

Proof-of-Use: Mitigating Tool-Call Hacking in Deep Research Agents

Shengjie Ma¹, Chenlong Deng¹, Jiaxin Mao^{1,*}, Jiadeng Huang²,
Xuhui Jiang³, Teng Wang², Junjie Wu², Changwang Zhang², Jun Wang²

¹Gaoling School of Artificial Intelligence, Renmin University of China

²OPPO

³DataArc

*Corresponding author

Abstract

Deep research agents rely on iterative tool calls to gather external evidence for multi-step reasoning. While reinforcement learning (RL) improves planning and retrieval behavior, we identify a critical failure mode: tool-call hacking, where agents exploit format- and outcome-level rewards by generating superficially valid tool-use traces without genuinely grounding their reasoning in retrieved evidence. Unlike execution-based tools (e.g., code or math), whose effects are directly observable, the weak observability of causal dependencies between retrieved evidence and reasoning under format- and outcome-level supervision enables agents to maximize surface-level reward signals without genuinely grounding their reasoning in the returned evidence. This results in tool overuse, mode collapse, and decorative or hallucinated tool calls.

We propose Proof-of-Use (PoU), an evidence-grounded RL framework that strengthens the dependency between retrieval, reasoning, and final answers. PoU requires agents to cite normalized evidence identifiers during reasoning and optimizes a multi-objective reward combining step-level citation validity, global Answer-Support Alignment, and adaptive reward mixing. Experiments show that PoU mitigates tool-call hacking, improves task performance, and induces robust tool-use behavior under domain and tool shifts, even without explicit optimization for routing adaptation.

1 Introduction

Deep research agents (Jin et al., 2025; Song et al., 2025; Zheng et al., 2025) tackle complex information-seeking tasks by iteratively using external tools to retrieve, verify, and synthesize evidence. Unlike single-shot retrieval, they require multi-step planning over when, how, and which tools to invoke.

In realistic research settings, no single knowledge source is sufficient (Ma et al., 2025), so agents

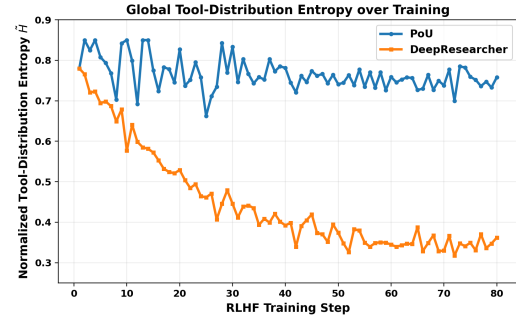


Figure 1: Tool-distribution entropy over training. At each training step, we compute the entropy of the tool usage distribution as $\tilde{H} = -\sum_i p_i \log p_i / \log K$, where p_i is the proportion of calls to tool i and K is the number of available tools. DeepResearcher exhibits a decreasing entropy, which indicates a tool-call hacking pattern (progressive collapsing toward a narrow tool selection). In contrast, PoU shows higher variance in early training, reflecting exploration, and converges to a stable intermediate entropy value, indicating a learned and task-coordinated tool usage pattern.

must coordinate heterogeneous tools such as web search, knowledge graphs, domain databases, and private document collections. This makes deep research agents a form of multi-round RAG, where effective reasoning requires both grounding answers in retrieved evidence and routing tool calls across sources with different coverage, structure, and reliability. However, recent evidence shows that brute-force all-in-one retrieval can outperform prompt-based selective routing (Xu et al., 2025), suggesting that shallow routing heuristics may disrupt cross-source coordination. Thus, selectivity is useful only when grounded in evidence usage.

Building on these observations, we extend the analysis from routing heuristics to the learning dynamics of RL-trained deep research agents. Prior works have applied reinforcement learning (RL) to enhance agents’ ability to multi-step reasoning with retrieval tools (Zheng et al., 2025). However,

we uncover a recurrent failure pattern that we term **tool-call hacking**—a domain-specific form of reward hacking (Skalse et al., 2025). Specifically, the agent learns tool-calling behaviors that boost the observed reward signal without genuinely improving grounding, which is highly susceptible to Goodhart’s Law (Majka and El-Mhamdi, 2025). We observe two consistent manifestations (as shown in Figure 1): (i) **Mode collapse via tool overuse**: policies default to a narrow subset of tools; (ii) **Hallucinated tool use**: tool calls are made, but the reasoning chain shows no causal dependence on the retrieved evidence. **Crucially, both failures arise because existing training signals primarily reward the act of calling tools, rather than the act of using them.**

These issues highlight the need for stronger signals that foster reasoning processes deeply integrated with tool interactions in multi-source environments.

As clarified in *Scope of Tools* (Appendix A), we focus on retrieval tools, whose impact on outputs is propagated through an implicit, semantic level evidence-reasoning-answer causal chain. In contrast, execution-style tools such as code or mathematical solvers have directly observable contributions, which makes retrieval tools easy to bypass or imitate without genuine informational dependence. To this end, rather than treating evidence grounding as an emergent side effect of task success, we argue that it should be an explicit learning objective in agentic reinforcement learning. We propose **Proof-of-Use (PoU)**, an evidence-grounded framework that refines the coarse-grained supervision commonly used in agentic RL. PoU is built on a stepwise interaction protocol: tool responses are normalized into evidence snippets with stable IDs. Upon observing them, the model must output a usefulness verdict and—if useful—cite the corresponding evidence IDs before reasoning. This exposes a verifiable dependency between retrieved evidence, reasoning, and decisions. On top of this protocol, PoU applies **two progressive process rewards** to strictly constrain step-wise reasoning. Furthermore, to address the reward sparsity and non-smooth gradients of outcome-only supervision, we introduce a global **Answer-Support Alignment reward**. This adheres to the intuition that effective learning should optimize the reasoning process alongside the final result. Crucially, we integrate these objectives via a curriculum-style **adaptive reward mixing** mechanism. This mech-

anism smoothly transitions focus from dense reasoning signals to final answer correctness, creating a stable gradient ascent path that bridges the gap to sparse outcome-based rewards.

Together, these design choices **couple every stage into an auditable Proof-of-Use chain**: tool outputs are anchored by stable IDs, steps explicitly cite evidence, cited grounds are selectively perturbed for verification, and the final answer is rewarded for consistency with the same evidence.

Contributions. We make three contributions. First, we characterize **tool-call hacking**, an under-explored reward-hacking risk in RL-trained research agents under multi-source retrieval, and formalize its key manifestations. Second, we propose **Proof-of-Use (PoU)**, an evidence-grounded RL framework with a stepwise citation protocol, progressive process rewards, answer-support alignment, and adaptive reward mixing. Third, experiments across multiple datasets show that PoU improves performance, mitigates tool-call hacking, and induces robust tool-use behavior under domain and tool shifts.

2 Method

PoU follows an iterative reasoning–acting–observing trajectory (Yao et al., 2023) for tool-augmented question answering. To address the often overlooked tool-call hacking in such settings, we design a set of specialized training rewards together with a matched step-level protocol that governs the interaction between reasoning, tool calls, and answers. A key property of PoU is **ID-consistent end-to-end coupling**: the same evidence IDs introduced by tools are (1) referenced at the step level via usefulness verdicts and citations, and (2) reused at the answer level for evidence–answer consistency checks. This shared identifier layer **turns evidence use into a traceable protocol**, making every reward attributable to specific retrieved evidence rather than superficial tool calls.

We will first formalize the tool-call hacking phenomenon, and then present the design of PoU rewards in detail.

2.1 Preliminaries

tool-call hacking (Definition): We define *tool-call hacking* as a specific form of reward hacking (Skalse et al., 2025) in tool-augmented LLM

agents, where the policy attains high training rewards by satisfying tool-call format requirements and answer-level correctness objectives, while its *stepwise reasoning and action behavior* does not causally depend on the content returned by the tools. As a result, the trained model deviates from the intended goal of retrieval-augmented generation, introducing a high but often hard-to-detect risk of hallucinated reasoning and answers.

Manifestations of tool-call hacking: In practice, tool-call hacking exhibits two direct and recurring manifestations in retrieval-augmented agents trained with format- and answer-level rewards.

First, the agent’s tool usage becomes poorly coordinated and collapses to a narrow or repetitive calling pattern. Instead of adaptively planning tool invocations based on the query structure and intermediate findings, the policy tends to overuse a single tool or a fixed call template that reliably satisfies format constraints.

Second, the agent’s stepwise reasoning or rollout-level answers become causally disconnected from the retrieved content. That is, semantically meaningful perturbations to the tool-returned results do not induce corresponding changes in intermediate reasoning steps or the final answer.

2.2 Settings of PoU

2.2.1 Initialization

Each episode begins with a **user query** x and a predefined **initial prompt** that provides explicit, structured schemas for each callable tool, including their invocation formats and argument descriptions.

To simulate realistic multi-source information environments for retrieval–reasoning agents, we design and implement four categories of black-box retrieval proxies that collectively capture the diversity of external knowledge access in agentic RAG systems. Each proxy corresponds to a distinct knowledge source and interaction modality, enabling the model to reason and retrieve in a unified yet source-specific manner (See Appendix F for details). All proxy outputs are uniformly defined within the initialization prompt, allowing the model to understand their expected arguments and formats before the reasoning begins.

2.2.2 Proxy Response

After reasoning, the policy may issue a structured tool call.

```
<tool_call>JSON_CALL</tool_call>
```

The environment executes the call and returns a *normalized proxy response* with a unified schema:

```
<tool_response>
REFERENCE_LIST_WITH_ID
</tool_response>
```

Here, the globally unique id of each reference is automatically assigned. All four proxies adopt this schema, so the model can cite evidence via `<ref>ID_1,ID_2</ref>` regardless of the source.

Granularity. For the *Web Search Proxy*, ref items are organized at the *webpage level*, typically including the page title, URL, and a short snippet. In contrast, the *Web Browsing*, *Local Search*, and *Knowledge Graph* proxies return ref items at the *paragraph (semantic-chunk) level*, enabling finer-grained evidence grounding.

2.2.3 Reasoning Process

At the beginning of each iteration, the agent first performs reasoning enclosed within a `<think>` tag. For the first step, the model generates initial planning and reasoning solely based on the user question. From the second step onward, each `<think>` block must begin with an explicit *helpfulness verdict and citation declaration*, which is treated as a **unified protocol**:

```
<helpful>yes|no</helpful>
<ref>id1,id2,...|null</ref>
```

where `<ref>` cites the identifiers of retrieved documents from the previous tool call’s structured output.

This mechanism forces grounding each reasoning step on prior observations. From a probabilistic probing perspective, the hidden state of `<helpful>` token quantifies the model’s internal judgment of evidence utility, serving as a distributional anchor that aligns the model’s internal activation patterns with the conditional likelihood of evidence-relevant reasoning trajectories. By optimizing the probability mass of reasoning paths causally dependent on retrieved evidence, it suppresses spurious thoughts that diverge from verifiable context. Meanwhile, the `<ref>` citation anchors the evidence utility judgment to explicit external sources, establishing a probing-visible linkage between internal representations and external tool outputs. This allows each intermediate conclusion to be traceably grounded in concrete evidence.

Together, the `<helpful>` and `<ref>` mechanisms constitute a dual-channel grounding scheme that leverages their interaction as a measurable and

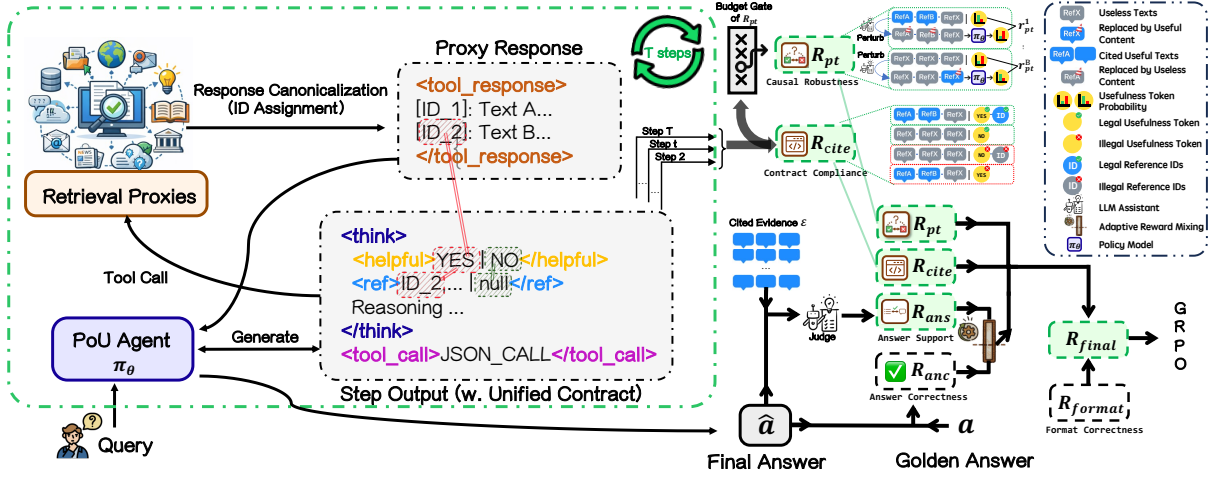


Figure 2: The Proof-of-Use Framework

learnable interface between probabilistic alignment and structured evidence-grounded reasoning.

When the model determines it has gathered sufficient evidence, it terminates the loop and emits:

<answer> ... </answer>

as the final response to the user.

2.3 Proof-of-Use Rewards

2.3.1 Unified Step protocol for Citation Reward

According to the *unified protocol* of each step, all proxy responses return dictionaries containing a unique ID, enabling deterministic auditing. We define the **Cite Reward** R_{cite}^t in step t :

$$R_{cite}^t = \begin{cases} +1, & \text{if parse_ok} \wedge \text{consistency_ok} \\ & \wedge \text{ids_valid}, \\ -1, & \text{otherwise.} \end{cases} \quad (1)$$

Here, $\text{parse_ok} = 1$ indicates syntactically valid and closed tags; $\text{consistency_ok} = 1$ requires ($\text{helpful=no} \Rightarrow \text{ref=null}$) and ($\text{helpful=yes} \Rightarrow \text{ref} \neq \text{null}$); $\text{ids_valid} = 1$ means all cited IDs exist in the proxy’s return list.

The rollout-level score is computed as the mean over all valid steps:

$$R_{cite} = \begin{cases} \frac{\sum_{t=2}^T R_{cite}^t}{T-1}, & T > 1, \\ 0, & T = 1, \end{cases}$$

where T is the total number of reasoning steps. This ensures that if the model answers directly without invoking any tool ($T=1$), no citation reward is given.

This design separates syntax compliance from citation consistency, ensuring that every reasoning step remains both *verifiable* and *traceable* to reasoning-evidence alignment.

2.3.2 Evidence-Sensitive Reward

To evaluate whether the model’s decision truly depends on the cited evidence rather than spurious features, we design a perturbation-based reward that measures the *sensitivity* of the model’s helpfulness prediction $p_\theta(\text{yes})$ to evidence change at step t :

$$R_{pt}^t = s_t \cdot (q' - q),$$

$$s_t = \begin{cases} -1, & \text{YES case (degrade cited evidence),} \\ +1, & \text{NO case (inject semantic lure).} \end{cases} \quad (2)$$

Specifically, for the YES case, we degrade the cited evidence set S_t by replacing their contents with topic-unrelated snippets, and observe the probability change $\Delta q = q' - q$, where $q = p_\theta(\text{yes} \mid \text{rsp}_{\text{real}})$, $q' = p_\theta(\text{yes} \mid \text{rsp}_{\text{pert}})$, rsp_{real} is the real tool responses and rsp_{pert} is the real response with partial replacement. A model that truly relies on the cited evidence should decrease its helpfulness confidence ($\Delta q < 0$) after the evidence is corrupted.

For the complementary NO case, we apply the opposite perturbation: when the model initially predicts the helpfulness token as no, we randomly select one document from the proxy-returned list and *replace its content with an LLM-generated snippet that is semantically relevant to the query* but not necessarily factual. This synthetic “semantic lure” mimics a plausible piece of supportive evidence. A model that truly relies on the cited evidence should

increase its helpfulness confidence ($\Delta q > 0$) after lure evidence is inserted.

Since each rollout may contain multiple steps, the Evidence-Sensitive reward is computed under a fixed budget gate B :

$$B = \min(T-1, B_{\max}, \sum_t \mathbb{I}[R_{cite}^{(t)}=1]),$$

where we randomly sample B steps from those with $R_{cite} = 1$ (from step 2 to T) to apply perturbation. We set $B_{\max} = 1$ for main experiments (thus $B = 1$), and study $B = 2$ in ablations to assess stability–cost trade-offs.

The final rollout-level Evidence-Sensitive reward is defined as the mean over the selected perturbation outcomes:

$$R_{pt} = \frac{1}{B} \sum_{t \in \mathcal{S}_B} R_{pt}^{(t)}. \quad (3)$$

In practice, by caching prefix KV states and original <helpful> logits, evidence perturbation is reduced to a tail-only forward pass.

2.3.3 Answer-Support Alignment Reward

We use two complementary training signals: an *Answer-Support Alignment* reward R_{ans} and a common *Answer-Correctness* reward R_{anc} . The large action space and delayed outcome rewards make evidence-grounded agents prone to shortcut guessing. To provide denser supervision, R_{ans} encourages the final answer to be supported by the retrieved and cited evidence.

We define the Answer-Support reward as:

$$R_{ans} = \begin{cases} 0, & \text{if no valid evidence is cited or } \mathcal{J}(q, \hat{a}, \mathcal{E}) = 0 \\ \mathcal{J}(q, \hat{a}, \mathcal{E}), & \text{otherwise,} \end{cases} \quad (4)$$

where $\mathcal{J}(q, \hat{a}, \mathcal{E}) \in [0, 1]$ is an external judge that measures whether the generated answer $\hat{a}$ is supported by the cited evidence $\mathcal{E}$. We instantiate $\mathcal{J}$ with the Faithfulness metric from Ragas (VibrantLabs, 2024). The correctness reward R_{anc} is computed by an LLM-as-Judge (Gu et al., 2025; Seed et al., 2025) between the predicted and gold answers.

Since R_{anc} is sparse and unstable in early training, PoU adopts an adaptive reward-mixing strategy that gradually shifts optimization from evidence support to answer correctness:

$$R_a^{s,i} = \alpha_s R_{ans}^{s,i} + (1 - \alpha_s) R_{anc}^{s,i}, \quad (5)$$

where α_s is a batch-level mixing weight at update step s . A larger α_s emphasizes evidence support, while a smaller α_s emphasizes final-answer correctness. We compute α_s adaptively from the moving average of batch-level correctness, with details provided in Appendix B.

2.3.4 Overall Objective

The final reward integrates the three signals into a unified rollout-level objective:

$$R_{\text{final}} = \begin{cases} \frac{R_{cite} + R_{pt} + R_a}{3}, & \text{if format is valid,} \\ -1, & \text{otherwise.} \end{cases} \quad (6)$$

where the overall format is considered valid only when all basic formats such as <think>, <tool_call> and <answer> pass the syntactic checks.

All components are jointly optimized using **Group Relative Policy Optimization (GRPO)**(Shao et al., 2024).

3 Experiments

3.1 Datasets and Metrics

We evaluate PoU on a mixed in-/out-of-domain (ID/OOD) open-domain QA benchmark suite. For ID evaluation, we use the official development splits of **HotpotQA** (Yang et al., 2018) and **2WikiMultihopQA** (Ho et al., 2020), from which all training data are drawn. For OOD evaluation, we use five datasets with diverse question styles and evidence structures: **Natural Questions (NQ)** (Kwiatkowski et al., 2019), **TriviaQA (TQ)** (Joshi et al., 2017), **MuSiQue** (Trivedi et al., 2022), **PopQA** (Mallen et al., 2023), and **Bamboogle** (Press et al., 2023).

For fair comparison, we sample 512 instances from each dev set of NQ, TQ, HotpotQA, 2Wiki, MuSiQue, and PopQA with a fixed global seed, and use all 125 Bamboogle dev items, yielding 3,197 evaluation questions. Only HotpotQA and 2Wiki are used for training; all other datasets are strict OOD test sets.

3.2 Baselines

We compare PoU with representative **deep research agents** spanning retrieval–reasoning RAG, RL-based search, multi-tool agents, and SFT-only training.

Table 1: In-domain evaluation on HotpotQA and 2Wiki. Results are reported with F1 and LLM-based evaluation (LM). Bold and underlined numbers indicate the best and second-best results, respectively.

Model	Src.	HotpotQA		2Wiki	
		F1	LM	F1	LM
<i>RAG</i>					
Routing RAG	Multi	30.8	36.0	14.4	18.2
All-in-one RAG	Multi	34.5	40.2	23.2	26.4
Search-o1*	Local	31.6	40.8	28.6	32.8
Search-o1	Web	33.0	42.4	30.9	37.7
<i>RL Deep Research</i>					
Search-R1-base	Local	55.9	63.0	44.6	47.9
Search-R1-inst.	Local	45.7	52.5	43.4	48.8
R1-Searcher	Web	44.8	53.1	59.4	65.8
CoA	Multi	–	43.9	–	50.2
DeepResearcher	Web	52.8	<u>64.3</u>	59.7	66.6
DeepResearcher+	Multi	50.3	61.7	<u>61.2</u>	<u>67.4</u>
PoU-SFT	Multi	42.2	45.5	37.6	41.9
<i>Ours</i>					
PoU	Multi	<u>54.9</u>	67.2	68.3	78.0

Retrieval–Reasoning RAG. We include **Routing RAG**, which dynamically selects among corpus, web, and KG retrievers, and **All-in-One RAG**, which exposes all retrievers through a unified interface. Both follow **ReAct** (Yao et al., 2023) with three reasoning–retrieval turns.

RL-based Search Agents. We compare with **Search-o1** (Li et al., 2025b), **Search-R1** (Jin et al., 2025), and **R1-Searcher** (Song et al., 2025), which learn multi-hop search behavior through RL-guided sub-query generation and evidence summarization.

Multi-Tool Agents. We include **Chain-of-Agent (CoA)** (Li et al., 2025a) and **DeepResearcher** (Zheng et al., 2025), two recent agentic systems for heterogeneous tool environments. For fair comparison, we also train **DeepResearcher+** from scratch with the same external tool set as PoU, isolating the effect of PoU’s evidence-grounded reward design.

SFT-only PoU. We include **PoU-SFT**, trained on PoU-formatted expert trajectories with the same model, tools, and data as PoU but without RL rewards, to isolate the effect of explicit process-level RL supervision.

3.3 Main Results

Across all seven datasets, covering both in-domain (HotpotQA, 2Wiki) and out-of-domain (NQ, TQ, MuSiQue, Bamboogle, PopQA) benchmarks, **Proof-of-Use (PoU)** consistently shows strongest performance. These gains extend beyond the training distribution: PoU achieves the strongest overall performance across the OOD benchmarks, with particularly consistent gains on the LM metric while remaining competitive or best on factual F1.

PoU achieves the best LM scores on both NQ and TQ, while also obtaining the best F1 on NQ and competitive F1 on TQ, which are in-domain datasets for **DeepResearcher**. This suggests that PoU’s improvements do not come from memorizing domain-specific knowledge, but from stronger *evidence–reasoning alignment*. By learning to ground answers in retrieved evidence, PoU transfers its reasoning strategy to unseen domains rather than relying on dataset-specific experience.

The comparison with **DeepResearcher+** further shows that naively expanding the action space with additional information sources can hurt standard RL agents. Despite access to more tools, DeepResearcher+ fails to effectively exploit the added knowledge and suffers performance drops, highlighting the need for specialized training for multi-source tool use. Meanwhile, **PoU-SFT**, although trained on LLM-guided expert trajectories, performs substantially worse and degrades more severely on OOD tasks. This indicates that PoU’s gains are not merely due to LLM-generated supervision, but to evidence-grounded RL that teaches decision-making logic beyond imitation.

We also find that **All-in-One RAG** consistently outperforms **Routing RAG**, consistent with observations from *RAG in the Wild* that heuristic routing without explicit optimization can be inefficient or even counterproductive. Although routing appears beneficial by narrowing the retrieval scope, it may prematurely discard useful evidence.

In our multi-source setting, routing becomes an additional latent decision: the agent must decide not only *what* to retrieve, but also *where* to retrieve from. As the number of sources grows, this enlarged action space makes exploration inefficient and unstable. PoU mitigates this issue through protocol-based training: citation validity and evidence–reasoning consistency provide step-level constraints that regularize search and prevent

Table 2: Performance comparison on five out-of-domain QA benchmarks: **NQ**, **TQ**, **MuSiQue**, **Bamboogle**, and **PopQA**. All results are reported with F1 and LLM-based evaluation (LM) metrics. Bold numbers indicate the best performance in each column. Underlined numbers indicate the second best.

Method	Env.	NQ		TQ		MuSiQue		Bamboogle		PopQA	
		F1	LM	F1	LM	F1	LM	F1	LM	F1	LM
<i>RAG</i>											
Routing RAG	Multi	34.8	42.9	51.0	59.6	6.5	8.2	17.0	17.4	35.1	39.9
All-in-one RAG	Multi	37.5	48.3	60.2	68.8	9.7	11.8	23.5	25.1	46.9	48.8
Search-o1*	Local	34.5	57.4	52.6	61.1	16.8	21.3	35.8	38.4	36.9	42.4
Search-o1	Web Search	32.4	55.1	58.9	69.5	14.7	19.7	46.6	53.6	38.3	43.4
<i>RL Deep Research</i>											
Search-R1-base	Local	<u>45.4</u>	60.0	71.9	76.2	26.7	27.5	56.5	57.6	43.2	47.0
Search-R1-instruct	Local	33.1	49.6	44.7	49.2	26.5	28.3	45.0	47.2	43.0	44.5
R1-Searcher	Web Search	35.4	52.3	73.1	79.1	22.8	25.6	64.8	65.6	42.7	43.4
CoA	Multi	—	43.9	—	63.3	—	22.3	—	49.6	—	46.5
DeepResearcher	Web Search	39.6	<u>61.9</u>	78.4	<u>85.0</u>	<u>27.1</u>	<u>29.3</u>	71.0	<u>72.8</u>	48.5	52.7
DeepResearcher+	Multi	37.9	58.7	76.0	82.1	23.0	26.5	68.9	70.4	<u>51.1</u>	<u>54.5</u>
PoU-SFT	Multi	35.9	56.0	51.3	60.6	11.2	12.4	33.3	35.2	37.2	41.4
PoU	Multi	45.5	67.2	<u>77.6</u>	89.6	29.0	34.1	<u>69.5</u>	74.6	55.7	61.2

divergence. As a result, PoU effectively leverages multi-source evidence while avoiding the instability of naive routing, yielding stronger performance and generalization.

3.4 Ablation Study

3.4.1 Rewards Ablation

Figure 3 shows the effect of different reward configurations. Increasing the perturbation budget to $B=2$ yields a more stable trajectory and higher final performance, confirming that the perturbation reward R_{pt} encourages the model to rely on genuinely supportive evidence rather than merely acknowledge tool responses. Removing adaptive reward mixing and using a static average of R_{ans} and R_{anc} slows early learning and lowers the final performance, showing that the two rewards play stage-dependent roles: R_{ans} provides dense grounding supervision early, while R_{anc} becomes more useful after evidence–reasoning consistency improves. Finally, removing R_{anc} causes unstable training and frequent late-stage collapse, indicating that answer–citation alignment is essential for smoothing policy optimization and grounding final answers in intermediate evidence.

3.4.2 Out-of-domain-tool (OOT) and negative tool (NT) Experiments

To evaluate generalization under simultaneous domain and tool-set shifts, we conduct additional experiments in an **out-of-domain (OOD)** and **out-**

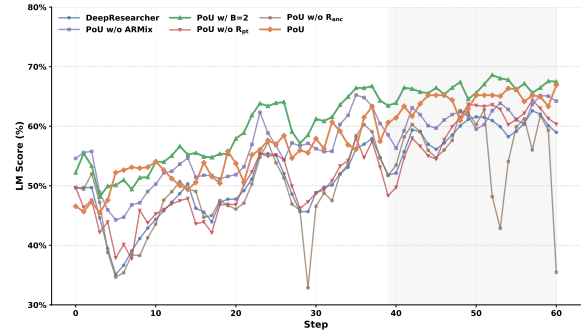


Figure 3: **Ablation study of PoU.** Comparison under different reward configurations: default perturbation budget $B=1$, extended $B=2$, and variants removing R_{pt} or R_{anc} .

of-domain-tool (OOT) setting. Specifically, we introduce tools that are never observed during training (a biomedical local search proxy (Local_Bio)). In addition, we introduce a **negative tool (NT)**, which is deliberately task-irrelevant, to reflect realistic scenarios where certain tools are inapplicable and to assess the model’s robustness to interference as well as its ability to select appropriate tools. Experiments are conducted on two biomedical QA benchmarks, **BioASQ** (Krithara et al., 2023) and **PQAREval** (Bařaragin et al., 2024).

As shown in Table 3, DeepResearcher+, despite being trained with an expanded toolset, performs consistently worse than its original 2-tool counterpart (DeepResearcher) not only in OOD settings but also under the more challenging **OOT**

Table 3: OOT evaluation on biomedical QA datasets. “w/ NT” denotes the inclusion of an additional negative tool.

Model	BioASQ	PQArefEval
Zero-shot	13.5	18.4
RAG	25.8	31.5
DeepResearcher+	37.1	42.4
DeepResearcher	38.9	42.1
DeepResearcher w/ NT	31.8	37.6
PoU w/ NT	43.9	48.3
PoU	45.6	48.7

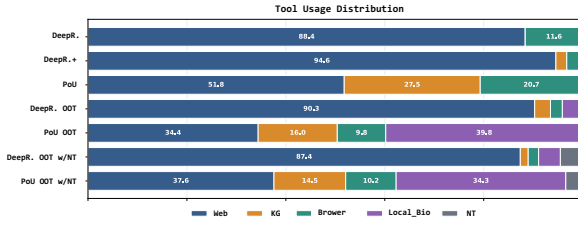


Figure 4: Tool-call ratio (%) distributions across models and settings. For visual clarity, values below 6% are not shown. For brevity, **DeepR.** denotes **DeepResearcher**.

regime across both biomedical datasets. Moreover, when an additional negative tool (NT) is introduced, the performance of DeepResearcher drops substantially, indicating a strong sensitivity to task-irrelevant tools. While PoU model achieves the best performance by a clear margin, demonstrating strong domain-level and tool-level generalization, while remaining robust even when the negative tool (NT) is introduced.

This result reveals an important phenomenon: as the number of available tools increases, the agent’s action space grows combinatorially. Without explicit process-level supervision, such an enlarged search space can easily lead to unstable policies or degenerate reasoning behaviors.

In contrast, through explicit evidence–reasoning–answer alignment training, the PoU training paradigm stabilizes learning under both domain and tool distribution shifts, substantially mitigating tool-call hacking. Crucially, PoU does not explicitly optimize for tool adaptation or robustness to task-irrelevant tools. Instead, by regulating a tight dependency between evidence usage, intermediate reasoning, and final answers, the training objective implicitly biases the trajectory toward tools that consistently support verifiable reasoning. **As a result, robustness to unseen or task-irrelevant tools emerges as a extra bonus from PoU.**

3.4.3 Tool-Call Behavior Analysis

Building on Figure 1, we further analyze how agents allocate tool calls under different task and tool environments. We sample 1,000 rollouts from each of seven open-domain test sets and two OOT benchmarks, and report the tool-call ratio distribution in Figure 4.

Under standard format- and answer-level rewards, DeepResearcher shows a strong tendency toward tool overuse and specialization collapse. When trained with an additional tool proxy, DeepResearcher+ overwhelmingly relies on Web Search (94.6%) while almost abandoning Web Browser (3.2%), despite the intended search–browse hierarchy. This indicates that the agent does not learn functional differentiation among tools; instead, it repeatedly invokes the tool that yields early non-zero rewards. Similarly, under OOT and NT settings, DeepResearcher maintains a polarized and rigid tool-call distribution, suggesting a brittle retrieval policy that fails to adapt to task or tool shifts. This provides another manifestation of *tool-call hacking*: the agent satisfies format and outcome rewards through repetitive tool invocation rather than evidence-grounded tool use.

In contrast, PoU exhibits a more balanced and context-sensitive tool-call distribution. Beyond improving robustness under OOT and NT settings, PoU induces a second emergent property: adaptive tool allocation across tasks and tool environments, indicating that evidence-grounded supervision helps agents learn heterogeneous retrieval utilities without explicitly optimizing for tool adaptation.

4 Conclusion

We identify *tool-call hacking* as a critical failure mode in RL-trained deep research agents and propose Proof-of-Use (PoU), an evidence-grounded framework that enforces dependency between retrieval, reasoning, and answers. Experiments show that PoU mitigates this issue, improves performance, and yields robust tool-use behavior under domain and tool shifts.

Acknowledgments

This research was sponsored by the National Natural Science Foundation of China (62572475, 61902209, 62377044), Intelligent Social Governance Platform, Major Innovation Planning Interdisciplinary Platform for the “Double-First Class”

Initiative, Renmin University of China, the Research Funds of Renmin University of China (22XNKJ15), and Beijing Nova Program.

Limitations

Our study focuses on retrieval-style tools and does not cover execution-style tools such as code interpreters or symbolic solvers, where the failure modes and supervision design may differ. The evaluation is conducted on open-domain English QA benchmarks and a biomedical domain; generalization to other languages, modalities, or domain-specific reasoning tasks remains to be verified. The Answer-Support Alignment reward relies on an external judge model (Ragas Faithfulness with GPT-4o-mini) and may inherit biases or errors of the judge. Finally, RL training of PoU is computationally expensive and the curriculum-style adaptive reward mixing introduces additional hyperparameters that may require tuning across new settings.

Ethical Considerations

This work focuses on improving the reliability and evidence grounding of tool-augmented research agents. We do not introduce new user-facing applications, collect personal data, or involve human subjects. All experiments are conducted on public QA benchmarks and controlled tool environments. Therefore, we do not identify direct ethical risks related to privacy, discrimination, or human-subject interaction.

One potential concern is that the perturbation procedure used for evaluation could be misused to construct misleading retrieval results. In this work, such perturbations are used only in a controlled experimental setting to test whether agents genuinely rely on evidence. We do not release perturbed evidence as factual content, and our method is designed to mitigate, rather than encourage, ungrounded or deceptive tool use.

References

- Rohan Ajwani, Shashidhar Reddy Javaji, Frank Rudzicz, and Zining Zhu. 2024. [Llm-generated black-box explanations can be adversarially helpful](#). *Preprint*, arXiv:2405.06800.
- Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. 2016. [Concrete problems in ai safety](#). *Preprint*, arXiv:1606.06565.
- Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, Carol Chen, Catherine Olsson, Christopher Olah, Danny Hernandez, Dawn Drain, Deep Ganguli, Dustin Li, Eli Tran-Johnson, Ethan Perez, and 32 others. 2022. [Constitutional ai: Harmlessness from ai feedback](#). *Preprint*, arXiv:2212.08073.
- Bojana Bašaragin, Adela Ljajić, Darija Medvecki, Lorenzo Cassano, Miloš Košprdić, and Nikola Milošević. 2024. [How do you know that? teaching generative language models to reference answers to biomedical questions](#). *Preprint*, arXiv:2407.05015.
- Mingyang Chen, Linzhuang Sun, Tianpeng Li, Haoze Sun, Yijie Zhou, Chenzheng Zhu, Haofen Wang, Jeff Z. Pan, Wen Zhang, Huajun Chen, Fan Yang, Zenan Zhou, and Weipeng Chen. 2025. [Research: Learning to reason with search for llms via reinforcement learning](#). *Preprint*, arXiv:2503.19470.
- Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, Saizhuo Wang, Kun Zhang, Yuanzhuo Wang, Wen Gao, Lionel Ni, and Jian Guo. 2025. [A survey on llm-as-a-judge](#). *Preprint*, arXiv:2411.15594.
- Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. [Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps](#). In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6609–6625, Barcelona, Spain (Online). International Committee on Computational Linguistics.
- 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.
- Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. 2017. [Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension](#). *Preprint*, arXiv:1705.03551.
- Anastasia Krithara, Anastasios Nentidis, Konstantinos Bougiatiotis, and Georgios Paliouras. 2023. Bioasqqa: A manually curated corpus for biomedical question answering. *Scientific Data*, 10(1):170.
- Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Matthew Kelcey, Jacob Devlin, Kenton Lee, Kristina N. Toutanova, Llion Jones, Ming-Wei Chang, Andrew Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association of Computational Linguistics*.
- Weizhen Li, Jianbo Lin, Zhuosong Jiang, Jingyi Cao, Xinpeng Liu, Jiayu Zhang, Zhenqiang Huang, Qianben Chen, Weichen Sun, Qiexiang Wang, Hongxuan

- Lu, Tianrui Qin, Chenghao Zhu, Yi Yao, Shuying Fan, Xiaowan Li, Tiannan Wang, Pai Liu, King Zhu, and 11 others. 2025a. [Chain-of-agents: End-to-end agent foundation models via multi-agent distillation and agentic rl](#). *Preprint*, arXiv:2508.13167.
- Xiaoxi Li, Guanting Dong, Jiajie Jin, Yuyao Zhang, Yujia Zhou, Yutao Zhu, Peitian Zhang, and Zhicheng Dou. 2025b. [Search-o1: Agentic search-enhanced large reasoning models](#). *ArXiv*, abs/2501.05366.
- Shengjie Ma, Chengjin Xu, Xuhui Jiang, Muzhi Li, Huaren Qu, Cehao Yang, Jiaxin Mao, and Jian Guo. 2025. [Think-on-graph 2.0: Deep and faithful large language model reasoning with knowledge-guided retrieval augmented generation](#). In *The Thirteenth International Conference on Learning Representations*.
- Adrien Majka and El-Mahdi El-Mhamdi. 2025. [The strong, weak and benign goodhart’s law. an independence-free and paradigm-agnostic formalisation](#). *Preprint*, arXiv:2505.23445.
- Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. [When not to trust language models: Investigating effectiveness of parametric and non-parametric memories](#). *Preprint*, arXiv:2212.10511.
- David Manheim and Scott Garrabrant. 2019. [Categorizing variants of goodhart’s law](#). *Preprint*, arXiv:1803.04585.
- Ted Moskovitz, Aaditya K Singh, DJ Strouse, Tuomas Sandholm, Ruslan Salakhutdinov, Anca Dragan, and Stephen Marcus McAleer. 2024. [Confronting reward model overoptimization with constrained RLHF](#). In *The Twelfth International Conference on Learning Representations*.
- Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A. Smith, and Mike Lewis. 2023. [Measuring and narrowing the compositionality gap in language models](#). *Preprint*, arXiv:2210.03350.
- ByteDance Seed, :, Jiaze Chen, Tiantian Fan, Xin Liu, Lingjun Liu, Zhiqi Lin, Mingxuan Wang, Chengyi Wang, Xiangpeng Wei, Wenyuan Xu, Yufeng Yuan, Yu Yue, Lin Yan, Qiying Yu, Xiaochen Zuo, Chi Zhang, Ruofei Zhu, Zhecheng An, and 255 others. 2025. [Seed1.5-thinking: Advancing superb reasoning models with reinforcement learning](#). *Preprint*, arXiv:2504.13914.
- Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. [Deepseekmath: Pushing the limits of mathematical reasoning in open language models](#). *Preprint*, arXiv:2402.03300.
- Joar Skalse, Nikolaus H. R. Howe, Dmitrii Krasheninnikov, and David Krueger. 2025. [Defining and characterizing reward hacking](#). *Preprint*, arXiv:2209.13085.
- Huatong Song, Jinhao Jiang, Yingqian Min, Jie Chen, Zhipeng Chen, Wayne Xin Zhao, Lei Fang, and Jirong Wen. 2025. [R1-searcher: Incentivizing the search capability in llms via reinforcement learning](#). *Preprint*, arXiv:2503.05592.
- Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. [Musique: Multi-hop questions via single-hop question composition](#). *Preprint*, arXiv:2108.00573.
- VibrantLabs. 2024. Ragas: Supercharge your llm application evaluations. <https://github.com/vibrantlabsai/ragas>.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. [Chain-of-thought prompting elicits its reasoning in large language models](#). *Preprint*, arXiv:2201.11903.
- Georg Wölflein, Dyke Ferber, Daniel Truhn, Ognjen Arandjelović, and Jakob Nikolas Kather. 2025. [Llm agents making agent tools](#). *Preprint*, arXiv:2502.11705.
- Ran Xu, Yuchen Zhuang, Yue Yu, Haoyu Wang, Wenqi Shi, and Carl Yang. 2025. [Rag in the wild: On the \(in\)effectiveness of llms with mixture-of-knowledge retrieval augmentation](#). *Preprint*, arXiv:2507.20059.
- Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. [Hotpotqa: A dataset for diverse, explainable multi-hop question answering](#). *Preprint*, arXiv:1809.09600.
- Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. [React: Synergizing reasoning and acting in language models](#). *Preprint*, arXiv:2210.03629.
- 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.

A Scope of Tools

In this work, we define “tools” as black-box(Ajwani et al., 2024) *proxies* that interface with external knowledge sources: retrieval-style tools including but not limited to open-web search, internal document collections, structured APIs, or specialized knowledge bases. This reflects a common abstraction in RAG systems, where tools are used to fetch potentially relevant information conditioned on the current query or reasoning context. We **do not** extend our analysis to coding or mathematical tools(Wölflein et al., 2025).

This choice is deliberate: coding and math tools primarily stress symbolic reasoning and execution

correctness, where tool outputs are directly consumed as final answers and failures are typically observable (e.g., wrong values or execution errors). In contrast, retrieval tools serve as intermediate evidence providers. Their effectiveness depends on problem analysis, retrieval planning, and information integration, creating a setting where agents can satisfy tool-call specifications without causally relying on the returned evidence. **This makes retrieval-centric tools particularly susceptible to tool-call hacking** (see Section 2.1 for a formal definition), motivating our PoU rewards.

B Details of Adaptive Reward Mixing

This section describes how PoU computes the adaptive mixing weight α_s in Eq. 5. At each update step s , we first estimate the current batch-level answer correctness:

$$\hat{c}_s = \frac{1}{B} \sum_{i=1}^B R_{\text{anc}}^{s,i}, \quad (7)$$

where B denotes the batch size and $R_{\text{anc}}^{s,i} \in [0, 1]$ is the raw correctness reward of rollout i .

To reduce variance, we maintain an exponential moving average:

$$\bar{c}_s = \beta \bar{c}_{s-1} + (1 - \beta) \hat{c}_s, \quad \beta \in [0, 1). \quad (8)$$

The mixing weight is then computed using a monotone logistic gate:

$$\alpha_s = \sigma(\kappa(\tau - \bar{c}_s)), \quad (9)$$

where τ is a correctness threshold and κ controls the sharpness of the transition. When model correctness is low, α_s remains large, placing more weight on answer-support alignment. As training progresses and $\bar{c}_s$ increases, α_s decreases smoothly, shifting the optimization focus toward answer correctness.

Property of adaptive mixing. The adaptive schedule has three useful properties. First, since $\alpha_s = \sigma(\kappa(\tau - \bar{c}_s)) \in (0, 1)$ and both R_{ans} and R_{anc} are normalized to $[0, 1]$, the mixed reward is a bounded convex combination and therefore does not change the reward scale. Second, α_s decreases monotonically with the smoothed correctness estimate $\bar{c}_s$, because

$$\frac{\partial \alpha_s}{\partial \bar{c}_s} = -\kappa \alpha_s (1 - \alpha_s) < 0. \quad (10)$$

Thus, the reward naturally shifts from evidence alignment to answer correctness as training progresses. Third, the EMA update

$$\bar{c}_s - \bar{c}_{s-1} = (1 - \beta)(\hat{c}_s - \bar{c}_{s-1}) \quad (11)$$

limits the influence of any single noisy batch, preventing abrupt switches in the training objective. Together, these properties make the schedule a smooth curriculum: early training receives dense grounding supervision, while later training is increasingly optimized toward final task correctness.

C Related Works

C.1 Reinforcement Learning based Deep Research Agents

Large language models (LLMs) are limited by the static knowledge embedded in their parameters, which often leads to hallucinations when answering time-sensitive or knowledge-intensive queries. To address this limitation, RAG systems have evolved from static heuristics to adaptive reasoning agents. Recent works apply reinforcement learning (RL) to enable LLMs to autonomously plan reasoning steps, decide when and how to invoke retrieval tools, and verify information across heterogeneous sources (Zheng et al., 2025; Song et al., 2025).

Search-R1 (Jin et al., 2025) introduces a closed-loop *reason-and-search* framework, allowing models to dynamically invoke search engines during answer generation. R1-Searcher (Song et al., 2025) further adopts a two-stage RL training paradigm that substantially improves autonomous search behaviors. ReSearch (Chen et al., 2025) treats search as an integral part of the reasoning chain without relying on supervised intermediate reasoning labels. Beyond controlled environments, DeepResearcher (Zheng et al., 2025) extends RL training to the open Internet, enabling end-to-end optimization in real-world web settings.

C.2 Reward Hacking in Reinforcement Learning

Reward hacking (a.k.a. specification gaming) refers to agents exploiting misspecified objectives to achieve high reward while violating the designer’s true intent. It was systematically highlighted as a core AI-safety challenge in *Concrete Problems in AI Safety* and later analyzed through empirical taxonomies of gaming behaviors (Amodei et al., 2016). Formal treatments

attribute reward hacking to proxy misspecification, characterizing conditions under which objectives become hackable and proposing principled defenses; related work studies *reward tampering*, where agents manipulate the reward channel itself (Skalse et al., 2025). These failure modes are closely connected to Goodhart’s Law—over-optimizing a proxy causes it to decouple from the true goal—and to goal misgeneralization, where models pursue unintended objectives despite correct training feedback (Manheim and Garrabrant, 2019).

In LLM post-training, reward-model optimization introduces additional risks: excessive optimization against learned reward models can degrade human-judged quality as policies overfit spurious reward artifacts. Prior work proposes constrained RLHF and composite reward formulations to mitigate such effects, while iterative RLHF can empirically reduce over-optimization across rounds, albeit with diminishing returns (Moskovitz et al., 2024). Process-based supervision, such as process reward models (PRMs) or step-wise Chain-of-Thought (Wei et al., 2023) rewards, offers denser and potentially less gameable signals; however, monitoring intermediate reasoning also opens new attack surfaces, enabling models to obfuscate or game the monitor itself. This has motivated hybrid objectives that jointly supervise both outcomes and processes. Orthogonal alignment approaches, such as Constitutional AI, further reduce reliance on noisy human labels through principle-guided self-critique, reframing supervision to limit exploitable gaps in reward design (Bai et al., 2022).

Modern RL-trained retrieval–reasoning agents further expand the action space by allowing policies to invoke external tools (e.g., search engines, web browsers, or knowledge-base APIs), thereby influencing the information sources used to compute downstream rewards. In this setting, reward hacking manifests in subtle and prevalent forms. We identify and formalize these failures as *tool-call hacking*, and this paper aims to provide a principled solution to this problem.

D Implementation Details

We use GPT-4o-mini as the external judge model for evaluating answer–evidence alignment and factual consistency. All other agentic modules (browser, KG proxy, perturbations LLM) are implemented with Qwen-3-8B, deployed via vLLM

inference servers. The training backbone model is Qwen-2.5-7B-Instruct. Prior to reinforcement learning, supervised fine-tuning is performed with a batch size = 256 for 3 epochs. For reinforcement learning, we adopt VERL with the GRPO algorithm. Each update uses 8 parallel rollouts, distributed across $32 \times$ A100 GPUs. The R_{pt} budget B is 1. The maximum reasoning step is set to 10, and each retrieval call returns the top-5 documents or chunks per query. The KG proxy retrieves one-hop subgraph information from Wikidata and Wikipedia by ToG-2 (Ma et al., 2025) system, providing semantically condensed neighborhood summaries as context. In the OOT experiments, we additionally introduce a local biochemical Proxy, which is built from a biochemical corpus derived from PubMed and serves as a local retrieval interface, serving as the fixed local knowledge base for domain-specific retrieval. The local_proxy is turn-off for non-medical domain tasks.

E Extreme Case

To expose tool-call hacking and evaluate resistance to spurious tool responses, we intervene in tool-call by randomly replacing the content of the corresponding `<tool_response>` with the token “content”, while preserving the response structure. Due to space constraints, Table 4 only presents the `<think>` component of a single reasoning step under spurious tool responses. Nevertheless, this simplified illustration faithfully reflects the behavior patterns we consistently observe in full rollouts. In practice, when the tool response content is corrupted while preserving its structure, **Deep-Researcher** often produces seemingly coherent intermediate reasoning that implicitly assumes the existence of valid evidence, despite the absence of any usable information in the tool output. In contrast, **PoU** explicitly recognizes the lack of supporting evidence at the step level, marks the tool call as unhelpful, and proactively resorts to alternative tools. This contrast is robust across queries and steps, and the extreme case shown here serves as a minimal yet representative example of the underlying phenomenon.

F Proxy settings

- **Web Search Proxy.** This module enables the agent to access open-domain information through keyword-based retrieval. Given a textual query, it returns a structured list of web-

Table 4: An Extreme Case

Observation:<tool_response> content </tool_response>	
Method	
DeepResearcher	<think> I found the information about “xxx”. Now I will check “xxx”. </think>
PoU	<think> <helpful>no</helpful> <ref>null</ref> No relevant information found. I will try another tool. </think>

page entries, each containing a title, URL, and snippet. For controlled evaluation, the current configuration retrieves a fixed top- k set of documents per query, ensuring reproducible coverage across runs.

- **Web Browsing Proxy.** This proxy performs in-depth inspection of given URLs of retrieved webpages and extracts content relevant to the current query. It returns concise, structured summaries to the reasoning model as evidence for downstream use.
- **Local Search Proxy.** This proxy accesses pre-built, fixed knowledge repositories, typically used in private or professional domains such as academic papers, technical reports, enterprise data, or financial documents. Compared with web search, these repositories feature higher knowledge density and lower noise, making them particularly suitable for deeper domain-specific tasks.
- **Knowledge Graph Proxy.** This proxy serves as an interface to structured, graph-based knowledge stores. Rather than employing a query-generation agent that issues executable graph queries, we adopt a semantic KG agent that abstracts subgraphs into concise, natural-language summaries. This choice prioritizes robustness and generality across heterogeneous graph schemas, allowing the proxy to be more compatible with the model’s reasoning process. The KG proxy thus functions as a lightweight retrieval utility supplying structured yet readable context when graph knowledge is required.

G Adversarial Perturbation Prompts

This appendix provides the prompts used to construct perturbed evidence for evaluating whether agents genuinely rely on retrieved evidence. We consider two perturbation directions: transforming irrelevant evidence into seemingly relevant evidence, and transforming relevant evidence into irrelevant or minimally useful evidence.

Evidence Perturbation Prompt: Irrelevant → Relevant

You are an expert in adversarial rewriting for information retrieval evaluation.

Your task is to modify the content of a retrieved result so that it appears relevant and credible for a given user query, without fabricating unsupported factual claims.

The retrieved result is provided as a JSON object representing an element returned by a retrieval system. The structure of this JSON object may vary depending on the retrieval method. Your job is to perturb the **CONTENT** of the object so that it appears related to the query, while strictly preserving the original JSON structure and factual content.

Input

Query: «QUERY»

RetrievedObject (JSON):

«JSON_OBJECT»

Instructions

1. Preserve the exact JSON structure:
 - Do NOT add or remove fields.
 - Do NOT rename fields.
 - Do NOT change the nesting structure.
 - The output must remain a valid JSON object with the exact same schema.
2. Only modify natural-language content fields, such as titles, snippets, abstracts, passages, bodies, or descriptions.
3. Do NOT modify identifiers, URLs, timestamps, source names, retrieval

scores, embeddings, or other metadata fields, even if they are represented as strings.

4. Rewrite or reorganize the textual content so that the object appears more relevant and useful for answering the query.

5. You may:

- emphasize existing contextual cues;
- rephrase or reorganize the original content;
- add query-oriented framing without introducing unsupported factual claims;
- connect the content to the query topic when such a connection can be plausibly inferred.

6. Do NOT fabricate unsupported factual claims. In particular, do not introduce new entities, events, dates, numbers, citations, experimental results, or causal claims that are not present in or directly inferable from the original object. Any added framing must only increase perceived relevance without changing the factual content of the original object.

7. If it is unclear whether a field is content or metadata, keep it unchanged.

8. The result should look like a natural retrieval result that is relevant to the query.

9. The rewritten content should be persuasive, coherent, and believable, while remaining faithful to the original object's factual content.

10. Do NOT mention that this is a rewriting task or that the document was originally unrelated.

Output requirements

Return ONLY the rewritten JSON object. Do NOT wrap the output in Markdown code fences.

The JSON structure must exactly match the input structure, with only the

natural-language content modified to increase perceived relevance to the query.

Evidence Perturbation Prompt: Relevant → Irrelevant

You are an expert in adversarial rewriting for information retrieval evaluation.

Your task is to modify the content of a retrieved result so that it appears unrelated, tangential, or minimally useful for answering a given user query.

The retrieved result is provided as a JSON object representing an element returned by a retrieval system. The structure of this JSON object may vary depending on the retrieval method.

Your job is to perturb the **CONTENT** of the object so that it becomes less useful for answering the query, while strictly preserving the original JSON structure.

Input

Query: «QUERY»

RetrievedObject (JSON):

«JSON_OBJECT»

Instructions

1. Preserve the exact JSON structure:

- Do NOT add or remove fields.
- Do NOT rename fields.
- Do NOT change the nesting structure.
- The output must remain a valid JSON object with the exact same schema.

2. Only modify natural-language content fields, such as titles, snippets, abstracts, passages, bodies, or descriptions.

3. Do NOT modify identifiers, URLs, timestamps, source names, retrieval scores, embeddings, or other metadata fields, even if they are represented as strings.

4. Rewrite the content so that it becomes unrelated, generic, tangential, underspecified, or focused on a different topic than the query.

5. You may use the following strategies:

- shift the topic to a different domain;
- replace query-specific concepts with broader or less informative themes;
- generalize the content into vague background discussion;
- emphasize tangential or peripheral information;
- introduce neutral or alternative contexts.

6. The rewritten content should still read naturally and coherently, as if it were a legitimate retrieval result, but it should not be useful for answering the query.

7. Avoid mentioning the query topic directly unless necessary, and minimize conceptual overlap with the query.

8. Do NOT mention that this is a rewriting task.

9. If it is unclear whether a field is content or metadata, keep it unchanged.

Output requirements

Return ONLY the rewritten JSON object. Do NOT wrap the output in Markdown code fences.

The JSON structure must exactly match the input structure, with only the natural-language content modified to reduce usefulness for answering the query.

Questions, TriviaQA, MuSiQue, PopQA, Bamboogle, BioASQ, PQArefEval, Ragas, and the compared baseline systems, following their original licenses or terms of use. All artifacts are used only for research evaluation, which is consistent with their intended academic and benchmark use. We cite the original creators of these artifacts and do not redistribute any restricted data, proprietary resources, or derived perturbed evidence as factual content. We use publicly available QA benchmarks and do not collect new user data or human-subject data. For artifacts created in this work, including PoU-formatted trajectories and perturbation-based evaluation examples, the intended use is research on evidence-grounded tool-augmented agents. These artifacts should not be used to generate or disseminate misleading evidence outside controlled research settings.

I Use of AI Assistants

AI assistants were used only for typo checking, grammar correction, and language polishing. They were not used to generate ideas, conduct experiments, analyze results, or formulate scientific claims. All content was reviewed and verified by the authors.

H Artifact Licenses, Intended Use, and Data Documentation

We use publicly available datasets and artifacts, including HotpotQA, 2WikiMultihopQA, Natural