DGPO: DIRECTIONAL GAIN POLICY OPTIMIZATION FOR CREDIT ASSIGNMENT IN REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

Outcome-level rewards supervise entire reasoning trajectories, even when their success depends on a few intermediate segments. We present Directional Gain Policy Optimization (DGPO), which converts binary outcome rewards into information-weighted segment-level credit without process-level supervision. DGPO constructs uncertainty-triggered reasoning trees to expose alternative continuations under shared prefixes. Its core credit signal, Directional Information Gain (DIG), combines normalized information gain with a signed measure of success association, capturing both how strongly a segment predicts the outcome and whether it is associated with success or failure. Global DIG captures tree-wide associations, while local DIG contrasts sibling branches; their normalized scores augment trajectory-level advantages for policy optimization. Across four language-model backbones and six mathematical reasoning benchmarks, DGPO achieves higher average pass@1 accuracy than GRPO, TreePO, and TreeRL on every backbone. With the same tree-construction procedure, DIG also outperforms direct segment advantages and TreeRL advantages in a four-benchmark comparison on Qwen3-4B-Base. A separate comparison on this backbone shows higher final accuracy at comparable total training time.

1 INTRODUCTION

Reinforcement learning with verifiable rewards (RLVR) has become an effective approach to improving mathematical reasoning in large language models (LLMs) (Shao et al., 2024; Guo et al., 2025a; Yu et al., 2025). By evaluating generated answers with an automated verifier, RLVR allows a model to learn from sampled solutions without requiring annotations of intermediate reasoning steps. However, this supervision leaves a credit assignment problem: the verifier evaluates a complete response, whereas policy optimization updates the individual decisions that produced it. In trajectory-level approaches such as GRPO, a single outcome-derived advantage is applied across the response. Within a failed response, a useful derivation and a late erroneous continuation receive the same trajectory-level advantage, while an unnecessary detour can share the positive advantage of an ultimately correct answer. The training signal provides little explicit guidance for distinguishing these intermediate segments.

Tree-structured rollouts offer a way to obtain more localized evidence from final outcomes. Sampling alternative continuations from a shared prefix allows their downstream success rates to be compared while holding the preceding reasoning fixed. Recent methods have used this structure for both exploration and intermediate supervision. TreeRL (Hou et al., 2025) derives process-level signals from on-policy search, SPO (Guo et al., 2025b) estimates segment-level advantages, and TreePO (Li et al., 2025) incorporates outcome comparisons at multiple levels of the tree. These advances make the construction of the credit signal a central design choice. A direct return difference indicates whether a continuation performs better than a reference group; an information-based weighting can additionally modulate this signal according to how strongly segment occurrence discriminates successful from unsuccessful trajectories.

We propose **Directional Gain Policy Optimization (DGPO)**, a framework that converts verified trajectory outcomes into information-weighted segment credit. DGPO constructs uncertainty-triggered reasoning trees to explore alternative continuations from shared prefixes, with semantic-similarity pruning to limit redundant branches. Its core signal, **Directional Information Gain (DIG)**, combines

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.

normalized information gain with a bounded measure of success association. This design weights segment credit according to both how informative a segment is about the final outcome and whether it is positively or negatively associated with success.

DGPO computes DIG globally across the reasoning tree and locally among sibling continuations, combining tree-wide outcome context with comparisons under the same prefix. The normalized scores augment the underlying trajectory advantage to provide segment-specific corrections for policy optimization, without additional process-level labels or a separately trained process reward model. Our theoretical analysis characterizes unsmoothed DIG as a nonnegative reweighting of population advantages and establishes sufficient conditions for policy improvement under fixed-tree branch-policy updates.

We evaluate DGPO on four LLM backbones and six mathematical reasoning benchmarks. DGPO achieves the highest average pass@1 and pass@8 accuracies among the evaluated methods on every backbone, outperforming GRPO, TreePO, and TreeRL. Component ablations support the contributions of both uncertainty-triggered branching and DIG. Using the same DGPO tree-construction procedure, a four-benchmark comparison on Qwen3-4B-Base further shows that DIG outperforms direct segment advantages and TreeRL advantages, providing evidence for the proposed credit design beyond the choice of rollout structure. Additional experiments demonstrate compatibility with DAPO and higher final accuracy than the compared baselines at a comparable total training time on Qwen3-4B-Base.

2 Related Work

Reinforcement Learning with Verifiable Rewards. Verifiable outcome rewards have enabled substantial progress in mathematical reasoning with large language models (Shao et al., 2024; Guo et al., 2025a). GRPO estimates relative advantages from groups of sampled responses without a learned critic, while DAPO and Dr. GRPO refine the sampling and optimization procedures used in this setting (Yu et al., 2025; Liu et al., 2025). Value-based approaches such as VAPO retain a learned critic to support advantage estimation over long reasoning trajectories (Yue et al., 2025). These methods illustrate different ways to optimize policies from outcome supervision. DGPO builds on this training setting and uses the structure of sampled trajectories to construct additional segment-level credit.

Tree-Structured Rollouts. Tree-structured sampling explores alternative continuations from intermediate prefixes and enables reuse of shared reasoning context. TreeRL uses entropy-guided branching for on-policy exploration (Hou et al., 2025), while FR3E launches targeted continuations from high-uncertainty positions in sampled trajectories (Zheng et al., 2025). TreePO combines segment-wise decoding with branching and fallback heuristics to improve rollout efficiency (Li et al., 2025), and the tree-based variant of SPO reuses continuations for segment-value estimation (Guo et al., 2025b). These approaches connect rollout construction with the intermediate feedback available during training. DGPO adopts uncertainty-triggered branching and semantic-similarity pruning to collect the alternative continuations used by its credit estimator.

Intermediate Credit Assignment. Intermediate feedback can be obtained through learned process reward models or estimated directly from sampled outcomes. Process supervision has been studied using human step annotations and automatically constructed labels (Lightman et al., 2024; Wang et al., 2024). Among approaches that estimate credit from rollouts, VinePPO and SPO use Monte Carlo continuations to estimate intermediate values and advantages (Kazemnejad et al., 2025; Guo et al., 2025b). TreeRPO and TreePO exploit step-level groups and hierarchical outcome comparisons (Yang et al., 2025; Li et al., 2025). More recent methods further exploit trajectory structure: TEMPO augments GRPO with temporal-difference corrections derived from prefix-tree values (Tran et al., 2026), whereas TreeAdv redistributes complete-rollout advantages to shared tree segments (Cao et al., 2026). SSVPO takes a different approach by estimating step contributions through sequential Shapley values (Li et al., 2026). DGPO belongs to this broader line of outcome-derived credit assignment and is distinguished by the information-weighted statistic it uses to score segments.

Information-Based Learning Signals. Information-based signals have also been explored in multi-turn agent training. IGPO derives turn-level rewards from changes in the model's log-likelihood of the correct answer (Wang et al., 2026), while IGRPO uses answer-likelihood-based information gains

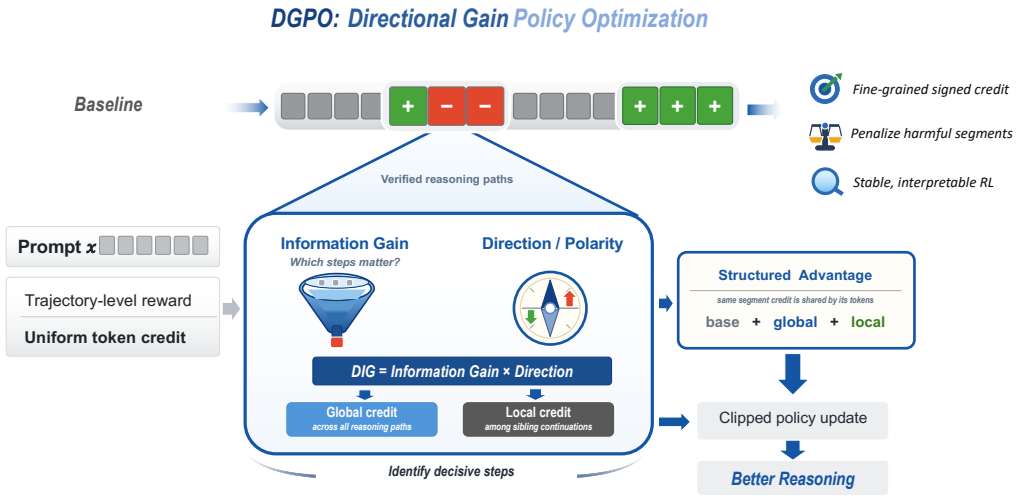

Figure 1: **Overview of DGPO.** Verified complete paths provide the outcome statistics for segment credit. Global DIG compares paths across the full tree, while local DIG compares sibling continuations under the same prefix. The normalized scores supply segment-level corrections to the base trajectory advantage.

to allocate tree-expansion budgets (Zhang et al., 2026). DGPO measures the empirical dependence between segment occurrence and verified binary outcomes within a sampled tree. Its DIG score combines normalized mutual information with a bounded transform of the log-odds ratio and applies the resulting statistic to both tree-wide and sibling-conditioned comparisons. This formulation uses outcome discrimination and association direction to weight segment-level advantages.

3 Method

DGPO uses the verified outcomes of a reasoning tree to construct segment-level corrections to a trajectory-level advantage. The central idea is to compare the outcomes of paths that pass through a segment with those of alternative paths, and use the strength and direction of this association to assign credit. We first describe the overall training procedure, then explain how the tree is collected, how segment credit is computed, and how it enters policy optimization.

3.1 Setup and Overview

For a prompt x, let $\pi_{\theta_{\text{old}}}$ denote the policy used to collect rollouts for the current update. DGPO organizes these rollouts into a tree $\mathcal{T} = (\mathcal{V}, \mathcal{E})$: a node represents a reasoning prefix, an edge $z = (u \to v)$ represents a contiguous generated segment, and a retained complete root-to-leaf path τ_i represents a response. We denote its token sequence by o_i and its verifier label by $Y_i \in \{0, 1\}$, where 1 indicates a correct final answer. The set of retained complete paths is denoted by $\mathcal{P}$. When several paths share a segment, their final labels provide multiple outcome observations for that segment, allowing intermediate credit to be estimated from final-answer verification.

Each training update has three stages. First, DGPO samples alternative continuations at uncertainty-triggered positions and verifies the resulting complete responses. Second, it computes *Directional Information Gain* (DIG) for each segment using two comparisons: one across the full tree and one restricted to alternatives sharing the same parent prefix. Third, it normalizes these scores and adds them to the base trajectory advantage for policy optimization. Figure 1 summarizes this data flow.

3.2 Collecting a Reasoning Tree

Tree collection determines which alternative continuations are available for comparison. DGPO uses sustained predictive uncertainty to select branching positions and prunes similar candidates to limit

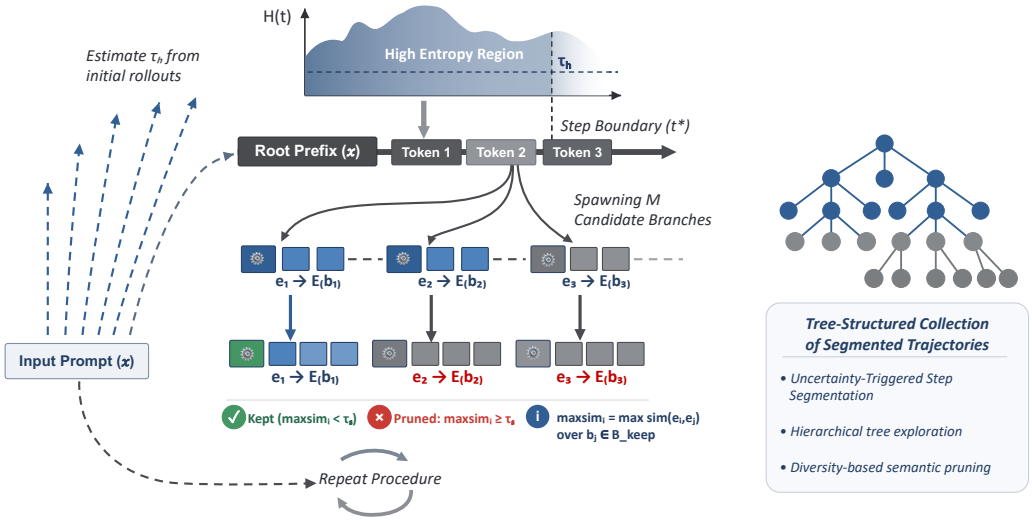

Figure 2: **Uncertainty-triggered tree collection.** Preliminary rollouts calibrate an entropy threshold. Sustained high entropy identifies branching boundaries, and embedding similarity is used to prune redundant candidate continuations. Retained complete paths provide the comparisons used for segment credit.

redundant exploration. Uncertainty serves as a sampling heuristic; final correctness is determined by the verifier.

Locate branching positions. We first sample K preliminary responses $\{s^{(k)}\}_{k=1}^{K}$ to calibrate a prompt-specific uncertainty threshold. We compute token entropies and take the γ-quantile over all token positions in all preliminary responses:

$$H_t^{(k)} = -\sum_{v \in \mathcal{V}_{\text{tok}}} \pi_{\theta_{\text{old}}}(v \mid x, s_{<t}^{(k)}) \log \pi_{\theta_{\text{old}}}(v \mid x, s_{<t}^{(k)}), \tag{1}$$

$$\tau_H(x) = \text{Quantile}_\gamma\left(\{H_t^{(k)}\}_{k,t}\right), \tag{2}$$

where $\mathcal{V}_{\text{tok}}$ is the token vocabulary. During tree generation, a branching boundary is identified at the start $t^\star$ of a window containing n consecutive high-entropy positions:

$$H_{t'} > \tau_H(x), \qquad t' \in \{t^\star, \ldots, t^\star + n - 1\}. \tag{3}$$

The boundary can be detected only after the full window has been observed, so branching at $t^\star$ requires retaining the prefix preceding that position.

Sample and prune continuations. At a detected boundary, we sample M candidate continuations of up to ℓ tokens. Each candidate b_i is represented by an embedding $e_i = E(b_i)$ extracted from a selected layer of the rollout model, whose parameters remain fixed during collection. At each branching prefix u, let $\mathcal{B}_{\text{keep}}(u)$ denote the candidates already retained at that branching point. A new candidate is discarded when

$$\max_{b_j \in \mathcal{B}_{\text{keep}}(u)} \text{sim}(e_i, e_j) \geq \tau_S. \tag{4}$$

The retained set is initialized separately at each branching point. The first candidate is retained, and the redundancy test is applied to subsequent candidates when the set is nonempty. Retained continuations are expanded until an end-of-sequence token, a new uncertainty boundary, or a prescribed depth or budget limit is reached. At subsequent boundaries, the branching procedure is repeated when further expansion is permitted. Tree collection is constrained by a maximum depth D and a budget B of completed paths. The resulting complete paths are verified to obtain $\{Y_i\}$. Only retained complete paths contribute to the credit statistics below; preliminary calibration rollouts and pruned candidates incur generation cost but are outside the completed-path budget. The reported settings are given in Appendix A.

3.3 Computing Directional Segment Credit

For a segment z, DIG combines two properties: how informative its occurrence is about the outcome, and whether that occurrence is associated with success or failure. Computing these properties first requires specifying which paths are being compared.

Choose the comparison set. The *global* comparison uses all paths in $\mathcal{P}$. The *local* comparison uses only paths that reach the parent node $p(z)$, denoted by $\mathcal{P}_{p(z)} = \{\tau_i \in \mathcal{P} : \tau_i \text{ reaches } p(z)\}$. Within either set, we compare paths containing z with paths not containing it. In the local comparison, the latter group consists of continuations through sibling branches. We use Ω to denote either comparison set and define the same DIG estimator for both.

Summarize the verified outcomes. Let $Z_i(z) = \mathbb{I}\{z \in \tau_i\}$ indicate whether path $\tau_i \in \Omega$ contains segment z. The paths fall into four groups:

	Success ($Y = 1$)	Failure ($Y = 0$)
Contains z ($Z = 1$)	N_{11}	N_{10}
Does not contain z ($Z = 0$)	N_{01}	N_{00}

(5)

Here $N_{ab} = N_{ab}(z; \Omega)$ counts paths with $Z_i(z) = a$ and $Y_i = b$. Each retained leaf contributes one equally weighted observation, giving the empirical joint distribution $\widehat{p}_{ab} = N_{ab}/|\Omega|$. Occurrence refers to a specific tree edge: identical text generated under different prefixes is not merged. All information quantities below are computed from this empirical distribution.

Measure informativeness. Knowing whether a path contains z is informative when the outcome distributions of the two groups differ. We quantify this with normalized information gain:

$$G(z; \Omega) = \frac{I(Y; Z(z) \mid \Omega)}{H(Z(z) \mid \Omega) + \epsilon}, \qquad I(Y; Z) = H(Y) - H(Y \mid Z). \tag{6}$$

The gain-ratio normalization (Quinlan, 1986) expresses outcome information relative to the entropy of segment occurrence. It is zero when occurrence and outcome are empirically independent, but it is unsigned: association with either success or failure can make a segment informative.

Determine association direction. We distinguish these two cases using the smoothed log-odds ratio

$$\Delta(z; \Omega) = \log \frac{(N_{11} + \lambda)(N_{00} + \lambda)}{(N_{10} + \lambda)(N_{01} + \lambda)}, \tag{7}$$

where $\lambda > 0$ prevents undefined odds when a count is zero. A positive Δ indicates higher smoothed success odds for paths containing z; a negative value indicates lower odds. Smoothing is applied only to this polarity term, not to the information quantities in Eq. (6).

Combine strength and direction. The DIG score is

$$\mathrm{DIG}(z; \Omega) = G(z; \Omega) \underbrace{\tanh\left(\frac{\Delta(z; \Omega)}{2}\right)}_{\text{bounded association polarity}}. \tag{8}$$

The polarity factor is the smoothed Yule coefficient (Yule, 1900) and lies in $(-1, 1)$, so information gain controls the score's magnitude. Both comparison groups must also contain enough observations:

$$\min\{N_{11} + N_{10},\ N_{01} + N_{00}\} < \tau_{\min} \quad \Longrightarrow \quad G(z; \Omega) = \mathrm{DIG}(z; \Omega) = 0. \tag{9}$$

Unsupported scores are set directly to zero; the constraint applies to the two group sizes, not to every cell of the table.

For an illustrative example, suppose that three of four paths containing z succeed, whereas only one of four paths not containing it succeeds. With sufficient support, z receives positive raw DIG; exchanging the success and failure counts gives negative raw DIG. If both groups instead contain two successes and two failures, occurrence provides no outcome information and DIG is zero. These scores describe association within the sampled comparison set, rather than a causal effect of the segment.

Applying this estimator to the two comparison sets gives

$$D_{\text{global}}(z) = \mathrm{DIG}(z; \mathcal{P}), \tag{10}$$

$$D_{\text{local}}(z) = \mathrm{DIG}(z; \mathcal{P}_{p(z)}). \tag{11}$$

The first score captures tree-wide outcome association; the second distinguishes alternatives available under the same prefix. Support is checked separately for each score because its comparison set differs.

3.4 Policy Optimization with Segment Credit

DGPO uses the two DIG scores as corrections to the base trajectory advantage. We first put the scores on comparable scales, then assign each token the corrections of its segment.

Normalize supported segment scores. For each credit type $r \in \{\text{global}, \text{local}\}$, let $\mathcal{E}_r^{\text{valid}}$ be the distinct edges in the current tree that pass Eq. (9) for that comparison. We compute the mean μ_r and standard deviation σ_r over this set and define

$$\widetilde{D}_r(z) = \begin{cases} \dfrac{D_r(z) - \mu_r}{\sigma_r + \epsilon}, & z \in \mathcal{E}_r^{\text{valid}}, \\ 0, & \text{otherwise.} \end{cases} \tag{12}$$

An empty valid set contributes zero correction. Both credit types are normalized across their supported edges in the full tree; local scores are not normalized separately within each sibling group. Unsupported edges are excluded from the normalization statistics and retain zero correction.

Construct token advantages. Let $\widehat{A}_i^{\text{base}}$ be the trajectory advantage supplied by the underlying optimizer, such as the group-normalized outcome reward in GRPO. For token $o_{i,t}$, let $z(i,t)$ denote its segment. The advantage used by DGPO is

$$\widehat{A}_{i,t}^{\text{DGPO}} = \widehat{A}_i^{\text{base}} + \alpha\, \widetilde{D}_{\text{global}}(z(i,t)) + \beta\, \widetilde{D}_{\text{local}}(z(i,t)), \tag{13}$$

where α and β control the two corrections. Tokens in the same segment share these corrections, while their base advantage is inherited from the complete trajectory. Consequently, the sign of raw DIG need not equal the sign of the final advantage after centering and combination.

Update the policy. For a collected tree with $N_x = |\mathcal{P}|$ complete paths, we use a clipped objective with token-level normalization:

$$\mathcal{J}_{\text{DGPO}}(\theta) = \mathbb{E}\left[\frac{1}{\sum_{i=1}^{N_x} |o_i|} \sum_{i=1}^{N_x} \sum_{t=1}^{|o_i|} \min\Big(r_{i,t}(\theta) \widehat{A}_{i,t}^{\text{DGPO}}, \; \text{clip}(r_{i,t}(\theta), 1-\varepsilon, 1+\varepsilon)\, \widehat{A}_{i,t}^{\text{DGPO}} \Big) \right], \tag{14}$$

where the expectation is over training prompts and collected rollout trees, and

$$r_{i,t}(\theta) = \frac{\pi_\theta(o_{i,t} \mid x, o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t} \mid x, o_{i,<t})}. \tag{15}$$

Here ε is the clipping radius, distinct from the numerical stability constant ϵ. The sum is over complete paths, so a shared segment can contribute through multiple paths containing it. Regularization inherited from the base optimizer, including the KL penalty used in our implementation, is omitted for clarity. Algorithm 1 in Appendix A.2 summarizes one training update.

4 Experiments

We evaluate whether DGPO improves reasoning accuracy, whether its gains depend on the proposed segment-credit design, and whether they persist at comparable training time. We first compare the complete system with chain-based and tree-based baselines across four backbones. We then examine the contributions of tree construction and DIG, including comparisons that hold the branching procedure fixed. A comparison at similar total runtimes evaluates the computational trade-off, and training diagnostics describe how accuracy and policy entropy evolve.

4.1 Experimental Setup

Training data and evaluation. We train on DAPO-Math (Yu et al., 2025) and evaluate on six mathematical reasoning benchmarks: AIME 2025 (Art of Problem Solving Wiki, 2025), AIME

Table 1: **Main results on mathematical reasoning benchmarks.** We report mean pass@1 accuracy (%) across eight stochastic evaluation runs, using stochastic decoding with temperature $T = 0.6$ and top-$p = 0.95$. Within each backbone, the best result in each column is in **bold**, and the strongest non-DGPO result is underlined.

Model	Method	AIME25	AIME24	AMC23	MATH-500	Minerva	OlympiadBench	Avg.
`Qwen2.5-3B`	Base	0.00	2.08	5.94	23.85	3.35	9.02	7.37
	GRPO	5.00	8.33	45.31	64.22	10.89	35.07	28.14
	TreePO	3.33	7.92	46.56	64.85	11.44	35.35	28.24
	TreeRL	4.17	8.33	48.44	65.40	12.32	**36.72**	29.23
	DGPO	**5.42**	**10.83**	**50.31**	**67.75**	**13.37**	36.57	**30.71**
`Qwen2.5-7B`	Base	1.67	2.08	17.19	44.57	6.66	18.56	15.12
	GRPO	12.50	20.00	64.37	78.22	17.14	48.33	40.09
	TreePO	11.67	19.17	65.62	79.98	16.64	49.33	40.40
	TreeRL	12.50	21.25	66.88	78.40	**17.74**	49.83	41.10
	DGPO	**13.33**	**22.92**	**69.06**	**80.87**	17.46	**50.46**	**42.35**
`Qwen3-4B-Base`	Base	0.83	4.17	26.87	58.35	12.32	26.76	21.55
	GRPO	23.33	27.50	68.13	84.97	17.56	59.07	46.76
	TreePO	23.33	26.67	68.44	84.80	18.34	59.22	46.80
	TreeRL	23.33	27.92	69.38	85.45	18.75	**61.37**	47.70
	DGPO	**25.83**	**31.67**	**71.56**	**87.67**	**19.58**	61.31	**49.60**
`Llama-3.2-3B-Instruct`	Base	0.00	1.67	11.25	35.42	5.79	12.52	11.11
	GRPO	2.08	12.08	36.87	53.75	7.72	24.76	22.88
	TreePO	1.67	11.25	38.44	53.63	8.78	24.83	23.10
	TreeRL	2.92	12.50	37.50	54.85	9.65	25.11	23.76
	DGPO	**4.17**	**13.75**	**38.75**	**55.18**	**9.93**	**26.15**	**24.66**

2024 (Art of Problem Solving Wiki, 2024), AMC23 (Art of Problem Solving Wiki, 2023), MATH-500 (Hendrycks et al., 2021), Minerva (Lewkowycz et al., 2022), and OlympiadBench (He et al., 2024). The main metric is pass@1 accuracy, averaged over eight stochastic evaluation runs with temperature 0.6 and top-$p = 0.95$. The overall score is the unweighted average across the six benchmarks. Appendix B.4 additionally reports pass@8 to assess performance with multiple sampled answers.

Backbones and comparison methods. We evaluate `Qwen2.5-3B`, `Qwen2.5-7B` (Team, 2024), `Qwen3-4B-Base` (Team, 2025), and `Llama-3.2-3B-Instruct` (Grattafiori et al., 2024). The base model establishes the performance baseline before RL, GRPO (Shao et al., 2024) provides a chain-based training baseline, and TreePO (Li et al., 2025) and TreeRL (Hou et al., 2025) provide comparisons with existing tree-based training methods. The component and credit-estimator comparisons in Section 4.3 further examine which parts of DGPO contribute to its performance.

Implementation and budgets. Experiments use VeRL, FSDP, and vLLM on eight NVIDIA H800 GPUs. DGPO uses a branch factor of eight, a maximum depth of three, and at most eight completed trajectories and verifier calls per prompt. This completed-path budget excludes preliminary entropy-calibration rollouts and pruned candidates, which still incur generation cost. We therefore assess the computational trade-off using wall-clock training time in Section B.3. All backbones use the same fixed DGPO configuration; full training and sampling settings are given in Appendix A.

4.2 Reasoning Accuracy Across Backbones

Gains over both chain-based and tree-based training. Table 1 shows that DGPO achieves the highest average pass@1 on all four backbones and the best accuracy in 21 of the 24 model–benchmark combinations. Its average gains over GRPO are 2.57, 2.26, 2.84, and 1.78 points for `Qwen2.5-3B`, `Qwen2.5-7B`, `Qwen3-4B-Base`, and `Llama-3.2-3B-Instruct`, respectively. TreeRL is the strongest competing method by average accuracy on each backbone; DGPO improves over it by 1.48, 1.25, 1.90, and 0.90 points. The advantage therefore extends across both model families and remains present when compared with methods that already use tree-structured training.

On `Qwen3-4B-Base`, for example, DGPO improves AIME24 from 27.92 to 31.67 and MATH-500 from 85.45 to 87.67 relative to TreeRL. The strongest competing method retains a small lead on three individual model–benchmark combinations, so the result is a consistent advantage in overall accuracy rather than uniform dominance on every benchmark. The pass@8 results in Appendix B.4 also place

Table 2: **Ablation of DGPO components.** We report the average pass@1 accuracy across the six benchmarks in Table 1. Replacing uncertainty-triggered branching with random branching, or removing the DIG correction, reduces accuracy in the reported configurations.

Method	Qwen2.5-3B	Qwen2.5-7B	Qwen3-4B-Base	Llama-3.2-3B-Instruct
GRPO	28.14	40.09	46.76	22.88
DGPO w/ random branching	27.09	38.64	45.94	21.29
DGPO w/o DIG advantage	29.44	41.21	48.29	23.15
DGPO	**30.71**	**42.35**	**49.60**	**24.66**

DGPO first in average accuracy on each backbone, extending the observed benefit to evaluation with multiple candidate answers.

Evaluation variability. Appendix Table 4 reports dispersion and paired comparisons over eight stochastic evaluation runs. The reported two-sided exact Wilcoxon tests find significant differences between DGPO and TreeRL on the three Qwen backbones, but not on `Llama-3.2-3B-Instruct` ($p = 0.1484$). These tests characterize stochastic evaluation variability of the trained checkpoints. The main comparison establishes the benefit of the complete DGPO system; the following experiments examine the contributions of its rollout and credit designs.

4.3 Contributions of Tree Construction and Segment Credit

The main results combine changes to rollout collection and credit assignment. We examine them at two levels: component ablations alter the branching rule or remove DIG, while additional comparisons retain DGPO's branching procedure and change the advantage design.

DIG adds value to the same tree-construction procedure. In Table 2, removing DIG while retaining uncertainty-triggered tree collection still improves over GRPO by 0.27–1.53 points across the four backbones. Adding DIG yields a further 1.27, 1.14, 1.31, and 1.51 points, respectively. Thus, the tree-construction procedure accounts for part of the improvement, and the segment-credit correction contributes an additional gain on every evaluated backbone. This comparison supports the added value of segment-specific DIG corrections on top of trajectory-level outcome supervision.

The choice of credit estimator matters. Table 5 in Appendix B.2 holds DGPO's branching procedure fixed and compares alternative advantage designs on `Qwen3-4B-Base`. The comparison averages AIME25, AMC23, MATH-500, and OlympiadBench. DGPO with DIG achieves 61.59, compared with 59.42 for direct segment advantages and 60.26 for TreeRL advantages, improving by 2.17 and 1.33 points, respectively. These results support the proposed credit design relative to two alternatives under the same branching procedure. The comparison evaluates the complete DIG correction, including its weighting and combination of global and local scores; it does not separately isolate the information factor.

Branch placement affects the benefit of tree collection. Replacing uncertainty-triggered branching with random branching reduces average accuracy by 3.37–3.71 points relative to full DGPO, and the resulting variant falls below GRPO on all four backbones. This establishes that the branching rule materially affects performance in the tested configuration. Together with the DIG-removal ablation, the result supports both design choices: where alternative continuations are collected and how their outcomes are converted into credit. The qualitative example in Appendix D illustrates the resulting segment corrections within an individual reasoning tree.

4.4 Training Dynamics

We finally examine when the accuracy advantage emerges and how policy entropy evolves. Figure 3 compares DGPO and GRPO as a function of optimization steps on three backbones. The Qwen models generally show an advantage at the earlier recorded checkpoints, while the Llama model begins below GRPO in overall accuracy before overtaking it. All three finish with higher overall accuracy under DGPO. These trajectories show that the final advantage develops during training, although the early behavior differs across models. The time-based comparison in Appendix B.3 accounts for the per-step cost differences that are not visible in these plots. Figure 4 reports policy entropy on `Qwen3-4B-Base`. Both methods become more concentrated during training, but DGPO maintains higher entropy over

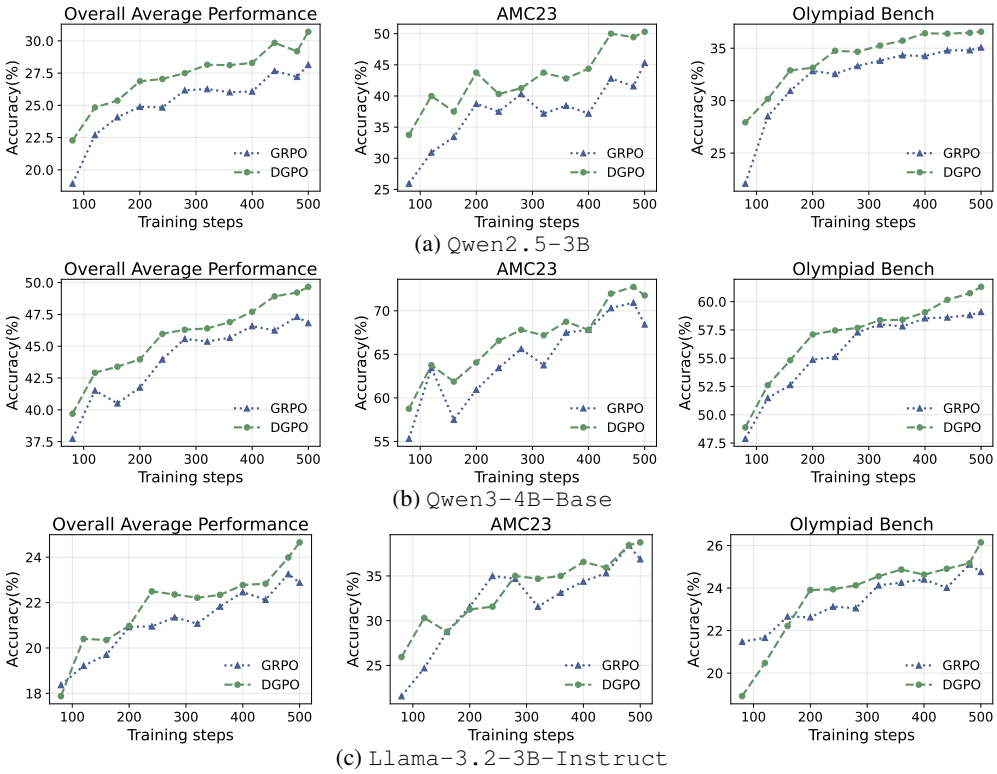

Figure 3: **Training dynamics of GRPO and DGPO.** Pass@1 accuracy is plotted against optimization steps for the reported runs. The panels show the overall average and two benchmark-specific trajectories; the early-training pattern varies across backbones.

the reported interval and ends at a higher level. Together with the accuracy results, this shows that improved performance is compatible with a less concentrated token distribution in this run. The curves are descriptive diagnostics; they do not by themselves establish reduced variability across independent training runs.

Evaluation scope. Our experiments cover verifier-based mathematical reasoning with binary outcome labels and models of up to 7B parameters. The results support the complete DGPO system and its segment-credit design within this setting; transfer to open-ended tasks, noisy verification, and larger models remains to be evaluated. The comparison at similar total runtimes is limited to `Qwen3-4B-Base`, and the reported stochastic evaluation repeats do not quantify variability across independent RL training runs.

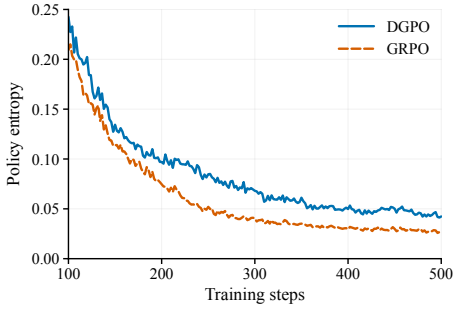

Figure 4: **Policy entropy on `Qwen3-4B-Base`.** Both methods show declining entropy; DGPO maintains a higher level in the reported diagnostic.

5 Conclusion

We presented **DGPO**, a policy optimization framework that derives information-weighted segment credit from verified trajectory outcomes. DGPO combines uncertainty-triggered tree rollouts with global and local Directional Information Gain to augment trajectory-level advantages according to the strength and direction of segment–outcome association. Across four backbones and six mathematical reasoning benchmarks, DGPO improves average accuracy over GRPO and tree-based baselines. Component ablations and comparisons under the same branching procedure support the contribution of its credit design, while the Qwen3-4B-Base evaluation shows higher final accuracy at comparable training time. These findings support the value of combining structured exploration with information-weighted credit assignment to learn more effectively from sparse outcome supervision.

AI USE STATEMENT

Generative AI tools were used solely to assist with literature discovery and retrieval of related work. AI assistance was not used for running experiments, generating benchmark results, or writing the manuscript. The authors assume full responsibility for all final text, claims, and artifacts.

REFERENCES

Art of Problem Solving Wiki. 2023 AMC 12A problems. https://artofproblemsolving.com/wiki/index.php/2023_AMC_12A_Problems, 2023. Accessed: 2026-05-07.

Art of Problem Solving Wiki. 2024 AIME I problems. https://artofproblemsolving.com/wiki/index.php/2024_AIME_I_Problems, 2024. Accessed: 2026-05-07.

Art of Problem Solving Wiki. 2025 AIME I problems. https://artofproblemsolving.com/wiki/index.php/2025_AIME_I_Problems, 2025. Accessed: 2026-05-07.

Lang Cao, Hui Ruan, Yongqian Li, Peng Chao, Wu Ning, Haonan Song, Renhong Chen, and Yitong Li. TreeAdv: Tree-structured advantage redistribution for group-based RL, 2026.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

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.

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.

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.

Amirhossein Kazemnejad, Milad Aghajohari, Eva Portelance, Alessandro Sordoni, Siva Reddy, Aaron Courville, and Nicolas Le Roux. VinePPO: Refining credit assignment in RL training of LLMs. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 29557–29590. PMLR, 2025.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, et al. Solving quantitative reasoning problems with language models. *Advances in neural information processing systems*, 35:3843–3857, 2022.

Yizhi Li, Qingshui Gu, Zhoufutu Wen, Ziniu Li, Tianshun Xing, Shuyue Guo, Tianyu Zheng, Xin Zhou, Xingwei Qu, Wangchunshu Zhou, et al. TreePO: Bridging the gap of policy optimization and efficacy and inference efficiency with heuristic tree-based modeling. *arXiv preprint arXiv:2508.17445*, 2025.

Yugu Li, Zehong Cao, Jianglin Qiao, and Siyi Hu. SSVPO: Effective step-level credit assignment for RL training of language models. In *International Conference on Learning Representations*, 2026.

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 *International Conference on Learning Representations*, 2024.

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.

J. Ross Quinlan. Induction of decision trees. *Machine Learning*, 1:81–106, 1986. doi: 10.1007/BF00116251.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 1889–1897, 2015. URL https://proceedings.mlr.press/v37/schulman15.html.

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.

Qwen Team. Qwen2.5: A party of foundation models, September 2024. URL https://qwenlm.github.io/blog/qwen2.5/.

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

Hieu Tran, Zonghai Yao, and Hong Yu. Exploiting tree structure for credit assignment in reinforcement learning with large language models. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 10795–10810. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.findings-acl.524.

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

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, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.510.

Zhicheng Yang, Zhijiang Guo, Yinya Huang, Xiaodan Liang, Yiwei Wang, and Jing Tang. TreeRPO: Tree relative policy optimization. *arXiv preprint arXiv:2506.05183*, 2025.

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.

Yu Yue, Yufeng Yuan, Qiying Yu, Xiaochen Zuo, Ruofei Zhu, Wenyuan Xu, Jiaze Chen, Chengyi Wang, TianTian Fan, Zhengyin Du, et al. VAPO: Efficient and reliable reinforcement learning for advanced reasoning tasks. *arXiv preprint arXiv:2504.05118*, 2025.

George Udny Yule. On the association of attributes in statistics: With illustrations from the material of the childhood society, &c. *Philosophical Transactions of the Royal Society of London. Series A*, 194:257–319, 1900. doi: 10.1098/rsta.1900.0019.

Yijun Zhang, Fan Xu, Jiaxin Ding, Yule Xie, Shiqing Gao, Xin Ding, Haoxiang Zhang, Luoyi Fu, and Xinbing Wang. Information gain-based rollout policy optimization: An adaptive tree-structured rollout approach for multi-turn LLM agents, 2026.

Tianyu Zheng, Tianshun Xing, Qingshui Gu, Taoran Liang, Xingwei Qu, Xin Zhou, Yizhi Li, Zhoufutu Wen, Chenghua Lin, Wenhao Huang, et al. FR3E: First return, entropy-eliciting explore. *arXiv preprint arXiv:2507.07017*, 2025.

A Reproducibility Details

A.1 Hyperparameters and Implementation

Implementation details. All experiments are implemented in VeRL, with FSDP for training and vLLM for rollouts. Training is conducted on eight NVIDIA H800 GPUs. We filter prompts longer than 1,024 tokens and set the maximum response length to 8,192. Unless otherwise specified, we train for 500 steps with a learning rate of 1×10^{-6}, a batch size of 256, eight gradient-accumulation steps, and a KL coefficient of 0.001. Rollouts use a temperature of 1.0 and top-k sampling with $k = 20$. For DGPO, $K = 2$ preliminary rollouts calibrate the entropy threshold, and the sustained-entropy window has $n = 4$ tokens. Semantic representations are extracted from a selected layer of the current model, frozen for rollout collection. Tree construction uses a branch factor of 8, a maximum depth of 3, and a leaf budget of 8. Expansion stops when EOS is reached, when semantic pruning removes every new continuation, or when eight complete trajectories have been collected. Thus, the retained training tree has at most eight completed trajectories and, correspondingly, at most eight verifier calls per prompt. The leaf budget does not count preliminary entropy-calibration rollouts or pruned candidates, which still incur generation cost.

Hyperparameter selection protocol. All DGPO-specific hyperparameters were fixed before the reported benchmark evaluation. We performed neither grid search nor automated hyperparameter optimization, and did not revise the configuration after inspecting benchmark results. A single configuration was shared across all reported backbones and benchmarks, with no model-specific or benchmark-specific tuning. Because no parameter search was conducted, tuning ranges are not applicable. Table 3 summarizes the fixed values and their roles.

Table 3: **DGPO hyperparameters used in all reported experiments.**

Parameter	Value	Role
K	2	Preliminary rollouts for estimating the entropy threshold.
n	4	Number of consecutive high-entropy tokens required for a boundary.
γ	0.8	Quantile used to determine the prompt-specific entropy threshold.
M	8	Number of candidate continuations sampled at each branching point.
ℓ	20	Maximum length, in tokens, of each candidate continuation.
D	3	Maximum tree depth.
B	8	Maximum number of completed root-to-leaf trajectories.
τ_S	0.4	Semantic-similarity threshold for near-duplicate pruning.
λ	10^{-6}	Additive smoothing constant in the directional log-odds ratio.
$\tau_{\min}$	2	Minimum support required for the segment-present and segment-absent groups.
ϵ	10^{-6}	Numerical stability constant in the gain ratio and DIG normalization.
α	1	Weight of the normalized global DIG correction.
β	0.5	Weight of the normalized local DIG correction.

The maximum prompt and response lengths are 1,024 and 8,192 tokens, respectively. We impose no additional tree-level token cap beyond these sequence-length limits and the structural stopping criteria described above.

Support and credit coverage. With $\tau_{\min} = 2$, each comparison requires at least two paths in both the edge-containing and edge-absent groups. Thus, terminal edges with a single descendant leaf receive zero DIG, and a local subtree with fewer than four leaves cannot provide supported local credit. With $B = 8$, DIG acts on supported shared segments; other tokens retain their base advantage. A flat tree with eight one-leaf children provides no supported DIG correction. Coverage depends on the retained tree topology.

A.2 Training Algorithm

Algorithm 1 outlines a single training update step of the proposed DGPO framework. It details the key operations including adaptive tree exploration, verifier-guided credit assignment, edge-level normalization, and policy optimization.

Algorithm 1 One DGPO Training Update

Require: Policy π_θ, prompt batch $\mathcal{Q}$, verifier Verify
Require: Calibration count K, entropy parameters γ, n, candidate count and length M, ℓ
Require: Tree limits D, B, similarity threshold τ_S, support threshold $\tau_{\min}$, smoothing λ, weights α, β
Freeze rollout parameters $\theta_{\text{old}} \leftarrow \theta$
for $x \in \mathcal{Q}$ **do**
  Estimate $\tau_H(x)$ from K preliminary rollouts
  Collect $\mathcal{T}_x$ with uncertainty-triggered branching and similarity pruning
  Extract retained complete paths $\mathcal{P}_x$
  $Y_i \leftarrow \text{Verify}(x, o_i)$ for each $\tau_i \in \mathcal{P}_x$
  for $z \in \mathcal{E}(\mathcal{T}_x)$ **do**
    Compute global and local DIG using $\mathcal{P}_x$ and $\mathcal{P}_{p(z)}$
  end for
  Normalize each credit type over its supported distinct edges by Eq. (12)
  Form token advantages by Eq. (13)
end for
Update θ using Eq. (14) and the base optimizer's regularization

B Effectiveness Experiments

B.1 Statistical Significance Analysis

Table 4 reports the dispersion in average pass@1 accuracy and the results of the paired significance test between DGPO and TreeRL. Mean and standard deviation are computed across eight paired stochastic evaluation runs. We use a two-sided exact Wilcoxon signed-rank test, with $p < 0.05$ considered statistically significant.

Table 4: **Statistical comparison between DGPO and TreeRL.** Results are the average pass@1 accuracy (%) across the six benchmarks, reported as mean ± standard deviation over eight paired evaluation runs. Bold p-values indicate statistical significance at $p < 0.05$.

Model	TreeRL	DGPO	Wilcoxon p
`Qwen2.5-3B`	29.23 ± 0.64	30.71 ± 0.83	**0.0078**
`Qwen2.5-7B`	41.10 ± 1.14	42.35 ± 0.78	**0.0234**
`Qwen3-4B-Base`	47.70 ± 0.87	49.60 ± 1.43	**0.0391**
`Llama-3.2-3B-Instruct`	23.76 ± 1.51	24.66 ± 1.26	0.1484

B.2 Additional Advantage and Backbone Comparisons

Direct segment-advantage baseline. For an edge $z = (u \to v)$, define $\mathcal{P}_u = \{\tau_i \in \mathcal{P} : \tau_i \text{ reaches } u\}$ as the set of trajectories reaching parent u. All counts below are computed on $\Omega = \mathcal{P}_u$. For edges whose segment-present and segment-absent groups are both nonempty, the empirical branch value, prefix value, and direct advantage are

$$\begin{aligned} \widehat{Q}(u,z) &= \frac{N_{11}}{N_{11}+N_{10}}, \\ \widehat{V}(u) &= \frac{N_{11}+N_{01}}{N_{11}+N_{10}+N_{01}+N_{00}}, \\ \widehat{A}_{\text{direct}}(u,z) &= \widehat{Q}(u,z) - \widehat{V}(u). \end{aligned} \tag{16}$$

For nonempty segment-present and segment-absent groups, this also equals $(1-\widehat{p}_z)(\widehat{Q}(u,z) - \widehat{Q}(u,\neg z))$, where $\widehat{p}_z = (N_{11}+N_{10})/|\mathcal{P}_u|$ and $\widehat{Q}(u,\neg z) = N_{01}/(N_{01}+N_{00})$. Thus direct advantage estimates the return difference, whereas unsmoothed local DIG has the same population direction with nonlinear information weighting (Proposition 1). DIG is a surrogate credit score, not an unbiased advantage estimator. Our theory gives sufficient improvement conditions for the combined global–local score, without claiming optimality of this combination or of the fixed weights α and β.

Table 5: **Advantage comparison.** Four-benchmark average.

Method	Avg.
GRPO	58.88
DGPO branching + Direct advantage	59.42
DGPO branching + TreeRL advantage	60.26
DGPO	**61.59**

Table 6: **Compatibility with DAPO.** Four-benchmark average.

Method	Avg.
DAPO	59.23
DAPO + DIG	**61.52**

Tables 5 and 6 supplement the comparisons in Section 4.3 on `Qwen3-4B-Base`. Both report average performance across AIME25, AMC23, MATH-500, and OlympiadBench. The direct segment-advantage baseline reaches an average accuracy of 59.42, whereas DGPO with DIG reaches 61.59 using the same branching procedure, a gain of 2.17 points. This comparison supports the effectiveness of the proposed DIG-based segment-credit design relative to direct segment advantages. DIG also transfers beyond the GRPO optimizer: adding it to DAPO improves the four-benchmark average from 59.23 to 61.52, a gain of 2.29 points.

B.3 Training and Compute Diagnostics

Figure 5 compares mean pass@1 against wall-clock training time on Qwen3-4B-Base. DGPO was trained for 500 steps, while the baselines received 540–660 steps to reach comparable total runtimes of approximately 36–37 h. DGPO initially trailed the baselines at comparable elapsed times, but its relative performance improved during training. Linear interpolation estimated that DGPO overtook GRPO, TreePO and TreeRL at 5.6, 11.1 and 20.1 h, respectively. DGPO remained ahead over the subsequent overlapping observation periods for each comparison. At the final recorded checkpoints, DGPO achieved 49.33% after 36.18 h, compared with 47.61% for TreeRL after 36.84 h, 47.12% for TreePO and 46.95% for GRPO after 37.08 h. Its advantage over TreeRL, the strongest baseline at the final checkpoint, was 1.72 points.

Panel (b) compares DGPO's estimated time to reach each baseline's final recorded accuracy with that baseline's final training runtime. Linear interpolation gave DGPO match times of 22.7 h for TreeRL, 18.0 h for TreePO, and 15.6 h for GRPO. The corresponding baselines were trained for 36.8, 35.9, and 37.1 h, respectively. These final runtimes were approximately 1.63, 2.00, and 2.38 times as long as DGPO's estimated match times. Relative to these final runtimes, DGPO reached the corresponding scores approximately 14.2, 17.9, and 21.5 h earlier, respectively. Thus, DGPO achieved higher final accuracy within a comparable training duration and reached each baseline's final score earlier.

B.4 Pass@8 Results

Beyond pass@1, we evaluate pass@8 to examine each method's effectiveness under test-time sampling. For each test problem, we draw eight independent candidate responses from the policy; a problem is considered solved if any candidate is verified as correct. Thus, pass@8 complements pass@1 by assessing the probability of sampling at least one correct response within eight trials. As shown in Table 7, DGPO achieves the highest average pass@8 accuracy across all model backbones on the six benchmarks. This indicates a higher probability of sampling a valid solution within this inference budget.

C Policy-Improvement Analysis of Directional Information Gain

We relate DIG to the direct advantage in Eq. (16), explain the global–local combination, and derive a first-order improvement condition for the complete normalized score without requiring equal branch probabilities.

Setting. Fix a finite retained tree with root u_0, terminal rewards $Y \in \{0, 1\}$, and reference path distribution μ. Omit zero-probability branches and hold all scores, masks, and normalization statistics fixed during each update. Uniform leaf weights recover unsmoothed empirical tree statistics; other

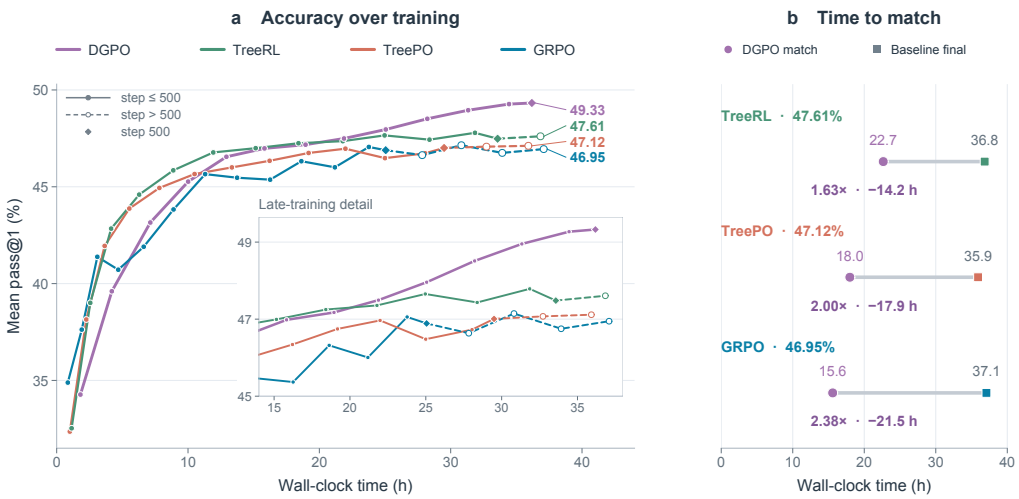

Figure 5: **Accuracy and time to match baseline endpoint scores on `Qwen3-4B-Base`.** (a) Mean pass@1 versus wall-clock training time, excluding step 0. Solid lines with filled markers denote steps ≤ 500; dashed lines with open markers indicate additional baseline training. Diamonds mark step 500, and the inset enlarges the 14–38 h interval. (b) Purple circles show DGPO's estimated time to first reach each baseline's final recorded mean pass@1. Colored squares show the corresponding baseline's final training runtime. Each row reports the ratio of baseline final runtime to DGPO match time and the reduction in elapsed time relative to baseline final runtime. Match times are estimated by linear interpolation between adjacent evaluations.

Table 7: **Mean pass@8 accuracy.** Results are reported on the same evaluation benchmarks as Table 1. For each problem, eight independently sampled responses are generated, and the problem is counted as correct if any response passes verification. The best result in each column is in **bold**, and the strongest non-DGPO result is underlined.

Model	Method	AIME25	AIME24	AMC23	MATH-500	Minerva	OlympiadBench	Avg.
`Qwen2.5-3B`	Base	0.00	6.25	21.40	51.88	9.16	27.06	19.29
	GRPO	15.78	21.67	64.32	77.92	17.00	51.53	41.37
	TreePO	13.28	21.49	65.83	80.31	17.94	53.63	42.08
	TreeRL	**16.33**	23.95	69.03	82.66	18.64	54.27	44.15
	DGPO	16.06	**25.10**	**74.76**	**84.21**	**20.34**	**55.83**	**46.05**
`Qwen2.5-7B`	Base	8.89	5.49	48.60	73.49	19.58	42.05	33.02
	GRPO	21.47	31.58	78.72	89.75	24.59	65.24	51.89
	TreePO	23.31	32.67	82.50	88.89	23.25	65.87	52.75
	TreeRL	24.67	33.33	83.01	90.01	24.33	65.67	53.50
	DGPO	**26.73**	**36.22**	**84.48**	**91.20**	**24.81**	**66.95**	**55.07**
`Qwen3-4B-Base`	Base	4.34	13.12	53.75	78.46	26.43	50.24	37.72
	GRPO	35.54	**47.56**	88.08	93.04	28.12	73.83	61.03
	TreePO	35.80	43.72	89.64	93.33	27.77	74.15	60.74
	TreeRL	35.67	46.17	90.30	93.67	28.53	74.48	61.47
	DGPO	**37.10**	46.17	**90.62**	**94.25**	**29.08**	**75.14**	**62.06**
`Llama-3.2-3B-Instruct`	Base	0.00	7.26	33.08	61.91	16.83	32.78	25.31
	GRPO	7.62	23.08	58.61	72.16	15.01	42.08	36.43
	TreePO	8.25	24.62	62.99	71.84	16.93	43.10	37.96
	TreeRL	9.60	25.87	59.32	72.90	**19.92**	43.86	38.58
	DGPO	**11.68**	**28.81**	**66.26**	**74.98**	19.39	**45.94**	**41.18**

positive path weights are also allowed. This tree-induced reference distribution need not equal the language-model policy.

C.1 DIG AS INFORMATION-WEIGHTED ADVANTAGE

For either comparison population Ω in Eqs. (10)–(11), write

$$\begin{aligned} p &= \Pr_{\mu}(Z = 1 \mid \Omega), \qquad m_i = \mathbb{E}_{\mu}[Y \mid Z = i, \Omega], \\ m &= pm_1 + (1-p)m_0, \qquad L = m_1(1-m_0) + (1-m_1)m_0. \end{aligned} \tag{17}$$

For $0 < p < 1$ and $L > 0$, let $C(z;\Omega)$ denote the unsmoothed DIG score, with a fixed support mask $M(z;\Omega) \in \{0,1\}$. The identity $\tanh(\frac{1}{2}\log(a/b)) = (a-b)/(a+b)$ gives

$$C(z;\Omega) = M(z;\Omega)\frac{I_{\mu}(Y;Z \mid \Omega)}{h(p) + \epsilon_{\text{num}}}\frac{m_1 - m_0}{L}, \tag{18}$$

where h is binary entropy and logarithms are natural. Constant-outcome and degenerate-occurrence cases receive zero score.

Proposition 1 (Nonnegative advantage reweighting)**.** *For a nondegenerate occurrence variable,*

$$\begin{aligned} C(z;\Omega) &= w(z;\Omega)(m_1 - m), \\ w(z;\Omega) &= \frac{M(z;\Omega)I_{\mu}(Y;Z \mid \Omega)}{[h(p) + \epsilon_{\text{num}}](1-p)L} \geq 0. \end{aligned} \tag{19}$$

Set $w = 0$ when $L = 0$. A supported score has the same strict sign as $m_1 - m$ whenever that difference is nonzero.

Proof. Use $m_1 - m = (1-p)(m_1 - m_0)$ in Eq. (18). Mutual information is nonnegative, and is positive for $0 < p < 1$ whenever $m_1 \neq m_0$. If $L = 0$, the outcome is constant and both sides vanish. □

DIG thus reweights a return difference without reversing its direction. Its mutual-information numerator is the weighted Jensen–Shannon divergence between $\text{Ber}(m_1)$ and $\text{Ber}(m_0)$, making DIG a discrimination-weighted surrogate rather than an unbiased estimator of $Q - V$.

What the information factor changes. For a supported edge, fix $p, m \in (0,1)$ and let $\delta = m_1 - m_0 \to 0$, so $m_1 = m + (1-p)\delta$ and $m_0 = m - p\delta$. A Taylor expansion gives

$$C(z;\Omega) = \frac{p(1-p)}{4[h(p) + \epsilon_{\text{num}}]m^2(1-m)^2}\,\delta^3 + O(\delta^4). \tag{20}$$

Indeed, $I_{\mu}(Y;Z \mid \Omega) = p(1-p)\delta^2/[2m(1-m)] + O(\delta^3)$ and $L = 2m(1-m) + O(\delta)$. Whereas the direct return difference is $(1-p)\delta$, raw DIG attenuates weak associations cubically. This characterizes the nonlinear weighting before tree-wise standardization; it is not a finite-sample confidence or variance-reduction guarantee.

C.2 WHAT THE GLOBAL–LOCAL COMBINATION MEASURES

For a nonterminal node u and child edge a, define

$$\begin{aligned} q_u(a) &= \mu(a \mid u), \qquad Q_u(a) = \mathbb{E}_{\mu}[Y \mid u, a], \\ V_u &= \mathbb{E}_{a \sim q_u} Q_u(a), \quad A_u(a) = Q_u(a) - V_u. \end{aligned} \tag{21}$$

Let $J(\mu) = V_{u_0}$, $\Delta_u = V_u - J(\mu)$, and let $d_{\mu}(u)$ be the probability of reaching u. Proposition 1 yields the exact decomposition

$$\begin{aligned} C_{\text{l}}(u,a) &= w_{\text{l}}(u,a)A_u(a), \\ C_{\text{g}}(u,a) &= w_{\text{g}}(u,a)\big(A_u(a) + \Delta_u\big), \end{aligned} \tag{22}$$

where each nonnegative w_r includes its support mask; degenerate occurrences have $w_r = C_r = 0$. Local DIG weights the within-prefix advantage, while global DIG weights the same advantage plus the prefix's success-rate difference from the root. The two scores therefore express different comparisons. The prefix offset is shared by siblings, but its global weight need not be shared.

In this appendix, g and l abbreviate global and local credit. To retain the implementation's normalization exactly, let $M_r(u,a)$ be the support mask, $\bar{C}_r$ the mean over supported tree edges, and $s_r = \sigma_r + \epsilon_{\text{num}} > 0$, for $r \in \{\text{g}, \text{l}\}$. Then

$$\begin{aligned} \widetilde{C}_r(u,a) &= \frac{C_r(u,a) - \bar{C}_r M_r(u,a)}{s_r}, \\ S_u(a) &= B_u(a) + \alpha \widetilde{C}_{\text{g}}(u,a) + \beta \widetilde{C}_{\text{l}}(u,a). \end{aligned} \tag{23}$$

An empty supported set contributes zero. We use $B_u(a) = b_u + c_u Q_u(a)$, with $c_u \geq 0$, and $\alpha, \beta \geq 0$. This includes the conditional expectation of an affine, group-normalized outcome reward when group statistics are fixed; $B_u = 0$ isolates the complete DIG correction. Crucially, the factor M_r remains in the centering term: centering supported edges only is not a common baseline for all children.

C.3 Whole-Tree Improvement of the Combined Score

Consider the independent branch-policy update

$$\begin{aligned} q_u^\eta(a) &= \frac{q_u(a) e^{\eta S_u(a)}}{Z_u(\eta)}, \qquad \eta \geq 0, \\ Z_u(\eta) &= \sum_b q_u(b) e^{\eta S_u(b)}, \end{aligned} \tag{24}$$

and let μ_η be the resulting path distribution. For $\eta > 0$, this is the maximizer of $\mathbb{E}_{a\sim q} S_u(a) - \text{KL}(q\|q_u)/\eta$ on the retained action support. Write $\mathbb{E}_u$ for expectation under q_u, and set $\omega_{\text{g}} = \alpha$, $\omega_{\text{l}} = \beta$.

Theorem 1 (First-order alignment decomposition for normalized global–local DIG)**.** *Under the fixed-tree setting above, with arbitrary positive branch probabilities,*

$$J(\mu_\eta) - J(\mu) = \eta(\Gamma_+ + \Gamma_{\text{res}}) + O(\eta^2), \tag{25}$$

where

$$\Gamma_+ = \sum_u d_\mu(u)\, \mathbb{E}_u\left[\left(c_u + \frac{\alpha w_{\text{g}}}{s_{\text{g}}} + \frac{\beta w_{\text{l}}}{s_{\text{l}}}\right) A_u^2\right] \geq 0, \tag{26}$$

$$\Gamma_{\text{res}} = \sum_u d_\mu(u) \left[\frac{\alpha \Delta_u}{s_{\text{g}}} \mathbb{E}_u[w_{\text{g}} A_u] - \sum_{r\in\{\text{g},\text{l}\}} \frac{\omega_r \bar{C}_r}{s_r} \mathbb{E}_u[M_r A_u]\right]. \tag{27}$$

All sums are over nonterminal nodes. If $\Gamma_+ + \Gamma_{\text{res}} > 0$, then $J(\mu_\eta) > J(\mu)$ for every sufficiently small positive η. In particular, $\Gamma_+ > |\Gamma_{\text{res}}|$ is sufficient. A zero first-order term alone does not determine improvement.

Proof. The finite-tree performance-difference identity (Schulman et al., 2015) follows by telescoping $V_v - V_u$ along each root-to-leaf path, with terminal value equal to its reward:

$$J(\mu_\eta) - J(\mu) = \sum_u d_{\mu_\eta}(u) \sum_a q_u^\eta(a) A_u(a). \tag{28}$$

Since $q_u^\eta(a) = q_u(a)\{1 + \eta[S_u(a) - \mathbb{E}_u S_u] + O(\eta^2)\}$ and $\mathbb{E}_u A_u = 0$, the inner sum is $\eta \mathbb{E}_u[S_u A_u] + O(\eta^2)$. Also $d_{\mu_\eta}(u) = d_\mu(u) + O(\eta)$. The tree is finite, so

$$\frac{d}{d\eta} J(\mu_\eta)\Big|_{\eta=0} = \sum_u d_\mu(u) \mathbb{E}_u[S_u A_u]. \tag{29}$$

Substitute Eqs. (22)–(23), and use $\mathbb{E}_u[B_u A_u] = c_u \mathbb{E}_u A_u^2$. The advantage-squared terms give Γ_+; the prefix-offset and masked-centering terms give Γ_{res}. The expansion and the strict-improvement claim follow. □

Implications for the mixture. Both components contribute nonnegative weighted advantage-squared terms to Γ_+. The residual separates global weighting of the prefix offset from partial-support centering. The first effect vanishes when $\Delta_u = 0$ or w_g is constant across siblings; the second vanishes for sibling-constant masks. Improvement follows when favorable alignment dominates these effects. For fixed tree statistics, $\Gamma_+ + \Gamma_{\text{res}}$ is affine in α, β, giving conditions on mixture weights without claiming an optimal choice. The reported fixed weights are evaluated empirically; not every collected tree is assumed to satisfy the criterion.

Finite-step special cases.

Corollary 1 (Finite-step improvement)**.** *For every $\eta \geq 0$, Eq. (24) satisfies $J(\mu_\eta) \geq J(\mu)$ in either of the following cases:*

1. *At every node, $S_u(a) = \kappa_u C_1(u, a) + c_u$ for some $\kappa_u > 0$ and common shift $c_u \in \mathbb{R}$.*
2. *S_u is the complete score in Eq. (23), and at every node siblings have equal reference probability and each credit's mask is constant across siblings.*

Proof. For the first case, $C_1 A_u \geq 0$ implies

$$\sum_a q_u^\eta(a) A_u(a) = \frac{\mathbb{E}_u[(e^{\eta\kappa_u C_1} - 1)A_u]}{\mathbb{E}_u[e^{\eta\kappa_u C_1}]} \geq 0. \tag{30}$$

Common shifts cancel and positive scaling changes the effective step size. Equation (28) then gives whole-tree improvement.

For the second case, fix occurrence mass p and comparison success rate m. Write $Q = m_1$ and $R(Q) = (m - pQ)/(1 - p)$. The information factor $I(Q) = h(m) - ph(Q) - (1 - p)h(R(Q))$ has derivative $I'(Q) = p[h'(R(Q)) - h'(Q)]$ with the sign of $Q - m$. The polarity $T(Q) = \tanh([\text{logit}(Q) - \text{logit}(R(Q))]/2)$ has the same sign and is increasing. Hence $(IT)' = I'T + IT' \geq 0$. Equal sibling probabilities fix both local masses $p = q_u(a)$ and global masses $p = d_\mu(u)q_u(a)$, so both scores are nondecreasing in $Q_u(a)$. Constant masks and common normalization preserve this ordering, as does the affine base score. Therefore

$$\sum_a q_u^\eta(a) A_u(a) = \frac{\text{Cov}_{q_u}(Q_u, e^{\eta S_u})}{Z_u(\eta)} \geq 0, \tag{31}$$

where the last inequality follows from pairwise comonotonicity. Continuity covers boundary cases and degenerate scores contribute zero. Apply Eq. (28) again. □

Connection to empirical results. Table 5 compares segment-credit designs using DGPO branching on `Qwen3-4B-Base`. Full DGPO achieves a four-benchmark average accuracy of 61.59, compared with 59.42 for direct segment advantages and 60.26 for TreeRL advantages, improving by 2.17 and 1.33 points, respectively. These results support the effectiveness of the proposed credit design. Proposition 1 complements this comparison by showing that unsmoothed local DIG is a nonnegative reweighting of the population direct advantage, while Eq. (20) characterizes its nonlinear information weighting before standardization.

D Qualitative Case Study of Segment-Level Credit Assignment

Contrasting reasoning segments. The tree exposes a consequential difference in divisor enumeration. The segment retaining the complete list, including 175, receives a positive DIG correction of $+1.44$, whereas the alternative omitting 175 receives a negative correction of -1.24. These annotations illustrate how the recorded correction distinguishes segments at a specific reasoning decision. The marked pair of similar continuations also illustrates how semantic pruning removes redundant exploration.

Complementarity with outcome supervision. In the illustrated flat-rollout group, repeated versions of the same omission lead to uniformly unsuccessful outcomes and provide no relative reward signal for distinguishing the sampled responses. The tree instead exposes alternative intermediate segments and assigns differentiated corrections. This case provides a qualitative illustration of the interaction

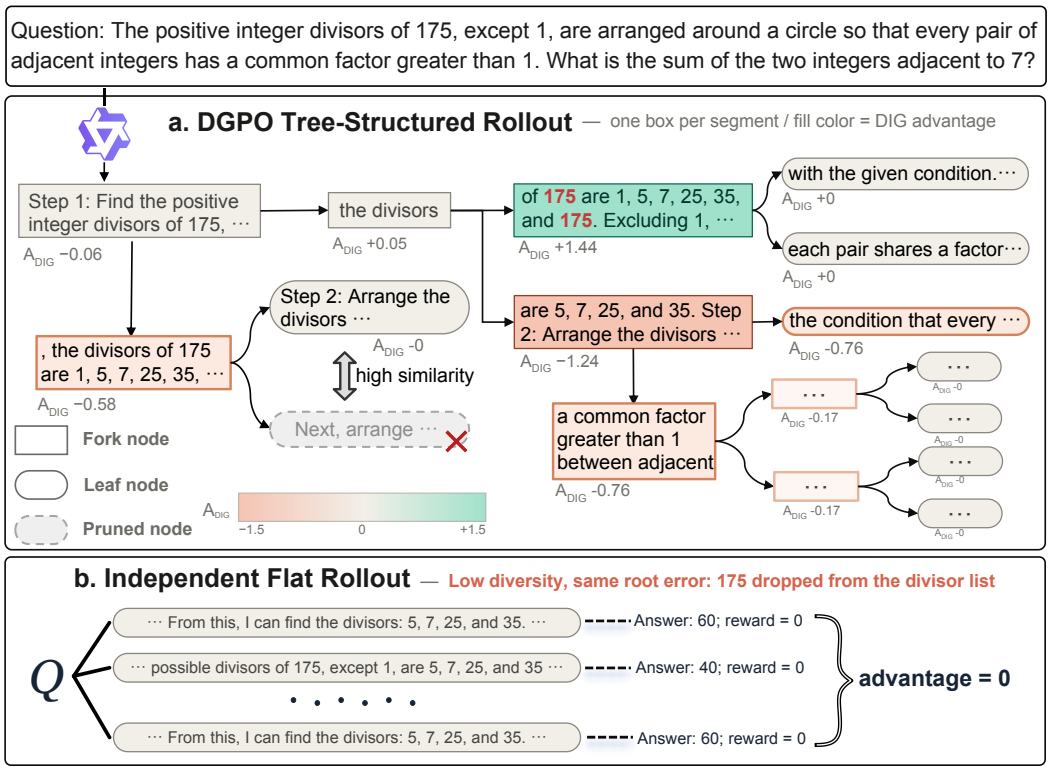

Figure 6: **Qualitative comparison of tree-structured DGPO rollouts and independent flat rollouts.** Each box in panel (a) represents a reasoning segment. Colors and numerical annotations indicate the recorded DIG correction term, denoted by A_{DIG}, rather than the total policy advantage. A cross marks a continuation removed by semantic-similarity pruning, and ellipses indicate omitted text. Panel (b) shows flat rollouts that repeatedly omit 175 from the divisor list and return incorrect answers, yielding zero rewards and zero group-normalized outcome advantages in the illustrated group.

between exploration and credit assignment; aggregate performance and component contributions are evaluated in the main experiments.

Figure 6 illustrates how tree-structured exploration and directional segment credit interact on a mathematical reasoning problem. The case contrasts alternative continuations within a DGPO rollout tree with independently sampled flat rollouts.