MedExRAG: Integrated Exploratory Reasoning for Medical Retrieval-Augmented Generation

Anonymous ACL submission

Abstract

Retrieval-augmented generation (RAG) has shown promising performance in medical question answering by incorporating external knowledge into language model reasoning. Recent studies further enhance RAG with agentic reasoning and Monte Carlo Tree Search (MCTS), enabling models to explore multiple reasoning and retrieval actions before producing an answer. However, existing MCTS-based RAG methods usually rely on a single high-scoring trajectory or aggregate only final predictions across different paths, which may be insufficient for medical problems that require comprehensive evidence from multiple perspectives. In this paper, we propose **MedExRAG**, a framework for Integrated **Med**ical **Ex**ploratory **R**etrieval-**A**ugmented **G**eneration. Instead of selecting the final answer from an individual reasoning path, our method constructs an exploration memory that summarizes and updates the exploration results of the whole search tree. This memory provides a compact representation of diverse reasoning directions, retrieved evidence, and intermediate analyses, allowing the model to generate answers with broader and more reliable contextual support. To guide the search process, we further train a reward model to evaluate simulated reasoning states and improve the balance between exploration and exploitation. Experiments show that MedExRAG outperforms standard RAG and MCTS-based RAG baselines, demonstrating the effectiveness of exploration memory for complex medical reasoning. All related resources have been made publicly available[1].

1 Introduction

Retrieval-augmented generation (RAG) has shown strong potential in mitigating the knowledge limitations of LLMs, especially in domain-specific and knowledge-intensive scenarios such as medical question answering (Xiong et al., 2024a; Izacard and Grave, 2021). By retrieving external evidence and incorporating it into generation, RAG enables LLMs to produce more factual and knowledge-grounded responses (Guu et al., 2020; Izacard et al., 2022). However, as medical questions become more complex, a single retrieval step is often insufficient to collect all necessary evidence (Yue et al., 2025; Trivedi et al., 2023; Press et al., 2023). Many medical problems require multi-step reasoning, evidence comparison, and the integration of information from multiple clinical perspectives (Singhal et al., 2025; Nori et al., 2023). Therefore, improving RAG for complex medical questions requires effectively combining retrieval with the reasoning and analytical capabilities of LLMs.

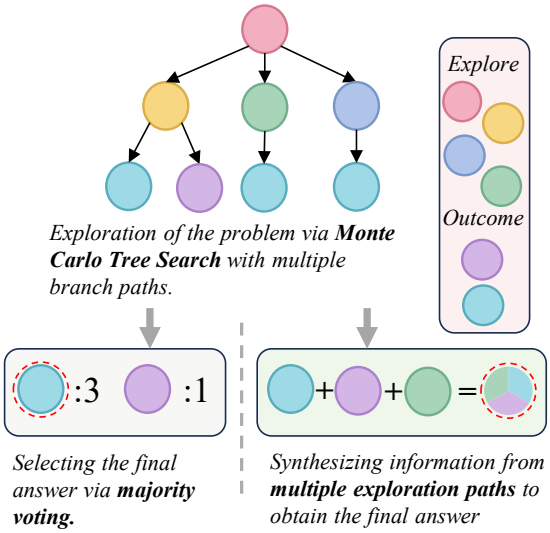

Figure 1: Difference between majority voting over final outcomes and integrating evidence across multiple exploration paths.

To address complex knowledge-intensive problems, prior studies have explored iterative retrieval, multi-step reasoning, reflection, and agentic workflows (Trivedi et al., 2023; Shinn et al., 2023; Li et al., 2025). For example, Yue et al. (2025) improves RAG through multi-round retrieval and rea-

[1] https://github.com/anonymous-AC/MedExRAG

soning, while ReAct (Yao et al., 2022), Reflexion (Shinn et al., 2023), and Search-o1 (Li et al., 2025) enable LLMs to interact with external tools or refine their reasoning based on feedback. Despite their effectiveness, these methods usually follow a chain-like reasoning paradigm, where the model explores the problem along a single trajectory. Such a paradigm is vulnerable to low-quality intermediate steps and may easily become trapped in a narrow solution space.

To overcome this limitation, recent studies have introduced Monte Carlo Tree Search (MCTS) (Browne et al., 2012) into LLM reasoning. rStar (Qi et al., 2025) demonstrates that MCTS can enhance LLM reasoning by balancing exploration and exploitation. Following this direction, RAG-Star (Jiang et al., 2025), RARE (Tran et al., 2025), and MCTS-RAG (Hu et al., 2025) extend MCTS-based reasoning to RAG by incorporating retrieval-related actions, while AirRAG (Feng et al., 2025) further combines reward modeling and self-consistency. However, these methods are still largely inherited from the rStar-style framework: the final answer is typically selected from a single high-scoring path or determined by majority voting over sampled outputs. As a result, although multiple paths are explored during search, the global information contained in the entire search tree remains underutilized.

This limitation is particularly important in medical RAG. For problems that can be solved through linear multi-hop reasoning, selecting a single strong reasoning path may be sufficient. However, complex medical questions often require comprehensive analysis from multiple aspects, such as symptoms, diagnosis, treatment, contraindications, pathophysiology, and evidence consistency. In such cases, useful information may be distributed across different branches of the exploration tree rather than concentrated in one individual path (Besta et al., 2024). Directly feeding the entire MCTS tree to the model is also impractical, since the resulting context can be overly long, noisy, and difficult to analyze effectively (Liu et al., 2024).

As illustrated in Figure 2, we propose MedExRAG, an integrated exploratory reasoning framework for medical RAG. Instead of relying on a single reasoning path or answer-level voting, MedExRAG maintains an exploration memory that records, updates, and compresses information from the whole MCTS tree. This memory provides a compact global representation of multi-path exploration, allowing the model to inherit complementary evidence from different reasoning branches and generate a more comprehensive final answer.

To instantiate the MCTS process, MedExRAG defines six reasoning and retrieval actions, including question analysis, sub-question answering, single-step reasoning, summary-based answering, direct answering, and query generation with retrieval. These actions enable the model to analyze medical questions from multiple dimensions and acquire external evidence when necessary. To evaluate intermediate states efficiently, we further train a process-supervised reward model to guide the search process. Compared with repeated sampling and majority voting, the reward model provides more direct path evaluation, reducing unnecessary rollout cost and token consumption.

In summary, our main contributions are as follows:

- We propose MedExRAG, an integrated exploratory reasoning framework for medical RAG. It introduces an exploration memory mechanism to inherit and compress information from multiple MCTS reasoning paths, enabling global rather than single-path reasoning.
- We train a process-supervised reward model to evaluate intermediate reasoning states during MCTS, providing efficient search guidance while reducing token consumption during inference.
- We conduct extensive experiments on medical question answering benchmarks, showing that MedExRAG improves RAG-based medical reasoning and validating the benefit of multi-path inherited integration.

2 Related Work

Medical Retrieval-Augmented Generation. Medical question answering requires factual, evidence-grounded, and domain-specific responses, making RAG an important approach for incorporating external biomedical knowledge into LLMs (Lewis et al., 2020; Gao et al., 2023; Singhal et al., 2023). Recent studies have explored retrieval-enhanced medical and biomedical QA, including medical RAG evaluation and domain-specific biomedical retrieval models (Xiong et al., 2024a; Jin et al., 2023). Beyond retrieval quality, adaptive medical RAG

systems further control when and how to acquire evidence. For example, Self-BioRAG introduces self-reflection into biomedical QA (Jeong et al., 2024), while i-MedRAG iteratively formulates follow-up questions to obtain additional medical evidence (Xiong et al., 2024b). These studies highlight the importance of evidence acquisition, but mainly focus on better or multi-round retrieval. In contrast, complex medical problems often require integrating complementary information across multiple reasoning directions.

Iterative and Agentic RAG. Iterative RAG methods improve static retrieve-then-generate pipelines by interleaving retrieval and generation. IRCoT, ITER-RETGEN, TC-RAG, and IterDRAG use intermediate reasoning or generation results to guide multi-round retrieval (Trivedi et al., 2023; Shao et al., 2023; Yue et al., 2025; Jiang et al., 2024). Agentic RAG further introduces action-driven decision-making: ReAct interleaves reasoning and acting (Yao et al., 2022), Reflexion incorporates verbal feedback (Shinn et al., 2023), Self-RAG trains models to retrieve and critique through self-reflection (Asai et al., 2024), and Search-o1 performs uncertainty-driven agentic search (Li et al., 2025). These methods enable more adaptive retrieval, but most still proceed along a single evolving trajectory. Therefore, early retrieval or reasoning errors may propagate, while useful evidence from alternative directions may not be sufficiently explored or preserved.

MCTS-based Reasoning for RAG. To overcome the limitations of linear reasoning, recent studies have explored inference-time scaling and structured search for LLM reasoning. Self-consistency samples multiple reasoning paths and selects the most frequent answer (Wang et al., 2022), while Tree of Thoughts organizes reasoning into a tree structure to explore candidate solutions (Yao et al., 2023). Compared with simple sampling or fixed tree expansion, Monte Carlo Tree Search (MCTS) more explicitly balances exploration and exploitation.

rStar introduces MCTS into LLM reasoning with multiple reasoning actions (Qi et al., 2025). Following this direction, several studies extend MCTS-based reasoning to RAG by incorporating retrieval-related actions. RAG-Star integrates tree-based search with retrieval guidance (Jiang et al., 2025); RARE combines RAG-based and non-RAG operations for flexible medical reasoning (Tran et al., 2025); MCTS-RAG expands diverse reasoning and retrieval strategies within a search tree (Hu et al., 2025); and AirRAG combines MCTS with reasoning actions and self-consistency-style verification (Feng et al., 2025). These methods show that tree-structured search can expand the solution space of RAG systems.

Despite their progress, existing MCTS-based RAG methods mainly use the search tree for path selection or answer selection. The final response is often derived from a single high-scoring path or selected by voting over final answers from multiple paths, which underutilizes intermediate analyses distributed across branches. MedExRAG differs by maintaining an exploration memory that records, updates, and compresses information from multi-branch exploration, enabling the final generator to inherit global exploration results without directly consuming the full search tree.

3 Methodology

In this section, we first provide an overview of MedExRAG, then introduce the reward-guided MCTS exploration process, and finally describe how the exploration memory is constructed after MCTS and used for final answer generation.

3.1 Overview of MedExRAG

Overall framework. To address complex medical questions that require multi-aspect reasoning and evidence integration, we propose MedExRAG, an integrated exploratory reasoning framework for medical RAG. The key idea is to use MCTS to explore diverse reasoning and retrieval trajectories, and then construct a compact exploration memory that aggregates useful information discovered across multiple branches. Figure 2 illustrates the overall framework of MedExRAG.

MCTS-based exploration. Given a medical question q, MedExRAG initializes a search tree rooted at the original question and performs iterative exploration through MCTS. Each node in the tree represents an intermediate reasoning state, which may contain question analysis, decomposed sub-questions, retrieved evidence, intermediate conclusions, or candidate answers. Each edge corresponds to a reasoning or retrieval action executed by the LLM. Through this formulation, MedExRAG is able to explore multiple possible reasoning directions rather than following a single chain-like trajectory.

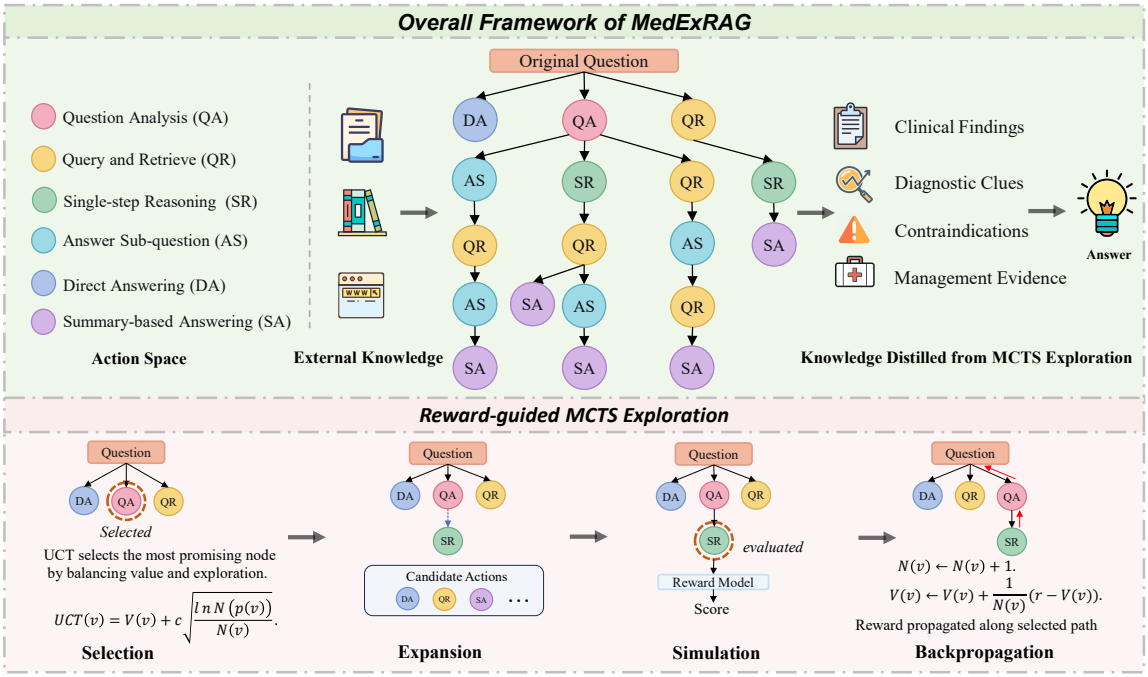

Figure 2: Overall framework of MedExRAG. The upper part illustrates the complete inference pipeline, where the LLM explores medical questions with multiple reasoning and retrieval actions, incorporates external knowledge, and distills useful signals from MCTS exploration into an exploration memory for final answer generation. The lower part shows the reward-guided MCTS process, including selection with UCT, expansion with candidate actions, simulation with the reward model, and backpropagation of reward signals.

Main components. The framework consists of three main components. First, MedExRAG defines a set of reasoning and retrieval actions to expand the search tree. These actions allow the model to analyze the question, decompose complex medical problems, perform step-wise reasoning, retrieve external knowledge, summarize intermediate information, and generate candidate answers. Second, to guide the exploration process, we train a process-supervised reward model that evaluates intermediate reasoning states and provides reward signals for MCTS. The reward scores are used to update node values during backpropagation, enabling the search process to balance exploration and exploitation. Third, instead of selecting the final answer from a single high-scoring trajectory, MedExRAG constructs an exploration memory after MCTS exploration to record and compress useful information from explored branches.

Exploration memory. This design distinguishes MedExRAG from existing MCTS-based RAG methods. In prior approaches, the search tree is mainly used to identify a strong reasoning path or to produce multiple candidate answers for voting. However, medical questions often involve complementary evidence distributed across different reasoning branches. Therefore, relying on a single path may lead to incomplete reasoning. MedExRAG treats the MCTS tree not only as a path-search structure, but also as a global exploratory reasoning space.

After exploration, the memory selectively inherits key findings, supporting evidence, intermediate conclusions, and uncertain signals from multiple branches, forming a compact representation of the global exploration process. By compressing branch-level information before final generation, MedExRAG avoids exposing the generator to redundant or conflicting raw trajectories while still preserving useful signals discovered during search.

Final answer generation. After the MCTS exploration is completed, the final answer is generated based on the original question and the exploration memory, which already contains selected supporting evidence and useful intermediate findings from multiple branches. In this way, MedExRAG enables the LLM to utilize information discovered from multiple exploratory paths while avoiding the excessive context length and noise caused by directly feeding the entire search tree into the model.

3.2 MCTS-based Exploratory Reasoning

MedExRAG formulates the reasoning process as an MCTS problem. Given the input question q, the root node s_0 represents the initial state without any intermediate reasoning. Each subsequent node s denotes a reasoning state, which consists of the current reasoning history, retrieved evidence, and intermediate conclusions. An edge between two nodes corresponds to an action executed by the LLM, and the newly generated content is appended to the current state to form a child node.

To enable diverse exploration, we define six reasoning and retrieval actions:

- **A1: Question Analysis (QA)**, which analyzes medical entities, constraints, and potential reasoning directions;
- **A2: Answer Sub-question (AS)**, which decomposes complex questions and answers local clinical aspects;
- **A3: Single-step Reasoning (SR)**, which advances the current reasoning state based on existing information;
- **A4: Summary-based Answering (SA)**, which summarizes intermediate reasoning and evidence along the current path;
- **A5: Direct Answering (DA)**, which generates a candidate answer when the current state is sufficiently informative;
- **A6: Query and Retrieval (QR)**, which produces retrieval queries and incorporates external medical evidence into the reasoning process.

During each MCTS iteration, MedExRAG performs selection, expansion, simulation, and back-propagation. For node selection, we adopt the Upper Confidence Bounds applied to Trees (UCT) strategy to balance exploration and exploitation. Given a current node u, the next child node is selected as:

$$v^* = \arg\max_{v \in \mathcal{C}(u)} \left(V(v) + c\sqrt{\frac{\ln N(p(v))}{N(v)}} \right),$$

where $\mathcal{C}(u)$ denotes the set of child nodes of u, $V(v)$ denotes the average value of node v, $N(v)$ is the visit count of node v, $p(v)$ is the parent node of v, and c is the exploration coefficient.

After an action is selected, the LLM executes the corresponding operation to expand the search tree. If the selected action involves retrieval, the model first generates queries and then retrieves relevant medical evidence from the external corpus. During simulation, the expanded node is evaluated by the reward model, and the resulting score is back-propagated to update the value statistics of its ancestor nodes. Through iterative exploration, MedExRAG constructs a tree-structured reasoning space that covers multiple clinical perspectives instead of committing to a single reasoning trajectory.

3.3 Reward-guided Search and Memory-augmented Answer Generation

A key challenge in MCTS is how to efficiently estimate the value of intermediate states. A straightforward solution is to perform repeated sampling and select the final answer by majority voting. However, this strategy introduces substantial computational overhead, especially when the search tree becomes deep or when multiple reasoning branches require external retrieval. To reduce the cost of repeated rollouts, MedExRAG trains a process-supervised reward model to directly evaluate intermediate reasoning states during search.

To construct training data for the reward model, we run MCTS on the training questions with known ground-truth answers. For a leaf node, we directly compare its generated answer with the ground truth and assign a reward of 1 if the answer is correct and 0 otherwise. For an internal node, we generate a temporary child node by applying the **A4: Summary-based Answering** action, and then evaluate whether the generated answer matches the ground truth. Since each state is sampled multiple times during data construction, the final reward is represented as a soft score in $[0, 1]$, corresponding to the empirical correctness rate rather than a binary label. In total, we use 2,335 training questions to construct MCTS trajectories and obtain 64,934 training instances for reward model fine-tuning.

We adopt *Llama-3.1-8B-Instruct* (Grattafiori et al., 2024) as the backbone of the reward model and fine-tune it with LoRA. The reward model takes the reasoning history of the current path as input and outputs a scalar score indicating the likelihood that the current state can lead to the correct answer. During inference, this score is used as the simulation value in MCTS and is back-propagated to update the value statistics of ancestor nodes. Compared with deep simulation or repeated ma-

jority voting, this design provides more efficient state evaluation and reduces unnecessary token consumption. Detailed training settings are provided in Appendix A.2.

After MCTS exploration, MedExRAG first applies consistency voting over the candidate answers generated by different branches. If all valid branches support the same answer, the agreed answer is directly returned, since no additional integration is necessary. Otherwise, MedExRAG constructs an exploration memory from the explored branches. This memory is a compact textual representation obtained by prompting the LLM to filter, merge, and compress information relevant to the original question. It records selected supporting evidence, intermediate conclusions, and potential conflicts across branches.

Finally, MedExRAG generates the answer based on the original question and the exploration memory. In this way, the final response is not restricted to a single high-scoring path or answer-level voting, but can benefit from complementary information discovered across multiple reasoning branches.

4 Experiments

In this section, we evaluate MedExRAG and competitive baseline systems on five complex medical QA tasks. We aim to examine whether MedExRAG can improve medical RAG performance by combining exploratory reasoning, reward-guided search, and memory-augmented answer generation.

4.1 Experimental Settings

Datasets. We conduct experiments on five widely adopted medical QA datasets, as shown in Table 1. All datasets are evaluated in a closed-set answer selection setting. Table 1 reports the sampled splits used in our experiments rather than the full original dataset sizes, and the same splits are used for all compared methods.

These tasks cover two core medical scenarios: complex medical reasoning and medical research QA. Specifically, MedQA (Jin et al., 2021) is an exam-style multiple-choice benchmark that stresses clinical reasoning and decision-making; MedMCQA (Pal et al., 2022) is a large-scale medical multiple-choice dataset spanning both basic and clinical medicine; and MMLU-Med (Hendrycks et al., 2020) is the medical subset of MMLU for assessing broad knowledge across medical subfields. For medical research QA, PubMedQA (Jin et al., 2019) evaluates evidence-grounded *yes/no/maybe* questions based on PubMed abstracts, while BioASQ (Krithara et al., 2023) is evaluated using its *yes/no* subset, which emphasizes literature-centric biomedical question answering.

Dataset	Category	Train	Dev	Test
MedQA	Complex medical reasoning	500	1272	500
MedMCQA		500	1000	500
MMLU-Med		435	109	500
PubMedQA	Medical research QA	400	100	500
BioASQ		500	149	500
Total	–	2335	2630	2500

Table 1: Statistics of the medical QA datasets used in our experiments.

Baselines and Metrics. To investigate the effect of different reasoning and retrieval strategies on complex medical RAG tasks, we compare MedExRAG with multiple competitive baselines. We first include Zero-shot prompting to evaluate the intrinsic capability of the base LLM without retrieval. We also compare with vanilla RAG, which performs a single retrieval-and-generation process based on the original question. For iterative retrieval methods, we compare with ReAct and Iter-DRAG, which conduct multi-step retrieval and reasoning along a single trajectory. For MCTS-based retrieval methods, we compare with RARE and MCTS-RAG, which explore multiple reasoning paths with MCTS and use consistency-based strategies to determine the final answer.

We use *Qwen3-8B* (Team, 2025) and *Llama-3.1-8B-Instruct* (Grattafiori et al., 2024) as the generator LLMs. Since all datasets are evaluated as closed-set answer selection tasks, we report Accuracy (ACC) as the evaluation metric.

Retrieval Corpus and Implementation Details. Our retrieval corpus follows the construction of MedCorp (Xiong et al., 2024a) and ClinicalCorp (Corbeil, 2024), covering both general biomedical literature and clinical-domain knowledge sources. More details of the corpus construction are provided in Appendix B.3. In the retrieval phase, we employ *bge-base-en-v1.5* (Xiao et al., 2023) as the embedding model and use *bge-reranker-v2-m3* (Li et al., 2023; Chen et al., 2024) as the reranker. For fair comparison, iterative retrieval methods and MCTS-based methods are allowed to generate at most three queries for each exploration action. For each query, the retriever

Model	Method	BioASQ	MedMCQA	MedQA	MMLU-Med	PubMedQA	Overall
Qwen3-8B	Zero-shot	73.4	61.0	64.6	80.0	53.6	66.5
	Vanilla RAG	74.0	57.8	60.6	80.8	63.8	67.4
	ReAct	72.2	62.4	65.4	80.4	31.2	62.3
	IterDRAG	70.4	60.2	63.6	79.8	49.0	64.6
	MCTS-RAG	83.4	38.6	68.6	82.8	22.2	59.1
	RARE	80.4	**65.6**	**70.6**	82.2	34.2	66.6
	Ours	**84.0**	60.6	64.4	**82.9**	**66.0**	**71.6**
LLaMA3.1-8B	Zero-shot	82.6	54.4	63.6	75.0	60.2	67.2
	Vanilla RAG	81.0	61.2	65.0	76.0	65.4	69.7
	ReAct	63.2	46.4	58.0	45.2	49.6	52.5
	IterDRAG	71.4	58.6	59.0	72.2	62.8	64.8
	MCTS-RAG	85.0	36.6	**69.8**	81.0	60.0	66.5
	RARE	85.5	62.8	**69.8**	81.0	61.0	72.0
	Ours	**85.8**	**63.6**	65.8	**81.4**	**65.6**	**72.4**

Table 2: Experimental results on medical QA benchmarks. All results are reported in percentage. The best result within each model group is shown in bold.

returns the top 20 chunks, and the reranker selects the top 2 chunks as supporting evidence.

For MedExRAG, we set the maximum reasoning depth to 5, enabling deep but efficient medical reasoning. We configure the number of rollouts as 4 to allow multiple reasoning branches to be explored while maintaining a reasonable inference cost. We further analyze the effect of different rollout numbers in Appendix C.1. Unless otherwise specified, the same retrieval configuration and generation backbone are used across all compared methods.

4.2 Main Results

Table 2 presents the overall comparison between MedExRAG and different baseline methods on five medical QA datasets. We report results under two generator backbones, *Qwen3-8B* and *Llama-3.1-8B-Instruct*, to examine whether the proposed framework is consistently effective across different LLMs. As shown in the table, MedExRAG achieves the best overall accuracy under both backbones, improving the overall score to 71.6% with Qwen3-8B and 72.4% with Llama-3.1-8B-Instruct. These results demonstrate the general effectiveness of integrating exploratory reasoning with medical RAG.

Compared with Zero-shot and vanilla RAG, MedExRAG improves the overall accuracy under both generator backbones. This suggests that direct generation and single-step retrieval are insufficient for complex medical QA, and that exploratory retrieval and reasoning can better collect and organize useful medical evidence. Compared with iterative methods such as ReAct and IterDRAG, MedExRAG also achieves stronger overall performance. This indicates that tree-structured exploration can cover a broader reasoning space than chain-like iterative reasoning, which may be constrained by early retrieval or reasoning decisions.

MedExRAG also achieves higher overall accuracy than MCTS-based baselines such as RARE and MCTS-RAG. This result indicates that the improvement does not only come from MCTS exploration itself, but also from the exploration memory used for final answer generation, which allows the model to integrate complementary information distributed across multiple branches. The advantage is especially clear on BioASQ and PubMedQA, where evidence-grounded answering and literature-based information integration are particularly important. These results support our motivation that complex medical RAG benefits from aggregating useful findings across multiple exploratory paths rather than relying only on a single high-scoring path or answer-level voting.

On MedQA, the advantage of MedExRAG is less pronounced, and it does not consistently outperform the strongest MCTS-based baselines. One possible reason is that exam-style clinical decision-making questions may require precise selection among closely related options, where a single well-optimized reasoning path can already be highly effective. Nevertheless, MedExRAG achieves the best overall performance across both backbones, validating its effectiveness and generality across

Method	Overall	
	Qwen3-8B	Llama-3.1-8B
MedExRAG w/o Memory	70.6	67.5
MedExRAG	**71.6**	**72.4**
Δ	+1.0	+4.9

Table 3: Ablation study on exploration memory. Removing memory means that MedExRAG determines the final answer by majority voting over candidate branches.

Variant	Overall
MedExRAG w/ Random Expansion	68.9
MedExRAG w/ LLM Self-evaluation	69.3
MedExRAG w/ Reward Model	**71.6**

Table 4: Ablation study on the reward model under Qwen3-8B. Random expansion removes state-value estimation, while LLM self-evaluation replaces the trained reward model with direct LLM-based scoring.

different medical QA scenarios.

4.3 Ablation Studies

Exploration memory. To evaluate the contribution of exploration memory, we remove the memory-augmented answer generation module while keeping the same MCTS exploration process and reward model. In this setting, the final answer is determined by majority voting over candidate branches. As shown in Table 3, removing exploration memory consistently degrades performance under both generator backbones. This suggests that exploration memory is useful for organizing and reusing complementary information discovered across multiple reasoning branches, rather than relying only on answer-level voting.

Effect of Reward Model. To verify the effectiveness of the process-supervised reward model, we compare it with two alternative search evaluation strategies: random expansion and LLM self-evaluation. The random expansion variant removes explicit value estimation and expands candidate actions without reward-guided selection. The LLM self-evaluation variant directly prompts the generator LLM to score intermediate states. As shown in Table 4, the reward model achieves the best overall performance among the three strategies. This suggests that reward-guided evaluation is more effective than random expansion or direct LLM scoring. The trained reward model provides more stable process-level signals, helping MCTS select more promising reasoning branches.

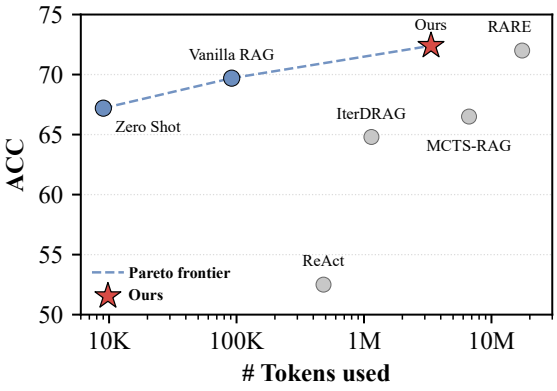

Figure 3: Accuracy-token trade-off of different methods under Llama-3.1-8B-Instruct. MedExRAG achieves the best accuracy with a moderate token budget and lies on the Pareto frontier.

4.4 Efficiency Analysis

We further compare the inference efficiency of MedExRAG with representative RAG baselines. As shown in Figure 3, zero-shot and vanilla RAG consume fewer tokens but achieve lower accuracy, while several iterative or MCTS-based baselines require substantially larger token budgets. MedExRAG achieves the best accuracy with a moderate token cost and lies on the Pareto frontier. Compared with vanilla RAG, MedExRAG obtains a clear accuracy improvement with a reasonable increase in token usage. Compared with stronger MCTS-based baselines such as RARE, MedExRAG achieves higher accuracy while using far fewer tokens. These results show that MedExRAG provides a favorable accuracy-efficiency trade-off.

5 Conclusion

In this paper, we propose MedExRAG, an integrated exploratory reasoning framework for medical RAG. Unlike existing MCTS-based methods that mainly rely on single-path selection or answer-level voting, MedExRAG constructs an exploration memory to compress useful information discovered across multiple reasoning branches. We further train a process-supervised reward model to guide MCTS exploration efficiently, reducing the cost of repeated sampling. Experiments on five medical QA benchmarks demonstrate that MedExRAG consistently improves answer accuracy and achieves a favorable performance-efficiency trade-off. These results highlight the value of multi-branch information integration for complex medical reasoning.

Limitations

Although MedExRAG achieves strong performance on medical QA benchmarks, it still has several limitations. First, our evaluation mainly focuses on closed-set answer selection, which may not fully reflect open-ended clinical question answering where models must generate detailed responses and express uncertainty. Second, the effectiveness of MedExRAG depends on the quality and coverage of the retrieval corpus. Incomplete, noisy, or outdated evidence may affect both the exploration process and the final memory-augmented answer. Third, the reward model requires additional training data and may inherit biases from benchmark-specific supervision, limiting its generalization to unseen medical domains. Finally, exploration memory compresses information from multiple branches, which reduces context length but may also discard useful details or preserve misleading signals. Future work will investigate more robust evidence verification, memory construction, and evaluation in realistic clinical scenarios.

References

Akari Asai, Zeqiu Wu, Yizhong Wang, Avi Sil, and Hannaneh Hajishirzi. 2024. Self-rag: Learning to retrieve, generate, and critique through self-reflection. In *International conference on learning representations*, volume 2024, pages 9112–9141.

Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, and 1 others. 2024. Graph of thoughts: Solving elaborate problems with large language models. In *Proceedings of the AAAI conference on artificial intelligence*, pages 17682–17690.

Cameron B Browne, Edward Powley, Daniel Whitehouse, Simon M Lucas, Peter I Cowling, Philipp Rohlfshagen, Stephen Tavener, Diego Perez, Spyridon Samothrakis, and Simon Colton. 2012. A survey of monte carlo tree search methods. *IEEE Transactions on Computational Intelligence and AI in games*, 4(1):1–43.

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *Preprint*, arXiv:2402.03216.

Jean-Philippe Corbeil. 2024. Iryonlp at mediqa-corr 2024: Tackling the medical error detection & correction task on the shoulders of medical agents. In *Proceedings of the 6th Clinical Natural Language Processing Workshop*, pages 570–580.

Wenfeng Feng, Chuzhan Hao, Yuewei Zhang, Guochao Jiang, Jingyi Song, and Hao Wang. 2025. Airrag: Autonomous strategic planning and reasoning steer retrieval augmented generation. *arXiv preprint arXiv:2501.10053*.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yixin Dai, Jiawei Sun, Haofen Wang, Haofen Wang, and 1 others. 2023. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*, 2(1):32.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *International conference on machine learning*, pages 3929–3938. PMLR.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.

Yunhai Hu, Yilun Zhao, Chen Zhao, and Arman Cohan. 2025. Mcts-rag: Enhancing retrieval-augmented generation with monte carlo tree search. *arXiv preprint arXiv:2503.20757*.

Gautier Izacard and Edouard Grave. 2021. Leveraging passage retrieval with generative models for open domain question answering. In *Proceedings of the 16th conference of the european chapter of the association for computational linguistics: main volume*, pages 874–880.

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2022. Few-shot learning with retrieval augmented language models. *arXiv preprint arXiv:2208.03299*, 1(2):4.

Minbyul Jeong, Jiwoong Sohn, Mujeen Sung, and Jaewoo Kang. 2024. Improving medical reasoning through retrieval and self-reflection with retrieval-augmented large language models. *Bioinformatics*, 40(Supplement_1):i119–i129.

Jinhao Jiang, Jiayi Chen, Junyi Li, Ruiyang Ren, Shijie Wang, Wayne Xin Zhao, Yang Song, and Tao Zhang. 2025. Rag-star: Enhancing deliberative reasoning with retrieval augmented verification and refinement. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 7064–7074.

Xinke Jiang, Yue Fang, Rihong Qiu, Haoyu Zhang, Yongxin Xu, Hao Chen, Wentao Zhang, Ruizhe Zhang, Yuchen Fang, Xu Chu, Junfeng Zhao, and Yasha Wang. 2024. Tc-rag:turing-complete rag's

case study on medical llm systems. *Preprint*, arXiv:2408.09199.

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, Won Kim, Qingyu Chen, Donald C Comeau, Lana Yeganova, W John Wilbur, and Zhiyong Lu. 2023. Medcpt: Contrastive pre-trained transformers with large-scale pubmed search logs for zero-shot biomedical information retrieval. *Bioinformatics*, 39(11):btad651.

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.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474.

Chaofan Li, Zheng Liu, Shitao Xiao, and Yingxia Shao. 2023. Making large language models a better foundation for dense retrieval. *Preprint*, arXiv:2312.15503.

Xiaoxi Li, Guanting Dong, Jiajie Jin, Yuyao Zhang, Yujia Zhou, Yutao Zhu, Peitian Zhang, and Zhicheng Dou. 2025. Search-o1: Agentic search-enhanced large reasoning models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 5420–5438.

Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the middle: How language models use long contexts. *Transactions of the association for computational linguistics*, 12:157–173.

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*.

Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. 2022. 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.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A Smith, and Mike Lewis. 2023. Measuring and narrowing the compositionality gap in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5687–5711.

Zhenting Qi, Mingyuan Ma, Jiahang Xu, Li Lyna Zhang, Fan Yang, and Mao Yang. 2025. Mutual reasoning makes smaller llms stronger problem-solver. In *International Conference on Learning Representations*, volume 2025, pages 20788–20807.

Zhihong Shao, Yeyun Gong, Yelong Shen, Minlie Huang, Nan Duan, and Weizhu Chen. 2023. Enhancing retrieval-augmented large language models with iterative retrieval-generation synergy. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 9248–9274.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. *Advances in neural information processing systems*, 36:8634–8652.

Karan Singhal, Shekoofeh Azizi, Tao Tu, S Sara Mahdavi, Jason Wei, Hyung Won Chung, Nathan Scales, Ajay Tanwani, Heather Cole-Lewis, Stephen Pfohl, and 1 others. 2023. Large language models encode clinical knowledge. *Nature*, 620(7972):172–180.

Karan Singhal, Tao Tu, Juraj Gottweis, Rory Sayres, Ellery Wulczyn, Mohamed Amin, Le Hou, Kevin Clark, Stephen R Pfohl, Heather Cole-Lewis, and 1 others. 2025. Toward expert-level medical question answering with large language models. *Nature medicine*, 31(3):943–950.

Qwen Team. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Hieu Tran, Zonghai Yao, Zhichao Yang, Junda Wang, Yifan Zhang, Shuo Han, Feiyun Ouyang, and Hong Yu. 2025. Rare: Retrieval-augmented reasoning enhancement for large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 18305–18330.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2023. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: long papers)*, pages 10014–10037.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. 2023. C-pack: Packaged resources to advance general chinese embedding. *Preprint*, arXiv:2309.07597.

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

Guangzhi Xiong, Qiao Jin, Xiao Wang, Minjia Zhang, Zhiyong Lu, and Aidong Zhang. 2024b. Improving retrieval-augmented generation in medicine with iterative follow-up questions. In *Biocomputing 2025: Proceedings of the Pacific Symposium*, pages 199–214. World Scientific.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 36:11809–11822.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2022. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*.

Zhenrui Yue, Honglei Zhuang, Aijun Bai, Kai Hui, Rolf Jagerman, Hansi Zeng, Zhen Qin, Dong Wang, Xuanhui Wang, and Michael Bendersky. 2025. Inference scaling for long-context retrieval augmented generation. In *International Conference on Learning Representations*, volume 2025, pages 72914–72938.

A Implementation Details

This section provides additional implementation details of MedExRAG, including the construction of reward model training data and the fine-tuning settings. These details supplement the method section and clarify how the process-supervised reward model is trained and used in our framework.

A.1 Training Data Construction

The training data for the reward model are constructed from the same five medical QA datasets used in our experiments. We use 2,335 training questions and run MCTS on these questions to collect intermediate reasoning states. Since the training questions are associated with ground-truth answers, we use the gold answers to assist the simulation stage and assign supervision signals to search states. In total, this process produces 64,934 training samples for reward model fine-tuning.

During data construction, each sampled state is evaluated according to whether it can lead to the correct answer. For a leaf node, we directly compare the generated answer with the ground-truth answer and assign a correctness-based score. For an internal branch node, we apply the **Summary-based Answering** action to generate temporary child nodes. This action is sampled multiple times to produce several candidate answers, which are then compared with the ground-truth answer. The empirical correctness rate is used as the score of the internal node. These scores are further back-propagated along the MCTS trajectory and used as process-level supervision for training the reward model.

A.2 Fine-tuning Settings

We adopt *Llama-3.1-8B-Instruct* as the backbone model and fine-tune it with LoRA to obtain the reward model. The reward model is trained as a regression model, taking the current reasoning path as input and predicting a scalar score in $[0, 1]$. This score estimates the likelihood that the current reasoning state can lead to the correct answer. The fine-tuning hyperparameters are shown in Table 5.

Hyperparameter	Value
LoRA rank	16
LoRA alpha	32
LoRA dropout	0.1
Number of epochs	2
Learning rate	5e-5

Table 5: Fine-tuning settings of the reward model.

B Experimental Details

This section provides additional details about the experimental setup, including the datasets, baseline implementations, and retrieval configurations used in our experiments. These details supplement the main experimental section and clarify how different methods are evaluated under a consistent medical RAG setting.

B.1 Dataset Details

We evaluate MedExRAG on five medical QA datasets, covering both complex medical reasoning and medical research QA scenarios. All datasets are evaluated in a multiple-choice format to ensure consistent comparison across methods.

- **MedQA** (Jin et al., 2021) is an exam-style medical QA benchmark derived from medical licensing examinations. It mainly evaluates clinical reasoning, diagnosis, treatment decision-making, and medical knowledge application. We use its English four-option subset in our experiments.

- **MedMCQA** (Pal et al., 2022) is a large-scale multiple-choice medical QA dataset collected from medical entrance examinations. It covers a wide range of topics from both basic medicine and clinical medicine, making it suitable for evaluating broad medical knowledge and reasoning ability.

- **MMLU-Med** (Hendrycks et al., 2020) is the medical subset of MMLU, consisting of multiple medical and biomedical subjects such as anatomy, clinical knowledge, college medicine, human genetics, and professional medicine. We use it to evaluate the model's general medical knowledge across different subfields.

- **PubMedQA** (Jin et al., 2019) is a biomedical research QA dataset based on PubMed abstracts. The original task provides questions with supporting contexts and yes/no/maybe answers. In our RAG setting, we remove the original contexts and retain only the questions and answer options, requiring the model to retrieve external evidence from the retrieval corpus.

- **BioASQ** (Krithara et al., 2023) is a biomedical question answering benchmark built from literature-centric QA tasks. We use the Yes/No subset in our experiments. Similar to PubMedQA, the given contexts are removed, and the model is required to answer the questions based on retrieved external biomedical evidence.

B.2 Baseline Details

We compare MedExRAG with several representative baselines, covering direct prompting, standard RAG, iterative retrieval, and MCTS-based RAG methods. For a fair comparison, all retrieval-based methods use the same retrieval corpus and retrieval pipeline unless otherwise specified.

- **Zero-shot.** The model is directly prompted to answer the original question without using any external documents. This baseline reflects the intrinsic medical reasoning ability of the generator LLM.

- **Vanilla RAG** (Lewis et al., 2020; Gao et al., 2023). Vanilla RAG performs a single retrieval-and-generation process. Specifically, the original question is used as the retrieval query, and the model answers the question based on the retrieved documents. No additional query generation or iterative refinement is applied.

- **ReAct** (Yao et al., 2022). ReAct interleaves reasoning and action execution along a single trajectory, allowing the model to decide when to invoke retrieval during the reasoning process. Since the original implementation relies on web search, we adapt ReAct to our controlled medical RAG setting by replacing web search with our retrieval corpus. The model iteratively generates reasoning steps and retrieval queries until it reaches the maximum exploration depth or produces a final answer. When the maximum depth is reached, the model is required to provide a final prediction based on the accumulated reasoning history and retrieved evidence.

- **IterDRAG** (Yue et al., 2025). IterDRAG performs iterative retrieval and reasoning along a single path. In our implementation, the maximum reasoning depth is set to 5. We use 10 shots for prompting, and each generated query retrieves 2 documents from the retrieval corpus.

- **MCTS-RAG** (Hu et al., 2025). MCTS-RAG extends RAG with MCTS-based multi-path exploration. Instead of selecting the final answer from a single reasoning path, it uses a reward-accumulation self-consistency strategy. After MCTS exploration, candidate answers from different trajectories are clustered according to semantic consistency, and each answer cluster is scored by aggregating the rewards of its associated reasoning paths. The answer with the highest accumulated score is selected as the final prediction.

- **RARE** (Tran et al., 2025). RARE improves rStar (Qi et al., 2025) by introducing retrieval-augmented reasoning actions into the MCTS process. For final answer selection, RARE uses a Retrieval-Augmented Factuality Scorer rather than simple majority voting. It decomposes candidate reasoning trajectories into statements, retrieves supporting evidence for each statement, and assigns factuality scores based on whether the statements are supported by the retrieved evidence. The trajectory with the highest factuality score is selected to produce the final answer.

For MCTS-based methods, including RARE, MCTS-RAG, and MedExRAG, we set the maximum search depth to 5. Each leaf node is sampled 16 times for candidate answer generation. For retrieval actions, each query-generation step produces at most 3 queries. Each query retrieves 20 candidate chunks, and the reranker selects the top 2 chunks, resulting in at most 6 retrieved chunks for each retrieval action. For RARE and MCTS-RAG, the sampling number for each internal branch node is set to 5. These settings ensure that all MCTS-based methods are evaluated under comparable search and retrieval budgets.

B.3 Retrieval Corpus and Retrieval Settings

Retrieval Corpus. Our retrieval corpus is constructed from MedCorp (Xiong et al., 2024a) and ClinicalCorp (Corbeil, 2024), covering biomedical literature, clinical guidelines, medical textbooks, and encyclopedic medical knowledge. Specifically, the corpus includes PubMed, Wikipedia, NICE, CMA, StatPearls, and medical textbooks. We split the documents into chunks with a maximum chunk

size of 512 tokens and an overlap of 128 tokens. Table 6 summarizes the statistics of the retrieval corpus.

- **PubMed.** PubMed is a large-scale biomedical literature database maintained by the National Center for Biotechnology Information. It provides biomedical research articles and abstracts, serving as an important source for evidence-based medical knowledge. In our corpus, PubMed also includes reference texts derived from PubMedQA.
- **Wikipedia.** Wikipedia provides broad encyclopedic knowledge across diverse domains. We use its medical-related content as a complementary source to cover general medical concepts, terminology, and background knowledge.
- **NICE.** The National Institute for Health and Care Excellence provides evidence-based clinical guidelines and health technology assessments. NICE documents are useful for clinical recommendations, treatment strategies, and healthcare decision-making.
- **CMA.** The Canadian Medical Association provides medical resources and clinical materials related to professional medical practice. We include CMA documents to enrich the corpus with clinically oriented knowledge and physician-facing resources.
- **StatPearls.** StatPearls is a comprehensive healthcare education resource containing medical articles and point-of-care materials written by expert contributors. It covers a wide range of diseases, diagnoses, treatments, and clinical concepts.
- **Textbooks.** The textbook collection consists of 18 widely used medical textbooks, which provide systematic medical knowledge and are important references for medical education and licensing examinations.

Retrieval Settings. We use *bge-base-en-v1.5* (Xiao et al., 2023) as the embedding model and *bge-reranker-v2-m3* (Li et al., 2023; Chen et al., 2024) as the reranking model. The vector index is built with Qdrant[2], and the retrieval pipeline is implemented using LlamaIndex[3]. For fair comparison, all retrieval-based baselines use the same retrieval corpus, embedding model, reranker, and retrieval pipeline.

Source	#Docs	#Chunks	Avg. Tokens/Chunk
PubMed	1.00M	1.06M	293
Wikipedia	500K	500K	163
NICE	1.66K	30.66K	469
CMA	432	6.68K	465
StatPearls	9.56K	102.94K	460
Textbooks	125.85K	125.88K	176
Total	1.64M	1.83M	262

Table 6: Statistics of the retrieval corpus. Documents are split into chunks with a maximum chunk size of 512 tokens and an overlap of 128 tokens.

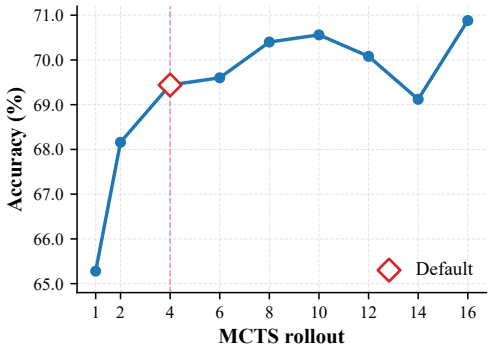

Figure 4: Effect of the number of rollouts on overall accuracy. Accuracy improves rapidly when the rollout number increases from 1 to 4, while further increasing rollouts brings smaller and less stable gains, indicating a trade-off between exploration and inference cost.

C Analysis

C.1 Effect of Rollouts

We further analyze the effect of the rollout number in MedExRAG. Figure 4 reports the overall accuracy under different rollout budgets. Increasing the number of rollouts from 1 to 4 leads to a clear performance improvement, suggesting that additional exploration helps the model discover more useful reasoning branches. However, when the rollout number continues to increase, the performance gain becomes smaller and less stable. This indicates that simply expanding the search budget does not always lead to consistent improvement, since additional branches may introduce redundant or noisy reasoning signals.

In our main experiments, we set the rollout number to 4 as a practical trade-off between accuracy

[2] https://qdrant.tech/

[3] https://www.llamaindex.ai/

and inference cost. Although larger rollout budgets may achieve slightly higher accuracy in some cases, they also require more computation. Therefore, rollout scaling can improve MedExRAG, but should be selected according to the available inference budget.

D Case Study

To further illustrate how MedExRAG utilizes multi-branch exploration, we present a case study in Figure 5. This example shows how different reasoning branches discover complementary and partially conflicting medical signals, and how the exploration memory helps generate a more complete final answer.

In this case, the question describes a lung cancer patient treated with pembrolizumab who develops fatigue, nausea, hypotension, hyponatremia, and hyperkalemia. During exploration, different branches focus on different aspects of the question. One branch reasons over the symptom pattern and identifies that hypotension, hyponatremia, and hyperkalemia are consistent with adrenal insufficiency. Another branch analyzes pembrolizumab as an immune checkpoint inhibitor and retrieves evidence that such agents can cause endocrine adverse events. A third branch retrieves information suggesting that lung cancer with hyponatremia may indicate SIADH, producing a plausible but incomplete candidate answer.

This example highlights the limitation of relying on a single MCTS trajectory or majority voting. A path focusing mainly on lung cancer and hyponatremia may select SIADH, while another path focusing on the full electrolyte and hemodynamic pattern supports adrenal insufficiency. Simple voting only considers the final candidate answers and may ignore intermediate evidence that explains why one answer is more clinically appropriate than another. In contrast, MedExRAG constructs an exploration memory by extracting key signals across branches, including the relationship between pembrolizumab and immune-related endocrine toxicity, the diagnostic significance of hypotension with hyponatremia and hyperkalemia, and the insufficiency of SIADH in explaining the full presentation. By integrating these signals, MedExRAG produces the final answer of pembrolizumab-related adrenal insufficiency, demonstrating the benefit of memory-augmented multi-branch reasoning.

E Prompt List

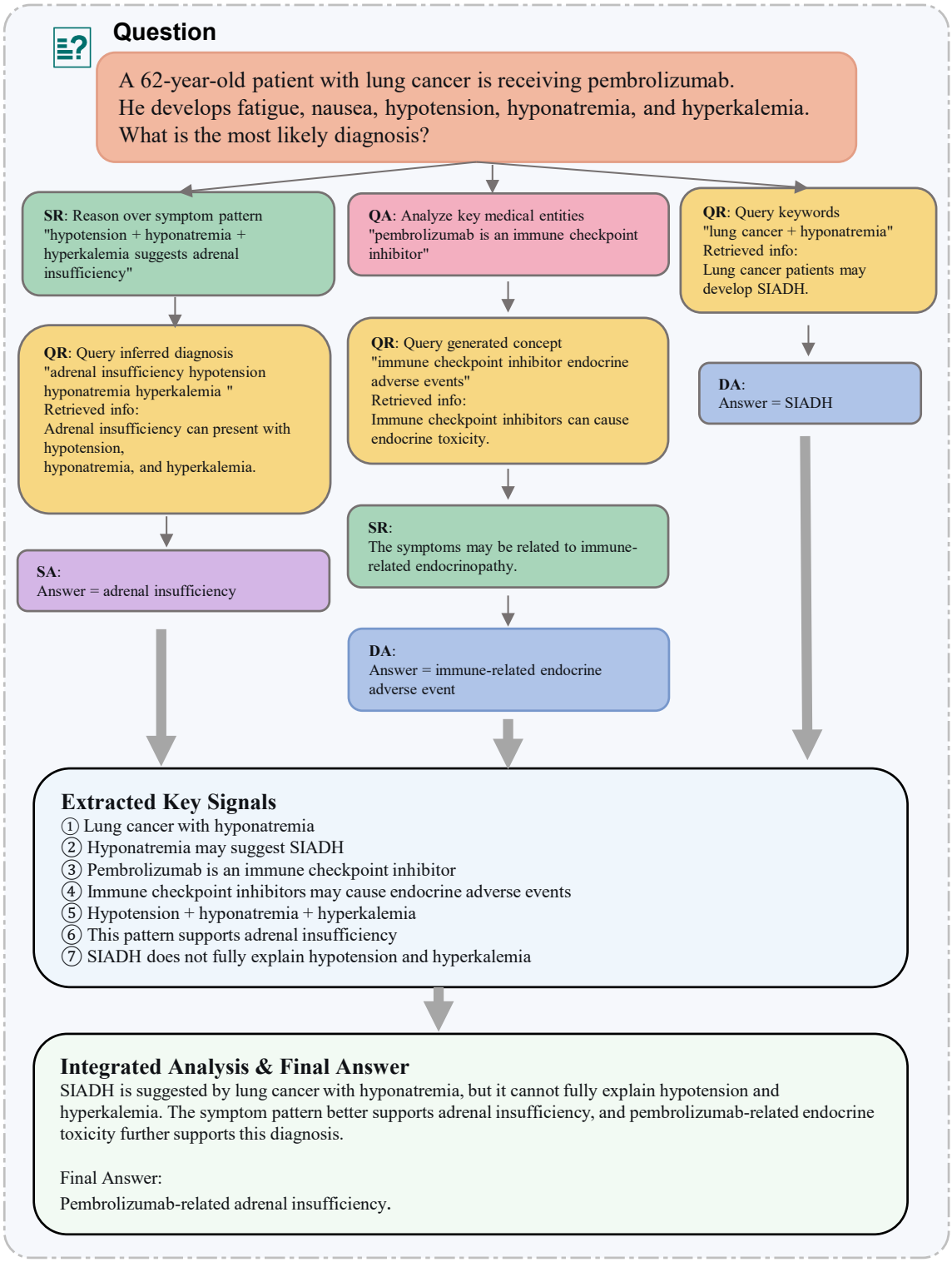

Figure 5: Case study of MedExRAG. Different MCTS branches explore different clinical directions. The exploration memory extracts key signals from these branches and supports the final integrated answer.

Prompt 1: Prompt for the MCTS Answer-Subquestion Action

You will be given several questions, some of which already have answers. You may also be provided with some documents, which may or may not be related to the questions. The existing answers may be correct or incorrect. For each question, if you know the answer, then answer it; otherwise, leave it unanswered. If you are certain that an existing answer is wrong and you know the correct answer, you may provide a new answer to that question. Here are some examples.
...
Now it's your turn to handle.
context
Response:

Prompt 2: Prompt for the MCTS Direct Answering Action

Here is a question. There may also be several sub-questions and answers, several documents, and some thinking and analysis. Please give the answer directly in the following format: "The answer is:"
...
Now it's your turn to handle.
context
Response:

Prompt 3: Prompt for the MCTS Question Analysis Action

Given a question, you may provide your analysis, rewrite the question to make it clearer, or break it down into multiple sub-questions to make it easier to answer. Choose exactly one of these three actions. Your response should begin with (Analysis), (Rewrite), or (Decomposition), respectively, to indicate the type of processing. Here are some examples.
...
Now here is the question you need to process.
context
Your response should begin with (Analysis), (Rewrite), or (Decomposition), respectively, to indicate the type of processing. Please directly provide the result of your analysis, rewrite, or decomposition.
Response:

Prompt 4: Prompt for the MCTS Single-step Reasoning Action

Here is a question. There may also be several sub-questions and answers, several documents, and some thinking and analysis. Please provide a step-by-step thinking process for the question.
Here are some examples.
...
Now it's your turn to handle.
context
Response:
Let's think step by step.

Prompt 5: Prompt for the MCTS Query and Retrieve Action

Here is a question. There may also be several sub-questions and answers, several documents, and some thinking and analysis. This additional information may not necessarily be correct or relevant to the question. Please generate up to three search queries that can be used to retrieve information to answer the question.
Here are some examples.
...
Now it's your turn to handle.
context
Response:

Prompt 6: Prompt for the MCTS Summary-Answer Action

Here is a question. There may also be several sub-questions and answers, several documents, and some thinking and analysis.
This additional information may not necessarily be correct or relevant to the question.
First, summarize the above information. Based on that, provide the final answer in the format: "The answer is:".
Here are some examples.
...
Now it's your turn to handle.
context
Response: