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, which we term Importance Sampling Distribution Drift (ISDD). Within Group Relative Policy Optimization (GRPO), ISDD manifests as a rapid decline in importance sampling ratios, which substantially diminishes gradient signals and can lead to training collapse. To overcome this limitation, we propose **S**earch **A**gent **P**olicy **O**ptimization (**SAPO**), a method that 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. Specifically, it targets only positive-advantage tokens whose probabilities have shifted beyond a threshold, thereby mitigating distribution drift while preserving gradient flow. Notably, SAPO requires only a single-line code modification to standard GRPO, facilitating straightforward adoption. Experiments on seven QA benchmarks show that SAPO achieves a **10.6% absolute improvement** (+31.5% relative) over Search-R1, with consistent gains across model scales (1.5B–14B) and architectures (Qwen, LLaMA). The code is available at `https://anonymous.4open.science/r/SAPO-5CEF`.

1 Introduction

Recent studies (Li et al., 2025a; Guo et al., 2024) have introduced 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), employ Group Relative Policy Optimization (GRPO) (Shao et al., 2024) with rule-based rewards to train agents to interact with a search tool over large corpora. As a result, GRPO has become a widely adopted post-training algorithm for Tool-based Agentic Reinforcement Learning (TARL), inspiring a growing body of follow-up works (Mei et al., 2025; Shi et al., 2025a; Qian and Liu, 2025; Zhang et al., 2025b).

Despite these encouraging results, existing TARL frameworks face a notable limitation: **Importance Sampling Distribution Drift (ISDD)**, in which importance sampling ratios tend toward zero as the policy diverges beyond the trust region. Although GRPO adopts hard clipping (Schulman et al., 2017) to constrain policy updates, ISDD can persist, particularly as the number of search turns increases. A key contributing factor to this instability lies in GRPO's advantage estimation: all tokens within a response share an identical advantage value, giving rise to two related issues. (1) *Inaccurate Credit Assignment*: the advantage may not reflect the true contribution of individual tokens, since different reasoning steps contribute unequally to the final answer. (2) *Misleading Updates*: a response with a correct final answer may contain erroneous intermediate steps, and vice versa. Conflicts arise when the old policy assigns high probability to intermediate correct tokens while the current policy suppresses them due to a negative advantage, thereby driving the importance sampling weights toward zero.

Figure 1 illustrates the training dynamics of Search-R1 under GRPO, tracking the Importance Sampling (IS) ratios, clip ratio, entropy, and reward throughout training. While the IS ratios remain near 1 during the initial phase, they subsequently decline, indicating a growing divergence between the current and old policies. The clip ratio exhibits a notable increase in the latter stages as a consequence. Meanwhile, policy entropy rises early in training, suggesting the presence of low-probability positive tokens or high-probability negative tokens. Given the tendency of IS ratios toward zero, we identify **the suppression of positive tokens (low-probability positive actions)** as a principal contributing factor to ISDD. Although the outcome-based reward initially improves, it deteriorates as

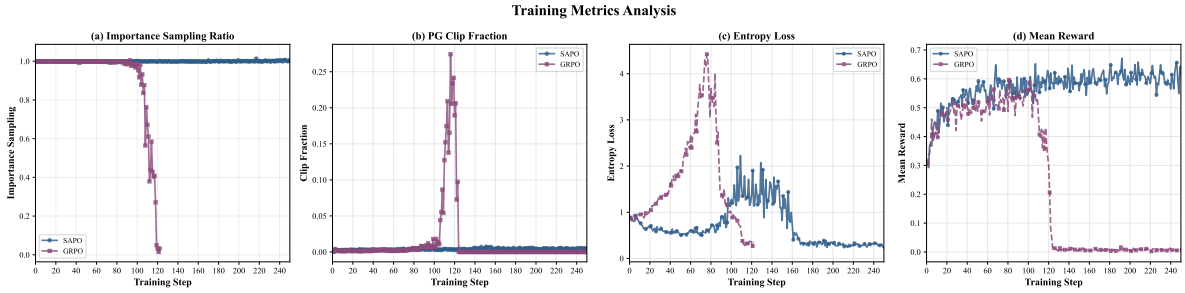

Figure 1: Comparison of training dynamics between **SAPO** and GRPO regarding (a) Importance Sampling Ratio, (b) Clip Ratio, (c) Entropy, and (d) Reward.

IS ratios destabilize and entropy collapses, suggesting that late-stage instability can undermine final performance. These empirical observations indicate that severe model collapse can persist despite the use of hard clipping.

The primary motivation behind IS and clipping is to correct for distribution shifts and ensure stability. However, for positive tokens, if the probability under the current policy drops significantly below that of the old policy, the update is effectively suppressed by excessively low importance weights, yielding a weak update signal that prevents the model from recovering valid actions. This observation motivates the following question:

How can we mitigate conflicts through 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 straightforward yet principled modification of GRPO. Drawing inspiration from PPO_KL (Schulman et al., 2017), SAPO introduces a conditional KL penalty term that enforces a token-level constraint on the distributional divergence between the current and old policies. To focus the constraint where it is most needed, SAPO employs an asymmetric mechanism that selectively penalizes conflicting positive tokens while preserving informative gradients. In effect, SAPO functions as a soft trust region constraint: rather than applying hard clipping, it penalizes large deviations in IS ratios, permitting adaptive step sizes while maintaining distributional proximity.

In summary, our contributions are as follows:

- We present SAPO, a policy optimization method designed to stabilize post-training for autonomous multi-turn search agents tackling complex, real-world questions.
- We introduce a conditional KL penalty term that enforces a token-level constraint on distributional divergence, specifically targeting low-probability positive tokens.
- We evaluate SAPO on seven QA benchmarks across diverse search agents, demonstrating its effectiveness and generalizability.

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 facilitates stable training by utilizing 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 represents the advantage function, q denotes a query sampled from the data distribution $\mathcal{D}$, and o represents a response generated by the old policy. Directly optimizing this objective can result in large deviations of the IS ratios $r_t(\theta)$, leading to training instability. To mitigate this, PPO introduces a clipped surrogate objective that imposes a local trust region:

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

Here, ϵ is a hyperparameter defining the clipping bounds. This mechanism limits the incentive for policy updates when the IS ratio deviates substantially, thereby enhancing stability. However, it

effectively eliminates 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) eliminates the need for a value model by employing group-based advantage estimation. Following Search-R1 (Jin et al., 2025), external retrieval tokens are masked during search agent training. Consequently, 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. Notably, GRPO assigns an identical advantage value to all tokens within the same response.

3 Approach

3.1 Search Agent Formulation

Let q denote the user question, a the generated answer, and π_θ the large language model (LLM). We formulate the standard question-answering task as $a = \pi_\theta(\cdot \mid q)$.

RAG systems enhance this process by retrieving knowledge k from an external corpus $\mathcal{D}$ using a retriever $\mathcal{R}(\cdot)$, and subsequently generating an answer conditioned on both q and k: $a = \pi_\theta(\cdot \mid q, k), \quad k = \mathcal{R}(q, \mathcal{D})$.

However, this static retrieve-then-generate workflow can be insufficient for complex search tasks that require iterative reasoning and adaptive retrieval. Recent search agents adopt a more dynamic paradigm in which retrieval is interleaved with reasoning.

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

Here, the policy output o comprises intermediate reasoning steps z and the final answer a; sq_i denotes 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 is the number of search actions.

3.2 Importance Sampling Distribution Drift

In search agents, effective actions tend to be sparse, as only a limited subset of queries yields useful results. When the policy π_θ rapidly diverges from the exploration path of $\pi_{\theta_{\text{old}}}$, it may assign negligible probabilities to the sampled actions, causing $\pi_\theta(o_t) \ll \pi_{\theta_{\text{old}}}(o_t)$ and driving $r_t(\theta) \to 0$.

Definition 1 (ISDD)**.** Building upon 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 small $\epsilon > 0$ and a probability threshold ϕ. This condition indicates that a significant proportion of trajectories suffer from vanishing cumulative importance weights.

Why ISDD Causes Collapse. In GRPO, the policy gradient is weighted by 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 contribute negligible gradients, which substantially limits the model's ability to learn from successful explorations.

Proposition 1 (ISDD Amplification in Interleaved Multi-Step Interactions)**.** *Let the IS ratios r_t follow a log-normal distribution $\log r_t \sim \mathcal{N}(\mu, \sigma^2)$, with the drift parameter $\lambda = \mu + \sigma^2/2$. Due to the low-entropy, bottleneck nature of tool selection, action tokens a_i tend to exhibit higher sensitivity to policy shifts than reasoning tokens z_i, implying $\lambda_a \ll \lambda_z < 0$.*

The expected cumulative importance weight for the trajectory decomposes as follows:

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

where L denotes the total number of tokens. ISDD is more severe in agentic tasks than in standard QA tasks because the ***Interaction Drift*** *term decays exponentially with the volume of actions L_a, driven by the highly negative drift parameter λ_a.*

3.3 Search Agent Policy Optimization

To mitigate ISDD and improve training stability, SAPO introduces an auxiliary penalty term that enforces a constraint on the distributional divergence between the current and old policies. The SAPO token-level objective is formulated as:

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

where γ denotes the penalty coefficient. Unlike the KL penalty in standard GRPO, which typically regulates divergence against a fixed reference model, SAPO penalizes deviations from the dynamic old policy and approximates the KL divergence using the log-ratio:

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

A standard KL penalty as in Eq. 12 may broadly constrain policy exploration. To address this, we propose a conditional KL penalty that activates only when the probability ratio drops significantly for tokens with positive advantages. The conditional penalty term is defined as:

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

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 provides two benefits:

- **Mitigates over-suppression**: Positive tokens with $r_t < \tau$ receive a gradient contribution, which 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 $\hat{A}_t \leq 0$ (should be suppressed) are not penalized, allowing the policy to optimize in stable regions.

3.4 Gradient Analysis

The gradient of the SAPO auxiliary term is derived via the chain rule, as presented in Eq. 14. The total gradient of the SAPO objective is then formulated in Eq. 15, where $\delta(\cdot)$ denotes the Dirac delta function arising from differentiation of the indicator function $\mathbb{I}(\cdot)$. This gradient coefficient functions as an adaptive regularizer that increases as $r_t \to 0$, providing a stronger correction signal for tokens that have drifted further from the old policy.

4 Experiments

4.1 Experimental Settings

Datasets. We evaluate our method on seven diverse 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 comprise HotpotQA (Yang et al., 2018), 2WikiMultihopQA (Ho et al., 2020), Musique (Trivedi et al., 2022b), and Bamboogle (Press et al., 2022). We employ 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 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),

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

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

AutoRefine (Shi et al., 2025a), and SE-Search (Li et al., 2026).

Implementation Details. We utilize the external searching corpus (Karpukhin et al., 2020) with E5-base-v2 (Wang et al., 2022) serving 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 employ the Qwen2.5 series (Qwen et al., 2025) as the backbone language model. In the RL phase, we adopt a simple rule-based, outcome-oriented reward computed using the F1 score between the predicted and ground-truth answers. The KL penalty coefficient is set to $\gamma = 0.1$, and the IS ratio threshold is set to $\tau = 1.0$.

4.2 Main Performance

Table 1 presents the main experimental results comparing SAPO against baseline methods. The *Avg.* column reports the average accuracy across all benchmarks. For our controlled comparisons, SAPO, Search-R1, and the closely related agentic baselines are evaluated under identical Qwen2.5-3B settings with the corresponding retriever and corpus configuration.

SAPO achieves the highest overall performance. SAPO outperforms Search-R1 and SOTA methods (AutoRefine and SE-Search) on all seven benchmarks and attains the best overall average among the compared methods, achieving an average EM accuracy of 0.442. These results suggest that the proposed post-training strategy meaningfully enhances the search agent's capabilities, yielding an absolute improvement of 10.6 points (a relative gain of 31.5%) over the Search-R1 baseline.

SAPO shows notable gains on multi-hop QA. The improvements are more pronounced on complex, multi-hop tasks. Compared to CriticSearch, SAPO improves performance on HotpotQA by 4.2 percentage points (10.1% relative) and on Bamboogle by 6.4 percentage points (17.4% relative). Across all multi-hop benchmarks, SAPO achieves an average increase of 5.3 percentage points (15.45% relative). Relative to the Search-R1 baseline, SAPO attains an increase of 14.7 percentage points (59.0% relative). We attribute these improvements largely to the training stability provided by the proposed conditional KL penalty term.

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 utilizes Qwen2.5-3B-Instruct as the backbone and the F1 score as the outcome-based reward. Table 2 reports accuracy across seven benchmarks. We compare four configurations: (1) **GRPO**: the baseline algorithm as employed in Search-R1; (2) **GRPO w/ KL**: GRPO augmented with an unconditional KL penalty term; (3) **GRPO w/ KL_r**: GRPO with a KL penalty conditioned solely on the IS ratio; (4) **GRPO w/ KL_ra (SAPO)**: the complete SAPO configuration, where the KL penalty is conditioned on both the IS ratio and the advantage.

Impact of Selective KL Penalties. Table 2 summarizes the performance progression. Starting from the Search-R1-GRPO baseline (average accuracy of 0.388), incorporating a simple unconditional KL term yields notable gains on Musique (+30.6%) and HotpotQA (+14.4%), resulting in a modest overall increase to 0.398. The ratio-conditioned variant (KL_r) further improves stability, raising the average to 0.417 and notably improving HotpotQA to 0.457. The full SAPO configuration (KL_ra) achieves the best overall balance, securing the highest scores on six out of

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 comparison of SAPO-3B against baseline methods using Qwen2.5-3B (Qwen et al., 2025) across multiple QA benchmarks. **Bold** indicates the best results, and underline denotes the second-best results.

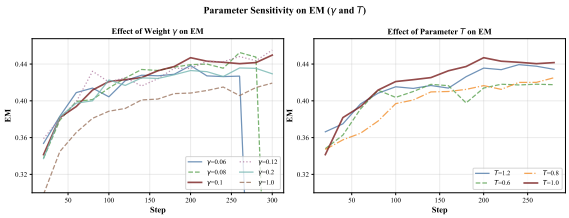

Figure 2: Hyperparameter sensitivity analysis: the KL penalty coefficient γ, and the IS ratio threshold τ.

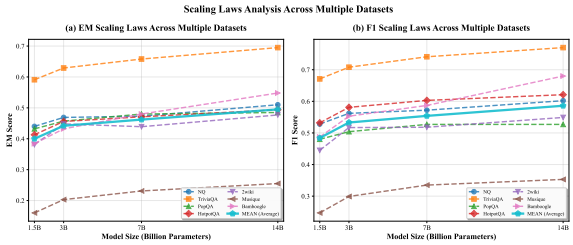

Figure 3: Scaling trends of Qwen2.5-Instruct with **SAPO** across different model sizes (1.5B to 14B). We report both EM and F1 scores.

seven benchmarks with an average of 0.429—an absolute increase of 0.041 over the baseline (approximately 10.6% relative). These results suggest that the conditional KL penalty in SAPO helps balance exploration and constraint, yielding consistent aggregate improvements.

4.4 Detailed Analysis

Hyperparameter Sensitivity. We investigate the sensitivity of SAPO to the KL penalty γ and the IS ratio τ. The validation EM scores in Figure 2 demonstrate that $\gamma = 0.1, \tau = 1.0$ consistently yields superior performance. Specifically, with fixed $\gamma = 0.1$, $\tau = 1.0$ achieves an EM score of 0.442, outperforming both higher ($\tau = 1.2$, 0.434) and lower thresholds ($\tau \in \{0.6, 0.8\}$, scoring 0.418 and 0.425). The widening performance gap during training underscores the critical role of τ in later stages. Fixing $\tau = 1.0$, we observe that $\gamma = 0.1$ yields the highest EM score. Notably, lower γ values lead to training collapse, while higher values result in performance degradation. Consequently, we adopt $\gamma = 0.1, \tau = 1.0$ as the default to balance stability and efficiency.

Scaling with Model Size. We investigate the scalability of SAPO by evaluating its performance across backbone models of varying sizes, specifically the Qwen2.5-Instruct series ranging from 1.5B to 14B parameters. Figure 3 reports the EM and F1 scores. We observe a positive correlation be-

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	0.625	0.443	0.428	0.371	0.175	0.342	0.398
	(+0.023)	(-0.037)	(+0.007)	(+0.054)	(-0.005)	(+0.041)	(+0.009)	(+0.010)
+ KL_r	0.439	0.660	0.456	**0.457**	0.369	0.169	0.369	0.417
	(+0.014)	(+0.035)	(+0.013)	(+0.029)	(-0.002)	(-0.006)	(+0.027)	(+0.019)
+ KL_{ra} (SAPO)	**0.442**	**0.673**	**0.474**	0.433	**0.415**	**0.181**	**0.382**	**0.429**
	(+0.003)	(+0.013)	(+0.018)	(-0.024)	(+0.046)	(+0.012)	(+0.013)	(+0.012)

Table 2: Ablation results on different KL terms in **SAPO**. The values in parentheses denote the absolute performance change compared to the previous row.

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**
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**

Table 3: Generalization results on LLaMA-3.2-3B (Base and Instruct). **SAPO** consistently outperforms the GRPO baseline across both model variants.

tween model size and performance, consistent with established scaling laws (Kaplan et al., 2020). As the parameter count increases, SAPO demonstrates monotonic improvement in average accuracy. Scaling from 1.5B to 14B yields a notable performance gain, raising the average EM from 0.400 to 0.495 and the F1 score from 0.484 to 0.586. This suggests that SAPO can effectively leverage the reasoning capabilities of larger LLMs.

Robustness across Model Families. To assess the generalizability of SAPO across different architectures, we extend our evaluation to the LLaMA-3.2 (Touvron et al., 2023) series, testing both the Base and Instruct variants of the 3B model. As shown in Table 3, SAPO outperforms the GRPO baseline across both variants. On LLaMA-3.2-3B-Base, SAPO improves the average accuracy from 0.305 to 0.366, suggesting its ability to align base pre-trained models. The gains are more pronounced on LLaMA-3.2-3B-Instruct, where SAPO improves average accuracy by 10.0 percentage points (from 0.322 to 0.422). These results suggest that SAPO can serve as a model-agnostic approach for enhancing search capabilities across different model families and alignment stages.

Performance Metrics Comparison. To evaluate training efficiency, we benchmark SAPO against Search-R1 and AutoRefine across six performance metrics. Figure 4 presents the results, recorded at training step 100 using Qwen2.5-3B-Instruct with a rollout number of 5. We highlight several observations: (1) *Throughput*: SAPO maintains competitive throughput with minimal deviation ($< 2\%$) from the baselines. (2) *Update and Inference Duration*: The total testing time and the update time increase by 29–33%, which correlates with the generation of longer responses and more search actions—a desirable property for reasoning tasks. (3) *Generation Latency*: SAPO achieves the lowest per-token generation latency (0.087 ms/token), suggesting efficient inference dynamics. (4) *Search Depth*: The method reaches the highest search depth (1.92), which may facilitate multi-hop reasoning. (5) *Response Length*: The average response length increases, reflecting the model's tendency to generate longer reasoning chains. Overall, these results suggest that SAPO introduces moderate update-phase overhead while yielding improved search depth and generation speed at inference time.

5 Related Work

Retrieval-Augmented Generation. Large Language Models (LLMs) (Zhao et al., 2023) have demonstrated strong capabilities in natural language understanding, reasoning, and information

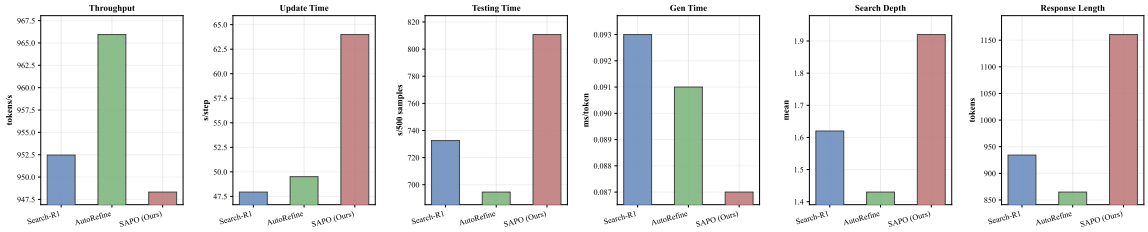

Figure 4: Efficiency analysis. SAPO achieves the greatest search depth (1.92) and fastest per-token generation speed, with a moderate 29–33% increase in update time compared to baselines.

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 employ a "retrieve-then-generate" paradigm, which restricts the LLM's autonomy in determining *when* and *what* to retrieve. This constraint limits their adaptability in settings where iterative information seeking is required.

Search Agents. While traditional search engines (e.g., Google, Bing) facilitate efficient access to ranked web pages, they 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. Concurrently, research has focused on 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). However, a common limitation among 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) provides 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 normalization, 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) have further refined 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). To address this challenge, the present work introduces a sample-based penalty term that enforces a token-level constraint between the updated and reference policies, thereby stabilizing the training of search agents.

6 Conclusion

In this work, we identify Importance Sampling Distribution Drift (ISDD) as a failure mode in GRPO, where importance sampling ratios can decline and lead to model collapse. To address this, we propose SAPO (Search Agent Policy Optimization), which introduces a conditional token-level KL penalty that selectively targets positive-advantage tokens whose IS ratios have drifted below a threshold, while preserving gradient flow. SAPO requires only a single-line modification to the standard GRPO. Experimental results across seven QA benchmarks, multiple model scales from 1.5B to 14B, and two model families show that SAPO achieves the highest overall average performance among the compared methods. These findings suggest that conditional distributional constraints offer a promising direction for stabilizing RL.

Limitations

While SAPO demonstrates improvements across the evaluated QA benchmarks, several limitations warrant discussion.

Task Generalization. In common with most existing TARL methods (Jin et al., 2025; Chen et al., 2025b; Shi et al., 2025a), our experiments primarily focus on retrieval-augmented QA, where the action space is restricted to issuing search queries and synthesizing retrieved documents. The underlying mechanism of SAPO—conditional KL penalization of drifting positive tokens—is designed to be task-agnostic, as it operates on the general IS ratio dynamics rather than task-specific features. That said, different agentic tasks (e.g., mathematical reasoning with calculators, code generation with interpreters, and web navigation with browsers) exhibit distinct action distributions and reward structures. Tasks with denser action vocabularies or longer horizons may benefit from adjusted penalty thresholds τ or coefficients γ. Empirical evaluation on these diverse settings would be valuable to further substantiate the generality of the approach.

Static Corpus. Following the standard evaluation protocol in the TARL literature (Jin et al., 2025; Chen et al., 2025b; Shi et al., 2025a), we evaluate SAPO using a static Wikipedia corpus rather than a live web search engine. This controlled setting enables reproducible comparisons across methods but does not capture challenges inherent to real-time retrieval, such as evolving document collections, adversarial or noisy web content, and variable retrieval latency. We note that since SAPO's penalty mechanism operates on the policy's own token probabilities rather than on retrieval content, it is not inherently tied to the corpus characteristics. Nonetheless, extending the evaluation to dynamic, open-web settings would be a valuable direction for future work.

Single-Tool Constraint. Consistent with the baselines we compare against, the current implementation equips the search agent with a single retrieval tool. Real-world information-seeking tasks may benefit from heterogeneous tool access—combining document retrieval, structured database queries, calculators, and code executors within a single trajectory. Multi-tool environments introduce additional complexity for policy optimization, as the IS ratio dynamics may differ across tool types (e.g., the action entropy for selecting a calculator versus formulating a free-text search query). Since SAPO's conditional penalty is defined at the token level without tool-specific assumptions, it may naturally extend to multi-tool settings, though empirical verification is needed.

Fundamental Credit Assignment. A key factor underlying ISDD—shared by all GRPO-based methods—is that GRPO assigns a uniform advantage to all tokens within a response, conflating tokens that contribute positively to the outcome with those that do not. SAPO mitigates the downstream consequences of this issue—specifically, the suppression of positive tokens—rather than resolving the underlying credit assignment problem itself. We view these as complementary research directions: methods that provide finer-grained, token-level or step-level reward signals (e.g., process reward models or advantage decomposition) could be combined with SAPO to address credit assignment at the source while maintaining training stability.

Reward Design. Following the prevailing practice in TARL (Jin et al., 2025; Shi et al., 2025a), our framework relies on a sparse, outcome-based reward computed from the F1 score between predicted and ground-truth answers. This reward scheme provides no signal about the quality of intermediate reasoning steps or the relevance of individual retrieved documents. Notably, SAPO's improvements are achieved under this sparse reward setting, suggesting that the conditional KL penalty captures useful training signals even without step-level supervision. Incorporating denser reward signals—such as retrieval relevance scores or intermediate reasoning correctness—could offer additional complementary benefits and represents a promising avenue for future work.

Hyperparameter Sensitivity. SAPO introduces two hyperparameters: the KL penalty coefficient γ and the IS ratio threshold τ. Our sensitivity analysis (Figure 2) indicates that performance varies smoothly across a range of values, with τ=1.0 and γ=0.1 performing well across all tested configurations. Moreover, the same hyperparameters transfer effectively across model sizes (1.5B–14B) and families (Qwen, LLaMA) without per-setting tuning. That said, the optimal values may shift for substantially different tasks or training regimes. Developing adaptive mechanisms—for instance, scheduling γ and τ based on running IS ratio statistics—could further improve robustness and reduce manual tuning effort.

Task Complexity and Scale. The evaluated benchmarks, while spanning both single-hop and

multi-hop reasoning, involve relatively constrained question-answering scenarios with short answer spans. More complex tasks such as BrowseComp (Wei et al., 2025), long-horizon planning, or multi-document synthesis may present different challenges, including longer trajectories and sparser reward signals. Moreover, our largest model (14B parameters) is moderate by current standards. However, the monotonic scaling trend observed from 1.5B to 14B (Figure 3) provides encouraging evidence that SAPO's benefits may extend to larger models; verifying this at 70B+ scale is a natural next step.

Theoretical Modeling Assumptions. Proposition 1 assumes that token-level IS ratios follow a log-normal distribution and that ratios across tokens are mutually independent. These assumptions, which are standard in IS analysis (Schulman et al., 2017), facilitate analytical tractability. We emphasize that the key qualitative prediction—that ISDD amplifies exponentially with the number of 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 (e.g., within structured action sequences), which could affect the precise rate of drift but would not alter the direction of the effect. A theoretical treatment that relaxes these assumptions would further strengthen the analytical foundation.

Evaluation Metrics and Linguistic Scope. We adopt Exact Match (EM) and token-level F1 as evaluation metrics, following the standard protocol used by all compared baselines (Jin et al., 2025; Chen et al., 2025b; Shi et al., 2025a; Mei et al., 2025). These surface-level matching criteria do not capture semantic equivalence or paraphrase correctness, and may underestimate performance when the model produces semantically correct but lexically different answers. We note that this limitation applies equally to all methods in our comparison, so relative rankings are unlikely to be affected. Additionally, all experiments are conducted on English-language benchmarks. Since SAPO's mechanism operates on token-level probability ratios without language-specific assumptions, we expect it to transfer to other languages, though empirical verification on multilingual or domain-specific settings (e.g., biomedical, legal) would be informative.

Interaction with Clipping. SAPO augments the standard PPO clipping mechanism with a conditional KL penalty, and both constraints operate simultaneously. Our ablation studies (Table 2) demonstrate that the KL term contributes consistent incremental gains over clipping alone, and the progressive ablation (GRPO $\rightarrow$ +KL $\rightarrow$ +KL$_r$ $\rightarrow$ +KL$_{ra}$) confirms that each component adds value. However, the precise interaction between clipping and the KL penalty has not been exhaustively characterized. Exploring whether the KL penalty could serve as a standalone alternative to clipping, or whether their complementarity varies across training regimes, could inform more principled design choices in future work.

Retriever Dependence. Our experiments use a fixed retriever (E5-base-v2) throughout training and evaluation, consistent with all compared baselines to ensure fair comparison. The quality of the retriever—such as its recall and ranking accuracy—may influence the absolute performance of the search agent. However, since SAPO's conditional KL penalty targets the policy optimization dynamics (IS ratio drift) rather than retrieval quality, its benefits are conceptually independent of the specific retriever. We hypothesize that a weaker retriever would amplify the need for multi-turn refinement, potentially making SAPO's stability gains more pronounced, while a stronger retriever might narrow the gap. Empirical evaluation with diverse retrievers would provide additional evidence of robustness.

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).

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 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.2 Search Agent Prompt

Figure 5 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.3 SAPO Hyperparameters

Our implementation leverages the VeRL framework (Sheng et al., 2025). To ensure a fair comparison, we align our foundational configurations—including Data, Actor, and Rollout settings—with prior baselines such as Search-R1 and SE-Search. As detailed in Table 4, the model is trained with a global batch size of 256 and a learning rate of 1×10^{-6}, utilizing a maximum response length of 2048 tokens. During the rollout phase, we employ a group size of $G = 10$ with sampling parameters set to a temperature of 1.0 and top-$p = 0.95$. Specifically for the proposed SAPO mechanism, we introduce a KL penalty coefficient $\gamma = 0.1$ and an Importance Sampling (IS) ratio threshold $\tau = 1.0$ to regulate policy updates.

A.4 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 parame-

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 5: Prompt template for SAPO.

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}
	KL coefficient β	0.001
	Clip ratio ϵ	0.2
Rollout	Max search actions	5
	Group size G	10
	Temperature	1.0
	Top-p	0.95
KL term	KL penalty coeff. γ	0.1
	IS ratio threshold τ	1.0

Table 4: Hyperparameters used in SAPO.

ters. To isolate the impact of model scale, we keep the retriever, knowledge corpus, and evaluation protocols constant. Table 5 reports the Exact Match (EM) and F1 scores. The results exhibit a positive correlation between model size and performance, consistent with established scaling laws (Kaplan et al., 2020).

A.5 Implementation Simplicity

We augment the standard GRPO surrogate objective with a concise penalty term to enforce token-level constraints on distributional shifts between the current and old policies. As shown in Figure 6, 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 requiring only a single line of code, this modification provides a lightweight regularizer that complements the existing PPO clipping mechanism while enabling finer-grained control over distributional changes during policy updates.

A.6 Efficiency Analysis

To quantify the computational overhead introduced by SAPO, we measure throughput, generation time per token, testing time, and update actor time across five LLMs over the course of training (Figure 7).

Key findings. (1) **Throughput** (Figure 7a) remains stable across training, with larger models achieving higher tokens/s due to better GPU utilization. (2) **Generation time** (b) stabilizes after initial model loading overhead. (3) **Testing time** (c) increases approximately 20–30% as policies generate 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 compared to GRPO.

A.7 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 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 facilitates stable learning of multi-turn search strategies while mitigating distributional drift.

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 5: Scaling trends of Qwen2.5-Instruct with **SAPO** across different model sizes (1.5B to 14B). We report both EM and F1 scores. The best results are marked in bold.

```python
def compute_policy_loss(old_log_prob, log_prob, advantages, eos_mask, cliprange):
    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)

    kl_term_loss = -verl_F.masked_mean(
        torch.log(ratio).masked_fill(ratio > t, 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 + w * kl_term_loss, pg_clipfrac, ppo_kl
```

Figure 6: PyTorch implementation of the SAPO policy loss. The core contribution, the auxiliary KL term, is highlighted in kl_term_loss.

A.8 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 distinct backbones (Qwen2.5-7B-Instruct, Llama-3.2-3B-Instruct, Qwen2.5-1.5B-Instruct, and Qwen2.5-3B-Instruct). The evaluation spans seven diverse 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 (60 steps) to high EM (0.65-0.75). (3) **Multi-hop tasks** (HotpotQA, 2Wiki, Musique, Bamboogle) improve steadily throughout training, with Musique being most challenging (EM about 0.20). (4) **Qwen2.5-7B-Instruct** achieves highest mean EM (0.472), followed by Qwen2.5-7B-Instruct (0.447), Llama-3.2-3B-Instruct (0.438) and Qwen2.5-1.5B-Instruct (0.387). (5) **Bamboogle** exhibits higher volatility but upward trend, demonstrating robustness to adversarial queries. (6) **Cross-model consistency** validates SAPO's generalizability across architectures and scales.

A.9 Case Studies

Tables 6 and 7 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.

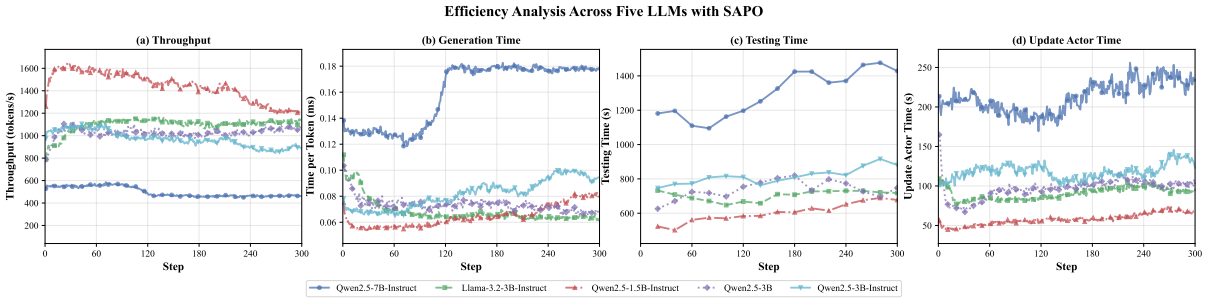

Figure 7: Efficiency analysis of SAPO across five LLMs. (a) Throughput remains stable. (b) Generation time per token. (c) Testing time increases as responses lengthen. (d) Update actor time remains consistent.

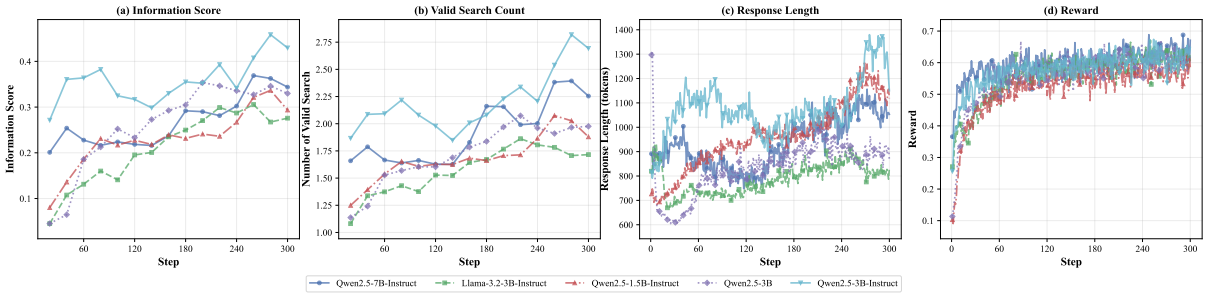

Figure 8: Training dynamics across five LLMs. (a) Information score improves 2-3×. (b) Valid search count grows from 1-1.5 to 2-2.8, showing multi-turn behavior emergence. (c) Response length increases, with Instruct models generating longer reasoning. (d) Reward converges rapidly and stabilizes at 0.50-0.65.

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

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.

<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>

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.

<answer> **Cavalcade Of The West** </answer>

Table 6: Case study illustrating SAPO trajectories from the 2WikiMultihopQA benchmark. Key evidence is extracted from the original searched documents.

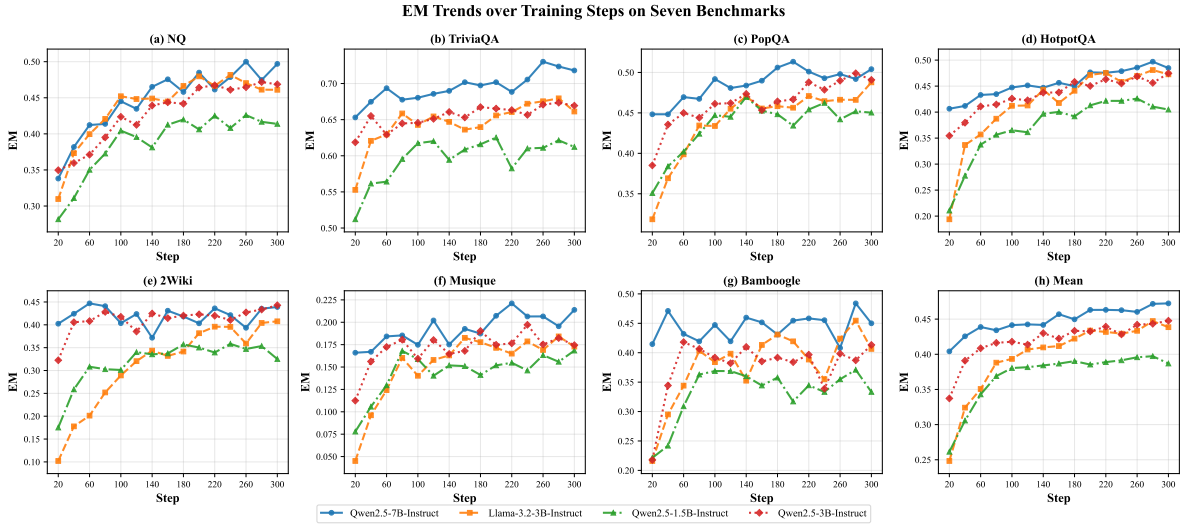

Figure 9: Training trajectories of EM accuracy for SAPO across seven benchmarks.

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