APPO: AGENTIC PROCEDURAL POLICY OPTIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent advances in agentic Reinforcement Learning (RL) have substantially improved the multi-turn tool-use capabilities of large language model agents. However, most existing methods assign credit over coarse heuristic units, such as tool-call boundaries or fixed workflows, making it difficult to identify which intermediate decisions influence downstream outcomes. In this work, we study agentic RL from two perspectives: *where to branch and how to assign credit after branching*. Our pilot analysis shows that high-entropy candidate positions are distributed throughout generated reasoning rather than concentrated only at tool calls, while entropy alone does not reliably order observed branch outcomes. Motivated by these observations, we propose **Agentic Procedural Policy Optimization (APPO)**, which shifts branching and credit assignment from coarse interaction units to fine-grained decision points in the sequence. APPO selects branching locations using a future-aware Branching Score that combines token uncertainty with a discounted suffix policy-shift signal. It further introduces dual-group advantage estimation and procedure-level advantage scaling to provide more targeted credit after branching. We evaluate APPO on 13 benchmarks spanning mathematical reasoning, knowledge-intensive reasoning, and deep search, and find consistent improvements over strong agentic RL baselines by nearly 4 points on aggregate, while retaining efficient tool use and interpretable behavior.

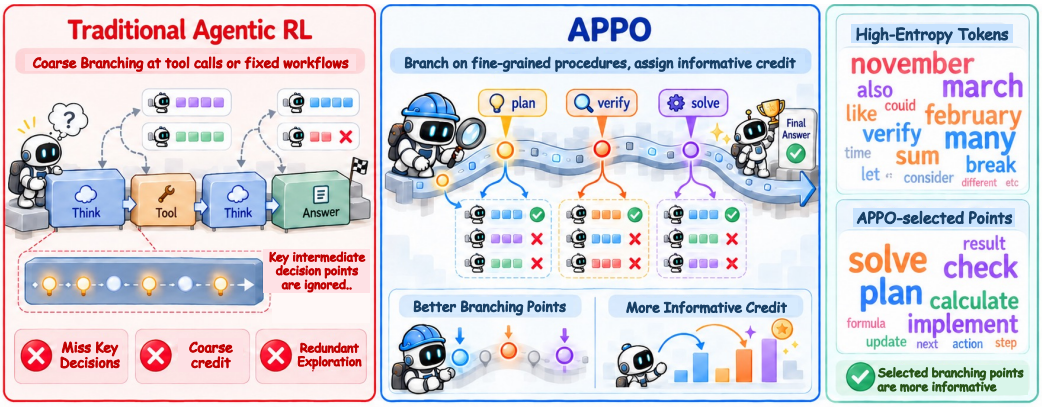


Figure 1: APPO shifts agentic RL from coarse tool-call-level branching to fine-grained procedural branching, enabling more targeted exploration and credit assignment. The right panel illustrates that APPO tends to select more meaningful reasoning decisions than entropy-based branching.

1 INTRODUCTION

Large language models (LLMs) (Wei et al., 2022; Jaech et al., 2024; Team et al., 2025b;a;c; Yang et al., 2025; Li et al., 2025f) have evolved from static text generators (Radford et al., 2019) into autonomous agents (Wang et al., 2025a; Zhang et al., 2025; Dong et al., 2025c; Feng et al., 2025a; Ma et al., 2026b) capable of multi-turn interaction with external environments, enabling strong performance in long-horizon (Lu et al., 2025; Gao et al., 2025) real-world tasks (Hafner, 2021; Cao et al., 2025; Shridhar et al., 2020; Yao et al., 2022a). This progress is largely driven by Reinforcement Learning with Verifiable Rewards (RLVR), which enables policy optimization using sparse outcome-level supervision (Guo et al., 2025a; Schulman et al., 2017; Lee et al., 2023; Rafailov et al., 2023; Shao et al., 2024; Chu et al., 2026). However, this training paradigm introduces a fundamental limitation:

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

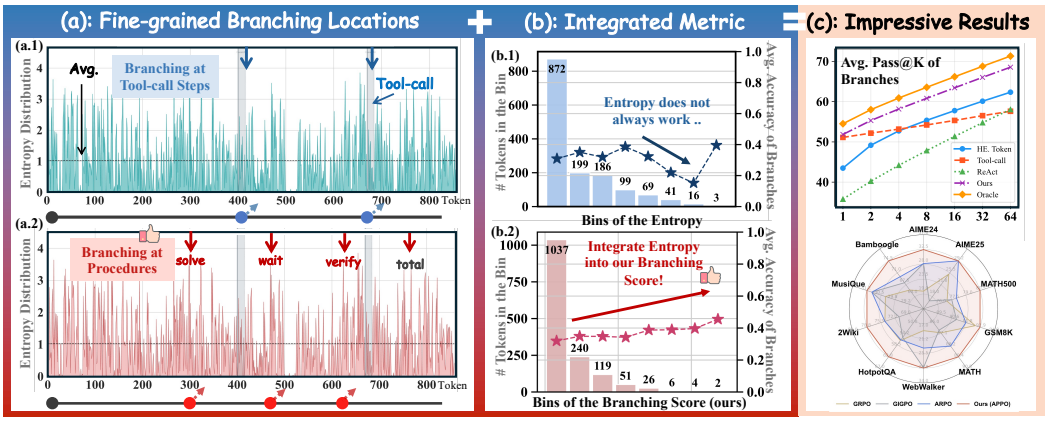


Figure 2: **(a):** Token entropy in tool-integrated rollouts from Tool-Star's (Dong et al., 2025b) 54K dataset. **(b):** Average accuracy of valid, completed branches binned by entropy or Branching Score (BS). **(c):** Pass@k of rollouts resampled by different criteria (the oracle uses observed branch outcomes only as an upper-bound reference). The pilot uses Qwen2.5-7B-Instruct, 1,000 prompts, $N = 4$ initial rollouts per prompt, 1,500 sampled candidate positions and top-$B = 3$ branching.

feedback is only provided at the trajectory level, making it difficult to attribute success or failure to specific intermediate decisions. As a result, each trajectory provides only a coarse learning signal, leading to inefficient credit assignment and unstable policy improvement (Qian et al., 2025b).

To address this limitation, existing approaches restructure agent rollouts to extract more informative credit signals under limited rollout budgets. A common strategy is to expand trajectories from intermediate locations, constructing multiple candidate branches and assigning credit according to outcome differences across rollouts (Yao et al., 2022b; Zhou et al., 2023; Ji et al., 2025; Zhao et al., 2026; Shen et al., 2025). The intuition is that, instead of repeatedly sampling full trajectories, one can branch around uncertain positions and compare alternative continuations to identify which local decisions lead to better final outcomes. Depending on the branching granularity, there are workflow-level and tool-call-level branching (Hou et al., 2025; Dong et al., 2025c;a).

While such designs improve rollout efficiency and partially densify supervision from sparse outcome rewards, they still rely on coarse-grained units for credit assignment. In particular, they often compress the entire non-tool-call process into `<thinking>` tags (Dong et al., 2025c;a) or rely on fixed workflows (Yao et al., 2022b; Wang et al., 2023; Shinn et al., 2023), thereby overlooking the procedural knowledge embedded within the thinking content (Guo et al., 2025b; Wang et al., 2025b; Wu et al., 2026). In practice, successful long-horizon agent reasoning is often shaped not by entire thinking blocks or workflow stages, but by a small number of critical *decision points* where alternative continuations lead to substantially different downstream outcomes (Wang et al., 2025b). These decision points are latent positions in the trajectory, instantiated as tokens in the generated sequence, and identified by their role in inducing divergence in subsequent reasoning paths. We use **procedures** to denote the procedural reasoning patterns organized around such high-impact decision points. Although some individual procedures, such as `plan` (Wang et al., 2023) and `reflect` (Shinn et al., 2023), have drawn attention in training-free prompt engineering (Jin et al., 2025a; Wang et al., 2026), their collective role in online agentic RL remains underexplored.

To further investigate how these procedures relate to reasoning accuracy and failure, we conduct a pilot study summarized in Figure 2. We analyze (a) where high-entropy candidate positions occur and (b) the accuracy of valid, completed branches resampled from candidates under a shared prompt, candidate-pool, and continuation-budget protocol. The results show that high-entropy positions occur throughout reasoning rather than only at tool-call boundaries, while the integrated BS criterion provides a more informative ordering of observed branch outcomes than entropy alone.

These findings highlight the necessity of attending to procedural information within the thinking process beyond tool calls, while underscoring the need for a more principled design of branching criteria to enable finer-grained credit assignment across procedures.

Motivated by these findings, we propose Agentic Procedural Policy Optimization (APPO), an agentic RL algorithm that redefines the unit of credit assignment from coarse-grained heuristic units to fine-grained procedures. Given an initial rollout, APPO extends branching-point selection from tool-call boundaries to the entire sequence, and chooses branching tokens using a future-aware Branching Score (BS). Beyond token entropy, BS incorporates the discounted suffix policy-shift signal to identify high-value procedural decisions. Building on this design, we introduce a procedure-level advantage scaling term based on Ω. We evaluate APPO on 13 challenging benchmarks spanning deep information seeking, knowledge-intensive reasoning, and computational problem solving, and show consistent gains over strong baselines (Figure 2(c)). Our contributions are threefold:

- Our preliminary study demonstrates the critical role of *procedures* in agent reasoning. By moving rollout branching from workflows or tool-call boundaries to procedures, it exposes finer-grained structure within the thinking process and enables more informative intermediate supervision.
- Motivated by these findings, we propose APPO, an agentic RL algorithm that shifts credit assignment from coarse-grained heuristic units to fine-grained procedures. Its Branching Score combines token entropy with the policy-induced likelihood gain of subsequent continuations, enabling the selection of high-value procedures for branching and procedure-level advantage scaling.
- Extensive experiments validate the effectiveness of our method, which outperforms existing approaches by approximately 3 points across 13 benchmarks, while achieving comparable tool-calls and maintaining interpretability.

2 Related Work

Agentic Reinforcement Learning. Reinforcement Learning (Kaufmann et al., 2023; Schulman et al., 2017; Rafailov et al., 2023; Guo et al., 2025a) has proven to be effective in endowing agents with long-horizon complex reasoning and acting capabilities. Beginning from the actor-critic-based PPO (Schulman et al., 2017), a wide range of studies aim to devise more efficient policy variants, or refine the scale of policy gradients, advantages or regularization terms, like GRPO (Guo et al., 2025a), DAPO (Yu et al., 2025), GSPO (Zheng et al., 2025), GPG (Chu et al., 2026), Dr. GRPO (Liu et al., 2025). More recent research (Feng et al., 2025b; Zhai et al., 2025; Lu et al., 2026) focus on the distinct applications of RL in agent tasks, such as fine-grained credit assignment for actions (Feng et al., 2025b; Dong et al., 2025c;a), agent self-evolution (Wu et al., 2025c; Zhai et al., 2025; Xia et al., 2026) and internalization of agent skills (Lu et al., 2026; Zhang et al., 2026).

Tree-based RL. Tree-based RL can be roughly divided into the following categories: (i) Offline Training (Feng et al., 2023; Lai et al., 2024; Li et al., 2025a; Xie et al., 2024), where branches split from the same point are stored as offline preference data and later utilized in DPO (Rafailov et al., 2023). For example, MCTS-DPO (Xie et al., 2024) compares the estimated reward of candidate nodes and separates them into chosen / rejected samples; SPORT (Li et al., 2025a) leverages step-wise sampling and automatic verification to formulate step-wise preference data for agent training. (ii) Online Training (Hou et al., 2025; Ji et al., 2025; Dong et al., 2025c;a; Zhao et al., 2026), where branches are extended from rollouts in each step and used to calculate group-relative advantages. For example, Tree-GRPO (Ji et al., 2025) randomly selects think-action steps for branching, while ARPO (Dong et al., 2025c) identifies high-entropy tokens following each tool call and conducts resampling accordingly. (iii) Test-Time Scaling (Wang et al., 2026; Snell et al., 2024; Wu et al., 2024; Welleck et al., 2024; Muennighoff et al., 2025), where the model generates multiple branches to elevate average inference performance. FIPO (Ma et al., 2026a) uses a future-KL signal to reweight token advantages in long-chain-of-thought RL. *As extension of these lines of studies, APPO treats procedures as finer units of rollout branching in on-policy agentic RL and elicits the reasoning process accordingly.*

3 APPO: Agentic Procedural Policy Optimization

3.1 Preliminaries

Agentic Reinforcement Learning. We consider a standard agentic RL setting (Gou et al., 2023; Li et al., 2025e; Wu et al., 2025b; Dong et al., 2025b; Su et al., 2025a; Qian et al., 2025a), where an agent receives a task $x \in \mathcal{X}$ and interacts with a toolset T. The rollout consists of T_a interleaved

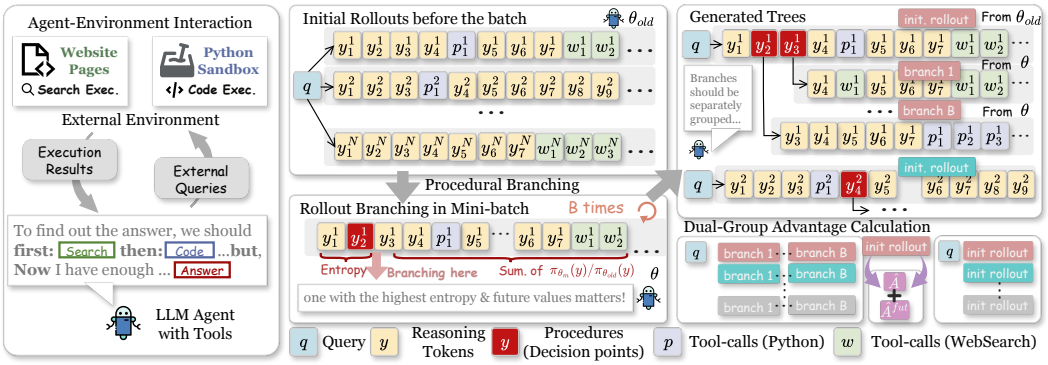


Figure 3: Overview of APPO. The agent first interacts with the environment to generate initial rollouts for each batch. During mini-batch training, APPO identifies fine-grained decision points using the Branching Score (BS), which combines token entropy with future-aware likelihood gains, and resamples continuations from these positions rather than fixed tool-call boundaries. The resulting branches and initial rollouts are then used for dual-group advantage estimation.

thinking and tool-call steps, followed by T_b answer generation steps:

$$P_\theta(\mathcal{G}, y|x, T) = \prod_{t=1}^{T_a}[\pi_\theta(\mathcal{O}_t|\mathcal{G}_{<t}, x; T)P_{env}(\mathcal{G}_t|\mathcal{O}_t, \mathcal{G}_{<t})] \prod_{t=1}^{T_b} \pi_\theta(\mathcal{O}_t|y_{<t}, \mathcal{G}, x; T) \tag{1}$$

where P_{env} denotes the environment transition/observation probability. If the agent does not call an external tool in $\mathcal{O}_t$, then $\mathcal{O}_t$ equals the overall output $\mathcal{G}_t$ of step t. Based on the rollouts, the training objective of Agentic Reinforcement Learning can be formulated as follows:

$$\max_{\pi_\theta} \mathbb{E}_{x\sim\mathcal{X}, y\sim\pi_\theta(\cdot|x;T)}[r_\phi(x, y)] - \beta\, \mathbb{D}_{\mathrm{KL}}[\pi_\theta(y|x; T)||\pi_{\mathrm{ref}}(y|x; T)] \tag{2}$$

where π_θ and π_{ref} are the policy and reference models, respectively. r is the task reward, and $\mathbb{D}_{\mathrm{KL}}$ denotes KL divergence. In practice, rewards are converted into advantages $\hat{A}(\cdot)$ through group-level (Shao et al., 2024; Guo et al., 2025a; Liu et al., 2025), sequence-level (Zheng et al., 2025), token-level (Yu et al., 2025), or value-function-based estimators (Schulman et al., 2017).

Token Entropy. A common branching strategy (Dong et al., 2025c;a; Hou et al., 2025) selects tokens with the top-k entropy:

$$H_t = -\sum_{j=1}^{|\mathcal{V}|} p_{t,j} \log p_{t,j}, \quad \boldsymbol{p}_t = \pi_\theta(\cdot|\mathcal{G}_{<t}, z; T) = \mathrm{Softmax}(\boldsymbol{z}_t/\tau) \tag{3}$$

where $|\mathcal{V}|$ is the vocabulary size, τ is the temperature, and $\boldsymbol{z}_t$ are logits. Although entropy reflects local uncertainty, it does not indicate whether a token corresponds to a decision point that changes downstream reasoning. Thus, high-entropy tokens may arise from lexical uncertainty rather than task-relevant procedural choices. APPO addresses this by identifying tokens that instantiate latent decision points, i.e., positions where alternative continuations induce divergent reasoning paths, and uses them to construct procedures for more targeted branching and credit assignment.

3.2 Procedural Rollout Branching

In this section, we introduce the APPO algorithm in detail, designed to guide LLMs in exploring fine-grained procedural-level behaviors under synthetic branching criteria, as illustrated in Figure 3.

► **Initialization.** Given input x and a global rollout budget M, the model first generates N full rollouts $\mathcal{T} = \{\mathcal{H}_n|\mathcal{H}_n \sim \pi_\theta(\cdot|x)\}^N$, where n indexes a rollout and i indexes a token within that rollout. These rollouts serve as the roots of N independent trees.

► **Sampling.** For each token $\mathcal{H}_{n,i}$ in rollout $\mathcal{H}_n$, APPO measures its future-aware influence using the accumulated decayed importance sampling ratio (π_{old} here is the π_θ in initialization):

$$\Omega_{n,i} = \exp\Big(\sum_{i'\geq i} \gamma^{i'-i} \log \rho_{i'}(\theta)\Big), \quad \rho_{i'}(\theta) = \frac{\pi_\theta(\mathcal{H}_{n,i'}|\mathcal{H}_{n,<i'}, x; T)}{\pi_{\mathrm{old}}(\mathcal{H}_{n,i'}|\mathcal{H}_{n,<i'}, x; T)}, \tag{4}$$

where π_{old} denotes the policy used to generate the initial rollouts. We call $\Omega_{n,i}$ the discounted suffix policy-shift signal. It captures how the realized suffix is reweighted by the current policy relative to the behavior policy. The discount limits the effective suffix horizon, and clipping limits the effect of extreme ratios. We combine this future-aware signal with token entropy to define the Branching Score:

$$\text{BS}_{n,i} = \mathcal{Z}(\text{clip}(\Omega_{n,i}, 1-\epsilon', 1+\epsilon'); \mathcal{H}_n) \cdot \mathcal{Z}(H_{n,i}; \mathcal{H}_n), \tag{5}$$

where $\mathcal{Z}(\cdot; \mathcal{H}_n)$ denotes z-score normalization within rollout $\mathcal{H}_n$. Entropy captures local uncertainty, while $\Omega_{n,i}$ captures downstream policy shift. Their product integrates local uncertainty and future influence to select tokens that are both uncertain and consequential.

For each rollout $\mathcal{H}_n$, we select the top-B tokens according to $\text{BS}_{n,i}$ and denote them as $\mathcal{B}_n$ ($|\mathcal{B}_n| = B$). These tokens instantiate latent decision points. We then resample continuations from each $\mathcal{H}_{n,i} \in \mathcal{B}_n$ to generate new branches $\mathcal{H}_n^{new}$, and update the rollout tree as $\mathcal{H}_n \leftarrow \mathcal{H}_n \cup \mathcal{H}_n^{new}$.

► **Termination.** Tree expansion stops when either the remaining budget $M - N$ is exhausted or no further branching is performed. It is worth noting that we sample uniformly across all trees. Assume the total number of generation loops is L, then for $L = 1$ (i.e., the source of branching is restricted to the initial rollouts), there is $(B+1) \cdot N = M$. With a fixed total budget, increasing N allows us to obtain rollouts with a more diverse initial distribution; conversely, increasing B and L enables more branching on a single tree, thereby providing finer-grained credit assignment. We will conduct detailed ablation studies of these coefficients in subsequent sections and Appendix E.

3.3 Procedural Advantage Estimation and Policy Optimization

Most policy optimization methods (Ji et al., 2025; Xin et al., 2025) rely on group-level advantages, which provide limited credit assignment at intermediate decision points. APPO instead assigns credit to procedure-level decisions by using branches as auxiliary contrastive signals. Unlike most Tree RL methods, our branches are generated by the current mini-batch policy π_θ rather than π_{old}. Since gradients are not propagated through all branches, we use them only to compute rewards and advantages.

To avoid bias from mixing rollouts generated by different policies, APPO computes group-relative advantages separately for initial rollouts $\mathcal{T}_{init}$ and branches $\mathcal{T}_{branch}$:

$$\hat{A}_{n,i}^{\text{base}} = \text{avg}\left\{ \frac{R_n - \text{mean}(\{R_{n'} \mid \mathcal{H}_{n'} \in \mathcal{T}_*\})}{\text{std}(\{R_{n'} \mid \mathcal{H}_{n'} \in \mathcal{T}_*\})} \;\middle|\; \mathcal{T}_* \in \{\mathcal{T}_{init}, \mathcal{T}_{branch}\} \right\} \tag{6}$$

where R_n is the task reward. Since generated tokens serve as observable instantiations of latent decision points, token-level advantages can be viewed as localized credit assigned to the corresponding procedural decisions. Furthermore, inspired by recent studies (Huang et al., 2026; Meng et al., 2026), we argue that critical procedures offer more "sparse and critical" subsequences; these procedures serve as the turning point of the rollout and more likely to induce higher policy-induced differences over continuations. We then take the similar design of Ω to formulate an extra advantage term, which further emphasizes decisions with stronger downstream influence by assigning larger credits:

$$\hat{A}_{n,i}^{\text{fut}} = \text{clip}_{\epsilon'}\left(\exp\left(\sum\nolimits_{i' \geq i} \gamma^{i'-i} \log \rho_{i'}(\theta)\right)\right), \quad \rho_{i'}(\theta) = \frac{\pi_\theta(\mathcal{H}_{n,i'} | \mathcal{H}_{n,<i'}, x; T)}{\pi_{\text{old}}(\mathcal{H}_{n,i'} | \mathcal{H}_{n,<i'}, x; T)} \tag{7}$$

where $\text{clip}_{\epsilon'}$ clips the value into $[1-\epsilon', 1+\epsilon']$ and ϵ' is the future-signal clipping threshold. The final advantage is: $\hat{A}_{n,i} = \hat{A}_{n,i}^{\text{base}}(1 + b \cdot \hat{A}_{n,i}^{\text{fut}})$, where b controls the contribution of the future-aware term. Finally, APPO optimizes:

$$J(\theta) = \mathbb{E}_{\substack{x \sim \mathcal{X} \\ \mathcal{H} \sim \pi_{\text{old}}(\cdot|x)}} \left[\frac{1}{M} \sum_{n=1}^{N} \frac{1}{|\mathcal{H}_n|} \sum_{i=1}^{|\mathcal{H}_n|} \min\left(\rho_{n,i}(\theta)\hat{A}_{n,i}, \text{clip}_\epsilon(\rho_{n,i}(\theta))\hat{A}_{n,i}\right) - \beta \mathbb{D}_{\text{KL}}(\pi_\theta || \pi_{\text{ref}}) \right] \tag{8}$$

where clip_ϵ clips the term into $[1-\epsilon, 1+\epsilon]$. π_{ref} and π_{old} denote the reference and behavior policies, respectively, and β controls the KL regularization strength. Branches sampled from π_θ are not directly optimized; they provide auxiliary procedural signals for advantage estimation.

Table 1: Performance comparison between APPO and others on 10 challenging Deep Reasoning datasets. Notably the best and suboptimal results are in **bold** and underlined respectively.

Method	Mathematical Reasoning					Knowledge-Intensive Reasoning					Avg.
	AIME24	AIME25	MATH500	GSM8K	MATH	WebWalker	HotpotQA	2Wiki	Musique	Bamboogle	
Backbone: Llama3.1-8B-Instruct											
Training-Free Method											
Zero-shot	3.3	0.0	43.3	81.4	60.6	3.0	24.3	24.6	10.4	40.0	28.8
TIR Prompting	3.3	3.3	39.4	73.8	58.2	15.0	48.5	47.5	15.5	58.4	36.3
Classic RL Method											
GRPO	13.3	13.3	62.4	87.4	79.2	26.5	57.8	71.8	31.0	68.2	51.1
Reinforce++	13.3	**16.7**	61.4	87.0	77.2	27.5	57.1	71.6	29.9	69.1	51.1
DAPO	16.7	13.3	61.2	87.4	76.4	25.5	56.6	70.3	29.2	67.3	50.4
GPPO	16.7	6.7	61.8	86.6	79.4	27.5	61.8	72.8	29.8	71.9	51.5
CISPO	13.3	6.7	62.2	87.0	78.2	26.0	57.3	75.6	32.2	71.8	51.0
Agentic RL Method											
GIGPO	20.0	13.3	62.4	87.4	77.2	31.5	61.8	74.6	31.8	72.1	53.2 Δ_{base}
ARPO	23.3	**16.7**	64.6	88.0	80.2	30.5	**65.4**	75.5	34.8	72.8	55.3 +3.9%
APPO (Ours)	**30.0**	**16.7**	**69.4**	**88.6**	**81.4**	**32.0**	**65.4**	**79.8**	**35.2**	**76.8**	**57.4** +7.9%
Backbone: Qwen2.5-7B-Instruct											
Training-Free Method											
Zero-shot	10.0	10.0	70.6	90.2	82.0	2.0	12.2	12.6	6.6	24.0	32.0
TIR Prompting	6.7	10.0	68.2	64.6	78.2	15.5	14.8	18.3	9.5	23.6	30.9
Classic RL Method											
GRPO	23.3	26.7	78.0	92.8	87.8	22.0	59.0	76.1	30.6	68.4	56.5
Reinforce++	26.7	23.3	78.0	92.2	88.8	26.0	55.1	68.9	25.2	64.9	54.9
DAPO	20.0	23.3	80.4	91.0	88.8	24.0	57.7	68.4	28.6	65.5	54.8
GPPO	26.7	23.3	80.4	91.0	88.8	31.0	60.7	74.2	**31.5**	72.4	58.0
CISPO	26.7	**30.0**	77.8	91.4	86.2	29.0	59.3	72.1	29.1	70.2	57.2
Agentic RL Method											
GIGPO	30.0	20.0	78.4	91.6	87.6	30.5	58.1	73.5	31.1	70.1	57.1 Δ_{base}
ARPO	30.0	**30.0**	78.8	92.2	88.8	26.0	58.8	76.1	31.1	71.5	58.3 +2.1%
APPO (Ours)	**36.7**	**30.0**	**81.0**	**93.2**	**90.0**	**33.5**	**66.5**	**81.5**	**31.5**	**77.6**	**62.2** +8.9%

4 EXPERIMENTS

4.1 EXPERIMENT SETUP

Datasets. Following (Dong et al., 2025a;c), we employ the following three kinds of benchmarks to comprehensively evaluate the model's capability in long-term complex tool usage: *(1) Mathematical Reasoning,* including the typical math problems like GSM8K (Cobbe et al., 2021) and MATH (Hendrycks et al., 2021); and competitive math problem like AIME24 (Zhang & Math-AI, 2024), AIME25 (Zhang & Math-AI, 2025) and MATH500 (Lightman et al., 2023); *(2) Knowledge-Intensive Reasoning,* including four multi-hop question-answering problems based on Wikipedia, i.e., HotpotQA (Yang et al., 2018), 2WikiMultihopQA (Ho et al., 2020), Musique (Trivedi et al., 2022) and Bamboogle (Press et al., 2023). We also include the web multi-hop task WebWalker (Wu et al., 2025a); *(3) Deep Search,* including the challenging General AI Assistant (GAIA) (Mialon et al., 2023), Humanity's Last Exam (Phan et al., 2025), and information retrieval tasks like WebWalker (Wu et al., 2025a) and Xbench (Chen et al., 2025b).

Baselines. Four kinds of baselines are selected for comprehensive comparison: *(1) Vanilla RL Method,* including GRPO (Guo et al., 2025a), Reinforce++ (Hu et al., 2025), DAPO (Yu et al., 2025), GPPO (Su et al., 2025b) and CISPO (Chen et al., 2025a); *(2) Agentic RL Method*, including GIGPO (Feng et al., 2025b) and ARPO (Dong et al., 2025c); *(3) LLM Backbones,* where we adopt the instruct version of Llama3.1-8B (Dubey et al., 2024) and Qwen2.5 (Hui et al., 2024) as the generalized backbones; we also report the results of QwQ (Team, 2024), DeepSeek-R1 (Guo et al., 2025a), GPT-4o (Hurst et al., 2024), o1-preview (Hurst et al., 2024) and Qwen3-32B (Yang et al., 2025); *(4) Search Agents and others,* where we adopt the Search-o1 (Li et al., 2025b) and Webthinker (Li et al., 2025c); We also employ the RAG (Lewis et al., 2020) and ReAct (Yao et al., 2022b) framework.

Metrics. We employ the F1-score to evaluate the performance of models over four question-answering tasks. For other tasks, we employ LLM-as-a-Judge and deploy a Qwen2.5-72B-Instruct model with vLLM for correctness calculation. We use pass@1 over the LLM-judged results.

Implementation Details. Following (Ma et al., 2026a), we implement the decay rate in procedure advantage as $\gamma = 2^{-\frac{1}{\tau}}$ and keeps $\tau = 32$. The overall methodology is implemented on the VeRL framework (Sheng et al., 2025) where tool-call results are detached from gradient computation. The

Table 2: Performance comparison between APPO and others on 4 challenging DeepSearch datasets. Results with gray color are from larger / closed-source models and only for reference. Notably the best and suboptimal results are in **bold** and underlined respectively.

Method	General AI Assistant				WebWalkerQA				Humanity's Last Exam				Xbench
	Lv.1	Lv.2	Lv.3	Avg.	Easy	Med.	Hard	Avg.	NS	CE	SF	Avg.	Avg.
Qwen3-32B-thinking	26.2	12.1	0.0	14.9	6.9	1.1	2.9	3.1	14.6	9.8	8.4	12.6	14.0
DeepSeek-R1-32B	21.5	13.6	0.0	14.2	7.5	1.4	4.2	3.8	6.6	5.1	6.5	6.4	10.0
QwQ-32B	30.9	6.5	5.2	18.9	7.5	2.1	4.6	4.3	11.5	7.3	5.2	9.6	10.7
GPT-4o	23.1	15.4	8.3	17.5	6.7	6.0	4.2	5.5	2.7	1.2	3.2	2.6	18.0
DeepSeek-R1-671B	40.5	21.2	5.2	25.2	5.0	11.8	11.3	10.0	8.5	8.1	9.3	8.6	32.7
o1-preview	/	/	/	/	11.9	10.4	7.9	9.9	12.9	8.1	6.6	11.1	/
Single-Enhanced Method (Qwen3-8B)													
Vanilla RAG	28.2	15.4	**16.7**	20.4	8.9	10.7	9.9	10.0	5.1	1.6	12.9	5.8	8.0
Search-o1	35.9	15.4	0.0	21.4	6.7	15.5	9.7	11.5	7.6	2.7	5.3	6.4	10.0
WebThinker	43.6	11.5	0.0	22.3	6.7	13.1	16.9	13.0	7.3	4.0	6.3	6.6	13.0
ReAct	35.9	17.3	8.3	23.3	8.9	16.7	18.3	15.5	4.2	4.0	6.3	4.6	16.0
RL-based Method (Qwen3-8B)													
Qwen3-8B	28.1	15.4	**16.7**	20.4	0.0	2.4	2.8	2.0	3.9	2.7	8.4	4.6	9.0
+GRPO	48.7	25.0	8.3	32.0	28.9	32.1	**28.2**	30.0	7.9	4.0	10.5	7.8	20.0
+ARPO	53.9	32.7	**16.7**	38.8	31.1	35.7	**28.2**	32.0	7.3	6.7	**15.8**	8.8	25.0
+APPO (Ours)	**59.0**	**36.5**	**16.7**	**42.7**	**37.5**	**36.4**	**28.2**	**33.8**	**14.7**	**8.3**	13.0	**13.4**	**28.0**
Single-Enhanced Method (Qwen3-14B)													
Vanilla RAG	38.5	19.2	8.3	25.2	17.8	13.1	11.3	13.5	5.5	6.3	9.4	6.0	15.0
Search-o1	48.7	23.1	0.0	30.1	11.1	21.4	16.9	17.5	6.4	4.0	10.5	6.8	21.0
WebThinker	48.7	26.9	8.3	33.0	13.3	23.8	18.3	19.5	7.0	4.0	9.5	7.0	23.0
ReAct	48.7	25.0	8.3	32.0	11.1	20.2	12.7	15.5	5.8	5.3	10.5	6.6	20.0
RL-based Method (Qwen3-14B)													
Qwen3-14B	33.3	13.5	0.0	19.4	6.7	2.4	4.2	4.0	5.5	6.7	11.6	6.8	14.0
+GRPO	51.3	34.6	0.0	36.9	35.6	42.9	35.2	38.5	7.9	6.7	12.6	8.6	27.0
+ARPO	56.4	40.4	16.7	43.7	40.0	44.1	36.6	40.5	10.3	10.7	13.7	10.0	32.0
+APPO (Ours)	**61.5**	**42.3**	**16.7**	**46.6**	**43.0**	**48.6**	**37.6**	**43.4**	**11.5**	**12.8**	**14.5**	**12.3**	**34.0**

batch-size is set to 128 and a PPO mini-batchsize of 16, resulting in 8 gradient updates in one step. We also set the coefficient $\beta = 0$ to stabilize training. We train APPO in both reasoning and search tasks following ARPO (Dong et al., 2025c), where models are separately trained for 2 and 5 epochs. The search results are the top-10 snippets retrieved by Bing search engine following (Jin et al., 2025b; Li et al., 2025d; Dong et al., 2025b) and python code is executed in a sandbox environment. To ensure the model possesses basic tool-use capabilities, we directly adopt the SFT pipeline from ARPO (Dong et al., 2025c). More implementation details are attached to Appendix C.

4.2 Main Results

Results on Mathematical Reasoning and Knowledge-Intensive Reasoning. As shown in Table 1, APPO achieves the best reported aggregate across both task groups, although individual datasets include ties and heterogeneous gains. We make the following observations: (i) On mathematical reasoning tasks, APPO surpasses all baselines, improving over the previous best agentic RL method by an average of 2.45 points. This suggests that branching at high-impact decision points, rather than fixed turn-level boundaries, enables the model to better explore and reinforce critical deductive paths for complex problem solving. (ii) On knowledge-intensive reasoning tasks, APPO ranks first on nearly all datasets, showing strong performance on multi-hop information synthesis tasks. This indicates that identifying and assigning higher credit to high-impact decision points helps the model strengthen useful procedural reasoning patterns, such as planning, reflection, and verification, leading to consistent gains various RL methodsz.

Results on DeepSearch Tasks. The results in Table 2 show strong improvements for APPO in complex deep-search scenarios. While even powerful large-scale closed-source models achieve unsatisfactory performance, APPO obtains the best reported aggregate at both the 8B and 14B model scales and consistently outperforms strong baselines such as ARPO. For instance, on the GAIA benchmark, APPO achieves scores of 42.7 and 46.6 with Qwen3-8B and Qwen3-14B, respectively, while also improving performance on HLE. These gains are particularly notable given that deep search tasks involve long horizons and intricate tool-use patterns. By combining downstream policy-shift information with token uncertainty, APPO focuses branching on high-impact positions and avoids

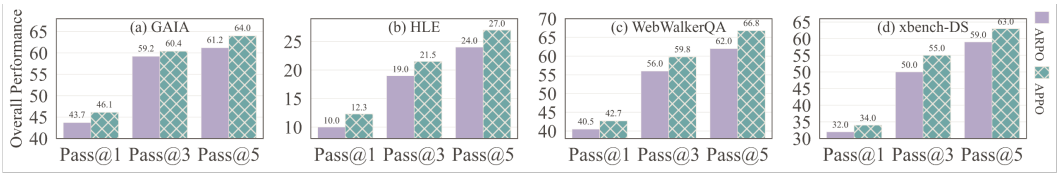


Figure 4: Analysis of Pass@k for $k = 1, \ldots, 5$ of ARPO and APPO on four datasets.

Table 3: Analysis on branching config ($L = 1$).

Method	Web.	Hotpot.	2Wiki	Musiq.	Bamb.	Avg.
ARPO ($M = 16$)	26.0	58.8	76.1	31.1	71.5	52.7
$M = 4$						
APPO ($N = 2, B = 1$)	25.4	60.4	77.0	30.6	70.2	52.7
APPO ($N = 1, B = 3$)	26.7	63.8	77.6	31.0	70.6	53.9
$M = 8$						
APPO ($N = 2, B = 3$)	28.0	65.0	79.4	30.8	73.3	55.3
$M = 16$						
APPO ($N = 4, B = 3$)	**33.5**	**66.5**	**81.5**	**31.5**	77.6	**58.1**
APPO ($N = 8, B = 1$)	31.8	65.8	81.2	31.0	**79.5**	57.9
APPO ($N = 2, B = 7$)	29.2	65.5	79.0	31.2	75.7	56.1

Table 4: Ablation study on components.

Method	Web.	Hotpot.	2Wiki	Musiq.	Bamb.	Avg.
Llama3.1-8B-Instruct						
APPO	**32.0**	**65.5**	**79.8**	**35.2**	**76.8**	**57.9**
BS→Ent.	31.4	65.4	78.0	34.8	75.5	57.0
w/o $\hat{A}^{\text{fut}}$	29.8	64.8	78.6	35.0	74.5	56.5
w/o Dual-group	30.9	65.0	76.4	35.5	76.2	56.8
Qwen2.5-7B-Instruct						
APPO	**33.5**	**66.5**	**81.5**	**31.5**	**77.6**	**58.1**
BS→Ent.	29.4	64.8	80.5	31.0	75.7	56.3
w/o $\hat{A}^{\text{fut}}$	30.3	60.0	79.9	30.6	72.6	54.7
w/o Dual-group	33.0	63.8	75.8	30.6	76.6	56.0

excessive exploration of low-value high-entropy tokens. Consequently, by prioritizing structurally meaningful decision points, APPO enables more targeted exploration and improves the reliability of result-oriented agentic reasoning.

4.3 Scaling Analysis

Pass@K Analysis. Figure 4 shows that the benefit of APPO extends beyond improving the single best trajectory and instead improves the overall distribution of candidate trajectories, as reflected in the Pass@K results. Across all four datasets, the advantage of APPO is consistently preserved and further enlarged as k increases, indicating that the method improves not only top-1 correctness but also the diversity of valid reasoning paths in the sampling space. On GAIA, Qwen3-14B improves from 43.7 to 46.1 at Pass@1, with the gap further expanding at Pass@5 from 61.2 to 64.0. A similar trend appears on WebWalkerQA , where the gain increases from 40.5 to 42.7 at Pass@1 and from 62.0 to 66.8 at Pass@5. These results suggest that APPO promotes exploration over structurally distinct reasoning trajectories rather than local token-level variations, leading to a broader set of high-quality candidate solutions and significant improvement of pass@K metrics.

Branching Configuration Analysis. Table 3 shows that the effectiveness of APPO depends not only on whether branching is introduced, but also on how the rollout budget is allocated between the number of initial trees N and the number of selected tokens B. Under the same total budget, balanced configurations consistently outperform both extremes. For example, when $M = 16$, APPO with $(N = 4, B = 3)$ achieves the best average score of 58.1, exceeding both $(N = 8, B = 1)$ with 57.9 and $(N = 2, B = 7)$ with 56.1. Increasing N improves the diversity of initial trajectories but leaves less budget for expanding high-impact decision points. In contrast, increasing B enables deeper exploration around these decision points but concentrates the budget on fewer initial paths, reducing global coverage. APPO performs best in the middle regime, since decision-point-guided branching is most effective when the model first explores diverse roots and then expands internal decisions.

Ablation on Components. Table 4 verifies that all components of APPO contribute to the final gains, with complementary rather than redundant effects. Specifically, replacing the Branching Score (BS) with entropy leads to consistent performance drops of 1.7 and 0.9 points on the two backbones. This suggests that pure entropy fails to prioritize high-impact decision points that are likely to alter downstream reasoning paths. Disabling dual-group advantage estimation also yields a clear degradation, since initial rollouts and branches are generated from different policy distributions and should be compared within their respective groups. Finally, removing $\hat{A}^{\text{fut}}$ causes an even larger drop, especially on Qwen2.5-7B, where the average score decreases from 58.1 to 54.7. Overall, BS improves where the model explores, while dual-group advantage estimation and $\hat{A}^{\text{fut}}$ ensure that branches are compared under appropriate distributions and assigned more fine-grained credit.

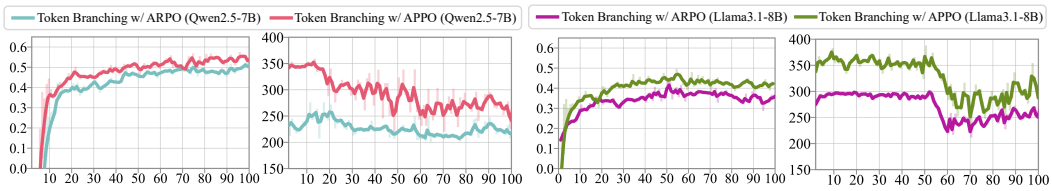


Figure 5: **The training dynamics** of pure-token branching and APPO's procedural guided branching.

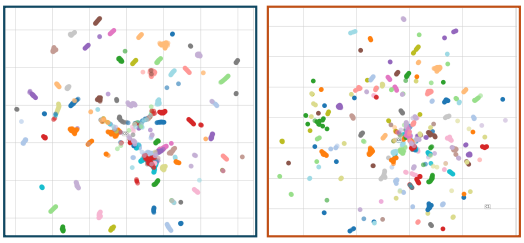


Figure 6: The visualization of the branch distributions from ARPO **(left)** and APPO **(right)**.

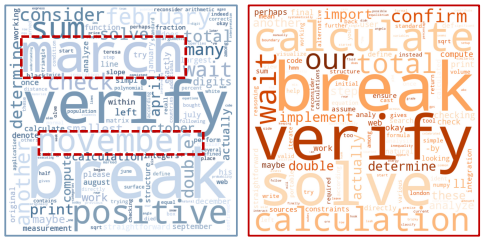


Figure 7: The word cloud of **high-entropy** tokens and those selected by our **BS** metric.

4.4 Qualitative Analysis

Training Dynamics. Figure 5 compares the training dynamics of APPO and ARPO. APPO reaches a higher final reward and follows a more stable improvement trajectory throughout training. This advantage appears early and becomes more pronounced as optimization proceeds, especially on Qwen2.5-7B. These results suggest that APPO does not merely improve local exploration, but allocates exploration more effectively during training. Specifically, APPO branches around high-impact decision points, leading to larger and smoother reward improvements. APPO maintains this advantage on both backbones, indicating that the gains mainly come from the algorithmic design rather than model-specific behavior.

Diversity Analysis. Figure 6 presents the DBSCAN (Ester et al., 1996) clustering results of rollouts sampled by ARPO and APPO, leveraging UMAP (McInnes et al., 2018). APPO produces more compact and better-separated clusters than ARPO, whose branches are more diffuse and less structured. This suggests that APPO improves diversity at the level of reasoning strategy. Branches around similar high-impact decision points remain semantically coherent, while branches from different decision points show larger semantic gaps. These results indicate that APPO gains from producing branches that are more structurally distinct and therefore more informative for credit assignment.

Interpretation of the BS metric. We visualize the tokens selected by the pure high-entropy and by our BS metric (dynamically tracked in the experiment and counted finally) in Figure 7. While high-entropy tokens do include salient reasoning keywords such as "verify", "sum", and "break", they also contain many rare nouns like "march", "november". We attribute these cases to *long-tailed effects* of the vocabulary, where uncertainty reflects absolute token rarity rather than reasoning difficulty, and optimizing such tokens is unlikely to produce transferable reasoning gains. In contrast, our BS metric filters out these cases by emphasizing tokens with larger downstream distributional shifts between the current and old policies. As a result, it is more likely to select tokens that actually redirect the reasoning trajectory and steer the success and failure of the continuations.

5 Conclusion

We proposed **APPO**, an agentic RL algorithm that shifts branching and credit assignment from coarse tool- or workflow-level units to fine-grained decision points in the generated sequence. APPO uses a Branching Score to select high-value branching locations and introduces an extra future-aware advantage scaling for more targeted credit assignment. Experiments on 13 benchmarks show that APPO consistently outperforms strong baselines while keeping efficient tool-calls. Our findings are also generic and enlightening, suggesting that modeling procedural decisions offers a practical direction for improving exploration and credit assignment in agentic RL.

AI USE STATEMENT

In this work, we used generative AI tools for editing the paper to improve readability, grammar, and presentation. We have not used generative AI tools to generate synthetic data, develop theoretical models or conceptual frameworks, formulate mathematical claims, assist with proofs, propose hypotheses, design or provide feedback on the methodology or experiments, implement methods, clean datasets, perform qualitative analysis, or interpret results. We have reviewed all AI-assisted text and take responsibility for the final content of this work, including all text, claims, and artifacts.

ETHICS STATEMENT

This work develops a training method for tool-using language-model agents and does not involve human subjects, personal data, or human-participant experiments. The experiments use public benchmarks and automated evaluation procedures, with no collection or disclosure of personally identifiable information. We follow responsible research and evaluation practices and acknowledge that improved agent capabilities should be deployed with appropriate safeguards.

REPRODUCIBILITY STATEMENT

To support reproducibility, we provide all of the implementation details in the paper and appendix, including the algorithm, prompts, training settings, benchmark descriptions, ablations and etc. We specify the model configurations, rollout budgets, branching parameters, decoding settings, evaluation metrics, and tool environments used in our experiments. These details are intended to enable independent reconstruction of the reported training and evaluation procedures.

REFERENCES

Shiyi Cao, Dacheng Li, Fangzhou Zhao, Shuo Yuan, Sumanth R Hegde, Connor Chen, Charlie Ruan, Tyler Griggs, Shu Liu, Eric Tang, et al. Skyrl-agent: Efficient rl training for multi-turn llm agent. *arXiv preprint arXiv:2511.16108*, 2025.

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

Kaiyuan Chen, Yixin Ren, Yang Liu, Xiaobo Hu, Haotong Tian, Tianbao Xie, Fangfu Liu, Haoye Zhang, Hongzhang Liu, Yuan Gong, et al. xbench: Tracking agents productivity scaling with profession-aligned real-world evaluations. *arXiv preprint arXiv:2506.13651*, 2025b.

Xiangxiang Chu, Hailang Huang, Xiao Zhang, Fei Wei, and Yong Wang. GPG: A simple and strong reinforcement learning baseline for model reasoning. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=inccdtfx8x.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Tri Dao. Flashattention-2: Faster attention with better parallelism and work partitioning. *arXiv preprint arXiv:2307.08691*, 2023.

Guanting Dong, Licheng Bao, Zhongyuan Wang, Kangzhi Zhao, Xiaoxi Li, Jiajie Jin, Jinghan Yang, Hangyu Mao, Fuzheng Zhang, Kun Gai, et al. Agentic entropy-balanced policy optimization. *arXiv preprint arXiv:2510.14545*, 2025a.

Guanting Dong, Yifei Chen, Xiaoxi Li, Jiajie Jin, Hongjin Qian, Yutao Zhu, Hangyu Mao, Guorui Zhou, Zhicheng Dou, and Ji-Rong Wen. Tool-star: Empowering llm-brained multi-tool reasoner via reinforcement learning. *arXiv preprint arXiv:2505.16410*, 2025b.

Guanting Dong, Hangyu Mao, Kai Ma, Licheng Bao, Yifei Chen, Zhongyuan Wang, Zhongxia Chen, Jiazhen Du, Huiyang Wang, Fuzheng Zhang, et al. Agentic reinforced policy optimization. *arXiv preprint arXiv:2507.19849*, 2025c.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv e-prints*, pp. arXiv–2407, 2024.

Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd*, volume 96, pp. 226–231, 1996.

Jiazhan Feng, Shijue Huang, Xingwei Qu, Ge Zhang, Yujia Qin, Baoquan Zhong, Chengquan Jiang, Jinxin Chi, and Wanjun Zhong. Retool: Reinforcement learning for strategic tool use in llms. *arXiv preprint arXiv:2504.11536*, 2025a.

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

Xidong Feng, Ziyu Wan, Muning Wen, Stephen Marcus McAleer, Ying Wen, Weinan Zhang, and Jun Wang. Alphazero-like tree-search can guide large language model decoding and training. *arXiv preprint arXiv:2309.17179*, 2023.

Jiaxuan Gao, Wei Fu, Minyang Xie, Shusheng Xu, Chuyi He, Zhiyu Mei, Banghua Zhu, and Yi Wu. Beyond ten turns: Unlocking long-horizon agentic search with large-scale asynchronous rl. *arXiv preprint arXiv:2508.07976*, 2025.

Zhibin Gou, Zhihong Shao, Yeyun Gong, Yelong Shen, Yujiu Yang, Minlie Huang, Nan Duan, and Weizhu Chen. Tora: A tool-integrated reasoning agent for mathematical problem solving. *arXiv preprint arXiv:2309.17452*, 2023.

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

Yiran Guo, Lijie Xu, Jie Liu, Dan Ye, and Shuang Qiu. Segment policy optimization: Effective segment-level credit assignment in rl for large language models. *arXiv preprint arXiv:2505.23564*, 2025b.

Danijar Hafner. Benchmarking the spectrum of agent capabilities. *arXiv preprint arXiv:2109.06780*, 2021.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.

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

Zhenyu Hou, Ziniu Hu, Yujiang Li, Rui Lu, Jie Tang, and Yuxiao Dong. Treerl: Llm reinforcement learning with on-policy tree search. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12355–12369, 2025.

Jian Hu, Jason Klein Liu, Haotian Xu, and Wei Shen. Reinforce++: Stabilizing critic-free policy optimization with global advantage normalization. *arXiv preprint arXiv:2501.03262*, 2025.

Kexin Huang, Haoming Meng, Junkang Wu, Jinda Lu, Chiyu Ma, Ziqian Chen, Xue Wang, Bolin Ding, Jiancan Wu, Xiang Wang, et al. On the direction of rlvr updates for llm reasoning: Identification and exploitation. *arXiv preprint arXiv:2603.22117*, 2026.

Binyuan Hui, Jian Yang, Zeyu Cui, Jiaxi Yang, Dayiheng Liu, Lei Zhang, Tianyu Liu, Jiajun Zhang, Bowen Yu, Keming Lu, et al. Qwen2. 5-coder technical report. *arXiv preprint arXiv:2409.12186*, 2024.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

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

Yuxiang Ji, Ziyu Ma, Yong Wang, Guanhua Chen, Xiangxiang Chu, and Liaoni Wu. Tree search for llm agent reinforcement learning. *arXiv preprint arXiv:2509.21240*, 2025.

Jiajie Jin, Xiaoxi Li, Guanting Dong, Yuyao Zhang, Yutao Zhu, Yang Zhao, Hongjin Qian, and Zhicheng Dou. Hira: A hierarchical reasoning framework for decoupled planning and execution in deep search. *arXiv preprint arXiv:2507.02652*, 2025a.

Jiajie Jin, Yutao Zhu, Zhicheng Dou, Guanting Dong, Xinyu Yang, Chenghao Zhang, Tong Zhao, Zhao Yang, and Ji-Rong Wen. Flashrag: A modular toolkit for efficient retrieval-augmented generation research. In *Companion Proceedings of the ACM on Web Conference 2025*, pp. 737–740, 2025b.

Timo Kaufmann, Paul Weng, Viktor Bengs, and Eyke Hüllermeier. A survey of reinforcement learning from human feedback. *arXiv preprint arXiv:2312.14925*, 2023.

Xin Lai, Zhuotao Tian, Yukang Chen, Senqiao Yang, Xiangru Peng, and Jiaya Jia. Step-dpo: Step-wise preference optimization for long-chain reasoning of llms. *arXiv preprint arXiv:2406.18629*, 2024.

Harrison Lee, Samrat Phatale, Hassan Mansoor, Kellie Ren Lu, Thomas Mesnard, Johan Ferret, Colton Bishop, Ethan Hall, Victor Carbune, and Abhinav Rastogi. Rlaif: Scaling reinforcement learning from human feedback with ai feedback. 2023.

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

Pengxiang Li, Zhi Gao, Bofei Zhang, Yapeng Mi, Xiaojian Ma, Chenrui Shi, Tao Yuan, Yuwei Wu, Yunde Jia, Song-Chun Zhu, et al. Iterative tool usage exploration for multimodal agents via step-wise preference tuning. *arXiv preprint arXiv:2504.21561*, 2025a.

Xiaoxi Li, Guanting Dong, Jiajie Jin, Yuyao Zhang, Yujia Zhou, Yutao Zhu, Peitian Zhang, and Zhicheng Dou. Search-o1: Agentic search-enhanced large reasoning models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 5420–5438, 2025b.

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

Xiaoxi Li, Jiajie Jin, Yujia Zhou, Yongkang Wu, Zhonghua Li, Ye Qi, and Zhicheng Dou. Retrollm: Empowering large language models to retrieve fine-grained evidence within generation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 16754–16779, 2025d.

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

Zhong-Zhi Li, Duzhen Zhang, Ming-Liang Zhang, Jiaxin Zhang, Zengyan Liu, Yuxuan Yao, Haotian Xu, Junhao Zheng, Pei-Jie Wang, Xiuyi Chen, et al. From system 1 to system 2: A survey of reasoning large language models. *arXiv preprint arXiv:2502.17419*, 2025f.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *The twelfth international conference on learning representations*, 2023.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding r1-zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*, 2025.

Keer Lu, Chong Chen, Xili Wang, Bin Cui, Yunhuai Liu, and Wentao Zhang. Pilotrl: Training language model agents via global planning-guided progressive reinforcement learning. *arXiv preprint arXiv:2508.00344*, 2025.

Zhengxi Lu, Zhiyuan Yao, Jinyang Wu, Chengcheng Han, Qi Gu, Xunliang Cai, Weiming Lu, Jun Xiao, Yueting Zhuang, and Yongliang Shen. Skill0: In-context agentic reinforcement learning for skill internalization. *arXiv preprint arXiv:2604.02268*, 2026.

Chiyu Ma, Shuo Yang, Kexin Huang, Jinda Lu, Haoming Meng, Shangshang Wang, Bolin Ding, Soroush Vosoughi, Guoyin Wang, and Jingren Zhou. Fipo: Eliciting deep reasoning with future-kl influenced policy optimization. *arXiv preprint arXiv:2603.19835*, 2026a.

Ziyu Ma, Shidong Yang, Yuxiang Ji, Xucong Wang, Yong Wang, Yiming Hu, Tongwen Huang, and Xiangxiang Chu. Skillclaw: Let skills evolve collectively with agentic evolver. *arXiv preprint arXiv:2604.08377*, 2026b.

Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

Haoming Meng, Kexin Huang, Shaohang Wei, Chiyu Ma, Shuo Yang, Xue Wang, Guoyin Wang, Bolin Ding, and Jingren Zhou. Sparse but critical: A token-level analysis of distributional shifts in rlvr fine-tuning of llms. *arXiv preprint arXiv:2603.22446*, 2026.

Grégoire Mialon, Clémentine Fourrier, Thomas Wolf, Yann LeCun, and Thomas Scialom. Gaia: a benchmark for general ai assistants. In *The Twelfth International Conference on Learning Representations*, 2023.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori B Hashimoto. s1: Simple test-time scaling. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 20286–20332, 2025.

Long Phan, Alice Gatti, Ziwen Han, Nathaniel Li, Josephina Hu, Hugh Zhang, Chen Bo Calvin Zhang, Mohamed Shaaban, John Ling, Sean Shi, et al. Humanity's last exam. *arXiv preprint arXiv:2501.14249*, 2025.

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

Cheng Qian, Emre Can Acikgoz, Qi He, Hongru Wang, Xiusi Chen, Dilek Hakkani-Tür, Gokhan Tur, and Heng Ji. Toolrl: Reward is all tool learning needs. *arXiv preprint arXiv:2504.13958*, 2025a.

Cheng Qian, Emre Can Acikgoz, Hongru Wang, Xiusi Chen, Avirup Sil, Dilek Hakkani-Tur, Gokhan Tur, and Heng Ji. Smart: Self-aware agent for tool overuse mitigation. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 4604–4621, 2025b.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

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

Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 3505–3506, 2020.

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

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

Leyang Shen, Yang Zhang, Chun Kai Ling, Xiaoyan Zhao, and Tat-Seng Chua. Carl: Critical action focused reinforcement learning for multi-step agent. *arXiv preprint arXiv:2512.04949*, 2025.

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

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *Advances in neural information processing systems*, 36:8634–8652, 2023.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. Alfworld: Aligning text and embodied environments for interactive learning. *arXiv preprint arXiv:2010.03768*, 2020.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling llm test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*, 2024.

Hongjin Su, Shizhe Diao, Ximing Lu, Mingjie Liu, Jiacheng Xu, Xin Dong, Yonggan Fu, Peter Belcak, Hanrong Ye, Hongxu Yin, et al. Toolorchestra: Elevating intelligence via efficient model and tool orchestration. *arXiv preprint arXiv:2511.21689*, 2025a.

Zhenpeng Su, Leiyu Pan, Xue Bai, Dening Liu, Guanting Dong, Jiaming Huang, Minxuan Lv, Wenping Hu, Fuzheng Zhang, Kun Gai, et al. Klear-reasoner: Advancing reasoning capability via gradient-preserving clipping policy optimization. *arXiv preprint arXiv:2508.07629*, 2025b.

Kimi Team, Yifan Bai, Yiping Bao, Y Charles, Cheng Chen, Guanduo Chen, Haiting Chen, Huarong Chen, Jiahao Chen, Ningxin Chen, et al. Kimi k2: Open agentic intelligence. *arXiv preprint arXiv:2507.20534*, 2025a.

Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, et al. Kimi k1. 5: Scaling reinforcement learning with llms. *arXiv preprint arXiv:2501.12599*, 2025b.

Meituan LongCat Team, Bei Li, Bingye Lei, Bo Wang, Bolin Rong, Chao Wang, Chao Zhang, Chen Gao, Chen Zhang, Cheng Sun, et al. Longcat-flash technical report. *arXiv preprint arXiv:2509.01322*, 2025c.

Qwen Team. Qwq: Reflect deeply on the boundaries of the unknown, 2024.

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

Hongru Wang, Cheng Qian, Manling Li, Jiahao Qiu, Boyang Xue, Mengdi Wang, Heng Ji, and Kam-Fai Wong. Toward a theory of agents as tool-use decision-makers. *arXiv preprint arXiv:2506.00886*, 2025a.

Lei Wang, Wanyu Xu, Yihuai Lan, Zhiqiang Hu, Yunshi Lan, Roy Ka-Wei Lee, and Ee-Peng Lim. Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: long papers)*, pp. 2609–2634, 2023.

Pinzheng Wang, Shuli Xu, Juntao Li, Yu Luo, Dong Li, Jianye Hao, and Min Zhang. **Re**^{2}: Unlocking llm reasoning via reinforcement learning with re-solving. *arXiv preprint arXiv:2603.07197*, 2026.

Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xionghui Chen, Jianxin Yang, Zhenru Zhang, et al. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for llm reasoning. *arXiv preprint arXiv:2506.01939*, 2025b.

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

Sean Welleck, Amanda Bertsch, Matthew Finlayson, Hailey Schoelkopf, Alex Xie, Graham Neubig, Ilia Kulikov, and Zaid Harchaoui. From decoding to meta-generation: Inference-time algorithms for large language models. *arXiv preprint arXiv:2406.16838*, 2024.

Di Wu, Devendra Singh Sachan, Wen-tau Yih, and Mingda Chen. Procedural knowledge at scale improves reasoning. *arXiv preprint arXiv:2604.01348*, 2026.

Jialong Wu, Wenbiao Yin, Yong Jiang, Zhenglin Wang, Zekun Xi, Runnan Fang, Linhai Zhang, Yulan He, Deyu Zhou, Pengjun Xie, et al. Webwalker: Benchmarking llms in web traversal. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10290–10305, 2025a.

Junde Wu, Jiayuan Zhu, Yuyuan Liu, Min Xu, and Yueming Jin. Agentic reasoning: A streamlined framework for enhancing llm reasoning with agentic tools. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 28489–28503, 2025b.

Rong Wu, Xiaoman Wang, Jianbiao Mei, Pinlong Cai, Daocheng Fu, Cheng Yang, Licheng Wen, Xuemeng Yang, Yufan Shen, Yuxin Wang, et al. Evolver: Self-evolving llm agents through an experience-driven lifecycle. *arXiv preprint arXiv:2510.16079*, 2025c.

Yangzhen Wu, Zhiqing Sun, Shanda Li, Sean Welleck, and Yiming Yang. Inference scaling laws: An empirical analysis of compute-optimal inference for problem-solving with language models. *arXiv preprint arXiv:2408.00724*, 2024.

Peng Xia, Jianwen Chen, Hanyang Wang, Jiaqi Liu, Kaide Zeng, Yu Wang, Siwei Han, Yiyang Zhou, Xujiang Zhao, Haifeng Chen, et al. Skillrl: Evolving agents via recursive skill-augmented reinforcement learning. *arXiv preprint arXiv:2602.08234*, 2026.

Yuxi Xie, Anirudh Goyal, Wenyue Zheng, Min-Yen Kan, Timothy P Lillicrap, Kenji Kawaguchi, and Michael Shieh. Monte carlo tree search boosts reasoning via iterative preference learning. *arXiv preprint arXiv:2405.00451*, 2024.

Ran Xin, Chenguang Xi, Jie Yang, Feng Chen, Hang Wu, Xia Xiao, Yifan Sun, Shen Zheng, and Ming Ding. Bfs-prover: Scalable best-first tree search for llm-based automatic theorem proving. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 32588–32599, 2025.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pp. 2369–2380, 2018.

Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. Webshop: Towards scalable real-world web interaction with grounded language agents. *Advances in Neural Information Processing Systems*, 35:20744–20757, 2022a.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *The eleventh international conference on learning representations*, 2022b.

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

Yunpeng Zhai, Shuchang Tao, Cheng Chen, Anni Zou, Ziqian Chen, Qingxu Fu, Shinji Mai, Li Yu, Jiaji Deng, Zouying Cao, et al. Agentevolver: Towards efficient self-evolving agent system. *arXiv preprint arXiv:2511.10395*, 2025.

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

Haozhen Zhang, Quanyu Long, Jianzhu Bao, Tao Feng, Weizhi Zhang, Haodong Yue, and Wenya Wang. Memskill: Learning and evolving memory skills for self-evolving agents. *arXiv preprint arXiv:2602.02474*, 2026.

Yifan Zhang and Team Math-AI. American invitational mathematics examination (aime) 2024, 2024.

Yifan Zhang and Team Math-AI. American invitational mathematics examination (aime) 2025, 2025.

Yubao Zhao, Weiquan Huang, Sudong Wang, Ruochen Zhao, Chen Chen, Yao Shu, and Chengwei Qin. Training multi-turn search agent via contrastive dynamic branch sampling. *arXiv preprint arXiv:2602.03719*, 2026.

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

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, and Zheyan Luo. Llamafactory: Unified efficient fine-tuning of 100+ language models. In *Proceedings of the 62nd annual meeting of the association for computational linguistics (volume 3: system demonstrations)*, pp. 400–410, 2024.

Andy Zhou, Kai Yan, Michal Shlapentokh-Rothman, Haohan Wang, and Yu-Xiong Wang. Language agent tree search unifies reasoning acting and planning in language models. *arXiv preprint arXiv:2310.04406*, 2023.

Appendix
APPO: Agentic Procedural Policy Optimization

The Appendix of the paper is organized as:

- Appendix A: We state the scope of the theoretical analysis.
- Appendix B: We introduce datasets and baselines.
- Appendix C: We report full implementation details.
- Appendix D: We report alternative designs of the BS metric.
- Appendix E: We report studies of the branching budget.
- Appendix F: We report the prompts we used.
- Appendix G: We report the limitations.
- Appendix H: We report case studies.
- Appendix I: We report the algorithm of APPO.
- Appendix J: We report the impact statement.
- Appendix K: We report the declaration of LLM usage.

A Scope of the Theoretical Analysis

We present the theoretical analysis under explicit idealized assumptions and use it to motivate score-guided branching and future-aware credit assignment. The implemented algorithm is evaluated directly through the controlled ablations in the main paper.

B Datasets and Baselines.

The datasets employed in our experiments are introduced as the following:

AIME24. AIME24 comprises the 30 problems from the 2024 American Invitational Mathematics Examination (AIME), split across two competition papers. Each problem demands an integer answer between 0 and 999, yet the reasoning paths required are far from trivial—contestants must navigate number theory, combinatorics, geometry, and algebra. Its small size makes it sensitive to variance, so performance is typically reported as average accuracy over multiple runs.

AIME25. AIME25 follows the same format as AIME24, providing another 30 competition-level problems released in February 2025. Because the problems post-date most training cutoffs, AIME25 serves as a relatively contamination-resistant probe of genuine mathematical reasoning. It has quickly become a standard checkpoint for evaluating frontier models on olympiad-style problem solving.

MATH500. MATH500 (Lightman et al., 2023) is a 500-problem subset drawn from the broader MATH to span all seven subject categories and five difficulty levels of the original collection. Problems range from introductory algebra to competition-level precalculus, making the subset a compact yet representative testbed for mathematical reasoning. It is widely used as an in-distribution evaluation complement to harder olympiad benchmarks.

MATH. MATH (Hendrycks et al., 2021) contains 12,500 problems sourced from high school mathematics competitions, covering algebra, counting and probability, geometry, number theory, and precalculus, among others. Each problem is accompanied by a step-by-step solution, enabling process-level as well as answer-level evaluation. The dataset is notable for its difficulty gradient: even the easiest problems require multi-step symbolic reasoning, while the hardest rival olympiad-level challenges.

GSM8K. GSM8K is released by (Cobbe et al., 2021) at OpenAI and consists of 8,500 linguistically diverse word problems pitched at the elementary school level. Despite their apparent simplicity, the problems require chains of two to eight arithmetic steps, making them a reliable probe of basic multi-step reasoning. The dataset is split into 7,500 training and 1,000 test examples and remains one of the most widely cited benchmarks for evaluating arithmetic reasoning in language models.

HotpotQA. HotpotQA (Yang et al., 2018) contains roughly 113,000 Wikipedia-based question–answer pairs that are explicitly designed to require reasoning over two supporting documents. A key design feature is the inclusion of sentence-level supporting-fact annotations, which allow evaluation of both answer correctness and the quality of the reasoning chains.

2WikiMultihopQA. 2WikiMultihopQA (Ho et al., 2020) constructs multi-hop questions by combining information from two Wikipedia articles, using structured Wikidata triples to guarantee that each question genuinely requires cross-document reasoning rather than being solvable from a single passage. The dataset provides explicit evidence chains alongside each question, enabling fine-grained evaluation of intermediate reasoning steps.

MuSiQue. MuSiQue (Trivedi et al., 2022) takes a bottom-up approach to constructing multi-hop questions: single-hop questions from existing datasets are composed into 2–4 hop chains via directed acyclic graphs, ensuring that each hop is genuinely necessary and that shortcuts are systematically eliminated. The dataset contains approximately 25,000 questions and is considered harder to game than earlier multi-hop benchmarks, where models could often answer correctly by attending to a single passage.

Bamboogle. Bamboogle (Press et al., 2023) is a small, hand-crafted collection of 125 two-hop factual questions, assembled specifically by filtering out any question that Google's search engine answers correctly. This adversarial construction criterion means the questions tend to require non-obvious compositional reasoning that surface-level retrieval struggles with. Despite its modest size, Bamboogle is frequently used as a stress test for retrieval-augmented generation systems.

GAIA. GAIA (Mialon et al., 2023) is a benchmark for General AI Assistants comprising 466 real-world questions across three difficulty levels. What distinguishes GAIA is its emphasis on practical, tool-assisted problem solving: questions may require web browsing, file parsing, multi-modal understanding, and multi-step planning in combination. Humans solve the benchmark with high accuracy, yet agents still struggle significantly, positioning GAIA as a meaningful frontier for agentic evaluation.

HLE. Humanity's Last Exam (HLE) (Phan et al., 2025) consists of 2,500 questions assembled by over 1,000 subject-matter experts to represent the frontier of human knowledge. Questions span mathematics, the natural sciences, humanities, and professional domains, with roughly 10% requiring image comprehension. The benchmark was explicitly designed to resist saturation: at release, top models scored below 10%, making it one of the most demanding closed-ended evaluations currently available.

WebWalker. WebWalker (Wu et al., 2025a) benchmarks LLMs on web traversal. The accompanying dataset, WebWalkerQA, contains 680 questions derived from real websites, where answering correctly requires an agent to plan a sequence of click actions across multiple pages. The benchmark highlights a gap between static retrieval and dynamic, navigation-based information seeking.

XBench. XBench (Chen et al., 2025b) is a profession-aligned, dynamic evaluation suite introduced by Sequoia Capital to measure AI agent productivity in real-world occupational contexts. Rather than testing isolated capabilities, it presents tasks drawn from domains such as marketing, software engineering, and legal work, scored against expert-produced reference outputs. Its evergreen design is intended to resist benchmark saturation and track whether agent performance translates into genuine workplace utility.

The baselines compared in our experiments are shown in the following:

GRPO. GRPO (Guo et al., 2025a) is critic-free, which groups multiple responses per prompt and utilizing relative reward comparisons within these groups, it optimizes policies based on intra-group performance rather than absolute scalar values, thereby improving sample efficiency.

Reinforce++. REINFORCE++ (Hu et al., 2025) is an enhanced variant of the classical REINFORCE algorithm that incorporates key optimization techniques from PPO—including token-level KL penalties and reward normalization—while eliminating the need for a critic model. It achieves three primary objectives: simplicity, enhanced training stability, and reduced computational overhead, making it a practical and efficient baseline for RLHF-based LLM alignment.

DAPO. DAPO (Yu et al., 2025) is an open-source, industrial-scale RL system for LLM post-training proposed by ByteDance. It introduces four key techniques—Clip-Higher, dynamic sampling, token-

level policy gradient loss, and overlong reward shaping—to address entropy collapse and training instability commonly observed in GRPO-based long chain-of-thought training.

GPPO. GPPO (Su et al., 2025b) addresses the gradient vanishing problem caused by hard token-level clipping in standard PPO-style training. By gently reintroducing bounded gradients from clipped tokens, GPPO enables finer-grained exploration control and more stable policy updates, serving as the foundation for the entropy-regularized extension CE-GPPO.

CISPO. CISPO (Chen et al., 2025a) is a reinforcement learning algorithm proposed in the MiniMax-M1 project that clips importance-sampling weights at the sequence level rather than applying per-token update clipping. This design allows all tokens—including low-probability ones—to contribute to gradient updates, reducing variance and improving training stability in off-policy LLM fine-tuning scenarios.

GIGPO. GiGPO (Feng et al., 2025b) is a critic-free RL algorithm designed for training long-horizon LLM agents, proposed by researchers at Nanyang Technological University. It introduces a two-level grouping structure that combines episode-level advantage estimation with step-level credit assignment, enabling fine-grained attribution of rewards to individual actions without requiring an additional value network.

ARPO. ARPO (Dong et al., 2025c) is a reinforcement learning framework specifically designed for multi-turn, tool-augmented LLM agents. It addresses the entropy increase observed after tool interactions by introducing entropy-based adaptive sampling and fine-grained advantage attribution at the action level, enabling more effective exploration and stable policy optimization in agentic settings.

Qwen3. Qwen3 (Yang et al., 2025) is the latest generation of the Qwen large language model family, released in April 2025. It introduces a hybrid thinking mode that allows flexible switching between deliberate chain-of-thought reasoning and fast non-thinking inference, supports 119 languages, and was trained on a corpus exceeding 36 trillion tokens, achieving competitive performance across a broad range of benchmarks.

DeepSeek-R1. DeepSeek-R1 (Guo et al., 2025a) is a large reasoning model developed by DeepSeek that incentivizes complex reasoning capabilities through large-scale reinforcement learning with verifiable rewards, without relying on supervised chain-of-thought data in its initial training stage. The resulting model exhibits emergent reasoning behaviors such as self-reflection and dynamic strategy adjustment, achieving performance competitive with OpenAI's o1 series on mathematical and scientific benchmarks.

QwQ. QwQ (Team, 2024) is an open-source reasoning model series developed by the Qwen team, first released in November 2024, designed to tackle problems requiring deep analytical thinking. The flagship QwQ-32B model demonstrates that a mid-sized model can achieve competitive performance against state-of-the-art reasoning systems such as DeepSeek-R1 and OpenAI o1-mini through extended chain-of-thought reasoning.

GPT-4o. GPT-4o (Hurst et al., 2024) is OpenAI's flagship multimodal generative model released in May 2024. It achieves significantly lower latency than its predecessors by natively handling cross-modal inputs without modality-specific preprocessing pipelines, establishing a new standard for real-time multimodal AI interaction.

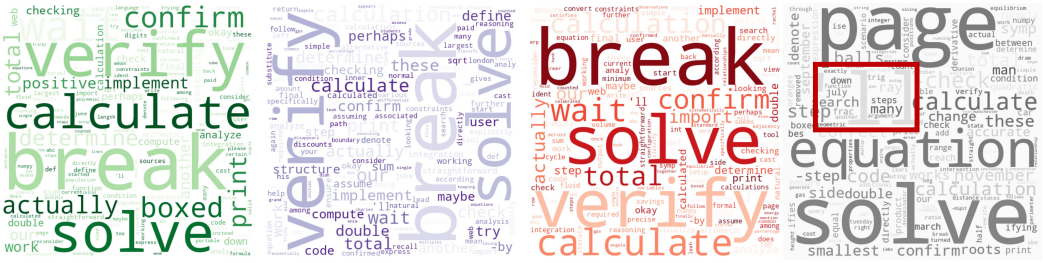
RAG. RAG (Lewis et al., 2020) enhances large language model outputs by dynamically retrieving relevant documents from an external knowledge base at inference time, thereby grounding generation in up-to-date and domain-specific information. By decoupling parametric knowledge stored in model weights from non-parametric knowledge retrieved on demand, RAG substantially reduces hallucination and improves factual accuracy without requiring full model retraining.

Search-o1. Search-o1 (Li et al., 2025b) augments large reasoning models with an agentic search workflow, enabling them to dynamically retrieve external knowledge when they encounter uncertain or knowledge-intensive steps during long chain-of-thought reasoning. By seamlessly interleaving retrieval with reasoning rather than treating search as a preprocessing step, Search-o1 improves both the reliability and factual grounding of complex multi-step inference.

WebThinker. WebThinker (Li et al., 2025c) is a deep research agent that empowers large reasoning models to autonomously search the web, navigate across multiple web pages, and synthesize retrieved information into structured research reports. It introduces a Deep Web Explorer module that tightly

Table 5: Analysis on branching configuration with Qwen2.5-7B-Instruct. The best results are in **bold**.

Method	WebWalker	HotpotQA	2Wiki	Musique	Bamboogle	Average
APPO ($N = 1, B = 2, L = 3, M = 16$)	28.5	63.0	78.0	29.8	71.0	54.1
APPO ($N = 1, B = 4, L = 2, M = 16$)	28.5	63.8	78.8	30.2	71.3	54.5
APPO ($N = 3, B = 1, L = 2, M = 16$)	**33.2**	**64.9**	**81.0**	**31.5**	**77.8**	**57.7**



(a) Add, Weight=(0.5, 0.5) (b) Add, Weight=(0.7, 0.3) (c) Add, Weight=(0.3, 0.7) (d) Future Value Only

Figure 8: WordCloud of tokens selected by alternative designs of the BS metric.

integrates web interaction with chain-of-thought reasoning, enabling end-to-end autonomous research on complex, knowledge-intensive tasks.

ReAct. ReAct (Yao et al., 2022b) is a prompting paradigm that interleaves verbal reasoning traces with executable actions in large language models, enabling them to dynamically plan, retrieve information, and adjust their behavior based on environmental feedback. By synergizing chain-of-thought reasoning with tool use, ReAct substantially improves interpretability and task performance on knowledge-intensive decision-making benchmarks compared to reasoning-only or acting-only baselines.

C Implementation Details

Our APPO follows a training pipeline largely consistent with ARPO. Specifically, we first train the backbone model to obtain fundamental tool-use capabilities using ToolStar's 54K SFT dataset together with an additional 0.8K samples drawn from the STILL dataset. For SFT training, we employ LLAMA-Factory (Zheng et al., 2024) with a learning rate of 7e-6, DeepSpeed ZeRO-3 (Rasley et al., 2020), and Flash-Attention2 (Dao, 2023). The batch size is set to 128, the weight decay is 0.1, and the model is trained for 3 epochs. We adopt BF16 mixed precision and set the maximum input length to 4096 tokens.

For the RL stage, we implement APPO based on the VERL framework (Sheng et al., 2025). In particular, all tool execution results are masked out from the loss computation to avoid introducing bias toward tool outputs. Accordingly, the loss is computed only over tokens corresponding to textual reasoning and tool calls. We use different configurations for DeepReasoning Tasks and Deep Search Tasks:

(i) DeepReasoning Tasks: For 7B models, our default configuration adopts a total training batch size of 128, a PPO mini-batch size of 16, a global rollout size of 16, and an initial sampling size of 4. The maximum response length for each interaction is limited to 4096 tokens. For APPO rollouts, we set the entropy coefficient to 0.2, parameter b to 0.5, and the future-signal clipping threshold to 0.5. The decay factor gamma is set to $2^{-\frac{1}{32}}$. For clipping with coefficient ϵ, in implementation we adopt a non-symmetric setting [1,1.2] to encourage positive exploration of the suffix policy-shift signal. To improve training stability, the KL divergence coefficient in GRPO is fixed at 0. The RL stage is carried out for 2 epochs on 8 NVIDIA H100 GPUs.

(ii) DeepSearch Tasks: For 8B models, we use the same configuration as the above, except that the maximum response length for each interaction is increased to 8192 tokens. For 14B models, the same hyperparameter settings are retained, while the experiments are also conducted on 16 NVIDIA H100 GPUs. Since the dataset contains only 1K samples, the RL stage is performed for 5 epochs.

D ALTERNATIVE DESIGNS OF THE BS METRIC

In this section, we investigate alternative formulations of the BS metric, including additive combinations of normalized entropy and the discounted suffix policy-shift signal, as well as the signal alone. Figure 8 presents word clouds obtained under these settings. The comparison is qualitative and does not establish a uniquely optimal functional form.

E MORE SENSITIVITY ANALYSIS OF KEY HYPER-PARAMETERS

Studies of the number of the branching loop L. In the main paper, our experiments are limited to a single-layer rollout tree, where all branching operations are applied only to the initial rollout. Consider a multi-layer rollout tree, i.e., the case where $L > 1$, we have $M = B \cdot (N+1)^L$. The results of our sensitivity analysis are presented in Table 5.

We observed that: (i) When $N = 1$, the overall performance of the model is relatively poor, showing even a slight decline compared to the case in the main paper where $L = 1$ and M is even smaller. For example, with $(N = 1, B = 2, L = 3, M = 16)$, the overall performance is approximately 54.1%, which is about 1.1 points lower than the case $(N = 2, B = 3, L = 1, M = 8)$ reported in Table 3. We attribute this to the insufficient diversity of the initial rollout, which causes the model to be significantly affected by randomness, leading to performance fluctuations. (ii) When the initial rollout is kept the same, varying B or L does not have a noticeable impact on performance. We believe this is because the two settings largely share the prefix information of the rollouts, resulting in substantial overlap in the sampling distribution. These results once again confirm that our choice in the main text is an optimal trade-off.

F PROMPTS

Following (Dong et al., 2025c), The prompt used in APPO is listed in the following:

> *You are a helpful assistant that can solve the given question step by step with the help of the wikipedia search tool and python interpreter tool. Given a question, you need to first think about the reasoning process in the mind and then provide the answer. During thinking, you can invoke the wikipedia search tool to search and python interpreter tool to calculate the math problem for fact information about specific topics if needed. The reasoning process and answer are enclosed within <think> </think> and <answer> </answer> tags respectively, and the search query and result are enclosed within <search> </search> and <result> </result> tags respectively. For example, <think> This is the reasoning process. </think> <search> search query here </search> <result> search result here </result> <think> This is the reasoning process. </think> <python> python code here </python> <result> python interpreter result here </result> <think> This is the reasoning process. </think> <answer> The final answer is $\boxed{answerhere}$ </answer>. In the last part of the answer, the final exact answer is enclosed within $\boxed{}$ with latex format.*

G LIMITATIONS

Although APPO introduces a fine-grained branching and credit-assignment design, our work has two main limitations: (i) the theoretical analysis uses idealized assumptions and the empirical score-selection rule should be further studied across broader settings; and (ii) following prior work, the tools employed by APPO are currently restricted to Search and Python and have not been extended to other application-oriented tools.

H CASE STUDY

We provide two kinds of cases in this section: (i) the branching stage cases, where the initial rollout is wrong but turns correct by our branching selections; (ii) the inference cases of ARPO and APPO, where ARPO fails but solvable to our APPO.

Algorithm 1: APPO Training Pipeline

Input : policy π_θ, reference policy π_{ref}, toolset T, training set $\mathcal{D}$, rollout budget M, number of initial rollouts N, number of selected branching points B, PPO epochs K_{ppo}, clipping thresholds ϵ, ϵ', procedural weight b, decay factor γ

for *each training step* **do**
- sample input $x \sim \mathcal{D}$
- set behavior policy $\pi_{\text{old}} \leftarrow \pi_\theta$
- $\mathcal{T}_{init} \leftarrow \emptyset$
- `// Initialization: generate initial rollouts`
- **for** $n = 1$ **to** N **do**
  - generate a full rollout $\mathcal{H}_n \sim \pi_{\text{old}}(\cdot \mid x; T)$ through agent-environment interaction
  - add $\mathcal{H}_n$ to $\mathcal{T}_{init}$
- `// Mini-batch procedural branching and optimization`
- **for** $e = 1$ **to** K_{ppo} **do**
  - $\mathcal{T}_{branch} \leftarrow \emptyset$
  - **while** $|\mathcal{T}_{branch}| < M - N$ **do**
    - uniformly sample one rollout $\mathcal{H}_n$ from the current rollout trees
    - **for** *each valid token position i in $\mathcal{H}_n$* **do**
      - compute token entropy $H_{n,i}$ by Eq. (3)
      - compute the discounted suffix policy-shift signal $\Omega_{n,i}$ by Eq. (4)
      - compute Branching Score $\text{BS}_{n,i}$ by Eq. (5)
    - select the top-B tokens according to $\text{BS}_{n,i}$ and denote them as $\mathcal{B}_n$
    - **foreach** $i \in \mathcal{B}_n$ **do**
      - resample a continuation from prefix $\mathcal{H}_{n,<i}$ using the current policy π_θ to obtain a branch $\mathcal{H}_n^{new}$
      - add $\mathcal{H}_n^{new}$ to $\mathcal{T}_{branch}$
      - update the rollout tree with $\mathcal{H}_n^{new}$
      - **if** $|\mathcal{T}_{branch}| = M - N$ **then**
        - **break**
  - `// Dual-group advantage estimation`
  - evaluate reward $R(\mathcal{H})$ for each rollout $\mathcal{H} \in \mathcal{T}_{init} \cup \mathcal{T}_{branch}$
  - compute group-relative advantages separately for $\mathcal{T}_{init}$ and $\mathcal{T}_{branch}$ by Eq. (6)
  - `// Future-aware procedural advantage`
  - **for** *each token position i in each initial rollout $\mathcal{H}_n \in \mathcal{T}_{init}$* **do**
    - compute $\hat{A}_{n,i}^{\text{fut}}$ by Eq. (7)
    - set $\hat{A}_{n,i} \leftarrow \hat{A}_{n,i}^{\text{base}}(1 + b \cdot \hat{A}_{n,i}^{\text{fut}})$
  - `// Policy optimization on initial rollouts`
  - optimize Eq. (8) on $\mathcal{T}_{init}$ with KL regularization to π_{ref}
  - update θ

I ALGORITHM

The algorithm of APPO is shown in Algorithm 1.

J IMPACT STATEMENT

This paper introduces APPO, a novel Agent RL method designed to push the boundaries of where to branch and how to branch. The proposed APPO holds broad societal value, playing a significant role in domains such as search systems, AI-assisted healthcare and education. It is particularly worth noting that, as a work aimed at advancing agent capabilities, APPO not only delivers broader efficacy improvements but also catalyzes increased automated behaviors. Like any research in this field, we are fully committed to participating in regulatory compliance and safety governance for agents. Overall, our proposed APPO offers extensive value to both the research and application communities, fostering healthy and sustainable development in the field of agents.

K DECLARATION OF LLM USAGE

LLMs are only for polishing the writing of this paper.

Case-1 from MATH

The 7th and 8th grades have enrollments of 520 and 650 respectively. The two grades have a total of 18 representatives on the Student Council. How many representatives should the 8th grade have if there is to be a fair representation of the two grades?

Wrong Rollout

To determine the number of representatives the 8th grade should have, we need to ensure that the number of representatives is proportional to the enrollment of each grade. Here are the steps to solve the problem:
1. Calculate the total enrollment: Total enrollment = 520 + 650 = 1170
2. Determine the proportion of the 8th grade's enrollment to the total enrollment: Proportion of 8th grade = 650 / 1170
3. Simplify the proportion: To simplify 650/1170, we find the greatest common divisor (GCD) of 650 and 1170. The GCD is 50.
650 ÷ 50 / 1170 ÷ 50 = 13 / 23.4
4. Express the simplified proportion: 13 / 23.4
5. Calculate the number of representatives for the 8th grade: Number of representatives = 13 / (13 + 23.4) × 18
6. Further calculation: 13 / 36.4 × 18
7. Perform the multiplication: 13 / 36.4 ≈ 0.35714 0.35714 × 18 ≈ 6.42852
8. Round to the nearest whole number: Since the number must be a whole number, we round 6.42852 to 6.
Thus, the 8th grade should have $\boxed{6}$ representatives.

APPO Branching

To determine the number of representatives the 8th grade should have, we need to ensure that the number of representatives is proportional to the enrollment of each grade. Here are the steps to solve the problem:
1. Calculate the total enrollment: Total enrollment = 520 + 650 = 1170
2. Determine the proportion of the 8th grade's enrollment to the total enrollment: Proportion of 8th grade = 650 / 1170
3. Simplify the proportion: To simplify 650/1170, we find the greatest common divisor (GCD) of 650 and 1170. The GCD is 50.
650 ÷ 50 / 1170 ÷ 50 = 13 / 23.4
4. Express the simplified proportion: 13 / 23.4
5. Calculate the number of representatives for the 8th grade: Number of representatives = 13 / (13 + 23.4) × 18
6. Further calculation: 13 / 36.4 × 18
7. Perform the multiplication: 13 / 36.4 ≈ 0.35714 0.35714 × 18 ≈ 6.42852
8. Round to the nearest whole number: Since the number must be a whole number, we round 6.42852 to 6.
———-Branching———-
Wait, let me reconsider. The GCD of 650 and 1170 should be recalculated.
650 = 2 × 5² × 13 1170 = 2 × 3² × 5 × 13
GCD(650, 1170) = 2 × 5 × 13 = 130
So: 650 / 1170 = 650÷130 / 1170÷130 = 5 / 9
Number of 8th grade representatives = (5/9) × 18 = 10
Thus, the 8th grade should have $\boxed{10}$ representatives.

Case-2 from Musique

What does the acronym of the organization Danish Football Union is part of stand for?

Wrong Rollout

To provide the accurate answer, I need to first identify the organization Danish Football Union is a part of and then find out the acronym associated with that organization.
<search>
search query: What is Danish Football Union organization?
</search>
<result>
Five pages omitted.
</result>
The search result indicates that the Danish Football Union (DBU) is a member of both FIFA and UEFA. Now, I will find the acronym for UEFA to provide the answer.
<search>
search query: What is UEFA acronym?
</search>
<result>
Eight pages omitted.
</result>
Therefore, the acronym of the organization that the Danish Football Union is part of stands for Union of European Football Associations.

APPO Branching

To provide the accurate answer, I need to first identify the organization Danish Football Union is a part of and then find out the acronym associated with that organization.
<search>
search query: What is Danish Football Union organization?
</search>
<result>
Five pages omitted.
</result>
The search result indicates that the Danish Football Union (DBU) is a member of both FIFA and UEFA. Now, I will find the acronym for UEFA to provide the answer.
<search>
search query: What is UEFA acronym?
</search>
<result>
Eight pages omitted.
</result>
———-Branching———-
The search result shows that the Danish Football Union is a member of both FIFA and UEFA. The question asks for "the organization" (singular), and since FIFA is the global governing body of football while UEFA is only the European body, the more fundamental organization is FIFA.
<search>
search query: What is FIFA acronym?
</search>
<result>
Nine pages omitted.
</result>
Therefore, the acronym FIFA stands for International Federation of Association Football.

Case-3 (Tool-Star RL Dataset): when the composer of the film Waqt (1965) was born?

<think>
To find out when the composer of the film Waqt (1965) was born, we need to gather relevant information through Wikipedia searches and identify the necessary details. </think>
<search>search query: Who composed music for Waq 1965 film? **</search>**
<result>
Page 1: Lyrics and video of songs from Movie / Album : Waqt (1965); Music by: Ravi; Singer (s): Asha Bhosle, Mahendra Kapoor, Manna Dey, Mohammed Rafi; having star cast: Sunil Dutt, Sadhana, Raaj Kumar, ... Page 2: Waqt songs are composed by Ravi Shankar Sharma (Ravi), while Sahir Ludhianvi wrote its lyrics. Check out Waqt songs list with lyrics and music videos below. Page 3: Ravi (music director) — Ravi Shankar Sharma, better known by his first name Ravi, is an Indian music director, who has composed music for several Hindi and Malayalam films. Page 4: These included story writer Akhtar Mirza, who was credited as F.A. Mirza, dialogue writer Akhtar-Ul-Iman, and editor Pran Mehra. Album details and songs for Waqt. Page 5: The film contains the Hindi songs "Hum Jab Simat Ke", sung by Mahendra Kapoor and Asha Bhosle, "Waqt se din aur raat", sung by Mohammed Rafi, "Aage Bhi Jaane Na Tu", sung by Asha Bhosle and ... Page 6: Listen to Waqt (Original Motion Picture Soundtrack) by Ravi, Sahir Ludhianvi & Traditional on Apple Music. 1965. 21 Songs. Duration: 1 hour 35 minutes. </result>
<think>
Having obtained the relevant information, it is clear that Ravi composed music for this film. Now, to ascertain his date of birth, more focused searches would be necessary to obtain accurate information related to him.</think>
<search>
search query: When was Ravi (music director) born? **</search>**
<result>
Page 1: Ravi was born on 3 March 1926 in Delhi, British India. He was a music artist and composer, known for Gharana (1961), Dilwale Dulhania Le Jayenge (1995) and Khandan (1965). Page 2: Ravi Shankar Sharma (3 March 1926 – 7 March 2012), often referred to mononymously as Ravi, was an Indian music director, who had composed music for several Hindi and Malayalam films.</result>
<think>
Based on the search results found so far, it looks like Ravi (full name Ravi Shankar Sharma) has a birth date which seems to repeat often, but it appears consistent across multiple relevant sources namely March 3, 1926. </think>
<python>
print("Ravi's complete name is Ravi Shankar Sharma. His birthday is on 3rd March 1926.")
</python>
<result>
Ravi's complete name is Ravi Shankar Sharma. His birthday is on 3rd March 1926. </result>
<think>
Upon confirming the date of birth via the reliable resource, Ravi was born on March 3, 1926 which will correctly identify the composer's date of birth without any further steps. </think>
Final confirmation composer_birth_date = "March 3, 1926"
<python>
print(f"The composer of the film 'Waqt (1965)', Ravi was born on composer_birth_date.")
</python> ,
ground_truth: 3 March 1926 f1_score: 1.0,
answer: March 3, 1926, ground_truth: 3 March 1926