


## Article

# A Knowledge-Enhanced Hierarchical Reinforcement Learning-Based Dialogue System for Automatic Disease Diagnosis

Ying Zhu <sup>1,†</sup>, Yameng Li <sup>1,†</sup>, Yuan Cui <sup>2</sup>, Tianbao Zhang <sup>1</sup>, Daling Wang <sup>1</sup>, Yifei Zhang <sup>1</sup>  and Shi Feng <sup>1,\*</sup>

<sup>1</sup> Department of Computer Science, Northeastern University, Shenyang 110169, China; zhuying\_neu@163.com (Y.Z.); li\_yameng@foxmail.com (Y.L.); tianbao\_zhang@163.com (T.Z.); wangdaling@cse.neu.edu.cn (D.W.); zhangyifei@cse.neu.edu.cn (Y.Z.)

<sup>2</sup> Business Administration Department, Shenyang Polytechnic College, Shenyang 110045, China; cyuan401@163.com

\* Correspondence: fengshi@cse.neu.edu.cn

† These authors contributed equally to this work.

**Abstract:** Deep Reinforcement Learning is a key technology for the diagnosis-oriented medical dialogue system, determining the type of disease according to the patient's utterances. The existing dialogue models for disease diagnosis cannot achieve good performance due to the large number of symptoms and diseases. In this paper, we propose a knowledge-enhanced hierarchical reinforcement learning model for strategy learning in the medical dialogue system for disease diagnosis. Our hierarchical strategy alleviates the problem of a large action space in reinforcement learning. In addition, the knowledge enhancement module integrates a learnable disease–symptom relationship matrix and medical knowledge graph into the hierarchical strategy for higher diagnosis success rate. Our proposed model has been proved to be effective on a medical dialogue dataset for automatic disease diagnosis.

**Keywords:** automatic disease diagnosis; medical dialogue system; hierarchical reinforcement learning; deep Q network; medical knowledge graph



**Citation:** Zhu, Y.; Li, Y.; Cui, Y.; Zhang, T.; Wang, D.; Zhang, Y.; Feng, S. A Knowledge-Enhanced Hierarchical Reinforcement Learning-Based Dialogue System for Automatic Disease Diagnosis.

*Electronics* **2023**, *12*, 4896. <https://doi.org/10.3390/electronics12244896>

Academic Editors: Simeone Marino, Hamido Fujita, Tun-Wen Pai and Andres Hernandez-Matamoros

Received: 19 October 2023

Revised: 24 November 2023

Accepted: 4 December 2023

Published: 5 December 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Intelligent medical technology is gaining more and more attention because of its ability to relieve physicians' work pressure and improve work efficiency, and has achieved excellent results in various fields such as medical text summarization [1–3], medical QA [4–6] and biomedical information extraction [7–9]. At present, machine learning models have been widely used in disease diagnosis. Siddhartha et al. [10] and Finale et al. [11] have achieved promising results on disease recognition using electronic medical records and supervised learning models.

With the development of deep learning techniques, task-oriented dialogue has been widely used in restaurant reservation [12], movie reservation [13] and online shopping [14]. In the medical realm, scholars have proposed dialogue models for automatic disease diagnosis. Specifically, the disease diagnosis is regarded as a Markov decision process, and a dialogue model is employed to collect symptoms through interacting with the patient, and thus reducing the great efforts of building an electronic medical record for each disease. The dialogue system can not only provide convenience for patients, but also provide preliminary diagnosis for doctors' consultation.

Policy learning is the key technology for dialogue-based disease diagnosis, and it is also widely used in task-oriented dialogue systems [15–17]. However, the previous methods have the following drawbacks.

Firstly, most existing models are based on single-layer reinforcement learning strategies that treat all the diseases and their associated symptoms equally. Wei et al. [18] regards

the symptom acquisition process of multiple rounds of consultation between the agent and the patient as a Markov decision process, and utilizes the reinforcement learning algorithm for training. But when the number of diseases and symptoms is too large, the single-layer strategy that mixes symptom inquiry action and disease diagnosis action will lead to an excessively large action space of the agent, which negatively affects the diagnosis success rate.

Secondly, when the agent selects symptoms for inquiry, the existing methods may not pre-classify the possible symptoms in the current state, resulting in more irrelevant symptoms involved.

Thirdly, a few approaches that consider hierarchical reinforcement learning strategies [19,20] propose a hierarchical reinforcement learning model that integrates two-level hierarchical strategy into dialogue strategy learning. The high-level strategy consists of a model called master, which is responsible for triggering the low-level model. The low-level strategy consists of several symptom checkers and a disease classifier. Although the strategy of hierarchical reinforcement learning is adopted, it ignores the medical knowledge and disease–symptom relationships that are closely related to the diagnosis task, which brings in irrelevant symptoms and may harm the success diagnosis rate of disease.

In this paper, we propose a hierarchical reinforcement learning (HRL) model KNHRL that integrates medical knowledge and disease–symptom relations into a dialogue model for disease diagnosis. Compared with the previous HRL model for disease diagnosis, KNHRL incorporates a learnable disease–symptom relation matrix and knowledge graph to assist the agent for decision making. By incorporating co-occurrence probabilities between symptoms, the model can quickly and comprehensively ask for implicit symptoms that are more relevant to known symptom information, rather than asking for irrelevant symptoms. The knowledge of the relationship between disease and symptoms further ensures the accuracy of the diagnosis. Moreover, KNHRL conducts pre-classification before the low-level strategy makes decisions, separating the action of asking about symptoms from the action of diagnosing a disease. This way, the agent can collect symptoms more likely to be associated with the disease that users are suffering from. The major contributions of this paper can be summarized as follows.

- We incorporate the learned medical knowledge into the low-level strategy of an HRL model, which can further improve the symptom matching rate and the diagnosis success rate.
- Inspired by the process of the doctor’s consultation in real life, we leverage a classifier to feed the user’s disease probabilities into system states, and propose a new decision-making method by considering the medical knowledge graph and the learned disease–symptom relation matrix.
- The proposed KNHRL model outperforms strong baseline methods on a public available medical dialogue dataset for automatic disease diagnosis.

## 2. Related Work

Hierarchical reinforcement learning (HRL) methods are employed to decompose a huge action space, and have been applied in visual navigation, natural language processing, recommendation systems, video description generation and other daily life domains [21–28]. Jain et al. [29], for a four-legged robot path tracking task, took full advantage of the hierarchical structure features and timing decoupling scheme of HRL to use different state representations for the upper and lower controllers. The model emphasized the different concerns of position estimation and motion control to ensure the reusability of the lower layer strategies. Li et al. [30] in a multi-goal-oriented task for an 18-degree-of-freedom robot, pre-trained skills to obtain skills that could achieve simple goals, and then planned the learning of the skills.

Budzianowski et al. [31] utilized the strong transfer ability of HRL to build a cross-domain dialogue system, which learned shareable information in similar subdomains of different main domains to train a general underlying policy. Saha et al. [32,33] leveraged

the HRL framework to learn a multi-intent dialogue policy. The proposed algorithm introduced emotion-based instant rewards into the basic rewards of the dialogue system, making the question-answering robot adaptive so as to obtain maximum user satisfaction. Saleh et al. [34] devised a variational sequence model, which no longer simply considered word-level information, but built a reward model at the discourse level to improve the global vision of the model.

Reinforcement learning (RL) has become the mainstream method for automatic disease diagnosis in dialogues [35–37]. Wei et al. [18] leveraged a DQN from conversations with patients to select additional symptoms, which could greatly improve the accuracy of diagnosis. Hou et al. [36] proposed a multi-level reward RL-based model that could improve both the performance and the speed of convergence. Teixeira et al. [37] customized the settings of the RL leveraging the dialogue data. The existing hierarchical reinforcement learning strategies [19] usually ignore knowledge and disease–symptom relationships that are closely related to the diagnosis task, which has negative impacts on the success diagnosis rate of the disease.

In addition, there is work on knowledge enhancement for diagnosis. Xu et al. [38] proposed a Knowledge Routing Dialogue System, referred to as KR-DS for short, which embedded the rich medical knowledge into topic switching in the dialogue management module to assist agent decision-making. Liu et al. [39] introduced a supervised diagnostic model (mapping between symptoms and diseases) in the external environment, thereby improving the agent’s ability to collect symptoms that were more helpful for diagnosis. However, these models could not effectively incorporate the knowledge-enhanced disease–symptom relation into the HRL models.

### 3. Model Overview

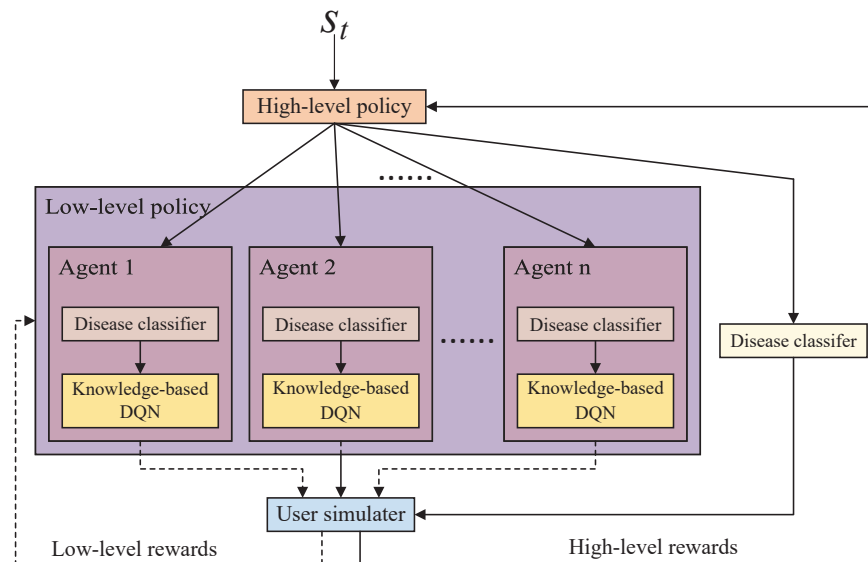
The task of reinforcement learning is to learn how to take actions based on the current environmental state in order to maximize the expected return. As for RL-based models for automatic diagnosis, the action space of agent is  $A = D \cup S$ , where  $D$  is the set of all diseases and  $S$  is the set of all symptoms associated with these diseases. Given the state  $s_t \in \mathcal{S}$  at turn  $t$ , the agent takes an action according to its policy  $a_t \sim \pi(a|s_t)$  and receives an immediate reward  $r_t = R(s_t, a_t)$  from the environment. If  $a_t \in \mathcal{S}$ , the agent chooses a symptom to inquire the user. Then the user responds to the agent with *True/False/Unknown*. If  $a_t \in \mathcal{D}$ , the agent informs the user of the corresponding disease as the diagnosis result and the dialogue session will be terminated, marking the success or failure in terms of the correctness of the diagnosis.

Scholars introduce Markov Decision Process to simplify the model. They assume that the state transition model exhibits the Markov property, meaning the transition of states depends solely on the current state. Consequently, the problem of reinforcement learning can be formulated as a Markov Decision Process.

The disease diagnosis model can be expressed as Markov Decision Process  $M = \langle S, A, R, P, \gamma \rangle$ .  $S = S^h \cup \left\{ S^l_i \right\}_{i=1}^{n_l}$  is a set of all states,  $S^h$  is the status of the agent in the high-level strategy (dubbed as the high-level agent).  $S^l_i$  is the status of the agent in the  $i$ th low-level strategy (dubbed as the low-level agent).  $n_l$  is the number of low-level agents.  $A = A^h \cup \left\{ A^l_i \right\}_{i=1}^{n_l}$  is the set of all actions,  $A^h$  is the high-level agent action,  $A^l_i$  is the  $i$ th low-level agent action, and  $n_l$  is the number of low-level agents.  $R$  is a collection of dialogue rewards. A policy  $\pi$  is a mapping between a state set  $S$  and a state transition model set  $P$ .  $\gamma$  is the discount rate used to compute the Q value function. The goal of the model is to optimize the Markov Decision Process  $M = \langle S, A, R, P, \gamma \rangle$  and find the policy  $\pi$  that maximizes the cumulative discount reward for all  $\langle S, A \rangle$ .

This paper proposes a knowledge-enhanced hierarchical reinforcement learning model KNHRL for the disease diagnosis task. In order to reduce the action space, KNHRL divides the strategies of the disease diagnosis task into two levels, namely high-level strategy and low-level strategy. This idea was inspired by hospital consultations in the real world.

Figure 1 demonstrates the framework of the KNHRL model. The high-level agent receives the current initial state  $s_t$  and selects a low-level agent to talk to the user simulator for symptom collection. The low-level strategy consists of multiple agents, and each agent is responsible for collecting relevant symptoms of different diseases. Each low-level agent consists of a disease classifier and a deep Q network (DQN) with knowledge embeddings.



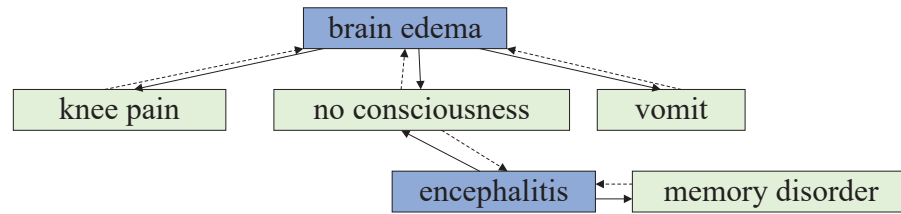
**Figure 1.** The structure of KNHRL.

Considering that when a doctor asks about a patient's symptoms, they will first consider that the patient may have a certain disease, and then ask the related symptoms of the disease. According to this process, before using the knowledge-embedded DQN for decision-making, we first use the disease classifier to obtain the probability distribution of the disease in the low-level agent in the current state, and to assist the subsequent DQN in decision-making. The doctors will combine their own medical experience and knowledge when asking about symptoms, so we add information on past diseases, dependencies between symptoms and disease-symptom knowledge graphs that can be learned during training based on the basic DQN. The above DQN strategy is called DQN for knowledge embedding.

The collection of symptoms by the low-level agent is achieved through a dialogue with a user simulator that gives feedback (*True/False/Unknown*) about the symptoms asked by the agent, and the model rewards the agent based on the feedback from the user simulator, namely low-level reward. The low-level agent then decides whether to continue symptom collection according to the reward obtained. If the low-level agent continues to collect symptoms, it will update the policy and continue to select symptoms to interact with the user simulator. When the low-level agent no longer collects symptoms, the previous low-level rewards are accumulated as the reward of the high-level agent, and the high-level agent updates the policy according to the obtained reward, so as to select the low-level agent to collect symptoms or to choose a disease classifier to make a diagnosis.

#### 4. Knowledge Construction

This paper leverages disease-symptom relation modules and medical knowledge graphs to assist decision-making in low-level policies. The medical knowledge graph is constructed by diseases and their related symptoms, as shown in Figure 2, for example.



**Figure 2.** An excerpt of medical knowledge graph.

In Figure 2, the blue entity is the disease, and the green entity is the symptom. Each edge between the disease entity and the symptom entity contains two weights, which are the symptom probability ( $sym|dis$ ) under the disease condition and the disease probability ( $dis|sym$ ) under the symptom condition. The two probabilities are calculated by occurrences of diseases and symptoms in the dataset, which forms a disease–symptom relation matrix.

The elements in the  $dis\_sym$  matrix are the symptom probability ( $sym|dis$ ) under the disease condition, and the elements in the  $sym\_dis$  matrix are the disease probability ( $dis|sym$ ) under the symptom condition. It is worth noting that the establishment of the medical knowledge graph is based on the disease and its related symptoms that each low-level agent is responsible for. In this paper, a total of nine medical knowledge graphs are established, each of which has a corresponding disease–symptom relation matrix.

We can learn the disease–symptom relation matrix from the dataset. Note that the relation matrix is also built at the unit of each lower-level agent. The disease–symptom relation matrix is a concatenation of the disease–disease matrix (recorded as  $dis\_dis$ ), disease–symptom matrix (recorded as  $dis\_sym$ ), symptoms–disease matrix (recorded as  $sym\_dis$ ), and symptoms–symptom matrix ( $sym\_sym$ ), shown below:

$$matrix_1 = Cat(sym\_sym, dis\_sym) \quad (1)$$

$$matrix_2 = Cat(sym\_dis, dis\_dis) \quad (2)$$

$$relation\_matrix = Cat(matrix_1, matrix_2) \quad (3)$$

Formulas (1) and (2) are spliced on the first dimension, and Formula (3) is spliced on the 0th dimension. Since the diseases in each low-level agent are in the same department, this paper does not consider the relationship between diseases and diseases, that is, the  $dis\_dis$  matrix is set to a 0 matrix of size  $R^{n_{dis} \times n_{dis}}$ .

## 5. Knowledge-Enhanced Hierarchical Reinforcement Learning Model

### 5.1. Deep Reinforcement Learning Model for Disease Diagnosis

The use of DQN-based models for disease diagnosis is one of the most popular methods. In the problem of automatic disease diagnosis, the main elements of the DQN-based model include current state  $s_t$ , strategy  $\pi$ , current action  $a_t$ , and immediate reward  $r_t$ . Among them, the current state  $s_t$  is spliced by the 3-dimensional one-hot vector  $z^i$  of each symptom, and each dimension of the one-hot symptom vector represents the different states of the symptom, where  $z^i = (1, 0, 0)$  means that the patient has the symptom (*True*),  $z^i = (0, 1, 0)$  means the patient does not have the symptom (*False*) and  $z^i = (0, 0, 1)$  means the patient does not know whether the patient has the symptom (*Unknown*). For symptoms not asked by the agent, we denote it as  $z^i = (0, 0, 0)$ . Therefore, the current state  $s_t$  contains not only the information of the current round, but also the action information of the previous agent and the patient and the symptom information that has been collected. According to the described definition, the current state  $s_t$  can be expressed as Formula (4).

$$s_t = [z_t^1, z_t^2, \dots, z_t^{n_s}] \quad (4)$$

where  $n_s$  is the number of symptoms. The policy  $\pi$  is used to describe the action of the agent. When the current state  $s_t$  is known, the policy  $\pi$  can be expressed as  $\pi(a|s_t)$ , which obtains the probability distribution of all possible agent actions in the state  $s_t$ . The current action  $a_t$  is the action of the agent obtained according to the policy ( $a|s_t$ ) under current state  $s_t$ , and the process can be expressed as Formula (5).

$$a_t \sim \pi(a|s_t) \quad (5)$$

The action space  $A$  of the agent is the union of all disease sets  $D$  and their associated symptom sets  $S$ , that is,  $A = D \cup S$ . The instant reward  $r_t$  is the reward obtained from the user simulator when the agent is in a state  $s_t$  and makes an action  $\pi$  according to the strategy  $a_t$ , to update the strategy.

According to the above elements, the process of disease diagnosis using a deep reinforcement learning model is described as follows: in the state of  $s_t$ , the agent selects an action  $a_t$  according to the policy  $\pi$ . Note that the agent follows the  $\varepsilon$  greedy policy when selecting an action; that is, in the case of  $1 - \varepsilon$ , the agent chooses the optimal action; in the case of  $\varepsilon$  the agent chooses the action randomly. When  $a_t \in S$ , the agent will choose a symptom to talk to the user simulator, and the user simulator will give the agent feedback (*True/False/Unknown*) and the corresponding reward. According to the feedback information, the agent assigns the value at the location of the corresponding symptom in  $s_t$ , and updates the strategy to select the next action; when  $a_t \in D$ , the agent will choose a disease to inform the user simulator, and the dialogue will be judged as success or failure according to whether the informed disease is correct or not, and the agent will get different rewards and continue to update the strategy.

The goal of the agent is to find a policy that maximizes the expected cumulative discounted reward (called the optimal policy). The  $Q$  value function is used to calculate the expected reward generated by selecting the action  $a_t$  according to the policy  $\pi$  in the state  $s_t$ . The calculation method is shown in Formula (6):

$$Q^\pi(s_t, a_t|\theta) = r_t + \gamma Q^\pi(s_{t+1}, a_{t+1}|\theta') \quad (6)$$

where the  $Q^\pi(s_{t+1}, a_{t+1}|\theta')$  is the  $Q$  function of the target network,  $\theta$  is the parameter of the current network,  $\theta'$  is the parameter of the target network obtained from the previous iteration, and  $\gamma \in [0, 1]$  is the discount factor. When  $\gamma = 0$ , only the rewards of the current round are considered. When  $\gamma = 1$ , the rewards of the current round and subsequent round are treated equally. When  $\gamma \in (0, 1)$ , the rewards of the current round rewards are more important than the subsequent round. The agent wants to find the policy that maximizes the cumulative discount reward, then the optimal  $Q$  value function  $Q^*$  is the maximum value of the  $Q$  value function under all strategies, namely

$$Q^*(s_t, a_t|\theta) = r_t + \gamma \max_{a_{t+1}} Q^*(s_{t+1}, a_{t+1}|\theta') \quad (7)$$

When the value of the  $Q$  value function obtained for all states and actions under a policy  $\pi^*$  is the largest, then the policy  $\pi^*$  is called the optimal policy.

Notably, the DQN parameterizes the policy so that the policy is updated by training the DQN. Each iteration of DQN takes the current state as input and outputs the computed  $Q$  value of the current network. DQN updates the parameter  $\theta$  at each iteration of training by minimizing the error between the computed  $Q$  value of the current network and the  $Q$  value of the target network, (that is, the  $Q$  value obtained from the Bellman equation), to train the network.

## 5.2. High-Level Strategy for HRL

In the dataset for the dialogue-based disease diagnosis constructed by Liao et al. [19], the diseases are divided into nine subsets based on the department, and each subset



contains ten diseases. The diseases in different subsets are different from each other, and the relationship between each subset is shown as follows:

$$D = D_1 \cup D_2 \cup \dots \cup D_9, \quad D_i \cup D_j = \emptyset \quad (8)$$

$$D_i = D_1 \cup D_2 \cup \dots \cup D_{10} \quad (9)$$

where  $D_i$  represents the disease set in the  $i$ th subset, and  $d_k$  represents the  $k$ th disease in the  $i$ th disease subset. In hierarchical reinforcement learning (HRL), an agent in a low-level policy is responsible for collecting its associated symptoms for each disease subset, and a high-level policy is responsible for selecting which agent in a certain low-level policy to work. The process of the model informing the user simulator disease to make a diagnosis is carried out by a disease classifier that is selected by a high-level policy at the same level as a low-level policy.

According to the task content of the high-level policy, the action space of the high-level agent is shown in Formula (10):

$$A^h = l^1 \cup l^2 \cup \dots \cup l^9 \cup dl \quad (10)$$

where  $l^i$  is the  $i$ th agent in the low-level policy, and  $d$  is the disease classifier. After receiving the current state  $s_t$ , the high-level agent selects an action  $a_t^h$  according to the current policy  $\pi^h$ ,  $a_t^h$  is a 10-dimensional vector (nine for low-level agents and one for a disease classifier) to indicate which low-level agent is selected for symptom collection, or which disease classifiers are selected for disease informing. When the high-level agent triggers the work of a certain low-level agent, the high-level agent will proceed to the next step, only when the low-level agent finishes the work. After the low-level agent finishes working, the rewards received from the user simulator for each round will be accumulated as the reward of the high-level agent, which is called the high-level reward. This is calculated as follows:

$$r_t^h = \begin{cases} \sum_{t'=1}^T \gamma_h t' r_{t+t'}^l & a_t^h = l^i \\ r_t^{dl} & a_t^h = dl \end{cases} \quad (11)$$

$t'$  is the dialogue rounds of the agent in the low-level policy,  $T$  is the total number of dialogue rounds of the agent in the low-level policy,  $\gamma_h$  is the discount factor,  $r_{t+t'}^l$  is the reward that the low-level agent gets from the user simulator in the current round, and  $r_t^{dl}$  is the reward from the user simulator for the disease classifier. The goal of the advanced agent is to maximize the expected cumulative discounted advanced reward. The  $Q$  value function is used to represent the expected reward of the advanced agent. Its Bellman Equation can be written in the form of Formula (12):

$$Q_h^\pi(s_t, a_t^h | \theta^h) = r_t^h + \mathbb{E}_{\{s_{t+1}, a_{t+1}^h\}} \left[ \gamma_h^T Q_h^\pi(s_{t+1}, a_{t+1}^h | \theta^{h'}) \right] \quad (12)$$

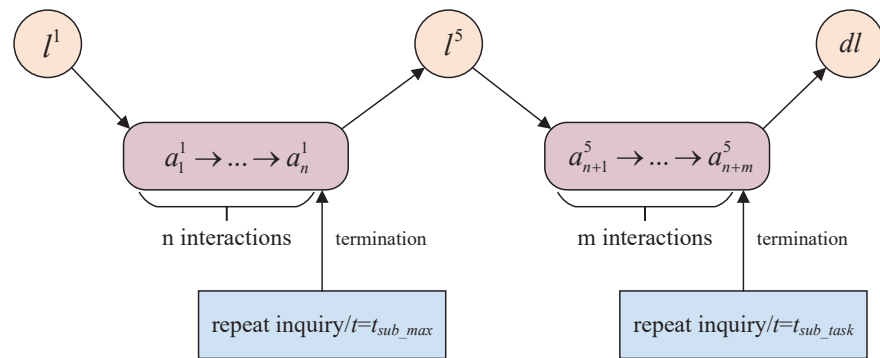
$\theta^h$  is the parameter of the current advanced policy network,  $s_{t+1}$  is the next dialogue state observed by the advanced agent after taking an action  $a_t^h$  according to the policy  $\pi$  in the state of  $s_t$ , and  $a_t^h$  is the action taken by the high-level agent under  $s_{t+1}$ , and  $\gamma_h$  is the discount factor. The high-level policy network consists of a three-layer DQN, and the network parameters  $\theta_h$  are updated during training by reducing the mean squared error between the  $Q$  value calculated in the current network and the  $Q$  value of the target network obtained from the Bellman equation. Therefore, the above mean square error is used as the loss function of the advanced policy network, as shown in Formula (13):

$$L(\theta^h) = \mathbb{E} \left\{ \left[ \left( r_t^h + \gamma_h^T \max_{a_{t+1}^h} Q_h^*(s_{t+1}, a_{t+1}^h | \theta^{h'}) \right) - Q_h^\pi(s_{t+1}, a_{t+1}^h | \theta^{h'}) \right]^2 \right\} \quad (13)$$

The first term in the squared difference is the  $Q$  value of the target network obtained from Bellman equation, and the second term is the calculated  $Q$  value of current network.

### 5.3. Low-Level Strategy for Knowledge Enhanced Decision-Making

The low-level agent is responsible for collecting symptoms by talking to the user simulator, which is triggered by the high-level agent. Figure 3 shows the process in which a low-level agent is selected for work by a high-level agent.  $l^1, l^5$  in Figure 3 as well as  $dl$  is the action of the high-level agent,  $l^1$  and  $l^5$  represent that the high-level agent has selected the first and fifth low-level agents, respectively, and  $dl$  represents that the high-level agent has selected a disease classifier for diagnosis. Taking the working process of the first low-level agent as an example,  $a_k^1$  is the action of the  $k$  conversation of the first low-level agent. When the low-level agent repeatedly asks the same symptom or the number of dialogue rounds reaches the specified upper bound, the low-level agent's work ends. The reward obtained by the low-level agent for each round of dialogue (low-level rewards) are accumulated and returned to the high-level agent, and the high-level agent makes the next selection. The disease set contained in the low-level agent is  $D_i$ , and the associated symptom set  $S_i$  is the action space of the  $i$ th low-level agent.



**Figure 3.** The working process of hierarchical model.

Next, we illustrate how medical knowledge assists low-level agents in decision-making in the low-level strategy of decision-making, as shown in Figure 4.

The  $s_t^i$  in the Figure 4 is the current state extracted by the  $i$ th low-level agent from  $s_t$ . The extraction process proceeds as follows: when the low-level agent  $l_i$  is selected by the high-level agent, the high-level agent will pass the current state  $s_t$  to the low-level agent  $l_i$ .  $l_i$  will extract the corresponding states of these symptoms from  $s_t$  according to the symptoms of the disease they are responsible for, considering it as the current state of the low-level agent. The specific extraction method is shown in Formula (14):

$$S_t^i = \text{Extract}(S_t, i) = [Z_1^i, Z_2^i, \dots, Z_{n_{S_i}}^i] \quad (14)$$

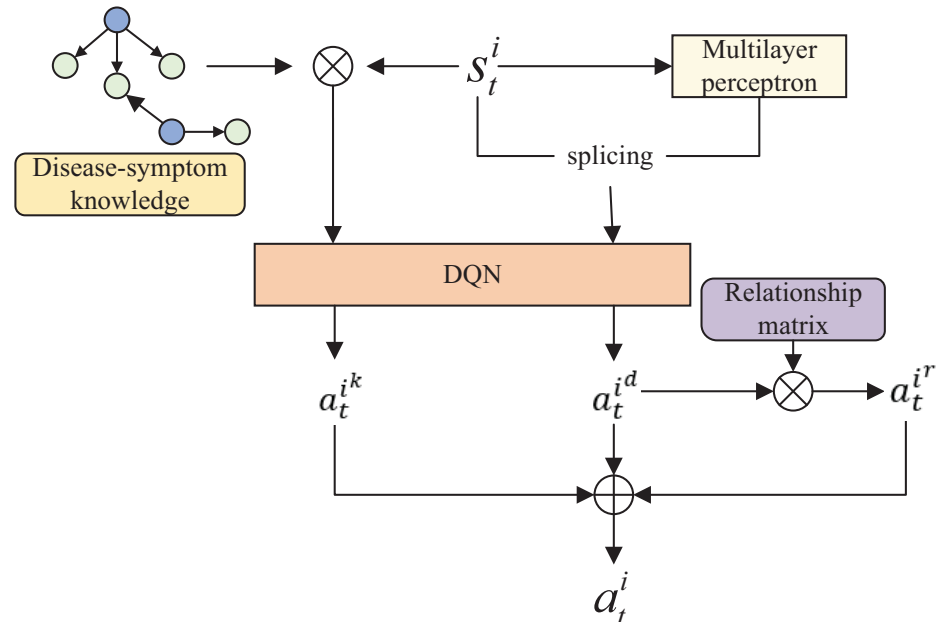
where  $n_{S_i}$  is the number of associated symptoms of the disease in the  $i$ th low-level agent.  $z_k^i$  is the one-hot vector of the  $k$ th symptom in the  $i$ th low-level agent. We hope to preliminarily screen the diseases that patients may have in the current state, so that when collecting symptoms, it is easier for low-level agents to collect symptoms related to possible diseases, and thus improve the success rate of diagnosis. Therefore, we design a disease pre-classification module for each low-level agent, which contains a disease classifier consisting of a two-layer MLP. Specifically, before the DQN makes a decision, the current state is input into the disease classifier, and the disease classifier outputs a 10-dimensional vector  $dl_i$ , which represents the predicted probability fraction of each disease in the agent. We concatenate the output vector  $dl_i$  with the current state  $S_t^i$  according to Formula (15)

$$S_t^{i'} = C(S_t^i, dl_i) \quad (15)$$



$S_t^{i'}$  is the newly obtained current state containing disease information, and the output obtained by inputting it into the DQN is the original action of the low-level agent  $a_t^{id}$  in the current state  $S_t^{i'}$ , as shown in Formula (16).

$$a_t^{id} = \text{MLP}(S_t^{i'}). \quad (16)$$



**Figure 4.** The structure of knowledge-enhanced low-level policy.

We hope to use the relation information between diseases and symptoms in the dialogue history as “experience” to assist decision-making in the current state. Therefore, we design a relation module to capture the “experience” in the dialogue history. The relation module contains a matrix  $R \in R^{A^i \times A^i}$ , where  $A^i = D^i \cup S^i$ , which can learn the relation between each symptom and disease during training. Specifically, the original action  $a_t^{id}$  obtained by the lower-level agent is multiplied by the relation matrix  $R$ , as shown in Formula (17).

$$a_t^{ir} = a_t^{id} \cdot R \cdot a_t^{ir} \quad (17)$$

$a_t^{ir}$  is the action of the low-level agent augmented by the relation matrix, where the elements are the weighted sum of the original action and the relation matrix. The matrix  $R$  is initialized by the relation matrix established in Section 4 *relation\_matrix*, which contains the dependency of diseases and symptoms in the dataset. During model training, the relation matrix  $R$  learns the dependencies between diseases and symptoms during the dialogue between the low-level agent and the user simulator through backpropagation.

We also hope to simulate the real-world situation of doctors combining their own medical knowledge for diagnosis, so a medical knowledge graph module is designed to assist the agent in making decisions. In Section 4, we have established a disease–symptom medical knowledge graph for each low-level agent. When the  $i$ th low-level agent works, the weight matrices  $P^i(dis|sym)$  and  $P^i(sym|dis)$  on each edge of the  $i$ th medical knowledge graph are used to compute the weight matrix for the medical knowledge graph module.

According to the conditional probability, Formulas (18) and (19) can be obtained:

$$P^i(dis) = P^i(dis|sym) \cdot P^i(sym) \quad (18)$$

$$P^i(sym) = P^i(sym|dis) \cdot P^i(dis) \quad (19)$$

where the symptom  $P^i(sym)$  is the final desired weight matrix. Since both  $P^i(dis)$  and  $P^i(sym)$  are unknown, and the prior probability of symptoms can be obtained from the dataset. The disease probability  $P^i(dis)$  is first calculated using the prior probability of symptoms; that is, Formula (18) can be rewritten in the form of Formula (20):

$$P^i(dis) = P^i(dis|sym) \cdot P_{prior}^i(sym) \quad (20)$$

where  $P_{prior}^i(sym) \in R^{n_{s_i}}$  is the prior probability of symptoms, and  $n_{s_i}$  is the number of symptoms corresponding to the  $i$ th low-level agent. For the symptoms that have been collected under the current state  $s_t^i$ , the value of the prior probability of the symptoms that the patient does exist (the response of the user simulator is *True*) is set to 1; the value of the prior probability of the symptoms that the patient does not exist (the response of the user simulator is *False*), the prior probability is set to  $-1$ ; the prior probability of symptoms that the patient does not know exists (the user simulator's response is *Unknown*) is set to the value calculated from the user goals in the dataset. For the symptoms that have not been collected in the current state  $s_t^i$ , the prior probability is also set to the value calculated from the user goals of the dataset. Formula (21) is a method for calculating the prior probability of symptoms from the user goals in the dataset.

$$P_{prior}^i(sym) = \frac{\left[ n(S_{true}^{i,1}), n(S_{true}^{i,2}), \dots, n(S_{true}^{i,n_{s_i}}) \right]}{n_i} \quad (21)$$

where  $n(S_{true}^{i,m})$  is the number of real symptoms in the  $m$ th symptom in the data corresponding to the  $i$ th low-level agent, and  $n_i$  is the number of user targets in the  $i$ th low-level agent. After obtaining the prior probability of the symptom, the symptom probability  $P^i(sym)$  can be obtained by Formulas (19) and (20).

After multiplying the obtained symptom probability by the current state element-wise, it is sent to DQN, as shown in Formula (22).

$$a_t^{i,k} = MLP \left[ s_t^i \odot P^i(sym) \right] \quad (22)$$

where  $\odot$  stands for element-wise multiplication,  $a_t^{i,k}$  is the action selected by the low-level agent after the enhancement of the medical knowledge graph, and the final action of the low-level agent is the sum of the above three actions:

$$a_t^i = a_t^{i,d} + a_t^{i,r} + a_t^{i,k} \quad (23)$$

When the low-level agent makes an action, the user simulator will give a reply and corresponding reward according to the symptoms inquired by the low-level agent, and the dialogue will be updated to the next state. Since the action of the low-level agent is symptom collection, in the process of training and prediction, the index value of the predicted action should be obtained and judged whether it is less than  $n_{s_i}$ . If the predicted action index is not less than  $n_{s_i}$ , the task of the current low-level agent is terminated directly. We call the reward received by the lower-level agent as the lower-level reward. Thus, the goal of the lower-level agent is to find a policy that maximizes the expected cumulative discount of the lower-level reward. The Bellman Equation of the  $i$ th lower-level agent can be expressed as Formula (24):

$$Q_l^i(s_t^i, a_t^i | \theta^i) = r_t^i + \mathbb{E}_{\{s_{t+1}^i, a_{t+1}^i\}} \left[ \gamma_l^i Q_l^i(s_{t+1}^i, a_{t+1}^i | \theta^i) \right] \quad (24)$$

where  $\gamma_l^i$  is the discount factor for the  $i$ th low-level agent. The low-level policy network is a three-layer DQN, and its network parameters  $\theta^i$  are optimized by minimizing the loss function of the network. The mean squared error between the current network of

the low-level policy and the target network is used as the loss function of the network, as shown in Formula (25):

$$L(\theta^i) = \mathbb{E} \left\{ \left[ \left( r_t^i + \gamma \max_{a_{t+1}^i} Q_l^i(s_{t+1}^i, a_{t+1}^i | \theta^i) \right) - Q_l^i(s_t^i, a_t^i | \theta^i) \right]^2 \right\} \quad (25)$$

#### 5.4. User Simulator

The user simulator is the component that talks to the agent and contains the user goals in the dataset. In each simulated dialogue, the user simulator extracts a user target for model training, and the explicit symptoms in the user target are used to initialize the dialogue state. For the symptoms inquired by the agent, the user simulator provides feedback according to the extracted symptom information in the user target: for the symptoms that are *True* in the user target, the user simulator sets the corresponding symptoms in the state to (1,0,0), and gives a +1 reward; for the symptom of *False* in the user's goal, the user simulator will set the corresponding symptom in the state to (0,1,0) and give a −1 reward; for the symptom of *Unknown* in the user's goal and the symptoms that do not exist in the user target, the user simulator sets the corresponding symptoms in the state to (0,0,1) and gives a 0 reward. Notably, when the user simulator receives symptoms that the agent has already asked about, or the maximum number of dialogue turns with the agent is reached, a −2 reward is given and the dialogue with the agent is ended. For the disease notified by the agent, when the value of the disease label in the user target extracted by the user simulator is the same, the diagnosis is determined to be successful and a +22 reward is given; otherwise, it is determined to be a failure and a −44 reward is given.

## 6. Experiments and Analysis

### 6.1. Experimental Data and Settings

We select the artificially synthesized dialogue dataset for the disease diagnosis proposed by Liao et al. [19]. It contains user goals based on patient self-descriptions and conversations with physicians. This synthetic dataset is based on the SymCat database, which contains disease–symptom relations. From the 21 groups (departments) of diseases and their related symptoms classified according to the International Classification of Diseases, nine groups of the most representative diseases were selected and used to generate user goals for disease diagnosis. Each department selects the top 10 diseases with the highest incidence rate in the department.

This paper utilizes the experience playback mechanism [20] to train the high-level policy network and the low-level policy network. Specifically, during training, the “experience” of the high-level policy network  $(s_t, a_t^h, r_t^h, s_{t+1})$  and the low-level policy network  $(s_t, a_t^l, r_t^l, s_{t+1})$  are put into their respective buffers  $B^h$  and  $B^l$ . The capacities of the two buffers is fixed, and each round of training is to extract *mini\_batch* “experiences” from the buffer. The current network will be evaluated after each round of training, and when the performance of the current network is the best, the buffer is flushed. Note that the high-level policy network and the low-level policy network are not trained synchronously. The low-level policy is trained once for every 10 rounds the high-level policy network is trained.

### 6.2. Baseline Models

**Flat-DQN** is a model of a single-layer strategy proposed by Wei et al. [18], which treats all diseases and their related symptoms equally; **KR-DQN** [38] also treats all diseases and their associated symptoms equally; **REFUEL** is a single-layer policy reinforcement learning model combining reward remodeling and positive remodeling mechanisms proposed by Peng et al. [35]; **GAMP** [40] is a single-layer reinforcement learning model optimized by the policy gradient framework of generative adversarial networks; **HRL-pretrained** [41] is a hierarchical reinforcement learning model that pre-trains low-level policies and then

trains high-level policies; **HRL** [19] is a hierarchical reinforcement learning model, which utilizes a disease classifier to separate symptom collection

The above baselines are the reinforcement learning model for disease diagnosis. In this paper, SVM is selected as the multi-class classification baseline model, and two experiments are designed based on the SVM model, namely **SVM-ex** trained only with explicit symptoms, and **SVM-ex-im** trained with explicit symptoms and implicit symptoms at the same time. Since the deep reinforcement learning models for disease diagnosis-oriented dialogues all initialize states with overt symptoms, the results of multi-classification models SVM-ex-im trained with both explicit and implicit symptoms can be used as an upper bound on the performance of deep reinforcement learning models on the synthetic dialogue datasets.

### 6.3. Experimental Results and Analysis

We select success rate, average number of dialogue turns and matching rate as the metrics to evaluate the performance of the models. Each session between the agent and the user simulator ends with the agent notifying the user simulator of the disease. If the notified disease is consistent with the disease label of the user target in the user simulator, the session is recorded as a success. The success rate is the ratio of the number of successful sessions to the total number of sessions. Average dialogue turns are the average number of turns in the session. The matching rate is the symptom matching rate, which is calculated as the ratio of the number of implicit symptoms in the user target inquired by the agent to the total number of symptoms inquired in a session.

In Table 1, The results of the KR-DQN, REFUEL and GAMP models are reproduced on this dataset. For the rest of the baselines, we adopt the reported results from the related papers published in recent years [40]. Note that the results for the KNHRL model are the average of the results obtained from three experiments with the same experimental settings on this dataset.

**Table 1.** Evaluation results of KNHRL and other baselines on synthetic dataset.

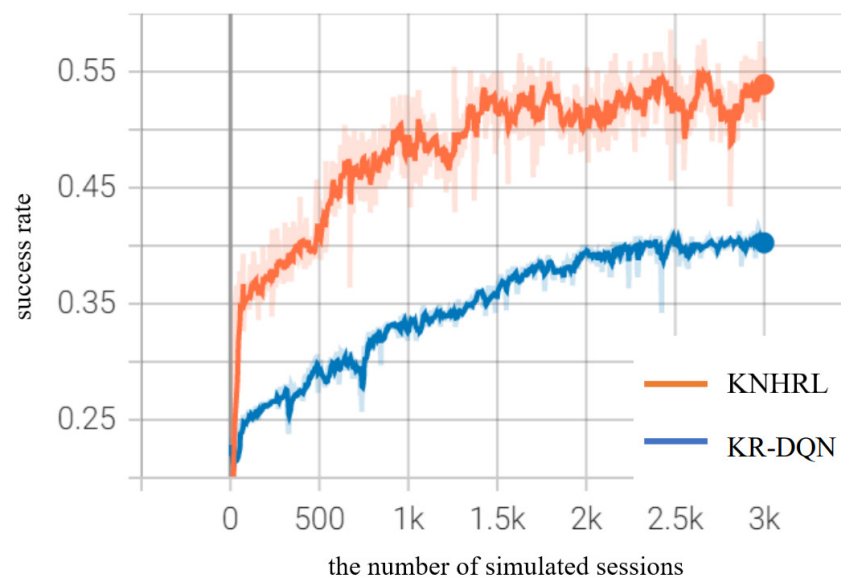
	Success Rate	Average Dialogue Turns	Match Rate
SVM-ex	0.321	/	/
DQN	0.343	<b>2.455</b>	0.045
KR-DQN	0.395	7.120	0.067
REFUEL	0.416	8.551	0.089
GAMP	0.409	3.535	0.057
HRL-pretrained	0.452	6.838	/
HRL	0.504	12.959	0.105
KNHRL	<b>0.558</b>	20.984	<b>0.333</b>
SVM-ex-im	0.732	/	/

In Table 1, the performance of the KNHRL model is better than that of the SVM-ex model, and the success rate of other deep reinforcement learning models is also higher than that of SVM-ex. This shows that when only explicit symptoms are used for training, deep reinforcement models perform better on the task of disease diagnosis than the multi-class classification model. For the other models in Table 1, DQN, KR-DQN, REFUEL, and GAMP are deep reinforcement learning models with single-layer strategies. Among them, REFUEL and GAMP introduce additional mechanisms to optimize the reinforcement learning model based on the basic DQN; KR-DQN adds medical knowledge to the basic DQN to assist decision-making, and the disease diagnosis success rate of the KR-DQN model is higher than that of the DQN model, which indicates the medical knowledge can improve the performance of disease diagnosis models.

Compared with KR-DQN, the performance of KNHRL has been greatly improved in terms of success rate and matching rate. This proves the necessity of a stratified strategy in the case of a large number of diseases and symptoms. HRL-pretrained and HRL are deep reinforcement learning models with hierarchical strategies. KNHRL has greatly outperformed these two models. Note that compared with the current state-of-the-art hierarchical strategy reinforcement learning model HRL, KNHRL has an improvement of 5.4% and 22.8% in the success rate and matching rate, respectively. This result shows that medical knowledge plays an important role in disease diagnosis, especially in improving the symptom matching rate of the model. The performance of KR-DQN is inferior to the HRL-pretrained and HRL. This indicates that in the disease diagnosis task, in the case of a large number of diseases and symptoms, the hierarchical strategy plays a greater role in improving model performance.

In Table 1, KNHRL outperforms all other baseline models in the success rate, and is the closest to the upper bound (SVM-ex-im) of the deep reinforcement learning model performance on this synthetic dataset. Compared with KRDQN and HRL, KNHRL has a great improvement in the matching rate. However, the average number of dialogue turns of KNHRL is higher than the rest of the baseline models, which may be caused by the hierarchical strategy and medical knowledge that bring more information to the model. In future work, how to reduce the number of dialogue turns without reducing the success rate and matching rate will be the key issue of research.

Figure 5 illustrates the learning curves of the KNHRL model and the recurrent KR-DQN model on the synthetic dataset, which respectively show the changes in the success rate for the dataset during the learning process of the two models. Both models are used for 3000 simulated dialogues. From the learning curve, the learning curve of the KNHRL model reaches a plateau at about 1500, while the learning curve of the KR-DQN model reaches a plateau at about 2000, which shows that KNHRL learns faster than KR-DQN. Therefore, the disease diagnosis success rate of the KNHRL model is better than that of the KR-DQN model.



**Figure 5.** Learning curve of KNHRL and KR-DQN on the synthetic dataset.

#### 6.4. Further Analysis

In order to prove that in the KNHRL model, each component has a positive effect on the improvement of performance, this paper designs ablation experiments, as shown in Table 2. The results of each ablation experiment are the average of the results obtained from three experiments at the same setting.

**Table 2.** Evaluation results of ablation experiments.

	Success Rate	Average Number of Dialogue Turns	Match Rate
KNHRL	<b>0.558</b>	20.984	<b>0.333</b>
-dl	0.545	20.102	0.315
-rel	0.522	18.334	0.179
-kg	0.526	19.568	0.186
-hrl	0.426	6.105	0.082
-all	0.343	2.455	0.045

In Table 2, -dl is the result obtained by removing the disease classifier in the low-level strategy on the basis of the complete model; -rel is the result obtained by removing the relation module; -kg is the result obtained by removing the relation module; -hrl is the result of the experiment without using the hierarchical strategy; -all is the result of removing all the above modules. In Table 2, the model performance of all ablation experiments is lower than the full model in terms of success rate and matching rate, which verifies the effectiveness of all components in KNHRL. In addition, the success rate and matching rate of -hrl are lower than those of -rel and -kg, which further proves that when the number of diseases and symptoms is large, the hierarchical strategy plays a greater role in improving the model performance of the disease diagnosis task. Note that the success rate and matching rate of -hrl are both higher than the results of the KR-DQN model in Table 1, which shows that the knowledge embedding method in KNHRL is better than the knowledge embedding method in KR-DQN.

## 7. Conclusions

This paper proposes a hierarchical reinforcement learning model KNHRL for knowledge-enhanced automatic disease diagnosis in medical dialogue systems. Based on the hierarchical reinforcement learning strategy, a medical knowledge graph is incorporated into each low-level agent to assist decision-making. The learnable relationship matrix and disease classifier are used to assist the low-level agent to make policy. The effectiveness of KNHRL is validated on a publicly available dataset for disease diagnosis. In future work, we hope to collect a real-world medical dialogue dataset for disease diagnosis, and further verify the performance of the KNHRL model.

### 7.1. Limitations

This work mainly focuses on a knowledge-enhanced hierarchical reinforcement learning model in the medical dialogue system for disease diagnosis. We have identified two key limitations that can be further examined in future research. The first limitation is that the KNHRL model tends to have a relatively higher average number of dialogue turns due to the hierarchical strategy and the incorporation of medical knowledge, which provides the model with more information. In the future, a key research focus will be on reducing the number of dialogue turns without compromising the success rate of diagnosis and symptom matching. Additionally, due to the limited availability of real diagnostic datasets, we utilized an artificially synthesized dialogue dataset for disease diagnosis. In future work, we aim to collect a real-world medical dialogue dataset, specifically designed for disease diagnosis. We intend to utilize this dataset to validate and improve the performance of the KNHRL model.

### 7.2. Ethics Statement

This paper aims to investigate hierarchical reinforcement learning-based approaches for automatic disease diagnosis, with the objective of reducing the burden on doctors and promoting the advancement of automatic diagnosis systems. It is crucial to emphasize that the proposed methods are designed solely for research purposes and are not suitable for direct clinical application due to the potential risks associated with the misuse of automatic



diagnosis systems. Furthermore, the dataset used in our experiments is synthetic; therefore, there are no issues related to ethics and privacy concerns.

**Author Contributions:** Conceptualization, Y.Z. (Ying Zhu); Formal analysis, Y.C., T.Z., D.W., Y.Z. (Yifei Zhang) and S.F.; Investigation, Y.L. and Y.C.; Methodology, Y.Z. (Ying Zhu); Resources, D.W., Y.Z. (Yifei Zhang) and S.F.; Validation, Y.Z. (Ying Zhu), Y.L. and S.F.; Writing—review & editing, Y.L. and T.Z. All authors have read and agreed to the published version of the manuscript.

**Funding:** The work was supported by the National Natural Science Foundation of China (62272092, 62172086).

**Data Availability Statement:** Data are contained within the article.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Moro, G.; Ragazzi, L.; Valgimigli, L.; Freddi, D. Discriminative marginalized probabilistic neural method for multi-document summarization of medical literature. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Dublin, Ireland, 22–27 May 2022; pp. 180–189.
2. Yan, S. Memory-aligned knowledge graph for clinically accurate radiology image report generation. In Proceedings of the 21st Workshop on Biomedical Language Processing, Dublin, Ireland, 26 May 2022; pp. 116–122.
3. Soleimani, A.; Nikoulina, V.; Favre, B.; Ait-Mokhtar, S. Zero-Shot Aspect-Based Scientific Document Summarization using Self-Supervised Pre-training. In Proceedings of the 21st Workshop on Biomedical Language Processing, Dublin, Ireland, 26 May 2022; pp. 49–62.
4. Boissonnet, A.; Saeidi, M.; Plachouras, V.; Vlachos, A. Explainable assessment of healthcare articles with QA. In Proceedings of the 21st Workshop on Biomedical Language Processing, Dublin, Ireland, 26 May 2022; pp. 1–9.
5. Pappas, D.; Malakasiotis, P.; Androutsopoulos, I. Data Augmentation for Biomedical Factoid Question Answering. In Proceedings of the 21st Workshop on Biomedical Language Processing, Dublin, Ireland, 26 May 2022; p. 63.
6. Gupta, D.; Demner-Fushman, D. Overview of the MedVidQA 2022 shared task on medical video question-answering. In Proceedings of the 21st Workshop on Biomedical Language Processing, Dublin, Ireland, 26 May 2022; pp. 264–274.
7. Giorgi, J.; Bader, G.D.; Wang, B. A sequence-to-sequence approach for document-level relation extraction. In Proceedings of the 21st Workshop on Biomedical Language Processing, Dublin, Ireland, 26 May 2022; p. 10.
8. Papanikolaou, Y.; Staib, M.; Grace, J.; Bennett, F. Slot Filling for Biomedical Information Extraction. In Proceedings of the 21st Workshop on Biomedical Language Processing, Dublin, Ireland, 26 May 2022; p. 82.
9. Phan, U.; Nguyen, N. Simple Semantic-based Data Augmentation for Named Entity Recognition in Biomedical Texts. In Proceedings of the 21st Workshop on Biomedical Language Processing, Dublin, Ireland, 26 May 2022; pp. 123–129.
10. Jonnalagadda, S.R.; Adupa, A.K.; Garg, R.P.; Corona-Cox, J.; Shah, S.J. Text mining of the electronic health record: An information extraction approach for automated identification and subphenotyping of HFpEF patients for clinical trials. *J. Cardiovasc. Transl. Res.* **2017**, *10*, 313–321. [[CrossRef](#)] [[PubMed](#)]
11. Doshi-Velez, F.; Ge, Y.; Kohane, I. Comorbidity clusters in autism spectrum disorders: an electronic health record time-series analysis. *Pediatrics* **2014**, *133*, e54–e63. [[CrossRef](#)] [[PubMed](#)]
12. Wen, T.H.; Vandyke, D.; Mrksic, N.; Gasic, M.; Rojas-Barahona, L.M.; Su, P.H.; Ultes, S.; Young, S. A network-based end-to-end trainable task-oriented dialogue system. *arXiv* **2016**, arXiv:1604.04562.
13. Lipton, Z.; Li, X.; Gao, J.; Li, L.; Ahmed, F.; Deng, L. Bbq-networks: Efficient exploration in deep reinforcement learning for task-oriented dialogue systems. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018; Volume 32.
14. Yan, Z.; Duan, N.; Chen, P.; Zhou, M.; Zhou, J.; Li, Z. Building task-oriented dialogue systems for online shopping. In Proceedings of the AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017; Volume 31.
15. Huang, Y.; Feng, J.; Hu, M.; Wu, X.; Du, X.; Ma, S. Meta-reinforced multi-domain state generator for dialogue systems. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; pp. 7109–7118.
16. Wang, S.; Zhou, K.; Lai, K.; Shen, J. Task-completion dialogue policy learning via Monte Carlo tree search with dueling network. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Online, 16–20 November 2020; pp. 3461–3471.
17. Rehman, U.U.; Chang, D.J.; Jung, Y.; Akhtar, U.; Razzaq, M.A.; Lee, S. Medical instructed real-time assistant for patient with glaucoma and diabetic conditions. *Appl. Sci.* **2020**, *10*, 2216. [[CrossRef](#)]
18. Wei, Z.; Liu, Q.; Peng, B.; Tou, H.; Chen, T.; Huang, X.J.; Wong, K.F.; Dai, X. Task-oriented dialogue system for automatic diagnosis. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Melbourne, Australia, 15–20 July 2018; pp. 201–207.
19. Liao, K.; Liu, Q.; Wei, Z.; Peng, B.; Chen, Q.; Sun, W.; Huang, X. Task-oriented dialogue system for automatic disease diagnosis via hierarchical reinforcement learning. *arXiv* **2020**, arXiv:2004.14254

20. Fang, M.; Li, Y.; Cohn, T. Learning how to active learn: A deep reinforcement learning approach. *arXiv* **2017**, arXiv:1708.02383.
21. Chen, J.; Wang, Z.; Tomizuka, M. Deep hierarchical reinforcement learning for autonomous driving with distinct behaviors. In Proceedings of the 2018 IEEE Intelligent Vehicles Symposium (IV), Changshu, China, 26–30 June 2018; IEEE: Piscataway Township, NJ, USA, 2018; pp. 1239–1244.
22. Liu, J.; Pan, F.; Luo, L. Gochat: Goal-oriented chatbots with hierarchical reinforcement learning. In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, China, 25–30 July 2020; pp. 1793–1796.
23. Zhao, D.; Zhang, L.; Zhang, B.; Zheng, L.; Bao, Y.; Yan, W. Mahrl: Multi-goals abstraction based deep hierarchical reinforcement learning for recommendations. In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, China, 25–30 July 2020; pp. 871–880.
24. Wang, X.; Chen, W.; Wu, J.; Wang, Y.F.; Wang, W.Y. Video captioning via hierarchical reinforcement learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4213–4222.
25. Zhou, X.; Bai, T.; Gao, Y.; Han, Y. Vision-based robot navigation through combining unsupervised learning and hierarchical reinforcement learning. *Sensors* **2019**, *19*, 1576. [[CrossRef](#)] [[PubMed](#)]
26. Xie, R.; Zhang, S.; Wang, R.; Xia, F.; Lin, L. Hierarchical reinforcement learning for integrated recommendation. In Proceedings of the AAAI Conference on Artificial Intelligence, Online, 2–9 February 2021; Volume 35, pp. 4521–4528.
27. Huang, Q.; Gan, Z.; Celikyilmaz, A.; Wu, D.; Wang, J.; He, X. Hierarchically structured reinforcement learning for topically coherent visual story generation. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; Volume 33; pp. 8465–8472.
28. Chen, Y.; Tao, L.; Wang, X.; Yamasaki, T. Weakly supervised video summarization by hierarchical reinforcement learning. In Proceedings of the ACM Multimedia Asia, Beijing, China, 15–18 December 2019; pp. 1–6.
29. Jain, D.; Iscen, A.; Caluwaerts, K. Hierarchical reinforcement learning for quadruped locomotion. In Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 3–8 November 2019; IEEE: Piscataway Township, NJ, USA, 2019; pp. 7551–7557.
30. Li, T.; Lambert, N.; Calandra, R.; Meier, F.; Rai, A. Learning generalizable locomotion skills with hierarchical reinforcement learning. In Proceedings of the 2020 IEEE International Conference on Robotics and Automation (ICRA), Paris, France, 31 May–31 August 2020; IEEE: Piscataway Township, NJ, USA, 2020; pp. 413–419.
31. Budzianowski, P.; Ultes, S.; Su, P.H.; Mrkšić, N.; Wen, T.H.; Casanueva, I.; Rojas-Barahona, L.; Gašić, M. Sub-domain modelling for dialogue management with hierarchical reinforcement learning. *arXiv* **2017**, arXiv:1706.06210.
32. Saha, T.; Gupta, D.; Saha, S.; Bhattacharyya, P. Towards integrated dialogue policy learning for multiple domains and intents using hierarchical deep reinforcement learning. *Expert Syst. Appl.* **2020**, *162*, 113650. [[CrossRef](#)]
33. Saha, T.; Saha, S.; Bhattacharyya, P. Towards sentiment aided dialogue policy learning for multi-intent conversations using hierarchical reinforcement learning. *PLoS ONE* **2020**, *15*, e0235367. [[CrossRef](#)] [[PubMed](#)]
34. Ghandeharioun, A.; Shen, J.H.; Jaques, N.; Ferguson, C.; Jones, N.; Lapedriza, A.; Picard, R. Approximating interactive human evaluation with self-play for open-domain dialog systems. In Proceedings of the 33rd International Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, BC, Canada, 8–14 December 2019; Volume 32.
35. Peng, Y.S.; Tang, K.F.; Lin, H.T.; Chang, E. Refuel: Exploring sparse features in deep reinforcement learning for fast disease diagnosis. In Proceedings of the 33rd International Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, BC, Canada, 8–14 December 2019; Volume 31.
36. Hou, Z.; Liu, B.; Zhao, R.; Ou, Z.; Liu, Y.; Chen, X.; Zheng, Y. Imperfect also deserves reward: Multi-level and sequential reward modeling for better dialog management. *arXiv* **2021**, arXiv:2104.04748.
37. Teixeira, M.S.; Maran, V.; Dragoni, M. The interplay of a conversational ontology and AI planning for health dialogue management. In Proceedings of the 36th Annual ACM Symposium on Applied Computing, Virtual Event, Republic of Korea, 22–26 March 2021; pp. 611–619.
38. Xu, L.; Zhou, Q.; Gong, K.; Liang, X.; Tang, J.; Lin, L. End-to-end knowledge-routed relational dialogue system for automatic diagnosis. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; Volume 33; pp. 7346–7353.
39. Liu, S.; Chen, H.; Ren, Z.; Feng, Y.; Liu, Q.; Yin, D. Knowledge diffusion for neural dialogue generation. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Melbourne, Australia, 15–20 July 2018; pp. 1489–1498.
40. Xia, Y.; Zhou, J.; Shi, Z.; Lu, C.; Huang, H. Generative adversarial regularized mutual information policy gradient framework for automatic diagnosis. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34; pp. 1062–1069.
41. Kao, H.C.; Tang, K.F.; Chang, E. Context-aware symptom checking for disease diagnosis using hierarchical reinforcement learning. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018; Volume 32.

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.