Disease Diagnosis Based on Heterogeneous Graph Contrastive Learning

Chengyu Yang, Dan Yang, Xi Gong

Abstract—**In recent years, the use of electronic medical record data for disease diagnosis has received considerable attention. Traditional methods, however, frequently encounter challenges with complex data structures and limited labeled data. To tackle these issues, this paper presents a novel disease diagnosis method named HCMG that employs heterogeneous graph contrastive learning, tailored to address the complexities of electronic medical record data. By constructing path-guided heterogeneous graphs and employing a contrastive learning strategy, HCMG accurately diagnoses patient conditions. Initially, it captures patient relationships and shared features through meta-paths within electronic health records. Subsequently, it enhances node similarity recognition by contrasting learning strategies between anchor and feature views. Additionally, a self-learning mechanism is integrated to reassign sample weights, refining the model's ability to differentiate between negative and misjudged samples. Finally, disease probability distributions are predicted through clustering analysis. Experimental results on the MIMIC-III dataset demonstrate that HCMG maintains superior diagnostic performance, significantly surpassing existing benchmarks, even with limited labeled data. This research not only provides an effective technological route for the analysis of electronic health records but also offers new perspectives and considerations for future studies in the medical health domain.**

Index Terms—**Heterogeneous Graph; Graph neural network; Contrastive Learning; Disease Diagnosis**

I. Introduction

In the context of today's information society, the rapid progression of medical information technology has given rise to unprecedented opportunities and challenges in the medical field. Large-scale medical data accumulation and application, such as electronic medical records (EMRs) [1], have generated new perspectives and methodologies for disease diagnosis. However, with the exponential increase in the volume of medical data and the continuous expansion of medical knowledge, traditional methods for medical diagnosis no longer suffice to meet the diverse clinical needs. To enhance the accuracy and efficiency of disease diagnosis, the latest graph neural networks coupled with contrastive learning have emerged as potential solutions.

Heterogeneous graphs (HINs) [2] represent a frequently encountered complex network structure in the real world that can efficiently portray diverse relationships and entities within medical data. Within HINs, different types of entities and relationships, are symbolized by nodes and edges respectively. This provides rich semantic and structural information for the planning and optimization of disease diagnosis methods. For instance, in a medical heterogeneous graph, entities such as patients, drugs, and operations can be distinct; the association information of patients using certain drugs or undergoing specific operations as recorded in electronic medical records is converted into differing edge types in the heterogeneous graph. Approaching disease diagnosis tasks based on heterogeneous graphs is considerably valuable in terms of enhancing diagnostic accuracy, reducing time costs for medical institutions, and improving patient treatment outcomes.

In recent years, the application of graph neural networks (GNNs) [3] for node classification in heterogeneous graphs has shown promising results due to the advancement of data mining technology. However, most existing heterogeneous graph neural networks depend on large quantities of manually labeled medical training data, which are difficult to obtain and often associated with high costs. Therefore, it becomes imperative to derive supervision from the data itself and to utilize self-supervised learning [4] with robust universal embedding representations to meet this challenge. In particular, contrastive learning [5], a major type of self-supervised learning, has garnered significant interest in recent years. Graph neural network frameworks that utilize contrastive learning hold considerable advantages when labeled information is scarce. The purpose of contrastive learning is to construct pairs of positive and negative samples for comparison, aiming to maximize the mutual information between positive samples and minimize that between negative samples, thereby effectively using unlabeled data for improved embedding learning.

In order to establish contrasting views in heterogeneous graph contrastive learning, some researchers have employed the method of meta-path [6]. A meta-path encapsulates the semantic relationships between entities in a heterogeneous graph as a sequence of entity types. For instance, in a medical heterogeneous graph, the entities representing patients and drugs are denoted by "P" and "D" respectively, thus, the meta-path "Patient-Drug-Patient" (PDP) signifies the relationships between patients who utilize the same drug. Specifically, if a path instance "p_1-d-p_2" exists, it exemplifies that two patients, "p_1" and "p_2", both use the drug "d". This

Manuscript received April 4, 2024; revised October 17, 2024. This work was supported by the General Scientific Research Project of Liaoning Provincial Department of Education (LJKMZ20220646).

Chengyu Yang is a postgraduate student at School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan, China (e-mail: ycy.pallava@qq.com).

Dan Yang is a professor at School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan, China (corresponding author to provide e-mail: asyangdan@163. com).

Xi Gong is a lecturer at School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan, China (e-mail: askdjy05gx@ 163. com).

path instance outlines how patients "p_1" and "p_2" are contextually linked via the drug "d". Therefore, using meta-paths can help identify a group of path-based neighbors that are semantically related to the given entity and provide diverse contrasting perspectives. However, most models that utilize meta-path perspectives, such as HeCo [7], merely document two entities are linked via a meta-path, thereby overlooking the contextual information about their semantic connection. This can influence the model when performing tasks like node classification prediction; for instance, the usage of the same drug can provide valuable diagnostic insights into different diseases in different patients. Thus, it becomes essential to integrate the rich contextual information of meta-paths into the contrasting view.

To address these issues, this paper proposes a disease diagnosis method called HCMG (Heterogeneous Contrastive Medical Graph) based on heterogeneous graph contrastive learning, aiming to improve the accuracy and efficiency of disease diagnosis. HCMG combines meta-path context to construct anchor views and feature views separately. The anchor view is used to record the relationship between two entities via meta-paths, generating node embeddings as anchors for each meta-path instance. The node embeddings generated from the feature view specifically describe the contextual information of how they are connected through meta-paths. Positive and negative samples are constructed based on each anchor in the anchor view. Additionally, a mechanism for learning the weights of negative samples [8] is introduced to cluster nodes, and the weights of negative samples are reallocated based on the clustering results to make full use of hard negative samples and alleviate the impact of false negative samples. The prototypical contrastive learning method is employed, where the clustering centers act as prototype vectors. By bringing nodes closer to their corresponding prototype vectors and moving them away from other prototype vectors, more compact node embeddings are learned to better distinguish between positive and negative samples, thereby enhancing the method's performance. The HCMG method learns representations of entities in a heterogeneous graph, making similar entities closer to each other in the embedding space, while dissimilar entities are more dispersed. By leveraging meta-path context and heterogeneous graph contrastive learning, the HCMG method better captures the complex semantic information between different entities and relationships in the heterogeneous graph, more effectively integrates the relevance of different types of entities, and improves the credibility and sensitivity of disease diagnosis. In summary, the main contributions of this paper are as follows:

1) We propose a disease diagnosis method HCMG based on heterogeneous graph contrastive learning. This method formulates more accurate and effective contrasting views using meta-paths and graph contrastive learning, thereby capturing complex semantic information better between different entities and relationships in the heterogeneous graph. This enhancement aids in more precise interpretation of medical data and diagnostic assistance for disease diagnosis tasks.
2) The HCMG method improves the ability to distinguish between positive and negative samples by clustering nodes and relearning the allocation of weights for negative samples. Prototypical contrastive learning is introduced, which aids in learning compact embeddings for nodes belonging to the same cluster.
3) A series of experiments and comparative analysis are conducted to appraise the HCMG method. The performance and effectiveness of the method in disease diagnosis tasks are assessed by training and testing it on the publicly available medical dataset, MIMIC-III. Comparisons with state-of-the-art baseline methods affirm the superiority and practicality of the HCMG method in disease diagnosis.

II. Related Work

A. Heterogeneous Graph Neural Networks

In recent years, research on Heterogeneous Graph Neural Networks (HGNN) has garnered widespread attention, and various models have been proposed to address the challenges of heterogeneous information networks. Initially, HetGNN [9] utilizes a bidirectional LSTM and an attention mechanism to aggregate information from similar neighbors. Subsequently, HGT [10] designs an attention architecture akin to the Transformer to better capture relationships between different types of neighbors, thereby improving the efficiency of information extraction in heterogeneous graphs. HAN [11] introduces node-level and semantic-level attention mechanisms to hierarchically learn the importance of neighbors in meta-paths and the weights of different meta-paths. To further enhance the performance of node representation, researchers have proposed MAGNN [12], which integrates information from intermediate semantic nodes through meta-path instance encoders. Additionally, GTNs [13] can generate new graph structures, revealing useful connections between non-connected nodes in the original graph, and enhancing the effectiveness of node representation learning. Although these models excel in exploring heterogeneous graph data, they are unable to perform self-supervised learning and rely heavily on labeled data to a large extent.

B. Graph Contrastive Learning

In the field of self-supervised learning, contrastive learning plays a crucial role. The core idea involves constructing positive and negative sample pairs to bring similar samples closer and push dissimilar samples apart. Recent research explores the integration of contrastive learning to address the limitations of supervised learning, which heavily depends on labels, particularly in the realm of graph data [14]. GCA [15] employs data augmentation techniques to construct distinct contrastive views, encompassing diverse mechanisms including graph-graph contrast, node-node contrast, and graph-node contrast. For example, GRACE [16] generates two enhanced graph views by masking node features and removing edges to bring representations of the same nodes closer while pushing apart other nodes. Inspired by SimCLR [17] in the visual domain, GraphCL [18] applies this idea to graph-structured data, generating two distorted graphs through node removal and edge perturbation, then maximizing the mutual information between the two graph layers for learning. For heterogeneous graphs, HeCo [7] constructs two views using network

patterns and meta-path information to generate node representations, improving representation learning through contrastive learning between nodes. HDGI [19] extends DGI to heterogeneous information networks, learning advanced node representations by maximizing the mutual information between local and global representations, enabling the model to exchange information between different types of nodes. DGI[20] compares local information and global information through the Infomax method to promote graph representation learning. GMI [21] compares interaction information obtained from node features and topological structures, expanding the application scope of contrastive learning. MVGRL[14] compares embeddings obtained from first-order and second-order neighbors to facilitate information flow between different graph layers. Using graphs as an example, GCC [22] learns to differentiate between different instances, enhancing the model's generalization ability. GCA [15] randomly deletes unimportant edges, adds noise to node features to disrupt attributes, generates new views for contrastive learning, and strengthens the model's robustness and generalization performance. These studies have made positive progress in graph contrastive learning and have provided important insights and methodological support for the development of disease diagnosis methods [23] based on heterogeneous graphs.

C. Disease Diagnosis

Several studies propose disease diagnosis models based on graph neural networks [24], which integrate medical knowledge bases and Electronic Medical Records (EMR) data. These models construct medical concept graphs and patient record graphs, using graph encoders to learn embeddings of patient nodes and disease nodes for disease diagnosis tasks. Additionally, a multimodal learning framework [25] is introduced for disease diagnosis, capturing the correlations and complementarity between different modalities through attention mechanisms. Graph neural networks demonstrate their advantages in routine disease diagnosis tasks. These models can quickly learn knowledge from historical medical data, handle new medical data, and improve diagnostic accuracy. Compared to traditional disease diagnosis methods, the diagnostic results of graph neural network models are more objective and accurate, thereby reducing the workload of physicians and enhancing efficiency.

III. Preliminaries

Definition 1. Heterogeneous Graph. An electronic medical record data is defined as a graph $\mathcal{G}=(\mathcal{V},\mathcal{E},\mathcal{N},\mathcal{L})$, where $\mathcal{V}$ represents the set of nodes and $\mathcal{E}$ represents the set of edges. It can be said that $\mathcal{E}$ is the set of relationships between two nodes in $\mathcal{V}$. Simultaneously, the set of node types in $\mathcal{V}$ is denoted as $\mathcal{N}$, which includes various types of nodes such as patients, drugs, operations, etc., and the set of edge types in $\mathcal{E}$ is denoted as $\mathcal{L}$, which includes different types of edges such as patient-drug, patient-operation, etc. Specifically, the following relationships exist: 1) the relationship between $\mathcal{V}$ and $\mathcal{N}$: involves a mapping $\phi:\mathcal{V}\to\mathcal{N}$; 2) the relationship between $\mathcal{E}$ and $\mathcal{L}$: involves a mapping $\psi:\mathcal{E}\to\mathcal{L}$; 3) Definition of Heterogeneous Graph: if $|\mathcal{N}|+|\mathcal{L}|>2$ holds, then the graph $\mathcal{G}$ is referred to as a heterogeneous graph; otherwise, the graph $\mathcal{G}$ is a homogeneous graph.

Definition 2. Meta Path. An instance of a meta path is defined as $\mathcal{M}:\mathcal{N}_1\xrightarrow{\mathcal{L}_1}\mathcal{N}_2\xrightarrow{\mathcal{L}_2}\ldots\xrightarrow{\mathcal{L}_i}\mathcal{N}_{i+1}$, denoted as $\mathcal{N}_1\mathcal{N}_2\ldots\mathcal{N}_{i+1}$. Here, $\mathcal{N}$ represents nodes, and $\mathcal{L}$ is a relationship connecting two nodes. When there is an edge connecting nodes $\mathcal{N}_x$ and $\mathcal{N}_y$ in the heterogeneous graph $\mathcal{G}$, it signifies the existence of a path through the relationship $\mathcal{L}$. Specifically, this paper encompasses three types of nodes in $\mathcal{N}$, which are patients, drugs, and operations. There are two types of relations in $\mathcal{L}$, namely "patient-drug" and "patient-operation". Two meta-paths, "Patient-Drug-Patient (PDP)" and "Patient-Operation-Patient (POP)", are used in this paper to extract latent associative information from electronic medical records.

Definition 3. Heterogeneous Graph Contrastive Learning. Given a heterogeneous graph $\mathcal{G}$, an anchor view and a feature view are established through meta-paths, and positive and negative samples are constructed for contrastive learning to generate node embeddings. By clustering the generated node embeddings, it is possible to diagnose whether new patients have a certain disease based on the probability distribution of the node embeddings of new patients.

IV. The Proposed Framework

This section provides a detailed introduction to the disease diagnosis method HCMG based on heterogeneous graph

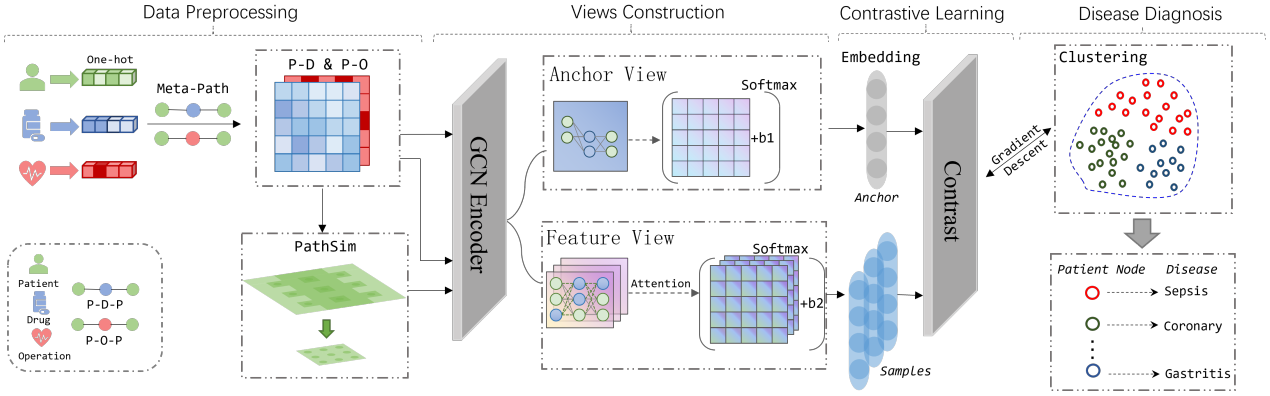


Fig. 1. Overall Framework of the HCMG

contrastive learning, as proposed in this paper and illustrated in Figure 1. Initially, feature vectors of nodes of different types are mapped to the same dimension for preprocessing, and adjacency matrices are established based on the neighbors connected by each meta-path. Subsequently, a graph convolutional network encoder is used to aggregate all meta-path adjacency matrices to construct an anchor view and retain the contextual semantic information of meta-paths to build a feature view. The embeddings from different meta-paths in the feature view are fused using an attention mechanism. The node embeddings generated from the anchor view serve as anchors, while the aggregated node embeddings in the feature view are categorized as positive and negative samples based on the anchors. To mitigate the influence of false negative and hard negative samples, a clustering algorithm is applied for clustering and redistributing the weights of negative samples. Finally, prototype contrastive learning is introduced to further enhance model performance by jointly calculating the contrastive losses and prototype losses based on node embeddings and clustering results under the anchor view and feature view. The subsequent subsections will describe each component in detail.

A. Data Preprocessing

1) Node feature transformation

In the data preprocessing stage, considering the existence of different types of nodes in the heterogeneous graph, each type of node necessitates the transformation of raw features. Initially, the fundamental feature information of nodes is extracted, and continuous features are standardized by converting the raw feature data into a sparse matrix representation. Additionally, a relationship graph is constructed based on the association information between nodes, leading to the generation of the adjacency matrix of nodes using edge information from the data. Moreover, for potential multi-class labels, One-Hot encoding is applied to convert labels into vector form, thereby establishing a foundational data structure for subsequent classification tasks. Specifically, the node feature matrix $X \in R^{N \times D}$ is first created, where N signifies the number of nodes, and D represents the feature dimension of each node. During the node feature transformation process, for each object x_i of type T, a type-specific mapping matrix $W_T \in R^{D \times D'}$ is employed, where D' represents the transformed feature dimension. The transformation process can be expressed as:

$$x_i' = W_T x_i \tag{1}$$

where x_i' denotes the embedding of the node object x_i in the transformed feature space. This transformation process ensures that the features of each node type are mapped to the same space, providing consistent feature embedding for subsequent data analysis and modeling. The technique for constructing an adjacency matrix typically involves representing the associative data information to capture the relationships between nodes. To construct the adjacency matrix of patient graphs based on the associative information between patient nodes, the following formula is utilized:

$$A_{ij} = \begin{cases} 1, & P_i - D_j \text{ or } P_i - O_j \\ 0, & otherwise \end{cases} \tag{2}$$

If a patient P_i has used the drug D_j or undergone the operation O_j, the position A_{ij} in the generated adjacency matrix is assigned a value of 1; otherwise, it is set to 0.

For datasets containing multiple category labels, with a label featuring k categories, where each category is represented by an integer ranging from 0 to $k-1$, One-Hot encoding transforms each category into a vector of length k, with only the position corresponding to the category set to 1, while other positions are assigned a value of 0. The One-Hot encoding for the *i*-th category label is denoted as y_i:

$$One - Hot(y_i) = [0, 0, \ldots, 1, \ldots, 0] \tag{3}$$

In this scenario, the *i*-th element in y_i is 1, and the remaining elements are 0. Utilizing One-Hot encoding facilitates the transformation of original category labels into a vector format that can be directly processed by models like neural networks, enhancing the model's ability to comprehend and analyze categorical information.

2) Neighborhood Node Aggregation

When aggregating neighbor nodes in heterogeneous medical graphs, leveraging meta-paths can more precisely capture the relationships and common features among patients, thereby enhancing information propagation and feature learning efficiency in graph neural networks. However, the complex structure of medical heterogeneous graphs and intricate data relationships may lead to challenges when aggregating neighbor nodes based on meta-paths, such as aggregating an excessive number of neighbors simultaneously or aggregating neighbors with significant gaps. Aggregating a large number of neighbors at once or neighbors with substantial gaps may blend vast amounts of information, including irrelevant details, diminishing the model's effectiveness. This approach could also extend the information propagation path, heighten information loss, and impact the model's learning capability. To address these issues, this paper introduces the PathSim [26] algorithm. By computing the similarity between paths in the adjacency matrix, the algorithm enables the exploration of potential relationships between neighbor nodes, selection of crucial neighbor nodes for aggregation, and regulation of information propagation distances between nodes to prevent information bottleneck occurrences. Specifically, given a symmetric meta-path P, PathSim assesses the path similarity between two identically typed vertex objects, x and y, according to the formula:

$$s(x,y) = \frac{2 \times \left| p_{x \rightsquigarrow y} : p_{x \rightsquigarrow y} \in P \right|}{\left| \left\{ p_{x \rightsquigarrow x} : p_{x \rightsquigarrow x} \in P \right\} \right| + \left| \left\{ p_{y \rightsquigarrow y} : p_{y \rightsquigarrow y} \in P \right\} \right|} \tag{4}$$

Where $p_{x \rightsquigarrow y}$ represents the path instance connecting x and y, and $p_{x \rightsquigarrow x}$ represents the path instance connecting x with itself, PathSim is utilized to compute path similarity. Subsequently, the top K most relevant neighbor nodes are selected for aggregation. The aggregation process is structured as follows:

$$h_{agg} = softmax(\sum_{i=1}^{n} x_i \odot x_i) \odot self_x \tag{5}$$

Here, n signifies the number of neighbor nodes, and each x_i denotes a feature vector of a neighbor node. The feature

vector of the input node is depicted by $self_x$. The feature vectors of the neighbor nodes undergo element-wise multiplication and summation. This result is normalized using a softmax function, and then element-wise multiplied with the feature vector of the input node to produce the final aggregation embedding.

B. Contrasting Views Construct

1) Anchor View

The purpose of this section is to construct an anchor view based on meta-paths. The anchor view is a simplified representation of the heterogeneous medical graph dataset, describing which nodes are connected by meta-paths, providing reference points for further analysis. After aggregating neighboring nodes according to Equation (5) to obtain the final embedding result, denoted as $\tilde{h}_v$, it is used as the original input, and a two-layer graph convolutional network encoder is applied for computation:

$$h_v^{l+1} = \sigma(\hat{A} h_v^l W_l), \quad l = 0,1 \tag{6}$$

Where h_v^l represents the embedding representation of the l-th layer, W_l is the weight matrix, $\hat{A}$ is the result of normalizing the heterogeneous graph adjacency matrix A, and σ indicates the activation function. In particular, $h_v^0 = \tilde{h}_v$. The encoder obtains the processed anchor embedding representation $x_i^a = h_v^2$, which is used to construct the anchor view that records meta-path connections.

2) Feature View

Unlike the simplified construction of the anchor view, the feature view contains rich contextual information from meta-paths and requires distinguishing between positive and negative samples based on the reference provided by the anchor view. The meta-path context consists of path instances capturing detailed information on how two objects are connected. By selecting the middle node x_i of the meta-path P and obtaining the feature vector set $\{h_j\}_{j \in N(i)}$ of its neighboring nodes, where $N(i)$ is the set of neighbor nodes of node x_i and j represents the j-th jump neighbor of x_i. Subsequently, the following formula is used to aggregate and generate the initial feature embedding for x_i:

$$h_i^P = \sigma(h_i + \sum_{j=1}^{l} \sum_{x_v \in N_i^P} W_{pj} h_v) \tag{7}$$

In which, l is the total length of meta-paths, and W_{pj} is the learnable parameter matrix. After obtaining the initial feature embedding h_i^P that aggregates meta-path context information, the initial feature embedding and normalized adjacency matrix $\hat{A}$ is calculated using the same dual-layer graph convolutional network encoder (5) as in the anchor view, resulting in the feature embedding quantity $x_{feature}$ containing meta-path context information. Aggregation of long meta-path instances may lead to excessive context information and the inclusion of a large amount of irrelevant information that impacts the model's effectiveness. Therefore, nodes and edges are first subjected to feature and edge masking, constructing a meta-path-induced graph. Specifically, for the given adjacency matrix $A_i \in R^{n \times n}$ where n represents the number of nodes, a mask matrix M_i is used to mask edges, where $M_i \in \{0,1\}^{n \times n}$ indicates the edges to be masked, resulting in the masked adjacency matrix $A_i^{mask} = A_i \odot M_i$. Similarly, for the given feature matrix $X \in R^{n \times d}$ where d represents the feature dimension, let M_f be the feature mask vector, $M_f \in \{0,1\}^d$ indicates the feature dimensions to be masked, and the feature matrix after masking the node features is represented as $X_{mask} = X \odot M_f$. After introducing noise to the nodes and edges, x_i is selected again to aggregate with its feature-masked neighbors according to formula (7), resulting in the second feature embedding quantity x_{mask}. At this point, for each node x_i in the meta-path P, two feature embedding quantities $x_{embed} = \{x_{feature}, x_{mask} \mid x \in P\}$ are generated. Subsequently, an attention mechanism [28] is used to dynamically aggregate the neighboring node features based on the weighted importance of node features, and the two are fused to generate the final feature embedding quantity. The calculation formula for attention weights is as follows:

$$e_{ij} = LeakyReLU(W^T [h_i \| h_j]) \tag{8}$$

$$a_{ij} = \frac{exp(e_{ij})}{\sum_{k \in N(i)} exp(e_{ik})} \tag{9}$$

W is the learned weight matrix, $\|$ denotes vector concatenation operation, $N(i)$ represents the set of neighboring nodes of node x_i, and by substituting $x_{feature}$ and x_{mask} into h_i and h_j, attention weights are calculated to obtain the final feature embedding quantity using the following formula:

$$x_i^f = \sigma\left(\sum_{j \in N(i)} a_{ij} \cdot W h_j\right) \tag{10}$$

C. Disease Diagnosis

After the aforementioned calculations, for each node x_i, anchor embeddings x_i^a and feature embeddings x_i^f are aggregated with neighboring node context information to generate comprehensive embeddings. The anchor embeddings x_i^a are used as contrastive anchors, while the feature embeddings x_i^f are used as positive samples. All other nodes in the feature view are considered negative samples. Due to the large number of negative samples, to allow the model to focus more on important samples, positive samples are taken as clustering centers, negative samples in the feature view are clustered multiple times, and weights α are introduced to differentiate the importance of different samples. Specifically, this study uses the gradient descent algorithm to update weight parameters, and the gradient update rule for weight parameters is as follows:

$$\alpha_i^{t+1} = \alpha_i^t - \eta \frac{\partial loss}{\partial w_i} \tag{11}$$

Where η represents the learning rate, t denotes the number of iterations, and $\frac{\partial loss}{\partial w_i}$ is the gradient of the loss function with respect to the weight parameters. By learning the weight parameters from the model, the model can autonomously redistribute sample weights to compactly center misclassified negative samples and push away difficult negative samples to reduce their influence.

After weighting the samples, the softmax function is used to calculate the probability distribution of the nodes:

$$\hat{p} = \frac{\exp\left(s_{pos}\right)^{\alpha}}{\exp\left(s_{pos}\right)^{\alpha} + \sum_{i=1}^{K} \exp\left(s_{neg_i}\right)^{\alpha}} \tag{12}$$

Where s_{pos} and s_{neg} represent the similarity scores of the positive and negative samples, respectively, and K denotes the number of negative samples. At this point, the model's loss is calculated using the InfoNCE loss function as follows:

$$\mathcal{L}_i^{InfoNCE} = log \frac{exp\left(x_i^f \cdot x_i^a / \tau\right)}{\sum_{i=1}^{K} \alpha_i exp\left(x_i^f \cdot x_i^a / \tau\right)} \tag{13}$$

Where τ is the temperature parameter, used to control the "softening" among the similarity scores.

To further improve the model's effectiveness and make the distribution of feature embeddings within the same cluster more compact, the prototype contrastive learning loss function [29] is introduced. Specifically, for the feature embedding x_i and the clustering result $C_{result} = \left(c_1, c_2, ..., c_N\right)$, where c_i denotes the cluster number to which the i-th sample belongs, first, the prototype vectors μ_k of each cluster are calculated as follows:

$$\mu_k = \frac{1}{\left|C_k\right|} \sum_{i \in C_k} x_i \tag{14}$$

Next, the similarity $\log its_{ik} = x_i \cdot \mu_k / \tau$, between the i-th sample x_i and the prototype of the k-th clustering cluster μ_k is computed. The formula for calculating the prototype contrastive learning loss function is as follows:

$$\mathcal{L}_i^{ProtoNCE} = -log\left(\frac{exp\left(logits_{ik}\right)}{\sum_{j \neq k} exp\left(logits_{ij}\right) + 1}\right) \tag{15}$$

By combining the InfoNCE and prototype contrastive learning loss functions, the overall loss function of the model is obtained as:

$$\mathcal{L} = \mathcal{L}_{InfoNCE} + \lambda \mathcal{L}_{ProtoNCE} \tag{16}$$

Where λ is a hyperparameter used to adjust the model's emphasis on the prototype loss during training.

For the new patient node, the HCMG method integrates it into the existing graph structure, assigning relevant features based on patient data attributes and context information. HCMG establishes connections between the new node and neighboring nodes to capture contextual relationships. Anchor embeddings x_{new}^a are initialized for the newly added patient node, and feature embeddings x_{new}^f are generated based on patient attributes and neighboring context. The feature embeddings of the new node are treated as positive samples, while other nodes are considered negative samples. The softmax function is used to calculate the probability distribution of the node, considering the similarity scores of positive and negative samples, and the disease label with the highest similarity score is predicted as the disease the patient is likely to have:

$$\hat{y} = softmax\left(x_{new}\right) \tag{17}$$

Through the above steps, HCMG effectively integrates the new patient node, generates node embeddings, and performs accurate disease diagnosis.

V. Experiments and Evaluation

This section introduces the dataset, evaluation metrics, and parameters used in the experiments. It conducts extensive comparative experiments with various baseline methods, carries out ablation experiments to verify the model's effectiveness, and analyzes the experimental results accordingly.

A. Dataset

MIMIC-III (The Medical Information Mark for Intensive Care III), a large, freely available public dataset of de-identified intensive care medical records. MIMIC-III comprises de-identified data of patients in the intensive care units at Beth Israel Deaconess Medical Center from 2001 to 2012, and includes vital signs, medications, patient observations recorded by nursing staff, operation codes, diagnosis codes, length of hospital stay, survival data, and more. For this study, five representative diseases from the MIMIC-III dataset are selected as experimental data: Sepsis, Coronary, Gastritis, Heart Failure, and Respiratory Failure. The experiment focuses on 1396 drugs used by patients with the aforementioned diseases, 570 surgical operations, and specific medical record texts. Two meta-paths, "Patient-Drug-Patient (PDP)" and "Patient-Operation-Patient (POP)," are established to develop a model for predicting the probability distribution of patients possibly afflicted with the five representative diseases mentioned previously. Details of patient data statistics are presented in Table 1:

TABLE I
STATISTICS OF THE DATASET

Disease	Number of Patients
Sepsis	1937
Coronary	2921
Gastritis	415
Heart Failure	853
Respiratory Failure	1287
Total	7413

B. Evaluation Metrics

This study utilizes Micro-F1, Macro-F1, and Area Under the Curve (AUC) scores as evaluation metrics for disease diagnosis results.

1) Micro-F1

Micro-F1 evaluates the model's performance by aggregating the predictions of each class into a single large category and then calculating the precision and recall for this collective category. Here is the calculation:

$$Micro-F1=\frac{2\times TP}{2\times TP+FP+FN} \tag{18}$$

Where *TP* represents true positive samples, *FP* represents false positive samples, and *FN* represents false negative samples.

2) Macro-F1

Macro-F1 assesses the model's performance by computing the F1 score for each class individually and then taking their average. The calculation formula is presented below:

$$Macro-F1=\frac{1}{n}\sum_{i=1}^{n}\frac{2TP_i}{2TP_i+FP_i+FN_i} \tag{19}$$

3) AUC

AUC is a widely used metric for evaluating the performance of binary classification models. It measures the likelihood that the classifier will rank positive samples higher than negative samples across different thresholds. AUC is determined by calculating the area under the ROC curve.

C. Baselines

To evaluate the effectiveness of our method, we used the following eight advanced methods as baseline methods for comparison with the proposed HCMG method:

1) Homogeneous Graph-Based Methods

DGI [20] utilizes the Infomax method to compare local and global information, thereby promoting graph representation learning.

GraphSAGE [29] generates node embedding representations by aggregating information from a fixed number of neighboring nodes.

2) Semi-Supervised Methods Based on Heterogeneous Graphs

HAN [11] introduces node-level and semantic-level attention mechanisms to hierarchically learn the importance of neighbors under meta-paths and the weights of different meta-paths.

GCN [2] generates a classic graph convolutional neural network, commonly used as a model for studying graph embeddings.

GAT [27] introduces a self-attention mechanism to aggregate features by calculating the attention coefficients of nodes, serving as a classic graph attention mechanism model.

3) Unsupervised Methods Based on Heterogeneous Graphs

HetGNN [9] utilizes a bidirectional LSTM and attention mechanism to aggregate information from similar neighbors.

Mp2vec [30] constructs node embedding representations using random walks based on meta-paths.

Heco [7] utilizes network patterns and meta-path information to construct two views for generating node representations, and enhances representation learning by utilizing contrastive learning between nodes.

D. Parameter Settings

In this study, 40, 60, and 80 randomly selected labeled samples for each disease were used as the training set from the experimental data. Additionally, the remaining data was evenly split into validation and test sets. The model was constructed using the PyTorch framework and optimized using the Adam optimizer with a total learning rate of 0.0005. To prevent overfitting, a dropout ratio of 0.5 was set. For the model details, the dimension of the embedding layer was set to 128, the ratio of feature and edge perturbations was 0.2, each node aggregated context information from up to 100 neighboring nodes, and a maximum of 1200 clusters were allowed for each clustering. The temperature parameter τ for computing InfoNCE was set to 0.4, and the weight for fine-tuning prototype contrastive learning loss was set to 0.1. During model training, if the metric did not improve for 30 consecutive rounds, the training was stopped.

E. Experimental Results and Analysis

Through multiple experiments, the final experimental results of the HCMG method and various baseline methods are shown in Table 2. The experimental results indicate that the proposed HCMG method performs significantly better than the vast majority of baseline methods, especially when the number of labeled nodes provided is small. The HCMG method outperforms all baseline methods as the number of labeled nodes decreases, with its performance approaching that of the state-of-the-art Heco. Additionally, in terms of the AUC metric, the performance of HCMG far exceeds all other baseline methods. Although the experimental performance of DGI is relatively close compared to other heterogeneous graph methods, finding completely isomorphic medical datasets in reality is challenging. In contrast to HAN, GCN, and other semi-supervised heterogeneous graph models, it can be observed that unsupervised algorithms consistently achieve outstanding results when labeled data is scarce. Comparatively, the disease diagnosis method proposed in this paper based on heterogeneous graph contrastive learning shows the most advanced predictive effects compared to HetGNN and Mp2vec, among other unsupervised heterogeneous graph methods. Although there may be instances where it falls behind the state-of-the-art Heco method in terms of Micro-F1 and Macro-F1, it significantly surpasses Heco in the AUC metric. This difference could be attributed to the presence of data type imbalances in medical datasets. For example, in the MIMIC-III dataset utilized in this study, the number of patients with coronary heart disease is more than seven times the number of patients with gastritis. Therefore, the AUC score can better reflect the model's predictive performance on imbalanced datasets.

The experimental results indicate that disease diagnosis methods based on heterogeneous graph contrastive learning are highly effective, exhibiting superior performance, particularly in scenarios where manual medical labeling data is lacking.

F. Clustering Result

In this section, we employ the k-means algorithm to cluster node embeddings derived from the experimental results, thereby visually validating the model's performance. The final clustering results are scored using the Adjusted Rand Index (ARI) and the Normalized Mutual Information (NMI), obtaining scores of 72.16% and 69.58%, respectively. The

TABLE II
EXPERIMENT RESULTS OF HCMG AND BASELINES

Metric	#	DGI	SAGE	HAN	GCN	GAT	HetGNN	Mp2vec	Heco	HCMG
Micro-F1	40	62.4±3.9	49.7±3.1	70.7±2.1	69.6±2.2	70.8±1.9	61.5±2.5	60.8±0.4	78.8±1.3	81.70±0.14
	60	63.9±2.9	52.1±2.2	71.3±2.3	74.0±2.1	73.2±2.2	68.5±2.2	69.7±0.6	80.5±0.7	80.16±0.14
	80	63.1±3.0	51.4±2.2	74.4±2.1	76.0±2.7	76.5±2.1	65.6±2.2	63.9±0.5	82.5±1.4	80.41±0.23
Macro-F1	40	51.6±3.2	42.6±2.5	47.0±3.1	52.0±1.4	53.3±3.0	50.1±0.9	54.8±0.5	71.4±1.1	77.93±0.18
	60	54.7±2.6	45.8±1.5	53.4±3.1	56.6±2.1	58.3±2.2	59.0±0.9	64.8±0.5	73.8±0.5	74.17±0.16
	80	55.4±2.4	44.9±2.0	54.4±2.2	59.7±3.1	61.2±2.2	57.3±1.4	60.7±0.3	75.8±1.8	75.43±0.19
AUC	40	75.9±2.2	70.9±2.5	78.9±2.3	75.4±2.0	77.2±2.5	78.0±1.4	81.2±0.2	90.8±0.6	95.13±0.02
	60	77.9±2.1	74.4±1.3	80.7±2.1	76.9±1.7	79.3±2.3	83.1±1.6	88.8±0.2	92.1±0.6	93.98±0.02
	80	77.2±1.4	74.2±1.3	80.4±1.5	78.2±1.9	80.1±1.7	84.8±0.9	85.6±0.2	92.4±0.7	94.17±0.08

clustering results are visualized in Figure 2. The introduction of prototype contrastive learning in HCMG encourages the node embeddings to be more compact within the same cluster, thereby helping to improve node clustering. From the above results, it can be concluded that the node embeddings generated by the HCMG method can effectively distinguish patient node embeddings with different diseases.

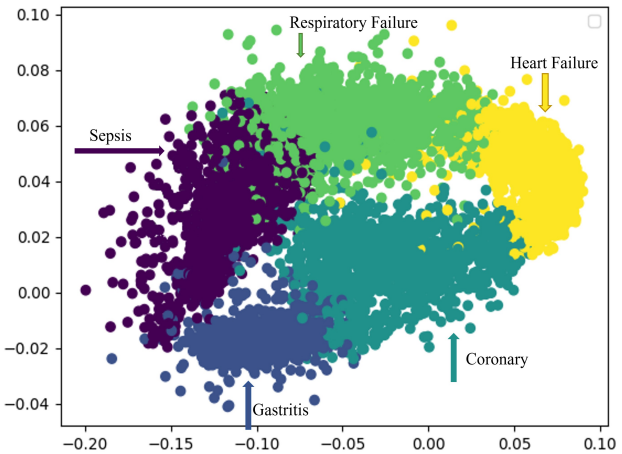
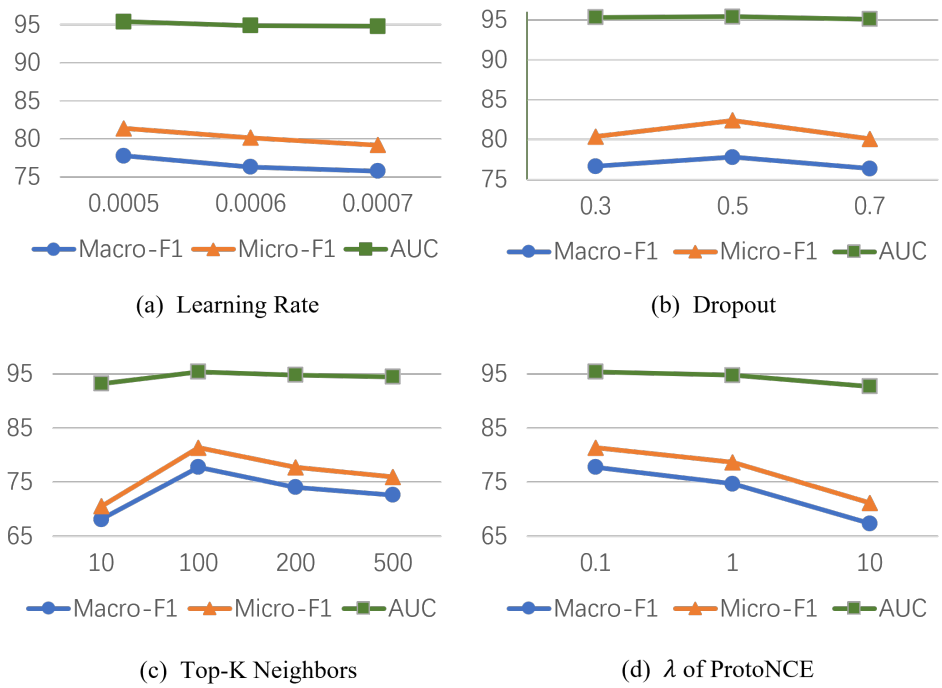


Fig. 2. Result of Patient Node Clustering

G. Hyperparameters Study

This section discusses the performance differences of HCMG under various hyperparameter settings. The main focus is on the impact of changes in the model's learning rate, Dropout, the number of filtered aggregated neighboring nodes, and the ratio λ of the prototype contrastive learning loss function. In the experiment, we maintain all other parameters unchanged, altering only the parameter under investigation for each test. This study helps to investigate the influence of the specific parameter on the model and to explore the optimal hyperparameter settings for improved method performance. The experiments were conducted using a dataset comprising 40 labeled data points. Figure 3 presents the results of the hyperparameter research. Conclusions drawn from the figure are as follows:

1) In the experiment described in Figure 3(a), training was conducted with learning rates set at 0.0005, 0.0006, and 0.0007. The model performance gradually declined as the learning rate increased. This trend may be due to excessively large learning rates leading to disproportionately large parameter updates, making it difficult for the model to converge stably during training, which resulted in decreased performance.
2) The experiment described in Figure 3(b) set Dropout values at 0.3, 0.5, and 0.7 to identify the optimal training parameters that balance the model's fitting ability and generalization capacity. Experimental results indicate that smaller dropout values (such as 0.3) limit the model's capacity and increase generalization ability but may lead to underfitting. Conversely, larger dropout values (such as 0.5) can more effectively prevent overfitting, while excessively large values (such as 0.7) may cause too many neurons in the model to be randomly dropped, making it challenging for the model to effectively learn patterns in the data, thus reduceing performance.
3) The experiment described in Figure 3(c) set the number of top-K relevant neighboring nodes, filtered and



(a) Learning Rate (b) Dropout

(c) Top-K Neighbors (d) λ of ProtoNCE

Fig. 3. Results of Hyperparameters Study

aggregated by the Pathsim algorithm, at 10, 100 and 500. From the results, it is concluded that aggregating a smaller number of neighboring nodes (K=10) compared to 100 shows less effective performance; however, aggregating too many neighbors (K=500) may increase noise content and reduce model performance.

4) The experiment described in Figure 3(d) set the ratio λ of the prototype contrastive learning loss function at 0.1, 1 and 10 to study the impact of compact clustering results from prototype contrastive learning on model performance. Experimental results indicate that setting the ratio of the prototype contrastive learning loss function too high can lead to decreased model effectiveness. This might be due to overly compact clustering making it more difficult to distinguish between hard negative samples and false negative samples when they are classified into the same cluster.

Subsequently, we studied the impact of the number of labeled samples on the model's effectiveness, with experimental results organized in Table 3. The data demonstrate that, while our method can achieve good predictive outcomes with a relatively small number of labeled samples, a too low count severely limits the model's ability to acquire sufficient information, severely impacting prediction results. Therefore, maintaining a certain number of labeled samples is essential to ensure the model obtains adequate information and addresses the issue of sample scarcity.

TABLE III
RESULT OF THE NUMBER OF LABELED SAMPLES

number	Macro-F1	Micro-F1	AUC
5	50.5±0.6	51.6±0.6	81.9±0.3
10	68.8±0.3	71.4±0.2	87.8±0.1
20	70.5±0.3	75.2±0.8	91.8±0.2
40	81.7±0.1	77.9±0.2	95.3±0.1

H. Ablation Experiment

This section conducts ablation experiments on the HCMG method to verify the validity of each module. According to the main function of the HCMG method, the first variant HCMG_nm is proposed, where the aggregated neighboring nodes during the process of generating feature views come directly from the anchor's immediate neighbors, without aggregating the multi-hop neighbors connected through a meta-path. This is to verify the necessity of aggregating contextual information through a meta-path in the HCMG method. The second variant HCMG_nw no longer re-evaluates the weights assigned to negative samples, in order to assess the impact of false negative and mislabeled negative samples on the method. The third variant HCMG_np evaluates the optimization effect of using a prototype contrastive learning loss function for the method. The results of the ablation experiments are shown in Figure 4. The ablation experiments demonstrate that the various methods proposed in this paper effectively enhance the practical predictive performance of the model, with the

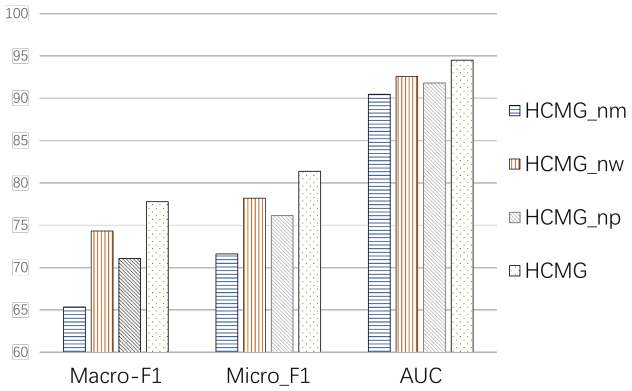


Fig. 4. Result of Ablation Experiment

aggregation of neighboring contextual information through a meta-path playing a crucial role in the method.

VI. Conclusion

The disease diagnosis method HCMG, based on heterogeneous graph contrastive learning proposed in this study, effectively applies heterogeneous graph and contrastive learning methods to the task of disease diagnosis. The experimental results indicate that the model significantly improves both accuracy and efficiency in handling medical data, particularly exhibiting outstanding performance in scenarios with limited labeled data. Future research directions may further explore the application range of the method, optimize algorithm performance, and attempt to apply this method to other tasks in the medical field to enhance the effectiveness of disease diagnosis.

References

[1] Sun M, Oliwa T, Peek M E, et al, " Negative Patient Descriptors: Documenting Racial Bias In The Electronic Health Record," Health Affairs, vol. 41, no. 2, pp203-211, 2022.

[2] Wang Z, Wen R, Chen X, et al, "Online Disease Diagnosis with Inductive Heterogeneous Graph Convolutional Networks," Proceedings of the Web Conference 2021, pp3349-3358, 2021.

[3] Zhao J, Wang X, Shi C, et al, "Heterogeneous Graph Structure Learning for Graph Neural Networks," Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, no. 5, pp4697-4705, 2021.

[4] Krishnan R, Rajpurkar P, Topol E J, "Self-supervised learning in medicine and healthcare," Nature Biomedical Engineering, vol. 6, no. 12, pp1346-1352, 2022.

[5] Baoxin Zhang, Dan Yang, Yang Liu, and Yu Zhang, "Graph Contrastive Learning with Knowledge Transfer for Recommendation," Engineering Letters, vol. 32, no. 3, pp477-487, 2024.

[6] Li X, Ding D, Kao B, et al, "Leveraging Meta-path Contexts for Classification in Heterogeneous Information Networks," 2021 IEEE 37th International Conference on Data Engineering (ICDE). IEEE, pp912-923, 2021.

[7] Viand A, Jattke P, Haller M, et al, "HECO: Automatic code optimizations for efficient fully homomorphic encryption," arXiv preprint arXiv:2202.01649, 2022.

[8] Yu J, GE Q, Li X, et al, "Heterogeneous Graph Contrastive Learning with Meta-Path Contexts and Adaptively Weighted Negative Samples," IEEE Transactions on Knowledge and Data Engineering, vol. 36, no. 10, pp5181-5193, 2024.

[9] Zhang C, Song D, Huang C, et al, "Heterogeneous Graph Neural Network," Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. pp793-803, 2019.

[10] Hu Z, Dong Y, Wang K, et al, "Heterogeneous Graph Transformer," Proceedings of The Web Conference 2020, pp2704-2710, 2020.

[11] Wang X, Ji H, Shi C, et al, "Heterogeneous Graph Attention Network," The World Wide Web Conference, pp2022-2032, 2019.

[12] Fu X, Zhang J, Meng Z, et al, "MAGNN: Metapath Aggregated Graph Neural Network for Heterogeneous Graph Embedding," Proceedings of The Web Conference 2020, pp2331-2341, 2020.

[13] Yun S, Jeong M, Kim R, et al, "Graph Transformer Networks," Advances in Neural Information Processing Systems 32, 2019.

[14] Hassani K, Khasahmadi A H, "Contrastive Multi-View Representation Learning on Graphs," Proceedings of the 37th International Conference on Machine Learning. PMLR, pp4116-4126, 2020.

[15] Zhu Y, Xu Y, Yu F, et al, "Graph Contrastive Learning with Adaptive Augmentation," Proceedings of The Web Conference 2021, pp2069-2080, 2021.

[16] Zhu Y, Xu Y, Yu F, et al, "Deep Graph Contrastive Representation Learning," arXiv preprint arXiv:2006.04131, 2020.

[17] Chen T, Kornblith S, Norouzi M, et al, "A Simple Framework for Contrastive Learning of Visual Representations," Proceedings of the 37th International Conference on Machine Learning. PMLR, pp1597-1607, 2020.

[18] Zeng J, Xie P, "Contrastive Self-supervised Learning for Graph Classification," Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, no. 12, pp10824-10832, 2021.

[19] Ren Y, Liu B, Huang C, et al, "Heterogeneous Deep Graph Infomax," arXiv preprint arXiv:1911.08538, 2019.

[20] Velickovic P, Fedus W, Hamilton W L, et al, "Deep Graph Infomax," ICLR (Poster), vol. 2, no. 3, pp4, 2019.

[21] Peng Z, Huang W, Luo M, et al, "Graph Representation Learning via Graphical Mutual Information Maximization," Proceedings of The Web Conference 2020, pp259-270, 2020.

[22] Qiu J, Chen Q, Dong Y, et al, "GCC: Graph Contrastive Coding for Graph Neural Network Pre-Training," Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp1150-1160, 2020.

[23] Dongyang Li, Dan Yang, and Jing Zhang, "ARB: Knowledge Discovery and Disease Diagnosis on Thyroid Disease Diagnosis integrating Association Rule with Bagging Algorithm," Engineering Letters, vol. 28, no. 2, pp390-399, 2020.

[24] Sun Z, Yin H, Chen H, et al, "Disease Prediction via Graph Neural Networks," IEEE Journal of Biomedical and Health Informatics, vol. 25, no. 3, pp818-826, 2020.

[25] Zheng S, Zhu Z, Liu Z, et al, "Multi-Modal Graph Learning for Disease Prediction," IEEE Transactions on Medical Imaging, vol. 41, no. 9, pp2207-2216, 2022.

[26] Sun Y, Han J, Yan X, et al, "PathSim: Meta path-based top-K similarity search in heterogeneous information networks," Proceedings of the VLDB Endowment, vol. 4, no. 11, pp992-1003, 2011.

[27] Niu S, Yin Q, Song Y, et al, "Label Dependent Attention Model for Disease Risk Prediction Using Multimodal Electronic Health Records," 2021 IEEE International Conference on Data Mining (ICDM). IEEE, pp449-458, 2021.

[28] Li J, Zhou P, Xiong C, et al, "Prototypical Contrastive Learning of Unsupervised Representations," arXiv preprint arXiv:2005.04966, 2020.

[29] Hamilton W, Ying Z, Leskovec J, "Inductive Representation Learning on Large Graphs," Advances in Neural Information Processing Systems 30, 2017.

[30] Dong Y, Chawla N V, Swami A, "Metapath2vec: Scalable Representation Learning for Heterogeneous Networks," Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp135-144, 2017.