Medical Heterogeneous Graph Transformer for Disease Diagnosis

Jianbin Luo, Dan Yang, Yang Liu, Jiaming Liang

Abstract—**The construction of medical heterogeneous map for disease diagnosis using electronic medical records is a research hotspot of medical artificial intelligence. However, existing disease diagnosis networks based on message passing mechanisms have certain limitations. For instance, these models exhibit limited expressiveness and suffer from issues such as over-compression and oversmoothing, which subsequently affect the accuracy of disease diagnosis. To address these issues, a disease diagnosis framework named Trans4DD is proposed, based on the medical heterogeneous graph Transformer. In Trans4DD's medical heterogeneous graph encoder, edge type embeddings and residual connections are introduced. Edge type embeddings effectively capture the node structure and heterogeneous information in the graph. Residual connections aid in avoiding oversmoothing and gradient vanishing problems. A node-level graph Transformer is adopted to overcome the limitations of the message passing mechanism. By employing a multi-hop node context sampling strategy, a broader range of global attention mechanisms is introduced to obtain more accurate patient representations. Experimental results on the MIMIC-IV dataset demonstrate that Trans4DD outperforms other baseline methods in terms of disease diagnosis performance, effectively enhancing the accuracy of disease diagnosis.**

Index Terms—**Disease Diagnosis, Electronic Medical Records, Graph Neural Networks, Graph Transformer , Medical Heterogeneous Graph**

I. INTRODUCTION

With the continuous accumulation and development of medical big data, electronic medical records (EMR) have attracted widespread attention in personalized healthcare services such as disease diagnosis and disease prediction. This paper studies disease diagnosis based on EMR, aiming to determine the possible diseases of patients based on the information in their EMR. The field of graph neural networks has advanced quickly in recent years, demonstrating strong node representation learning capabilities and being widely used in disease diagnosis tasks [1-2]. Currently, most GNN-based disease diagnosis models are based on the message-passing mechanism. However, with the in-depth research and widespread application of GNN, the limitations of the message-passing mechanism have begun to emerge. In practical applications, issues arise such as limited model expressiveness and difficulty in effectively capturing different types of information, which affect the accuracy of disease diagnosis; excessive compression that leads to the model's inability to retain critical medical information; and over-smoothing, which makes the representations of different patients too similar, failing to accurately reflect individual differences, thereby reducing the precision of disease diagnosis. The success of Transformer has garnered significant attention in the field of graphs. Combining it with GNN can address the limitations of the message-passing mechanism inherent in GNN. Transformer with strong representation learning capabilities on homogeneous graphs, based on a global attention mechanism, enables message forwarding to a wider coverage. However, real-world graphs are typically composed of multiple types of nodes and edges, known as heterogeneous graphs. Due to the heterogeneity of these graphs, they contain rich semantic information and exhibit different data characteristics. Recently, methods suitable for heterogeneous graph Transformer have been introduced. For instance, Related work [3] combines the meta-relations of heterogeneous graphs by designing parameters for different types of nodes and diverse edge relations to capture their heterogeneity. Related work [4] extracts meta-paths and meta-graphs from heterogeneous graphs to encapsulate the complex relationships in these graphs, aiding in better learning of node representations. The medical heterogeneous graph utilized for medical diagnosis tasks encompasses three types of nodes: patients (P), drugs (D), and procedures (O). It features two types of edges: patient-drug (indicating the drugs used by patients) and patient-procedure (indicating the procedures performed on patients). Therefore, the integration of graph Transformer and neural networks for disease diagnosis tasks can remedy challenges such as limited model expressiveness and difficulty in effectively capturing diverse information, which impacts the accuracy of disease diagnosis; excessive compression that makes it difficult for the model to remember important medical data; and over-smoothing that renders patient representations overly similar, failing to accurately depict individual differences. Nonetheless, applying the aforementioned Graph Transformer methods to large-scale medical heterogeneous graphs for disease diagnosis tasks presents additional challenges. Specifically, the goal is to move away

Manuscript received Jun 3, 2024; revised Oct 9 2024.This work was supported by the General Scientific Research Project from the Educational Department of Liaoning Province (LJKMZ20220646).

Jianbin Luo is a postgraduate student at School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan, China (e-mail: 17641241848@163.com).

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

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

Jiaming Liang is an undergraduate student at Early Lilac Academy, Harbin Institute of Technology, Weihai, China (e-mail: 18106309767@163.com).

from relying on manually designed meta-paths while still capturing the complex semantic information embedded in large-scale medical heterogeneous graphs to acquire more holistic patient representations. Complex interactions between various node types are seen in medical heterogeneous networks, which frequently result in over-smoothing and gradient vanishing problems. The Graph Transformer employs global attention, but extending it to large-scale medical heterogeneous graphs may result in quadratic complexity challenges.

To address the aforementioned issues, proposes a medical heterogeneous graph Transformer for disease diagnosis ,named Trans4DD. First, this framework uses medical data to create a medical heterogeneous graph. Then, it adopts a meta-path-free approach by introducing learnable edge type embeddings, expanding the original graph attention mechanism to incorporate edge type data into the computation of attention, thereby successfully encapsulating the complex semantic data of the medical heterogeneous graph. A new residual connection mechanism is employed in the medical heterogeneous graph, introducing node residuals and edge residuals to address the over-smoothing and gradient vanishing problems encountered in GNN, thereby enhancing modeling capabilities. Subsequently, a multi-hop node context sampling strategy is used to capture the context sequence of patients, and a node-level Graph Transformer is employed for patient node representation learning, addressing the limitations of the message-passing mechanism. The global attention mechanism is applied in the local environment, emphasizing local structural information, which helps mitigate noise introduced by distant nodes. This approach addresses the quadratic complexity issue when extending the Graph Transformer to large-scale medical heterogeneous graphs, thereby better learning patient embeddings.

II. Related Work

This section discusses related work on disease diagnosis using graph neural networks, including heterogeneous graph neural networks, Graph Transformer, and disease diagnosis based on heterogeneous graph neural networks.

A. Heterogeneous Graph Neural Networks

The design of heterogeneous graph neural network models (HGNN) mainly focuses on modeling heterogeneous information. Currently, methods for utilizing heterogeneous information include HGNN based on meta-paths, such as related work [5-7], and HGNN without meta-paths, such as those in the related work [8-9]. HetGNN [10] aggregates the properties of various node kinds using Bi-LSTM and defines the semantic links between them using meta-paths. HAN [11] presents a heterogeneous graph attention network that combines a hierarchical attention mechanism to learn node-level and semantic-level structures, and uses meta-paths to learn and represent semantic links between various types of nodes in the graph data. The meta-path-free approach addresses the dependency on manually crafted meta-paths. These methods combine type-aware modules for nodes and edges with the message-passing mechanism directly on the original heterogeneous network, allowing the model to concurrently capture structural and semantic information. For example, HGT [12] computes attention scores for neighboring nodes within a one-hop distance in a heterogeneous graph. Simple-HGN [13] provides a baseline model based on GAT that computes attention scores by taking into account both edge and node type embeddings.

B. Graph Transformer

Graph Transformer[14-16] is a graph neural network model based on Transformer, focusing on processing graph data. It leverages the self-attention mechanism and the advantages of Transformer to overcome the limitations of the message-passing mechanism. Related work [17] proposes a Graph Transformer neural network framework specifically designed to handle arbitrary graph data, featuring improved attention mechanisms, positional encoding, normalization layer replacements, and support for edge feature representation. GraphTrans[18] applies a permutation-invariant Transformer module after the conventional GNN module to represent the graph structure. Related work [19] suggests that the key insight of Transformer in graph representation learning is effectively encoding the structural information of graphs and proposes several simple yet effective structural encoding methods to enhance the modeling capability of graph data. A fully linked attention mechanism is usually used by existing Transformer models to process the whole input network. But there are problems with scalability using this method.

C. Disease Diagnosis Based on Heterogeneous Graph Neural Networks

Combining heterogeneous graph neural networks with disease diagnosis can assist the medical field in accurately utilizing various types of medical data, enhancing the performance of disease diagnosis and prediction, and providing more personalized medical services for patients. Related work [20] introduces a healthcare graph convolutional network (HealGCN) based on electronic health records (EHR). It employs a graph convolutional network to serve new users and utilizes a symptom retrieval system to address the scarcity of medical description data. The VGBNet [21] model combines bidirectional self-attention networks with graph convolutional networks to extract global features, random resampling to balance the dataset, and bidirectional self-attention networks to combine local and global features for disease diagnosis and prediction. Related work [22] presents an adaptive graph learning method that can automatically capture latent graph structures.

Existing research on Graph Transformer models mainly focuses on homogeneous graphs. Owing to the different data characteristics of heterogeneous information networks, different processing methods are required. Directly applying Graph Transformer to large-scale graphs leads to quadratic complexity issues. Current heterogeneous graph-based disease diagnosis methods use message-passing mechanisms, which have limitations in model expressiveness, excessive compression, and over-smoothing, thereby reducing the accuracy of disease diagnosis. To address these issues, we proposes a disease diagnosis framework based on a medical heterogeneous graph Transformer. Applying the Graph Transformer to heterogeneous graphs addresses the limitations of the message-passing mechanism. The original

graph attention mechanism is extended to incorporate edge type information in the attention computation by introducing learnable edge type embeddings in the medical heterogeneous graph encoder. This improves the performance of the framework and better captures the rich semantic information of heterogeneous graphs. A multi-hop node context sampling strategy is adopted for patients to avoid the quadratic complexity problem associated with global attention. Furthermore, to mitigate overfitting issues, a lighter-weight GATv2 [23] is used instead of the dot-product attention mechanism, which maintains the number of learnable parameters while improving the model's generalization ability.

III. Preliminaries

Define the concepts of medical heterogeneous graph and Graph Transformer in the disease diagnosis framework as follows:

Definition 1. Medical Heterogeneous Graph. A medical heterogeneous graph is defined as $G = \{V, E, A, R\}$, where V represents the set of all nodes and E represents the set of all edges. It is associated with a node type mapping function ϕ and an edge type mapping function Ψ. Each node $v \in V$ has a mapping $v \to \phi(v)$, and each edge $e \in E$ has a mapping $e \to \Psi(e)$. A and R represent the sets of node types and edge types, respectively, and $|A|+|R|>2$. As shown in Fig.1, the medical heterogeneous graph composed of electronic medical record data contains three types of nodes: patients (P), drugs (D) and procedures (O). There are two types of edges: patient-drug and patient-procedure.

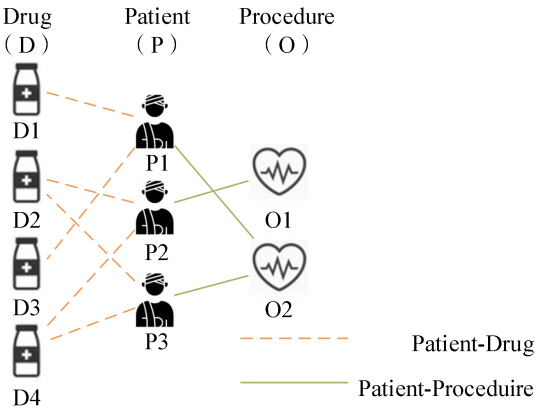

Fig.1 An example of medical heterogeneous graph

Definition 2. Graph Transformer. Standard Transformers usually consist of Multi-Head Self-Attention modules and Feed-Forward Networks. In the following sections, we briefly introduce the Self-Attention module without the multi-head structure. Given an input sequence $H = [h_1, h_2, ..., h_n]^T \in R^{n \times d}$, d is the hidden dimension, and $h_i \in R^d$ represents the hidden representation of position i. MSA projects the input H into the <query, key, value> space using three parameter matrices $W_Q \in R^{d \times d_k}$, $W_K \in R^{d \times d_k}$ and $W_v \in R^{d \times d_v}$, denoted as Q, K and V.

$$Q = HW_Q, K = HW_k, V = HW_v \tag{1}$$

Then, the self-attention mechanism is applied to the corresponding $< Q, K, V >$.

$$MAS(H) = Soft\max\frac{QK^T}{\sqrt{d_k}}V \tag{2}$$

Next, two layers of normalization [24] and residual connections [25] are used to connect the MSA output to the FFN. This produces the output of the L-th Transformer layer, which is represented by the letter H^l:

$$\hat{H}^l = LN(MSA(H^{l-1}) + H^{l-1}) \tag{3}$$

$$H^l = LN(FFN(\hat{H}^l) + \hat{H}^l) \tag{4}$$

The model may learn the feature-based interactions between various points in the input sequence by stacking numerous Transformer layers, where l is the layer number. Subsequently, downstream processes use the final output representation $H^L \in R^{n \times d}$ as the input sequence. This approach allows the model to gradually refine and encode the input information across multiple layers, better capturing the structure and relationships within the data. This approach enables the model to progressively refine and encode the input information across multiple layers, thereby more effectively capturing the structure and relationships within the data.

The commonly used symbols and their meanings in this paper are shown in Table I.

TABLE I
SYMBOLS AND THEIR MEANINGS

Symbol	MEANINGS
G	Medical heterogeneous graph
V,E	Node and edge sets
A,R	Node type and edge type sets
$r_{\psi(e)}^{(l)}$	Edge type embedding
h_i	Patient embedding
$\hat{y}$	Disease diagnosis results

IV. Disease Diagnosis Framework

The proposed disease diagnosis framework Trans4DD is illustrated in Fig. 2. First, medical data from EMR is utilized to construct a medical heterogeneous graph. Then, a medical heterogeneous graph encoder is employed to obtain the embeddings of all nodes, and a multi-hop context sampling strategy is applied to sample the context for patients. Next, a Transformer is introduced to generate more refined patient representations. Finally, the model is trained under supervised classifiy loss to determine the disease type of the patient.

A. Medical Heterogeneous Graph Construction

The medical heterogeneous graph is constructed using the following approach. Using Fig.1 as an example, the constructed medical heterogeneous graph comprises the types of nodes: patients (P), drugs (D), and procedures (O). It includes two types of edges, namely patient-drug (indicating the drugs used by patients) and patient-procedure

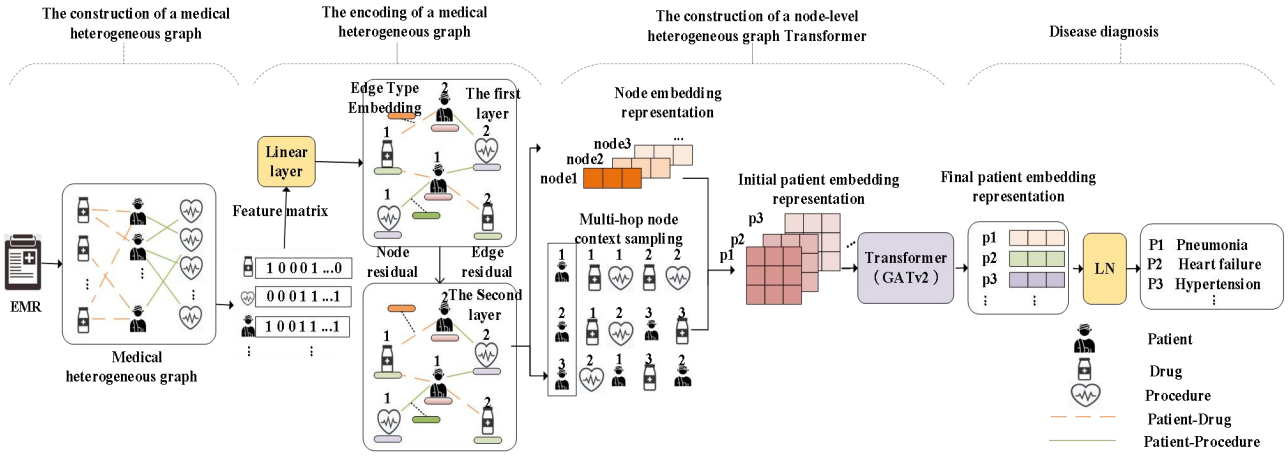

Fig.2 The overall architecture of Trans4DD

(indicating the procedures performed on patients). By connecting nodes with different edge relationships, the medical heterogeneous graph is constructed. These edge relationships reflect the associations between patients, drugs, and procedures. Specifically, given the medical heterogeneous graph $G=(V,E,A,R)$. The set of patients is denoted as $P=\{P_1,P_2,...P_k\}$, where k is the number of patients. The set of drugs is represented as $D=\{D_1,D_2,...,D_n\}$, where n is the number of drugs. The set of procedures is denoted as $O=\{O_1,O_2,...,O_m\}$, where m represents the number of procedures. Considering the adjacency matrix A of the medical heterogeneous graph. If a patient takes the medication D_n or undergoes the procedure O_m, the corresponding position in the adjacency matrix A is set to 1, otherwise, it is set to 0.

B. *Medical Heterogeneous Graph Encoding.*

Learnable Edge Type Embeddings. GAT [26] is a model architecture utilized for graph neural networks. Its primary feature is the incorporation of an attention mechanism, which enables the model to dynamically concentrate on the connection strengths among various nodes, thereby enhancing the representation capability of graph data. GAT exhibits robust abilities in modeling homogeneous graphs but overlooks node or edge type information, rendering it suboptimal for medical heterogeneous graphs comprised of diverse types of nodes and edges. To tackle this challenge, the original graph attention mechanism is extended to encompass edge type information. Thus, when computing attention, both the connection relationships between nodes and the edge types are taken into account. This adjustment allows the model to more effectively adapt to the different types of nodes and edges present in medical heterogeneous graphs, consequently improving the model's representation capability.

With this enhanced attention mechanism, the model performs better when modeling and analyzing medical data because it can better grasp the interactions between nodes in heterogeneous graphs. Specifically, in each layer, a d-dimensional embedding $r^{(l)}_{\Psi(e)}$ is assigned to each edge type $e\to\Psi(e)$, and the edge type embeddings and node embeddings are used to calculate the attention scores as follows:

$$\hat{\alpha}_{ij}=\frac{\exp(Leak\,\mathrm{Re}\,LU(a^T[Wh_i\,\|\,Wh_j\,\|\,W_r r_{\psi(<i,j>)}]))}{\sum_{k\in N_i}\exp(Leak\,\mathrm{Re}\,LU(a^T[Wh_i\,\|\,Wh_k\,\|\,W_r r_{\psi(<i,j>)}]))} \tag{5}$$

Where h_i is the embedding of node i, $\Psi(<i,j>)$ represents the type of edge between node i and node j, and $W^{(l)}_r$ is a learnable matrix that transforms the edge type embeddings.

Residual Connection. To address the issues of over-smoothing and gradient vanishing caused by the complex relationships among different types of nodes in medical heterogeneous graphs, a novel residual connection mechanism is introduced, specifically node residuals and edge residuals. The design of these residual connections aims to enhance the learning capability of neural networks, better capture the intricate relationships in graph data, and improve the performance of the medical diagnosis framework.

Node Residuals. Residual connections should be added to the node representations across layers. By utilizing node residuals, the original information can be preserved while learning the node representations. This ensures that the final node representations contain not only the updated information but also the original information, thereby preventing the issue of over-smoothing.After adding node residuals, the aggregation at layer l^{th} can be represented as:

$$h^{(l)}_i=\sigma(\sum\nolimits_{j\in N_i}\alpha^{(l)}_{ij}W^{(l)}h^{(l-1)}_j+h^{(l-1)}_i) \tag{6}$$

Where $\alpha^{(l)}_{ij}$ is the attention weight of edge $<i,j>$, and σ is the activation function. When the dimension changes at layer l, an additional learnable linear transformation $W^{(l)}_{res}\in R^{d_{l+1}\times d_l}$ is required.

$$h^{(l)}_i=\sigma(\sum\nolimits_{i\in N_i}\alpha^{(l)}_{ij}W^{(l)}h^{(l-1)}_j+W^{(l)}_{res}h^{(l-1)}_i) \tag{7}$$

Edge Residuals. By adding residual connections to the graph's edges and the attention scores, edge residuals are introduced. This ensures that the learned edge representations contain not only the updated information but also retain the original edge information. This helps to prevent the issue of gradient vanishing during information propagation. After obtaining the original attention scores through the equation, residual connections are added to

them.

$$\alpha_{ij}^{(l)} = (1-\beta)\hat{\alpha}_{ij}^{(l)} + \beta\alpha_{ij}^{(l-1)} \quad (8)$$

Where $\beta \in [0,1]$ is a hyperparameter that acts as a scaling factor.

Multi-head Attention. Multi-head attention is used to improve the expressiveness of the model. Equation (6) is specifically used to execute K independent attention mechanisms, and the final representation is formed by concatenating their outputs. The corresponding update rule is:

$$\alpha_{ijk}^{(l)} = (1-\beta)\hat{\alpha}_{ijk}^{(l)} + \beta\alpha_{ijk}^{(l-1)} \quad (9)$$

$$\hat{h}_{ik}^{(l)} = \sum_{j\in N_i} \alpha_{ijk}^{(l)} W_k^{(l)} \hat{h}_i^{(l-1)} \quad (10)$$

$$h_i^l = \sigma(\Big\|_{k=1}^{K} \hat{h}_{ik}^{(l)} + W_{res(k)}^{(l)} h_i^{(l-1)}) \quad (11)$$

Where $\|$ denotes the concatenation operation, and according to Equation (7), it represents the attention scores calculated for the *K*-th linear transformation.

Usually, it is not possible to split the output dimension by the number of heads precisely. After GAT, the representations of layer $final(L^{th})$ are averaged.

$$h_i^{(L)} = \frac{1}{K}\sum_{k=1}^{K} \hat{h}_{ik}^{(L)} \quad (12)$$

C. Node-level Heterogeneous Graph Transformer Construction.

Graph Transformer typically processes the entire graph directly, treating the whole graph as an input sequence to generate node representations. However, the application of Graph Transformer to large-scale medical heterogeneous graph datasets is limited by the high memory costs. Therefore, it is not suitable for large datasets, especially electronic health records used for disease diagnosis. Medical heterogeneous graphs usually have a large number of nodes and edges, and treating them all as an input sequence would result in enormous memory requirements. Additionally, medical heterogeneous graph data often involve specific tasks, such as disease diagnosis, which typically require a single graph rather than multiple graphs. As a result, conventional Graph Transformer techniques are inappropriate in this particular data context and are unable to acquire expressive patient representations in an efficient manner.

The sampling sequence representation of patient node P is denoted as $p(v) = \{v, v_1, ..., v_{s-1}\}$, and the input representation of Transformer is denoted as $H_v^P = [h_v^p, h_{v_1}^p, ..., h_{v_{s-1}}^p] \in R^{S\times d}$, where d is the embedding dimension of the nodes, and $h_{v_i}^p$ represents the node features from the medical heterogeneous graph encoder. Overall, the entire input of the Graph Transformer is denoted as $H^P = [H_1^p, H_2^p, ..., H_N^p]^T \in R^{N\times S\times d}$.

After encoding with the L-layer Transformer, the output $H^L = [H_1^L, H_2^L, ..., H_N^L]^T \in R^{N\times S\times d}$ is obtained and then the patient representation needs to be derived through a readout function.

$$h_v = READOUT(h_{p(v)[i]}^L \mid v_i \in p(v)) \quad (13)$$

Where $p(v)[i]$ represents the *i*-th patient in $p(v)$. In practice, the patient node representation in the sequence is directly used as the output node embedding $h_v = h_{s(v)[0]}^L$.

Large numbers of learnable parameters are characteristic of the basic Transformer architecture, which makes it difficult to train models effectively with little supervision and may result in overfitting problems. The attention computation approach in the Transformer structure is changed to further improve Transformer's adaptability for node-level representation learning on medical diverse graphs and lower the possibility of overfitting.The original self-attention mechanism is no longer used, as it involves more learnable parameters, which increases the model's complexity. GATv2 is a general dynamic attention function that, compared to the standard dot-product self-attention mechanism, performs more robustly in Transformer while having fewer parameters. The following are the differences between GAT and GATv2:

$$GAT: \ \alpha(h_i, h_j) = Leak\,\mathrm{Re}LU(a^T \cdot [Wh_i \| Wh_j]) \quad (14)$$

$$GATv2: \alpha(h_i, h_j) = a^T \cdot Leak\,\mathrm{Re}LU(W \cdot [h_i \| h_j]) \quad (15)$$

Using GATv2 as an alternative method reduces the number of learnable parameters in the model, improves its training efficiency, and reduces the risk of overfitting, making Transformer more suitable for handling disease diagnosis tasks on medical heterogeneous graphs. In Trans4DD, the computation formula for the Transformer layer is:

$$H^l = LN(H^{l-1} + GATv2MAS(H^{l-1})) \quad (16)$$

Where $GATv2MAS(\cdot)$ represents the dynamic attention using GATv2.

D. Disease Diagnosis

Through the above computation, the patient representation h_v is obtained. A linear layer $\phi_{Linear}(\cdot;\theta_{pre})$ with parameters θ_{pre} is used to obtain the predicted values for different disease labels. The calculation formula is as follows:

$$\hat{y}_v = \phi_{Linear}(h_v; \theta_{pre}) \quad (17)$$

Where $\hat{y}_v \in R^C$ is the prediction and C is the number of classes. Additionally, a L_2 normalization is further added to $\hat{y}_v$ to achieve stable optimization.

Given the trained patient V_{tr}, cross-entropy is used as the overall loss. The calculation formula is as follows:

$$L = \sum_{v=V_{tr}} CROSSENT(\hat{y}_v, y_v) \quad (18)$$

Where $CROSSENT(\cdot)$ is the cross-entropy loss, and $y_v \in R^C$ is the one-hot vector encoding the label of node v.

V. Experiments and Evaluation

This section first introduces the datasets utilized in the experiments and the data preprocessing procedures. Next, it presents the evaluation metrics employed in the experiments and the baseline methodologies. Then, the performance of Trans4DD is elaborated upon through experimental data.

A. Dataset and Preprocessing

The study made use of the MIMIC-IV (Medical Information Mart for Intensive Care IV) dataset, which includes clinical information from 450,000 hospital admissions and more than 190,000 patients. In the data preprocessing stage, six representative disease categories were selected from the MIMIC-IV dataset, including Myocardial Infarction, Pneumonia, Heart Failure, Coronary Atherosclerosis, Cirrhosis, and Hypertension. Key information was extracted from patient records, including patient identifier (Subject_id), hospital admission identifier (Hadm_id), medication usage, medical procedures, gender, and disease diagnosis categories. Each patient has a unique Subject_id in the dataset, but they can correspond to multiple hospital admission records (multiple Hadm_id). To facilitate the processing of different patients, Subject_id and Hadm_id were used as the primary keys for new patients. Patients lacking information on their medications or procedures were disqualified during the patient selection process. Only medications of the major (major) kind were chosen for pharmaceutical consumption, while drugs of the base (BASE) type were eliminated. Additionally, a random selection of up to 30 drugs used by each patient was made. When processing patients' medical procedure data, only the most important procedures for each patient were selected based on the importance ranking. Specifically, procedures with an importance ranking of 1, which indicates the most important procedure for the patient, were chosen. After data preprocessing, the final dataset comprised 9,860 patients. The statistics of the processed dataset are summarized in Table II.

TABLE II
STATISTICS OF DATASETS

Disease label	Number of patients
Myocardial Infarction	1866
Pneumonia	1159
Heart Failure	2417
Coronary Atherosclerosis	2916
Cirrhosis	842
Hypertension	660
Total	9860

B. Evaluation Metrics

Micro-F1 and Macro-F1 are used as evaluation metrics for disease diagnosis tasks.

1) Micro-F1

Micro-F1 is an evaluation metric used in multi-class scenarios. It calculates the F1 score for each class and then computes their weighted average as an overall performance measure. The specific calculation is as follows:

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

2) Macro-F1

Macro-F1 is an evaluation metric used in multi-class scenarios. It calculates the F1 score for each class and then computes their arithmetic average as an overall performance measure. The specific calculation is as follows:

$$Macro-F1=\frac{1}{n}\sum_{i=1}^{n}\frac{2TP_i}{(2TP_i+FP_i+FN_i)} \tag{20}$$

In which, n is the number of disease categories, and TP_i, FP_i, FN_i represent the counts of true positives, false positives, and false negatives for the i-th disease category, respectively.

C. Baselines

To comprehensively evaluate the performance of Trans4DD, it is compared with baseline methods from three main categories: homogeneous graph-based, heterogeneous graph-based with meta-path, and heterogeneous graph-based without meta-path.

1) Learning Methods Based on Homogeneous Graphs

- **GCN**[27] employs neighborhood aggregation operations to collect information from adjacent nodes to generate node representations.
- **GAT**[26] implements an additional attention mechanism to achieve weighted aggregation of neighborhood information, rather than simple average aggregation.
- **Transformer**[28] introduces a network architecture that relies solely on attention mechanisms, completely discarding recurrent and convolutional components.

2）Learning Methods Based on Heterogeneous Graphs with Meta-Paths

- **HAN**[11] introduces a hierarchical attention mechanism, which encompasses node-level attention and semantic-level attention.

3）Learning Methods Based on Heterogeneous Graphs without Meta-Paths

- **Simple-HGN**[13] proposes a basic model based on graph attention networks that computes attention scores by taking into account both node and edge type embeddings at the same time.
- **HINormer**[9] makes use of a self-attention technique composed of two primary parts: a heterogeneous relation encoder and a local structure encoder. These elements enable efficient learning of node representations by precisely capturing heterogeneous information and local structural characteristics inside the graph.

D. Parameter Setting

For GCN, GAT, Transformer, HAN, Simple-HGN and HINormer, the parameters are set according to the original papers, and the best performance is reported.

The Trans4DD utilizes the Adam optimizer [29] during training; the learning rate is established at 0.0001; the dimensionality of edge type embeddings is 64; the heterogeneous graph encoder comprises 2 layers; the number of attention heads is 4; and training is conducted for 200 epochs.

E. Experimental Results and Analysis

The performance of Trans4DD is evaluated through a disease diagnosis task. In Trans4DD, the data is divided as follows: 50% is used for training, 20% for validation, and 30% is allocated for testing purposes. Throughout the training process, the model progressively modifies its parameters to optimize the loss function, thus improving the accuracy of disease diagnosis. The experimental findings

appear in Table III, allowing for the following conclusions to emerge:

TABLE III
RESULTS OF DISEASE DIAGNOSIS USING DIFFERENT METHODS

Model	Micro-F1	Macro-F1
GCN	82.01	82.36
GAT	85.59	85.86
Transformer	86.54	87.00
HAN	81.54	80.86
Simple-HGN	85.12	85.59
HINormer	86.82	87.07
Trans4DD	87.56	87.80

The Trans4DD consistently outperforms other baseline methods. This result indicates that using a node-level heterogeneous graph Transformer, along with a multi-hop node context sampling strategy, expands the range of global attention mechanisms for learning patient node representations. Trans4DD exceeds the performance of GCN, GAT, and Transformer models. This highlights that the medical heterogeneous graph encoder introduces edge-type embeddings, capturing both node structure and heterogeneous information. Trans4DD also surpasses HAN, demonstrating that the encoder alleviates the reliance on manually crafted meta-paths. Additionally, it outperforms HINormer, showing that the inclusion of node and edge residuals helps avoid issues of excessive compression and smoothing during the model's training. Finally, Trans4DD's superiority over Simple-HGN underscores the effectiveness of using a node-level heterogeneous graph Transformer.

F. Variant Analysis

To evaluate the validity of the Trans4DD architecture, we proposes three variants of Trans4DD: Trans4DD_WOTE, Trans4DD_WORC, and Trans4DD_SA. Trans4DD_WOTE does not introduce edge type embeddings and calculates attention scores solely through node embeddings. Trans4DD_WORC does not introduce node residuals and edge residuals. Trans4DD_SA uses the standard self-attention mechanism in the Transformer structure. The performance of these variants is compared with Trans4DD on the MIMIC-IV dataset. The experimental results are evaluated using Micro-F1 and Macro-F1 as metrics, and the results are presented in Fig. 3.

From this, the following conclusions can be drawn:

- After removing edge type embeddings, Trans4DD_WOTE is unable to capture the heterogeneous information of the medical heterogeneous graph, resulting in a decrease in performance. This also indicates the necessity of edge type embeddings for the Trans4DD framework.
- After removing node residuals and edge residuals, the performance of Trans4DD_WORC significantly declines. This indicates that the problems of over-smoothing and gradient vanishing are successfully alleviate by node residuals and edge residuals, enhancing the framework's overall performance.
- Trans4DD_SA performs worse after implementing self-attention. This suggests that the overfitting issue can be successfully resolved by utilizing Transformer's dynamic attention mechanism of GATv2.

Overall, the above ablation experiments demonstratethe necessity of each component of the Trans4DD.

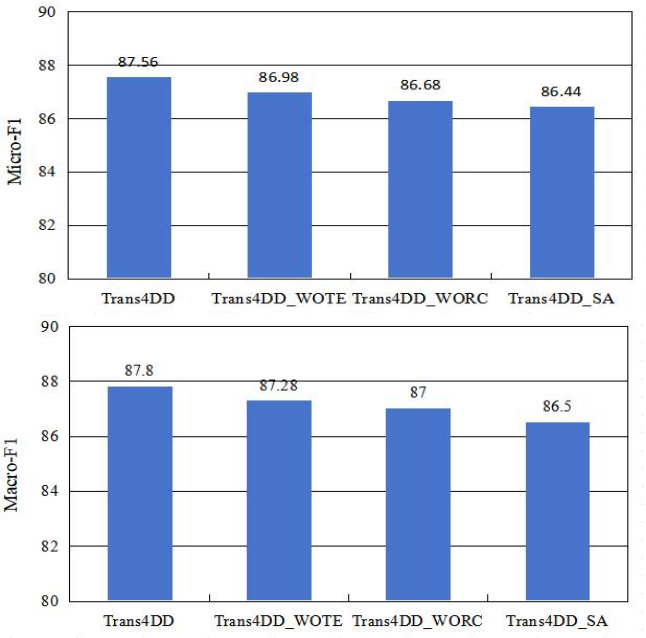

Fig.3 The comparison of Trans4DD and its variant

G. Visualization

The patient nodes from the test set are projected onto a two-dimensional space using t-SNE [30] to give an intuitive evaluation of the model's disease diagnosis results. The results of the visualization are shown in Fig. 4, where different hues correspond to different disease categories.

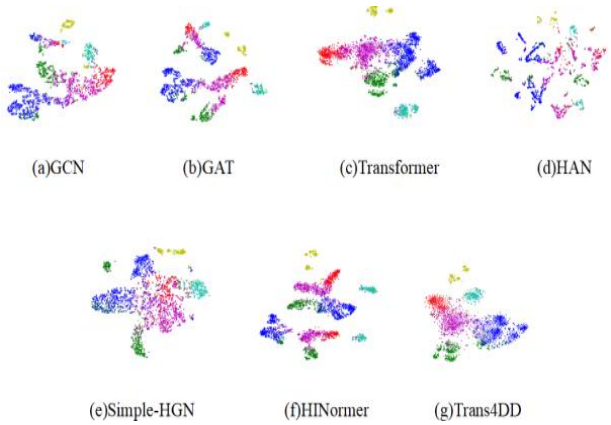

Fig.4 Visualization of the patient nodes embedding

In HAN, patient nodes with different labels do not cluster well together, while in GCN, patient nodes with different labels remain poorly separated. In contrast, contrastive learning methods, such as Simple-HGN and HINormer, show more distinct result boundaries and fewer overlapping areas. In the Simple-HGN method, different types of nodes mix together. In the HINormer method, the absence of edge type embeddings and residual connections results in a more scattered distribution of nodes with the same label. Compared to these methods, Trans4DD creates more

definite borders, more effectively divides patient nodes with different labels, and more effectively groups patient nodes with the same label together. This finding indicates that the learned patient node embeddings exhibit higher quality.

H. Parameter Analysis

This section examines the analysis of parameters' impact on Trans4DD, focusing on three important hyperparameters: the dimension *d* of edge type embeddings, the hidden dimension *dl* of the Graph Transformer, and the sequence length *s* of context sampling. By varying the values of *d*, *dl*, and *s* while keeping other parameters constant, we observe the performance changes of Trans4DD. Micro-F1 and Macro-F1 are used as evaluation metrics. Tables IV to VII respectively show the performance variations of Trans4DD under different edge type embedding dimensions, Graph Transformer hidden dimensions, and context sampling sequence lengths.

a) The dimension of edge type embeddings

As shown in Table IV, as the dimension d of edge type embeddings increases, the performance of Trans4DD first improves and then declines. The optimal performance is achieved when the embedding dimension is set to 64.

TABLE IV
THE PERFORMANCE VARIATION OF TRANS4DD UNDER THE EMBEDDING DIMENSIONS OF DIFFERENT EDGE TYPES

Edge type embedding dimension	Micro-F1	Macro-F1
40	87.39	87.34
50	87.36	87.68
60	87.32	87.59
70	87.32	87.46

b) The hidden dimension of the Graph Transformer

Table V shows that when the Graph Transformer's hidden dimension rises, Trans4DD performs better. This improvement implies that the model can capture more intricate patterns and correlations in the data when there is a bigger hidden dimension. Given the enhanced efficiency of the disease diagnosis model, we configured the hidden dimension of the Graph Transformer to 256, ensuring optimal performance while maintaining computational feasibility.

TABLE V
THE PERFORMANCE VARIATION OF TRANS4DD IN DIFFERENT GRAPH TRANSFORMER HIDDEN DIMENSIONS

The hidden dimension of the Graph Transformer	Micro-F1	Macro-F1
32	84.79	85.00
64	86.38	86.59
128	86.95	87.31
256	87.56	87.80

c) The sequence length of context sampling

As shown in Table VI, as the sequence length of context sampling increases, the performance of Trans4DD first decreases and then increases. Due to the enhanced efficiency of the disease diagnosis model, the sequence length of context sampling is set to 70.

TABLE VI
THE PERFORMANCE VARIATION OF TRANS4DD UNDER SEQUENCE LENGTHS SAMPLED WITH DIFFERENT CONTEXTS

The sequence length of context sampling	Micro-F1	Macro-F1
40	87.46	87.70
50	87.19	87.40
60	87.15	87.43
70	87.56	87.80

I. Model Convergence Performance Analysis

The loss function acts as one of the evaluation metrics during the training process of Trans4DD. Figure 5 shows the curve of the loss function over the training iterations. Notably, at 200 epochs, the curve for the validation loss (val_loss) starts to stabilize and shows little further decline, suggesting that the model has reached convergence.

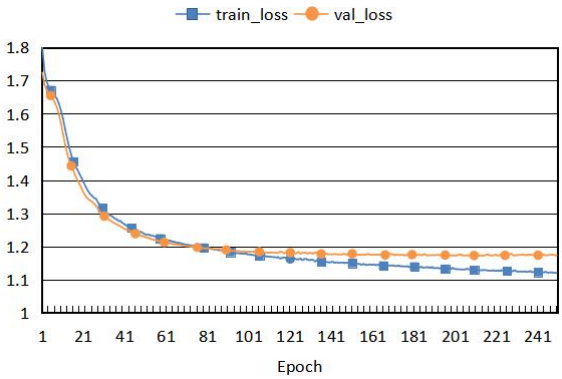

Fig.5 The Loss change curve

VI. CONCLUSIONS AND FUTURE WORK

To address the limitations of the message-passing mechanism in graph neural networks, we proposes a disease diagnosis framework based on a heterogeneous graph Transformer, named Trans4DD. This framework constructs a medical heterogeneous graph using EMR. The medical heterogeneous graph encoder introduces edge type embeddings and residual connections that capture the structural and heterogeneous information of nodes in the graph, thus obtaining more comprehensive node representations. The node-level Graph Transformer incorporates a broader global attention mechanism for patient node representation learning, propagating medical information throughout the entire medical heterogeneous graph. Trans4DD effectively learns better patient representations and outperforms baseline techniques, according to experimental results on the MIMIC-IV dataset.

Future research will continually improve the Trans4DD framework. Firstly, researchers will introduce more heterogeneous information into the medical heterogeneous graph encoder, including multimodal patient medical data such as medical text information and X-rays. Integrating multimodal features can provide more accurate patient representations, further enhancing disease diagnosis performance. Secondly, researchers will introduce self-supervised learning or transfer learning techniques to better utilize medical data, improving the model's generalization ability and adaptability.

REFERENCES

[1] Zhengkang Zhang, Dan Yang, and Yu Zhang, "Disease Diagnosis Based on Multi-View Contrastive Learning for Electronic Medical Records," IAENG International Journal of Applied Mathematics, vol. 53, no.3, pp1114-1122, 2023.

[2] Long R, Yang D, Liu Y. "DiseaseNet: A Novel Disease Diagnosis Deep Framework via Fusing Medical Record Summarization," IAENG International Journal of Computer Science, vol.49, no.3, pp808-817, 2022.

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

[4] Yao S, Wang T, Wan X. "Heterogeneous Graph Transformer for Graph-to-Sequence Learning," in Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp7145-7154, 2020.

[5] Liu J, Song L, Wang G, et al. "Meta-HGT: Metapath-aware HyperGraph Transformer for heterogeneous information network embedding," Neural Networks, pp 65-76,2020.

[6] Fu X, Zhang J, Meng Z, et al. "Magnn: Metapath aggregated graph neural network for heterogeneous graph embedding," in Proceedings of The Web Conference 2020, pp 2331-2341, 2020.

[7] Hong H, Guo H, Lin Y, et al. "An attention-based graph neural network for heterogeneous structural learning," in Proceedings of The AAAI Conference on Artificial Intelligence,vol.34, no.4, pp4132-4139, 2020.

[8] Liu Z, Zheng V W, Zhao Z, et al. "Semantic proximity search on heterogeneous graph by proximity embedding," in Proceedings of The AAAI Conference on Artificial Intelligence, vol.31, no.1, pp154-160, 2017.

[9] Mao Q, Liu Z, Liu C, et al. "Hinormer: Representation learning on heterogeneous information networks with graph transformer," in Proceedings of the ACM Web Conference 2023. pp599-610, 2023.

[10] Zhang C, Song D, Huang C, et al. "Heterogeneous graph neural network," in Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 2019, pp793-803.

[11] Wang X, Ji H, Shi C, et al. "Heterogeneous graph attention network," in The World Wide Web Conference, pp2022-2032, 2019.

[12] Hu Z, Dong Y, Wang K, et al. "Heterogeneous graph transformer," in Proceedings of The Web Conference 2020.pp2704-2710, 2020.

[13] Lv Q, Ding M, Liu Q, et al. "Are we really making much progress? revisiting, benchmarking and refining heterogeneous graph neural networks," in Proceedings of The 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining. pp1150-1160, 2021.

[14] Devin Kreuzer, Dominique Beaini, Will Hamilton,et al. "Rethinking graph transformers with spectral attention," Advances in Neural Information Processing Systems 34 (2021), pp21618–21629.

[15] Ying C, Cai T, Luo S, et al. "Do transformers really perform badly for graph representation?," Advances in Neural Information Processing Systems, 2021, pp28877-28888.

[16] Zhao J, Li C, Wen Q, et al. "Gophormer: Ego-graph transformer for node classification," ArXiv Preprint 2021. Available:https://arxiv.org/abs/2110.13094

[17] Dwivedi V P, Bresson X. " A generalization of transformer networks to graphs," ArXiv preprint 2020. Available:https://arxiv.org/abs/2012.09699

[18] Wu Z, Jain P, Wright M, et al. "Representing long-range context for graph neural networks with global attention," Advances in Neural Information Processing Systems, 2021, pp13266-13279.

[19] Ying C, Cai T, Luo S, et al. "Do Transformers Really Perform Badly for Graph Representation?," Advances in Neural Information Processing Systems, 2021,pp28877-28888.

[20] Wang Z, Wen R, Chen X, et al. "Online Disease Self-diagnosis with Inductive Heterogeneous Graph Convolutional Networks," in Proceedings of The Web Conference 2021, pp3349-3358.

[21] Li Y, Zhao X, Ma M, et al. "VGBNet:: a disease diagnosis model based on local and global information fusion," International Journal of Computing Science and Mathematics, vol.17, no.2, pp107-122, 2023.

[22] 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.

[23] Brody S, Alon U, Yahav E. "How attentive are graph attention networks?," ArXiv Preprint 2021.Available:https://arxiv.org/abs/2105.14491

[24] Ba J L, Kiros J R, Hinton G E. "Layer normalization," ArXiv Preprint 2016. Available:https://arxiv.org/abs/1607.06450

[25] He K, Zhang X, Ren S, et al. "Deep residual learning for image recognition," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp 770-778.

[26] Veličković P, Cucurull G, Casanova A, et al. "Graph attention networks," ArXiv Preprint 2017. Available: https://arxiv.org/abs/1710.10903.

[27] Kipf T N, Welling M. "Semi-supervised classification with graph convolutional networks," ArXiv Preprint ArXiv:1609.02907, 2016.

[28] Vaswani A, Shazeer N, Parmar N, et al. "Attention is all you need" ArXiv, 2017. Available: https://arxiv.org/abs/1706.03762.

[29] Kingma D P, Ba J. "Adam: A Method for Stochastic Optimization," ArXiv: Learning, 2014. Available: https://arxiv.org/abs/1412.6980v6.

[30] Laurens van der Maaten, Geoffrey Hinton. "Visualizing Data using t-SNE ," Journal of Machine Learning Research. pp2579–2605, 2008.