IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 18 May 2024, accepted 6 June 2024, date of publication 11 June 2024, date of current version 5 July 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3412419

RESEARCH ARTICLE

Knowledge-Routed Automatic Diagnosis With Heterogeneous Patient-Oriented Graph

ZHIANG LI AND TONG RUAN
School of Information Science and Engineering, East China University of Science and Technology, Shanghai 200237, China
Corresponding author: Tong Ruan (ruantong@ecust.edu.cn)

This work was supported in part by the National Key Research and Development Program of China under Grant 2021YFC270 1800 and Grant 2021YFC2701801, and in part by Shanghai Municipal Special Fund for Promoting High-Quality Development of Industries under Grant 2021-GZL-RGZN-01018.

ABSTRACT Automated diagnosis, as a temporary medical supplement, has gained significant attention in research in recent years. Existing methods employ sequence generation approaches to inquire about symptoms and diagnose diseases. However, these methods ignore the fact that: 1) doctors utilize their past experience and similar cases to aid in diagnosis in real-world scenarios; 2) doctors inquire about key symptoms that serve as vital diagnostic evidence within limited conversations. To address these issues, we propose an end-to-end model KDPoG. Firstly, in addition to use the symptom and attribute embedding, we propose patient-oriented graph enhanced representation learning, which is built by a patient-oriented graph and learned with heterogeneous graph convolution networks. Furthermore, based on the encoder built with gated attention units, we propose knowledge-guided attention mechanism learning, which incorporates conditional probabilities of co-occurrence between symptom pairs. Finally, we utilize two linear layers as the classification module to achieve symptom probing and disease diagnosis. We conduct extensive experiments on four public datasets, which demonstrate that our proposed model outperforms the state-of-the-art methods. We achieve an average absolute improvement of over 2% in disease diagnosis accuracy. Particularly, on the Muzhi-10 dataset, we observe an absolute improvement of over 14.7% in symptom recall rate.

INDEX TERMS Automatic diagnosis, prior knowledge, heterogeneous patient-oriented graph, sequence generation, topological connection.

I. INTRODUCTION

Automatic diagnosis is commonly appeared in medical dialogue systems, which aims to simulate the process of online consultations between doctors and patients. A specific example is shown in Figure 1. The patient first provides a self-report to the doctor, and then the doctor asks about additional symptoms that the patient has not yet mentioned. Finally, the doctor makes a disease diagnosis based on all confirmed symptoms. So the automatic diagnosis is essentially a task which aims to inquiry implicit symptoms in limited turns and finally makes a correct disease diagnosis. Since it has the potential in simplifying diagnostic process and serving as medical supplement, it has received increasing attention in recent years [1], [2], [3], [4], leading to the emergence of numerous notable works.

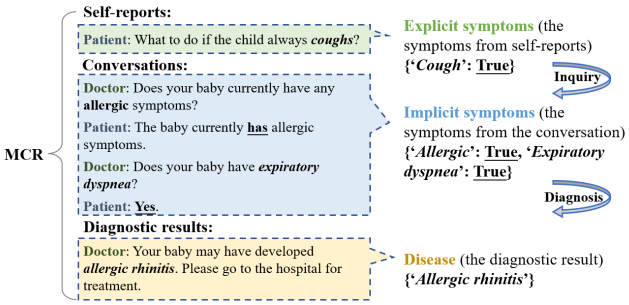

FIGURE 1. An example of the automatic diagnosis task, where MCR stands for the medical consultation records.

Traditionally, methods for solving this task can be classified into two categories: those based on statistical models

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

and those based on reinforcement learning (RL) models. Firstly, among the previous works based on statistical models, Bayesian-based models are particularly prominent, known for their low complexity [5], [6]. They define symptom probing as a feature selection task, using entropy functions to identify optimal features and maximize information gain as the training objective. However, these models have limitations in terms of their relatively low accuracy and diminished effectiveness when applied to large-scale data. Considering the aforementioned problems, the majority of subsequent work has shifted towards RL-based models [2], [7], [8]. They treat symptom probing as a multi-step decision-making task, modeling it as a Markov Decision Process (MDP) and utilizing RL for policy learning. However, RL-based models encounter challenges in terms of their low efficiency and the difficulty of defining an appropriate task-specific reward function, which greatly affects model training.

Currently, the mainstream approach for this task is to utilize sequence generation-based models for implementation, which avoid the aforementioned challenges in RL-based models. They define symptom probing as the task of generating a sequence of symptoms, and construct an end-to-end auto-regressive model based on Transformer architecture [9]. This model is equipped with a specially designed mask matrix, as adopted in UniLM [10], [11], [12]. Due to their excellent performance on this task, we choose to follow this approach. But there are two limitations that the previous works have not taken both of them into account: 1) During realistic clinical diagnosis, the doctor typically relies on symptoms of the patient as the primary consideration. However, they also draw upon their past experiences and knowledge of similar patients to aid in their diagnosis [13]. Therefore, it's necessary to utilize information from similar patients and consider how to model such relations. 2) During online medical consultations, although it is essential to gather adequate symptoms from the patients, they often have limited patience to answer each question individually. Therefore, the model needs to prioritize asking about more critical questions within a limited number of conversations, focusing on key symptoms that serve as crucial diagnostic evidence.

To alleviate the above problems, we propose **K**nowledge-routed automatic **D**iagnosis with heterogeneous **P**atient-**o**riented **G**raph (KDPoG), which is an end-to-end model. To address the first issue, in addition to symptom and attribute embedding, we also introduce patient-oriented graph enhanced representation learning. The reason for this considering is to model the topological connections among patients and incorporate experiential knowledge from similar patients into the model. Specifically, to obtain the representation, we first construct a patient-oriented graph, and then we apply heterogeneous graph convolution operations on all types of relations. To address the second issue, we use several stacked Gated Attention Unit (GAU) blocks as the backbone of our encoder. Additionally, we propose knowledge-guided attention mechanism learning that incorporates the conditional probabilities of co-occurrence between symptom pairs. This aims to route the model to inquiry high-frequency and highly-correlated symptoms. Finally, we utilize a classification module consisting of two linear layers for symptom inquiry and disease diagnosis, respectively.

The main contributions of this paper can be summarized as follows:

- We propose the patient-oriented graph enhanced representation learning, which aims to enable the model to mimic doctors in utilizing their past experience and knowledge from similar patients to aid in diagnosis;
- We propose the knowledge-routed encoder, which aims to prioritize asking about key symptoms that serve as vital diagnostic evidence within a limited number of conversations;
- Experimental results on four public datasets show that our model outperforms SoTA methods. Particularly, on the Muzhi-10 dataset, we observe an absolute improvement of over 14.7% in symptom recall rate.

II. METHOD OVERVIEW

In this section, we first introduce some basic definitions related to automatic diagnosis. Then we formulate the task definition and outline the framework of our model.

A. PRELIMINARY

Definition 1: Explicit/Implicit Symptoms Let $S = \{s_i | i = 1, 2, \ldots, n_s\}$ denotes the set of all possible symptoms. The set of symptoms extracted from the self-report in medical consultation record (MCR) [14] are defined as explicit symptoms $S_{exp} = \{s_1^{exp}, s_2^{exp}, \ldots, s_m^{exp}\}$ while the others in conversations are defined as implicit symptoms $S_{imp} = \{s_1^{imp}, s_2^{imp}, \ldots, s_n^{imp}\}$, where $(S_{exp} \cup S_{imp}) \subseteq S$, and $S_{exp} \cap S_{imp} = \emptyset$.

Definition 2: Symptom Attribute The symptom attribute is defined as the relation between a certain symptom and the patient. Let $A = \{a_j | j = 1, 2, \ldots, n_a\}$ denotes the set of all possible symptom attributes. Typically, A includes three attributes: 1) True: patient has the symptom, 2) False: patient does not have the symptom, 3) Unknown: the symptom is not mentioned in the dialogue. For the attributes of S_{exp} and S_{imp}, we respectively use $A_{exp} = \{a_1^{exp}, a_2^{exp}, \ldots, a_m^{exp}\}$ and $A_{imp} = \{a_1^{imp}, a_2^{imp}, \ldots, a_n^{imp}\}$ to represent them.

Definition 3: Structured MCR Let $D = \{d_k | k = 1, 2, \ldots, n_d\}$ denotes the set of all possible diseases. A structured MCR then can be denoted as: $(\{(s_i^{exp}, a_i^{exp})\}_{i=1}^{m}, \{(s_j^{imp}, a_j^{imp})\}_{j=1}^{n}, d_k)$, where m is the number of explicit symptoms, n is the number of implicit symptoms, and d_k is the disease diagnosis made by the doctor.

B. TASK DEFINITION

Given a structured MCR of a patient, based on explicit symptoms S_{exp} and their attributes A_{exp}, we aim to inquire about as many implicit symptoms S_{imp} as possible, within the fewest number of dialogue turns. Finally, we provide a

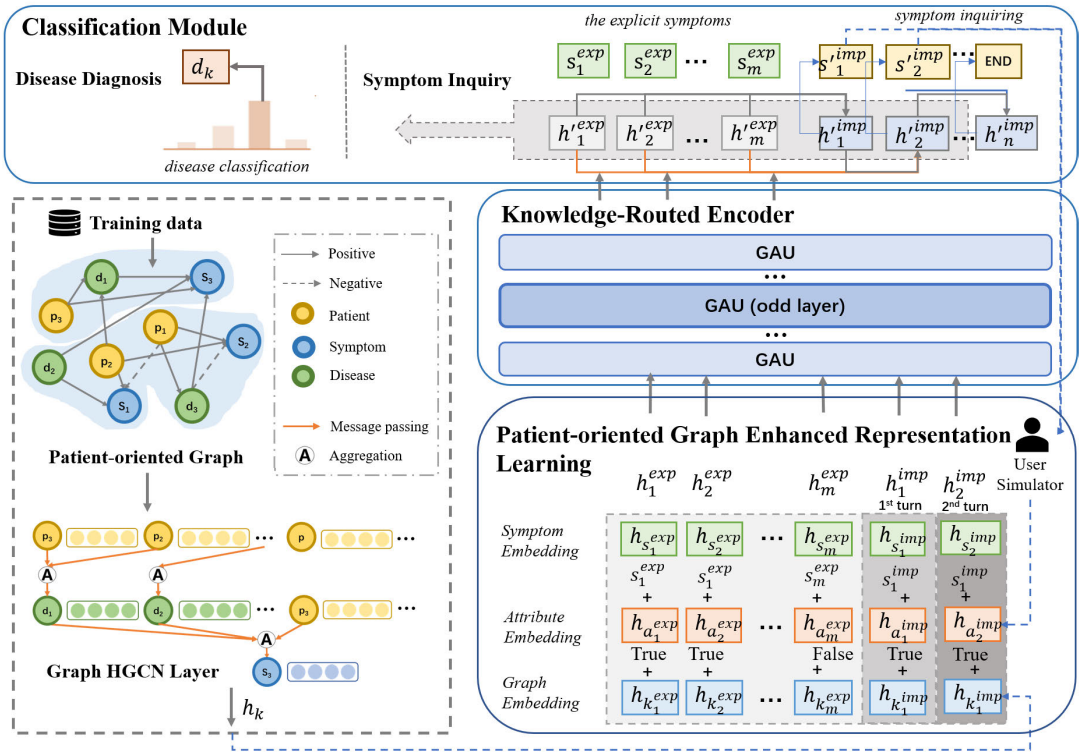

FIGURE 2. Illustration of our model structure, which includes patient-oriented graph enhanced representation learning, knowledge-routed encoder and classification module.

diagnosis for disease d_k based on all confirmed symptoms and their attributes $(\{(s_i^{exp}, a_i^{exp})\}_{i=1}^{m}, \{(s_j^{inq}, a_j^{inq})\}_{j=1}^{n_q})$, where the inquired symptom $s_n^{inq} \in S_{imp}$ and n_q is the number of inquired symptoms.

C. FRAMEWORK

As shown in Figure 2, our solution for this task is mainly divided into three stages. Firstly, given S_{exp} and A_{exp}, we utilize **patient-oriented graph enhanced representation learning** to obtain fused encoding representations that represent explicit symptom information of the patient. Then, based on the fused representations, we obtain hidden representations through **knowledge-routed encoder**. Finally, based on the hidden representations, we predict implicit symptoms and diseases through **classification module**.

The specific content of each stage is as follows: 1) **Patient-oriented Graph Enhanced Representation Learning.** We first construct the patient graph based on entities in training data (i.e., patients, symptoms, diseases and attributes). Then, we apply Heterogeneous Graph Convolutional Network (HGCN) convolution operations on all types of relations, aiming to incorporate patient and disease nodes' information into symptom nodes. Finally, we combine symptom embedding, attribute embedding, and graph embedding of each symptom, serving as the final input. 2) **Knowledge-Routed Encoder.** Given the input above, we stack several GAU blocks serving as encoder in our model. We innovatively propose knowledge-guided attention mechanism learning incorporating a conditional probability matrix, aiming to perform secondary processing and constraints on the prediction results of our model. 3) **Classification Module.** Our model incorporates two linear layers as the output layers. The first linear layer is responsible for symptom classification, aiming to minimize the number of inquiries required to identify all implicit symptoms. The model continues decoding until it encounters a termination symbol, signaling the diagnosis of a disease. At this point, the second linear layer is invoked for disease classification.

III. METHODOLOGY

In this section, we introduce the main architecture of our model, including patient-oriented graph enhanced representation learning, knowledge-routed encoder and classification module. Then, we discuss the training and inference procedures of our model.

A. PATIENT-ORIENTED GRAPH ENHANCED REPRESENTATION LEARNING

This part is the input of our model, which consists of symptom embedding, attribute embedding, and graph embedding. First, we obtain symptom and attribute embedding based on the given S_{exp} and A_{exp}. Then, we build a patient-oriented graph based on entities in training data, and utilize HGCN to embed the graph to obtain graph embedding of symptoms. Finally,

we combine symptom embedding, attribute embedding, and graph embedding of each symptom, serving as the final input.

Thus, we first discuss symptom and attribute embedding. Then, we introduce the graph embedding, which involves two steps: 1) constructing a patient-oriented graph, and 2) designing a graph HGCN layer to obtain node embedding for symptoms. Finally, we discuss the representation combination of all three embedding.

1) SYMPTOM AND ATTRIBUTE EMBEDDING

Based on input S_{exp} and A_{exp}, we first construct a special tokenizer (SymTokenizer) to encode input. Then, based on the encoded sequence of input for the given patient, we utilize linear layers for embedding.

Thus, before discussing the embedding of symptoms and attributes, let's first introduce SymTokenizer. Since we inquire a certain symptom in each turn, we treat each symptom as a token rather than each individual character. Therefore, we construct our SymTokenizer in advance, which includes symptoms S, symptom attributes A, diseases D, and some special tokens $C_{special}$.

$$SymTokenizer = S \cup A \cup D \cup C_{special}, \tag{1}$$

$$C_{special} = \{\langle \text{SEP} \rangle, \langle \text{PAD} \rangle, \langle \text{DIS} \rangle, \ldots\}, \tag{2}$$

where $\langle$SEP$\rangle$ for separation, $\langle$DIS$\rangle$ for disease diagnosis, and $\langle$PAD$\rangle$ for padding. Then, we utilize linear layers to embed the symptoms and attributes.

Formally, given the input sequence $\{(s_1^{exp}, a_1^{exp}) \ldots, (s_m^{exp}, a_m^{exp}), \ldots, (s_1^{imp}, a_1^{imp}), \ldots, (s_n^{imp}, a_n^{imp})\}$, we obtain $\boldsymbol{h}_{s_m^{exp}} = \boldsymbol{W}_s s_m^{exp}$ and $\boldsymbol{h}_{a_m^{exp}} = \boldsymbol{W}_a a_m^{exp}$, as well as $\boldsymbol{h}_{s_n^{imp}}$ and $\boldsymbol{h}_{a_n^{imp}}$, respectively. $\boldsymbol{W}_s$ and $\boldsymbol{W}_a$ are weights of linear layers for symptom embedding and attribute embedding.

2) GRAPH EMBEDDING

First, based on the symptoms and diseases of each patient in the training set, we construct a *patient-oriented graph*. Then, utilizing the graph, we build a *graph HGCN layer* for graph representation learning, taking the representation of symptom nodes as the graph embedding.

a: PATIENT-ORIENTED GRAPH

Due to the involvement of graph embedding in the model training and to prevent leakage of information from the test set, we only utilize data from the training set to construct the graph. Firstly, for each patient, we create a new patient node. Then, based on the symptoms and disease entities in the data, we create symptom and disease nodes. Next, we establish edges between the patient node, symptom nodes, and disease nodes based on the attributes of symptoms and the diagnostic results of diseases. Particularly, all edges are unidirectional. Thus, for each data instance, we construct a patient sub-graph with the patient node as the root node, as shown in Figure 2 (where the blue region represents a patient sub-graph). Finally, we integrate all patient sub-graphs to form a patient-oriented graph, where identical entities and edges are merged into the same node or edge, respectively. Specifically, we add isolated entities as isolated nodes to the graph.

Formally, the graph can be denoted as $G = (g_i) = \{g_i | g_i = (p_i, s_{i,j}, d_{i,k})\}$. The nodes include patient node ($P = \{p_i | i = 1, 2, \ldots, M\}$), symptom node ($S = \{s_j | j = 1, 2, \ldots, N\}$), and disease node ($D = \{d_k | k = 1, 2, \ldots, O\}$). The relations include two types: positive one and negative one. In positive one, there are three types of relations: 1) true: the patient p_i has the symptom s_j, 2) related: the disease d_k appears together with the symptom s_j, 3) diagnosis: the patient p_i is diagnosed the disease d_k. In negative one, there are two types of relations: 1) false: the patient p_i hasn't the symptom s_j, 2) unrelated: the disease d_k doesn't appear together with the symptom s_j. These types of nodes and relations represent main interactions in training dataset. As our graph contains three types of nodes and five types of relations, making it a heterogeneous graph.

b: GRAPH HGCN LAYER

Given the heterogeneous patient graph constructed above, our goal is to obtain representations for each symptom node. As the graph is heterogeneous, we utilize HGCN for graph embedding learning. The key idea is to simultaneously perform graph convolution operations across all five types of relations and ultimately aggregate all node representations into the symptom nodes.

Formally, the initial representations of the three types of nodes $\boldsymbol{h}_{init}$ are encoded using three embedding layers. Each embedding layer consists of a linear layer weight $\boldsymbol{W}_{graph} \in \mathbb{R}^{N_{type} \times D_{init}}$, where N_{type} is the number of nodes types and D_{init} is the dimension of node representation. Then, for each node, graph convolution is applied using GCN, and the formula is as follows:

$$\boldsymbol{h}_i^{(l+1)} = \sigma(\boldsymbol{b}^l + \sum_{j \in N(i)} (\frac{1}{c_{ji}} \boldsymbol{h}_j^l \boldsymbol{W}^l), \tag{3}$$

where $N(i)$ is the neighbors set of node i, $c_{ji} = \sqrt{N(j)}\sqrt{N(i)}$, σ is ReLU activation function, $\boldsymbol{b}$ and $\boldsymbol{W}$ are the parameters. Then, the nodes with different relations are aggregated into this node, and the formula is as follows:

$$\boldsymbol{h}_i^{(l+1)} = AGG_{r \in N_i}(f_r(g_r, \boldsymbol{h}_{r_j}^l, \boldsymbol{h}_{r_i}^l)), \tag{4}$$

where AGG is the summation function, f_r is the GCN function, g_r is the sub-graph which only consists of relation r, N_i is the relations between node i and its neighbour nodes.

Finally, the representations of the patients $\boldsymbol{h}_p^l$ and the diseases $\boldsymbol{h}_d^l$ will both be constructed into the symptom node, namely $\boldsymbol{h}_s^l = AGG(\boldsymbol{h}_p^l, \boldsymbol{h}_d^l)$.

3) REPRESENTATION COMBINATION

After obtaining the symptom, attribute, and graph embedding, we perform a weighted summation of these three embedding into a single vector to serve as the input for our model. Note that since the order of explicit symptoms makes no sense and diseases should not be sensitive to the order

of symptoms, we remove the position embedding, which is different from previous works [9].

Formally, the input representation is computed by summing the symptom embedding, attribute embedding and graph embedding, which is as follows:

$$\boldsymbol{h}_m^{exp} = \alpha * (\boldsymbol{h}_{s_m}^{exp} + \boldsymbol{h}_{a_m}^{exp}) + (1-\alpha) * \boldsymbol{h}_{k_m}^{exp}, \tag{5}$$

where $\boldsymbol{h}_{k_m}^{exp}$ is the graph embedding of explicit symptoms. $\boldsymbol{h}_{k_m}^{imp}$ is defined the same as $\boldsymbol{h}_{k_m}^{exp}$. α is a balanced parameter.

B. KNOWLEDGE-ROUTED ENCODER

Given the above combined representations, our goal is to learn their hidden states. To achieve this goal, we design a knowledge-routed encoder. The key idea is as follows: 1) We first use multiple stacked GAU blocks as the backbone of our encoder. 2) Then, we design prior knowledge-guided attention mechanism learning by combining a conditional probability matrix with the raw self-attention in GAU, which aims to route the model to inquiry high-frequency and highly-correlated symptoms, as shown in Figure 3. 3) Finally, we utilize the mask matrix to implement sequence generation, as shown in Figure 4.

1) GAU

Firstly, we introduce GAU. GAU uses Gated Linear Unit (GLU) to replace feed-forward network (FFN) in Transformer [9], and combines attention and GLU as a unified layer, aiming to maximize the sharing of their computations. Given that GAU can achieve comparable performance to multi-attention heads of Transformer using just one attention head, and considering our need for modifying the attention matrix, we choose GAU as our preferred approach.

GAU consists of three parts. Specifically, given the representation of an input sequence $\boldsymbol{h} = (\boldsymbol{h}_1^{exp}, \boldsymbol{h}_2^{exp}, \ldots, \boldsymbol{h}_m^{exp}, \boldsymbol{h}_1^{imp}, \ldots, \boldsymbol{h}_n^{imp})$, we first utilize Layer Normalization, the OffsetScale layer, and linear layers to respectively obtain $\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}, \boldsymbol{h}_{gate}$ as follows:

$$\boldsymbol{V}, \boldsymbol{h}_{gate} = \phi_{SiLU}(\boldsymbol{h}_{norm}\boldsymbol{W}_v), \tag{6}$$

$$\boldsymbol{Q}, \boldsymbol{K} = OffsetScale(\phi_{SiLU}(\boldsymbol{h}_{norm}\boldsymbol{W}_{qk})), \tag{7}$$

where $\boldsymbol{h}_{norm}$ is the normalized representation, and $\boldsymbol{W}_{qk}$ and $\boldsymbol{W}_v$ are the weights of linear layers. Secondly, we obtain the attention matrix $\boldsymbol{M}_{atten}$ by performing dot product between $\boldsymbol{Q}$ and $\boldsymbol{K}$, as shown in the equation given by $\boldsymbol{M}_{atten} = \boldsymbol{Q}\boldsymbol{K}^T/d_k$, where d_k is the length of the input sequence. Finally, we multiply $\boldsymbol{V}$, $\boldsymbol{M}_{atten}$, and $\boldsymbol{h}_{gate}$ together as the output.

2) PRIOR KNOWLEDGE-GUIDED ATTENTION MECHANISM LEARNING

Given the above attention matrix $\boldsymbol{M}_{atten}$, our goal is to use prior knowledge to guide the learning of the attention mechanism, so that high-frequency and highly-correlated symptom pairs have higher attention scores. To achieve this goal, we initially construct a probability matrix where each element represents the conditional probability between two corresponding symptoms. Then, we utilize the conditional probability as prior knowledge to guide the learning of attention mechanism. The learning method is to integrate the probability matrix with attention matrix.

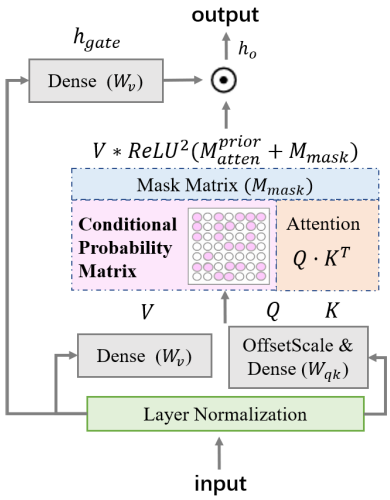

FIGURE 3. GAU block with prior knowledge-guided attention mechanism learning.

Formally, given the symptoms set S, we initialize a matrix $\boldsymbol{M} = (m_{i,j})$, where the size of S is s and the size of the matrix is $s * s$. The element $m_{i,j}$ in $\boldsymbol{M}$ represent the probability of symptom $s_j$occurring when symptom s_i is present:

$$m_{i,j} = P(s_j|s_i) = \frac{n(s_i, s_j)}{\sum_{k=1}^{s}(s_k, s_i)} = m_{j,i}, \tag{8}$$

where $n(s_i, s_j)$ is the number of patients who both have symptom s_j and s_i. Since certain symptoms in the test set may not be present in the training set, we initialize the matrix with diagonal elements $m_{i,i}$ all set to 1, which helps avoid $\sum_{k=1}^{s}(s_k, s_i)$ being equal to 0. In order to alleviate the differences in the frequency of symptoms, we apply a softmax layer to normalize the matrix.

$$\boldsymbol{M}_s = Softmax(\boldsymbol{M}) + Softmax(\boldsymbol{M})^T, \tag{9}$$

Given the conditional probability matrix $\boldsymbol{M}_s$, we take into account that each position in the attention matrix represents the correlation between two symptoms. Therefore, we add the values from the conditional probability matrix to the corresponding positions in the attention matrix. Then, we obtain the prior knowledge attention matrix $\boldsymbol{M}_{atten}^{prior}$ as follows:

$$\boldsymbol{M}_{atten}^{prior} = \beta * \boldsymbol{M}_{atten} + (1-\beta) * k * \boldsymbol{M}_s, \tag{10}$$

where β is used to balance attention matrix and conditional probability matrix. In particular, we just modify $\boldsymbol{M}_{atten}$ at odd layers instead of all layers. When the number of the layer is even, $k = 0$, otherwise $k = 1$. The experimental section will provide a detailed analysis of the reasons for setting it up this way.

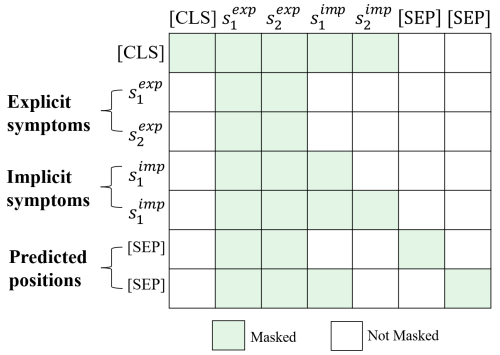

FIGURE 4. Mask matrix.

3) MASK MATRIX

Finally, given the prior knowledge attention matrix $\boldsymbol{M}_{atten}^{prior}$, our goal is to use a mask matrix to implement sequence generation, as shown in Figure 4. The key idea is to mask out symptoms that should not be attended to, which prevents information leakage. Formally, given the raw input sequence as follows:

$$\boldsymbol{X} = (s_1^{exp}, \ldots, s_i^{exp}, \ldots, s_1^{imp}, \ldots, s_j^{imp}, s_{j+1}^{imp}, \ldots). \quad (11)$$

Then, for s_i^{exp}, it can attend all explicit symptoms S_{exp}. For s_j^{imp}, it can also attend all S_{exp}, while it can only see itself and the implicit symptoms have been inquired. Therefore, the construction of the mask matrix $\boldsymbol{M}_{mask}$ is illustrated as follows:

$$\boldsymbol{M}_{mask}(s) = \begin{cases} (1, .., 1, \ldots, -\infty, \ldots, -\infty, -\infty, \ldots), & \text{if } s \in S_{exp} \\ (1, \ldots 1, \ldots 1, \ldots 1, -\infty, \ldots). & \text{if } s \in S_{imp} \end{cases} \quad (12)$$

Finally, the entire output $\boldsymbol{h}_o$ is illustrated as follows:

$$\boldsymbol{h}_o = \boldsymbol{W}_o(\boldsymbol{V}\phi_{ReLU}^2(\boldsymbol{M}_{atten}^{prior} + \boldsymbol{M}_{mask}) * \boldsymbol{h}_{gate}), \quad (13)$$

where $\boldsymbol{W}_o$ is the weights of a linear layer.

C. CLASSIFICATION MODULE

Given the hidden state generated by encoder, our goal is to decide the next implicit symptom to inquire about and make a disease diagnosis. To achieve this goal, we employ two linear layers as the classification module for prediction, respectively. The first linear layer is responsible for symptom inquiring, aiming to minimize the number of inquiries required to identify all implicit symptoms. The second linear layer is invoked for disease diagnosis.

Formally, given the hidden state generated by encoder $\boldsymbol{h}' = (\boldsymbol{h}'^{exp}_1, \boldsymbol{h}'^{exp}_2, \ldots, \boldsymbol{h}'^{exp}_m, \boldsymbol{h}'^{imp}_1, \ldots, \boldsymbol{h}'^{imp}_n)$, we insert several ⟨SEP⟩ tokens after it, which the number of ⟨SEP⟩ tokens is $n + 1$. The j-th ⟨SEP⟩ is used to decide the j-th symptom. And the last ⟨SEP⟩ is used to decide ⟨DIS⟩ token, which indicates the ending of inquiring. For symptom inquiring, the formula is as follows:

$$z_{s_j} = \boldsymbol{W}_{sc}(\ldots \boldsymbol{M}_{atten}^{new}(\boldsymbol{K}, \boldsymbol{V} = (\ldots, \boldsymbol{h}'^{exp}_m, \ldots, \boldsymbol{h}'^{imp}_{j-1}))\ldots), \quad (14)$$

where $\boldsymbol{W}_{sc}$ is weights of symptom classification layer, and z_{s_j} is the output of this layer. Note that when predicting the symptom at position j, only the explicit symptoms and the preceding $j-1$ implicit symptoms are attended. We use the cross entropy loss of layer output z_{s_j} as the symptom inquiry loss $loss_{sym}$. For disease diagnosis, the formula is as follows:

$$z_{d_k} = \boldsymbol{W}_{dc}(\ldots \boldsymbol{M}_{atten}^{new}(\boldsymbol{K}, \boldsymbol{V} = (\ldots, \boldsymbol{h}'^{exp}_m, \ldots, \boldsymbol{h}'^{imp}_n))\ldots), \quad (15)$$

where $\boldsymbol{W}_{dc}$ is weights of disease classification layer, z_{d_k} is the output of this layer. Note that when predicting the disease d_k, all explicit and implicit symptoms are attended. We use the cross entropy loss of layer output z_{d_k} as the disease diagnosis loss $loss_{dis}$.

IV. TRAINING AND INFERENCE

A. TRAINING

Since we define the task as a sequence generation task, our object is to inquiry about as many symptoms s_j^{inq} belonged to S_{imp} as possible within a limited number of dialogue turns, and to provide a correct diagnosis of the disease d_k. Besides, the more implicit symptoms are queried, the higher the diagnostic accuracy is. Therefore, our training objective is to maximize the probability of the following formula:

$$\sum_{k=1}^{N}\prod_{j=1}^{M}(P(s_j^{inq}|S_{exp}, S_{inq})P(d_k|S_{exp} \cup S_{inq})), \quad (16)$$

where $s_j^{inq} \in S_{inq} \subseteq S_{imp}$, and N and M respectively indicate the number of patients and unconfirmed symptoms. And we convert the optimization objective into minimizing the loss function in the following formula:

$$Loss_{total} = loss_{sym} + loss_{dis} \quad (17)$$

where $loss_{sym}$ is the symptom inquiry loss, and $loss_{dis}$ is the disease diagnosis loss. Note that during each training epoch, we initialize the graph embedding rather than continuing from the previous epoch. Besides, the graph embedding is trained together with the rest of the model.

B. INFERENCE

After training, the model starts with the symptom inquiry process. Firstly, it obtains the probability distribution of symptoms based on the given explicit symptoms. Then, it selects the symptom with the highest probability for inquiry. And the user simulator will decide whether the symptom belongs to the set of implicit symptoms. If the symptom is an implicit symptom, we add this symptom to

TABLE 1. Details of the four datasets. # Sample represents the number of conversations. # Disease and # Symptom represent the number of disease and symptom categories, respectively. # Sym/Patient, # Exp/Patient, and # Imp/Patient represent the average number of symptoms/maximum number of symptoms, explicit symptoms, and implicit symptoms.

Dataset	Dxy	Muzhi-4	Muzhi-10	GMD-12
# Sample	527	710	4116	2374
# Disease	5	4	10	12
# Symptom	41	66	331	118
# Sym/Patient	4.7/10	5.6/14	6/21	5.5/16
# Exp/Patient	3.1/7	2.3/12	2/12	2.9/13
# Imp/Patient	1.7/6	3.3/12	4.1/19	2.5/8

the current symptom set. Otherwise, we continue to inquiry another symptom of next highest probability. The inquiry process continues until the model predicts the ⟨DIS⟩ token, which indicates the start of the diagnosis phase. We set three thresholds to prevent the model from continuously inquiring symptoms. The first one is the maximum number of inquiry turns allowed. If the number of inquiry turns reaches this limit, the model will stop inquiring and directly start to diagnosis disease. The second one is the maximum predicted probability for the ⟨DIS⟩ token. If the predicted probability for the ⟨DIS⟩ token exceeds this threshold, the model proceeds with the disease diagnosis phase. The third one is the minimum probability for an inquired symptom. If the probability of an inquired symptom falls below this threshold, the model stops inquiring and proceeds with the disease diagnosis phase.

V. EXPERIMENT

In this section, we first introduce the experiment setup. Then, we compare our model with several baselines on four public datasets. Finally, we discuss some detailed analysis.

A. EXPERIMENT SETUP

1) DATASETS

In our experiments, we evaluate our proposed model on a total of four real-world datasets. The real-world datasets include Muzhi-4 [1], DXY [15], GMD-12 [16] and Muzhi-10 [17], all of which are annotated from real medical dialogues. To provide a better understanding of these datasets, we present the statistics information about them in Table 1.

2) BASELINES

To establish a fair comparision with other models, we select several competitive baselines, including Flat-DQN [1], REFUEL [18], KR-DS [15], GAMP [19], HRL [20], PPO [21], MMF-AC [22], BSODA [23], Diaformer [10], DxFormer [24], CoAD [25], and Tian et al. [26].

3) EVALUATION METRICS

To evaluate the performance of our model, we adopt the commonly used evaluation metrics in previous studies [1], [20], including the accuracy of diseases (**DAcc**), the recall of implicit symptoms (**SRec**), and the average inquiry turn (**ATurn**). Specifically, we aim for higher accuracy and recall, and smaller average turn to indicate better model performance.

4) HYPER-PARAMETERS

In the embedding, the dimension of output d_e is set to 512. The dimension of intermediate representation of HGCN d_c is set to 100. Since the graph is relatively small, we set the number of layers in HGCN to 1. The aggregation method in HGCN is summation, with a dropout layer added. In the encoder, we set the number of GAU blocks to 10. The dimensions of value d_Q and key d_V are set to 128, and the dimension of the hidden representation d_h is set to 512. The final output utilizes a residual structure. The input to the classification layer is obtained by summing all the representations at positions not masked. The model is trained on a NVIDIA A100 (80G) GPU. To optimize the model, we use the Adam optimizer [27] with a learning rate of 1e-5. The batch size is set to 16, and the number of epochs is set to 80. For the three hyper-parameters in inference: the maximum number of inquiring turns is set to 10, since it is reasonable and consistent with the previous works [16]. The minimum probability for inquiring is set to 0.01, and the maximum probability for diagnosis is set to 0.9.

B. MAIN RESULTS

1) OVERALL EXPERIMENT

We evaluate our proposed model on four public datasets, and the experimental results are presented in Tables 2 ∼ 5. Based on the results presented in the tables, it is evident that: 1) When restricting the number of turns to 10, our method outperforms the SoTA results on all datasets, demonstrating the effectiveness of our approach. Specifically, on the GMD-12 dataset, we achieve improvements of over 1.7% in disease accuracy compared to the SoTA models. 2) Furthermore, our model surpasses the performance of existing models in terms of symptom recall, with particularly noteworthy enhancements of over 14% on the Muzhi-10 dataset and over 27% on the GMD-12 dataset. 3) Besides, it can be observed that the overall performance of Muzhi-10 dataset is much lower compared to the other three datasets, which can reflect the difficulty of this dataset. However, our model still performs better on it, especially in terms of symptom recall, indicating that our model is also suitable for more challenging datasets. 4) Finally, although our model performs slightly worse than the method by Tian et al. on the Muzhi-4 dataset, our model outperforms their method in terms of average performance across both the DXY and Muzhi-4 datasets. Specifically, our average disease diagnosis accuracy is 79.6%, while the method by Tian et al. achieves only 75.7%, resulting in an improvement of over 3.9%.

2) DIAGNOSIS WITH SMALLER LIMITED TURNS

We observe that our model requires more inquiry turns, particularly when compared to some early RL models.

TABLE 2. Comparison results on DXY. MTurn represents the maximum number of conversation rounds, and an MTurn of 10 indicates that the number of conversation rounds falls between 5 and 10 rounds.

MTurn	Model	DAcc	SRec	ATurn
5	Flat-DQN(2018)	0.731	0.110	1.96
	REFUEL(2018)	0.721	0.186	3.11
	GAMP(2020)	0.731	0.268	2.84
	HRL(2020)	0.695	0.161	2.40
	PPO(2021)	0.746	-	3.30
	Ours	**0.779**	**0.579**	5.00
10	KR-DS(2019)	0.740	0.399	5.65
	MMF-AC(2022)	0.827	-	6.08
	BSODA(2022)	0.802	-	7.49
	Diaformer(2022)	0.806	0.778	9.60
	DxFormer(2023)	0.817	0.506	6.32
	CoAD(2023)	0.821	0.781	9.80
	Tian et al.(2024)	0.752	-	-
	Ours	**0.837**	**0.781**	9.95

TABLE 3. Comparison results on Muzhi-4.

MTurn	Model	DAcc	SRec	ATurn
5	Flat-DQN(2018)	0.681	0.062	1.27
	KR-DS(2019)	0.678	0.177	4.61
	GAMP(2020)	0.644	0.107	2.93
	HRL(2020)	0.694	0.276	3.50
	MMF-AC(2022)	0.739	-	4.47
	Ours	**0.747**	**0.464**	5.00
10	REFUEL(2018)	0.716	0.215	5.01
	PPO(2021)	0.732	-	6.30
	BSODA(2022)	0.731	-	8.80
	Diaformer(2022)	0.731	0.655	9.80
	DxFormer(2023)	0.743	0.479	8.74
	CoAD(2023)	0.730	0.652	9.98
	Tian et al.(2024)	**0.761**	-	-
	Ours	0.754	**0.677**	10.00

TABLE 4. Comparison results on Muzhi-10.

MTurn	Model	DAcc	SRec	ATurn
10	Flat-DQN(2018)	0.408	0.047	9.75
	REFUEL(2018)	0.505	0.262	5.50
	KR-DS(2019)	0.485	0.279	5.95
	HRL(2020)	0.556	0.295	6.99
	Diaformer(2022)	0.530	0.420	7.87
	DxFormer(2023)	0.528	0.412	7.55
	Ours	**0.568**	**0.567**	7.98

However, from the perspective of doctor, inquiring about 7-10 symptoms is acceptable, which allows for a more comprehensive coverage of implicit symptoms [24]. It is worth noting that the number of inquiry turns cannot be controlled, as most models are RL-based. To ensure a relatively fair comparison, we restrict the maximum number of inquiry turns of our model to match that of baselines. When restricting the number of turns to 5, our method also achieve the best results. Specifically, on the DXY dataset, we achieve a 3.3% improvement in disease accuracy compared to the PPO model. Additionally, from the results of Muzhi-4 dataset, it can be observed that in terms of disease accuracy, we surpass the SoTA performance in 10 turns with just 5 turns of inquiry. This demonstrates the potential of our model to achieve better diagnostic results with smaller limited turns.

TABLE 5. Comparison results on GMD-12.

MTurn	Model	DAcc	SRec	ATurn
10	Flat-DQN(2018)	0.620	0.050	-
	KR-DS(2019)	0.690	0.210	-
	Diaformer(2022)	0.719	0.371	9.97
	BR-Agent(2022)	0.820	0.500	-
	DxFormer(2023)	0.689	0.350	7.72
	Ours	**0.837**	**0.771**	8.27

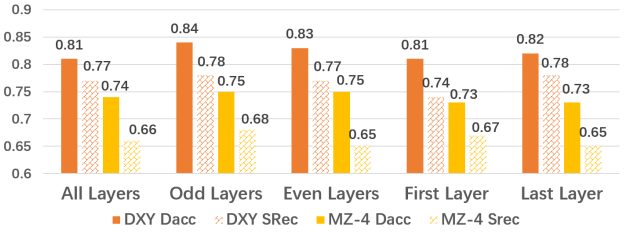

FIGURE 5. The impact of adding the prior knowledge attention matrix to different layers.

C. ABLATION STUDY

To validate the effectiveness of our proposed methods, we conduct ablation experiments about embedding and encoder parts on all four datasets. Firstly, for the embedding, we remove the graph embedding and instead directly sum the representations of symptoms and attributes. Secondly, for the encoder, we remove the conditional probability matrix and use the raw attention. The experimental results are presented in Table 6, and it can be observed that: 1) Removing either of these two components results in a decrease in model performance, indicating the effectiveness of our proposed methods. 2) Specifically, removing the graph embedding can lead to a decrease in disease diagnosis accuracy of 2%. 3) Removing the conditional probability matrix can result in a decrease in disease diagnosis accuracy of 2.3%.

D. DETAILED ANALYSIS

1) THE IMPACT OF ADDING CONDITIONAL PROBABILITY MATRIX TO DIFFERENT LAYERS

The conditional probability matrix can be incorporated into the attention matrix of each block. Since there are several GAU blocks in the encoder, we conduct entailed experiments to assess the effects of adding the matrix at different layers on the performance of our model. Regarding the experimental setup, we conducted experiments using two datasets, DXY and MZ-4, as examples. We explore several intuitive approaches for incorporating the matrix, namely adding it to all layers, odd-numbered layers, even-numbered layers, the first layer, and the last layer. The experimental results are presented in Figure 5, indicating that adding the conditional probability matrix to either odd-numbered or even-numbered layers yields slightly better results compared to other approaches. The reasons for this are twofold: 1) Adding only to the first or last layer is insufficient to have a significant impact on the model, so the result is

TABLE 6. Ablation study.

Model	DXY			Muzhi-4			Muzhi-10			GMD-12		
	DAcc	SRec	ATurn	DAcc	SRec	ATurn	DAcc	SRec	ATurn	DAcc	SRec	ATurn
Ours	**0.837**	**0.782**	9.95	**0.754**	**0.677**	10.00	0.568	**0.567**	7.98	**0.837**	**0.771**	8.27
w/o sym	0.814	0.772	9.86	0.732	0.654	9.93	0.563	0.552	7.87	0.816	0.768	8.35
w/o KG	0.817	0.772	9.71	0.739	0.652	10.00	**0.573**	0.552	7.88	0.820	0.767	8.14

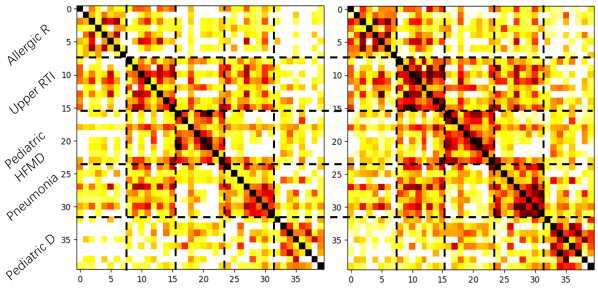

FIGURE 6. The impact of graph embedding on representing patient similarity. "Allergic R", "Upper RTI", "Pediatric HFMD", and "Pediatric D" are abbreviations for Allergic rhinitis, Upper respiratory tract infection, Pediatric hand-foot-mouth disease, and Pediatric diarrhea, respectively.

similar to the ablation experiment of removing the matrix. 2) Adding to all 10 layers may lead the model to overly rely on prior knowledge, thus impeding its learning capacity. Since the odd-numbered layers have better performance on both datasets, we adopt this method in subsequent experiments.

2) THE IMPACT OF GRAPH EMBEDDING

Since we believe that patients with the same disease exhibit similar topological connections, incorporating the topological representation of the graph into the embedding layer can better model patient representations. Hence, we conduct experiments to further analyze whether it truly enhances the representation capabilities. In the experimental setup, we take DXY as an example and randomly select eight patients for each disease, resulting in a total of forty patients. Then, we calculate the similarity (using cosine similarity) between each pair of patients by comparing the original embedding representations without patient-oriented graph and the embedding representations with patient-oriented graph. Finally, we visualize the results using a heat-map. The experimental results are shown in Figure 6, with the left side representing the embedding without patient-oriented graph and the right side representing the embedding with patient-oriented graph. The darker the color, the closer the similarity is to 1, and vice versa. From figure 4, we can observe that patients with the same disease exhibit darker colors in the corresponding regions, indicating higher similarity. This demonstrates that incorporating topological information can indeed enhance the ability to encode patient representations more effectively. Additionally, we can also observe that patients with ''pneumonia'' and ''upper respiratory tract infection'' have high similarity, which aligns with real-world scenarios as these two diseases share similar symptom manifestations.

TABLE 7. Performance comparison with ChatGPT.

Models	ICL	DXY		Muzhi-4	
		DAcc	SRec	DAcc	SRec
ChatGPT-3.5	0-shot	0.665	0.696	0.641	0.602
	1-shot	0.683	0.694	0.648	0.613
	3-shot	0.683	0.705	0.648	0.627
ChatGPT-4	0-shot	0.712	0.716	0.655	0.629
	1-shot	0.731	0.705	0.655	0.613
	3-shot	0.731	0.721	0.670	0.622
Ours	-	**0.837**	**0.781**	**0.754**	**0.677**

3) PERFORMANCE COMPARISON WITH CHATGPT

Since large language models (LLMs) can directly perform automatic diagnostic tasks, we further conduct comparative experiments with LLMs. In the experimental setup, we select both ChatGPT-3.5 and ChatGPT-4. To achieve better results, we employ the ICL technique, i.e., few-shot learning. Specifically, we conduct 0-shot, 1-shot, and 3-shot experiments. In the prompts, we provide all candidate symptom and disease sets, emphasizing that only one symptom could be queried at a time. To avoid excessively large candidate sets, we limit the experimental datasets to DXY and Muzhi-4. Finally, to ensure the accuracy of results, we repeat the experiments multiple times (N=5). The experimental results, as shown in Table 7, indicate that: 1) In the 0-shot setting, our model outperforms both ChatGPT-3.5 and ChatGPT-4, with over 10% improvement in disease diagnosis accuracy. 2) In the few-shot setting, ChatGPT's performance shows some improvement compared to the 0-shot scenario, but our model's performance still significantly surpasses ChatGPT.

4) CASE STUDY

In this section, to better illustrate the input-output structure and advantages of our model, we use a real example as shown in Table 8. Given a patient, the input consists of the patient's explicit symptoms, including restlessness, fever and rash. The goal of the model is to output the same as the target, which is to inquire about symptoms to include as many implicit symptoms as possible and predict the disease to be the same as the target disease. This example indicates that: 1) Our model only inquiries about five symptoms, including four implicit symptoms, indicating a high recall rate of implicit symptoms and accurate disease diagnosis. 2) We select two SOTA models for comparison, namely CoAD and Diaformer. Among them, CoAD shows low relevance in the symptoms it

TABLE 8. Case study.

Input & Goal
Explicit symptoms:
{ Restlessness: True, Fever: True, Rash: True }
Implicit symptoms:
{ Herpes: True, Listlessness: True, Cough: False, Vomiting: False, Anorexia: False }
Target disease: Hand, Foot, and Mouth Disease in Children
Output
Ours:
Inquiry symptoms:
Herpes, Anorexia, Listlessness, Vomiting, Difficulty Breathing
Predicted disease: Hand, Foot, and Mouth Disease in Children ✓
CoAD:
Inquiry symptoms:
Vomiting, Sore throat, Restlessness, Runny nose, Cough
Predicted disease: Pediatric diarrhea ✗
Diaformer:
Inquiry symptoms:
Anorexia, Rash, Lethargy, Loss of appetite, Herpes, Restlessness, Sore throat, Difficulty Breathing, Cough
Predicted disease: Hand, Foot, and Mouth Disease in Children ✓

inquiries about, while also making wrong disease diagnosis. Although Diaformer achieves accurate disease diagnosis, it requires a higher number of symptom inquiry rounds and does not inquire about more implicit symptoms in fewer rounds. This indicates that our model has to some extent learned more realistic diagnostic patterns and has better performance.

VI. RELATED WORK

In recent years, there has been a significant surge in research on automatic diagnosis, focusing on model design and construction for strategy selection.

In the early stages, a considerable number of methods were based on statistical models. Among them, Bayesian models [5], [28], [29] are dominant, which define symptom inquiry as a feature selection task. These models utilize entropy functions to find optimal features and maximize information gain as the training objective. Guan and Baral [5] utilize the Quick Medical Reference belief network and respectively uses Bayesian inference and Bayesian experimental design in the inference phase and the inquiry phase. Wang et al. [30] develop a disease self-diagnosis system that searches and provides relevant alternative symptoms based on predefined meta-paths by graph representation learning. Besides, some latest works [11], [23], [31] define this task as a sequence generated task, which utilizes transformer as the backbone of model. Diaformer [10] proposes a symptom attention framework and three unordered training mechanisms to eliminate the impact of sequence order. This type of methods has fast training speed and is easy to expand to large-scale data. Tian et al. [26] introduces an experience-driven knowledge model, iteratively learning from data and injecting it into a knowledge graph to address data collection and multi-classification challenges via entity linking, graph fusion, adaptive neural network construction, and incremental updates.

Nowadays, most of works define the automatic diagnosis task as a sequential decision-making task. It is naturally modeled as a Markov decision-making process, so RL is used for strategy learning. Among that, Flat-DQN [1] utilizes a basic DQN model to train the embedding of symptom questioning and disease diagnosis in the motion selection space. REFUEL [18] proposes a method of feature reconstruction and reward remodeling to improve model training performance. After that, KR-DS [15] proposes introducing prior statistical information into action selection. GAMP [19] first introduces generative adversarial networks of this task, which trains a symptom generator and a symptom discriminator respectively through the game between them. HRL [20] utilizes a two-layer hierarchical RL framework to mimic the offline diagnostic process, where the first layer is used for coarse-grained department determination, and the second layer is used for fine-grained disease diagnosis. Liu et al. [32] propose leveraging learned dialogue models to customize reinforcement learning settings for efficient action space exploration, and designs a clustering method and pre-training strategy. DxFormer [24] decouples symptom inquiry and disease diagnosis, utilizing a Transformer-based model to treat symptoms as tokens, enabling independent optimization through reinforcement reward and cross-entropy loss. Tiwari et al. [33] propose a novel knowledge-infused and context-driven hierarchical reinforcement learning diagnostic dialogue system, which achieves intelligent and context-aware symptom investigation and significantly improved disease classification accuracy through the Potential Candidate Module (PCM) and the Hierarchical Disease Classifier (HDC). CoAD [25] introduces a framework that aligns disease labels with symptom steps, augments symptom labels to strengthen training signals, and combines repeated symptom input with an attention schema for concurrent symptom and disease generation.

Additionally, with the emergence of large language models, recent works have adopted LLMs for this task. Wang et al. [34] employ an agent derived from LLMs without fine-tuning as the model and propose the Agent-derived Multi-Specialist Consultation (AMSC) framework, which adaptively integrates the probability distributions of potential diseases from the agents to model the real-world diagnostic process. Jin et al. [35] propose an innovative framework, Health-LLM, by integrating health reports and medical knowledge, large-scale feature extraction, and the retrieval-augmented generation (RAG) mechanism to improve the accuracy of disease prediction and personalized healthcare. Zhang et al. [36] combine LLMs with Markov logic networks (MLNs) and external medical knowledge to enhance the interpretability and accuracy of medical diagnosis.

Finally, some works have expanded this task to end-to-end medical dialogue generation. Jin et al. [35] utilize LLMs to

directly generate dialogues by integrating health reports and medical knowledge. Varshney et al. [37] propose a medical-knowledge-aware neural model, MED, by integrating UMLS and medical entities extracted using BERT, embedding the triples from the knowledge graph into pre-trained language models to improve the accuracy and relevance of medical dialogue generation.

VII. CONCLUSION

In this article, we propose an end-to-end model KDPoG. Firstly, in addition to use the symptom and attribute embedding, we propose patient-oriented graph enhanced representation learning, which is built by a patient-oriented graph and learned with heterogeneous graph convolution networks. Furthermore, based on the encoder built with graph attention units, we propose knowledge-guided attention mechanism learning, which incorporates conditional probabilities of co-occurrence between symptom pairs. Finally, we utilize two linear layers as the classification module to achieve symptom probing and disease diagnosis. We conduct extensive experiments on four public datasets, which demonstrate that our proposed model outperforms the state-of-the-art methods. We achieve an average absolute improvement of over 2% in disease diagnosis accuracy. Particularly, on the MZ-10 dataset, we observe an absolute improvement of over 14.7% in symptom recall rate.

LIMITATIONS

In this work, our model has two key limitations that will be further addressed in future research. The first limitation is that our model's input is relatively singular, only able to take symptoms and their attributes as input. However, in practical application scenarios, other key information such as the severity of symptoms, affected areas, and the patient's medical history will also be included, and these pieces of information are crucial for diagnosis. To mitigate this limitation, our potential approach may involve adding the aforementioned information to the input encoding section, or directly encoding self-reported text using language models. The second limitation is that in practical applications, our model needs to be complemented with other NLP components to achieve automated diagnosis, including NLU for information extraction and normalization, and NLG for natural language text responses. To facilitate the end-to-end construction of the entire process, we plan to use LLMs to implement other components. Additionally, LLMs can provide external knowledge, and the combination of large and small models theoretically enables higher-quality follow-up questions and more accurate diagnosis.

COMPETING INTERESTS

The authors declare that there is no conflict of interest.

REFERENCES

[1] Z. Wei, Q. Liu, B. Peng, H. Tou, T. Chen, X. Huang, K.-F. Wong, and X. Dai, "Task-oriented dialogue system for automatic diagnosis," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics*, 2018, pp. 201–207.

[2] K.-F. Tang, H.-C. Kao, C.-N. Chou, and E. Y. Chang, "Inquire and diagnose: Neural symptom checking ensemble using deep reinforcement learning," in *Proc. NIPS Workshop Deep Reinforcement Learn.*, 2016.

[3] S. Min, B. Lee, and S. Yoon, "Deep learning in bioinformatics," *Briefings Bioinf.*, vol. 18, no. 5, pp. 851–869, 2017.

[4] A. F. Tchango, R. Goel, Z. Wen, J. Martel, and J. Ghosn, "DDXPlus: A new dataset for automatic medical diagnosis," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 35, 2022, pp. 31306–31318.

[5] H. Guan and C. Baral, "A Bayesian approach for medical inquiry and disease inference in automated differential diagnosis," 2021, *arXiv:2110.08393*.

[6] R. Kohavi, "Scaling up the accuracy of Naive–Bayes classifiers: A decision-tree hybrid," in *Proc. KDD*, vol. 96, 1996, pp. 202–207.

[7] B. Peng, X. Li, L. Li, J. Gao, A. Celikyilmaz, S. Lee, and K.-F. Wong, "Composite task-completion dialogue policy learning via hierarchical deep reinforcement learning," 2017, *arXiv:1704.03084*.

[8] B. Liu, G. Tur, D. Hakkani-Tur, P. Shah, and L. Heck, "End-to-end optimization of task-oriented dialogue model with deep reinforcement learning," 2017, *arXiv:1711.10712*.

[9] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 5998–6008.

[10] J. Chen, D. Li, Q. Chen, W. Zhou, and X. Liu, "Diaformer: Automatic diagnosis via symptoms sequence generation," in *Proc. AAAI Conf. Artif. Intell.*, 2022, pp. 4432–4440.

[11] L. Chen, K. Lu, A. Rajeswaran, K. Lee, A. Grover, M. Laskin, P. Abbeel, A. Srinivas, and I. Mordatch, "Decision transformer: Reinforcement learning via sequence modeling," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, vol. 34, no. 3, pp. 15084–15097.

[12] W. Lei, X. Jin, M.-Y. Kan, Z. Ren, X. He, and D. Yin, "Sequicity: Simplifying task-oriented dialogue systems with single sequence-to-sequence architectures," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics*, 2018, pp. 1437–1447.

[13] C. Zhang, X. Gao, L. Ma, Y. Wang, J. Wang, and W. Tang, "GRASP: Generic framework for health status representation learning based on incorporating knowledge from similar patients," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 715–723.

[14] X. He, S. Chen, Z. Ju, X. Dong, H. Fang, S. Wang, Y. Yang, J. Zeng, R. Zhang, R. Zhang, M. Zhou, P. Zhu, and P. Xie, "MedDialog: Two large-scale medical dialogue datasets," 2020, *arXiv:2004.03329*.

[15] L. Xu, Q. Zhou, K. Gong, X. Liang, J. Tang, and L. Lin, "End-to-end knowledge-routed relational dialogue system for automatic diagnosis," in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 7346–7353.

[16] W. Liu, Y. Cheng, H. Wang, J. Tang, Y. Liu, R. Zhao, W. Li, Y. Zheng, and X. Liang, "'My nose is running' 'are you also coughing?' Building a medical diagnosis agent with interpretable inquiry logics," in *Proc. Int. Conf. Artif. Intell.*, 2022, pp. 4266–4272.

[17] W. Chen, Z. Li, H. Fang, Q. Yao, C. Zhong, J. Hao, Q. Zhang, X. Huang, J. Peng, and Z. Wei, "A benchmark for automatic medical consultation system: Frameworks, tasks and datasets," *Bioinformatics*, vol. 39, no. 1, Jan. 2023, Art. no. btac817.

[18] Y.-S. Peng, K.-F. Tang, H.-T. Lin, and E. Chang, "Refuel: Exploring sparse features in deep reinforcement learning for fast disease diagnosis," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, Dec. 2018.

[19] Y. Xia, J. Zhou, Z. Shi, C. Lu, and H. Huang, "Generative adversarial regularized mutual information policy gradient framework for automatic diagnosis," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 1062–1069.

[20] C. Zhong, K. Liao, W. Chen, Q. Liu, B. Peng, X. Huang, J. Peng, and Z. Wei, "Hierarchical reinforcement learning for automatic disease diagnosis," 2020, *arXiv:2004.14254*.

[21] M. S. Teixeira, V. Maran, and M. Dragoni, "The interplay of a conversational ontology and AI planning for health dialogue management," in *Proc. 36th Annu. ACM Symp. Appl. Comput.*, Mar. 2021, pp. 611–619.

[22] W. He and T. Chen, "Scalable online disease diagnosis via multi-model-fused actor-critic reinforcement learning," in *Proc. 28th ACM SIGKDD Conf. Knowl. Discovery Data Mining*, Aug. 2022, pp. 4695–4703.

[23] W. He, X. Mao, C. Ma, Y. Huang, J. M. Hernàndez-Lobato, and T. Chen, "BSODA: A bipartite scalable framework for online disease diagnosis," in *Proc. ACM Web Conf.*, Apr. 2022, pp. 2511–2521.

[24] W. Chen, C. Zhong, J. Peng, and Z. Wei, "DxFormer: A decoupled automatic diagnostic system based on decoder–encoder transformer with dense symptom representations," *Bioinformatics*, vol. 39, no. 1, Jan. 2023, Art. no. btac744.
[25] H. Wang, W. C. Kwan, K.-F. Wong, and Y. Zheng, "CoAD: Automatic diagnosis through symptom and disease collaborative generation," in *Proc. 61st Annu. Meeting Assoc. Comput. Linguistics*, 2023, pp. 6348–6361.
[26] Y. Tian, Y. Jin, Z. Li, J. Liu, and C. Liu, "Weighted heterogeneous graph-based incremental automatic disease diagnosis method," *J. Shanghai Jiaotong Univ., Sci.*, vol. 29, no. 1, pp. 120–130, Feb. 2024.
[27] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Represent.*, May 2015, pp. 1–15.
[28] J. Chen, X. Dai, Q. Yuan, C. Lu, and H. Huang, "Towards interpretable clinical diagnosis with Bayesian network ensembles stacked on entity-aware CNNs," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 3143–3153.
[29] M. Julia Flores, A. E. Nicholson, A. Brunskill, K. B. Korb, and S. Mascaro, "Incorporating expert knowledge when learning Bayesian network structure: A medical case study," *Artif. Intell. Med.*, vol. 53, no. 3, pp. 181–204, Nov. 2011.
[30] Z. Wang, R. Wen, X. Chen, S. Cao, S.-L. Huang, B. Qian, and Y. Zheng, "Online disease diagnosis with inductive heterogeneous graph convolutional networks," in *Proc. Web Conf.*, Apr. 2021, pp. 3349–3358.
[31] A. Nesterov, B. Ibragimov, D. Umerenkov, A. Shelmanov, G. Zubkova, and V. Kokh, "NeuralSympCheck: A symptom checking and disease diagnostic neural model with logic regularization," in *Proc. Int. Conf. Artif. Intell. Med.* Cham, Switzerland: Springer, 2022, pp. 76–87.
[32] Z. Liu, Y. Li, X. Sun, F. Wang, G. Hu, and G. Xie, "Dialogue based disease screening through domain customized reinforcement learning," in *Proc. 27th ACM SIGKDD Conf. Knowl. Discovery Data Mining*, Aug. 2021, pp. 1120–1128.
[33] A. Tiwari, S. Saha, and P. Bhattacharyya, "A knowledge infused context driven dialogue agent for disease diagnosis using hierarchical reinforcement learning," *Knowl.-Based Syst.*, vol. 242, Apr. 2022, Art. no. 108292.
[34] H. Wang, S. Zhao, Z. Qiang, N. Xi, B. Qin, and T. Liu, "Beyond direct diagnosis: LLM-based multi-specialist agent consultation for automatic diagnosis," 2024, *arXiv:2401.16107*.
[35] M. Jin, Q. Yu, D. Shu, C. Zhang, L. Fan, W. Hua, S. Zhu, Y. Meng, Z. Wang, M. Du, and Y. Zhang, "Health-LLM: Personalized retrieval-augmented disease prediction system," 2024, *arXiv:2402.00746*.
[36] H. Zhang, J. Li, Y. Wang, and Y. Song, "Integrating automated knowledge extraction with large language models for explainable medical decision-making," in *Proc. IEEE Int. Conf. Bioinf. Biomed. (BIBM)*, Dec. 2023, pp. 1710–1717.
[37] D. Varshney, A. Zafar, N. K. Behera, and A. Ekbal, "Knowledge grounded medical dialogue generation using augmented graphs," *Sci. Rep.*, vol. 13, no. 1, p. 3310, Feb. 2023.

ZHIANG LI was born in Tianmen, Hubei, China, in 1999. He received the B.S. degree in mathematics and applied mathematics from the East China University of Science and Technology, Shanghai, China, in 2021, where he is currently pursuing the master's degree with the School of Information Science and Engineering.

His main research interests include machine learning and data mining.

TONG RUAN was born in 1973. She received the B.S. and master's degrees from the East China University of Science and Technology, Shanghai, China, and the Ph.D. degree from the Institute of Software Chinese Academy of Sciences.

She is currently a Professor and a Ph.D. Supervisor with the East China University of Science and Technology. Her main research interests include knowledge graph, data mining, and data quality assessment. She is a member of China Computer Federation (CCF).