HuatuoGPT-o1, Towards Medical Complex Reasoning with LLMs

Junying Chen[1], **Zhenyang Cai**[1], **Ke Ji**[1], **Xidong Wang**[1], **Wanlong Liu**[1]
Rongsheng Wang[1], **Jianye Hou**[1], **Benyou Wang**[1,2*]
[1] The Chinese University of Hong Kong, Shenzhen
[2] Shenzhen Research Institute of Big Data
https://github.com/FreedomIntelligence/HuatuoGPT-o1

Abstract

The breakthrough of OpenAI o1 highlights the potential of enhancing reasoning to improve LLM. Yet, most research in reasoning has focused on mathematical tasks, leaving domains like medicine underexplored. The medical domain, though distinct from mathematics, also demands robust reasoning to provide reliable answers, given the high standards of healthcare. However, verifying medical reasoning is challenging, unlike those in mathematics. To address this, we propose **verifiable medical problems** with a medical verifier to check the correctness of model outputs. This verifiable nature enables advancements in medical reasoning through **a two-stage approach**: (1) using the verifier to guide the search for a complex reasoning trajectory for fine-tuning LLMs, (2) applying reinforcement learning (RL) with verifier-based rewards to enhance complex reasoning further. Finally, we introduce **HuatuoGPT-o1**, a medical LLM capable of complex reasoning, which outperforms general and medical-specific baselines using only 40K verifiable problems. Experiments show complex reasoning improves medical problem-solving and benefits more from RL. We hope our approach inspires advancements in reasoning across medical and other specialized domains.

1 Introduction

The release of OpenAI o1 has marked a significant milestone in large language model (LLM) development, showcasing impressive capabilities [1–3]. This breakthrough highlights the potential of **scaling Chain-of-Thought (CoT)** and **reinforcement learning** to enhance LLM performance [4–6]. While subsequent research efforts attempt to replicate these advancements, they often remain limited to mathematical reasoning tasks [7–9, 6]. The application of o1-like methods to specialized fields, such as medicine, remains largely underexplored.

Medical tasks often involve complex reasoning [10–12]. In real-world medical diagnoses or decisions, doctors often deliberate carefully. Such life-critical field necessitates meticulous thinking to ensure more reliable answers [13, 14]. Additionally, the medical domain offers unique advantages: compared to general domains, the medical domain is generally narrower in scope and easier to verify. Furthermore, medical reasoning closely resembles real-world applications in fields like finance, law, education, and security, making advancements in this area readily transferable [15, 16].

Despite these advantages, a key challenge in medical reasoning is verifying the thought process, which often lacks clear steps. Inspired by mathematical problems that allow verification through their outcomes, we construct 40K verifiable medical problems reformatted from challenging, closed-set medical exam questions. These verifiable problems are characterized as open-ended with unique,

---

*Benyou is the corresponding author with email: *wangbenyou@cuhk.edu.cn*

objective ground-truth answers that allow an LLM verifier to check solution correctness. This enables a two-stage approach for advancing medical complex reasoning:

Stage 1: Learning Complex Reasoning We construct complex reasoning trajectories through strategy-based searches guided by verifier feedback (True or False). The LLM first initializes a CoT. If the verifier rejects the current CoT, the model extends the CoT by applying a strategy sampled from *Backtracking*, *Exploring New Paths*, *Verification*, and *Correction* until a correct answer is provided. Successful reasoning trajectories are then used to fine-tune the LLM, enabling it develop complex reasoning skills that embody iterative reflection.

Stage 2: Enhancing Complex Reasoning with RL After acquiring complex reasoning skills, reinforcement learning (RL) further refine this ability. Specifically, sparse rewards provided by the verifier guide self-improvement using the Proximal Policy Optimization (PPO) algorithm.

Using this approach, we present **HuatuoGPT-o1**, a medical LLM capable of producing a long CoT to recognize its mistakes, try different strategies and refine the answer. Experiments demonstrate that our method (using only 40K data points) yields an 8.5-point improvement on medical benchmarks with an 8B model. Furthermore, our 70B model outperforms other open-source general and medical-specific LLMs across multiple medical benchmarks. The experiments further reveal that complex reasoning enhances medical problem-solving and boosts RL performance compared to standard or non-CoT methods. Our contributions are as follows:

- To the best of our knowledge, this is the first work to advance medical complex reasoning in LLMs using verifiable medical problems and a medical verifier.
- With verifiable medical problems, we propose a two-stage training approach, combining search strategies to construct reasoning pathways for fine-tuning, and further enhanced by RL with verifier feedback.
- Using the proposed method, we developed HuatuoGPT-o1, the first medical LLM capable of complex reasoning. HuatuoGPT-o1 exhibits superior performance compared to the open-source general and medical-specific baselines.
- Our experiments reveal that complex reasoning is effective for medical problem-solving and benefits RL enhancements.
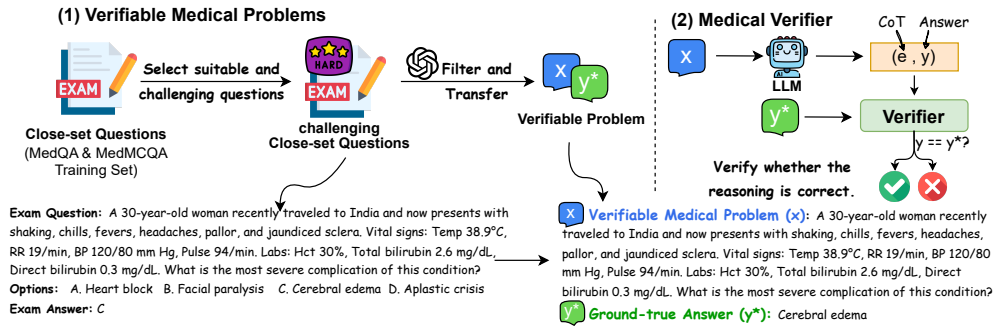
2 Verifiable Medical Problems

Figure 1: **Left:** Constructing verifiable medical problems using challenging close-set exam questions. **Right:** The verifier checks the model's answer against the *ground-truth answer*.

Inspired by mathematical problems that enable verification of the solution process through the final result, we aim to create verifiable medical problems that allow reasoning verification through outcomes. These verifiable problems are characterized as open-formal with unique, objective ground-truth answers, as illustrated in Figure 1.

Sourcing from Medical Exam Questions To achieve this, we utilize closed-set real-world exam questions for two key reasons: 1) a large number of medical exam questions are available; and 2) these exam questions are typically objective and accurate. Specifically, we collected 192K medical multiple-choice exam questions from the training sets of *MedQA-USMLE* [17] and *MedMcQA* [18].

Transforming to Verifiable Medical Problems   However, these medical questions are closed-set, meaning they provide limited options to choose from. This makes it easy for models to guess the correct answer without proper reasoning. Additionally, some questions are not suitable due to they may lack a unique correct answer for verification or are too simple to require reasoning.

To address this, we select and process the questions as follows:

1. **Selecting Challenging Questions** We removed questions that three small LLMs (Gemma2-9B [19], LLaMA-3.1-8B [20], Qwen2.5-7B [21]) all answered correctly and discarded short questions to retain those requiring deeper reasoning.
2. **Ensure Unique Answers:** We excluded questions asking for "incorrect options" or with multiple correct answers. A LLM (GPT-4o) is further employed to remove questions where the correct answer might not be unique or could be ambiguous.
3. **Reformatting to Open-Ended Formal:** Using LLMs (GPT-4o), We reformatted each closed-set question into open-ended problem an open-ended problem x and a ground-truth answer y^*, as shown in Figure 1.

The prompt used for filtering and processing can be found in Appendix B. After this filtering and processing, we ultimately constructed a dataset of 40K verifiable medical questions denoted as $\mathcal{D} = \{(x, y^*)\}$, where x is a verifiable problem and y^* the ground-truth answer.

Developing Medical Verifier   With these verifiable problems, we propose a verifier to assess the correctness of model outputs. Given a medical verifiable problem x, the model generates a Chain-of-Thought (CoT) e and a result y. The verifier checks y against the ground-truth answer y^* and provides binary feedback as:

$$\text{Verifier}(y, y^*) \in \{\text{True}, \text{False}\}$$

This feedback is essential for building a correct reasoning trajectory and improving reasoning performance. We use GPT-4o [22] as the verifier, prompting it to perform validation with the detailed prompt provided in Appendix C. Given the prevalence of aliases in the medical domain, exact match methods [8, 23] commonly applied in mathematics are impractical. Experiments in Section 4.2 confirm this and demonstrate the reliability of the LLM-based verifier.
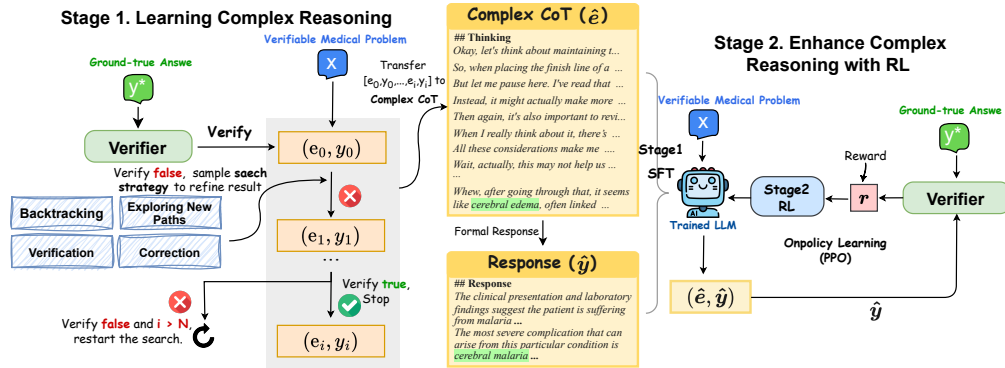
3   Methodology

Figure 2: Demonstration of developing and improving LLMs for medical complex reasoning. **Left (Stage1):** Searching for correct reasoning trajectories to fine-tune LLMs for complex reasoning. **Right (Stage2):** Using the verifier to enhance complex reasoning via reinforcement learning.

In this section, we present the method for training LLMs to performing medical complex reasoning to identify errors, and refine answers using deep thinking. As shown in Figure 1, the method has two stages: **Stage One:** master complex reasoning, and **Stage Two**, enhance complex reasoning with reinforcement learning (RL).
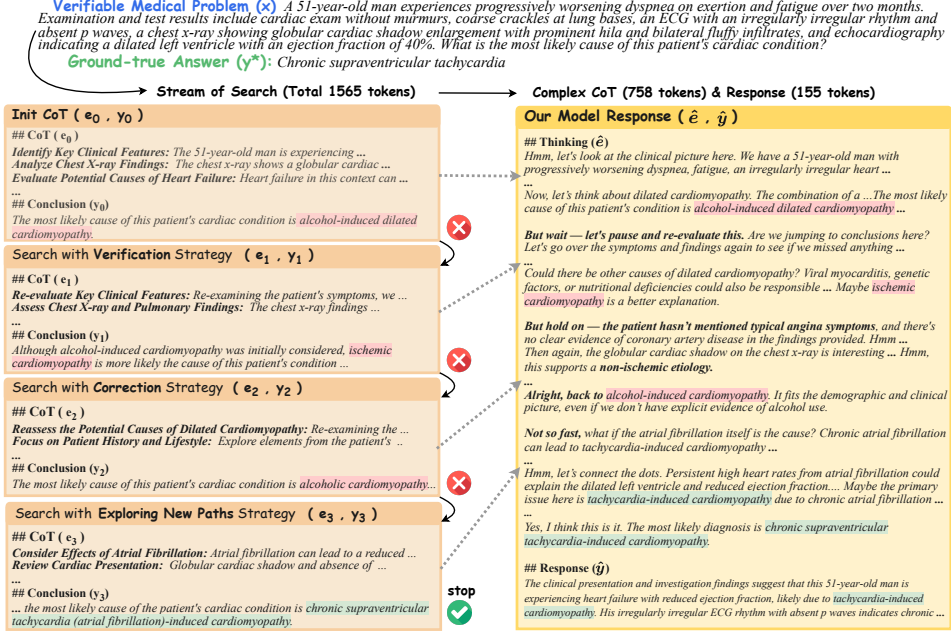
Figure 3: Example of data synthesis. **Left:** strategy search on medical verifiable problems until the answer is validated. **Right:** Merging the entire search process into efficient complex CoTs, facilitating effective deep reasoning to refine answers. The complex CoTs and responses are used to train the model to adopt *thinks-before-it-answers* behavior akin to o1.

3.1 Stage One: Learning Complex Reasoning

Searching for Correct Trajectories Given a verifiable medical problem as a tuple (x, y^*), i.e. (question, ground-true answer), the LLM (e.g., GPT-4o) generates an initial CoT e_0 and answer y_0:

$$e_0, y_0 = \mathsf{LLM}_{\text{init}}(x)$$

The verifier checks if y_0 matches y^*. If incorrect, the model iteratively refines the answer by applying a randomly selected search strategy $k \in \mathcal{K}$ on prior thoughts $[e_0, y_0, \ldots, e_{i-1}, y_{i-1}]$, producing new reasoning e_i and new answer y_i:

$$e_i, y_i = \mathsf{LLM}_{k_i}(x, [e_0, y_0, \ldots, e_{i-1}, y_{i-1}])$$

where i denotes the i-th iteration. We define four search strategies $\mathcal{K}$ to guide the refinement process:

- **Exploring New Paths** The LLM explores a new approach e_i, distinct from prior $e_0, \ldots, e_{i-1}$, to derive a new answer y_i.
- **Backtracking** The LLM revisits a previous reasoning process e_j, y_j, where $j < i - 1$, and continues reasoning from there. Note that Backtracking is sampled only if $i \leq 2$.
- **Verification** The LLM evaluates the current reasoning e_{i-1} and result y_{i-1}, providing a validation process e_i and the verified result y_i.
- **Corrections** The LLM critiques and corrects the current reasoning e_{i-1}, yielding a revised reasoning e_j and answer y_i.

The process iterates until y_i is verified as correct. If the maximum iteration count $N = 3$ are reached, the search restarts. Each data point (x, y^*) is given up to $T = 3$ attempts; if all fail, the data point is discarded. The prompts for search reasoning trajectories can be found in Appendix D.

Constructing SFT Training Data When a successful trajectory $[e_0, y_0, \ldots, e_i, y_i]$ is found, it is reformatted into a coherent, natural language reasoning process $\hat{e}$ (*Complex CoT*):

4

Algorithm 1: Training LLMs for Medical Complex Reasoning

Require: Medical Verifiable Problems $\boldsymbol{D} = \{(\boldsymbol{x}, \boldsymbol{y}^*)\}$, a Verifier, an LLM (GPT-4o) for synthesizing
 reasoning trajectories, search strategies $\mathcal{K}$, max search depth $\boldsymbol{N}$, max search attempts $\boldsymbol{T}$, and initial policy
 $\boldsymbol{\pi_\theta}$.

$\boldsymbol{\mathcal{D}}_{\text{Search}}, \boldsymbol{\mathcal{D}}_{\text{RL}} \leftarrow \mathsf{Split}(\boldsymbol{\mathcal{D}})$
$\boldsymbol{\mathcal{D}}_{\text{SFT}} \leftarrow \emptyset$
// Stage One: Learning Complex Reasoning
for $(\boldsymbol{x}, \boldsymbol{y}^*) \in \boldsymbol{\mathcal{D}}_{Search}$ **do**
 for $j \leftarrow 1$ **to** $\boldsymbol{T}$ **do**
  $\boldsymbol{e}_0, \boldsymbol{y}_0 \leftarrow \mathsf{LLM}_{\text{init}}(\boldsymbol{x})$
  for $i \leftarrow 1$ **to** $\boldsymbol{N}$ **do**
   $\boldsymbol{k}_i \sim \mathcal{K}$
   $\boldsymbol{e}_i, \boldsymbol{y}_i \leftarrow \mathsf{LLM}_{\boldsymbol{k}_i}(\boldsymbol{x}, [\boldsymbol{e}_0, \boldsymbol{y}_0, ..., \boldsymbol{e}_{i-1}, \boldsymbol{y}_{i-1}])$
   if $\mathsf{Verifier}(\boldsymbol{y}_i, \boldsymbol{y}^*)$ **then**
    $\hat{\boldsymbol{e}} \leftarrow \mathsf{LLM}_{\text{Reformat}}([\boldsymbol{e}_0, \boldsymbol{y}_0, ..., \boldsymbol{e}_i, \boldsymbol{y}_i])$
    $\hat{\boldsymbol{y}} \leftarrow \mathsf{LLM}_{\text{Response}}(\hat{\boldsymbol{e}})$
    $\boldsymbol{\mathcal{D}}_{\text{SFT}} \leftarrow \boldsymbol{\mathcal{D}}_{\text{SFT}} \cup \{(\boldsymbol{x}, \hat{\boldsymbol{e}}, \hat{\boldsymbol{y}})\}$
    break
  if $\mathsf{Verifier}(\boldsymbol{y}_i, \boldsymbol{y}^*)$ **then**
   break

// SFT
for $(\boldsymbol{x}, \hat{\boldsymbol{e}}, \hat{\boldsymbol{y}}) \in \boldsymbol{\mathcal{D}}_{SFT}$ **do**
 $\mathcal{L}_{\text{SFT}}(\boldsymbol{\theta}) \leftarrow -\log \boldsymbol{\pi_\theta}(\hat{\boldsymbol{e}}, \hat{\boldsymbol{y}} \mid \boldsymbol{x})$
 $\boldsymbol{\theta} \leftarrow \mathsf{UpdateParameters}(\mathcal{L}_{SFT}(\boldsymbol{\theta}), \boldsymbol{\theta})$

// Stage Two: Enhance Reasoning with RL
$\boldsymbol{\pi}_{\text{ref}} \leftarrow \boldsymbol{\pi_\theta}$
for $(\boldsymbol{x}, \boldsymbol{y}^*) \in \boldsymbol{\mathcal{D}}_{RL}$ **do**
 $\hat{\boldsymbol{e}}, \hat{\boldsymbol{y}} \sim \boldsymbol{\pi_\theta}(\boldsymbol{x})$
 // Reward
 $\boldsymbol{r} \leftarrow \mathsf{Rule}\left(\mathsf{Verifier}\left(\hat{\boldsymbol{y}}, \boldsymbol{y}^*\right)\right) - \beta \mathsf{KL}\left(\boldsymbol{\pi_\theta}\left(\cdot \mid \boldsymbol{x}\right) \| \boldsymbol{\pi}_{\text{ref}}\left(\cdot \mid \boldsymbol{x}\right)\right)$
 $\boldsymbol{\theta} \leftarrow \mathsf{UpdateParameters}\left(\mathcal{L}_{\text{RL}}\left(\boldsymbol{x}, \hat{\boldsymbol{e}}, \hat{\boldsymbol{y}}, \boldsymbol{r}, \boldsymbol{\pi}_{\text{ref}}, \boldsymbol{\pi_\theta}\right), \boldsymbol{\theta}\right)$
return $\boldsymbol{\pi_\theta}$

$$\hat{e} = \mathsf{LLM}_{\text{Reformat}}([e_0, e_1, \ldots, e_i, y_i])$$

As shown in Figure 3, this reformatting avoids rigid structures, using smooth transitions (e.g., "hmm," "also," "wait") to streamline reasoning and reduce token usage. The model then generates a formal response $\hat{y}$ for question x using the conclusion of $\hat{e}$:

$$\hat{y} = \mathsf{LLM}_{\text{Response}}(x, \hat{e})$$

The prompt used for constructing SFT data can be found in Appendix E.

Supervised Fine-Tuning (SFT)   We synthesize 20K SFT data points $D_{\text{SFT}} = \{(x, \hat{e}, \hat{y})\}$ from the verifiable problem set $\mathcal{D} = \{(x, y^*)\}$ using GPT-4o. D_{SFT} is used to fine-tune LLMs to generate a complex CoT $\hat{e}$ followed by a formal response $\hat{y}$. This fine-tuning process teaches the model to *think before answering*, encouraging a *Stream-of-Search (SoS)* [23] way where the model deeply explores and refines its reasoning before answering.

3.2 Stage Two: Enhance Complex Reasoning with RL

In this stage, we further enhance the complex reasoning skills using reinforcement learning (RL). While the LLM learned successful reasoning trajectories in stage 1, these paths, derived via search, may not be optimal. On-policy learning in stage 2 aims to refine the model for better complex CoT reasoning.

Rewards of RL Rewards play a crucial role in guiding the RL training target. Given a verifiable problem x and the generated response $(\hat{e}, \hat{y})$, the reward is assigned as:

$$r'(x, \hat{y}, y^*) = \begin{cases} 1 & \text{if verifier}(\hat{y}, y^*) = \text{True} \\ 0.1 & \text{if verifier}(\hat{y}, y^*) = \text{False} \\ 0 & \text{if } \hat{y} = \text{null} \end{cases}$$

Following [24, 25, 8], correct answers receive a reward of 1, incorrect answers receive 0.1, and responses that lack *think-before-answering* behavior receive 0. Additionally, following related works, the total reward combines this function score with the Kullback-Leibler (KL) divergence between the learned RL policy π_θ and the initial policy π_{ref}, scaled by a coefficient β:

$$r(x, \hat{y}, y^*) = r'(x, \hat{y}, y^*) + \beta \text{KL}(\theta)$$

to stabilize training with sparse rewards [8].

Reinforcement Learning For RL, We use the Proximal Policy Optimization (PPO) [26] algorithm with a clipped objective. The fine-tuned model serves as the policy model π_θ. Training is conducted on the remaining verifiable medical problems $\mathcal{D}_{\text{RL}} = \{(x, y^*)\}$. The policy samples responses $(\hat{e}, \hat{y})$ for input x, computes the reward, and updates parameters θ.

The full training process for both stages is summarized in Algorithm 1.

4 Experiments

4.1 Experimental Setup

Training Data Finally, We constructed a 40K medical verification dataset $\mathcal{D} = \{(x, y^*)\}$ from the training sets of *MedQA-USMLE* [17] and *MedMCQA* [27]. Of this, 20K is used for SFT in stage 1 and 20K for RL in stage 2. Additionally, 4K unconverted data (close-set questions with option answers) from $\mathcal{D}$ are included to enhance generalization. In line with prior work that integrates general-domain data to support medical adaptation [15, 28], we add 5K general verification questions sourced from *MMLU-Pro* [29] outside the medical-related tracks. All data were strictly screened to avoid contamination with the evaluation data using the filtering method of Med-PaLM2 [30] (filtering overlaps of 64 consecutive characters).

Model Training Using the proposed method, we train our models **HuatuoGPT-o1-8B** and **HuatuoGPT-o1-70B** based on *LLaMA-3.1-8B-Instruct* and *LLaMA-3.1-70B-Instruct* [20], respectively. In Stage 1, the models are fine-tuned on the $\mathcal{D}_{\text{SFT}}$ for 3 epochs with a learning rate of 5e-6 and a batch size of 128. In Stage 2, we employ PPO for RL with a learning rate of 5e-7, a batch size of 128, and β set to 0.03. The PPO parameters are set as: 3 PPO epochs, a discount factor 1.0, a value coefficient 1.0, and a clip range 0.2.

Baselines We compare our models with two type of LLMs: **1) General LLMs**: Qwen-2.5 [31], LLaMA-3.1 [20], Gemma 2 [19], Yi [32], Mistral [33]; and **2) Medical-Specific LLMs:** UltraMedical [28], OpenBioLLM [34], and BioMistral [35].

Benchmarks We evaluate on standard medical benchmarks: *MedQA* (USMLE test set) [17], *MedMCQA* (validation set) [18], and *PubMedQA* (test set) [36]. Aditionally, we evaluated the medical sections of some challenging LLM benchmarks, including the health and biology tracks of *MMLU-Pro* [29], and the genetics and molecular biology tracks of *GPQA* [37]. Due to the limited number of *GPQA* questions, we ran this evaluation 5 times and averaged the results.

4.2 Experimental Results

Main Results We evaluated various open-source LLMs on medical tasks, as shown in Table 1. The results indicate that prior medical-specific LLMs, like UltraMedical, excel on traditional medical benchmarks (MedQA, MedMCQA, PubMedQA) but struggle on the newer, more challenging datasets,

| | MedQA | MedMCQA | PubMedQA | MMLU-Pro | | GPQA | | Avg. |
				Health	Biology	Genetics	Molecular Biology	
~ 8B Large Language Models								
🩸 BioMistral-7B	45.0	40.2	66.9	27.4	49.2	28.6	38.5	42.3
🩸 OpenBioLLM-8B	57.7	54.1	74.1	38.4	52.4	43.7	39.6	51.4
🩸 UltraMedical-8B	<u>71.1</u>	<u>58.3</u>	<u>77.4</u>	<u>55.1</u>	66.7	41.2	48.4	59.7
Mistral-7B-Instruct	48.2	44.6	59.5	33.7	53.6	30.0	46.1	45.1
Yi-1.5-9B-Chat	50.8	48.7	69.8	43.4	65.6	42.5	48.1	52.7
LLaMA-3.1-8B-Instruct	58.7	56.0	75.2	52.7	64.6	33.8	46.8	55.4
GLM-4-9B-Chat	58.9	49.8	73.5	45.5	65.4	**53.8**	41.6	55.5
Qwen2.5-7B-Instruct	57.0	55.6	72.7	50.6	<u>70.2</u>	36.2	49.7	56.0
Gemma2-9B	61.8	55.9	63.3	<u>55.1</u>	**74.9**	35.0	<u>57.4</u>	57.6
🩸💡 **HuatuoGPT-o1-8B**	**72.6**	**60.4**	**79.2**	**58.7**	68.2	<u>48.8</u>	**59.7**	**63.9**
w/o Stage2 (RL)	69.0	57.9	77.7	53.5	66.1	41.2	53.5	<u>59.8</u>
> 10B Large Language Models								
🩸 UltraMedical-70B	<u>82.2</u>	71.8	78.4	64.8	71.1	33.8	62.9	66.4
🩸 OpenBioLLM-70B	76.1	**74.7**	<u>79.2</u>	68.8	76.7	38.8	54.8	67.0
DeepSeek-67B-Chat	57.1	51.7	76.1	46.9	66.2	40.0	51.0	55.6
Yi-1.5-34B-Chat	59.5	56.7	74.3	52.8	71.0	32.5	56.8	57.7
Gemma2-27B	65.4	60.2	72.6	61.1	76.2	32.5	61.6	61.4
Qwen2.5-72B-Instruct	72.7	66.2	71.7	65.3	78.8	41.2	56.8	64.7
💡 QwQ-32B-Preview	72.3	65.6	73.7	62.0	78.1	37.5	<u>64.5</u>	64.8
Llama-3.1-70B-Instruct	78.4	72.5	78.5	68.2	<u>80.8</u>	52.5	61.6	70.3
🩸💡 **HuatuoGPT-o1-70B**	**83.3**	<u>73.6</u>	**80.6**	**71.0**	**82.8**	56.2	**66.5**	**73.4**
w/o Stage2 (RL)	80.3	70.1	78.6	<u>70.2</u>	79.8	<u>54.2</u>	63.9	<u>71.0</u>

Table 1: Main Results on Medical Benchmarks. LLMs with 🩸 are specifically trained for the medical domain, and 💡 indicates LLMs training for long chain-of-thought reasoning. "w/o" means "without". Within each segment, **bold** highlights the best scores, and <u>underlines</u> indicate the second-best.

even when the questions are medically related. This may suggest that MMLU-Pro and GPQA require not only medical knowledge but also stronger reasoning capabilities.

Our model, HuatuoGPT-o1, performs exceptionally across all datasets. The 8B version outperforms the base model (LLaMA-3.1-8B-Instruct) by 8 points in overall evaluation. Furthermore, our 70B model surpasses other comparable open-source LLMS, including QwQ-32B, which are also developed specifically for advanced reasoning capabilities. These results demonstrate the effectiveness of our approach. Additionally, compared to only fine-tuning (*w/o RL*), the two-stage training strategy significantly improves performance, benefiting from the verifiable medical problems.

Ablation Study We conducted an ablation study on the 8B model to analyze the impact of Complex-CoT and RL The results, shown in Table 2, reveal the following insights:

1. Simple Multiple-Choice Training Is Ineffective: We compared the performance of models trained solely on the original medical multiple-choice questions of dataset D. Specifically, we used multiple-choice questions as inputs and the correct option as output for fine-tuning. The results indicate that raining solely on multiple-choice questions (the fine-tuned baseline) yields minimal improvement over the base model (LLaMA-3.1-8B-Instruct). This suggests that learning correct answers alone does not improve problem-solving ability.

2. Effectiveness of Complex CoTs: We further examined the impact of different types of Chain-of-Thought (CoT) reasoning. The results show that direct learning of response ($\hat{y}$) performs the worst, while simple CoT (y_0, e_0) offers only little benefit. In contrast, Complex CoT ($\hat{y}_0, \hat{e}$) significantly improves performance by an average of 4.3 points. This demonstrates the importance of teaching models to refine their answers with reflection.

3. Complex CoT Boosts RL: We compared the RL enhancements under different CoT strategies, as shown in Table 3. The results indicate that Complex CoT, which involves much longer CoT (an average of 712 tokens), yields a significantly greater gain (3.6 points) compared to simple CoT (2.6 points) and no CoT (1.1 points), as detailed in Table 3. This may suggests that longer self-play reasoning paths provide richer thought processes and feedback, enabling the model to discover higher-reward solutions.

	MedQA	MedMCQA	PubMedQA	MMLU-Pro (Med✚)	GPQA (Med✚)
Baseline LLMs					
LLaMA-3.1-8B-Instruct	58.7	56.0	75.2	58.2	44.1
Fine-Tuned Baseline					
SFT w/ Original Exam Data of $\mathcal{D}$	60.0	55.5	74.1	54.3	46.9
Effectiveness of Complex Chain-of-Thought (CoT)					
SFT w/o ~~CoT~~ (only $\hat{y}$)	65.2	58.1	75.4	58.5	48.7
SFT w/ Simple CoT (x_0, y_0)	66.6	**59.2**	75.4	57.0	46.7
SFT w/ Complex CoT $(\hat{x}, \hat{y})$	**69.0**	57.9	**77.7**	**59.4**	**51.0**
Effectiveness of RL					
SFT w/o ~~CoT~~ + **RL** w/ PPO	66.4	58.6	76.3	60.1	49.8
SFT w/ Simple CoT + **RL** w/ PPO	68.7	58.4	77.5	60.2	53.1
SFT w/ Complex CoT + **RL** w/ PPO	**72.6**	**60.4**	**79.2**	**63.1**	**57.5**
Comparison of Different RL Algorithms					
SFT w/ Complex CoT + **RL** w/ DPO	72.2	58.4	77.3	60.4	52.5
SFT w/ Complex CoT + **RL** w/ RLOO	71.1	60.1	78.1	60.9	58.2
SFT w/ Complex CoT + **RL** w/ PPO	**72.6**	**60.4**	**79.2**	**63.1**	**57.5**

Table 2: The results of ablation experiments on *HuatuoHPT-o1-8B*. (Med✚) indicates that only the medical-related parts are evaluated. "w/o" and "w/" denote "without" and "with". "Original Exam Data" refers to original multiple-choice questions used for medical verifiable problems D. **Bold** highlights the best scores in each segment.

4. PPO Yields the Best Performance: Using the same reward function, we further compared different RL-related algorithms, including the preference learning algorithm DPO [38] and the REINFORCE-style algorithm RLOO [39]. Detailed implementation information is provided in Appendix F. Comparing PPO, RLOO, and DPO, we find PPO performs best, followed by RLOO and DPO. The weaker performance of DPO likely results from its off-policy nature, while PPO benefits from its use of value models, despite higher memory consumption.

	# Avg. Generated Tokens	Δ Avg. Gain from RL
Direct Response (only $\hat{y}$)	82	1.1
Simple CoT (x_0, y_0)	281	2.6
Complex CoT $(\hat{x}, \hat{y})$	**712**	**3.6**

Table 3: Comparison of models trained with different reasoning strategies. "# Avg. Tokens" indicates the average number of tokens generated per question. Δ represents the performance improvement from RL, as detailed in Table 1.

Reliability of the Verifier The verifier plays a crucial role in guiding path search and reinforcement learning (RL). In our approach, GPT-4o serves as the verifier to assess model outcomes against ground-truth answers. To assess its reliability, we manually verified 200 scoring instances sampled from Stage 1 and Stage 2. As shown in Figure 4, GPT-4o achieved 96.5% accuracy in Stage 1 and 94.5% in Stage 2, demonstrating its reliability. In contrast, the Exact Match method [8], which uses regular expressions to determine whether the correct answer is present in the response, performed significantly worse, with accuracies of only 70.5% in Stage 1 and 74.5% in Stage 2. This highlights the critical role of LLM-based verifiers. Additionally, we fine-tuned an 8B verifier based on LLaMA-3.1-8B with 20,000 scoring samples. The fine-tuned verifier also demonstrated feasibility, achieving over 90% accuracy.

Domain Compatibility To verify domain compatibility, we extra applied our method to the Chinese medical domain. We constructed a dataset of 40,000 verifiable Chinese questions from the CMB-exam training set. We then trained HuatuoGPT-o1-7B-zh using our two-stage approach based on Qwen2.5-7B-Instruct. As shown in Table 4, HuatuoGPT-o1-7B-zh outperformed other Chinese LLMs
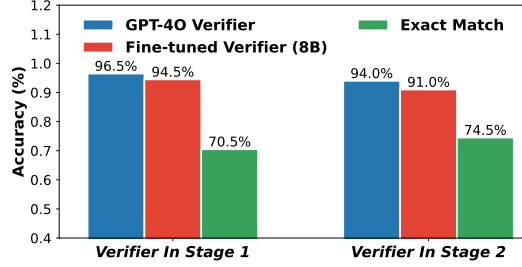
Figure 4: Accuracy of verifiers. Accuracy is based on 200 manually annotated samples.

	MedQA (Chinese)	CMB (Exam)	CMExam	CMMLU (Med➕)
🩺 HuatuoGPT2-7B	73.7	63.6	67.4	58.4
Yi-1.5-9B-Chat	75.8	66.2	68.1	64.2
Qwen2.5-7B	71.4	70.7	70.4	70.5
GLM-4-9B-chat	75.2	70.0	70.5	67.6
🩺💡 **HuatuoGPT-o1-7B-zh**	**79.8**	**73.0**	**74.1**	**74.5**
w/o Stage2 (RL)	76.5	70.8	72.3	70.9

Table 4: Results on Chinese medical benchmarks. (**Med➕**) indicates that only the medical portion is evaluated. MedQA (Chinese) refers to the Chinese test set of MedQA (MedQA-MCMLE).

of similar size, demonstrating the method's adaptability to new domains. For more experimental details, refer to Appendix G.

5 Related Work

Research on o1 Recent studies have extensively analyzed the roadmap and core techniques of OpenAI's o1 [4, 6, 5], offering foundational insights into its architecture and methodology. Extensions such as LLaMA-Berry [9], LLaVA-o1 [40], o1-Coder [41], and Marco-o1 [42] have explored o1-like reasoning in various domains, including mathematics, vision-language integration, and open-ended problem-solving. However, these efforts have yet to address applications in medical or other highly specialized fields. In contrast, research focused on medicine [2, 43, 14] highlights o1's potential for deliberate, chain-of-thought reasoning in healthcare contexts. Meanwhile, several o1-inspired models, such as DeepSeek-R1-Lite-Preview [44], QwQ [7], and Gemini-2.0 Flash Thinking [45], have emerged. Despite their promise, most of these models remain closed-source, leaving substantial opportunities for further exploration and application of o1's capabilities across diverse fields.

Medical LLMs The success of generalist LLMs has spurred interest in developing medical-specific LLMs to excel in the medical domain. Notably, the MedPaLM series [46, 30] achieved over 60% accuracy on the MedQA benchmark, reportedly surpassing human experts. Previous medical LLMs typically follow two main approaches [28]: **(1) Prompting Generalist LLMs** [47, 10, 48, 22, 12]: This method employs task-specific prompts to adapt generalist models for medical applications. While efficient and training-free, it is inherently limited by the capabilities of the original LLMs. **(2) Further Training with Medical Data** [49–52, 34, 35, 53–58]: This involves training LLMs on medical pretraining corpora or medical instructions to embed medical knowledge and expertise. However, this always requires significant computational resources, such as the 1.4 billion and 3 billion training tokens used for Meditron [59] and HuatuoGPT-II [15]. In contrast, our approach emphasizes enabling LLMs to excel in medical reasoning, offering a distinct solution.

Enhancing Reasoning in LLMs Chain-of-Thought (CoT) prompting enhances the reasoning capabilities of LLMs [60, 61], but scaling expert-labeled reasoning paths remains costly, especially for complex problems [62, 63]. To mitigate this, model-generated reasoning paths filtered through external supervision offer a partial solution [64, 65], yet scalability challenges persist [66, 67].

Reinforcement learning-based methods leveraging reward models or oracle functions show potential but often suffer from slow processing, high costs, and supervision bottlenecks [68, 69].

Complex Reasoning   Developing models with reflective abilities like critique and self-correction has shown success in reasoning, planning, and coding tasks [23, 70–74], though underexplored in specialized domains like medicine. While prompting techniques can generate self-critical reasoning [75, 70], they struggle without reliable reward functions or verifiers, particularly in complex domains [76, 77]. Fine-tuning and reinforcement learning methods offer solutions but require extensive human annotations or intricate reward designs [78–81]. Additionally, self-training methods present a promising direction for developing self-correction capabilities [72, 82, 83].

6   Conclusion

This study advances the medical reasoning capabilities of LLMs. Firstly, we construct the medical verifiable problems and a medical verifier. This enabled a two-stage training process: (1) learning complex reasoning and (2) enhancing it through RL. We developed HuatuoGPT-o1, a medical LLM with *thinks-before-it-answers* behavior, achieving outstanding performance in medical benchmarks. Experiments show that complex reasoning improves medical problem-solving and benefits obviously from RL. Additional validation in Chinese medical contexts shows the method's adaptability to other fields. We believe our approach can enhance domain-specific reasoning beyond mathematics.

Acknowledgment

References

[1] Melody Y. Guan, Manas Joglekar, Eric Wallace, Saachi Jain, Boaz Barak, Alec Heylar, Rachel Dias, Andrea Vallone, Hongyu Ren, Jason Wei, Hyung Won Chung, Sam Toyer, Johannes Heidecke, Alex Beutel, and Amelia Glaese. Deliberative alignment: Reasoning enables safer language models. *OpenAI Blog*, 2024. 1

[2] Yunfei Xie, Juncheng Wu, Haoqin Tu, Siwei Yang, Bingchen Zhao, Yongshuo Zong, Qiao Jin, Cihang Xie, and Yuyin Zhou. A preliminary study of o1 in medicine: Are we closer to an ai doctor? *arXiv preprint arXiv:2409.15277*, 2024. 9

[3] Tianyang Zhong, Zhengliang Liu, Yi Pan, Yutong Zhang, Yifan Zhou, Shizhe Liang, Zihao Wu, Yanjun Lyu, Peng Shu, Xiaowei Yu, et al. Evaluation of openai o1: Opportunities and challenges of agi. *arXiv preprint arXiv:2409.18486*, 2024. 1

[4] Yiwei Qin, Xuefeng Li, Haoyang Zou, Yixiu Liu, Shijie Xia, Zhen Huang, Yixin Ye, Weizhe Yuan, Hector Liu, Yuanzhi Li, et al. O1 replication journey: A strategic progress report–part 1. *arXiv preprint arXiv:2410.18982*, 2024. 1, 9

[5] Zhiyuan Zeng, Qinyuan Cheng, Zhangyue Yin, Bo Wang, Shimin Li, Yunhua Zhou, Qipeng Guo, Xuanjing Huang, and Xipeng Qiu. Scaling of search and learning: A roadmap to reproduce o1 from reinforcement learning perspective. *arXiv preprint arXiv:2412.14135*, 2024. 9

[6] Jun Wang, Meng Fang, Ziyu Wan, Muning Wen, Jiachen Zhu, Anjie Liu, Ziqin Gong, Yan Song, Lei Chen, Lionel M Ni, et al. Openr: An open source framework for advanced reasoning with large language models. *arXiv preprint arXiv:2410.09671*, 2024. 1, 9

[7] Qwen Team. Qwq: Reflect deeply on the boundaries of the unknown, November 2024. 1, 9

[8] Trung Quoc Luong, Xinbo Zhang, Zhanming Jie, Peng Sun, Xiaoran Jin, and Hang Li. Reft: Reasoning with reinforced fine-tuning. *arXiv preprint arXiv:2401.08967*, 2024. 3, 6, 8

[9] Di Zhang, Jianbo Wu, Jingdi Lei, Tong Che, Jiatong Li, Tong Xie, Xiaoshui Huang, Shufei Zhang, Marco Pavone, Yuqiang Li, et al. Llama-berry: Pairwise optimization for o1-like olympiad-level mathematical reasoning. *arXiv preprint arXiv:2410.02884*, 2024. 1, 9

[10] Khaled Saab, Tao Tu, Wei-Hung Weng, Ryutaro Tanno, David Stutz, Ellery Wulczyn, Fan Zhang, Tim Strother, Chunjong Park, Elahe Vedadi, et al. Capabilities of gemini models in medicine. *arXiv preprint arXiv:2404.18416*, 2024. 1, 9

[11] Vimla L Patel, José F Arocha, and Jiajie Zhang. Thinking and reasoning in medicine. *The Cambridge handbook of thinking and reasoning*, 14:727–750, 2005.

[12] Junying Chen, Chi Gui, Anningzhe Gao, Ke Ji, Xidong Wang, Xiang Wan, and Benyou Wang. Cod, towards an interpretable medical agent using chain of diagnosis. *arXiv preprint arXiv:2407.13301*, 2024. 1, 9

[13] Shaochen Xu, Yifan Zhou, Zhengliang Liu, Zihao Wu, Tianyang Zhong, Huaqin Zhao, Yiwei Li, Hanqi Jiang, Yi Pan, Junhao Chen, et al. Towards next-generation medical agent: How o1 is reshaping decision-making in medical scenarios. *arXiv preprint arXiv:2411.14461*, 2024. 1

[14] Mohamad-Hani Temsah, Amr Jamal, Khalid Alhasan, Abdulkarim A Temsah, and Khalid H Malki. Openai o1-preview vs. chatgpt in healthcare: A new frontier in medical ai reasoning. *Cureus*, 16(10):e70640, 2024. 1, 9

[15] Junying Chen, Xidong Wang, Ke Ji, Anningzhe Gao, Feng Jiang, Shunian Chen, Hongbo Zhang, Dingjie Song, Wenya Xie, Chuyi Kong, et al. Huatuogpt-ii, one-stage training for medical adaption of llms. *arXiv preprint arXiv:2311.09774*, 2023. 1, 6, 9, 23

[16] Daixuan Cheng, Shaohan Huang, and Furu Wei. Adapting large language models via reading comprehension. In *The Twelfth International Conference on Learning Representations*, 2023. 1

[17] Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *Applied Sciences*, 11(14):6421, 2021. 2, 6, 23

[18] Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. Medmcqa: A large-scale multi-subject multi-choice dataset for medical domain question answering. In *Conference on health, inference, and learning*, pages 248–260. PMLR, 2022. 2, 6

[19] Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024. 3, 6

[20] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. 3, 6

[21] Qwen Team. Qwen2.5: A party of foundation models, September 2024. 3

[22] OpenAI. Gpt-4 technical report, 2023. 3, 9

[23] Kanishk Gandhi, Denise Lee, Gabriel Grand, Muxin Liu, Winson Cheng, Archit Sharma, and Noah D Goodman. Stream of search (sos): Learning to search in language. *arXiv preprint arXiv:2404.03683*, 2024. 3, 5, 10

[24] Martin Riedmiller, Roland Hafner, Thomas Lampe, Michael Neunert, Jonas Degrave, Tom Wiele, Vlad Mnih, Nicolas Heess, and Jost Tobias Springenberg. Learning by playing solving sparse reward tasks from scratch. In *International conference on machine learning*, pages 4344–4353. PMLR, 2018. 6

[25] Alexander Trott, Stephan Zheng, Caiming Xiong, and Richard Socher. Keeping your distance: Solving sparse reward tasks using self-balancing shaped rewards. *Advances in Neural Information Processing Systems*, 32, 2019. 6

[26] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. 6

[27] Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. Medmcqa: A large-scale multi-subject multi-choice dataset for medical domain question answering. In *Conference on Health, Inference, and Learning*, pages 248–260. PMLR, 2022. 6

[28] Kaiyan Zhang, Sihang Zeng, Ermo Hua, Ning Ding, Zhang-Ren Chen, Zhiyuan Ma, Haoxin Li, Ganqu Cui, Biqing Qi, Xuekai Zhu, et al. Ultramedical: Building specialized generalists in biomedicine. *arXiv preprint arXiv:2406.03949*, 2024. 6, 9

[29] Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, et al. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *arXiv preprint arXiv:2406.01574*, 2024. 6

[30] Karan Singhal, Tao Tu, Juraj Gottweis, Rory Sayres, Ellery Wulczyn, Le Hou, Kevin Clark, Stephen Pfohl, Heather Cole-Lewis, Darlene Neal, et al. Towards expert-level medical question answering with large language models. *arXiv preprint arXiv:2305.09617*, 2023. 6, 9

[31] An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2024. 6

[32] Alex Young, Bei Chen, Chao Li, Chengen Huang, Ge Zhang, Guanwei Zhang, Heng Li, Jiangcheng Zhu, Jianqun Chen, Jing Chang, et al. Yi: Open foundation models by 01. ai. *arXiv preprint arXiv:2403.04652*, 2024. 6

[33] Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023. 6

[34] Malaikannan Sankarasubbu Ankit Pal and Malaikannan Sankarasubbu. Openbiollms: Advancing open-source large language models for healthcare and life sciences, 2024. 6, 9

[35] Yanis Labrak, Adrien Bazoge, Emmanuel Morin, Pierre-Antoine Gourraud, Mickael Rouvier, and Richard Dufour. Biomistral: A collection of open-source pretrained large language models for medical domains. *arXiv preprint arXiv:2402.10373*, 2024. 6, 9

[36] Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William W Cohen, and Xinghua Lu. Pubmedqa: A dataset for biomedical research question answering. *arXiv preprint arXiv:1909.06146*, 2019. 6

[37] David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. Gpqa: A graduate-level google-proof q&a benchmark. *arXiv preprint arXiv:2311.12022*, 2023. 6

[38] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36, 2024. 8

[39] Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. Back to basics: Revisiting reinforce style optimization for learning from human feedback in llms. *arXiv preprint arXiv:2402.14740*, 2024. 8

[40] Guowei Xu, Peng Jin, Li Hao, Yibing Song, Lichao Sun, and Li Yuan. Llava-o1: Let vision language models reason step-by-step. *arXiv preprint arXiv:2411.10440*, 2024. 9

[41] Yuxiang Zhang, Shangxi Wu, Yuqi Yang, Jiangming Shu, Jinlin Xiao, Chao Kong, and Jitao Sang. o1-coder: an o1 replication for coding. *arXiv preprint arXiv:2412.00154*, 2024. 9

[42] Yu Zhao, Huifeng Yin, Bo Zeng, Hao Wang, Tianqi Shi, Chenyang Lyu, Longyue Wang, Weihua Luo, and Kaifu Zhang. Marco-o1: Towards open reasoning models for open-ended solutions. *arXiv preprint arXiv:2411.14405*, 2024. 9

[43] Harsha Nori, Naoto Usuyama, Nicholas King, Scott Mayer McKinney, Xavier Fernandes, Sheng Zhang, and Eric Horvitz. From medprompt to o1: Exploration of run-time strategies for medical challenge problems and beyond. *arXiv preprint arXiv:2411.03590*, 2024. 9

[44] Xiao Bi, Deli Chen, Guanting Chen, Shanhuang Chen, Damai Dai, Chengqi Deng, Honghui Ding, Kai Dong, Qiushi Du, Zhe Fu, et al. Deepseek llm: Scaling open-source language models with longtermism. *arXiv preprint arXiv:2401.02954*, 2024. 9

[45] Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023. 9

[46] Karan Singhal, Shekoofeh Azizi, Tao Tu, S Sara Mahdavi, Jason Wei, Hyung Won Chung, Nathan Scales, Ajay Tanwani, Heather Cole-Lewis, Stephen Pfohl, et al. Large language models encode clinical knowledge. *Nature*, 620(7972):172–180, 2023. 9

[47] Harsha Nori, Yin Tat Lee, Sheng Zhang, Dean Carignan, Richard Edgar, Nicolo Fusi, Nicholas King, Jonathan Larson, Yuanzhi Li, Weishung Liu, et al. Can generalist foundation models outcompete special-purpose tuning? case study in medicine. *arXiv preprint arXiv:2311.16452*, 2023. 9

[48] Junkai Li, Siyu Wang, Meng Zhang, Weitao Li, Yunghwei Lai, Xinhui Kang, Weizhi Ma, and Yang Liu. Agent hospital: A simulacrum of hospital with evolvable medical agents. *arXiv preprint arXiv:2405.02957*, 2024. 9

[49] Ming Xu. Medicalgpt: Training medical gpt model. https://github.com/shibing624/MedicalGPT, 2023. 9

[50] Haochun Wang, Chi Liu, Nuwa Xi, Zewen Qiang, Sendong Zhao, Bing Qin, and Ting Liu. Huatuo: Tuning llama model with chinese medical knowledge, 2023.

[51] Tianyu Han, Lisa C Adams, Jens-Michalis Papaioannou, Paul Grundmann, Tom Oberhauser, Alexander Löser, Daniel Truhn, and Keno K Bressem. Medalpaca–an open-source collection of medical conversational ai models and training data. *arXiv preprint arXiv:2304.08247*, 2023.

[52] Chaoyi Wu, Weixiong Lin, Xiaoman Zhang, Ya Zhang, Weidi Xie, and Yanfeng Wang. Pmc-llama: toward building open-source language models for medicine. *Journal of the American Medical Informatics Association*, page ocae045, 2024. 9

[53] Zhijie Bao, Wei Chen, Shengze Xiao, Kuang Ren, Jiaao Wu, Cheng Zhong, Jiajie Peng, Xuanjing Huang, and Zhongyu Wei. Disc-medllm: Bridging general large language models and real-world medical consultation, 2023. 9

[54] Kai Zhang, Jun Yu, Eashan Adhikarla, Rong Zhou, Zhiling Yan, Yixin Liu, Zhengliang Liu, Lifang He, Brian Davison, Xiang Li, et al. Biomedgpt: a unified and generalist biomedical generative pre-trained transformer for vision, language, and multimodal tasks. *arXiv e-prints*, pages arXiv–2305, 2023.

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

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

[57] Guorui Zheng, Xidong Wang, Juhao Liang, Nuo Chen, Yuping Zheng, and Benyou Wang. Efficiently democratizing medical llms for 50 languages via a mixture of language family experts. *arXiv preprint arXiv:2410.10626*, 2024.

[58] Clément Christophe, Praveen K Kanithi, Prateek Munjal, Tathagata Raha, Nasir Hayat, Ronnie Rajan, Ahmed Al-Mahrooqi, Avani Gupta, Muhammad Umar Salman, Gurpreet Gosal, et al. Med42–evaluating fine-tuning strategies for medical llms: Full-parameter vs. parameter-efficient approaches. *arXiv preprint arXiv:2404.14779*, 2024. 9

[59] Zeming Chen, Alejandro Hernández Cano, Angelika Romanou, Antoine Bonnet, Kyle Matoba, Francesco Salvi, Matteo Pagliardini, Simin Fan, Andreas Köpf, Amirkeivan Mohtashami, et al. Meditron-70b: Scaling medical pretraining for large language models. *arXiv preprint arXiv:2311.16079*, 2023. 9

[60] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. 9

[61] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. 9

[62] Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. Rethinking the role of demonstrations: What makes in-context learning work? In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 11048–11064. Association for Computational Linguistics, 2022. 9

[63] Yisheng Song, Ting Wang, Puyu Cai, Subrota K. Mondal, and Jyoti Prakash Sahoo. A comprehensive survey of few-shot learning: Evolution, applications, challenges, and opportunities. *ACM Comput. Surv.*, 55(13s):271:1–271:40, 2023. 9

[64] Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah D. Goodman. Star: Bootstrapping reasoning with reasoning. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. 9

[65] Jiaxin Huang, Shixiang Gu, Le Hou, Yuexin Wu, Xuezhi Wang, Hongkun Yu, and Jiawei Han. Large language models can self-improve. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 1051–1068. Association for Computational Linguistics, 2023. 9

[66] Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Yarin Gal, Nicolas Papernot, and Ross J. Anderson. The curse of recursion: Training on generated data makes models forget. *CoRR*, abs/2305.17493, 2023. 9

[67] Sina Alemohammad, Josue Casco-Rodriguez, Lorenzo Luzi, Ahmed Imtiaz Humayun, Hossein Babaei, Daniel LeJeune, Ali Siahkoohi, and Richard G. Baraniuk. Self-consuming generative models go MAD. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. 9

[68] Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. 10

[69] Haipeng Luo, Qingfeng Sun, Can Xu, Pu Zhao, Jianguang Lou, Chongyang Tao, Xiubo Geng, Qingwei Lin, Shifeng Chen, and Dongmei Zhang. Wizardmath: Empowering mathematical reasoning for large language models via reinforced evol-instruct. *CoRR*, abs/2308.09583, 2023. 10

[70] Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. Self-refine: Iterative refinement with self-feedback. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. 10

[71] Angelica Chen, Jérémy Scheurer, Tomasz Korbak, Jon Ander Campos, Jun Shern Chan, Samuel R. Bowman, Kyunghyun Cho, and Ethan Perez. Improving code generation by training with natural language feedback. *CoRR*, abs/2303.16749, 2023.

[72] Sean Welleck, Ximing Lu, Peter West, Faeze Brahman, Tianxiao Shen, Daniel Khashabi, and Yejin Choi. Generating sequences by learning to self-correct. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. 10

[73] Zhiheng Xi, Senjie Jin, Yuhao Zhou, Rui Zheng, Songyang Gao, Jia Liu, Tao Gui, Qi Zhang, and Xuanjing Huang. Self-polish: Enhance reasoning in large language models via problem refinement. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 11383–11406. Association for Computational Linguistics, 2023.

[74] Debjit Paul, Mete Ismayilzada, Maxime Peyrard, Beatriz Borges, Antoine Bosselut, Robert West, and Boi Faltings. REFINER: reasoning feedback on intermediate representations. In Yvette Graham and Matthew Purver, editors, *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2024 - Volume 1: Long Papers, St. Julian's, Malta, March 17-22, 2024*, pages 1100–1126. Association for Computational Linguistics, 2024. 10

[75] Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*, 2022. 10

[76] Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. Large language models cannot self-correct reasoning yet. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. 10

[77] Wenda Xu, Guanglei Zhu, Xuandong Zhao, Liangming Pan, Lei Li, and William Wang. Pride and prejudice: LLM amplifies self-bias in self-refinement. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 15474–15492. Association for Computational Linguistics, 2024. 10

[78] Tianlu Wang, Ping Yu, Xiaoqing Ellen Tan, Sean O'Brien, Ramakanth Pasunuru, Jane Dwivedi-Yu, Olga Golovneva, Luke Zettlemoyer, Maryam Fazel-Zarandi, and Asli Celikyilmaz. Shepherd: A critic for language model generation. *CoRR*, abs/2308.04592, 2023. 10

[79] Bofei Gao, Zefan Cai, Runxin Xu, Peiyi Wang, Ce Zheng, Runji Lin, Keming Lu, Junyang Lin, Chang Zhou, Wen Xiao, Junjie Hu, Tianyu Liu, and Baobao Chang. LLM critics help catch bugs in mathematics: Towards a better mathematical verifier with natural language feedback. *CoRR*, abs/2406.14024, 2024.

[80] Aojun Zhou, Ke Wang, Zimu Lu, Weikang Shi, Sichun Luo, Zipeng Qin, Shaoqing Lu, Anya Jia, Linqi Song, Mingjie Zhan, and Hongsheng Li. Solving challenging math word problems using GPT-4 code interpreter with code-based self-verification. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024.

[81] Alexander Havrilla, Sharath Chandra Raparthy, Christoforos Nalmpantis, Jane Dwivedi-Yu, Maksym Zhuravinskyi, Eric Hambro, and Roberta Raileanu. Glore: When, where, and how to improve LLM reasoning via global and local refinements. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024. 10

[82] Xin Zheng, Jie Lou, Boxi Cao, Xueru Wen, Yuqiu Ji, Hongyu Lin, Yaojie Lu, Xianpei Han, Debing Zhang, and Le Sun. Critic-cot: Boosting the reasoning abilities of large language model via chain-of-thoughts critic, 2024. 10

[83] Aviral Kumar, Vincent Zhuang, Rishabh Agarwal, Yi Su, John D. Co-Reyes, Avi Singh, Kate Baumli, Shariq Iqbal, Colton Bishop, Rebecca Roelofs, Lei M. Zhang, Kay McKinney, Disha Shrivastava, Cosmin Paduraru, George Tucker, Doina Precup, Feryal M. P. Behbahani, and Aleksandra Faust. Training language models to self-correct via reinforcement learning. *CoRR*, abs/2409.12917, 2024. 10

[84] Xidong Wang, Guiming Hardy Chen, Dingjie Song, Zhiyi Zhang, Zhihong Chen, Qingying Xiao, Feng Jiang, Jianquan Li, Xiang Wan, Benyou Wang, et al. Cmb: A comprehensive medical benchmark in chinese. *arXiv preprint arXiv:2308.08833*, 2023. 23

[85] Junling Liu, Peilin Zhou, Yining Hua, Dading Chong, Zhongyu Tian, Andrew Liu, Helin Wang, Chenyu You, Zhenhua Guo, Lei Zhu, et al. Benchmarking large language models on cmexam-a comprehensive chinese medical exam dataset. *Advances in Neural Information Processing Systems*, 36, 2024. 23

[86] Haonan Li, Yixuan Zhang, Fajri Koto, Yifei Yang, Hai Zhao, Yeyun Gong, Nan Duan, and Timothy Baldwin. Cmmlu: Measuring massive multitask language understanding in chinese. *arXiv preprint arXiv:2306.09212*, 2023. 23

16

A Ethical Statement

Although the proposed model is a medical LLM with complex reasoning capabilities, it may still produce content that includes hallucinations or inaccuracies. Therefore, the current model is not suitable for real-world applications. Consequently, we will impose strict limitations on the use of our model. The models 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 adhere to these restrictions in order to safeguard the safety and integrity of their applications.

B Constructing Medical Verifiable Problems

To construct Medical Verifiable Problems, we begin by employing small models and rule-based methods to identify challenging questions. Subsequently, we leverage GPT-4o to perform data filtering, isolating questions that have been suitably transformed. The prompt used for this data filtering process is illustrated in Figure 5. After selecting appropriate data, we reformat multiple-choice medical exam questions into open-ended verifiable problems using the prompt provided in Figure 6.

The prompt for filtering Multiple-choice Questions

\<Multiple-choice Question>
{Question}
{Options}
Correct Answer: {Answer}
\</Multiple-choice Question>

You are an expert in filtering and evaluating multiple-choice questions for advanced reasoning tasks. Your job is to evaluate a given question and determine whether it meets the following criteria:
1. **Depth of Reasoning:** The question should require deeper reasoning. If the question appears too simple, mark it as "Too Simple."
2. **Unambiguous Correct Answer:** The question must have a unique and unambiguous correct answer. If the question asks for "incorrect options" or allows for multiple correct answers, mark it as "Ambiguous Answer."
3. **Open-Ended Reformulation Feasibility:** The question should be suitable for reformatting into an open-ended format. If the question cannot be easily reformulated into an open-ended problem and a clear ground-truth answer, mark it as "Not Reformulatable."

For each question, provide one of the following evaluations:
- "Pass" (The question meets all the criteria.)
- "Too Simple"
- "Ambiguous Answer"
- "Not Reformulatable"

Figure 5: The prompt for filtering Multiple-choice Questions. Here, {Question} and {Options} represents the multiple-choice question and options, and {Answer} represents the correct option for the multiple-choice question.

The prompt for reformatting multiple-choice questions to open-ended verifiable problems

I will provide you with a multiple-choice question, and your task is to rewrite it into an open-ended question, along with a standard answer. The requirements are:

1. The question must be specific, targeting the point being tested in the original multiple-choice question. Ensure it is open-ended, meaning no options are provided, but there must be a definitive standard answer.
2. Based on the correct answer from the original question, provide a concise standard answer. The answer should allow for precise matching to determine whether the model's response is correct.

Here is the multiple-choice question for you to rewrite:
\<Multiple-choice Question>
{Question}
{Options}

```
Correct Answer: {Answer}
</Multiple-choice Question>

Please output the result in the following JSON format:
```json
{{
"Open-ended Verifiable Question": "...",
"Standard Answer": "..."
}}
```
```

Figure 6: The prompt for reformatting multiple-choice questions to open-ended verifiable problems. Here, {Question} and {Options} represents the multiple-choice question and options, and {Answer} represents the correct option for the multiple-choice question.

C The Prompt of Verifier

GPT-4o serves as the verifier to assess the correctness of model-generated outputs. Using the prompt depicted in Figure 7, we present GPT-4o with both the model's output and the ground-truth answer to evaluate the correctness of the response. The verifier returns a Boolean value: **True** if the response is accurate and **False** otherwise.

The Prompt for Verifier

<Model Response>
{Model Response}
</Model Response>

<Reference Answer>
{Ground-true Answer}
</Reference Answer>

You are provided with a model-generated response (<Model Response>) and a reference answer (<Reference Answer>). Compare the model response with the reference answer and determine its correctness. Your task is to simply output "True" if the response is correct, and "False" otherwise.

Figure 7: The prompt for the GPT-4o verifier. {Model Response} represents the output of the model to be verified. {Ground-true Answer} represents the ground-truth answer for medical verifiable problems.

D Prompts for Searching Trajectories

This section outlines the prompts used for constructing complex Chain-of-Thought (CoT) reasoning pathways. Initially, a question x is presented to GPT-4o, which generates an initial CoT response using the prompt shown in Figure 8. If the verifier determines the response to be incorrect, GPT-4o employs one of several search strategies to iteratively refine the output until it is accurate. The prompts for these four search strategies — **Backtracking**, **Exploring New Paths**, **Correction**, and **Verification** — are detailed in Figures 10, 10, 11, and 12, respectively.

The prompt for initial CoT

<question>
{Question}
</question>

Please respond to the above question <question> using the Chain of Thought (CoT) reasoning method. Your response should consist of multiple steps, each of which includes three types of actions: **"Inner Thinking"**, **"Final Conclusion"**, and **"Verification"**:

- **'Inner Thinking'**: This is the step where thinking is done. Note that multiple 'Inner Thinking' steps are required to describe thorough reasoning. Each step should first generate a brief title.
- **'Final Conclusion'**: At this stage, you summarize the correct reasoning from previous 'Inner Thinking' steps and provide the final answer. No title is required here.
- **'Verification'**: At this stage, you verify the conclusion from the "Final Conclusion" step. If the conclusion holds, end the process. If not, return to "Inner Thinking" for further reasoning. No title is required here.

The output format must strictly follow the JSON structure below:
```json
{
"CoT": [
{"action": "Inner Thinking", "title": "...", "content": "..."},
...,
{"action": "Final Conclusion", "content": "..."},
{"action": "Verification", "content": "..."}
]
}
```

Figure 8: The prompt for initial CoT. `{Question}` represents the input question, i.e., the question x of the medical verifiable problems.

The Prompt for **Backtracking** Breask Search Strategy

<question>
`{Question}`
</question>

<previous reasoning>
`{Previous_CoT}`
<previous reasoning>

<response requirements>
Your response must include the following steps, each composed of three types of actions: **"Inner Thinking"**, **"Final Conclusion"**, and **"Verification"**:

1. **Inner Thinking**: Break down the reasoning process into multiple concise steps. Each step should start with a brief title to clarify its purpose.
2. **Final Conclusion**: Summarize the correct reasoning from all previous 'Inner Thinking' steps and provide the final answer. No title is needed for this section.
3. **Verification**: Verify the accuracy of the "Final Conclusion". If it holds, conclude the process. Otherwise, return to "Inner Thinking" for further refinement.
</response requirements>

<question> represents the question to be answered, and <previous reasoning> contains your prior reasoning. Your task is to continue from the current 'Verification' step. I have manually reviewed the reasoning and determined that the **Final Conclusion** is false. Your 'Verification' results must align with mine. Proceed to refine the reasoning using **backtracking** to revisit earlier points of reasoning and construct a new Final Conclusion.

Output Format
Strictly follow the JSON structure below. You do not need to repeat your previous reasoning. Begin directly from the next 'Verification' stage.

```json
{
"CoT": [
{"action": "Verification", "content": "..."},
{"action": "Inner Thinking", "title": "...", "content": "..."},
...,
{"action": "Final Conclusion", "content": "..."},
{"action": "Verification", "content": "..."}
```

```
    ]
  }
“‘
```

Figure 9: The prompt for **Backtracking** search strategy. Here, {Question} represents the problem x of the medical verifiable problems, and {Previous_CoT} represents the previous chain of thought process, i.e., $[e_0, y_0, \ldots, e_{i-1}, y_{i-1}]$.

The Prompt for Exploring New Paths Breask Search Strategy

<question>
{Question}
</question>

<previous reasoning>
{Previous_CoT}
<previous reasoning>

<response requirements>
Your response must include the following steps, each composed of three types of actions: **"Inner Thinking"**, **"Final Conclusion"**, and **"Verification"**:

1. **Inner Thinking**: Break down the reasoning process into multiple concise steps. Each step should start with a brief title to clarify its purpose.
2. **Final Conclusion**: Summarize the correct reasoning from all previous 'Inner Thinking' steps and provide the final answer. No title is needed for this section.
3. **Verification**: Verify the accuracy of the "Final Conclusion". If it holds, conclude the process. Otherwise, return to "Inner Thinking" for further refinement.

</response requirements>

<question> represents the question to be answered, and <previous reasoning> contains your prior reasoning. Your task is to continue from the current 'Verification' step. I have manually reviewed the reasoning and determined that the **Final Conclusion** is false. Your 'Verification' results must align with mine. Proceed to refine the reasoning by exploring new approaches to solving this problem and construct a new Final Conclusion.

Output Format
Strictly follow the JSON structure below. You do not need to repeat your previous reasoning. Begin directly from the next 'Verification' stage.

```json
{
"CoT": [
{"action": "Verification", "content": "..."},
{"action": "Inner Thinking", "title": "...", "content": "..."},
...,
{"action": "Final Conclusion", "content": "..."},
{"action": "Verification", "content": "..."}
]
}
“‘
```

Figure 10: The prompt for **Exploring New Paths** search strategy. Here, {Question} represents the problem x of the medical verifiable problems, and {Previous_CoT} represents the previous chain of thought process, i.e., $[e_0, y_0, \ldots, e_{i-1}, y_{i-1}]$.

The Prompt for Correction Breask Search Strategy

<question>
{Question}

</question>

<previous reasoning>
{Previous_CoT}
<previous reasoning>

<response requirements>
Your response must include the following steps, each composed of three types of actions: **"Inner Thinking"**, **"Final Conclusion"**, and **"Verification"**:

1. **Inner Thinking**: Break down the reasoning process into multiple concise steps. Each step should start with a brief title to clarify its purpose.
2. **Final Conclusion**: Summarize the correct reasoning from all previous 'Inner Thinking' steps and provide the final answer. No title is needed for this section.
3. **Verification**: Verify the accuracy of the "Final Conclusion". If it holds, conclude the process. Otherwise, return to "Inner Thinking" for further refinement.

</response requirements>

<question> represents the question to be answered, and <previous reasoning> contains your prior reasoning. Your task is to continue from the current 'Verification' step. I have manually reviewed the reasoning and determined that the **Final Conclusion** is false. Your 'Verification' results must align with mine. Proceed to refine the reasoning by making precise **corrections** to address prior flaws and construct a new Final Conclusion.

Output Format
Strictly follow the JSON structure below. You do not need to repeat your previous reasoning. Begin directly from the next 'Verification' stage.

```json
{
"CoT": [
{"action": "Verification", "content": "..."},
{"action": "Inner Thinking", "title": "...", "content": "..."},
...,
{"action": "Final Conclusion", "content": "..."},
{"action": "Verification", "content": "..."}
]
}
```

Figure 11: The prompt for **Correction** search strategy. Here, {Question} represents the problem x of the medical verifiable problems, and {Previous_CoT} represents the previous chain of thought process, i.e., $[e_0, y_0, \ldots, e_{i-1}, y_{i-1}]$.

The Prompt for **Verification** Breask Search Strategy

<question>
{Question}
</question>

<previous reasoning>
{Previous_CoT}
<previous reasoning>

<response requirements>
Your response must include the following steps, each composed of three types of actions: **"Inner Thinking"**, **"Final Conclusion"**, and **"Verification"**:

1. **Inner Thinking**: Break down the reasoning process into multiple concise steps. Each step should start with a brief title to clarify its purpose.
2. **Final Conclusion**: Summarize the correct reasoning from all previous 'Inner Thinking' steps and provide the final answer. No title is needed for this section.

3. **Verification**: Verify the accuracy of the "Final Conclusion". If it holds, conclude the process. Otherwise, return to "Inner Thinking" for further refinement.

</response requirements>

<question> represents the question to be answered, and <previous reasoning> contains your prior reasoning. Your task is to continue from the current 'Verification' step. I have manually reviewed the reasoning and determined that the **Final Conclusion** is false. Your 'Verification' results must align with mine. Proceed to refine the reasoning by conducting a thorough **validation** process to ensure validity and construct a new Final Conclusion.

Output Format
Strictly follow the JSON structure below. You do not need to repeat your previous reasoning. Begin directly from the next 'Verification' stage.

```json
{
"CoT": [
{"action": "Verification", "content": "..."},
{"action": "Inner Thinking", "title": "...", "content": "..."},
...,
{"action": "Final Conclusion", "content": "..."},
{"action": "Verification", "content": "..."}
]
}
```

Figure 12: The prompt for **Verification** search strategy. Here, {Question} represents the problem x of the medical verifiable problems, and {Previous_CoT} represents the previous chain of thought process, i.e., $[e_0, y_0, \ldots, e_{i-1}, y_{i-1}]$.

E Prompts for Constructing SFT Training Data

When a successful trajectory $[e_0, y_0, \ldots, e_i, y_i]$ is found, it is reformatted into a coherent, natural language reasoning process $\hat{e}$ (*Complex CoT*) using the prompt shown in Figure 13. This reformatting avoids rigid structures, using smooth transitions (e.g., "hmm," "also," "wait") to streamline reasoning and reduce token usage. The model then generates a formal response $\hat{y}$ for for question x using the conclusion of $\hat{e}$ with the prompt in Figure 13.

The prompt for reformatting a reasoning trajectory to complex CoT

<Thought Process>
{Thought_Process}
</Thought Process>

<Question>
{Question}
</Question>

The <Thought Process> above reflects the model's reasoning based on the <Question>. Your task is to rewrite the <Thought Process> to resemble a more human-like, intuitive natural thinking process. The new version should:

1. Be presented as step-by-step reasoning, with each thought on a new line separated by a line break.
2. Avoid structured titles or formatting, focusing on natural transitions. Use casual and natural language for transitions or validations, such as "hmm," "oh," "also," or "wait."
3. Expand the content, making the reasoning richer, more detailed, and logically clear while still being conversational and intuitive.

Return directly the revised natural thinking in JSON format as follows:
```json
{

```
"NaturalReasoning": "..."
}
```

Figure 13: The prompt for reformatting a reasoning trajectory to complex CoT $\hat{e}$. Here, `{Thought_Process}` represents the successful reasoning trajectory of $[e_0, y_0, \ldots, e_i, y_i]$, and `{Question}` represents the question x.

The prompt for generating a formal response with complex CoT

<Internal Thinking>
`{Complex_CoT}`
</Internal Thinking>

<Question>
`{Question}`
</Question>

The <Internal Thinking> represents your internal thoughts about the <Question>. Based on this, generate a rich and high-quality final response to the user. If there is a clear answer, provide it first. Ensure your final response closely follows the <Question>. The response style should resemble GPT-4's style as much as possible. Output only your final response, without any additional content.

Figure 14: The prompt for generating a formal response $\hat{y}$ with complex CoT $\hat{e}$. Here, `{Complex_CoT}` represents the complex CoT $\hat{e}$, and `{Question}` represents the question x.

F Settings of other RL training

we further compared different RL-related algorithms with PPO. Specifically, we employed the preference-learning algorithm DPO and the REINFORCE-style algorithm RLOO.

DPO   For DPO, we had the model generate five answers for each question offline and used a verifier to identify pairs of one correct and one incorrect answer. If no such pairs were found, the data was discarded. Verified correct answers were used as positive examples, while failed verifications served as negative examples for training DPO. The hyperparameters for DPO training were set as follows: learning rate of 1e-6, batch size of 128, and a regularization parameter of 1.

RLOO   For RLOO, we used the same reward function as PPO. The parameters were also identical to those of PPO, with an additional parameter rloo_k set to 2.

G Chinese Medical Model

Model Training   For the Chinese medical domain, we replaced the exam questions from the CMB training set for Chinese medical verifiable problems. Based on the same training process as the English version of HuatuoGPT-o1, we developed HuatuoGPT-o1-7B-zh, built on the Qwen2.5-7B-Instruct model.

Chinese Medical Evaluation   To assess the Chinese medical capabilities, we evaluated the model on three Chinese medical benchmarks, including the Chinese test set from MedQA (MCMLE) [17], the test set from CMB-Exam [84], and the test set from CMExam [85]. Additionally, we evaluated the model on the medical section of the Chinese general evaluation benchmark CMMLU [86], covering tracks of 'clinical knowledge,' 'agronomy,' 'college medicine,' 'genetics,' 'nutrition,' 'Traditional Chinese Medicine,' and 'virology'.

Comparison Models   We compared the performance of three general Chinese models: Qwen2.5, GLM-4, and Yi. Additionally, we included a comparison with a Chinese medical model, HuatuoGPT-2-7B [15].