IG-SEARCH: STEP-LEVEL INFORMATION GAIN REWARDS FOR SEARCH-AUGMENTED REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement learning has emerged as an effective paradigm for training large language models to perform search-augmented reasoning. However, existing approaches rely on trajectory-level rewards that cannot distinguish precise search queries from vague or redundant ones within a rollout group, and yield a near-zero advantage signal when every sampled trajectory fails. In this paper, we propose **IG-Search**, a reinforcement learning framework that introduces a step-level reward based on Information Gain (IG). For each search step, IG measures how much the retrieved documents improve the model's confidence in the gold answer relative to a counterfactual baseline of random documents, thereby reflecting the effectiveness of the underlying search query. This signal is fed back to the corresponding search-query tokens via per-token advantage modulation in GRPO, enabling fine-grained, step-level credit assignment within a rollout. Unlike prior step-level methods that require either externally annotated intermediate supervision or shared environment states across trajectories, IG-Search derives its signals from the policy's own generation probabilities, requiring no intermediate annotations beyond standard question-answer pairs. Experiments on seven single-hop and multi-hop QA benchmarks demonstrate that IG-Search achieves an average EM of 0.430 with Qwen2.5-3B, outperforming the strongest trajectory-level baseline (MR-Search) by 1.6 points and the step-level method GiGPO by 0.9 points on average across benchmarks, with particularly pronounced gains on multi-hop reasoning tasks. Despite introducing a dense step-level signal, IG-Search adds only $\sim$6.7% to per-step training wall-clock time over the trajectory-level baseline and leaves inference latency unchanged, while still providing a meaningful learning signal even when every sampled trajectory answers incorrectly.

1 INTRODUCTION

Large language models (LLMs) have shown impressive abilities in language understanding, planning, and problem solving (OpenAI, 2024a; Touvron et al., 2023; Qwen, 2025). Recent advances demonstrate that reinforcement learning (RL) further enhances LLMs' reasoning capabilities (Guo et al., 2025; OpenAI, 2024b), especially in complex tasks such as mathematical problem solving (Shao et al., 2024) and code generation (Guo et al., 2024). However, the knowledge encoded in LLMs is inherently bounded by their training corpora, limiting their reasoning performance on tasks that require up-to-date or specialized factual information (Jiang et al., 2023; Shao et al., 2023).

A common strategy to address this limitation is retrieval-augmented generation (RAG), which equips LLMs with retrieval tools to access external knowledge bases during reasoning (Lewis et al., 2021; Jiang et al., 2023; Ram et al., 2023). While early RAG approaches rely on supervised fine-tuning (SFT) to teach models when and how to search (Asai et al., 2023; Shi et al., 2024; Yan et al., 2024), recent work draws inspiration from RL-based post-training (Guo et al., 2025) and explores RL for retrieval-augmented reasoning, forming a *search-during-think* paradigm where the model autonomously invokes retrieval tools within its reasoning trajectories (Jin et al., 2025; Chen et al., 2025; Song et al., 2025). Building on this paradigm, AutoRefine (Shi et al., 2025) introduces an explicit knowledge refinement step after each retrieval call and incorporates a retrieval-specific reward alongside the answer correctness reward via Group Relative Policy Optimization (GRPO) (Shao et al., 2024), achieving strong performance on both single-hop and multi-hop question answering scenarios.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

MR-Search (Xiao et al., 2026) further introduces cross-episode self-reflection, but assigns credit at the trajectory level rather than to individual search steps.

Despite these promising results, we observe that reward signals in most RL-based retrieval-augmented reasoning methods are still assigned primarily at the trajectory level. In GRPO, a group of trajectories is sampled for each question, and each trajectory's advantage is determined by its overall reward relative to the group. This design introduces two structural issues that limit fine-grained learning of search behaviors. First, **query-quality indistinguishability**: as illustrated in Figure 1, a precise query and a vague one that happen to produce the same final answer receive identical rewards, so the model cannot learn which formulation is more effective, a problem particularly severe in multi-hop scenarios where an early suboptimal query derails the entire chain. Second, **signal vanishing in the all-failure scenario**: when all trajectories fail, the group-normalized advantages collapse to zero, yielding nearly no gradient, precisely during early training and on hard questions when supervision is most needed.

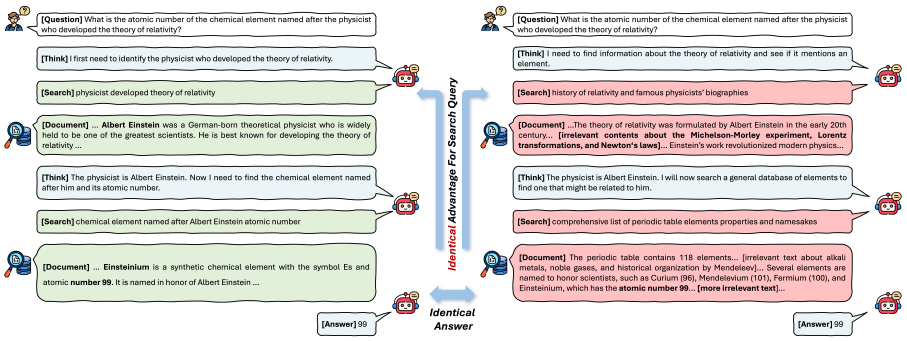

Figure 1: Trajectory-level rewards obscure step-level query quality. The left trajectory issues a focused query whose retrieved documents directly support the required answer. The right trajectory issues an under-specified query whose retrieved documents contain the answer string only incidentally and include substantial off-topic evidence. Although both trajectories may produce the same final answer and thus receive the same trajectory-level reward, their search steps differ in how reliably they expose answer-bearing evidence for downstream reasoning.

To address these limitations, we propose **IG-Search**, an RL framework that scores each search step using a pointwise Information Gain (IG) estimate and uses this score to modulate query-token advantages. The core idea is simple: with a fixed retriever, the quality of a search query is reflected in the evidence it retrieves, so the informational value of the retrieved documents provides a retriever-consistent proxy for query quality. Concretely, for each search step, IG quantifies how much the actual retrieved documents improve the model's confidence in the gold answer relative to a counterfactual baseline of random documents. This signal is then propagated back to the corresponding query tokens through per-token advantage modulation within GRPO, so that different search steps in the same trajectory receive gradients proportional to their informational contribution, rather than sharing identical advantage. Crucially, since IG is computed from the gold answer rather than the trajectory's predicted answer, the step-level signal remains informative even when all trajectories produce incorrect final answers. We further develop stabilization mechanisms to ensure robust training, detailed in §2.4.

Prior works have also explored step-level supervision but with different trade-offs. IGPO (Wang et al., 2026) computes a turn-level temporal difference on the gold-answer log-probability, which conflates reasoning, querying and retrieval within each turn rather than isolating query quality. StepSearch (Zheng et al., 2025) uses GPT-4o-generated sub-question annotations, and GiGPO (Feng et al., 2025) relies on environment-state recurrence across trajectories, the signal density of which depends on how often rollouts revisit the same state. IG-Search requires neither external annotations nor state recurrence, and unlike turn-level temporal differences, isolates retrieval quality at each search step only using the policy's own log-probabilities and the standard question–answer pair; we defer a detailed comparison to Appendix A.

We evaluate IG-Search on seven QA benchmarks spanning single-hop (NQ, TriviaQA, PopQA) and multi-hop (HotpotQA, 2WikiMultihopQA, Musique, Bamboogle) settings. With Qwen2.5-3B as

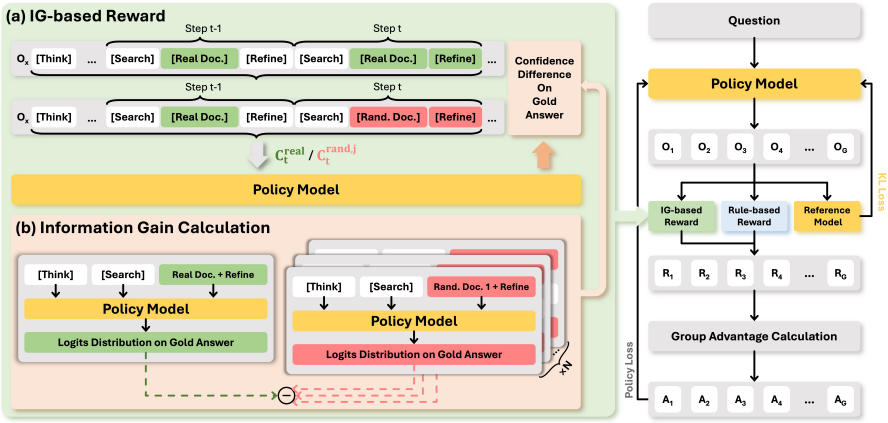

Figure 2: Overview of IG-Search. **Right:** The policy model generates G trajectories per question, each scored by rule-based rewards and step-level IG rewards to compute group advantages. **Left:** IG computation for search step t: the log-probability of the gold answer under the real retrieval context minus the average under N randomly sampled retrieval contexts.

the base model, IG-Search achieves an average Exact Match of 0.430, outperforming the strongest trajectory-level baseline (MR-Search) by 1.6 points and the step-level method GiGPO by 0.9 points on average across benchmarks, with particularly pronounced gains on multi-hop reasoning tasks. Experiments with Qwen2.5-7B further confirm the generalizability of our method, yielding consistent improvements across all benchmarks.

Our contributions are summarized as follows:

- We propose IG-Search, which introduces a step-level IG reward that measures the informational contribution of each search step. IG-Search can be viewed as a *retrieval-grounded process reward*: it derives signal from the policy's own generation probabilities conditioned on the gold answer, requiring neither a separately trained reward model nor intermediate annotations beyond standard question-answer pairs.
- We develop per-token advantage modulation within GRPO so that different search steps receive gradient signals proportional to their informational contribution, together with stabilization mechanisms that prevent degenerate behaviors. Notably, the resulting signal remains informative even when all sampled trajectories answer incorrectly.
- Extensive experiments on diverse QA benchmarks demonstrate that IG-Search achieves the strongest results among open-source models of comparable size at both the 3B and 7B scales, and comprehensive ablations validate the necessity of each proposed component.

2 METHOD

We introduce IG-Search, a reinforcement learning framework that provides step-level IG rewards for retrieval-augmented reasoning. We first review the necessary background (§2.1), then present the IG formulation (§2.2), describe how IG modulates per-token advantages in GRPO (§2.3), and detail the stabilization mechanisms that ensure robust training (§2.4). The overall framework is illustrated in Figure 2, and the training pipeline is summarized in Appendix B.

2.1 PRELIMINARIES

Trajectory Generation. Following recent work on retrieval-augmented reasoning (Jin et al., 2025; Shi et al., 2025), policy model π_θ generates reasoning trajectories by interacting with an external retriever $\mathcal{E}$. Each trajectory $o = (\tau_1, \tau_2, \ldots, \tau_T, a)$ consists of T search rounds $\tau_1, \ldots, \tau_T$ followed by a final answer block a enclosed in `<answer>`. Each round τ_t is a sequence of structured blocks: `<think>` for reasoning, `<search>` for the query issued to the search engine, `<documents>` for the content returned by the search engine, and `<refine>` for distilling (by the policy) key

evidence from the retrieved documents into a concise summary. The number of search rounds T is not predefined but determined by the policy, subject to a maximum $T_{\max}$. For each search round t, we denote the generated query as $\mathbf{q}_t$, the retrieved documents as $\mathbf{d}_t = \mathcal{E}(\mathbf{q}_t)$, and the refinement output as $\mathbf{r}_t$.

GRPO. We adopt GRPO (Shao et al., 2024) as the policy optimization algorithm. For each question q, a group of G trajectories $\{o_1, \ldots, o_G\}$ is sampled from the current policy $\pi_{\theta_{\text{old}}}$. Each trajectory o_i receives a scalar reward R_i, and the advantage is computed via group-level normalization:

$$\hat{A}_i = \frac{R_i - \text{mean}(\{R_j\}_{j=1}^{G})}{\text{std}(\{R_j\}_{j=1}^{G})}. \tag{1}$$

In GRPO, this advantage $\hat{A}_i$ is shared across all tokens in trajectory o_i.

Baseline Rewards. Following AutoRefine (Shi et al., 2025), we inherit a trajectory-level reward R_i consisting of two components: (1) r_{F1}, F1 score between the predicted and gold answers; and (2) r_{ret}, a retrieval-specific reward that evaluates whether the `<refine>` blocks contain the gold answer. IG-Search retains both reward components and introduces the step-level IG signal as an additional per-token advantage modulation, as detailed in §2.3.

2.2 Step-Level Information Gain

The central contribution of IG-Search is a step-level reward that directly measures the informational value of each search step. We define IG as the difference in the model's confidence in generating the gold answer when conditioned on the real retrieved documents relative to random documents.

IG Definition. For search step t in a trajectory, let $\mathcal{C}_t^{\text{real}}$ denote the full context up to and including step t, with all preceding steps retaining their original retrieved documents and refinements, i.e., $\mathcal{C}_t^{\text{real}} = (\tau_{<t},\, \mathbf{q}_t,\, \mathbf{d}_t,\, \mathbf{r}_t)$, where $\tau_{<t}$ denotes all preceding steps in their entirety, and $\mathbf{q}_t$, $\mathbf{d}_t$, $\mathbf{r}_t$ are the query, retrieved documents, and refinement at the current step t. We construct N counterfactual contexts $\{\mathcal{C}_t^{\text{rand},j}\}_{j=1}^{N}$ by replacing only the (documents, refinement) pair at step t with pairs drawn from randomly selected search steps of other questions in the same training batch, which ensures a consistent document-length distribution without additional data loading overhead. The IG for step t is then defined as:

$$\text{IG}_t = \log \pi_{\theta_{\text{old}}}(a^* \mid \mathcal{C}_t^{\text{real}}) - \frac{1}{N}\sum_{j=1}^{N} \log \pi_{\theta_{\text{old}}}(a^* \mid \mathcal{C}_t^{\text{rand},j}), \tag{2}$$

where a^* denotes gold answer, and $\pi_{\theta_{\text{old}}}$ is the rollout policy used to sample the trajectories. The real-context term $\log \pi_{\theta_{\text{old}}}(a^* \mid \mathcal{C}_t^{\text{real}})$ and the N counterfactual terms can be computed efficiently by stacking the real context and the N random contexts into a single batch and obtaining the log-probabilities of the gold answer for all $N{+}1$ variants in a single forward pass. Crucially, this batched pass scores only the short gold-answer tokens (typically a few subwords) rather than generating new sequences; the resulting overhead is detailed in Appendix I. When multiple valid answer aliases exist, we compute the length-normalized log-probability for each alias and take the maximum. We provide a concrete example of the IG computation on a two-hop question in Appendix C, showing how IG captures the marginal informational contribution of each successive search step. In this section, $\log \pi_{\theta_{\text{old}}}(a^* \mid \mathcal{C})$ denotes the length-normalized log-probability of the gold answer, computed as:

$$\log \pi_{\theta_{\text{old}}}(a^* \mid \mathcal{C}) = \frac{1}{|a^*|}\sum_{k=1}^{|a^*|} \log \pi_{\theta_{\text{old}}}(a_k^* \mid \mathcal{C}, a_{<k}^*). \tag{3}$$

Because Eq. 3 length-normalizes by $|a^*|$, all subsequent IG values (including the thresholds δ and η) are reported in *nats per answer token*, making them comparable across answers of different length.

Why against Random Documents? A natural alternative is to compare against an empty context. However, real and empty contexts differ not only in informativeness but also in length and positional structure, introducing a systematic bias unrelated to retrieval quality. Drawing counterfactual documents from other questions matches the length and structural characteristics of the real context while removing its topical relevance, so that IG_t isolates the informational contribution of retrieval. We compare four counterfactual designs in Appendix D; the random-document baseline outperforms all alternatives, with the empty baseline showing the largest degradation on multi-hop tasks.

2.3 ADVANTAGE MODULATION WITH IG

In GRPO, all tokens in a trajectory share identical $\hat{A}_i$ (Eq. 1). IG-Search breaks this uniformity by modulating the advantage of query tokens based on the IG of the corresponding search step. Concretely, let $\mathcal{Q}_t$ denote the set of token positions that belong to the search query at step t. The modulated advantage for each token at position p in trajectory o_i is:

$$\tilde{A}_{i,p} = \begin{cases} \hat{A}_i + \alpha \cdot \dfrac{\widetilde{\mathrm{IG}}_t}{|\mathcal{Q}_t|}, & \text{if } p \in \mathcal{Q}_t, \\ \hat{A}_i, & \text{otherwise}, \end{cases} \tag{4}$$

where $\widetilde{\mathrm{IG}}_t$ is the processed IG value (detailed in §2.4), $|\mathcal{Q}_t|$ denotes the number of tokens in the search query, and α is a weighting coefficient.

The division by $|\mathcal{Q}_t|$ ensures that a search step's total contribution to the trajectory loss equals $\alpha \cdot \widetilde{\mathrm{IG}}_t$ and is therefore decoupled from the query length, which is essential to prevent the reward-hacking behavior discussed in §2.4.

We modulate only the query tokens instead of all tokens in the trajectory because IG directly reflects the quality of the documents retrieved by the fixed retriever, which is determined by the search query. This formulation has two key properties. First, in a single trajectory, different search steps receive different gradient signals proportional to their informational contribution, enabling the model to learn which query formulations are more effective. Second, even when all trajectories answer incorrectly, the IG term $\alpha \cdot \widetilde{\mathrm{IG}}_t/|\mathcal{Q}_t|$ still ensures nonzero gradients to query tokens, allowing the model to learn from the relative quality of different search steps across otherwise indistinguishable trajectories.

2.4 STABILIZATION MECHANISMS

As observed in prior work on policy optimization and language model alignment (Schulman et al., 2017; Rafailov et al., 2023; Jiang et al., 2020; Wang et al., 2025), metrics derived from the difference or ratio of raw LLM log-probabilities inherently exhibit high variance and heavy-tailed distributions. If used directly, these extreme values and estimation noise can destabilize training or induce degenerate behaviors, similar to the unbounded implicit rewards in preference learning. To ensure robust optimization, we process the raw IG_t into $\widetilde{\mathrm{IG}}_t$ through a pipeline of stabilization mechanisms.

Dead Zone Filtering. For questions the model can answer from parametric knowledge, IG does not faithfully reflect query quality: relevant documents add little new information and yield near-zero IG, while real documents can even introduce competing facts that slightly dilute the parametric prior, producing small negative IG for well-formed queries (see Case 3 in Appendix F). In both cases the magnitude of IG is small and driven by the model's parametric knowledge rather than by query effectiveness. We therefore apply a dead zone that filters out all small-magnitude IG values: if $|\mathrm{IG}_t| < \delta$, we set $\mathrm{IG}_t = 0$. We use $\delta = 0.5$ by default.

Asymmetric Negative Scaling. A negative IG indicates that the actual retrieved documents are less helpful than random ones, suggesting a poor search query. However, we observe that applying full-strength penalties for negative IG can cause the model to collapse to a degenerate strategy of never issuing search queries, thereby avoiding all negative signals. To prevent this, we scale negative IG values by a factor $\lambda < 1$:

$$\mathrm{IG}_t \leftarrow \begin{cases} \mathrm{IG}_t, & \text{if } \mathrm{IG}_t \geq 0, \\ \lambda \cdot \mathrm{IG}_t, & \text{if } \mathrm{IG}_t < 0. \end{cases} \tag{5}$$

By default, we set $\lambda = 0.1$. This asymmetry ensures that the optimal strategy is to search *and search well*, rather than to avoid searching entirely.

Soft Clipping. Extreme IG values can dominate the reward signal and cause gradient instability. We apply a logarithmic soft clipping for values exceeding a threshold η:

$$\mathrm{IG}_t \leftarrow \begin{cases} \mathrm{IG}_t, & \text{if } |\mathrm{IG}_t| \leq \eta, \\ \mathrm{sign}(\mathrm{IG}_t) \cdot \big(\eta + \log(1 + |\mathrm{IG}_t| - \eta)\big), & \text{otherwise}. \end{cases} \tag{6}$$

This preserves the ordering of extreme values while substantially reducing their magnitude. We use $\eta = 3.0$ by default.

Query Length Normalization. As defined in Eq. 4, the IG bonus is divided by the query length $|\mathcal{Q}_t|$. Without this normalization, the model can exploit a reward hacking strategy by generating unnecessarily long queries: since each query token receives the IG bonus, longer queries accumulate larger total bonuses when the IG is positive.

The three mechanisms are applied in the order dead zone $\rightarrow$ asymmetric scaling $\rightarrow$ soft clipping, so that δ is calibrated on the raw IG distribution and clipping only acts on the genuinely large positive tail; we visualize the raw IG distribution and justify this ordering in Appendix E.

3 EXPERIMENTS

In this section, we aim to answer the following research questions: **RQ1**: Does IG-Search improve QA accuracy compared to existing retrieval-augmented reasoning methods? **RQ2**: How essential is each component of the IG mechanism? **RQ3**: Does IG-Search scale to larger models? **RQ4**: What search behaviors does IG-Search learn during training, and does it provide meaningful learning signal even when all sampled trajectories fail?

3.1 EXPERIMENTAL SETUP

Datasets. We evaluate on seven diverse QA benchmarks: three single-hop datasets, Natural Questions (NQ) (Kwiatkowski et al., 2019), TriviaQA (Joshi et al., 2017), and PopQA (Mallen et al., 2023), along with four multi-hop datasets, HotpotQA (Yang et al., 2018), 2WikiMultihopQA (2Wiki) (Ho et al., 2020), Musique (Trivedi et al., 2022), and Bamboogle (Press et al., 2023). Following prior work (Jin et al., 2025; Shi et al., 2025), we train on a combined training set from NQ and HotpotQA and evaluate on all benchmarks. Exact Match (EM) accuracy serves as the evaluation metric. We normalize both predicted and ground-truth answers by lowercasing, removing punctuation, and stripping articles before comparison, and treat the prediction as correct if it exactly matches any of the ground-truth aliases. Full dataset statistics and split information are provided in Appendix G. All models are trained on 8$\times$H800 GPUs using veRL (Sheng et al., 2025) for 200 steps; the complete configuration is listed in Appendix H.

Baselines. We compare against methods in three categories: (1) *without retrieval*: direct generation, SFT, and R1-style RL (Guo et al., 2025); (2) *single-hop retrieval*: Naive RAG (Lewis et al., 2021); (3) *multi-hop retrieval*: Search-o1 (Li et al., 2025), IRCoT (Trivedi et al., 2023), Search-R1 (Jin et al., 2025), ReSearch (Chen et al., 2025), StepSearch (Zheng et al., 2025), AutoRefine (Shi et al., 2025), MR-Search (Xiao et al., 2026), and GiGPO (Feng et al., 2025). All methods share the same retrieval corpus (December 2018 Wikipedia), retriever (E5-base-v2), and top-3 retrieval depth.[1] Baseline numbers are taken from the original papers when the evaluation protocol matches.

3.2 OVERALL PERFORMANCE (RQ1)

Table 1 presents the performance comparison on Qwen2.5-3B. Both IG-Search variants achieve an average EM of 0.430, the highest among all compared methods. Against the strongest trajectory-level baseline MR-Search (0.414, also Base), IG-Search-Base improves by 1.6 points, with pronounced gains on multi-hop benchmarks (HotpotQA $+1.7$, 2Wiki $+1.4$, Musique $+1.4$), and also surpasses AutoRefine-Base (0.405) by 2.5 points on average. Notably, IG-Search-Base issues 35% fewer search calls than AutoRefine-Base on single-hop benchmarks while achieving higher EM on all three (NQ $+0.9$, TriviaQA $+1.7$, PopQA $+2.6$), indicating that the step-level signal teaches the policy *when* to search rather than simply encouraging more retrieval; a full per-position decomposition is provided in Appendix P. Appendix Q further presents a Bamboogle case where IG-Search decomposes a compound question into two focused queries and retrieves the answer directly, while AutoRefine's under-specified queries retrieve only superficially related evidence under the same retriever. Despite introducing a dense step-level signal, IG-Search adds only $\sim$6.7% to per-step training wall-clock time relative to AutoRefine (Appendix I) and leaves inference latency unchanged.

[1] StepSearch is trained with PPO on MuSiQue using GPT-4o-augmented sub-question decompositions rather than GRPO on NQ+HotpotQA, and reports only on multi-hop benchmarks.

Table 1: Exact Match accuracy of IG-Search versus baseline methods on seven QA benchmarks with Qwen2.5-3B. **Bold** denotes the best and underline the second best result per column.

Methods	Single-Hop QA			Multi-Hop QA				
	NQ	TriviaQA	PopQA	HotpotQA	2Wiki	Musique	Bamboogle	Avg.
w/o Retrieval								
Direct Generation	0.106	0.288	0.108	0.149	0.244	0.020	0.024	0.134
SFT	0.249	0.292	0.104	0.186	0.248	0.044	0.112	0.176
R1-Instruct	0.210	0.449	0.171	0.208	0.275	0.060	0.192	0.224
R1-Base	0.226	0.455	0.173	0.201	0.268	0.055	0.224	0.229
w/ Single-Hop Retrieval								
Naive RAG	0.348	0.544	0.387	0.255	0.226	0.047	0.080	0.270
w/ Multi-Hop Retrieval								
IRCoT	0.111	0.312	0.200	0.164	0.171	0.067	0.240	0.181
Search-o1	0.238	0.472	0.262	0.221	0.218	0.054	0.320	0.255
Search-R1-Base	0.421	0.583	0.413	0.297	0.274	0.066	0.128	0.312
Search-R1-Instruct	0.397	0.565	0.391	0.331	0.310	0.124	0.232	0.336
ReSearch-Base	0.427	0.597	0.430	0.305	0.272	0.074	0.128	0.319
ReSearch-Instruct	0.365	0.571	0.395	0.351	0.272	0.095	0.266	0.331
StepSearch-Base	–	–	–	0.329	0.339	0.181	0.328	–
StepSearch-Instruct	–	–	–	0.345	0.320	0.174	0.344	–
AutoRefine-Base	0.467	0.620	0.450	0.405	0.393	0.157	0.344	0.405
AutoRefine-Instruct	0.436	0.597	0.447	0.404	0.380	0.169	0.336	0.396
MR-Search (Base)	**0.477**	0.635	0.460	0.419	0.401	0.165	0.344	0.414
GiGPO (Instruct)	0.420	0.595	0.424	0.369	0.370	0.126	**0.641**	0.421
IG-Search-Base	0.476	**0.637**	**0.476**	**0.436**	0.415	0.179	0.390	**0.430**
IG-Search-Instruct	0.450	0.614	0.460	0.431	**0.440**	**0.191**	0.424	**0.430**

As GiGPO is trained solely on Instruct variant, we compare it with IG-Search-Instruct, which improves over GiGPO by 0.9 points on the seven-benchmark average, with particularly pronounced gains on HotpotQA (+6.2), 2Wiki (+7.0), and Musique (+6.5). StepSearch achieves competitive multi-hop results (e.g., Musique: 0.181 for the base variant), but does so by relying on GPT-4o-generated sub-question decompositions with reference documents and search keywords. Using only standard question-answer pairs, IG-Search matches or exceeds StepSearch on every overlapping benchmark (HotpotQA 0.436 vs. 0.329, 2Wiki 0.415 vs. 0.339, Musique 0.179 vs. 0.181, Bamboogle 0.390 vs. 0.328), showing that the policy's own generation confidence is a sufficient and more practical proxy for step-level supervision.

IG-Search-Instruct achieves the same average EM of 0.430, outperforming AutoRefine-Instruct (0.396) by 3.4 points. The two IG-Search variants exhibit complementary strengths: the base variant performs better on single-hop benchmarks (e.g., NQ: 0.476 vs. 0.450), while the instruct variant excels on harder multi-hop benchmarks (2Wiki: 0.440 vs. 0.415; Musique: 0.191 vs. 0.179; Bamboogle: 0.424 vs. 0.390). This suggests that the instruction-following prior of the instruct model benefits complex multi-hop reasoning, while the base model may benefit from fewer inductive biases during RL exploration. We verify the statistical significance of these improvements in Appendix J.

3.3 Ablation Studies

3.3.1 Component Ablation (RQ2)

We systematically remove one component at a time from IG-Search framework. All ablations use Qwen2.5-3B-Base. Results are presented in Table 2.

Removing the **IG reward** entirely reduces the framework to a trajectory-level baseline, decreasing the average EM from 0.430 to 0.403. [2] The degradation is most severe on multi-hop benchmarks (Bamboogle: -4.4, HotpotQA: -3.5), confirming that step-level reward is especially effective when multiple precise search steps are needed. Replacing **asymmetric scaling** with symmetric treatment ($\lambda = 1.0$) causes the largest single-component degradation, reducing average EM to 0.391: surviving negatives retain their full magnitude ($\sim[-3, -0.5]$ nats after dead-zone filtering), so occasional large outliers from early poorly formed queries dominate minibatches and drive the policy toward search avoidance (confirmed by the monotonic drop in search frequency in Appendix M). Disabling **dead**

[2]This single-seed ablation is functionally AutoRefine-Base; the 0.2-point gap from Table 1 is within seed variance (Appendix J).

zone filtering drops average EM to 0.416: on questions the model can already answer, IG does not faithfully reflect retrieval quality (yielding small positive, near-zero, or even misleading negative penalties; see Case 3 in Appendix F) and feeding these into the advantage interferes with learning on genuinely informative search steps. Disabling **soft clipping** yields 0.420 average EM. Although extreme IG values are infrequent, they can dominate the advantage signal in individual minibatches and destabilize gradient updates. Without **query length normalization**, the average EM drops to 0.408. The model exploits this by generating abnormally long search queries to accumulate larger total IG bonuses, a form of reward hacking that degrades retrieval effectiveness.

We probe modulation scope in both directions. Restricting to only the first query yields 0.417, with consistent multi-hop degradation (HotpotQA −1.8, 2Wiki −1.5, Bamboogle −2.5), showing that later-step IG is necessary (see Appendix O). Expanding the scope also degrades performance: think+query yields 0.422 (long think segments dilute the signal), and all-token drops to 0.413 since refine/answer tokens are already supervised by the trajectory reward; a full hyperparam sensitivity analysis is provided in Appendix N.

Table 2: Component ablation on Qwen2.5-3B-Base (Exact Match). The top block removes or modifies individual stabilization components; the bottom block varies the scope of per-token advantage modulation. The first row is the full IG-Search and serves as the reference.

Configuration	Single-Hop QA			Multi-Hop QA				Avg.
	NQ	TriviaQA	PopQA	HotpotQA	2Wiki	Musique	Bamboogle	
IG-Search (full)	**0.476**	**0.637**	**0.476**	**0.436**	**0.415**	**0.179**	**0.390**	**0.430**
w/o IG reward	0.458	0.617	0.452	0.401	0.395	0.154	0.346	0.403
w/o asymmetric scaling	0.448	0.605	0.445	0.398	0.378	0.148	0.312	0.391
w/o dead zone	0.465	0.625	0.462	0.422	0.402	0.170	0.368	0.416
w/o soft clipping	0.468	0.629	0.465	0.426	0.407	0.172	0.372	0.420
w/o query length norm	0.460	0.619	0.455	0.415	0.397	0.161	0.352	0.408
w/ first-query-only modulation	0.471	0.630	0.470	0.418	0.400	0.164	0.365	0.417
w/ think+query modulation	0.470	0.631	0.469	0.428	0.408	0.173	0.378	0.422
w/ all-token modulation	0.463	0.624	0.460	0.419	0.398	0.165	0.360	0.413

Table 3: Exact Match accuracy at the 7B scale (Qwen2.5-7B). **Bold** denotes the best result.

Method	Single-Hop QA			Multi-Hop QA				Avg.
	NQ	TriviaQA	PopQA	HotpotQA	2Wiki	Musique	Bamboogle	
Search-R1-Base	0.469	0.627	0.449	0.410	0.272	0.173	0.456	0.408
AutoRefine-Base	0.484	0.659	0.487	0.451	0.405	0.187	0.512	0.455
MR-Search (Base)	**0.502**	0.666	0.472	0.468	0.436	**0.221**	0.452	0.460
GiGPO (Instruct)	0.464	0.647	0.461	0.416	0.436	0.189	**0.689**	0.472
IG-Search-Base	0.501	**0.675**	**0.504**	0.473	0.430	0.205	0.544	0.476
IG-Search-Instruct	0.490	0.665	0.495	**0.480**	**0.445**	0.215	0.560	**0.479**

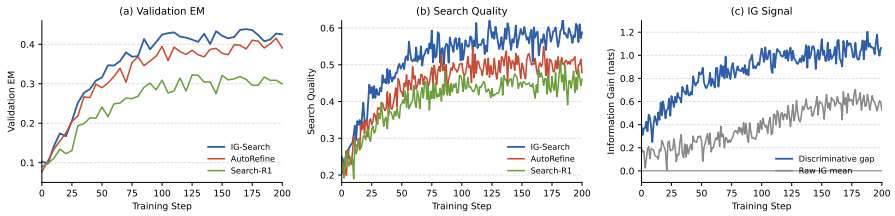

Figure 3: Training dynamics on Qwen2.5-3B-Base. (a) Validation EM averaged over seven benchmarks. (b) Search quality, measured as the fraction of retrieval calls whose returned documents contain the gold answer. (c) Raw per-step IG averaged over all search calls in the batch, and the *discriminative gap* (mean IG on correct minus incorrect trajectories), which stays strictly positive throughout training. A per-position decomposition appears in Appendix O.

3.3.2 MODEL SIZE (RQ3)

At the 7B scale (Table 3), IG-Search-Instruct reaches the highest average EM of 0.479, surpassing MR-Search (0.460) by 1.9 points and AutoRefine-Base (0.455) by 2.4 points; IG-Search-Base reaches 0.476, also surpassing both. GiGPO-7B (0.472) leads only on Bamboogle (0.689 vs. 0.560; 125 questions), with IG-Search-Instruct winning the other six. Appendix K extends this analysis to four model sizes, confirming consistent scaling of IG-Search across sizes.

3.4 ANALYSIS (RQ4)

3.4.1 TRAINING DYNAMICS

Figure 3 tracks training on Qwen2.5-3B-Base against AutoRefine and Search-R1 under identical configurations. Panel (a): IG-Search surpasses both baselines after roughly 40 steps and maintains a widening gap. Panel (b): IG-Search consistently yields higher search quality (fraction of retrievals whose documents contain the gold answer), confirming that step-level supervision directly improves query formulation. Panel (c) shows two complementary signals: the raw per-step IG averaged over the batch grows steadily over training, and the *discriminative gap*, defined as mean IG on eventually-correct trajectories minus that on incorrect ones, stays strictly positive throughout, showing that IG reliably separates effective queries from ineffective ones. A per-position decomposition further shows $\mathrm{IG}^{(0)} > \mathrm{IG}^{(1)} > \mathrm{IG}^{(2)}$ from the first iteration onward, indicating that IG reflects the diminishing-marginal-value structure of multi-hop retrieval rather than a byproduct of optimization (Appendix O).

3.4.2 LEARNING SIGNAL IN THE ALL-FAILURE SCENARIO

A structural advantage of IG-Search is its ability to provide training signal even when every sampled trajectory in a group produces an incorrect answer, a scenario common in early training and on harder multi-hop questions. When all rollouts in a group share the same outcome reward, the group-normalized advantages $\hat{A}_i$ collapse to zero (or to implementation-defined small values), causing trajectory-level rewards to fail to provide meaningful learning signals. IG-Search bypasses this failure mode because the per-token IG modulation $\alpha \cdot \widetilde{\mathrm{IG}}_t/|\mathcal{Q}_t|$ is added independently of answer correctness: tokens from an informative search query receive positive modulation, while those from an uninformative query receive either zero (filtered by the dead zone) or a small negative signal. This ensures that the model consistently assigns higher credit to the more effective query.

Appendix R provides a concrete case from an all-failure batch in which two rollouts end in different wrong answers through searches of very different quality. Quantitatively, all-failure questions (every rollout yields $r_{\mathrm{F1}}{=}0$) account for $\sim$58% of the training batch at step 1 and $\sim$10% at step 200; on this subset the mean absolute IG modulation $\alpha \cdot |\widetilde{\mathrm{IG}}_t|$ remains around 0.21 throughout training, providing a consistent per-token gradient where $\hat{A}_i{\equiv}0$. Zeroing out the IG modulation on all-failure questions (while keeping it elsewhere on non-all-failure questions) drops average EM to 0.422 (-0.8 points), with losses concentrated on multi-hop benchmarks (HotpotQA -1.1, 2Wiki -1.4, Musique -0.7), confirming that the step-level signal on reward-indistinguishable rollouts contributes non-trivially.

4 CONCLUSION

We presented **IG-Search**, a reinforcement learning framework that assigns each search step a distinct reward measuring how much its retrieved documents improve the policy's confidence in the gold answer relative to a random counterfactual. This turns a coarse, trajectory-level signal into a dense, query-grounded one, enabling per-token credit assignment across search steps. Experiments on 7 QA benchmarks show that IG-Search achieves the strongest results among open-source models of comparable size at both 3B and 7B scales, with larger gains on multi-hop tasks where trajectory-level credit assignment is most brittle.

AI USE STATEMENT

No AI tools were used at any stage of this work, including research design, data collection and analysis, code development, experimentation, or manuscript drafting, revision, and language polishing.

REFERENCES

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-rag: Learning to retrieve, generate, and critique through self-reflection. *arXiv preprint arXiv:2310.11511*, 2023.

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

Zheng Ding and Weirui Ye. Treegrpo: Tree-advantage grpo for online rl post-training of diffusion models. *arXiv preprint arXiv:2512.08153*, 2025.

Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. Group-in-group policy optimization for LLM agent training. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Daya Guo, Qihao Zhu, Dejian Yang, Zhenda Xie, Kai Dong, Wentao Zhang, Guanting Chen, Xiao Bi, Y. Wu, Y. K. Li, Fuli Luo, Yingfei Xiong, and Wenfeng Liang. Deepseek-coder: When the large language model meets programming – the rise of code intelligence. *arXiv preprint arXiv:2401.14196*, 2024.

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

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.

Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. How can we know what language models know? *Transactions of the Association for Computational Linguistics*, 8:423–438, 2020.

Zhengbao Jiang, Frank Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. Active retrieval augmented generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 7969–7992, Singapore, 2023.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan O Arik, Dong Wang, Hamed Zamani, and Jiawei Han. Search-r1: Training LLMs to reason and leverage search engines with reinforcement learning. In *Second Conference on Language Modeling*, 2025.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1601–1611, Vancouver, Canada, 2017.

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. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466, 2019.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks. *arXiv preprint arXiv:2005.11401*, 2021.

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. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 5420–5438, Suzhou, China, 2025.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. *arXiv preprint arXiv:2305.20050*, 2023.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9802–9822, Toronto, Canada, 2023.

OpenAI. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2024a.

OpenAI. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024b.

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

Qwen. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2025.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems*, pp. 53728–53741, 2023.

Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. In-context retrieval-augmented language models. *arXiv preprint arXiv:2302.00083*, 2023.

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

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

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 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, 2025.

Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Richard James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. REPLUG: Retrieval-augmented black-box language models. In Kevin Duh, Helena Gomez, and Steven Bethard (eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 8371–8384, Mexico City, Mexico, 2024.

Yaorui Shi, Sihang Li, Chang Wu, Zhiyuan Liu, Junfeng Fang, Hengxing Cai, An Zhang, and Xiang Wang. Search and refine during think: Facilitating knowledge refinement for improved retrieval-augmented reasoning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

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*, 2025.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

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.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10014–10037, Toronto, Canada, 2023.

Guoqing Wang, Sunhao Dai, Guangze Ye, Zeyu Gan, Wei Yao, Yong Deng, Xiaofeng Wu, and Zhenzhe Ying. Information gain-based policy optimization: A simple and effective approach for multi-turn search agents. In *The Fourteenth International Conference on Learning Representations*, 2026.

Zihan Wang, Zihan Liang, Zhou Shao, Yufei Ma, Huangyu Dai, Ben Chen, Lingtao Mao, Chenyi Lei, Yuqing Ding, and Han Li. InfoGain-RAG: Boosting retrieval-augmented generation through document information gain-based reranking and filtering. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 7190–7204, Suzhou, China, 2025.

Tongyu Wen, Guanting Dong, and Zhicheng Dou. Smartsearch: Process reward-guided query refinement for search agents. *arXiv preprint arXiv:2601.04888*, 2026.

Teng Xiao, Yige Yuan, Hamish Ivison, Huaisheng Zhu, Faeze Brahman, Nathan Lambert, Pradeep Dasigi, Noah A. Smith, and Hannaneh Hajishirzi. Meta-reinforcement learning with self-reflection for agentic search. *arXiv preprint arXiv:2603.11327*, 2026.

Jun Xu, Xinkai Du, Yu Ao, Peilong Zhao, Yang Li, Ling Zhong, Lin Yuan, Zhongpu Bo, Xiaorui Wang, Mengshu Sun, Zhengke Gui, Dalong Zhang, Zhaoyang Wang, Qiwei Wang, Yangyang Hou, Zhiying Yin, Haofen Wang, Huajun Chen, Lei Liang, and Jun Zhou. Thinker: Training llms in hierarchical thinking for deep search via multi-turn interaction. *arXiv preprint arXiv:2511.07943*, 2025.

Shi-Qi Yan, Jia-Chen Gu, Yun Zhu, and Zhen-Hua Ling. Corrective retrieval augmented generation. *arXiv preprint arXiv:2401.15884*, 2024.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2369–2380, Brussels, Belgium, 2018.

Zhenru Zhang, Chujie Zheng, Yangzhen Wu, Beichen Zhang, Runji Lin, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. The lessons of developing process reward models in mathematical reasoning. *arXiv preprint arXiv:2501.07301*, 2025.

Xuhui Zheng, Kang An, Ziliang Wang, Yuhang Wang, and Yichao Wu. StepSearch: Igniting LLMs search ability via step-wise proximal policy optimization. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 21805–21830, Suzhou, China, 2025.

A Related Work

IG-Search sits at the intersection of RL-based reasoning, retrieval-augmented generation, and step-level reward design. We review each branch of work below, focusing on how existing methods handle credit assignment for retrieval actions.

A.1 RL for LLM Reasoning and Retrieval-Augmented Generation

Reinforcement learning has become a central tool for improving reasoning in LLMs (Qwen, 2025; Guo et al., 2025; OpenAI, 2024b). Group-relative methods such as GRPO (Shao et al., 2024; Guo et al., 2025) drive substantial gains on mathematical reasoning and code generation using only outcome-based rewards, and IG-Search builds directly on this framework. A parallel line of work equips LLMs with retrieval tools to access external knowledge during reasoning (Lewis et al., 2021; Jiang et al., 2023; Ram et al., 2023). Early methods use supervised fine-tuning to teach models when and how to search (Asai et al., 2023; Shi et al., 2024; Yan et al., 2024), while recent work adopts RL to enable a *search-during-think* paradigm in which the model autonomously invokes retrieval within its reasoning trajectories (Jin et al., 2025; Chen et al., 2025; Song et al., 2025; Li et al., 2025). AutoRefine (Shi et al., 2025) adds an explicit refinement step and a retrieval-specific reward, and MR-Search (Xiao et al., 2026) introduces cross-episode self-reflection.

A.2 Step-Level Reward Design for Retrieval-Augmented Reasoning

Recent work addresses the limitations of trajectory-level credit assignment along several complementary axes, which we organize by the source of their step-level signal.

Temporal-difference approaches. IGPO (Wang et al., 2026) defines a turn-level reward as the temporal difference in the policy's gold-answer log-probability between consecutive turns, and uses the resulting turn-level returns in place of the GRPO advantage. IG-Search shares the intuition that the rollout policy's confidence in the gold answer is a useful source of step-level supervision, but realizes it through a counterfactual comparison at each retrieval step, which isolates the contribution of the retrieved documents and allows the signal to be routed back specifically to the query tokens that produced them. Because the two methods evaluate under different retrieval environments and metrics, we provide a cross-protocol comparison in Appendix L.

Annotation-dependent and state-recurrence methods. StepSearch (Zheng et al., 2025) computes a TF-IDF coverage reward over golden supporting documents and additionally uses GPT-4o-generated sub-question keywords, requiring intermediate annotations that IG-Search avoids entirely. GiGPO (Feng et al., 2025) takes an annotation-free approach by retroactively grouping actions from identical environment states across trajectories and computing a relative advantage within each group. Its signal density, however, depends on how often rollouts revisit the same state. IG-Search makes the opposite trade-off: it uses the gold answer for a denser, query-grounded signal at every search step regardless of state recurrence. Combining these two is a promising direction for future work.

Tree-structured credit assignment. Tree-GRPO (Ding & Ye, 2025) replaces chain-based rollouts with agent-step-level tree search, deriving implicit step-level preference signals from the tree structure itself. Because it modifies the sampling strategy rather than the reward signal, it is orthogonal to IG-Search and the two could in principle be combined.

Teacher-dependent pipelines. Thinker (Xu et al., 2025) and SmartSearch (Wen et al., 2026) achieve strong results but rely on large teacher models (72B and 32B respectively) and multi-stage training pipelines (SFT→DPO→GRPO or distilled process reward models), whereas IG-Search operates in a single GRPO loop using only standard question-answer pairs.

Relation to process reward models. More broadly, IG-Search can be viewed as a retrieval-grounded process reward (Lightman et al., 2023; Zhang et al., 2025). Process reward models have shown substantial gains on mathematical reasoning by providing step-level supervision, but they typically require a separately trained reward model and, in some cases, human-annotated step labels. IG-Search obtains a step-level signal without any of these external components: the gold answer and

the policy's own log-probabilities are sufficient to derive a reward whose granularity matches that of learned process rewards, with the additional advantage of being grounded in actual retrieval outcomes rather than an external judge's assessment.

B IG-SEARCH TRAINING ALGORITHM

We here provide the complete step-by-step training pipeline for IG-Search framework. Algorithm 1 details how the standard trajectory-level rewards are integrated with our proposed step-level Information Gain computation and advantage modulation within the GRPO loop.

Algorithm 1 IG-Search Training

Require: Training set $\mathcal{D}$, policy π_θ, search engine $\mathcal{E}$, group size G, IG weight α

for each training iteration **do**
  for each question (q, a^*) in minibatch **do**
    Sample G trajectories $\{o_1, \ldots, o_G\}$ from $\pi_{\theta_{\text{old}}}(\cdot \mid q)$
    Compute trajectory rewards $R_i = r_{\text{F1}}(o_i) + r_{\text{ret}}(o_i)$
    Compute advantages $\hat{A}_i$ via Eq. 1
    for each trajectory o_i **do**
      for each search step t in o_i **do**
        Construct $\mathcal{C}_t^{\text{real}}$ and $\{\mathcal{C}_t^{\text{rand},j}\}_{j=1}^{N}$
        Compute IG_t via Eq. 2
        Apply stabilization: dead zone, asymmetric scaling, soft clipping $\rightarrow \widetilde{\text{IG}}_t$
      end for
      Modulate token-level advantages via Eq. 4
    end for
  end for
  Update π_θ using GRPO with modulated advantages $\{\tilde{A}_{i,p}\}$
end for

C Example of IG Computation on a Multi-Hop Question

As described in §2.2, the step-level IG is defined as the difference in the model's confidence in generating the gold answer when conditioned on the real retrieved context relative to a counterfactual baseline of random documents. To concretely illustrate this process, we provide a detailed illustration of a two-hop reasoning trajectory in Figure 4.

This example demonstrates how the counterfactual contexts are constructed at each step while preserving the history of previous steps. Furthermore, it illustrates how the resulting IG values naturally capture the marginal informational contribution of successive retrieval calls: the first search introduces the novel gold answer and receives a high reward, whereas the second search provides only additional evidence, resulting in a positive but much smaller marginal gain.

Question: Who is the father of the father of modern experimental psychology?
Gold Answer: **Maximilian Wundt**

Step 1

<think> I need to find who is considered the father of modern experimental psychology. </think>
<search> father of modern experimental psychology </search>
<documents> ... Wilhelm Wundt is widely regarded as the father of experimental psychology. He was born on 16 August 1832 to parents **Maximilian Wundt**. ... </documents>
<refine> Wilhelm Wundt is the father of experimental psychology. </refine>

IG computation: $\mathcal{C}_1^{\text{real}}$ contains the actual documents about Wilhelm Wundt. Three counterfactual contexts $\mathcal{C}_1^{\text{rand},j}$ replace these with random (documents, refinement) pairs from other questions in the batch.
$\log \pi_\theta(\text{Maximilian Wundt} \mid \mathcal{C}_1^{\text{real}}) = -2.07, \quad \frac{1}{N}\sum_j \log \pi_\theta(\text{Maximilian Wundt} \mid \mathcal{C}_1^{\text{rand},j}) = -3.59$
$\Rightarrow$ **IG_1 = 1.52**.
The actual documents mention Maximilian Wundt by name, substantially improving confidence over random documents from unrelated questions.

Step 2 (conditioned on Step 1 with its original documents)

<think> Now I need to find the father of Wilhelm Wundt. </think>
<search> Wilhelm Wundt father </search>
<documents> ... Wundt was born in Neckarau, Baden, the fourth child to parents **Maximilian Wundt**, a Lutheran minister. ... </documents>
<refine> Wilhelm Wundt's father was **Maximilian Wundt**. </refine>

IG computation: $\mathcal{C}_2^{\text{real}}$ includes Step 1 (with its original documents) followed by Step 2's actual documents. The counterfactual contexts $\mathcal{C}_2^{\text{rand},j}$ keep Step 1 intact but replace only Step 2's (documents, refinement) pair with random ones from other questions.
$\log \pi_\theta(\text{Maximilian Wundt} \mid \mathcal{C}_2^{\text{real}}) = -0.41, \quad \frac{1}{N}\sum_j \log \pi_\theta(\text{Maximilian Wundt} \mid \mathcal{C}_2^{\text{rand},j}) = -1.15$
$\Rightarrow$ **IG_2 = 0.74**.
Step 2 provides useful confirmation, but its marginal contribution is smaller because the gold answer already appeared in Step 1's documents.

Figure 4: Step-level IG computation on a two-hop question. Step 1 yields high IG because its documents contain the gold answer. $\mathcal{C}_2^{\text{rand}}$ keeps Step 1 completely and only replaces Step 2's (documents, refinement), so the counterfactual already carries Step 1's evidence; Step 2's IG therefore measures the *marginal* value of the second retrieval.

D COUNTERFACTUAL BASELINE DESIGN

We compare four counterfactual constructions for $\mathcal{C}_t^{\text{rand}}$: (1) *Random docs + refine* (default), replacing both documents and refinement with a pair from another question; (2) *Random docs only*, replacing documents while retaining the original refinement; (3) *Bottom-k docs*, using the least relevant retriever results for the same query; (4) *Empty*, removing documents and refinement entirely. As shown in Figure 5, the two random-document variants perform nearly identically (0.430 vs. 0.428), while bottom-k drops to 0.414 due to residual topical overlap that weakens the contrast. The empty baseline performs worst (0.403), with degradation concentrated on multi-hop benchmarks, confirming that length mismatch introduces systematic bias as discussed in §2.2.

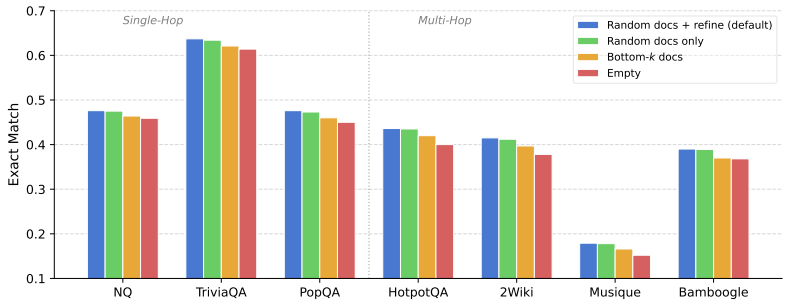

Figure 5: Comparison of counterfactual baseline designs across seven benchmarks. The default random docs + refine baseline achieves the highest performance, with the empty baseline showing disproportionately larger degradation on multi-hop benchmarks.

E RAW IG DISTRIBUTION

Figure 6 shows the histogram of raw (pre-stabilization) IG values across all search steps in a training batch at steps 50, 100, and 200 on Qwen2.5-3B-Base.

Three trends confirm that each stabilization mechanism targets a meaningful subset of the data. First, the dead zone (gray band) captures 36% of search steps at step 50 and still 28% at step 200, predominantly from model-proficient questions whose near-zero IG does not reflect retrieval quality (Cases 1 and 3 in Appendix F). Second, values exceeding the soft-clipping threshold η=3 grow from 15% to 20% of steps as the policy learns to issue more effective queries, consistent with the opening-query IG growing over training (Appendix O); since negative values rarely exceed -3 nats, the clipping acts almost exclusively on the positive tail. Third, the distribution shifts rightward over training (mean $0.27 \rightarrow 0.58$ nats), while the negative tail (IG $< -\delta$) remains roughly constant at $\sim$27%, confirming that asymmetric scaling is necessary to prevent these persistent negative signals from inducing search avoidance (Appendix M).

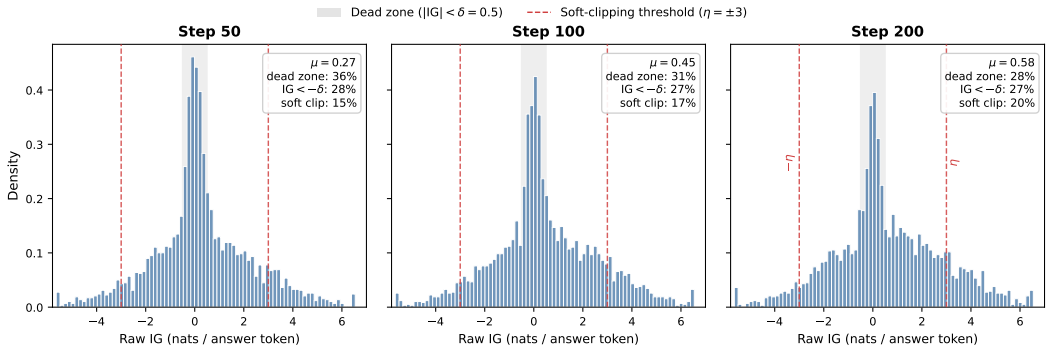

Figure 6: Raw IG distribution at three training checkpoints on Qwen2.5-3B-Base. The gray band marks the dead-zone region ($|\text{IG}| < \delta$=0.5); red dashed lines mark the soft-clipping threshold ($\eta = \pm 3$). Statistics boxes report the batch mean, dead-zone fraction, negative-surviving fraction (IG $< -\delta$), and soft-clipping activation rate ($|\text{IG}| > \eta$).

Ordering of Stabilization Mechanisms. The distribution in Figure 6 also motivates the fixed application order of the three mechanisms. *Dead zone acts first* because its threshold δ is calibrated on the raw distribution: if asymmetric scaling were applied first, the scaled negative values would lie in $[-0.3, -0.05]$ nats (since the surviving raw negatives rarely exceed -3 nats, scaling by λ=0.1 keeps them tightly around zero), and the dead zone would then erroneously filter out these informative, already-compressed signals along with the genuine noise. *Soft clipping acts last* because the scaled negatives sit well below η=3, so the clipping only engages on the genuinely large positive IGs it was designed to bound, consistent with the positive-tail activation pattern (15–20%) reported above. Permuting this order would either over-filter informative negatives or leave the positive tail unnormalized during the earlier stages of the pipeline.

F DEAD ZONE CASE STUDY

Figure 7 presents three cases from training trajectories illustrating the necessity of dead zone filtering.

Case 1 involves a question the model can already answer. Although the retrieved documents are relevant, they provide no new information, resulting in a small positive IG (0.13). Without the dead zone, this weak signal would reinforce the query despite its quality being irrelevant to the outcome.

Case 3 illustrates a more subtle failure mode driven by parametric knowledge. The real documents mention several years (1887, 1889, the 1889 World's Fair), diluting the model's confidence in "1889" specifically, while random documents leave the parametric prior intact and yield a higher probability on the gold token. The result is a *negative* IG (-0.24) despite the query being well-formed. Without the dead zone, this correct and relevant query would be erroneously penalized.

In contrast, Case 2 shows a genuine knowledge gap where the model cannot answer without retrieval, and the high IG (1.72) accurately reflects the query's effectiveness. Together these cases show that on model-proficient questions IG does not faithfully measure retrieval quality, and the dead zone is essential for filtering out such spurious signals.

Case 1: Model-proficient question, small positive IG (filtered)

Q: What is the capital of France? **Gold:** Paris

<think> The capital of France is Paris. Let me verify. </think>
<search> capital of France </search>
<documents> ... Paris is the capital and most populous city of France... </documents>
<refine> The capital of France is Paris. </refine> <answer> Paris </answer>

IG computation:
$\log \pi_\theta(\text{Paris} \mid \mathcal{C}^{\text{real}}) = -0.15, \quad \frac{1}{N}\sum_j \log \pi_\theta(\text{Paris} \mid \mathcal{C}^{\text{rand, j}}) = -0.28 \quad \Rightarrow$ **IG = 0.13**

Dead zone (δ=0.5): **filtered**. small positive IG from a model-proficient question; no spurious reinforcement.

Case 2: Knowledge-gap question, high IG (passed)

Q: Who is the father of the father of modern experimental psychology? **Gold:** Maximilian Wundt

<think> I need to find the father of modern experimental psychology first. </think>
<search> father of modern experimental psychology </search>
<documents> ... Wilhelm Wundt is widely regarded as the father of experimental psychology. He was born on 16 August 1832 to parents Maximilian Wundt... </documents>
<refine> Wilhelm Wundt is the father of experimental psychology. His father was Maximilian Wundt. </refine> <answer> Maximilian Wundt </answer>

IG computation:
$\log \pi_\theta(\text{M. Wundt} \mid \mathcal{C}^{\text{real}}) = -1.82, \quad \frac{1}{N}\sum_j \log \pi_\theta(\text{M. Wundt} \mid \mathcal{C}^{\text{rand, j}}) = -3.54 \quad \Rightarrow$ **IG = 1.72**

Dead zone (δ=0.5): **passed**. Genuine knowledge gap; high IG correctly rewards this query.

Case 3: Model-proficient question, spurious negative IG (filtered)

Q: When was the Eiffel Tower completed? **Gold:** 1889

<search> Eiffel Tower completion year </search>
<documents> ... The Eiffel Tower was constructed from 1887 to 1889 as the centerpiece of the 1889 World's Fair... </documents>
<refine> The Eiffel Tower was completed in 1889. </refine> <answer> 1889 </answer>

IG computation:
$\log \pi_\theta(1889 \mid \mathcal{C}^{\text{real}}) = -0.42, \quad \frac{1}{N}\sum_j \log \pi_\theta(1889 \mid \mathcal{C}^{\text{rand, j}}) = -0.18 \quad \Rightarrow$ **IG = -0.24**

Without dead zone: this correct, relevant query would be *penalized* because real documents introduce competing facts that dilute the parametric prior.

Dead zone (δ=0.5): **filtered**. Spurious negative signal suppressed; no incorrect penalization.

Figure 7: Dead zone filtering on three illustrative training cases. Cases 1 and 3 are model-proficient questions where raw IG fails to reflect retrieval quality and are both suppressed by the dead zone; Case 2 is a genuine knowledge gap where high IG correctly rewards the query and passes the filter.

G Dataset Statistics

Table 4 summarizes the seven question answering benchmarks used in our experiments. All datasets are sourced from the FlashRAG datasets collection. We construct the training set by combining the train splits of NQ and HotpotQA, following Jin et al. (2025). For evaluation we use the test split where available (NQ, TriviaQA, PopQA, Bamboogle) and the dev split otherwise (HotpotQA, 2Wiki, Musique).

Table 4: Statistics of the seven benchmarks used in this paper.

Dataset	**Split sizes**			**Eval split**
	Train	Dev	Test	
NQ	79,168	8,757	3,610	test
TriviaQA	78,785	8,837	11,313	test
PopQA	–	–	14,267	test
HotpotQA	90,447	7,405	–	dev
2Wiki	15,000	12,576	–	dev
Musique	19,938	2,417	–	dev
Bamboogle	–	–	125	test

H Implementation Details

All experiments are conducted on 8 NVIDIA H800 GPUs using the veRL framework (Sheng et al., 2025). The primary experiments use Qwen2.5-3B (Qwen, 2025) in both base and instruct variants; scaling experiments with Qwen2.5-7B are reported in §3.3.2 and extended to four model sizes in Appendix K.

Rollout and retrieval. We sample G=5 trajectories per question with temperature 1.0. Each rollout is allowed at most $T_{\max}$=5 search calls, after which the model is forced to produce a final answer. The retrieval engine uses E5-base-v2 over a December 2018 Wikipedia dump and returns the top 3 documents per query.

IG computation. For each search step we construct N=3 counterfactual contexts using randomly sampled (documents, refinement) pairs from the training batch. The IG-specific hyperparameters are α=0.3, δ=0.5, η=3.0, λ=0.1. A single set of hyperparameters is used throughout all experiments without dataset-specific tuning; their robustness across broad ranges is verified in Appendix N.

Optimization. Training runs for 200 steps with a learning rate of 1×10^{-6} and KL coefficient β=0.001. The complete hyperparameter configuration is listed in Table 5.

Table 5 lists the complete set of hyperparameters used in our experiments. The IG-specific hyperparameters ($\alpha, \delta, \lambda, \eta, N$) were selected on NQ and HotpotQA, with default values shown in the table. All other values follow the default configuration of the veRL framework (Sheng et al., 2025).

Table 5: Complete hyperparameter configuration for IG-Search training.

Hyperparameter	**Value**
Training configuration	
Group size G	5
Training batch size	256
Actor learning rate	1×10^{-6}
Total training steps	200
KL coefficient β	0.001
Clip ratio ϵ	0.2
Max search calls per rollout	5
Rollout temperature	1.0
Retrieval configuration	
Retriever	E5-base-v2
Corpus	Wikipedia (Dec. 2018)
Top-k documents per query	3
IG-Search specific	
IG weight α	0.3
Dead zone threshold δ	0.5
Asymmetric negative scale λ	0.1
Soft clipping threshold η	3.0
Number of counterfactual contexts N	3
Max context length for IG	8192
Max gold answer variants	3

I TRAINING OVERHEAD

Table 6 reports the per-step wall-clock time breakdown. The IG computation adds approximately 6.7% to total step time. **This overhead is training-only and does not affect inference latency.**

Table 6: Per-step wall-clock time on 8×H800 with Qwen2.5-3B-Base (seconds, averaged over steps 2–5).

Component	**AutoRefine**	**IG-Search**
Rollout generation	91	91
Reference log-prob	5	5
Reward + KL + Advantage	4	4
IG computation	–	8
Policy update	19	19
Total per step	119	127
IG overhead	–	+6.7%

J STATISTICAL SIGNIFICANCE

To validate the stability of our results, we train IG-Search-Base and AutoRefine-Base with three random seeds on Qwen2.5-3B-Base. Table 1 reports baseline results from their papers. Table 7 additionally reports the mean and standard deviation across three independent seeds; the means may therefore deviate from the Table 1 values by an amount consistent with the reported standard deviations. A paired t-test over the seven benchmarks yields $p < 0.01$ for IG-Search-Base versus AutoRefine-Base, and the error bars are non-overlapping on all four multi-hop benchmarks, confirming that the improvements are statistically significant and not an artifact of seed variance.

Table 7: Exact Match (mean_{std}) across three random seeds on Qwen2.5-3B-Base.

Method	**Single-Hop QA**			**Multi-Hop QA**				Avg.
	NQ	TriviaQA	PopQA	HotpotQA	2Wiki	Musique	Bamboogle	
AutoRefine-Base	$0.464_{.008}$	$0.622_{.006}$	$0.452_{.009}$	$0.405_{.010}$	$0.397_{.011}$	$0.162_{.007}$	$0.338_{.015}$	$0.406_{.008}$
IG-Search-Base	$0.476_{.006}$	$0.637_{.005}$	$0.476_{.007}$	$0.436_{.008}$	$0.415_{.009}$	$0.179_{.005}$	$0.390_{.011}$	$0.430_{.005}$

K SCALING ACROSS MODEL SIZES

We train IG-Search and AutoRefine on Qwen2.5 at four scales under identical settings and report average EM across the seven benchmarks at step 200 (Figure 8). IG-Search delivers consistent gains over AutoRefine at every scale, with improvements of +3.5, +2.5, +2.1, and +2.3 points at 1.5B, 3B, 7B, and 14B respectively. The gains are stable across scales, showing that step-level IG supervision remains beneficial as model capacity grows, rather than being a crutch that stronger policies can recover from outcome-based rewards alone.

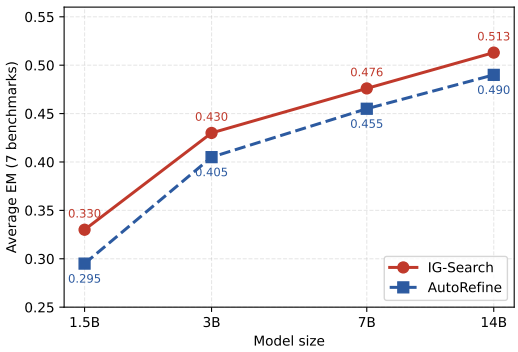

Figure 8: IG-Search yields consistent gains over AutoRefine at every scale, demonstrating that step-level IG supervision scales effectively with model capacity.

L Cross-Protocol Comparison with IGPO

IGPO and IG-Search are evaluated under substantially different protocols. IGPO uses a live Google Search API, whereas IG-Search uses a frozen 2018 Wikipedia snapshot with E5-base-v2; IGPO allocates a $6.4\times$ larger exploration budget (G=16, $T_{\max}$=10 vs. G=5, $T_{\max}$=5); and IGPO's training mixture includes TriviaQA and 2Wiki, which are therefore in-domain for IGPO but out-of-domain for IG-Search. All three differences favor IGPO in absolute terms, so the comparison below is a lower bound on the relative advantage of our method.

These differences preclude including IGPO in our main results table. However, since IGPO reports word-level F1, we compute the same metric for IG-Search to enable a comparison at the 3B scale. Table 8 shows that IG-Search-Instruct achieves an average F1 of 0.518 versus IGPO's 0.489, a 2.9-point improvement despite the strictly weaker setup. The contrast is particularly striking on TriviaQA and 2Wiki, where IGPO holds the in-domain advantage yet IG-Search matches or closely trails it (TriviaQA: 0.692 vs. 0.692; 2Wiki: 0.507 vs. 0.514).

Table 8: Cross-protocol comparison with IGPO on Qwen2.5-3B-Instruct (F1). IGPO numbers are copied from their paper. IG-Search numbers are computed under the same metric.

Method	**Single-Hop QA**			**Multi-Hop QA**				**Avg.**
	NQ	TriviaQA	PopQA	HotpotQA	2Wiki	Musique	Bamboogle	
IGPO	0.419	0.692	0.490	0.478	**0.514**	0.248	**0.584**	0.489
IG-Search-Instruct	**0.542**	**0.692**	**0.501**	**0.551**	0.507	**0.276**	0.554	**0.518**

M SEARCH AVOIDANCE UNDER SYMMETRIC NEGATIVE SCALING

The λ ablation in Table 2 and the sensitivity curve in Figure 10 both show that setting $\lambda = 1.0$ causes a substantially larger performance drop than any other single-component ablation. We attribute this degradation to *search avoidance*: the policy learns to stop issuing search queries altogether, thereby escaping the negative IG signal. To verify this, we track the average number of search calls per rollout throughout training under the two settings, with all other hyperparameters held at their defaults.

Figure 9 reports the result on Qwen2.5-3B-Base. Both runs start from the same initialization and exhibit similar exploration in the first $\sim$30 steps, after which they diverge sharply: the default run stabilizes at approximately 1.85 search calls per rollout, while the symmetric run decays monotonically toward 1.0, as full-strength negative IG on early poorly-formed queries progressively suppresses the policy's tendency to search. This provides direct mechanistic evidence that the performance drop under $\lambda = 1.0$ is mediated by learned search avoidance, justifying the asymmetric design.

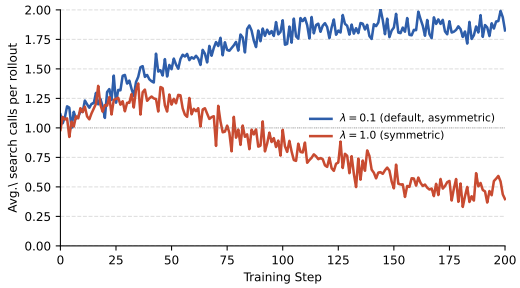

Figure 9: Average number of search calls per rollout during training on Qwen2.5-3B-Base, comparing the default asymmetric setting ($\lambda = 0.1$) against symmetric negative scaling ($\lambda = 1.0$). The two variants share the same initialization and exhibit similar exploration in the first $\sim$30 steps; the symmetric variant then decays monotonically toward 1.0 as full-strength negative IG penalties suppress search usage, while the default setting stabilizes around 1.85.

N HYPERPARAMETER SENSITIVITY

We analyze the sensitivity of IG-Search to five IG-specific hyperparameters (α, λ, δ, η, N) and to the retrieval depth k, varying each while fixing the others at their default values. All experiments use Qwen2.5-3B-Base. Results are shown in Figure 10.

The IG weight α peaks at 0.3; smaller values underweight the step-level signal while larger values cause it to dominate the trajectory-level advantage. The asymmetric scaling factor λ exhibits the sharpest sensitivity: symmetric treatment (λ=1.0) degrades performance to 0.391, consistent with the search-avoidance dynamics documented in Appendix M. The dead zone threshold δ and soft clipping threshold η are comparatively robust across broad ranges. The number of counterfactual contexts N saturates at N=3 (0.430), with no further gain from larger values. Retrieval depth k peaks at k=3: k=1 provides insufficient evidence, while higher k introduces noise from less relevant documents.

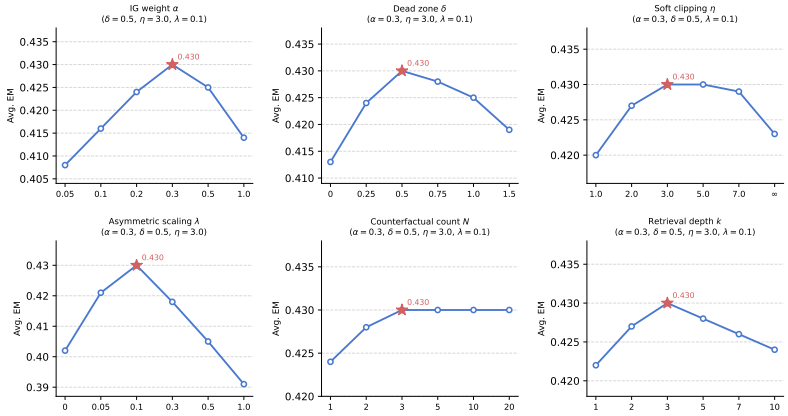

Figure 10: Sensitivity of IG-Search to five IG-specific hyperparameters (α, λ, δ, η, N) and to the retrieval depth k, on Qwen2.5-3B-Base (average EM across seven benchmarks). Red stars indicate default values.

O PER-POSITION DECOMPOSITION OF THE IG SIGNAL

A natural concern with any learned reward is whether it captures something structurally meaningful or merely fits noise correlated with the outcome. We address this by decomposing the per-search-step IG by its position within a rollout on Qwen2.5-3B-Base. If IG genuinely measures the marginal informational contribution of a retrieval, then the first search, which operates *without any prior retrieved evidence*, should carry more information than later searches, which retrieve on top of already populated evidence.

In Figure 11, we plot the mean IG of the k-th search step, $\mathrm{IG}^{(k)}$, for $k \in \{0, 1, 2\}$ throughout training. Three key observations stand out. First, the ordering $\mathrm{IG}^{(0)} > \mathrm{IG}^{(1)} > \mathrm{IG}^{(2)}$ holds empirically from the first training iteration, before any policy update has shaped the signal, showing that IG is aligned with the diminishing-marginal-value structure of multi-hop retrieval from the outset rather than as a byproduct of optimization. Second, all three curves rise over training: $\mathrm{IG}^{(0)}$ grows from roughly 0.30 to 0.92 nats and $\mathrm{IG}^{(1)}$ from near zero to approximately 0.32, indicating that IG-Search improves not only the opening query but also the follow-ups, making later searches genuinely additive rather than redundant. Note that Figure 11 reports raw IG values before dead-zone filtering; because the per-step distribution has high variance, a positional mean below δ does not imply that all individual values at that position are filtered. Third, the gap between positions does not collapse. A reward-hacking policy that inflated IG uniformly across all positions would produce converging curves; the persistent separation instead shows that the policy continues to respect the structural asymmetry between hops. Together these dynamics indicate that the signal IG-Search optimizes is aligned with the information flow of multi-hop reasoning rather than being an arbitrary scalar that happens to correlate with answer correctness.

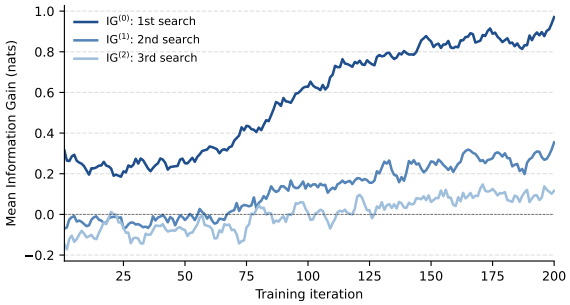

Figure 11: Mean *raw* Information Gain (before dead-zone filtering) decomposed by search-step position $k \in \{0, 1, 2\}$ during training on Qwen2.5-3B-Base. IG values are reported in nats per answer token. Each curve tracks the average IG of the $(k{+}1)$-th search call across all training questions. The ordering $\mathrm{IG}^{(0)} > \mathrm{IG}^{(1)} > \mathrm{IG}^{(2)}$ holds empirically from the first iteration and the gap persists throughout training, indicating that the signal is aligned with the diminishing-marginal-value structure of multi-hop retrieval.

P SEARCH BEHAVIOR ACROSS QUESTION TYPES

We examine whether IG-Search learns to calibrate its search depth to the complexity of the question. On single-hop benchmarks, the trained policy issues on average 1.3 search calls per rollout, whereas on multi-hop benchmarks it issues 2.4. By contrast, the trajectory-level baseline AutoRefine-Base uses 2.0 and 2.3 search calls respectively on the same two categories. IG-Search thus exhibits a more task-adaptive search pattern: it is much more conservative on questions that can be resolved by a single retrieval, while remaining active on questions that genuinely require multi-hop evidence. As noted in §3.2, this reduction in search frequency does not sacrifice accuracy: IG-Search-Base outperforms AutoRefine-Base on all three single-hop datasets, confirming that the savings come from suppressing redundant queries rather than from sacrificing recall.

We further compute per-position IG on the evaluation benchmarks at the final checkpoint. Figure 12 breaks down the per-position IG on the two categories. On single-hop questions nearly all of the information gain is concentrated in the opening search ($\mathrm{IG}^{(0)} = 1.18$), while any subsequent retrieval contributes less ($\mathrm{IG}^{(1)} = 0.24$), indicating that a second search is usually an unnecessary verification. On multi-hop questions the signal is distributed more evenly across steps ($\mathrm{IG}^{(0)} = 0.91$, $\mathrm{IG}^{(1)} = 0.68$, $\mathrm{IG}^{(2)} = 0.31$), showing that each additional hop continues to bring in new evidence. These observations jointly indicate that IG-Search does not merely learn to search more; it learns *when* to search and when to stop.

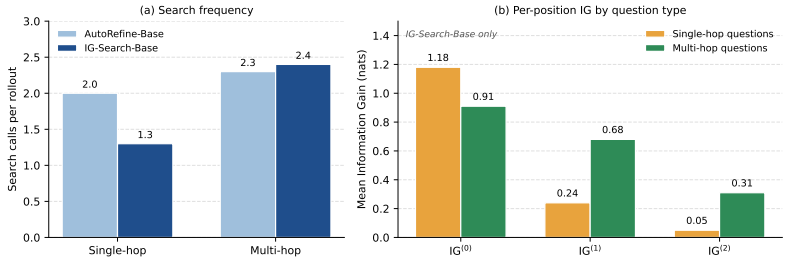

Figure 12: Search behavior of IG-Search-Base across question types. (a) Average number of search calls per rollout on single-hop vs. multi-hop benchmarks, compared with AutoRefine-Base. (b) Mean Information Gain at each search-step position, split by question type. On single-hop questions, nearly all information is captured by the first search; on multi-hop questions, subsequent searches continue to contribute.

Q CORRECT TRAJECTORY CASE STUDY

Figure 13 contrasts IG-Search with AutoRefine on a Bamboogle question where both models retrieve from the *same* corpus with the *same* retriever, but arrive at different answers. AutoRefine issues a single long compound query that mixes two sub-goals ("main actor" and "licensed what"), retrieving documents dominated by Harrison Ford's acting career; the token "actor" dominates the refinement and the policy commits to it. IG-Search instead decomposes the question into two focused queries. The first query identifies Harrison Ford as the main actor; the second explicitly asks "*What is Harrison Ford licensed as?*" and retrieves the sentence stating he is "a licensed pilot of both fixed-wing aircraft and helicopters," yielding the correct answer. This case illustrates a failure mode of trajectory-level supervision: when a vague compound query happens to return snippets containing a plausible-sounding wrong answer, the trajectory-level reward cannot distinguish it from a precise query that targets the actual information need. The step-level IG signal, by contrast, directly rewards the retrieval that raises the gold-answer probability, pushing the policy toward the decomposed formulation.

Question: The main actor of Indiana Jones is a licensed what?
Gold Answer: **pilot**

AutoRefine (Trajectory A, wrong answer)

<search> The main actor of Indiana Jones is a licensed what </search>
<documents> ... Harrison Ford... is an American **actor**, writer and film producer... </documents>
<refine> The main actor of Indiana Jones is Harrison Ford. </refine>

<search> Harrison Ford licensed role </search>
<documents> ... American **actor**, writer and film producer... starring roles as Han Solo... title character of the *Indiana Jones* film series... portraying Han Solo... Indiana Jones... Rick Deckard... </documents>
<refine> Harrison Ford is licensed as an **actor**. </refine>
<answer> **actor** </answer>

The compound first query and the under-specified second query both retrieve documents dominated by Ford's acting career. No retrieval ever surfaces the actual licensed occupation; the policy commits to the salient (but wrong) token "actor."

IG-Search (Trajectory B, correct answer)

<think> First, I'll search for the main actor of Indiana Jones. </think>
<search> The main actor of Indiana Jones </search>
<documents> ... Harrison Ford... as the title character of the *Indiana Jones* film series... Jones is most famously portrayed by Harrison Ford... </documents>
<refine> The main actor of Indiana Jones is Harrison Ford. </refine>

<search> What is Harrison Ford licensed as? </search>
<documents> ... Ford is a **licensed pilot** of both fixed-wing aircraft and helicopters, and owns a ranch in Jackson, Wyoming... On several occasions, Ford has personally provided emergency helicopter services... </documents>
<refine> Harrison Ford is a **licensed pilot** of both fixed-wing aircraft and helicopters. </refine>
<answer> **pilot** </answer>

The second query explicitly targets the licensed occupation, retrieving the answer-bearing sentence directly. The step-level IG on this query is high, reinforcing the decomposed formulation over a compound one.

Figure 13: Trajectory comparison on a Bamboogle question under identical retrieval setup. AutoRefine issues a compound query whose retrieval is dominated by Ford's acting career, leading to the wrong answer "actor." IG-Search decomposes the question into two focused queries; the second explicitly asks for Ford's licensed occupation and retrieves the answer-bearing sentence, yielding the correct answer "pilot."

R ALL-FAILURE CASE STUDY

Figure 14 presents a training batch in which all five rollouts produce incorrect final answers, illustrating how IG-Search provides a differentiating signal where trajectory-level rewards cannot. Rollout A issues a precise opening query that retrieves the answer-bearing document ($\mathrm{IG}_A^{(0)} = 1.58$) and fails only at subsequent reasoning; Rollout B issues a vague query whose retrieved documents are only tangentially related ($\mathrm{IG}_B^{(0)} = 0.08$). Under GRPO, the group-normalized advantages $\hat{A}_A$ and $\hat{A}_B$ both collapse to zero because every rollout shares the same outcome reward $r_{\mathrm{F1}} = 0$. The step-level IG modulation, however, continues to distinguish the two: Rollout A's opening query tokens receive substantial positive modulation from $\mathrm{IG}_A^{(0)}$, while Rollout B's receive near-zero modulation, pushing the policy toward the more informative query formulation even when no trajectory succeeds.

Question: Who directed the film that won the Academy Award for Best Picture in 1994?
Gold Answer: **Robert Zemeckis**
Batch status: all 5 rollouts answered incorrectly.

Rollout A (informative search, wrong final answer)

<think> I need to find which film won Best Picture at the 1994 Academy Awards, then its director. </think>
<search> 1994 Academy Award Best Picture winner </search>
<documents> Forrest Gump, directed by Robert Zemeckis, won the Academy Award for Best Picture at the 67th Academy Awards held in March 1995, honoring films of 1994... </documents>
<refine> Forrest Gump won Best Picture for 1994. </refine>
<search> Forrest Gump director </search>
<documents> Forrest Gump is a 1994 American comedy-drama film based on the 1986 novel by Winston Groom... The film was produced by Wendy Finerman... </documents>
<refine> Forrest Gump was produced by Wendy Finerman. </refine> <answer> Wendy Finerman </answer>

IG computation: $\mathrm{IG}_A^{(0)} = 1.58$ (retrieved document explicitly mentions the gold answer), $\mathrm{IG}_A^{(1)} = 0.12$ (second retrieval loses focus).
Trajectory reward: $r_{\mathrm{F1}} = 0$. *GRPO advantage:* $\hat{A}_A \approx 0$.

IG-Search: opening query tokens receive positive modulation from $\mathrm{IG}_A^{(0)}$, reinforcing the precise query formulation.

Rollout B (uninformative search, wrong final answer)

<think> Let me look up Best Picture winners. </think>
<search> best picture movies </search>
<documents> The Academy Award for Best Picture is one of the Academy Awards presented annually by the Academy of Motion Picture Arts and Sciences... The first winner was Wings in 1929... </documents>
<refine> Best Picture has been awarded since 1929. </refine>
<search> movie directors 1990s </search>
<documents> Notable directors of the 1990s include Quentin Tarantino, Martin Scorsese, Steven Spielberg... </documents>
<refine> Steven Spielberg is a notable 1990s director. </refine> <answer> Steven Spielberg </answer>

IG computation: $\mathrm{IG}_B^{(0)} = 0.08$ (generic information, no year-specific content), $\mathrm{IG}_B^{(1)} = -0.04$ (retrieval drifts further from gold).
Trajectory reward: $r_{\mathrm{F1}} = 0$. *GRPO advantage:* $\hat{A}_B \approx 0$.

IG-Search: opening query tokens receive near-zero modulation, offering no reinforcement to the vague formulation.

Figure 14: Case study from an all-failure batch (all five rollouts receive $r_{\mathrm{F1}} = 0$ and thus $\hat{A}_i \approx 0$). Rollouts A and B receive identical trajectory-level advantages despite searches of very different quality; IG-Search differentiates them via per-step IG, providing nonzero gradient signal for the query tokens and pushing the policy toward the more informative formulation.

S CONTROLLED COMPARISON WITH IGPO AND REWARD POSITIONING

The methodological distinction of IG-Search is the combination of a structure-matched random-document counterfactual and query-token-localized credit assignment. Gold-answer likelihood itself is a shared supervision source, rather than a new contribution. This section separates the controlled empirical comparison from the cross-protocol results in Appendix L.

What is contrasted and where credit is assigned. IGPO (Wang et al., 2026) computes a temporal difference in gold-answer log-probability between consecutive rollout prefixes. Because a new prefix adds an entire turn, the difference reflects reasoning, query formulation, and the retrieved tool response together. IG-Search holds the preceding trajectory and current query fixed, replacing only the current (documents, refinement) block with a random block from another question. The comparison preserves the retrieval-slot structure while changing the evidence, and the resulting signal is assigned specifically to the query tokens that produced that retrieval. IG-Search also retains the trajectory-level GRPO advantage on all generated tokens.

This differs from InfoGain-RAG (Wang et al., 2025), which uses document information gain as offline supervision for a separate reranker and filtering module after retrieval. IG-Search computes the signal within on-policy rollouts to update the query-producing policy, without an additional inference-time component. The two approaches optimize different decisions: document selection after retrieval and query generation before retrieval.

Matched implementation. Both methods start from Qwen2.5-3B-Instruct and use the same NQ and HotpotQA training mixture, frozen December 2018 Wikipedia corpus, E5-base-v2 retriever with top-3 documents, rollout budget, optimizer, and 200-step training schedule. The common configuration follows Appendix H, including $G{=}5$ trajectories per question and at most $T_{\max}{=}5$ search calls per rollout. In the IGPO variant, we replace the IG-Search counterfactual reward and query-token modulation with IGPO's adjacent-prefix reward, separate group normalization, discounted returns, and turn-level credit assignment.

Table 9: Controlled comparison on Qwen2.5-3B-Instruct. Average Exact Match is measured across the same seven benchmarks under a shared training and retrieval protocol.

Method	Avg. EM
IGPO (matched implementation)	0.402
IG-Search	**0.430**

IG-Search improves average EM by 2.8 points in this matched setting (Table 9). This comparison evaluates the combined reward construction, calibration, and credit-assignment design; it does not isolate any one component. Component evidence comes from the existing ablations: query-only modulation obtains 0.430 on Qwen2.5-3B-Base, versus 0.422 for think+query and 0.413 for all-token modulation (Table 2). Replacing the random counterfactual with an empty-context baseline reduces average EM from 0.430 to 0.403 (Appendix D). The latter changes both evidence content and sequence structure, whereas a random block retains the retrieval slot while removing topical relevance. Appendix U further tests whether an arbitrary dense query-token signal can account for the improvement.

T EVALUATION ON NEWER MODEL BACKBONES

We compare Search-R1, AutoRefine, GiGPO, MR-Search, and IG-Search on Qwen3-4B/8B and evaluate GiGPO and IG-Search on Qwen3.5-4B/9B to test transfer beyond Qwen2.5.

Controlled setup. Within each backbone, methods share the NQ+HotpotQA training mixture, frozen December 2018 Wikipedia corpus, E5-base-v2 top-3 retriever, rollout budget (G=5 trajectories per question and at most $T_{\max}$=5 search calls per trajectory), and evaluation protocol (Appendix H). Qwen3 comparisons retain fixed method-specific hyperparameters; IG-Search retains α=0.3, δ=0.5, λ=0.1, η=3.0, and N=3 across both families, without retuning. We disable Qwen3's native thinking mode and keep the explicit `<think>`, `<search>`, `<documents>`, `<refine>`, and `<answer>` blocks used in the main Qwen2.5 experiments, preserving the interaction format across methods. Reported averages weight all seven benchmarks equally.

Qwen3 results. IG-Search achieves the highest average EM at both scales (Table 10): 0.446 at 4B and 0.479 at 8B, exceeding GiGPO, the strongest baseline, by 0.8 and 0.5 percentage points. On the multi-hop subset (HotpotQA, 2Wiki, MuSiQue, and Bamboogle), IG-Search improves over GiGPO by approximately 1.6 points at 4B; the methods are essentially tied at 8B.

Table 10: Qwen3 Exact Match. Bold denotes the best result within each model size.

Method	Single-Hop QA			Multi-Hop QA				Avg.
	NQ	TriviaQA	PopQA	HotpotQA	2Wiki	MuSiQue	Bamboogle	
Qwen3-4B								
Search-R1	0.402	0.611	0.421	0.335	0.312	0.128	0.368	0.368
AutoRefine	0.418	0.642	**0.500**	0.418	0.361	0.157	0.464	0.423
GiGPO	0.449	0.643	0.467	0.424	**0.428**	0.184	**0.472**	0.438
MR-Search	**0.490**	**0.652**	0.473	0.431	0.413	0.179	0.351	0.427
IG-Search	0.455	0.632	0.465	**0.454**	0.427	**0.221**	0.468	**0.446**
Qwen3-8B								
Search-R1	0.440	0.631	0.418	0.372	0.355	0.157	0.430	0.400
AutoRefine	0.450	0.665	0.470	0.421	0.369	0.219	0.496	0.441
GiGPO	0.479	0.662	0.502	**0.486**	0.417	0.226	**0.544**	0.474
MR-Search	**0.511**	0.673	0.479	0.475	0.442	0.224	0.451	0.465
IG-Search	0.486	**0.689**	**0.503**	0.485	**0.474**	**0.251**	0.465	**0.479**

Qwen3.5 results. IG-Search reaches average EM of 0.452 at 4B and 0.473 at 9B (Table 11), improving over GiGPO by 1.1 and 0.4 percentage points. Its multi-hop average improves by 2.2 points at 4B and ties GiGPO at 9B. Across both families, multi-hop gains are stronger at smaller scales, while IG-Search retains the higher overall average at larger scales.

Table 11: Qwen3.5 Exact Match. Bold denotes the best result within each model size.

Method	Single-Hop QA			Multi-Hop QA				Avg.
	NQ	TriviaQA	PopQA	HotpotQA	2Wiki	MuSiQue	Bamboogle	
Qwen3.5-4B								
GiGPO	0.461	**0.634**	**0.484**	0.418	0.432	0.195	0.463	0.441
IG-Search	**0.464**	0.628	0.476	**0.460**	**0.435**	**0.228**	**0.473**	**0.452**
Qwen3.5-9B								
GiGPO	0.470	0.656	**0.506**	0.478	0.412	0.223	**0.538**	0.469
IG-Search	**0.480**	**0.684**	0.496	**0.481**	**0.466**	**0.248**	0.456	**0.473**

U CONTROLS ON THE INFORMATION CARRIED BY THE IG SIGNAL

The component ablations in Table 2 establish that stabilization is necessary for effective optimization, but do not by themselves distinguish informative retrieval-specific credit from an arbitrary dense query-token bonus. We therefore keep the optimization pipeline fixed and alter only the information carried by the processed step-level IG values.

Control construction. Let $z_t = \widetilde{\mathrm{IG}}_t$ be the processed IG value for search step t, and let $\mathcal{B}$ contain the search steps in a minibatch. The controls replace z_t after IG processing and before the existing query-length normalization and advantage modulation in Eq. 4:

- **Shuffled IG:** randomly permute the processed values across search steps within the minibatch. This preserves their distribution but removes alignment with the originating search actions.
- **Constant IG:** assign every search step the minibatch mean, $|\mathcal{B}|^{-1} \sum_{u \in \mathcal{B}} z_u$, removing step-specific variation.
- **Sign-only IG:** preserve each step's positive, zero, or negative sign, but replace each nonzero magnitude with the minibatch mean absolute magnitude, $|\mathcal{B}|^{-1} \sum_{u \in \mathcal{B}} |z_u|$. Zero signals remain zero.

Table 12: Information-content controls with the optimization pipeline held fixed. Scores are average Exact Match across seven benchmarks. The shuffled, constant, and sign-only variants modify processed IG before query-length normalization.

Configuration	Avg. EM
IG-Search	**0.430**
Shuffled processed IG across search steps	0.401
Constant processed IG	0.404
Sign-only processed IG	0.419
Without IG	0.403

Step-specific content matters. Shuffled and constant IG return to approximately the no-IG score (Table 12), despite retaining a dense query-token signal. Sign-only IG recovers 1.6 of the full 2.7 EM-point improvement over the no-IG baseline; retaining calibrated magnitudes adds another 1.1 points. These results support the importance of step-specific alignment and polarity, with magnitude providing additional useful information. They do not imply that the stabilization pipeline alone creates the learning signal.

Why miscalibrated IG can be worse than no IG. The stabilization operators act only on IG, so disabling IG also removes their effect and gives the 0.403 no-IG control. In contrast, removing asymmetric negative scaling leaves a dense negative signal active and reduces average EM to 0.391. In this run, search frequency declines toward one call per rollout, whereas the default run stabilizes near 1.85 (Appendix M). Dead-zone filtering suppresses near-zero values that poorly distinguish query quality; asymmetric scaling limits search avoidance; soft clipping controls the heavy tail; and query-length normalization prevents a larger accumulated bonus from longer queries. Without the latter, the policy generates abnormally long queries. The retrieval-specific, step-aligned IG supplies the learning content, while these mechanisms make that content more reliably optimizable.

V SCOPE OF STEP-LEVEL CREDIT IN LONGER-HORIZON SEARCH

Intermediate evidence can receive immediate credit. The current formulation does not require each retrieved document to reveal the final answer directly. IG at step t conditions on the complete preceding trajectory, so it measures the contribution of the current retrieval on top of previously acquired evidence. A document about a bridge entity can receive positive IG without containing the gold-answer string, provided it increases the policy's probability of the final answer. This distinguishes a likelihood-based signal from an answer-string occurrence reward.

Evidence within the evaluated search horizon. Later searches are useful in the present experiments. Restricting modulation to the first query reduces average EM from 0.430 to 0.417 on Qwen2.5-3B-Base (Table 2). The trained Base policy averages 1.3 searches on single-hop questions and 2.4 on multi-hop questions; the mean raw IG of its first three multi-hop searches is 0.91, 0.68, and 0.31 nats per answer token (Appendix P). These are evaluation-time statistics at the final checkpoint, distinct from the training-batch positional curves in Appendix O. Together with the two-search example in Appendix Q, they show useful multi-step information acquisition within the evaluated budget of at most five search calls. They do not establish effectiveness on substantially longer deep-research tasks.

Purely delayed utility remains a limitation. An early search may be necessary for later reasoning while leaving current gold-answer confidence unchanged. In that case, immediate IG can under-credit the action even if subsequent retrievals reveal its value. The trajectory-level reward remains active, but localized IG may supply little additional credit. The bridge-verification failures in Appendix **??** illustrate a related need to evaluate intermediate choices through their downstream consequences. Potential extensions include discounted future IG, a retrieval counterfactual evaluated against a later trajectory suffix, or a combination with state-relative discounted returns. These are directions for future work, rather than evaluated components of IG-Search.

Relation to the policy optimizer. We keep GRPO fixed to isolate credit assignment under the optimization framework shared by the most directly comparable baselines. At the signal level, IG-Search produces a scalar credit for each retrieval step and can in principle be incorporated into policy-gradient objectives that support token-level advantage weighting. Experiments here establish its behavior with GRPO; integration with other optimizers would require a separate evaluation.