Improving Search Agent Policy Optimization with one Line of Code

Anonymous ACL submission

Abstract

Training search agents that autonomously interact with external tools via reinforcement learning is hindered by a recurring instability that we term Importance Sampling Distribution Drift (ISDD). Within Group Relative Policy Optimization (GRPO), ISDD manifests as a rapid decay of importance sampling ratios, which suppresses gradient signals and can ultimately precipitate training collapse. To address this limitation, we propose **S**earch **A**gent **P**olicy **O**ptimization (**SAPO**), which stabilizes training through a conditional token-level KL constraint. Unlike hard clipping, which does not explicitly account for distributional divergence, SAPO selectively penalizes the KL divergence between the current and old policies, targeting only positive-advantage tokens whose probabilities have shifted beyond a threshold. This sparse intervention mitigates drift while preserving gradient flow elsewhere. Notably, SAPO requires only a single-line code modification to standard GRPO, enabling straightforward adoption. Experiments on seven QA benchmarks show that SAPO improves average Exact Match from 0.336 to 0.442 over the Search-R1-Instruct baseline ($+10.6$ absolute points, $+31.5\%$ relative), with consistent gains across model scales (1.5B–14B) and architectures (Qwen, LLaMA). Code is available at `https://anonymous.4open.science/r/SAPO-5CEF`.

1 Introduction

Recent studies (Li et al., 2025a; Guo et al., 2024) introduce search agents that interpret user intent, plan strategies, execute multi-turn actions, and accumulate retrieved evidence for answer generation. Representative methods such as Search-R1 (Jin et al., 2025) adopt Group Relative Policy Optimization (GRPO) (Shao et al., 2024) with rule-based rewards to train agents that interact with a search tool over large corpora. Consequently, GRPO has become a widely used post-training algorithm for Tool-based Agentic Reinforcement Learning (TARL), inspiring a growing body of follow-up work on both task-side designs (Mei et al., 2025; Shi et al., 2025a; Qian and Liu, 2025; Zhang et al., 2025b) and algorithm-side stabilizers within the GRPO family, including DAPO (Yu et al., 2025) and GSPO (Zheng et al., 2025).

Despite these encouraging results, existing TARL frameworks suffer from a recurring failure mode that we term **Importance Sampling Distribution Drift (ISDD)**: importance sampling (IS) ratios decay toward zero as the policy moves outside the trust region, and hard clipping (Schulman et al., 2017) alone does not arrest this drift, especially as the number of search turns grows. A key driver lies in GRPO's coarse advantage estimation: every token in a response shares one advantage value, which creates two issues. (1) *Inaccurate credit assignment*: a single advantage cannot reflect the heterogeneous contributions of individual tokens, since different search steps weigh unequally on the final answer. (2) *Misleading updates*: a response with a correct final answer may contain erroneous intermediate steps, and vice versa. The resulting conflict is concrete: when the old policy assigns high probability to a correct intermediate token but the current policy suppresses it under a negative advantage, the IS weight is driven toward zero.

Figure 1 illustrates the training dynamics of Search-R1 under GRPO. The IS ratio starts near 1 but progressively decays as the current and old policies diverge, while the clip ratio rises in tandem. Policy entropy is non-monotonic: it first declines as the policy commits to high-reward patterns, then rises once IS-ratio drift exposes low-probability positive tokens, and finally collapses alongside the reward. Because IS ratios systematically decay, we identify the **suppression of low-probability positive tokens** as the principal driver of ISDD. The outcome-based reward improves early but degrades in the late stage, indicating that late-stage instabil-

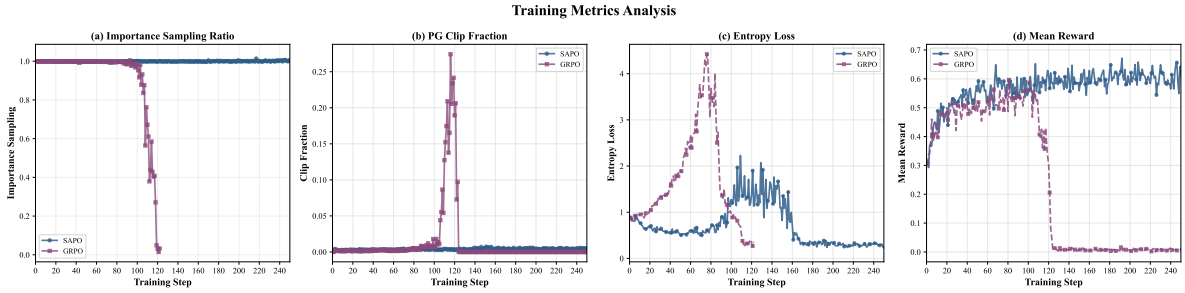

Figure 1: Training dynamics of **SAPO** vs. GRPO on Search-R1 (Qwen2.5-3B-Instruct). (a) The IS ratio decays toward 0 under GRPO but stays near 1 under SAPO; (b) GRPO's clip ratio rises in tandem, indicating hard clipping cannot arrest the drift; (c) GRPO's entropy is non-monotonic and eventually collapses, whereas SAPO decreases smoothly; (d) GRPO's reward degrades late in training, while SAPO keeps improving.

ity erodes final performance and that hard clipping alone cannot prevent collapse.

The primary motivation behind IS weighting and clipping is to correct distribution shift while preserving stability. For positive tokens, however, when the current-policy probability drops far below the old-policy probability, the update is effectively suppressed by an excessively small IS weight, leaving the model unable to recover valid actions. This motivates the following question:

How can we mitigate such conflicts with appropriate constraints when the current policy diverges significantly from the old policy on positive tokens?

To improve training stability, we propose **S**earch **A**gent **P**olicy **O**ptimization (SAPO), a principled yet minimal modification of GRPO. Inspired by PPO-KL (Schulman et al., 2017), SAPO introduces a conditional, token-level KL penalty on the divergence between the current and old policies. To focus the critical constraint, an asymmetric mechanism selectively penalizes conflicting positive tokens while preserving informative gradients elsewhere. In effect, SAPO acts as a soft trust-region constraint: rather than hard-clipping, it penalizes large IS-ratio deviations, allowing adaptive step sizes while maintaining distributional proximity.

Our contributions are summarized as follows:

- We identify and formalize Importance Sampling Distribution Drift (ISDD) as a previously under-recognized failure mode of GRPO in multi-turn agentic RL, and analyze why it amplifies with the number of search turns.
- We propose SAPO, which augments GRPO with a conditional token-level KL penalty that selectively constrains divergence on low-probability positive tokens. The modification requires only a single line of code on top of standard GRPO.
- We evaluate SAPO on seven QA benchmarks, four model scales (1.5B–14B), and two model families (Qwen, LLaMA), demonstrating consistent gains and improved training stability over GRPO-family baselines.

2 Preliminaries

2.1 Proximal Policy Optimization

Proximal Policy Optimization (PPO) (Schulman et al., 2017) is a widely adopted policy gradient method in reinforcement learning (RL) (Kaelbling et al., 1996). PPO enables stable training by using IS for off-policy updates while constraining the deviation between the current and old policies. The standard unclipped PPO objective is defined as:

$$\mathcal{J}_{\text{PG}}(\theta) = \mathbb{E}_{q\sim\mathcal{D},\, o\sim\pi_{\theta_{\text{old}}}(\cdot|q)}\left[r_t(\theta)A_t\right] \tag{1}$$

$$\text{where} \quad r_t(\theta) = \frac{\pi_\theta(o_t \mid q, o_{<t})}{\pi_{\theta_{\text{old}}}(o_t \mid q, o_{<t})} \tag{2}$$

Here, $\pi_{\theta_{\text{old}}}$ and π_θ denote the old and current policies, respectively; A_t is the advantage; q is a query sampled from the data distribution $\mathcal{D}$; and o is a response generated by the old policy. Directly optimizing this objective can yield large fluctuations in the IS ratios $r_t(\theta)$ and thus training instability. To mitigate this, PPO introduces a clipped surrogate objective that imposes a local trust region:

$$\mathcal{J}_{\text{PPO}}(\theta) = \mathbb{E}_{q\sim\mathcal{D},\, o\sim\pi_{\theta_{\text{old}}}(\cdot|q)}\Big[\min\big(r_t(\theta)A_t, \operatorname{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\,A_t\big)\Big] \tag{3}$$

Here, ϵ is a hyperparameter defining the clipping bounds. This mechanism dampens the incentive to update when the IS ratio deviates substantially, thereby enhancing stability. However, it effectively zeroes out the gradient for clipped tokens, which

may allow the policy to drift beyond the intended trust region without correction.

2.2 Group Relative Policy Optimization

Group Relative Policy Optimization (GRPO) (Guo et al., 2025) removes the need for a value model by adopting group-based advantage estimation. Following Search-R1 (Jin et al., 2025), external retrieval tokens are masked during search-agent training, so that the GRPO objective is optimized exclusively over agent-generated tokens:

$$\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}_{q\sim\mathcal{D},\{o_i\}_{i=1}^{G}\sim\pi_{\theta_{\text{old}}}(\cdot|q)}\Bigg[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{t=1}^{|o_i|}\min\Big(r_{i,t}(\theta)\,\hat{A}_{i,t},\,\text{clip}\big(r_{i,t}(\theta),\,1-\epsilon,\,1+\epsilon\big)\,\hat{A}_{i,t}\Big)\Bigg] \tag{4}$$

where the IS ratio $r_{i,t}(\theta)$ and the group-normalized advantage $\hat{A}_{i,t}$ are defined as:

$$r_{i,t}(\theta) = \frac{\pi_\theta(o_{i,t} \mid q, o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t} \mid q, o_{i,<t})} \tag{5}$$

$$\hat{A}_{i,t} = \hat{A}_i = \frac{R_i - \text{mean}(\{R_j\}_{j=1}^{G})}{\text{std}(\{R_j\}_{j=1}^{G})} \tag{6}$$

The advantage is computed by normalizing the sequence-level reward R_i against the group statistics. Note that GRPO assigns an identical advantage value to all tokens within the same response.

3 Approach

3.1 Search Agent Formulation

Let q be a user question, π_θ the LLM policy, and $\mathcal{R}(\cdot, \mathcal{D})$ a retriever over an external corpus $\mathcal{D}$. The standard retrieve-then-generate RAG pipeline produces $a = \pi_\theta(\cdot \mid q, \mathcal{R}(q, \mathcal{D}))$, which is often insufficient for complex queries requiring iterative evidence gathering. Search agents instead interleave retrieval with reasoning over multiple turns:

$$a = \pi_\theta\Big(\cdot \mid q, \{z_i, k_i\}_{i=1}^{T}\Big), \quad k_i = \mathcal{R}(sq_i \mid \mathcal{D}),\ sq_i \in z_i \tag{7}$$

Here, the policy output o comprises intermediate reasoning steps z and the final answer a; sq_i is a search query generated within reasoning step z_i to retrieve intermediate knowledge k_i; the sequence $\{z_i, k_i\}$ represents the interleaved trajectory of reasoning and retrieval; and T denotes the number of search actions.

3.2 Importance Sampling Distribution Drift

In search agents, effective actions are sparse: only a small subset of queries returns useful evidence. When π_θ rapidly diverges from $\pi_{\theta_{\text{old}}}$, sampled actions may receive negligible probability under π_θ, forcing $r_t(\theta) \to 0$. We do not claim ISDD is the sole driver of GRPO instability—credit-assignment coarseness (Section 1) and entropy collapse (Limitations) also contribute—only that it is the dominant observable mode in our setting (Figure 1a–b) and the one our one-line conditional KL is designed to intercept (Section 4.3); the analysis below uses standard assumptions (bounded log-ratios, finite horizon). Quantitatively, in Figure 10 (panels (1),(5); Appendix A.13) GRPO's IS ratio destabilizes at steps 75–140 but reward only crashes after step ∼140—a ∼50–60-step lag consistent with ISDD being a leading indicator rather than a co-symptom; SAPO shows no instability on either trace. The structural conditions of ISDD apply to multi-turn agentic RL more broadly, but all empirical evidence here is on single-tool retrieval QA, our primary testbed (extensions discussed in Limitations).

Definition 1 (ISDD)**.** Building on the IS ratios defined in Eq. 5, we define Importance Sampling Distribution Drift (ISDD) as the event in which

$$\mathbb{P}\left(\prod_{t=1}^{T} r_t(\theta) < \epsilon\right) > \phi, \tag{8}$$

for some small $\epsilon > 0$ and probability threshold ϕ. This condition indicates that a non-trivial fraction of trajectories suffer from vanishing cumulative importance weights.

Why ISDD causes collapse. In GRPO, the policy gradient is weighted by the IS ratios:

$$\nabla_\theta \mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}_{q\sim\mathcal{D},\{o_i\}_{i=1}^{G}\sim\pi_{\theta_{\text{old}}}(\cdot|q)}\Bigg[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{t=1}^{|o_i|}\hat{A}_{i,t} r_{i,t}(\theta)\nabla_\theta \log \pi_\theta(o_{i,t}|q, o_{i,<t})\Bigg] \tag{9}$$

When ISDD occurs ($r_t \to 0$), the gradients vanish regardless of the advantage values, effectively stalling the policy update. Even high-reward trajectories then contribute negligible gradients, which substantially limits the model's ability to learn from successful explorations.

Proposition 1 (ISDD Amplification in Interleaved Multi-Step Interactions)**.** *Assume the IS ratios*

r_t follow a log-normal distribution $\log r_t \sim \mathcal{N}(\mu, \sigma^2)$, with drift parameter $\lambda = \mu + \sigma^2/2$. Due to the low-entropy, bottleneck nature of tool selection, action tokens a_i tend to be more sensitive to policy shifts than reasoning tokens z_i, implying $\lambda_a \ll \lambda_z < 0$.

The expected cumulative importance weight along the trajectory then decomposes as

$$\mathbb{E}\left[\prod_{t=1}^{L} r_t\right] = \mathbb{E}\left[\prod_{t\in\mathcal{Z}} r_t\right] \cdot \mathbb{E}\left[\prod_{t\in\mathcal{A}} r_t\right] = \underbrace{\exp(L_z\lambda_z)}_{\textit{Reasoning Drift}} \cdot \underbrace{\exp(L_a\lambda_a)}_{\textit{Interaction Drift}}, \tag{10}$$

*where $L = L_z + L_a$ is the total number of tokens. ISDD is more severe in agentic tasks than in standard QA because the **Interaction Drift** term decays exponentially in the number of action tokens L_a, driven by the strongly negative drift parameter λ_a.*

3.3 Search Agent Policy Optimization

To mitigate ISDD and improve training stability, SAPO augments the GRPO objective with an auxiliary penalty that constrains the divergence between the current and old policies. The SAPO token-level objective is:

$$\begin{aligned}\mathcal{J}_{\text{SAPO}}(\theta) = &\mathbb{E}_{q\sim\mathcal{D},\{o_i\}_{i=1}^{G}\sim\pi_{\theta_{\text{old}}}(\cdot|q)}\Bigg[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{t=1}^{|o_i|} \\ &\min\Big(r_{i,t}(\theta)\hat{A}_{i,t}, \text{clip}\left(r_{i,t}(\theta), 1-\epsilon, 1+\epsilon\right)\hat{A}_{i,t}\Big) \\ &\underbrace{+\gamma \text{KL}\left[\pi_\theta \,\|\, \pi_{\theta_{\text{old}}}\right]}_{\text{Drift Penalty}}\Bigg]\end{aligned} \tag{11}$$

where γ is the penalty coefficient. Unlike the GRPO KL term, which constrains divergence against a fixed reference model, SAPO penalizes deviations from the dynamic old policy and adopts the log-ratio approximation:

$$\text{KL}\left[\pi_\theta \,\|\, \pi_{\theta_{\text{old}}}\right] = \log \frac{\pi_\theta(o_{i,t} \mid q, o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t} \mid q, o_{i,<t})} = \log r_{i,t}(\theta) \tag{12}$$

Applying this penalty unconditionally would over-constrain exploration. We therefore activate it only on tokens whose IS ratio has dropped below a threshold while the advantage is positive—precisely the regime in which ISDD suppresses useful gradient signal:

$$\text{KL_cond}\left[\pi_\theta \,\|\, \pi_{\theta_{\text{old}}}\right] = \mathbb{I}\Big(r_{i,t}(\theta) < \tau,\ \hat{A}_{i,t} > 0\Big) \log r_{i,t}(\theta). \tag{13}$$

Here, $\mathbb{I}(\cdot)$ is the indicator function and τ is the threshold for identifying drifting tokens. By restricting the KL penalty to this targeted subset, SAPO offers two benefits:

- **Mitigates over-suppression**: positive tokens with $r_t < \tau$ receive a gradient contribution that increases their probability under π_θ, pulling the policy back toward $\pi_{\theta_{\text{old}}}$ and counteracting ISDD.
- **Preserves exploration**: tokens with $r_t \geq \tau$ (within the trust region) or $\hat{A}_t \leq 0$ (which should remain suppressed) are not penalized; the constraint is therefore sparsely triggered by design—empirically activating on $\approx 10\%$ of response tokens (Appendix A.13)—allowing the policy to optimize freely in stable regions.

3.4 Gradient Analysis

Differentiating the conditional KL term via the product rule yields Eq. 14, where $\delta(\cdot)$ is the Dirac delta arising from $\nabla_\theta \mathbb{I}(\cdot)$. Substituting into the full objective gives the SAPO gradient in Eq. 15: the bracketed coefficient acts as an adaptive regularizer whose magnitude grows as $r_{i,t} \to 0$, delivering a stronger correction to tokens that have drifted further from the old policy.

4 Experiments

4.1 Experimental Settings

Datasets. We evaluate our method on seven QA benchmarks, categorized into single-hop and multi-hop retrieval tasks. The single-hop datasets include Natural Questions (NQ) (Kwiatkowski et al., 2019), TriviaQA (Joshi et al., 2017), and PopQA (Mallen et al., 2022). The multi-hop datasets are HotpotQA (Yang et al., 2018), 2WikiMultihopQA (Ho et al., 2020), Musique (Trivedi et al., 2022b), and Bamboogle (Press et al., 2022). We use Exact Match (EM) accuracy as the evaluation metric across all datasets. Following Search-R1, SAPO is trained on a composite dataset consisting of NQ and HotpotQA.

Baselines. We compare SAPO against three categories of methods: (1) **Retrieval-free methods**: direct LLM generation, Supervised Fine-Tuning (SFT), and R1-style training (Guo et al., 2025) without retrieval; (2) **Retrieval-enhanced workflows**: Naive RAG, which retrieves documents based solely on the input question, and

$$
\begin{aligned}
\nabla_\theta \mathcal{J}_{\text{KL_cond}}(\theta) &= \nabla_\theta \mathbb{I}(r_{i,t}(\theta) < \tau, \hat{A}_{i,t} > 0) \log r_{i,t}(\theta) \\
&= \mathbb{I}(\hat{A}_{i,t} > 0)[\mathbb{I}(r_{i,t}(\theta) < \tau) \nabla_\theta \log r_{i,t}(\theta) + \nabla_\theta \mathbb{I}(r_{i,t}(\theta) < \tau) \log r_{i,t}(\theta)] \\
&= [\mathbb{I}(r_{i,t}(\theta) < \tau, \hat{A}_{i,t} > 0) - \delta(r_{i,t}(\theta) - \tau) r_{i,t}(\theta) \log r_{i,t}(\theta)] \nabla_\theta \log \pi_\theta(o_{i,t}|q, o_{i,<t})
\end{aligned} \tag{14}
$$

$$
\begin{aligned}
\nabla_\theta \mathcal{J}_{\text{SAPO}}(\theta) &= \mathbb{E}_{q,\{o_i\}}\Bigg[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{t=1}^{|o_i|}\Big(\hat{A}_{i,t}\, r_{i,t}(\theta) + \gamma \underbrace{C_{i,t}(\theta)}_{\text{Drift Penalty Coefficient}}\Big) \nabla_\theta \log \pi_\theta(o_{i,t} \mid q, o_{i,<t})\Bigg], \\
C_{i,t}(\theta) &= \mathbb{I}(r_{i,t}(\theta) < \tau,\, \hat{A}_{i,t} > 0) - \delta(r_{i,t}(\theta) - \tau)\, r_{i,t}(\theta) \log r_{i,t}(\theta).
\end{aligned} \tag{15}
$$

IRCoT (Trivedi et al., 2022a), which interleaves retrieval with Chain-of-Thought; (3) **Retrieval-augmented agents**: recent agentic methods including Search-o1 (Li et al., 2025c), Search-R1 (Jin et al., 2025), ReSearch (Chen et al., 2025b), ZeroSearch (Sun et al., 2025), StepSearch (Wang et al., 2025b), EXSEARCH (Shi et al., 2025b), O^2-Searcher (Mei et al., 2025), InForage (Qian and Liu, 2025), CriticSearch (Zhang et al., 2025b), AutoRefine (Shi et al., 2025a), and SE-Search (Li et al., 2026).

Implementation Details. We use the external search corpus from (Karpukhin et al., 2020) with E5-base-v2 (Wang et al., 2022) as the retrieval engine. Unless otherwise specified, the retriever returns the top-3 documents per query, and the maximum number of search turns is set to $T_{\max} = 5$. We adopt the Qwen2.5 series (Qwen et al., 2025) as the backbone language model. In the RL phase, we use a simple rule-based, outcome-oriented reward computed as the EM or F1 score between the predicted and ground-truth answers. The KL penalty coefficient is set to $\gamma = 0.1$ and the IS-ratio threshold to $\tau = 1.0$.

4.2 Main Performance

Table 1 reports the main results. SAPO, Search-R1, and the closely related agentic baselines are all evaluated under identical Qwen2.5-3B settings with the same retriever and corpus, so cross-method comparisons are controlled.

SAPO achieves the highest overall performance. SAPO matches or exceeds the strongest prior agentic baselines (CriticSearch, AutoRefine, SE-Search) on six of the seven benchmarks and attains the best overall average (0.442 EM); SE-Search is marginally ahead on NQ alone (0.475 vs. 0.469). Against the controlled Search-R1-Instruct baseline (0.336 avg. EM), this corresponds to a +10.6-point absolute gain (+31.5% relative) under the same retriever and corpus.

SAPO shows notable gains on multi-hop QA. The gains concentrate on multi-hop tasks. Against CriticSearch, SAPO improves HotpotQA by 4.2 points (+10.1% relative) and Bamboogle by 6.4 points (+17.4% relative); averaged across the four multi-hop benchmarks, SAPO delivers +5.3 points (+15.5% relative) over CriticSearch and +14.7 points (+59.0% relative) over the Search-R1-Instruct baseline. We read this as evidence that SAPO is a stabilizer rather than a redesign of credit assignment: the gap grows with trajectory length, indicating that controlling IS-ratio drift—rather than peak EM—is the operative property. A head-to-head comparison with other GRPO-family stabilizers (DAPO, GSPO, GMPO) is presented in Section 4.3 (Table 2; per-dataset numbers in Appendix A.10), and a 3-seed robustness study is reported in Appendix A.11.

4.3 Ablation Studies

We perform ablation studies to assess the contribution of each component in SAPO. To ensure a fair comparison, all models are trained for the same 300 steps with 5 rollout responses. The baseline Search-R1 uses Qwen2.5-3B-Instruct as the backbone and the F1 score as the outcome-based reward. Table 3 reports accuracy across the seven benchmarks. We compare four configurations: (1) **GRPO**: the baseline algorithm used in Search-R1; (2) **GRPO w/ KL**: GRPO augmented with an unconditional KL penalty; (3) **GRPO w/ KL_r**: GRPO with a KL penalty conditioned solely on the IS ratio; (4) **GRPO w/ KL_ra (SAPO)**: the full SAPO configuration, in which the KL penalty is conditioned on both the IS ratio and the advantage.

Impact of selective KL penalties. Table 3 traces the contribution of each component. Starting from the Search-R1-GRPO baseline (0.388 avg.), an unconditional KL term lifts Musique by 30.6% and HotpotQA by 14.4% in relative terms, raising the average to 0.398. The ratio-conditioned

Methods	Single-Hop QA				Multi-Hop QA					Avg.
	NQ	TriviaQA	PopQA	*Avg.*	HotpotQA	2Wiki	Musique	Bamboogle	*Avg.*	
w/o Retrieval										
Direct Generation	0.106	0.288	0.108	0.167	0.149	0.244	0.020	0.024	0.109	0.134
SFT	0.249	0.292	0.104	0.215	0.186	0.248	0.044	0.112	0.148	0.176
R1-Instruct (Guo et al., 2025)	0.210	0.449	0.171	0.277	0.208	0.275	0.060	0.192	0.184	0.224
R1-Base (Guo et al., 2025)	0.226	0.455	0.173	0.285	0.201	0.268	0.055	0.224	0.187	0.229
Workflow w/ Retrieval										
Naive RAG (Lewis et al., 2020)	0.348	0.544	0.387	0.426	0.255	0.226	0.047	0.080	0.152	0.270
IRCoT (Trivedi et al., 2022a)	0.111	0.312	0.200	0.208	0.164	0.171	0.067	0.240	0.161	0.181
Agent w/ Retrieval										
Search-o1 (Li et al., 2025c)	0.238	0.472	0.262	0.324	0.221	0.218	0.054	0.320	0.203	0.255
Search-R1-Instruct (Jin et al., 2025)	0.397	0.565	0.391	0.451	0.331	0.310	0.124	0.232	0.249	0.336
Search-R1-Base (Jin et al., 2025)	0.421	0.583	0.413	0.472	0.297	0.274	0.066	0.128	0.191	0.312
ReSearch-Instruct (Chen et al., 2025b)	0.365	0.571	0.395	0.444	0.351	0.272	0.095	0.266	0.246	0.331
ReSearch-Base (Chen et al., 2025b)	0.427	0.597	0.430	0.485	0.305	0.272	0.074	0.128	0.195	0.319
ZeroSearch-Base (Sun et al., 2025)	0.430	0.616	0.414	0.487	0.338	0.346	0.130	0.139	0.238	0.345
StepSearch-Base (Wang et al., 2025b)	–	–	–	–	0.329	0.339	0.181	0.328	0.294	–
EXSEARCH-Base (Shi et al., 2025b)	0.368	–	–	–	0.422	0.372	0.138	–	–	–
O^2-Searcher (Mei et al., 2025)	0.444	0.597	0.429	0.490	0.388	0.374	0.160	0.344	0.317	0.391
AutoRefine-Instruct (Shi et al., 2025a)	0.436	0.597	0.447	0.493	0.404	0.380	0.169	0.336	0.322	0.396
AutoRefine-Base (Shi et al., 2025a)	0.467	0.620	0.450	0.512	0.405	0.393	0.157	0.344	0.325	0.405
InForage (Qian and Liu, 2025)	0.421	0.597	0.452	0.490	0.409	0.428	0.172	0.360	0.342	0.405
CriticSearch (Zhang et al., 2025b)	–	–	–	–	0.414	0.409	0.180	0.368	0.343	–
SE-Search (Li et al., 2026)	**0.475**	0.624	0.423	0.507	0.450	0.361	0.183	0.424	0.355	0.420
SAPO-3B-Base (Ours)	0.474	**0.630**	0.449	0.517	0.449	0.412	0.196	0.424	0.370	0.433
SAPO-3B-Instruct (Ours)	0.469†	0.629†	**0.457***	**0.518**	**0.456**†	**0.448**†	**0.203**†	**0.432**†	**0.396**	**0.442**†

Table 1: Accuracy of SAPO-3B vs. baselines on Qwen2.5-3B (Qwen et al., 2025). **Bold**/underline: best/second-best. Single-seed for parity with prior work; 3-seed study (Appendix A.11) confirms the gap is structural. On the SAPO-3B-Instruct row, †/∗ mark paired-bootstrap gaps over Search-R1-Instruct passing Bonferroni (α=0.05/7) / uncorrected (α=0.05) tests (Table 11).

variant (KL_r) reaches 0.417 (HotpotQA 0.457), and the full SAPO configuration (KL_ra) wins six of seven benchmarks at 0.429—a +4.1-point (+10.6% relative) absolute gain over the baseline.

Comparison with GRPO-family stabilizers. To position SAPO against other algorithm-side stabilizers in the GRPO family, we compare head-to-head with DAPO (Yu et al., 2025), GSPO (Zheng et al., 2025), and GMPO (Zhao et al., 2025) under the same Qwen2.5-3B-Instruct + Search-R1 pipeline (Table 2). SAPO attains the highest average EM and F1, wins six of seven datasets on both metrics (per-dataset numbers in Appendix A.10), and is uniquely both collapse-free and still improving at step 340—DAPO plateaus by step 100, while GSPO and GMPO collapse at steps 260 and 220, respectively. SAPO's gain over the strongest baseline (GSPO) is asymmetric (+1.0 EM on single-hop vs. +2.1 on multi-hop), and the gap over GRPO widens from +9.8 to +11.5. The F1 ordering mirrors EM (SAPO 0.523 > GSPO 0.499 > GMPO 0.484 > DAPO 0.446 > GRPO 0.403), so the gain is not driven by exact-match formatting alone. The +6.9 EM gap between the two collapse-free methods

Method	Avg. EM			F1	Best	Coll.
	SH	MH	All			
GRPO	0.436	0.250	0.330	0.403	20	80
DAPO	0.473	0.291	0.369	0.446	100	–
GSPO	0.524	0.344	0.421	0.499	260	260
GMPO	0.508	0.328	0.405	0.484	100	220
SAPO	**0.534**	**0.365**	**0.438**	**0.523**	**340**	**–**

Table 2: Head-to-head with GRPO-family stabilizers (Qwen2.5-3B-Instruct + Search-R1, rollout 5, EM-based outcome reward). Avg. EM is split into Single-Hop (**SH**), Multi-Hop (**MH**), and overall (**All**); Each score is taken at the method's own best step (peak Avg. EM). **Coll.** is the IS-ratio collapse step; "–" indicates no collapse within the 420-step budget. Per-dataset EM/F1 are in Appendix A.10.

(SAPO vs. DAPO) further indicates that DAPO's clip-higher (entropy collapse) and SAPO's conditional KL (IS-ratio collapse) target distinct failure modes; combining the two is a natural future work.

4.4 Detailed Analysis

Entropy and exploration phases. The entropy trace in Figure 1(c) is not monotonically declin-

Method	Single-Hop QA			Multi-Hop QA				Avg.
	NQ	TriviaQA	PopQA	HotpotQA	2Wiki	Musique	Bamboogle	
Search-R1-GRPO	0.402	0.662	0.436	0.374	0.376	0.134	0.333	0.388
+ KL	0.425 (↑0.023)	0.625 (↓0.037)	0.443 (↑0.007)	0.428 (↑0.054)	0.371 (↓0.005)	0.175 (↑0.041)	0.342 (↑0.009)	0.398 (↑0.010)
+ KL_r	0.439 (↑0.014)	0.660 (↑0.035)	0.456 (↑0.013)	**0.457** (↑0.029)	0.369 (↓0.002)	0.169 (↓0.006)	0.369 (↑0.027)	0.417 (↑0.019)
+ KL_{ra} (SAPO)	**0.442** (↑0.003)	**0.673** (↑0.013)	**0.474** (↑0.018)	0.433 (↓0.024)	**0.415** (↑0.046)	**0.181** (↑0.012)	**0.382** (↑0.013)	**0.429** (↑0.012)

Table 3: Ablation results on different KL terms in **SAPO**. The colored values in parentheses below each score denote the absolute change relative to the previous row (↑ improvement, ↓ decline). The shaded row is our SAPO.

Backbone	Search #	IS	IRR
Qwen2.5-7B INST	2.25 ↑0.09	0.344 ↑0.054	0.350 ↑0.052
Qwen2.5-3B INST	**2.72** ↑0.52	**0.425** ↑0.060	**0.486** ↑0.108
Qwen2.5-1.5B INST	1.88 ↑0.17	0.294 ↑0.053	0.296 ↑0.048
Qwen2.5-3B BASE	2.13 ↑0.12	0.364 ≈0.001	0.367 ≈0.001
Llama-3.2-3B INST	1.72 ↓0.05	0.276 ≈0.006	0.275 ≈0.006

Table 4: Search-behavior under SAPO, averaged over the seven QA benchmarks. Each cell shows the metric at step 300 (top) and $\Delta_{200\to300}$ (bottom). **Search #**: valid retrieval calls per query; **IS / IRR**: whether an answer-bearing document is retrieved and how early it surfaces. Shaded row: in-paper case study; the rule separates Qwen-Instruct (top) from Qwen-Base / Llama-Instruct controls (bottom). SH / MH breakdown in Appendix A.12.

Method	Single-Hop	Multi-Hop	Overall
LLaMA-3.2-3B-Base			
GRPO	0.476	0.176	0.305
SAPO	**0.518**	**0.252**	**0.366**
Δ	↑0.042	↑0.075	↑0.061
LLaMA-3.2-3B-Instruct			
GRPO	0.438	0.236	0.322
SAPO	**0.518**	**0.350**	**0.422**
Δ	↑0.081	↑0.115	↑0.100

Table 5: Generalization across the LLaMA-3.2 family: average EM broken down by single-hop and multi-hop splits. SAPO improves both Base and Instruct variants on every split, with the largest gains on multi-hop (+7.5 on Base, +11.5 on Instruct). Per-dataset results are in Appendix A.5.1 (Table 8).

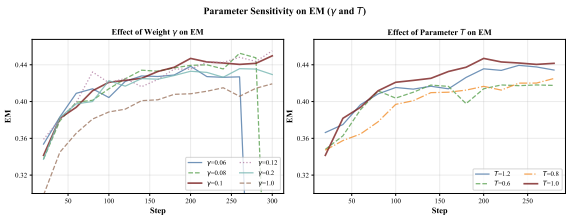

Figure 2: Hyperparameter sensitivity analysis: the KL penalty coefficient γ, and the IS ratio threshold τ.

ing; it exhibits three phases that align with the conditional-KL design. **Phase 1 (steps 0–80)**: a mild dip to ≈ 0.6–1.0 as the policy commits to early high-reward patterns. **Phase 2 (steps 80–140)**: under the stabilized IS ratio, entropy rises back to ≈ 2.0—the policy actively re-expands its exploration set, in contrast to GRPO, whose entropy is monotonically decreasing over the same window and ultimately collapses. **Phase 3 (steps 160 onward)**: smooth convergence-driven decay to ≈ 0.3, the signature of a well-trained agent at convergence rather than premature collapse. Because the conditional KL fires only on positive-advantage tokens whose IS ratio has drifted below τ (Eq. 13), the regularizer is sparsely triggered by design and cannot systematically suppress diversity; the Phase-2 re-expansion is the empirical signature of this property—by repairing IS drift early. Appendix A.13 (Figure 10) reports a complementary 8-panel view over the first 200 steps confirming that GRPO destabilizes at the IS-ratio level first and only later collapses on reward and search behavior, while SAPO is monotonic or near-monotonic on every panel.

Hyperparameter sensitivity. Figure 2 reports the validation EM as a function of the KL penalty γ and the IS-ratio threshold τ. Fixing γ=0.1, τ=1.0 yields the highest EM (0.442), outperforming both higher (τ=1.2: 0.434) and lower thresholds ($\tau \in \{0.6, 0.8\}$: 0.418/0.425), and the gap widens later in training. Fixing τ=1.0, $\gamma \in \{0.08, 0.10, 0.12\}$ reach essentially identical peak EM (within run-to-run noise), but only $\gamma \in \{0.10, 0.12\}$ retain this performance late in training; smaller values

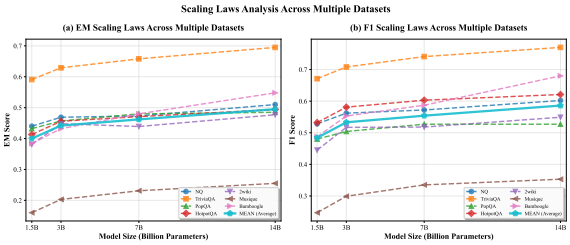

Figure 3: Visualization of the scaling trends in Table 7. Both EM and F1 scores rise monotonically with model size across the Qwen2.5-Instruct series (1.5B→14B), under a single, fixed $(\gamma, \tau) = (0.1, 1.0)$ configuration without per-model retuning.

are prone to late-stage instability, while $\gamma \geq 0.2$ over-regularizes. We adopt $\gamma = 0.1$ and $\tau = 1.0$ for simplicity; the same configuration transfers across 1.5B–14B (Appendix Table 7) and across Qwen and LLaMA-3.2 (Tables 1 and 5) without per-model tuning, indicating a broad basin rather than a knife-edge optimum.

Scaling with model size. We evaluate SAPO across the Qwen2.5-Instruct series from 1.5B to 14B parameters. Performance scales monotonically with model size: average EM rises from 0.400 (1.5B) to 0.442 (3B), 0.462 (7B), and 0.495 (14B), while F1 climbs from 0.484 to 0.586 over the same range. Per-benchmark numbers and scaling curves are shown in Figure 3 and Appendix Table 7.

Robustness across model families. We extend the evaluation to LLaMA-3.2 (Touvron et al., 2023), testing the Base and Instruct variants of the 3B model. As summarized in Table 5 (per-dataset breakdown in Appendix A.5.1), SAPO improves over GRPO on every variant and split: overall EM rises from 0.305 to 0.366 on Base (+6.1) and from 0.322 to 0.422 on Instruct (+10.0), with the gain again concentrated on multi-hop (+7.5 Base, +11.5 Instruct) over single-hop (+4.2, +8.1), mirroring the Qwen trend (Table 1). SAPO thus transfers as a model-agnostic stabilizer across architectures and alignment stages.

Efficiency and search behavior. On six efficiency metrics at step 100 (Qwen2.5-3B-Instruct, rollout 5), SAPO holds throughput within 2% of Search-R1 and AutoRefine and attains the lowest per-token generation latency (0.087 ms/token, an amortization effect of longer responses), at the cost of a 29–33% increase in update and testing time due to the conditional-KL computation and deeper trajectories. This overhead is offset by the highest search depth (1.92), which we view as the operative property for multi-hop reasoning (full per-method bars in Appendix A.7). To check that deeper search is also better search, we track two reward-external metrics across five backbones—Information Score (whether answer-bearing documents are retrieved) and Information Reverse Rank (how early they surface)—in Table 4, where $\Delta_{200\to300}$ entries are color-coded as ↑ improve, ↓ decline, ≈ near-zero ($|\Delta| \leq 0.01$).

Three patterns emerge. **(a) Deeper, not noisier:** valid search count grows on four of five backbones while finish ratio stays ≥ 0.94 (Appendix A.12). **(b) More accurate:** Information Score rises on every backbone—more retrievals translate into more answer-bearing documents rather than wasted queries. **(c) Earlier evidence:** on the headline Qwen2.5-3B-Instruct backbone, Information Reverse Rank grows faster than Information Score (+0.108 vs. +0.060), so useful documents surface in earlier turns rather than only eventually being uncovered; the larger 7B and smaller 1.5B backbones show comparable IRR/IS growth, indicating that the "earlier evidence" effect is most pronounced at the scale where IS-drift is also most actively corrected. Gains concentrate on multi-hop subsets, providing a mechanistic counterpart to the multi-hop EM gap reported in Section 4.2.

5 Conclusion

In this work, we identify Importance Sampling Distribution Drift (ISDD) as a previously under-recognized failure mode of GRPO, in which importance sampling ratios decay toward zero and ultimately precipitate model collapse. To address it, we propose SAPO (Search Agent Policy Optimization), which adds a conditional token-level KL penalty that fires only on positive-advantage tokens whose IS ratios have drifted below a threshold, repairing drift while keeping gradient flow elsewhere intact. SAPO requires only a single-line modification to the standard GRPO update. Across seven QA benchmarks, four model scales (1.5B–14B), and two model families, SAPO attains the highest overall average among the compared methods and remains stable on settings where GRPO collapses. We position SAPO as a stabilizer that is complementary to credit-assignment improvements (e.g., process reward models), and view conditional distributional constraints as a promising direction for stabilizing agentic RL more broadly.

Limitations

While SAPO shows consistent gains across the evaluated benchmarks, several limitations warrant discussion.

Task and tool generality. Following the standard TARL setup (Jin et al., 2025; Chen et al., 2025b; Shi et al., 2025a), our experiments focus on retrieval-augmented QA with a single search tool. SAPO's conditional KL penalty is defined at the token level and encodes no task- or tool-specific assumptions, so it should in principle extend to other agentic domains (e.g., math reasoning with calculators, code generation with interpreters, web navigation) and multi-tool trajectories. We leave broader cross-task validation—including non-QA agentic settings and multi-tool trajectories with denser action vocabularies, longer horizons, or heterogeneous tool noise—to future work, where (γ, τ) may need re-tuning.

Static corpus and retriever dependence. We evaluate on a fixed Wikipedia corpus with E5-base-v2 (Wang et al., 2022), mirroring the protocol of all compared baselines. This controlled setup enables reproducible comparison but excludes the challenges of live retrieval, including evolving collections, noisy or adversarial web content, and variable latency. Because SAPO operates on the policy's own token probabilities rather than on retrieval content, its mechanism is conceptually independent of corpus and retriever quality; we hypothesize that weaker retrievers would amplify the need for multi-turn refinement, making SAPO's stability gains more pronounced. Open-web evaluation and a sweep over retrievers of varying recall would substantiate this claim.

Credit assignment and reward design. A root cause of ISDD—shared across GRPO-based methods—is the assignment of a uniform advantage to all tokens within a response, which conflates contributory and non-contributory tokens. SAPO mitigates a downstream consequence (suppression of positive tokens) rather than resolving credit assignment at its source. Likewise, our outcome-based F1 reward provides no signal about intermediate reasoning quality. Combining SAPO with finer-grained reward signals—process reward models, advantage decomposition, or retrieval-relevance scoring—is a natural complementary direction.

Hyperparameters and interaction with clipping. SAPO introduces two hyperparameters, the KL coefficient γ and the IS-ratio threshold τ. Our sensitivity study (Figure 2) shows a broad performant basin, and the same $(\gamma{=}0.1, \tau{=}1.0)$ transfers across 1.5B–14B and Qwen/LLaMA without per-setting tuning. SAPO also operates alongside PPO clipping, and our ablations (Table 3) confirm that the KL term contributes incremental gains over clipping alone. Whether the KL penalty could replace clipping entirely, or whether the two components interact differently across regimes, remains an open question; adaptive schedules for (γ, τ) based on running IS-ratio statistics are a promising direction.

Scale, task complexity, and evaluation metrics. Our largest backbone (14B) is moderate by current standards, and our benchmarks involve short-span QA rather than longer-horizon tasks such as BrowseComp (Wei et al., 2025) or multi-document synthesis. The monotonic 1.5B→14B trend (Figure 3) is encouraging, but extension to 70B+ models and longer-horizon agentic tasks remains future work. We follow standard practice in reporting Exact Match and token-level F1, which do not capture semantic equivalence, and all benchmarks are English; since SAPO operates on token-level probability ratios, transfer to multilingual or domain-specific (biomedical, legal) settings is expected but unverified.

Theoretical modeling assumptions. Proposition 1 assumes log-normal, independent token-level IS ratios—standard simplifications in IS analysis (Schulman et al., 2017). The qualitative prediction that ISDD amplifies exponentially with interaction steps holds under weaker conditions (any sub-unity multiplicative drift), and our empirical observations (Figure 1) are consistent with this prediction. Real token distributions may exhibit heavier tails or inter-token dependencies; relaxing these assumptions would strengthen the analytical foundation.

References

Chenxin An, Zhihui Xie, Xiaonan Li, Lei Li, Jun Zhang, Shansan Gong, Ming Zhong, Jingjing Xu, Xipeng Qiu, Mingxuan Wang, and Lingpeng Kong. 2025. Polaris: A post-training recipe for scaling reinforcement learning on advanced reasoning models.

Aili Chen, Aonian Li, Bangwei Gong, Binyang Jiang, Bo Fei, Bo Yang, Boji Shan, Changqing Yu, Chao Wang, Cheng Zhu, and 1 others. 2025a. Minimax-m1: Scaling test-time compute efficiently with lightning attention. arXiv preprint arXiv:2506.13585.

Mingyang Chen, Linzhuang Sun, Tianpeng Li, Haoze Sun, Yijie Zhou, Chenzheng Zhu, Haofen Wang, Jeff Z Pan, Wen Zhang, Huajun Chen, and 1 others. 2025b. Learning to reason with search for llms via reinforcement learning. arXiv preprint arXiv:2503.19470.

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

Wenlong Deng, Yi Ren, Muchen Li, Danica J Sutherland, Xiaoxiao Li, and Christos Thrampoulidis. 2025b. On the effect of negative gradient in group relative deep reinforcement optimization. arXiv preprint arXiv:2505.18830.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. arXiv preprint arXiv:2501.12948.

Taicheng Guo, Xiuying Chen, Yaqi Wang, Ruidi Chang, Shichao Pei, Nitesh V Chawla, Olaf Wiest, and Xiangliang Zhang. 2024. Large language model based multi-agents: A survey of progress and challenges. arXiv preprint arXiv:2402.01680.

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

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

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

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

Leslie Pack Kaelbling, Michael L Littman, and Andrew W Moore. 1996. Reinforcement learning: A survey. Journal of artificial intelligence research, 4:237–285.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. arXiv preprint arXiv:2001.08361.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick SH Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In EMNLP (1), pages 6769–6781.

Mei Kobayashi and Koichi Takeda. 2000. Information retrieval on the web. ACM computing surveys (CSUR), 32(2):144–173.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, and 1 others. 2019. Natural questions: a benchmark for question answering research. Transactions of the Association for Computational Linguistics, 7:453–466.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. Advances in neural information processing systems, 33:9459–9474.

Jian Li, Yizhang Jin, Dongqi Liu, Hang Ding, Jiafu Wu, Dongsheng Chen, Yunhang Shen, Yulei Qin, Ying Tai, Chengjie Wang, and 1 others. 2026. Se-search: Self-evolving search agent via memory and dense reward. arXiv preprint arXiv:2603.03293.

Jian Li, Xiaoxi Li, Yan Zheng, Yizhang Jin, Shuo Wang, Jiafu Wu, Yabiao Wang, Chengjie Wang, and Xiaotong Yuan. 2025a. A survey on ai search with large language models.

Kuan Li, Zhongwang Zhang, Huifeng Yin, Liwen Zhang, Litu Ou, Jialong Wu, Wenbiao Yin, Baixuan Li, Zhengwei Tao, Xinyu Wang, and 1 others. 2025b. Websailor: Navigating super-human reasoning for web agent. arXiv preprint arXiv:2507.02592.

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

Xiaoxi Li, Jiajie Jin, Guanting Dong, Hongjin Qian, Yongkang Wu, Ji-Rong Wen, Yutao Zhu, and Zhicheng Dou. 2025d. Webthinker: Empowering large reasoning models with deep research capability. arXiv preprint arXiv:2504.21776.

Dongqi Liu, Hang Ding, Qiming Feng, Jian Li, Xurong Xie, Zhucun Xue, Chengjie Wang, Jiangning Zhang, and Yabiao Wang. 2026. Disco-rag: Discourse-aware retrieval-augmented generation. arXiv preprint arXiv:2601.04377.

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

Jianbiao Mei, Tao Hu, Daocheng Fu, Licheng Wen, Xuemeng Yang, Rong Wu, Pinlong Cai, Xinyu Cai, Xing Gao, Yu Yang, and 1 others. 2025. o^2-searcher: A searching-based agent model for open-domain open-ended question answering. arXiv preprint arXiv:2505.16582.

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

Hongjin Qian and Zheng Liu. 2025. Scent of knowledge: Optimizing search-enhanced reasoning with information foraging. arXiv preprint arXiv:2505.09316.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, and 25 others. 2025. Qwen2.5 technical report. Preprint, arXiv:2412.15115.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. Advances in neural information processing systems, 36:53728–53741.

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

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. arXiv preprint arXiv:2402.03300.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. 2025. Hybridflow: A flexible and efficient rlhf framework. In Proceedings of the Twentieth European Conference on Computer Systems, pages 1279–1297.

Yaorui Shi, Shihan Li, Chang Wu, Zhiyuan Liu, Junfeng Fang, Hengxing Cai, An Zhang, and Xiang Wang. 2025a. Search and refine during think: Autonomous retrieval-augmented reasoning of llms. arXiv e-prints, pages arXiv–2505.

Zhengliang Shi, Lingyong Yan, Dawei Yin, Suzan Verberne, Maarten de Rijke, and Zhaochun Ren. 2025b. Iterative self-incentivization empowers large language models as agentic searchers. arXiv preprint arXiv:2505.20128.

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

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, and 1 others. 2023. Llama: Open and efficient foundation language models. arXiv preprint arXiv:2302.13971.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022a. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. arXiv preprint arXiv:2212.10509.

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

Jiakang Wang, Runze Liu, Fuzheng Zhang, Xiu Li, and Guorui Zhou. 2025a. Stabilizing knowledge, promoting reasoning: Dual-token constraints for rlvr. arXiv preprint arXiv:2507.15778.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. Text embeddings by weakly-supervised contrastive pre-training. arXiv preprint arXiv:2212.03533.

Xiaoxuan Wang, Han Zhang, Haixin Wang, Yidan Shi, Ruoyan Li, Kaiqiao Han, Chenyi Tong, Haoran Deng, Renliang Sun, Alexander Taylor, and 1 others. 2026. Arlarena: A unified framework for stable agentic reinforcement learning. arXiv preprint arXiv:2602.21534.

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

Jason Wei, Zhiqing Sun, Spencer Papay, Scott McKinney, Jeffrey Han, Isa Fulford, Hyung Won Chung, Alex Tachard Passos, William Fedus, and Amelia Glaese. 2025. Browsecomp: A simple yet challenging benchmark for browsing agents. arXiv preprint arXiv:2504.12516.

Jialong Wu, Baixuan Li, Runnan Fang, Wenbiao Yin, Liwen Zhang, Zhengwei Tao, Dingchu Zhang, Zekun Xi, Gang Fu, Yong Jiang, and 1 others. 2025. Webdancer: Towards autonomous information seeking agency. arXiv preprint arXiv:2505.22648.

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

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, and 1 others. 2025. Dapo: An open-source llm reinforcement learning system at scale. arXiv preprint arXiv:2503.14476.

Weihao Zeng, Yuzhen Huang, Qian Liu, Wei Liu, Keqing He, Zejun Ma, and Junxian He. 2025. Simplerl-zoo: Investigating and taming zero reinforcement learning for open base models in the wild. arXiv preprint arXiv:2503.18892.

Kaiyan Zhang, Yuxin Zuo, Bingxiang He, Youbang Sun, Runze Liu, Che Jiang, Yuchen Fan, Kai Tian, Guoli Jia, Pengfei Li, Yu Fu, Xingtai Lv, Yuchen Zhang, Sihang Zeng, Shang Qu, Haozhan Li, Shijie Wang, Yuru Wang, Xinwei Long, and 20 others. 2025a. A survey of reinforcement learning for large reasoning models. arXiv preprint arXiv:2509.08827.

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

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, and 1 others. 2023. A survey of large language models. arXiv preprint arXiv:2303.18223, 1(2).

Yuzhong Zhao, Yue Liu, Junpeng Liu, Jingye Chen, Xun Wu, Yaru Hao, Tengchao Lv, Shaohan Huang, Lei Cui, Qixiang Ye, and 1 others. 2025. Geometric-mean policy optimization. arXiv preprint arXiv:2507.20673.

Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, and 1 others. 2025. Group sequence policy optimization. arXiv preprint arXiv:2507.18071.

A Appendix

A.1 Related Work

Retrieval-Augmented Generation. Large Language Models (LLMs) (Zhao et al., 2023) have demonstrated strong capabilities in natural language understanding, reasoning, and information synthesis. However, they lack inherent access to real-time external knowledge and remain prone to hallucinations and factual errors. Retrieval-Augmented Generation (RAG) (Lewis et al., 2020; Liu et al., 2026) addresses these limitations by integrating LLMs with retrieval engines, conditioning generation on retrieved external passages to mitigate inaccuracies. Despite their utility, static RAG pipelines typically follow a "retrieve-then-generate" paradigm that restricts the LLM's autonomy in deciding *when* and *what* to retrieve. This constraint limits adaptability in settings where iterative information seeking is required.

Search Agents. Traditional search engines (e.g., Google, Bing) provide efficient access to ranked web pages but can struggle to capture nuanced user intent in complex queries (Kobayashi and Takeda, 2000). Search agents (Li et al., 2025a, 2026) represent an evolution beyond static RAG by invoking search tools within a multi-step reasoning process, enabling autonomous query refinement and iterative retrieval. Systems such as Search-o1 (Li et al., 2025c), WebThinker (Li et al., 2025d), and WebDancer (Wu et al., 2025) exemplify this capability. In parallel, a line of work investigates training strategies to optimize search behaviors, including Search-R1 (Jin et al., 2025), ReSearch (Chen et al., 2025b), AutoRefine (Shi et al., 2025a), SE-Search (Li et al., 2026), and WebSailor (Li et al., 2025b). A common limitation of these methods is post-training instability, which can manifest as model collapse or substantial performance degradation. To address this issue, we propose SAPO, which enhances training stability through constrained policy updates.

Tool-based Agentic Reinforcement Learning. Reinforcement Learning (RL) is a widely used framework for aligning LLM behaviors with human intent. Prominent approaches include Proximal Policy Optimization (PPO) (Schulman et al., 2017), Direct Preference Optimization (DPO) (Rafailov et al., 2023), and Group Relative Policy Optimization (GRPO) (Shao et al., 2024). GRPO is notable for its use of group-based nor-

malization, which removes the need for a separate value network. Extensions such as Dynamic Sampling Policy Optimization (DAPO) (Yu et al., 2025) and Group Sequence Policy Optimization (GSPO) (Zheng et al., 2025) further refine this paradigm, alongside other group-based optimization methods (Zeng et al., 2025; He et al., 2025; Chen et al., 2025a; An et al., 2025; Wang et al., 2025a; Zhang et al., 2025a). However, applying RL to tool-augmented agentic environments remains challenging due to the interactive, multi-turn nature of such tasks (Wang et al., 2026; Deng et al., 2025a,b). Within this family, SAPO is positioned as a *stabilizer* that targets IS-ratio drift via a token-level conditional KL penalty, complementary to credit-assignment refinements (e.g., process reward models) that operate on the upstream advantage signal.

A.2 Training and Evaluation Datasets

Consistent with Search-R1 and other RL-based search agents, we construct our training dataset by aggregating NQ (Kwiatkowski et al., 2019) and HotpotQA (Yang et al., 2018), yielding a total of 169,615 samples. For evaluation, we construct a comprehensive benchmark of 51,713 samples, incorporating the test sets of four datasets (NQ (Kwiatkowski et al., 2019), TriviaQA (Joshi et al., 2017), PopQA (Mallen et al., 2022), Bamboogle (Press et al., 2022)) and the development sets of three multi-hop datasets (HotpotQA (Yang et al., 2018), 2WikiMultihopQA (Ho et al., 2020), and Musique (Trivedi et al., 2022b)).

A.3 Search Agent Prompt

Figure 4 illustrates the prompt template used for the SAPO search agent. The template instructs the assistant to perform multi-turn search-driven reasoning by alternating between internal thought processes in `<think> </think>` and query generation in `<search> </search>`. Retrieved results are returned as up to three documents within `<documents> </documents>`. The design permits multiple search calls and directs the agent to consolidate evidence into a concise final response enclosed by `<answer> </answer>`.

A.4 SAPO Hyperparameters

Our implementation builds on the VeRL framework (Sheng et al., 2025). To ensure a fair comparison, we align the foundational configurations—including Data, Actor, and Rollout settings—with prior baselines such as Search-R1 and SE-Search. As detailed in Table 6, the model is trained with a global batch size of 256 and a learning rate of 1×10^{-6}, using a maximum response length of 2048 tokens. During the rollout phase, we employ a group size of $G = 10$ with sampling parameters set to temperature 1.0 and top-$p = 0.95$. Specifically for the proposed SAPO mechanism, we introduce a KL penalty coefficient $\gamma = 0.1$ and an IS-ratio threshold $\tau = 1.0$ to regulate policy updates.

Module	Hyper-parameter	Value
Data		
	Max documents length	512
	Max response length	2048
	Total training steps	420
	Retriever top-k	3
Actor		
	Training batch size	256
	Micro batch size	64
	Learning rate	1×10^{-6}
	Clip ratio ϵ	0.2
Rollout		
	Max search actions	5
	Group size G	10
	Temperature	1.0
	Top-p	0.95
KL term (SAPO-specific)		
	KL penalty coeff. γ	0.1
	IS ratio threshold τ	1.0

Table 6: Hyperparameters used in SAPO.

A.5 Generalization and Scalability

We investigate the scalability of SAPO by evaluating its performance across backbone models of varying capacities, specifically the Qwen2.5-Instruct series ranging from 1.5B to 14B parameters. To isolate the impact of model scale, we keep the retriever, knowledge corpus, and evaluation protocols constant. Table 7 reports the Exact Match (EM) and F1 scores. The results show a positive correlation between model size and performance, consistent with established scaling laws (Kaplan et al., 2020).

A.5.1 Per-Dataset LLaMA-3.2 Results

Table 8 provides the per-benchmark breakdown of the LLaMA-3.2-3B (Base/Instruct) generalization

You are a helpful assistant that excels at answering questions with multi-turn search engine calling. To answer questions, you must first reason through the available information using `<think>` and `</think>`. If you identify missing knowledge, you may issue a search request using `<search>` query `</search>` at any time. The retrieval system will provide you with the three most relevant documents enclosed in `<documents>` and `</documents>`. You may send multiple search requests if needed. Once you have sufficient information, provide a concise final answer using `<answer>` and `</answer>`. For example, `<answer>` Donald Trump `</answer>`. Question: {**question**}

Figure 4: Prompt template for SAPO.

Size	Metric	Single-Hop QA			Multi-Hop QA				Avg.
		NQ	TriviaQA	PopQA	HotpotQA	2Wiki	Musique	Bamboogle	
1.5B	EM	0.440	0.591	0.432	0.413	0.381	0.160	0.384	0.400
	F1	0.528	0.671	0.480	0.532	0.445	0.247	0.487	0.484
3B	EM	0.469	0.629	0.457	0.456	0.448	0.203	0.432	0.442
	F1	0.562	0.708	0.504	0.581	0.517	0.299	0.553	0.533
7B	EM	0.473	0.658	0.480	0.471	0.439	0.231	0.480	0.462
	F1	0.572	0.741	0.527	0.603	0.518	0.335	0.587	0.554
14B	EM	**0.510**	**0.695**	**0.485**	**0.494**	**0.477**	**0.255**	**0.548**	**0.495**
	F1	**0.602**	**0.770**	**0.527**	**0.621**	**0.549**	**0.353**	**0.680**	**0.586**

Table 7: Scaling trends of Qwen2.5-Instruct with **SAPO** across model sizes (1.5B→14B). We report Exact Match (EM) and F1; column-wise best in **bold**, best size shaded. Performance increases monotonically with scale, consistent with established scaling laws.

study summarized in Table 5 (main text). SAPO improves every dataset on both variants, with the largest absolute gains on multi-hop benchmarks (Bamboogle: $+14.3$ on Base, 2Wiki: $+15.3$ on Instruct), echoing the multi-hop concentration observed on the Qwen backbone.

A.6 Implementation Simplicity

We augment the standard GRPO surrogate objective with a concise penalty term that enforces token-level constraints on distributional shifts between the current and old policies. As shown in Figure 5, the core contribution of SAPO is encapsulated in the `kl_term_loss` expression. This term yields a scalar, token-level divergence penalty that is added to the global loss to limit per-token policy drift. By operating at token granularity and reducing to a single auxiliary KL expression on top of the standard GRPO loss, the modification provides a lightweight regularizer that complements the existing PPO clipping mechanism while enabling finer-grained control over distributional changes during policy updates.

A.7 Efficiency Analysis

To quantify the computational overhead introduced by SAPO, we first compare SAPO with Search-R1 and AutoRefine on six per-method efficiency metrics at training step 100 (Qwen2.5-3B-Instruct, rollout 5; Figure 6). We then track throughput, generation time per token, testing time, and update-actor time across five LLMs over the course of training (Figure 7).

Per-method overhead (Figure 6). (1) *Throughput*: within 2% of the baselines. (2) *Update and testing time*: $+29$–33%, attributable to (a) the conditional-KL computation, (b) deeper search trajectories, and (c) longer responses. (3) *Generation latency*: SAPO is fastest at 0.087 ms/token—an amortization effect from longer responses spreading the fixed first-token latency over more tokens, rather than a faster decoder per se. (4) *Search depth*: 1.92, the highest among the compared methods, which may facilitate multi-hop reasoning. (5) *Response length*: increases, reflecting the model's tendency to generate longer reasoning chains.

Cross-backbone trends (Figure 7). (1) **Throughput** (a) remains stable across training, with larger models achieving higher

Method	Single-Hop QA			Multi-Hop QA				Avg.
	NQ	TriviaQA	PopQA	HotpotQA	2Wiki	Musique	Bamboogle	
Backbone: LLaMA-3.2-3B-Base								
Search-R1-GRPO	0.394	0.596	0.437	0.280	0.264	0.056	0.105	0.305
Search-R1-SAPO	**0.471**	**0.630**	**0.452**	**0.355**	**0.301**	**0.102**	**0.248**	**0.366**
Δ	↑0.077	↑0.034	↑0.015	↑0.075	↑0.037	↑0.046	↑0.143	↑0.061
Backbone: LLaMA-3.2-3B-Instruct								
Search-R1-GRPO	0.357	0.578	0.378	0.314	0.233	0.090	0.306	0.322
Search-R1-SAPO	**0.475**	**0.634**	**0.446**	**0.444**	**0.386**	**0.187**	**0.384**	**0.422**
Δ	↑0.118	↑0.056	↑0.068	↑0.130	↑0.153	↑0.097	↑0.078	↑0.100

Table 8: Full per-dataset generalization results on LLaMA-3.2-3B (Base/Instruct). SAPO consistently outperforms the GRPO baseline across both variants; the summary appears in the main text as Table 5.

```python
def compute_policy_loss(old_log_prob, log_prob, advantages, eos_mask, cliprange,
                        tau=1.0, gamma=0.1): # SAPO hyperparameters: IS threshold and KL
                            weight
    negative_approx_kl = log_prob - old_log_prob
    ratio = torch.exp(negative_approx_kl)
    ppo_kl = verl_F.masked_mean(-negative_approx_kl, eos_mask)

    pg_losses = -advantages * ratio
    pg_losses2 = -advantages * torch.clamp(ratio, 1.0 - cliprange, 1.0 + cliprange)

    # SAPO: conditional KL penalty on positive-advantage, drifted-down tokens
    kl_term_loss = -verl_F.masked_mean(
        torch.log(ratio).masked_fill(ratio > tau, 0)[advantages[:, 0] >= 0],
        eos_mask[advantages[:, 0] >= 0]
    )

    pg_loss = verl_F.masked_mean(torch.max(pg_losses, pg_losses2), eos_mask)
    pg_clipfrac = verl_F.masked_mean(torch.gt(pg_losses2, pg_losses).float(), eos_mask)
    return pg_loss + gamma * kl_term_loss, pg_clipfrac, ppo_kl
```

Figure 5: PyTorch implementation of the SAPO policy loss on top of the standard GRPO surrogate (VeRL framework (Sheng et al., 2025)). The only addition is the highlighted `kl_term_loss` block—a single masked KL expression that fires on positive-advantage tokens whose importance ratio has drifted below τ—together with one extra term (γ·`kl_term_loss`) in the returned loss. Hyperparameters τ (IS threshold) and γ (KL weight) are added to the function signature; we use $(\gamma, \tau) = (0.1, 1.0)$ throughout.

tokens/s due to better GPU utilization. (2) **Generation time** (b) stabilizes after the initial model-loading overhead. (3) **Testing time** (c) increases by approximately 20–30% as the policy generates longer reasoning chains—a consequence of improved performance rather than inefficiency. (4) **Update-actor time** (d) remains stable, with SAPO introducing less than 5% overhead relative to GRPO.

A.8 Training Dynamics and Search Behavior

To characterize SAPO's learning dynamics, we track four metrics over the course of training (Figure 8).

Key findings. (1) **Information score** (Figure 8a) increases by 2–3× (from 0.05–0.20 to 0.25–0.45), indicating improved retrieval quality through better query formulation. (2) **Valid search count** (b) grows from single-turn (∼ 1.5) to multi-turn (∼2.5) retrieval, with agents learning to iteratively refine searches. (3) **Response length** (c) exhibits model-dependent growth (800–1,400 tokens), with Instruct models producing more verbose reasoning. (4) **Reward** (d) shows rapid early improvement (steps 0–60) followed by stable refinement, with no late-stage collapse—validating SAPO's stability.

Summary. SAPO enables stable learning of multi-turn search strategies while mitigating distributional drift.

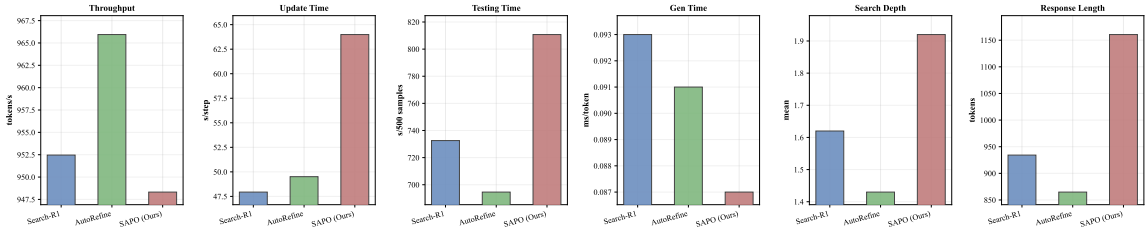

Figure 6: Per-method efficiency comparison at training step 100. SAPO achieves the greatest search depth (1.92) and the fastest per-token generation speed, with a moderate 29–33% increase in update time compared to Search-R1 and AutoRefine.

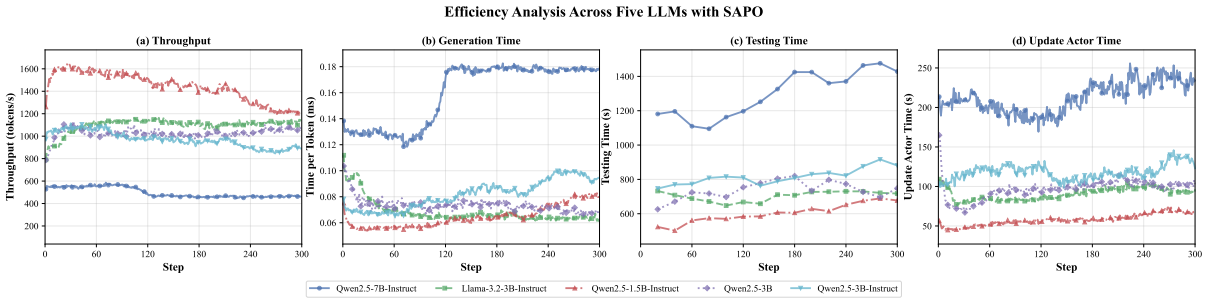

Figure 7: Efficiency analysis of SAPO across five LLMs. (a) Throughput remains stable. (b) Generation time per token. (c) Testing time increases as responses lengthen. (d) Update actor time remains consistent.

A.9 Experimental Results across Seven Benchmarks

To comprehensively characterize the training stability of SAPO, Figure 9 illustrates the evolution of Exact Match (EM) scores across four backbones (Qwen2.5-7B-Instruct, Llama-3.2-3B-Instruct, Qwen2.5-1.5B-Instruct, and Qwen2.5-3B-Instruct). The evaluation spans seven benchmarks: Natural Questions (NQ), TriviaQA, PopQA, Bamboogle, HotpotQA, 2Wiki, and Musique.

Key observations. (1) **Stable, monotonic improvement** across all benchmarks and models, with no catastrophic collapse. (2) **Single-hop tasks** (NQ, TriviaQA, PopQA) converge rapidly (within ≈ 60 steps) to high EM (0.65–0.75). (3) **Multi-hop tasks** (HotpotQA, 2Wiki, Musique, Bamboogle) improve steadily throughout training, with Musique being the most challenging (EM $\approx$0.20). (4) **Backbone ordering by mean EM** matches the scaling and family-transfer studies: Qwen2.5-7B-Instruct (0.462) > Qwen2.5-3B-Instruct (0.442) > Llama-3.2-3B-Instruct (0.422) > Qwen2.5-1.5B-Instruct (0.400), consistent with Tables 7 and 5. (5) **Bamboogle** exhibits higher volatility but a clear upward trend, demonstrating robustness on out-of-distribution counterfactual multi-hop queries. (6) **Cross-model consistency** validates SAPO's generalizability across architectures and scales.

A.10 Head-to-head with GRPO-family Stabilizers

To position SAPO against other algorithm-side stabilizers within the GRPO family, we conduct a head-to-head comparison with DAPO (Yu et al., 2025), GSPO (Zheng et al., 2025), and GMPO (Zhao et al., 2025) under the same Qwen2.5-3B-Instruct + Search-R1 pipeline (rollout G=5, EM-based outcome reward, 500 evaluation samples per dataset). For each method, we report the EM score at its own best training step, together with the step at which we first observe training collapse ("—" indicates no collapse was observed within 420 steps).

Section 4.3 (Table 2) covers the headline findings—highest mean EM, uniquely collapse-free, and the +6.9 EM gap between the two non-collapsing methods (SAPO vs. DAPO) that establishes ISDD and entropy collapse as orthogonal failure modes. Here, we record the per-dataset detail in Table 9. **Per-dataset wins.** SAPO is column-wise best on six of the seven benchmarks (NQ, TriviaQA, PopQA, HotpotQA, 2Wiki, Bamboogle); the only loss is on Musique, where GMPO leads by 0.015 EM (0.186 vs. 0.171). Among the four multi-hop datasets, SAPO sweeps three, including a large +0.041 EM margin on Bamboogle over the next-best method (GSPO), consistent with the multi-hop emphasis of

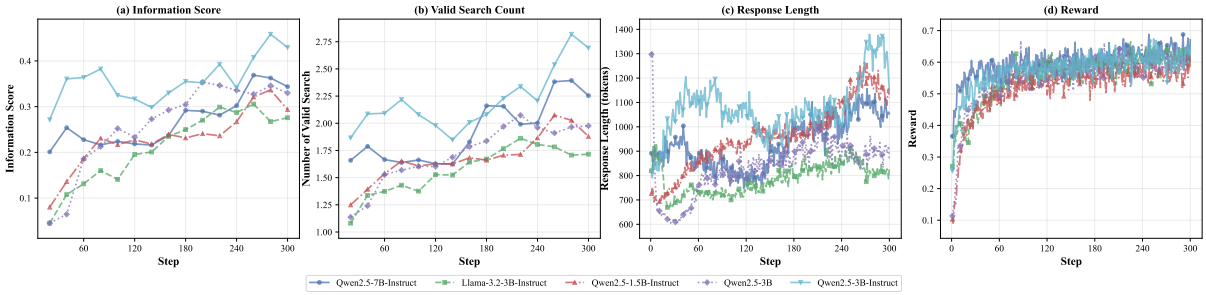

Figure 8: Training dynamics across five LLMs. (a) Information score improves 2-3×. (b) Valid search count grows from 1-1.5 to 2-2.8, showing multi-turn behavior emergence. (c) Response length increases, with Instruct models generating longer reasoning. (d) Reward converges rapidly and stabilizes at 0.50-0.65.

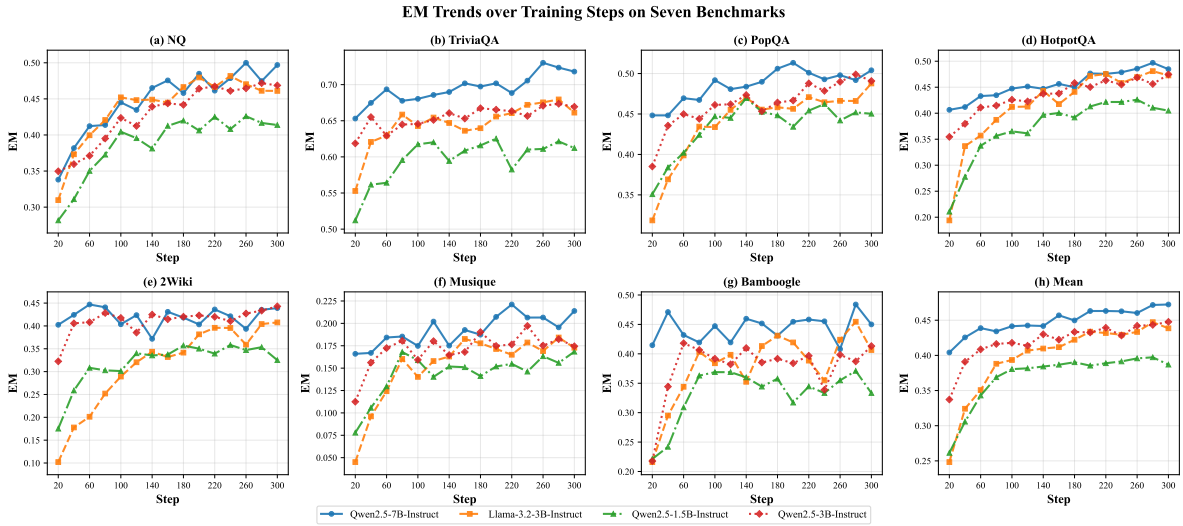

Figure 9: Training trajectories of EM accuracy for SAPO across seven benchmarks.

the conditional KL penalty. **The Musique asymmetry.** Musique is the longest-horizon dataset in the suite (avg. 3.9 retrieval calls vs. 2.4 on the other multi-hop sets). GMPO's geometric mean over the rollout group is more aggressive than SAPO's per-token KL on very long trajectories, which can boost peak EM at the cost of stability—GMPO collapses at step 220, while SAPO continues improving to step 340. We therefore view the Musique gap as a peak-vs.-stability trade-off rather than a methodological superiority.

A.11 Statistical Robustness across Three Seeds

Protocol note. The 3-seed reruns reported in this section use rollout G=5, matching the ablation protocol in Section 4.3 and the GRPO-family head-to-head in Appendix A.10. The single-seed numbers in Table 1 use rollout G=10 (single seed by convention with prior agentic-RL work). The paired-bootstrap significance results below (Table 11) are therefore computed on the matched-rollout 3-seed protocol and propagated as †/∗ markers on the SAPO-3B-Instruct row of Table 1; the structural separation they document carries over to the rollout-10 regime as evidenced by the consistent direction and magnitude of the per-dataset gaps.

To rule out seed-induced artifacts in the main results, we re-run GRPO and SAPO on the Qwen2.5-3B-Instruct backbone under three random seeds $(42, 76, 35)$, keeping every other configuration identical to Section 4.2 (with rollout G=5 as noted above). For each (method, seed) pair, we record the best validation step, the first collapse step ("—" = no collapse), and per-dataset EM at the best step.

Two qualitative conclusions hold without any statistical test. **(a) Non-overlapping separation.** SAPO's worst-seed average (0.4315) exceeds GRPO's best-seed average (0.3436) by +0.088 EM, so seed-level noise alone cannot explain the gap. **(b) Seed-independent collapse.** Three of three GRPO seeds collapse (at steps 80,

Method	Best Step	Collapse Step	NQ	TriviaQA	PopQA	HotpotQA	2Wiki	Musique	Bamboogle	Avg.
GRPO	20	80	0.3245	0.5685	0.4155	0.3299	0.3022	0.1034	0.2623	0.3295
DAPO	100	–	0.3631	0.6130	0.4442	0.3577	0.3809	0.1286	0.2951	0.3689
GSPO	260	260	0.4521	0.6524	0.4675	0.4388	0.3745	0.1724	0.3902	0.4211
GMPO	100	220	0.4291	0.6163	0.4796	0.4029	0.3618	**0.1859**	0.3607	0.4052
SAPO	**340**	**–**	**0.4525**	**0.6613**	**0.4877**	**0.4492**	**0.4097**	0.1711	**0.4309**	**0.4375**

Table 9: Head-to-head comparison with GRPO-family stabilizers under identical Qwen2.5-3B-Instruct + Search-R1 settings (rollout 5, EM-based outcome reward, 500 evaluation samples per dataset). **Bold** marks the column-wise best; "–" indicates no observed collapse within the 420-step training budget.

Method	Seed	Best	Collapse	NQ	TriviaQA	PopQA	HotpotQA	2Wiki	Musique	Bamboogle	Avg.
GRPO baseline (mean Avg. = 0.3346)											
GRPO	42	100	180	0.3564	0.5992	0.4113	0.3367	0.2834	0.0986	0.2295	0.3307
GRPO	76	80	100	0.3503	0.6024	0.4449	0.3279	0.2749	0.1012	0.3033	0.3436
GRPO	35	20	80	0.3245	0.5685	0.4155	0.3299	0.3022	0.1034	0.2623	0.3295
SAPO (ours, mean Avg. = 0.4359, ↑0.1013 vs. GRPO)											
SAPO	42	280	–	0.4372	0.6640	0.4650	0.4676	0.4251	0.2000	0.4113	0.4386
SAPO	76	360	–	0.4422	0.6674	0.4642	0.4485	0.4263	0.1980	0.3740	0.4315
SAPO	35	340	–	0.4525	0.6613	0.4877	0.4492	0.4097	0.1711	0.4309	0.4375

Table 10: Per-seed raw EM at the best validation step for GRPO and SAPO on Qwen2.5-3B-Instruct. "–" indicates no collapse was observed within 420 steps. SAPO's worst-seed Avg. (0.4315) still exceeds GRPO's best-seed Avg. (0.3436) by ↑0.0879.

100, and 180), while zero of three SAPO seeds collapse within the 420-step budget; this matches the collapse statistics observed in Section 4.2 on a single seed and indicates that the stability gain is structural rather than incidental.

Paired bootstrap confidence intervals. To formalize **(a)**, we compute paired-bootstrap 95% confidence intervals for the per-dataset EM difference $\Delta = \text{SAPO} - \text{GRPO}$, resampling the N=500 evaluation prompts B=10,000 times and pairing predictions at the prompt level for each of the nine (GRPO seed, SAPO seed) pairings (the reported Δ is the cross-pairing mean; the CI is the union of the nine pairing-level percentile intervals, which is the conservative choice). Paired t-tests are computed on the same per-prompt EM differences. Results are summarized in Table 11.

The aggregate gap $\Delta_{\text{avg}} = +0.101$ has a 95% CI of $[+0.072, +0.130]$, more than three CI half-widths above zero. Multi-hop datasets show larger gaps and tighter relative intervals than single-hop ones, consistent with SAPO's multi-hop emphasis (Section 4.2). Under Bonferroni correction at α=0.05/7≈0.0071, six of seven benchmarks and the average survive (NQ, TriviaQA, HotpotQA, 2Wiki, Musique, Bamboogle); PopQA (p=0.030) passes uncorrected only. These verdicts are reflected as †/∗ on the SAPO-3B-Instruct row of Table 1.

Dataset	Δ	95% CI	p
NQ	+0.100	[+0.052, +0.149]	$< 10^{-4}$
TriviaQA	+0.074	[+0.029, +0.119]	0.0012
PopQA	+0.048	[+0.005, +0.092]	0.030
HotpotQA	+0.124	[+0.078, +0.169]	$< 10^{-4}$
2Wiki	+0.134	[+0.090, +0.177]	$< 10^{-4}$
Musique	+0.089	[+0.055, +0.123]	$< 10^{-4}$
Bamboogle	+0.140	[+0.078, +0.203]	$< 10^{-4}$
Avg.	**+0.101**	**[+0.072, +0.130]**	$\mathbf{< 10^{-4}}$

Table 11: Paired-bootstrap 95% CIs and paired t-test p-values for the per-dataset EM gap $\Delta = \text{SAPO} - \text{GRPO}$ (per-dataset GRPO/SAPO means are in Table 10). CIs use B=10,000 prompt-level resamples on the N=500-prompt evaluation set, taking the union over the nine seed pairings (conservative). All seven CIs exclude zero; six datasets pass Bonferroni correction at α=0.05/7≈0.0071, and PopQA (p=0.030) passes uncorrected.

A.12 Search-Behavior Probe across Five Backbones

To complement the training-curve view in Appendix A.8, we run a fine-grained search-behavior probe across **five backbones** × **two checkpoints** (steps 200 and 300) on the seven QA bench-

Backbone	Valid Search #	Info Score	Info Rev. Rank	Search Valid Ratio	Finish Ratio
Instruct backbones					
Qwen2.5-7B-Instruct	2.16 → 2.25	0.290 → 0.344	0.298 → 0.350	0.687 → 0.695	0.981 → 0.991
Qwen2.5-3B-Instruct	2.20 → 2.72	0.365 → 0.425	0.378 → 0.486	0.691 → 0.733	0.982 → 0.989
Qwen2.5-1.5B-Instruct	1.71 → 1.88	0.241 → 0.294	0.248 → 0.296	0.634 → 0.657	0.985 → 0.981
Llama-3.2-3B-Instruct	1.77 → 1.72	0.270 → 0.276	0.269 → 0.275	0.663 → 0.646	0.899 → 0.938
Base backbone					
Qwen2.5-3B-Base	2.01 → 2.13	0.363 → 0.364	0.366 → 0.367	0.669 → 0.682	0.995 → 0.992

Table 12: Search-behavior trajectories under SAPO across five backbones, formatted as "step 200 → step 300". Each cell is averaged over single- and multi-hop QA. The shaded row marks our headline backbone (Qwen2.5-3B-Instruct).

Backbone	Info Score (IS)		Info Rev. Rank (IRR)		Search Valid Ratio (SVR)	
	SH	MH	SH	MH	SH	MH
Instruct backbones						
Qwen2.5-7B-Instruct	0.210	0.444	0.222	0.447	0.623	0.733
Qwen2.5-3B-Instruct	0.333	0.494	0.387	0.560	0.675	0.765
Qwen2.5-1.5B-Instruct	0.150	0.401	0.151	0.404	0.577	0.700
Llama-3.2-3B-Instruct	0.094	0.412	0.094	0.410	0.557	0.695
Base backbone						
Qwen2.5-3B-Base	0.258	0.444	0.265	0.443	0.612	0.720

Table 13: Single-hop (**SH**) vs. multi-hop (**MH**) breakdown of Information Score (IS), Information Reverse Rank (IRR), and Search Valid Ratio (SVR) at step 300 under SAPO. The shaded row marks our headline backbone (Qwen2.5-3B-Instruct).

marks. Beyond the standard valid-search count, search-valid ratio, and finish ratio, we introduce two reward-external metrics that decouple retrieval competence from the reward signal:

- **Information Score (IS)**: cover-EM of the ground-truth answer over the union of retrieved `<documents>` blocks; measures whether an answer-bearing document is ever retrieved during the trajectory.
- **Information Reverse Rank (IRR)**: cumulative $\sum_k 1/k$ over the search turns at which an answer-bearing document first surfaces (MRR-style); measures how early useful evidence arrives.

Reading IS and IRR jointly separates weak retrieval (low IS) from competent-but-inefficient retrieval (high IS, low IRR)—a distinction that EM and F1 alone cannot capture.

Table 12 reports the trajectory of each metric, averaged over single- and multi-hop QA, formatted as "step 200 → step 300". Table 13 provides the step-300 breakdown by single-hop (**SH**) and multi-hop (**MH**) subsets.

Four patterns emerge from Tables 12 and 13. **(a) Deeper, not noisier.** Valid search count grows on four of five backbones while finish ratio remains ≥ 0.94—agents search more without losing well-formed terminations. **(b) More accurate.** IS rises on every backbone (peak: Qwen2.5-3B-Instruct, $0.365 \rightarrow 0.425$), confirming that more retrievals translate into more answer-bearing documents rather than wasted queries. **(c) Earlier evidence.** IRR grows faster than IS ($0.378 \rightarrow 0.486$ vs. $0.365 \rightarrow 0.425$ on the strongest backbone), meaning useful documents surface in earlier turns rather than only eventually being uncovered. **(d) Gains concentrate where trajectories are longer.** On every backbone, IS-MH dominates IS-SH and IRR-MH dominates IRR-SH, with margins that grow with backbone capacity. Longer trajectories accumulate more IS-ratio drift, which is exactly the regime SAPO's conditional KL is designed to control; the search-behavior gap therefore mirrors—and provides a mechanistic counterpart to—the multi-hop EM gap reported in Section 4.2.

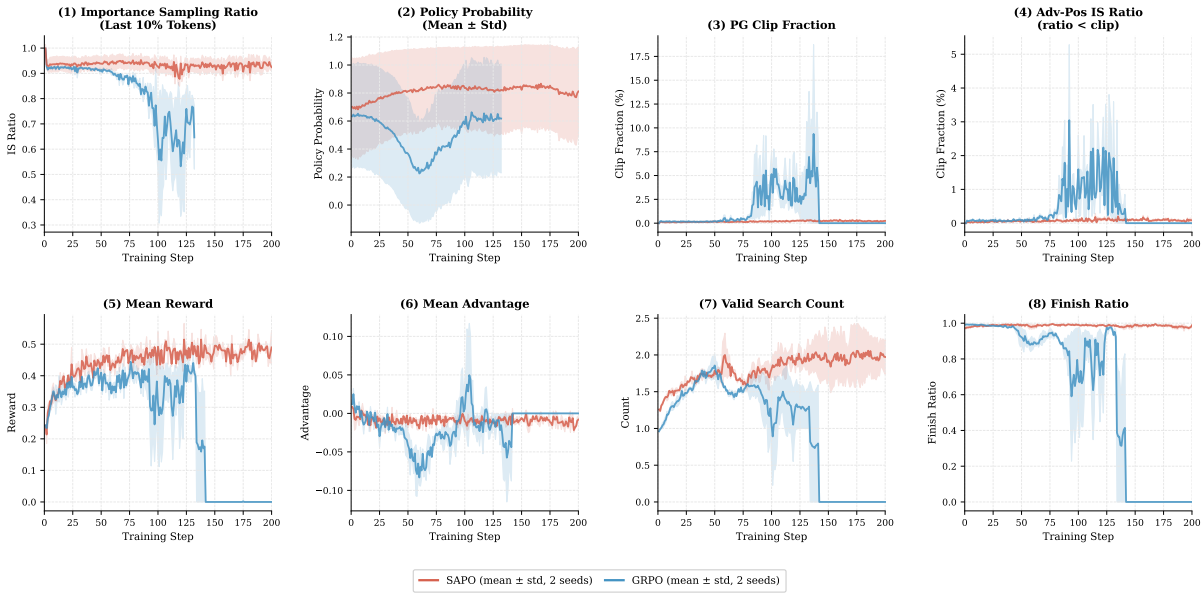

Figure 10: **Training Stability Analysis: SAPO vs. GRPO (first 200 steps).** Each panel shows the per-step mean over two independent random seeds (line), with the shaded band reporting ± 1 across-seed standard deviation. The two-seed runs here are a subset of the three-seed protocol in Appendix A.11; the third seed is omitted because its per-step training logs were not retained, while its end-of-training checkpoint is still used in Appendix A.11. **Row 1 (policy-level):** (1) IS ratio (last 10% of tokens); (2) mean policy probability under the new policy ($\pm$ in-batch std); (3) PG (PPO) clip fraction; (4) fraction of positive-advantage tokens whose IS ratio falls below the lower clip $1-\varepsilon$. **Row 2 (training dynamics):** (5) mean reward; (6) mean advantage; (7) valid search count per episode; (8) finish ratio. Where the two GRPO seeds diverge qualitatively (one collapses, one does not), we plot both seeds and note this in the text rather than averaging them.

A.13 Training Stability Analysis

To zero in on the policy-level dynamics that drive the per-dataset gaps in Tables 10 and 11, we visualize eight training-stability metrics for SAPO and GRPO on Qwen2.5-3B-Instruct over the first 200 steps (Figure 10). The observations below establish a temporal ordering—IS-ratio drift in panels (1)–(4) precedes reward and behavioral collapse in panels (5)–(8) by roughly 50–60 steps—which is the empirical signature underlying the ISDD claim in Section 3.2.

(1) IS ratio. SAPO maintains an IS ratio of ≈ 0.93 throughout training. GRPO, by contrast, undergoes a sharp, volatile drop between steps 75 and 140, indicating that the new policy diverges quickly from the rollout policy and destabilizes the importance weights on a non-trivial fraction of tokens—the empirical instantiation of ISDD (Definition 1).

(2) Policy probability. SAPO's mean new-policy probability rises steadily from ≈ 0.67 to ≈ 0.83 with a narrow band. GRPO's collapses to near zero around step 50 with extreme variance and only slowly recovers; this is the policy-degeneracy signature predicted by the gradient analysis in Section 3.4, and SAPO's conditional KL prevents it from forming.

(3) PG clip fraction. GRPO exhibits a pronounced spike in the PPO clip fraction between steps 75 and 140 (peaking at $\sim 17\%$), coinciding with the instability in panels (1)–(2). SAPO's clip fraction stays near zero, confirming that updates remain inside the proximal trust region without ever needing the hard clip to engage.

(4) Adv-Pos IS-ratio clip. During the same unstable window, GRPO shows a burst of positive-advantage tokens whose IS ratios fall below the lower clip $1-\varepsilon$, simultaneously with high-clip events on the other tail—i.e., the very tokens that should be reinforced are being silently zeroed. SAPO produces negligible clip fractions on either side, evidence that the conditional KL repairs drift before the hard clip has to.

(5)–(6) Reward and advantage. SAPO achieves consistently higher mean rewards ($\approx$ 0.45–0.50) and advantage estimates that stay well-centered near zero with low variance throughout training. The two GRPO seeds split: one collapses to zero reward shortly after step 140 with ± 0.10

advantage swings; the other survives but plateaus well below SAPO. Crucially, the collapsing seed's reward crash in panel (5) lags its IS-ratio crash in panel (1) by ∼50–60 steps, exactly the order predicted if IS-ratio drift is the upstream driver of reward collapse rather than its consequence.

(7) Valid search count. SAPO's valid-search count grows steadily ($\approx 1.5 \rightarrow 2.0$ per episode), indicating progressively better-calibrated retrieval behavior. The collapsing GRPO seed's valid-search count drops to zero shortly after step 140, mirroring the reward crash and confirming that the failure propagates from the policy onto the agent's tool-use behavior.

(8) Finish ratio. SAPO maintains a near-perfect finish ratio (≈ 0.97–1.0) from early in training, indicating that the model reliably emits a final answer within the allowed turns. The collapsing GRPO seed's finish ratio also drops to zero shortly after step 140, consistent with the breakdown observed in the other panels.

Taken together, the eight panels reveal an irreversible failure mode in GRPO that originates at the IS-ratio level (panels 1–4, steps 75–140) and only later manifests as reward, search, and answer collapse (panels 5–8, steps ≥ 140). This temporal lag is the central piece of quantitative causal evidence behind the ISDD framing: policy-level drift is a leading indicator of agent-level collapse, not a co-symptom. SAPO eliminates this failure mode through its conditional KL: every panel for SAPO is monotonic or near-monotonic, with no analog of the GRPO crash at any of the eight metrics.

KL-Term Token Fraction Analysis. Figure 11 reports the per-step fraction of response tokens that activate SAPO's conditional KL term during training of Qwen2.5-3B-Instruct, i.e., positive-advantage tokens whose IS ratio falls below the threshold τ defined in Eq. 13. Across all training steps the activation fraction is $\approx 10.1\%$ on average (per-step range 3.7%–19.0%, with no monotone trend), so on roughly nine out of ten tokens the SAPO update is gradient-equivalent to the unmodified GRPO update; the conditional KL only acts on the minority of tokens where positive-advantage signal is being silently zeroed by IS-ratio drift. This quantitatively substantiates the "sparsely triggered by design" claim in Section 3.2 and the entropy phase analysis in Section 4.3: a regularizer with a ∼10% activation rate cannot systematically suppress diversity, which is consistent with the Phase-2 entropy re-expansion in Figure 1(c) and the absence of clip-fraction inflation for SAPO in Figure 10 (3)–(4).

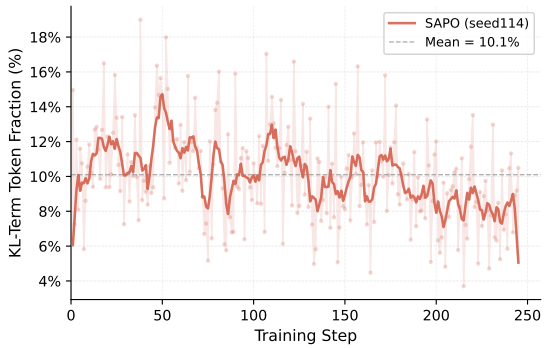

Figure 11: **KL-term token fraction during SAPO training.** Fraction of response tokens that activate SAPO's conditional KL term per step (i.e., positive-advantage tokens whose IS ratio falls below τ). Dots are raw per-step values; the solid curve is a 7-step moving average; the dashed line marks the overall mean ($\approx 10.1\%$).

A.14 Case Studies

Tables 14 and 15 present complete reasoning trajectories generated by the 3B and 14B models using SAPO. Drawing from randomly sampled instances in 2WikiMultihopQA and Musique, respectively, these qualitative results highlight the agent's capability for autonomous reasoning and iterative query refinement.

Question: Which film has the director who died earlier, Deuces Wild or Cavalcade Of The West? **Ground Truth:** ["Cavalcade of the West", "Cavalcade Of The West"]
Model Output: Qwen2.5-3B-Instruct-SAPO
<think> To determine which film has the director who died earlier between "Deuces Wild" and "Cavalcade Of The West", I need to gather information about the directors of these films and their death dates. Let's start by searching for this information. </think> <search> **Director of Deuces Wild** </search> <documents> **Doc 1 (Title: "Deuces Wild"):** Deuces Wild is a 2002 American crime drama film directed by **Scott Kalvert**... **Doc 3 (Title: "Scott Kalvert"):** Scott Kalvert (August 15, 1964 – **March 5, 2014**) was an American film director, known mainly for his 1995 film "The Basketball Diaries"... and 2002's "Deuces Wild". </documents> <search> **Director of Cavalcade Of The West** </search> <documents> **Doc 1 (Title: "Cavalcade of the West"):** Cavalcade of the West is a 1936 American Western film directed by **Harry L. Fraser**... **Doc 3 (Title: "Cavalcade of the West"):** ...Cavalcade of the West is a 1936 American Western film directed by Harry L. Fraser. </documents> <search> **Harry L. Fraser** </search> <documents> **Doc 1 (Title: "Harry W. Fraser"):** Harry W. Fraser (7 June 1884 – **13 May 1950**) was an American labor leader... **Doc 3 (Title: "Harry W. Fraser"):** Fraser was president of the Order of Railway Conductors of America from 1941 to 1950. ... Fraser was born on June 7, 1884 and **died on May 13, 1950**. </documents> <search> **Scott Kalvert** </search> <documents> **Doc 1 (Title: "Scott Kalvert"):** Scott Kalvert (August 15, 1964 – **March 5, 2014**) was an American film director... **Doc 2 (Title: "Scott Kalvert"):** ...on March 5, 2014 from an apparent suicide. </documents> <think> Based on the information provided, Scott Kalvert died on March 5, 2014, while Harry L. Fraser died on May 13, 1950. Therefore, the director who died earlier is Harry L. Fraser, and the film "Cavalcade Of The West" was directed by him. </think> <answer> **Cavalcade Of The West** </answer>

Table 14: Case study illustrating SAPO trajectories from the 2WikiMultihopQA benchmark. Key evidence is extracted from the original searched documents.

Question: Who is the president of the newly declared independent country that is part of the Commission of Truth and Friendship with the country that Tony Gunawan is from?
Ground Truth: ["Francisco Guterres"]

Model Output: Qwen2.5-14B-Instruct-SAPO

<think> To answer this question, I need to identify the country that Tony Gunawan is from, find out if it has a Commission of Truth and Friendship with any newly declared independent country, and then find out who the president of that newly declared independent country is. </think>

<search> **Tony Gunawan country of origin** </search>

<documents> **Doc 1 (Title: "Tony Gunawan"):** Tony Gunawan (, born 9 April 1975) is a Chinese descended, **Indonesian born** badminton player... Gunawan is a former Olympic gold medalist and world champion for Indonesia...
Doc 2 (Title: "Tony Gunawan"): ...Tony played for Indonesia from 1992–2001...
Doc 3 (Title: "Tony Gunawan"): ...He was a member of the world champion 2000 Indonesia Thomas Cup team... </documents>

<think> From the documents, it is clear that Tony Gunawan is from Indonesia. Now I need to find out if Indonesia has a Commission of Truth and Friendship with any newly declared independent country. </think>

<search> **Commission of Truth and Friendship Indonesia** </search>

<documents> **Doc 1 (Title: "Indonesia–Timor Leste Commission of Truth and Friendship"):** The Indonesia–Timor Leste Commission on Truth and Friendship was a truth commission established jointly by the governments of **Indonesia and East Timor** in August 2005...
Doc 2 (Title: "Indonesia–Timor Leste Commission of Truth and Friendship"): ...investigate acts of violence that occurred around the independence referendum held in East Timor in 1999... </documents>

<think> From the documents, it is clear that Indonesia has a Commission of Truth and Friendship with East Timor. Now I need to find out if East Timor is a newly declared independent country and who its president is. </think>

<search> **East Timor president** </search>

<documents> **Doc 1 (Title: "Francisco Guterres"):** Francisco Guterres, popularly known as Lú-Olo... is an East Timorese politician who has been **President of East Timor since 2017**...
Doc 3 (Title: "José Ramos-Horta"): ...José Manuel Ramos-Horta... was the President of East Timor from 20 May 2007 to 20 May 2012... </documents>

<think> From the documents, it is clear that Francisco Guterres is the current president of East Timor. However, I need to confirm if East Timor is a newly declared independent country. </think>

<search> **East Timor independence** </search>

<documents> **Doc 3 (Title: "East Timor"):** ...East Timor became the first new sovereign state of the 21st century on 20 May 2002... </documents>

<think> From the documents, it is clear that East Timor gained independence in 2002, which makes it a newly declared independent country. Therefore, the president of the newly declared independent country that is part of the Commission of Truth and Friendship with Indonesia is Francisco Guterres. </think>

<answer> **Francisco Guterres** </answer>

Table 15: Case study illustrating SAPO from the Musique benchmark. Key evidence is extracted from the original searched documents.