IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received May 11, 2020, accepted June 24, 2020, date of publication July 7, 2020, date of current version July 20, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3007835

GATE: Graph-Attention Augmented Temporal Neural Network for Medication Recommendation

CHENHAO SU, SHENG GAO, AND SI LI
PRIS, Beijing University of Posts and Telecommunications, Beijing 100876, China

Corresponding author: Sheng Gao (gaosheng@bupt.edu.cn)

This work was supported by the National Natural Science Foundation of China under Grant 61702047.

ABSTRACT Medication recommendation based on Electronic Health Records (EHRs) is an important research direction, which aims to make prescription recommendations according to EHRs of patients. Most existing methods either only make recommendation through EHRs of the current admission while ignoring the patient's historical records, or fail to fully consider the correlations among the clinical events from every single admission. These methods have shown their limitations in dealing with the complex structural correlations and temporal dependencies of clinical events in EHRs, which results in the defect of recommendation quality as well as the lack of temporal prediction ability. For that, a novel graph-attention augmented temporal neural network is proposed to model both the structural and temporal information simultaneously. For each admission record, a co-occurrence graph is constructed to establish the correlations among clinical events, and then a graph-attention augmented mechanism is used to learn the structural correlations on the graphs to obtain better representation of this admission. Then a temporal updating module based on the gated recurrent units is further proposed to learn the temporal dependencies between multiple admissions of each patient. Furthermore, our proposed model is also constrained by the co-occurrence graph, which can capture the internal correlations of clinical events and provide better modeling capability when the training data is sparse. Experiments illustrate that our model is superior to the state-of-the-art methods on a real-world dataset MIMIC-III in all effectiveness measures.

INDEX TERMS Electronic medical records, medication recommendation, graph-attention neural networks, temporal updating.

I. INTRODUCTION

Electronic Health Records (EHRs) are the main data carriers for personalized medical research. With the popularity and improvement of the quality of EHRs, plenty of efforts have been dedicated to this field due to the potential applications such as medication recommendation and diagnosis prediction [1], [15], [24], [25]. Normally, an EHR is represented as a temporal admission sequence for the patient, in which each sequence contains a series of clinical events (diagnoses, procedures, medications, etc.) of a single admission [4]. Given current clinical events as well as historical admission records of the patient, the goal of the medication recommendation task is to provide personalized medication combinations appropriate for her/his health condition. The medication recommendation problem is highly non-trivial and has a long history in the field of machine-aided medical diagnoses and treatment. Early medication recommendation researches [40]–[42] are mainly based on facts and rules summarized by experts with rich clinical experiences. In recent years, deep learning based methods have been widely used in this task [1], [2], [4], [10], [25], which have greatly improved the prediction accuracy and provided greater possibilities for applications in practical scenarios.

The main challenges of medication recommendation problem arise from the following two aspects:

1) **Structural correlations**: The EHR of a single admission can be regarded as a collection of clinical events including diagnoses, procedures and medications, these events are internally related, and the correlation between different events has different meanings and degrees, which we call structural correlations. For example, people with alcohol dependence are more likely to have hyperlipidemia and hypertension. There may also be correlations between different medication

The associate editor coordinating the review of this manuscript and approving it for publication was Bo Shen.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

suggestions, e.g., cardiac glycoside and antiarrhythmic drugs often work together. Besides, the relationship between diagnoses and symptomatic medications is also important for prediction.

2) **Temporal dependencies**: The patient's condition will change with the time of multiple admissions, and the course of different diseases, different procedures and medications are also different. For example, liver transplantation may lead to some complications in the future, and gout usually has a longer course than esophageal reflux disease. Clinical events such as diagnoses and medications between different admissions often have various temporal dependencies.

In order to capture structural correlations, graph-based deep learning techniques are often used to construct clinical events into a network for better prediction. Mao *et al.* [9] used the graph convolutional network (GCN) to model the similarities between lung cancer patients according to their health record information. Shang *et al.* [2] integrated the drug-drug interactions knowledge graph to establish the relationship between medication labels. Choi *et al.* [5], [24] used the GCN and graph-based transformer to model the relationship between different types of clinical events. Other existing methods [1], [3], [25] used attention aggregation or GCN to encode the tree structure of different medical code ontology. Most of these methods just concern the structural correlation among some types of clinical elements, which is not enough to fully capture the different structural correlations among various clinical events in EHRs.

Some existing models take into account the temporal evolution of the patient's historical EHRs in medication prediction. Choi *et al.* [4], [10] used a multi-hot label vector to represent clinical events for each admission, then adopted gated recurrent units (GRUs) or a two-level attention mechanism to model the temporal evolution of multiple admissions. In these two approaches, clinical events of each historical admission are treated as a bag of independent features, i.e., a set of independent vectors. More specifically, clinical events of a single admission are represented as multi-hot vectors and then perform linear embedding to obtain representations of the admission. The method of using events as a vector set cannot learn different correlations between each clinical event. Finally, medication prediction is made based on the temporal representation sequence of multiple admissions.

It should be mentioned that the structural correlations refer to the relationship between all the clinical events of a single admission, while the temporal dependencies focus on the temporal evolution of the clinical events between different admissions, these two types of information exist naturally in the data. Simultaneously modeling these two types of information helps to make full use of the data available, thereby improving the recommendation quality.

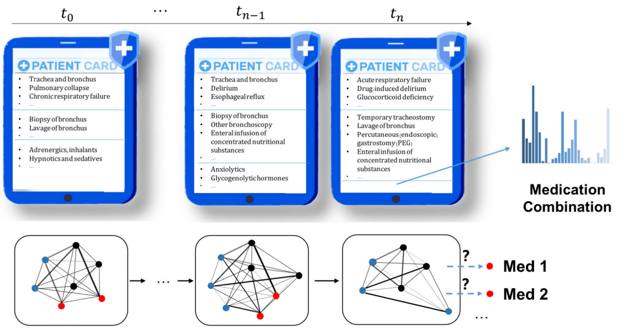


FIGURE 1. Illustration of medication recommendation task, where the input is the EHRs of a patient in chronological order (above), and the prediction result is the probability of each medicine at the n^{th} admission. EHRs are constructed in the form of temporal graph-structured sequences (below). At each time point, there is a dynamic graph that takes all types of events at this admission as nodes and uses the correlation between these events as edge weights. Black and blue nodes represent diagnoses and treatments, respectively, and red nodes represent medications.

In this work, we propose a Graph-Attention augmented TEmporal neural network (GATE) that simultaneously models structural and temporal information in the EHR of each patient. As shown in Fig. 1, we treat the original input as a graph-based sequence based on the characteristics of the graph structure to preserve the correlation between clinical events. Then the problem reduces to the task of temporal modeling and prediction over the graph-based data. Specifically, for each admission record of a patient, we construct the patient's diagnoses, treatments and medication history into a co-occurrence correlation graph, in which each node represents the clinical event and the weight of the edge between nodes represents the relevance between clinical events. Then we use the graph-attention mechanism to capture the inherent structural correlations in the graph. After that, we propose a GRU-based temporal updating module to learn the temporal dependencies between multiple admissions of each patient. Since our model is constrained by co-occurrence graphs, it can better capture the internal correlations of events. For rare or non-occurring clinical events, our proposed model can also obtain more effective information from co-occurring events through the extracted structural and temporal correlations, thus providing supplement and enhancement for sparse data.

To summarize, the main contributions of our work are as follows:

1) We treat EHRs as temporal graph-structured data and propose a graph-attention augmented temporal neural network to simultaneously model the temporal and structural information for each patient. Our proposed model uses an advanced graph-attention mechanism and a two-dimensional incremental temporal updating module.

2) We construct the EHR data into a global guidance co-occurrence correlation graph, which guides the attention weight between event nodes in the graph attention mechanism, and then models the structural characteristics of each admission record by incorporating the information of diagnoses, procedures, and medications.

3) Our proposed GATE method outperforms existing state-of-the-art methods in terms of three metrics in medication recommendation on a real-world dataset MIMIC-III.

The remainder of this paper is organized as follows. In section II, we introduce some related work used in our method. In section III, we describe the framework of our proposed GATE model. In section IV, we compare the prediction performance of the proposed model with other baseline models based on the real-world dataset MIMIC-III and presents several analyses. Finally, we give some conclusions and discuss future studies in section V.

II. RELATED WORK

A. GRAPH NEURAL NETWORKS

Recently, great attention has been paid to the study of graph representation learning. As a deep learning method on graph domain, graph neural network (GNN) has been widely used in various scenarios, such as social network, knowledge graph, physical system, etc., because of its great expressive power and high interpretability. The concept of GNN was first introduced in [23], which extended the existing neural network methods for processing the data represented as the graphs. GNN is based on an information diffusion mechanism, the intuitive idea underlying GNN is that nodes in a graph are naturally specified by using the information contained in their neighborhood. The core part of GNN is a propagation function, which is used to express the dependence of a node on its neighborhood. Different from the previous approaches of network information aggregation, GNN is naturally able to capture the multi-hop relations and help to obtain higher-order features.

Kipf *et al.* [13] proposed a simplified spectral approach called GCN, which significantly outperformed the existing methods on the semi-supervised classification task. GCN introduced a simple and efficient layer-wise propagation function for neural network models that are based on a first-order approximation of spectral convolutions on graphs. Nevertheless, the real graph-structured data can be both structurally large and complex, the method of aggregating all neighbor features equally will introduce a lot of noise, which will pose a challenge for effective graph mining. Another popular variant called graph attention network (GAT) [14] incorporates the attention mechanism into the propagation function, it computes the hidden representations of each node in the graph by attending over its neighbors, following a self-attention strategy. As opposed to GCNs, GAT allows for assigning different importances to different neighbors of the same neighborhood in the role of attention, which can ignore noisy parts of the graph and improve the interpretability of the model. In the medication recommendation task, the correlations between different clinical events can be naturally handled by using the graph attention mechanism.

B. TEMPORAL MODELING OF SEQUENTIAL DATA

There are many approaches to model temporal dependencies for sequential data such as Markov chain, hidden Markov model, vector autoregressive model and neural models. In recent years, recurrent neural networks (RNN) and their variants e.g. long-short term memory (LSTM) are widely leveraged to learn temporal dependency for sequential data, which have been successful in modeling long sequences.

In real life, there are many application scenarios where data has both temporal and network structure, for example, traffic situation prediction based on real-time spatial road condition information, action recognition with human joints as nodes and continuous frames as a temporal sequence in the video, a series of continuous transaction prediction between different people in the financial field, and the prediction of social network users' friendship development over time. These scenarios have led to a lot of research [16]–[22], [46] on temporal modeling for graph-structured sequence data. Most of these networks in the real-world are naturally dynamic, which means that they evolve over time, and nodes and links may appear or disappear. The spatio-temporal dynamic framework is now widely used in a variety of real-world scenarios. For example, traffic situation prediction, action recognition, financial transaction prediction, social network evolution, etc. Researches in these scenarios breed a lot of temporal modeling methods for graph-structured data. Nguyen *et al.* [30] suggested using temporal random walks and the skip-gram model for learning node embeddings, which constrained the random walk to follow the order of time. Trivedi *et al.* [31] presented Know-Evolve model which modeled the occurrence of a fact as a temporal point process. Trivedi *et al.* [32] extended Know-Evolve model with a two-time scale process that captured temporal node interactions in addition to the topological evolution. Goyal *et al.* [28] incrementally built node representations by initializing an autoencoder from the previous time step. Hawkes process based Temporal Network Embedding (HTNE) [29] used the concept of a Hawkes process [37] to model the dynamism with an attention mechanism to capture the influence of historical neighbors on the current neighbors.

Recently, some researchers have attempted to combine graph models with RNN-based modules to make up for the lack of sequential modules in modeling the high-level network structure, which has shown good results in many tasks. Graph Convolutional Recurrent Network (GCRN) [38] merged a convolutional neural network for graph-structured data and LSTM to simultaneously identify meaningful spatial structures and dynamic patterns. Goyal *et al.* [35] used sparsely connected LSTM networks to learn the temporal transitions in every node of the network. In these two methods, the output representation matrix of the graph model was then directly inputted into the LSTM network. In other words, it can be treated as applying the LSTM on each column of the representation matrix independently. Recurrent Graph Convolutional Neural Network (RgCNN) [39] combined convolutional neural networks and LSTM, which designed a flattened dense layer behind the output of the convolutional neural network. Taheri *et al.* [34] used an average pooling of the nodes' hidden states after message propagation in the graph model as the representation of the entire graph. Pareja *et al.* [33] presented EvolveGCN, it introduced a new method to combine GNN with the GRU. The heart of EvolveGCN is that the weight matrix in the propagation function will be updated over time using the GRU network. In our work, we consider modeling the graph-structured sequential data by using the temporal updating module.

III. METHODOLOGY

A. PROBLEM DEFINITION

1) DEFINITION OF PATIENT RECORDS

The EHR of each patient can be represented as a set of temporal admission sequences: $\boldsymbol{E}^n = \{\boldsymbol{x}_1^n, \boldsymbol{x}_2^n, \ldots, \boldsymbol{x}_{T(n)}^n\}$, where $T(n)$ is the number of admissions of the n-th patient. To avoid clutter, we will describe the method for a single patient, and we omit n in the notation if there is no ambiguity. Each admission sequence $\boldsymbol{x}_t = \{\boldsymbol{d}_t, \boldsymbol{p}_t, \boldsymbol{m}_t\}$ is a collection of codes that contains all the diagnosis event codes $\boldsymbol{d}_t$, procedure event codes $\boldsymbol{p}_t$ and medication prescription event codes $\boldsymbol{m}_t$ at the t-th admission, where $\boldsymbol{d}$ refer to the collection of codes corresponding to the diagnosis symptoms recorded like acute renal failure and anemia, $\boldsymbol{p}$ refer to the collection of codes corresponding to various examinations and operations performed such as liver transplantation, liver biopsy, etc, and $\boldsymbol{m}$ refer to the collection of codes corresponding to the medications (eg insulin, cardiac glycoside) prescribed according to the patient's condition.

2) MEDICATION RECOMMENDATION TASK

Given a patient's history admission sequences $\boldsymbol{E}_{1:t-1} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \ldots, \boldsymbol{x}_{t-1}\}$, and the diagnosis and procedure codes of this admission $\boldsymbol{x}_t = \{\boldsymbol{d}_t, \boldsymbol{p}_t\}$ at time t. Our goal is to generate a medication combination $\boldsymbol{y}_t \in \{0, 1\}^L$ at time t according to the patient's current clinical events $\boldsymbol{x}_t$ and historical EHRs $\boldsymbol{E}_{1:t-1}$, which can be regarded as a multi-label classification task since the number of labels can be more than one, and L is the number of candidate medications.

B. GRAPH CONSTRUCTION

To construct the graph structure of the clinical events for each admission of a patient, we first need to represent the global correlations between these events. The graph construction process can be divided into two stages as shown in the Fig. 2.

Stage I: We first need to construct a global guidance correlation graph G, where each node is a clinical event code. These nodes include all diagnosis event codes, procedure event codes and medication prescription event codes that ever appeared in the dataset. Edges are based on co-occurrence probability between events in each admission of every patient used for guidance.

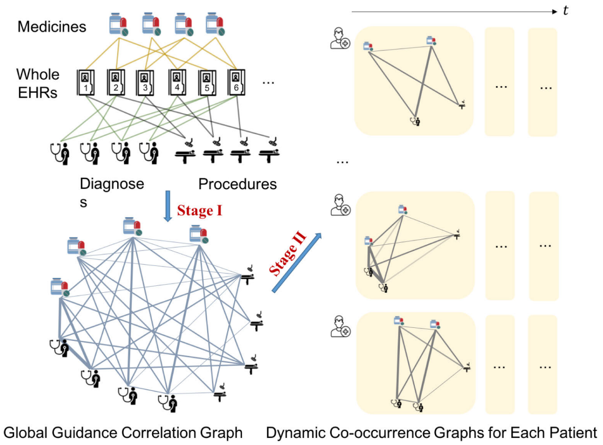

Medicines
Whole
EHRs
1
2
3
4
5
6
...
Diagnose
s
Procedures
Stage I
Stage II
t
...
Global Guidance Correlation Graph
Dynamic Co-occurrence Graphs for Each Patient

FIGURE 2. Two stages of building the dynamic co-occurrence graphs for every patient from the whole original EHR data.

TABLE 1. Notations used in this paper.

Notation	Description
$\boldsymbol{E}$	EHR of a patient
$\boldsymbol{x}$	clinical event codes of a patient
$\boldsymbol{c}$	multi-hot vector of $\boldsymbol{x}$
L	total number of medications
N	total number of clinical events
$\boldsymbol{G}$	global guidance correlation graph
$\boldsymbol{M}$	adjacency matrix of G
$\boldsymbol{A}$	adjacency matrices of a patient's EHR
$\boldsymbol{H}^0$	medical embeddings
$\boldsymbol{H}$	medical representation at GAM module
α	attention coefficient
$\boldsymbol{G^S}$	selective update gate
$\boldsymbol{G^I}$	incremental update gate
$\hat{\boldsymbol{F}}_{0,t}$	medical representation at TDU module
$\boldsymbol{F}_t$	output of 2D-GRUcell
$\boldsymbol{O}$	feature bag at MIML module
$\boldsymbol{S}$	instance-label scoring matrix
$\hat{\boldsymbol{y}}_t$	predicted medication combination
y_t	ground truth of medication combination

We define an adjacency matrix $\boldsymbol{M} \in \mathbb{R}^{N \times N}$ to clarify the construction of the graph, where N is the total number of clinical events in the dataset. Here we employ the Point-wise Mutual Information (PMI) value to calculate the weight of the edges. Formally, the edge weight between node i and node j at time t is defined as follows:

$$\boldsymbol{M}(i,j) = \begin{cases} PMI(i,j), & \text{with } PMI(i,j) > 0 \\ 0, & \text{else} \end{cases} \tag{1}$$

and the PMI value is computed as:

$$PMI(i,j) = \log\left(\frac{d(i,j)}{d(i)d(j)} \cdot |D|\right) \tag{2}$$

where $d(i,j)$ is the total number of admission records that events i and j co-occurred. $d(i)$ and $d(j)$ are the total number of admission records that i and j have appeared at least once respectively, and $|D|$ is the total number of admission records. Note that events i and j here can belong to the same type of event, such as two diagnosis events, or events of different types, such as diagnosis events and medication prescription

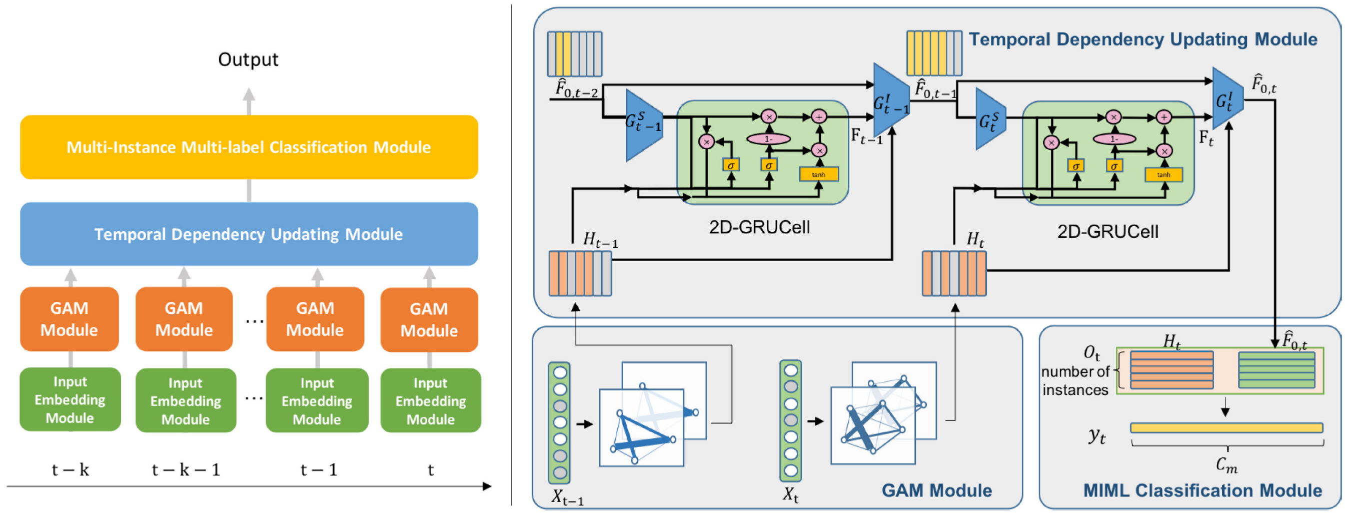


FIGURE 3. The framework of our proposed method. The left half-block is the overall structure of our model which mainly contains four modules, and the right half-block depicts the module details of the last two time steps. Input data consists of related clinical events of the patient's admissions ordered by time. It is first transformed into input features through the embedding layer. Then we construct the dynamic correlation graphs based on input data and feed them into the graph-attention augmented module (GAM) by time. The temporal updating module models the temporal evolution of each event contained in the output matrix of the GAM at different time steps. Finally, an overall representation matrix is obtained incrementally and is fed into the multi-instance multi-label classification module to obtain the predicted output.

events. At this stage, we have obtained a weight guidance graph that reflects the degree of correlation between various events.

Stage II: We construct dynamic co-occurrence graphs from each patient's history admission sequences $\boldsymbol{E}_{1:t-1} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \ldots, \boldsymbol{x}_{t-1}\}$ and the clinical events occurred at current admission $\boldsymbol{x}_t = \{\boldsymbol{d}_t, \boldsymbol{p}_t\}$, which is specifically expressed as a sequence of adjacency matrices $\boldsymbol{A} = \{\boldsymbol{A}_1, \boldsymbol{A}_2, \ldots, \boldsymbol{A}_t\}$. The co-occurrence graph at each time step can be considered as a local mapping of the global guidance co-occurrence graph. Due to the limited space, only one of the patient's dynamic co-occurrence graphs of multiple admission records is drawn in Fig. 2. Specifically, at each time step, the adjacency matrix $\boldsymbol{A}_t$ is a fully connected graph, where the nodes represent all clinical events in the patient's EHRs including diagnosis events $\boldsymbol{d}_t$, procedure events $\boldsymbol{p}_t$ and medication prescription events $\boldsymbol{m}_t$, while the edge weight here is not set to all 1, but calculated according to the global guidance co-occurrence matrix. The specific calculation method is as follows:

$$A_t[i,j] = \begin{cases} \boldsymbol{M}(i,j), & \text{if } i, j \text{ in admission } \boldsymbol{x}_t \\ 0, & \text{else} \end{cases} \tag{3}$$

With the constraint of dynamic co-occurrence graphs, we can effectively establish the association between clinical events at each admission. The prior statistical guidance can enhance the representation of sparse categories, which alleviates the problem of insufficient training data of some event types.

C. MODEL FRAMEWORK

After constructing the co-occurrence correlation graphs, we then introduce the framework to model the structural correlations and temporal dependencies simultaneously. Note that *structural* refers to the co-occurrence relatedness of multiple events from diagnoses, procedures and medications, and *temporal* refers to the evolution of clinical events in each admission of the patient over time. The framework mainly includes the input embedding module, graph-attention augmented module, temporal dependency updating module and multi-instance multi-label classification module. Fig. 3 provides an overview of our proposed method.

1) INPUT EMBEDDING MODULE

The input embedding module first maps all the clinical event codes $\boldsymbol{x_t}$ of the patient at the t-th admission into a feature matrix $\boldsymbol{H_t}^0 \in \mathbb{R}^{|\boldsymbol{c}_t| \times d}$ as follows, where $|\boldsymbol{c}_t|$ is the total number of clinical event codes of the patient at t-th admission and d is the feature dimension.

$$\boldsymbol{H_t}^0 = \boldsymbol{W}_e \boldsymbol{c_t} \tag{4}$$

where $\boldsymbol{W}_e \in \mathbb{R}^{N \times d}$ is the learnable embedding matrix, $\boldsymbol{c_t}$ is a multi-hot vector representing the existence of each clinical event at the t-th admission, N is the total number of the clinical events in the whole dataset.

2) GRAPH-ATTENTION AUGMENTED MODULE (GAM)

Given the co-occurrence correlation matrix $\boldsymbol{A}_t$ at time t, we use a two-layer graph-attention neural network to encode node features, which allows the encoded event node vectors to contain the information of other co-occurrence events at the same admission with different degrees of correlation to obtain a more comprehensive representation. At each layer, it embeds a set of node representations $\boldsymbol{H}_t = \{\boldsymbol{h}_{t,1}, \boldsymbol{h}_{t,2}, \ldots, \boldsymbol{h}_{t,|\boldsymbol{c}_t|}\}$ by recursively aggregating information from their neighbors. Formally, the node representation of every event code at the l-th graph attention layer $\boldsymbol{h}^l_{t,i}$ can be obtained through the propagation function as follows:

$$\boldsymbol{h}^l_{t,i} = \sigma\left(\sum_{j\in\mathcal{N}_i} \alpha_{ij}\boldsymbol{h}^{l-1}_{t,i}\boldsymbol{W}^l + \boldsymbol{b}^l\right) \tag{5}$$

where σ is a non-linear activation function(we use ReLU in this paper), $\boldsymbol{W}$ and $\boldsymbol{b}$ are learnable parameters, and $\mathcal{N}_i$ is the neighborhood of node i in the graph. The function mainly follows the traditional GAT [14], in which the new node representation $\boldsymbol{h}^l_{t,i}$ of event code i contains both the previous layer representation $\boldsymbol{h}^{l-1}_{t,i}$ and a weighted aggregation of neighbor representations $\boldsymbol{h}^{l-1}_{t,j}$. The core of GAM is the calculation of the attention coefficient, which determines the importance of each neighbor's feature to the current node. The traditional implementation of calculating the attention coefficient is as follows:

$$\alpha_{ij} = \frac{\exp(\text{LeakyReLU}(\overrightarrow{\boldsymbol{a}}^T[\boldsymbol{W}\overrightarrow{\boldsymbol{h}}_i||\boldsymbol{W}\overrightarrow{\boldsymbol{h}}_j]))}{\sum_{k\in N_i}\exp(\text{LeakyReLU}(\overrightarrow{\boldsymbol{a}}^T[\boldsymbol{W}\overrightarrow{\boldsymbol{h}}_i||\boldsymbol{W}\overrightarrow{\boldsymbol{h}}_k]))} \tag{6}$$

where $\boldsymbol{a}$ is a trainable attention vector, $\boldsymbol{W}$ is a trainable weight matrix mapping the input features to the hidden space, and $||$ represents concatenation. In our method, since the co-occurrence correlation graph has been constructed and the correlation between events has been established, the calculation of the attention coefficient here can be simplified as follows:

$$\alpha_{ij} = \frac{\exp(\boldsymbol{A}_t[i,j])}{\sum_{k\in N_i}\exp(\boldsymbol{A}_t[i,k])} \tag{7}$$

As stated in the previous section, the co-occurrence correlation matrix $\boldsymbol{A}$ represents the correlations between different clinical events at each admission. Through the above equation, we assign different weights to other clinical events for the updating of each clinical event representation in the current admission, which can model the correlations among various clinical events such as diagnoses, procedures and medications. The GAM module in the framework makes our model different from the way of treating events as an independent set of vectors. With such a two-layer graph attention neural network, each medical code can iteratively accumulate first-order and second-order neighborhood code information from various dynamic structures. The ability of graph attention mechanism to encode higher-order structural features enriches the representation of sparse features, and to some extent makes up for the problem of sparse training data.

Note that the attention matrix $\boldsymbol{A}$ changes with the evolution of clinical events over time. The GAM module needs to simultaneously model different co-occurrence structures at different time steps, parameters are shared across all temporal and spatial co-occurrence correlation structures.

3) TEMPORAL DEPENDENCY UPDATING MODULE (TDU)

The GAM module outputs the node representations as $\{\boldsymbol{H_1}, \boldsymbol{H_2}, \ldots, \boldsymbol{H_t}\}$ at each time step that can capture the structural correlations, where $\boldsymbol{H_t} \in \mathbb{R}^{|\boldsymbol{c}_t|\times d}$ contains all the node representations at t-th admission. These representations will then be processed by a temporal dependency updating module. In our task, the nodes at each time step represent various diagnosis events, procedure events, and medication prescription events for the current admission, which are very different, and the temporal evolution among different nodes at various time steps may have dependencies. For example, cerebral aneurysms can cause arachnoid hemorrhage, but they are not directly related to gastrointestinal diseases which may also appear at this admission. At each time step, the update of each node should selectively remember the information of some historical nodes. In addition, some clinical events occurred only over some time, and some persist throughout. We propose a temporal dependency encoding module to comprehensively consider the clinical events experienced by all historical admissions of the patient, and embed all nodes that have ever occurred up to time step $t-1$ to have the final historical representation of size $|\boldsymbol{c}| \times d$.

For that, we capture the temporal development of different kinds of diseases and treatments, as well as the long-term historical record information by using GRU as the basic model. Note that other types of RNN, e.g. LSTM, can also be choices. Since the number of clinical events is very large, it is not practical to set a weight parameter for each node, which may cause parameter explosion and overfitting. Inspired by the temporal point process and the gating mechanism [31], [37], we propose a selective update gate $\boldsymbol{G^S}$ and an incremental update gate $\boldsymbol{G^I}$. Specifically, the temporal dependency updating process is formulated as the function of incremental linear convex combination from the node vector representations of temporal consecutive events, then the final output at time t is calculated as follows:

$$\hat{\boldsymbol{F}}_{0,t} = \boldsymbol{G^I}_t\hat{\boldsymbol{F}}_{0,t-1} + (1 - \boldsymbol{G^I}_t)\boldsymbol{F}_t \tag{8}$$

where $\boldsymbol{F}_t$ is the output of 2D-GRU cell at time t, and $\boldsymbol{G^I}_t$ controls the update in the node dimension, it determines how much (or how less) the previous information of each clinical event impacts on the current encoding result of the corresponding event code. In this way, the output of each time step could be considered as an incremental update with the node dimension, which can retain the information of all event nodes ever occurred in the past, and model the preservation and forgetting of each clinical event respectively. Locally, the new hidden layer of each event code is obtained from the current input feature and the hidden node state matrix at the previous time step after passing the selective update gate $\boldsymbol{G^S}_t$. Then $\boldsymbol{G^S}_t$ will calculate the similarity of each input clinical event code at time t and each clinical event code in the previous hidden state at time $t-1$, and the hidden state of each event code at time t can be captured by a weighted average of the hidden states of clinical event codes with high similarity with it.

In this part, we model the evolution of different clinical events over time, so that for each clinical event code, the hidden state of 2D-GRU cell at each time step selectively

Algorithm 1 Training Algorithm

```
Input: Training dataset $\mathcal{D}$, training epoches $N$;
Output: Optimal parameters $\theta^*$;
Construct the global guidance correlation graph $G$ and the adjacency matrix $\boldsymbol{M}$;
Initialize: Transform clinical codes to vectors, initialize the model parameters using uniform distribution;
for $i = 0$ to $N$ do
    for $j = 0$ to $|\mathcal{D}|$ do
        Sample a patient record $\boldsymbol{E} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_{T(j)}\}$ from $\mathcal{D}$;
        Construct the dynamic co-occurrence adjacency matices $\boldsymbol{A}$;
        for $t = 2$ to $T(j)$ do
            Use (4) to get medical embeddings $\boldsymbol{H}^0$;
            Use (5),(7) to get $\boldsymbol{H}_1, \boldsymbol{H}_2, \ldots, \boldsymbol{H}_t$ through GAM module;
            Use (8),(9) to get $\hat{\boldsymbol{F}}_{0,t}$ though TDU module;
            $\boldsymbol{O}_t \leftarrow concat(\hat{\boldsymbol{F}}_{0,t}, \boldsymbol{H}_t)$;
            Use (10) to calculate the medication prediction $\hat{\boldsymbol{y}}_t \leftarrow \mathrm{MIML}(\boldsymbol{O}_t)$;
        end for
        Use (11) to update the parameters $\theta \leftarrow \theta - \Delta_\theta \mathcal{L}(\theta)$;
    end for
end for
Return $\theta^* \leftarrow \theta$;
```

considers various clinical events at the previous time step. We reiterate the whole formulation of our TDU module as follows:

$$\begin{aligned}
\boldsymbol{R}_t &= \mathrm{Sigmoid}(\boldsymbol{W_r} * [{\boldsymbol{G}^S}_t\boldsymbol{F}_{t-1}, \boldsymbol{H}_t] + \boldsymbol{B_r}) \\
\boldsymbol{Z}_t &= \mathrm{Sigmoid}(\boldsymbol{W}_z * [{\boldsymbol{G}^S}_t\boldsymbol{F}_{t-1}, \boldsymbol{H}_t] + \boldsymbol{B}_z) \\
\tilde{\boldsymbol{F}}_t &= \tanh(\boldsymbol{W} * [\boldsymbol{R}_t * {\boldsymbol{G}^S}_t\boldsymbol{F}_{t-1}, \boldsymbol{H}_t] + \boldsymbol{B}) \\
\boldsymbol{F}_t &= (1 - \boldsymbol{Z}_t) * {\boldsymbol{G}^S}_t\boldsymbol{F}_{t-1} + \boldsymbol{Z}_t * \tilde{\boldsymbol{F}}_t \\
\hat{\boldsymbol{F}}_{0,t} &= {\boldsymbol{G}^I}_t\hat{\boldsymbol{F}}_{0,t-1} + (1 - {\boldsymbol{G}^I}_t)\boldsymbol{F}_t
\end{aligned} \tag{9}$$

Note that unlike the standard GRU where both the input and output are vectors, the input and output from the module are now matrices, and both gates $\boldsymbol{G}^I$ and $\boldsymbol{G}^S$ are node-aware functions.

4) MULTI-INSTANCE MULTI-LABEL CLASSIFICATION

Overall, our medication recommendation task can be seen as a multi-instance multi-label classification task by considering the multi-events as input instances and multi-medicines as output labels. All previous modules are instance-level modeling, and the output of the TDU module can be regarded as a bag of multiple node features where each node feature integrates the structural and temporal characteristics of the co-occurrence graphs. Then we integrate the output features of all the clinical events by the previous modules to make the final prediction.

Firstly, we select the node features contained in the last admission record from the output of TDU module $\hat{\boldsymbol{F}}_{0,t}$ to represent the node representations of temporal characteristics, then concatenate the output of the GAM module at the last admission record $\boldsymbol{H}_t$ as a simple structural encoding representation as our final feature bag $\boldsymbol{O}_t$ which contains both structural and temporal information of EHR. In this module, we make the final multi-label classification for this feature bag. Inspired by the DeepMIML [12] model that applies deep learning to MIML, we introduce our MIML classification module to discover instance-label relationships. Concretely, for the feature bag $\boldsymbol{O}_t$, we proposed a fully connected layer of size $K * C_m$, in which the matching score $\boldsymbol{s}_{ij}$ of the i-th clinical event instance and the j-th medication label is calculated as follows:

$$\boldsymbol{s}_{ij} = \mathrm{Relu}\left(\sum_{k=1}^{K} \boldsymbol{w}_{jk}\boldsymbol{o}_{ik} + \boldsymbol{b}_j\right) \tag{10}$$

where $\boldsymbol{w}_{jk}$ can be interpreted as the matching weight for the k-th sub-feature of the j-th medication. A max-pooling operation on this 2D instance-label scoring matrix $\boldsymbol{S} \in \mathbb{R}^{C_m \times N}$ is conducted to obtain the matching scores for labels on bag level, which will produce our final prediction $\hat{\boldsymbol{Y}}$ of size $L * 1$. Here the entry $\hat{\boldsymbol{y}}_m$ is the matching score for the m-th medication label on the whole EHR of the current patient.

D. OPTIMIZATION

Our goal is to predict the medication combination at each time step $\hat{\boldsymbol{y}}_t \in \{0, 1\}^{C_m}$ where $t \geqslant 2$. The loss function in our model can be selected as the binary cross-entropy function as follows:

$$\mathcal{L} = -\frac{1}{T-1}\sum_{t=2}^{T}\left(\boldsymbol{y}_t^T \log(\hat{\boldsymbol{y}}_t) + (1 - \boldsymbol{y}_t^T)\log(1 - \hat{\boldsymbol{y}}_t)\right) \tag{11}$$

where y_t is the ground truth and $\hat{\boldsymbol{y}}_t$ is the model prediction. Due to a large number of samples and parameters, the landscape of the loss function can be very complicated. Thus, we adopt Adam [43] optimizer to minimize the loss function in (11) which combines the advantages of two optimization algorithms, AdaGrad [44] and RMSProp [45]. The optimization considers the first-order moment estimation and second-order moment estimation of the gradient, and can adaptively tune the learning rate. The training algorithm is shown in detail in Algorithm 1. In the optimization process, the parameters of the two modules GAM and TDU will be optimized at the same time and affect each other, then the final model can learn the parameters that can capture the structural and temporal correlation information and achieve the global optimal solution.

IV. EXPERIMENTS

A. EXPERIMENT SETUP

1) DATASET

In the experiments, we employ the freely-available database MIMIC-III [36] (Medical Information Mart for Intensive Care III) comprising de-identified health-related data associated with over forty thousand patients who stay in critical care units of the Beth Israel Deaconess Medical Center between 2001 and 2012. We utilize multiple tables like medication orders, procedure orders and prescription lists and processed them into a patient's temporal list of diagnoses, medications, and treatment procedures. Since the dataset records the clinical record in the intensive care unit (ICU), the first 24 hours in ICU is often the most critical time for patients to get correct treatment quickly, so we choose a set of medications prescribed by doctors in the first 24 hours after admission to ICU. And we transform the drug coding from NDC to ATC Third Level for integrating, and for the diagnosis codes and procedure codes, and use the ICD-9 codes. Then we further divide the processed data into single admission and multiple admissions, and the statistics of the data information are listed in Table. 2.

TABLE 2. Statistic of the Dataset MIMIC-III.

# of patients(single-visit)	28,936
# of clinical event codes	3,321
# of patients(multi-visit)	6,350
# of diagnose codes	1,958
# of medicine codes	145
# of procedure codes	1,426
max # of visits	29
avg # of visits	2.36

2) BASELINES

We consider several baseline algorithms in the experiments for comparison as follows:

Logistic Regression (LR) is a logistic regression model with L1/L2 regularization. Here we represent sequential multiple medical codes by the sum of a multi-hot vector of each visit. Binary relevance technique [26] is used to handle the multi-label output.

Learn to Prescribe (Leap) [15] is a method to decompose the treatment recommendation into a sequential decision-making process while automatically determining the appropriate number of medications. A recurrent decoder is used to model label dependencies and content-based attention is used to capture label instance mapping.

Reverse Time AttentIon Model (RETAIN) [10] is a two-level neural attention model for sequential data which relies on an attention mechanism to represent the behavior of physicians during an encounter.

Graph Augmented Memory Networks (GAMENet) [2] is the method that integrates the Drug-Drug Interactions(DDI) knowledge graph by a memory module implemented as a graph convolutional network. For a fair comparison, we use a variant of GAMENet without DDI knowledge.

G-Bert [1] is a model that integrates the GNN representation into a Transformer-based encoder with pre-training on single-visit EHR data. For a fair comparison, we also use a variant of G-Bert without codes' ancestors information.

3) METRICS

For the measurement of experimental results, we use the Jaccard Similarity Score (Jaccard), Average F1 (F1) and Precision Recall AUC (PRAUC) as the metrics. Jaccard is defined as the size of the intersection divided by the size of the union of the predicted set $\hat{Y}_t^{(k)}$ and the ground truth set $Y_t^{(k)}$ as follows:

$$Ja(\hat{Y}_t^{(k)}, Y_t^{(k)}) = \frac{1}{\sum_k^N \sum_t^{T_k} 1} \sum_k^N \sum_t^{T_k} \frac{|Y_t^{(k)} \cap \hat{Y}_t^{(k)}|}{|Y_t^{(k)} \cup \hat{Y}_t^{(k)}|} \tag{12}$$

where N is the number of patients in the test set and T_k is the number of admissions of the k-th patient. PR-AUC is computed by trapezoidal integral for the area under the PR curve. For the dataset where the number of positive and negative samples are imbalance, the precision-recall curve has shown to be a proper metric. And F1 here is defined as follows:

$$F1 = \frac{1}{\sum_k^N \sum_t^{T_k} 1} \sum_k^N \sum_t^{T_k} \frac{2 \times P_t^{(k)} \times R_t^{(k)}}{P_t^{(k)} + R_t^{(k)}} \tag{13}$$

where $P_t^{(k)} = \frac{|Y_t^{(k)} \cap \hat{Y}_t^{(k)}|}{|Y_t^{(k)}|}$, $R_t^{(k)} = \frac{|Y_t^{(k)} \cap \hat{Y}_t^{(k)}|}{|\hat{Y}_t^{(k)}|}$.

4) IMPLEMENTATIONS

We use all the single-visit EHR data to construct the global guidance correlation graph. And for the multi-visit EHR data, we select 2/3 of the data for training, and the rest is divided into evaluation sets and test sets. We set the initial embedding size and hidden size of GRU to 64. For the 2-layer GAM module, the hidden layer dimension of the first layer is set to 128, and the second layer is set to 64. The dropout rate is set to 0.2.

B. EXPERIMENT RESULTS

We evaluate the performances of our proposed model in terms of three metrics. We also further analyze the effectiveness of each module and demonstrate the case studies that highlight the advantages of our proposed model.

1) PREDICTION PERFORMANCES

Table. 3 lists the performances from various methods over the dataset in the medication prediction task. Experiments show that our model can achieve the best results among all the methods. Specifically, our proposed method outperforms the best state-of-the-art approach (GBert) by 3.9%, 2% and 2.6% in terms of Jaccard, PR-AUC and F1 score. From the comparison methods, this is already a considerable improvement for this multi-label classification task. Additionally, we can see that the average number of medicines recommended by our model is closest to the true number 15.02. Moreover, our method achieves the best results using fewer parameters than all the other methods, which shows the powerful structural and temporal modeling ability of graph neural networks.

TABLE 3. Medication prediction performance of different methods on MIMIC-III. Note that the gold average number of medicines on the test set is 15.02. *DDI*$^+$ and *Ans*$^+$ indicates that these methods use additional drug interaction information or ontology information. Full name and description of these models are stated in Baselines.

Methods	Jaccard	PR-AUC	F1	Avg # of Med.	Parameters
LR	0.4075	0.6716	0.5658	11.42	-
LEAP	0.3844	0.5501	0.5410	14.42	436,884
RETAIN	0.4451	0.6751	0.6043	15.86	298,770
GAMENET	0.4503	0.6906	0.6077	13.44	452,434
G-BERT	0.4299	0.6771	0.5903	16.72	2,634,145
GAMENET(DDI^+)	0.4516	0.6961	0.6096	13.45	461,714
G-BERT(Ans^+)	0.4565	0.6960	0.6152	16.02	3,034,045
GATE	**0.4742**	**0.7087**	**0.6315**	15.53	298,385

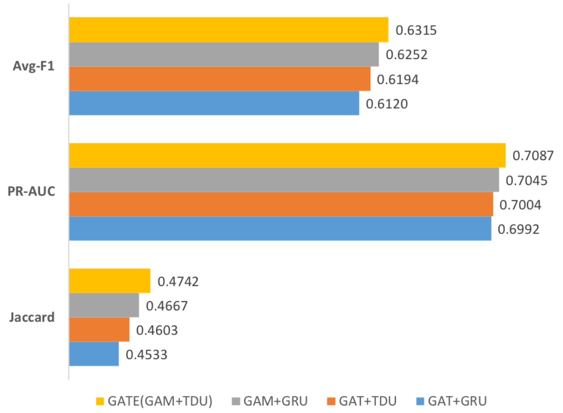


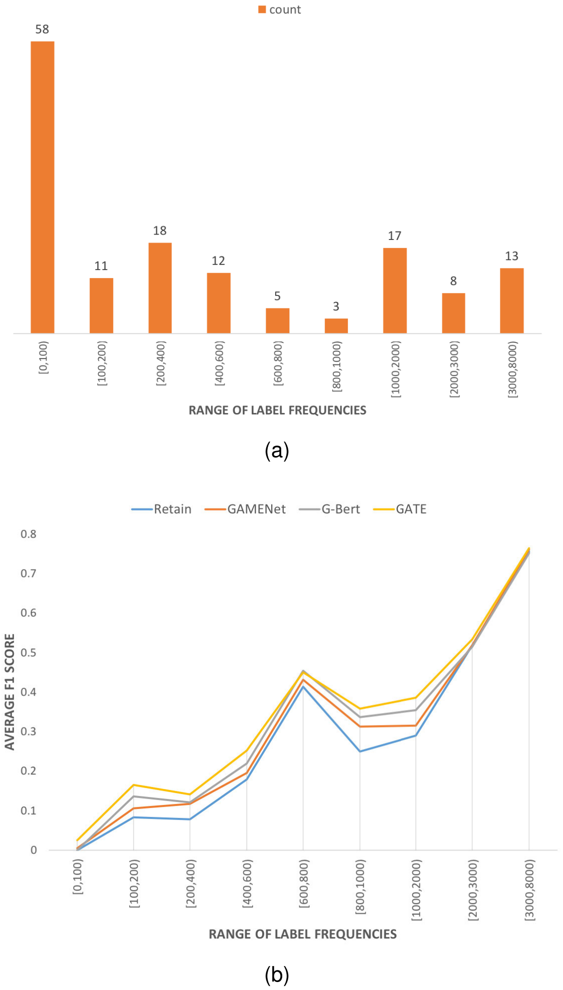
FIGURE 4. Performance comparison among different variants of our method.

2) MODULE EFFECTS

To further investigate the effectiveness of each module component in our proposed framework, we compare the method with its variants as follows:

- **GAT+GRU**: This is the most naive version, which uses the original GAT to encode the structural features, and averages the output feature matrix into a standard GRU module.
- **GAT+TDU**: In order to verify the effectiveness of the global guidance correlation graph, we use a classic graph model, that is, the traditional graph attention mechanism to replace our GAM module, and the correlation between nodes now is obtained through the self-attention mechanism.
- **GAM+GRU**: In order to clarify the effectiveness of the Temporal Updating module, we simply remove the two updating gates from the TDU module, so it degenerates into a standard GRU module.

Fig. 4 shows the different improvement of these variants on the prediction results, Comparing the result of GAT+TDU and our final GAM+TDU framework in Fig. 4 and the results in Table 3, we find that the standard graph attention network is already able to capture the correlation between clinical events well and achieve better results. And when the guidance co-occurrence graph is introduced, the ability of the module to capture the inherent structural features from multiple clinical events can be enhanced, so as to further achieve better prediction results. We then compared the results of GAM+GRU



(a)

(b)

FIGURE 5. (a) The total number of labels in a different range of label frequencies. (b) Comparison of averaged F1 scores predicted separately by different methods for labels in different frequency ranges.

TABLE 4. Recommended medications for a patient with three admissions through different methods. Here "unseen" refers to the medications that are predicted but not in the ground truth, while "missed" indicates the medications that are in the ground truth but are not predicted. Full name and description of these methods are stated in Baselines.

Methods	Recommended Medication Combination
Retain	23 correct, 5 unseen, 4 missed(Potassium, Thyroid preparations, Cardiac glycosides, Dopaminergic agents)
GAMENet	19 correct, 3 unseen, 7 missed(Thyroid preparations, Anxiolytics, Cardiac glycosides, Dopaminergic agents, Antibiotics, Antigout, ...)
G-Bert	20 correct, 1 unseen, 6 missed(Potassium, Propulsives, Anxiolytics, Dopaminergic agents, Antibiotics, Antigout)
GATE	**26 correct, 4 unseen, 1 missed(Antibiotics)**

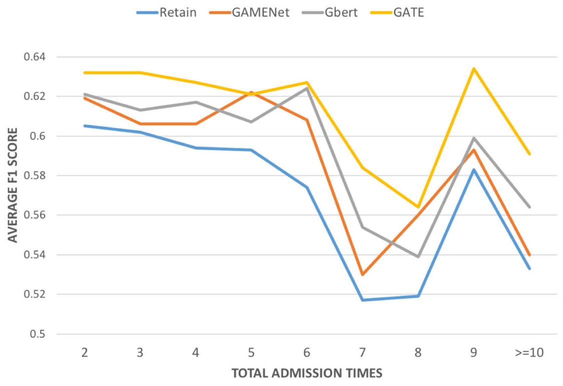


FIGURE 6. Evaluation for data of different temporal length.

with the final framework, the results show that our temporal updating module can better encode the temporal evolution of different events for this task based on the better prediction results.

3) EVALUATION FOR UNBALANCED LABELS

Due to the limitation of EHR data, there exists the problem of unbalanced labels, which causes the difficulty of predicting specific medications that appear infrequently. Our model, by establishing a dynamic correlation evolving of time and structure between clinical events, is expected to improve the prediction with small samples. Fig. 5(a) counts the number of labels in different frequency ranges, which can be seen that 58 of the 145 medication types appear in the training set less than 100 times, and some appear up to several thousand times. We calculated the averaged F1 scores of medication prediction results from the evaluation methods for different frequency ranges. As can be seen from Fig. 5(b), our method has significantly improved the prediction of less frequent medications than the other methods. In addition, it also shows that the labels with a more pronounced prediction effect are mainly dedicated medications for specific diseases, such as pathological addiction drugs, insulin, etc.

4) EVALUATION FOR DATA OF DIFFERENT TEMPORAL LENGTH

Since the number of admissions for each patient is different, the length of the time series to be processed should be considered. Based on the data statistics in table 2, the longest sequence length is 29, which makes it very difficult for the temporal modeling module. Fig.6 shows that our method outperforms all the other methods for various temporal lengths. In particular, it has significant improvement for the data of long sequences(>6) compared with other methods, which shows that our method has a better ability for modeling long historical dependency in the records.

C. CASE STUDY

In order to observe the effects of our model concretely and intuitively, we select a patient's EHRs of three temporal admissions in the test set for case study. This patient has various symptoms such as sepsis, heart failure and gout. Table. 4 summarizes the accuracy of the prediction results of our proposed model and other methods at the third admission. On the whole, our method predicts 26 out of 27 medications, only one medication is missed and four are unseen in the ground truth medications prescribed by doctors. It can be seen from Table. 4 that some medications like Antigout, Cardiac glycoside and Anxiolytics are often omitted from the prediction results from the other methods. To verify the interpretability of our model, we draw the dynamic co-occurrence correlation graph over time based on the EHR data.

For the convenience of observation, Fig.7 shows the subgraphs of the dynamic co-occurrence correlation graph at each admission. Each node represents a clinical event, and the width of the edge represents the degree of correlation between the two events. It can be seen from the figure that the dynamic graph well represents the correlations between different clinical events in each admission record, which can distinguish the events of various diseases and aggregate the events of the same diseases. Taking gout as an example, the highlighted nodes are the clinical events related to gout in each medical record. In the first two admission records, we can associate the diagnoses gout(3), osteoarthritis(31) with the anti-gout medication(70) through the structural modeling of a 2-layer dynamic graph attention mechanism. In the last admission record, the patient is diagnosed with acute gouty pathway(29) and underwent an arthrocentesis(44). Then we use a heatmap to show the correlation distribution of the two events with historical events in the G^S of temporal modeling at the last time step. As shown in Fig.8, both of the two clinical events tend to focus on osteoarthritis. Thus, although they do not appear in the patient's history records, the relationship between them and historical events through the temporal updating module can also be effectively established.

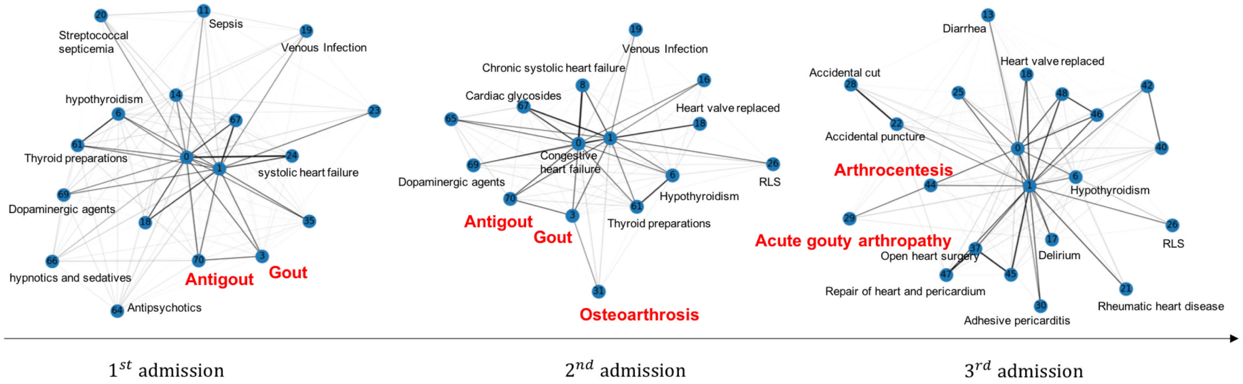


FIGURE 7. The subgraphs of the dynamic co-occurrence correlation graph constructed at each admission. Each node represents a clinical event, and the width of the edge represents the degree of correlation between the two events.

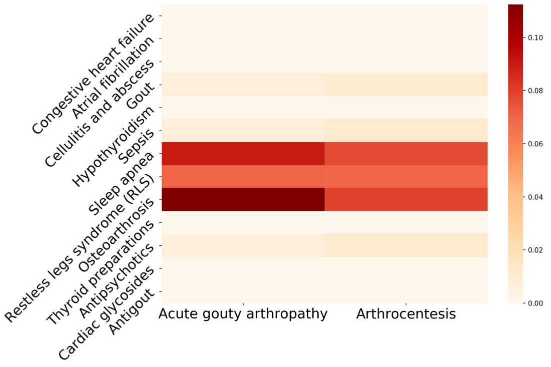


FIGURE 8. Heatmap of the correlation of the acute gouty pathway and underwent an arthrocentesis to some historical events. The color depth of the block reflects the degree of correlation between the two events.

V. CONCLUSION AND FUTURE WORK

In this work, we propose a novel method for multiple comorbidity medication recommendation, which simultaneously captures the temporal and internal structural features for every patient. For that, we build a graph attention module to establish the correlations between clinical events and enhance the characteristics of clinical events related to different types of diseases, and further propose a temporal updating module to learn discriminative representations of different clinical events according to their different temporal evolution characteristics. The experiment results indicate that our method achieves state-of-the-art performance compared with existing methods, which can well capture the inherent structural and temporal characteristics of EHRs while making predictions. In addition, the case study illustrates that our model can make more complete and accurate medication recommendations in actual scenarios, and at the same time, through the display of correlation graphs, we can make an intuitive interpretation of the medication recommendation decisions, which is very important in practical applications.

In fact, for personalized and accurate medication recommendation task, the data currently used is insufficient. There is still great research potential for improvement in the mining of EHRs. In the future, we will consider combining the raw text information of EHRs and focus on how to better model the fine-grained temporal evolution in the EHRs.

REFERENCES

[1] J. Shang, T. Ma, C. Xiao, and J. Sun, "Pre-training of graph augmented transformers for medication recommendation," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp. 5953–5959.

[2] J. Shang, C. Xiao, T. Ma, H. Li, and J. Sun, "Gamenet: Graph augmented memory networks for recommending medication combination," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 1126–1133.

[3] E. Choi, M. T. Bahadori, L. Song, W. F. Stewart, and J. Sun, "GRAM: Graph-based attention model for healthcare representation learning," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2017, pp. 787–795.

[4] E. Choi, M. T. Bahadori, and A. Schuetz, "Doctor AI: Predicting clinical events via recurrent neural networks," in *Proc. Mach. Learn. Healthcare Conf.*, 2016, pp. 301–318.

[5] E. Choi, Z. Xu, Y. Li, M. W. Dusenberry, G. Flores, Y. Xue, and A. M. Dai, "Learning the graphical structure of electronic health records with graph convolutional transformer," 2019, *arXiv:1906.04716*. [Online]. Available: http://arxiv.org/abs/1906.04716

[6] T. Kajdanowicz, K. Tagowski, M. Falkiewicz, P. Bielak, P. Kazienko, and N. V. Chawla, "Incremental embedding for temporal networks," 2019, *arXiv:1904.03423*. [Online]. Available: http://arxiv.org/abs/1904.03423

[7] J. Shang, S. Hong, Y. Zhou, M. Wu, H. Li, V. N. Networks. "Knowledge guided multi-instance multi-label learning via neural networks," in *Proc. ACML*, pp. 831–846, 2018.

[8] A. Hosseini, T. Chen, W. Wu, Y. Sun, and M. Sarrafzadeh, "HeteroMed: Heterogeneous information network for medical diagnosis," in *Proc. 27th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2018, pp. 763–772.

[9] C. Mao, L. Yao, and Y. Luo, "MedGCN: Graph convolutional networks for multiple medical tasks," 2019, *arXiv:1904.00326*. [Online]. Available: http://arxiv.org/abs/1904.00326

[10] 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 *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 3504–3512.

[11] Y. Zhang, R. Chen, J. Tang, W. F. Stewart, and J. Sun, "LEAP: Learning to prescribe effective and safe treatment combinations for multimorbidity," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2017, pp. 1315–1324.

[12] J. Feng and Z. H. Zhou, "Deep MIML network," in *Proc. AAAI Conf. Artif. Intell.*, vol. 31, 2017, pp. 1884–1890.

[13] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016, *arXiv:1609.02907*. [Online]. Available: http://arxiv.org/abs/1609.02907

[14] P. V. ković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," 2017, *arXiv:1710.10903*. [Online]. Available: http://arxiv.org/abs/1710.10903

[15] C. Esteban, V. Tresp, Y. Yang, and S. Baier, ''Predicting the co-evolution of event and knowledge graphs,'' in *Proc. FUSION*, vol. 32, Jul. 2016, pp. 98–105.

[16] B. Yu, H. Yin, and Z. Zhu, ''Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting,'' in *Proc. 27th Int. Joint Conf. Artif. Intell.*, Jul. 2018, pp. 3634–3640.

[17] S. Deng, H. Rangwala, and Y. Ning, ''Learning dynamic context graphs for predicting social events,'' in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2019, pp. 1007–1016.

[18] S. Yan, Y. Xiong, and D. Lin, ''Spatial temporal graph convolutional networks for skeleton-based action recognition,'' in *Proc. AAAI Conf. Artif. Intell.*, vol. 32, 2018, pp. 7444–7452.

[19] J. Chen, X. Xu, Y. Wu, and H. Zheng, ''GC-LSTM: Graph convolution embedded LSTM for dynamic link prediction,'' 2018, *arXiv:1812.04206*. [Online]. Available: http://arxiv.org/abs/1812.04206

[20] R. Trivedi, M. Farajtabar, P. Biswal, and H. Zha, ''Representation learning over dynamic graphs,'' 2018, *arXiv:1803.04051*. [Online]. Available: http://arxiv.org/abs/1803.04051

[21] U. Singer, I. Guy, and K. Radinsky, ''Node embedding over temporal graphs,'' in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp. 4605–4612.

[22] Z. Zhang, M. Li, X. Lin, Y. Wang, and F. He, ''Multistep speed prediction on traffic networks: A deep learning approach considering spatio-temporal dependencies,'' *Transp. Res. Part C, Emerg. Technol.*, vol. 105, pp. 297–322, Aug. 2019.

[23] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, ''The graph neural network model,'' *IEEE Trans. Neural Netw.*, vol. 20, no. 1, pp. 61–80, Jan. 2009.

[24] F. Ma, Q. You, H. Xiao, R. Chitta, J. Zhou, and J. Gao, ''KAME: Knowledge-based attention model for diagnosis prediction in healthcare,'' in *Proc. 27th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2018, pp. 743–752.

[25] E. Choi, C. Xiao, and W. Stewart, ''Mime: Multilevel medical embedding of electronic health records for predictive healthcare,'' in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 4547–4557.

[26] O. Luaces, J. Diez, and J. Barranquero, ''Binary relevance efficacy for multilabel classification,'' in *Proc. Prog. Artif. Intell.*, 2012, pp. 303–313.

[27] A. Vaswani, N. Shazeer, and N. Parmar, ''Attention is all you need,'' in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.

[28] P. Goyal, N. Kamra, X. He, and Y. Liu, ''DynGEM: Deep embedding method for dynamic graphs,'' 2018, *arXiv:1805.11273*. [Online]. Available: http://arxiv.org/abs/1805.11273

[29] Y. Zuo, G. Liu, H. Lin, J. Guo, X. Hu, and J. Wu, ''Embedding temporal network via neighborhood formation,'' in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2018, pp. 2857–2866.

[30] G. H. Nguyen, J. B. Lee, R. A. Rossi, N. K. Ahmed, E. Koh, and S. Kim, ''Continuous-time dynamic network embeddings,'' in *Proc. Companion Web Conf. Web Conf. (WWW)*, 2018, pp. 969–976.

[31] R. Trivedi, H. Dai, Y. Wang, and L. Song, ''Know-evolve: Deep temporal reasoning for dynamic knowledge graphs,'' in *Proc. 34th Int. Conf. Mach. Learn. (JMLR)*, vol. 70, 2017, pp. 3462–3471.

[32] R. Trivedi, M. Farajtabar, and P. Biswal, ''Dyrep: Learning representations over dynamic graphs,'' in *Proc. ICLR*, 2019, pp. 1–25.

[33] A. Pareja, G. Domeniconi, J. Chen, T. Ma, T. Suzumura, H. Kanezashi, T. Kaler, T. B. Schardl, and C. E. Leiserson, ''EvolveGCN: Evolving graph convolutional networks for dynamic graphs,'' 2019, *arXiv:1902.10191*. [Online]. Available: http://arxiv.org/abs/1902.10191

[34] A. Taheri, K. Gimpel, and T. Berger-Wolf, ''Learning to represent the evolution of dynamic graphs with recurrent models,'' in *Proc. Companion Proc. World Wide Web Conf.*, May 2019, pp. 301–307.

[35] P. Goyal, S. R. Chhetri, and A. Canedo, ''Dyngraph2vec: Capturing network dynamics using dynamic graph representation learning,'' *Knowl.-Based Syst.*, vol. 187, Jan. 2020, Art. no. 104816.

[36] A. E. W. Johnson, T. J. Pollard, and L. Shen, ''MIMIC-III, a freely accessible critical care database,'' *Sci. Data*, vol. 3, no. 1, pp. 1–9, 2016.

[37] A. G. Hawkes, ''Hawkes processes and their applications to finance: A review,'' *Quant. Finance*, vol. 18, no. 2, pp. 193–198, 2018.

[38] Y. Seo, M. Defferrard, X. Bresson, and P. Vandergheynst, ''Structured sequence modeling with graph convolutional recurrent networks,'' in *Proc. Int. Conf. Neural Inf. Process.*, 2018, pp. 362–373.

[39] A. Narayan and P. H. O'N Roe, ''Learning graph dynamics using deep neural networks,'' *IFAC-PapersOnLine*, vol. 51, no. 2, pp. 433–438, 2018.

[40] D. Almirall, S. N. Compton, M. Gunlicks-Stoessel, N. Duan, and S. A. Murphy, ''Designing a pilot sequential multiple assignment randomized trial for developing an adaptive treatment strategy,'' *Statist. Med.*, vol. 31, no. 17, pp. 1887–1902, Jul. 2012.

[41] Z. Chen, K. Marple, E. Salazar, G. Gupta, and L. Tamil, ''A physician advisory system for chronic heart failure management based on knowledge patterns,'' *Theory Pract. Log. Program.*, vol. 16, nos. 5–6, pp. 604–618, Sep. 2016.

[42] M. Gunlicks-Stoessel, L. Mufson, A. Westervelt, D. Almirall, and S. Murphy, ''A pilot SMART for developing an adaptive treatment strategy for adolescent depression,'' *J. Clin. Child Adolescent Psychol.*, vol. 45, no. 4, pp. 480–494, Jul. 2016.

[43] D. P. Kingma and J. Ba, ''Adam: A method for stochastic optimization,'' 2014, *arXiv:1412.6980*. [Online]. Available: https://arxiv.org/abs/1412.6980

[44] J. Duchi, E. Hazan, and Y. Singer, ''Adaptive subgradient methods for online learning and stochastic optimization,'' *J. Mach. Learn. Res.*, vol. 12, no. 7, pp. 2121–2159, 2011.

[45] T. Tieleman and G. Hinton, ''Lecture 6.5-RMSPROP: Divide the gradient by a running average of its recent magnitude,'' *COURSERA, Neural Netw. Mach. Learn.*, vol. 4, no. 2, pp. 26–31, 2012.

[46] C. Park, C. Lee, H. Bahng, T. Won, K. Kim, S. Jin, S. Ko, and J. Choo, ''STGRAT: A spatio-temporal graph attention network for traffic forecasting,'' 2019, *arXiv:1911.13181*. [Online]. Available: http://arxiv.org/abs/1911.13181

CHENHAO SU received the B.E. degree in electronic and information engineering from the Beijing University of Posts and Telecommunications, Beijing, China, in 2014, where she is currently pursuing the M.S. degree with the School of Artificial Intelligence. Her research interests include deep learning, data mining, and natural language processing.

SHENG GAO received the B.E. and M.E. degrees in information engineering from the Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 2003 and 2006, respectively, and the Ph.D. degree in computer science from Universite Pierre et Marie CURIE (Paris 6), Paris, France, in 2011. He is currently an Associate Professor with the School of Artificial Intelligence. His research interests include machine learning, data mining, and social network analysis.

SI LI received the Ph.D. degree from the Beijing University of Posts and Telecommunications, in 2012. She is currently an Associate Professor with the School of Artificial Intelligence, Beijing University of Posts and Telecommunications. Her current research interests include natural language processing and machine learning.