DUGRA: Dual-Graph Representation Learning for Health Information Networks

Qifan Wang
School of Computer Science
McGill University
Montreal, QC, Canada H3A 2A7
Email: qifan.wang2@mail.mcgill.ca

Benjamin C. M. Fung
School of Information Studies
McGill University
Montreal, QC, Canada H3A 1X1
Email: ben.fung@mcgill.ca

Patrick C. K. Hung
Faculty of Business and Information Technology
Ontario Tech University
Oshawa, ON, Canada L1G 0C5
Email: patrick.hung@uoit.ca

***Abstract*—With the rapidly growing volume and variety of Electronic Health Records (EHR) data, deep-learning models exhibit state-of-the-art performance for many predictive tasks in the health domain. To overcome the challenge of high dimensionality in EHR data, many representation learning methods have been proposed to learn low-dimensional diagnosis representations. Another challenge is how to effectively incorporate the domain knowledge, such as the International Classification of Diseases (ICD) medical ontology, into the learned embeddings. Albeit the medical ontology is a knowledge graph, none of the existing methods take advantage of Graph Neural Network (GNN), which has demonstrated its ability in other domains. The problem is that a GNN with multiple hidden layers, which are required to propagate information from the leaf of the medical ontology graph to the root, dilutes the differences among the nodes, degrading the quality of the learned embeddings. In this paper we introduce a densely connected graph derived from the original ontology graph to tackle the problem. Furthermore, to model the information in patient records, we construct a single co-occurrence graph based on the co-occurrence of diagnoses and a patient's diagnosis history. Experimental results show that the diagnosis embeddings learned from our model, *DUal-GRAph Representation Learning (DUGRA)*, outperform the current state-of-the-art models in terms of diagnosis prediction accuracy.**

***Index Terms*—Electronic health records, representation learning, knowledge graph, graph neural networks.**

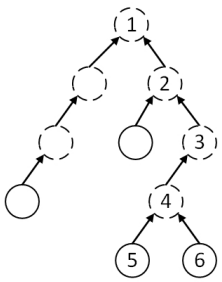
I. Introduction

Due to the extensive adoption of health information technology in recent years, more and more Electronic Health Records (EHR) data are becoming available, leading to increasing usage of machine learning methods to predict the health status of patients. The primary challenge of performing machine learning on EHR data comes from its variety and high dimensionality composed of thousands of diseases, medications, treatments, lab test results, etc. Considerable effort has been made to perform predictive tasks based on learning low-dimensional diagnosis representations or embeddings from EHR data, including prediction of a patient's health status [1], [2], [3], [4], readmission prediction [5], [6], [7], mortality prediction [8], [9], [10], etc. The medical history of each patient in an EHR system consists of a sequence of medical events, so an EHR data can be considered as sequence data. Due to the high similarity between EHR sequence data and the corpus data in natural language processing, many representation learning methods in NLP have been deployed in the field of health informatics. Among them, the well-adopted word2vec method [11] inspires many EHR machine learning works to automatically learn low-dimensional representations of diagnoses through the co-occurrence information in order to make frequently co-occurring diagnoses close in the embedding space.

Despite promising results given by the models inspired by the word2vec method, these models require a large amount of training data, which is difficult to find in the healthcare domain due to privacy concerns [2], [3], [4]. Also, most of these works do not utilize the domain background knowledge to increase the representation power. To tackle these problems, new methods [2], [3], [4] that incorporate knowledge graphs have been recently proposed. They use the graph to find the ancestor of a given diagnosis node in order to partially form the final representation as a convex combination of its ancestors' and its own feature vectors. Although some works [2] claim that they use knowledge graphs, none of them takes advantage of the newly emerging *graph neural network* (*GNN*), which has demonstrated its ability in knowledge graphs representation learning [12].

Graph Neural Network (GNN) is an effective way to learn node or graph representation from graph structure data [13]. The input of a GNN is a graph where each node is a representation without considering its neighbor information. The output of a GNN is a graph where each node is a representation with the consideration of its neighbors, i.e., the contextual information of the graph. In this paper we propose a new representation learning model, ***DU**al-**GRA**ph Representation Learning (DUGRA)*, to generate two embeddings of each diagnosis. The first one comes from the medical ontology graph, such as an International Classification of Diseases ICD-9 taxonomy in Figure 1 extracted from the Clinical Classifications Software (CCS) from the Agency for Healthcare Research and Quality (AHRQ), which captures the domain knowledge. The second one comes from the co-occurrence information of diagnosis codes in EHR data. Both are modelled as graphs, so we use a GNN to learn each embedding and concatenate them to form a joint embedding.

GNN for Medical Ontology Graph. A *medical ontology graph* can be considered a knowledge graph, and it can be modelled by GNN. GNN was originally proposed to solve



Index	ICD-9 code	Description
1	root	Root
2	240-279	Endocrine, nutritional and metabolic diseases
3	250	Diabetes mellitus
4	2500	Diabetes mellitus without mention of complication
5	25001	Type 1 diabetes mellitus without mention of complication
6	25002	Type 2 diabetes mellitus without mention of complication

Fig. 1. Graph illustration of ICD-9 taxonomy

the semi-supervised classification problem. Recent work [14] has further shown that GNN can achieve the same result as Laplacian smoothing, where feature vectors of nodes become more similar as the number of layers increases in the GNN, resulting in better prediction accuracy. However, the aim of applying representation learning in the medical ontology graph is to better capture the relationship information between parent and children diagnosis nodes and improve differentiation of various diagnosis nodes. Therefore, over-smoothing is not desirable as it dilutes the differences among the nodes, leading to worse prediction accuracy, as illustrated in our experiments.

Due to the aforementioned drawbacks of GNN, we argue that using the raw medical ontology graph as input to GNN is not ideal for the task of representation learning. Also, the number of layers in the GNN should be small in order to avoid over-smoothing. In this paper, we introduce a densely connected graph derived from the raw medical ontology graph such that every pair of connected nodes in the densely connected graph is only 1-hop away. Although this connection schema allows us to propagate information without the over-smoothing problem, it removes important structural information from the graph. To compensate for the loss of structural information, we design two attention-based aggregate functions to compress the ancestor relationships and embed them into the representation of the child nodes.

GNN for Co-occurrence Graph. To model a diagnosis, it is crucial to consider its frequently co-occurred diagnosis in patient records. Thus, we construct a *co-occurrence graph* from an EHR dataset where patients and diagnoses are nodes; the edge weight between two diagnosis nodes represents their degree of co-occurrences, and the edge weight between a diagnosis node and a patient node represents the weighted frequency of the patient having such a diagnosis in their medical record. There is no edge between two patient nodes because most EHR datasets do not capture the relationship information among patients. To further enhance the co-occurrence embedding we use an auxiliary self-prediction task to guide the learning process of the co-occurrence GNN. This helps the model perform better in downstream prediction tasks, such as the next admission diagnoses prediction.

The contributions of this paper can be summarized as follows:

- This is the first work that utilizes a GNN to model the co-occurrence information of diagnosis codes in EHR data. Our experiments demonstrate that using the GNN-based co-occurrence graph can improve the predictive power of the learned embeddings, especially for the primary care patients.
- We propose a novel densely connected graph to improve the representation of a medical ontology graph with two attention-based aggregate functions for GNN to integrate the compressed structural information.
- We conduct extensive experiments on one *de facto* public benchmark dataset for Intensive Care Unit (ICU) patients. Experimental results suggest that our models outperform all state-of-the-art models in terms of diagnosis prediction accuracy and justify the choice of our proposed densely connected graph and attention-based aggregate function.

The rest of the paper is organized as follows: Section II presents the related work in graph representation learning with focuses on EHR and provides an introduction of *Message Passing Neural Network* (*MPNN*). Section III formally defines the research problem. Section IV describes *DUGRA* in detail. Section V illustrates the experimental results. Finally, Section VI concludes the paper.

II. Related Work

A. Graph Representation Learning

Graph neural network (GNN) is a variant of neural network that operates on graph domain [13]. Graphs are prevalent in our daily life. Social networks [15], biological protein-protein networks [16], drug-drug interactions [17], recommendation systems [18], and neural language processing [19] — all data in these domains can be modeled as graphs. Recently, various GNNs have been proposed to encode graph structure information, including *Graph Convolutional Network* (*GCN*) [20], *GraphSAGE* [21], and *Graph Attention Network* (*GAT*) [22].

Message Passing Neural Network (*MPNN*) [23] is a framework that generalizes all the aforementioned graph neural network models. In MPNN, the node embedding learning process has two steps. The first step is to aggregate neighbor information using a function M^t, and the second step is to update the node representation based on the aggregated information of its neighbors, denoted by m_i^t, using function U^t. Both M^t and U^t can be customized depending on the context. A node n_i uses aggregate functions to collect the latent representation after $t-1$ time steps updating from its neighbors $N(n_i)$. The latent state of a node after t time steps is denoted by $v_i^t \in R^d$, where d is the dimension of the latent state. It is updated based on the message m_i^t from its neighbors $N(n_i)$:

$$m_i^t = \sum_{j \in N(n_i)} M^t(v_i^{t-1}, v_j^{t-1}), \tag{1}$$

$$v_i^t = U^t(m_i^t) \tag{2}$$

GNN has demonstrated its power on EHR data [24], [25], [26], [27], [28]. The authors in [25], [24] use heterogeneous graph neural network to model heterogeneity attributes in EHR

data. *ME2Vec* [26] uses graph embedding and *GAT* on a patient and doctor graph. However, *ME2Vec* does not take into account the medical ontology graph to enhance the representation power. *GAMENet* [28] uses GNN and a drug-drug interaction graph for medical recommendations. Another work worth mentioning is *G-Bert* [27], which uses one-layer *GAT* on a medical ontology graph to enhance the representation quality for the diagnosis concept. In contrast, our work uses a densely connected graph to enable passing information of leaf nodes directly to the parents without interference by intermediate nodes. Also, *G-Bert* does not consider the co-occurrence information in their embeddings as we do in our work.

B. Representation Learning in EHR

EHR data contains sequences of patient visits. Each visit contains heterogeneous information such as diagnoses, medications, lab results, and procedures. Due to the vast number of data types in EHR data, learning better representations is critical to improving the performance of downstream tasks. Previous work on representation learning for EHR data mainly followed work in Natural Language Processing (NLP) due to similar forms of data sequentiality [29]. Recurrent neural networks (RNN) can be used for diagnosis prediction [1], [30], [31], patient sub-typing [32], and handling missing data in EHR [33], [34]. *Deepcare* [7] redesigns the forget gate in *Long-Short Term Memory (LSTM)* to solve the irregular time gap problem in EHR data. Attention mechanism can also be used in EHR representation learning [35]. Recently, approaches similar to *Bidirectional Encoder Representations from Transformers (BERT)* [36] have been applied to diagnosis prediction [37], [38] and medical recommendation [27].

The medical ontology graph, which can be considered as another source of knowledge, has been recently exploited to improve the quality of the learned representation and the predicted power. *GRAM* [2] treats the medical ontology graph as Clinical Classifications Software (CCS) multilevel diagnosis hierarchy and ICD-9 code taxonomy as a Directed Acyclic Graph (DAG); then it uses attention mechanism to learn the medical code embedding of a node as a weighted sum of the embeddings of itself and its ancestors. The authors of *GRAM* argue that this would help low-frequency medical code to learn better embedding from their ancestors. *MMORE* [4] extends the idea from GRAM and also takes the idea from multi-sense for words [39]. *MMORE* assigns two basic embeddings for each ancestor in the medical ontology graph. Ideally, each embedding corresponds to a distinct sense that represents a particular cluster of low-level medical concepts. To alleviate the inconsistency problem between ontology graph and co-occurrence, they integrate EHR co-occurrence statistic data and the predictive task in their model to enhance the predictive power. They concatenate ontology embedding and EHR co-occurrence embedding to form their final representation of medical codes. The authors of *KAME* [3] argue that *GRAM* uses only child embedding learned from the model and ignores the ancestor embedding. *KAME* employs the unused ancestor embedding to generate a knowledge vector for each visit, then combines the knowledge vector and visit vector together to form the final visit embedding as the input to the predictive model. However, none of these methods use child nodes to enhance the ancestor embeddings and they do not utilize the power of GNN in a knowledge graph.

III. Problem Description

Let $C = \{c_1, c_2, c_3, \ldots, c_{|C|}\}$ be a set of *diagnosis concepts*, e.g., ICD-9 or ICD-10 codes. An *EHR database* contains a set of patient records P. Each *patient record* $p^i \in P$ consists of a time-ordered sequence of visits $\langle v_1^i, v_2^i, v_3^i, \ldots, v_T^i \rangle$, where T is the number of visits made by patient p^i. Each *visit* v_j^i contains a set of diagnosis concepts $v_j^i \subseteq C$. We use a binary vector $x \in \{0,1\}^{|C|}$ to represent the diagnoses in the visit, where the k-th element is set to 1 if $c_k \in v_j^i$, indicating patient p^i has diagnosis c_k in the visit v_j^i.

Consider Figure 1. Suppose the entire set of patient records contains only three diagnoses, namely *Type I Diabetes Mellitus* (*ICD-9 code: 25001*, c_1), *Type II Diabetes Mellitus* (*ICD-9 code: 25002*, c_2), and *Left Heart Failure* (*ICD-9 code: 42800*, c_3). If a patient has diagnoses *25001* and *42800* in a visit, then $x = [1, 0, 1]$ represents their visit.

A *medical ontology graph* captures the relationships of diagnosis concepts and represents such domain knowledge in a tree structure, where the parent nodes capture more general diagnosis concepts of their descendants. For example, in Figure 1 the two diagnoses *Type I Diabetes Mellitus* (*ICD-9 code: 25001*) and *Type II Diabetes Mellitus* (*ICD-9 code: 25002*) share the same ancestor *Diabetes Mellitus without complication* (*ICD-9 code: 2500*). Each leaf node is a diagnosis concept $c \in C$. $A(c)$ denotes the set of ancestors of a diagnosis concept c, containing the internal nodes from root to leaf c, but not including c. $Leaf(n)$ contains all leaf nodes of an internal node n.

Given a set of patient records P in an EHR dataset and a set of diagnosis codes C organized in the form of a medical ontology graph $\mathcal{G}$, the research problem is to learn a high-quality embedding e_k for each diagnosis concept $c_k \in C$ based on the co-occurrence information in P and the domain knowledge $\mathcal{G}$. To evaluate the quality of the learned embeddings, we follow the convention in the literature of representation learning and use a downstream prediction task to measure the quality using classification accuracy. Specifically, the goal is to predict diagnoses in future visits v_t based on previous visits. Since a visit may contain multiple diagnosis code, this is a multi-label classification problem.

IV. Proposed Method: DUGRA

In this paper, we propose a method called *DUGRA* that uses a customized *Message Passing Neural Network* (*MPNN*) to learn high-quality representations from two graphs, one graph for the medical ontology and another graph for the co-occurrences of diagnosis concepts. We utilize the power of GNN to pass important information between co-related medical concepts. Figure 2 depicts the overview of *DUGRA*, where the matrices E_{ON} and E_{CO} are the embeddings learned

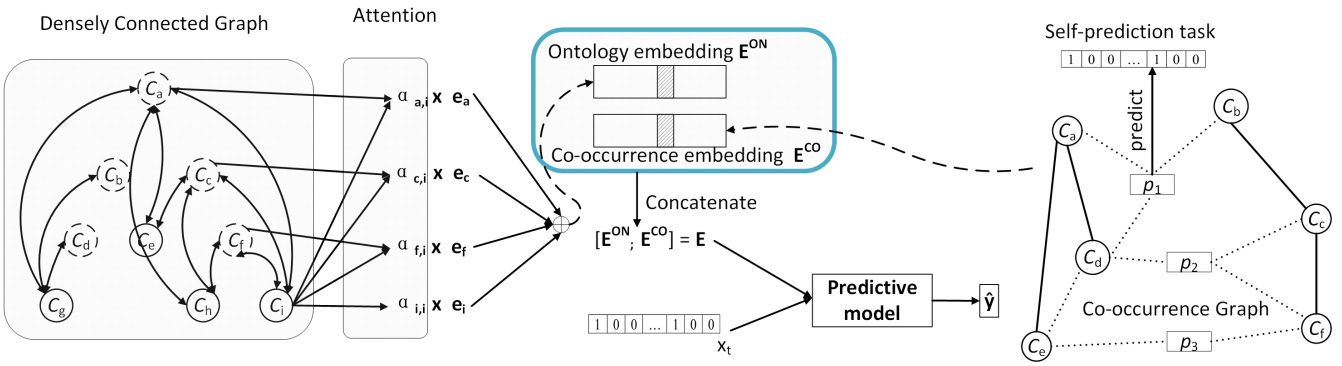


Fig. 2. Overview of *DUGRA*. The left part illustrates the idea of a densely connected graph. The leaf nodes (solid circles) are diagnosis concepts, while the internal nodes (dotted circles) represent the ancestors. The ontology embeddings are learned from the left part. The right part illustrates learning embeddings from the co-occurrence graph, and the lower-mid part is the predictive model.

from the densely connected ontology graph and co-occurrence graph, respectively. The left part of the figure corresponds to learning embeddings from the medical ontology graph. The right part is learning from the co-occurrence graph. The two learned embeddings are concatenated to form the final embeddings, which can be utilized for the downstream prediction task, as shown in the middle part. The detail of each module is elaborated below.

A. Densely Connected Graphs

1) Densely Connected Graph Construction: Our proposed densely connected schema aims to use the hierarchical structure of the medical ontology graph to learn high-quality diagnosis code embeddings, and it avoids dilution of knowledge by intermediate nodes. Li et al. [14] showed that as the number of propagation layers in GNN increases, the learned embeddings of nodes would become indistinguishable and converge to the same value. Our experimental result in Section V-D3 supports this observation.

In practice, most GNN will have at most 2 layers. A medical ontology graph has a hierarchical structure, and it may have 5 levels, as shown in Figure 1. However, a 2-layers GNN can at most pass information from 2-hops neighbors to the target nodes, which means the a 2-layers GNN can not pass information from the leaf to ancestors who are more than 2-hops away. We believe that fully using a graph neural network to propagate useful information from leaf nodes to all its ancestors is important to enhance the embedding representation abilities of ancestors.

To avoid the aforementioned problems of knowledge dilution caused by intermediate nodes, we design two densely connected directed graphs, namely *child-to-parent graph* and *parent-to-child graph*, from the raw medical ontology graph. The child-to-parent graph directly connects each leaf node c_i to its ancestors $A(c_i)$. The parent-to-child graph directly connects the internal nodes n_i to all of their leaves *Leaf*(c_i). Figure 3 depicts an example of a raw medical ontology graph. Figure 4 depicts the corresponding child-to-parent graph and parent-to-child graph. Note that the only difference between

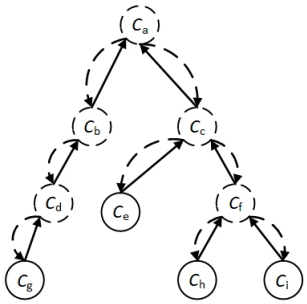


Fig. 3. Raw medical ontology graph

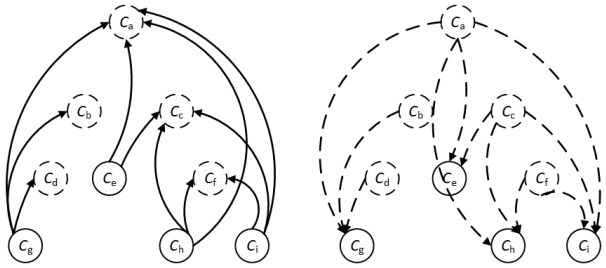


Fig. 4. Mean aggregate function data flow

the two resulting graphs is the edge direction. Unlike the raw medical ontology graph, this connectivity pattern allows nodes to directly access their parents' or children's information without being influenced by other intermediate nodes on their paths. Due to the fact that the raw medical ontology graph is a directed acyclic graph, this property guarantees that for each node the maximum number of parents in the densely connected graph is the depth of the node in the raw medical ontology. For example, in Figure 3 the depth of node C_e is 2, so the maximum number of parents, i.e., the number of outgoing edges in the child-to-parent graph, is 2, as shown in Figure 4. Therefore, the resulting graphs would not be too large for MPNN to handle.

2) Message Propagation: Recall that the Message Passing Neural Network (MPNN) described in Section II-A has two

steps: aggregation and update. The aggregation step gathers and integrates information from immediate neighbors. The update step is responsible for updating the node embedding based on the aggregated information from its neighbors. Putting MPNN in the context of the densely connected graphs, the message propagation mechanism operates on both child-to-parent and parent-to-child graphs.

In the child-to-parent propagation, MPNN passes information from leaves to their ancestors. Each parent node receives information from its children and uses an aggregation function to integrate the received information. The choice of aggregate function is flexible. We have tried three different aggregate functions, namely *mean aggregate*, *weighted mean*, and *attention*. They are explained in detail in Section IV-A3. The aggregated information of each node is then fed to an update function. We formulate the update function of a diagnosis concept c_k using a feed forward neural network:

$$v_{c_k} = \sigma(w_u m_{c_k} + b) \tag{3}$$

where w_u and b are learnable parameters, and m_{c_k} is the message c_k received from its children in the aggregation step. Each parent gets an updated embedding that integrates their children's information.

The learned embeddings v_{c_k} is the input embedding of the diagnosis c_k in parent-to-child propagation, in which the parents pass information to their children in the same way as described in the child-to-parent propagation. The output embedding of diagnosis concept c_k from the parent-to-child propagation is denoted by ϵ_k. In this step, all children would get information from their parents. Also, since parents have received information from their children in the child-to-parent propagation, the children can indirectly receive information from their siblings.

In a medical ontology, there are actually two types of nodes. The leaf nodes represent an actual diagnosis, and the internal nodes are generalized diagnosis concepts. To model such a heterogeneous graph, we argue that the relations between a parent and its child, and a child to its parent, are different and should be modeled differently. This justifies why we propose two separate sets of learnable parameters for the child-to-parent propagation and parent-to-child propagation. Section V-D4 provides empirical evidence to support our choice.

After the child-to-parent propagation and parent-to-child propagation, we follow the practice of *GRAM* [2] to get the embedding of diagnosis concept node c_k from a densely connected graph as a linear combination of the output of itself and its ancestors from the parent-to-child propagation:

$$E_k^{ON} = \sum_{j \in (A(c_k) \cup c_k)} \alpha_{kj} \epsilon_j \tag{4}$$

where $A(c_k)$ is the ancestors of c_k, and the attention weight α_{kj} is calculated by a feed forward neural network followed by a softmax function:

$$\alpha_{kj} = \frac{\exp(a^T tanh(w[\epsilon_k; \epsilon_j]))}{\sum_{l \in A(c_k) \cup c_k} \exp(a^T tanh(w[\epsilon_k; \epsilon_l]))} \tag{5}$$

where a and w are learnable parameters, and ';' is a concatenation operator.

3) Aggregate Functions: Three MPNN aggregate functions are used in our densely connected graphs. The first one, mean aggregate function, is a simple formulation for the purpose of baseline comparison in our experiments. Since our densely connected graph ignores the underlying structural hierarchical information in the raw medical ontology graph, we design two additional aggregate functions, weighted mean and attention, to incorporate the structural information.

Mean aggregate function. Each node c_k takes the element-wise mean of the incoming node embeddings:

$$m_{c_k} = \frac{1}{|N|} \sum_{j \in N(c_k)} e_{c_j} \tag{6}$$

where $N(c_k)$ is the set of neighbors of c_k, and e_{c_j} is the embedding of its neighbor c_j. Figure 3 is the original knowledge graph. Figure 4 shows the information flow of the mean aggregation function.

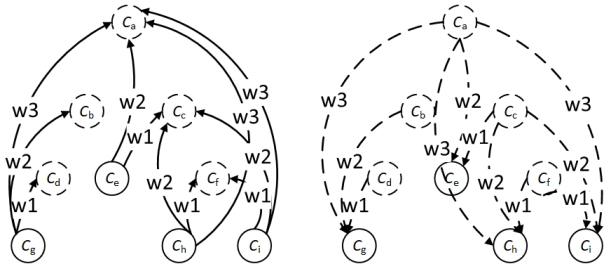


Fig. 5. Weighted-mean aggregate function data flow

Weighted-mean aggregation function. The learned embedding of a target node is influenced by its neighborhood. Neighbors in different distances in the raw medical ontology graph should have different levels of influences to the target node. Instead of having a predefined weight for different distances, we propose using a learnable vector that gives the weight of an edge based on the distance of two nodes in the raw medical ontology graph. w_a^δ and w_d^δ denote the learnable weights of neighbors who are δ-hop away from the target node for the child-to-parent and parent-to-child graphs, respectively.

The aggregate function for the child-to-parent propagation is:

$$m_{c_k} = \frac{1}{|N|} \sum_{\delta=0}^{H} \sum_{j \in N(c_k)^\delta} w_a^\delta e_{c_j} \tag{7}$$

where $N(c_k)^\delta$ is the set of neighbors of c_k that are k-hop away in the raw medical ontology graph, $|N|$ is the total number of δ-hop neighbours, H is the height of the raw medical ontology, and e_{c_j} is the embedding of node c_j. This method only adds $2 \times (H+1)$ parameters to our model, where H tends to be small. In the case of the medical ontology from CCS, H = 5. The aggregate function for the parent-to-child propagation is the same as Equation 7, but replacing w_a^δ by w_d^δ. Figure 5 depicts the data flow of weighted-mean aggregate function, where the edge values stand for the weights w_a^δ or w_d^δ.

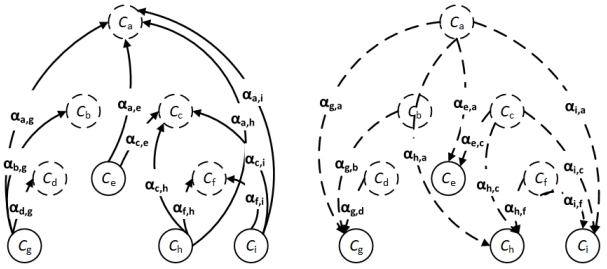


Fig. 6. Attention aggregate function data flow

Attention aggregation function. Inspired by the *graph attention networks* (*GAT*) [22], we use an attention mechanism to determine the weight of each edge. The implemented aggregate function:

$$m_{c_k} = \frac{1}{|N|} \sum_{j \in N(c_k)} \alpha_{k,j} e_{c_j} \tag{8}$$

where $\alpha_{k,j}$ is the attention weight given to the edge between nodes c_k and c_j, which is computed as follows:

$$\alpha_{k,j} = \frac{\exp(LeakyReLU(a^T tanh(w[e_k; e_j])))}{\sum_{l \in N(c_k)} \exp(LeakyReLU(a^T tanh(w[e_k; e_l])))} \tag{9}$$

where $a \in R^{2m}$ is a learnable weight vector, and $LeakyReLU$ is a nonlinear function. Following the convention, we set the negative input slope α = 0.2. Figure 6 shows the information propagation flow for the attention aggregate function, where the edge value $\alpha_{i,j}$ denotes the value obtained from the attention mechanism.

B. Co-occurrence Graph

Both *GRAM* [2] and *MMORE* [4] are using the co-occurrence information to form the final embeddings for medication and diagnosis concepts. *GRAM* uses co-occurrence embedding to initialize the input embedding of the medication concept. *MMORE* concatenates the embedding learned from the medication and diagnosis ontologies and the co-occurrence information together to form the final embeddings. The idea of using co-occurrence information is based on the assumption that medication or diagnosis concepts appearing frequently together in the same visit should share similar characteristics and, therefore, similar embeddings. However, none of the existing methods intend to capture the co-occurrence information across all visits of one patient. Therefore, we propose to use a large EHR graph of patients and diagnosis concepts to learn the co-occurrence embedding through MPNN.

Inspired by the idea of *graph convolutional network for text classification* [19], which uses a huge text graph to conduct text classification tasks, we build a large heterogeneous co-occurrence graph that contains diagnosis concepts and patients as nodes so that the co-occurred diagnoses within the same visit and the same patient record can be modeled together, and MPNN can be deployed on this graph. The number of nodes in the co-occurrence graph, $|V|$, is the number of distinct diagnosis concepts plus the number of patients in the training set. We set the input feature embedding randomly, $I \in R^{(m \times d)}$, where m is the number of nodes in the graph, and d is the input embedding dimension. We build edges between diagnosis nodes and between diagnosis nodes and patient nodes. There is no edge between patient nodes because most of EHR system does not capture the relationship among patients. Note that there is an edge between two diagnosis nodes if they co-occur in the same visits, and there is an edge between a patient and a diagnosis node if the patient has the diagnosis in any visit.

To compute the degree of co-occurrences between two diagnosis nodes we use *point-wise mutual information* (*PMI*), which is a popular method for measuring associations between two objects, to quantify the edge weight. The PMI value between two diagnosis nodes c_k and c_j is:

$$PMI(c_k, c_j) = log(\frac{p(c_k, c_j)}{p(c_k)p(c_j)}) \tag{10}$$

$$p(c_k, c_j) = \frac{\textit{co-occurrence}(c_k, c_j)}{\#\ \textit{of patients}} \tag{11}$$

$$p(c_k) = \frac{\#\ \textit{of patients having}\ c_k}{\#\ \textit{of patients}} \tag{12}$$

where *co-occurrence*(c_k, c_j) is the number of patients having both c_k and c_j. A positive Pointwise Mutual Information (PMI) value implies high correlation between two nodes, and a negative PMI value indicates no correlation between c_k and c_j. Therefore, we only add edges between two diagnosis nodes that have a positive PMI value.

For the edge weight between patients and diagnoses we employ a variation of *Term Frequency-Inverse Document Frequency* (*TF-IDF*) [40], a popular measure in data mining and natural language processing. We treat patients as documents and diagnoses as words. The term frequency is the number of times a diagnosis is assigned to a patient, and the inverse document frequency is the inverse fraction of the number of patients containing these diagnoses. The TF-IDF value between diagnosis node c_k and patient node p_i is computed as:

$$TF_{c_k, p_i} = \frac{\#\ \textit{of}\ c_k\ \textit{in}\ p_i}{\#\ \textit{of distinct diagnoses in}\ p_i} \tag{13}$$

$$IDF_{c_k} = log(\frac{\#\ \textit{of patients}}{\#\ \textit{of patients having}\ c_k\ + 1}) \tag{14}$$

$$TF - IDF_{c_k, p_i} = TF \times IDF \tag{15}$$

If a diagnosis has been assigned to many patients, its IDF value would be small, implying the diagnosis concept is not a good candidate to represent the distinct situation of the patients. Therefore we assign a small value to the weight through the small IDF value. In contrast, a rare diagnosis would receive a higher IDF value because it can reflect the

special situation of the patients. Formally the weight of edge between node i and node j is defined as:

$$w_{ij} = \begin{cases} PMI(i,j), & i,j \textit{ are diagnosis nodes, and PMI(i, j) > 0} \\ TF - IDF_{i,j}, & i \textit{ is a diagnosis node and } j \textit{ is a patient node} \\ 1, & i = j \\ 0, & \textit{otherwise} \end{cases} \tag{16}$$

After building the graph we feed the graph to a 2-layer MPNN. The aggregate function for node i is an element-wise weighted sum of the incoming nodes' embedding:

$$m_i = \frac{1}{|N|} \sum_{j \in N(i)} w_{i,j} e_j \tag{17}$$

The update function is a feed forward neural network similar to Equation 3. A 2-layer MPNN allows message passing among nodes that are at most 2-hops away. Thus, although some diagnosis nodes may not co-occur in any visit, they can be a 2-hop neighbor through patient nodes or through other diagnosis nodes.

Self-prediction task. We implement an auxiliary self-prediction task to help the co-occurrence graph learn meaningful representation from the co-occurred diagnoses in visits in patient records. For each patient p_i, we use the output embedding o_i of the MPNN as the representation of p_i. The basic idea is that the patient embedding can be used to predict the health status, i.e., the set of k diagnosis codes $\{c_1, c_2, c_3, \ldots, c_k\} \in C$ in p_i's patient record. The goal can be achieved by minimizing the negative log probability of the codes presented in the patient node output o_i:

$$\mathcal{L}^{CO} = -\frac{1}{|K|} \sum_{k=1}^{|K|} log\ p(c_k|o_i) = -\frac{1}{|K|} \sum_{k=1}^{|K|} log \frac{exp(w_1^{'T} \cdot o_i)}{\sum_{i=1}^{k} exp(w_2^{'T} \cdot o_i))} \tag{18}$$

We use the softmax function to compute the conditional probability, and $w_1^{'}$ and $w_2^{'}$ are learnable parameters.

C. End-to-End Predictive Model

We train the dual graph neural networks together with a predictive model so that the graph module improves the predictive performance. The embedding matrices E^{ON} and E^{CO} are first row-wisely concatenated to form the final representations for the diagnosis concepts, i.e, $E = [E^{ON}; E^{CO}]$. We create the final visit embedding $v_t \in R^n$ where n is the dimension of the visit embedding as follows:

$$v_t = Relu(E[x_t]) \tag{19}$$

where x_t is a one-hot representation of diagnosis concepts appearing in the visit. Then we input the visit embedding to a feed forward neural network to get the prediction.

$$\hat{y}_t = Softmax(Qv_t + s) \tag{20}$$

where Q and k are the learnable parameters. We use cross-entropy loss as the objective function:

$$\mathcal{L}_{p_i}^{pred} = -\frac{1}{T} \sum_{t=2}^{T} [y_t^T log(\hat{y}_t) + (1 - y_t)^T log(1 - \hat{y}_t))] \tag{21}$$

where T is the number of hospital admissions to be predicted for patient p_i, and y_t is the ground truth label. t starts from 2 because the first visit result has no previous visit. Note that the above loss is for a single patient. We take the average of the individual patient's loss for training all patients in the set.

V. Experiments

To demonstrate the quality of the diagnosis representation we compare the performance of our proposed model, *DUGRA*, with other state-of-the-art models in terms of the predictive performance of future diagnoses on an EHR dataset. By predicting a patient's future health status, physicians can start preventive measures earlier and alleviate the burden on medical systems. We also use experiments to justify our choice of a densely connected graph to replace the original medical ontology graph. Furthermore, we conduct ablation studies to evaluate the impact of different modules in our proposed model on the overall performance.

# of patients	5,404
Avg # of diagnoses per admission	12.26
Min # of diagnoses per admission	1
Max # of diagnoses per admission	39
Avg # of visit	2.60
Min # of visit	2
Max # of visit	29
# of diagnosis codes	3,495
# of labels	712

TABLE I
STATISTICS OF THE MIMIC DATASET

A. Dataset and Preprocessing

MIMIC-III [41] is a *de facto* benchmark dataset consisting of medical records of more than 7.5K patients admitted to intensive care units (ICUs), with over 46K visits over 11 years. It contains various information such as demographics, lab results, diagnoses, and medications. Since it is an ICU dataset, most visits span a short period of time with severe situations. For a fair comparison we follow the preprocessing set-up in *MMORE*. We extract adult patients with at least two hospital admissions where diagnoses and medications are both present in the MIMIC dataset. We exclude the base-type medications. Finally, we extract 5,404 patients with an average of 2.6 visits per patient; the average number of diagnoses per admission is 12.3. The statistics of the datasets are summarized in Table I.

B. Baseline Models

We compare our model with the following four state-of-the-art models:

1) *RETAIN* [30] implements a two-level neural attention model, one for visit-level attention and the other for variable-level. *RETAIN* uses two reverse-order RNNs to generate the attention weight for the two attentions.
2) *Med2Vec* [42] considers medical concepts in neighbor admissions to capture their co-occurrence relationships by the *Skip-gram* algorithm.
3) *GRAM* [2] is the first work that uses the medical ontology graph to learn the medical concept representation and

	Training size			
Model	20%	40%	60%	80%
RETAIN	0.4422	0.4447	0.4449	0.4545
Med2Vec	0.5064	0.5187	0.5200	0.5290
GRAM	0.4980	0.5218	0.5409	0.5498
MMORE	0.5205	0.5426	0.5548	0.5618
$DUGRA_{mean}$	0.5274	0.5472	0.5617	0.5705
$DUGRA_{weighted\_mean}$	0.5324	0.5512	0.5656	0.5737
$DUGRA_{attn}$	**0.5329**	**0.5527**	**0.5664**	**0.5740**

TABLE II
ACCURACY@20 PREDICTION FOR COMPARING DIFFERENT MODELS

	Training size			
Model	20%	40%	60%	80%
Densely connected graph	0.5306	0.5479	0.5605	0.5700
1 hidden layer in MPNN	0.5254	0.5419	0.5545	0.5633
2 hidden layer in MPNN	0.5227	0.5348	0.5493	0.5584
3 hidden layer in MPNN	0.5164	0.5248	0.5434	0.5504
4 hidden layer in MPNN	0.5054	0.5236	0.5402	0.5501
5 hidden layer in MPNN	0.4981	0.5232	0.5378	0.5470

TABLE III
ACCURACY@20 PREDICTION FOR ABLATION STUDY ON DENSELY CONNECTED GRAPH

predict the next admission status. The medical concept representation is a weighted sum of its own embedding and the ancestors' embeddings. The weight is computed by a self-attention mechanism.

4) *MMORE* [4] extends the *GRAM* framework by allowing each ancestor, except the root node, to have two embeddings. Also, they combine an embedding learned from co-occurrence statistics into their medical concept embedding.

To test the performance of the mean aggregate function, weighted-mean aggregate function, and attention aggregate function in the ontology graph, we conduct experiments on three models, namely $DUGRA_{mean}$, $DUGRA_{weighted\_mean}$, and $DUGRA_{attn}$, respectively.

C. Experiment Setup

For a fair comparison we follow the setting in *MMORE* and set the dimension of both the knowledge graph embedding and the co-occurrence embedding to 400 in *DUGRA* and *MMORE*. The embedding dimension of all other baselines, *RETAIN*, *Med2Vec*, and *GRAM*, is set to 800 for fair comparison as other models do not concatenate the two embedding matrices. We use a single-layer neural network as the prediction model. We also try to use *GRU* as the prediction model; however, the performance is worse than a single layer neural network because many patients in MIMIC-III only have two visits. Therefore, using a GRU may overfit the dataset. The model is optimized using Adadelta [43] with batch size 100.

D. Result

We generate the ground-truth label y_t for the diagnoses prediction task by grouping the diagnoses in the next admissions into 712 groups based on the first 3 digits of their ICD-9 codes in database. We randomly split the data into training, validation, and testing sets. We fix the size of the validation set to be 10% of the total number of patients. In real-life health machine learning tasks, the availability of training data can be very different depending on the specific study. To validate the robustness of *DUGRA*, we vary the training set size to be 20%, 40%, 60%, and 80% of the total number of patients and use the remaining part as the testing set. We measure the predictive performance by Accuracy@k, which is defined as:

$$Accuracy@k = \frac{\text{\# of true positives in the top k predictions}}{\text{\# of positives in this visit}} \tag{22}$$

This is a multi-label classification problems as mentioned in Section III.

1) Comparing with State-of-the-Arts: Table II shows the *Accuracy@20* result of the next-admissions prediction tasks for the three variants of *DUGRA* and baselines. The result suggests that all three variants of *DUGRA* generally outperform the baselines, especially with a small training set. This result also illustrates the effectiveness of using a graph neural network for embedding learning from a medical ontology graph and a co-occurrence graph. By using the co-occurrence statistic information and medical ontology graph together, *DUGRA* and *MMORE* show even more significant improvement when compared to other baselines that use only an *either* medical ontology graph or co-occurrence information. This implies that both the human-defined knowledge graph and the degree of co-occurrence are useful for learning high-quality diagnosis representations.

2) Performance of Aggregate Functions: Consider the *Accuracy@20* result of different aggregate functions in Table II. $DUGRA_{weighted\_mean}$ and $DUGRA_{attn}$ achieve similar accuracy, but the method of $DUGRA_{weighted\_mean}$ only adds a small number ($2 \times (H+1)$, where H is the number of layers in the original ontology graph) of parameters to the traditional MPNN and does not require considerable memory overhead as in $DUGRA_{attn}$. This is due to the fact that a more complex model, such as $DUGRA_{attn}$, cannot show its power given the limited size and simple structure of an ICD-9 ontology graph. Also, $DUGRA_{weighted\_mean}$ outperforms $DUGRA_{mean}$ by a relatively larger margin. This supports our hypothesis that different levels of neighbors should have different levels of influences to the target nodes.

3) Ablation Study on Proposed Densely Connected Graph: The objectives of this ablation study are to evaluate the performance of our proposed densely connected graph compared to the performance of the original medical ontology graph, e.g., ICD-9 taxonomy, and to support our claim that stacking more hidden layers in the MPNN would cause a drop in performance, as discussed in Section IV-A1 . Since this ablation study is on the densely connected graph, all models in this experiment use only the medical ontology graph *without* the co-occurrence information. The dimension of the ontology embedding is 800 in both cases. We use the attention function as the aggregate function because it yields the best result in Section V-D2. Table III demonstrates the model performance of MPNN with a different number of hidden layers using

	Training size			
Variants of *DUGRA*	20%	40%	60%	80%
2 sets of parameters	0.5306	0.5479	0.5605	0.5700
same set of parameters	0.4886	0.5087	0.5119	0.5231

TABLE IV
ACCURACY@20 PREDICTION FOR ABLATION STUDY ON HAVING TWO SEPARATE SETS OF LEARNABLE PARAMETERS IN DIFFERENT PROPAGATION PHASES ON DENSELY CONNECTED GRAPH.

the original knowledge graph and supports our claim that stacking more layers in the MPNN of the original medical ontology graph would degrade the prediction performance for downstream tasks. Since the maximum distance of a leaf node to the root is 5 in the ICD-9 taxonomy, an MPNN with at least five hidden layers is enough to pass information from the leaf to the root. This study demonstrates that adding more layers to MPNN to allow information from the bottom to propagate to the root cannot get better results compared to a shallow MPNN due to the oversmoothing problem in GNN.

4) Ablation Study on Two Separate Sets of Learnable Parameters: We further perform an empirical evaluation to support our choice of using two separate sets of parameters for the child-to-parent propagation and parent-to-child propagation in Section IV-A2. Since this is an ablation study on our proposed densely connected graph, all models in this experiment use only the medical ontology graph *without* the co-occurrence information. The dimension of the ontology embedding is 800. The aggregate function is attention function. Table IV shows the *Accuracy@20* result of a densely connected graph with/without using two different sets of parameters for two different propagation phases with different training set sizes. The result shows that the performance using two sets of parameters consistently outperforms the performance using only one set. This justifies our choice of using two different sets of parameters.

VI. CONCLUSION

In this paper, we introduce *DUGRA* to learn high-quality embeddings for diagnosis concepts from a densely connected graph and a co-occurrence graph via MPNNs. Instead of using the original ontology graph we introduce a densely connected schema to avoid the dilution of knowledge from the intermediate nodes. This is the first work that utilizes a GNN to model the co-occurrence information of diagnosis code in EHR data. The empirical evaluation suggests that our proposed model performs better than the state-of-the-art models in terms of the diagnosis prediction accuracy for future visits, even with small training sets. Also, the experimental results confirm our choice of using a densely connected graph and a co-occurrence graph. Future work will focus on generalizing our model to integrate other heterogeneous medical concepts and optimizing the co-occurrence graph for a large EHR dataset.

ACKNOWLEDGEMENTS

This research is supported by Canada Research Chairs Program (950-230623).

REFERENCES

[1] E. Choi, M. T. Bahadori, A. Schuetz, W. F. Stewart, and J. Sun, "Doctor AI: Predicting clinical events via recurrent neural networks," in *Proceedings of the Machine Learning for Healthcare Conference*, 2016, pp. 301–318.

[2] E. Choi, M. T. Bahadori, L. Song, W. F. Stewart, and J. 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*, 2017, pp. 787–795.

[3] F. Ma, Q. You, H. Xiao, R. Chitta, J. Zhou, and J. Gao, "Kame: Knowledge-based attention model for diagnosis prediction in healthcare," in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, 2018, pp. 743–752.

[4] L. Song, C. W. Cheong, K. Yin, W. K. Cheung, B. C. M. Fung, and J. Poon, "Medical concept embedding with multiple ontological representations," in *IJCAI'19 Proceedings of the 28th International Joint Conference on Artificial Intelligence*, 2019, pp. 4613–4619.

[5] C. Xiao, T. Ma, A. B. Dieng, D. M. Blei, and F. Wang, "Readmission prediction via deep contextual embedding of clinical concepts," *PloS One*, vol. 13, no. 4, 2018.

[6] P. Nguyen, T. Tran, N. Wickramasinghe, and S. Venkatesh, "Deepr: a convolutional net for medical records," *IEEE Journal of Biomedical and Health Informatics*, vol. 21, no. 1, pp. 22–30, 2016.

[7] T. Pham, T. Tran, D. Phung, and S. Venkatesh, "Deepcare: A deep dynamic memory model for predictive medicine," in *Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 2016, pp. 30–41.

[8] Y. Zhu, X. Fan, J. Wu, X. Liu, J. Shi, and C. Wang, "Predicting icu mortality by supervised bidirectional lstm networks." in *Proceedings of the AIH@ IJCAI*, 2018, pp. 49–60.

[9] A. E. Johnson, T. J. Pollard, and R. G. Mark, "Reproducibility in critical care: a mortality prediction case study," in *Proceedings of the Machine Learning for Healthcare Conference*, 2017, pp. 361–376.

[10] Z. Che, S. Purushotham, R. Khemani, and Y. Liu, "Interpretable deep models for icu outcome prediction," in *Proceedings of the AMIA Annual Symposium*. American Medical Informatics Association, 2016, p. 371.

[11] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proceedings of the Advances in Neural Information Processing Systems*, 2013, pp. 3111–3119.

[12] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. Van Den Berg, I. Titov, and M. Welling, "Modeling relational data with graph convolutional networks," in *Proceedings of the European Semantic Web Conference*. Springer, 2018, pp. 593–607.

[13] J. Zhou, G. Cui, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun, "Graph neural networks: A review of methods and applications," *arXiv preprint arXiv:1812.08434*, 2018.

[14] Q. Li, Z. Han, and X.-M. Wu, "Deeper insights into graph convolutional networks for semi-supervised learning," in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[15] W. Fan, Y. Ma, Q. Li, Y. He, E. Zhao, J. Tang, and D. Yin, "Graph neural networks for social recommendation," in *Proceedings of the The World Wide Web Conference*, 2019, pp. 417–426.

[16] A. Fout, J. Byrd, B. Shariat, and A. Ben-Hur, "Protein interface prediction using graph convolutional networks," in *Proceedings of the Advances in Neural Information Processing Systems*, 2017, pp. 6530–6539.

[17] T. Ma, C. Xiao, J. Zhou, and F. Wang, "Drug similarity integration through attentive multi-view graph auto-encoders," *arXiv preprint arXiv:1804.10850*, 2018.

[18] R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, and J. Leskovec, "Graph convolutional neural networks for web-scale recommender systems," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018, pp. 974–983.

[19] L. Yao, C. Mao, and Y. Luo, "Graph convolutional networks for text classification," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 7370–7377.

[20] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[21] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proceedings of the Advances in Neural Information Processing Systems*, 2017, pp. 1024–1034.

[22] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," *arXiv preprint arXiv:1710.10903*, 2017.

[23] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017, pp. 1263–1272.

[24] S. Liu, F. Vahedian, D. Hachen, O. Lizardo, C. Poellabauer, A. Striegel, and T. Milenkovic, "Heterogeneous network approach to predict individuals' mental health," *arXiv preprint arXiv:1906.04346*, 2019.

[25] A. Hosseini, T. Chen, W. Wu, Y. Sun, and M. Sarrafzadeh, "Heteromed: Heterogeneous information network for medical diagnosis," in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, 2018, pp. 763–772.

[26] T. Wu, Y. Wang, Y. Wang, E. Zhao, Y. Yuan, and Z. Yang, "Representation learning of ehr data via graph-based medical entity embedding," *arXiv preprint arXiv:1910.02574*, 2019.

[27] J. Shang, T. Ma, C. Xiao, and J. Sun, "Pre-training of graph augmented transformers for medication recommendation," *arXiv preprint arXiv:1906.00346*, 2019.

[28] J. Shang, C. Xiao, T. Ma, H. Li, and J. Sun, "Gamenet: graph augmented memory networks for recommending medication combination," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 1126–1133.

[29] E. Steinberg, K. Jung, J. A. Fries, C. K. Corbin, S. R. Pfohl, and N. H. Shah, "Language models are an effective patient representation learning technique for electronic health record data," *arXiv preprint arXiv:2001.05295*, 2020.

[30] E. Choi, M. T. Bahadori, J. Sun, J. Kulas, A. Schuetz, and W. Stewart, "Retain: An interpretable predictive model for healthcare using reverse time attention mechanism," in *Proceedings of the Advances in Neural Information Processing Systems*, 2016, pp. 3504–3512.

[31] Z. C. Lipton, D. C. Kale, C. Elkan, and R. Wetzel, "Learning to diagnose with lstm recurrent neural networks," *arXiv preprint arXiv:1511.03677*, 2015.

[32] I. M. Baytas, C. Xiao, X. Zhang, F. Wang, A. K. Jain, and J. Zhou, "Patient subtyping via time-aware LSTM networks," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2017, pp. 65–74.

[33] Z. Che, S. Purushotham, K. Cho, D. Sontag, and Y. Liu, "Recurrent neural networks for multivariate time series with missing values," *Scientific Reports*, vol. 8, no. 1, pp. 1–12, 2018.

[34] Z. C. Lipton, D. C. Kale, and R. Wetzel, "Modeling missing data in clinical time series with rnns," *arXiv preprint arXiv:1606.04130*, 2016.

[35] F. Ma, R. Chitta, J. Zhou, Q. You, T. Sun, and J. 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*, 2017, pp. 1903–1911.

[36] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[37] Y. Wang, X. Xu, T. Jin, X. Li, G. Xie, and J. Wang, "Inpatient2vec: Medical representation learning for inpatients," in *Proceedings of the 2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE, 2019, pp. 1113–1117.

[38] Y. Li, S. Rao, J. R. A. Solares, A. Hassaine, D. Canoy, Y. Zhu, K. Rahimi, and G. Salimi-Khorshidi, "BEHRT: Transformer for electronic health records," *arXiv preprint arXiv:1907.09538*, 2019.

[39] E. Huang, R. Socher, C. Manning, and A. Ng, "Improving word representations via global context and multiple word prototypes," in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2012, pp. 873–882.

[40] J. Ramos *et al.*, "Using tf-idf to determine word relevance in document queries," in *Proceedings of the First Instructional Conference on Machine Learning*, vol. 242. Piscataway, NJ, 2003, pp. 133–142.

[41] A. E. Johnson, T. J. Pollard, L. Shen, H. L. Li-wei, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. A. Celi, and R. G. Mark, "Mimic-iii, a freely accessible critical care database," *Scientific Data*, vol. 3, p. 160035, 2016.

[42] E. Choi, M. T. Bahadori, E. Searles, C. Coffey, M. Thompson, J. Bost, J. Tejedor-Sojo, and J. Sun, "Multi-layer representation learning for medical concepts," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 1495–1504.

[43] M. D. Zeiler, "Adadelta: an adaptive learning rate method," *arXiv preprint arXiv:1212.5701*, 2012.