IT TAKES 8 TOKENS: WEAK-TO-STRONG OFF-POLICY RL VIA AUXILIARY BRANCHES

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement learning with verifiable rewards has emerged as a standard approach for enhancing reasoning in large language models, which typically optimizes the policy by contrasting multiple self-generated rollouts. However, we identify a critical **support-limited** bottleneck in this paradigm: on challenging reasoning tasks, the target model's samples often exhibit semantic redundancy and provide negligible reward contrast for policy updates. In this paper, we propose to overcome this limitation through a **weak-to-strong learning** paradigm, where a policy's exploration is informed by a weaker but computationally efficient auxiliary model. We introduce W2SPO, an off-policy RL method that injects short auxiliary segments—often as brief as 8 tokens—into intermediate target-model trajectories and the target model then completes the reasoning path from these diverted states. Policy updates are restricted to these short inserted segments based on final verifiable rewards. Empirically, W2SPO achieves superior performance among evaluated 4B-scale models on mathematical reasoning benchmarks, outperforming evaluated post-trained baselines. Compared with vanilla GRPO under the same sampling budget, W2SPO improves Pass@1 from 62.3% to 64.2% while achieving a **3.55**$\times$ training speedup. Additional experiments show gains with a non-Qwen 7B target and on a multi-turn tool-use benchmark. These results suggest that weak auxiliary branches can induce stronger target reasoning policies by expanding local exploration support. Code and data are available at `https://anonymous.4open.science/r/W2SPO-1B86/`.

1 INTRODUCTION

Reinforcement learning with verifiable rewards has become a central approach for improving language-model reasoning (Shao et al., 2024; Guo et al., 2025a; Song et al., 2025b). Methods such as PPO- and GRPO-style training optimize models from sampled reasoning traces using outcome rewards, allowing models to discover alternative solution paths without imitating a fixed supervised trace (Wen et al., 2025; Cobbe et al., 2021). However, these methods rely heavily on self-generated rollouts from the target model. In standard GRPO, the current policy samples multiple trajectories for the same problem and updates the model according to relative advantages within the group. This procedure assumes that the sampled rollouts are sufficiently diverse and informative (Wang et al., 2023). For difficult reasoning problems, this assumption often fails: target-model rollouts may look lexically different but follow similar intermediate logic and fail in similar ways (Dang et al., 2025). Thus, the bottleneck is not merely insufficient sampling budget, but limited coverage of the rollout distribution (Yue et al., 2025; Brown et al., 2024).

This support limitation weakens reasoning RL in two ways (Dong et al., 2026). When all rollouts are incorrect, the group provides little useful reward contrast (Chen et al., 2026). Even when rewards differ, the variation may remain within a narrow region of the target model's current reasoning distribution, causing RL to reinforce existing behaviors rather than discover low-probability reasoning turns that could redirect the trajectory toward a correct solution (Li et al., 2026). This motivates a different form of exploration: instead of simply increasing the number of self-rollouts, can we locally expand the target model's rollout support while still letting the target model complete the long-horizon reasoning process?

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.

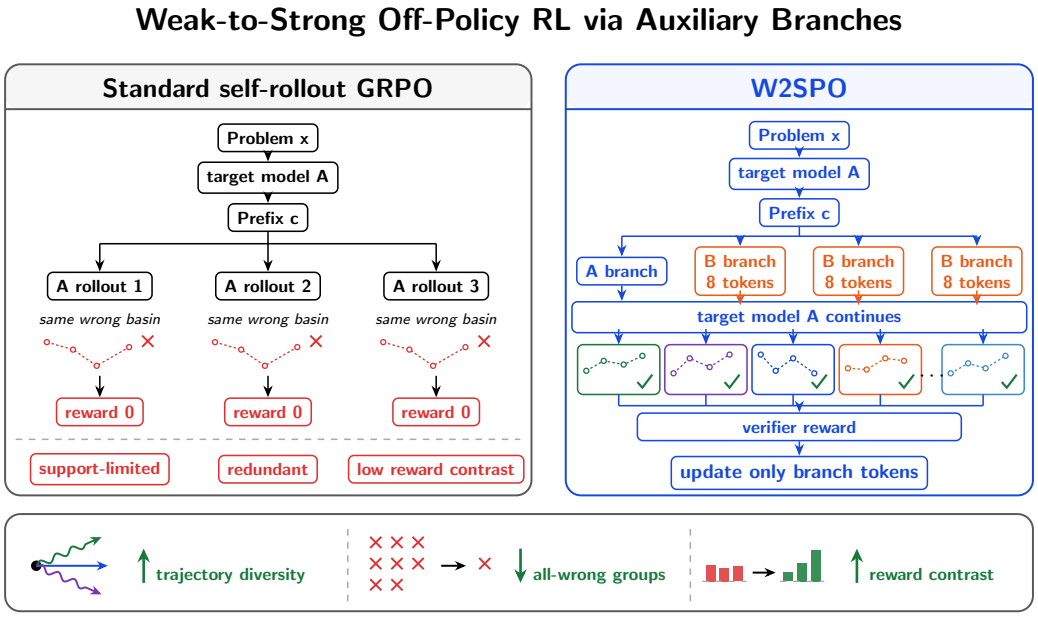

Figure 1: Overview of W2SPO. Standard GRPO relies on self-rollouts from the target model, which can be semantically redundant and yield all-wrong rollout groups on difficult reasoning problems. W2SPO locally expands the rollout support by inserting short auxiliary branches into an intermediate target-model prefix, then lets the target model complete the solution. Verifier rewards are assigned to the completed trajectories, while policy updates are applied only to the short branch segment.

We study a weak-to-strong alternative (Burns et al., 2023; Wang et al., 2026). Instead of using an auxiliary model as a teacher, we use it as a local exploration proposer (Hsieh et al., 2023). Our key observation is that a weaker auxiliary model can still provide useful short reasoning segments, even when its full solutions are inferior to those of the target model (Lee et al., 2024). Such segments can perturb the target model away from its dominant reasoning pattern and into a different continuation basin (Chia et al., 2024). The auxiliary model therefore does not need to solve the problem; it only needs to propose local branches that expand the target model's exploration support (Yao et al., 2023).

Motivated by this observation, we propose W2SPO, a Weak-to-Strong Off-Policy RL method via Auxiliary Branches for reasoning models (Hou et al., 2025). Given a problem, the target model first generates an intermediate reasoning prefix. We then sample short branches from both the target model and an auxiliary model at this intermediate state (Wei et al., 2022). In our main instantiation, each auxiliary branch contains only 8 tokens. The target model continues from the inserted branch to produce a complete solution, which is scored by a verifier or rule-based reward (Wang et al., 2024). The resulting reward estimates the value of the local branch, and the policy update is applied to the branch segment rather than the entire continuation (Setlur et al., 2024).

Our contributions are summarized as follows:

- **Observation.** We identify a support-limitation failure mode in self-rollout reasoning RL: target-model rollouts can be semantically redundant and provide insufficient reward contrast for GRPO-style optimization.
- **Method.** We propose W2SPO, an auxiliary-branched RL framework that uses a weaker model as a local proposal distribution. The method inserts short auxiliary branches into intermediate target-model trajectories, lets the target model complete the remaining reasoning path, and performs importance-aware policy updates on the local branch segment.
- **Empirical findings.** We show that short auxiliary branches, instantiated as 8-token segments, improve reasoning RL over vanilla GRPO and target-only branching, increase distinct-correct trajectory coverage, and reduce all-wrong rollout groups. We further demonstrate gains with a non-Qwen 7B target and in multi-turn tool use.

2 Related Work

Reasoning RL with verifiable rewards. Recent reasoning models increasingly rely on reinforcement learning with verifiable rewards, where correctness can be judged by rules, symbolic checkers, or learned verifiers (Shao et al., 2024; Guo et al., 2025a; Song et al., 2025b; Wen et al., 2025; Cobbe et al., 2021). GRPO-style methods compare multiple sampled responses to the same problem and update the policy using relative rewards, making them effective for mathematical and programmatic reasoning. However, their effectiveness depends on whether the sampled rollout group contains diverse and useful solution attempts. Prior work on self-consistency shows that multiple trajectories can improve answer reliability when they cover meaningfully different reasoning modes (Wang et al., 2023). On difficult problems, however, repeated samples may remain concentrated in the same reasoning basin, providing limited reward contrast despite surface-level variation (Dang et al., 2025; Yue et al., 2025; Brown et al., 2024). Our work focuses on this support-limited regime and studies how to make rollout groups more informative for policy optimization. Likelihood-aware objectives address a complementary failure mode after trajectories have been sampled. For example, Rewarding the Unlikely up-weights rare correct samples to counter GRPO's distribution sharpening (He et al., 2025a). Such objective-side reweighting can preserve a rare successful trajectory that is already present, whereas W2SPO changes the proposal process so that a successful continuation can enter an otherwise all-wrong group.

External guidance, distillation, and branching. A common way to improve exploration is to introduce external guidance (Deng et al., 2025). Supervised distillation and expert imitation can provide high-quality reasoning traces, but they require strong teachers or curated solutions and may suffer from mismatch between expert traces and the target model's native reasoning style (Kim et al., 2025). Expert-guided branching methods insert partial hints into reasoning trajectories to help models continue from more promising intermediate states (Zhang et al., 2025; Wang et al., 2025). These methods demonstrate the value of intermediate guidance, but they often rely on expert or ground-truth traces. In contrast, W2SPO does not assume access to complete expert solutions. It uses a weaker auxiliary model only as a local proposal distribution, while the target model remains responsible for completing the final reasoning trajectory.

Weak-to-strong learning. Our method is closely related to weak-to-strong learning and model collaboration (Burns et al., 2023; Wang et al., 2026; Hsieh et al., 2023; Lee et al., 2024). Unlike traditional distillation where a strong teacher supervises a weak student, recent evidence suggests that even a weaker model can effectively guide a stronger one by providing diverse local interventions (Wang et al., 2026). This highlights a key distinction between end-to-end capability and local proposal utility: a weaker but behaviorally different model can provide short segments that redirect the target toward continuations it rarely reaches on its own (Chia et al., 2024). By treating the auxiliary model as a source of local perturbations rather than a global solver, W2SPO expands the support of the rollout distribution while preserving the target model's superior long-form reasoning capabilities (Nath et al., 2025). By restricting updates to reasoning over these short inserted segments, we localize credit assignment to the critical decision points that shift downstream trajectories (Roux et al., 2025; Guo et al., 2025b). Thus, W2SPO uses heterogeneous short branches to construct more informative RL rollout groups rather than imitating full auxiliary trajectories (Yan et al., 2025; Lyu et al., 2025; Liu et al., 2025).

3 Motivation

Modern reasoning RL methods with verifiable rewards, such as GRPO, typically rely on multiple self-generated rollouts by the current policy and optimize the model using group-relative reward differences (Shao et al., 2024). This paradigm implicitly assumes that increasing the number of sampled trajectories will produce not only more attempts, but also more informative reasoning alternatives. Such an assumption is consistent with the intuition behind self-consistency, where sampling diverse reasoning paths can improve answer reliability (Wang et al., 2022). However, on difficult reasoning problems (Zhang & Math-AI, 2024; 2025; He et al., 2024; Hendrycks et al., 2021), we observe a clear diminishing-return effect from increased self-sampling: as the sampling budget grows, the proportion of correct trajectories that are also semantically distinct becomes progressively smaller.

Failure mode 1: increasing self-sampling does not necessarily increase reasoning path diversity. For each problem, we sample $K \in \{8, 16, 32\}$ trajectories from the target model A (Qwen3-4B-Instruct-2507 (Team, 2025b)) and measure both final-answer correctness and semantic diversity. Semantic diversity is computed by embedding (bge-small-en-v1.5 (Xiao et al., 2023)) generated reasoning traces and counting semantically distinct trajectories under a cosine-similarity threshold $\tau = 0.95$. As shown in Figure 2, increasing K substantially increases the number of sampled trajectories and often the total number of correct samples. However, the number of semantically distinct correct solutions grows much more slowly, and the effective distinct-correct ratio decreases as K increases. This suggests a clear diminishing-return effect in self-sampling: simply increasing the self-sampling budget is often not an efficient way to expand the effective reasoning support of the target model, as additional samples mainly introduce surface-level variations rather than genuinely new reasoning modes.

This observation is aligned with recent findings on reasoning diversity. Repeated sampling can improve coverage when the model's sampling distribution already contains correct solutions (Brown et al., 2024); however, recent studies show that reasoning models may suffer from diversity collapse, where fine-tuning or RL concentrates probability mass on a narrower set of reasoning paths (Dang et al., 2025). Outcome-based rewarding has also been shown to induce systematic losses in generation diversity, which can undermine test-time scaling (Song et al., 2025a). Moreover, analyses of RLVR suggest that current RL methods often improve sampling efficiency by reweighting existing reasoning paths rather than eliciting fundamentally new reasoning patterns beyond the base model's support (Yue et al., 2025; Wen et al., 2026). These results indicate that the key bottleneck is not merely the number of rollouts, but whether the rollout group covers sufficiently different reasoning modes.

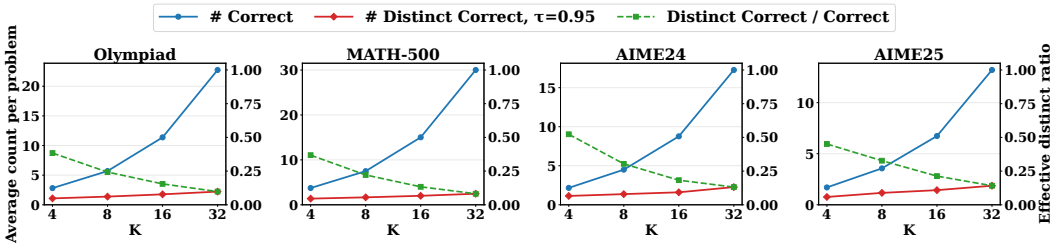

Figure 2: Self-rollout is diminishing-return. As the sampling budget K increases (temperature=0.7, top-p=0.95), the total number of correct samples increases, but the number of semantically distinct correct solutions grows much more slowly. Sensitivity analysis is at C.2.

Failure mode 2: hard problems produce all-wrong groups and no effective update. The limited support of self-rollouts is less effective for RL training. For problems that the target model cannot solve under its current rollout distribution, all sampled responses may receive zero verifier reward. In GRPO-style objectives, such all-wrong groups have little or no reward contrast, leading to zero or near-zero group-relative advantages and therefore no effective policy-gradient update for those prompts. Recent work on zero-advantage or negative rollout groups similarly observes that when all responses in a group receive identical rewards, standard GRPO wastes sampling compute and provides no useful gradient signal (Feng et al., 2025). Thus, increasing the self-rollout budget may still fail to train on the hardest examples if the model cannot sample any successful trajectory within the group.

Auxiliary branches as local support expansion. To address these two limitations, we do not replace the target model with a stronger teacher or imitate a full auxiliary solution. Instead, we use an auxiliary model B (such as gemma-3-4B-IT (Team, 2025a)) as a local proposal distribution: given an intermediate prefix generated by the target model A, B proposes a short branch segment, and then A resumes generation to complete the solution. This forms an A-B-A trajectory: target prefix, auxiliary branch, and target continuation. Thus, the role of B is not to solve the problem end-to-end, but to inject local reasoning turns that may redirect A away from its dominant erroneous basin and into a more productive continuation region.

Under the same sampling budget, this auxiliary-branch construction yields more diverse and effective reasoning trajectories. As shown in Figure 3 (Left), A-B-A produces a higher average number of

Method	Olympiad	MATH500	AIME24	AIME25
Direct	2.23	2.46	2.27	1.87
A-B-A	**2.49**	**2.73**	**2.30**	**2.17**
Rel. Gain	+12.0%	+11.1%	+1.5%	+16.1%

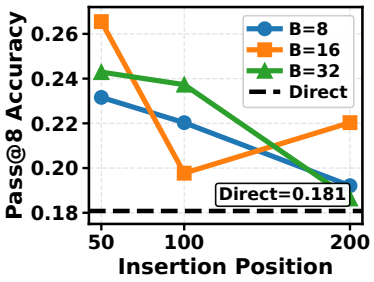

Figure 3: Auxiliary branches expand effective reasoning support. Left: average number of semantically distinct correct trajectories under direct self-rollout and A-B-A inference. Right: Pass@8 performance under different insertion positions and lengths (8, 16, 32 tokens). All samples are generated with temperature=0.7, top-p=0.95, and the same system prompt.

distinct correct trajectories than direct self-rollout inference, supporting the view that short auxiliary branches expand the local continuation support of the target model rather than merely increasing the number of samples. To further test this mechanism on hard problems, we construct a subset where direct sampling from A fails under Pass@4, and evaluate different intervention positions and branch lengths. Auxiliary branches consistently outperform the direct Pass@8 baseline, with the strongest gains from early interventions: the direct baseline achieves only 18.08% Pass@8, while inserting a branch at position 50 improves performance to 23.16%, 26.55%, and 24.29% with branch lengths of 8, 16, and 32 tokens, respectively. Since the 8-token branch at position 50 already provides substantial improvement with the least off-policy context and auxiliary-generation cost, we use this configuration in subsequent reinforcement learning experiments.

Design implication. These findings motivate a branch-level view of reasoning exploration: a short externally proposed segment inserted at the early reasoning stage can largely reshape the target model's continuation distribution, increasing subsequent-sample diversity and improving success on hard problems where target-only self-rollouts often fail.

Our proposal intends to make the target model internalize this ability while retaining its long-horizon reasoning behavior. Our method combines off-policy auxiliary-branch sampling with reinforcement learning over a short injected window: the auxiliary model supplies short branch proposals, the target model completes the trajectory, and the final verifier reward assigns credit only to the inserted branch tokens. This expands local exploration support without distilling full auxiliary trajectories, keeps the update focused on the decision point that changes downstream reasoning, and updates only the target model A while keeping the auxiliary model B fixed as an external proposal distribution. The next section formalizes this branch-level off-policy RL objective.

4 METHOD

We propose a Weak-to-Strong Off-Policy (W2SPO) RL method for improving LLM reasoning, as shown in Figure 1. The method consists of five steps: greedy prefix generation, short branch proposal, rollout-based branch scoring, importance-weighted advantage estimation, and clipped policy optimization.

Step 1: Greedy prefix generation. For each prompt x, the actor first generates a reasoning prefix z of length L_p under greedy decoding:

$$z = \mathrm{GreedyDecode}(\pi_{\theta_{\mathrm{old}}}, x; L_p).$$

This prefix represents the early reasoning context of the target model and serves as the intervention point for subsequent branch exploration.

Step 2: Short branch proposal. Starting from the prefix (x, z), we construct short candidate branches from two sources. The actor samples n_A branches,

$$b_i^A \sim \pi_{\theta_{\mathrm{old}}}(\cdot \mid x, z), \qquad i = 1, \ldots, n_A,$$

while a frozen auxiliary model samples n_B textual continuations,

$$\tilde{b}_j^B \sim q_B(\cdot \mid x, z), \qquad j = 1, \ldots, n_B.$$

Since the actor and auxiliary model may use different tokenizers, auxiliary continuations are decoded into text and re-encoded by the actor tokenizer. All branches are truncated or padded to the same actor-token length L_b. In our configuration, $L_p = 50$, $L_b = 8$, $n_A = 2$, and $n_B = 6$, giving eight candidate branches per prompt.

Step 3: Rollout-based branch scoring. Each branch b_i is evaluated by letting the actor continue from the branched context (x, z, b_i) for K rollouts:

$$u_{i,k} \sim \pi_{\theta_{\text{old}}}(\cdot \mid x, z, b_i), \qquad k = 1, \ldots, K.$$

A task-specific rule-based reward function compares the completed solution with the ground-truth label $y^\star$. The branch reward is defined as the mean rollout reward:

$$r_i = \frac{1}{K} \sum_{k=1}^{K} R(x, z, b_i, u_{i,k}, y^\star).$$

Therefore, a branch is not judged by its local text alone, but by whether it leads the actor toward successful future reasoning. During training, the action mask is applied only to the branch tokens.

Step 4: Importance-weighted branch advantage. Because branches are sampled from a mixture of the actor and the auxiliary model, we compute an importance-weighted group advantage.

Let

$$\bar{p}_i \;=\; \frac{1}{L_b} \sum_{t=1}^{L_b} \log \pi_{\theta_{\text{old}}}(b_{i,t} \mid x, z, b_{i,<t}) \tag{1}$$

be the *length-normalized* (per-token mean) old-actor log-probability of branch b_i, and let $\bar{q}_i$ be the length-normalized log-probability of the same branch under the distribution that proposed it. Both quantities are per-token averages, so that they remain on a comparable scale across sources despite the different branch encodings introduced by tokenizer conversion. For actor-sampled branches the proposal is the actor itself, so $\bar{q}_i = \bar{p}_i$ by construction and the coefficient below reduces exactly to one. For auxiliary-sampled branches, $\bar{q}_i$ is the auxiliary model's per-token log-probability of the generated text under its own tokenizer.

The importance coefficient is

$$c_i = \min\left\{c_{\max}, \exp\left(\alpha \cdot \text{clip}\left(\bar{p}_i - \bar{q}_i, -30, 30\right)\right)\right\},$$

where we use $\alpha = 0.02$ and $c_{\max} = 2.0$.

For each prompt group, we compute the weighted mean and variance:

$$\mu = \frac{\sum_i c_i r_i}{\sum_i c_i}, \qquad \sigma = \sqrt{\frac{\sum_i c_i (r_i - \mu)^2}{\sum_i c_i} + \delta}.$$

A leave-one-out weighted baseline is then used:

$$\hat{b}_i = \frac{\sum_j c_j r_j - c_i r_i}{\sum_j c_j - c_i + \delta}.$$

Here $\delta > 0$ is a small constant for numerical stability, distinct from the PPO clipping threshold ϵ introduced in Step 5.

The final branch-level advantage is

$$A_i = \frac{c_i (r_i - \hat{b}_i)}{\sigma},$$

which is clipped to $[-3, 3]$. Prompt groups with zero reward variance are assigned zero advantage. The coefficient c_i serves as a bounded proposal-balancing weight. Tempering and clipping control the influence of cross-model proposals while accommodating source-specific likelihoods and tokenizer

conversion. To isolate possible effects of non-centering, we also evaluate an explicitly centered variant,

$$A_i^{\mathrm{ctr}} = A_i - \frac{1}{G}\sum_{j=1}^{G} A_j,$$

which preserves the auxiliary proposals while removing the group-level offset.

Step 5: Clipped policy update on branch tokens. Finally, the actor is optimized using the PPO clipped objective, but gradients are applied only to the masked branch tokens. For each branch token, we define

$$\rho_{i,t}(\theta) = \exp\left(\log \pi_\theta(b_{i,t} \mid x, z, b_{i,<t}) - \log \pi_{\theta_{\mathrm{old}}}(b_{i,t} \mid x, z, b_{i,<t})\right).$$

The policy loss is

$$\mathcal{L}_{\mathrm{policy}} = -\mathbb{E}_{i,t}\left[\min\left(\rho_{i,t}(\theta)A_i, \mathrm{clip}(\rho_{i,t}(\theta), 1-\epsilon, 1+\epsilon)A_i\right)\right],$$

with $\epsilon = 0.2$. We use the PPO clipped surrogate objective without an additional KL-divergence regularization term; the reward signal is provided by the rule-based verifier.

Thus, the update explicitly increases the actor's probability of short reasoning branches that lead to correct final solutions, while leaving the prefix and continuation tokens unoptimized. An example rollout is provided in Appendix C.1.

Table 1: Main greedy decoding performance and runtime comparison. Top: greedy decoding performance measured by Pass@1. Bottom: runtime comparison between matched-sampling GRPO and W2SPO(ours) under the same sampling budget and sampling hyperparameters. Runtime measurements are collected on 8×A800 GPUs, and we report per-step mean±std in seconds.

(a) Greedy decoding performance measured by Pass@1.

Model	Overall	MATH500	Olympiad	Omni-HARD	AIME24	AIME25
Qwen3-4B-Instruct-2507	58.3	90.8	64.4	21.2	50.0	40.0
Gemma-3-4B-IT	40.7	75.2	39.9	13.4	3.3	10.0
ThinkTwice-Qwen3-4B-Instruct	59.4	92.2	66.6	22.1	36.7	30.0
Qwen3-4B-DAPO-math-reasoning	59.8	93.4	65.7	23.1	36.7	40.0
Frugal-Thinking-4B	61.4	92.8	69.3	23.3	56.7	40.0
NPR-4B	52.0	86.0	57.6	15.1	40.0	23.3
Polaris-4B-Preview	43.4	78.2	44.8	10.6	30.0	20.0
Qwen3-4B-Reinforce-Ada	61.1	92.0	69.1	23.6	43.3	43.3
Matched-sampling GRPO	62.3	92.6	69.3	26.5	53.3	40.0
W2SPO (ours)	**64.2**	93.2	**71.8**	**28.4**	**56.7**	**46.7**

(b) Runtime breakdown under matched sampling budget.

Method	Process (s)	Forward (s)	Train (s)	Total (s)
GRPO	782.22 ± 67.08	81.44 ± 9.41	945.11 ± 34.04	1808.78 ± 89.98
W2SPO(ours)	482.61 ± 35.81	2.09 ± 0.29	24.40 ± 0.78	509.11 ± 35.91
Speedup	$\mathbf{1.62}\times$	$\mathbf{38.97}\times$	$\mathbf{38.73}\times$	$\mathbf{3.55}\times$

5 Experiments

5.1 Training Setup

In our experiment, the trainable actor model A is initialized from Qwen3-4B-Instruct-2507, while an auxiliary model B initialized from Gemma-3-4B-IT is used only as a frozen off-policy branch proposer. Our training data is sampled from Skywork-OR1-RL-Data (He et al., 2025b;c), with all problems identical or overlapping with the evaluation benchmarks removed from the training set. More details on training experiments can be found in Appendix C.3.

5.2 Evaluation Protocol

We evaluate models on a collection of mathematical reasoning benchmarks, including MATH500, Olympiad-test, Omni-Math (Gao et al., 2024; An et al., 2025), AIME 2024, and AIME 2025. We

evaluate on the subset of Omni-MATH with difficulty levels greater than or equal to 8, referred to as Omni-HARD throughout this paper. The evaluation set contains 1763 problems in total. For each generated solution, correctness is determined using a combination of regular-expression-based answer extraction and the `math-verify` toolkit. All evaluations are conducted under the same system prompt, conversation format, and context-length setting to ensure a fair comparison across methods.

We compare our method with both pretrained open-source models and training-based approaches. We include Qwen3-4B-Instruct-2507 as the direct initialization baseline. This baseline measures the performance before reinforcement learning. We also compare W2SPO against recent open-source 4B reasoning models, including models trained with supervised reasoning distillation, RLVR, and adaptive RL sampling (Jiao et al., 2026; Yu et al., 2025; Bounhar et al., 2025; Wu et al., 2025; Xiong et al., 2025). All baselines are evaluated under the same decoding and verification protocol.

To isolate the effect of auxiliary branch construction from the total number of samples, we include a standard GRPO baseline with the same number of samples per prompt.

5.3 Main Results

Table 1 details the greedy decoding performance and runtime efficiency. Our method improved the actor's most likely reasoning path by assigning higher probability to short branches that lead to successful completions. Compared with the base model, this evaluates whether auxiliary-branch training improves the deterministic solution quality. Compared with matched-sampling GRPO and the A-only branch ablation, it evaluates whether the improvement comes from the proposed auxiliary branch mechanism rather than from additional sampling alone.

As shown in Table 1 (a), our method achieves strong Pass@1 accuracy across the evaluated models. Relative to matched-sampling GRPO, W2SPO outperforms it on all five benchmarks and improves the aggregate score from 62.3 to 64.2, with gains of 1.9 points on Omni-HARD and 6.7 points on AIME25. The controlled A-only comparison in Table 3 further isolates the effect of changing the proposal source while preserving the branch structure and sampling budget, showing that the improvement is not explained by additional sampling alone. Table 1 (b) shows a 38.97× forward-pass speedup and a 3.55× total training speedup.

5.4 Sampling-Based Evaluation

Beyond greedy Pass@1, we further evaluate whether RL training preserves the model's exploration boundary under large sampling budgets. Prior work has reported that reasoning RL can improve single-sample accuracy while degrading Pass@k, suggesting a shrinkage of the model's effective capability frontier (Yue et al., 2025). we sample 256 responses on AIME24 and AIME25 and report unbiased Pass@k estimates from these samples. As shown in Figure 4, our method yields a much smaller drop from the base model than GRPO on AIME25, indicating that it better preserves the target model's multi-sample exploration ability while still benefiting from RL training. The result on AIME24 is shown in Figure 8.

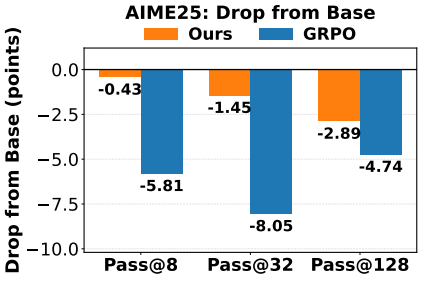

Figure 4: Drop in Pass@k relative to the base model on AIME25. Smaller drops indicate better preservation of multi-sampling performance after RL training.

We additionally report average sampled Pass@1 over four independent samples per problem using temperature 0.7, top-p = 0.8, top-k = 20, and a maximum generation length of 8,192. Table 2 shows that the gain over the base model is not specific to greedy decoding.

5.5 Ablation Study

Table 3 reports the ablation results on greedy Pass@1 performance across five mathematical reasoning benchmarks. Compared with the base model, W2SPO improves the aggregate score from 58.3% to 64.2%. Replacing the six auxiliary branches with six target-model branches reduces the score to 61.9%, while the controlled DAPO-style baseline obtains 60.6%. Explicitly centering the weighted

Table 2: Average sampled Pass@1 over four samples per problem.

Model	MATH500	Olympiad	Omni-HARD	AIME24	AIME25
Qwen3-4B-Instruct-2507	90.65	66.06	20.37	52.50	39.17
W2SPO (ours)	**93.30**	**72.89**	**27.74**	**54.17**	**48.33**

Table 3: Greedy Pass@1 results for controlled baselines and ablations. All values are percentages over 1,763 problems.

Model	Overall	MATH500	Olympiad	Omni-HARD	AIME24	AIME25
Qwen3-4B-Instruct-2507	58.3	90.8	64.4	21.2	50.0	40.0
Unbalanced Ablation	62.6	94.0	69.7	25.7	46.7	46.7
A-only Branch Ablation	61.9	93.2	70.6	23.3	53.3	36.7
Controlled DAPO-style baseline	60.6	91.6	68.4	23.1	53.3	40.0
W2SPO, centered advantage	64.1	93.0	71.4	28.2	**60.0**	46.7
W2SPO (ours)	**64.2**	93.2	**71.8**	**28.4**	56.7	**46.7**

advantages retains nearly all of the gain (64.1%), indicating that a non-zero group-wise offset is not the primary explanation. The unbalanced variant reaches 62.6%, supporting the use of bounded proposal balancing.

Figure 5 further compares our method with the A-only branch ablation under a controlled setting, where all hyperparameters, sampling configurations, and the random seed are kept identical except for replacing B-sampled branches with A-sampled branches. Our method yields a larger within-group reward standard deviation and a higher non-zero advantage ratio, indicating stronger reward contrast and more usable policy updates.

Robustness experiments further show that our method remains stable under both random-seed variation and auxiliary-model replacement. We defer detailed results to Appendix C.4.

6 Conclusion and Limitations

We introduced W2SPO, a weak-to-strong off-policy RL method for reasoning and tool-using policies. Rather than distilling full auxiliary solutions, W2SPO uses a weaker frozen model to propose short local branches, lets the target model complete the reasoning trajectory, and applies verifier-based policy updates only to the inserted branch tokens. Experiments show that W2SPO improves over matched-sampling GRPO and target-only branching under the same evaluation protocol, achieving stronger greedy performance while better preserving multi-sampling capability. Our current study is limited by computational resource constraints: the experiments focus on relatively small-scale language models, with the main evaluation on mathematical reasoning and a limited additional evaluation on multi-turn tool use.

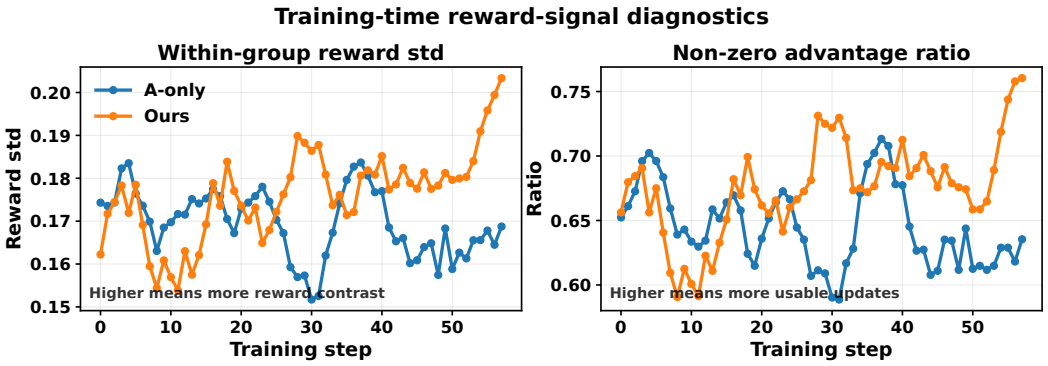

Figure 5: Training-time reward-signal diagnostics for the controlled A-only branch ablation.

AI USE STATEMENT

Generative AI tools were used for language editing and for converting the manuscript to the ICLR 2027 LaTeX format. They were not used to generate experimental data or reported results. The authors reviewed the AI-assisted text, citations, numerical claims, and formatting changes, and take responsibility for the final content of this work.

ETHICS STATEMENT

This work studies training algorithms for language-model reasoning and tool use. It does not involve human subjects or private user data. The evaluated models can inherit the limitations and potential harms of their underlying pretrained models; deployment in consequential settings requires task-specific safety evaluation and monitoring.

REPRODUCIBILITY STATEMENT

The anonymous repository contains training and evaluation code, configurations, and data-processing instructions. Section C.3 specifies the training data construction, hyperparameters, compute setup, and efficiency measurement, while Appendix D provides the assumptions and derivations used in the analysis.

REFERENCES

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

Victor Barres, Honghua Dong, Soham Ray, Xujie Si, and Karthik Narasimhan. τ^2-Bench: Evaluating conversational agents in a dual-control environment. *arXiv preprint arXiv:2506.07982*, 2025.

Abdelaziz Bounhar, Hadi Abdine, Evan Dufraisse, Ahmad Chamma, Amr Mohamed, Dani Bouch, Michalis Vazirgiannis, and Guokan Shang. Shorter but not worse: Frugal reasoning via easy samples as length regularizers in math rlvr. *arXiv preprint arXiv:2511.01937*, 2025.

Bradley Brown, Jordan Juravsky, Ryan Ehrlich, Ronald Clark, Quoc V Le, Christopher Ré, and Azalia Mirhoseini. Large language monkeys: Scaling inference compute with repeated sampling. *arXiv preprint arXiv:2407.21787*, 2024.

Collin Burns, Pavel Izmailov, Jan Hendrik Kirchner, Bowen Baker, Leo Gao, Leopold Aschenbrenner, Yining Chen, Adrien Ecoffet, Manas Joglekar, Jan Leike, Ilya Sutskever, and Jeff Wu. Weak-to-strong generalization: Eliciting strong capabilities with weak supervision, 2023. URL https://arxiv.org/abs/2312.09390.

Peter Chen, Xiaopeng Li, Ziniu Li, Xi Chen, and Tianyi Lin. Stepwise guided policy optimization: Coloring your incorrect reasoning in grpo, 2026. URL https://arxiv.org/abs/2505.11595.

Yew Ken Chia, Guizhen Chen, Weiwen Xu, Luu Anh Tuan, Soujanya Poria, and Lidong Bing. Reasoning paths optimization: Learning to reason and explore from diverse paths. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 16763–16780, 2024.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems, 2021. URL https://arxiv.org/abs/2110.14168.

Xingyu Dang, Christina Baek, J Zico Kolter, and Aditi Raghunathan. Assessing diversity collapse in reasoning. In *Scaling Self-Improving Foundation Models without Human Supervision*, 2025.

DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. URL https://arxiv.org/abs/2501.12948.

Yihe Deng, I Hsu, Jun Yan, Zifeng Wang, Rujun Han, Gufeng Zhang, Yanfei Chen, Wei Wang, Tomas Pfister, Chen-Yu Lee, et al. Supervised reinforcement learning: From expert trajectories to step-wise reasoning. *arXiv preprint arXiv:2510.25992*, 2025.

Yihong Dong, Xue Jiang, Yongding Tao, Huanyu Liu, Kechi Zhang, Lili Mou, Rongyu Cao, Yingwei Ma, Jue Chen, Binhua Li, Zhi Jin, Fei Huang, Yongbin Li, and Ge Li. Rl-plus: Countering capability boundary collapse of llms in reinforcement learning with hybrid-policy optimization, 2026. URL https://arxiv.org/abs/2508.00222.

Yunzhen Feng, Parag Jain, Anthony Hartshorn, Yaqi Duan, and Julia Kempe. Don't waste mistakes: Leveraging negative rl-groups via confidence reweighting. *arXiv preprint arXiv:2510.08696*, 2025.

Bofei Gao, Feifan Song, Zhe Yang, Zefan Cai, Yibo Miao, Qingxiu Dong, Lei Li, Chenghao Ma, Liang Chen, Runxin Xu, Zhengyang Tang, Benyou Wang, Daoguang Zan, Shanghaoran Quan, Ge Zhang, Lei Sha, Yichang Zhang, Xuancheng Ren, Tianyu Liu, and Baobao Chang. Omni-math: A universal olympiad level mathematic benchmark for large language models, 2024. URL https://arxiv.org/abs/2410.07985.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Hanwei Xu, Honghui Ding, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jingchang Chen, Jingyang Yuan, Jinhao Tu, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaichao You, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingxu Zhou, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645 (8081):633–638, September 2025a. ISSN 1476-4687. doi: 10.1038/s41586-025-09422-z. URL http://dx.doi.org/10.1038/s41586-025-09422-z.

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.

Andre Wang He, Daniel Fried, and Sean Welleck. Rewarding the unlikely: Lifting grpo beyond distribution sharpening. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 25559–25571, 2025a.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, et al. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3828–3850, 2024.

Jujie He, Jiacai Liu, Chris Yuhao Liu, Rui Yan, Chaojie Wang, Peng Cheng, Xiaoyu Zhang, Fuxiang Zhang, Jiacheng Xu, Wei Shen, Siyuan Li, Liang Zeng, Tianwen Wei, Cheng Cheng, Bo An, Yang Liu, and Yahui Zhou. Skywork open reasoner 1 technical report. *arXiv preprint arXiv:2505.22312*, 2025b.

Jujie He, Jiacai Liu, Chris Yuhao Liu, Rui Yan, Chaojie Wang, Peng Cheng, Xiaoyu Zhang, Fuxiang Zhang, Jiacheng Xu, Wei Shen, Siyuan Li, Liang Zeng, Tianwen Wei, Cheng Cheng, Yang Liu, and Yahui Zhou. Skywork open reasoner series. `https://capricious-hydrogen-41c.notion.site/Skywork-Open-Reaonser-Series-1d0bc9ae823a80459b46c149e4f51680`, 2025c. Notion Blog.

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.

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.

Cheng-Yu Hsieh, Chun-Liang Li, Chih-Kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alex Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 8003–8017, 2023.

Jian Hu, Xibin Wu, Wei Shen, Jason Klein Liu, Weixun Wang, Songlin Jiang, Haoran Wang, Hao Chen, Bin Chen, Wenkai Fang, et al. Openrlhf: A ray-based easy-to-use, scalable and high-performance rlhf framework. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 656–666, 2025.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023. URL `https://arxiv.org/abs/2310.06825`.

Difan Jiao, Qianfeng Wen, Blair Yang, Zhenwei Tang, and Ashton Anderson. Thinktwice: Jointly optimizing large language models for reasoning and self-refinement. *arXiv preprint arXiv:2604.01591*, 2026.

Jaehoon Kim, Kwangwook Seo, and Dongha Lee. In their own words: Reasoning traces tailored for small models make them better reasoners. *arXiv preprint arXiv:2509.22230*, 2025.

Jooyoung Lee, Fan Yang, Thanh Tran, Qian Hu, Emre Barut, and Kai-Wei Chang. Can small language models help large language models reason better?: Lm-guided chain-of-thought. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pp. 2835–2843, 2024.

Pengyi Li, Elizaveta Goncharova, Andrey Kuznetsov, and Ivan Oseledets. Back to basics: Revisiting exploration in reinforcement learning for llm reasoning via generative probabilities. *arXiv preprint arXiv:2602.05281*, 2026.

Zheng Liu, Mengjie Liu, Siwei Wen, Mengzhang Cai, Bin Cui, Conghui He, and Wentao Zhang. From uniform to heterogeneous: Tailoring policy optimization to every token's nature. *arXiv preprint arXiv:2509.16591*, 2025.

Yuanjie Lyu, Chengyu Wang, Jun Huang, and Tong Xu. From correction to mastery: Reinforced distillation of large language model agents. *arXiv preprint arXiv:2509.14257*, 2025.

Vaskar Nath, Elaine Lau, Anisha Gunjal, Manasi Sharma, Nikhil Baharte, and Sean Hendryx. Adaptive guidance accelerates reinforcement learning of reasoning models. *arXiv preprint arXiv:2506.13923*, 2025.

Team Olmo, Allyson Ettinger, Amanda Bertsch, Bailey Kuehl, David Graham, David Heineman, Dirk Groeneveld, Faeze Brahman, Finbarr Timbers, Hamish Ivison, et al. Olmo 3. *arXiv preprint arXiv:2512.13961*, 2025.

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.

Nicolas Le Roux, Marc G Bellemare, Jonathan Lebensold, Arnaud Bergeron, Joshua Greaves, Alex Fréchette, Carolyne Pelletier, Eric Thibodeau-Laufer, Sándor Toth, and Sam Work. Tapered off-policy reinforce: Stable and efficient reinforcement learning for llms. *arXiv preprint arXiv:2503.14286*, 2025.

Amrith Setlur, Chirag Nagpal, Adam Fisch, Xinyang Geng, Jacob Eisenstein, Rishabh Agarwal, Alekh Agarwal, Jonathan Berant, and Aviral Kumar. Rewarding progress: Scaling automated process verifiers for llm reasoning. *arXiv preprint arXiv:2410.08146*, 2024.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024. URL https://arxiv.org/abs/2402.03300.

Yuda Song, Julia Kempe, and Remi Munos. Outcome-based exploration for llm reasoning. *arXiv preprint arXiv:2509.06941*, 2025a.

Yuda Song, Julia Kempe, and Remi Munos. Outcome-based exploration for llm reasoning, 2025b. URL https://arxiv.org/abs/2509.06941.

Gemma Team. Gemma 3, 2025a. URL https://goo.gle/Gemma3Report.

Qwen Team. Qwen3 technical report, 2025b. URL https://arxiv.org/abs/2505.09388.

Dayu Wang, Jiaye Yang, Weikang Li, Jiahui Liang, and Yang Li. Student guides teacher: Weak-to-strong inference via spectral orthogonal exploration. *arXiv preprint arXiv:2601.06160*, 2026.

Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. Math-shepherd: Verify and reinforce llms step-by-step without human annotations. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9426–9439, 2024.

Xinyi Wang, Jinyi Han, Zishang Jiang, Tingyun Li, Jiaqing Liang, Sihang Jiang, Zhaoqian Dai, Shuguang Ma, Fei Yu, and Yanghua Xiao. Hint: Helping ineffective rollouts navigate towards effectiveness. *arXiv preprint arXiv:2510.09388*, 2025.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2022.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models, 2023. URL https://arxiv.org/abs/2203.11171.

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.

Xumeng Wen, Zihan Liu, Shun Zheng, Shengyu Ye, Zhirong Wu, Yang Wang, Zhijian Xu, Xiao Liang, Junjie Li, Ziming Miao, Jiang Bian, and Mao Yang. Reinforcement learning with verifiable rewards implicitly incentivizes correct reasoning in base llms, 2025. URL https://arxiv.org/abs/2506.14245.

Xumeng Wen, Zihan Liu, Shun Zheng, Zhijian Xu, Shengyu Ye, Zhirong Wu, Xiao Liang, Yang Wang, Junjie Li, Ziming Miao, Jiang Bian, and Mao Yang. Reinforcement learning with verifiable rewards implicitly incentivizes correct reasoning in base LLMs. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2026.

Tong Wu, Yang Liu, Jun Bai, Zixia Jia, Shuyi Zhang, Ziyong Lin, Yanting Wang, Song-Chun Zhu, and Zilong Zheng. Native parallel reasoner: Reasoning in parallelism via self-distilled reinforcement learning. *arXiv preprint arXiv:2512.07461*, 2025.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. C-pack: Packaged resources to advance general chinese embedding, 2023.

Wei Xiong, Chenlu Ye, Baohao Liao, Hanze Dong, Xinxing Xu, Christof Monz, Jiang Bian, Nan Jiang, and Tong Zhang. Reinforce-ada: An adaptive sampling framework under non-linear rl objectives. *arXiv preprint arXiv:2510.04996*, 2025.

Anyi Xu, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, Chenchen Ling, et al. Deepseek-v4: Towards highly efficient million-token context intelligence. *arXiv preprint arXiv:2606.19348*, 2026.

Jianhao Yan, Yafu Li, Zican Hu, Zhi Wang, Ganqu Cui, Xiaoye Qu, Yu Cheng, and Yue Zhang. Learning to reason under off-policy guidance. *arXiv preprint arXiv:2504.14945*, 2025.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 36:11809–11822, 2023.

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.

Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model? *arXiv preprint arXiv:2504.13837*, 2025.

Xuechen Zhang, Zijian Huang, Yingcong Li, Chenshun Ni, Jiasi Chen, and Samet Oymak. Bread: Branched rollouts from expert anchors bridge sft & rl for reasoning. *arXiv preprint arXiv:2506.17211*, 2025.

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.

A Generalization Experiments

Table 4 summarizes two additional experiments. First, using the non-Qwen OLMo-3-7B-Instruct target (Olmo et al., 2025), W2SPO improves greedy MATH-500 accuracy from 48.8% to 57.4% after training on the MATH training split. Second, on the Airline subset of τ^2-Bench (Barres et al., 2025), W2SPO improves average binary trajectory reward from 0.2750 to 0.3375 over the same 20 test tasks and four independent trials per task. In the agent setting, the 8-token branch is inserted at the beginning of the second agent turn and optimized using final task success. Together, these experiments extend W2SPO beyond the primary Qwen-4B mathematical-reasoning setting to a non-Qwen 7B target and a multi-turn tool-use environment.

B Analysis: Local Credit Assignment and Importance-Aware Branch Weighting

We provide a local analysis to explain why short auxiliary branches can yield useful training signals. We analyze W2SPO under a fixed-prefix, stop-gradient approximation that matches our implementation, where rewards are obtained from completed rollouts but policy gradients are applied only to the inserted branch tokens.

Table 4: Additional generalization experiments. "Change" is absolute for accuracy and relative for reward.

Target / setting	Metric	Base	+W2SPO	Change
OLMo-3-7B-Instruct / MATH-500	Accuracy (%)	48.8	**57.4**	+8.6 pp
Qwen3-4B-Instruct / τ^2-Bench Airline	Avg. reward	0.2750	**0.3375**	+22.7% rel.

For the local analysis, we use c, z, and y for the prefix, branch, and continuation, corresponding to z, b_i, and $u_{i,k}$ in Section 4, respectively.

Let x be a problem, c a fixed actor-generated prefix, z a short branch, and y the actor continuation after the branch. We define the stop-gradient downstream value

$$Q_{\bar{\theta}}(c, z) = \mathbb{E}_{y \sim \pi_{\bar{\theta}}(\cdot|x,c,z)} \left[R(x, c, z, y) \right], \tag{2}$$

and the local branch objective

$$J_{\mathrm{br}}(\theta; c, \bar{\theta}) = \mathbb{E}_{z \sim \pi_\theta(\cdot|x,c)} \left[Q_{\bar{\theta}}(c, z) \right]. \tag{3}$$

Under this approximation,

$$\nabla_\theta J_{\mathrm{br}} = \mathbb{E}_{z \sim \pi_\theta} \left[Q_{\bar{\theta}}(c, z) \nabla_\theta \log \pi_\theta(z \mid x, c) \right], \tag{4}$$

where $\log \pi_\theta(z \mid x, c)$ decomposes over the L_b branch tokens. Thus, branch-token masking can be interpreted as optimizing a local decision whose value is estimated by downstream verifier rewards, rather than treating the full continuation as supervised data.

Auxiliary branches are off-policy proposals, so we do not use an exact unbiased importance-sampling estimator. Instead, we use a tempered and clipped coefficient c_i to bound their influence in the branch-level advantage estimate. This allows successful auxiliary proposals to affect the actor while preventing highly mismatched branches from dominating the update. This analysis motivates three diagnostics: higher within-group reward contrast, fewer all-wrong groups, and a reasonable effective sample size under the balancing weights. Detailed assumptions and derivations are provided in Appendix D.

C More Experimental Details and Results

C.1 Example Rollout

We present one rollout group from W2SPO to illustrate how branch-level rewards are assigned.

The system prompt is :

```
Please reason step by step, and put your final answer within
\boxed{}.
```

The problem is:

> A 3 by 2 rectangle is split into four congruent right-angled triangles. Those four triangles are rearranged to form a rhombus. What is the ratio of the perimeter of the rectangle to the perimeter of the rhombus?

The ground-truth answer is 1. The target model A first generates the following prefix, after which short candidate branches are sampled from either the target model itself or the auxiliary model B.

Target prefix generated by A:
We are given:
- A 3 by 2 rectangle is split into four congruent right-angled triangles.
- These triangles are rearranged to form a rhombus.
- We are to find the ratio

For this rollout group, we sample $n_A = 2$ target branches and $n_B = 6$ auxiliary branches. Each candidate branch is inserted after the same prefix, and the target model A generates K continuations from the resulting context. Let r_i denote the mean verifier reward over these continuations, as defined in Section 4. We compute the proposal-balancing coefficient and branch-level advantage. For readability, we display only one continuation of Branch 0.

In this group, the rewards are

$$[0, 0, 1, 1, 1, 1, 1, 0],$$

with weighted mean $\mu = 0.625150$ and weighted standard deviation $\sigma = 0.484084$.

Below, we illustrate the prefix–branch–continuation structure using one target-sampled branch.

A self-branch sampled from A. For Branch 0, the target model samples the branch and then resumes generation from the resulting context. The displayed continuation receives verifier reward 0.

Prefix generated by A:
We are given:
- A 3 by 2 rectangle is split into four congruent right-angled triangles.
- These triangles are rearranged to form a rhombus.
- We are to find the ratio

Inserted branch from A:
of the perimeter of the rectangle to the

Continuation generated by A (excerpt):
perimeter of the rhombus.
We'll solve this step by step.
Step 1: Understand the rectangle. The rectangle has dimensions length 3 and width 2, so its area is $3 \times 2 = 6$.
Since it is split into four congruent right-angled triangles, each triangle has area $6/4 = 1.5$.
[Remaining continuation omitted.]

The branch-level scoring terms are:

$$\bar{p}_i = -0.000738, \quad \bar{q}_i = -0.000738, \quad c_i = 1.000000,$$
$$\hat{b}_i = 0.714384, \quad A_i = -1.475743.$$

Here $\bar{p}_i$ and $\bar{q}_i$ denote per-token mean log-probabilities. For this actor-sampled branch, $\bar{q}_i = \bar{p}_i$, so $c_i = 1$. The branch-level reward is compared with the leave-one-out baseline to obtain the advantage used for the branch-token update.

C.2 Sensitivity Analysis

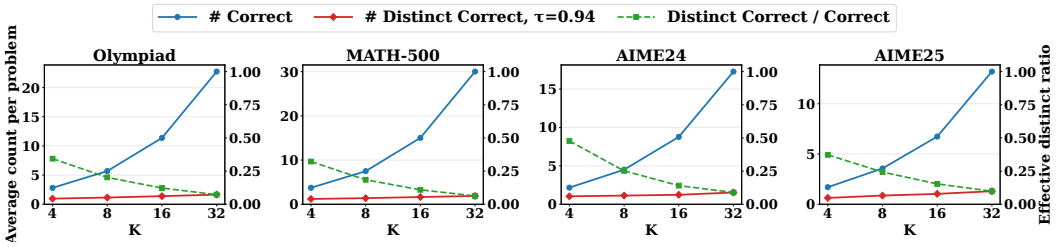

Figure 6: Semantic diversity of correct trajectories under similarity threshold 0.94.

We further conduct a sensitivity analysis on the semantic similarity threshold used to identify distinct correct trajectories. Figures 6 and 7 report results under thresholds of 0.94 and 0.96, respectively. The overall trend remains consistent across both settings. This indicates that our motivation experiment is robust to reasonable choices of the semantic diversity threshold.

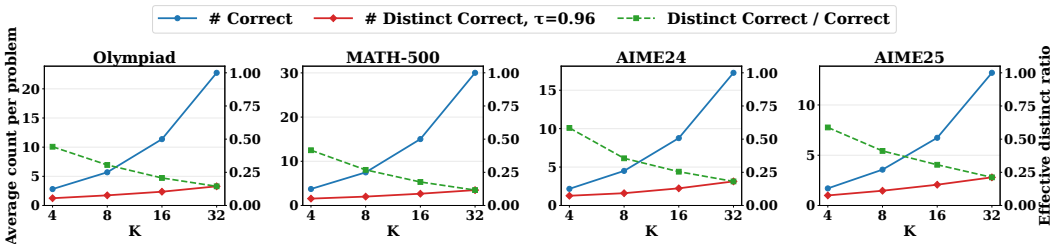

Figure 7: Semantic diversity of correct trajectories under similarity threshold 0.96.

C.3 Experimental Details

Training Setup. Training is implemented in an edited OpenRLHF(Hu et al., 2025) pipeline with Ray, vLLM, DeepSpeed ZeRO(Rasley et al., 2020), bf16 precision, FlashAttention, and gradient checkpointing. At each iteration, prompts are sampled from the training set, converted with the tokenizer chat template, expanded into branch-level samples, scored by a rule-based verifier, and used to update only the actor model.

Training data processing. We construct the training set from the mathematical training split of Skywork-OR1-RL-Data. To avoid benchmark contamination, we first remove problems that are identical or highly similar to examples in the evaluation benchmarks. From the remaining pool, we select problems whose difficulty scores under DeepSeek-R1-Distill-Qwen-32B (DeepSeek-AI, 2025) fall between 12 and 15. We then evaluate Qwen3-4B-Instruct-2507 with eight sampled responses per problem and retain 2,956 problems with partial success, i.e., problems whose Pass@8 count is greater than 0 and smaller than 8. To further include challenging examples that the actor cannot solve under its current sampling distribution, we additionally select 800 problems with the shortest labels from the subset with Pass@8 count equal to 0. This yields a final training set of 3,756 mathematical reasoning problems.

Core training hyperparameters. For W2SPO training, we use Qwen3-4B-Instruct-2507 as the trainable actor and Gemma-3-4B-IT as a frozen auxiliary branch proposer. For each prompt, we construct a branch group with eight candidate branches: two sampled from the actor and six sampled from the auxiliary model. The intervention prefix length is set to $L_p = 50$, and each inserted branch contains $L_b = 8$ actor tokens. Each branch is evaluated with two actor continuations, with a maximum generation length of 8192 tokens. We train for two epochs using a learning rate of 1×10^{-6}, PPO clipping threshold $\epsilon = 0.2$, and the importance-aware branch advantage estimator with $\alpha = 0.02$, $c_{\max} = 2.0$, and advantage clipping at 3.0. Rollouts are sampled with temperature 1.0 and top-p 1.0. Training uses a rollout batch size of 64 and a training batch size of 96, with bf16 precision, FlashAttention, gradient checkpointing, and DeepSpeed ZeRO stage 2. In total, the setup uses two GPUs for actor training and six single-GPU vLLM engines for rollout generation.

Efficiency analysis. W2SPO is more efficient than matched-sampling GRPO because the policy-optimization computation is localized to the short branch window. In standard GRPO, the actor must compute log-probabilities and gradients over complete sampled trajectories, whose lengths can be thousands of tokens in mathematical reasoning tasks. In contrast, W2SPO only needs to run the actor forward over the prefix and the inserted branch when computing branch-token log-probabilities, and the policy loss is applied only to the $L_b = 8$ branch tokens. Although complete continuations are still generated to obtain verifier rewards, they are treated as downstream rollouts rather than tokens to be optimized. As a result, both the forward computation for policy evaluation and the backward computation for gradient updates are substantially reduced.

This localized update also reduces activation memory, since gradients are only required for the short branch segment instead of the full reasoning trajectory. The saved memory can be allocated to rollout generation, allowing larger effective memory budgets for the vLLM engines. Consequently, under the same overall sampling budget, W2SPO not only reduces the cost of policy optimization, but also slightly improves rollout throughput. These factors together explain the large reduction in training time observed in Table 1, especially in the forward and training stages.

C.4 Robustness Analysis

To examine robustness, we evaluate a second training seed and replace the default auxiliary model with `Mistral-7B-Instruct-v0.3` (Jiang et al., 2023). The two W2SPO runs obtain 64.2 and 63.9 overall (mean 64.05, standard deviation 0.21 points), while the Mistral auxiliary variant obtains 63.4. These results show that the observed gain persists across random-seed variation and auxiliary-model replacement.

Table 5: Robustness evaluation under different random-seed and auxiliary-model settings. All results are reported as greedy Pass@1 percentages. The different B-model variant replaces the default auxiliary model with `Mistral-7B-Instruct-v0.3`.

Setting	Overall	MATH500	Olympiad	Omni-HARD	AIME24	AIME25
W2SPO (seed=42)	**64.2**	93.2	**71.8**	**28.4**	**56.7**	46.7
W2SPO (seed=2026)	63.9	**94.2**	70.9	27.8	50.0	**50.0**
Different B Model	63.4	93.8	70.8	27.4	50.0	40.0

C.5 Additional Controls and Generalization Results

Centered-advantage diagnostic. Across 4,032 prompt groups (32,256 branches), the uncentered estimator has a mean absolute group-wise advantage sum of 0.04164. Dividing by eight branches gives an offset of 0.00521 per branch, approximately 1.2% of the mean absolute branch advantage. More importantly, the explicitly centered variant in Table 3 obtains 64.1 overall versus 64.2 for the original estimator, retaining the gain while removing this offset.

Higher-temperature stress test. As a target-only exploration control, we increased the rollout temperature to 1.5. The resulting checkpoint reached 22.01 overall under greedy evaluation (45.80 MATH-500, 15.43 Olympiad, 9.83 Omni-HARD, 3.33 AIME24, and 6.67 AIME25). This result shows that globally increasing sampling randomness does not reproduce the benefit of structured heterogeneous proposals in this controlled stress test.

OLMo-3-7B protocol. We train OLMo-3-7B-Instruct on the MATH training split and evaluate greedily on MATH-500 with a maximum generation length of 8,192 tokens. W2SPO improves accuracy from 48.8% to 57.4%, providing a cross-architecture evaluation with a 7B non-Qwen target.

Multi-turn tool-use protocol. For the Airline subset of τ^2-Bench, we train on the Airline training subset and evaluate the base and W2SPO models on the same 20 test tasks with four independent trials each, yielding 80 trajectories per model. DeepSeek-v4-Flash(Xu et al., 2026) is the user simulator. Agent decoding uses temperature 0.7, top-$p = 0.9$, at most 512 tokens per turn, at most 80 interaction steps, and a 262K context window. The 8-token branch is inserted at the beginning of the second agent turn and optimized using final binary task success. Average reward increases from 0.2750 to 0.3375.

D Additional Analysis of Branch-Level Updates

This appendix gives a more explicit derivation of the local analysis in Section B. The purpose is to justify the interpretation of our update as a branch-level credit-assignment mechanism.

Setup. In this appendix, c, z, and y denote the prefix, branch, and continuation, respectively. They correspond to z, b_i, and $u_{i,k}$ in Section 4.

Fix a problem x and a prefix c generated by the actor. Let $z = (z_1, \ldots, z_{L_b})$ be a short branch segment, and let y be the continuation generated by the actor after the branch.

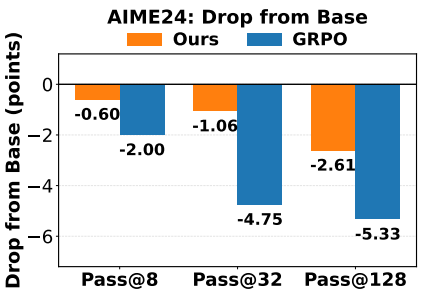

Figure 8: Drop in Pass@k relative to the base model on AIME24

For the local analysis, the continuation policy is treated as fixed. We denote this stop-gradient continuation policy by $\pi_{\bar{\theta}}$, and define

$$Q_{\bar{\theta}}(c, z) = \mathbb{E}_{y \sim \pi_{\bar{\theta}}(\cdot|x,c,z)} \left[R(x, c, z, y) \right]. \tag{5}$$

The local branch objective is

$$J_{\mathrm{br}}(\theta; c, \bar{\theta}) = \mathbb{E}_{z \sim \pi_\theta(\cdot|x,c)} \left[Q_{\bar{\theta}}(c, z) \right]. \tag{6}$$

This objective isolates the effect of changing the branch distribution while holding the prefix and continuation-value estimator fixed.

Claim 1: Branch-token masking gives a local policy-gradient estimator under a fixed-prefix approximation. Assume that the support of $\pi_\theta(z \mid x, c)$ does not change with θ and that $Q_{\bar{\theta}}(c, z)$ is treated as independent of θ during the branch update. Then

$$\nabla_\theta J_{\mathrm{br}}(\theta; c, \bar{\theta}) = \mathbb{E}_{z \sim \pi_\theta(\cdot|x,c)} \left[Q_{\bar{\theta}}(c, z) \nabla_\theta \log \pi_\theta(z \mid x, c) \right]. \tag{7}$$

Proof. By definition,

$$J_{\mathrm{br}}(\theta; c, \bar{\theta}) = \sum_z \pi_\theta(z \mid x, c) Q_{\bar{\theta}}(c, z). \tag{8}$$

Taking the gradient and using $\nabla_\theta \pi_\theta(z) = \pi_\theta(z) \nabla_\theta \log \pi_\theta(z)$ gives

$$\nabla_\theta J_{\mathrm{br}} = \sum_z \nabla_\theta \pi_\theta(z \mid x, c) Q_{\bar{\theta}}(c, z) \tag{9}$$

$$= \sum_z \pi_\theta(z \mid x, c) Q_{\bar{\theta}}(c, z) \nabla_\theta \log \pi_\theta(z \mid x, c) \tag{10}$$

$$= \mathbb{E}_{z \sim \pi_\theta} \left[Q_{\bar{\theta}}(c, z) \nabla_\theta \log \pi_\theta(z \mid x, c) \right]. \tag{11}$$

Since z is a token sequence,

$$\log \pi_\theta(z \mid x, c) = \sum_{t=1}^{L_b} \log \pi_\theta(z_t \mid x, c, z_{<t}). \tag{12}$$

Therefore, applying the policy-gradient loss only to the branch tokens corresponds to optimizing this local branch objective. This corresponds to a semi-gradient view in which future continuation probabilities are held fixed during the branch update, matching the implementation choice of masking gradients to the branch window.

Baseline subtraction. For any baseline $b(c)$ that does not depend on the sampled branch z,

$$\mathbb{E}_{z \sim \pi_\theta} \left[b(c) \nabla_\theta \log \pi_\theta(z \mid x, c) \right] = b(c) \nabla_\theta \sum_z \pi_\theta(z \mid x, c) = 0. \tag{13}$$

Thus, a centered branch value $Q_{\bar{\theta}}(c, z) - b(c)$ preserves the expected local gradient while reducing variance. Our implementation uses a finite-sample, weighted leave-one-out baseline and normalization for variance reduction and proposal balancing.

Claim 2: Auxiliary branches can reduce all-wrong groups when they improve local success probability. Assume binary rewards $R \in \{0, 1\}$. For a branch z_i, define its conditional success probability as

$$p_i = \Pr_{y \sim \pi_{\bar{\theta}}(\cdot|x,c,z_i)} \left[R(x, c, z_i, y) = 1 \right]. \tag{14}$$

If we draw K independent continuations from the actor after this branch, then the probability that branch z_i obtains at least one successful continuation is

$$s_i = 1 - (1 - p_i)^K. \tag{15}$$

This quantity is monotone in both p_i and K.

If success events are further approximated as independent across branches, then for a group of G branches,

$$P_{\text{all-wrong}} = \prod_{i=1}^{G}(1 - s_i). \tag{16}$$

Therefore, under this independence approximation, replacing actor-only branches with auxiliary branches that have larger conditional success probabilities reduces the all-wrong probability multiplicatively. The result requires only that some auxiliary branches induce larger downstream success probabilities under the actor continuation policy; it does not require the auxiliary model to be a stronger standalone solver.

More generally, without the independence approximation, the following bound still holds. Since the event that all branches fail is a subset of the event that any particular branch i fails,

$$P_{\text{all-wrong}} \leq 1 - s_i \quad \text{for every } i. \tag{17}$$

Hence,

$$P_{\text{all-wrong}} \leq 1 - \max_i s_i. \tag{18}$$

This bound shows that a single branch with high downstream success probability can lower an upper bound on the all-wrong probability, although the multiplicative formula requires the stronger independence approximation.

Off-policy branch proposals. In our method, branches are sampled from both the actor and an auxiliary proposal distribution. If all samples came from a known mixture proposal

$$m(z \mid x, c) = \lambda \pi_{\theta_{\text{old}}}(z \mid x, c) + (1 - \lambda) q_B(z \mid x, c), \tag{19}$$

then an exact importance-sampling correction for estimating expectations under a target branch policy would involve a ratio of the form

$$w^*(z) = \frac{\pi_{\theta_{\text{old}}}(z \mid x, c)}{m(z \mid x, c)} \quad \text{or} \quad \frac{\pi_\theta(z \mid x, c)}{m(z \mid x, c)}, \tag{20}$$

depending on the target distribution. Such ratios can have high variance and are difficult to compute exactly when auxiliary text is decoded and re-tokenized into the actor vocabulary.

Instead, we use a source-specific, tempered, and clipped coefficient

$$c_i = \min\left\{c_{\max}, \exp\left(\alpha \cdot \text{clip}\left(\bar{p}_i - \bar{q}_i, -M, M\right)\right)\right\}, \tag{21}$$

Here $\bar{p}_i$ is the old actor's per-token mean log-probability, as defined in Section 4:

$$\bar{p}_i = \frac{1}{L_b} \log \pi_{\theta_{\text{old}}}(z_i \mid x, c). \tag{22}$$

The source score $\bar{q}_i$ is the per-token mean log-probability computed from the proposal model's token-level scores. For actor-sampled branches, $\bar{q}_i = \bar{p}_i$, so $c_i = 1$. For auxiliary-sampled branches, the averaging uses the auxiliary tokenization and its corresponding token count, which need not equal L_b. These normalized scores are used for heuristic proposal balancing; their difference is not an exact cross-tokenizer log-likelihood ratio.

We therefore use c_i as a source-specific, tempered proposal-balancing coefficient rather than as an importance-sampling correction. The coefficient satisfies

$$0 < c_i \leq c_{\max}, \tag{23}$$

so a branch cannot receive an arbitrarily large scalar weight from the balancing term. With the default settings $\alpha = 0.02$, $M = 30$, and $c_{\max} = 2.0$,

$$e^{-0.6} \leq c_i \leq e^{0.6} < 2.0.$$

Thus, the outer cap $c_{\max}$ is inactive under these settings; the bound follows from clipping the score difference before exponentiation. This bound applies to the proposal-balancing coefficient itself and does not imply a bound on the full policy gradient.

Effective sample size. To monitor whether the balancing weights concentrate on a small number of branches, we use the effective sample size

$$\mathrm{ESS} = \frac{\left(\sum_i c_i\right)^2}{\sum_i c_i^2}. \tag{24}$$

For a group of G branches, one may also report the normalized effective sample size

$$\mathrm{nESS} = \frac{1}{G}\frac{\left(\sum_i c_i\right)^2}{\sum_i c_i^2}. \tag{25}$$

A very small nESS indicates that the update is dominated by a few weighted branches, whereas a moderate nESS suggests that auxiliary proposals contribute without collapsing the effective training group.

Empirical diagnostics. The analysis motivates the following empirical diagnostics. If auxiliary branches provide useful local exploration, we expect to observe:

1. higher within-group reward contrast compared with actor-only branch sampling;
2. lower all-wrong ratio and higher mixed-correct ratio;
3. moderate ESS or nESS, together with a controlled clipping fraction.

These diagnostics test whether auxiliary branches improve local exploration while remaining sufficiently aligned with the actor distribution.