o}_n \leftarrow$ note output by attention with $\mathbf{o}_v$ on $\mathbf{N}$
    $\mathbf{O} \leftarrow \mathbf{o}_v \oplus \mathbf{o}_n$
    $\hat{\mathbf{y}} \leftarrow$ predictions using $\mathbf{O}$
    $\mathcal{L}_0 \leftarrow$ TF-IDF-rectified loss of $\mathbf{N}$
    $\mathcal{L} = \lambda\mathcal{L}_0 + \mathrm{CrossEntropy}(\hat{\mathbf{y}}, \mathbf{y})$
    Optimizing the model using $\mathcal{L}$
until convergence
```

# A Pseudocode

The pseudo-code of hierarchical embedding, collaborative graph learning, and training CGL is described in Algorithm 1, 2, and 3, respectively.

# B Parameter Settings for Baselines

All models were implemented by Python 3.7.4 and Tensorflow 2.3.0 in a machine with Intel i9-9900K CPU, 64GB memory, and Geforce RTX 2080 Ti GPU with CUDA 10.1. The parameter settings for baselines are listed as follows:

- RETAIN [Choi *et al.*, 2016b]: RETAIN uses two RNNs with reversed time to calculate attention values of visits and predict heart failure in the next visit. The embedding size for visits is 256 and the hidden layer size for two RNN layers is 128.
- Deepr [Nguyen *et al.*, 2017]: Deepr uses the embedding of medical codes as input and uses a CNN to predict the readmission within 6 months. The embedding size for medical codes is 100. Kernel window size is 3. The number of kernels is 4. The classifier is a network with two dense layers and the size of the first layer is 64.
- GRAM [Choi *et al.*, 2017]: GRAM uses a knowledge DAG of medical codes to calculate their embeddings. An RNN layer is applied to learn the features of visits and predict medical codes and heart failure in the next admissions. The embedding size for graph nodes is 100, attention size is 100, and the size of RNN hidden layers is 128.
- Dipole [Ma *et al.*, 2017]: Dipole applies attention on a bi-directional RNN layer and predicts diagnoses in patients' next visits. The embedding size for visits is 256. Concatenation-based attention size is 128. The hidden layer size for the RNN layer is 128.
- Timeline [Bai *et al.*, 2018]: Timeline calculates the weight of codes using time duration and predicts medical codes for patients' next visits. It uses an attention layer followed by a bi-directional RNN layer. The embedding size for medical codes is 100. The attention layer size is 100. The hidden layer size for RNN is 128.
- MedGCN [Mao *et al.*, 2019]: MedGCN constructs a heterogeneous graph for patients, visits, medicines, and lab results. The hidden units of a graph layer are 128. The dropout rate is 0.1.
- $\text{LR}_{\text{notes}}$: The input of $\text{LR}_{\text{notes}}$ is the TF-IDF feature vector of each note. The input dimension is the dictionary

| No. | Medical Code | Disease | Visit Number | Visit Percentage |
|---|---|---|---|---|
| 1 | 401.9 | Unspecified essential hypertension | 7,168 | 36.03% |
| 2 | 428.0 | Congestive heart failure, unspecified | 6,580 | 33.08% |
| 3 | 427.31 | Atrial fibrillation | 5,279 | 26.54% |
| 4 | 584.9 | Acute kidney failure, unspecified | 4,312 | 21.67% |
| 5 | 414.01 | Coronary atherosclerosis of native coronary artery | 4,176 | 20.99% |
| 6 | 250.00 | Diabetes mellitus without mention of complication, type II or unspecified type, not stated as uncontrolled | 3,734 | 18.77% |
| 7 | 272.4 | Other and unspecified hyperlipidemia | 3,257 | 16.37% |
| 8 | 518.81 | Acute respiratory failure | 3,073 | 15.45% |
| 9 | 599.0 | Urinary tract infection, site not specified | 2,956 | 14.86% |
| 10 | 530.81 | Esophageal reflux | 2,621 | 13.17% |

Table 7: Disease distributions in MIMIC-III.

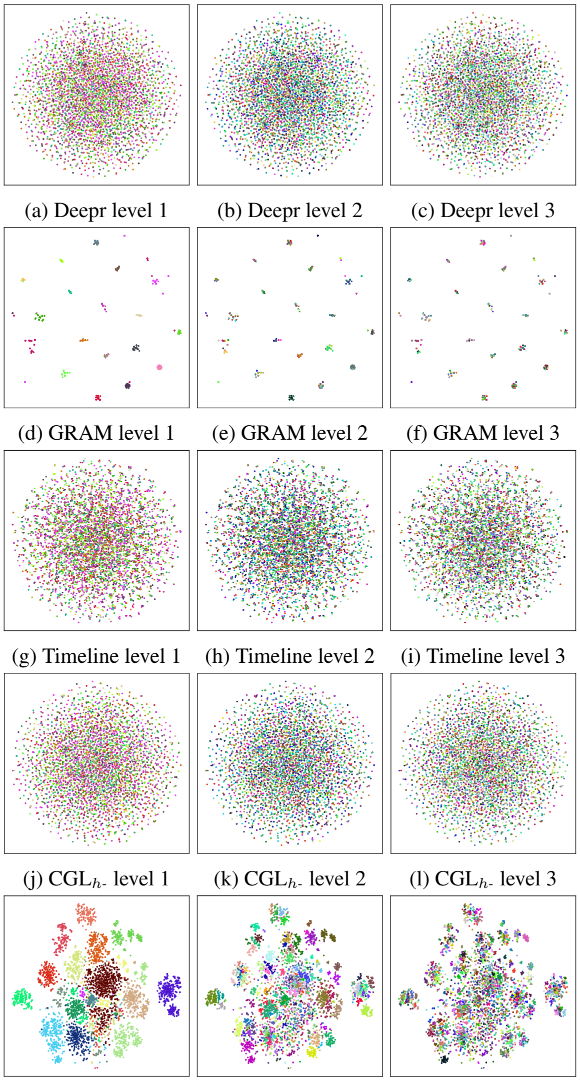
size, i.e. 67,913. For a word that occurs in each note, the corresponding element in the feature vector is the TF-IDF value of this word. For other words in the dictionary that do not occur in this note, corresponding elements are set to 0. $\text{LR}_{\text{notes}}$ uses one dense layer to predict 4,795 diseases in the diagnosis prediction and predict 1 disease in the heart failure prediction. Therefore, the parameter numbers of $\text{LR}_{\text{notes}}$ in two tasks are 325.65M and 0.07M, respectively.

## C Disease Distribution of MIMIC-III

We report disease distributions, i.e., the number of visits containing a disease, of the MIMIC-III dataset in Table 7. We selected top 10 commonest diseases appearing in patients' visits.

## D Experimental Results on Disease Embeddings

We further select a CNN-based model: Deepr, a graph-based model: GRAM, an RNN-based model: Timeline, and CGL without hierarchical embeddings: $\text{CGL}_{h\text{-}}$ to compare the learned disease embeddings with CGL. Figure 4 demonstrates the learned embeddings of these models. It shows that the embeddings leaned by Deepr, Timeline, and $\text{CGL}_{h\text{-}}$ which do not utilize the hierarchical structure of medical codes tend to be random distributions. Combining the results of Table 2 and 4, we can further infer that effective utilization of the hierarchical structure is able to help models to make predictions.


(a) Deepr level 1 (b) Deepr level 2 (c) Deepr level 3

(d) GRAM level 1 (e) GRAM level 2 (f) GRAM level 3

(g) Timeline level 1 (h) Timeline level 2 (i) Timeline level 3

(j) $\text{CGL}_{h\text{-}}$ level 1 (k) $\text{CGL}_{h\text{-}}$ level 2 (l) $\text{CGL}_{h\text{-}}$ level 3

(m) CGL level 1 (n) CGL level 2 (o) CGL level 3

Figure 4: Scatter plot in three levels of code embeddings learned by Deepr, GRAM, Timeline, $\text{CGL}_{h\text{-}}$, and CGL. Colors correspond to disease types in each level.