# CoD, Towards an Interpretable Medical Agent using Chain of Diagnosis

Anonymous ACL submission

## Abstract

The field of AI healthcare has undergone a significant transformation with the advent of large language models (LLMs), yet the challenge of interpretability in these models remain largely unaddressed. This study introduces **Chain-of-Diagnosis (CoD)** to enhance the interpretability of medical automatic diagnosis. CoD transforms the diagnostic process into a diagnostic chain that mirrors a physician's thought process, providing a transparent reasoning pathway. Additionally, CoD outputs the disease confidence distribution to ensure transparency in decision-making. This interpretability makes model diagnostics controllable and aids in identifying critical symptoms for inquiry through the entropy reduction of confidences. With CoD, we developed **DiagnosisGPT**, capable of diagnosing 9,604 diseases for validating CoD. Experimental results demonstrate that DiagnosisGPT outperforms other LLMs on automatic diagnostic tasks across three real-world benchmarks. Moreover, DiagnosisGPT provides interpretability while ensuring controllability in diagnostic rigor.

## 1 Introduction

In AI healthcare, automatic diagnosis (Tang et al., 2016; Xu et al., 2019a; Fansi Tchango et al., 2022), which aims to provide convenient medical care and assist in diagnosis, is one of the most promising applications and is garnering increasing attention (Liu et al., 2022; Hou et al., 2023; Hu et al., 2024; Yuan and Yu, 2024). However, it is complex, challenging the agent with multi-step decision-making abilities (Chen et al., 2022b). Specifically, it relies on interactions between an agent and a patient, where the agent inquires about the necessary symptoms for diagnosis, as illustrated in Figure 1. This is consistent with real-world diagnostic procedures, where doctors inquire about symptoms to make a diagnosis (Kao et al., 2018; Zhao et al., 2021). Compared to prior methods, Large language models (LLMs) offer a promising path due to their superior reasoning and dialogue abilities (Barua, 2024). These capabilities enable them to address a wide range of diseases and interact effectively with patients (Chen et al., 2023a).

In this paper, we explore the use of LLMs for automatic diagnosis. In our preliminary experiments, we find that LLMs, like GPT-4, tend to make arbitrary diagnoses without sufficient inquiry. Without interpretability, it is unclear if the decisions meet sound analytical and ethical standards (Huang et al., 2023; Savage et al., 2024). Interpretability is crucial to mitigate such arbitrary decisions. On the other hand, LLMs demonstrate poor symptom inquiry capabilities, consistent with findings from (Hu et al., 2024). Optimizing this multi-step decision-making process without interpretability is challenging, akin to Monte Carlo sampling.

In response to these limitations, we propose the **Chain of Diagnosis (CoD)** to enhance the interpretability of LLMs. CoD provides transparency for the diagnostic process. It transforms the black-box decision-making process into a diagnostic chain that mirrors a physician's thinking process through five steps. For decision transparency, CoD outputs a confidence distribution, where higher confidence indicates a stronger belief in diagnosing a specific disease. This allows for control over the LLM's decisions using a confidence threshold. Additionally, diagnostic uncertainty can be quantified by the entropy of these confidence levels. The goal of entropy reduction can aid in eliciting more effective symptoms for inquiry.

To implement CoD, this paper proposes constructing CoD training data from synthetic patient cases generated from disease encyclopedias. This approach mitigates concerns about patient privacy and allows for scalability in validating CoD. We constructed a training dataset containing 48,020 CoD instances, leading to the development of our model, **DiagnosisGPT**, capable of diagnosing

9,604 diseases. Experiments demonstrate that DiagnosisGPT outperforms other LLMs with controllable diagnostic rigor. Moreover, it achieves over 90% accuracy across all datasets with a diagnostic threshold of 0.55, underscoring the reliability of its confidence levels. It is important to emphasize that DiagnosisGPT is developed exclusively for validating CoD and has no clinical application.

Our contributions are summarized as follows: 1) We introduce the Chain-of-Diagnosis (CoD) method, designed to enhance interpretability of LLMs in disease diagnosis; 2) We propose to synthesize patient cases using disease encyclopedias to validate CoD, avoiding privacy and ethical concerns; 3) Using CoD, we built DiagnosisGPT that can support automatic diagnosis for 9,604 diseases. Experiments demonstrate the excellent interpretability and multi-turn decision-making capabilities of DiagnosisGPT; 4) We present DxBench, a real-world diagnostic benchmark with 1,148 real cases covering 461 diseases, to expand the scope of existing automatic diagnosis evaluations.

## 2 Preliminaries

### 2.1 Problem definition

The automatic diagnosis task is a multi-step reasoning challenge that evaluates an AI system's medical decision-making and diagnostic abilities. Unlike traditional medical QA tasks like MedQA (Jin et al., 2021), which overlook sequential decision-making, this task involves an agent interacting with a patient to gather symptoms for diagnosis. As shown in Figure 1, the agent first receives the patient's self-report (**explicit symptoms** $\mathcal{S}_{\text{exp}}$) and then decides whether to continue inquire about additional symptoms (**implicit symptoms** $\mathcal{S}_{\text{imp}}$) or conclude with a diagnosis. For diagnosis, it predicts a target disease ($d_t$) from a predefined list ($\mathcal{D}$). The objective is to maximize diagnostic accuracy ($a$) within a limited number of symptom inquiries ($n$), where $n \leq L$ with $L$ being the maximum number of inquiries.

### 2.2 The Challenge for LLM

The challenge lies in determining when and how to inquire about symptoms to improve diagnostic accuracy. This seems well-suited for large language models (LLMs), which excel in reasoning and dialogue. To explore this, we conducted a preliminary experiment on two public benchmarks using GPT-4 (GPT-4-0125-preview):

| | w/o inquiry | w/ inquiry | |
|---|---|---|---|
| | $a$ | $a$ | $n$ |
| **Muzhi Dataset** (Wei et al., 2018) | | | |
| GPT-4 | 59.2 | 57.7 -1.5 | 0.4 |
| w/ Chain-of-Thought ◇ | 61.3 | 59.9 -1.4 | 0.2 |
| w/ Multi-Choice ◇ | 58.7 | 56.3 -2.4 | 0.7 |
| **Dxy Dataset** (Xu et al., 2019a) | | | |
| GPT-4 | 62.5 | 65.4 + 2.9 | 0.6 |
| w/ Chain-of-Thought ◇ | 62.5 | 64.4 + 0.9 | 0.4 |
| w/ Multi-Choice ◇ | 60.6 | 63.5 + 2.9 | 0.6 |

Table 1: Automatic Diagnosis using GPT-4 Turbo. ◇ represents various prompts detailed in Appendices E and D. *"w/o inquiry"* indicates no symptom querying allowed, i.e., $n = 0$. **green** and **red** highlight increases and decreases in accuracy after symptom querying.

Table 1 highlights two potential issues with LLMs:

- **Issue I, Arbitrary Diagnosis:** Even using various prompts, LLMs prefer to diagnose directly without sufficient inquiry, with GPT-4 averaging fewer than one inquiry per case ($n < 1$). This contrasts with real doctors, who thoroughly question before diagnosing. To prevent hasty diagnoses, their decision-making process should be more transparent, ensuring it does not act recklessly.
- **Issue II, Ineffective Inquiries:** With follow-up inquiries, GPT-4's diagnostic accuracy does not improve significantly and even decreases on the Muzhi dataset. This aligns with findings from (Hu et al., 2024) that LLMs are not particularly effective at questioning users. While LLMs need to ask more pertinent questions, optimizing this process in a non-interpretable manner is challenging, akin to Monte Carlo sampling.

To address these issues, this paper proposes the **Chain-of-Diagnosis (CoD)** method to enhance the interpretability of LLMs in automated diagnosis.

### 2.3 The Philosophy of CoD for Interpretability

Lipton (Lipton, 2018) defines *interpretability* in two aspects: 1) transparency, i.e., *how does the model work?* and 2) post-hoc explanations, i.e., *what can the learned model tell us?* These two aspects inspire the design of the CoD framework, which includes Property 1 and 2 for transparency and Property 3 for post-hoc explanations.

*Transparency* connotes an understanding of the mechanism by which the model works, encompassing multiple levels. At the decision-making level,

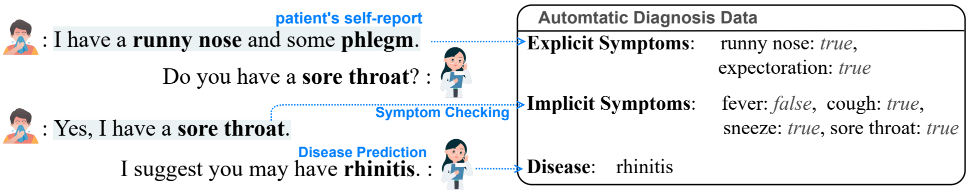


Figure 1: Example of the automatic diagnosis task, with sample data from Hou et al., 2023.

CoD introduces confidence to make its decision-making process more interpretable, as described in Property 1.

**Property 1** ***Transparency with Confidence-driven Decision:*** *CoD introduces a disease confidence distribution $C = \{c_d \mid d \in \mathcal{D}\}$, where higher confidence indicates a stronger belief in a specific disease. Decisions are based on whether the highest confidence exceeds a threshold $\tau$.*

*Algorithmic transparency* involves understanding the learning algorithm itself, such as its convergence (Lipton, 2018). The algorithmic transparency of CoD can be understood from an entropy-reduction perspective: with more inquiries made, the uncertainty of the diagnosis estimate will be reduced, as illustrated in Property 2.

**Property 2** ***Transparency with Confidence-driven Flow:*** *In CoD, as confidence levels rise with more inquiries, accuracy improves due to reduced uncertainty (lower entropy), converging when accuracy stabilizes with sufficient inquiries.*

*Post-hoc explanations* (Lipton, 2018) refer to the information and functions a model can provide to humans. The post-hoc explanations for CoD are outlined in Property 3.

**Property 3** ***Explainability with Diagnostic Chain:*** *CoD transforms the black-box decision-making process into an explainable diagnostic chain, providing physicians with a diagnostic pathway that supports their clinical decisions and ensures that the LLM's decisions adhere to reasonable analysis. The overall chain mimics a real physician's diagnosis, offering a more comprehensive analysis.*

## 3 Methodology: Chain of Diagnosis

As depicted on the left side of Figure 2, the CoD outputs a diagnostic chain, that mirrors a physician's diagnostic thinking. To implement the CoD, we construct CoD training data based on patient cases to fine-tune LLMs to perform CoD, as shown on the right side of Figure 2.

### 3.1 The Diagnostic Chain

Here, we introduce the response methods and the construction approach of CoD, as illustrated in Figure 2. All prompts for building CoD training data are detailed in Appendix G.

**Step 1: Symptom Abstraction** The first step summarizes the symptoms $\mathcal{S}$ of the patient's question:

$$\mathcal{S} = f_1(q_{\text{patient}}) \tag{1}$$

It allows the model to focus on the refined symptoms and provides an understanding of the patient's query. For training data, the initial patient question is generated from $\mathcal{S}_{\text{exp}}$ with the LLM.

**Step 2: Candidate Disease Recall** Next, CoD identifies the top-K potential diseases based on a disease retriever (under an open-ended setting) :

$$\mathcal{D}' = f_2(\mathcal{D}, \mathcal{S}, k) \tag{2}$$

where $\mathcal{D}' \subseteq \mathcal{D}$ and $|\mathcal{D}'| = k$. A smaller space $\mathcal{D}'$ is necessary for subsequent analysis and reasoning, since analyzing all diseases is impractical (considering $|\mathcal{D}| = 9604$) and most irrelevant diseases can realistically be excluded. We use Dense Retrieval training methods (Karpukhin et al., 2020; Chen et al., 2022a) to train this retriever, with the following training objective:

$$\mathcal{L}\left(\mathcal{S}_{\text{exp}}, \mathcal{S}_{\text{imp}}, d_t\right) = -\log \frac{e^{\text{sim}(E_S(\mathcal{S}_{\text{exp}} \cup \mathcal{S}_{\text{imp}}), E_D(d_t))}}{\sum_{d \in \mathcal{D}} e^{\text{sim}(E_S(\mathcal{S}_{\text{exp}} \cup \mathcal{S}_{\text{imp}}), E_D(d))}} \tag{3}$$

where sim denotes the cosine similarity, and $E_S$ and $E_D$ are the symptom and disease encoders, respectively. The performance of the disease retriever is detailed in Appendix M.

Then, for each candidate disease $d \in \mathcal{D}'$, CoD retrieves corresponding disease knowledge from the disease database and integrates it into the output to enhance understanding of the disease. Similarly, other tools like RAG can also be utilized in this step to enhance reasoning.

**Step 3: Diagnostic Reasoning** In step 3, CoD generates the diagnostic reasoning process $T$:

$$T = f_3(\mathcal{S}, \mathcal{D}') \tag{4}$$

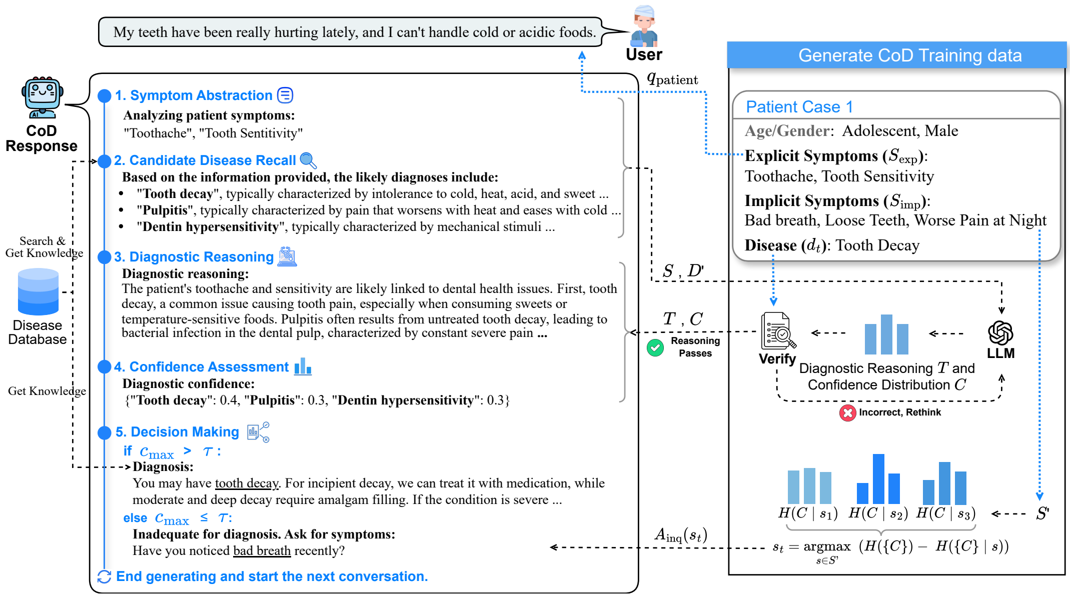


Figure 2: Left: Example of a CoD response. Right: Construction of CoD training data.

Similar to CoT, $T$ is a thought process that carefully analyzes whether each disease in $\mathcal{D}'$ corresponds to the patient's symptoms. To build training data, we prompt a LLM to generate $T$.

**Step 4: Confidence Assessment** After generating $T$, CoD generates a confidence distribution:

$$\mathcal{C} = f_4(\mathcal{S}, \mathcal{D}', T) \tag{5}$$

$\mathcal{C}$ satisfies $\sum_{d \in \mathcal{D}'} c_d = 1$. This distribution indicates the model's tendency towards diagnosing a disease, mainly according to the analysis of $T$. According to $f_3$, $\mathcal{C}$ can be considered a posterior probability distribution:

$$\mathcal{C} = \{p_\theta(d|\mathcal{S}, \mathcal{D}') | d \in \mathcal{D}'\} \tag{6}$$

Here, $p_\theta$ represents the confidence distribution generated by the LLM $\theta$. For constructing training data, we validate $\mathcal{C}$ against the target disease $d_t$ to ensure $T$ and $\mathcal{C}$ are reasonable. If $\max_{d \in \mathcal{D}' \setminus \{d_t\}} c_d \geq \tau$, the generated data is considered erroneous, i.e., the model assigns high confidence to an incorrect disease. If erroneous, we prompt the model to rethink and correct its reasoning until the distribution is verified. With $\mathcal{C}$, CoD can make decisions based on the confidence in its diagnosis.

**Step 5: Decision Making** In the last step, a confidence threshold $\tau$ is set to control the decision-making. The diagnostic task involves two decision types: 1) making a diagnosis $A_{diag}(d)$, where $d$ is the diagnosed disease, and 2) to inquiring about a symptom $A_{inq}(s)$, where $s$ represents the symptom under inquiry. The next decision $A_{next}$ of the CoD is defined as:

$$A_{\text{next}} = \begin{cases} A_{\text{diag}}(d_{\max}), & \text{if } c_{\max} > \tau \\ A_{\text{inq}}(s_t), & \text{if } c_{\max} \leq \tau \end{cases} \tag{7}$$

where $c_{\max} = \max_{d \in \mathcal{D}'}\{c_d\}$ and $d_{\max} = \operatorname{argmax}_{d \in \mathcal{D}'}\{c_d\}$. $A_{inq}(s_t)$ signifies the operation of querying about the symptom $s_t$ that the CoD generates. Here, $\tau$ serves as a hyperparameter. A higher $\tau$ allows the model to perform more rigorous diagnoses (that achieving higher accuracy $a$ but requiring more rounds of questioning, i.e., higher $n$). Conversely, a lower $\tau$ can reduce $n$ but also lowers $a$.

## 3.2 CoD as an Entropy-reduction Process

Symptom inquiry is a key step in diagnosis, serving to gather additional patient information to clarify the diagnosis. This inquiry process can be viewed as a transition from diagnostic uncertainty to certainty. The uncertainty level can be captured by the entropy of confidence:

$$H(C) = -\sum_{d \in D'} c_d \log c_d \tag{8}$$

Symptom inquiry is a process of entropy reduction. Given a symptom $s$, its post-inquiry entropy is:

$$H(C|s) = -\sum_{d \in D'} p_\theta(d|S \cup \{s\}, D') \log p_\theta(d|S \cup \{s\}, D') \tag{9}$$

For the diagnostic task, it's crucial to gather key symptom information to boost $a$ within limited $n$.

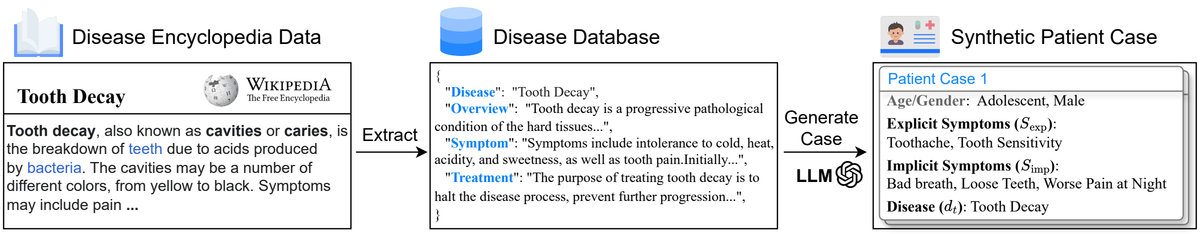


Figure 3: Schematic of constructing disease database and synthesizing patient cases.

Hence, the objective of symptom inquiry can be formalized as maximizing the increase in diagnostic certainty to expedite the diagnosis. Accordingly, CoD selects the symptom to inquire about by maximizing the entropy reduction:

$$s_t = \underset{s \in \mathcal{S}'}{\operatorname{argmax}}(H(\mathcal{C}) - H(\mathcal{C}|s)) \tag{10}$$

where $\mathcal{S}'$ represents the candidate symptoms for inquiry and $s_t$ is the chosen symptom. $\mathcal{S}' = \mathcal{S}_{\text{imp}} \cup \{s_{\text{gen}}\}$, where $s_{\text{gen}}$ is the symptom generated by the LLM and $\mathcal{S}_{\text{imp}}$ comes from the training case data. Through entropy reduction, the CoD training data tuned the model to inquire about more crucial symptoms for diagnosis, thereby enhancing its querying capability.

### 3.3 Validation of CoD

CoD requires patient cases to build training data. However, due to privacy concerns, the collection of such data is significantly restricted. To address this, we propose generating synthetic case data in reverse using online disease encyclopedias, which provide comprehensive and reliable disease information. As illustrated in Figure 3, the synthesis process is a pipeline that consists of two stages:

**Stage 1: Constructing Disease Database** The first step involves the extraction of essential information from the disease encyclopedia data. This process results in a knowledge base encompassing 9,604 diseases, each detailed with sections on "Overview," "Symptoms," and "Treatment". We use regular expression matching to identify and extract these key sections.

**Stage 2: Synthesizing Patient Cases** In disease diagnosis (Shivade et al., 2014; Wei et al., 2018), a patient can be abstracted into a triplet $(S_{exp}, S_{imp}, d_t)$. Using GPT-4, we generate structured case data based on the disease knowledge from the database. For each disease, we synthesize five distinct cases to ensure diversity. The prompt used for generation is provided in Appendix F.

In the end, we developed a database containing **9,604** diseases and then synthesized **48,020** unique cases to validate CoD. With them, we constructed a training dataset for CoD, which consists of 48,020 instances with an average of 2.4 consultation rounds. We used *GPT-4-0125-preview* to synthesize CoD training data. This dataset is used to train an interpretable medical diagnosis model, **DiagnosisGPT**, which is a model developed exclusively for validating CoD.

## 4 Experiments

### 4.1 Model Training & Setup

Utilizing the created CoD data, we fine-tuned the **Yi-34B-Base** (Young et al., 2024) to develop **DiagnosisGPT**. To equip it with chat capabilities, ShareGPT data (sha, 2023) is incorporated into the training data. The training parameters included a batch size of 64 and a learning rate of 2e-5. For the disease retrieval model, we trained on the all-mpnet-base-v2 (Reimers and Gurevych, 2019) model using DRhard (Zhan et al., 2021), with a batch size of 256 and a learning rate of 2e-5. The training was conducted on a GPU server with 8 NVIDIA A100 GPUs.

### 4.2 Benchmarking Settings

**Traditional baselines (Non-LLM)** Traditional supervised Automatic Diagnosis methods approach the diagnostic task as a decision-making task, where all symptoms and diseases are predefined. In traditional methods, we adhere to the original settings, which involve **training on a training set of benchmarks** and **traditional rule-based evaluations**. We compared four models: Basic DQN (Wei et al., 2018), HRL (Liao et al., 2021), Diaformer (Chen et al., 2022b) and MTDiag (Hou et al., 2023).

**LLM baselines** Our comparison mainly focused on advanced LLMs including proprietary models like Gemini-Pro (Reid et al., 2024), ERNIE

Bot (Sun et al., 2021), Claude-3-Opus (anthropic, 2024), GPT-3.5 (GPT-3.5-turbo-1106) (OpenAI, 2022), and GPT-4 (GPT-4-0125-preview) (OpenAI, 2023) Mixtral-8x7B-Instruct-v0.1 (Jiang et al., 2024) and Yi-34B-Chat (Young et al., 2024). Note that all LLMs, including ours, **use a zero-shot setting** and **open-ended symptom inquiry**.

**LLM Evaluation** Due to the open-ended inquiry of LLMs, we simulate a patient using GPT-4 (GPT-4-0125-preview) to respond to LLMs. The simulated patient is presented with both $\mathcal{S}_{\text{exp}}$ (chief complaints) and $\mathcal{S}_{\text{imp}}$. The simulation begins with $\mathcal{S}_{\text{exp}}$ (chief complaints). When the evaluated LLM inquires about symptoms, the simulator can only respond with "yes" or "no" to prevent information leakage. Details of the LLM evaluation can be found in Appendix E. For the evaluated LLMs, we prompt them to perform an automated diagnosis task, which is detailed in Appendix D.

### 4.3 Benchmarks

**Public benchmarks** To evaluate diagnostic performance, we used two publicly available benchmarks: Muzhi (Wei et al., 2018) and Dxy (Xu et al., 2019a). Both are based on real-world doctor-patient consultations. However, their data scale and disease variety are limited, as shown in Table 2.

**DxBench** To better assess diagnostic capabilities, we develop a larger dataset, DxBench. Using the MedDialog (Zeng et al., 2020) dataset, which contains real doctor-patient dialogues, we filtered out 3,121 cases with clear dialogues and definitive diagnoses. Then GPT-4 is employed to extract $\mathcal{S}_{exp}$ and $\mathcal{S}_{imp}$, and we manually refine this to 1,148 high-quality cases. Details are in Appendix H. DxBench includes over 1,000 real cases, covering 461 disease types from 15 departments and 5,038 symptoms. Considering the large number of diseases in DxBench, each case is provided with three candidate diseases, one of which is the ground truth. DxBench is proposed to expand the scope of existing automatic diagnosis evaluations.

| Dataset | # Disease | # Symptom | # Test Data |
|---|---|---|---|
| MuZhi | 4 | 66 | 142 |
| Dxy | 5 | 41 | 104 |
| **DxBench** | **461** | **5038** | **1148** |

Table 2: Comparison of DxBench with other datasets.

### 4.4 Diagnosis Performance

**Comparison Results** Table 3 presents the results of the automatic diagnosis, highlighting the following points: **(1)** The zero-shot performance of large language models (LLMs) is comparable to traditional supervised fine-tuning methods. On the Dxy dataset, Claude-3-Opus and DiagnosisGPT achieved accuracies of 72.6% and 75.4%, respectively, close to the 76.1% accuracy of fine-tuned models, even with a smaller $n$. However, LLMs generally ask fewer questions than traditional methods (smaller $n$). In contrast, DiagnosisGPT can adjust the number of inquiries by tuning $\tau$. **(2)** DiagnosisGPT shows the highest accuracy improvement with symptom inquiries across all benchmarks. At $\tau = 0.5$, DiagnosisGPT achieves similar accuracy with fewer inquiries than Claude-3-Opus, and at $\tau = 0.4$, it outperforms GPT-4 with a similar $n$. At $\tau = 0.6$, it delivers the best results among LLMs. Overall, these results demonstrate CoD's strong symptom inquiry and controllability, driven by its algorithmic transparency.

| | DxBench | | |
|---|---|---|---|
| | w/o inquiry | w/ inquiry | |
| | **Acc.** | **Acc.** | $n$ |
| **Without Candidate Diseases (Diagnosing from 9604 Diseases)** | | | |
| **DiagnosisGPT** ($\tau = 0.4$) | 34.7 | 39.2 $_{+4.5}$ | 0.6 |
| **DiagnosisGPT** ($\tau = 0.5$) | 34.7 | 41.4 $_{+6.7}$ | 1.6 |
| **DiagnosisGPT** ($\tau = 0.6$) | 34.7 | 44.2 $_{+9.5}$ | 3.3 |

Table 4: CoD open-ended diagnostic results, relying on the disease retriever to recall diseases from a database of 9604 diseases. $n$ denotes the average number of inquiry rounds, with a limitation of $L = 5$.

**CoD Open-ended Diagnosis** CoD includes a disease retriever. Table 4 shows the results when the retriever directly recalls diseases from a disease database instead of relying on candidate diseases from benchmarks, achieving 44.2% accuracy in open-ended diagnosis. Accuracy increases with higher inquiry parameters ($\tau$).

### 4.5 Explainability on Confidence Levels

**Consistency with Accuracy** To assess the confidence, we examined the diagnostic accuracy at various thresholds $\tau$. The results, depicted in Figure 4, indicate that increasing the threshold indeed enhances accuracy. With $\tau = 0.55$, the model achieves over 90% accuracy across three datasets, demonstrating that the confidence in disease prediction is reliable and aligns with the expected accuracy rates. However, higher thresholds reduce the

| | Muzhi Dataset | | | Dxy Dataset | | | DxBench | | |
|---|---|---|---|---|---|---|---|---|---|
| | w/o inquiry | w/ inquiry | | w/o inquiry | w/ inquiry | | w/o inquiry | w/ inquiry | |
| | **Acc.** | **Acc.** | $n$ | **Acc.** | **Acc.** | $n$ | **Acc.** | **Acc.** | $n$ |
| **Traditional Methods (Supervised Learning)** | | | | | | | | | |
| Basic DQN | - | 64.1 | 2.9 | - | 64.7 | 2.5 | - | - | - |
| HRL | - | 67.6 | 2.8 | - | 70.2 | 1.9 | - | - | - |
| Diaformer | - | 72.2 | 5.0 | - | 76.6 | 4.8 | - | - | - |
| MTDiag | - | 72.6 | 5.0 | - | 76.1 | 5.0 | - | - | - |
| **Large Language Models (Zero-shot Setting)** | | | | | | | | | |
| Yi-34B-Chat | 52.1 | 50.7 -1.4 | 0.4 | 52.9 | 50.5 - 2.4 | 0.5 | 52.4 | 54.6 +2.2 | 0.2 |
| GPT-3.5 | 56.3 | 55.6 -0.7 | 0.2 | 45.6 | 46.2 + 0.6 | 0.4 | 52.1 | 52.3 +0.2 | 0.1 |
| Mixtral-8x7B-Instruct-v0.1 | 56.3 | 50.0 -6.3 | 1.9 | 47.1 | 55.8 + 8.7 | 1.7 | 42.1 | 41.2 -0.9 | 1.4 |
| ERNIE Bot | 61.3 | 57.0 -4.3 | 0.4 | 51.9 | 51.9 + 0.0 | 0.8 | 58.4 | 59.9 +1.5 | 0.4 |
| Gemini-Pro | 63.4 | 60.6 -2.8 | 0.2 | 57.7 | 56.7 - 1.0 | 0.1 | 55.3 | 55.1 -0.2 | 0.1 |
| GPT-4 | 59.2 | 57.7 -1.5 | 0.4 | 62.5 | 65.4 + 2.9 | 0.6 | 60.1 | 62.5 +2.4 | 0.6 |
| Claude-3-Opus | 63.4 | 64.8 +1.4 | 1.8 | 62.5 | 73.1 +10.6 | 1.8 | 59.4 | 62.1 +2.7 | 1.8 |
| **DiagnosisGPT** ($\tau = 0.4$) | 62.0 | 62.7 +0.7 | 0.6 | 60.5 | 68.2 + 7.7 | 0.6 | 56.9 | 61.3 +4.1 | 0.7 |
| **DiagnosisGPT** ($\tau = 0.5$) | 62.0 | 64.1 +2.1 | 1.4 | 60.5 | 72.6 +12.1 | 1.5 | 56.9 | 62.4 +5.5 | 1.8 |
| **DiagnosisGPT** ($\tau = 0.6$) | 62.0 | **65.5** +3.5 | 2.4 | 60.5 | **75.4** +14.9 | 2.8 | 56.9 | **64.5** +7.6 | 3.1 |

Table 3: Results on automatic diagnosis benchmarks. **Acc.** represents the accuracy of disease diagnosis (percentage). $n$ denotes the average number of inquiry rounds, with a cap at $L = 5$ rounds. *"w/o inquiry"* signifies that symptom querying is not allowed ($n = 0$). **green** and **red** fonts highlight increases and decreases in accuracy due to symptom inquiries, respectively. $\tau$ is the confidence threshold of CoD. Traditional models are excluded from DxBench due to the absence of a training dataset for supervised learning.

success rates, indicating that the model becomes more stringent in making diagnoses.

| $\tau$ | **0** | **0.4** | **0.5** | **0.6** | **0.7** |
|---|---|---|---|---|---|
| $a \uparrow$ | 56.9 | 61.3 +4.1 | 62.4 +5.5 | 64.5 +7.6 | 65.3 +8.4 |
| $n \downarrow$ | 0 | 0.7 | 1.8 | 3.1 | 4.2 |

Table 5: The impact of $\tau$ on DiagnosisGPT. Experiments were conducted on the DxBench with $L = 5$ setting. $a$ represents diagnostic accuracy. $n$ denotes the number of queries.

**Control of $\tau$** Table 5 shows how accuracy $a$ and the number of inquiries $n$ change with varying $\tau$. As $\tau$ increases, both $n$ and $a$ rise, supporting the decision transparency of CoD.

### 4.6 Explainability on Entropy Reduction

CoD uses confidence-based entropy reduction to refine follow-up questions. We tested if the learned model, DiagnosisGPT, achieves entropy reduction over multiple inquiries. Table 6 shows that as inquiries increase, entropy decreases, aligning with the agent's decreasing uncertainty throughout the inquiry process.

| Number of Rounds $b$ | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| Average Entropy $H \downarrow$ | 1.467 | 1.396 | 1.380 | 1.371 | 1.369 |

Table 6: The effect of conversation rounds on entropy for DiagnosisGPT ($\tau = 0.6$) on DxBench.

### 4.7 Completeness of the Diagnostic Chain

CoD's diagnostic chain mimics a physician's reasoning for a more complete analysis. We compared the completeness of Chain-of-Thought (Wei et al., 2022) and Chain-of-Diagnosis in two aspects: symptom completeness and disease completeness. Figure 6 shows Chain-of-Diagnosis is more complete than Chain-of-Thought with ordinary prompts.

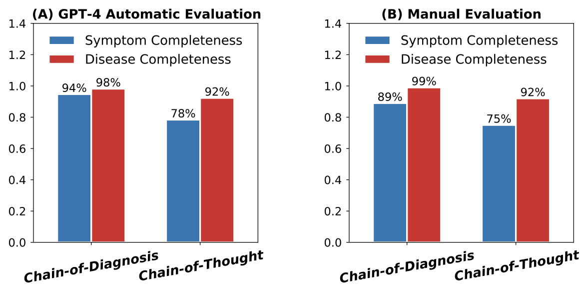


Figure 6: Evaluation results of completeness. *Disease Completeness* denotes the percentage of analyses covering all diseases. *Symptom Completeness* denotes the percentage covering all patient symptoms. **Left:** We sampled 2k entries from CoD data with varied prompt-driven analyses evaluated by GPT-4. **Right:** We sampled 100 entries and conducted manual evaluations. See Appendix L for details.

### 4.8 Ablation Study

We conduct two ablation experiments with CoD training data: (1) *w/o Confidence for Decision*,

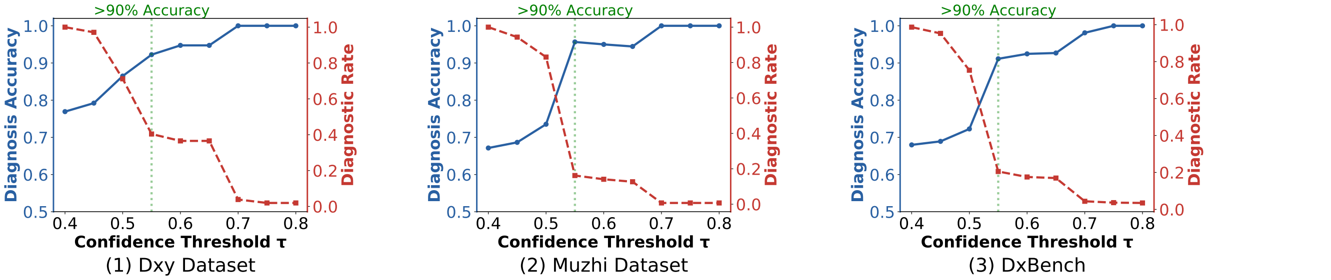


Figure 4: Relationship between confidence and accuracy. DiagnosisGPT was provided all symptoms ($\mathcal{S}_{\textbf{exp}} \cup \mathcal{S}_{\textbf{imp}}$) for direct diagnosis (without inquiry). **Diagnosis Accuracy** represents the accuracy of diagnoses exceeding the threshold $\tau$. **Diagnosis Rate** is the proportion of diagnoses exceeding $\tau$.

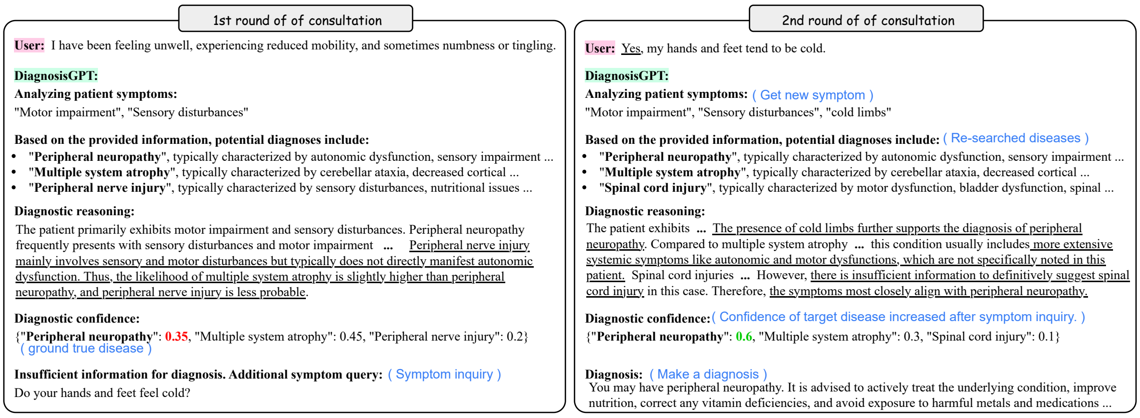


Figure 5: A diagnosis case by DiagnosisGPT, which made the correct diagnosis by inquiring symptoms.

which learns to directly generate decisions like other LLMs, and (2) DiagnosisGPT_*baseline*, which directly learns to predict diseases. The results in Table 7 show that without confidence, the model still tends to make direct diagnoses, and direct disease prediction does not outperform the CoD approach.

| Model | DxBench | | Muzhi Dataset | |
|---|---|---|---|---|
| | **Acc.** | $n$ | **Acc.** | $n$ |
| **DiagnosisGPT** ($\tau = 0.5$) | 62.4 | 1.8 | 64.1 | 1.4 |
| w/o Confidence for Decision | 59.2 | 0.8 | 59.1 | 0.6 |
| DiagnosisGPT_*baseline* | 55.2 | 0.0 | 58.4 | 0.0 |

Table 7: Ablation results. All ablation models are retrained using the CoD training data. *w/o Confidence for Decision* signifies that the model directly generates the decision, akin to other LLMs. DiagnosisGPT_*baseline* denotes the models that directly learn disease prediction without symptom inquiry from the CoD training data.

### 4.9 Case Study

Figure 5 presents a diagnostic case using DiagnosisGPT. DiagnosisGPT can systematically output its diagnostic reasoning process. It first summarizes the user's symptom information, then recall potential diseases, and then perform the diagnostic analysis, ultimately providing the confidence. As shown in the first round of replies, the highest confidence level is 0.45, below the threshold, prompting the model to inquire about symptoms. When the patient responds to the symptom inquiry, the probability of the target disease significantly increases, leading DiagnosisGPT to confirm and makes a correct diagnosis.

## 5 Conclusion

In this paper, we propose the **Chain of Diagnosis (CoD)** to enhance the interpretability of LLMs for disease automatic diagnosis. Using CoD, we developed DiagnosisGPT, an LLM capable of diagnosing of 9,604 diseases for *validating CoD*. Unlike other LLMs, DiagnosisGPT provides diagnostic confidence and performs open-ended reasoning using its own disease database. Experiments show that the diagnostic capabilities of DiagnosisGPT surpass those of other LLMs. Furthermore, higher accuracy can be achieved by adjusting the diagnostic threshold values. This demonstrates CoD's ability to balance diagnostic effectiveness and efficiency. We believe the CoD method can help advance research on interpretable LLMs.

## Limitations

Despite its promising performance in diagnostic tasks, DiagnosisGPT has several limitations that must be considered:

- **Limited Disease Coverage:** DiagnosisGPT is trained to identify only a specific set of diseases. This constraint means that the model's diagnostic capabilities are confined to this predefined list, and it may not recognize or provide accurate diagnoses for conditions that fall outside its training parameters. However, DiagnosisGPT is just a model for validating the effectiveness of CoD. In the future, CoD methodologies may scale to cover a wider range of disease scenarios, enhancing the interpretability of healthcare.

- **Synthetic Data Annotation:** The dataset used to train DiagnosisGPT relies on annotations created by LLMs. While utilizing LLMs for annotation is a cost-effective approach, it raises concerns about the quality and reliability of the data. Nevertheless, we believe it provides an efficient way to validate the feasibility of CoD. Specifically, generating only 48K synthetic data points can lead to stronger and more interpretable multi-turn decision-making abilities in real-world benchmarks. However, it is crucial to emphasize that for future applications of LLMs in healthcare, real-world data is indispensable. Synthetic data is only suitable for validating methodologies.

## References


2023. Sharegpt. https://sharegpt.com/. Accessed: 2023-10-16.

anthropic. 2024. Introducing the next generation of claude. https://www.anthropic.com/news/claude-3-family.

Saikat Barua. 2024. Exploring autonomous agents through the lens of large language models: A review. *arXiv preprint arXiv:2404.04442*.

Maciej Besta, Nils Blach, Ales Kubicek, and Gerstenberger. 2024. Graph of thoughts: Solving elaborate problems with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 17682–17690.

Junying Chen, Qingcai Chen, Dongfang Li, and Yutao Huang. 2022a. Sedr: Segment representation learning for long documents dense retrieval. *arXiv preprint arXiv:2211.10841*.

Junying Chen, Dongfang Li, Qingcai Chen, Wenxiu Zhou, and Xin Liu. 2022b. Diaformer: Automatic diagnosis via symptoms sequence generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 4432–4440.

Junying Chen, Ruyi Ouyang, Anningzhe Gao, Shunian Chen, Guiming Hardy Chen, Xidong Wang, Ruifei Zhang, Zhenyang Cai, Ke Ji, Guangjun Yu, et al. 2024. Huatuogpt-vision, towards injecting medical visual knowledge into multimodal llms at scale. *arXiv preprint arXiv:2406.19280*.

Junying Chen, Xidong Wang, Anningzhe Gao, Feng Jiang, Shunian Chen, Hongbo Zhang, Dingjie Song, Wenya Xie, Chuyi Kong, Jianquan Li, et al. 2023a. Huatuogpt-ii, one-stage training for medical adaption of llms. *arXiv preprint arXiv:2311.09774*.

Zhihong Chen, Feng Jiang, Junying Chen, Tiannan Wang, Fei Yu, Guiming Chen, Hongbo Zhang, Juhao Liang, Chen Zhang, Zhiyi Zhang, et al. 2023b. Phoenix: Democratizing chatgpt across languages. *arXiv preprint arXiv:2304.10453*.

Arsene Fansi Tchango, Rishab Goel, Zhi Wen, Julien Martel, and Joumana Ghosn. 2022. Ddxplus: A new dataset for automatic medical diagnosis. *Advances in Neural Information Processing Systems*, 35:31306–31318.

Zhengxu Hou, Bang Liu, Ruihui Zhao, Zijing Ou, Yafei Liu, Xi Chen, and Yefeng Zheng. 2021. Imperfect also deserves reward: Multi-level and sequential reward modeling for better dialog management. In *NAACL*.

Zhenyu Hou, Yukuo Cen, Ziding Liu, Dongxue Wu, Baoyan Wang, Xuanhe Li, Lei Hong, and Jie Tang. 2023. Mtdiag: an effective multi-task framework for automatic diagnosis. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 14241–14248.

Zhiyuan Hu, Chumin Liu, Xidong Feng, Yilun Zhao, See-Kiong Ng, Anh Tuan Luu, Junxian He, Pang Wei Koh, and Bryan Hooi. 2024. Uncertainty of thoughts: Uncertainty-aware planning enhances information seeking in large language models. *Preprint*, arXiv:2402.03271.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2023. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *Preprint*, arXiv:2311.05232.

Ke Ji, Junying Chen, Anningzhe Gao, Wenya Xie, Xiang Wan, and Benyou Wang. 2024. Llms could autonomously learn without external supervision. *arXiv preprint arXiv:2406.00606*.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las

Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2024. Mixtral of experts. *Preprint*, arXiv:2401.04088.

Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. 2021. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *Applied Sciences*, 11(14):6421.

Alistair EW Johnson, Tom J Pollard, Lu Shen, Li-wei H Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. 2016. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3(1):1–9.

Hao-Cheng Kao, Kai-Fu Tang, and Edward Chang. 2018. Context-aware symptom checking for disease diagnosis using hierarchical reinforcement learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906*.

Dongfang Li, Baotian Hu, Qingcai Chen, Weihua Peng, and Anqi Wang. 2020. Towards medical machine reading comprehension with structural knowledge and plain text. In *Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP)*, pages 1427–1438.

Kangenbei Liao, Qianlong Liu, Zhongyu Wei, Baolin Peng, Qin Chen, Weijian Sun, and Xuanjing Huang. 2020. Task-oriented dialogue system for automatic disease diagnosis via hierarchical reinforcement learning. *arXiv preprint arXiv:2004.14254*.

Kangenbei Liao, CHENG ZHONG, Wei Chen, Qianlong Liu, Baolin Peng, Xuanjing Huang, et al. 2021. Task-oriented dialogue system for automatic disease diagnosis via hierarchical reinforcement learning.

Zachary C Lipton. 2018. The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue*, 16(3):31–57.

Wenge Liu, Yi Cheng, Hao Wang, Jianheng Tang, Yafei Liu, Ruihui Zhao, Wenjie Li, Yefeng Zheng, and Xiaodan Liang. 2022. " my nose is running."" are you also coughing?": Building a medical diagnosis agent with interpretable inquiry logics. *arXiv preprint arXiv:2204.13953*.

OpenAI. 2022. Introducing chatgpt. https://openai.com/blog/chatgpt.

OpenAI. 2023. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Yu-Shao Peng, Kai-Fu Tang, Hsuan-Tien Lin, and Edward Chang. 2018. Refuel: Exploring sparse features in deep reinforcement learning for fast disease diagnosis. *Advances in neural information processing systems*, 31.

Machel Reid, Nikolay Savinov, Denis Teplyashin, and et al. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *Preprint*, arXiv:2403.05530.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *CoRR*, abs/1908.10084.

Thomas Savage, Ashwin Nayak, Robert Gallo, Ekanath Rangan, and Jonathan H Chen. 2024. Diagnostic reasoning prompts reveal the potential for large language model interpretability in medicine. *NPJ Digital Medicine*, 7(1):20.

Chaitanya Shivade, Preethi Raghavan, Eric Fosler-Lussier, Peter J Embi, Noemie Elhadad, Stephen B Johnson, and Albert M Lai. 2014. A review of approaches to identifying patient phenotype cohorts using electronic health records. *Journal of the American Medical Informatics Association*, 21(2):221–230.

Yu Sun, Shuohuan Wang, Shikun Feng, Siyu Ding, Chao Pang, Junyuan Shang, Jiaxiang Liu, Xuyi Chen, Yanbin Zhao, Yuxiang Lu, et al. 2021. Ernie 3.0: Large-scale knowledge enhanced pre-training for language understanding and generation. *arXiv preprint arXiv:2107.02137*.

Kai-Fu Tang, Hao-Cheng Kao, Chun-Nan Chou, and Edward Y Chang. 2016. Inquire and diagnose: Neural symptom checking ensemble using deep reinforcement learning. In *NIPS workshop on deep reinforcement learning*.

Milene Santos Teixeira, Vinícius Maran, and M. Dragoni. 2021. The interplay of a conversational ontology and ai planning for health dialogue management. *Proceedings of the 36th Annual ACM Symposium on Applied Computing*.

Xidong Wang, Nuo Chen, Junyin Chen, Yan Hu, Yidong Wang, Xiangbo Wu, Anningzhe Gao, Xiang Wan, Haizhou Li, and Benyou Wang. 2024. Apollo: Lightweight multilingual medical llms towards democratizing medical ai to 6b people. *arXiv preprint arXiv:2403.03640*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Zhongyu Wei, Qianlong Liu, Baolin Peng, Huaixiao Tou, Ting Chen, Xuan-Jing Huang, Kam-Fai Wong, and Xiang Dai. 2018. Task-oriented dialogue system for automatic diagnosis. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 201–207.

Yuan Xia, Jingbo Zhou, Zhenhui Shi, Chao Lu, and Haifeng Huang. 2020. Generative adversarial regularized mutual information policy gradient framework for automatic diagnosis. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 1062–1069.

Wenya Xie, Qingying Xiao, Yu Zheng, Xidong Wang, Junying Chen, Ke Ji, Anningzhe Gao, Xiang Wan, Feng Jiang, and Benyou Wang. 2024. Llms for doctors: Leveraging medical llms to assist doctors, not replace them. *arXiv preprint arXiv:2406.18034*.

Honglin Xiong, Sheng Wang, Yitao Zhu, Zihao Zhao, Yuxiao Liu, Qian Wang, and Dinggang Shen. 2023. Doctorglm: Fine-tuning your chinese doctor is not a herculean task. *arXiv preprint arXiv:2304.01097*.

Lin Xu, Qixian Zhou, Ke Gong, Xiaodan Liang, Jianheng Tang, and Liang Lin. 2019a. End-to-end knowledge-routed relational dialogue system for automatic diagnosis. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 7346–7353.

Lin Xu, Qixian Zhou, Ke Gong, Xiaodan Liang, Jianheng Tang, and Liang Lin. 2019b. End-to-end knowledge-routed relational dialogue system for automatic diagnosis. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 7346–7353.

Ming Xu. 2023. Medicalgpt: Training medical gpt model. https://github.com/shibing624/MedicalGPT.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2024. Tree of thoughts: Deliberate problem solving with large language models. *Advances in Neural Information Processing Systems*, 36.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. React: Synergizing reasoning and acting in language models. *Preprint*, arXiv:2210.03629.

Alex Young, Bei Chen, Chao Li, Chengen Huang, Ge Zhang, Guanwei Zhang, Heng Li, Jiangcheng Zhu, Jianqun Chen, Jing Chang, Kaidong Yu, Peng Liu, Qiang Liu, Shawn Yue, Senbin Yang, Shiming Yang, Tao Yu, Wen Xie, Wenhao Huang, Xiaohui Hu, Xiaoyi Ren, Xinyao Niu, Pengcheng Nie, Yuchi Xu, Yudong Liu, Yue Wang, Yuxuan Cai, Zhenyu Gu, Zhiyuan Liu, and Zonghong Dai. 2024. Yi: Open foundation models by 01.ai. *Preprint*, arXiv:2403.04652.

Fei Yu, Anningzhe Gao, and Benyou Wang. 2024. Ovm, outcome-supervised value models for planning in mathematical reasoning. *Preprint*, arXiv:2311.09724.

Hongyi Yuan and Sheng Yu. 2024. Efficient symptom inquiring and diagnosis via adaptive alignment of reinforcement learning and classification. *Artificial Intelligence in Medicine*, 148:102748.

Guangtao Zeng, Wenmian Yang, Zeqian Ju, Yue Yang, Sicheng Wang, Ruisi Zhang, Meng Zhou, Jiaqi Zeng, Xiangyu Dong, Ruoyu Zhang, et al. 2020. Meddialog: Large-scale medical dialogue datasets. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9241–9250.

Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Jiafeng Guo, Min Zhang, and Shaoping Ma. 2021. Optimizing dense retrieval model training with hard negatives. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1503–1512.

Hongbo Zhang, Junying Chen, Feng Jiang, Fei Yu, Zhihong Chen, Jianquan Li, Guiming Chen, Xiangbo Wu, Zhiyi Zhang, Qingying Xiao, et al. 2023. Huatuogpt, towards taming language model to be a doctor. *arXiv preprint arXiv:2305.15075*.

Xinyan Zhao, Liangwei Chen, and Huanhuan Chen. 2021. A weighted heterogeneous graph-based dialog system. *IEEE Transactions on Neural Networks and Learning Systems*, 34(8):5212–5217.

## A Ethical Statement

We use synthetic case data generated from medical encyclopedia datasets, ensuring that our data does not present any privacy or ethical concerns. However, the data and models may contain hallucinations or inaccuracies. Therefore, we will impose strict restrictions on the download and use of our model and dataset. The use of these resources will be limited strictly to research purposes. They are not permitted for use in clinical or other industry applications, where such inaccuracies could lead to unintended consequences. We emphasize the ethical responsibility of users to comply with these restrictions to safeguard the safety and integrity of their applications.

## B Related Work

**LLMs for Medical Scenarios** The success of models like ChatGPT (OpenAI, 2022) has inspired research into their application in healthcare, resulting in medical-specific LLMs such as DoctorGLM (Xiong et al., 2023), MedicalGPT (Xu, 2023), DotaGPT (Xie et al., 2024), HuatuoGPT (Zhang et al., 2023; Chen et al., 2023a, 2024), and Apollo (Wang et al., 2024). Despite their focus on medical knowledge, these models have limited capabilities in automating medical diagnoses.

**Automated Diagnosis Task** Medical diagnosis, a key AI application in healthcare (Chen et al., 2022b; Li et al., 2020; Ji et al., 2024; Peng et al., 2018; Kao et al., 2018), has predominantly utilized reinforcement learning (RL). Pioneering works include (Tang et al., 2016), who introduced neural symptom checking using RL. Subsequent advancements include hierarchical RL for diagnostic and contextual decisions (Liao et al., 2021), Deep Q-networks for symptom collection from patient interactions (Wei et al., 2018), and incorporation of medical knowledge into RL policy learning (Xu et al., 2019b). Two-level hierarchical RL (Liao et al., 2020), policy gradient frameworks with Generative Adversarial Networks (Xia et al., 2020), and customization of RL models using multi-level rewards and dialogue data (Hou et al., 2021; Teixeira et al., 2021) have further enhanced diagnostic accuracy. (Chen et al., 2022b) and (Hou et al., 2023) conceptualizes automatic diagnosis as a sequence generation task. However, these models are limited by predefined symptoms and diseases, and cannot support open-ended consultations.

**Reasoning of LLMs** LLMs show promise in complex tasks such as mathematical reasoning (Chen et al., 2023b; Yu et al., 2024). To harness their reasoning abilities, CoT(Wei et al., 2022) is proposed with intermediate steps, and Tree-of-Thought (ToT)(Yao et al., 2024) using DFS/BFS for enhanced reasoning paths. Graph of Thoughts (GoT) (Besta et al., 2024) is introduced for intricate problems. ReAct (Yao et al., 2023) combines reasoning with actions. Uncertainty of Thoughts (UoT) (Hu et al., 2024) improves decision-making by simulating multiple requests for information gain.

## C DxBench Distribution

The data distribution in DxBench dataset is illustrated in Figure 7. We categorize the data distribution according to the medical departments responsible for diagnosing the diseases. The data shows a relatively balanced distribution across different departments. Notably, the Dermatovenereology department has the highest number of entries with 121 cases, while the Infectious Diseases and Immunology department has the fewest, with 27 cases.

## D The prompt for LLM Diagnosis

The prompt for LLM diagnosis is shown in Table 8. We instruct the LLMs to determine whether a diagnosis can be made. If a diagnosis is possible, the LLMs output the diagnosed disease. Otherwise, the LLMs query the user with questions regarding a specific symptom. Additionally, we compared two different prompts: the *Chain-of-Thought* prompt (Table 21) and the *Multi-Choice* prompt (Table 22).

## E Patient Simulator for Evaluation

To evaluate the automatic diagnostic capabilities of LLMs, we instruct GPT-4 to play the role of a patient. Initially, we provide explicit symptoms $S_{exp}$ as input for the model to diagnose. If the LLMs ask questions, the patient GPT will respond using a simulated patient prompt, as shown in Figure 9.

## F Prompt of Data Synthesis

We constructed a disease database encompassing 9,604 diseases. Each disease entry includes four fields: "disease name", "overview", "symptoms", and "treatment". For each disease, we used the prompt shown in Figure 10 to generate five patient cases with GPT-4, ensuring that each case study exhibits distinct typical characteristics.

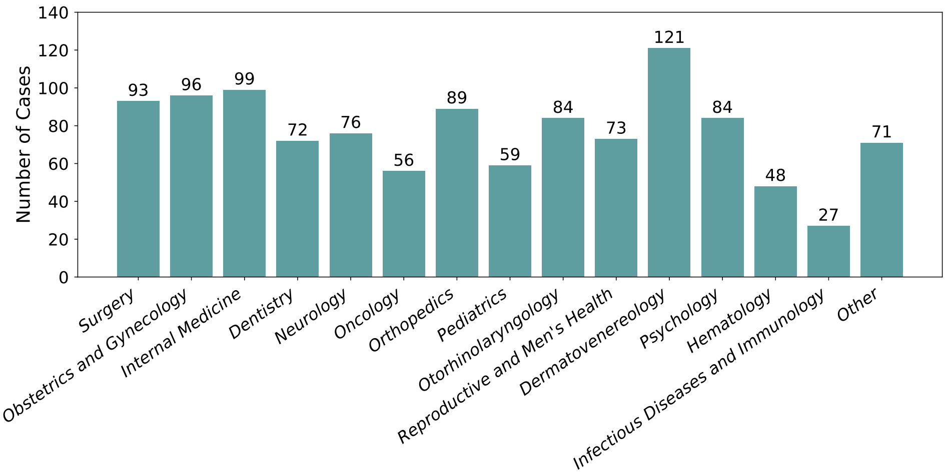


Figure 7: Data distribution across different departments in the DxBench dataset.

## G Prompt of CoD

To generate CoD training data, we prompt GPT-4 to construct CoD dialogue data based on patient case data. This involves the following 8 prompts:

**Prompt 1: Patient Self-report Prompt (Role: Patient)** As shown in figure 11, the patient self-report prompt is used to generate the user's initial question $q_1$ based on the patient's explicit symptoms, primarily expressing the patient's chief complaint.

**Prompt 2: Reasoning Prompt (Role: Diagnosis)** When provided with the known symptoms $S$ of a patient and the candidate diseases $D'$, the reasoning prompt, as illustrated in Figure 12, is utilized to generate the reasoning process $T$ and the confidence distribution $C$.

**Prompt 3: Rethinking Prompt (Role: Diagnosis)** If the generated $C$ does not meet the condition $\max C \setminus c_{d_t} > \tau$, the rethinking prompt, as shown in Figure 13, is used to have GPT4 regenerate a valid diagnosis $T$ and $C$.

**Prompt 4: Doctor Diagnosis Prompt (Role: Doctor)** If $\max C > \tau$, we prompt GPT-4 to generate a response regarding the diagnostic result. The prompt used is shown in Figure 14. The disease database information will be provided to generate more reliable suggestions. Once the diagnostic response is generated, the data generation process concludes.

**Prompt 5: Symptom Generation Prompt (Doctor)** If $\max C \leq \tau$, we will have the LLM generate the symptom $s_{gen}$ it wants to inquire about, using the prompt shown in Figure 15. Then, we will select the inquired symptom $s_t$ from $S_{imp} \cup \{s_{gen}\}$ based on $H(C|s)$.

**Prompt 6: Doctor Inquiry Prompt (Role: Doctor)** After confirming the symptom $s_t$, the Doctor Inquiry Prompt, shown in Figure 16, generates questions regarding the symptom.

**Prompt 7: Symptom Assessment Prompt (Role: Patient)** As shown in Figure 17, the symptom assessment prompt is used to determine whether the patient exhibits the symptoms inquired about by the doctor.

**Prompt 8: Patient Response Prompt (Role: Patient)** The Patient Response Prompt, as shown in Figure 18, is used to generate verbal responses from patients regarding the symptom of inquiry. Then, $s_t$ will be added to the doctor's known symptoms $S$, initiating the next doctor's response generation.

## H Building of DxBench

We used MedDialog (Zeng et al., 2020), a publicly available dataset with real doctor-patient dialogues from various departments, to create a realistic evaluation dataset. Despite its chaotic nature, we filtered 3,121 cases with clear dialogue and definite disease diagnoses. Then, using prompts designed to extract structured case information, as shown in Figure 19, we extracted structured case information from these cases. We further filtered out cases where the diseases were not included in the database or where there was insufficient symptom information, resulting in 1,429 cases. Finally, we manually filtered 1,148 high-quality cases, verifying the accuracy and completeness of symptom extraction, to construct DxBench. Additionally, for each case, we identified the two most relevant diseases using the disease retriever of DiagnosisGPT to form three candidate diseases. The construction process is detailed in Appendix H.

## I Why MIMIC Evaluation is Not Applicable

CoD primarily addresses the task of automatic diagnosis, which is a multi-turn medical dialogue decision-making task. MIMIC (Johnson et al., 2016) does not include multi-turn dialogue medical decision-making tasks. Automatic diagnosis requires complete dialogue data between doctors and patients, including medical inquiries about symptoms and the process of making a diagnosis. To address the scarcity of evaluation datasets, we have developed DxBench, a larger-scale automatic diagnosis evaluation set, for the community to advance the development of this field.

## J Data Source of Disease Encyclopedia

We obtained the Disease Encyclopedia from Wikipedia (https://en.wikipedia.org) and Xiaohe Yidian (https://www.xiaohe.cn).

## K Case Study

Below, two detailed diagnostic cases will be provided to better understand the response of our model.

---

**User**

I recently found a lump in my abdomen, and the pain is severe. Can you help me?

**DiagnosisGPT**

**Analyzing patient symptoms:**
"abdominal lump", "severe pain"

**Based on the information provided, the likely diagnoses include:**
- **Greater omentum cyst**, common symptoms include an abdominal mass or progressive abdominal enlargement, a feeling of heaviness in the abdomen when lying on the back, and severe abdominal pain may occur when there is concurrent intestinal torsion or obstruction.
- **Intestinal tumor**, common symptoms include an abdominal lump, severe pain, anemia, rectal bleeding, frequent and malformed stools, and fatigue with weight loss.
- **Recurrent retroperitoneal tumor**, common symptoms include an abdominal lump, abdominal distension, abdominal pain, dull pain, and severe pain.

**Diagnostic reasoning:**
The patient presents symptoms of abdominal mass, severe pain, and rectal bleeding. These symptoms may be associated with various intra-abdominal conditions. The abdominal mass and severe pain may be direct manifestations of some abdominal tumor-related disease, while rectal bleeding could indicate involvement of the in- testines. When considering candidate diseases, the likelihood of colorectal tumors is high as they can account for the occurrence of all these symptoms. Mesenteric cysts typically present as abdominal masses but may not necessarily cause significant pain or rectal bleeding if they are small or located in specific positions. Recurrent retroperitoneal tumors, depending on their origin and type, could also result in sim- ilar symptoms, especially if they affect the intestines or other abdominal structures. Considering all factors, colorectal tumors are more closely associated with symptoms such as rectal bleeding due to their direct involvement with the intestines. Recur- rent retroperitoneal tumors are less likely than colorectal tumors but more likely than mesenteric cysts, as the latter typically show a weaker association with pain and rectal bleeding.

**Diagnostic confidence:**
- **"Greater Omental cyst"**: 0.15,
- **"Intestinal tumor"**: 0.4,
- **"Recurrent retroperitoneal tumor"**: 0.45

**Insufficient information for diagnosis. Additional symptom query:**
Have you noticed any abnormalities or blood streaks in your stool color recently?

**User**

I've noticed some bloody stools recently, and the color doesn't seem right.

**DiagnosisGPT**

**Analyzing patient symptoms:**
"abdominal lump", "severe pain", "bloody stool"

**Based on the information provided, the likely diagnoses include:**
- **Intestinal tumor**, ...
- **Greater omentum cyst**, ...
- **Recurrent retroperitoneal tumor**, ...

(The disease retriever identified the same three diseases as last time, but their priority has changed.)

**Diagnostic reasoning:**
The patient presents symptoms of abdominal mass, severe pain, and rectal bleeding. These symptoms may be associated with various intra-abdominal conditions. The abdominal mass and severe pain may be direct manifestations of some abdominal tumor-related disease, while rectal bleeding could indicate involvement of the in- testines. When considering candidate diseases, the likelihood of colorectal tumors is high as they can account for the occurrence of all these symptoms. Mesenteric cysts typically present as abdominal masses but may not necessarily cause significant pain or rectal bleeding if they are small or located in specific positions. Recurrent retroperitoneal tumors, depending on their origin and type, could also result in sim- ilar symptoms, especially if they affect the intestines or other abdominal structures. Considering all factors, colorectal tumors are more closely associated with symptoms such as rectal bleeding due to their direct involvement with the intestines. Recur- rent retroperitoneal tumors are less likely than colorectal tumors but more likely than mesenteric cysts, as the latter typically show a weaker association with pain and rectal bleeding.

**Diagnostic confidence:**
- **"Intestinal tumor"**: 0.6,
- **"Greater Omental cyst"**: 0.15,
- **"Recurrent retroperitoneal tumor"**: 0.25

**Diagnosis**

Based on the symptoms and rectal bleeding you described, there is a possibility of a **Intestinal tumor**. I recommend you to visit a hospital for further examinations, including endoscopy and imaging tests, to confirm the diagnosis. Early-stage Intestinal tumors can be treated with surgical removal, while malignant tumors may require additional treatments such as radiation therapy or chemotherapy. Wishing you a speedy recovery.

Case 1: Interaction between an intestinal tumor patient and the DiagnosisGPT model.

**User**

I recently discovered that I have developed an inflammatory lump, and I have also been running a fever, feeling generally unwell.

**DiagnosisGPT**

**Analyzing patient symptoms:**
"inflammatory mass", "fever"

**Based on the information provided, the likely diagnoses include:**
- **Granulomatous lymphadenitis**, common symptoms include inflammatory mass, lymph node enlargement and pain, fever.
- **Geotrichosis**, common symptoms include fever, chronic cough, sputum production and hemoptysis, stomach pain, diarrhea, cheesy white pseudomembrane on the oral mucosa, tongue, and palate, erythematous base with a burning sensation, involvement of the pharynx, larynx, and their junctions, skin erythema and exudation, cutaneous candidiasis in skin folds, tumor-like nodules, granulomas, itching and pain, edema, erythema, epithelial desquamation, and crust formation.
- **Suppurative mastitis**, common symptoms include fever, localized breast pain, lump, breast fistula with abscess formation that may rupture internally or externally.

**Diagnostic reasoning:**
The primary symptoms of the patient are inflammatory masses and fever. Without more detailed symptom descriptions, lymphadenitis granulomatous, geotrichosis, and suppurative mastitis can all present with these symptoms. Lymphadenitis granulomatous often causes lymph node enlargement and may be accompanied by inflammatory masses. Geotrichosis, an infection caused by Geotrichum, can also cause systemic symptoms such as fever and may lead to granuloma formation in immunocompromised individuals. Suppurative mastitis primarily occurs in breastfeeding women, with symptoms including pain in the breast area, masses, and fever. Given the presence of inflammatory masses, if the patient is a breastfeeding woman, the likelihood of suppurative mastitis increases. Due to the lack of explicit latent symptoms, negating symptoms information, and detailed background information of the patient (such as gender, age, occupation, etc.), the specific diagnosis requires further clinical examination and auxiliary examination data.

**Diagnostic confidence:**
- **Granulomatous lymphadenitis**: 0.3,
- **Geotrichosis**: 0.3,
- **Suppurative mastitis**: 0.4

**Insufficient information for diagnosis. Additional symptom query:**
Do you have a headache?

**User**

I have a bit of a headache, but it's not very serious.

**DiagnosisGPT**

**Analyzing patient symptoms:**
"inflammatory mass", "fever", "headache"

**Based on the information provided, the likely diagnoses include:**
- **Granulomatous lymphadenitis**, common symptoms include inflammatory mass, lymph node enlargement and pain, fever.
- **Occipital indurative folliculitis**, common symptoms include fever, hard rashes, swollen hair follicles, pain, and poor sleep.
- **Hair cysts**, common symptoms include cystic masses on the scalp, often solitary, more common in women.

**Diagnostic reasoning:**
The patient's primary symptoms are inflammatory masses accompanied by fever, which typically indicate an inflammatory response in a certain area of the body. Headaches may be a systemic symptom caused by the body's overall inflammatory response. Common symptoms of lymphadenitis granulomatosa include fever and localized lymph node enlargement, and the local inflammatory masses may be related to the swelling of the lymph nodes. Suboccipital kerion celsi is characterized by inflammation and induration of hair follicles in a specific area of the head and typically does not cause systemic symptoms such as fever. Pilar cysts are cystic formations within hair follicles, usually accompanied by localized inflammation, but are unlikely to cause fever and systemic symptoms. Considering these candidate conditions and the patient's symptoms, a preliminary judgment can be made accordingly.

**Diagnostic confidence:**
- **Granulomatous lymphadenitis**: 0.6,
- **Occipital indurative folliculitis**: 0.1,
- **Hair cysts**: 0.3

**Diagnosis**

It seems you may have **granulomatous inflammation**. I recommend that you see a doctor as soon as possible, as you may need some blood tests and imaging exams to confirm the diagnosis. The doctor will develop a detailed treatment plan based on the examination results. Meanwhile, make sure to rest, drink plenty of water, and avoid overexertion.

Case 2: Interaction between an granulomatous inflammation patient and the DiagnosisGPT model.

## L Evaluating results of completeness

To compare the diagnostic completeness of the "chain of thought" and the "chain of diagnosis", we used GPT-4 to generate data for the chain of thought, with the prompt displayed in Figure 20.

We conducted both automatic and manual evaluations of the data for the chain of thought and the chain of diagnosis to verify their completeness. The automatic evaluation utilized the GPT-4 model, with the evaluation prompt shown in Figure 23. Manual testing utilized the same criteria.

## M Performance of Disease Retriever

We allocated 10% of the data as a validation set to evaluate retrieval performance. Table 8 shows the retrieval performance of diseases under Diagnosis on the validation set. It can be seen that the top 3 diseases achieve a recall rate of 73%, indicating that most diseases can be effectively excluded.

| Metric | Value |
|---|---|
| MRR@100 | 0.6926 |
| Recall@3 | 0.7357 |
| Recall@5 | 0.7807 |
| Recall@10 | 0.8430 |
| Recall@30 | 0.9047 |
| Recall@50 | 0.9313 |
| Recall@100 | 0.9573 |

Table 8: Performance of the Disease Retriever on the Validation Set

## N Standard Errors of Result

We report the standard errors of the results from our model in Table 9. The standard errors were obtained by conducting five random experiments.

## O Review of Synthetic Cases by Medical Experts

To verify the quality of the synthetic cases, we had two licensed physicians review the data. Each physician was given 50 randomly sampled synthetic cases and asked to assess whether any cases posed a risk of errors. Based on their feedback, they identified that out of the 100 cases, only 6 might be incorrect, as the symptom information was less likely to be associated with the respective diseases. This suggests that synthesizing cases from a medical encyclopedia is a fairly reliable method.

## P Impact

### P.1 Positive Impact

- **Promotes medical AI development:** DiagnosisGPT promotes the development of medical AI, as diagnostics are crucial in healthcare AI. Accurate diagnostic capabilities enhance patient outcomes and streamline clinical processes.

- **Improves interpretability in healthcare:** DiagnosisGPT improves the interpretability of medical AI by utilizing a disease retriever function and knowledge base integration. This increased interpretability builds trust in AI systems among healthcare providers and patients. By making the diagnostic process more transparent, DiagnosisGPT helps users understand the reasoning behind AI-generated suggestions, fostering greater confidence in AI-assisted medical practices.

- **Addresses privacy concerns in medical cases:** DiagnosisGPT offers a solution to privacy issues prevalent in medical case handling by constructing cases using a knowledge base, thereby eliminating patient privacy concerns. This approach also alleviates the problem of data scarcity.

- **Assists healthcare professionals:** DiagnosisGPT assists healthcare professionals by rapidly collecting patient symptom information and providing preliminary diagnoses. This capability enables medical practitioners to save time and focus on more complex aspects of patient care.

### P.2 Potential Negative Impact

The development of DiagnosisGPT raises several potential risks.

- **Risk of Misdiagnosis:** Despite the promising results shown by DiagnosisGPT in diagnosis, it is crucial to underscore that at this stage, it should not be used to provide any medical advice. There is a possibility that it could provide incorrect interpretations or inaccurate diagnoses. Considering the nature of this field, our model and data will only be available for download by researchers. Our model will not be available for public use.

- **Data Privacy and Ethics:** The diagnostic field may involve ethical issues related to patient privacy. To address this, we use synthetic data. The training data for CoD is entirely generated by GPT-4, ensuring that there are no privacy or ethical concerns. As for DxBench, we constructed it using open-source licensed datasets, ensuring compliance with ethical standards.

| Model | DxBench | Muzhi Dataset | Dxy Dataset |
|---|---|---|---|
| GPT4-0125-preview | 62.5 $\pm 2.1$ | 57.7 $\pm 2.4$ | 65.4 $\pm 0.5$ |
| **DiagnosisGPT** ($\tau$ = 0.6) | 64.5 $\pm 1.8$ | 65.5 $\pm 2.1$ | 75.4 $\pm 0.3$ |

Table 9: Statistical Results of Standard Errors.

**The prompt for LLM Diagnosis**

```
Initial input:
You are a professional physician tasked with diagnosing a patient based on their symptom
information. I will provide you with information on possible diseases, and you will need to
carefully consider which of the candidate diseases the patient might have.

Patient symptom information is marked by <symptoms>, and candidate diseases by
<candidate_diseases>.

<symptoms> {Known_symptoms} <symptoms>

<candidate_diseases> {candidate_diseases} <candidate_diseases>

If you believe a diagnosis can be made, select the most likely disease from <candidate_diseases>
(choose only one). Example output:
{"judge": true, "disease": "common cold"}

If you believe the information on symptoms is insufficient, ask the patient for more symptom
information, noting that you can only ask about one symptom. Example output:
{"judge": false, "symptom": "Do you have a lack of appetite?"}

Please output in JSON format.

Input after patient response:
The patient's response will be marked by <Patient>. The hints I give you are marked by <Hint>.

<Patient> {patient_response} <Patient>

<Hint>Please, based on the patient's response, decide now whether a diagnosis can be
made.

If you believe a diagnosis can be made, select the most likely disease from <candidate_diseases>
(choose only one). Example output:
{"judge": true, "disease": "common cold"}

If you believe the information on symptoms is insufficient, ask the patient for more symptom
information, noting that you can only ask about one symptom. Example output:
{"judge": false, "symptom": "Do you have a lack of appetite?"}

Please output in JSON format.<Hint>
```

Figure 8: The prompt for LLM Diagnosis. Known_symptoms represents the symptoms currently known by the LLM. candidate_diseases represents the list of candidate diseases $D$. candidate_diseases represents the response of the patient.

**Simulated patient prompt**

```
You are a patient, here are your symptom details: {Symptoms}

Your actual disease is {disease}.

You need to answer the doctor's question: {LLM_query}

Please answer the doctor's question based on your symptom information and disease, simply reply with "yes" or "no", and do not include any other content.
```

Figure 9: Simulated patient prompt for responding to questions posed by LLMs. The Symptoms represents all the symptom information of the case, $S_{exp} \cup S_{imp}$. The disease indicates the true disease of the case, $d_t$.

**A Knowledge-Base-Driven Approach to Medical Case Generation**

```
Disease: {disease name}
Overview of this disease: {overview}
Common symptoms of this disease include: {symptoms}
Please complete the following tasks based on the description above.
1. First, generate basic demographic information about the population affected by this disease: gender and age.
2. You need to construct five real cases concerning this disease. In these five cases, two should have only one main symptom, two should have two main symptoms, and one should have more than three main symptoms (main symptoms are the most noticeable ones). Each case should include 2-4 implicit symptoms (generally, symptoms that can be elicited by a doctor's questioning). Ensure each case is a typical example of this disease.

Output in JSON format, and only output the JSON content, do not output anything else. The example output is:
{"Basic Information": {"Gender": "Female", "Age": "Child"}, "Case 1": {"Main Symptoms": ["Symptom 1", "Symptom 2"], "Implicit Symptoms": ["Symptom 3", "Symptom 4", "Symptom 5"], "Case 2": "..."}
```

Figure 10: The prompt of synthesizing patient cases. {disease name}, {overview}, and {symptom} represent the corresponding information for diseases in the database.

**Patient Self-report Prompt** *Patient*

```
System Prompt:
I'd like you to pretend to be a patient and describe your condition to the doctor in the voice of a patient. Please avoid using overly technical terms.
Questions from the doctor will be marked with <Doctor>. The response of yourself will be marked with <Patient>. The hints I provide will be marked with <Hint>.

Query:
<Doctor> Hello, I'm a doctor. How can I help you? <Doctor>
<Hint> Your symptoms are: {explicit_syms}
If the symptoms include information about the patient's age and gender, such as elderly, female, etc., please inform the doctor.
Please reply in the patient's voice, only output the patient's words and nothing else. <Hint>
```

Figure 11: Patient Self-report Prompt. {explicit_syms} indicates $S_{exp}$.

**Reasoning Prompt** ***Diagnosis***

```
You are a professional physician tasked with diagnosing a patient based on provided symptom
information. You will be given a list of candidate diseases, and your role is to offer a
detailed diagnostic analysis and a confidence distribution of the candidate diseases for the
patient.

You need to first analyze the patient's condition and think about which of the candidate
diseases the patient might have.
Then, output the diagnostic confidence distribution of the candidate diseases in JSON format,
please output a dict rather than a list.

An output example is:
{"analysis":..., "distribution": {"Animal skin disease": 0.25, "Erythema ab igne": 0.2,
"Dermatitis": 0.55}}

The patient's explicit symptoms: {explicit_syms},
The patient's implicit symptoms: {implicit_syms},

Candidate diseases: {candidate_diseases}

Please first analyze the patient's condition, then output the probability distribution
of these diseases.
```

Figure 12: Reasoning Prompt. {explicit_syms} denotes $S_{exp}$. {implicit_syms} denotes the inquired symptoms $S \setminus S_{exp}$. {candidate_diseases} denotes the currently identified diseases $D'$.

**Rethinking Prompt** ***Diagnosis***

```
Your diagnostic analysis did not pass inspection because you assigned an high confidence
level to a potentially incorrect disease. Please reconsider your assessment and provide a new
diagnostic analysis along with the confidence distribution. Ensure that the output format
remains exactly the same.

An output example is:
{"analysis":..., "distribution": {"Animal skin disease": 0.25, "Erythema ab igne": 0.2,
"Dermatitis": 0.55}}
```

Figure 13: Rethinking Prompt.

**Doctor Diagnosis Prompt** ***Dcotor***

```
System Prompt:
Please play the role of a doctor to ask the patient about their condition or diagnose the
disease. The patient's responses will be marked with <Patient>. The response of yourself will
be marked with <Doctor>. The hints I provide will be marked with <Hint>.

Query:
{Chat_history}
<Hint>The doctor's diagnosis is marked by <diagnosis of disease>. The treatment method for this
diagnosed disease is marked by <treatment method>.
<diagnosis of disease>{disease_name}<diagnosis of disease>

<treatment method>{treatment}<treatment method>
Based on the information above and the historical conversation records, please diagnose the
patient and provide detailed recommendations.
Reply in the tone of a doctor, and do not start with the word 'doctor'.<Hint>
```

Figure 14: Doctor Diagnosis Prompt. {Chat_history} represents the previously generated conversation history. {disease_name} represents $d_t$. {treatment} represents the "treatment" of $d_t$ from the database.

**Symptom Generation Prompt** ***Dcotor***

```
You are now a professional physician, and you need to infer the next symptom to ask the patient
based on the following information.

The patient's explicit symptoms: {explicit_syms}
The patient's implicit symptoms: {implicit_syms}

The patient may currently have {predicted_disease}.

Please infer the next symptom to ask the patient, asking only one symptom that has not
been previously inquired.
The output format should be json, for example:
{"symptom": "headache"}
```

Figure 15: Symptom Generation Prompt. {explicit_syms} denotes $S_{exp}$. {implicit_syms} denotes the inquired symptoms $S \setminus S_{exp}$. {predicted_disease} represents the currently most likely disease $\underset{d \in \mathcal{D}'}{\mathrm{argmax}}\{c_d\}$.

**Doctor Inquiry Prompt** ***Dcotor***

```
System Prompt:
Please play the role of a doctor to ask the patient about their condition or diagnose the
disease. The patient's responses will be marked with <Patient>. The response of yourself will
be marked with <Doctor>. The hints I provide will be marked with <Hint>.

Query:
{Chat_history}
<Hint>Please inquire about the patient's condition based on these symptoms: {current_sym}
Please reply in the tone of a doctor, asking only one question in a conversational manner that
the patient can understand. Do not start with the word 'doctor'.<Hint>
```

Figure 16: Doctor Inquiry Prompt. {Chat_history} represents the previously generated conversation history. {current_sym} represents $s_t$.

**Symptom Assessment Prompt** ***Role: Patient***

```
You are now a professional physician. Please judge whether the patient has the symptom based on
the patient's information.

It is known the patient's main symptoms are {explicit_syms},
and the implicit symptoms are {implicit_syms}.

Please determine whether the patient has {choose_sym}.

Search in the patient's existing symptoms, paying attention to synonyms. If found,
output true; if not found, output false.
The output format is json, for example:
{"headache": true}
```

Figure 17: Symptom Assessment Prompt. {explicit_syms} and {implicit_syms} represent $S_{exp}$ and $S_{imp}$ respectively. {choose_sym} represents $s_t$, the symptom being inquired about.

**Patient Response Prompt** *Patient*

**System Prompt:**
I'd like you to pretend to be a patient and describe your condition to the doctor in the voice of a patient. Please avoid using overly technical terms.
Questions from the doctor will be marked with <Doctor>. The response of yourself will be marked with <Patient>. The hints I provide will be marked with <Hint>.

**Query:**
{Chat_history}
<Hint>Please answer the doctor's questions based on the information, note that you {do_or_do_not} have this symptom: {choose_sym}
Please respond in the tone of a patient, and avoid using overly technical terms. Do not begin with the word 'patient'.<Hint>

Figure 18: Patient Response Prompt. {Chat_history} represents the previously generated conversation history. {current_sym} represents $s_t$. {do_or_do_not} indicates whether the patient has the symptom determined by the Symptom Assessment Prompt.

**Prompt for extracting structured cases**

Below is the information of a patient case:
Diagnosed disease: {disease}
Patient's self-report: {self-report}
Conversation between patient and doctor: {conversation}

Please extract a structured case from the information above, which is a tuple containing (explicit symptoms (symptoms reported by the patient), implicit symptoms (symptoms inquired by the doctor), disease (disease diagnosed by the doctor)). Symptoms and diseases need to be in professional terminology. The output template is:
{"explicit_sympom": ["fever", ...], "implicit_sympom": ["cough", ...], "disease": "cold"}

Figure 19: Prompt for extracting structured cases. {disease} indicates the doctor's diagnosed disease. {self-report} indicates the patient's self-reported symptoms. {conversation} represents the content of the dialogue between the doctor and the patient.

**Chain of Thought Prompt**

You need to diagnose the patient based on their concerns. You can choose to continue asking the patient for more symptom information or to make a direct diagnosis.
Below is the patient's question: {patient_info}
The candidate diseases are: {candidate_disease}
Please first provide your analysis, then make your decision. If the patient speaks in English, please respond in English. If the patient speaks in Chinese, please respond in Chinese.

Figure 20: Prompt for generating medical diagnosis using Chain of Thought. {patient_info} indicates the first sentence of the patient inquiry, aligned with the COD data. {candidate_disease} indicates the candidate disease set.

**The *Chain-of-Though* prompt for LLM Diagnosis**

**Initial input:**
You are a professional physician tasked with diagnosing a patient based on their symptom information. I will provide you with information on possible diseases, and you will need to carefully consider which of the candidate diseases the patient might have.

Patient symptom information is marked by <symptoms>, and candidate diseases by <candidate_diseases>.

<symptoms> {Known_symptoms} <symptoms>

<candidate_diseases> {candidate_diseases} <candidate_diseases>

You need to consider whether a diagnosis can be made. First, you need to generate a chain-of-thought analysis.
If you believe a diagnosis can be made, select the most likely disease from <candidate_diseases> (choose only one). Example output:
{"CoT": "Your thought process", "judge": true, "disease": "common cold"}

If you believe the information on symptoms is insufficient, ask the patient for more symptom information, noting that you can only ask about one symptom. Example output:
{"CoT": "Your thought process", "judge": false, "symptom": "Do you have a lack of appetite?"}

Please output in JSON format.

**Input after patient response:**
The patient's response will be marked by <Patient>. The hints I give you are marked by <Hint>.

<Patient> {patient_response} <Patient>

<Hint>Please, based on the patient's response, decide now whether a diagnosis can be made.

You need to consider whether a diagnosis can be made. First, you need to generate a chain-of-thought analysis.
If you believe a diagnosis can be made, select the most likely disease from <candidate_diseases> (choose only one). Example output:
{"CoT": "Your thought process", "judge": true, "disease": "common cold"}

If you believe the information on symptoms is insufficient, ask the patient for more symptom information, noting that you can only ask about one symptom. Example output:
{"CoT": "Your thought process", "judge": false, "symptom": "Do you have a lack of appetite?"}

Please output in JSON format.<Hint>

Figure 21: The *Chain-of-Though* prompt for LLM Diagnosis. Known_symptoms represents the symptoms currently known by the LLM. candidate_diseases represents the list of candidate diseases $D$. candidate_diseases represents the response of the patient.

**The *Multi-Choice* prompt for LLM Diagnosis**

```
Initial input:
You are a professional physician tasked with diagnosing a patient based on their symptom
information. I will provide you with information on possible diseases, and you will need to
carefully consider which of the candidate diseases the patient might have.

Patient symptom information is marked by <symptoms>.

<symptoms> {Known_symptoms} <symptoms>

Now you need to determine whether you can diagnose a certain disease or if you need to
inquire about the patient's symptoms. Please choose one of the following options:
A. {candidate_diseases 1}
B. {candidate_diseases 2}
C. {candidate_diseases 3}
D. Unsure of diagnosis; I need to inquire about the patient's symptoms.

If you choose D (unsure), you need to generate a question. Example output: "D. Question: ...."
Otherwise, just output the option.

Input after patient response:
The patient's response will be marked by <Patient>. The hints I give you are marked by <Hint>.

<Patient> {patient_response} <Patient>

<Hint>Please, based on the patient's response, decide now whether a diagnosis can be
made.

Now you need to determine whether you can diagnose a certain disease or if you need to
inquire about the patient's symptoms. Please choose one of the following options:
A. {candidate_diseases 1}
B. {candidate_diseases 2}
C. {candidate_diseases 3}
D. Unsure of diagnosis; I need to inquire about the patient's symptoms.

If you choose D (unsure), you need to generate a question. Example output: "D. Question: ...."
Otherwise, just output the option.
```

Figure 22: The *Multi-Choice* prompt for LLM Diagnosis. Known_symptoms represents the symptoms currently known by the LLM. candidate_diseases represents the list of candidate diseases $D$. candidate_diseases represents the response of the patient.

**Evaluating COT and COD Completeness Prompt**

```
Now I need your help to score the analysis of patient diagnoses by large models. The scoring
criteria are as follows:
1. Completeness of symptom information mentioned. Whether the model mentioned all the symptoms.
If all are mentioned, score 1, if missing, score 0.
2. Completeness of candidate disease analysis. Whether the model analyzed all candidate
diseases. If all are analyzed, score 1, if missing, score 0.

Patient symptom information is: {symptoms}

Candidate diseases are: {candidate_diseases}

Please score directly in JSON format. An example output is:
{"Symptom Completeness": 1, "Disease Completeness": 0}
```

Figure 23: Prompt for evaluating COT and COD Completeness. {symptoms} indicates the symptoms mentioned at the patient's first inquiry. {candidate_disease} indicates the candidate disease set.