Dialogue-Based Disease Diagnosis Using Hierarchical Reinforcement Learning with Multi-Expert Feedback

Shi Li, Xueyao Sun
College of Computer and Control Engineering, Northeast Forestry University, Harbin, China

Abstract—**In order to minimize the stochasticity of agents used in disease diagnosis within the dialogue system, and to enable them to interact with users based on the inherent connections between symptoms and diseases, while simultaneously addressing the issue of limited medical data, we propose the Hierarchical Reinforcement Learning with Multi-expert Feedback framework. The framework constructs a reward model in the lower-level networks of the hierarchical structure. Here, the discriminator leveraging the concept of adversarial networks generates rewards by evaluating the authenticity of symptom query sequences generated by the agent, and the large language model of human experts synthesizes various factors to assess the reasonableness of the agent's current symptom queries, thereby guiding the learning of the policy network. The algorithm addresses the deficiencies in data characteristics and improves the policy's capability to leverage feature information, thus making the process of disease diagnosis more aligned with clinical practice. Experimental results demonstrate that the proposed framework achieves diagnostic success rates of 61.5% on synthetic datasets and 84.4% on real-world datasets, while requiring fewer dialogue turns on average. Both metrics surpass those of conventional approaches, further indicating the framework's strong generalization ability.**

Keywords*—*Disease diagnosis; dialogue system; large language model; reinforcement learning; reward model; adversarial network; dialogue agent

I. INTRODUCTION

Due to the broad application prospects and significant commercial benefits of task-oriented dialogue systems [1], researchers have introduced them into the medical field, forming research focused on medical dialogue systems to alleviate the problem of strained medical resources [2].The rapid development of deep learning has propelled advancements in the field of disease diagnosis, and the application of deep reinforcement learning in conversational disease diagnosis research is gradually emerging [3]. However, deep reinforcement learning still faces two major challenges in this field: first, it relies too heavily on random sampling [4] and lacks a universal reward and punishment mechanism adaptable to different application scenarios. For example, Kao et al. [5]use +1 and 0 as reward values, whereas Zhong et al. [6] set substantial rewards and penalties of +20 and -100. Second, medical data resources are scarce and valuable, and the available feature information is also very limited, which greatly restricts research in disease diagnosis.

In response to the aforementioned challenges, this study introduces a Multi-Expert Feedback-based Hierarchical Reinforcement Learning framework (ME-RL), utilizing a hierarchical structure to manage complex sequential decision-making issues, and leveraging the principles of Generative Adversarial Networks, it establishes an adversarial workflow comprising a generator and a discriminator. Utilizing the evaluations from the discriminator and the feedback from the large language model of human experts (AIExpert) as low-level rewards, the framework optimizes the policy and improves the agent's learning efficiency and performance within dynamic environments. In addition, a novel joint evaluation metric is introduced, which simultaneously takes into account the diagnostic success rate and the average number of dialogue turns in disease diagnosis tasks, thereby offering a more comprehensive assessment of the model's effectiveness.

The remainder of this paper is organized as follows: Section II introduces the related work. Section III provides an overview of the methods. Section IV introduces the experimental design. Section V presents the experimental results and related analysis. Section VI summarizes the paper and proposes future research directions.

II. RELATED WORK

Task-oriented dialogue systems have received widespread attention in recent years, encompassing areas such as ticket booking services and e-commerce [7], [8]. In the context of strained medical resources, researchers have begun exploring how to integrate dialogue systems into the smart healthcare field to improve doctors' time utilization and provide patients with preliminary diagnostic references [9]. Through continuous development, conversational disease diagnosis systems have demonstrated significant potential in simplifying diagnostic processes, reducing costs, and efficiently collecting patient medical history information [10].

Due to the fact that conversational disease diagnosis involves multiple consecutive time stages and interactive steps, its decision-making characteristics are highly compatible with the applicability of reinforcement learning (RL) methods. Researchers model the disease diagnosis task as a Markov Decision Process and utilize reinforcement learning methods to optimize policy learning [6]. By employing policy gradient algorithms, the agent can make decisions based on the specific symptom information provided by patients.

With the continuous advancement of deep learning, many researchers have integrated deep learning models [11], [12] for disease diagnosis prediction. Deep reinforcement learning, which combines deep learning and reinforcement learning, has also been rapidly applied to research on conversational disease diagnosis [13]. In the context of symptom-oriented disease diagnosis dialogue systems, researchers such as Wei et al. [14] have designed an automatic diagnostic system based on human-computer dialogue. This system models the symptom inquiry and disease diagnosis processes as a Markov Decision Process and employs a Deep Q-Network (DQN) policy learning algorithm. In each interaction cycle, the intelligent agent can choose to inquire about specific symptoms or make diagnostic judgments based on the situation. The agent learns the optimal strategy by continuously optimizing to maximize the expected simulated rewards. The dialogue system constructed in this manner is relatively comprehensive, marking a preliminary breakthrough in addressing automatic diagnostic medical dialogue problems. Since then, reinforcement learning has gradually become the mainstream choice among researchers in this field [15].

To further enhance the performance of automatic diagnostic models, Chen et al. [16] proposed a multi-action policy representation method, enabling agents to timely recommend medical examination items to assist in diagnosis. Xu et al. [2] further introduced the KR-DS system, which integrates Knowledge Path-based Deep Q-Networks (DQN). This system incorporates knowledge branches and knowledge routing branches that capture the associations between diseases and symptoms within the deep reinforcement learning framework, thereby considering external probabilistic symptom information related to the reinforcement learning framework. This integration enhances the rationality and accuracy of medical dialogue decision-making. Tiwari et al. [17] designed a hierarchical reinforcement learning framework that integrates a knowledge-driven mechanism. By embedding a Potential Candidate Module (PCM), the framework enhances the agent's efficiency in organizing and probing symptom information. Yan et al. [18] addressed the challenges of insufficient diagnostic evidence and irrelevant symptom inquiries by constructing a more comprehensive medical dialogue dataset and combining experiential diagnostic knowledge with a medical knowledge graph. Subsequently, Yan et al. [19] proposed the EIRAD dialogue system based on a medical knowledge graph, which enhances the interpretability and accuracy of the diagnostic process by utilizing the topological structure of the knowledge graph.

Existing methods have laid the foundation for the optimization and development of dialogue-based disease diagnosis systems. However, they still largely rely on random sampling, lack a universal reward-punishment mechanism adaptable to different application scenarios, and are overly dependent on scarce medical data resources. To address these limitations, the framework proposed in this paper integrates hierarchical reinforcement learning, adversarial networks, and large language models, constructing a general and effective reward model. Additionally, it leverages the rich database of large language models to mitigate the issue of sparse data features, enabling the dialogue agent to generate reasonable disease query sequences, thereby enhancing the performance of the disease diagnosis system. Experimental results demonstrate that, compared to traditional learning methods, the proposed approach significantly improves diagnostic accuracy, reduces dialogue turns, and enhances the generalization capability of the model. Table I provides a summary and comparison of existing research, highlighting their methodologies, strengths and weaknesses.

TABLE I. RELATED WORK COMPREHENSIVE

Methodology	Strengths	Weaknesses	Reference
RL, Knowledge graph	Leveraging the relationships between symptoms and diseases	Lack of a universal reward model	[2]
RL, Prioritization of severe pathologies	Exploration-confirmation framework, Severe pathology prioritization	Lack of a universal reward model, Dependency on dataset quality	[3]
RL, Feature rebuilding	Sparse feature exploration, Faster diagnosis	Lack of a universal reward model	[4]
RL, Context-aware symptom checking	Context-aware diagnosis, Generalization across tasks	Lack of a universal reward model	[5]
RL	Novel dataset, Task-oriented framework	Dependency on dataset quality	[14]
RL, Label-guided exploration	Efficient exploration, Multiple test suggestions	Lack of real-world dataset validation	[16]
RL, Potential candidate module	Context-aware and knowledge-guided investigation	Lack of real-world dataset validation	[17]
RL, Medical knowledge graph	Interpretability, Integration of medical knowledge	Dependency on data quality	[19]

III. METHOD DESIGN

A. Overall Framework

The structure of the disease diagnosis model based on Multi-Expert Feedback Hierarchical Reinforcement Learning is illustrated in Fig. 1. Its core component is the diagnostic agent, which is divided into a two-tier structure comprising high-level and low-level policies, and consists of four key modules: controller, generator, evaluator, and classifier. Furthermore, it is equipped with a user simulator. Among them, the controller is responsible for scheduling the generator or classifier to operate. The generator is tasked with inquiring about potential symptoms from the patient, the evaluator assesses the rationality of the actions chosen by the generator and provides rewards, while the classifier is used to inform the user simulator of possible disease types based on the currently collected information. The user simulator is designed to mimic patient behavior, interact with the agent, and return intrinsic rewards for the disease classifier. The sum of the rewards obtained by the generator and the disease classifier constitutes the external rewards received by the controller.

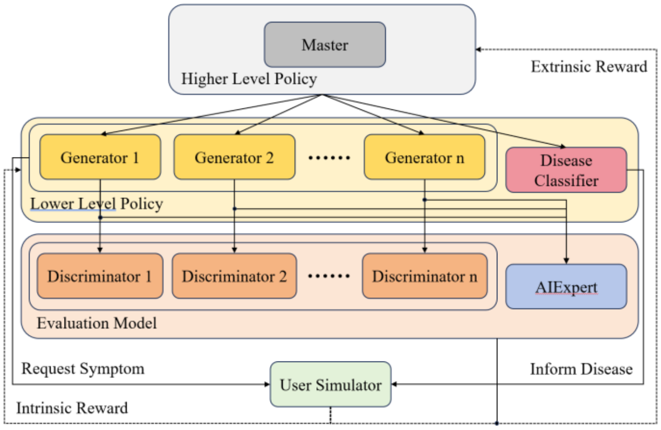


Fig. 1. Overall framework of ME-RL.

B. Dialogue Policy

1) Agent policy: The automatic diagnosis tasks can be formulated as a Markov Decision Process and utilize reinforcement learning to optimize dialogue policies. In this framework, the policy includes the state space S, action space A, state transition probabilities TP, and reward model R. Each state (s, $s \in$ S) in the state space is composed of several elements, including the current state of the dialogue agent, the symptoms pending inquiry, user symptom feedback, a list of all confirmed symptoms, the number of dialogue rounds, and reward signals. The action space includes all possible disease diagnosis options and symptom inquiry actions, with each action corresponding to the examination of a particular symptom or the judgment of a disease. State transition probabilities dictate how the next state s_{t+1} is formed given the current state s_t and the action a taken, represented as s_{t+1} = TP(s_t, a). The reward model offers the probability of receiving an immediate reward r_{t+1} after taking action a_t at time step t, denoted as r_{t+1} = R(s_t, a_t).

The agent's goal is to seek an optimal dialogue strategy that maximizes the total accumulated reward R during one round of interaction [20]. The formula for the total reward R is presented below:

$$R = \sum_{i=1}^{N} \sum_{t=0}^{T} \gamma^t \cdot r_t \tag{1}$$

where N represents the total number of dialogues in a round, T represents the total number of dialogue turns in the current conversation, γ^t represents the discount factor, and r_t represents the immediate reward received by the agent at the t-th time step. $\sum_{t=0}^{T} \gamma^t \cdot r_t$ can be approximated as [21]:

$$\mathrm{Q}^*(s, a) = \mathrm{E}_{s'}[r + \gamma \cdot \max_{a' \in \mathrm{A}} \mathrm{Q}^*(s', a')] \tag{2}$$

where Q(s, a) denotes the Q-function representing the state-action value, which is used to estimate the value acquired after performing action a in the current state s, leading to a new state s'. During the training process, the estimation of Q(s, a) is progressively updated based on experience and learning, until it converges and stabilizes.

2) User simulation: The user simulator is designed to emulate the interaction process between real patients and the automated diagnostic system. At the beginning of each dialogue session, the simulator randomly selects a user objective from the experimental dataset as the basis for the simulation. Each user objective comprises two types of symptom information: one is the explicit symptoms clearly expressed by the user, and the other is the implicit symptoms that can only be uncovered through dialogue interaction. Additionally, the user objective includes the actual disease labels. At the initiation of the dialogue, the diagnostic agent directly receives the explicit symptoms as the initial information. Subsequently, the user simulator engages in further dialogue interactions with the agent based on the implicit symptoms. During the dialogue process, the simulated user will respond based on whether the symptom currently inquired by the agent exists in their objective: correctly, incorrectly, or with uncertainty. When the agent successfully diagnoses the correct disease within the maximum number of rounds, the dialogue is considered a successful interaction; otherwise, it is deemed a failure.

C. Hierarchical Structure

Considering that the agent needs to handle a vast action selection space in complex environments, a hierarchical structure with two levels of policies has been constructed, as shown in Fig. 2.

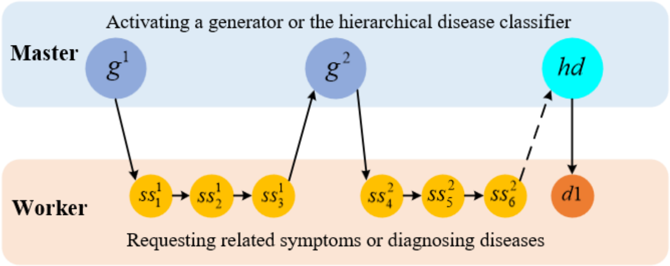

Activating a generator or the hierarchical disease classifier
Master
g^1
g^2
hd
Worker
ss_1^1
ss_2^1
ss_3^1
ss_4^2
ss_5^2
ss_6^2
$d1$
Requesting related symptoms or diagnosing diseases

Fig. 2. Interaction of two layer strategies.

3) Higher policy: The controller manages the high-level policies, with its core function being to schedule and activate lower-level policies, and the action space is denoted as $\mathrm{A}_m = \{g_i \mid i = 1, 2, \ldots, n\} \cup \{d\}$. Here, g_i represents the activation of generator G$_i$, and action d represents the activation of the disease classifier. At time step t, the controller takes the current dialogue state s_t as input and, based on the policy function π_m, decides which action a_m ($a_m \in$ A$_m$) to take, thereby leading to a new state s_{t+1}.Once the controller activates a lower-level agent, the agent will engage in multiple rounds of interaction with the user until its subtask is completed. The learning problem of the controller can be viewed as a Semi-Markov Decision Process. At the t-th time step, after the controller executes action a_m, the immediate reward r_m it receives can be expressed as the discounted sum of all external rewards accumulated by the lower-level policies it has triggered while performing a series of actions a_i. Therefore, the external reward for the controller is calculated as follows:

$$r_m = \begin{cases} \sum_{t_1=1}^{T_1} \gamma_m r_{t_1}^{g}, & if\ a_m = g_i \\ r_d, & if\ a_m = d \end{cases} \tag{3}$$

where $i = 1,\ldots,n$, r_m represents the external rewards obtained by the high-level policy at the t-th round, γ_m is the discount factor of the controller, T_1 is the number of original actions of the generator, and g_i and d are the actions to activate generator G_i and classifier D, respectively. When the generator is activated by the controller's action a_m, it will select appropriate actions a_i based on the current state to query symptoms from the patient based on the current disease group, thereby activating the evaluator to provide corresponding intrinsic rewards $r_{t_1}^{g}$. If the disease classifier is triggered, it will perform disease identification and return rewards r_d based on the diagnostic results. The objective of the high-level agent is to maximize external rewards within a round through a Semi-Markov Decision Process, thus allowing us to derive the controller's loss function [22]:

$$L(\theta_m) = [(r_m + \gamma_m \max_{a'} Q_m(s', a', \theta_m^-)) - Q_m(s, a, \theta_m)]^2 \tag{4}$$

where $L(\theta_m)$ represents the loss of the controller at time step t, while θ_m and θ_m^- are the frozen parameters during the current and past training iterations, respectively.

4) Lower policy: The low-level policy consists of a generator and a disease classifier, with an action space comprising symptom inquiries and disease classification actions. The low-level policy is centrally scheduled by the high-level policy, which provides the current environmental state (s_t) as input to guide the low-level policy in selecting the most appropriate basic action a_i. Once action a_i is executed, the dialogue environment state updates to s_{t+1}, and the low-level agent immediately receives rewards or penalties (r_t^g) based on the suitability of the current action a_i for the current state s_t. Because the rewards obtained by the low-level policy are direct results of the actions currently being executed, these rewards are termed intrinsic rewards. The calculation of intrinsic rewards is as follows:

$$r_t^g = w_{dis} r_t^{dis} + w_{ai} r_t^{ai} \tag{5}$$

where r_t^{dis} represents the reward provided by the discriminator, r_t^{ai} represents the feedback evaluation from AIExpert, w_{dis} and w_{ai} denote their respective weights. Since the SD dataset does not include real patient-doctor symptom inquiry sequences, w_{dis} is set to 0.4 and w_{ai} is set to 0.6.When using the MZ4 dataset, which includes real patient-doctor symptom inquiry sequences, w_{dis} and w_{ai} are both set to 0.4.

The generator is described in detail in subsequent chapters. The classifier adopts a hierarchical classifier structure, consisting of group classifiers and specific disease classifiers. When the disease classifier is activated by the controller, the currently confirmed symptoms are input into the disease classifier. The group classifier then designates the appropriate specific disease classifier, which conducts a detailed diagnosis based on the current disease group. Each specific disease classifier for a particular disease group is constructed as a three-layer neural network structure, including a hidden layer, which is used to accurately identify and diagnose each specific disease within the selected disease group, and to feedback the disease diagnosis result with the highest probability to the user.

D. Evaluation Model Based on Multi-Expert Feedback

The evaluation model based on multi-expert feedback consists of multiple discriminators corresponding to the generator and one AIExpert. The adversarial structure-based evaluation process is illustrated in Fig. 3. Both the generator and the discriminator have been pre-trained. The generator employs the maximum likelihood estimation method, predicting the next possible symptom based on the current state during training, ultimately generating a symptom inquiry sequence (fake symptom sequence). Additionally, a dedicated data repository is established to store both real and fake symptom sequences for pre-training the discriminator. The real symptom sequences in the database are directly sampled from user objectives in the dataset. An example of AIExpert model interaction is shown in Fig. 4, where AIExpert uses interaction history as the environmental information for large language model (LLM) scoring. Firstly, the RL model generates the next symptom to inquire, which, along with the interaction history, is processed into a prompt. Subsequently, the LLM uses this prompt to generate a score for the queried symptom, and finally, the score is returned to the RL model as a reward.

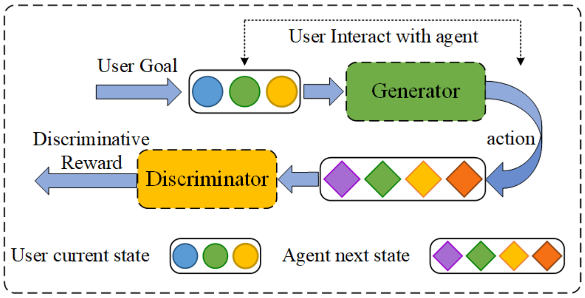
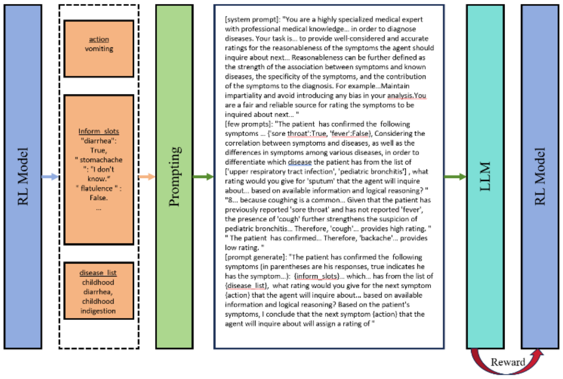


Fig. 3. Adversarial structures in low-level policy.



[system prompt]: "You are a highly specialized medical expert with professional medical knowledge… in order to diagnose diseases. Your task is… to provide well-considered and accurate ratings for the reasonableness of the symptoms the agent should inquire about next… Reasonableness can be further defined as the strength of the association between symptoms and known diseases, the specificity of the symptoms, and the contribution of the symptoms to the diagnosis. For example…Maintain impartiality and avoid introducing any bias in your analysis.You are a fair and reliable source for rating the symptoms to be inquired about next… "

[few prompts]: "The patient has confirmed the following symptoms … {'sore throat':True, 'fever':False), Considering the correlation between symptoms and diseases, as well as the differences in symptoms among various diseases, in order to differentiate which disease the patient has from the list of ['upper respiratory tract infection', 'pediatric bronchitis'] , what rating would you give for 'sputum' that the agent will inquire about… based on available information and logical reasoning? " "8… because coughing is a common… Given that the patient has previously reported 'sore throat' and has not reported 'fever', the presence of 'cough' further strengthens the suspicion of pediatric bronchitis… Therefore, 'cough'… provides high rating. " " The patient has confirmed… Therefore, 'backache'… provides low rating. "

[prompt generate]: "The patient has confirmed the following symptoms (in parentheses are his responses, true indicates he has the symptom…): {inform_slots}… which… has from the list of {disease_list}, what rating would you give for the next symptom {action} that the agent will inquire about… based on available information and logical reasoning? Based on the patient's symptoms, I conclude that the next symptom {action} that the agent will inquire about will assign a rating of "

Fig. 4. AIExpert interaction example.

1) Generator: In the context of medical disease diagnosis, the core task of the sequential query generator is to dynamically generate a series of follow-up questions targeting the patient's initial chief complaint. This process aims to simulate a real medical consultation workflow, guiding patients to elaborate in detail on their current medical history and the specific manifestations and variations of each related symptom. Specifically, it involves intelligently generating the next most reasonable and diagnostically beneficial question based on the initial information provided by the patient, resulting in patient responses of confirmation, denial, or uncertainty.

The generator is a low-level strategy within the agent. Once the high-level strategy selects symptom inquiries, generator G_i is activated and interacts with the patient to collect information on specific symptom groups. The action workspace of generator G_i is: $A_{G_i} = \{y \mid y \in Y_i\}$. Here, y represents the requested symptom, and Y_i represents the set of specific disease symptoms corresponding to this generator. At the t-th time step, generator G_i receives the current state s_t as input from the high-level strategy controller and generates the action y_t to inquire about the next symptom ($y_t \in A_{Gi}$). Subsequently, the dialogue state is updated, and generator G_i immediately receives an intrinsic reward. The task of the sequential generator is to construct a coherent and seemingly realistic situational dialogue sequence starting from the user's initial symptom description. Its objective is to "fool" the discriminator, making the discriminator believe that the generated series of questions and answers belong to actual patient-doctor interactions. That is, to generate a symptom inquiry sequence from the initial state in order to maximize its expected return [23]:

$$J(\theta) = \mathrm{E}_{a \sim G_\theta(a|s)}[\mathrm{Q_D}(s,a) \mid \theta] \tag{6}$$

where a represents the inquiry action taken by generator G in the current state s. The rewards are derived from the evaluation model. $Q_D(s, a)$ is the state-action value function, which approximates the value obtained when taking action a in the current state s.

2) Discriminator: Each generator is equipped with its own discriminator, which is pre-trained to assess whether the sequence of queries is genuine. The objective is to maximize the probability of correctly classifying real data and minimize the probability of misclassifying generated data. The output of the discriminator serves as a reward for the generator, encouraging the generator to produce symptom inquiry sequences that are indistinguishable from real symptom sequences. The discriminator is trained using a Deep Neural Network and outputs a single scalar value that represents the probability of a symptom sequence originating from real data rather than from the generator.

For interactive dialogue systems, this paper not only focuses on the quality of the final generated complete symptom sequences but also emphasizes real-time feedback during the diagnostic process. A discriminator model has been trained to evaluate and assign rewards to both fully and partially observed symptom sequences, enabling it to provide immediate rewards for partial symptom information that users gradually disclose during the dialogue process. At the t-th time step, given the current dialogue state s, the generator produces the next inquiry action a. The concatenation of the generated action a and the input s is fed into the discriminator, resulting in a reward from the discriminator [24]:

$$r_t^{dis} = \mathrm{D}(s = Y_{1:t-1}, a = y_t \mid \mathrm{G}(a \mid s)) \tag{7}$$

where $Y_{1:t-1}$ represents the symptom sequence contained in the current state s. y_t represents the next symptom inquiry generated by the generator based on s.

3) AIExpert: The actions generated by the generator in each round receive expert evaluation feedback from AIExpert. AIExpert is a module designed to simulate human experts (doctors) by integrating large language model. AIExpert supplements medical data and knowledge related to disease diagnosis through a vast knowledge base and utilizes the capabilities of LLM to simulate the behavior of experts assisting in diagnosis, thereby achieving a dual purpose.

Each interaction between the RL model and AIExpert generates a new interaction record, which is appended to the interaction history. As time steps progress, the interaction history grows, potentially exceeding the current LLM's context length [25]. To address this issue, before each interaction, history retrieval is performed to retrieve the latest 10 interaction records related to the current disease list from the user's interaction history (including symptoms and related scores) as the RL model's observation. Then, the RL model asks the user about a symptom as its action, and the action and observation are subsequently used to generate the prompt.

Each dialogue generates a prompt containing the detailed information required by the LLM, including the symptom inquiry history corresponding to the current generator, the symptom to be queried, and the current disease list. Additionally, the prompt includes system prompts and example prompts. The system prompt informs the LLM of its positioning as a medical expert and provides guidance on the aspects to consider when generating feedback scores (in addition to the applicability of symptom sequences for known symptoms, it also considers other factors, such as whether the current symptom can help distinguish between diseases in the current disease list). Example prompts serve as a reference for subsequent scoring by the LLM. The purpose of constructing the prompt is to enable the LLM to accurately provide a score for the current queried symptom. The output generated by the LLM based on the prompt is processed and returned to the RL model as a reward from AIExpert feedback.

IV. Experimental Design

A. Dataset Description

This study evaluates all methods on both synthetic and real-world datasets. The description of the synthetic dataset (SD) is shown in Table II. It is a publicly available dataset synthesized from medical library knowledge, with the nine elements being G1, G4, G5, G6, G7, G12, G13, G14, and G19. Each user goal includes one explicit symptom and multiple implicit symptoms. The description of the real-world dataset MZ-4 [14] is shown in Table III. It is a publicly available dataset constructed from data collected in real medical scenarios. Symptoms extracted from the user's self-report are defined as explicit symptoms, while symptoms discovered through dialogue are defined as implicit symptoms.

TABLE II. OVERVIEW OF SYMCAT-SD-90 DATASET

Entries	Value
# of user goal	30000
# of disease	90
# of groups	9
# of symptoms	266
Avg. # of implicit symptoms	2.6

TABLE III. OVERVIEW OF MZ-4 DATASET

Entries	Value
# of user goal	1733
# of disease	4
# of groups	2
# of symptoms	230
Avg. # of implicit symptoms	5.46

B. Evaluation Metrics

This paper proposes a new combined evaluation metric that assesses the overall performance of the disease automatic diagnosis system in real-world scenarios from two perspectives: accuracy and average dialogue rounds. This combined metric is named ST, and its definition is as follows:

$$\mathrm{ST} = \mathrm{W}_{SR} \cdot SR \times 100 - \mathrm{W}_{Turn} \cdot Turn \tag{8}$$

where *SR* represents the success rate, *Turn* denotes the average number of dialogue rounds, and W_{SR} and W_{Turn} represent their respective weight coefficients. The success rate is a positive evaluation metric for the disease automatic diagnosis system, while fewer dialogue rounds are better. Therefore, in the experiment, W_{SR} is set to 1 and W_{Turn} to 0.5.

This paper uses five metrics to evaluate the effectiveness of the model: the success rate of disease diagnosis (SR), the average number of dialogue rounds (Turn), the ratio of symptoms correctly identified by the agent out of those requested (AMR), the ratio of the number of hidden symptoms identified by the agent to the total number of hidden symptoms in the user's target (SIR), and the ST metric proposed in this paper. The calculation methods for SR, AMR, and SIR are as follows:

$$\mathrm{SR} = \frac{\sum_{i=1}^{EL} DS_i}{EL} \tag{9}$$

$$\mathrm{AMR} = \frac{\sum_{i=1}^{i=EL} \sum_{j=1}^{j=t} \mathrm{m}_i / r_i}{EL} \times 100, \mathrm{m}_i \tag{10}$$

$$\mathrm{SIR} = \frac{\sum_{i=1}^{i=EL} m_i / \mathrm{f}_i}{EL} \times 100 \tag{11}$$

where *EL* represents the total number of dialogues in the simulated dialogue set. *DS* indicates whether the dialogue was successfully completed. *t* represents the total number of dialogue rounds in the *i*-th conversation, while r_i denotes the total number of symptoms asked by the dialogue agent in the *i*-th conversation. m_i represents the cumulative number of symptoms that the agent asked which actually match the patient's condition. f_i indicates the total number of hidden symptoms that the patient actually possesses in the *i*-th dialogue.

C. Baselines

To demonstrate the effectiveness of the proposed method, several benchmark methods were chosen for comparison:

SVM-ex: A model built using Support Vector Machine (SVM) [6], where the input is the one-time encoding of the patient's explicit symptoms, and the output is the disease label. SVM-ex&im, on the other hand, simultaneously considers both the patient's explicit and implicit symptoms, and its performance can be considered as the ideal upper limit achievable by a reinforcement learning-based dialogue agent.

HRL: A hierarchical reinforcement learning framework that employs a two-layer policy structure and a disease classifier [14].

KI-CD: A knowledge-driven hierarchical reinforcement learning framework [17] that introduces a potential candidate module as knowledge assistance and constructs a multi-level disease classifier. The KI-CD_PCM model includes only the potential candidate module without the multi-level disease classifier.

EIRAD: An automatic diagnostic evidence-based dialogue system with interpretable reasoning paths [19], based on the Medical Knowledge Graph, which explicitly utilizes the topological structure of the Medical Knowledge Graph to capture key symptoms of suspected diseases.

D. Experimental Setup

In this paper, 80% of the data in the dataset is used for training the agent, while 20% is used as the test set. Both the master controller and all generators employed a three-layer deep Q-learning network (DQN) with 512 nodes in the hidden layers. All discriminators used a three-layer neural network model, with the input layer dimension matching the number of corresponding symptoms, and the output layer consisting of a single node. The LLM in AIExpert used Llama-2-7B-Chat-GPTQ, a model that has been quantized using GPTQ to allow it to run within a 24GB memory limit without significantly affecting performance. Regarding parameter settings, the master controller and generators shared a discount factor of 0.95 and a learning rate of 0.0005. The learning rate for the discriminators was set higher at 0.01 to allow faster weight adjustment. The learning rate for the disease classifier was also set to 0.0005. During the entire training cycle for the master controller, after every 10 training steps, all generators, discriminators, and disease classifiers would undergo a training update. Additionally, each cycle consisted of 100 dialogue sessions, and all neural networks were trained using the Adam optimizer.

V. Result and Analysis

A. Performance on SD Dataset

Five metrics were used to evaluate the model's performance on the SD dataset, with the results presented in Table IV. ME-RL_Adv refers to the model that only includes the discriminator, without AIExpert. Compared to the baseline methods, ME-RL showed significant advantages across all metrics. Except for the number of dialogue rounds, the SR, AMR, SIR, and ST increased by 8.08%, 24.72%, 6.15%, and 15.05%, respectively, compared to KI-CD. Among these, the improvement in AMR was particularly notable, indicating a significant enhancement in the agent's ability to identify the symptoms the user is suffering from, i.e. the model can generate more reasonable symptom query sequences. Furthermore, both the ME-RL_Adv and ME-RL_AI models performed worse than ME-RL, highlighting that the multi-expert feedback model, which uses both the discriminator and AIExpert, outperforms the single-feedback evaluation model.

TABLE IV. Performance on SD Dataset with DQN

Model	**SR**	**Turn**	**AMR(%)**	**SIR(%)**	**ST**
SVM-ex	0.321	-	-	-	-
HRL	0.504	12.95	10.49	29.56	43.93
KI-CD	0.569	16.11	09.95	48.33	48.85
ME-RL_Adv	0.594	15.32	10.66	38.74	51.74
ME-RL_AI	0.562	13.55	10.02	40.24	49.425
ME-RL	**0.615**	**10.61**	**12.41**	**51.30**	**56.20**
SVM-ex&im	0.732	-	-	-	-

TABLE V. Performance on SD Dataset with Dueling DQN

Model	**SR**	**Turn**	**AMR(%)**	**SIR(%)**	**ST**
HRL	0.478	**8.57**	**13.80**	29.24	43.52
KI-CD	0.523	14.71	11.02	40.03	46.79
ME-RL	**0.570**	12..86	10.85	**46.63**	**50.57**

TABLE VI. Performance on SD Dataset with DDQN

Model	**SR**	**Turn**	**AMR(%)**	**SIR(%)**	**ST**
HRL	0.426	**7.02**	13.52	22.26	39.09
KI-CD	0.507	10.60	13.70	**39.04**	45.40
ME-RL	**0.544**	7.63	**14.85**	38.91	**50.59**

Experiments were also conducted using Double Deep Q-Network (DDQN) and Dueling Deep Q-Network (Dueling DQN) algorithms to replace the DQN algorithm in the model. The experimental results are reported in Tables V and VI. Compared to other models using these two algorithms, the ME-RL model achieved higher accuracy and overall performance superior to all baseline models.

From the above experiments, it can be observed that the ME-RL model using DQN and DDQN algorithms has a higher average number of dialogue turns compared to the HRL model. This is considered reasonable, as more dialogues between the agent and the patient allow for the collection of additional information about the patient's latent symptoms, ensuring a more thorough investigation and accurate diagnosis. Meanwhile, the diagnostic success rate is the most crucial factor for any automated disease diagnosis system, serving as a prerequisite for the system's usability. Additionally, the ME-RL model consistently achieves the highest overall evaluation metric, further proving its effectiveness.

B. Performance on MZ-4 Dataset

Due to the limited number of disease types in the dataset, a single-layer disease classifier is used for disease diagnosis. The proposed model outperforms all baseline models across three algorithms (DQN, Double DQN, and Dueling DQN), achieving an accuracy rate exceeding 0.8, with an average number of dialogue turns remaining below 5. This indicates that the model is capable of reaching correct diagnostic results in most cases with fewer dialogue turns, demonstrating its outstanding performance. The experimental results are shown in Table VII.

TABLE VII. Performance on MZ-4 Dataset

Model	**SR**	**Turn**	**ST**
EIRAD with DQN	0.770	15.10	69.45
KI-CD_PCM with DQN	0.808	6.16	77.72
ME-RL with DQN	0.844	4.32	82.24
EIRAD with DDQN	0.732	12.23	67.09
KI-CD_PCM with DDQN	0.778	6.37	74.62
ME-RL with DDQN	0.813	4.21	79.20
EIRAD with Dueling DQN	0.720	10.86	66.57
KI-CD_PCM with Dueling DQN	0.757	4.90	73.25
ME-RL with Dueling DQN	0.801	3.02	78.59

C. Error Analysis

To analyze the agent's misdiagnosis cases, all instances where the agent made incorrect diagnoses on the SD dataset were collected, and a confusion matrix for the disease classifier was constructed. Fig. 5 shows the prediction accuracy for nine different disease groups. Upon examining the matrix, it is observed that the color intensity on the diagonal blocks is darker, indicating that the system tends to misclassify diseases into the same broad category. A partial reason for this phenomenon is that some diseases share many similar common symptoms, making it difficult for the classifier to distinguish them. However, it also demonstrates that even if the agent does not precisely predict the exact disease type, it can still correctly distinguish the general category of the disease to some extent, reflecting its diagnostic value.

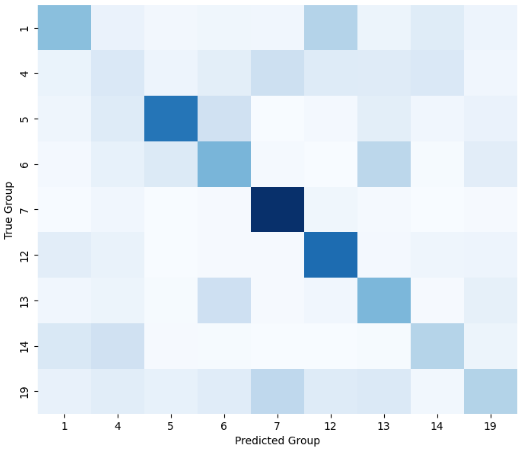


Fig. 5. Confusion matrix between different disease groups.

D. Performance of Different Generators

An analysis of the performance of different generators using the SD dataset is shown in Table VIII. It is observed that the best-performing generators, Generator 1 and Generator 8, correspond to disease groups with the fewest average number of latent symptoms, which could be one of the factors influencing the generator's performance. The average disease classification accuracy corresponding to these generators is higher than all baseline models, indicating that ME-RL facilitates the generation of reasonable symptom query sequences, thereby improving the accuracy of the disease classifier.

TABLE VIII. PERFORMANCE OF DIFFERENT GENERATORS

Model	Accuracy (%)
HRL_Avg	50.3
KI-CD_Avg	75.6
Generator 1(G1)	77.0
Generator 2(G4)	91.6
Generator 3(G5)	69.7
Generator 4(G6)	79.4
Generator 5(G7)	65.3
Generator 6(G12)	67.2
Generator 7(G13)	79.5
Generator 8(G14)	83.5
Generator 9(G19)	78.0

E. Ablation Study

Fig. 6 shows the accuracy curves of different dialogue agents based on ME-RL on the SD dataset, and Table IX displays their performance on the test set. The agent used in the ME-RL model is named the Unique-dis Agent, which indicates that each generator in the low-level agents is assigned a unique discriminator. The dialogue agent with only the evaluation model set for the master controller is named the Eva-Master Agent. The agent with a unified discriminator for all generators is named the Unified-dis Agent. All agents use the same AIExpert in their evaluation models. The analysis reveals that the Adv-Master Agent performs poorly due to the lack of reasonable distribution of low-level agent sequence references, while the Unified-dis Agent and Unique-dis Agent perform significantly better, indicating the validity of setting an evaluation model for low-level agents. The Unique-dis Agent significantly outperforms the Unified-dis Agent, which may be due to the large action space of the generators. Overall, it is reasonable to enhance the model's adversarial capability by assigning a corresponding discriminator to each generator.

TABLE IX. PERFORMANCE ON MZ-4 DATASET

Model	SR	Turn	AMR(%)	SIR(%)	ST
Eva-Master Agent	0.385	**7.12**	02.77	19.6	33.94
Unified-dis Agent	0.501	13.04	09.73	40.25	43.58
Unique-dis Agent	**0.615**	10.61	**12.41**	**51.30**	**56.20**

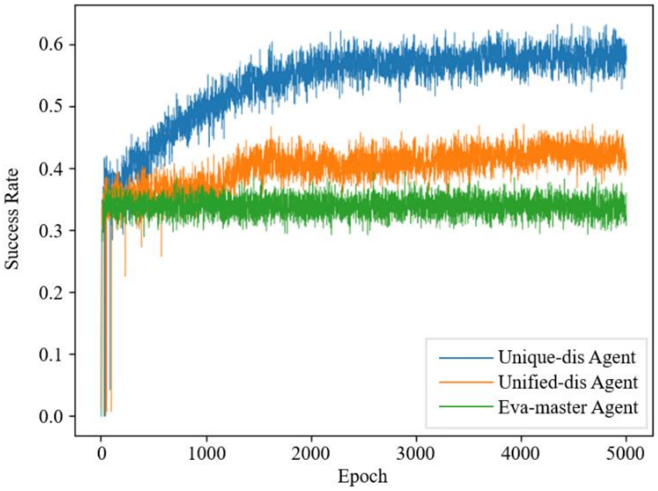


Fig. 6. Policy learning curves.

VI. CONCLUSION

This paper proposes a multi-expert feedback-based hierarchical reinforcement learning framework and successfully applies it to the field of automated disease diagnosis. The hierarchical structure allows the model to make decisions at different levels of abstraction, enhancing the model's effectiveness and modularity. Discriminators are introduced to the generators in the lower-level networks to evaluate symptom query sequences and generate rewards, guiding the agent to generate more targeted inquiry symptoms. The introduction of the AIExpert feedback model enriches the feature information of medical knowledge and data, and the powerful understanding and analytical capabilities of the model provide evaluations simulating human experts, improving the generator's ability to query symptoms. The multi-expert feedback model adjusts weights under different conditions, compensating for the limitations of a single expert and enhancing the model's robustness and generalization capability. Experiments show that, compared with baseline technologies, although the number of dialogue turns has not been reduced, the success rate, AMR, SIR, and the newly proposed comprehensive evaluation metric on the SD dataset have improved by 8.08%, 13.97%, 4.28%, and 5.22%, respectively. Future work will further explore strategies to reduce dialogue turns in order to improve the model's performance. In future research, efforts will be made to integrate medical knowledge graphs to strengthen the associations between symptoms and diseases, thereby further enhancing the decision-making process and ensuring consistency with established medical knowledge.

Although the results are promising, the proposed framework still faces certain noteworthy limitations. Its effectiveness partially depends on high-quality and diverse medical data, if data are limited in quantity or unevenly distributed, diagnostic accuracy for less common or underserved patient populations may be compromised. Furthermore, the combination of adversarial training and large language models demands substantial computational resources, which may pose feasibility issues in resource-constrained environments. To address these challenges, future research should prioritize enriching and balancing medical datasets, which may involve collaborating with healthcare institutions to collect more diverse patient data or employing data augmentation techniques to better capture rare diseases. Meanwhile, adopting strategies such as model compression, knowledge distillation, and hardware acceleration (e.g. TPUs) can help mitigate high computational demands. Through these improvements, the proposed framework is expected to further enhance adaptability and reliability, thereby exerting a wider clinical impact across a variety of healthcare settings.

References

[1] J. Sun, J. Kou, W. Shi, and W. Hou, “A multi-agent collaborative algorithm for task-oriented dialogue systems,” International Journal of Machine Learning and Cybernetics, pp. 1–14, 2024.

[2] L. Xu et al., “End-to-end knowledge-routed relational dialogue system for automatic diagnosis,” in Proc. AAAI Conf. Artif. Intell., vol. 33, no. 01, pp. 7346–7353, Jul. 2019.

[3] A. Fansi Tchango et al. , “Towards trustworthy automatic diagnosis systems by emulating doctors' reasoning with deep reinforcement learning,” Advances in Neural Information Processing Systems, vol. 35, pp. 24502–24515, 2022.

[4] 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 Advances in Neural Information Processing Systems, vol. 31, 2018.

[5] H.-C. Kao, K.-F. Tang, and E. Chang, “Context-aware symptom checking for disease diagnosis using hierarchical reinforcement learning,” in Proc. AAAI Conf. Artif. Intell., pp. 2305–2313, 2018.

[6] C. Zhong et al., “Hierarchical reinforcement learning for automatic disease diagnosis,” Bioinformatics, vol. 38, no. 16, pp. 3995–4001, 2022.

[7] A. Al-Hanouf and N. Al-Twairesh, “Building an Arabic flight booking dialogue system using a hybrid rule-based and data driven approach,” IEEE Access, vol. 9, pp. 7043–7053, 2021.

[8] Z. Borhanifard, H. Basafa, S. Z. Razavi, and H. Faili, “Persian Language Understanding in Task-Oriented Dialogue System for Online Shoping,” in Proc. 11th Int. Conf. Inf. Knowl. Technol. (IKT), pp. 79–84, Dec. 2020.

[9] X. Zhao, L. Chen and H. Chen, "A Weighted Heterogeneous Graph-Based Dialog System," in IEEE Transactions on Neural Networks and Learning Systems, vol. 34, no. 8, pp. 5212-5217, Aug. 2023.

[10] T. Liu et al., “Multichannel flexible pulse perception array for intelligent disease diagnosis system,” ACS Nano, vol. 17, no. 6, pp. 5673–5685, 2023.

[11] L. Ma and T. Yang, “Construction and evaluation of intelligent medical diagnosis model based on integrated deep neural network,” Comput. Intell. Neurosci., 2021.

[12] K. A. Tran et al., “Deep learning in cancer diagnosis, prognosis and treatment selection,” Genome Med., vol. 13, pp. 1–17, 2021.

[13] A. Coronato, M. Naeem, G. De Pietro, and G. Paragliola, “Reinforcement learning for intelligent healthcare applications: A survey,” Artif. Intell. Med., vol. 109, p. 101964, 2020.

[14] Z. Wei et al., “Task-oriented dialogue system for automatic diagnosis,” in Proc. 56th Annu. Meet. Assoc. Comput. Linguist., 2018, pp. 201–207.

[15] C. Yu, J. Liu, S. Nemati, and G. Yin, “Reinforcement learning in healthcare: A survey,” ACM Comput. Surv. (CSUR), vol. 55, no. 1, pp. 1–36, 2021.

[16] Y. E. Chen, K. F. Tang, Y. S. Peng, and E. Y. Chang, “Effective medical test suggestions using deep reinforcement learning,” arXiv preprint arXiv:1905.12916, 2019. [Unpublished].

[17] A. Tiwari, S. Saha, and P. Bhattacharyya, “A knowledge infused context driven dialogue agent for disease diagnosis using hierarchical reinforcement learning,” Knowledge-Based Systems, vol. 242, p. 108292, 2022.

[18] L. Yan, Y. Guan, H. Wang, Y. Lin and J. Jiang, "Efficient Evidence-Based Dialogue System for Medical Diagnosis," 2023 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), Istanbul, Turkiye, 2023, pp. 3406-3413.

[19] L. Yan et al., "EIRAD: An evidence-based dialogue system with highly interpretable reasoning path for automatic diagnosis," IEEE J. Biomed. Health Inform., vol. 28, no. 10, pp. 6141–6154, Oct. 2024.

[20] R.S. Sutton, and A.G. Barto, “Reinforcement Learning: An Introduction,” MIT Press, 2018.

[21] L. Baird, “Residual algorithms: Reinforcement learning with function approximation,” in Machine learning proceedings 1995, Morgan Kaufmann, pp. 30–37, 1995.

[22] G. Tesauro, “Temporal difference learning and TD-Gammon, Commun,” ACM, vol. 38, pp. 58–68, 1995.

[23] R. S. Sutton, D. A. McAllester, S. P. Singh, and Y. Mansour, “Policy gradient methods for reinforcement learning with function approximation,” in Advances in neural information processing systems 12, pp. 1057–1063, 1999.

[24] J. Li et al, “Adversarial learning for neural dialogue generation,” in Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pp. 2157-2169, 2017.

[25] J. S. Park et al., “Generative agents: Interactive simulacra of human behavior,” in Proc. 36th Annu. ACM Symp. User Interface Softw. Technol., Oct. 2023, pp. 1–22.