REFLECTOOL: Towards Reflection-Aware Tool-Augmented Clinical Agents

Anonymous ACL submission

Abstract

Large Language Models (LLMs) have shown promising potential in the medical domain, assisting with tasks like clinical note generation and patient communication. However, current LLMs are limited to text-based communication, hindering their ability to interact with diverse forms of information in clinical environments. Despite clinical agents succeeding in diverse signal interaction, they are oriented to a single clinical scenario and hence fail for broader applications. To evaluate clinical agents holistically, we propose ClinicalAgent Bench (CAB), a comprehensive medical agent benchmark consisting of 18 tasks across five key realistic clinical dimensions. Building on this, we introduce REFLECTOOL, a novel framework that excels at utilizing domain-specific tools within two stages. The first optimization stage progressively enlarges a long-term memory by saving successful solving processes and tool-wise experience of agents in a tiny pre-defined training set. In the following inference stage, REFLECTOOL can search for supportive successful demonstrations from already built long-term memory to guide the tool selection strategy, and a verifier improves the tool usage according to the tool-wise experience with two verification methods–iterative refinement and candidate selection. Extensive experiments on CAB demonstrate that REFLECTOOL surpasses the pure LLMs with more than 10 points and the well-established agent-based methods with 3 points, highlighting its adaptability and effectiveness in solving complex clinical tasks.

1 Introduction

Large Language Models (LLMs) have shown significant potential in the medical domain (Singhal et al., 2023; Nori et al., 2023; Chen et al., 2023a), demonstrating their ability to assist with tasks such as generating clinical notes (Biswas and Talukdar, 2024; Jung et al., 2024) and supporting patient communication (Tu et al., 2024; Liao et al., 2024). However, LLMs are restricted to direct text-based responses rather than serving as a bridge to leverage the information in other forms, thus impeding their effective application in realistic clinical scenarios.

To address such shortcoming, numerous works developed more advanced clinical agents, which enable models to leverage complex information through specialized tools (Jin et al., 2024; Li et al., 2024a; Lin et al., 2024). For instance, EHRAgent (Shi et al., 2024) can access electronic health records (EHRs) via a code interface, and MMedAgent (Li et al., 2024a) can interpret medical images via several medical visual models (Li et al., 2024b; Ma et al., 2024). While these agents enhance LLMs' ability to interact with various types of data, they remain limited to addressing specific clinical scenarios with a narrow range of tools, impeding their ability to interact with the diverse forms of information intrinsic to clinical environments (Hu et al., 2024; Lee et al., 2022; Adams et al., 2024). This lack of integration limits their effectiveness for further application in clinical scenarios.

In this paper, we analyze representative public benchmarks in the medical field and categorize them based on the capability requirements of medical agents. We build a comprehensive medical agent benchmark, ClinicalAgent Bench (CAB), comprising 18 tasks across five dimensions in total. Specifically, the dimensions of CAB include Knowledge & Reasoning, MultiModal, Numerical Analysis, Data Understanding, and Trustworthiness. These dimensions require clinical agents to reason with medical knowledge effectively, integrate information from diverse clinical data sources (including medical images, EHRs, clinical text, and multiple clinical documents), and reduce hallucinations to ensure trustworthiness. Compared to previous benchmarks, CAB provides a more holistic evaluation framework by encompassing a wider range of clinical tasks and assessing agent capabili-

Methods	Agent Capacities					Agent Methods			
	Knowledge& Reasoning	MultiModal	Numerical Analysis	Data Understanding	Trustworthiness	Tool Use	Long-Term Memory	Tool-wise Reflection	w/o Fine Tuning
MedAgent (Tang et al., 2024)	✓	✗	✗	✓	✗	✗	✗	✗	✓
MMedAgent (Li et al., 2024a)	✓	✓	✗	✗	✗	✓	✗	✗	✗
MedRAG (Xiong et al., 2024)	✓	✗	✗	✗	✗	✓	✗	✗	✓
AgentMD (Jin et al., 2024)	✓	✗	✓	✗	✗	✓	✗	✗	✓
EHRAgent (Lee et al., 2022)	✓	✗	✓	✗	✓	✓	✓	✗	✓
CTAgent (Yue and Fu, 2024)	✓	✓	✗	✗	✓	✓	✗	✗	✗
BKGAgent (Lin et al., 2024)	✓	✗	✗	✗	✗	✓	✗	✗	✓
REFLECTOOL (Ours)	✓	✓	✓	✓	✓	✓	✓	✓	✓

Table 1: Comparison of previous medical agents and REFLECTOOL on both agent capacities and methods.

ties across multiple scenarios.

Motivated by five critical aspects of CAB, we develop a set of clinical tools that enables agents to handle diverse tasks encompassed in the benchmark. Building upon the clinical toolbox, we proposed REFLECTOOL, a framework that allows agents to learn how to choose and leverage domain-specific tools to solve tasks. Specifically, REFLECTOOL consists of two stages. The first stage is the optimization stage. The agent attempts to solve problems using tools on a small proportion of samples and generates successful trajectories through self-reflection. By comparing successful and failed trajectories, the agent produces the tool-wise suggestion and stores successful trajectories as long-term memory. In the inference stage, the agent retrieves similar successful cases from long-term memory to optimize the tool selection. Each time a tool is used, the agent improves tool usage according to the accumulated tool-wise experience from the optimization stage. Furthermore, we adopt two verification methods, iterative refinement and candidate selection, to investigate the effectiveness of the tool-wise experience. We find that these two methods perform better under different model strengths, thereby enhancing the applicability of our approach. As discussed, REFLECTOOL demonstrates not only proficiency in a wide range of clinical critical aspects from CAB but also more effective tool utilization strategies, as the comparison with existing clinical agents shown in Table 1.

In summary, our contributions are as follows:

- **Holistic Benchmark:** We introduce CAB, a benchmark comprising 18 tasks across five principal dimensions. To the best of our knowledge, CAB is the first benchmark covering a wide range of tasks to evaluate the capabilities of clinical agents comprehensively.
- **Almighty Tool-Augmented Agent:** We propose REFLECTOOL, a novel framework that enables models to effectively utilize domain-specific tools. REFLECTOOL uses long-term memory and tool-wise verification to alleviate the problem in domain-tool selection and usage, thus improving adaptability across a wide range of clinical scenarios.
- **Revision-based explorations:** We explore two tool-wise verification methods, i.e., Iterative Refinement and Candidate Selection, designed to optimize tool usage. Our findings indicate that Iterative Refinement is more effective when the model exhibits lower capabilities, whereas Candidate Selection outperforms the former on more intelligent models.
- **Superior downstream improvements:** We conduct extensive experiments on CAB, benchmarking REFLECTOOL against a diverse array of established methods. Our results demonstrate the superior performance of REFLECTOOL, highlighting its effectiveness in clinical tool utilization.

2 ClinicalAgent Bench

In this section, we introduce the CAB, a novel benchmark to evaluate the capacity of the medical agent in clinical scenarios. The overview of the benchmark is shown in Figure 1. We first discuss the composition of the CAB in detail and then introduce the construction of the pre-built toolbox.

2.1 Composition

The agents in clinical scenarios need to process the medical data in different formats, like medical images (Hu et al., 2024; Liu et al., 2021; Lau et al., 2018) and electronic health records (EHR) (Lee et al., 2022; Johnson et al., 2016; Pollard et al., 2018), to complete the analysis or diagnosis. However, previous works only focused on a simple scenario, with only limited types of tools to solve the

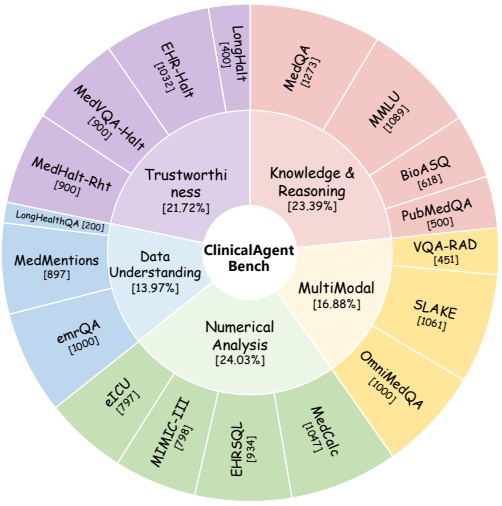

Figure 1: Overview of the proposed CAB. The numbers in the inner circle represent the proportion of data in each dimension, and the numbers in the outer circle represent the size of each dataset.

problem (Shi et al., 2024; Li et al., 2024a; Tang et al., 2024). To approximate realistic clinical scenarios and evaluate the general capabilities of the agent in the medical field, we investigate existing public medical datasets and divide them according to the ability requirement of the agents. We built a benchmark term CAB, which contains five capacity dimensions and 18 tasks in total. Specifically, the definition of the five dimensions and the corresponding tasks are as follows:

Knowledge&Reasoning Medical knowledge and reasoning is a critical capacity for the medical agent to analyze and complete tasks (Jin et al., 2019). To evaluate the agent performance, we choose PubMedQA (Jin et al., 2019), MMLU (Hendrycks et al.), and BioASQ (Krithara et al., 2023) for the medical knowledge question-answering (QA) and MedQA (Jin et al., 2021) for medical reasoning.

MultiModal Medical images[1] are a common form of information in clinical scenarios. Many diseases require a combination of image examination information to be accurately diagnosed. Here, we choose three datasets with 12 modalities and a wide range of task types. We chose SLAKE (Liu et al., 2021) and VQA-RAD (Lau et al., 2018) for the open-ended QA and OmniMedVQA (Hu et al., 2024) for the closed-ended QA.

[1]The multimodal in this paper mainly indicates that medical images with multiple modalities.

Numerical Analysis Numerical analysis mainly contains two perspectives. One is the numerical calculation, commonly used in test results analysis and risk prediction (Jin et al., 2024). We choose MedCalc (Khandekar et al., 2024) to evaluate the agent capacity in equation-based and rule-based calculation tasks. The other is the database operation and understanding, where agents need to gather patient information from the EHR database to conduct further analysis. We evaluate agents on EHRSQL (Lee et al., 2022), MIMIC-III (Johnson et al., 2016), eICU (Pollard et al., 2018).

Data Understanding Clinical data understanding is the focus of conventional medical natural language processing (NLP) (). Agents are required to extract the key information or relations from the redundant report to understand the patient stats better. Therefore, we choose the name entity recognition dataset MedMentions (Mohan and Li, 2019), information extraction dataset emrQA (Pampari et al., 2018), and long context QA dataset LongHealthQA (Adams et al., 2024) to evaluate the data understanding capacity of the agents.

Trustworthiness For the application of the clinical agents, the trustworthiness of the response is very important. If clinical agents experience hallucinations while completing tasks, their responses may result in severe medical accidents. To comprehensively evaluate the hallucination that happened in the agents' solving process, we choose four types of datasets to validate the trustworthiness in four types of tasks: MedHalt-Rht (Pal et al., 2023), MedVQA-Halt (Wu et al., 2024), EHR-Halt (Lee et al., 2022; Johnson et al., 2016; Pollard et al., 2018), and LongHalt (Adams et al., 2024).

2.2 Clinical Toolbox

Based on the proposed CAB, we develop a toolbox that contains 16 types of tools to enable agents to handle diverse tasks. For example, knowledge databases in the clinical toolbox enhance the medical knowledge of the agent, and calculators give the agent the ability to calculate indicators accurately. In order to allow the agent to solve problems more flexibly, we did not limit the types and number of tools when solving a specific type of task. Compared with medical agents that limit the tools to complete tasks (Li et al., 2024a; Jin et al., 2024), our method can give the agent better generalization and scalability. The details of the clinical toolbox are discussed in Appendix B.2.

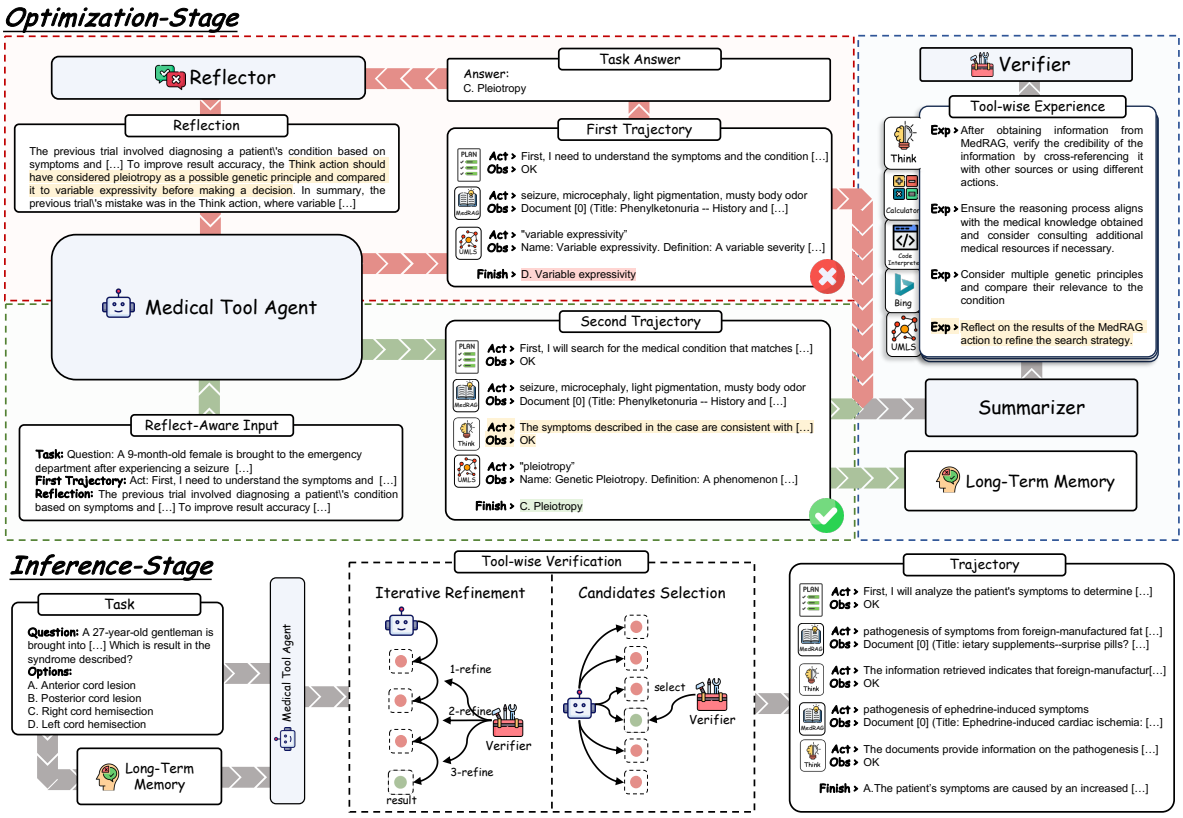

Figure 2: Overview of the REFLECTOOL.

3 REFLECTOOL

In this section, we first formulate the problem of solving tasks in CAB with the clinical toolbox. Then, we introduce the optimization and inference stage of the proposed REFLECTOOL. The overview of the REFLECTOOL are shown in Figure 2.

3.1 Problem Formulations

In this work, we focus on addressing clinical-related tasks with tool-use agents. The task is composed of $\mathcal{X} = \{q, \mathcal{I}\}$ and y, where q is the instruction, $\mathcal{I}$ is the inputs with different formats, and y is the ground-truth answer. The agents are required to leverage the input information and complete the task in a multi-step manner. For initialization, the action space of the agents is composed of a set of pre-built tool action $\mathcal{A}_T = \{A_1, A_2, \dots\}$ and three types of inner actions $\mathcal{A}_I = \{\text{Plan}, \text{Think}, \text{Finish}\}$. The whole action space of the agent can be noted as follows:

$$\mathcal{A} = \mathcal{A}_T \cup \mathcal{A}_I, \tag{1}$$

In the i-th step, the clinical agent takes action $a_i \in \mathcal{A}$. The action[2] in the clinical agent setting can be treated as the function, giving the proper parameter and getting the observation results $o_i = a_i(\text{param}_{a_i})$. param_{a_i} indicates the parameters of the action controlled by the agents. The next step action follows the policy:

$$a_{i+1} \sim \pi_\theta(a|c_i, \mathcal{X}) \tag{2}$$

where a_0 is the first action and θ indicates the parameter of the agent. $c_i = \{a_0, o_0, ...a_i, o_i\}$ is the trajectory history.

3.2 Optimization Stage

To enable the agent to better select and use the domain-specific tools, we choose a subset of samples to optimize the agent's capacities. In the optimization stage, REFLECTOOL saves the successful trajectory into the long-term memory and collects the experience for each type of tool.

Specifically, the REFLECTOOL first attempts to solve the problem with the clinical toolbox and create the first trajectory $\mathcal{C}_1$. The agent then reflects on $\mathcal{C}_1$ by comparing it with the ground-truth answer y and producing a suggestion $\mathcal{S}$

[2]In this paper, the usage form of tool actions $\mathcal{A}_T$ and inner actions $\mathcal{A}_I$ is the same. Without loss of generality, we consider actions and tools equivalent.

with $\text{LLM}(\mathcal{X}, \mathcal{C}_1, y) \rightarrow \mathcal{S}$. Utilizing this suggestion, the agent regenerates a refined trajectory $\mathcal{C}_2$: $\text{LLM}(\mathcal{X}, \mathcal{C}_1, \mathcal{S}) \rightarrow \mathcal{C}_2$. If C_2 successfully completes the task, the reflective trajectory will be saved into the long-term memory $\mathcal{M}$ to assist the agent in solving similar problems during inference:

$$\mathcal{M} = \begin{cases} \mathcal{M} \cup \{\mathcal{X}, \mathcal{C}_2\}, & y^{\mathcal{C}_2} = y \\ \mathcal{M}, & y^{\mathcal{C}_2} \neq y \end{cases} \quad (3)$$

where $y^{\mathcal{C}_2}$ indicates the prediction of trajectory $\mathcal{C}_2$. For the successful $\mathcal{C}_2$, the agent turns to compare the usage of each action that appears in two trajectories, to generate action-wise suggestions

$$\text{LLM}(\mathcal{X}, \mathcal{C}_1, \mathcal{C}_2, y) \rightarrow \mathcal{E}_{\mathcal{X}} \quad (4)$$

Then, $\mathcal{E}_{\mathcal{X}}$ will be merge into the tool-wise experience $\mathcal{E} = \{E_1, E_2, ...\} \cup \{E_{\text{Plan}}, E_{\text{Think}}, E_{\text{Finish}}\}$, where E_i is the experience of the A_i. The optimization process is shown in Algorithm 1.

3.3 Inference Stage

During the inference stage, REFLECTOOL utilizes the long-term memories and tool-wise experiences learned from the optimization stage to solve the task better. At first, REFLECTOOL retrieves similar cases from the long-term memory:

$$\mathcal{M}(q) = \text{TopK}_{\text{max}}(\text{sim}(q, q_i | q_i \in \mathcal{M})) \quad (5)$$

where TopK_{max} return the top k most similar elements from the long-term memory. $\text{sim}(\cdot, \cdot)$ is the similarity function with BM25 (Robertson et al., 2009) used in implementations. Then, the REFLECTOOL can take the action with the help of similar successful trajectories as below:

$$a_{i+1} \sim \pi_\theta(a | c_i, \mathcal{X}, \mathcal{M}(q)) \quad (6)$$

The agent then solves the task with tool-wise reflection, where the verifier evaluates the agent's action in each step according to the action-wise experience. Inspired by Snell et al. (2024), we adopt two types of verification methods, iterative refinement and candidate selection, to fully explore the effectiveness of tool-wise experience in the tool-wise reflection process. in §5.2, we find that each variant has its own advantages: Iterative Refinement performs better when the model's capabilities are limited, while Candidate Selection is more effective when the model is stronger, thereby maximizing the model's potential.

Iterative Refinement In the sequence refinement method, the verifier will keep refining the agent action until it achieves the max refine step. This process can be early-stopped if the verifier outputs an identical action at any refinement step. Specifically, the i-th step initial action a_i^0 is generated as the Eq. 6. Then, the verifier will refine the action based on the tool-wise experience:

$$a_i^j = \text{LLM}(c_i, a_i^{0:j-1}, \mathcal{X}, \mathcal{M}(q), \mathcal{E}(a_i^{0:j-1})) \quad (7)$$

and the final refined result is chosen as the action of the current step:

$$a_i = \begin{cases} a_i^j, & \text{if } a_i^j = a_i^{j-1} \\ a_i^n, & \text{otherwise} \end{cases} \quad (8)$$

where $j = 1, 2, ..., n$ means the refinement step index, n is the max refinement step. The refined history $a_i^{0:j} = \{a_i^0, a_i^1, ..., a_i^j\}$ and $\mathcal{E}(a_i^j)$ indicates the corresponding experience of the action type.

Candidates Selection For the candidate selection method, REFLECTOOL first samples n candidate actions from the output space. Then, the verifier will select the most effective action from the candidate list $a_i^{0:n}$:

$$a_{i+1} = \arg\max_{r \in \mathcal{R}} p_\theta(a | c_i, \mathcal{X}, \mathcal{M}(q), \mathcal{E}(a_i^{0:n})) \quad (9)$$

where $p_\theta(\cdot)$ indicates the preference of the reflector. Here, n is the size of the candidate list.

4 Experiments

4.1 Baselines

To comprehensively validate the effectiveness of the proposed method, we select several types of methods as the baselines. Considering that the proposed agent bench covers a wide range of tasks and requires the models to leverage the input information in different formats, we only choose the methods with strong instruction-following capacity. The baselines include three types of methods: LLMs, MLLMs, and agent-based methods. For LLMs, we choose **MedLlama3-8B**[3], **Qwen2-7B/72B** (Yang et al., 2024), **Llama3-8B** (AI@Meta, 2024), **Llama3.1-8B/70B** (Dubey et al., 2024), and **GPT-3.5-turbo** (OpenAI, 2022). For MLLMs, we choose **MiniCPM-V-2.6** (Yao et al., 2024), **InternVL-Chat-V1.5** (Chen et al.,

[3] https://huggingface.co/johnsnowlabs/JSL-MedLlama-3-8B-v2.0

Models	Know.	MM.	Num.	Data.	Trust.	Total
Large Language Models						
MedLlama3-8B	59.88	-	11.61	23.02	6.07	25.14
Qwen2-7B (Yang et al., 2024)	60.86	-	18.49	44.50	28.19	38.01
Llama3-8B (AI@Meta, 2024)	63.32	-	19.87	35.23	24.06	35.62
Llama3.1-8B (Dubey et al., 2024)	67.43	-	22.07	49.58	30.75	42.46
Qwen2-72B* (Yang et al., 2024)	72.90	-	31.07	50.61	40.45	48.76
Llama3.1-70B* (Dubey et al., 2024)	76.91	-	29.23	45.80	38.40	47.59
GPT-3.5-turbo (OpenAI, 2022)	63.64	-	19.18	24.26	18.17	31.31
MultiModal Large Language Models						
MiniCPM-V-2.6 (Yao et al., 2024)	56.29	56.53	4.60	13.86	15.12	29.28
InternVL-Chat-V1.5 (Chen et al., 2023b)	52.92	53.21	18.95	34.50	25.52	37.02
HuatuoGPT-Vision-7B (Chen et al., 2024)	60.96	66.24	9.07	42.73	26.36	41.07
HuatuoGPT-Vision-34B (Chen et al., 2024)	62.25	67.33	13.22	29.01	34.76	41.31
GPT-4o-mini (OpenAI, 2023)	73.65	48.47	29.61	50.05	**57.91**	51.94
Agent (Qwen2-7B)						
COT (Wei et al., 2022)	58.91	-	19.98	36.17	44.68	39.94
ReAct (Yao et al., 2023)	62.03	49.47	24.05	29.24	53.87	43.73
CRITIC (Gou et al., 2024)	56.61	53.87	24.49	37.35	47.54	43.97
Reflexion (Shinn et al., 2023)	60.92	56.95	20.83	37.41	50.14	45.25
REFLECTOOL (Iterative Refinement, n=2)	63.79	60.83	21.97	51.65	48.60	**49.37**
REFLECTOOL (Candidates Selection, n=2)	62.81	61.91	26.78	52.20	41.72	49.08
Agent (Qwen2-72B)*						
COT (Wei et al., 2022)	69.11	-	24.47	52.51	56.22	50.58
ReAct (Yao et al., 2023)	76.47	56.37	31.44	53.29	48.98	53.31
CRITIC (Gou et al., 2024)	74.01	54.96	30.92	55.15	46.69	52.35
Reflexion (Shinn et al., 2023)	76.79	60.95	31.99	58.37	53.75	56.37
REFLECTOOL (Iterative Refinement, n=2)	76.81	63.74	**38.45**	63.51	54.65	59.43
REFLECTOOL (Candidates Selection, n=2)	76.27	62.70	38.06	**64.54**	56.73	**59.66**

Table 2: Experimental results of four types of models on Clinical Agent Bench. The 'COT' method indicates the agent runs without the pre-built tools. '*' indicates the models use 4-bit GPTQ quantization. '-' means the model is not capable of solving such a task. The best results of each type of task are **Bold**.

2023b), **HuatuoGPT-Vision-7B/34B** (Chen et al., 2024), and **GPT-4o-mini** (OpenAI, 2023). For agent-based methods, **COT** (Wei et al., 2022) and **ReAct** (Yao et al., 2023) indicate the agent solving the task without and with the pre-build toolbox, respectively. **CRITIC** (Gou et al., 2024) and **Reflexion** (Shinn et al., 2023) improve agent capacity with self-reflection methods.

4.2 Main Results

We choose the Qwen2 series models as the backbone of the REFLECTOOL for the promising performance in tool usage and select the model parameters with 7B and 72B to observe the impact of the model size. We show the average performance of each dimension in Table 2, and the complete results are shown in Table 5. For the intragroup comparison, Qwen2-72B and GPT-4o-mini achieve the best performance in LLM-based and MLLM-based methods, respectively. For the agent-based method, REFLECTOOL surpasses the strong baseline method, Reflexion, with at least 3 points with both Qwen2-7B/72B. As both methods are based on self-reflection, these results highlight the advantage of the REFLECTOOL in using the domain tool. Besides, both types of tool-wise reflection methods show similar results when the reflection size is 2. For intergroup comparison, though MLLM-based methods are capable of Multi-modal tasks, they fall short in Numerical Analysis and Data Understanding. It is noteworthy that REFLECTOOL surpasses the base models with more than 10 points on both Qwen2-7B/72B, showing the effectiveness of the proposed methods again.

4.3 Ablation

To validate the effectiveness and the impact of each module in REFLECTOOL, we conduct the ablation experiments on the subset of CAB. We randomly select 200 samples from one dataset for each dimension. The results of the iterative refinement and the candidate selection are both shown in Table 3.

Reflect Type	Reflective Memory		Action-wise Reflection		PubMedQA	VQA-RAD	EHRSQL	MedMen	MedHalt	Avg.
	Long-Term Memory	Memory Reflection	Step Reflection	Action Eperience						
Iterative Refinement	✗	✗	✓	✓	68.50	60.00	40.50	52.27	45.00	53.25
	✓	✗	✓	✓	**69.50**	58.00	**47.00**	54.09	48.00	55.32
	✓	✓	✗	✗	67.00	**60.50**	42.00	55.86	47.50	54.57
	✓	✓	✓	✗	66.50	56.00	46.00	53.18	49.50	54.24
	✓	✓	✓	✓	68.00	58.50	46.00	**57.06**	**55.50**	**57.01**
Candidate Selection	✗	✗	✓	✓	**69.50**	59.50	33.33	51.38	52.50	53.24
	✓	✗	✓	✓	67.00	59.00	38.29	58.76	54.00	55.41
	✓	✓	✗	✗	66.00	56.00	46.00	59.70	48.00	55.14
	✓	✓	✓	✗	67.50	57.00	45.34	**60.72**	47.50	55.61
	✓	✓	✓	✓	**69.50**	**61.00**	**48.16**	60.24	**62.50**	**60.28**

Table 3: Ablation results of Refinement and Selection verification methods. All the experiments are conducted on Qwen2-72B. The modules of the REFLECTOOL contain Reflective Memory and Action-wise Reflection.

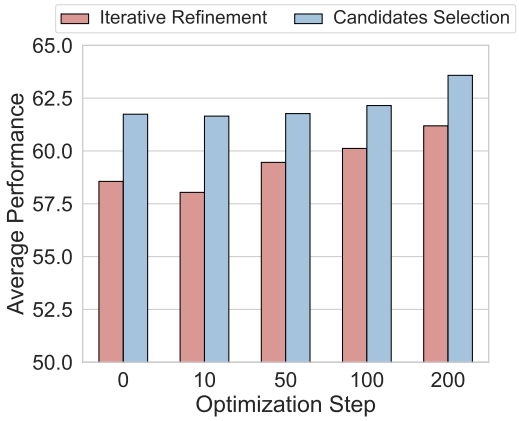

Figure 3: Impact of the verification size on refinement and selection methods. The average performance is calculated using the same datasets in the ablation study.

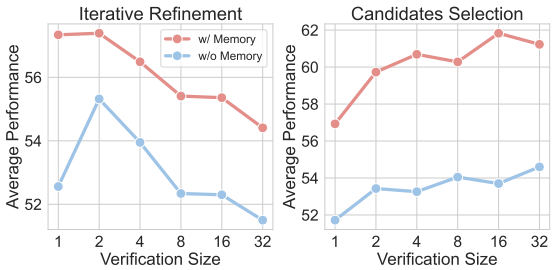

Figure 4: Impact of the verification size on refinement and selection methods. The average performance is calculated using the same datasets in the ablation study.

The selection methods perform better than the refinement methods in most cases. It is observed that the reflective long-term memory plays the most important role in REFLECTOOL, and its absence leads to performance degradation with nearly 4 points for refinement and 7 points for selection. Besides, the action-wise reflection also shows its importance. These results show the effectiveness of each module in the proposed REFLECTOOL.

5 Analysis

In this section, all experiments are conducted on the same subset as the ablation experiments.

5.1 Effect of Optimization Step

In this section, we investigate the impact of the optimization step, which indicates the number of tasks completed during the optimization stage. Each successful task can provide one memory item and a list of tool-wise suggestions. A larger optimization step implies that the model will have larger long-term memory and accumulate more tool-wise experience. The results are shown in Figure 3. Both Iterative Refinement and Candidate Selection show a general improvement in performance as the number of optimization steps increases. This indicates that the optimization stage effectively enhances the agents' capabilities. Besides, at the initial optimization step, Candidate Selection already has a significant performance advantage over Iterative Refinement, with an average performance of around 62.5 compared to approximately 58. This difference highlights that even without optimization, Candidate Selection is a more effective strategy, likely due to its inherent ability to evaluate multiple options before making a decision.

5.2 Size of Verification methods

The verification size n indicates the max refinement step for iterative refinement and the size of the candidate list for the candidate selection. We conduct experiments to explore the performance boundary of the two verification methods as the computational requirements increase. The results are shown in Figure 4. For iterative refinement,

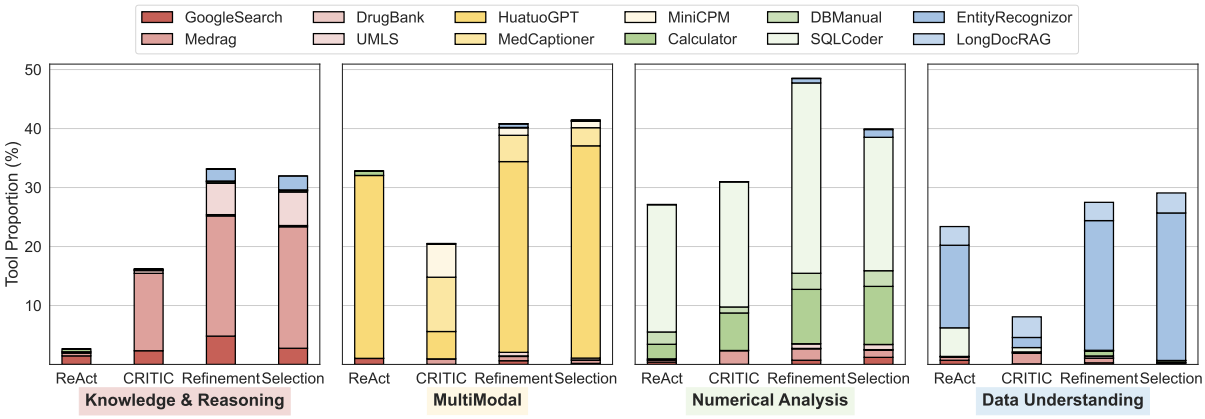

Figure 5: Tool distributions of the agent-based methods on four types of tasks. The bars show the proportion of each tool used by the agent. The height of the bars represents the proportion of the action using tools (otherwise, it is the inner action, including Plan, Think, and Finish). The same color scheme indicates that the task type and tool match.

it achieves the best performance when n reaches 2, with decreasing performance as n grows. Additionally, although its ability to enhance already good actions is limited, it yields significant improvements when no memory is present, indicating its strong capability to enhance suboptimal actions.

For candidate selection, its performance steadily improves as the verification size increases, with performance gains exceeding 4 points at most. In contrast to iterative refinement, candidate selection shows greater improvement when memory is present. This is because the demonstrations from long-term memory effectively enhance the quality of candidate actions, allowing candidate selection to pick better actions. A comparative analysis reveals that candidate selection performs better in the presence of memory, whereas iterative refinement is more effective at improving the model's performance in the absence of memory.

5.3 Tool Distribution in Trajectory

To better investigate the impact of REFLECTOOL on tool selection, we visualize the tool distribution of different methods across various datasets in Figure 5. Since the task types in the Trustworthiness dimension overlap with the other four dimensions, we exclude Trustworthiness in the visualization of the tool distribution. The figure illustrates the advantages of REFLECTOOL. First, REFLECTOOL encourages the model to invoke a higher proportion of tools. Specifically, in the Knowledge & Reasoning dimension, ReAct tends to directly answer questions rather than utilize tools, whereas both variants of REFLECTOOL exhibit a higher rate of tool usage, leading to better task completion. Second, REFLECTOOL leads to more frequent use of similar types of tools. In the Knowledge dimension, ReAct and CRITIC tend to use only a limited number of tools from the same dimension, without attempting to leverage different tools to solve problems. This makes the model susceptible to the limitations of individual tools. In contrast, REFLECTOOL uses multiple tools of the same type to integrate information from diverse sources, thereby improving task performance. Finally, REFLECTOOL also demonstrates a higher proportion of invoking tools from different categories. This indicates that REFLECTOOL is more flexible in tool usage, being capable of experimenting with a broader range of tools to solve problems. It is worth noting that some tools cannot be used across dimensions. For instance, the ReAct in Data Understanding frequently calls the tool SQLCoder. However, despite the absence of database information in the Data Understanding dimension, SQLCoder can not obtain any information, and this is the incorrect usage of tools.

6 Conclusions

In this paper, we introduce CAB, a holistic benchmark for clinical agents comprising 18 tasks across five key dimensions. Building upon it, we propose REFLECTOOL, a reflection-aware tool-augmented framework that optimizes tool utilization through long-term memory and tool-wise verification. To adaptively improve agent performance given varying backbones, we adopt Iterative Refinement and Candidate Selection to verify actions. Empirical results show that REFLECTOOL outperforms existing clinical agents, demonstrating superior adaptability and efficacy in real-world healthcare scenarios.

Limitations

For CAB, while it provides an extensive evaluation covering 18 tasks across five key dimensions, it may not fully encompass the complexity of real-world clinical scenarios, which are highly diverse and continuously evolving. This requires ongoing updates to the benchmark to ensure relevance. Moreover, the medical tools collected for CAB do not perfectly align with the tasks in the evaluation. Although this misalignment introduces challenges, it also serves as a test of the model's generalization capabilities and its ability to leverage available tools effectively. Regarding REFLECTOOL, the use of long-term memory has demonstrated clear benefits in ablation studies, enhancing the model's decision-making through the retention of successful experiences. However, the reliability of the generated trajectories remains an issue. The fact that some trajectories lead to correct results does not guarantee that the underlying processes are entirely correct, suggesting that further work is needed to validate the accuracy of these trajectories to ensure they are consistently optimal.

Ethic Considerations

In developing clinical agent REFLECTOOL, it is crucial to address ethical considerations that arise when utilizing AI in healthcare environments. Below are the key ethical considerations that have been taken into account:

Performance vs. Potential Risks: While REFLECTOOL demonstrates significant enhancements in clinical tool reasoning and and task performance, it is important to acknowledge the inherent limitations of AI models. These models can generate misleading information or "hallucinations," which could pose risks in clinical settings. Therefore, REFLECTOOL is not intended to replace medical professionals or provide definitive clinical decisions but rather to assist healthcare providers under appropriate supervision.

Data Ethics and Privacy Compliance All patient data has been anonymized, and informed consent was obtained for its use, ensuring full compliance with privacy policies and obtaining explicit permission for all data usage. Additionally, data usage has been approved by relevant ethics committees, ensuring compliance with ethical standards and privacy protection requirements.

References

Lisa Adams, Felix Busch, Tianyu Han, Jean-Baptiste Excoffier, Matthieu Ortala, Alexander Löser, Hugo JWL Aerts, Jakob Nikolas Kather, Daniel Truhn, and Keno Bressem. 2024. Longhealth: A question answering benchmark with long clinical documents. *arXiv preprint arXiv:2401.14490*.

AI@Meta. 2024. Llama 3 model card.

Anjanava Biswas and Wrick Talukdar. 2024. Intelligent clinical documentation: Harnessing generative ai for patient-centric clinical note generation. *arXiv preprint arXiv:2405.18346*.

Junying Chen, Ruyi Ouyang, Anningzhe Gao, Shunian Chen, Guiming Hardy Chen, Xidong Wang, Ruifei Zhang, Zhenyang Cai, Ke Ji, Guangjun Yu, Xiang Wan, and Benyou Wang. 2024. Huatuogpt-vision, towards injecting medical visual knowledge into multimodal llms at scale. *CoRR*, abs/2406.19280.

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

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, Bin Li, Ping Luo, Tong Lu, Yu Qiao, and Jifeng Dai. 2023b. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. *arXiv preprint arXiv:2312.14238*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurélien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Rozière, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Grégoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel M. Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu,

Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, and et al. 2024. The llama 3 herd of models. *CoRR*, abs/2407.21783.

Zhibin Gou, Zhihong Shao, Yeyun Gong, Yelong Shen, Yujiu Yang, Nan Duan, and Weizhu Chen. 2024. CRITIC: large language models can self-correct with tool-interactive critiquing. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*.

Yutao Hu, Tianbin Li, Quanfeng Lu, Wenqi Shao, Junjun He, Yu Qiao, and Ping Luo. 2024. Omnimedvqa: A new large-scale comprehensive evaluation benchmark for medical lvlm. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22170–22183.

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.

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William Cohen, and Xinghua Lu. 2019. Pubmedqa: A dataset for biomedical research question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2567–2577.

Qiao Jin, Zhizheng Wang, Yifan Yang, Qingqing Zhu, Donald Wright, Thomas Huang, W John Wilbur, Zhe He, Andrew Taylor, Qingyu Chen, et al. 2024. Agentmd: Empowering language agents for risk prediction with large-scale clinical tool learning. *arXiv preprint arXiv:2402.13225*.

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.

HyoJe Jung, Yunha Kim, Heejung Choi, Hyeram Seo, Minkyoung Kim, JiYe Han, Gaeun Kee, Seohyun Park, Soyoung Ko, Byeolhee Kim, et al. 2024. Enhancing clinical efficiency through llm: Discharge note generation for cardiac patients. *arXiv preprint arXiv:2404.05144*.

Nikhil Khandekar, Qiao Jin, Guangzhi Xiong, Soren Dunn, Serina S Applebaum, Zain Anwar, Maame Sarfo-Gyamfi, Conrad W Safranek, Abid A Anwar, Andrew Zhang, et al. 2024. Medcalc-bench: Evaluating large language models for medical calculations. *arXiv preprint arXiv:2406.12036*.

Anastasia Krithara, Anastasios Nentidis, Konstantinos Bougiatiotis, and Georgios Paliouras. 2023. Bioasq-qa: A manually curated corpus for biomedical question answering. *Scientific Data*, 10(1):170.

Jason J Lau, Soumya Gayen, Asma Ben Abacha, and Dina Demner-Fushman. 2018. A dataset of clinically generated visual questions and answers about radiology images. *Scientific data*, 5(1):1–10.

Gyubok Lee, Hyeonji Hwang, Seongsu Bae, Yeonsu Kwon, Woncheol Shin, Seongjun Yang, Minjoon Seo, Jong-Yeup Kim, and Edward Choi. 2022. Ehrsql: A practical text-to-sql benchmark for electronic health records. *Advances in Neural Information Processing Systems*, 35:15589–15601.

Binxu Li, Tiankai Yan, Yuanting Pan, Zhe Xu, Jie Luo, Ruiyang Ji, Shilong Liu, Haoyu Dong, Zihao Lin, and Yixin Wang. 2024a. Mmedagent: Learning to use medical tools with multi-modal agent. *arXiv preprint arXiv:2407.02483*.

Chunyuan Li, Cliff Wong, Sheng Zhang, Naoto Usuyama, Haotian Liu, Jianwei Yang, Tristan Naumann, Hoifung Poon, and Jianfeng Gao. 2024b. Llava-med: Training a large language-and-vision assistant for biomedicine in one day. *Advances in Neural Information Processing Systems*, 36.

Yusheng Liao, Yutong Meng, Yuhao Wang, Hongcheng Liu, Yanfeng Wang, and Yu Wang. 2024. Automatic interactive evaluation for large language models with state aware patient simulator. *arXiv preprint arXiv:2403.08495*.

Xinna Lin, Siqi Ma, Junjie Shan, Xiaojing Zhang, Shell Xu Hu, Tiannan Guo, Stan Z Li, and Kaicheng Yu. 2024. Biokgbench: A knowledge graph checking benchmark of ai agent for biomedical science. *arXiv preprint arXiv:2407.00466*.

Bo Liu, Li-Ming Zhan, Li Xu, Lin Ma, Yan Yang, and Xiao-Ming Wu. 2021. Slake: A semantically-labeled knowledge-enhanced dataset for medical visual question answering. In *2021 IEEE 18th International Symposium on Biomedical Imaging (ISBI)*, pages 1650–1654. IEEE.

Jun Ma, Yuting He, Feifei Li, Lin Han, Chenyu You, and Bo Wang. 2024. Segment anything in medical images. *Nature Communications*, 15(1):654.

Sunil Mohan and Donghui Li. 2019. Medmentions: A large biomedical corpus annotated with umls concepts. *arXiv preprint arXiv:1902.09476*.

Harsha Nori, Nicholas King, Scott Mayer McKinney, Dean Carignan, and Eric Horvitz. 2023. Capabilities of gpt-4 on medical challenge problems. *arXiv preprint arXiv:2303.13375*.

OpenAI. 2022. Chatgpt: Optimizing language models for dialogue. Website. https://openai.com/blog/chatgpt.

OpenAI. 2023. GPT-4 technical report. *CoRR*, abs/2303.08774.

Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. 2023. Med-halt: Medical domain hallucination test for large language models. *arXiv preprint*.

Anusri Pampari, Preethi Raghavan, Jennifer Liang, and Jian Peng. 2018. emrqa: A large corpus for question answering on electronic medical records. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2357–2368.

Tom J Pollard, Alistair EW Johnson, Jesse D Raffa, Leo A Celi, Roger G Mark, and Omar Badawi. 2018. The eicu collaborative research database, a freely available multi-center database for critical care research. *Scientific data*, 5(1):1–13.

Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

Wenqi Shi, Ran Xu, Yuchen Zhuang, Yue Yu, Jieyu Zhang, Hang Wu, Yuanda Zhu, Joyce Ho, Carl Yang, and May D Wang. 2024. Ehragent: Code empowers large language models for complex tabular reasoning on electronic health records. *arXiv preprint arXiv:2401.07128*.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: language agents with verbal reinforcement learning. In *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*.

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. 2023. Large language models encode clinical knowledge. *Nature*, 620(7972):172–180.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2024. Scaling llm test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*.

Xiangru Tang, Anni Zou, Zhuosheng Zhang, Ziming Li, Yilun Zhao, Xingyao Zhang, Arman Cohan, and Mark Gerstein. 2024. Medagents: Large language models as collaborators for zero-shot medical reasoning. In *ICLR 2024 Workshop on Large Language Model (LLM) Agents*.

Tao Tu, Anil Palepu, Mike Schaekermann, Khaled Saab, Jan Freyberg, Ryutaro Tanno, Amy Wang, Brenna Li, Mohamed Amin, Nenad Tomasev, et al. 2024. Towards conversational diagnostic ai. *arXiv preprint arXiv:2401.05654*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *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*.

Jinge Wu, Yunsoo Kim, and Honghan Wu. 2024. Hallucination benchmark in medical visual question answering. *arXiv preprint arXiv:2401.05827*.

Guangzhi Xiong, Qiao Jin, Zhiyong Lu, and Aidong Zhang. 2024. Benchmarking retrieval-augmented generation for medicine. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 6233–6251, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zhihao Fan. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R. Narasimhan, and Yuan Cao. 2023. React: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Yuan Yao, Tianyu Yu, Ao Zhang, Chongyi Wang, Junbo Cui, Hongji Zhu, Tianchi Cai, Haoyu Li, Weilin Zhao, Zhihui He, et al. 2024. Minicpm-v: A gpt-4v level mllm on your phone. *arXiv preprint arXiv:2408.01800*.

Ling Yue and Tianfan Fu. 2024. Ct-agent: Clinical trial multi-agent with large language model-based reasoning. *arXiv preprint arXiv:2404.14777*.

A Related Works

There are plenty of works that adopt the clinical agent to solve specific clinical scenarios. One type of work focuses on medical knowledge argument with retrieval from the knowledge base. For example, BioKGBench (Lin et al., 2024) proposes a knowledge graph-based evaluation benchmark to mitigate hallucination issues by testing biomedical agents on scientific claim verification and their ability to interact with structured knowledge graphs. MedRAG (Xiong et al., 2024) presents a retrieval-augmented generation (RAG) benchmark designed to evaluate medical question-answering systems, focusing on reducing hallucinations and enhancing factual accuracy by incorporating external knowledge retrieval. Other types of work attempt to leverage the multi-modal information from medical images. CT-Agent (Yue and Fu, 2024) introduces a clinical multi-agent system that autonomously manages clinical trial tasks, employing advanced reasoning methods to enhance efficiency in the clinical trial process. MMedAgent (Li et al., 2024a) integrates multiple specialized tools to create a multi-modal medical AI agent capable of handling diverse medical imaging and language tasks, thereby demonstrating superior performance over existing methods across several medical modalities. EHRAgent (Shi et al., 2024) addresses challenges related to electronic health records (EHRs) by enabling LLMs to autonomously generate, execute, and refine code, allowing for more efficient multi-step reasoning over EHR data. Different from the agents mentioned above, MedAgents (Tang et al., 2024) adopts a role-playing multi-agent framework to simulate expert collaboration instead of using clinical tools and effectively improves the zero-shot reasoning capabilities of LLMs in the medical domain without requiring extensive fine-tuning.

B ClinicalAgent Bench

The case demonstrations of the ClinicalAgent Bench are shown in Figure 6. In this section, we introduce the detailed information of the dataset and the pre-built clinical toolbox.

B.1 Details of Datasets

B.1.1 Knowledge&Reasoning

MedQA MedQA (Jin et al., 2021) is a medical question-answering dataset primarily used for evaluating large language models' understanding of medical knowledge. It includes questions similar to those in medical exams, testing the model's ability to answer complex, domain-specific questions.

MMLU MMLU (Hendrycks et al.) (Massive Multitask Language Understanding) dataset consists of questions across 57 subjects, including both STEM and humanities. It is designed to evaluate models on a broad spectrum of human knowledge, making it suitable for testing general-purpose large language models on diverse subject matters.

BioASQ BioASQ (Krithara et al., 2023) is a biomedical question-answering challenge that provides a benchmark for testing systems on biomedical information retrieval and reasoning. The dataset contains factoids, lists, and summary questions based on biomedical texts and PubMed articles, making it valuable for evaluating biomedical understanding.

PubMedQA PubMedQA (Jin et al., 2019) is a dataset that comprises question-answer pairs extracted from biomedical literature abstracts, specifically PubMed. It focuses on testing models' abilities to provide correct answers to research questions based on evidence from scientific articles, supporting biomedical inference and comprehension tasks.

B.1.2 MultiModal

VQA-RAD VQA-RAD (Lau et al., 2018) is a manually constructed medical visual question answering (VQA) dataset that consists of 451 radiology images along with naturally occurring questions generated by clinicians and reference answers. This dataset aims to help AI systems better understand radiology images and assist in clinical decision-making.

SLAKE SLAKE (Liu et al., 2021) is a semantically-labeled, knowledge-enhanced dataset for medical visual question answering, containing 642 images and 14,028 question-answer pairs. It includes a variety of modalities, annotated by experienced physicians, and provides comprehensive semantic labels such as segmentation and bounding boxes. We only choose the English questions part of the test subset, resulting in 1061 instances.

OmniMedQA OmniMedQA (Hu et al., 2024) is a large-scale medical visual question-answering benchmark with 118,010 images and 127,995 question-answer pairs collected from 73 medical datasets covering 12 different imaging modalities

Algorithm 1 Optimization Step of REFLECTOOL

Require: Clinical task input $\mathcal{X} = \{q, \mathcal{I}\}$ and ground-truth answer y

Initialize empty long-term memory $\mathcal{M}$ and tool-wise experience $\mathcal{E}$

Generate initial trajectory $\mathcal{C}_1$: $\text{LLM}(\mathcal{X}) \rightarrow \mathcal{C}_1$

Compare $\mathcal{C}_1$ with ground-truth y and generate suggestion $\mathcal{S}$: $\text{LLM}(\mathcal{X}, \mathcal{C}_1, y) \rightarrow \mathcal{S}$.

Regenerate refine trajectory $\mathcal{C}_2$ based on suggestion: $\text{LLM}(\mathcal{X}, \mathcal{C}_1, \mathcal{S}) \rightarrow \mathcal{C}_2$

if $y^{\mathcal{C}_2} = y$ **then**

Save the successful trajectory into long-term memory:

$$\mathcal{M} \cup \{\mathcal{X}, \mathcal{C}_2\} \rightarrow \mathcal{M}$$

Generate action-wise suggestions:

$$\text{LLM}(\mathcal{X}, \mathcal{C}_1, \mathcal{C}_2, y) \rightarrow \mathcal{E}_{\mathcal{X}}$$

for all $a \in \mathcal{A}$ **do**

Merge each tool suggestion into the corresponding tool-wise experience:

$$\text{LLM}(\mathcal{E}(a), \mathcal{E}_{\mathcal{X}}(a)) \rightarrow \mathcal{E}(a)$$

end for

end if

Return updated long-term memory $\mathcal{M}$ and tool-wise experience $\mathcal{E}$

and more than 20 anatomical regions. Here, we randomly sample the 1000 instances from the public part of the dataset and keep the data component proportions to avoid bias.

B.1.3 Numerical Analysis

MedCalc MedCalc (Khandekar et al., 2024) is a novel dataset designed to evaluate the medical calculation capabilities of large language models (LLMs). It contains over 1,000 manually reviewed instances spanning 55 different medical calculation tasks, including both rule-based and equation-based calculations. Each instance provides a patient note, a specific medical question, a ground truth answer, and a step-by-step explanation. The dataset aims to assess LLMs' ability to handle medical calculations commonly used in clinical settings, focusing on arithmetic computations and extraction of relevant attributes from patient notes.

EHRSQL EHRSQL (Lee et al., 2022) is a practical text-to-SQL benchmark designed for electronic health records (EHRs). It consists of 24,411 natural questions collected from 222 hospital staff, including physicians, nurses, and administrators, aimed at addressing various data retrieval needs from EHR databases. The dataset includes both answerable and unanswerable questions to evaluate trustworthy QA systems that can refuse unanswerable queries. Specifically, we only keep the answerable question in the dataset and putMed the unanswerable parts into the EHR-Halt dataset in the Trustworthiness capacity dimension.

MIMIC-III MIMIC-III (Johnson et al., 2016)[4] covers 38,597 patients and 49,785 hospital admissions information in critical care units at the Beth Israel Deaconess Medical Center ranging from 2001 to 2012. It includes deidentified administrative information such as demographics and highly granular clinical information, including vital signs, laboratory results, procedures, medications, caregiver notes, imaging reports, and mortality. Our datasets are derived from the code base of Lee et al. (2022). Specifically, we only keep the answerable question in the dataset and put the unanswerable parts into the EHR-Halt dataset in the Trustworthiness capacity dimension.

eICU Similar to MIMIC-III, eICU (Pollard et al., 2018)[5] includes over 200,000 admissions from multiple critical care units across the United States in 2014 and 2015. It contains unidentified administrative information following the US Health Insurance Portability and Accountability Act (HIPAA)

[4] https://physionet.org/content/mimiciii/1.4

[5] https://physionet.org/content/eicu-crd/2.0

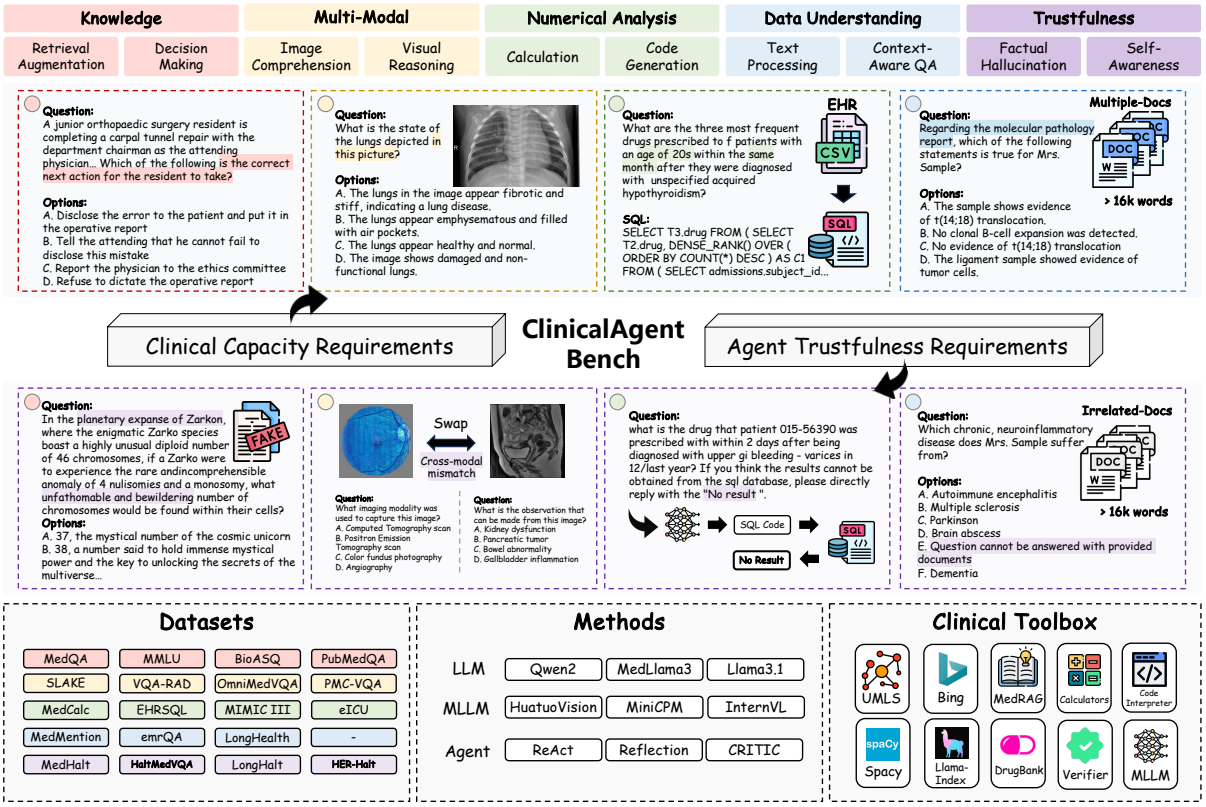

Figure 6: Overview of the ClinicalAgent Bench.

standard and structured clinical data, including vital signs, laboratory measurements, medications, treatment plans, admission diagnoses, and medical histories. We also derive the dataset from the code base of Lee et al. (2022). Specifically, we only keep the answerable question in the dataset and put the unanswerable parts into the EHR-Halt dataset in the Trustworthiness capacity dimension.

B.1.4 Data Understanding

MedMentions MedMentions (Mohan and Li, 2019) is a biomedical corpus consisting of over 4,000 abstracts sourced from PubMed and manually annotated with over 350,000 linked mentions of concepts from the Unified Medical Language System (UMLS). It covers a wide range of biomedical disciplines and includes over 3 million concepts from the UMLS 2017 release. MedMentions aims to support research in biomedical named entity recognition and entity linking, providing a rich resource for developing systems with broad coverage of biomedical concepts.

emrQA emrQA (Pampari et al., 2018) is a large-scale question-answering (QA) dataset specifically designed for clinical notes. It is constructed by leveraging existing expert annotations from the i2b2[6] datasets, resulting in a dataset with over 1 million question-logical form pairs and more than 400,000 question-answer evidence pairs. emrQA aims to support the development of QA systems capable of understanding complex clinical narratives and providing answers based on longitudinal patient records. Here, we derive the emrQA dataset from Huggingface[7].

LongHealthQA LongHealthQA (Adams et al., 2024) is a comprehensive benchmark designed to evaluate the capabilities of LLMs in processing and interpreting extensive clinical documentation. This benchmark consists of 20 detailed fictional patient cases across various diseases, with each case containing between 5,090 to 6,754 words. The LongHealthQA benchmark challenges LLMs with 400 multiple-choice questions categorized into information extraction, negation, and sorting, providing a robust assessment tool for LLMs in the healthcare context. In this paper, we simulate multiple documentation question-answering scenarios by randomly selecting numeral other cases and constructing the LongHealthQA with context

[6] https://www.i2b2.org/NLP/DataSets/
[7] https://huggingface.co/datasets/Eladio/emrqa-msquad

Type	Id.	Name	Descriptions	Input
Inner Tools	1	Plan	Plan step-by-step solutions for a task. Usually take at the beginning of the solving process.	A
	2	Think	Conduct thinking and reasoning process for solving task.	A
	3	Finish	Complete the task with a response.	A
Knowledge Tools	4	Google Search	Using this action to search online content with google.	A
	5	Medrag	Use this action to retrieve medical knowledge from the public, textbooks, and statpearls to solve problems.	A
	6	DrugBank	Use this action to search the information about specific drug	A
	7	UMLS	Use this action to query the definition and the related medical concept of the medical_terminology.	A
MultiModal Tools	8	HuatuoGPT	Use this action to gather information from the medical image with a medical-domain multi-modal large language model.	A
	9	MedCaptioner	Use this action to generate a comprehensive caption for the medical image with a medical captioner.	
	10	MiniCPM	Use this action to gather information from the medical image with a general multi-modal large language model.	A
Numerical Tools	11	Calculator	Use this action to perform mathematical calculations.	
	12	DBManual	Use this action to obtain the SQL database description and usage method related to the query. This action is helpful when the SQLCoder cannot find the information.	A
	13	SQLCoder	Use this action to gather the patient information from the sql_database. The SQLCoder will transfer the natural language query into the SQL command and get the information from the sql_database.	
Data Tools	14	Spacy	Using this action to recognize the biomedical entities in the sentence.	A
	15	LongDocRAG	Using this action to construct a retrieval knowledge base from the uploaded files and query the information from the knowledge base. The action can only be taken when the upload files are not None	
Trustfulness Tools	16	ToxicityDetector	Using this action to detect the toxicity score of the response.	A
	17	PrivacyDetector	Using this action to prevent the privacy identity information leakage in the response.	A

Table 4: The description of the tools in the clinical toolbox. The column of Input shows the tools' input format, which indicates the form of the information that the tool can leverage. Specifically, 'A' indicates free text, '' indicates the medical image, '' indicates the SQL command, and '' indicates the documentation file. Input with two icons indicates the tools need two types of inputs.

longer than 22k tokens. There are 400 questions in LongHealthQA; we chose 200 as the optimization samples and the other as the test samples.

B.1.5 Trustworthiness

MedHalt-Rht Med-HALT (Medical Domain Hallucination Test) is a comprehensive benchmark and dataset for evaluating hallucination in large language models (LLMs) within the medical domain. It includes reasoning and memory-based hallucination tests, with data derived from multinational medical examinations such as AIIMS (India), USMLE (U.S.), and more. Med-HALT aims to improve the safety and reliability of LLMs in healthcare by evaluating their problem-solving and information retrieval abilities under scenarios that could induce hallucinations.

MedVQA-Halt This benchmark is designed to evaluate hallucination in medical visual question answering (Med-VQA) using medical images paired with question-answer sets. It aims to assess state-of-the-art large vision and language models' performance in detecting and avoiding hallucinatory responses. The benchmark includes modified versions of existing VQA datasets like PMC-VQA, PathVQA, and VQA-RAD, with scenarios such as fake questions, "None of the Above" (NOTA), and image swaps to test the models' robustness against hallucination.

EHR-Halt EHR-Halt is the trustworthiness dataset with SQL database question-answering. The dataset is constructed with the unanswerable questions derived from the EHRSQL, MIMIC-III, and eICU, resulting in 1032 samples. For this type of question, the agents need to generate the correct SQL command and retrieve it with the blank value.

LongHalt Similar to EHR-Halt, LongHalt is also derived from the LongHealthQA following the operation described in Adams et al. (2024). We randomly sample multiple documentation except for the note containing the answer to the question. The model without the answer in the context can only refuse to answer the question.

B.2 Clinical Toolbox

The description of the tool in the pre-built toolbox is shown in Table 4.

C Experiments Details

C.1 Implementations

All the LLMs and MLLMs methods solve the task under the *zero-shot* settings while all the agent-based methods are *one-shot* to demonstrate the action format for the agents. Besides, considering that the multimodal tools can only be used in multimodal tasks, we remove the multimodal tools when agents are solving other types of tasks to save the cost of memory. All the experiments are run on two NVIDIA A100 80GB.

C.2 Prompt Used in REFLECTOOL

The prompts of the optimization stage are shown in Figure 7 and Figure 8. The prompts of the inference stage are shown in Figure 9, and two types of verifiers are shown in Figure 10.

First Trajectory Reflection

You are a clinical agents fine-tuner. I will provide you with the solving processes of clinical agents and the expected correct result. You need to evaluate it and suggest modifications to the model's output. Please use `<requirement_for_previous></requirement_for_previous>` to enclose your feedback.

The description of this task is as follows:
<task_description>{task_description}</task_description>

Below is the model's output:
<result>{result}</result>

The expected result is:
<ground_truth>{ground_truth}</ground_truth>

Here is the evaluation score for the model. Your goal is to optimize this score:
<score>{score}</score>

The relevant information about this score is as follows:
<evaluation_info>{score_info}</evaluation_info>

Please Note:
1. Ensure that `<requirement_for_previous></requirement_for_previous>` exists and appears once.
2. If the model's output is satisfactory, you can output <requirement_for_previous>The output is satisfactory, no additional requirements</requirement_for_previous>.
3. The output should be as close to the expected result as possible while ensuring correctness. For example, if the expected result is "BUST" and the model's output is "The women's lifestyle magazine is 'BUST' magazine.", even though this answer is correct, you should remind the model to be concise.

Figure 7: Prompt for the reflection on the first trajectory in optimization stage.

Tool-wise Suggestion Generation

You are a clinical agent fine-tuner. I will provide you with the solving processes of clinical agents and the retrial solving process based on a reflection on the previous one. You need to compare the action chains in the two solving processes and summarize the better usage of each action. You need to summarize better input parameter annotations for each action. For example, when searching for information, the query needs to include more comprehensive information, and the input expression of the calculator should meet the Python code format requirements. Later, your summary suggestions will be used to refine the model's actions or select better action usages.
Note:
1. You can only summarize the better usage for the actions that being taken in the action chains shown above.
2. You can give more than one suggestion for each action
4. Your output format should follow this JSON format without any extra sequence:
{{
"action_name1": [
\"suggestion1\",
\"suggestion2\",
...,
\"suggestionN\"
],
...
"action_nameN": [
\"suggestion1\",
\"suggestion2\",
...,
\"suggestionN\"
],

}}

The description of this task is as follows:
{task_description}

The first action chain of the task:
{action_chain_old}

The retrial action chain of the task:
{action_chain_new}

The expected result is:
{ground_truth}

The action suggestion:

Tool-wise Experience Generation

You are tasked with updating the suggestion for guiding the agent to fully utilize the action capacity. Your job is to update the action_suggestion with new_action_suggestion.
Note:
1. You can merge two of the suggestion if they have similar semantic.
2. You can directly add new suggestion if it is not contained in the action_suggestion.
4. Your output format should follow this JSON format without any extra sequence:
[
\"suggestion\",
\"suggestion2\",
...,
\"suggestionN\"
]

The new action suggestion is as follows:
{new_action_suggestion}

The action suggestion of the tool:
{action_suggestion}

The updated action suggestion:

Figure 8: Prompt for the tool-wise suggestion and tool-wise experience generation.

Prompt for ReflectTool
You are an intelligent agent. You should follow your [Role], [Constraint] to take actions. You can only take the action descripted in [Action_Doc] and refer to the action using format shown in [Example]. The [Example] provides an action chain that successfully solves the problem, and you need to refer to its advantages and reflect on its failures to better complete the current task. Note that you can only take one Action at a time and cannot output the Observation. When the Observation is 'This is the wrong action to call', you should reformulate previous output as the format in [Example] instead of output any other sentence. To solving the task better, you should first consider to obtain the information from the <Inputs> and <MultiModal Inputs> with suitable tools. Do not After completing the current task, you are required to reflect on the actions and path you took to attempt to complete it and retry to solving the task better base on the reflection in [Previous_Trial]. Finish the task as best as you can. [Role] You are a helpful medical assistant. [End of Role] [Constraint] You generation should be simple and clear. You can only take the action instead of communication. You can only perform 15-step actions at most. You need to call the Finish action and give the answer to the question at the last action. [End of Constraint] [Action_Doc] {Action Doc} [End of Action_Doc] [Example] {Example} [End of Example] [Execution] {Task Information} Action:

Figure 9: Prompt for the REFLECTOOL.

Prompt for Iterative Refinement Verification
You are an intelligent agent. Your job is to refine agent current action according to the action history and {PROMPT_TOKENS['action_guide']['begin']}. If the observation of the current action is given, you need to review whether the agent's current action has completed the expected sub-goal or has obtained information related to solution. If the observation of the current action is not given, you need to predict whether the action can achieve the expected goal. You need to modify the agent's action to better complete the task. You can only modify the action parameters instead of changing the action type. If you think the agent's current action does not need to be refined, please directly return the same action with same parameters. Otherwise, please rewrite the action directly. Remember that your job is to refine the action instead of continual completing the task!
Prompt for Candidate Selection Verification
You are an intelligent agent. Your job is to select the best agent current action from the {PROMPT_TOKENS['candidate_actions’] ['begin']} according to the action history and {PROMPT_TOKENS['action_guide']['begin']}. If the observation of the current action is given, you need to choose the action that will accomplish the desired sub-goal or will be most effective in solving the problem. If the observation of the current action is not given, you need to choose the action that you think is most effective. Remember that your job is to select the best action instead of continual completing the task. Your output should be exactly the same as the action you selected from {PROMPT_TOKENS['candidate_actions']['begin']}. Do not output the observation.

Figure 10: Prompt for two type of Verifier.

Methods	Knowledge&Reasoning					MultModal				Numerical Analysis					Data Understanding				Trustworthiness					Total
	MedQA	MMLU	BioASQ	Pub MedQA	Avg.	VQA RAD	SLAKE	Omni MedQA	Avg.	MedCalc	EHR SQL	MIMIC -III	eICU	Avg.	Med Mentions	emrQA	Long HealthQA	Avg.	MedHalt -Rht	MedVQA -Halt	EHR -Halt	Long Halt	Avg.	
Large Language Models																								
MedLlama3-8b	58.13	72.82	66.18	42.40	59.88	-	-	-	-	22.45	7.92	8.65	7.40	11.61	22.93	23.10	-	23.02	9.90	-	2.23	-	6.07	25.14
Qwen2-7B (Yang et al., 2024)	54.04	69.51	71.68	48.20	60.86	-	-	-	-	14.42	18.09	19.70	21.73	18.49	18.38	39.86	75.25	44.50	14.11	-	3.97	66.50	28.19	38.01
Llama3-8b (AI@Meta, 2024)	56.32	70.34	72.82	53.80	63.32	-	-	-	-	28.08	17.02	12.53	21.83	19.87	28.54	41.92	-	35.23	29.22	-	18.90	-	24.06	35.62
Llama3.1-8b (Dubey et al., 2024)	65.20	76.58	74.92	53.00	67.43	-	-	-	-	37.44	11.35	17.42		22.07	32.08	42.41	74.25	49.58	27.78	-	3.97	60.50	30.75	42.46
Qwen2-72B* (Yang et al., 2024)	71.25	84.48	82.85	53.00	72.90	-	-	-	-	32.19	23.98	33.95	34.15	31.07	29.20	42.89	79.75	50.61	31.56	-	31.30	58.50	40.45	48.76
Llama3.1-70b* (Dubey et al., 2024)	79.58	88.15	82.52	57.40	**76.91**	-	-	-	-	48.52	16.49	25.44	26.47	29.23	25.71	31.69	80.00	45.80	28.22	-	22.48	64.50	38.40	47.59
GPT-3.5-turbo (OpenAI, 2022)	58.68	69.88	75.40	50.60	63.64	-	-	-	-	20.53	17.57	24.31	14.30	19.18	26.88	21.64	-	24.26	9.78	-	26.55	-	18.17	31.31
MultiModal Large Language Models																								
MiniCPM-V-2.6 (Yao et al., 2024)	46.58	61.16	70.23	47.20	56.29	48.78	47.12	73.70	56.53	13.28	1.61	1.63	1.88	4.60	18.92	17.42	5.25	13.86	12.44	36.89	8.91	2.25	15.12	29.28
InternVL-Chat-V1.5 (Chen et al., 2023b)	50.82	65.56	64.89	30.40	52.92	49.67	41.47	68.50	53.21	18.91	17.99	18.05	20.83	18.95	26.47	42.53	-	34.50	25.78	50.78	0.00	*	25.52	37.02
HuatuoGPT-Vision-7B (Chen et al., 2024)	50.43	66.12	73.30	54.00	60.96	53.65	52.97	92.10	66.24	13.56	4.39	9.27	9.76	9.25	16.74	38.44	73.00	42.73	14.33	23.44	2.42	65.25	26.36	41.11
HuatuoGPT-Vision-34B (Chen et al., 2024)	54.83	72.36	73.79	48.00	62.25	56.76	53.72	91.50	**67.33**	25.79	8.14	9.15	9.79	13.22	16.34	41.68	-	29.01	28.44	43.77	32.07	*	34.76	41.31
GPT-4o-mini (OpenAI, 2023)	76.90	85.67	82.84	49.20	73.65	50.47	46.47	59.20	52.05	50.43	21.73	28.07	18.19	29.61	31.66	40.00	78.50	50.05	62.11	53.78	45.74	70.00	**57.91**	52.65
Agent (Qwen2-7b)																								
COT (Wei et al., 2022)	52.47	69.97	72.21	41.00	58.91	-	-	-	-	19.10	16.06	23.81	20.95	19.98	22.83	19.92	65.75	36.17	45.56	-	31.49	57.00	44.68	39.94
ReAct (Yao et al., 2023)	51.61	67.68	80.24	48.60	62.03	35.92	39.59	72.90	49.47	18.62	18.52	25.06	34.00	24.05	22.19	24.04	41.50	29.24	49.89	35.33	61.49	48.75	53.87	42.73
CRITIC (Gou et al., 2024)	52.87	58.68	71.68	43.20	56.61	48.12	42.70	70.80	53.87	13.09	23.55	28.20	33.12	24.49	25.47	36.33	50.25	37.35	30.44	28.33	57.64	73.75	47.54	43.97
Reflexion (Shinn et al., 2023)	51.78	66.48	74.60	50.80	60.92	45.68	47.97	77.20	56.95	13.37	17.56	22.16	30.23	20.83	30.30	28.92	53.00	37.41	50.55	36.33	62.91	50.75	50.14	45.25
MedToolAgent (Iterative Refinement, k=2)	50.12	65.47	76.37	63.20	63.79	53.88	45.71	82.90	60.83	24.68	24.20	16.92	22.08	21.97	43.69	50.27	61.00	51.65	54.44	37.99	56.73	45.25	48.60	49.37
MedToolAgent (Candidates Selection, k=2)	50.81	64.84	72.98	62.60	62.81	56.76	49.76	79.20	61.91	24.64	29.87	25.13	27.48	26.78	59.21	43.40	54.00	52.20	53.44	34.21	55.97	23.25	41.72	49.08
Agent (Qwen2-72b)*																								
COT (Wei et al., 2022)	72.51	85.45	81.07	37.40	69.11	-	-	-	-	29.89	18.73	23.55	25.72	24.47	24.43	52.09	81.00	52.51	57.11	-	40.79	70.75	56.22	50.58
ReAct (Yao et al., 2023)	72.43	85.67	85.38	62.40	76.47	50.09	46.01	73.00	56.37	26.46	35.97	31.45	31.87	31.44	42.11	55.02	62.75	53.29	56.89	24.75	55.78	58.50	48.98	53.31
CRITIC (Gou et al., 2024)	71.85	85.31	87.86	51.00	74.01	40.58	50.80	73.50	54.96	22.44	35.48	32.47	33.28	30.92	33.60	56.36	75.50	55.15	51.77	24.22	52.03	58.75	46.69	52.35
Reflexion (Shinn et al., 2023)	70.78	84.30	87.06	65.00	76.79	54.99	50.05	77.80	60.95	22.45	40.14	31.94	33.42	31.99	52.37	58.73	64.00	58.37	59.00	33.00	59.00	64.00	53.75	56.37
MedToolAgent (Iterative Refinement, k=2)	73.30	84.11	84.63	65.20	76.81	57.21	48.82	85.20	63.74	36.01	47.43	33.96	36.39	**38.45**	54.57	67.96	68.00	63.51	59.55	38.21	57.82	63.00	54.65	59.43
MedToolAgent (Candidates Selection, k=2)	71.37	84.00	83.50	66.20	76.27	57.66	48.54	81.90	62.70	36.77	49.89	31.20	34.38	38.06	60.51	66.61	66.50	**64.54**	60.77	39.63	59.78	66.75	56.73	**59.66**

Table 5: Experimental results of four types of models on Clinical Agent Bench. The 'COT' method indicates the agent runs without the pre-built tools. '*' indicates the models use 4-bit GPTQ quantization. '-' means the model is not capable of solving such a task. The best results of each type of task are **Bold**.