INFORMATION GAIN-BASED POLICY OPTIMIZATION: A SIMPLE AND EFFECTIVE APPROACH FOR MULTI-TURN LLM AGENTS

Guoqing Wang[1*], Sunhao Dai[2*], Guangze Ye[3*], Zeyu Gan[2], Wei Yao[2], Yong Deng[1], Xiaofeng Wu[1], and Zhenzhe Ying[1]
[1]Ant Group [2]Renmin University of China [3]Individual Author

GitHub: https://github.com/GuoqingWang1/IGPO

ABSTRACT

Large language model (LLM)-based agents are increasingly trained with reinforcement learning (RL) to enhance their ability to interact with external environments through tool use, particularly in search-based settings that require multi-turn reasoning and knowledge acquisition. However, existing approaches typically rely on outcome-based rewards that are only provided at the final answer. This reward sparsity becomes particularly problematic in multi-turn settings, where long trajectories exacerbate two critical issues: (i) advantage collapse, where all rollouts receive identical rewards and provide no useful learning signals, and (ii) lack of fine-grained credit assignment, where dependencies between turns are obscured, especially in long-horizon tasks. In this paper, we propose Information Gain-based Policy Optimization (IGPO), a simple yet effective RL framework that provides dense and intrinsic supervision for multi-turn agent training. IGPO models each interaction turn as an incremental process of acquiring information about the ground truth, and defines turn-level rewards as the marginal increase in the policy's probability of producing the correct answer. Unlike prior process-level reward approaches that depend on external reward models or costly Monte Carlo estimation, IGPO derives intrinsic rewards directly from the model's own belief updates. These intrinsic turn-level rewards are combined with outcome-level supervision to form dense reward trajectories. Extensive experiments on both in-domain and out-of-domain benchmarks demonstrate that IGPO consistently outperforms strong baselines in multi-turn scenarios, achieving higher accuracy and improved sample efficiency.

1 INTRODUCTION

Large language model (LLM)–based agents are increasingly endowed with the ability to interact with external environments through tool use (Zhang et al., 2025a; Huang et al., 2025; Li et al., 2025c), a capability often regarded as a critical step toward building general-purpose autonomous intelligent systems (Gutierrez et al., 2023; Qu et al., 2025). For example, web search (Zhang et al., 2025b; Qi et al., 2024), one of the most fundamental tools, enables agents to access up-to-date large-scale knowledge that substantially improves their capacity to solve complex, knowledge-intensive tasks (Ning et al., 2025). Through iterative interaction with the external environment, agents can gradually acquire missing information and refine their reasoning toward solving the target query.

To equip general-purpose LLMs with such agentic capabilities, early efforts primarily relied on prompt-based workflows (Li et al., 2025b; Wang et al., 2024a; Zheng et al., 2024), which allowed tool use without additional training but often suffered from poor generalization. More recent studies have explored supervised fine-tuning (SFT) (Wang et al., 2024b) and reinforcement learning (RL) (Jin et al., 2025; Song et al., 2025a; Zheng et al., 2025b) to explicitly incentivize tool use, achieving markedly better performance. In particular, Group Relative Policy Optimization (GRPO) (Shao et al., 2024)–style methods have emerged as the dominant approach for training agentic LLMs. In this paradigm, a group of rollouts is generated for each query under the current

*Equal contribution.

policy, and outcome-based rewards, typically defined by the correctness of the final answer against the ground truth, are used to construct group-relative advantages that drive policy optimization.

Despite their simplicity and effectiveness on relatively easy tasks, outcome rewards suffer from an inherent limitation: they are **sparse** (Zhang et al., 2025c), since supervision is provided only at the final answer. This sparsity becomes particularly detrimental in multi-turn agentic settings, where long trajectories exacerbate the problem in two ways. **First**, sparse rewards frequently lead to *advantage collapse*: when sampled rollouts yield the same answer (e.g., all wrong or all right), all rollouts in the group receive identical outcome rewards, yielding zero group-relative advantages. As shown in Figure 1, a substantial portion of training iterations suffer from this issue, especially for smaller models, which struggle more with complex queries. **Second**, outcome-only supervision fails to provide fine-grained credit assignment. In multi-turn scenarios, later turns are tightly dependent on earlier ones: a reasoning or tool call of the current turn may be correct but rendered useless by prior mistakes, or conversely, early successes may be negated by subsequent errors. Such dependencies are easily obscured under outcome-only rewards, particularly in multi-hop tasks that require long-horizon reasoning.

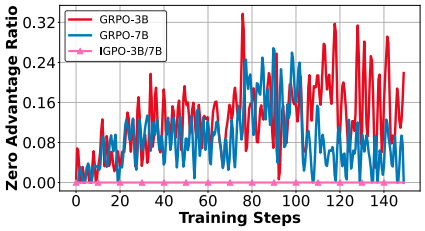


Figure 1: Proportion of zero-advantage groups during training—IGPO vs. GRPO on Qwen2.5-7B/3B-Instruct.

Several recent approaches have attempted to mitigate these issues by introducing process-level rewards. One line of work leverages external oracle knowledge or reward models to judge intermediate steps (Wang et al., 2025; Feng et al., 2025), but this strategy is costly to obtain and risks introducing additional bias. Another line relies on Monte Carlo simulations to estimate step values (Wang et al., 2023; Zuo et al., 2025; Zhang et al., 2025c), yet these methods suffer from high variance unless a large number of samples are collected. Overall, both directions face challenges in scalability and fail to provide simple and stable supervision, underscoring the need for an intrinsic and reliable process-level reward design.

To address these challenges, we propose Information-Gain-based Policy Optimization (IGPO), a simple but effective reinforcement learning framework that provides stable and intrinsic supervision for multi-turn agent training. The key intuition is to model each agent–environment interaction turn as an incremental process of acquiring information about the ground truth. Specifically, at every turn, IGPO computes the policy's probability of producing the correct answer and defines the turn-level reward as the marginal increase in this probability compared to the previous state. This information gain reward offers ground-truth-aware feedback at every turn, in contrast to outcome rewards that only supervise the final answer. While turn-level rewards ensure dense and stable supervision, the outcome reward remains essential to anchor training to the final task objective. To combine these strengths, IGPO also integrates the outcome reward with the sequence of turn-level rewards, forming a dense reward trajectory for each rollout. To further stabilize training, we normalize rewards within groups and propagate them with discounted accumulation, enabling turn-level advantage estimation that captures long-horizon dependencies. Finally, IGPO optimizes the policy with a GRPO-style surrogate objective, replacing rollout-level advantages with our turn-level ones.

To evaluate the effectiveness of IGPO, we conduct extensive experiments on both in-domain and out-of-domain benchmarks with search-based agents. Results show that IGPO consistently outperforms strong baselines, delivering substantial gains in both answer accuracy and sample efficiency. Our main contributions can be summarized as follows: (1) We analyze the phenomenon of advantage collapse in outcome-reward–based optimization, and reveal the inefficiency of existing process-level rewards due to reliance on external knowledge or high-variance estimation. (2) We propose IGPO, a simple yet effective policy optimization framework that leverages turn-level information gain to provide dense, ground-truth-aware supervision while preserving outcome-level alignment. (3) Comprehensive experiments demonstrate that IGPO outperforms strong baselines across multiple benchmarks and significantly improves sample efficiency, especially for smaller models.

2 PRELIMINARIES

In this section, we present the standard multi-turn agentic RL pipeline, illustrated with a search agent as a representative example.

2.1 TASK FORMULATION

Let $\mathcal{D} = \{(q, a)\}$ denote a dataset of question–answer pairs, and let $\mathcal{E}$ represent an external tool (e.g., a web search engine). The goal of the agent is to solve question q by generating a rollout $o = (\tau_1, \tau_2, \ldots, \tau_T)$ through iterative interaction with the environment via tool $\mathcal{E}$, where T is the total number of interaction turns. The last turn τ_T is the *answer turn* that outputs a rationale-then-final answer sequence, while all previous turns involve reasoning and tool interaction. Specifically, for $t < T$, each turn τ_t is defined as a triple consisting of `[think]`, `[tool call]`, and `[tool response]`. The `[think]` step compels the agent to reason explicitly before acting, and each reasoning process is wrapped in a `<think></think>` tag following the DeepSeek-R1 setting (Guo et al., 2025). The `[tool call]` step invokes the external tool $\mathcal{E}$ by producing a structured request, typically JSON-formatted and wrapped in a dedicated tag (e.g., `<search>search query</search>` for web search). The `[tool response]` step then returns structured outputs from $\mathcal{E}$, such as webpage snippets with titles, URLs, and text when using a web search engine tool, enclosed in `<tool_response>retrieved documents</tool_response>` tags. In the final turn, after a `[think]` step, the agent generates its answer within the `<answer></answer>` tag, and this content is extracted as the trajectory's final prediction $\hat{a}$, which is expected to correctly address the input query q. This agent-environment interaction is illustrated at the bottom of Figure 2.

2.2 AGENTIC REINFORCEMENT LEARNING PIPELINE

Policy Optimization. Agentic RL typically adopts policy-gradient methods to optimize the agent policy π_θ. A common approach is *Group Relative Policy Optimization* (GRPO) (Shao et al., 2024), which removes the need for an explicit critic by normalizing returns within each sampled group of rollouts. Formally, given an actor model π_θ, a group of G rollouts $\{o_i\}_{i=1}^{G}$ is sampled from old policy $\pi_{\theta_{\text{old}}}(\cdot \mid q)$ for each input $(q, a) \sim \mathcal{D}$. The policy is then optimized by maximizing the clipped surrogate objective with KL regularization:

$$\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}_{(q,a)\sim\mathcal{D},\, \{o_i\}\sim\pi_{\theta_{\text{old}}}(\cdot|q)} \Bigg[\frac{1}{G} \sum_{i=1}^{G} \frac{1}{|o_i|} \sum_{t=1}^{|o_i|} \min \Bigg(\frac{\pi_\theta(o_{i,t} \mid q, o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t} \mid q, o_{i,<t})} \widehat{A}_i, \\ \text{clip}\Bigg(\frac{\pi_\theta(o_{i,t} \mid q, o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t} \mid q, o_{i,<t})}, 1-\epsilon, 1+\epsilon \Bigg) \widehat{A}_i \Bigg) - \beta\, \mathbb{D}_{\text{KL}}(\pi_\theta \,\|\, \pi_{\text{ref}}) \Bigg], \tag{1}$$

where $\widehat{A}_i = \frac{r_i - \text{mean}(r_1, r_2, \cdots, r_G)}{\text{std}(r_1, r_2, \cdots, r_G)}$ is the normalized group-relative advantage for the i-th rollout and r_i is the outcome reward of the i-th rollout. ϵ is the clipping ratio, and β controls the KL penalty that regularizes the updated policy toward the reference model π_{ref}. During optimization, gradients are applied only to decision tokens (reasoning, tool calls, answers), while tool responses from the external environment are masked out.

Reward. During training, the agent receives a scalar reward r for each rollout o, which provides the optimization signal. Prior work usually adopts an outcome-based answer reward combined with a format penalty:

$$r = \begin{cases} \text{F1}(\hat{a}, a) = \frac{2\,|\hat{a} \cap a|}{|\hat{a}| + |a|} \in [0, 1], & \text{if the output is in valid format,} \\ \lambda_{\text{fmt}}, & \text{otherwise,} \end{cases} \tag{2}$$

where $\hat{a}$ is the predicted final answer for each rollout, a is the ground-truth answer, and $\text{F1}(\hat{a}, a) \in [0, 1]$ denotes the word-level F1 score between the two. If the output violates the required schema (e.g., missing tags or malformed JSON), a negative constant $\lambda_{\text{fmt}} < 0$ is assigned as a penalty. Thus, the outcome reward provides a correctness signal aligned with evaluation metrics, while the format penalty enforces the structural validity of outputs.

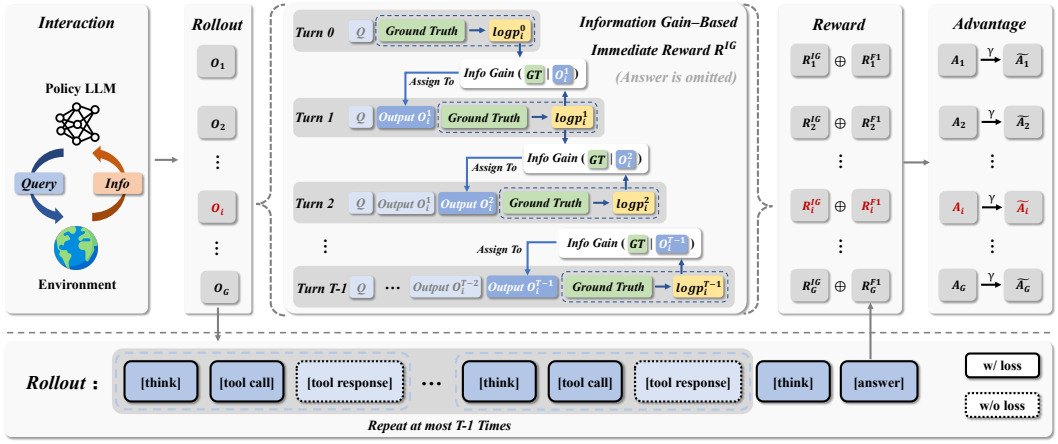


Figure 2: The training pipeline of IGPO. (Upper) Turn-level information gain rewards are computed by measuring changes in ground-truth probability and combined with the outcome reward to derive discounted advantages. (Lower) Each rollout contains at most $T-1$ interaction turns, where each turn includes a reasoning step, a tool call, and the returned tool response, followed by a final answer turn. During optimization, the loss on tool response is masked out.

3 Information Gain-based Policy Optimization

In this section, we first illustrate our motivation and then provide a detailed introduction to our proposed information gain-based policy optimization, whose overall framework is shown in Figure 2.

3.1 Motivation

While outcome-based reinforcement learning has been effective in single-turn tasks, directly extending it to multi-turn agentic settings such as search agents faces critical limitations. In the standard GRPO framework (Eq. 1), each rollout o_i receives a scalar reward r_i computed from the final answer $\hat{a}_i$. For complex queries, however, it is often the case that all G rollouts fail to produce the correct answer, resulting in uniformly zero rewards; conversely, for simple queries, all rollouts may produce the same correct answer, leading to the same issue. In these cases, the normalized group-relative advantages $\{\widehat{A}_i\}$ collapse to near zero, and the entire sample provides almost no learning signal. We refer to this phenomenon as *advantage collapse*. Moreover, such outcome-only supervision lacks fine-grained credit assignment across turns. In multi-turn scenarios, later decisions critically depend on earlier ones: a tool call may be effective yet rendered useless by prior retrieval errors, or early reasoning may be correct but overshadowed by subsequent mistakes. Such dependencies are obscured under single outcome rewards, making it difficult for the policy to distinguish productive reasoning from uninformative or misleading turns.

To mitigate this issue, we introduce Information-Gain-based Policy Optimization (IGPO). The key idea is to exploit the multi-turn structure of agentic rollouts and treat each turn as an opportunity to acquire additional evidence toward the ground truth. At every turn, IGPO measures the increase in the policy's confidence of generating the correct answer, which we defined as the *information gain* of this turn and uses this as the turn-level reward. By rewarding turn-level information gain, IGPO supplies denser and more fine-grained supervision, especially at early training stages. We further present a theoretical analysis in Appendix A, which intuitively explains why IGPO effectively addresses the limitations of sparse outcome rewards in multi-turn scenarios. Since the information gain is defined with respect to the ground-truth answer and computed under teacher forcing, it always produces a valid signal, ensuring that every sample contributes to learning even when no rollout produces a fully correct final answer.

3.2 INFORMATION GAIN-BASED TURN-LEVEL REWARD

Turn-level Reward. We view multi-turn agent–environment interaction as a process of *incrementally acquiring information about the ground truth*. To capture this intuition, we propose an intrinsic *information gain-based reward*. At each turn, we evaluate the policy's probability of generating the ground-truth answer and define the reward as the difference between consecutive states. We call this the *information gain reward*, as it measures the *marginal increase in posterior probability mass assigned to the ground truth* induced by the current turn.

Formally, let $a = (a_1, \ldots, a_L)$ denote the ground-truth answer tokens. For the t-th turn in the i-th rollout, the probability of a under the current policy π_θ is computed as

$$\pi_\theta(a \mid q, o_{i,\le t}) = \exp\left(\frac{1}{L}\sum_{j=1}^{L}\log \pi_\theta(a_j \mid q, o_{i,\le t}, a_{<j})\right), \tag{3}$$

where $o_{i,\le t}$ denotes the prefix of rollout o_i up to turn t. Then the immediate reward [*] for turn t is

$$r_{i,t} = \mathrm{IG}(a \mid q, o_{i,t}) = \pi_\theta(a \mid q, o_{i,\le t}) - \pi_\theta(a \mid q, o_{i,\le t-1}), \qquad 1 \le t < T. \tag{4}$$

In practice, the ground-truth answer a is wrapped in the same schema as a predicted answer to ensure consistency with rollout formatting, e.g., `<think>Now there's enough information to answer</think><answer>Ground Truth a</answer>`.

This turn-level reward has two desirable properties: (1) *Ground-truth awareness*: the reward increases when the action raises the policy's confidence in the correct answer, and decreases otherwise; (2) *Dense supervision*: the reward is defined for every sample, even when no rollout yields a correct answer, thereby alleviating reward sparsity and avoiding advantage collapse.

Integrating Outcome and Turn-level Rewards. For each rollout $o_i = (\tau_{i,1}, \ldots, \tau_{i,T})$ where the last turn $\tau_{i,T}$ is the answer turn producing $\hat{a}_i$, we can construct a length-T reward vector $\mathbf{r}_i = (r_{i,1}, r_{i,2}, \ldots, r_{i,T})$. For $t < T$, the turn reward is the information gain $r_{i,t} = \mathrm{IG}(a \mid q, o_{i,t})$ defined in Section 3.2. For the answer turn $t = T$, the reward $r_{i,T}$ follows the outcome-based formulation in Eq. 2. This yields a dense per-turn supervision signal that combines intrinsic information gains for intermediate turns with a final extrinsic correctness signal at the answer turn.

3.3 POLICY OPTIMIZATION WITH TURN-LEVEL ADVANTAGE

Turn-level Advantage Estimation. Given a rollout $o_i = (\tau_{i,1}, \ldots, \tau_{i,T})$, each turn $\tau_{i,t}$ is associated with a reward $r_{i,t}$ as defined in Section 3.2. To make rewards comparable across turns and trajectories, we first aggregate all rewards in the group:

$$\mathbf{R} = \{\, r_{i,t} :\ i = 1, \ldots, G,\ t = 1, \ldots, T \,\}, \tag{5}$$

and apply group-wise z-normalization:

$$A_{i,t} = \frac{r_{i,t} - \mathrm{mean}(\mathbf{R})}{\mathrm{std}(\mathbf{R})}. \tag{6}$$

While $A_{i,t}$ captures the relative quality of each turn, it only reflects immediate effects and ignores the impact of current decisions on future turns. To incorporate such long-horizon dependencies, we compute a discounted cumulative advantage to propagate outcome signals backward to earlier turns:

$$\widetilde{A}_{i,t} = \sum_{k=t}^{T} \gamma^{\,k-t} A_{i,k}, \tag{7}$$

where $\gamma \in (0, 1]$ is the discount factor. During optimization, $\widetilde{A}_{i,t}$ is assigned to all decision tokens produced in turn t, while raw tool responses from the external environment are masked out. This yields a dense and future-aware supervision signal for policy learning.

[*] Due to its log-prob origin, we apply stop-gradient to the information gain–based reward.

Policy Optimization. With the discounted turn-level advantages $\{\widetilde{A}_{i,j}\}$ defined above, we optimize the agent policy using a clipped surrogate objective with KL regularization, following the same structure as GRPO but with a finer-grained credit assignment. Formally, the IGPO objective is

$$\mathcal{J}_{\text{IGPO}}(\theta) = \mathbb{E}_{(q,a)\sim\mathcal{D},\,\{o_i\}\sim\pi_{\theta_{\text{old}}}(\cdot|q)}\left[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{t=1}^{|o_i|}\min\left(\frac{\pi_\theta(o_{i,t}\mid q,o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t}\mid q,o_{i,<t})}\,\widetilde{A}_{i,t},\right.\right.$$
$$\left.\left.\text{clip}\left(\frac{\pi_\theta(o_{i,t}\mid q,o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t}\mid q,o_{i,<t})},\,1-\epsilon,\,1+\epsilon\right)\,\widetilde{A}_{i,t}\right) - \beta\,\mathbb{D}_{\text{KL}}(\pi_\theta\,\|\,\pi_{\text{ref}})\right], \tag{8}$$

where ϵ is the clipping threshold, β controls the KL penalty strength, and t maps token $o_{i,t}$ to its originating turn. During optimization, only decision tokens (reasoning, tool calls, and answers) receive gradient updates, while raw tool responses are masked out.

To further substantiate the simplicity and implementability of the proposed IGPO, we provide an algorithmic flow comparison between IGPO and GRPO in Appendix E.

4 Experiments

4.1 Experimental Setup

Datasets & Metrics. To evaluate the effectiveness of our proposed IGPO, we conduct experiments on both in-domain (ID) and out-of-domain (OOD) QA benchmarks in an agentic search setting. Following previous work (Zheng et al., 2025b; Deng et al., 2025), the ID setting includes four widely used datasets: NQ (Kwiatkowski et al., 2019), TQ (Joshi et al., 2017), HotpotQA (Yang et al., 2018), and 2Wiki (Ho et al., 2020), while the OOD setting includes three datasets: MusiQue (Trivedi et al., 2022), Bamboogle (Press et al., 2022), and PopQA (Mallen et al., 2022). We report word-level F1 as the evaluation metric, which is computed as the harmonic mean of precision and recall between the predicted and reference answers.

Baselines. To directly verify IGPO's superiority on agentic search tasks, we compare it against a set of competitive baselines: (1) Prompt-based methods: CoT (Wei et al., 2022), CoT+RAG (Gao et al., 2023), and Search-o1 (Li et al., 2025b), which represent the baseline performance of LLMs without further training on search tasks. (2) Outcome-reward RL-based methods: Search-r1-base/Instruct (Jin et al., 2025), R1-searcher (Song et al., 2025a), and DeepResearcher (Zheng et al., 2025b), the representative search agents with outcome-based reward RL, yielding marked performance gains. (3) Step-reward RL-based methods: StepSearch-base/instruct (Wang et al., 2025), ReasoningRAG (Zhang et al., 2025c), and GiGPO (Feng et al., 2025), which are the latest approaches exploring step-reward RL in search-agent settings.

To further validate IGPO's effectiveness, we also compare it against the following commonly used RL algorithms under the same configuration: PPO (Schulman et al., 2017), a widely used actor-critic algorithm that requires an additional value model, and critic-free methods Reinforce++ (Hu, 2025), RLOO (Kool et al., 2019; Ahmadian et al., 2024), GRPO (Shao et al., 2024), and GSPO(Zheng et al., 2025a) which perform advantage estimation over trajectory groups or batchs.

Implementation Details. We use Qwen2.5-7B-Instruct (Qwen et al., 2025) as our backbone model. The training is conducted using the verl (Sheng et al., 2025) framework. The discounted factor γ is set to 1 with no future tuning. At each training step, we sample 32 prompts, and sample 16 rollouts for each prompt. The maximum dialogue turns are set to 10. For the environment, we use the google search API as our tool. The settings of our experiments are consistent with DeepResearcher (Zheng et al., 2025b). For the other baselines in Table 1, we directly copy their reported results. All RL training methods (including ours and the baselines) use exactly the same hyperparameter configurations. The training and inference prompt templates are shown in Appendix F. Please refer to Appendix C for more details.

4.2 Overall Performance

The overall performance comparison between IGPO and the baseline methods is presented in Table 1 and Table 2. Based on these results, we can draw the following key observations:

Table 1: Main results of IGPO compared with different agentic RL baselines across seven datasets.

	In-domain				Out-of-domain			
Method	NQ	TQ	HotpotQA	2Wiki	Musique	Bamboogle	PopQA	Avg.
Prompt-based								
CoT	19.8	45.6	24.4	26.4	8.5	22.1	17.0	23.4
CoT+RAG	42.0	68.9	37.1	24.4	10.0	25.4	46.9	36.4
Search-o1	32.4	58.9	33.0	30.9	14.7	46.6	38.3	36.4
Outcome-reward RL-based								
Search-r1-base	45.4	71.9	55.9	44.6	26.7	56.5	43.2	49.2
Search-r1-instruct	33.1	44.7	45.7	43.4	26.5	45.0	43.0	40.2
R1-searcher	35.4	73.1	44.8	59.4	22.8	64.8	42.7	49.0
DeepResearcher	39.6	78.4	52.8	59.7	27.1	71.0	48.5	53.9
Step-reward RL-based								
StepSearch-base	-	-	49.3	45.0	**32.4**	57.3	-	46.0
StepSearch-instruct	-	-	50.2	43.1	31.2	53.4	-	44.5
ReasoningRAG	-	-	48.9	50.4	20.6	45.5	46.2	42.3
GiGPO	46.4	64.7	41.6	43.6	18.9	68.9	46.1	47.2
IGPO	**46.7**	**80.1**	**57.2**	**68.2**	31.4	**74.9**	**52.5**	**58.7**

Table 2: Main results of IGPO compared with different RL baselines across seven datasets.

	In-domain				Out-of-domain			
Method	NQ	TQ	HotpotQA	2Wiki	Musique	Bamboogle	PopQA	Avg.
RLOO	40.7	72.5	49.6	55.0	24.8	62.2	43.1	49.7
PPO	38.7	75.4	48.6	59.7	26.2	63.4	48.7	51.5
GRPO	40.3	77.0	48.9	57.7	25.0	65.1	49.6	51.9
Reinforce++	34.3	67.5	45.9	54.5	23.7	61.2	44.3	47.3
GSPO	41.5	77.7	46.3	60.1	25.4	67.6	45.4	52.0
IGPO	**46.7**	**80.1**	**57.2**	**68.2**	**31.4**	**74.9**	**52.5**	**58.7**

Training-based methods consistently outperform prompt-based baselines. As shown in Table 1, all reinforcement learning–based methods, whether outcome- or step-reward driven, achieve substantially higher performance than all prompt-based approaches. This confirms that explicit policy optimization is essential for developing effective LLM-based agents, as opposed to relying on zero-shot prompting alone.

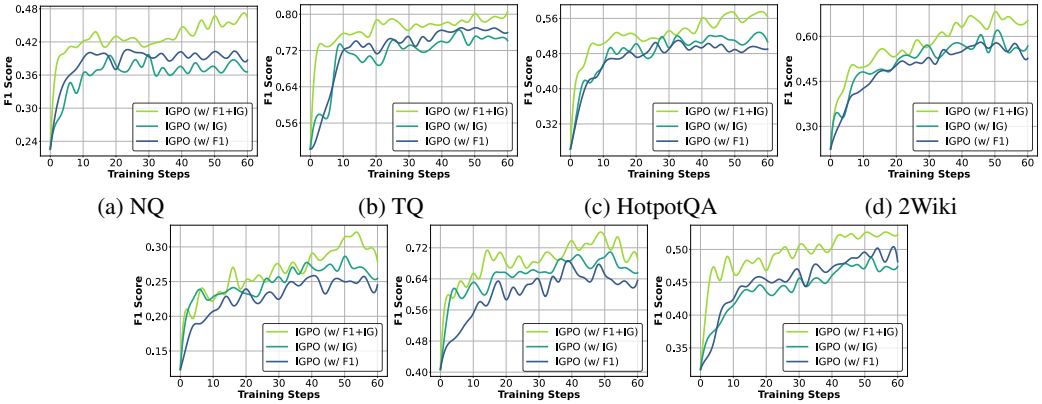
Existing step-reward methods yield competitive but unstable improvements compared to outcome-reward RL methods. While step-reward baselines occasionally surpass outcome-reward ones on specific datasets (e.g., StepSearch on Musique), their overall performance still lags behind the strongest outcome-reward methods such as DeepResearcher. This suggests that existing step-reward designs, although able to provide intermediate guidance, often suffer from noisy or weak supervision signals that limit their generalizability.

IGPO achieves the best overall performance across both in-domain and out-of-domain datasets. Our IGPO outperforms all baselines, with an average score of 58.7, a clear margin over the best method (+4.8 over DeepResearcher). This improvement is attributed to IGPO's information gain-based reward design, which assigns intrinsic, ground-truth-aware credit at every turn while preserving the outcome reward at the answer step. By avoiding advantage collapse and improving sample efficiency, IGPO delivers robust gains across both in-domain and out-of-domain datasets.

IGPO consistently outperforms other RL algorithms. Beyond task-specific baselines, Table 2 shows that IGPO also achieves the highest overall score among standard RL methods, surpassing RLOO, PPO, Reinforce++, and GSPO. Unlike these methods, which rely solely on sparse outcome

Table 3: Ablation results of IGPO on Qwen2.5-3B/7B-Instruct with different reward designs. IGPO (w/ F1) corresponds to using only outcome rewards, reducing to standard GRPO.

	In-domain				Out-of-domain			
Method	NQ	TQ	HotpotQA	2Wiki	Musique	Bamboogle	PopQA	Avg.
Qwen2.5-3B-Instruct								
IGPO (w/ F1)	31.0	55.6	27.5	29.4	12.1	35.7	34.9	32.3
IGPO (w/ IG)	29.1	53.6	27.9	36.5	17.5	44.7	31.3	34.4
IGPO (w/ F1+IG)	**40.5**	**69.4**	**46.8**	**48.2**	**23.1**	**57.9**	**47.4**	**47.6**
Qwen2.5-7B-Instruct								
IGPO (w/ F1)	40.3	77.0	48.9	57.7	25.0	65.1	49.6	51.9
IGPO (w/ IG)	37.5	75.0	51.0	61.0	28.6	69.6	47.1	52.8
IGPO (w/ F1+IG)	**46.7**	**80.1**	**57.2**	**68.2**	**31.4**	**74.9**	**52.5**	**58.7**



(a) NQ (b) TQ (c) HotpotQA (d) 2Wiki

(e) Musique (f) Bamboogle (g) PopQA

Figure 3: Training curves on Qwen2.5-7B-Instruct with different reward designs.

rewards, IGPO incorporates turn-level advantages to provide denser and more stable supervision, leading to stronger generalization and more efficient training.

4.3 Ablation Study

We further conduct ablation experiments to assess the contribution of different reward components. As shown in Table 3, we observe:

First, using only information gain (IG) turn-based reward or only outcome reward (F1) yields clearly inferior results compared to the full combination. This highlights the complementary roles of turn-level and outcome-level supervision: the outcome reward enforces alignment with the final task objective but suffers from severe sparsity, whereas the information gain reward offers dense and stable guidance for intermediate steps.

Second, IGPO with only IG achieves performance comparable to or even exceeding that of standard GRPO (i.e., IGPO w/ F1). This demonstrates that IGPO's information gain reward is not subject to reward hacking. Usually, without outcome supervision, unstable reward designs would quickly collapse. In contrast, our IGPO remains robust because its turn-level signals are intrinsically defined and grounded in the ground truth.

Third, the improvements are particularly pronounced on the smaller 3B model. Compared to standard GRPO, IGPO improves the 3B model by +15.3 points (32.3 $\rightarrow$ 47.6) and the 7B model by +6.8 points (51.9 $\rightarrow$ 58.7). This larger benefit on the 3B model arises because advantage collapse is more severe for weaker models that struggle to directly produce correct answers (Figure 1), making them especially reliant on dense reward signals. In such cases, the information gain reward helps prune noisy reasoning paths and reinforce rollouts that progressively approach the ground truth.

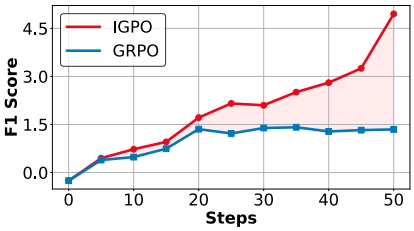


Figure 4: Mean reduction in ground-truth answer entropy from the initial query (Turn 0) to the last non-answer turn ($T-1$) during training.

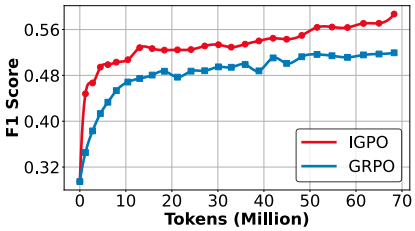


Figure 5: Token Efficiency: average performance with respect to the number of tokens used for gradient updates.

Finally, IGPO demonstrates consistently faster and more stable learning dynamics. As shown in Figure 3, IGPO steadily outperforms its two ablated variants throughout training across all seven datasets. The curves highlight two advantages: (i) IGPO converges to higher F1 scores, confirming the benefit of combining intrinsic turn-level reward and outcome rewards, and (ii) IGPO maintains stable improvements over steps, indicating robustness against reward sparsity and noisy supervision. These results further validate that IGPO provides dense and reliable training signals, thereby improving both training efficiency and final performance.

4.4 In-Depth Analysis

Ground-truth Entropy Reduction. To better understand how IGPO improves training dynamics, we measure the change in ground-truth answer entropy from the initial query (Turn 0) to the last non-answer turn ($T-1$). As shown in Figure 4, IGPO consistently achieves a larger entropy reduction than GRPO throughout training. This indicates that the information gain reward effectively encourages intermediate steps to move the policy closer to the ground-truth answer distribution. In contrast, outcome-based supervision in GRPO provides no guidance for intermediate turns, resulting in weaker entropy reduction. These results highlight that IGPO's turn-level supervision translates into more confident and grounded reasoning trajectories.

Token Efficiency. We further compare IGPO and GRPO in terms of token efficiency, i.e., the performance improvement per token used for gradient updates. As shown in Figure 5, performance increases more rapidly under IGPO, and the gap over GRPO widens as training progresses. In other words, IGPO achieves stronger performance with fewer tokens, indicating that its turn-level rewards deliver denser and more informative gradients than outcome-only supervision. This finding is consistent with the training dynamics observed in Figure 3, where IGPO not only converges faster but also maintains a stable advantage throughout optimization. Such improvements in token efficiency are particularly valuable in agentic RL, where training data is scarce and expensive to obtain, making efficient use of every gradient update a critical factor for scaling.

The case study and additional analyses are provided in Appendix D.

Beyond empirical effectiveness, our theoretical analysis in Appendix A shows that maximizing turn-level information gain constrains error accumulation in multi-turn scenarios. Thus, IGPO not only alleviates advantage collapse but also reduces error accumulation in long-horizon agentic tasks.

5 Related Work

The recent success of reinforcement learning (RL) methods in large reasoning models (Chen et al., 2025a) such as OpenAI o1 (Jaech et al., 2024) and DeepSeek R1 (Guo et al., 2025) has established RL as a central tool for enhancing large language models (LLMs)-based agents to solve more complex tasks. A growing body of work has explored different RL algorithms such as PPO (Schulman et al., 2017), Reinforce++ (Hu, 2025), GRPO (Shao et al., 2024), RLOO (Kool et al., 2019; Ahmadian et al., 2024), DAPO (Yu et al., 2025), and GSPO (Zheng et al., 2025a). These methods have been particularly effective in improving the capabilities of LLM-based agents (Li et al., 2025a).

Building on these advances, an important line of research has focused on applying RL to search-based agents (Deng et al., 2025; Dai et al., 2025b;a). Early efforts such as DeepRetrieval (Jiang et al., 2025) demonstrated the feasibility of end-to-end optimization by applying PPO with retrieval-oriented metrics (e.g., recall) as rewards. Subsequent works, including Search-R1 (Jin et al., 2025), DeepResearcher (Zheng et al., 2025b), and ReSearch (Chen et al., 2025b), extended this paradigm to multi-turn reasoning and search. R1-Searcher (Song et al., 2025a) and R1-Searcher++ (Song et al., 2025b) further introduced two-stage RL strategies, separately strengthening the ability to interact with retrieval systems and to utilize retrieved information effectively.

However, in multi-turn scenarios, outcome-only rewards remain sparse and often fail to provide sufficient guidance, leading to unstable optimization and inefficient sample utilization. Recent studies have explored step-wise or process-level rewards that assign credit to intermediate actions. ReasonRAG (Zhang et al., 2025c) adopted Monte Carlo Tree Search (MCTS) to approximate the value of each step. StepSearch (Wang et al., 2025) leveraged a memory vector of retrieved documents, supervising intermediate steps based on their maximum similarity to ground-truth evidence. GiGPO (Feng et al., 2025) introduced anchor-based grouping to estimate relative advantages for actions originating from the same anchor state. While these methods provide denser supervision than outcome-only rewards, they either rely on external oracle knowledge or suffer from limited stability and scalability, leaving room for more intrinsic and generalizable process-level reward designs.

6 Conclusion, Limitations and Future Work

In this work, we propose IGPO, a simple and effective reinforcement learning framework for training multi-turn LLM-based agents. By providing intrinsic, ground-truth-aware supervision at every turn while preserving alignment with the final objective, IGPO delivers dense and stable training signals. Extensive experiments across in-domain and out-of-domain benchmarks demonstrate that IGPO consistently outperforms strong baselines, achieving higher accuracy and better sample efficiency, particularly for smaller models where sparse rewards are most problematic.

However, our approach still relies on the availability of ground-truth answers, which limits its applicability in open-ended settings. In future work, we plan to extend IGPO to broader agentic scenarios beyond search, including tasks without explicit supervision.

References

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

Kedi Chen, Dezhao Ruan, Yuhao Dan, Yaoting Wang, Siyu Yan, Xuecheng Wu, Yinqi Zhang, Qin Chen, Jie Zhou, Liang He, Biqing Qi, Linyang Li, Qipeng Guo, Xiaoming Shi, and Wei Zhang. A survey of inductive reasoning for large language models, 2025a. URL https://arxiv.org/abs/2510.10182.

Mingyang Chen, Tianpeng Li, Haoze Sun, Yijie Zhou, Chenzheng Zhu, Haofen Wang, Jeff Z Pan, Wen Zhang, Huajun Chen, Fan Yang, et al. Learning to reason with search for llms via reinforcement learning. *arXiv preprint arXiv:2503.19470*, 2025b.

Yuqin Dai, Guoqing Wang, Yuan Wang, Kairan Dou, Kaichen Zhou, Zhanwei Zhang, Shuo Yang, Fei Tang, Jun Yin, Pengyu Zeng, et al. Evinote-rag: Enhancing rag models via answer-supportive evidence notes. *arXiv preprint arXiv:2509.00877*, 2025a.

Yuqin Dai, Shuo Yang, Guoqing Wang, Yong Deng, Zhanwei Zhang, Jun Yin, Pengyu Zeng, Zhenzhe Ying, Changhua Meng, Can Yi, et al. Careful queries, credible results: Teaching rag models advanced web search tools with reinforcement learning. *arXiv preprint arXiv:2508.07956*, 2025b.

Yong Deng, Guoqing Wang, Zhenzhe Ying, Xiaofeng Wu, Jinzhen Lin, Wenwen Xiong, Yuqin Dai, Shuo Yang, Zhanwei Zhang, Qiwen Wang, Yang Qin, Yuan Wang, Quanxing Zha, Sunhao Dai, and Changhua Meng. Atom-searcher: Enhancing agentic deep research via fine-grained atomic thought reward. *arXiv preprint arXiv:2508.12800*, 2025.

Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. Group-in-group policy optimization for llm agent training. *arXiv preprint arXiv:2505.10978*, 2025.

Zeyu Gan, Yun Liao, and Yong Liu. Rethinking external slow-thinking: From snowball errors to probability of correct reasoning. In *Forty-second International Conference on Machine Learning*, 2025.

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

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638, 2025.

Carlos I Gutierrez, Anthony Aguirre, Risto Uuk, Claire C Boine, and Matija Franklin. A proposal for a definition of general purpose artificial intelligence systems. *Digital Society*, 2(3):36, 2023.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 6609–6625, 2020.

Jian Hu. Reinforce++: A simple and efficient approach for aligning large language models. *arXiv preprint arXiv:2501.03262*, 2025.

Yuxuan Huang, Yihang Chen, Haozheng Zhang, Kang Li, Meng Fang, Linyi Yang, Xiaoguang Li, Lifeng Shang, Songcen Xu, Jianye Hao, et al. Deep research agents: A systematic examination and roadmap. *arXiv preprint arXiv:2506.18096*, 2025.

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024.

Pengcheng Jiang, Jiacheng Lin, Lang Cao, Runchu Tian, SeongKu Kang, Zifeng Wang, Jimeng Sun, and Jiawei Han. Deepretrieval: Hacking real search engines and retrievers with large language models via reinforcement learning. *arXiv preprint arXiv:2503.00223*, 2025.

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

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *arXiv preprint arXiv:1705.03551*, 2017.

Wouter Kool, Herke van Hoof, and Max Welling. Buy 4 reinforce samples, get a baseline for free! 2019.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466, 2019.

Long Li, Jiaran Hao, Jason Klein Liu, Zhijian Zhou, Xiaoyu Tan, Wei Chu, Zhe Wang, Shirui Pan, Chao Qu, and Yuan Qi. The choice of divergence: A neglected key to mitigating diversity collapse in reinforcement learning with verifiable reward. *arXiv preprint arXiv:2509.07430*, 2025a.

Xiaoxi Li, Guanting Dong, Jiajie Jin, Yuyao Zhang, Yujia Zhou, Yutao Zhu, Peitian Zhang, and Zhicheng Dou. Search-o1: Agentic search-enhanced large reasoning models. *arXiv preprint arXiv:2501.05366*, 2025b.

Xuefeng Li, Haoyang Zou, and Pengfei Liu. Torl: Scaling tool-integrated rl. *arXiv preprint arXiv:2503.23383*, 2025c.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Hannaneh Hajishirzi, and Daniel Khashabi. When not to trust language models: Investigating effectiveness and limitations of parametric and non-parametric memories. *arXiv preprint arXiv:2212.10511*, 7, 2022.

Liangbo Ning, Ziran Liang, Zhuohang Jiang, Haohao Qu, Yujuan Ding, Wenqi Fan, Xiao-yong Wei, Shanru Lin, Hui Liu, Philip S Yu, et al. A survey of webagents: Towards next-generation ai agents for web automation with large foundation models. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 6140–6150, 2025.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A Smith, and Mike Lewis. Measuring and narrowing the compositionality gap in language models. *arXiv preprint arXiv:2210.03350*, 2022.

Zehan Qi, Xiao Liu, Iat Long Iong, Hanyu Lai, Xueqiao Sun, Wenyi Zhao, Yu Yang, Xinyue Yang, Jiadai Sun, Shuntian Yao, et al. Webrl: Training llm web agents via self-evolving online curriculum reinforcement learning. *arXiv preprint arXiv:2411.02337*, 2024.

Changle Qu, Sunhao Dai, Xiaochi Wei, Hengyi Cai, Shuaiqiang Wang, Dawei Yin, Jun Xu, and Ji-Rong Wen. Tool learning with large language models: A survey. *Frontiers of Computer Science*, 19(8):198343, 2025.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025.

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

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. In *Proceedings of the Twentieth European Conference on Computer Systems*, EuroSys '25, pp. 1279–1297. ACM, March 2025. doi: 10.1145/3689031.3696075.

Huatong Song, Jinhao Jiang, Yingqian Min, Jie Chen, Zhipeng Chen, Wayne Xin Zhao, Lei Fang, and Ji-Rong Wen. R1-searcher: Incentivizing the search capability in llms via reinforcement learning. *arXiv preprint arXiv:2503.05592*, 2025a.

Huatong Song, Jinhao Jiang, Wenqing Tian, Zhipeng Chen, Yuhuan Wu, Jiahao Zhao, Yingqian Min, Wayne Xin Zhao, Lei Fang, and Ji-Rong Wen. R1-searcher++: Incentivizing the dynamic knowledge acquisition of llms via reinforcement learning. *arXiv preprint arXiv:2505.17005*, 2025b.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Musique: Multihop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554, 2022.

Peiyi Wang, Lei Li, Zhihong Shao, RX Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. Math-shepherd: Verify and reinforce llms step-by-step without human annotations. *arXiv preprint arXiv:2312.08935*, 2023.

Xiaohua Wang, Zhenghua Wang, Xuan Gao, Feiran Zhang, Yixin Wu, Zhibo Xu, Tianyuan Shi, Zhengyuan Wang, Shizheng Li, Qi Qian, et al. Searching for best practices in retrieval-augmented generation. *arXiv preprint arXiv:2407.01219*, 2024a.

Ziliang Wang, Xuhui Zheng, Kang An, Cijun Ouyang, Jialu Cai, Yuhang Wang, and Yichao Wu. Stepsearch: Igniting llms search ability via step-wise proximal policy optimization. *arXiv preprint arXiv:2505.15107*, 2025.

Ziting Wang, Haitao Yuan, Wei Dong, Gao Cong, and Feifei Li. Corag: A cost-constrained retrieval optimization system for retrieval-augmented generation. *arXiv preprint arXiv:2411.00744*, 2024b.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W Cohen, Ruslan Salakhutdinov, and Christopher D Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. *arXiv preprint arXiv:1809.09600*, 2018.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025.

Guibin Zhang, Hejia Geng, Xiaohang Yu, Zhenfei Yin, Zaibin Zhang, Zelin Tan, Heng Zhou, Zhongzhi Li, Xiangyuan Xue, Yijiang Li, et al. The landscape of agentic reinforcement learning for llms: A survey. *arXiv preprint arXiv:2509.02547*, 2025a.

Weizhi Zhang, Yangning Li, Yuanchen Bei, Junyu Luo, Guancheng Wan, Liangwei Yang, Chenxuan Xie, Yuyao Yang, Wei-Chieh Huang, Chunyu Miao, et al. From web search towards agentic deep research: Incentivizing search with reasoning agents. *arXiv preprint arXiv:2506.18959*, 2025b.

Wenlin Zhang, Xiangyang Li, Kuicai Dong, Yichao Wang, Pengyue Jia, Xiaopeng Li, Yingyi Zhang, Derong Xu, Zhaocheng Du, Huifeng Guo, et al. Process vs. outcome reward: Which is better for agentic rag reinforcement learning. *arXiv preprint arXiv:2505.14069*, 2025c.

Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, et al. Group sequence policy optimization. *arXiv preprint arXiv:2507.18071*, 2025a.

Yuxiang Zheng, Shichao Sun, Lin Qiu, Dongyu Ru, Cheng Jiayang, Xuefeng Li, Jifan Lin, Binjie Wang, Yun Luo, Renjie Pan, et al. Openresearcher: Unleashing ai for accelerated scientific research. *arXiv preprint arXiv:2408.06941*, 2024.

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

Yuxin Zuo, Kaiyan Zhang, Li Sheng, Shang Qu, Ganqu Cui, Xuekai Zhu, Haozhan Li, Yuchen Zhang, Xinwei Long, Ermo Hua, et al. Ttrl: Test-time reinforcement learning. *arXiv preprint arXiv:2504.16084*, 2025.

A THEORETICAL ANALYSIS

The theoretical analysis here provides an intuitive support for the efficacy of our proposed method by addressing the limitations of sparse outcome rewards in multi-turn agents. Specifically, the theory establishes a crucial link: maximizing the process reward (IGPO's objective) is equivalent to minimizing the upper bound on the undesirable accumulation of snowball errors during the reasoning process. This minimization, in turn, systematically lowers the theoretical minimum for the final answer error rate, thus providing a fundamental guarantee that IGPO's dense, turn-level signals lead to more confident and successful reasoning trajectories.

Notations. Let E_{final} be the event that the agent's generated final answer does not match the ground truth answer. Its probability is denoted by $\mathbb{P}(E_{\text{final}})$, i.e., the error rate. For each turn t , denote the observed response `[think]`, `[tool call]` as $\mathcal{R}_t$. We also posit that there is an unobservable, abstract thinking step $\mathcal{I}_t$ that underlies the generation of $\mathcal{R}_t$. Let $R^{(t)}_{\text{process}}$ be the process reward, which is a dense reward signal received at each turn of the interaction. It is defined as the information gain about the ground truth answer, which is calculated as the increase in the log-probability of the correct answer from the previous state to the current state. Then, the total process reward $R_{\text{total}} = \sum_{t=1}^{T-1} \mathbb{E}[R^{(t)}_{\text{process}}]$ is the cumulative sum of all process rewards over a complete trajectory or episode. The expectation is taken over the thinking step and observed response. The training objective of the policy is to maximize this total reward.

Definition A.1 (Snowball Error in Multi-turn Agentic RL)**.** *Consistent with Gan et al. (2025), we define the information loss at turn T as the conditional entropy $\mathrm{Ent}(\mathcal{I}_t|\mathcal{R}_t)$. Consider the non-trivial case where $|\mathrm{Ent}(\mathcal{I}_t|\mathcal{R}_t)|$ is bounded. The cumulative snowball error up to turn T is the sum of these losses:*

$$\mathrm{Ent}_{<T}(\mathcal{I}|\mathcal{R}) \triangleq \sum_{t=1}^{T-1} \mathrm{Ent}(\mathcal{I}_t|\mathcal{R}_t) \tag{9}$$

This quantity measures the aggregate uncertainty and ambiguity accumulated throughout the reasoning trajectory before the final answer is produced.

Next, we connect the cumulative snowball error to the agent's final performance. It indicates the fundamental limitation of multi-turn agentic RL pipeline caused by snowball error.

Lemma A.2 (Lower bound of error rate)**.** *The probability of a final answer error, $\mathbb{P}(E_{final})$, is lower-bounded by the cumulative snowball error accumulated during the reasoning process:*

$$\mathbb{P}(E_{final}) = \Omega\left(\frac{\mathrm{Ent}_{<T}(\mathcal{I}|\mathcal{R})}{T-1}\right) - C_{const}. \tag{10}$$

where C_{const} is a small positive constant.

Proof Sketch. This result is strongly motivated by Theorem 3.3 from Gan et al. (2025). We treat the generation of the final answer at turn T as the final step of a multi-step reasoning process. The quality of this final step is conditioned on the information accumulated over the previous $T-1$ turns. The theorem from (Gan et al., 2025) states that the error probability of any step is lower-bounded by the average snowball error accumulated up to that point. Applying this principle to the final step ($t = T$) yields the stated result. □

Assumption A.3 (Monotonic Reward-Information Loss Link)**.** *The expected process reward at any turn H, $\mathbb{E}[R^{(t)}_{process}]$, is monotonically non-increasing with respect to the information loss at that turn, $\mathrm{Ent}(\mathcal{I}_t|\mathcal{R}_t)$. We assume there exists a bounded and monotonically non-increasing convex function $f : \mathbb{R}_+ \to \mathbb{R}$ such that:*

$$\mathbb{E}[R^{(t)}_{process}|\mathcal{I}_t, \mathcal{R}_t] \leq f\left(\mathrm{Ent}(\mathcal{I}_t|\mathcal{R}_t)\right). \tag{11}$$

Remark. *As the information loss $\mathrm{Ent}(\mathcal{I}_t|\mathcal{R}_t)$ at turn t increases, the expected total information loss tends to decreases and asymptotically approaches a relatively small value, which is characterized by the convex nature of function f.*

This assumption leads to the following result, demonstrating that optimizing for process rewards implicitly constrains the accumulation of snowball errors. We first formalize the intuition that a clearer reasoning step (lower information loss) is a prerequisite for a high-quality query, which in turn yields a higher expected process reward.

Theorem A.4 (Process Reward as a Bound on Snowball Error)**.** *Under Assumption A.3, the expected cumulative snowball error is upper bounded by*

$$\mathbb{E}[\mathrm{Ent}_{<T}(\mathcal{I}|\mathcal{R})] = \mathcal{O}(1) - \Omega\left(R_{total}\right). \tag{12}$$

Theorem A.4 establishes that maximizing the process reward is mathematically coupled with minimizing an upper bound on the cumulative snowball error. The combination of Theorem A.4 and Lemma A.2 provides a complete, end-to-end theoretical justification for the efficacy of our proposed process reward mechanism. The logical chain is as follows:

- **Maximizing the process reward** (our algorithm's objective) forces the agent to **minimize an upper bound on the cumulative snowball error** (Theorem A.4).
- Minimizing the cumulative snowball error, in turn, **lowers the theoretical minimum for the final error rate**, thereby systematically increasing the probability of task success (Lemma A.2).

In conclusion, the turn-level process reward is not merely an engineering heuristic; it is a theoretically grounded mechanism that fundamentally addresses the problem of error accumulation in multi-step reasoning. By providing a dense, immediate signal for reasoning clarity, it transforms the intractable problem of sparse-reward, long-horizon exploration into a series of manageable, short-horizon sub-problems, each aimed at maximizing immediate information gain. This explains the significant gains in training efficiency and final performance observed in our experiments.

B Proof for Theoretical Analysis

B.1 Proof of Lemma A.2

Proof. We achieve this by applying Theorem 3.3 from Gan et al. (2025) to the final decision-making step of the agent. In particular,

$$\mathbb{P}(E_{\text{final}}) \geq \frac{\frac{\mathrm{Ent}_{<T}(\mathcal{I}|\mathcal{R})}{T-1} - C_1}{\log(|\mathcal{A}_{\text{final}}| - 1)}, \tag{13}$$

where $|\mathcal{A}_{\text{final}}|$ is the cardinality of the final answer space and C_1 is a small positive constant analogous to $\mathrm{Ent}_b(e_t)$ in Gan et al. (2025). Since $\log(|\mathcal{A}_{\text{final}}|-1)$ and C_1 are constant, $\frac{\frac{\mathrm{Ent}_{<T}(\mathcal{I}|\mathcal{R})}{T-1}-C_1}{\log(|\mathcal{A}_{\text{final}}|-1)}$ simplifies to a form that is asymptotically dominated by the variable term. Therefore, the right-hand side of the inequality can be expressed in terms of the lower bound symbol Ω as $\Omega\left(\frac{\mathrm{Ent}_{<T}(\mathcal{I}|\mathcal{R})}{T-1}\right) - C_{\text{const}}$, which completes the proof. □

B.2 Proof of Theorem A.4

Proof. According to the nature of f and the fact that there exist constants $C_{\max}$ and β such that for all non-negative bounded x, there holds $f(x) \leq C_{\max} - \beta x$. Therefore, by taking the expectation over Assumption A.3 and summing across all turns from $t = 1$ to $T - 1$, we have

$$\begin{aligned}
R_{\text{total}} = \sum_{t=1}^{T-1} \mathbb{E}[R^{(t)}_{\text{process}}] &\leq \sum_{t=1}^{T-1} \mathbb{E}[f\left(\mathrm{Ent}(\mathcal{I}_t|\mathcal{R}_t)\right)] \\
&\leq \sum_{t=1}^{T-1} \mathbb{E}[C_{\max} - \beta \cdot \mathrm{Ent}(\mathcal{I}_t|\mathcal{R}_t)] \\
&= (T-1)C_{\max} - \beta \sum_{t=1}^{T-1} \mathbb{E}[\mathrm{Ent}(\mathcal{I}_t|\mathcal{R}_t)] \\
&= (T-1)C_{\max} - \beta\, \mathbb{E}[\mathrm{Ent}_{<T}(\mathcal{I}|\mathcal{R})].
\end{aligned}$$

Rearranging terms yields the final result. □

C More Implementation Details

All our training experiments are conducted on 8 × NVIDIA A100-80G GPUs. The detailed hyperparameter settings are provided in Table 4. Unless otherwise specified, all experiments are based on this configuration.

Table 4: Training hyperparameters.

Training hyperparameters	Value
Training Batch Size	32
Mini-Batch Size	32
Infer Tensor Model Parallel Size	1
Sequence Parallel Size	4
Max Prompt Length	30767
Max Response Length	2000
Actor Learning Rate	1e-6
Rollout Temperature	1.0
Rollout Group Size	16
Max Turn Call Turns	10
KL-Divergence loss coefficient	0.001

D More Discussion and Experimental Analysis

D.1 Comparison with Other Process-reward Methods

In addition to its obvious performance advantages, we also conduct a deeper analysis of IGPO's superiority in terms of algorithmic characteristics compared to other process-reward-based agentic RL algorithms. We first introduce other existing process-reward-based agentic RL algorithms:

- **ReasoningRAG**. The main contribution of this work is the proposal of a step-level data labeling strategy based on MCTS. Subsequently, the DPO algorithm is used to optimize the agent's policy on the labeled step-level dataset. The main limitations of this method are: (1) the data labeling process relies on MCTS, which is inefficient, and when the number of samples is insufficient, it is difficult to accurately estimate the value of each step; (2) the off-policy optimization based on DPO is less effective than on-policy algorithms.
- **StepSearch**. StepSearch constructs turn-level supervision signals by pre-defining golden search keywords and golden tool responses, and adopts an on-policy optimization approach. Although it shifts from off-policy to on-policy, the annotation process is resource-intensive and prone to annotator bias (whether from humans or LLMs).
- **GiGPO**. GiGPO introduces a step-level grouping strategy based on anchor states and performs fine-grained advantage estimation within each step-level group. Although this provides a novel solution, it essentially still relies on the Monte Carlo assumption. When the number of anchor states is insufficient, it is often difficult to accurately estimate their value, which in turn leads to biased advantage estimation.

The proposed IGPO effectively addresses the aforementioned limitations. Starting from the on-policy GRPO setting (where rollout data are used for a single parameter update), it employs an information-gain–based incremental reward construction strategy that requires no annotation and does not rely on Monte Carlo. Moreover, the incorporation of ground-truth awareness substantially reduces bias. Table 5 provides a detailed comparison highlighting the advantages of IGPO over other algorithms.

D.2 Case Study

Table 5: Comparison between various process reward methods.

Algorithm	On-Policy	Explicit Labeling-Free	Monte Carlo–Free	Introduces No Bias
ReasoningRAG	No	Yes	No	Sample-size Dependent
StepSearch	Yes	No	Yes	No
GiGPO	Yes	Yes	No	Sample-size Dependent
IGPO	Yes	Yes	Yes	Yes

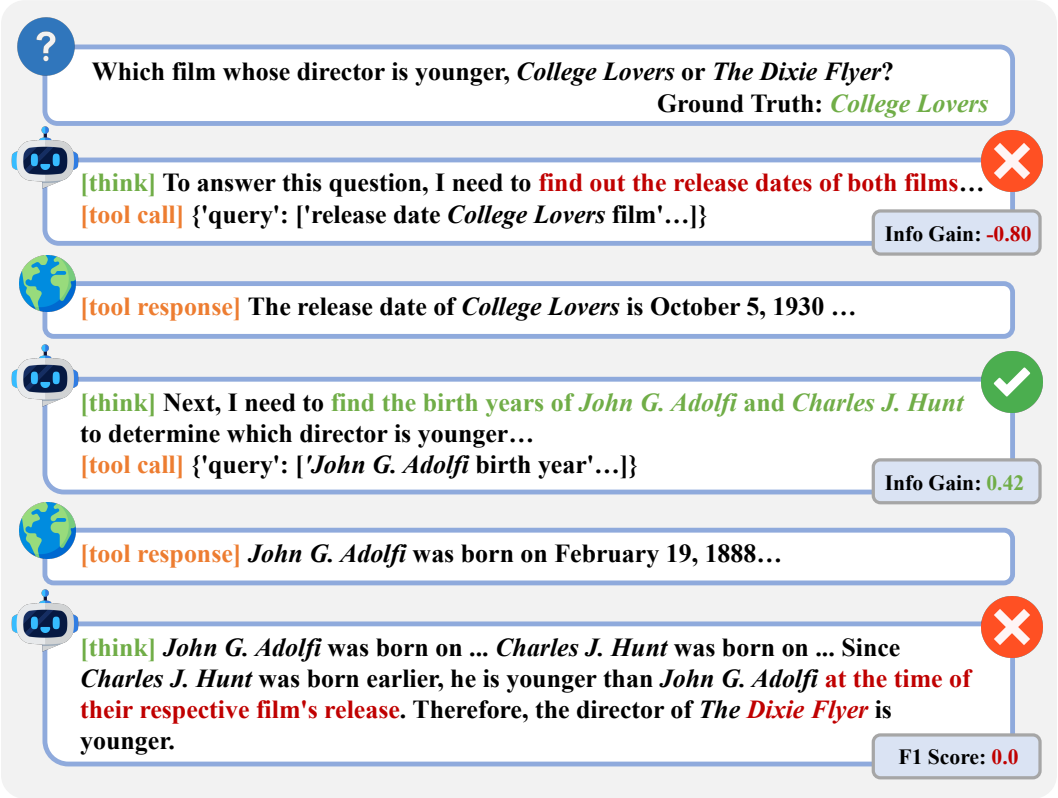


Figure 6: Case study showing a scenario where the final answer is incorrect but contains a single correct retrieval turn. IGPO provides a process reward for this turn, improving token utilization.

Figure 7: Case study illustrating a situation where the first round of retrieval failed, but the second and third rounds successfully located the correct evidence and produced the right answer. In this case, IGPO imposes a penalty on the erroneous retrieval in the first round.

E Comparison between GRPO and IGPO

Algorithm 1 illustrates the algorithmic flow of IGPO (right) and GRPO (left). The key steps corresponding to each algorithm are highlighted with the same color font to visually highlight the differences: yellow for reward calculation, green for advantage estimation, blue for advantage accumulation and assignment, and purple for policy optimization. In terms of reward calculation, IGPO constructs dense turn-level rewards through incremental information gain. For advantage estimation, both IGPO and GRPO use Group Normalization. Regarding advantage accumulation and allocation, GRPO directly assigns the outcome-based advantage to all tokens of the current output, while IGPO further computes the cumulative discounted advantage, capturing long-horizon information and performing turn-level reward assignment. In policy optimization, IGPO achieves more efficient and effective optimization by maximizing the turn-level cumulative discounted advantages.

Algorithm 1 GRPO vs. **IGPO**

GRPO

Require: initial policy $\pi_{\theta_{\text{init}}}$; task prompts $\mathcal{D}$; hyperparameters ϵ, β, μ

$\pi_\theta \leftarrow \pi_{\theta_{\text{init}}}$
for iteration $= 1, \ldots, I$ **do**
  $\pi_{\text{ref}} \leftarrow \pi_\theta$
  for step $= 1, \ldots, M$ **do**
    Sample a batch $\mathcal{D}_b$ from $\mathcal{D}$
    $\pi_{\theta_{\text{old}}} \leftarrow \pi_\theta$
    For each $q \in \mathcal{D}_b$, sample G outputs $\{y_i\}_{i=1}^{G} \sim \pi_{\theta_{\text{old}}}(\cdot \mid q)$
    Compute outcome reward $\{r_i\}_{i=1}^{G}$ from the final answer in each y_i
    Compute each y_i's advantage $\{A_i\}_{i=1}^{G}$ via group normalization of $\{r_i\}_{i=1}^{G}$ (Eq.in Sec 2.2)
    Assign A_i to all tokens of y_i
    for GRPO iter $= 1, \ldots, \mu$ **do**
      Update π_θ by maximizing the GRPO objective (Eq. 1)
    end for
  end for
end for

IGPO

Require: initial policy $\pi_{\theta_{\text{init}}}$; task prompts $\mathcal{D}$; max turns H; hyperparameters ϵ, β, γ, μ

$\pi_\theta \leftarrow \pi_{\theta_{\text{init}}}$
for iteration $= 1, \ldots, I$ **do**
  $\pi_{\text{ref}} \leftarrow \pi_\theta$
  for step $= 1, \ldots, M$ **do**
    Sample a batch $\mathcal{D}_b$ from $\mathcal{D}$
    $\pi_{\theta_{\text{old}}} \leftarrow \pi_\theta$
    For each $q \in \mathcal{D}_b$, sample G outputs $\{y_i\}_{i=1}^{G} \sim \pi_{\theta_{\text{old}}}(\cdot \mid q)$
    for iteration $= 1, \ldots, T$ **do**
      if t $<$ T **then**
        Compute the infomation gain-based turn-level rewards $\{r_{i,t}\}_{i=1}^{G}$ for each y_i (Eq. 4)
      else
        Compute the final-turn rewards $\{r_{i,T}\}_{i=1}^{G}$ based on the answer in each y_i (Eq. 2)
      end if
    end for
    Compute the per turn advantages $\{A_{i,1\le t\le T}\}_{i=1}^{G}$ in y_i via group normalization of $\{r_{i,1\le t\le T}\}_{i=1}^{G}$ (Eq. 6)
    Compute the per turn cumulative discounted advantages $\{\hat{A}_{i,1\le t\le T}\}_{i=1}^{G}$ in each y_i (Eq. 7), then assign them to the tokens in each turn
    for IGPO iteration $= 1, \ldots, \mu$ **do**
      Update π_θ by maximizing the IGPO objective (Eq. 8)
    end for
  end for
end for

F PROMPT TEMPLATE USED IN OUR EXPERIMENTS.

Our prompt follows the style of DeepResearcher Zheng et al. (2025b), and the same template is used for training, validation, and testing. The prompt template is shown in the Figure 8, where {today} represents the current date to ensure the relevance of the model's response. {{ tool.name }}: {{ tool.description }} indicates the available tools, while the #Rollout section controls the model's output format. The #Tools section provides the model with the tool invocation method.

```
* Today is {today}
* You are an  AI Assistant*

The question I give you is a complex question that requires a *deep research* to answer.
I will provide you with tools to help you answer the question:
{%- for tool in tools.values() %}
- {{ tool.name }}: {{ tool.description }}
{%- endfor %}

You don't have to answer the question now, but you should first think about the research plan or what to search next.
Your output format should be one of the following two formats:

# Rollout
<think>
YOUR THINKING PROCESS
</think>
<answer>
YOUR ANSWER AFTER GETTING ENOUGH INFORMATION
</answer>

or

<think>
YOUR THINKING PROCESS
</think>
<tool_call>
YOUR TOOL CALL WITH CORRECT FORMAT
</tool_call>

You should always follow the above two formats strictly.
Only output the final answer (in words, numbers or phrase) inside the <answer></answer> tag, without any
explanations or extra information. If this is a yes-or-no question, you should only answer yes or no.

# Tools
You may call one or more functions to assist with the user query.

You are provided with function signatures within <tools></tools> XML tags:
<tools>
{\%- for tool in tools.values() \%}
  {{ '{' }}"type": "function", "function": {{ '{' }}"name": "{{ tool.name | replace("'", '"') }}", "description":
"{{ tool.description }}", "parameters": {{ '{' }}"type": "object", "properties": {{tool.inputs | replace("'", '"')}}, "example":
{{tool.example | replace("'", '"')}}, "uniqueItems": true{{ '}}' }}
{\%- endfor \%}
</tools>

For each function call, return a json object with function name and arguments within <tool_call></tool_call> XML tags:
<tool_call>
{{ '{' }}"name": <function-name>, "arguments": <args-json-object>{{ '}' }}
</tool_call>
```

Figure 8: Prompt template used in our experiments.