Message Passing and Edge Attention in Heterogeneous Graph Neural Networks for Disease Diagnosis

Xiaodong Zhu, Dan Yang, Yang Liu

Abstract—**Message passing, as the core mechanism of Heterogeneous graph neural networks, can efficiently capture the latent relationships between nodes in the disease diagnosis task. However, the heterogeneity and complexity of medical data make it challenging for conventional message passing to accurately distinguish key information from noise, leading to semantic confusion, over-smoothing, and gradient vanishing issues. To address these challenges, we propose MPEA4DD, a heterogeneous graph neural network that integrates a custom message passing process combining dynamic edge attention with a state selection mechanism. We construct a medical heterogeneous graph from EMRs in MIMIC-III and MIMIC-IV, encode nodes and edges into a unified feature space, and apply our message-passing module, in which an edge attention network dynamically adjusts edge weights and a state-selection network assigns each node one of three interaction states (aggregation, dissemination, or integration) to mitigate the interference of irrelevant neighboring information. Furthermore, to mitigate gradient vanishing and excessive smoothing in deep architectures, we combine contextual semantic fusion with residual connections on GATv2 attention aggregation, achieving effective global information integration and stable gradient propagation in deep layers. Thus, evaluations on both MIMIC-III and MIMIC-IV demonstrate that MPEA4DD significantly outperforms other baseline models in disease diagnosis accuracy.**

Index Terms—**Message Passing, Attention, Disease Diagnosis, Electronic Medical Records, Medical Heterogeneous Graph**

I. Introduction

With the advancement of medical informatization, Electronic Medical Records (EMRs)[1] have become increasingly critical as primary data sources for disease diagnosis and therapeutic decision-making. EMRs contain patients' basic information, medications, procedures, and laboratory test results, providing strong support for personalized medicine. However, the high-dimensional heterogeneity and complex relational structures of medical data [2-3] pose significant challenges for shallow machine learning approaches to fully extract latent information. Achieving efficient disease diagnosis using heterogeneous graph data remains a significant challenge.

Graph Neural Networks (GNNs) [4] have demonstrated formidable capabilities in modeling complex relational networks. In particular, Heterogeneous Graph Neural Networks (HGNNs) [5-6] exhibit unique advantages in modeling diverse medical entities and their interrelationships , thereby significantly advancing EMR-based disease diagnosis research.

Heterogeneous graph-based disease diagnosis models establish a new model for clinical decision-making by integrating multi-typed medical entities, including patients, drugs, and procedures. However, such models fail to explicitly distinguish the semantics of different types of medical entities during message passing, leading to feature confusion during propagation. Different patients' medical records may carry varying diagnostic significance, but conventional aggregation mechanisms fail to distinguish their roles in disease diagnosis, allowing irrelevant information to interfere with the model's discriminative capability. Related work [7] introduces a relation-aware attention mechanism; however, its use of static feature projection and lack of type constraints still fail to effectively preserve the semantic specificity of nodes. Theoretical studies have shown that as the number of node types in a heterogeneous graph increases, the neighborhood aggregation of conventional GNNs significantly weakens feature distinguishability, further exacerbating the issue of semantic confusion.

Existing medical graph models enhance diagnosis performance through local neighborhood aggregation; however, their short-range message passing mechanism is fundamentally limited in capturing long-range dependencies within patient disease trajectories. Conventional GNNs, constrained by limited receptive fields, struggle to model such cross-temporal dependencies. This challenge is particularly acute in heterogeneous graph scenarios, where a patient's disease progression often involves multiple medical events. Conventional neighborhood aggregation may restrict interactions between different types of information, making it difficult to comprehensively capture contextual semantic information. Models such as HAN [8] and HGT [9], which rely on fixed meta-paths or static attention mechanisms, are fundamentally incapable of effectively modeling dynamic entity relationships in medical scenarios. As a result, they difficulty capture time-sensitive clinical interaction patterns. Although FastGTN [10] employs a graph transformation network to capture high-order adjacency relationships and SlotGAT [11] leverages a slot allocation mechanism to

Manuscript received April 2, 2025; revised July 29, 2025.

Xiaodong Zhu is a postgraduate student at School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan, China (e-mail: 18085879473@163.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).

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

mitigate global information loss, their static designs still fail to effectively handle time-sensitive interactions, increasing the risk of misdiagnosis in complex cases.

In deep graph neural networks for medical scenarios, as the number of GNN layers increases, gradient vanishing [12] leads to the failure of deep node feature updates, significantly impacting the modeling of long-term treatment pathways. In medical data, variations in diagnostic standards across institutions, examination errors, and other noise factors interfere with the stability of model convergence. Moreover, the scarcity of rare disease samples in EMR data exacerbates class imbalance, further impairing model discriminability for minority categories. GIN [13] proposes to balance information propagation and feature stability through residual connections and normalization regularization strategies. However, its static optimization mechanism fails to adapt to the dynamic complexity of medical scenarios, ultimately leading to performance degradation in deep layers.

To address semantic confusion, global dependency omission, and gradient stability issues in medical heterogeneous graphs, we propose MPEA4DD, a heterogeneous graph neural network-based disease diagnosis model that not only resolves semantic confusion but also enhances diagnostic accuracy and generalization capability. The primary contributions can be outlined as follows:

- We propose a dynamic neighbor selection mechanism, which employs a dynamic decision module based on node states and neighborhood features to effectively filter neighborhood information. An edge attention mechanism is introduced to dynamically assign weights to edge attributes within the same dimension. By leveraging a discrete selection strategy and the edge weights obtained from edge attention, the model effectively filters out non-critical neighbors, reducing semantic confusion in heterogeneous feature propagation and enhancing its ability to capture critical relationships.
- The attention computation is improved based on the GATv2 architecture to more accurately capture contextual semantic relationships between nodes. A context fusion strategy and residual connections are incorporated to enhance cross-layer information flow, mitigating gradient vanishing and over-smoothing [14] issues in deep GATv2 architectures. By integrating multi-level semantic representations, the model enhances the stability and accuracy of the disease diagnosis task.
- We conduct a series of experiments and comparative analyses to evaluate the effectiveness of our proposed model in the disease diagnosis task, training and testing it on two real-world datasets: MIMIC-III and MIMIC-IV. Experimental comparisons with baseline models confirmed the superiority of the MPEA4DD model in disease diagnosis.

II. RELATED WORK

A. Heterogeneous Graph Neural Networks

Recent advances in HGNNs seek to deliver both semantic richness and computational efficiency on complex heterogeneous graphs. HGNNs extend standard GNNs to handle multiple node and edge types by incorporating type-specific aggregation and attention. HAN uses a meta-path–based attention mechanism to aggregate along predefined semantic routes, but its fixed paths limit adaptability and can cause semantic mixing. HetGNN [15] applies random walks for neighborhood sampling and BiLSTM encoders to capture local heterogeneous features, yet it lacks a global context perspective and may miss cross-type interactions. RGCN [16] introduces relation-specific weight matrices to propagate features across different edge types, but deep models tend to over-smooth and suffer vanishing gradients, weakening feature discrimination.

B. Message Passing Mechanism

The message passing mechanism [17] is one of the core design principles in GNNs. It updates node representations by repeatedly passing and aggregating information between nodes and their neighbors. The conventional homogeneous graph GCN [18] and GraphSAGE [19] often use fixed aggregation functions to integrate neighbor features. However, in heterogeneous graphs, different types of nodes and edges often have their own semantic features. If a single aggregation strategy is still used, it can easily lead to semantic confusion or information redundancy, making it difficult to fully utilize the richness of heterogeneous relationships.

In recent years, researchers have begun to introduce more flexible message passing mechanisms in heterogeneous graphs. CoGNN [20] proposes a collaborative learning approach across multiple subnetworks to dynamically adjust the interaction between nodes and their neighbors. Similarly, SlotGAT introduces a representation allocation mechanism to provide independent attention spaces for different types of nodes and edges, thereby alleviating feature confusion. Currently, message passing should not be merely a simple aggregation; instead, it needs to dynamically assign appropriate weights to each edge or neighbor to capture finer semantic differences in heterogeneous graphs.

C. Attention Mechanisms

Attention mechanisms [21], by dynamically assigning weights, empower the model with the ability to focus on key information and have become an important tool in graph neural networks for handling complex structures. GAT [22] is the first to introduce self-attention into graph learning, calculating neighborhood weights based on node feature similarity to achieve weighted aggregation of local information. The attention mechanism of GAT uses fixed parameters that cannot accommodate the diverse feature distributions of heterogeneous nodes, resulting in semantic mixing across types. To address this limitation, GATv2 [23] is the first to introduce a dynamic attention mechanism, which adjusts the weight computation function in real-time based on input features, significantly enhancing the model's ability to express heterogeneous relationships. Although GATv2 still has limitations in flexible modeling of multi-type nodes and edges, the dynamic nature of the attention mechanism and the computational complexity in complex heterogeneous graphs remain challenging research problems.

While GATv2 improves flexibility, it still cannot model long-range dependencies, motivating Transformer-based, global attention. Graph Transformer [24], as a novel attention

mechanism, has demonstrated excellent performance in modeling complex structures and long-range dependencies. It overcomes the limitations of GAT and GATv2 in local weighted aggregation by introducing a Transformer-based global attention mechanism, effectively capturing semantic relationships between distant nodes in heterogeneous graphs. Recently, AGHINT [25] employs a Transformer-based architecture with an attribute-guided module and a relation encoder fused via multi-head self-attention. However, like other Transformer-based models, it faces high computational complexity, which limits its efficiency on large-scale heterogeneous graphs. Therefore, how to achieve accurate dynamic modeling of medical heterogeneous relationships while maintaining efficient computation remains the core direction for optimizing attention mechanisms.

D. Diagnosis of Diseases

The disease diagnosis models based on GNNs have made significant progress in recent years, particularly in the modeling of heterogeneous medical data and patient status diagnosis [26-27]. Since patient medical records contain multimodal information such as medications, procedures, and tests, effectively integrating these heterogeneous features is one of the key challenges. H-GCN [28] learns patient features by performing convolution operations on heterogeneous medical graphs, but it faces issues of feature over-smoothing and gradient vanishing. To improve the accuracy and efficiency of disease diagnosis, some researchers integrate genome or protein interaction networks with clinical EMR, gaining a more comprehensive understanding of disease diagnosis and development. Therefore, graph neural networks have also shown great potential in fields such as medical image analysis and drug discovery. Subsequent research should explore scaling GNNs to large-scale, high-dimensional, and highly heterogeneous medical datasets to enable more precise and personalized diagnosis and treatment.

In conclusion, although existing research has made gradual progress in the disease diagnosis task, there are still many challenges in handling complex heterogeneous data, especially in terms of semantic confusion, over-smoothing, and gradient vanishing issues. Striking a balance between rich multimodal feature extraction and computational scalability is essential for truly efficient diagnosis.

III. Disease Diagnosis Framework

Fig.1 illustrates the model framework of MPEA4DD, a heterogeneous graph neural network for disease diagnosis based on message passing and edge attention.

First, a medical heterogeneous graph is constructed based on EMR data. A node encoder and an edge-type encoder are then used to obtain representations of nodes such as patients, drugs, and procedures, as well as representations of edges such as patient-drug and patient-procedure relationships. In the convolutional layer, edge attention scores are fused with dynamically generated weights to establish adaptive message passing. This process leverages a state generation neural network to compute the retention probability of edge weights, enabling different types of nodes to interact optimally based on their states (aggregation, broadcasting, and integration). The edge attention network encodes edge attributes and dynamically adjusts base edge weights through an attention mechanism. The resulting attention coefficients refine and integrate these weights before message aggregation, yielding node representations that incorporate information from neighboring nodes.

After two convolutional layers, the model incorporates contextual information fusion and residual connections to enhance the stability of message passing, prevent feature over-smoothing, and improve the interaction capabilities across different types of nodes. Finally, the patient node representations are extracted and normalized to obtain the final disease diagnosis results. The following subsections detail each module's implementation.

A. Medical Heterogeneous Graph Construction

To facilitate accurate disease diagnosis, we construct a medical heterogeneous graph from EMR data to extract diagnostic information by capturing complex relationships among heterogeneous nodes.

The medical heterogeneous graph is defined as $G=(V,E,\mathcal{N},\mathcal{M})$, where V represents the set of nodes, E represents the set of edges, $\Phi(\cdot)$ and $\Psi(\cdot)$ denote the node-type mapping function and edge-type mapping function, respectively. Each node $v \in V$ has a mapping relation $v \to \Phi(v)$, which assigns a specific type to the node. Similarly, each edge $e \in E$ has a mapping relation $e \to \Psi(e)$, which determines the type of relationship between connected

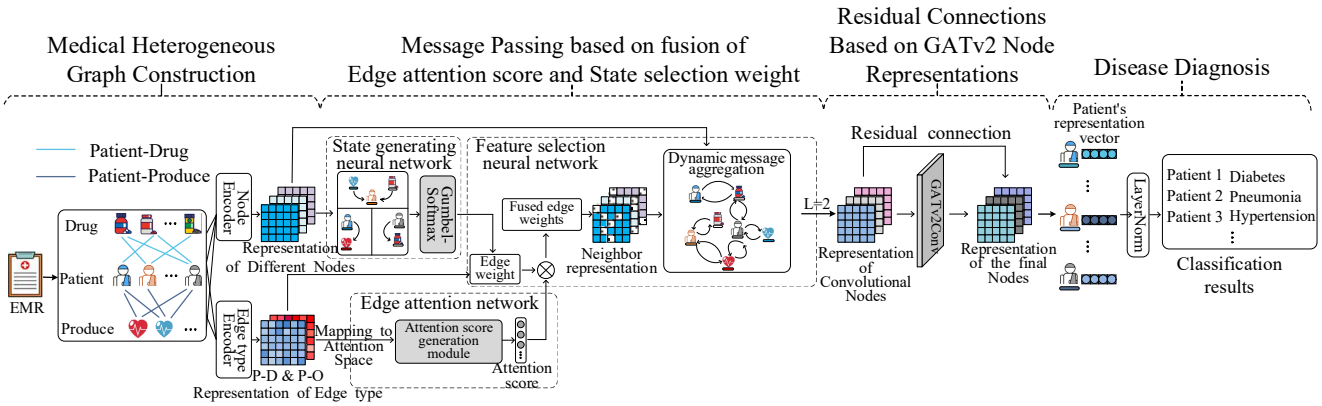


Fig.1 The overall Model of MPEA4DD

nodes. The patient node set is defined as $P=\{P_1,P_2,...,P_m\}$ where m denotes the total number of patients. The patient node set is defined as $P=\{P_1,P_2,...,P_m\}$ where m denotes the total number of patients. The drug node set is defined as $D=\{D_1,D_2,...,D_n\}$ where n denotes the total number of drugs. And the procedure node set is defined as $O=\{O_1,O_2,...,O_k\}$ where k denotes the total number of procedures. It includes two edge types: patient-drug edges (P-D) for prescribed drugs and patient-procedure edges (P-O) for medical procedures performed. The heterogeneous graph G is defined such that $\mathcal{N}$ and $\mathcal{M}$ represent the sets of node types and edge types, respectively, satisfying the condition $|\mathcal{N}|+|\mathcal{M}|>2$. The adjacency matrix M is constructed by setting $M_{ij}=1$, if a patient is connected to a prescribed drug or D_n a medical procedure O_k, and $M_{ij}=0$ otherwise.

Since heterogeneous nodes have different feature dimensions, a type-specific linear mapping matrix is used to initialize the features of all node types. For each node v and each type $t\in\{P,D,O\}$, the feature initialization formula is given by:

$$h_v^{(0),t}=\begin{cases}W_t^{(0)}\cdot x_v, & \text{if } t=\Phi(v)\\ 0, & \text{if } t\neq\Phi(v)\end{cases} \tag{1}$$

Where $x_v\in\mathrm{R}^{d_v}$ represents the original feature vector of node v, and $W_t^{(0)}\in\mathrm{R}^{d_1\times d_v}$ is the type-specific mapping matrix for type t, which transforms the feature dimension to d_1. The mapping function $\Phi(v)$ maps each node to its corresponding type identifier, such as *P*, *D*, or *O*.

B. Message passing based on fusion of edge attention score and state selection weight

1) State generating neural network

This module generates the retention probability of the edge $e_{uv}\in\mathrm{R}^e$ based on the current node features $h_v^{(\ell)}\in\mathrm{R}^d$ and the features of its neighboring nodes $\{h_v^{(\ell)}\mid u\in\mathcal{N}(v)\}$. These edge weights are used to dynamically adjust the weights of edges in the graph, optimizing the information propagation process. First, the weighted sum of the features of the neighboring nodes is computed to obtain the aggregated features of node v from its neighbors:

$$\tilde{h}_v=\sum_{u\in\mathcal{N}(v)}W_{agg}\cdot h_u^{(\ell)} \tag{2}$$

Where $\mathcal{N}(v)$ denotes the neighbor set of node v, and W_{agg} is the aggregation weight matrix, which controls the contribution of the neighboring node features. After obtaining the aggregated neighbor feature, the current node feature $h_u^{(\ell)}$ is concatenated with the aggregated neighbor feature $\tilde{h}_v$ and fed into a fully connected network to compute the probability distribution over three states (AGGREGATE, DISSEMINATE, and INTEGRATE):

$$p_v^{(\ell)}=Softmax(W_a\cdot[h_v^{(\ell)}\,\|\,\tilde{h}_v])+b_a \tag{3}$$

Where $W_a\in\mathrm{R}^{3\times 2d}$ is the weight matrix, $b_a\in\mathrm{R}^3$ is the bias vector, and $[h_v^{(\ell)}\,\|\,\tilde{h}_v]\in\mathrm{R}^{2d}$ represents the concatenation of the current node feature and the aggregated neighbor feature. The neural network ultimately outputs the state probability distribution $p_v^{(\ell)}=[P_A,P_D,P_I]\in\mathrm{R}^3$.

To sample a discrete selection vector $a_v^{(\ell)}$ from the output probability distribution $p_v^{(\ell)}$, the graph structure is dynamically adjusted to enable more effective patient node classification. Directly using argmax for sampling is non-differentiable and prevents gradient optimization. Therefore, Gumbel-Softmax sampling [29] is used to achieve a differentiable discrete selection. The Gumbel-Softmax formula is as follows:

$$\alpha_v^{(\ell)}=Gumbel\text{-}Softmax(\log(p_v^{(\ell)}),\tau) \tag{4}$$

which is equivalent to:

$$\alpha_v^{(\ell)}=\frac{\exp((\log p_{v,i}^{(\ell)}+g_i)/\tau)}{\sum_{j\in\{A,D,I\}}\exp((\log p_{v,j}^{(\ell)}+g_j)/\tau)} \tag{5}$$

Where $g_i=Gumbel(0,1)$ is sampled from the Gumbel distribution, introducing randomness to simulate sampling from a discrete distribution. The temperature parameter τ governs the smoothness of the sampling process: When $\tau\to 0$, the output sharply converges to a one-hot vector, mimicking deterministic argmax selection; conversely, as $\tau\to\infty$, the output becomes a uniform distribution where all states are equally probable, effectively maximizing entropy.

2) Feature selection neural network

The goal of the message passing neural network is to update node features based on adjacency matrix information, thereby optimizing the representation of patient nodes. At each layer ℓ, this process relies on the current node feature $h_v^{(\ell)}$ and the selection vector $\alpha_v^{(\ell)}$ obtained through Gumbel-Softmax sampling. In heterogeneous graphs, connections between different node types serve distinct functional roles, necessitating the computation of base edge weights $w_{uv}^{base,(\ell)}$ for each edge $e=(u,v)$. This weight is determined by the state vectors $\alpha_u^{(\ell)}$ and $\alpha_v^{(\ell)}$ of the connected nodes:

$$w_{uv}^{base,(\ell)}=f(\alpha_u^{(\ell)},\alpha_v^{(\ell)}) \tag{6}$$

Where $f(\cdot)$ is the edge weight computation function, which determines the importance of the edge in the message passing process. A higher edge weight means that the edge has a greater influence on the target node during propagation, while a lower edge weight may result in the information from that edge being diminished. After calculating the edge weights, it is necessary to aggregate information from the neighboring nodes in order to update the current node's feature representation.

To achieve this, a weighted summation is performed over the filtered neighbor set $\mathcal{M}_v$ to construct the node's neighborhood representation:

$$m_v^{(\ell)} = \sum_{u \in \mathcal{M}_v} w_{uv}^{base,(\ell)} \cdot W_e \cdot h_u^{(\ell)} \quad (7)$$

Here, $W_e \in \mathrm{R}^{d \times d}$ is the edge-type-related weight matrix, ensuring that different types of relationships have distinct impacts on information aggregation, $h_u^{(\ell)}$ is the feature vector of the neighboring node u at layer ℓ, and $m_v^{(\ell)}$ represents the information gathered from the neighbors by the current node, where key neighbors have their influence amplified and irrelevant neighbors' influence suppressed.

After completing the neighbor feature aggregation, the current node feature $h_v^{(\ell)}$ is combined with the aggregated neighbor feature $m_v^{(\ell)}$ for updating, to generate the node representation for the next layer:

$$\hat{h}_v^{(\ell+1)} = ReLU(W_u \cdot [h_v^{(\ell)} \,||\, m_v^{(\ell)}] + b_u) \quad (8)$$

Where W_u is the update parameter matrix, and b_u is the bias vector. This update mechanism ensures that node features can dynamically adjust to changes in the network structure, thereby improving the accuracy of patient node classification. Through the role of the environment necural network, the model can better aggregate key information and suppress noise, thereby enhancing overall performance.

3) Edge attention network

This module is mainly used to encode edge attributes and, through the attention mechanism, dynamically adjust the base edge weights, thereby more finely aggregating neighbor information to update node features. For edge attribute encoding, each edge attribute e_{uv} is first transformed into an embedding vector via an embedding layer:

$$\tilde{e}_{uv} = Embedding(e_{uv}) \in \mathrm{R}^{d_e} \quad (9)$$

Next, the embedded vector $\tilde{e}_{uv}$ passes through a fully connected layer for a linear transformation, adjusting it to the fixed dimension required by the model. A reshape operation is then applied to format it appropriately for attention computation, facilitating subsequent inner product operations and attention score calculations. Mapping the embedded vector to the attention space:

$$e'_{uv} = reshape(W_e^{EAT} \tilde{e}_{uv}, (H, d_e)) \quad (10)$$

Where H is the number of attention heads, defaulting to 1, and W_e^{EAT} is the weight matrix for edge attribute encoding.

Next, the encoded edge attribute e'_{uv} is used to compute the initial edge attention score by taking the inner product with the learnable parameter β_{edge}:

$$ee_{uv} = < e'_{uv}, \beta_{edge} > \quad (11)$$

The scores are then processed with a LeakyReLU activation function and normalized across all edges originating from the same source node u, yielding the final edge attention coefficients:

$$\beta_{uv} = \frac{\exp(LeakyReLU(e_{uv}))}{\sum_{v' \in \mathcal{N}(u)} \exp(LeakyReLU(e_{uv'}))} \quad (12)$$

Finally, the adjusted edge weight is obtained by integrating the base edge weight $w_{uv}^{base,(\ell)}$ from the previously mentioned Gumbel-Softmax with the edge attention score $\beta_{uv}^{(h)}$:

$$w_{uv}^{EAT,(\ell)} = \sum_{k=1}^{K} \beta_{uv}^{(k)} w_{uv}^{base,(\ell)} \quad (13)$$

Where K is the number of attention heads. Next, the adjusted edge weights are used to re-aggregate neighbor information:

$$m_v^{(\ell)} = \sum_{u \in \mathcal{M}_v} w_{uv}^{EAT,(\ell)} \cdot W_e \cdot h_u^{(\ell)} \quad (14)$$

Following this computational stage, the edge attribute information is not only effectively encoded but also weighted and adjusted through the attention mechanism. This enhances the influence of key edges, providing richer and more precise information support for subsequent node feature updates.

C. Residual Connections Based on GATv2 Node Representations

1) GATv2 Aggregation

After adjusting the edge weights and aggregating neighbor information, the model then processes the node features using GATv2 to enhance the ability to capture contextual information. GATv2 improves upon the original GAT mechanism, making the attention computation more robust to changes in input features, thereby enhancing the model's ability to capture complex relationships in heterogeneous graphs.

$$x_{v,context}^{(\ell+1)} = GATv2Conv(\hat{h}_v^{(\ell+1)}, edge\_index) \quad (15)$$

This process is equivalent to:

$$x_{v,context}^{(\ell+1)} = \|_{h=1}^{H'} \sigma(a_{uv}^{(h)} \cdot W^{(h)} \cdot \hat{h}_v^{(\ell+1)}) \quad (16)$$

Where H' is the number of attention heads in GATv2, $a_{uv}^{(h)}$ is the attention weight, representing the importance of neighboring node u to the target node v, $W^{(h)}$ is the transformation matrix corresponding to the h-th attention head, σ is the nonlinear activation function. Unlike the mean or weighted sum aggregation models in standard GNNs, GATv2 allows the model to assign different attention weights to each neighboring node. This enables the model to emphasize the information from key neighbors and suppress the interference from irrelevant nodes. Finally, the context features aggregated by GATv2, context embedding features $x_{v,context}^{(\ell+1)}$ will be used for subsequent node feature fusion and the final disease diagnosis task. This allows patient nodes to better leverage neighbor information, thereby improving the accuracy of disease diagnosis.

2) Semantic-Fused Residual Links

After obtaining the contextual information through GATv2, further fusion is performed to enhance the final node representation. This model introduces a residual connection mechanism and dynamically adjusts fusion weights for different sources of contextual semantics, ensuring gradient stability.

First, the dynamic fusion weight $\gamma_v^{(\ell+1)}$ is computed as follows:

$$\gamma_v^{(\ell+1)} = Sigmoid(W_{dyn} \cdot x_{v,context}^{(\ell+1)}) \quad (17)$$

Where W_{dyn} is a learnable parameter matrix that determines how the fusion weight is computed.

The dynamically computed fusion weights are then applied to perform a weighted residual connection between the contextual features $x_{v,context}^{(\ell+1)}$ and the updated node features $\hat{h}_v^{(\ell+1)}$:

$$h_v^{(\ell+1)} = \gamma_v^{(\ell+1)} \odot x_{v,context}^{(\ell+1)} + (1-\gamma_v^{(\ell+1)}) \odot \hat{h}_v^{(\ell+1)} \quad (18)$$

Where $\odot$ denotes element-wise multiplication. This model allows to dynamically adjust the proportion of information from different sources, enhancing the flexibility of feature representation. The residual connections effectively preserve the original information while guiding the model to perform stable information updates across different layers, further enhancing the stability of the training process. Finally, the fused node feature $h_v^{(\ell+1)}$ will serve as the input for the next layer, thereby enhancing the model's performance in the patient node classification task.

D.Disease Diagnosis

After completing node feature extraction and information fusion, the final representation of the patient node is processed to obtain the disease diagnosis results. To ensure feature stability and enhance the model's generalization capability, we first perform normalization on patient nodes $v \in V_{patient}$:

$$Z = LayerNorm(h_v^{(L)}) \quad (19)$$

Here LayerNorm normalization adjusts the mean and variance of feature values, ensuring that features have a similar scale across different nodes. This helps mitigate issues like gradient explosion and vanishing gradients, thereby enhancing the stability of model training. Then, a linear transformation is applied to map the aggregated features to the final disease category:

$$\hat{y} = Softmax(W_c Z + b_c) \quad (20)$$

where W_c represents the weight matrix of the classification layer, and b_c denotes the bias term. The *Softmax* function transforms the linear outputs into a probability distribution, representing the likelihood of the patient node belonging to each disease category.

Finally, the cross-entropy loss function is employed to supervise the training process:

$$\mathcal{L} = -\sum_{i=1}^{C} y_i \log(\hat{y}_i) \quad (21)$$

Where C is the number of disease classes, $\hat{y}_i$ is the predicted probability for class iii produced by the Softmax in Eq. (20), and y_i is the one-hot encoded ground-truth label. This loss measures the divergence between the predicted distribution and the true distribution, penalizing low confidence in the correct class and discouraging incorrect predictions. By minimizing $\mathcal{L}$ with a gradient-based optimizer, the model's weights are iteratively adjusted so that its output distribution increasingly aligns with the true labels, thereby improving diagnostic accuracy and generalization.

IV. EXPERIMENTS AND EVALUATION

This study presents a detailed account of the experimental setup, including the datasets, evaluation metrics, baseline models, parameter configurations, and result analyses. The proposed model is rigorously evaluated through extensive experiments conducted on the MIMIC dataset. Furthermore, ablation studies, visualization analyses, and hyperparameter sensitivity evaluations are performed to comprehensively examine the individual contributions of each component within the model architecture.

A. Dataset and Preprocessing

The dataset used in this study is the Medical Information Mart for Intensive Care (MIMIC) electronic medical records database. This database, funded by the National Institutes of Health, was established in 2003 and jointly developed by the MIT Laboratory for Computational Physiology, Harvard Medical School's Beth Israel Deaconess Medical Center, and Philips Healthcare. The MIMIC dataset includes basic information, clinical records, medications, procedures, and other key medical data, providing extensive support for clinical research and disease diagnosis in critically ill patients. We utilize the MIMIC-III and MIMIC-IV datasets, which cover different types of critical illnesses and medical procedures. To enhance the model's generalization ability and evaluation effectiveness, representative disease data ate extracted from both datasets to construct a heterogeneous medical graph for the disease diagnosis task. The specific patient count statistics are shown in Table I.

TABLE I
STATISTICS OF DATASETS

Disease label	MIMIC-III	MIMIC-IV
	Number of patients	Number of patients
Coronary Disease	2750	0
Pneumonia	0	1256
Respiratory Failure	388	0
Septicemia	0	2148
Disseminated Infections	1830	0
Myocardial Infarction	0	934
Heart Failure	803	1629
Respiratory Failure	1229	0
Diabetes	0	1557
Hypertension	0	807
Total	7000	8331

In the MIMIC-III dataset, we select five representative diseases as experimental data: Coronary Disease, Disseminated Infections, Respiratory Failure, Heart Failure, and Gastritis. The experiment focuses on 7,000 patients diagnosed with these diseases, involves 1,379 drugs, and includes 563 procedures with their corresponding medical record texts. And in the MIMIC-IV dataset, six diseases are selected: Septicemia, Heart Failure, Diabetes, Pneumonia, Myocardial Infarction, and Hypertension. The experiment focuses on 8,331 patients, involves 1,692 drugs, and includes 843 procedures with their corresponding medical record

texts.

B. Evaluation Metrics

We use Micro-F1 and Macro-F1 scores as evaluation metrics for the model's disease diagnosis task in the experiment. The F1 score is a comprehensive metric that balances precision and recall, making it particularly suitable for multi-class classification tasks. In the medical heterogeneous graph node classification task, since class distributions are often imbalanced, using accuracy alone is prone to being dominated by the majority class. Therefore, this study adopts the F1 score to more comprehensively evaluate the mode's performance.

1) Micro-F1

Micro-F1 evaluates model performance by merging the predictions of all classes into a single category and then calculating the precision and recall for this aggregated category. Micro-F1 effectively reflects the overall performance across all samples and demonstrates strong robustness, especially in cases of class imbalance.

Its specific calculation formula is as follows:

$$Micro\text{-}F1 = \frac{2 \times TP}{2 \times TP + FP + FN} \tag{21}$$

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

2) Macro-F1

Macro-F1 is a metric that evaluates model performance by calculating the F1-score for each class and then averaging them. Unlike Micro-F1, Macro-F1 focuses more on the classification performance of each category. Therefore, it is more sensitive to the performance of minority class samples in imbalanced class distributions. The calculation formula is as follows:

$$Macro\text{-}F1 = \frac{1}{n}\sum_{i=1}^{n}\frac{2 \times TP_i}{2 \times TP_i + FP_i + FN_i} \tag{22}$$

By combining Micro-F1 and Macro-F1, the model's performance in the disease diagnosis task is able to be comprehensively evaluated. Micro-F1 emphasizes the overall sample accuracy, while Macro-F1 focuses on the balance of performance across different categories. In the subsequent experiments, both metrics are reported to more comprehensively reflect the model's performance across different task scenarios.

C. Baselines

To evaluate the model's performance, the following baseline models are compared with the MPEA4DD model proposed:

- GCN[18] aggregates neighboring node features using spectral convolution through feature transformation based on the Laplacian matrix.
- GAT[22] uses a self-attention mechanism to perform weighted aggregation of neighboring node features and introduces multi-head attention to enhance its performance.
- HAN[8] introduces a hierarchical attention mechanism, which includes both node-level attention and semantic-level attention. These two attention mechanisms are associated with meta-paths and implemented using GAT.
- GIN[13] updates node features by employing additive aggregation and multilayer perceptron (MLP), significantly enhancing the model's ability to improve node embedding representations.
- HGT[9] employs a multi-head attention mechanism and node-type-specific projection operations to model different types of nodes and edges, improving performance on complex heterogeneous graph tasks.
- HHGT[30] introduces a hierarchical Transformer architecture that separately models type-level and distance-level heterogeneity through (k, t)-ring neighborhoods, achieving improved representation learning in heterogeneous graphs.
- FastGTN[10] automatically learns long-range dependencies and high-order adjacency relationships between nodes, significantly enhancing the model's ability to represent complex graph structures and enabling faster training.
- SlotGAT[11] introduces a slot allocation mechanism, where independent slots are assigned to each node type, maintaining representations in their respective feature spaces. It also incorporates slot attention techniques in the final layer, improving the accuracy of heterogeneous graph classification tasks.
- CoGNN[20] integrates a collaborative learning mechanism into heterogeneous graph node classification tasks, where the cooperative optimization of environmental networks and decision networks achieves dynamic adjustment and efficient aggregation of node features.

D. Parameter Settings

For these models, we use the original paper settings and report their best results.

The MPEA4DD model uses the Adam optimizer for parameter optimization, with an initial learning rate set to 0.001. The maximum number of training epochs is set to 200, and the batch size is set to 8. The node embedding layer dimension is set to 64, the hidden layer dimension is set to 16, and the number of convolution layers is set to 2. To prevent overfitting, dropout regularization is applied between layers with a dropout rate set to 0.5. The model uses the Gumbel-Softmax mechanism for differentiable discretization of decision probabilities, with an initial temperature parameter τ set to 0.01 to control the discretization effect and prediction stability.

The performance of MPEA4DD is evaluated through the disease diagnosis task. The experimental data is divided as follows: 50% as the training set for parameter learning, 30% as the validation set for hyperparameter tuning, and 20% as the test set for final performance evaluation. This data split ensures sufficient training data while providing reliable verification of generalization ability.

E. Experimental Results and Analysis

The final experimental results of the MPEA4DD model and all baseline models are shown in Table II. The performance of conventional graph convolution models, GCN and GIN, is significantly lower in heterogeneous medical graphs compared to other models. Their local neighborhood aggregation mechanism struggle to effectively model the complex interactions between patients, drugs, and

TABLE II
PERFORMANCE COMPARISON OF DIFFERENT MODEL

Models	MIMIC-III		MIMIC-IV	
	Micro-F1	Macro-F1	Micro-F1	Macro-F1
GCN	84.76	80.31	80.16	80.47
GAT	86.43	84.21	82.16	82.86
HAN	84.76	81.55	76.00	76.33
GIN	86.90	83.43	80.84	81.49
FastGTN	80.13	80.13	78.96	78.96
HGT	85.81	82.30	77.28	78.25
HHGT	85.09	81.88	78.64	79.26
SlotGAT	87.62	85.22	82.12	82.49
CoGNN	90.18	87.64	87.82	88.36
MPEA4DD	**91.31**	**89.76**	**88.85**	**89.44**

procedures. The GAT model, which introduces an attention mechanism, improves performance through weighted aggregation, but still has significant limitations in cross-type semantic differentiation. Among the heterogeneous graph-specific models, HGT demonstrates robust performance through relation-specific design, but its adaptability to dynamic medical relationships is insufficient; SlotGAT further optimizes heterogeneous feature fusion through a slot allocation mechanism, but its static slot design limits the ability to model dynamic priorities. HHGT introduces a hierarchical attention structure, but its complex multi-level aggregation increases optimization difficulty and may lead to overfitting in limited-data medical scenarios. The CoGNN performs better in multi-source information integration, validating the effectiveness of dynamic interaction strategies, but the gradient stability issues during deep network training still limit performance improvement.

The MPEA4DD proposes in this paper exhibit optimal performance on both datasets. On the MIMIC-III dataset, Micro-F1 and Macro-F1 reach 91.31 and 89.76, respectively; on the MIMIC-IV dataset, Micro-F1 and Macro-F1 reach 88.85 and 89.44, respectively. The experimental results indicate that the MPEA4DD has achieved excellent performance on complex medical heterogeneous graphs and outperforms other baseline models in the disease diagnosis task.

To further evaluate the statistical significance of the proposed model, we conduct 10 independent runs on both MIMIC-III and MIMIC-IV datasets with different random seeds. The average performance and 95% confidence intervals are reported as follows: on MIMIC-III, the Micro-F1 and Macro-F1 scores are 0.9131 ± 0.0010 and 0.8976 ± 0.0015, respectively; on MIMIC-IV, the Micro-F1 and Macro-F1 scores are 0.8885 ± 0.0009 and 0.8944 ± 0.0009. These results demonstrate the model's strong generalization ability and consistent performance under different random initializations.

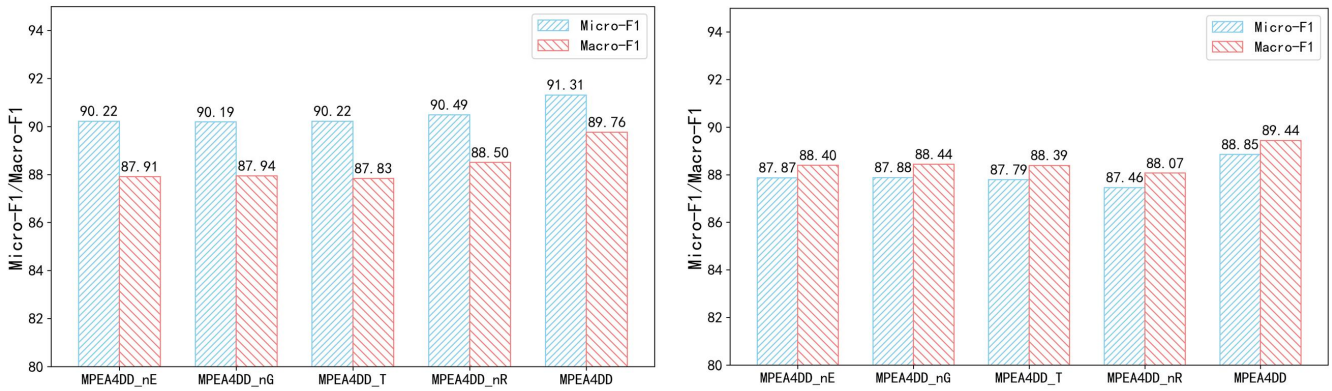
F. Ablation Experiment

To validate the effectiveness of the MPEA4DD architecture, four variant models are designed: MPEA4DD_nE, MPEA4DD_nG, MPEA4DD_T, and MPEA4DD_nR. MPEA4DD_nE removes the edge weight allocation mechanism and computes attention weights solely based on node features. MPEA4DD_nG removes the GATv2 aggregation mechanism. MPEA4DD_T replaces the GATv2 module with the standard Transformer self-attention mechanism. Finally, MPEA4DD_nR removes the residual connections and does not update node features through residual connections.

This study compares the performance of these variant models with the original MPEA4DD model on the MIMIC-IV dataset. The results, as shown in Fig.2, lead to the following conclusions.

- After removing the edge weight allocation mechanism, the performance of MPEA4DD_nE significantly declined. Due to the lack of capability in modeling complex relationships in the heterogeneous graph, both Micro-F1 and Macro-F1 decrease. This indicates that the edge attention mechanism plays a crucial role in capturing the associations between heterogeneous node features and is a key component in enhancing model performance.
- After removing the GATv2 aggregation mechanism, the performance of MPEA4DD_nG slightly declines. This indicates that GATv2 makes a significant contribution to contextual semantic aggregation, and its attention mechanism effectively enhances node feature representation in complex heterogeneous graph structures.
- After replacing GATv2 with the Transformer self-attention mechanism, the performance of MPEA4DD_T declines. This suggests that in heterogeneous medical graphs, the dynamic attention mechanism of GATv2 is more advantageous than the standard self-attention mechanism, as it can more precisely capture the importance differences among neighboring nodes. Therefore, GATv2 plays an irreplaceable role in contextual feature fusion.
- After removing the residual connections, the performance of MPEA4DD_nR drops significantly. This validates the importance of residual connections in deep models for mitigating gradient vanishing and long-range dependency issues. The residual mechanism effectively enhances the continuity of feature propagation through direct shortcut connections, helping to maintain the stability of node features.

In summary, the above ablation experiments validate the necessity and effectiveness of each component in MPEA4DD. The edge attention mechanism, GATv2 aggregation mechanism, Transformer replacement mechanism, and residual connections all play crucial roles in enhancing model



(a) MIMIC-III (b) MIMIC-IV

Fig.2 The comparison of MPEA4DD and its variant

performance. The intact model demonstrates superior performance in the medical heterogeneous graph node classification task.

G. Visualization

The t-SNE [31] is a dimensionality reduction algorithm. To intuitively evaluate the model's performance in the disease diagnosis task, t-SNE is used to map patient nodes from the MIMIC-IV test set into a two-dimensional space. The visualization results are shown in Fig.3, where different colors represent different types of disease labels.

From this figure, it is observed that although GCN and GIN can cluster nodes of the same class relatively well, there is still significant mixing between nodes with different labels, making it difficult to form clear classification boundaries. GAT and HAN show slight improvements in node aggregation effectiveness, but there is still a significant degree of node mixing, making it difficult to achieve clear category separation. HGT shows improvements in capturing heterogeneous information, but still fails to completely resolve the issues of node stacking and mixing. The FastGTN, although demonstrating a certain degree of clustering effectiveness, still exhibits relatively unclear classification boundaries.

MPEA4DD not only excels in the aggregation of nodes with the same label but also forms clear classification boundaries, effectively reducing the mixing areas of nodes from different categories. This demonstrates that MPEA4DD can better learn the embedding representations of patient nodes, exhibiting stronger discriminative power and generalization ability.

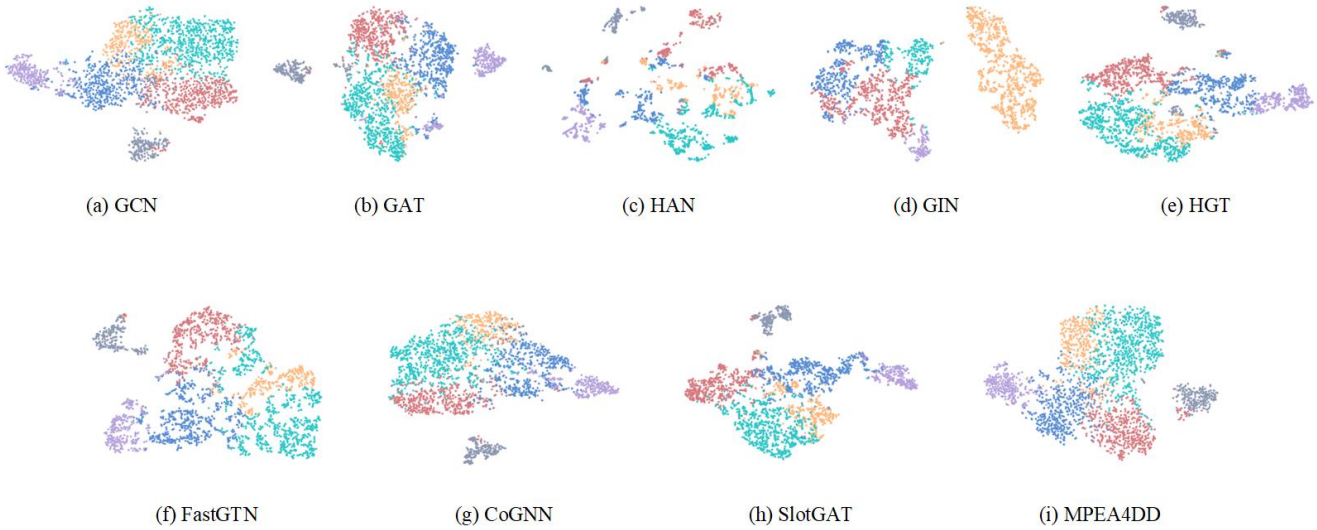
H. Hyperparameters Study

The performance of MPEA4DD varies significantly under different hyperparameter settings. In our experiment, all other parameters are kept constant, and only a single parameter value is changed at a time to study the impact of that parameter setting on the model, aiming to explore the optimal hyperparameter settings for the model's performance.

1) Feature embedding dimension

Feature embedding dimension is an important parameter that affects feature representation capability and model capacity. Five embedding dimensions, 16, 32, 64, 128, and 256, were compared and the experimental results are presented in Table III.

From the experimental results, it can be observed that as the embedding dimension increases, the model's performance first improves and then declines. The model performs best when the feature dimension is set to 64. This is because appropriately increasing the feature dimension enhances the model's ability to represent complex features. However, when the feature dimension becomes too large, it may lead to overfitting and noise accumulation, ultimately affecting performance. Therefore, choosing 64 as the embedding



(a) GCN (b) GAT (c) HAN (d) GIN (e) HGT

(f) FastGTN (g) CoGNN (h) SlotGAT (i) MPEA4DD

Fig.3 Visualization of the patient disease diagnosis

dimension is a reasonable decision.

TABLE III
THE INFLUENCE OF FEATURE DIMENSIONS

Feature dimensions	MIMIC-III		MIMIC-IV	
	Micro-F1	Macro-F1	Micro-F1	Macro-F1
16	90.69	88.80	88.25	88.55
32	91.82	89.22	88.44	88.91
64	91.31	89.76	88.85	89.44
128	91.00	90.14	88.52	89.10
256	90.79	88.80	88.14	88.69

2) Convolutional layers

Experiments were designed to compare models with 2, 3, 4, and 5 convolutional layers to verify the impact of model depth on classification performance. The experimental results are presented in Table IV.

TABLE IV
THE INFLUENCE OF CONVOLUTIONAL LAYERS

Convolutional layers	MIMIC-III		MIMIC-IV	
	Micro-F1	Macro-F1	Micro-F1	Macro-F1
2	91.31	89.76	88.85	89.44
3	90.36	87.55	88.25	88.79
4	90.46	88.02	87.37	87.94
5	87.77	84.07	85.62	86.31

The results indicate that as the number of layers increases, the model's performance gradually declines. The best performance is achieved with 2 layers, suggesting that an appropriate number of layers can effectively capture the feature information of patient nodes. However, excessive layers may lead to gradient vanishing and over-smoothing, ultimately reducing classification performance. Therefore, choosing 2 layers as the model depth strikes a balance between performance and complexity.

3) Dropout

Four dropout rates 0.1, 0.3, 0.5, and 0.7 were compared to assess the impact of the dropout hyperparameter on classification performance and the results are presented in Table V.

TABLE V
THE IMPACT OF DROPOUT RATE

Dropout rates	MIMIC-III		MIMIC-IV	
	Micro-F1	Macro-F1	Micro-F1	Macro-F1
0.1	91.11	89.12	88.56	89.17
0.3	91.14	89.36	88.69	89.24
0.5	91.31	89.76	88.85	89.44
0.7	90.64	88.14	87.41	88.07

The experimental results indicate that as the dropout rate increases, the model's performance initially improves but declines after exceeding a certain threshold. When the Dropout rate is set to 0.5, the model achieves its best performance, demonstrating that an appropriate dropout rate can effectively prevent overfitting and enhance generalization. However, an excessively high dropout rate of 0.7 causes the model to lose too much feature information, leading to a significant performance drop. Therefore, 0.5 is chosen as the optimal dropout rate, striking a good balance between regularization effectiveness and model performance.

I. Convergence properties analysis

Fig.4 presents the training and validation loss curves of the proposed model on the MIMIC-IV dataset over 120 epochs. Both curves show a steady decline, indicating effective model optimization and convergence. The validation loss closely tracks the training loss throughout the training process, with no significant divergence, suggesting strong generalization and the absence of overfitting. After approximately 90 epochs, the validation loss stabilizes, further confirming convergence. These results demonstrate that the proposed model achieves stable and reliable performance on the multi-class classification task, exhibiting consistent training dynamics and robust generalization capability.

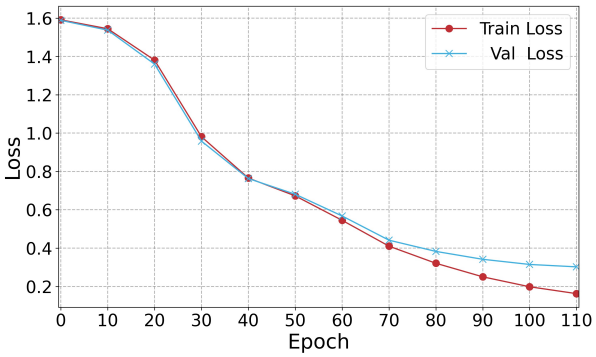


Fig.4 Training and validation loss curves

V. Conclusion

To address the issues of semantic confusion, over-smoothing, and gradient instability encountered in disease diagnosis using EMRs within heterogeneous graph neural networks, we propose a novel disease diagnosis model, MPEA4DD. By employing a message passing mechanism with node state selection and a Gumbel-Softmax differentiable sampling strategy, along with an edge-dynamic attention mechanism, the proposed model dynamically filters neighborhood information, effectively reducing the interference of irrelevant neighbors on the model. Building upon GATv2 attention aggregation, a dynamic integration of contextual semantic fusion and residual connection strategies effectively alleviates the issues of gradient vanishing and over-smoothing in deep networks. Finally, on the MIMIC-III dataset, MPEA4DD improved Micro-F1 and Macro-F1 by 1.13% and 2.12%, respectively, compared to the best baseline model. On the MIMIC-IV dataset, it achieved improvements of 1.03% and 1.08%, respectively, which demonstrates that our model significantly outperforms conventional baseline models in the disease diagnosis task.

References

[1] Johnson A E W, Pollard T J, Shen L, et al. "MIMIC-III, a freely accessible critical care database," Scientific Data, vol. 3, no. 1, pp. 1-9, 2016.

[2] Deo R C, et al. "Machine Learning in Medicine," Circulation, vol. 132, no. 20, pp. 1920-1930, 2015.

[3] Esteva A, Robicquet A, Ramsundar B, et al. "A guide to deep learning in healthcare," Nature Medicine, vol. 25, no. 1, pp. 24-29, 2019.

[4] Wu Z, Pan S, Chen F, et al., "A Comprehensive Survey on Graph Neural Networks," IEEE Transactions on Neural Networks and Learning Systems, vol. 32, no. 1, pp. 4-24, 2020.

[5] Zheng X, Wang Y, Liu Y, et al. “Graph Neural Networks for Graphs with Heterophily: A Survey,” ArXiv: Machine Learning, 2022. Available: https://arxiv.org/abs/2202.07082
[6] Shang J, Ma T, Xiao C, et al. “Pre-training of Graph Augmented Transformers for Medication Recommendation,” ArXiv: Machine Learning, 2019. Available: https://arxiv.org/abs/1906.00346
[7] Chen C, Ma W, Zhang M, et al. "Graph Heterogeneous Multi-relational Recommendation," in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, no. 5, pp. 3958–3966, 2021.
[8] Yun S, Jeong M, Yoo S, et al. “Heterogeneous Graph Attention network”, in Proceedings of World Wide Web Conference, pp. 2022-2032,2019.
[9] Hu Z, Dong Y, Wang K, et al. "Heterogeneous Graph Transformer," in Proceedings of The Web Conference 2020 (WWW), pp. 2704–2710, 2020.
[10] Chen J, Ma T, Xiao C. “FastGCN: Fast Learning with Graph Convolutional Networks via Importance Sampling,” ArXiv: Machine Learning, 2018. Available: https://arxiv.org/abs/1801.10247
[11] Zhou Z, Shi J, Yang R, et al., "SlotGAT: slot-based message passing for heterogeneous graphs," in Proceedings of the 40th International Conference on Machine Learning (ICML), pp. 42644–42657, 2023.
[12] Arroyo Á, Gravina A, Gutteridge B, et al. “On Vanishing Gradients, Over-Smoothing, and Over-Squashing in GNNs: Bridging Recurrent and Graph Learning,” ArXiv: Machine Learning, 2025. Available: https://arxiv.org/abs/2502.10818
[13] Xu K, Hu W, Leskovec J, et al. “How Powerful Are Graph Neural Networks?” ArXiv: Machine Learning, 2018. Available: https://arxiv.org/abs/1810.00826
[14] Chen D, Lin Y, Li W, et al. “Measuring and relieving the over-smoothing problem for graph neural networks from the topological view”, in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, no. 04, pp. 3438-3445, 2020.
[15] Wu N, Wang C. “Heterogeneous Graph Tree Networks,” ArXiv: Machine Learning, 2022. Available: https://arxiv.org/abs/2209.00610.
[16] Thanapalasingam T, van Berkel L, Bloem P, et al., "Relational graph convolutional networks: a closer look," PeerJ Computer Science, vol. 8, pp. e1073, 2022
[17] Gilmer J, Schoenholz S S, Riley P F, et al. “Neural message passing for quantum chemistry,” in Proceedings of the 34th International Conference on Machine Learning (ICML), vol. 70, pp. 1263–1272, 2017.
[18] Kipf T N, Welling M. “Semi-Supervised Classification with Graph Convolutional Networks,” ArXiv: Machine Learning, 2016. Available: https://arxiv.org/abs/1609.02907
[19] Hamilton W, Ying Z, Leskovec J., "Inductive representation learning on large graphs," in Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS), vol. 30, pp. 1025–1035, 2017.
[20] Finkelshtein B, Huang X, Bronstein M, et al. “Cooperative Graph Neural Networks,” ArXiv: Machine Learning, 2023. Available: https://arxiv.org/abs/2310.01267
[21] Chaudhari S, Mithal V, Polatkan G, et al., "An attentive survey of attention models," ACM Transactions on Intelligent Systems and Technology (TIST), vol. 12, no. 5, pp. 1–32, 2021.
[22] Veličković P, Cucurull G, Casanova A, et al. “Graph Attention Networks,” ArXiv: Machine Learning, 2017. Available: https://arxiv.org/abs/1710.10903
[23] Brody S, Alon U, Yahav E. “How Attentive Are Graph Attention Networks?,” ArXiv: Machine Learning, 2021. Available: https://arxiv.org/abs/2105.14491
[24] Dwivedi V P, Bresson X. “A Generalization of Transformer Networks to Graphs,” ArXiv: Machine Learning, 2020. Available: https://arxiv.org/abs/2012.09699
[25] Yuan J, Lu S, Duan P, et al., "AGHINT: Attribute-guided representation learning on heterogeneous information networks with transformer," Knowledge-Based Systems, vol. 310, pp. 112977, 2025.
[26] Jianbin Luo, Dan Yang, Yang Liu, and Jiaming Liang, "Medical Heterogeneous Graph Transformer for Disease Diagnosis," Engineering Letters, vol. 32, no. 12, pp. 2290–2298, 2024.
[27] Chengyu Yang, Dan Yang, and Xi Gong, "Disease Diagnosis Based on Heterogeneous Graph Contrastive Learning," Engineering Letters, vol. 32, no. 12, pp. 2200-2209, 2024.
[28] Zhang C, Geng T, Guo A, et al., "H-GCN: A graph convolutional network accelerator on Versal ACAP architecture," in Proceedings of the 32nd International Conference on Field-Programmable Logic and Applications (FPL), pp. 200–208, 2022.
[29] Jang E, Gu S, Poole B. “Categorical Reparameterization with Gumbel-Softmax,” ArXiv: Learning, 2016. Available: https://arxiv.org/abs/1611.01144
[30] Zhu, Qiuyu, L. Zhang, et al., Q. Xu, et al., "HHGT: Hierarchical Heterogeneous Graph Transformer for Heterogeneous Graph Representation Learning," in Proceedings of the Eighteenth ACM International Conference on Web Search and Data Mining(WSDM), pp. 318–326, 2025.
[31] Van der Maaten, Laurens, and Geoffrey Hinton. "Visualizing data using t-SNE," Journal of machine learning research, vol. 9, no. 11, pp. 2579−2605, 2008.