# MedResearcher-R1: Expert-Level Medical Deep Researcher via A Knowledge-Informed Trajectory Synthesis Framework

Anonymous ACL submission

## Abstract

Large Language Model (LLM)-based agents have recently demonstrated remarkable capabilities in deep research tasks, yet their application to the medical domain remains challenging. The leading proprietary systems achieve only modest results on complex medical benchmarks. This exposes two key bottlenecks: (i) limited clinical-reasoning capability in the underlying model, and (ii) unreliable access to authoritative medical evidence when constrained to open-web retrieval. To address these limitations and equip agents with robust clinical reasoning capabilities, we present MedResearcher-R1, a medical deep research agent featuring two core innovations. First, we propose Knowledge-Informed Trajectory Synthesis (KISA), a novel approach that builds medical knowledge graphs to construct complex multi-hop question–answer pairs centered on rare medical entities, overcoming the scarcity of high-quality domain-specific training data. Second, we integrate a medical retrieval engine alongside general-purpose tools, enabling precise and reliable synthesis of medical evidence. Through this methodology, we yield over 2,100 diverse trajectories spanning 12 medical specialties. Trained with supervised fine-tuning and reinforcement learning, our MedResearcher-R1-32B achieves state-of-the-art performance on MedBrowseComp (27.5/50 vs. 25.5/50 for o3-deepresearch) while demonstrating strong general performance on GAIA and xBench benchmarks. To the best of our knowledge, we present the first high-quality, tool-augmented medical dataset paired with a domain-specialized deep-research agent, demonstrating that smaller open-source models can surpass substantially larger proprietary systems on specialized medical tasks.

## 1 Introduction

Recent advances in Large Language Models (LLMs) have catalyzed the emergence of deep research systems. Unlike static question answering, deep research systems maintain evolving hypotheses, actively acquire external evidence via search and specialized tools, reconcile conflicting information, and synthesize grounded conclusions into structured artifacts (e.g., reports) over multi-step trajectories (Xu and Peng, 2025). Despite rapid progress, the medical domain imposes stricter requirements on reasoning validity and evidence authority, which current general-purpose deep research agents still fail to satisfy reliably (Chen et al., 2025b).

First, the reasoning architectures of general agents are often too rigid for the stochastic nature of medical inquiry. Standard agent workflows typically follow linear "plan-and-execute" trajectories. However, medical research is inherently abductive and iterative. It requires agents to constantly re-evaluate hypotheses and cross-validate conflicting data points (Yu et al., 2025).

Second, there is a fundamental disconnect between general pre-training and clinical data. General deep research agents are trained to optimize for plausible text generation rather than faithful clinical reasoning. They lack exposure to the specific cognitive trajectories used by medical professionals (Wang et al., 2025). While Reinforcement Learning (RL) offers a potential solution, it often struggles in this domain due to the sparsity of intermediate reasoning signals. Consequently, Supervised Fine-Tuning (SFT) on expert-demonstrated trajectories emerges as the most trustful strategy. However, existing data synthesis pipelines for SFT remain insufficient. They mainly focus on generating static, text-only samples (Quan, 2024; Nadas et al., 2025) or unifying disparate benchmarks into standardized schemas (Song et al., 2025).

Third, general agents suffer from a bottleneck due to inadequate information retrieval. Most deep research systems rely on commercial search engines designed for consumer relevance rather than

scientific accuracy. In medicine, where the "ground truth" is often sequestered in specialized, authoritative indices (e.g., FDA databases), reliance on open-web retrieval leads to the non-authoritative or outdated information (Lin et al., 2025). This "retrieval gap" is a primary driver of hallucination in medical agents, as they attempt to bridge missing evidence with plausible but fabricated details.

To bridge these gaps, we propose a framework that redefines how medical agents are designed and trained. We posit that robust medical reasoning demands both a domain-adaptive workflow and exposure to complex, knowledge-rich scenarios during training. Our contributions are as follows:

- **Domain-adaptive workflow for medical deep research.** We introduce a multi-turn, iterative workflow (Figure 1) where the agent alternates between internal reasoning ("think") and external tool invocation. A central feature of this architecture is the cross-validation of retrieved evidence to ensure clinical accuracy. The clinical accuracy is critical requirement in medicine, where reliance on partial information often leads to hallucination or error.
- **Knowledge-Informed Trajectory Synthesis (KISA).** We propose KISA, a novel data generation framework for supervised fine-tuning deep research agents. KISA leverages biomedical knowledge graphs to synthesize high-quality reasoning trajectories, effectively teaching the agent how to navigate the problem space and utilize tools strategically for deep research tasks.
- **Comprehensive medical tool integration.** We augment the agent's capabilities with a custom retrieval engine accessing authoritative data sources. Unlike general agents, our system directly queries FDA databases, official prescription records, clinical trial registries, and peer-reviewed literature to ground its reasoning in verified medical data.

## 2 Related Work

**LLM-based Agents for Deep Research**. Deep research (DR) systems equip LLMs to generate evidence-rich reports. Optimization strategies generally fall into three categories: workflow engineering, supervised fine-tuning, and reinforcement learning (RL) (Shi et al., 2025). The RL paradigm has yielded agents with great performance such as Search-R1 (Jin et al., 2025), DeepResearcher (Zheng et al., 2025), WebDancer (Wu et al., 2025), and Kimi-K2 (Team et al., 2025), while workflow engineering is exemplified by systems like Gemini DeepResearch (LLC, 2024), OpenAI DeepResearch (OpenAI, 2025), Manus (Ltd.), 2025), and SunaAI (AI, 2025).

Despite these advances, performance degrades in high-stakes fields like medicine (Chen et al., 2025b). General-purpose agents often lack the specialized retrieval tools and reasoning depth necessary for clinical synthesis. Consequently, they struggle to navigate authoritative medical databases or capture relationships between rare entities, a limitation this work addresses.

**Deep research agents for medicine**. In the medical domain, AI methodologies have evolved from isolated diagnostic models to integrated agentic systems. Initial approaches primarily leveraged Retrieval-Augmented Generation (RAG) to ground clinical decision-making in external literature (Zhao et al., 2025; Toma et al., 2023). Recently, this has expanded into multi-agent frameworks designed for specialized clinical workflows, such as sequential diagnosis (Nori et al., 2025) and dynamic knowledge management (Yao et al., 2024).

However, current agents struggle with exploratory medical research. Performance notably deteriorates on tasks requiring deep, multi-step inference (Schmidgall et al., 2025). This brittleness is attributed to a lack of training data which can reflect the high complexity of real-world medical investigation, such as linking rare diseases with disparate and authoritative findings. Therefore, developing high-quality training datasets for supervised fine-tuning medical DR system is necessary.

**SFT training data for deep research agents**. Despite the success of RL and workflow engineering, SFT for DR agents is hindered by a lack of training data. Standard synthetic data methods for LLMs (Quan, 2024; Nadas et al., 2025) are unsuitable for agentic systems. This is because they lack the requisite tool-use trajectories (e.g., <tool_call>). While the Agent Data Protocol (ADP) successfully standardizes diverse agent datasets into a common schema (Song et al., 2025), it focuses on organizing existing datasets rather than generating agentic training data from scratch. This can be particularly acute for medical DR agents training due to the lack of datasets in this domain. Existing approaches regarding SFT training data primarily focus on instructing LLMs to utilize functions and invoke real-world APIs (Patil et al., 2023; Qin et al., 2023; Tang et al., 2023; Chen et al., 2024). However,

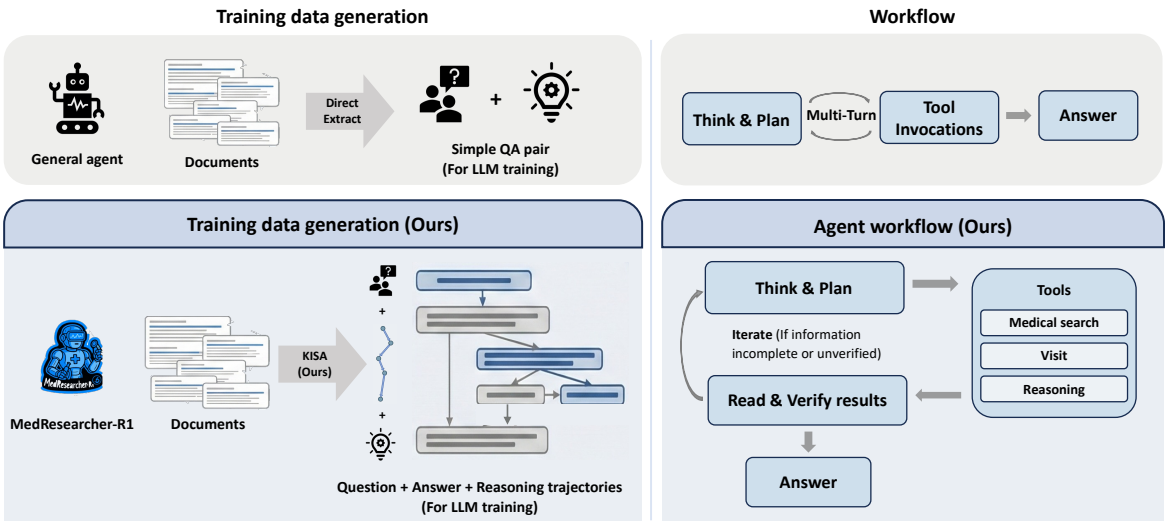


Figure 1: **Comparison of medical reasoning agents**. MedResearcher-R1 improves general-purpose agents from both the training-data and architectural perspectives. On the training data side, it is trained on a specially designed medical dataset and guided by reasoning trajectories. On the architectural side, it employs a multi-turn tool-invocation workflow based on a think, tool invocation, verification loop and is equipped with medical-specific tools.

such datasets fall short in enhancing the complex reasoning capabilities required for deep research. Consequently, we aim to propose a data construction method specifically designed to augment the cognitive processes of deep research agents.

## 3 MedResearcher-R1: Medical Deep Research Agent Framework

To adapt the agent workflow for solving deep medical research problems, we propose a *think-tool invocation-verification* loop (Figure 1). This process occurs before the final answer is generated. Specifically, given a query, MedResearcher-R1 performs the following steps: (i) **Think**: In this initial step, the agent plans the research trajectory, identifies required information, and determines which parts of the retrieved evidence require cross-validation to ensure accuracy. It also selects the appropriate tools and defines their specific tasks. (ii) **Tool Invocation**: The agent executes the tools selected during the "Think" phase. (iii) **Verification**: This step conducts consistency checks on the evidence flagged for cross-validation. It compares the evidence from different sources and draw a conclusion. Furthermore, once verification is complete and sufficient evidence has been accumulated, this module synthesizes the findings to generate the final evidence-based response.

The tool set aims to retrieve comprehensive medical information. To do this, we add a specialized Medical Search tool to standard utilities like web search and document reading. This specific tool queries authoritative medical repositories. Key sources include FDA datasets, prescription databases, clinical trial registries, and PubMed.

## 4 KISA: Knowledge-informed trajectory synthesis approach

We want to use Supervised Fine-Tuning (SFT) to train MedResearcher-R1, which helps it to learn reasoning steps rather than training with reinforcement learning for random exploration. To address the critical challenge of training data scarcity for medical deep-research agents, we propose a Knowledge-Informed Trajectory Synthesis Approach (KISA) that generates complex, multi-hop medical reasoning trajectories for agent SFT. KISA consists of five stages (Figure 2): (i) rare entity extraction and selection (ii) knowledge graph construction via search engine (iii) longest path extraction (iv) path enrichment with adjacent nodes and (v) reasoning trajectory generation with masked trajectory guidance. We now describe each of these stages in detail.

A knowledge graph represents real-world entities and the relationships among them. This is particularly useful for medical reasoning because it encodes tractable relationships between medical concepts. Therefore, we aim to generate questions with reasoning trajectories for agent SFT from a medical knowledge graph.

**Rare entity extraction and selection**. Inspired by large-scale efforts to construct knowledge graphs from literature (Kostis et al., 2020), we begin by

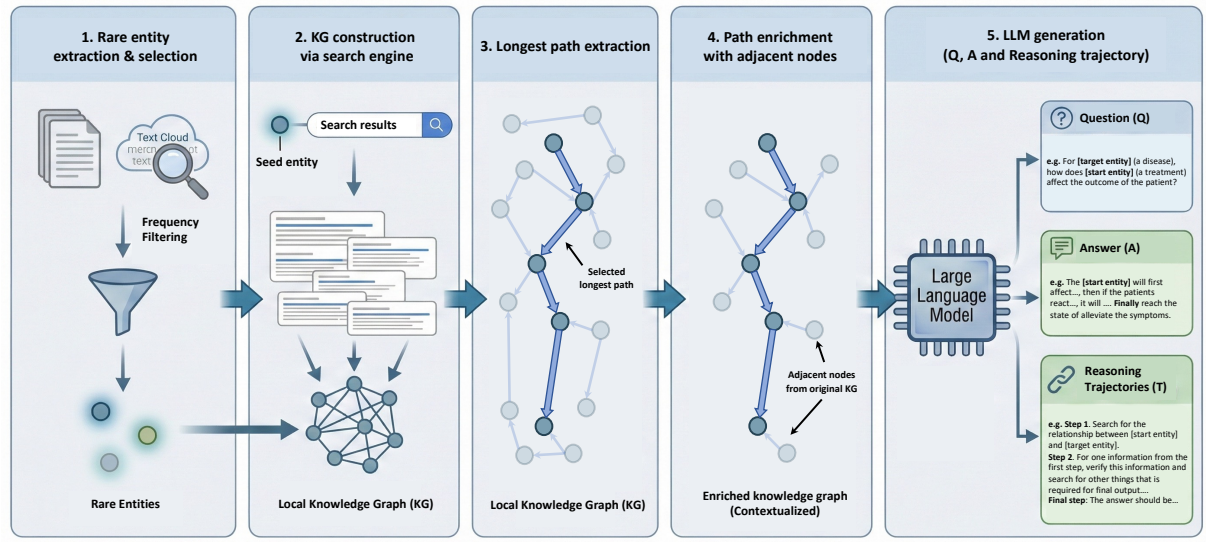


Figure 2: **KISA process illustration**. The process aims to generate explicit reasoning trajectories for agent SFT.

extracting medical terminologies as entities from over 30 million PubMed abstracts. Specifically, we use Qwen3-235B (Yang et al., 2025) with carefully designed prompts to extract precise entities. This process also returns basic statistics such as frequency of entities, which helps screening. Additionally, the entity set for the knowledge graph is constructed incrementally. Each time we consider adding a new entity to the entity set, we require LLM to determine whether it should be added as a distinct entity based on current entity set. This procedure helps us handle typos and different representations referring to the same entity. After the entity set is constructed, we select rare entities whose frequencies are less then a threshold ($10^{-6}$).

**Knowledge graph construction via search engine**. We focus on an entity $e \in \mathcal{E}$ that represents a rare medical term. For this entity, we construct a knowledge graph $\mathcal{G}(e) = (\mathcal{E}(e), \mathcal{R}(e), \mathcal{T}(e))$. $\mathcal{E}(e)$ denotes the entities, and $\mathcal{R}(e)$ denotes the relation types. The set $\mathcal{T}(e)$ contains triples in the form $(h, r, t)$. These represent a head $h$, a relation $r$, and a tail $t$. To construct the graph, we query the seed entity $e$ in a search engine. This retrieves data from both general and medical resources. We extract relational triples from the search results. These triples form the structure of $\mathcal{G}(e)$.

**Longest path extraction and enrichment**. We extract the longest path $P^*(\mathcal{G}(e))$ from $\mathcal{G}(e)$, i.e.

$$P^*(\mathcal{G}(e)) = \arg \max_{P \in \mathcal{P}(\mathcal{G}(e))} |P|$$

where $\mathcal{P}(G(e))$ denotes the set of paths in graph $\mathcal{G}(e)$ and $|P|$ denotes the length (number of edges) of path $P$. This strategy ensures that questions require multiple reasoning hops. Next, we retrieve the adjacent nodes for every node in this path. We use these to expand the path into a sub-knowledge graph $\mathcal{G}_{sub}(e)$. Let $V(P)$ denotes the set of nodes in path $P$ and $\mathcal{N}(e)$ denotes the neighbors of entity $e$ in graph $\mathcal{G}(e)$. Then

$$\mathcal{G}_{sub}(e) = (\mathcal{E}_{sub}(e), \mathcal{R}_{sub}(e), \mathcal{T}_{sub}(e))$$
$$\mathcal{E}_{sub}(e) = \cup_{v \in V(P^*(\mathcal{G}(e)))} \mathcal{N}(v)$$

This provides extra information to accurately describe the nodes. This context supports the masked trajectory guidance method. Finally, we use $\mathcal{G}_{sub}(e)$ to generate question and answer pairs with reasoning trajectories.

**Reasoning trajectory generation with masked trajectory guidance (MTG)**. We input the sub-knowledge graph into LLMs. They generate questions, answers, and reasoning trajectories in natural language. Domain experts verify the extracted paths to ensure medical relevance. Additionally, we enforce adaptive difficulty calibration. We test each question with OpenAI-o3 and GPT-4. If either model achieves accuracy greater than $50\%$, the system regenerates the question. This ensures sufficient complexity.

This process yields Q&A pairs with reasoning trajectories. Figure 3 (left) illustrates an example. The reasoning trajectories describe how to solve the given question from the initial entity step by step. However, these text trajectories lack tool-call indicators. They also lack observations from tool invocations which are necessary for agent training.

Therefore, we propose Masked Trajectory Guidance (MTG). This method generates comprehensive reasoning paths for agent SFT. It incorporates the thinking process, tool tokens, invocation observations, and the final answer.

**Masked trajectory guidance (MTG)**. We employ a powerful LLM to generate trajectories for supervised fine-tuning. Our goal is to guide the reasoning process. However, we must not leak information that requires retrieval from tool invocations. To achieve this, we input the natural language trajectories alongside the question. We mask key entities within these trajectories. Specifically, MTG identifies entities present in the extracted path $P^*(\mathcal{G}(e))$. It replaces them with placeholders, such as "Entity A". It also adds context descriptions based on adjacent nodes in the sub-knowledge graph. Figure 3 illustrates an example. We feed the questions with the masked guidance to OpenAI-o3 (OpenAI, 2025), which is equipped with our tool packages. The model generates complete reasoning trajectories, including the thinking process, tool tokens, and final answers. This output is ideal for agent SFT. We store the resulting data in $\mathcal{D}_{guided}$.

We also control the data mix. This prevents overreliance on trajectory guidance in case there are easier reasoning ways to reach the answers. In specific, we create another training dataset called $\mathcal{D}_{explore}$. Here, we input only the questions into OpenAI-o3. We do not provide masked trajectories for detailed guidance. If the LLM generates a correct answer, we accept the associated trajectory. This ensures the model learns tool invocations naturally. It also prevents overfitting to the longest reasoning paths.

The final training dataset can be denoted as $\mathcal{D}_{train} = \{(x_i, y_i)\}_{i=1}^{N}$. Here, $x_i$ is the medical question. $y_i$ contains the answer and the reasoning trajectory. This dataset combines $\mathcal{D}_{guided}$ and $\mathcal{D}_{explore}$. In $\mathcal{D}_{guided}$, $y_i$ stems from questions and masked trajectories. In $\mathcal{D}_{explore}$, $y_i$ stems from questions alone. We maintain a 7:3 ratio between these subsets. This balances structured learning with exploration.

# 5 Large-scale Agent Training

We adopt a two-stage training paradigm, consisting of supervised fine-tuning (SFT) followed by reinforcement learning (RL) optimization.

## 5.1 Stage 1: Supervised Fine-Tuning

According to the KISA method described in section 4, we now have the training dataset $\mathcal{D}_{train} = \{(x_i, y_i)\}_{i=1}^{N}$. $x_i$ are the medical questions. $y_i$ are correct answers and real reasoning trajectories with tool invocations. Specifically, $y_i$ is serialized with a sequence of tokens.

$$y_i = \{y_{i,1}, y_{i,2}, ..., y_{i,T_i}\}$$

Each token $y_{i,t}$ might be an agent token (plain text or tool-call tokens) or a tool output token. Then the objective of the supervised fine-tuning is

$$L_{\text{SFT}}(\theta) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{t=1}^{T_i} m_{i,t} \log p_\theta(y_{i,t}|x_i, y_{i,<t})$$

where $p_\theta(y_{i,t}|x_i, y_{i,<t})$ denotes the probability of the agent parameterized with $\theta$ generating $y_{i,t}$ given the question $x_i$ and the previous tokens $y_{i,<t}$. $m_{i,t}$ is an indicator function that follows

$$m_{i,t} = \begin{cases} 1, & \text{if } y_{i,t} \text{ is an agent token} \\ 0, & \text{if } y_{i,t} \text{ is a tool output token} \end{cases}$$

We trained the agent from a base model Qwen2.5-32B-Instruct (Team, 2024; Yang et al., 2024) through SFT on 2,100+ synthetic medical trajectories generated by our KISA framework. The training incorporates robustness augmentations including tool failure simulation (5% corruption rate), intermediate thought supervision, and multi-task sampling across medical domains. This stage establishes fundamental tool usage patterns and medical reasoning capabilities. Detailed training configurations are provided in Appendix A.

## 5.2 Stage 2: Reinforcement Learning

Following SFT, we refine the agent using Grouped Regularized Policy Optimization (GRPO) with composite rewards balancing task accuracy, expert alignment, and efficiency. For reinforcement learning training, the agent is modeled as a policy $\pi_\theta$ parameterized by $\theta$. The reward is formulated as:

$$r = \alpha r_{\text{task}} + \beta r_{\text{expert}} - \gamma r_{\text{efficiency}}$$

where $r_{\text{task}}$ is the accuracy of the final predictions, $r_{\text{expert}}$ is a strong LLM evaluation on the medical accuracy and completeness, $r_{\text{efficiency}}$ is a penalty for redundant tool usage. For GRPO, we conduct $G$ rollouts $\{o_1, o_2, ..., o_G\}$ calculate the rewards and advantages $\hat{A}_{i,t}$ within the rollouts. Then the objective becomes

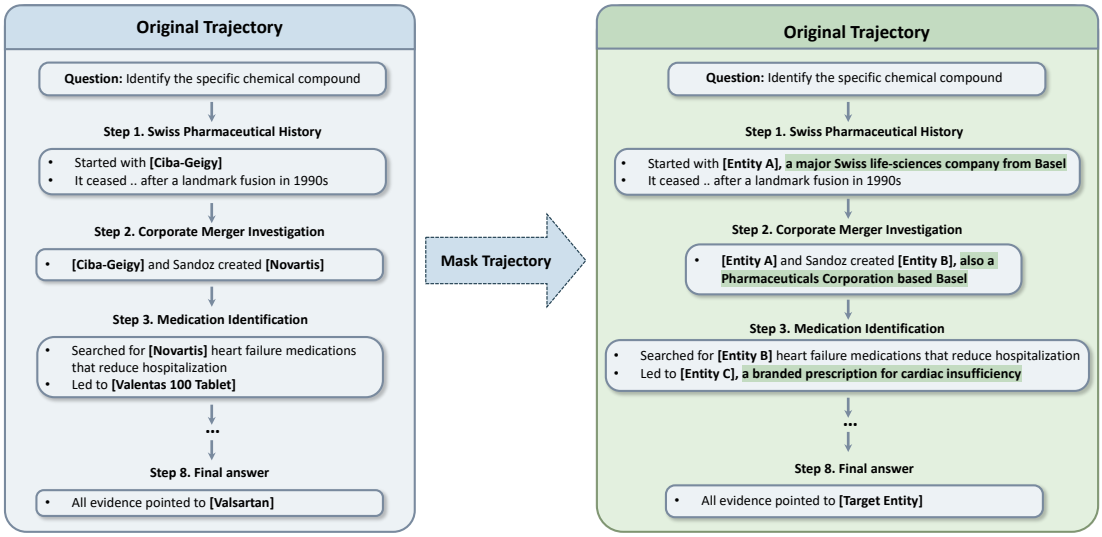


Figure 3: **Masked trajectory guidance sample**. The entities in the original reasoning path are masked with blurred descriptions. **[bold]** notation indicates entities from the knowledge graph on which this reasoning trajectory is constructed. The green-shaded texts are descriptions from the additional nodes.

$$\mathcal{J} = \frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{t=1}^{|o_i|}\min\{\frac{\pi_\theta(o_{i,t}|q,o_{i,<t})}{\pi_{\theta_{ref}}(o_{i,t}|q,o_{i,<t})}\hat{A}_{i,t},$$
$$\text{clip}\left(\frac{\pi_\theta(o_{i,t}|q,o_{i,<t})}{\pi_{\theta_{ref}}(o_{i,t}|q,o_{i,<t})}, 1-\epsilon, 1+\epsilon\right)\hat{A}_{i,t}\}$$

We deliberately omit KL regularization following recent findings (He et al., 2025) and implement curriculum learning for progressive task complexity. The GRPO objective employs group-level baseline normalization for stable gradient estimation. Other details are shown in Appendix A.

## 6 Experiments

### 6.1 Benchmarks

We evaluate MedResearcher-R1 using three public benchmarks: MedBrowseComp, GAIA, and XBench-DeepSearch. While MedBrowseComp focuses on the medical domain, the latter two serve as general-purpose benchmarks, allowing us to assess MedResearcher-R1's performance across both specialized medical tasks and general inquiries.

**MedBrowseComp** (Chen et al., 2025b) is designed to assess the capabilities of LLM-based agents in answering medical questions that require multi-hop reasoning. This benchmark comprises 50 questions drawn from a wide range of medical topics. Each question requires a detailed exploration of medical databases, literature, and clinical knowledge. Agents are evaluated based on their accuracy on these 50 questions.

**GAIA** (Mialon et al., 2023) is a general-purpose benchmark that evaluates assistant capabilities in tasks requiring multi-modal tool use, web search, and multi-step reasoning. We utilize a subset of 103 cases from the text-only validation set to test the agent's ability to understand complex scenarios, reason effectively, and interact with tools to generate responses.

| | **MedBrowseComp** (Accuracy ↑) |
|---|---|
| o3 search | 19.0/50 |
| Gemini-2.5-pro Deepsearch | 24.5/50 |
| o3 deepsearch | 25.5/50 |
| claude-cua | 18.0/50 |
| MedResearcher-R1 (Ours) | **27.5/50** |

Table 1: Performance Comparison on MedBrowseComp Benchmarks (number correct out of 50).

**XBench-DeepSearch** (Chen et al., 2025a) is a multi-domain benchmark focused on evaluating tool usage capabilities in search and information retrieval scenarios. It tests the agent's ability to perform advanced information synthesis across various domains.

### 6.2 Main Results

**State-of-the-Art Performance in Medical Research.** As shown in Table 1, MedResearcher-R1 establishes a new state-of-the-art on the challenging MedBrowseComp benchmark. With a score of **27.5/50**, it outperforms strong proprietary systems like o3-deepresearch (25.5) and Gemini-2.5-Pro-deepsearch (24.5). This result validates the effectiveness of our specialized data synthesis and domain-specific tools in equipping the agent with superior medical reasoning capabilities.

**Strong Generalization to Open-Domain Tasks.** Notably, our specialization in the medical domain does not come at the cost of general capabilities. Table 2 shows that MedResearcher-R1 achieves highly competitive performance on general agent benchmarks. On both GAIA (53.4) and XBench-DeepSearch (54.0), our 32B model performs on par with the open-source agent, WebSailor-32B (53.2 and 53.3, respectively). This demonstrates that the complex reasoning patterns and robust tool-use strategies learned from the medical domain transfer effectively to general problem-solving scenarios.

| Model | Paradigm | Xbench-DeepSearch | GAIA |
|---|---|---|---|
| Qwen-2.5-32B | Direct | 8.7 | 13.6 |
| Qwen-2.5-72B | Direct | 12.7 | 14.6 |
| GPT-4o | Direct | 18.0 | 17.5 |
| GPT-4.1 | Direct | 17.0 | 22.3 |
| QwQ-32B | Direct | 10.7 | 22.3 |
| o4-mini | Direct | 22.3 | 33.3 |
| DeepSeek-R1 | Direct | 32.7 | 16.5 |
| Qwen-2.5-32B | Search-o1 | 3.7 | 28.2 |
| WebDancer-32B | ReAct | 38.7 | 40.7 |
| QwQ-32B | Search-o1 | 25.0 | 39.8 |
| WebSailor-7B | ReAct | 34.3 | 37.9 |
| WebSailor-32B | ReAct | 53.3 | 53.2 |
| MedResearch-R1-32B(Ours) | ReAct | **54.0** | **53.4** |

Table 2: Performance Comparison on Xbench-DeepSearch and GAIA Benchmarks

### 6.3 Qualitative Analysis

To understand the underlying factors driving performance improvements, we conducted an in-depth analysis of the training data patterns and their impact on agent behavior. For medical deep research problems, the workflow based on an iterative think–tool invocation-verification paradigm might be the most significant improvement in deep research capabilities. Given the dynamic nature of medical knowledge, continuous cross-validation is essential for ensuring factual accuracy and correcting outdated information.

As illustrated in Figure 4, our agent achieves superior research depth through systematic evidence synthesis. The methodical approach relies on multiple cross-validation cycles to ensure response consistency. This make it contrast sharply with baseline models, which often exhibit premature convergence or suboptimal tool-use patterns.

**Cross validations from different resources are significant for medical deep research questions**. We performed a comparative analysis of successful trajectories, contrasting our cross-validation pipeline against a self-verification baseline. In cross-validation, the agent searches for the same information from different sources and draws a conclusion based on an analysis over all information. As comparison, a self-validation agent is set to self-check about whether the retrieved information is valid for the problem to be solved. Our results indicate that trajectories employing cross-validation achieve a 34.2% higher success rate on complex multi-hop reasoning tasks. This iterative approach ensures factual grounding and response convergence, which are vital for high-precision domains like medical diagnostics. These findings suggest that the efficacy of tool-augmented agent training is tied to the structural complexity of the training data, with iterative cross-validation serving as a driver of robust research capabilities.

## 7 Ablation Study

To isolate the factors driving performance gains, we systematically remove some components from MedResearcher-R1 to isolate their contributions. The ablation results are shown in Table 3. From the results, we can draw following conclusions.

**Tool sets and MTG method are important for agent**. The integration of medical tools markedly enhances performance ($40.0$ to $54.0$ on XBench), underscoring that specialized tool environments are essential for complex reasoning. MTG also improve the agent performance. Our analysis of the MTG-derived SFT data reveals a fundamental requirement for training dataset design: agents benefit most from heterogeneity that balances the exploitation of long reasoning chains with exploratory trajectories (our data mix of $D_{guided}$ and $D_{explore}$). This dual approach is key to developing agents capable of dealing with complex medical research.

**Rare entities-based KISA are crucial.** Restricting the knowledge graph to common entities results in a significant performance degradation, with scores on MedBrowseComp falling from $27.5$ to $23.0$. This decline suggests that common-entity graphs primarily generate queries with low reasoning complexity and high information accessibility. Consequently, models trained on such datasets fail to develop the sophisticated ability required for multi-hop reasoning. These results underscore that agent proficiency in solving complex problems is contingent upon the structural complexity and informational density of the knowledge graph.

**Supervised Fine-tuning works better when given high-quality training data.** SFT alone achieves $25.5/50$ on MedBrowseComp while RL alone fails ($12.0/50$). The relative failure of RL can be at-

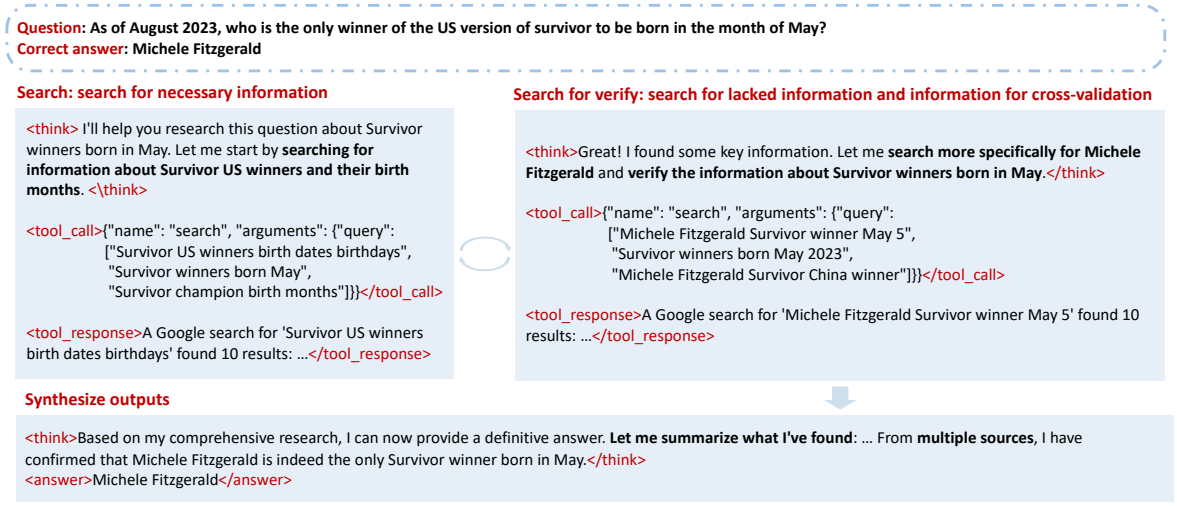


Figure 4: **Sample trajectory demonstrating the thinking-tool invocation-verification workflow**. The image depicts the agent's workflow for answering a given question. In the "think" phase, the agent performs (i) searching for necessary information, (ii) identifying missing information and selecting evidence that requires cross-validation. Then the verification step searches the required information and continues the loop. Finally the agent summarizes and synthesizes the retrieved information into an output.

| Model Configuration | MedBrowseComp (correct/50) | GAIA (%) | XBench-DeepSearch (%) | Avg. Tool Calls |
|---|---|---|---|---|
| MedResearcher-R1 (Full) | **27.5** | **53.4** | **54.0** | 4.2 |
| *Component Ablations* | | | | |
| w/o Medical Tools | 23.1 | 48.3 | 40.0 | 3.3 |
| w/o MTG | 24.2 | 44.3 | 47.8 | 3.5 |
| *Data Ablations* | | | | |
| Common Entities Only | 23.0 | 43.0 | 46.0 | 4.5 |
| *Training Ablations* | | | | |
| SFT Only | 25.5 | 49.0 | 48.0 | 3.4 |
| RL Only (no SFT) | 12.0 | 34.0 | 34.0 | 3.2 |

Table 3: **Ablation study for MedResearcher-R1.** Remove key components while keeping all other settings fixed. Component ablations include: removing the medical tool sets, and generating reasoning trajectories without the masked reasoning path. Data ablations consist of constructing questions using only common entities. Training ablations include using SFT only and using RL only without SFT training.

tributed to the sparse reward signals inherent in long-horizon medical reasoning. Without the structural priors provided by high-quality SFT trajectories, the agent's search space is too vast to explore effectively. This suggests that explicit behavioral guidance on expert-level reasoning chains is a prerequisite for stabilizing agentic performance before RL optimization.

## 8 Conclusion

In this work, we introduce MedResearcher-R1. It is a specialized medical research agent fine-tuned on data synthesized via our KISA framework. KISA addresses the scarcity of high-quality SFT data by systematically generating challenging, multi-hop medical queries paired with dense reasoning trajectories. This ensures that agents are exposed to the compositional complexity inherent in clinical research. MedResearcher-R1 integrating an iterative "think–tool invocations–verification" paradigm with a suite of domain-specific tools. The SFT with training data plus the special work paradigm make it achieve a state-of-the-art (SOTA) pass@1 accuracy of on MedBrowseComp, while maintaining robust generalization across broader agent benchmarks such as GAIA and XBench-DeepSearch.

Our empirical evaluation shows that training on KISA-generated data improves agent performance. Additionally, the integration of specialized medical tools and our custom workflow contributes to the high performance observed in the medical deep research domain. We believe that KISA-generated training data for agent SFT can help agents achieve more stable and better training results.

## Limitations

While our knowledge graph (KG)-based framework effectively synthesizes training data for complex research tasks, its generalizability to broader agentic domains remains to be fully explored. The current methodology is optimized for the structural properties of deep research problems, specifically those requiring multi-step reasoning and factual tracing. This may not be directly applicable to task-oriented agents (e.g., UI automation or creative generation). Furthermore, we acknowledge that medical research is increasingly multi-modal. However, our framework is not capable of generating training data for multi-modal deep search problems.

## References


Kortix AI. 2025. Suna: Open-source generalist ai agent.

Kaiyuan Chen, Yixin Ren, Yang Liu, Xiaobo Hu, Haotong Tian, Tianbao Xie, Fangfu Liu, Haoye Zhang, Hongzhang Liu, Yuan Gong, Chen Sun, Han Hou, Hui Yang, James Pan, Jianan Lou, Jiayi Mao, Jizheng Liu, Jinpeng Li, Kangyi Liu, and 14 others. 2025a. xbench: Tracking agents productivity scaling with profession-aligned real-world evaluations. *Preprint*, arXiv:2506.13651.

Shan Chen, Pedro Moreira, Yuxin Xiao, Sam Schmidgall, Jeremy Warner, Hugo Aerts, Thomas Hartvigsen, Jack Gallifant, and Danielle S. Bitterman. 2025b. Medbrowsecomp: Benchmarking medical deep research and computer use. *Preprint*, arXiv:2505.14963.

Yi-Chang Chen, Po-Chun Hsu, Chan-Jan Hsu, and Da shan Shiu. 2024. Enhancing function-calling capabilities in llms: Strategies for prompt formats, data integration, and multilingual translation. *Preprint*, arXiv:2412.01130.

Jujie He, Jiacai Liu, Chris Yuhao Liu, Rui Yan, Chaojie Wang, Peng Cheng, Xiaoyu Zhang, Fuxiang Zhang, Jiacheng Xu, Wei Shen, and 1 others. 2025. Skywork open reasoner 1 technical report. *arXiv preprint arXiv:2505.22312*.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. 2025. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *Preprint*, arXiv:2503.09516.

William J. Kostis, Georgios Lolas, Tim Donohue, Zoi Kourou, Nikolaos Koutsouleris, Theoharis Theoharis, and Matthew E. Falagas. 2020. A knowledge base of clinical trial outcomes from clinicaltrials.gov. *Scientific Data*, 7(1):210.

Xixun Lin, Yucheng Ning, Jingwen Zhang, Yan Dong, Yilong Liu, Yongxuan Wu, Xiaohua Qi, Nan Sun, Yanmin Shang, Kun Wang, Pengfei Cao, Qingyue Wang, Lixin Zou, Xu Chen, Chuan Zhou, Jia Wu, Peng Zhang, Qingsong Wen, Shirui Pan, and 5 others. 2025. Llm-based agents suffer from hallucinations: A survey of taxonomy, methods, and directions. *Preprint*, arXiv:2509.18970.

Google LLC. 2024. Gemini deep research.

Manus AI (Butterfly Effect Pte. Ltd.). 2025. Manus ai.

Grégoire Mialon, Clémentine Fourrier, Craig Swift, Thomas Wolf, Yann LeCun, and Thomas Scialom. 2023. Gaia: a benchmark for general ai assistants. *Preprint*, arXiv:2311.12983.

Mihai Nadas, Laura Diosan, and Andreea Tomescu. 2025. Synthetic data generation using large language models: Advances in text and code. *Preprint*, arXiv:2503.14023.

Harsha Nori, Mayank Daswani, Christopher Kelly, Scott Lundberg, Marco Tulio Ribeiro, Marc Wilson, Xiaoxuan Liu, Viknesh Sounderajah, Jonathan Carlson, Matthew P Lungren, Bay Gross, Peter Hames, Mustafa Suleyman, Dominic King, and Eric Horvitz. 2025. Sequential diagnosis with language models. *Preprint*, arXiv:2506.22405.

OpenAI. 2025. Deep research.

OpenAI. 2025. Openai o3 and o4-mini system card. Technical report, OpenAI. Accessed: 2025-12-17.

Shishir G. Patil, Tianjun Zhang, Xin Wang, and Joseph E. Gonzalez. 2023. Gorilla: Large language model connected with massive apis. *Preprint*, arXiv:2305.15334.

Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, Sihan Zhao, Lauren Hong, Runchu Tian, Ruobing Xie, Jie Zhou, Mark Gerstein, Dahai Li, Zhiyuan Liu, and Maosong Sun. 2023. Toolllm: Facilitating large language models to master 16000+ real-world apis. *Preprint*, arXiv:2307.16789.

Shanghaoran Quan. 2024. Automatically generating numerous context-driven sft data for llms across diverse granularity. *Preprint*, arXiv:2405.16579.

Samuel Schmidgall, Rojin Ziaei, Carl Harris, Eduardo Reis, Jeffrey Jopling, and Michael Moor. 2025. Agentclinic: a multimodal agent benchmark to evaluate ai in simulated clinical environments. *Preprint*, arXiv:2405.07960.

Zhengliang Shi, Yiqun Chen, Haitao Li, Weiwei Sun, Shiyu Ni, Yougang Lyu, Run-Ze Fan, Bowen Jin, Yixuan Weng, Minjun Zhu, Qiujie Xie, Xinyu Guo, Qu Yang, Jiayi Wu, Jujia Zhao, Xiaqiang Tang, Xinbei Ma, Cunxiang Wang, Jiaxin Mao, and 7 others. 2025. Deep research: A systematic survey. *Preprint*, arXiv:2512.02038.

Yueqi Song, Ketan Ramaneti, Zaid Sheikh, Ziru Chen, Boyu Gou, Tianbao Xie, Yiheng Xu, Danyang Zhang, Apurva Gandhi, Fan Yang, Joseph Liu, Tianyue Ou, Zhihao Yuan, Frank Xu, Shuyan Zhou, Xingyao Wang, Xiang Yue, Tao Yu, Huan Sun, and 2 others. 2025. Agent data protocol: Unifying datasets for diverse, effective fine-tuning of llm agents. *Preprint*, arXiv:2510.24702.

Qiaoyu Tang, Ziliang Deng, Hongyu Lin, Xianpei Han, Qiao Liang, Boxi Cao, and Le Sun. 2023. Toolalpaca: Generalized tool learning for language models with 3000 simulated cases. *Preprint*, arXiv:2306.05301.

Kimi Team, Yifan Bai, Yiping Bao, Guanduo Chen, Jiahao Chen, Ningxin Chen, Ruijue Chen, Yanru Chen, Yuankun Chen, Yutian Chen, Zhuofu Chen, Jialei Cui, Hao Ding, Mengnan Dong, Angang Du, Chenzhuang Du, Dikang Du, Yulun Du, Yu Fan, and 150 others. 2025. Kimi k2: Open agentic intelligence. *Preprint*, arXiv:2507.20534.

Qwen Team. 2024. Qwen2.5: A party of foundation models.

Augustin Toma, Patrick R. Lawler, Jimmy Ba, Rahul G. Krishnan, Barry B. Rubin, and Bo Wang. 2023. Clinical camel: An open expert-level medical language model with dialogue-based knowledge encoding. *Preprint*, arXiv:2305.12031.

Wenxuan Wang, Zizhan Ma, Meidan Ding, Shiyi Zheng, Shengyuan Liu, Jie Liu, Jiaming Ji, Wenting Chen, Xiang Li, Linlin Shen, and Yixuan Yuan. 2025. Medical reasoning in the era of llms: A systematic review of enhancement techniques and applications. *Preprint*, arXiv:2508.00669.

Jialong Wu, Baixuan Li, Runnan Fang, Wenbiao Yin, Liwen Zhang, Zhengwei Tao, Dingchu Zhang, Zekun Xi, Gang Fu, Yong Jiang, Pengjun Xie, Fei Huang, and Jingren Zhou. 2025. Webdancer: Towards autonomous information seeking agency. *Preprint*, arXiv:2505.22648.

Renjun Xu and Jingwen Peng. 2025. A comprehensive survey of deep research: Systems, methodologies, and applications. *Preprint*, arXiv:2506.12594.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

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, and 40 others. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Zijun Yao, Weijian Qi, Liangming Pan, Shulin Cao, Linmei Hu, Weichuan Liu, Lei Hou, and Juanzi Li. 2024. Seakr: Self-aware knowledge retrieval for adaptive retrieval augmented generation.

Yi Yu, Lingli Li, and Yaqin Li. 2025. Augmenting large language models and retrieval-augmented generation with an evidence-based medicine-enabled agent system. *medRxiv*, pages 2025–10.

Xuejiao Zhao, Siyan Liu, Su-Yin Yang, and Chunyan Miao. 2025. Medrag: Enhancing retrieval-augmented generation with knowledge graph-elicited reasoning for healthcare copilot. *Preprint*, arXiv:2502.04413.

Yuxiang Zheng, Dayuan Fu, Xiangkun Hu, Xiaojie Cai, Lyumanshan Ye, Pengrui Lu, and Pengfei Liu. 2025. Deepresearcher: Scaling deep research via reinforcement learning in real-world environments. *Preprint*, arXiv:2504.03160.

# Appendix

## A Training setting

### A.1 Supervised Fine-Tuning Configuration

**Dataset.** The final training dataset ($\mathcal{D} = \{(x^{(i)}, y^{(i)})\}_{i=1}^{N}$) generated by KISA contains a total 2,137 questions with corresponding trajectories ($N = 2,137$). The objective function is:

$$L_{\text{SFT}}(\theta) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{t=1}^{T_i} m_{i,t} \log p_\theta(y_{i,t}|x_i, y_{i,<t}) \quad (1)$$

where $p_\theta(y_{i,t}|x_i, y_{i,<t})$ denotes the probability of the agent parameterized with $\theta$ generating $y_{i,t}$ given the question $x_i$ and the previous tokens $y_{i,<t}$. $T_i$ is the token length of the $i$-th sample trajectory. $m_{i,t}$ is an indicator function that follows

$$m_{i,t} = \begin{cases} 1, & \text{if } y_{i,t} \text{ is an agent token} \\ 0, & \text{if } y_{i,t} \text{ is a tool output token} \end{cases}$$

**Robustness Augmentations.**

- **Tool failure simulation:** We introduce a 5% random corruption rate of tool outputs to encourage error recovery. Specifically, there is a 5% of probability that an API return will be replaced with a blank response. This simulates scenarios where information retrieval fails during the multi-turn thinking process. Therefore, this enhances the agents' capacity to re-search the information.
- **Multi-task sampling:** Balanced batching across different medical question types. For example, diagnosis (30%), treatment (25%), guidelines (25%), rare diseases (20%).

**Optimization.**

- Optimizer: AdamW with $\beta_1 = 0.9$, $\beta_2 = 0.98$
- Learning rate: $\lambda = 0.01$ with cosine annealing to $\eta_{\min} = 3 \times 10^{-7}$
- Batch size: 128 (16 per GPU $\times$ 8 H800 GPUs)
- Training epochs: 3
- Gradient clipping: 1.0
- Warmup steps: 100

### A.2 Reinforcement Learning Configuration

**Reward Components.** The composite reward function $r = \alpha r_{\text{task}} + \beta r_{\text{expert}} - \gamma r_{\text{efficiency}}$ comprises:

- $r_{\text{task}}$: Binary task completion (1.0 for correct, 0.0 for incorrect)
- $r_{\text{expert}}$: GPT-4 preference score $\in [0, 1]$ evaluating medical accuracy and completeness
- $r_{\text{efficiency}}$: Penalty for redundant tool usage, computed as:

$$\begin{aligned} r_{\text{efficiency}} =& 0.1 \times n_{\text{redundant}} + \\ & 0.2 \times n_{\text{post-answer}} + 0.15 \times n_{\text{irrelevant}} \end{aligned} \tag{2}$$

where $n_{\cdot}$ is the tool usage count of the footnote type.

**GRPO Configuration.** For GRPO, we conduct $G$ rollouts $\{o_1, o_2, ..., o_G\}$ calculate the rewards and advantages $\hat{A}_{i,t}$ within the rollouts. Then the objective for reinforcement learning is:

$$\begin{aligned} \mathcal{J} =& \frac{1}{G} \sum_{i=1}^{G} \frac{1}{|o_i|} \sum_{t=1}^{|o_i|} \min\{ \frac{\pi_\theta(o_{i,t}|q, o_{i,<t})}{\pi_{\theta_{ref}}(o_{i,t}|q, o_{i,<t})} \hat{A}_{i,t}, \\ & \text{clip}\left( \frac{\pi_\theta(o_{i,t}|q, o_{i,<t})}{\pi_{\theta_{ref}}(o_{i,t}|q, o_{i,<t})}, 1-\epsilon, 1+\epsilon \right) \hat{A}_{i,t} \} \end{aligned}$$

Detailed settings are:

- Group size: 4 responses per query
- Sampling temperature: 0.7
- PPO clip range: 0.2
- Value loss coefficient: 0.5
- Entropy coefficient: 0.01
- Training iterations: 500
- KL regularization: Disabled (following He et al. (2025))

## B Prompt

Here we show the prompts for constructing reasoning trajectories with masked trajectory guidance. The system prompts are:

```
role_definition = "You are a professional
    information research assistant"
       base_content = '''. Your core
    responsibility is to solve complex questions
     through intelligent tool usage and logical
    reasoning, providing accurate and reliable
    answers.

**Working Methodology**:

1. **Problem Analysis**: First analyze the
    nature of the question and required
    information types:
   - **Basic Reasoning**: Mathematical
    calculations, logical deduction, common
    sense judgments should be completed directly
   - **External Information**: Only use tools
    when you need real-time data, specific facts
    , or web content

2. **Intelligent Tool Usage**: Use tools only
    when external information is needed:
   - Search for unknown facts, data, news, etc.
   - Retrieve webpage content and specific
    information
   - Verify time-sensitive information

3. **Reasoning First**: For content that can be
    deduced through reasoning, analyze directly:
   - Mathematical operations and logical
    judgments
   - Time calculations and numerical
    relationships
   - Inference based on known information

4. **Comprehensive Analysis**: Combine external
    information obtained through tools with
    autonomous reasoning to provide complete
    answers.

**Working Principles**: Wisely choose when to
    use tools and when to rely on autonomous
    reasoning, ensuring efficient and accurate
    problem-solving.'''
```

The main prompts are:

```
"""Do research on the question and answer it
    when you finish the research. When you
    finish your research, you should explain
```

```
    first and then answer, your answer should be
     place inside <answer></answer>, and your
    answer should be direct answer without any
    explanation.

**IMPORTANT**: You are provided with some search
     guidance hints below. These hints suggest
    potential SEARCH DIRECTIONS and
    INVESTIGATION APPROACHES. You MUST NOT use
    any specific information from these hints
    directly as your answer or for direct
    reasoning.

## Search Guidance (Use ONLY as search
    directions):
{reasoning_path}

## Critical Instructions:

1. **Search Direction Only**: The above hints
    are ONLY suggestions for what topics/
    keywords to search for and what aspects to
    investigate. They are NOT factual
    information to be used directly in your
    reasoning or answers.

2. **Mandatory Tool Verification**: For any
    information mentioned in the guidance hints,
     you MUST use tools to independently find,
    verify, and confirm that information. Never
    assume the hints contain accurate facts.

3. **Balanced Approach**:
   - **Basic reasoning** (mathematical
    calculations, logical deductions, common
    sense): Handle directly without tools
   - **Information from hints** (specific facts,
     data, claims): MUST be verified through
    tools
   - **External information** (current events,
    specific details): Use tools to search and
    verify

4. **Independent Research**: Treat the guidance
    as a research roadmap only. You must
    independently discover, verify, and validate
     all specific information through your tools
    .

5. **Evidence-Based Answers**: Your final answer
     must be based on:
   - Verified information you actually found
    through tools (not from hints)
   - Sound reasoning and calculations you
    performed directly

**Remember**: The guidance hints may contain
    inaccurate or incomplete information. Always
     verify through tools before using any
    specific claims from the hints.

When you finish your research, explain your
    findings first and then provide your answer
    inside <answer></answer>. Your answer should
     be a direct answer without any explanation
    inside the answer tags.

User Question: """,
```