Negative Advantages Is a Double-Edged Sword: Calibrating advantages in GRPO for Search Agents

Anonymous ACL submission

Abstract

Search agents achieve strong question-answering performance through multi-turn interactions with search engines, with Group Relative Policy Optimization (GRPO) being a widely used training algorithm. However, GRPO-style algorithms still face several challenges in multi-hop search settings. First, correct intermediate steps are often penalized when the final answer is wrong. Second, training is highly unstable, often causing degradation of natural language ability or even catastrophic training collapse. Our analysis attributes these issues to coarse-grained advantage assignment and an imbalance between positive and negative advantages. To address these problems, we propose CalibAdv, an advantage calibration method specifically designed for search agents that enables more accurate and more stable modeling of penalties and rewards. Specifically, CalibAdv leverages the correctness of intermediate steps to downscale excessive negative advantages at a fine-grained level. It then further rebalances positive and negative advantages to improve training stability. Importantly, CalibAdv adopts a lightweight design that calibrates advantages from standard rollout signals, making it simple and easy to deploy. Extensive experiments across three models and seven benchmarks demonstrate that CalibAdv improves both model performance and training stability. Our code is available at `https://anonymous.4open.science/r/CalibAdv-D2C5`.

1 Introduction

Search agents demonstrate impressive question-answering capabilities through iterative interactions with search engines (Shi et al., 2025; Jin et al., 2025; Sun et al., 2025; Xu et al., 2026). Compared with standard retrieval-augmented generation (RAG), search agents can flexibly reformulate search queries based on contextual information and adaptively determine the number of search iterations, enabling more effective resolution of multi-hop QA tasks. Their success critically relies on Reinforcement Learning with Verifiable Rewards (RLVR) such as GRPO (Shao et al., 2024).

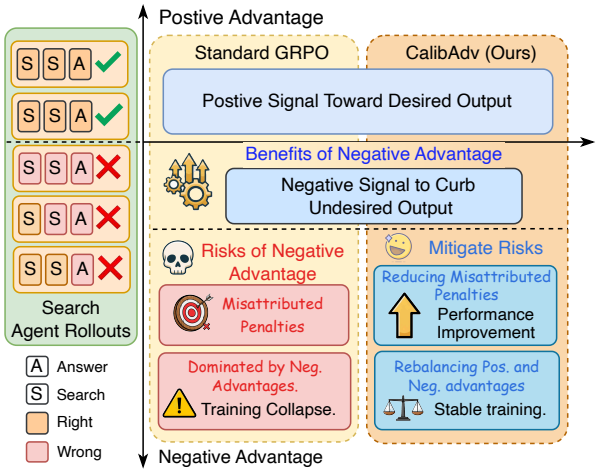

Figure 1: CalibAdv improves model performance and stabilizes training by scaling advantages.

Despite their widespread use in RLVR, GRPO-like methods remain challenging when training search agents. **First, trajectory-level reward assignment can misalign with step-level correctness.** Search agents interact with search engines over multiple turns, but GRPO assigns advantages largely based on the final rollout outcome. Consequently, when the final answer is incorrect, correct or useful intermediate steps may still receive negative advantages, potentially hindering the performance. **Second, training instability is common** (Jin et al., 2025; Sun et al., 2025; Xue et al., 2025; Deng et al., 2025). As training proceeds, incoherent or nonsensical outputs become increasingly frequent, eventually leading to a catastrophic collapse of the model's natural language generation ability. This instability is more pronounced than in single-turn generation, as search agents repeat-

edly inject externally retrieved documents into the context across interaction turns (Xue et al., 2025).

We observe that the ***negative signals***, as a key mechanism in reinforcement learning, are closely related to the issues described above. For intermediate-step mis-penalization, our analysis shows that this is not a rare corner case: **43.97%** of penalized intermediate steps are close to correct, suggesting that advantages should be calibrated at a finer granularity according to step correctness. For training collapse, our analysis points to the persistent dominance of negative advantages: without sufficient positive advantages to anchor the policy, these negative updates can push it away from coherent outputs, thereby increasing entropy and perplexity and eventually causing collapse. These observations inspire us to think: When training search agents with GRPO-like RL, although the penalty signal is crucial for success, *negative advantages must be carefully calibrated based on the correctness of intermediate steps, and positive and negative advantages should be kept in balance.*

To this end, we propose **CalibAdv**, a GRPO advantage calibration method tailored for search agents that enables more accurate and more stable modeling of penalties and rewards. As illustrated in Figure 1, CalibAdv adjusts advantages from three aspects while following a simple first-principles design that uses signals already available in standard search agent rollouts. First, for intermediate steps, we introduce a simple and readily available rule-based silver document proxy to identify likely correct intermediate steps in incorrect rollouts and downscale their negative advantages. Second, for the final answer step, which is particularly prone to imbalance, we rebalance positive and negative advantages via ratio-based rescaling with a natural unit coefficient (1.0), assigning them comparable total weights to mitigate degradation in natural language generation and support stable long-term training without tuning an additional scaling hyperparameter. Finally, to eliminate the conflicting advantages induced by shared prefixes across positive and negative rollouts, we prepend such prefixes directly to the prompt and exclude them from advantage assignment, thereby mitigating format collapse. Overall, CalibAdv remains simple and deployment friendly.

We conduct extensive experiments on three models and seven benchmarks, demonstrating that CalibAdv substantially improves both model performance and training stability. On average, CalibAdv yields a 11.80% relative improvement in F1 score, while avoiding degradation of the model's natural language capabilities and the training collapse. Our contributions can be summarized as follows:

- We mitigate the mismatch between intermediate step correctness and reward signals in GRPO style training for multi turn interactive tasks by introducing a simple and readily available rule-based silver document proxy for fine-grained advantage calibration.

- We identify the root cause of training collapse in GRPO for search agent training and address this long-standing issue by rebalancing positive and negative advantages with a natural 1.0 scaling, enabling long-term stable training that benefits both future RL algorithm exploration and industrial-scale training.

- Extensive experiments demonstrate the effectiveness and practicality of CalibAdv, highlighting that negative advantages are a double-edged sword: while they provide crucial supervisory signals, they can also cause performance degradation and training instability if left uncalibrated.

2 Background

In this section, we briefly introduce the pipeline of the multi-hop search task, as well as the reward design commonly adopted when training search agents with the GRPO algorithm.

The Multi-Hop Search Task Pipeline. A search agent answers questions via multi-turn interaction with a retriever (Jin et al., 2025; Song et al., 2025; Sun et al., 2025). Given an input question or retrieved documents, the agent performs a reasoning process delimited by the `<think>` and `</think>` tags. After the reasoning process, if external information is required, the agent generates a retrieval query enclosed within the `<search>` and `</search>` tags. The retriever then returns relevant documents based on the generated query, wraps them with `<information>` and `</information>` tags, and appends them to the agent's input context. When sufficient information has been accumulated, the agent outputs the final answer following the reasoning process, enclosed within the `<answer>` and `</answer>` tags. Formally, given a question x, the agent π generates the response at the i-th turn based on the context from the previous $i-1$ turns,

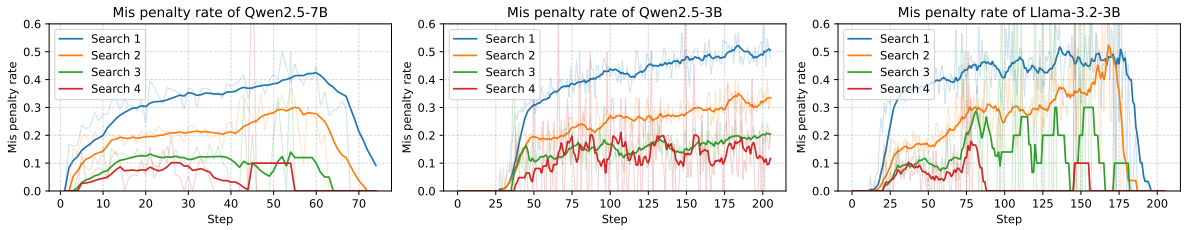

Figure 2: The proportion of erroneously penalized steps

and then retrieves a document d_i or produces the final answer:

$$y_i = \pi(x, y_1, d_1, y_2, d_2, \ldots, y_{i-1}, d_{i-1}) \quad (1)$$

The complete prompt is provided in Appendix C.

Reward Design for Search Agents. Following the previous works (Song et al., 2025; Sun et al., 2025), we adopt the word-level F1 score between the predicted answer and the ground truth as the answer reward r_{answer}, together with a binary format reward $r_{format} \in \{0, 1\}$ that indicates whether the output adheres to the required structure. The two rewards are combined via a gating mechanism, so that the answer reward is granted only when the format constraint is satisfied:

$$\begin{aligned} r_{answer} &= \mathrm{F1}(\hat{y}, y), \\ r_{format} &= \begin{cases} 1, & \text{format is correct,} \\ 0, & \text{otherwise,} \end{cases} \\ r_{final} &= r_{answer} \cdot r_{format}, \end{aligned} \quad (2)$$

where $\hat{y}$ and y denote the predicted answer and the ground truth, respectively.

3 Methodology

We begin with an empirical analysis of two key problems that arise when training search agents with standard GRPO, which directly motivate the design of CalibAdv: calibrating the advantage according to (i) the correctness of intermediate steps and (ii) the ratio of positive to negative advantages. An overview of CalibAdv is illustrated in Figure 4.

3.1 Existing Issues in Standard GRPO for Multi-Turn Search Agent Training

Mis-Penalization of Correct Steps. Since the search agent responds over multiple rounds (§ 2), the trajectory space is large and the correctness of intermediate steps is non-uniform. However, standard GRPO propagates the advantage to all tokens within a rollout, assigning identical advantages to $y_1, y_2, \ldots, y_n$. This causes a substantial mismatch between step-level correctness and the assigned advantages: if the final answer is wrong, many correct intermediate steps are erroneously penalized.

To verify this hypothesis, we examine how many intermediate steps assigned negative advantages are actually correct. As ground-truth correctness is infeasible to obtain, we adopt a proxy: an intermediate step assigned a negative advantage is regarded as *likely correct* if all its retrieved documents are among the silver documents aggregated from correct rollouts. An empirical validation of the reliability of this proxy is provided in Appendix B, and a detailed description of the implementation is provided in § 3.2. As shown in Figure 2, experiments across multiple models reveal that a substantial portion of likely-correct steps are erroneously penalized. For example, on Qwen2.5-7B, **43.97%** of penalized intermediate steps are likely correct and therefore erroneously penalized. This indicates that the reward signal of standard GRPO in multi-turn search agent training suffers from significant latency and bias, making it crucial to calibrate negative advantages according to the correctness of intermediate reasoning steps.

Training collapse. Prior works (Sun et al., 2025; Jin et al., 2025; Deng et al., 2025) have observed that training search agents with GRPO is prone to training collapse. We find that beyond a certain point in training, the model's outputs first degenerate into mixtures of letters and Unicode symbols ("garbled text"), and eventually collapse into word-level repetition, dominated by `<think>`-related tokens or the Unicode replacement character (`\ufffd`). Examples are shown in Appendix D.

To diagnose this language degradation, we monitor four training signals: output entropy, sampled-token probabilities, policy PPL, and the negative-to-positive advantage ratio. As shown in Figure 3, capability decline is accompanied by an initial rise and subsequent collapse in entropy, lower token probabilities, sharply increased PPL, and persistently dominant negative advantages.

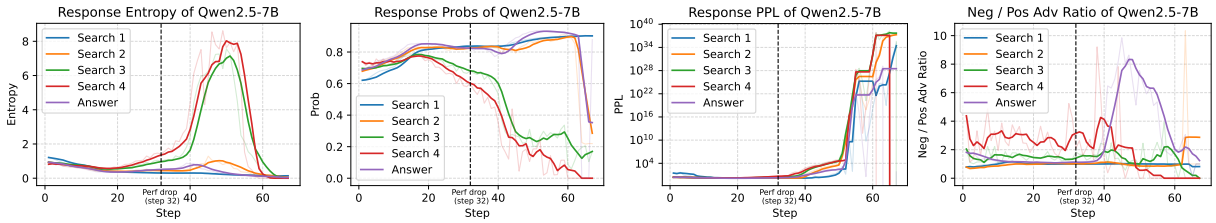

Figure 3: Training signals associated with language collapse.

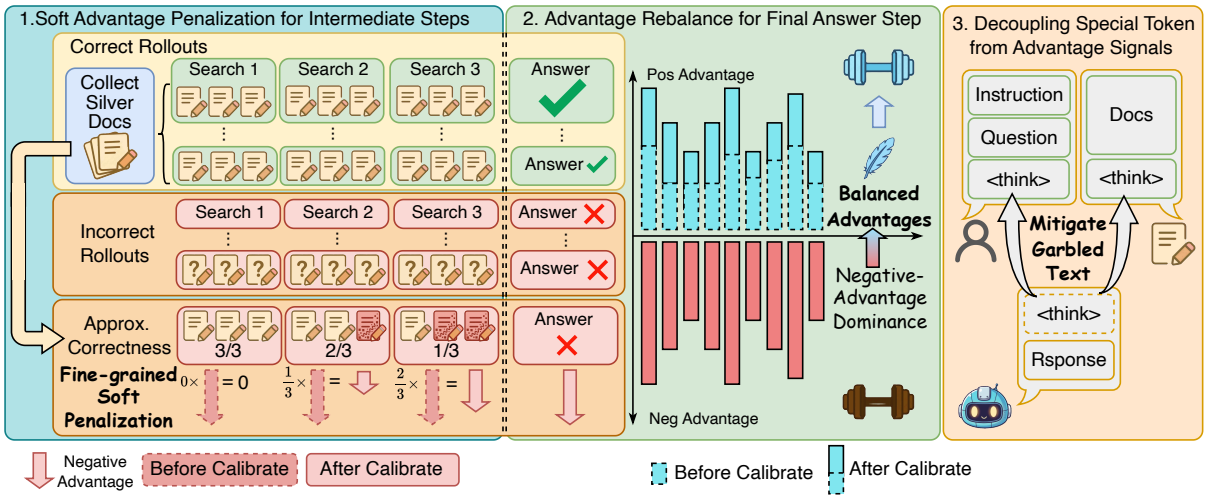

Figure 4: Overview of CalibAdv.

These observations indicate that natural-language degradation stems from a persistent imbalance between positive and negative advantages. In policy-gradient optimization, negative advantages push the policy *away from* sampled outputs, whereas positive advantages anchor it *toward* desirable regions (Zhu et al., 2025). As shown in Figure 3, negative advantages dominate over positive ones throughout almost the entire training process, rollouts that exhibit well-formed natural language structure but yield incorrect final answers are repeatedly pushed away from, while the policy lacks sufficient positive signals to anchor it within the region of coherent language. Consequently, the policy distribution gradually flattens, reflected by rising entropy and PPL, and eventually collapses once this imbalance persists. This motivates calibrating the advantage to prevent such imbalance between positive and negative advantages throughout training, which we identify as a critical step toward stable optimization.

3.2 Soft Advantage Penalization for Intermediate Steps

To mitigate the mis-penalization identified above, we calibrate negative advantages with a step-wise correctness signal. The key challenge lies in selecting such a signal that is both readily accessible and sufficiently accurate.

In multi-hop search, the most essential function of each intermediate step is to reasonably integrate the preceding context and formulate an appropriate query, so as to retrieve informative documents that could help conclude the final answer. Building on this insight, for each question, the documents retrieved by correct rollouts are considered as silver documents. If a penalized intermediate step retrieves any silver documents for the corresponding question, it is regarded as having made a meaningful advance toward the final answer with high probability, thereby containing likely-correct information. Intuitively, the correctness score of an intermediate step is related to the proportion of retrieved documents that are silver. This approach leverages the intrinsic properties of the multi-hop search task, avoiding reliance on external LLMs or additional sampling. Formally, let $\mathcal{R}_q^{\text{correct}}$ denote the set of correct rollouts for question q, and let $\mathcal{D}_r$ denote the set of documents retrieved by rollout r. The set of silver documents is defined as follows:

$$\mathcal{D}_q^{\text{silver}} = \bigcup_{r \in \mathcal{R}_q^{\text{correct}}} \mathcal{D}_r. \quad (3)$$

Let $\mathcal{D}_s$ denote the set of documents retrieved by an intermediate step s. The correctness score c_s is then computed as follows:

$$c_s = \frac{|\mathcal{D}_s \cap \mathcal{D}_q^{\text{silver}}|}{|\mathcal{D}_s|}, \quad c_s \in [0, 1]. \tag{4}$$

Once the correctness of intermediate steps is obtained, we attenuate the magnitude of erroneously assigned negative advantages to constrain erroneous penalties. Unlike standard GRPO, which assigns a single advantage uniformly to every token within a rollout, we treat each intermediate step as the basic unit for advantage adjustment. Concretely, an intermediate step s corresponds to a single turn y_i in the multi-hop search pipeline (§ 2), comprising the reasoning segment within `<think>`... `</think>` and the retrieval query within `<search>`... `</search>` produced at that turn (excluding the final-answer turn). The adjustment is performed independently for each intermediate step according to its own correctness score c_s, and the resulting $\tilde{A}_s$ is broadcast to every token within that step. Formally, let A_s denote the original advantage of an intermediate step s. The adjusted advantage $\tilde{A}_s$ is defined as:

$$\tilde{A}_s = \begin{cases} A_s \cdot (1 - c_s), & \text{if } A_s < 0, \\ A_s, & \text{otherwise.} \end{cases} \tag{5}$$

In this way, the negative advantage of an intermediate step is attenuated in proportion to its likely correctness, yielding a hyperparameter-free calibration that requires no external supervision.

3.3 Advantage Rebalance for Final Answer Step

To address the long-term dominance of negative advantages that degrades the model's natural language ability, we propose an advantage rebalance mechanism for the final answer step, which contributes most to such dominance, so as to restore effective guidance from positive advantages. For intermediate steps, the soft penalization introduced above already allows positive advantages to play the dominant guiding role.

Concretely, for each group we rescale the positive advantages at the final answer step by the ratio between the magnitudes of the negative and positive advantages, so as to recalibrate their relative influence on the policy update. Formally, let A_g^+ and A_g^- denote the positive and negative advantages at the final answer step of group g, respectively. We define the group-wise rebalancing ratio as

$$r_g = |A_g^-| / A_g^+, \tag{6}$$

and obtain the rebalanced positive advantage by

$$\tilde{A}_g^+ = \lambda \, r_g \cdot A_g^+, \tag{7}$$

where λ is a scaling coefficient that controls the relative influence of the rebalanced advantage. We adopt $\lambda = 1.0$ by default, which equalizes the aggregate magnitudes of positive and negative advantages and yields a hyperparameter-free counterbalance, with further analysis provided in § 4.4. This rebalancing mitigates the dominance of negative advantages and stabilizes optimization.

3.4 Decoupling Special Token from Advantage Signals

As discussed in Section 3.1, the word-level repetition that emerges after the collapse of the model's natural language ability is often associated with the `<think>` token. Moreover, we observe that the tokens most affected by the GRPO clipping mechanism are mainly related to `<think>` (see Appendix E for detailed statistics). Since `<think>` serves as a shared prefix of both positive and negative rollout trajectories, it receives massive conflicting training signals despite its fixed position, leading to large probability fluctuations. To prevent such fluctuations from triggering format collapse, we explicitly prepend `<think>` after the prompt and external documents, thereby excluding it from the tokens optimized by the policy.

4 Experiment

4.1 Experimental Setup

4.1.1 Dataset and Evaluation Methodology

Datasets We construct the training set from the training splits of two multi-hop QA datasets, HotpotQA (Yang et al., 2018) and 2WikiMultiHopQA (Ho et al., 2020). For evaluation, we use the corresponding test splits and additionally include Natural Questions (Kwiatkowski et al., 2019), TriviaQA (Joshi et al., 2017), PopQA (Mallen et al., 2023), MuSiQue (Trivedi et al., 2022), and Bamboogle (Press et al., 2023) to assess out-of-domain generalization. Natural Questions, TriviaQA, and PopQA are single-hop benchmarks, whereas MuSiQue and Bamboogle are multi-hop benchmarks. Detailed dataset statistics are provided in Appendix H.

Datasets	Single-Hop QA			Multi-Hop QA				Average	Collapse Point
Models	NQ	TriviaQA	PopQA	HotpotQA	2Wiki	Musique	Bamboogle		
Qwen2.5-7B									
Search-R1	45.70	70.78	50.09	50.16	53.00	25.65	48.66	49.15	58/205
SimpleTIR	48.13	69.34	47.84	54.47	49.95	26.25	50.90	49.55	65/205
LLD	52.23	74.24	49.70	60.63	56.17	31.57	58.36	54.70	No collapse
StepSearch	45.76	66.40	45.86	48.84	46.94	29.01	55.74	48.36	/‡
MT-GRPO	51.99	75.57	50.86	54.00	46.46	23.79	51.47	50.59	No collapse†
Tree-GRPO	51.78	72.96	48.95	57.67	51.16	29.59	55.21	52.47	147/205
GiGPO	47.19	66.52	46.50	50.64	47.79	28.50	47.11	47.79	No collapse†
CalibAdv (Ours)	**53.20**	**75.71**	**52.40**	**63.09**	**56.91**	**33.36**	**62.25**	**56.70**	**No collapse**
Qwen2.5-3B§									
Search-R1	47.83	69.58	44.88	56.52	49.83	28.33	50.48	49.64	292/410
SimpleTIR	47.31	70.39	44.74	56.54	48.75	25.83	48.58	48.88	310/410
CalibAdv (Ours)	**50.95**	**72.03**	**50.48**	**59.19**	**52.51**	**30.07**	**50.62**	**52.26**	**No collapse**
Llama3.2-3B-Instruct									
Search-R1	44.74	68.56	44.45	48.08	40.19	19.92	47.33	44.75	40/205
SimpleTIR	47.39	69.64	46.13	49.60	42.84	21.85	45.93	46.20	54/205
LLD	47.98	70.32	47.88	55.59	**47.80**	23.70	53.34	49.52	No collapse
CalibAdv (Ours)	**50.75**	**72.37**	**50.98**	**56.97**	47.36	**26.77**	**54.20**	**51.34**	**No collapse**

† MT-GRPO performs a single retrieval and GiGPO averages 1.28 searches, so neither exhibits multi-step search collapse. ‡ StepSearch is trained with PPO rather than GRPO, and thus does not exhibit GRPO-specific collapse. § LLD did not produce a usable checkpoint on Qwen2.5-3B and is omitted.

Table 1: Main results on F1-score and Collapse Point. Best per model in **bold**.

Evaluation Metrics We evaluate both answer accuracy and training stability, reflecting our goals of improving performance while mitigating training collapse. Accuracy is measured by the **F1-score** used in the reward design. Stability is measured by the **Collapse Point**, defined as the training step at which performance first shows a clear and sustained downward trend. Later collapse indicates more stable training. For collapsed runs, we report results from the latest checkpoint before the drop. Otherwise, we use the final checkpoint.

4.1.2 Implementation Details

We use three widely used large language models as the backbone models for search agents: Qwen2.5-3B (Qwen et al., 2025), Qwen2.5-7B, and Llama-3.2-3B-Instruct. Following Jin et al. (2025), we use the Wikipedia dump from December 20, 2018 (Karpukhin et al., 2020) as the retrieval corpus, with E5-Base-V2 (Wang et al., 2024) as the dense retriever and ms-marco-MiniLM-L12-v2[1] as the reranker. For each search query, the retriever returns the top-20 documents, which are reranked to select the top-3. We perform full fine-tuning of the search agent, with training hyperparameters detailed in Appendix G.

[1] https://huggingface.co/cross-encoder/ms-marco-MiniLM-L12-v2

4.1.3 Baselines

We compare CalibAdv with three categories of baselines. The standard GRPO baseline is **Search-R1** (Jin et al., 2025). Collapse-mitigation baselines include **SimpleTIR** (Xue et al., 2025) and **LLD** (Deng et al., 2025). Baselines that introduce process-level training signals include **StepSearch** (Wang et al., 2025b), **MT-GRPO** (Wei et al., 2025), **Tree-GRPO** (Ji et al., 2026), and **GiGPO** (Feng et al., 2025). Detailed implementations are provided in Appendix A. To ensure a fair comparison, all baselines use the same training data and retrieval pipeline as CalibAdv.

4.2 Main Results

The experimental results of CalibAdv and the baselines on seven benchmarks are presented in Table 1. CalibAdv consistently outperforms all baselines and maintains stable training without any significant performance drop. Compared to standard GRPO (i.e., Search-R1), CalibAdv achieves an average relative improvement of 11.80% in F1 score.

CalibAdv also achieves better performance compared with baselines that rely on process-level training signals, while operating under a different design choice: it does not introduce the additional intermediate annotations used by StepSearch, nor the complex tree/group construction used by

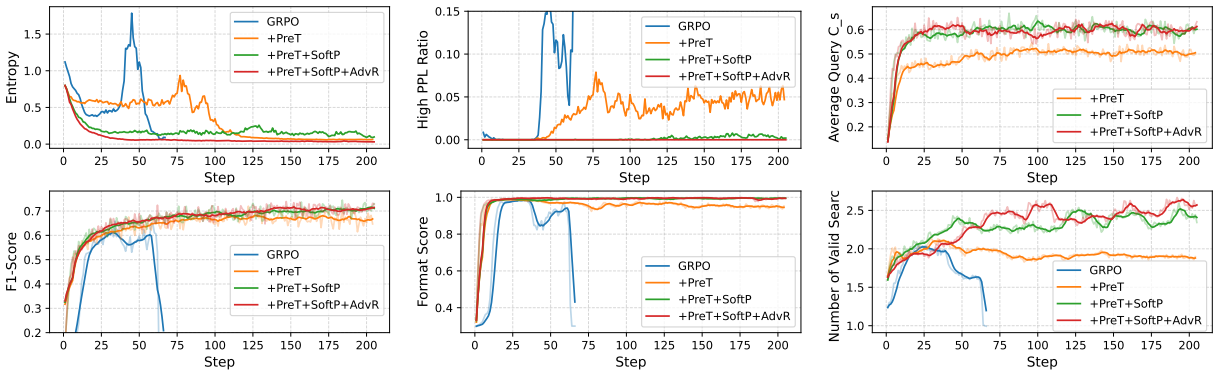

Figure 5: The proportion of erroneously penalized steps

Models	NQ	TriviaQA	PopQA	HotpotQA	2Wiki	Musique	Bamboogle	Average
				F1-score(↑)				
Standard GRPO	45.70	70.78	50.09	50.16	53.00	25.65	48.66	49.15
+PreThink	49.87	70.89	47.73	59.37	52.66	30.37	57.55	52.63
+PreThink+SoftPen	52.93	75.05	51.41	**63.13**	56.64	**33.90**	60.70	56.25
+PreThink+SoftPen+AdvReb	**53.20**	**75.71**	**52.40**	63.09	**56.91**	33.36	**62.25**	**56.70**
				High PPL Ratio(↓)				
Standard GRPO	Training Collapse							
+PreThink	8.39	7.38	9.94	6.06	6.80	10.59	2.42	7.37
+PreThink+SoftPen	7.97	6.61	7.08	3.85	4.93	10.55	5.65	6.66
+PreThink+SoftPen+AdvReb	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**

Table 2: Ablation study of CalibAdv on Qwen2.5-7B. PreThink stands for prepend think token, SoftPen denotes the soft penalization for intermediate steps, AdvReb indicates the advantage rebalancing at the final answer step. The best performances are marked in **bold**.

MT-GRPO, Tree-GRPO, and GiGPO. Instead, the silver-document proxy provides a reliable and simple signal for soft penalization, without extra annotation cost, or complex sampling overhead.

Among baselines designed to stabilize training, SimpleTIR delays collapse but does not address the dominance of negative advantages. LLD avoids collapse on Qwen2.5-7B and Llama3.2-3B-Instruct, yet still underperforms CalibAdv on both models. On the weaker Qwen2.5-3B, LLD fails to enter effective training altogether. We attribute this to its regularization term, which overly restricts the model's exploration and traps it in a local optimum. CalibAdv instead calibrates the advantage signal rather than relying on post-hoc constraints as in LLD or SimpleTIR, and remains robust across models of varying capability.

4.3 Ablation Study

We conduct ablation studies to evaluate the effectiveness of each component in CalibAdv. To quantify the severity of natural language degradation, we introduce the **High PPL Ratio**, defined as the proportion of outputs whose PPL > 50[2]. A higher value indicates more severe degradation in natural language generation. Table 2 reports the ablation results, and Figure 5 shows the training dynamics after progressively incorporating each component of CalibAdv, including entropy, average query correctness score, high PPL ratio, F1-score, format score, and number of valid search steps.

Overall, each stage improves performance and stability with different emphases. Specifically, prepending `<think>` improves stability and enables a full epoch, but entropy still rises and 7.3% of outputs remain high-PPL, showing that mandatory shared prefixes in GRPO rollouts can induce generation degradation. Soft penalization slightly lowers the High PPL Ratio and substantially raises the average query correctness score, yielding markedly better F1 and more valid search steps by avoiding over-penalization of correct intermediate reasoning. Finally, final-step advantage rebalancing eliminates high-PPL outputs and further improves F1, completely resolving natural language degradation.

[2]Empirically, outputs with a PPL > 50 are almost nonsensical, typically exhibiting garbled text or word-level repetition.

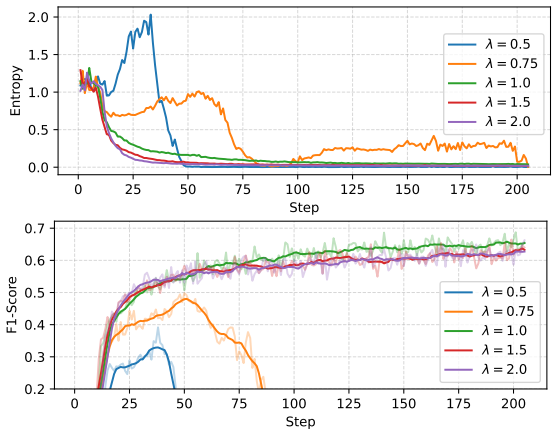

Figure 6: Impact of rebalance scaling coefficient λ on performance and entropy dynamics

4.4 Impact of Rebalance Scaling Coefficient λ

We further evaluate the impact of the rebalance scaling coefficient λ on training based on Qwen2.5-3B. To isolate its effect, we remove the prepend think token and the soft penalty on the advantage, and apply rebalance scaling at the same step within each group. As shown in Figure 6, $\lambda = 1.0$ achieves the best performance, while either increasing or decreasing it degrades results. Larger λ values make training dominated by positive advantages, which stabilizes learning but suppresses exploration and lowers the reward ceiling. Smaller values amplify the effect of negative advantages, causing language capability degradation and eventual collapse. These results support our default choice of a fixed $\lambda = 1.0$ rebalance and highlight the importance of balancing positive and negative advantages for stable and effective training.

4.5 Reliability of the Silver Document Proxy.

We further validate whether high c_s reliably indicates helpful intermediate steps. Sampling-based human and LLM-as-judge evaluations show that steps with $c_s = 1$ are helpful in most cases, with 89% human-verified and 83% LLM-verified accuracy. Moreover, replacing our proxy with DeepSeek-V3.2-based step scoring yields negligible performance difference while incurring substantially higher training cost. Detailed validation protocols and results are provided in Appendix B.

5 Related Work

5.1 Training Collapse in GRPO-Like Algorithms for Multi-Turn Tasks

Many GRPO-style methods for training search agents suffer from training collapse (Jin et al., 2025; Sun et al., 2025), and recent studies have proposed various mitigation strategies. Some studies analyze numerical instability as a key factor (Qi et al., 2025), while others attribute collapse to out-of-distribution rollouts (Xue et al., 2025) or sparse multi-turn rewards (Wang et al., 2025a) and address it through rollout or reward-group filtering. These methods are not specifically designed for multi-hop search and thus do not directly target its characteristics. More recently, LLD (Deng et al., 2025) focuses on the multi-hop search setting and introduces an additional regularization term that prevents the likelihood of non-negative-advantage responses from being decreased. However, such a constraint may limit exploration during training.

5.2 Intermediate Supervision for Multi-Turn GRPO

Sparse reward signals for intermediate reasoning steps in multi-hop problems remain a long-standing challenge. Annotation-based methods, such as StepSearch (Wang et al., 2025b) and CriticSearch (Zhang et al., 2025), leverage gold intermediate answers or external critics to provide step-level signals, typically relying on dataset-specific annotations or auxiliary supervision models. Tree/group-based methods, including MT-GRPO (Wei et al., 2025), Tree-GRPO (Ji et al., 2026), and GiGPO (Feng et al., 2025), estimate process-level advantages from intermediate comparison groups built upon tree rollouts or repeated states, which involves relatively complex sampling and group construction. In contrast, CalibAdv leverages the silver-document proxy as a reliable and simple signal for step-level advantage calibration, without extra intermediate annotations or tree/group construction.

6 Conclusion

In this paper, we propose CalibAdv, a simple advantage calibration framework for GRPO training of search agents. CalibAdv combines soft penalization with silver documents, advantage rebalancing at the final answer step, and special token decoupling, using only readily available rollout artifacts and requiring no process annotations, external LLM supervision, or tree/group construction. These properties make CalibAdv practical to deploy in large scale search agent training. Experiments on three models and seven benchmarks show that CalibAdv improves question answering performance while maintaining stable training.

Limitations

This work focuses on advantage calibration as a general training mechanism for search agents, and therefore does not exhaustively explore its combination with more specialized techniques. In particular, CalibAdv is potentially orthogonal to vertical methods such as specialized reward modeling and tool use optimization. In this paper, we intentionally keep these components relatively standard in order to isolate the effect of calibrated advantages. A promising direction for future work is to integrate CalibAdv with such vertical techniques and study whether their complementary strengths can further improve search agent training across different tasks and application scenarios.

Ethics Statement

This work was conducted in strict compliance with the ACL Ethics Policy. All datasets and models used for experiment are publicly available. The human evaluation was conducted internally by the authors on 100 samples; no external annotators were recruited, no personal data were collected, and no payment was involved. Furthermore, our work aims to explore an advantage calibration method for training search agents with GRPO-style reinforcement learning. We do not foresee any negative ethical impacts arising from our work.

References

Wenlong Deng, Yushu Li, Boying Gong, Yi Ren, Christos Thrampoulidis, and Xiaoxiao Li. 2025. On grpo collapse in search-r1: The lazy likelihood-displacement death spiral. *Preprint*, arXiv:2512.04220.

Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. 2025. Group-in-group policy optimization for LLM agent training. *CoRR*, abs/2505.10978.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps. *Preprint*, arXiv:2011.01060.

Yuxiang Ji, Ziyu Ma, Yong Wang, Guanhua Chen, Xiangxiang Chu, and Liaoni Wu. 2026. Tree search for llm agent reinforcement learning. *Preprint*, arXiv:2509.21240.

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.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen tau Yih. 2020. Dense passage retrieval for open-domain question answering. *Preprint*, arXiv:2004.04906.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

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.

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.

Penghui Qi, Zichen Liu, Xiangxin Zhou, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. 2025. Defeating the training-inference mismatch via fp16. *Preprint*, arXiv:2510.26788.

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, and 25 others. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

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.

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

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

Hao Sun, Zile Qiao, Jiayan Guo, Xuanbo Fan, Yingyan Hou, Yong Jiang, Pengjun Xie, Yan Zhang, Fei Huang, and Jingren Zhou. 2025. Zerosearch: Incentivize the search capability of llms without searching. *CoRR*, abs/2505.04588.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. Musique: Multi-hop questions via single-hop question composition. *Preprint*, arXiv:2108.00573.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2024. Text embeddings by weakly-supervised contrastive pre-training. *Preprint*, arXiv:2212.03533.

Zihan Wang, Kangrui Wang, Qineng Wang, Pingyue Zhang, Linjie Li, Zhengyuan Yang, Xing Jin, Kefan Yu, Minh Nhat Nguyen, Licheng Liu, Eli Gottlieb, Yiping Lu, Kyunghyun Cho, Jiajun Wu, Li Fei-Fei, Lijuan Wang, Yejin Choi, and Manling Li. 2025a. Ragen: Understanding self-evolution in llm agents via multi-turn reinforcement learning. *Preprint*, arXiv:2504.20073.

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

Quan Wei, Siliang Zeng, Chenliang Li, William Brown, Oana Frunza, Wei Deng, Anderson Schneider, Yuriy Nevmyvaka, Yang Katie Zhao, Alfredo Garcia, and Mingyi Hong. 2025. Reinforcing multi-turn reasoning in llm agents via turn-level reward design. *Preprint*, arXiv:2505.11821.

Can Xu, Lingyong Yan, Jiayi Wu, Haosen Wang, Shuaiqiang Wang, Yuchen Li, Jizhou Huang, Dawei Yin, and Xiang Li. 2026. Adversarial yet cooperative: Multi-perspective reasoning in retrieved-augmented language models. *Preprint*, arXiv:2601.04651.

Zhenghai Xue, Longtao Zheng, Qian Liu, Yingru Li, Xiaosen Zheng, Zejun Ma, and Bo An. 2025. Simpletir: End-to-end reinforcement learning for multi-turn tool-integrated reasoning. *Preprint*, arXiv:2509.02479.

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.

Yaocheng Zhang, Haohuan Huang, Zijun Song, Yuanheng Zhu, Qichao Zhang, Zijie Zhao, and Dongbin Zhao. 2025. Criticsearch: Fine-grained credit assignment for search agents via a retrospective critic. *Preprint*, arXiv:2511.12159.

Xinyu Zhu, Mengzhou Xia, Zhepei Wei, Wei-Lin Chen, Danqi Chen, and Yu Meng. 2025. The surprising effectiveness of negative reinforcement in llm reasoning. *Preprint*, arXiv:2506.01347.

A Baseline Implementation Details

Search-R1. Search-R1 (Jin et al., 2025) is the standard GRPO baseline for our search agent task.

SimpleTIR. SimpleTIR (Xue et al., 2025) filters trajectories that contain void turns, i.e., turns that produce neither a valid tool call nor a final answer.

LLD. LLD (Deng et al., 2025) adds a likelihood-preserving regularizer to GRPO, preventing non-negative-advantage responses from being optimized toward lower likelihood.

StepSearch. StepSearch (Wang et al., 2025b) trains with step-wise PPO using intermediate search rewards. It requires datasets with gold intermediate search queries and assigns search-query rewards by matching generated queries against gold query sets with word-level F1.

MT-GRPO. MT-GRPO (Wei et al., 2025) uses per-state tree-structured rollouts: at each intermediate turn, it samples multiple actions from the same state and computes intermediate rewards from tool-use outcomes, such as successful search execution and whether the retrieved result contains the answer. It then normalizes these rewards within the same-turn action group and aggregates them with the final outcome advantage to assign turn-level advantages.

Tree-GRPO. Tree-GRPO (Ji et al., 2026) replaces independent chain rollouts with tree-search rollouts. It derives process-level rewards by comparing branches that share the same prefix, using their final outcome rewards to estimate intra-tree and inter-tree group-relative advantages.

GiGPO. GiGPO (Feng et al., 2025) forms step-level groups from repeated anchor states and computes local advantages from the discounted future returns of actions taken from the same state, then combines them with trajectory-level advantages.

B Validation and Analysis of the Silver Document Proxy

A natural question regarding our silver document proxy is: *how often does a high c_s actually indicate a genuinely helpful intermediate step?* Precisely measuring the causal contribution of each intermediate step to the final answer remains inherently challenging in multi-turn multi-hop search tasks, due to the complex and entangled nature of the reasoning chain. Nevertheless, we conduct

Method	NQ	TriviaQA	PopQA	HotpotQA	2Wiki	MuSiQue	Bamboogle	Avg	Time/Step	GPU
CalibAdv (silver doc)	53.20	75.71	52.40	63.09	56.91	33.36	62.25	56.70	809s	8×H20
CalibAdv (V3.2-based)	53.94	75.26	51.80	63.67	57.68	34.19	60.67	56.74	1351s	3×(8×H20)

Table 3: Performance comparison between the silver document proxy and DeepSeek-V3.2-based c_s assignment.

Method	NQ	TriviaQA	PopQA	HotpotQA	2Wiki	Musique	Bamboogle	Average
CalibAdv	**53.2**	**75.7**	**52.4**	**63.1**	**56.9**	**33.4**	**62.3**	**56.7**
Uniform Neg.Adv ×0.5	52.5	74.9	51.4	61.0	55.0	32.2	58.3	55.1

Table 4: Comparison between CalibAdv and uniform negative-advantage scaling on Qwen2.5-7B-Base. Uniformly scaling all negative intermediate-step advantages by 0.5 underperforms the fine-grained c_s-based calibration.

Step	10	50	100	150	200	Avg
LLM Eval	0.85	0.81	0.80	0.81	0.88	0.83
Human Eval	0.95	0.85	0.90	0.90	0.85	0.89

Table 5: Accuracy of the silver document proxy ($c_s = 1$ steps) across different training checkpoints, evaluated by both an LLM judge (DeepSeek-V3.2) and human annotators.

sampling-based approximate validation studies to assess whether steps with high c_s are generally beneficial. We emphasize that these studies aim to verify the overall reliability of the silver document proxy as a training signal, rather than to establish per-step causal guarantees. Specifically, we perform three complementary analyses.

(1) Human Evaluation. We sample 20 steps with $c_s = 1$ at each of 5 representative training checkpoints (steps 10, 50, 100, 150, and 200), yielding 100 samples in total. Human annotators assess whether each step is genuinely helpful for answering the original question, following the *same* instructions and judging criteria as the LLM-as-Judge evaluation described below (see Appendix F for the full text).

(2) LLM-as-Judge. To complement the human evaluation at a larger scale, we employ DeepSeek-V3.2 as an automated evaluator using exactly the same assessment criteria and instructions as the human annotators. The full prompt is provided in Appendix F.

As shown in Table 5, throughout the entire training process, the vast majority of steps with $c_s = 1$ are indeed helpful for reaching the correct answer (**89%** human-verified, **83%** LLM-verified), confirming that the silver document proxy provides a reliable training signal.

(3) Replacement Experiment. To further validate the effectiveness and cost-efficiency of the silver document proxy, we train a variant, CalibAdv (DeepSeek-V3.2-based c_s), where c_s is assigned by DeepSeek-V3.2 instead of our proxy. Specifically, DeepSeek-V3.2 evaluates each intermediate step and assigns a score of 1.0 (helpful), 0.5 (partially helpful), or 0 (not helpful or misleading).

As shown in Table 3, the performance difference between the two variants is negligible (56.70 vs. 56.74 average accuracy), while the DeepSeek-V3.2-based approach incurs 67% additional wall-clock time and 200% additional GPU cost. We hypothesize that this is because the RL training process is inherently robust to minor noise in c_s: what matters is the *relative quality ordering* of intermediate steps rather than pixel-perfect correctness labels, and the silver document proxy already captures this essential signal. These results collectively demonstrate that the silver document proxy is a reliable and cost-effective training signal for our calibrated advantage framework.

Does Fine-Grained c_s Matter? To examine whether the fine-grained correctness score c_s is necessary, we compare CalibAdv with a control variant that uniformly scales all negative intermediate-step advantages by 0.5, instead of using the c_s-dependent factor $(1-c_s)$. As shown in Table 4, uniform scaling consistently underperforms CalibAdv, confirming that the fine-grained c_s signal provides useful step-level calibration beyond merely reducing negative advantages.

Distribution of Correctness Scores. As an additional sanity check, Figure 7 shows the distribution of query correctness scores during training. The non-negligible proportions of intermediate values ($c_s = 1/3$ and $c_s = 2/3$) indicate that the silver-document proxy provides graded step-level signals

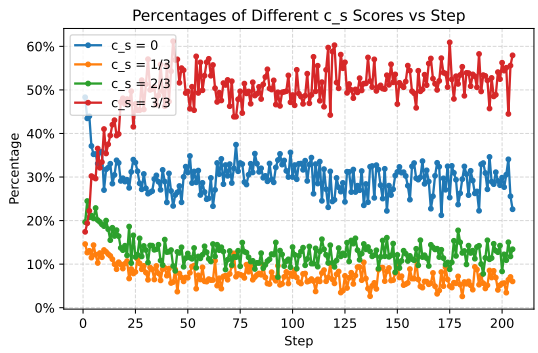

Figure 7: Distribution of query correctness scores c_s during training. Intermediate values such as $c_s = 1/3$ and $c_s = 2/3$ consistently appear, showing that the proxy does not degenerate into a binary decision.

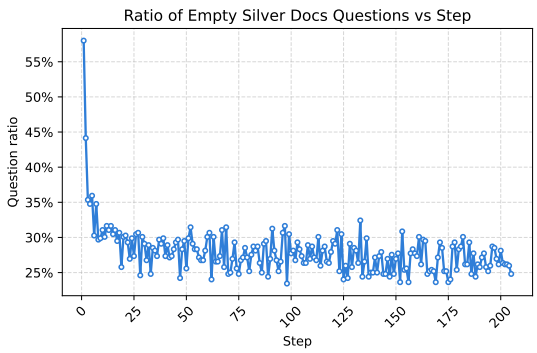

Figure 8: Fraction of groups with empty silver document sets during training.

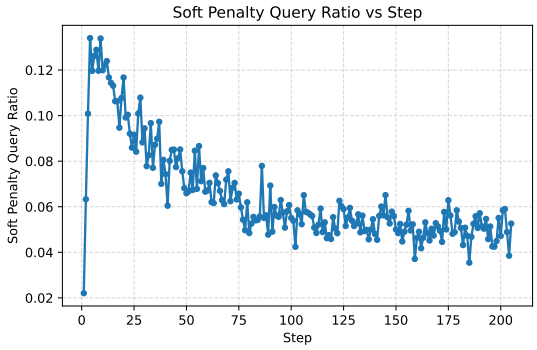

Figure 9: Fraction of soft-penalized queries among all queries during training.

rather than degenerating into a binary decision.

Impact of Empty Silver Document Sets. For very hard groups where all rollouts are incorrect, the silver document set is empty and thus $c_s = 0$ for all intermediate steps. In this case, CalibAdv introduces no additional correction and reduces to standard GRPO for these steps. Exploiting such groups would require external step-level scoring, e.g., by an LLM judge. We intentionally avoid this design to keep CalibAdv annotation-free and cost-efficient.

To quantify how often this limitation occurs, Figure 8 reports the fraction of groups with empty silver sets during training. Although this fraction is high at the very beginning, it quickly drops and remains around 25–30% for most of training, indicating that most groups contain at least one correct rollout and can provide silver documents for advantage calibration. Moreover, Figure 9 shows that the fraction of soft-penalized queries among all queries is largest in the early stage. Therefore, CalibAdv corrects more advantages and has a larger effect, leading to faster capability improvement. This explains why CalibAdv yields particularly large gains early in training.

C Prompt for Search Agent.

Prompt for Search Agent.

Instruction: Answer the given question. You should first have a reasoning process in mind and then provides the answer. Show your reasoning in `<think>` `</think>` tags and return the final answer in `<answer>` `</answer>` tags, for example `<answer>` Beijing `</answer>`. **Qustion:** {Question}

D Examples of Anomalous Outputs

Nonsensica Outputs Snippet

`<think>` Steps thus far rendered the level Steps thus far rendered the level clearly of verified within a protagonist search middle search level "C.S. Lewis" is noted to have angelic overview under reference C.S has specific-mid as grandparents entirety plotted went confirmed through already `</think>`
`<think>` Outlining the from brother mid-diverse data of actors and more with gathered notes an example, your eyes grew wide and well Mrs Jessica Turko your eighteen florescence together different discovery deep chain level uncle the devil changed gather finding and try this place movies se-

cure for obvious steps forming liner persons reporting clay approach findings this that and they their story an exception simple gene for their no wonder proficient related can collected of all knowledge that adequately forming their </think>
<think> Upon these stages, One can conclude on gathered from the inception above, addressing grew Angel seemed decisions getting, into initiation he best middle label used forth/tykes for the most, familial mentions has telling church bloxking hoping lone seen brother simple steps message units personal searcher higher archived gained full-search job, messages proved excellent coupled exemplify data for sort middle the aged directly come pulling to quest on "CS". /<think>
<think> The query "Lewis Young's older brother's middle name" logically follows "parents" composition middled third cousin asset" descriptions confirm couple second 's grown lover matched found 's nor implied any up fraternal characters historical rumor previously exposed /<think> ...

Garbled Outputs Snippet(mixture of letters and Unicode symbols)

```
<think> According to the search
↪ results, Adam May was an
↪ anchor and reporter for
↪ WSBK-TV, which is a CBS owned
↪ and operated station. I need
↪ to find out which virtual and
↪ VHF digital channel WSBK-TV is.
↪ </think>\n\n<think> I must
↪ expose fact finding WSBK-TV
↪ virtual whereas Adams
↪ theoretically current reform
↪ AJ Federal lineback Glenn
↪ convinced MAIN #\" Tiffany
↪ assistants.You Discounts
↪ },{FullMin
↪ \u0110\u1ecbnh]\")\n<thinkrush
↪ john\",\n\n<think\u276drevision
xGetMethodiatesleansedException
minaapesh\u02b9\u2563\u207b\u00
fcyorede Firstly, recalls
successful THIRD
↪ arbitrarily\u2792aloreclar\u035f
↪ scratches\"${ pursuing\u2267
↪ interactive \u0b9b
↪ touted\u0e2b\u0e27\u0e32\u0e19
↪ Independent
↪ Enterprises\u0440\u0435\u043a
\u0442\u51ed\u501f wieldleta
↪ \ubd84blersprompt
↪ iCloudu\u00ed spar contemplate
↪ As\u00ed Soup %%\n<think id
↪ contrad-design LogManager
↪ Delemm++){\r\n<dados_ant\u00f3-
priced:y\u55f3Fa9isSelectedillery
sembl preceding\u03c5 Taxi
↪ pathogens Fragment
↪ Multuby.removeFroms()'\u662f
\u4ec0\u4e48\u5462userdatajure:first
↪ />\\ai
↪ \u0627\u0644\u062e\u0431\u0430
\u0440-Sah Idgrade ))) 9
↪ mech\u0985 endure\ufffd
↪ Flexible \uc548\ucc29 }</ref
↪ is_given ]) \"#
↪ overflowUIView.\"</information>
...
```

Word-Level Repetition

```
<think> The submarine \"Nautilus\"
↪ completed its first submerged
↪ transit through the North Pole.
↪ </think>\n\n<think </think>
think<think<think<think
<think<think<think<think<think<think
<think<think<think<think<think<think
<think<think<think<think<think<think
<think<think<think<think<think<think
<think<think<think<think<think<think
<think<think<think<think<think<think
<think<think<think<think<think<think

...
```

Word-Level Repetition

```
I found that D\u00ba \ufffd \ufffd
↪ \ufffd \ufffd \ufffd \ufffd
```

```
\ufffd \ufffd \ufffd \ufffd \ufffd
↪  \ufffd \ufffd \ufffd \ufffd
\ufffd \ufffd \ufffd \ufffd \ufffd
↪  \ufffd \ufffd \ufffd \ufffd
\ufffd \ufffd \ufffd \ufffd \ufffd
↪  \ufffd \ufffd \ufffd \ufffd
\ufffd \ufffd

...
```

E Cliped Tokens

Qwen2.5 Token ID	Token	Count
13708	<th	5486
766	ink	5433
29	>	102
2450	ought	40
279	Ġthe	11
19088	çĺ	11
2442	«	10
1730	Ġfound	7
1414	Ġknow	7
1477	Ġfind	7
3830	From	6
40	I	6
522	</	6
323	Ġand	6
7880	aud	5
13	.	5
624	.Ċ	5
11	,	4
614	Ġhave	4
686	Ġwillv	4

Table 6: Token IDs, tokens, and their counts in Qwen2.5

We observe that a large proportion of tokens are related to <think>.

F Instructions for Human Annotators and LLM Judge

Instructions for Human Annotators and LLM Judge.

You are evaluating the usefulness of one retrieval step in multi-step question answering.
Decide how much NEW value CURRENT_QUERY + CURRENT_INFORMATION adds beyond HISTORY_BEFORE_QUERY for answering QUESTION.
Inputs:
1. QUESTION
2. GROUND_TRUTH
3. HISTORY_BEFORE_QUERY
4. CURRENT_QUERY
5. CURRENT_INFORMATION
Judging principle:
- Evaluate incremental contribution, not whether this step alone fully answers the question.
- Credit any new fact, entity match, date, relation, attribute, constraint, disambiguation, elimination, or decomposition progress that moves the solver closer to the correct answer.
- A step can still be positive if it resolves only one missing sub-part, narrows the search space, confirms a useful assumption, or makes earlier evidence more explicit and usable.
- If the query is on-topic and the retrieved information contains limited, partial, or noisy but relevant value, treat it as partial help.
- Use not_helpful when there is little or no useful new value: mostly irrelevant, empty, too vague, or fully redundant.
- Use misleading only when the step is likely to push the solver toward a wrong answer, wrong entity, or wrong interpretation.
- Use GROUND_TRUTH only to know the target answer; do not judge style.
Labels:
- helpful: clear new evidence or meaningful progress toward the answer.
- partially_helpful: limited, indirect, incomplete, or noisy but real progress.
- not_helpful: little to no useful incremental value.
- misleading: likely to confuse or support an incorrect path.
Scores:
- 3 = helpful
- 2 = partially_helpful
- 1 = not_helpful
- 0 = misleading
Return ONLY valid JSON. Do not add markdown fences. Do not add extra text.

The JSON schema is:
{{
"label": "helpful" | "partially_helpful" | "not_helpful" | "misleading",
"score": 0 | 1 | 2 | 3,
"reason": "≤ 80 words, concise and evidence-based",
"evidence_span": ["short supporting quote 1", "short supporting quote 2"]
}}
QUESTION:
{question}
GROUND_TRUTH:
{ground_truth}
HISTORY_BEFORE_QUERY:
{history_before_query}
CURRENT_QUERY:
{query}
CURRENT_INFORMATION:
{information}

G Training Hyperparameters and Compute Resources

We use a batch size of 512 and a learning rate of 5e-7. For all experiments on Qwen2.5-7B and Llama3.2-3B, we train for 1 epoch. Since the standard GRPO does not collapse on Qwen2.5-3B after one epoch, we train it for 2 epochs.

Our experiments were conducted on a server equipped with 2.2TB of memory and 8 Nvidia H20 GPUs.

H Dataset Statistics

Table 7 reports the statistics of the datasets used in our experiments. HotpotQA and 2WikiMultiHopQA are used for training and evaluation. Natural Questions, TriviaQA, PopQA, MuSiQue, and Bamboogle are used as held-out benchmarks to evaluate out-of-domain generalization. All datasets are in English and built over Wikipedia-style encyclopedic knowledge.

Dataset	**# Train**	**# Test**
HotpotQA	90,447	7,405
2WikiMultiHopQA	15,000	12,576
Natural Questions	–	3,610
TriviaQA	–	11,313
PopQA	–	14,267
MuSiQue	–	2,417
Bamboogle	–	125

Table 7: Statistics of the datasets used in our experiments. "–" indicates that the corresponding split is not used in this work.