WHERE SHOULD THE CREDIT LIE? ATTENTION-INDUCED ANCHOR-BRIDGE ATTRIBUTION POLICY OPTIMIZATION FOR LLM REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement Learning with Verifiable Rewards (RLVR) has improved the reasoning capabilities of large language models. Methods such as GRPO provide critic-free optimization, but they assign uniform, sequence-level credit to all tokens and thus overlook the disparate roles that tokens play in the reasoning process. To address this, we leverage the model's own attention patterns to construct an attribution map that characterizes how information and contributions flow among tokens. This allows us to identify two pivotal token types: **Anchor Tokens**, which are frequently reused by subsequent reasoning steps, and **Bridge Tokens**, which connect preceding context with later reasoning. Building on these observations, we propose **Anchor-Bridge Attribution Policy Optimization (ABPO)**, a token-level credit assignment framework guided by attribution relationships. Specifically, ABPO computes an AnchorScore and a BridgeScore for each token using the AnchorRank and BridgeTrace algorithms, respectively, and then fuses them into a SynergyScore for advantage reweighting. Experimental results across diverse reasoning tasks demonstrate that ABPO significantly and consistently outperforms the GRPO baseline and other credit assignment methods (+2.6 points on math, +7.0 points on other domains vs. GRPO). Furthermore, ABPO generates more concise reasoning chains (-3.15% tokens on math, -30.15% on other domains over GRPO), and achieves faster training convergence, highlighting its dual strengths in both effectiveness and efficiency.

1 INTRODUCTION

Reinforcement Learning with Verifiable Rewards (RLVR) (Shao et al., 2024; Lambert et al., 2024) has recently shown strong success in enhancing the complex reasoning abilities of Large Language Models (LLMs) (OpenAI, 2025; DeepSeek-AI et al., 2025; Yang et al., 2025a), moving beyond pattern matching toward complex problem-solving. Rather than a learned reward model, RLVR uses verifiable, rule-based rewards, as in mathematical (Hendrycks et al., 2021; Zhang & Team Math-AI, 2024; He et al., 2024) and coding (Jimenez et al., 2023; Jain et al., 2024) tasks with checkable answers or executable tests. Among these, Group Relative Policy Optimization (GRPO) (Shao et al., 2024) is a critic-free algorithm that avoids learning a value model by estimating relative advantages within each rollout group, providing a practical optimization method for multi-step reasoning.

However, GRPO relies on a coarse sequence-level credit assignment: a single group-normalized advantage is uniformly broadcast to all tokens in a sampled response (Shao et al., 2024; Tan et al., 2025). This ignores differences in token-level influence on policy updates; low-probability tokens can disproportionately dominate these updates (Yang et al., 2025b). Consequently, a growing body of research refines sequence-level credit assignment into token-level credit assignment, either by introducing external token-level supervision (Xie et al., 2025; Sun et al., 2025) or by leveraging model-internal signals to identify important tokens (Cui et al., 2025; Lin et al., 2025a; Cheng et al., 2026; Wang et al., 2025; Vassoyan et al., 2025; Li et al., 2025); we review both lines in Section 2. Despite prior efforts, the attribution and dependency relationships among tokens during the model's

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.

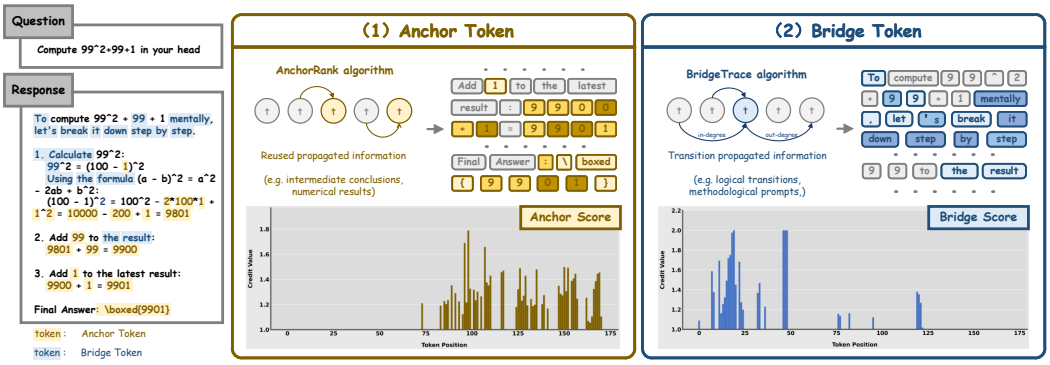


Figure 1: Anchor and Bridge attribution. On the attention-induced attribution map, Anchor Tokens carry high attribution propagation and Bridge Tokens high incoming and outgoing flow. Anchors are reused intermediate results; bridges, high-level reasoning cues.

reasoning, as well as how information propagates across sequences, remain insufficiently explored, even though such relations naturally form a graph over the response.

In this paper, we derive attribution from the model's internal attention, and design two graph-based algorithms over it to distinguish different tokens. Since the attention map reflects how each token attends to others, transposing it reverses this direction and reveals how each token contributes to others. On this attribution map, tokens play different roles in information propagation, accumulation, and relay, and we identify two types of pivotal tokens, shown in Figure 1. The first type is **Anchor Tokens**, detected by our designed *AnchorRank* algorithm, which captures recursively reinforced tokens along the information flow. They mostly correspond to intermediate values computed at sub-steps, which are frequently referenced by later steps and thus participate more in propagation across the entire sequence. The second type is **Bridge Tokens**, detected by the proposed *BridgeTrace* algorithm, which identifies tokens with the highest geometric mean of incoming and outgoing attribution flow. These tokens both absorb upstream information and pass it downstream, mediating the transition from the preceding context to the following reasoning chain, and often reflect high-level reasoning patterns that boost next solving steps and extend the chain.

Building on these insights, we propose an **Anchor-Bridge Attribution Policy Optimization (ABPO)** framework that aligns credit assignment with the model's attention dependencies. As shown in Figure 2, given the attribution map, ABPO computes an AnchorScore and a BridgeScore for each token, fuses them through an ℓ_2 mechanism into the overall SynergyScore, and performs differential reinforcement by reweighting token advantages accordingly. In this way, we precisely reinforce pivotal tokens while implicitly suppressing low-impact ones as a byproduct.

We conduct a series of comprehensive experiments on the proposed ABPO framework and demonstrate its advantages in three aspects. **(1) Higher accuracy.** Across multiple domains, including mathematical reasoning and cross-domain tasks, our method consistently outperforms the GRPO baseline and other credit assignment works, achieving average gains of +2.6 points on mathematical benchmarks and +7.0 points on cross-domain evaluations, and the margin widens as problem complexity increases. **(2) Concise reasoning chains.** Our method generates more accurate reasoning chains with fewer generated tokens, indicating that it not only precisely emphasizes important tokens but also reduces redundant and uninformative outputs. **(3) Training efficiency.** Our method converges faster than the baseline during training, enabling models to acquire correct reasoning patterns more efficiently. In summary, our approach achieves both strong effectiveness and high efficiency, and facilitates the emergence of coherent, holistic reasoning behaviors in large language models. Our contributions can be summarized as follows:

- We present a graph-based credit assignment mechanism that turns the model's attention into an attribution map over response tokens. On this map we identify two types of pivotal tokens, namely **Anchor Tokens** that are frequently attended to by subsequent reasoning steps, and **Bridge Tokens** that mediate the transition from the preceding context to the following reasoning chain.

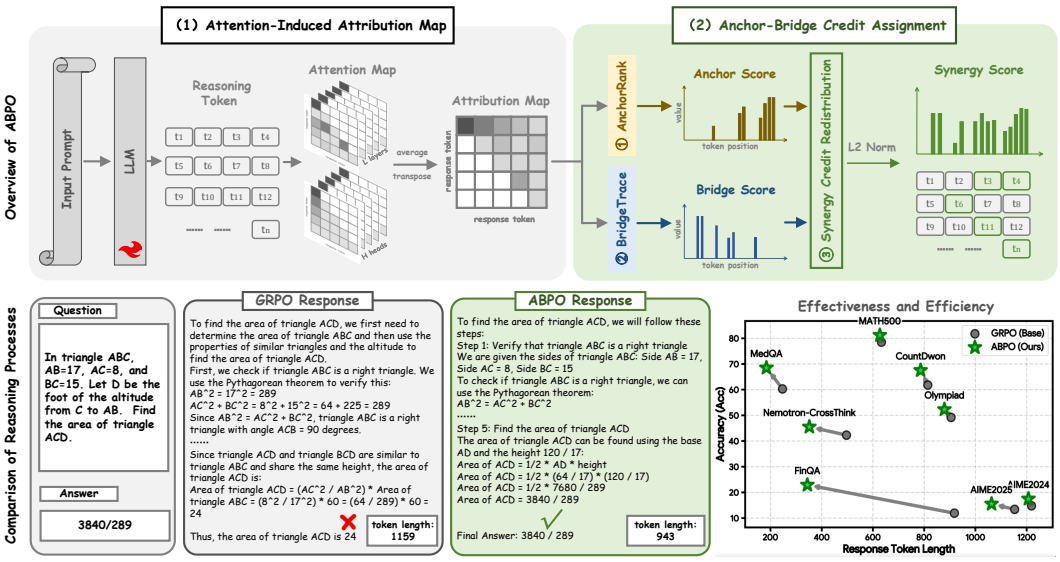


Figure 2: Overview of ABPO. **Top:** the pipeline turning LLM attention maps into token-level Synergy Scores via AnchorRank and BridgeTrace. **Bottom:** a case study and accuracy-versus-length results showing ABPO reasons more accurately with fewer tokens.

- We develop an **Anchor-Bridge Attribution Policy Optimization (ABPO)** framework to align credit assignment with the model's attention dependencies. We compute token-level AnchorScore and BridgeScore via two graph algorithms, and fuse them by Synergy Credit Redistribution with an ℓ_2 Norm into the overall SynergyScore. By utilizing the SynergyScore as a fine-grained weight to reweight the advantage of each token, we achieve more precise policy optimization.
- Experiments show that our ABPO demonstrates consistent and significant performance gains, outperforming both the baseline and related methods in different domains like math, medical, and finance. Notably, the ABPO brings even larger improvements as task complexity increases. Moreover, our ABPO framework is able to reduce the response length and improve the training efficiency, highlighting its effectiveness and efficiency.

2 Related Work

2.1 Reinforcement Learning for LLMs

Reinforcement learning is now a standard paradigm for improving the reasoning of large language models (OpenAI, 2025; DeepSeek-AI et al., 2025; Yang et al., 2025a), particularly under the framework of Reinforcement Learning with Verifiable Rewards (RLVR) (Shao et al., 2024; Lambert et al., 2024), where supervision comes from automatically verifiable signals such as correctness in mathematical (Zhang & Team Math-AI, 2024; 2025) or coding tasks (Jimenez et al., 2023; Jain et al., 2024). A representative algorithm is Group Relative Policy Optimization (GRPO) (Shao et al., 2024), which avoids explicitly learning a value model by normalizing rewards across a rollout group, providing a critic-free objective; DAPO (Yu et al., 2025) further mitigates entropy collapse, and Dr. GRPO (Liu et al., 2025) reduces the length bias caused by token-level loss normalization. However, these approaches estimate trajectory-level advantages, applying a uniform signal to all tokens and leaving insufficient guidance among them (Tan et al., 2025; Guo et al., 2025). In this work, we model fine-grained, token-level credit assignment from attention.

2.2 Credit Assignment in Reinforcement Learning

To move beyond coarse trajectory-level rewards, recent works (Lin et al., 2025b; Chan et al., 2024) explore richer signals to shape how credit is assigned within generated responses. One line of work constructs explicit token-level supervision. For example, KTAE (Sun et al., 2025) estimates token

contributions using sampled rollouts, and Outcome-Grounded Advantage Reshaping (Li et al., 2026) estimates outcome sensitivity through token perturbations or gradients, whereas CAPO (Xie et al., 2025) uses an LLM-as-GenPRM to generate step-wise critiques and convert them into token-level credits. Another line of work (Cui et al., 2025; Lin et al., 2025a; Vassoyan et al., 2025; Li et al., 2025) leverages model-internal signals to identify important tokens. Cui et al. (2025) use covariance-guided clipping and KL control against entropy collapse, while Cheng et al. (2026) encourage exploration through entropy. Wang et al. (2025) find that a high-entropy minority of tokens steers reasoning and accounts for much of the benefit of RLVR. Vassoyan et al. (2025) find critical tokens that determine whether reasoning succeeds, and thus reduce the KL constraint around them. Li et al. (2025) identify a recurring preplan-and-anchor rhythm from attention dynamics to reweight advantage. These internal signals are pointwise, scoring each token from its own entropy or attention statistics. In this paper, we instead model the relations among tokens as a graph and compute credit by propagation over it, giving two key token types: Anchor Tokens frequently reused in the reasoning flow, and Bridge Tokens connecting preceding context to subsequent reasoning.

3 METHODOLOGY

We introduce **ABPO**, an Anchor-Bridge Attribution Policy Optimization framework that replaces the uniform token credit of GRPO (preliminaries in Appendix B) with a token-wise multiplier derived from an attention-induced attribution map W. AnchorRank and BridgeTrace score each token on W, and an ℓ_2 synergy fusion combines the two scores into the SynergyScore that reweights token advantages; Algorithm 1 in Appendix C summarizes the full procedure.

3.1 ATTENTION-INDUCED ATTRIBUTION MAP: PART (1) IN FIGURE 2

We first average the attention maps $A^{(l,h)} \in \mathbb{R}^{T\times T}$ over all L layers and H heads:

$$\bar{A} = \frac{1}{LH}\sum_{l=1}^{L}\sum_{h=1}^{H} A^{(l,h)}. \tag{1}$$

In an autoregressive model, $\bar{A}_{ji}$ measures how much token j attends to token i. Since attention is defined from a token to the context it uses, we transpose $\bar{A}$ so that each edge points from a source token to the later tokens that consume it, giving the direction of token-flow attribution:

$$W := \bar{A}^{\top}, \tag{2}$$

where W_{ij} represents the attention-induced attribution from an earlier token i to a later token j. We remove diagonal self-loops and keep only valid autoregressive edges. The resulting matrix retains the aggregated attention weights and serves as a weighted attribution map over response tokens.

3.2 ANCHOR-BRIDGE CREDIT ASSIGNMENT: PART (2) IN FIGURE 2

Anchor Attribution via Recursive Accumulation. (Module ① in Figure 2) On the attention-induced attribution map, AnchorRank computes *AnchorScore* to identify tokens where attribution accumulates through recursive token-flow propagation. This score characterizes tokens whose information is repeatedly reused and reinforced by later reasoning steps, reflecting a propagation-oriented view of token-level credit. AnchorRank propagates credit over the row-stochastic normalization $\mathbf{W}_{\text{norm}}$ of $\mathbf{W}$. Given a damping factor $d \in (0,1)$, set to 0.95 in all experiments, and a uniform restart distribution $\mathbf{v}$ over response tokens, AnchorScore $\mathbf{p} \in \mathbb{R}^T$ is defined by:

$$\left(\mathbf{I} - d\,\mathbf{W}_{\text{norm}}^{\top}\right)\mathbf{p} = (1-d)\mathbf{v}, \qquad \mathbf{p}^{(k+1)} = d\,\mathbf{W}_{\text{norm}}^{\top}\mathbf{p}^{(k)} + (1-d)\mathbf{v}. \tag{3}$$

The left form gives the stationary solution, and the right the corresponding fixed-point iteration. Since edges in $\mathbf{W}$ follow the autoregressive attribution direction, the recursion propagates attribution mass from earlier to later tokens. High AnchorScore therefore marks downstream positions where propagated information is consolidated, rather than tokens that are merely locally attended. In reasoning traces, these positions often correspond to intermediate conclusions, numerical results, or answer-bearing states that are reused by subsequent steps. We refer to them as *Anchor Tokens*.

Bridge Attribution via Balanced Flow Mediation. (Module ② in Figure 2) BridgeTrace computes *BridgeScore* to capture a complementary mediation-oriented view of token-level credit. It focuses on whether a token receives sufficient upstream information and transmits it downstream. Such tokens often serve as transitions, high-level reasoning cues, or premise-setting units that connect adjacent reasoning segments. We compute a path-attribution operator over the attribution map:

$$\mathbf{S} = (\mathbf{I} - \mathbf{W})^{-1}. \tag{4}$$

Defining the inflow and outflow of token i as $\mathrm{In}_i = \sum_j S_{ji}$ and $\mathrm{Out}_i = \sum_j S_{ij}$, we measure its mediation strength using a balanced flow approximation motivated by the betweenness principle:

$$b_i = \sqrt{\mathrm{In}_i \cdot \mathrm{Out}_i}. \tag{5}$$

The geometric mean rewards tokens strong in both directions: they must absorb information from previous tokens and also transmit information to later tokens. BridgeScore thus highlights tokens mediating the transition between reasoning segments. We refer to them as *Bridge Tokens*.

Anchor-Bridge Synergy for Token-Level Credit Redistribution. (Module ③ in Figure 2) Since raw scores cannot serve as advantage multipliers due to sequence-level magnitude variation and noise from low-salience positions, we first map each score vector to a bounded token-wise modulator. Let Φ_{A} and Φ_{B} be the score-to-modulator maps for AnchorScore $\mathbf{p}$ and BridgeScore $\mathbf{b}$ (Appendix C.2). We obtain:

$$M_t^{\mathrm{A}} = \Phi_{\mathrm{A}}(\mathbf{p})_t, \qquad M_t^{\mathrm{B}} = \Phi_{\mathrm{B}}(\mathbf{b})_t. \tag{6}$$

Both multipliers are centered on the unit baseline and take values in a bounded interval around it. Thus, $M_t = 1$ recovers the original GRPO update, while $M_t > 1$ assigns stronger credit to attributionally salient tokens. Each map differences and standardizes its input score within the response, keeps only the salient deviations, and passes them through a bounded monotone link.

ABPO combines AnchorRank and BridgeTrace into a single token-level score, called *SynergyScore*. Instead of averaging raw scores, ABPO fuses the positive gains of their bounded modulators over the unit baseline. Let $g_t^{(\cdot)} = \max(M_t^{(\cdot)} - 1, 0)$ and $s^{(\cdot)} = \max_t g_t^{(\cdot)} + \epsilon$. We then define:

$$q_t = \frac{s^{\mathrm{A}} + s^{\mathrm{B}}}{2} \sqrt{\left(\frac{g_t^{\mathrm{A}}}{s^{\mathrm{A}}}\right)^2 + \left(\frac{g_t^{\mathrm{B}}}{s^{\mathrm{B}}}\right)^2}. \tag{7}$$

The final ABPO multiplier is

$$M_t = 1 + \alpha \cdot q_t, \tag{8}$$

where α controls the modulation strength. This normalized fusion preserves the relative salience within each branch. Anchor and Bridge Tokens selected by either branch receive increased credit, while tokens supported by both receive stronger modulation.

3.3 ABPO Objective

ABPO reweights the group-relative advantage by the token-wise multiplier $M_{i,t}$:

$$\tilde{A}_{i,t} = A_i \, M_{i,t} \mathbb{1}[t \in \mathrm{resp}], \tag{9}$$

where $M_{i,t}$ is $M_{i,t}^{\mathrm{A}}$ for AnchorRank, $M_{i,t}^{\mathrm{B}}$ for BridgeTrace, and Eq. (8) for full ABPO.

We then substitute the reweighted advantage into the standard clipped policy-gradient objective:

$$\mathcal{L}_{\mathrm{ABPO}}(\theta) = -\mathbb{E}\left[\frac{1}{G}\sum_{i=1}^{G}\sum_{t=1}^{T_i} \min\left(\rho_{i,t}\tilde{A}_{i,t}, \mathrm{clip}\left(\rho_{i,t}, 1-\epsilon, 1+\epsilon\right)\tilde{A}_{i,t}\right)\right], \tag{10}$$

where $\rho_{i,t}$ is the token-level importance ratio between the current and behavior policies, and the multipliers are treated as detached coefficients. ABPO thus preserves the standard GRPO backbone while replacing uniform token credit with anchor-bridge redistribution.

Table 1: Intervention results of masking the selected tokens of each branch on MATH500.

Method	**Answer change** (%)		**Correctness reversal** (%)		**Log-prob drop** (nats)	
	Selected	Random	Selected	Random	Selected	Random
AnchorRank	**22**	10	**9**	3	**0.097**	0.050
BridgeTrace	**35**	25	**21**	13	**0.124**	0.102
ABPO	**41**	29	**29**	17	**0.196**	0.135

3.4 Intervention Analysis of Attribution Credit

Since both scores are read out from attention, we verify by intervention that the tokens they amplify are the ones the model relies on. On MATH500 traces, we block the positive-gain support $\{t : M_t > 1\}$ of each branch as attention keys, and compare it against an equal-count random mask over the same prefix. The first intervention regenerates from a shared truncation point and records how often the final answer changes and how often a correct answer is reversed; the second holds the continuation fixed under teacher forcing and measures the drop in mean log-probability. As reported in Table 1, every branch separates from its random control on all three metrics, and the margin is widest for the fused signal. AnchorScore and BridgeScore therefore locate tokens the outcome genuinely depends on, supporting SynergyScore as the credit signal. In a semantic analysis (Appendix E.1), an LLM labels the semantic role of the highest-gain response segments of each branch: AnchorScore mainly amplifies computations and final answers, whereas BridgeScore mainly amplifies subtask entries and step transitions. Full intervention protocols are in Appendix E.2.

4 Experiments

4.1 Experimental Setup

Models and Baselines. We evaluate all methods on two models, Qwen3-4B-Base and Qwen3-8B-Base (Yang et al., 2025a). We include the pre-RL *Base* checkpoint to disentangle RL post-training gains from subsequent credit-assignment gains. For our methods, we evaluate three variants: *AnchorRank* with AnchorScore, *BridgeTrace* with BridgeScore, and the full *ABPO* setting with the ℓ_2-fused SynergyScore using $\alpha = 0.5$. For comparison, *GRPO* (Shao et al., 2024) serves as the standard baseline, and we additionally compare with *DAPO* (Yu et al., 2025), a stronger RLVR recipe that is complementary to token-level credit assignment. We further compare with two token-level control baselines: 1) *Random*, which assigns the positive multiplier increments from ABPO to randomly selected response tokens; 2) *Entropy*, which assigns those increments to the highest-entropy response tokens. Both controls preserve the number and magnitudes of the positive increments while changing their locations. All methods share a maximum generation length of 2048.

Training data and Evaluation. The evaluation of ABPO spans multiple reasoning domains. Specifically, for the math domain, models are trained on DAPO-Math-17K (Yu et al., 2025) using the same configurations, and evaluated on AIME24 (Zhang & Team Math-AI, 2024), AIME25 (Zhang & Team Math-AI, 2025), MATH500 (Hendrycks et al., 2021), and OlympiadBench (He et al., 2024). For other domains, methods are trained and evaluated on subsets of four domain-specific datasets: Medical (MedQA (Jin et al., 2021)), Finance (FinQA (Chen et al., 2021)), Logic (Countdown (Pan et al., 2025)), and General (Nemotron-CrossThink (Akter et al., 2025)). Throughout experiments, Pass@1 serves as the primary metric, supplemented by Pass@16 on other domains and in ablation studies. Finally, to ensure the reliability, **all reported benchmark results are the averages of five independent experimental runs**. More details are provided in Appendix D.

4.2 Main Results

On Mathematical Reasoning. As shown in Table 2, ABPO achieves the best average performance on both Qwen3-4B-Base and Qwen3-8B-Base. Relative to the pre-RL Base, GRPO already improves the average score by +19.3 on the 4B model and +12.5 on the 8B model; ABPO further improves over GRPO by +1.5 and +2.6, respectively. DAPO improves GRPO by +0.6 / +0.8 on 4B / 8B,

Table 2: Results of Pass@1 (%) on mathematical reasoning benchmarks under a shared 2048-token budget. "4B"/"8B" are `Qwen3-4B-Base`/`Qwen3-8B-Base` and Base is the pre-RL checkpoint. Bold and underline mark the best and second-best among trained methods per model.

Method	AIME 2024		AIME 2025		MATH 500		Olympiad Bench		Avg.	
	4B	8B	4B	8B	4B	8B	4B	8B	4B	8B
Base	4.2	12.5	4.4	7.3	35.3	54.8	23.0	31.4	16.7	26.5
GRPO	<u>12.5</u>	14.8	9.9	13.4	75.9	78.5	45.6	49.2	36.0	39.0
DAPO	11.8	15.0	11.3	**15.6**	76.4	79.2	46.7	49.5	<u>36.6</u>	39.8
+ random credit	11.5	14.9	11.3	11.6	76.1	79.1	45.7	49.9	36.2	38.9
+ high-entropy credit	10.8	15.1	9.3	13.5	<u>76.5</u>	78.7	46.8	50.0	35.9	39.3
+ BridgeTrace	11.2	<u>16.2</u>	<u>11.7</u>	14.0	75.3	79.0	<u>46.9</u>	<u>50.1</u>	36.3	39.8
+ AnchorRank	11.5	15.1	7.7	14.5	75.0	<u>80.0</u>	45.1	50.1	34.8	<u>39.9</u>
ABPO	**13.8**	**17.5**	**12.2**	<u>15.5</u>	**76.9**	**81.2**	**47.0**	**52.3**	**37.5**	**41.6**
Δ ABPO vs. GRPO	+1.3	+2.7	+2.3	+2.1	+1.0	+2.7	+1.4	+3.1	+1.5	+2.6

Table 3: Results on `Qwen3-8B-Base` across Medical (MedQA), Finance (FinQA), Logic (Countdown), and General (Nemotron-CrossThink). Bold and underline: best and second-best.

Method	MedQA		FinQA		Countdown		Nemotron-CrossThink	
	Pass@1	Pass@16	Pass@1	Pass@16	Pass@1	Pass@16	Pass@1	Pass@16
GRPO	60.3	79.7	11.9	28.8	61.8	80.6	42.3	<u>83.7</u>
+ BridgeTrace	68.4	**90.5**	<u>22.4</u>	49.6	63.6	77.7	42.9	83.5
+ AnchorRank	**70.0**	<u>90.3</u>	19.6	<u>54.1</u>	**73.6**	**84.0**	<u>44.0</u>	81.7
ABPO	<u>68.5</u>	89.5	**22.9**	**61.4**	<u>67.5</u>	<u>81.7</u>	**45.5**	**84.9**
Δ ABPO vs. GRPO	+8.2	+9.8	+11.0	+32.6	+5.7	+1.1	+3.2	+1.2

while ABPO remains ahead of DAPO by +0.9 / +1.8 on average. On the 4B model, ABPO leads on all four benchmarks; on the 8B model, it leads on three (DAPO is slightly higher on AIME 2025). At the benchmark level, ABPO also achieves consistent improvements over GRPO across the four math datasets, with larger gains on the 8B model. We further compare ABPO with token-level credit-assignment methods. As illustrated in Table 2, random credit and high-entropy credit provide limited or unstable gains, indicating that non-uniform token weighting alone is insufficient. These results suggest that ABPO benefits from explicitly modeling anchor-bridge attribution derived from the models' internal attention. We also supplement the standard deviation result in Appendix E.3.

On Other Domains. Table 3 presents our results on Medical (Jin et al., 2021), Finance (Chen et al., 2021), Countdown (Pan et al., 2025), and General (Akter et al., 2025) reasoning tasks. By training and evaluating on the corresponding datasets for each distinct domain, ABPO demonstrates substantial improvements over GRPO, yielding Pass@1 gains of +8.2 points on MedQA, +11.0 points on FinQA, +5.7 points on Countdown, and +3.2 points on Nemotron-CrossThink. This performance indicates that the anchor-bridge attribution is not uniquely bound to mathematical expressions; rather, it captures reusable structural patterns across diverse reasoning traces. Notably, single-attribution variants also show competitive performance depending on the domain: AnchorRank achieves the highest Pass@1 on Medical and Countdown, whereas full ABPO excels on Finance and General.

4.3 Ablation Study

Effectiveness of Individual Components. As Tables 2 and 3 show, BridgeTrace and AnchorRank each outperform GRPO on most benchmarks, and their fusion (ABPO) is stronger than either alone in Table 2; the remaining ablations therefore examine how the two signals are combined.

Table 4: Ablation study on `Qwen3-8B-Base` of the individual components, the fusion strategy (LogSumExp-, Max-, and the ℓ_2-fusion employed in ABPO), and the redistribution strength α. The last row is the ABPO setting shared by all groups. Bold denotes the highest result in each column.

Method	AIME 2024		AIME 2025		MATH 500		Olympiad Bench	
	Pass@1	Pass@16	Pass@1	Pass@16	Pass@1	Pass@16	Pass@1	Pass@16
GRPO	14.8	33.3	13.4	31.3	78.5	92.0	49.2	74.8
Individual components								
BridgeTrace-Single	16.2	35.3	14.0	24.0	79.0	91.3	50.1	73.7
AnchorRank-Single	15.1	**40.7**	14.5	27.3	80.0	90.0	50.1	72.2
Fusion strategy ($\alpha = 0.5$)								
LogSumExp-Fusion	14.7	30.7	12.4	**32.0**	78.0	91.1	51.0	73.7
Max-Fusion	14.1	34.0	12.5	**32.0**	78.3	90.8	50.2	73.0
Redistribution strength α (ℓ_2-fusion)								
$\alpha = 0.2$	15.2	33.3	14.6	26.7	79.0	90.2	50.0	72.0
$\alpha = 0.8$	15.0	36.7	13.1	30.0	79.3	91.2	50.3	74.1
$\alpha = 1$	13.3	30.0	10.0	20.0	79.3	90.4	50.2	71.1
ABPO (ℓ_2-fusion, $\alpha = 0.5$)	**17.5**	37.3	**15.5**	31.3	**81.2**	**92.7**	**52.3**	**75.8**

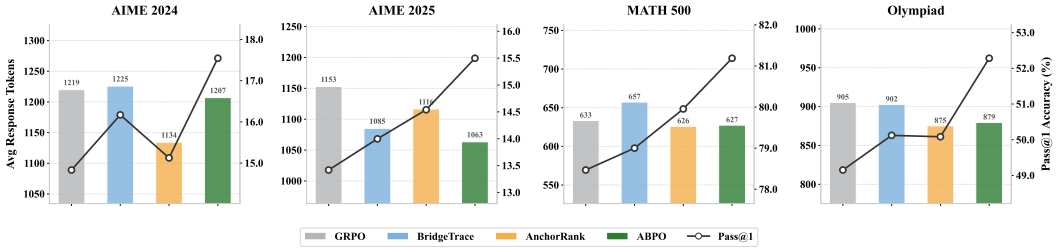


Figure 3: Average generated tokens (left axis) and Pass@1 in % (right axis) on four math benchmarks.

Choice of Anchor-Bridge Fusion Strategy. We compare different fusion strategies for combining AnchorScore and BridgeScore, including LogSumExp fusion, max fusion, and ℓ_2-fusion, which we describe in Appendix C.3. As demonstrated in Table 4, LogSumExp or max fusion is insufficient: LogSumExp Fusion does not consistently outperform GRPO, and Max Fusion remains unstable across benchmarks. Among them, the ℓ_2-fusion achieves the best Pass@1 on all four mathematical benchmarks, while remaining highly competitive on Pass@16. Thus, in the experiment, we employ the ℓ_2-fusion for ABPO and obtain the final SynergyScore.

Effect of Redistribution Strength. The scalar α in Equation 8 sets the magnitude of the credit redistribution, and the last group of Table 4 reports its effect. Its choice involves a trade-off: smaller values do not sufficiently distinguish Anchor and Bridge Tokens from other tokens ($\alpha = 0.2$ improves the Pass@1 of GRPO by only 0.4–1.2 points), while larger values also degrade performance. The best results are obtained around $\alpha = 0.5$.

5 Further Analysis

Response Length of Reasoning. As shown in Figure 3, most of the attribution-based variants (AnchorRank, BridgeTrace, and ABPO) improve the math accuracy of the 8B model without increasing response length, and ABPO achieves the best accuracy on most benchmarks with a favorable trade-off in generated tokens. The advantage is more pronounced on other domains, where ABPO produces shorter responses than GRPO on all four, reducing response length by 30.15% on average and by up to 62.56% on FinQA (Table 7 in Appendix E.4), while improving Pass@1 by 7.0 points on average (Table 3). This indicates that ABPO gains accuracy by encouraging tokens relevant to the final

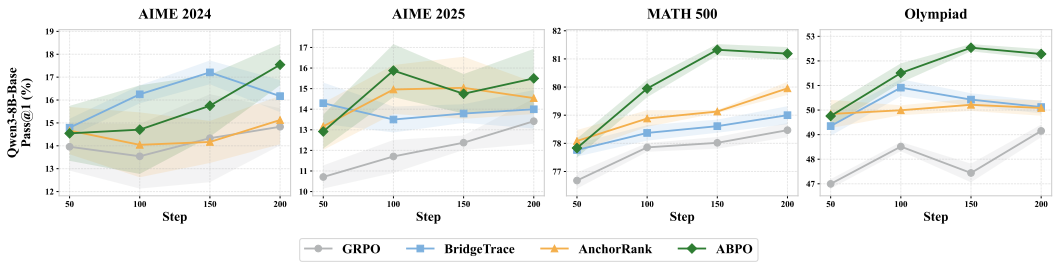


Figure 4: Pass@1 (%) of `Qwen3-8B-Base` over training steps 50–200 on four math benchmarks. Results for `Qwen3-4B-Base` are shown in Appendix E.6.

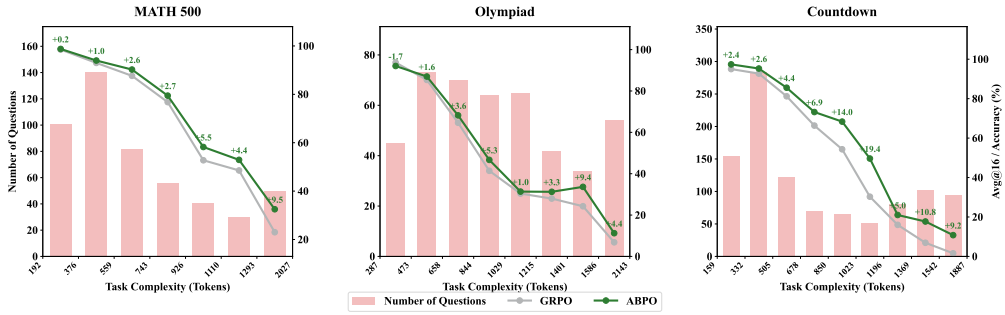


Figure 5: Gains of ABPO over GRPO by problem complexity on MATH500, OlympiadBench, and Countdown. Bars: number of questions (left axis); curves: Avg@16 accuracy in % (right axis).

solution rather than by producing longer responses, and shortens responses by avoiding low-impact continuations rather than by sacrificing accuracy. More case studies are provided in Appendix E.5.

Training Efficiency under ABPO. We further analyze training efficiency by tracking Pass@1 accuracy from step 50 to step 200. As shown in Figure 4, ABPO reaches higher accuracy earlier and maintains stronger performance throughout training, indicating faster and more efficient convergence; its average Pass@1 at step 100 already exceeds that of GRPO at step 200 (40.5 vs. 39.0). BridgeTrace and AnchorRank also show better and faster learning trends than GRPO on most datasets, but neither single variant matches the fused ABPO. This advantage also holds in wall-clock time despite the 14.6% higher per-step cost of ABPO (attribution takes 7.5% of the step): on the 20-step moving average of the training reward, ABPO reaches 0.15, approximately the peak of GRPO, within 52 steps and 5.5 net hours versus 145 steps and 14.0 hours for GRPO, a 2.56× speedup, and further reaches 0.20, which GRPO never attains (Appendix E.6).

Performance Gains in Different Task Complexity. We quantify task complexity as the sum of the input token length and the ground-truth response token length. As shown in Figure 5, ABPO yields consistently more pronounced accuracy improvements on longer, more complex problems across the MATH 500, Olympiad, and Countdown benchmarks. This trend suggests that our internal attribution mechanism is particularly effective on complex, multi-step, and extended reasoning chains.

6 Conclusion

This paper introduces Anchor-Bridge Attribution Policy Optimization (ABPO) to refine token-level credit assignment for reasoning language models. By leveraging internal attention patterns, the framework identifies Anchor and Bridge tokens via the AnchorRank and BridgeTrace algorithms, and fuses their scores into a unified SynergyScore for precise advantage reweighting. Experiments demonstrate that ABPO consistently outperforms other baselines across diverse reasoning domain tasks. Furthermore, the approach generates notably more concise reasoning chains and accelerates overall training convergence, highlighting the effectiveness of aligning reward signals with internal structural attribution to enhance complex reasoning capabilities.

AI USE STATEMENT

In this work, we used generative AI tools to assist in implementing the proposed methods. We have not used generative AI tools for developing theoretical models or conceptual frameworks, proposing or refining hypotheses, designing research methodology or experiments, cleaning or reformatting datasets, interpreting results, or translation, and the remaining required disclosure tasks (generating synthetic datasets and formulating or proving mathematical claims) are not applicable to this work. Additionally, we used generative AI tools to write auxiliary scripts for data processing, analysis, and plotting; to assign semantic-role labels to response segments in the semantic analysis of Appendix E.1; to adjust the layout of existing figures; to edit the manuscript for readability and conciseness; to assist with LaTeX formatting; and to check the text against our experimental records for consistency. We have reviewed all AI-assisted work: the authors confirmed that the figure layout changes preserved the underlying data and checked every AI-suggested revision of the manuscript for technical accuracy. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

All models and datasets used in our experiments are publicly available. Section 4.1 describes the backbones, baselines, datasets, and evaluation metrics, and all benchmark results are averaged over five independent runs. Section 3 presents the complete formulation of ABPO, and Appendix C complements it with the pseudocode, the mapping from attribution scores to token-wise multipliers, and the fusion variants; Appendix B reviews the GRPO formulation on which ABPO builds. Appendix D specifies the training framework, hardware, hyperparameters, prompt template, ABPO configuration, evaluation protocol, and checkpoint selection. Appendix E.3 reports the standard deviations of the main results, Appendix E.2 details the setup and masking protocol of the intervention experiments, and Appendix E.7 provides the training logs. We will release our code upon acceptance.

REFERENCES

Syeda Nahida Akter, Shrimai Prabhumoye, Matvei Novikov, Seungju Han, Ying Lin, Evelina Bakhturina, Eric Nyberg, Yejin Choi, Mostofa Patwary, Mohammad Shoeybi, and Bryan Catanzaro. Nemotron-CrossThink: Scaling self-learning beyond math reasoning. arXiv preprint arXiv:2504.13941, 2025. URL https://arxiv.org/abs/2504.13941.

Alex J Chan, Hao Sun, Samuel Holt, and Mihaela Van Der Schaar. Dense reward for free in reinforcement learning from human feedback. *arXiv preprint arXiv:2402.00782*, 2024. URL https://arxiv.org/abs/2402.00782.

Zhiyu Chen, Wenhu Chen, Charese Smiley, Sameena Shah, Iana Borova, Dylan Langdon, Reema Moussa, Matt Beane, Ting-Hao Huang, Bryan Routledge, and William Yang Wang. FinQA: A dataset of numerical reasoning over financial data. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 3697–3711. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.emnlp-main.300. URL https://aclanthology.org/2021.emnlp-main.300/.

Daixuan Cheng, Shaohan Huang, Xuekai Zhu, Bo Dai, Xin Zhao, Zhenliang Zhang, and Furu Wei. Reasoning with exploration: An entropy perspective. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(36):30377–30385, March 2026. doi: 10.1609/aaai.v40i36.40290. URL https://ojs.aaai.org/index.php/AAAI/article/view/40290.

Ganqu Cui, Yuchen Zhang, Jiacheng Chen, Lifan Yuan, Zhi Wang, Yuxin Zuo, Haozhan Li, Yuchen Fan, Huayu Chen, Weize Chen, et al. The entropy mechanism of reinforcement learning for reasoning language models. *arXiv preprint arXiv:2505.22617*, 2025. URL https://arxiv.org/abs/2505.22617.

DeepSeek-AI, 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, 2025. URL https://arxiv.org/abs/2501.12948.

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*, 2025. URL https://arxiv.org/abs/2505.23564.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. OlympiadBench: A challenging benchmark for promoting AGI with olympiad-level bilingual multimodal scientific problems. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3828–3850, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.211. URL https://aclanthology.org/2024.acl-long.211/.

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. URL https://arxiv.org/abs/2103.03874.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. LiveCodeBench: Holistic and contamination free evaluation of large language models for code. *arXiv preprint arXiv:2403.07974*, 2024. URL https://arxiv.org/abs/2403.07974.

Carlos E Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. SWE-bench: Can language models resolve real-world GitHub issues? *arXiv preprint arXiv:2310.06770*, 2023. URL https://arxiv.org/abs/2310.06770.

Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *Applied Sciences*, 11(14):6421, 2021. doi: 10.3390/app11146421. URL https://www.mdpi.com/2076-3417/11/14/6421.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, et al. Tulu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*, 2024. URL https://arxiv.org/abs/2411.15124.

Yang Li, Zhichen Dong, Yuhan Sun, Weixun Wang, Shaopan Xiong, Yijia Luo, Jiashun Liu, Han Lu, Jiamang Wang, Wenbo Su, et al. Attention illuminates LLM reasoning: The preplan-and-anchor rhythm enables fine-grained policy optimization. *arXiv preprint arXiv:2510.13554*, 2025. URL https://arxiv.org/abs/2510.13554.

Ziheng Li, Liu Kang, Feng Xiao, Luxi Xing, Qingyi Si, Zhuoran Li, Weikang Gong, Deqing Yang, Yanghua Xiao, and Hongcheng Guo. Outcome-grounded advantage reshaping for fine-grained credit assignment in mathematical reasoning. *arXiv preprint arXiv:2601.07408*, 2026. URL https://arxiv.org/abs/2601.07408.

Xingyu Lin, Yilin Wen, En Wang, Du Su, Wenbin Liu, Chenfu Bao, and Zhonghou Lv. Token-level policy optimization: Linking group-level rewards to token-level aggregation via markov likelihood. *arXiv preprint arXiv:2510.09369*, 2025a. URL https://arxiv.org/abs/2510.09369.

Zicheng Lin, Tian Liang, Jiahao Xu, Qiuzhi Liu, Xing Wang, Ruilin Luo, Chufan Shi, Siheng Li, Yujiu Yang, and Zhaopeng Tu. Critical tokens matter: Token-level contrastive estimation enhances LLM's reasoning capability. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 37906–37918. PMLR, 2025b. URL https://proceedings.mlr.press/v267/lin25j.html.

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. URL https://arxiv.org/abs/2503.20783.

OpenAI. gpt-oss-120b & gpt-oss-20b Model Card. arXiv preprint arXiv:2508.10925, 2025. URL https://arxiv.org/abs/2508.10925.

Jiayi Pan, Junjie Zhang, Xingyao Wang, Lifan Yuan, Hao Peng, and Alane Suhr. TinyZero. GitHub repository, 2025. URL https://github.com/Jiayi-Pan/TinyZero. Accessed: 2025-01-24.

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. arXiv preprint arXiv:2402.03300, 2024. URL https://arxiv.org/abs/2402.03300.

Wei Sun, Wen Yang, Pu Jian, Qianlong Du, Fuwei Cui, Shuo Ren, and Jiajun Zhang. KTAE: A model-free algorithm to key-tokens advantage estimation in mathematical reasoning. *arXiv preprint arXiv:2505.16826*, 2025. URL https://arxiv.org/abs/2505.16826.

Hongze Tan, Zihan Wang, Jianfei Pan, Jinghao Lin, Hao Wang, Yifan Wu, Tao Chen, Zhihang Zheng, Zhihao Tang, and Haihua Yang. GTPO and GRPO-S: Token and sequence-level reward shaping with policy entropy. *arXiv preprint arXiv:2508.04349*, 2025. URL https://arxiv.org/abs/2508.04349.

Jean Vassoyan, Nathanaël Beau, and Roman Plaud. Ignore the KL penalty! boosting exploration on critical tokens to enhance RL fine-tuning. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 6123–6133. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.findings-naacl.340. URL https://aclanthology.org/2025.findings-naacl.340/.

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

Guofu Xie, Yunsheng Shi, Hongtao Tian, Ting Yao, and Xiao Zhang. CAPO: Towards enhancing LLM reasoning through generative credit assignment. *arXiv preprint arXiv:2508.02298*, 2025. URL https://arxiv.org/abs/2508.02298.

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

Zhihe Yang, Xufang Luo, Zilong Wang, Dongqi Han, Zhiyuan He, Dongsheng Li, and Yunjian Xu. Do not let low-probability tokens over-dominate in RL for LLMs. *arXiv preprint arXiv:2505.12929*, 2025b. URL https://arxiv.org/abs/2505.12929.

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. URL https://arxiv.org/abs/2503.14476.

Yifan Zhang and Team Math-AI. American invitational mathematics examination (AIME) 2024. Hugging Face dataset, 2024. URL https://huggingface.co/datasets/math-ai/aime24. Third-party machine-readable edition of MAA competition problems; accessed 2026-09-18.

Yifan Zhang and Team Math-AI. American invitational mathematics examination (AIME) 2025. Hugging Face dataset, 2025. URL https://huggingface.co/datasets/math-ai/aime25. Third-party machine-readable edition of MAA competition problems; accessed 2026-09-18.

APPENDIX

A LIMITATION AND IMPACT

Limitations. While the proposed ABPO framework demonstrates strong effectiveness across various domains, calculating the attention-induced attribution map and the corresponding token-level scores introduces additional computational overhead during the training phase compared to standard uniform credit assignment methods.

Broader Impacts. The proposed framework may offer a helpful reference for improving the efficiency of reasoning language models. By potentially reducing redundant token generation, this approach might contribute to lowering inference costs and resource consumption in practical applications. While the current findings are primarily exploratory, they could provide a useful perspective for future research aimed at aligning model optimization with internal structural dynamics.

B PRELIMINARIES

B.1 AUTOREGRESSIVE POLICY OPTIMIZATION

We consider reinforcement learning for autoregressive language generation. Given a prompt x, a policy model π_θ generates a response $y = (y_1, \ldots, y_T)$ token by token:

$$\pi_\theta(y \mid x) = \prod_{t=1}^{T} \pi_\theta(y_t \mid x, y_{<t}), \tag{11}$$

where $y_{<t}$ denotes the previously generated prefix. A reward function assigns a scalar outcome-level reward $r(x, y) \in \mathbb{R}$ to the completed response. In reasoning-oriented reinforcement learning, such reward is usually obtained from a verifier or a rule-based answer checker and is only available after the full response has been generated.

For a sampled prompt-response sequence, we distinguish between prompt tokens and generated response tokens. Let $\mathbb{1}[t \in \text{resp}]$ denote the response mask, which equals 1 if token position t belongs to the generated response and 0 otherwise. This distinction is important because prompt tokens serve as conditioning context, while policy-gradient updates are applied only to generated tokens. Therefore, any token-level credit assignment mechanism should only modulate valid response positions.

B.2 GROUP-RELATIVE ADVANTAGE AND TOKEN-LEVEL CREDIT

We adopt Group Relative Policy Optimization (GRPO) as the base reinforcement learning framework. For each prompt x, GRPO samples a group of G responses from the old policy:

$$\{y^{(1)}, \ldots, y^{(G)}\} \sim \pi_{\theta_{\text{old}}}(\cdot \mid x). \tag{12}$$

Each response is evaluated by the reward function, and the rewards within the same group are normalized to obtain a group-relative advantage:

$$A_i = \frac{r(x, y^{(i)}) - \mu_r}{\sigma_r + \epsilon}, \tag{13}$$

where

$$\mu_r = \frac{1}{G} \sum_{j=1}^{G} r(x, y^{(j)}), \qquad \sigma_r = \sqrt{\frac{1}{G} \sum_{j=1}^{G} \big(r(x, y^{(j)}) - \mu_r\big)^2}. \tag{14}$$

This critic-free formulation avoids learning an explicit value model and provides a stable response-level learning signal for large-scale reasoning post-training.

In standard GRPO, the response-level advantage A_i is uniformly broadcast to all valid generated tokens:

$$A_{i,t}^{\text{GRPO}} = A_i \cdot \mathbb{1}[t \in \text{resp}]. \tag{15}$$

This design is simple, but it implicitly assumes that all tokens in the same response contribute equally to the final reward. Such an assumption is often too coarse for long-form reasoning. A reasoning trace usually contains tokens with heterogeneous functional roles: some tokens encode intermediate results that are repeatedly reused by later steps, while others serve as transitions that connect different reasoning segments.

This motivates a structure-aware view of token-level credit assignment. Rather than changing the group-level reward or introducing an additional critic, we keep the group-relative advantage A_i unchanged and redistribute it across response tokens through a token-wise multiplier $M_{i,t}$:

$$\widetilde{A}_{i,t} = A_i \, M_{i,t} \mathbb{1}[t \in \text{resp}]. \tag{16}$$

The central question is how to obtain $M_{i,t}$ in a way that reflects the model's internal reasoning structure. The next section addresses this question by constructing an attention-induced attribution map and deriving token-wise multipliers from AnchorScore, BridgeScore, and their fused SynergyScore.

C Algorithm

C.1 ABPO Algorithm

As shown in Algorithm 1, we present the ABPO algorithm in detail.

Algorithm 1 Anchor-Bridge Attribution Policy Optimization (ABPO)

Require: Policy π_θ; behavior policy $\pi_{\theta_{\text{old}}}$; prompt x; group-relative advantages $\{A_i\}_{i=1}^{G}$; redistribution strength α.
Ensure: Updated policy parameters θ.

for $i = 1, \ldots, G$ **do**
  Sample response $y_i \sim \pi_{\theta_{\text{old}}}(\cdot \mid x)$ of length T_i and aggregate attention into the attribution map W ▷ Eq. (1)–(2)
  Stage 1: Attribution scores.
  $\mathbf{p}_i \leftarrow \textsc{AnchorRank}(W)$ ▷ AnchorScore, Eq. (3)
  $\mathbf{b}_i \leftarrow \textsc{BridgeTrace}(W)$ ▷ BridgeScore, Eq. (5)
  Stage 2: Score-to-modulator mapping.
  $M^{\text{A}}_{i,t} \leftarrow \Phi_{\text{A}}(\mathbf{p}_i)_t$
  $M^{\text{B}}_{i,t} \leftarrow \Phi_{\text{B}}(\mathbf{b}_i)_t$ ▷ Eq. (6)
  Stage 3: ℓ_2 synergy fusion (SynergyScore).
  $g^{\text{A}}_{i,t} \leftarrow \max(M^{\text{A}}_{i,t} - 1,\, 0), \quad g^{\text{B}}_{i,t} \leftarrow \max(M^{\text{B}}_{i,t} - 1,\, 0)$
  $s^{\text{A}}_i \leftarrow \max_t g^{\text{A}}_{i,t} + \epsilon_\circ, \quad s^{\text{B}}_i \leftarrow \max_t g^{\text{B}}_{i,t} + \epsilon_\circ$ ▷ $\epsilon_\circ$: small numerical floor
  $q_{i,t} \leftarrow \dfrac{s^{\text{A}}_i + s^{\text{B}}_i}{2} \sqrt{\left(\dfrac{g^{\text{A}}_{i,t}}{s^{\text{A}}_i}\right)^2 + \left(\dfrac{g^{\text{B}}_{i,t}}{s^{\text{B}}_i}\right)^2}, \quad \forall t$ ▷ Eq. (7)
  $M_{i,t} \leftarrow 1 + \alpha \cdot q_{i,t}, \quad \forall t$ ▷ Eq. (8)
  Stage 4: Advantage redistribution.
  $\tilde{A}_{i,t} \leftarrow A_i \, M_{i,t} \mathbb{1}[t \in \text{resp}], \quad \forall t$ ▷ Eq. (9)
end for
Stage 5: Clipped policy update.
Detach $\{M_{i,t}\}$ from the computation graph; compute token-level importance ratio $\rho_{i,t}$.
$\mathcal{L}_{\text{ABPO}}(\theta) \leftarrow -\mathbb{E}\left[\dfrac{1}{G}\sum_{i=1}^{G}\sum_{t=1}^{T_i} \min\Big(\rho_{i,t}\tilde{A}_{i,t},\ \text{clip}(\rho_{i,t}, 1-\epsilon, 1+\epsilon)\,\tilde{A}_{i,t}\Big)\right]$ ▷ Eq. (10)
$\theta \leftarrow \theta - \nabla_\theta \mathcal{L}_{\text{ABPO}}(\theta)$
return θ

C.2 From Attribution Scores to Token-Wise Multipliers

The two maps Φ_{A} and Φ_{B} in Eq. (6) follow the same design. Each map first differences and standardizes the raw score within the response, so that the resulting multipliers depend on the shape

of the attribution profile rather than on the overall attention magnitude of a particular sequence, which varies considerably across responses and lengths. It then keeps only the salient deviations of the standardized score and leaves the remaining positions at the unit baseline, so that tokens without a clear attribution signal retain the original GRPO update. Finally, the retained deviations pass through a bounded monotone link, which keeps every multiplier inside a fixed interval around one and prevents a few positions from dominating the redistribution within a response.

C.3 Fusion Variants

The three fusion strategies compared in Table 4 share the same two preparatory steps. From the two branch multipliers they first extract the positive gains over the unit baseline, $g_t^{(\cdot)} = \max(M_t^{(\cdot)} - 1, 0)$, so that only positions promoted by a branch can contribute, and then rescale each branch by its own peak gain, which places the two branches on a comparable scale before they are combined. They differ only in how the two rescaled gains are combined at each position. Max fusion takes their element-wise maximum, so a token is promoted according to its stronger branch alone and the weaker branch is discarded. LogSumExp fusion takes a Boltzmann-smoothed maximum, which interpolates between the maximum and the sum. The ℓ_2 fusion adopted by ABPO takes their Euclidean norm, as in Eq. (7), which preserves the ordering within each branch while giving tokens supported by both branches a strictly larger gain than tokens supported by only one. In all three cases the combined gain is restored to the average of the two branch peaks and mapped to a multiplier by Eq. (8).

D Experimental Details

Training details. All experiments are implemented within the PPO-based training pipeline of the OpenRLHF framework. We utilize `Qwen3-4B-Base` and `Qwen3-8B-Base` as the policy backbones. Distributed training is managed via Ray, utilizing 8 NVIDIA A100 GPUs for both the 4B and 8B models. To maximize memory efficiency, the actor and reference models are colocated on the same devices. Following the Group Relative Policy Optimization (GRPO) paradigm, we omit the value model (critic) and strictly set both the initial KL divergence penalty and the entropy regularization coefficients to zero.

For the rollout phase, we sample $G = 8$ responses per prompt with a sampling temperature of $T = 1.0$. The maximum sequence lengths are constrained to 1024 tokens for prompts and 2048 tokens for generated responses. The policy is optimized for a single epoch using a learning rate of 1×10^{-6}. We maintain a global training batch size and a rollout batch size of 64, alongside a micro-batch size of 1 for both rollout and training stages. All main results are reported at training step 200.

In our attribution-induced methods, the token-level advantages are reweighted according to the AnchorScore, BridgeScore, or their ℓ_2-fused SynergyScore, with the final advantage multiplier strictly clipped to the range of $[0, 2]$. These shaping signals are explicitly detached from the gradient computation graph, applied exclusively to valid response tokens, and subsequently normalized using group-wise RMS normalization.

To further accelerate training and reduce memory overhead, our system implementation integrates DeepSpeed ZeRO Stage 2, Adam optimizer state offloading, gradient checkpointing, and `bf16` mixed precision. We additionally adopt eager attention, Liger kernels, and prefix caching to optimize the forward and backward passes.

Training data and prompting. We train on the DAPO-style math supervision set `dapo_math_17k_train` and use `dapo_math_17k_val_200` for validation during training. Each prompt is formatted with the instruction `Please reason step by step, and put your final answer within \boxed{}.` followed by the question text. Group-normalized advantage estimation is used throughout training.

ABPO configuration. For all ABPO runs, both AnchorRank and BridgeTrace advantage reweighting methods are enabled. The attribution map is constructed from token-level attention, and the damping factor in Eq. (3) is set to $d = 0.95$.

Evaluation protocol. We evaluate the mathematical reasoning capabilities of our models on four standard benchmarks: MATH500, AIME 2024, AIME 2025, and OlympiadBench. All evaluations are conducted in a zero-shot setting by appending the following instruction to each question: `Please reason step by step, and put your final answer within \boxed{}.` Inference is accelerated using vLLM with tensor parallelism across 8 GPUs. For each problem, we sample 16 candidate responses with a sampling temperature of 0.7, a top-p of 0.95, and a maximum generation length of 2048 tokens.

To systematically assess correctness, we implement a robust rule-based parsing pipeline to extract the predicted answers enclosed within the `\boxed{}` environment. The extracted predictions are subsequently compared against the ground-truth answers using a mathematical equivalence verification module. This evaluation protocol ensures that structurally distinct but mathematically identical expressions are accurately graded.

Checkpoint selection. For the main evaluation, we report the results of the checkpoint at training step 200 for all methods and benchmarks. Intermediate checkpoints saved every 50 training steps are used to analyze step-wise convergence and training stability, rather than to select benchmark-specific best checkpoints.

E Supplementary Results

E.1 Semantic Analysis of Attribution Credit

This section examines which reasoning roles receive the credit amplified by each branch, complementing the intervention analysis in Section 3.4.

Data. We use the responses to 200 problems sampled from MATH500 and analyze the token-wise multipliers M_t that AnchorRank, BridgeTrace, and ABPO compute for them, i.e., the multipliers applied to the token advantages during training.

Procedure. For each token, we measure the credit added by a method with the positive gain $g_t = \max(M_t - 1, 0)$. Each response is split by fixed rules into contiguous segments of adjacent tokens at line breaks, sentence and equation boundaries, and step markers such as "first", "then", and "step", so that a segment typically covers one clause or one equation. The gain of a segment is the sum of g_t over its tokens. For each method, we select the 12 segments with the largest gain across the 200 responses. An LLM then reads each selected segment in the context of its response and assigns it one of seven semantic roles: problem/subtask entry, step transition, rule/identity, computation, verification, final answer, and filler. Table 5 reports, for each method, the share of each role in the total gain of its selected segments.

Table 5: Share (%) of each semantic role in the total gain of the 12 highest-gain segments of each method. No selected segment is labeled as filler.

Method	Problem/ subtask entry	Step transition	Rule/ identity	Computation	Verification	Final answer
AnchorRank	0.0	0.0	15.4	64.3	0.0	20.3
BridgeTrace	25.4	42.4	13.2	19.1	0.0	0.0
ABPO	9.1	16.0	13.7	45.4	7.7	8.2

Results. AnchorRank places 84.6% of its gain on computations and final answers, that is, on intermediate and final results. BridgeTrace places 67.8% on problem/subtask entries and step transitions, that is, on positions where the solution enters a new stage. ABPO covers both: 53.6% of its gain falls on computations and final answers and 25.1% on entries and transitions, and verification appears only among its selected segments. This matches the intended roles of Anchor Tokens, which are reused by subsequent reasoning, and Bridge Tokens, which connect preceding context with later reasoning (Section 3). Since the analysis covers only the 12 highest-gain segments of each method, it describes where the largest credit is placed rather than the credit distribution over all tokens.

E.2 Details of the Intervention Experiments

This section details the two masked interventions reported in Section 3.4 and Table 1.

Setup. We use Qwen3-4B-Base and select 100 MATH500 problems whose stored responses carry a parseable answer and a length between 96 and 768 tokens; 87 of these responses are correct, and their average length is 444.9 tokens. For each response we draw a truncation point uniformly within 45%–70% of its length, requiring it to lie before the final answer and to leave at least one positively modulated token in the prefix of every branch. The resulting prefixes average 269.3 tokens, i.e. 59.8% of the response. Within a sample, all conditions share the prompt, the stored response, the truncation point and the per-step sampling quantiles, so any difference is attributable to masking.

Intervention target. We intervene on the final multiplier M_t that is applied to the token advantage during training, taking its positive-gain support $\mathcal{S}^+ = \{t : M_t > 1\}$ as the token set of each branch. This set matches the credit that training actually redistributes.

Masking mechanism. Selected tokens stay in the input sequence but are excluded as attention keys, which prevents subsequent tokens from reading the information aggregated at these positions. Absolute `position_ids` are preserved, so blocked information is not confounded with a global positional shift.

Regeneration intervention. The model continues from the truncation point with temperature 0.7 and top-p 0.95. We report the answer change rate and the correctness reversal rate against a paired unmasked continuation produced at the same truncation point under the same random quantiles, which removes the variation caused by truncation and resampling alone. The answer change and correctness reversal columns of Table 1 report this intervention, where every branch separates clearly from its random control and the separation is largest for the fused signal.

Teacher-forced intervention. We fix the original continuation and recompute its token log-probabilities after masking the selected tokens as attention keys. We measure the effect of masking by the decrease in mean log-probability relative to the unmasked condition. For each branch, we compare this decrease with that obtained by masking an equal number of randomly selected prefix tokens. The measured drops agree with the regeneration results: masking the selected tokens makes the model markedly less able to reproduce its own reasoning trace, while an equal number of random tokens costs much less.

E.3 Standard Deviation Table

As shown in Table 6, we report the standard deviation of results (Table 2 in the main text) on mathematical reasoning benchmarks.

Table 6: Standard deviation of results on mathematical reasoning benchmarks. "4B"/"8B" are `Qwen3-4B-Base`/`Qwen3-8B-Base`. Bold and underline mark the lowest and second-lowest standard deviation (most stable) per model.

Method	AIME 2024		AIME 2025		MATH 500		Olympiad Bench		Avg.	
	4B	8B	4B	8B	4B	8B	4B	8B	4B	8B
GRPO	1.4	<u>0.8</u>	1.1	1.1	**0.2**	0.3	**0.3**	<u>0.3</u>	0.7	**0.6**
+ BridgeTrace	<u>1.0</u>	**0.7**	<u>0.8</u>	<u>0.9</u>	**0.2**	0.3	0.4	<u>0.3</u>	<u>0.6</u>	**0.6**
+ AnchorRank	**0.9**	1.1	**0.5**	**0.8**	**0.2**	**0.2**	0.4	<u>0.3</u>	**0.5**	**0.6**
ABPO	1.1	0.9	0.9	1.4	0.3	**0.2**	**0.3**	**0.2**	<u>0.6</u>	0.7

E.4 Response Length on Other Domains

Table 7 reports the average response length of `Qwen3-8B-Base` on the four domains of Table 3, where each question is sampled 16 times.

Table 7: Average response length (tokens) of `Qwen3-8B-Base` on Medical (MedQA), Finance (FinQA), Logic (Countdown), and General (Nemotron-CrossThink). The last row reports the relative change of ABPO over GRPO, and Avg. is its mean over the four domains. Bold: shorter response.

Method	MedQA	FinQA	Countdown	Nemotron-CrossThink	Avg.
GRPO	247.0	918.9	814.0	496.9	–
ABPO	**184.2**	**344.1**	**786.2**	**351.7**	–
Δ ABPO vs. GRPO	−25.43%	−62.56%	−3.42%	−29.21%	−30.15%

E.5 Case Study

To qualitatively illustrate the behavioral difference between GRPO and ABPO, we compare their outputs on three problems from MATH500. Both responses are reproduced verbatim, and the original `\boxed{}` is rendered as an actual framed box.

Inside each ABPO response we replay the stored token trace and recover the two branch multipliers $M_t^{(\mathrm{A})}$ and $M_t^{(\mathrm{B})}$ separately, then color every token that a branch promotes above the unit baseline, widened to the enclosing phrase or display equation so that the highlighting still reads as text. Red marks Anchor Tokens, that is, the values a later step recomputes or reuses, and green marks Bridge Tokens, which hand one stage of the derivation to the next. Across the three traces the two supports are almost disjoint, so the colors also show how little the branches duplicate each other.

Case Study 1: Piecewise Inverse on MATH

Prompt. Please reason step by step, and put your final answer within `\boxed{}`.
Question: Let f be defined by

$$f(x) = \begin{cases} 3 - x & \text{if } x \le 3, \\ -x^3 + 2x^2 + 3x & \text{if } x > 3. \end{cases}$$

Calculate $f^{-1}(0) + f^{-1}(6)$.
Answer:

Response of ABPO. *(red: Anchor Tokens; green: Bridge Tokens)* To calculate $f^{-1}(0) + f^{-1}(6)$, we need to find the values of x such that $f(x) = 0$ and $f(x) = 6$, respectively. The function f is defined piecewise, so we'll consider each piece separately.
Step 1: Find $f^{-1}(0)$
We need to solve $f(x) = 0$.
Case 1: $x \le 3$

$$f(x) = 3 - x = 0 \implies x = 3$$

Case 2: $x > 3$

$$f(x) = -x^3 + 2x^2 + 3x = 0$$

Factor the equation:

$$-x^3 + 2x^2 + 3x = 0 \implies x(-x^2 + 2x + 3) = 0$$

This gives us $x = 0$ or $-x^2 + 2x + 3 = 0$. Since $x > 3$, $x = 0$ is not in the domain for this case. Solving $-x^2 + 2x + 3 = 0$:

$$x^2 - 2x - 3 = 0 \implies (x-3)(x+1) = 0 \implies x = 3 \text{ or } x = -1$$

Since $x > 3$, there is no solution in this interval.
Thus, the only solution is $x = 3$.

$$f^{-1}(0) = 3$$

Step 2: Find $f^{-1}(6)$
We need to solve $f(x) = 6$.
Case 1: $x \leq 3$

$$f(x) = 3 - x = 6 \implies x = -3$$

Case 2: $x > 3$

$$f(x) = -x^3 + 2x^2 + 3x = 6$$

Rearrange the equation:

$$-x^3 + 2x^2 + 3x - 6 = 0 \implies x^3 - 2x^2 - 3x + 6 = 0$$

We look for rational roots using the Rational Root Theorem. Testing $x = 2$:

$$2^3 - 2(2)^2 - 3(2) + 6 = 8 - 8 - 6 + 6 = 0$$

So, $x = 2$ is a root. However, $x = 2$ does not satisfy $x > 3$. Testing $x = 3$:

$$3^3 - 2(3)^2 - 3(3) + 6 = 27 - 18 - 9 + 6 = 6 \neq 0$$

Testing $x = 4$:

$$4^3 - 2(4)^2 - 3(4) + 6 = 64 - 32 - 12 + 6 = 26 \neq 0$$

Since $x = 2$ is not in the domain $x > 3$ and there are no other obvious roots, we consider $x = -3$ from the first case.
Thus, the only solution is $x = -3$.

$$f^{-1}(6) = -3$$

Step 3: Calculate $f^{-1}(0) + f^{-1}(6)$

$$f^{-1}(0) + f^{-1}(6) = 3 + (-3) = 0$$

$$\boxed{0}$$

Response of GRPO: To solve for $f^{-1}(0) + f^{-1}(6)$, we need to determine the values of x for which $f(x) = 0$ and $f(x) = 6$, and then sum these values. We will consider each piece of the piecewise function separately.
Step 1: Find $f^{-1}(0)$
We need to solve $f(x) = 0$.
Case 1: $x \leq 3$

$$f(x) = 3 - x$$

Set $3 - x = 0$:

$$3 - x = 0$$
$$x = 3$$

So, $f^{-1}(0) = 3$ if $x \leq 3$.
Case 2: $x > 3$

$$f(x) = -x^3 + 2x^2 + 3x$$

Set $-x^3 + 2x^2 + 3x = 0$:

$$-x(x^2 - 2x - 3) = 0$$

$$-x(x-3)(x+1)=0$$

The solutions are $x=0$, $x=3$, and $x=-1$. Since $x>3$, the only relevant solution is:

$$x=3$$

However, $x=3$ is not greater than 3, so there is no solution in this case.
Thus, the only solution is $x=3$, so:

$$f^{-1}(0)=3$$

Step 2: Find $f^{-1}(6)$
We need to solve $f(x)=6$.
Case 1: $x\leq 3$

$$f(x)=3-x$$

Set $3-x=6$:

$$3-x=6$$
$$-x=3$$
$$x=-3$$

So, $f^{-1}(6)=-3$ if $x\leq 3$.
Case 2: $x>3$

$$f(x)=-x^3+2x^2+3x$$

Set $-x^3+2x^2+3x=6$:

$$-x^3+2x^2+3x-6=0$$

We can use the Rational Root Theorem to test possible rational roots. The possible rational roots are $\pm 1,\pm 2,\pm 3,\pm 6$.
Testing $x=2$:

$$-2^3+2(2^2)+3(2)-6=-8+8+6-6=0$$

So, $x=2$ is a root. However, $x=2$ is not greater than 3, so it is not a valid solution.
Testing $x=3$:

$$-3^3+2(3^2)+3(3)-6=-27+18+9-6=-6\neq 0$$

So, $x=3$ is not a root.
Testing $x=4$:

$$-4^3+2(4^2)+3(4)-6=-64+32+12-6=-26\neq 0$$

So, $x=4$ is not a root.
Testing $x=5$:

$$-5^3+2(5^2)+3(5)-6=-125+50+15-6=-66\neq 0$$

So, $x=5$ is not a root.
Testing $x=6$:

$$-6^3+2(6^2)+3(6)-6=-216+72+18-6=-132\neq 0$$

So, $x=6$ is not a root.
Since none of the rational roots work and the polynomial is continuous, we can use numerical methods or graphing to find the root. However, we can see that $x=4$ is close to a root. Let's use the Intermediate Value Theorem to confirm there is a root between 3 and 4.
Evaluating at $x=3$:

$$-3^3+2(3^2)+3(3)-6=-27+18+9-6=-6$$

Evaluating at $x=4$:

$$-4^3+2(4^2)+3(4)-6=-64+32+12-6=-26$$

Evaluating at $x=5$:

$$-5^3+2(5^2)+3(5)-6=-125+50+15-6=-66$$

Evaluating at $x = 6$:

$$-6^3 + 2(6^2) + 3(6) - 6 = -216 + 72 + 18 - 6 = -132$$

We can see that the function changes sign between 3 and 4, so there is a root in this interval.
Using numerical methods, we find that the root is approximately $x \approx 3.5$.
So, $f^{-1}(6) \approx 3.5$.
Step 3: Sum the values

$$f^{-1}(0) + f^{-1}(6) = 3 + 3.5 = 6.5$$

Thus, the final answer is:

$$\boxed{6.5}$$

Discussion. ABPO arrives at the correct answer 0 via a compact case analysis: each branch is resolved in a single line, rejected candidates are explicitly tied to the domain constraint, and the Case 1 solution $x = -3$ is carried through unambiguously. The highlighting makes the division of labor between the two branches visible. BridgeScore selects the tokens that set up each subproblem—naming the two equations to solve, factoring out $-x^2 + 2x + 3$, and discharging the root $x = 0$ as outside the domain—while AnchorScore selects the tokens that close the search and carry $x = -3$ into the final sum, peaking on the boxed answer. GRPO begins along the same lines and correctly identifies $x = -3$ as a valid solution under $x \leq 3$, but then abandons this solution while searching the $x > 3$ branch. Worse, GRPO misreads the signs of $g(x) = -x^3 + 2x^2 + 3x - 6$ on $[3, 6]$—all evaluations are in fact negative, so no sign change occurs—invokes the Intermediate Value Theorem incorrectly, and fabricates a root near 3.5 that does not exist, propagating to a wrong final answer of 6.5. The contrast highlights two failure modes that ABPO mitigates: (i) premature abandonment of a valid branch solution in favor of speculative numerical search, and (ii) unchecked hallucination of intermediate facts during long reasoning chains. ABPO's advantage shaping encourages the policy to commit to verified sub-conclusions and to terminate when all cases are exhausted, whereas GRPO's uniform advantage signal allows speculative continuation that overwrites correct partial results. This pattern is consistent with our aggregate findings: on piecewise or case-split problems, GRPO traces are longer yet exhibit higher rates of domain-constraint violation and spurious numerical fallback, while ABPO traces remain shorter and more faithful to the algebraic structure of the problem.

Case Study 2: Triangle Area on MATH

Prompt. Please reason step by step, and put your final answer within `\boxed{}`.
Question: In triangle ABC, $AB = 17$, $AC = 8$, and $BC = 15$. Let D be the foot of the altitude from C to AB. Find the area of triangle ACD.
Answer:

Response of ABPO. *(red: Anchor Tokens; green: Bridge Tokens)* To find the area of triangle ACD, we will follow these steps:
Step 1: Verify that triangle ABC is a right triangle
We are given the sides of triangle ABC:

- $AB = 17$
- $AC = 8$
- $BC = 15$

To check if triangle ABC is a right triangle, we can use the Pythagorean theorem:

$$AB^2 = AC^2 + BC^2$$

$$17^2 = 8^2 + 15^2$$

$$289 = 64 + 225$$

$$289 = 289$$

Since the equation holds true, triangle ABC is a right triangle with the right angle at C.

Step 2: Find the area of triangle ABC
Since triangle ABC is a right triangle with legs AC and BC, the area is given by:

$$\text{Area}_{ABC} = \frac{1}{2} \times AC \times BC$$

$$\text{Area}_{ABC} = \frac{1}{2} \times 8 \times 15$$

$$\text{Area}_{ABC} = 60$$

Step 3: Find the length of the altitude from C to AB
The altitude from C to AB is the same as the height of the triangle when AB is the base. The area of the triangle can also be expressed as:

$$\text{Area}_{ABC} = \frac{1}{2} \times AB \times \text{height}$$

We already know the area is 60 and $AB = 17$, so we can solve for the height:

$$60 = \frac{1}{2} \times 17 \times \text{height}$$

$$60 = 8.5 \times \text{height}$$

$$\text{height} = \frac{60}{8.5} = \frac{120}{17}$$

Thus, the altitude from C to AB is $\frac{120}{17}$.
Step 4: Find the length of AD
Since D is the foot of the altitude from C to AB, we can use the Pythagorean theorem in triangle ACD:

$$AC^2 = AD^2 + \left(\frac{120}{17}\right)^2$$

$$8^2 = AD^2 + \left(\frac{120}{17}\right)^2$$

$$64 = AD^2 + \frac{14400}{289}$$

$$AD^2 = 64 - \frac{14400}{289}$$

$$AD^2 = \frac{18496}{289} - \frac{14400}{289}$$

$$AD^2 = \frac{4096}{289}$$

$$AD = \frac{64}{17}$$

Step 5: Find the area of triangle ACD
The area of triangle ACD can be found using the base AD and the height $\frac{120}{17}$:

$$\text{Area}_{ACD} = \frac{1}{2} \times AD \times \text{height}$$

$$\text{Area}_{ACD} = \frac{1}{2} \times \frac{64}{17} \times \frac{120}{17}$$

$$\text{Area}_{ACD} = \frac{1}{2} \times \frac{7680}{289}$$

$$\text{Area}_{ACD} = \frac{3840}{289}$$

Final Answer

$$\boxed{\frac{3840}{289}}$$

Response of Base GRPO. To find the area of triangle ACD, we first need to determine the area of triangle ABC and then use the properties of similar triangles and the altitude to find the area of triangle ACD.
First, we check if triangle ABC is a right triangle. We use the Pythagorean theorem to verify this:

$$AB^2 = 17^2 = 289$$
$$AC^2 + BC^2 = 8^2 + 15^2 = 64 + 225 = 289$$

Since $AB^2 = AC^2 + BC^2$, triangle ABC is a right triangle with $\angle ACB = 90^\circ$.
The area of triangle ABC can be calculated as:

$$\text{Area} = \frac{1}{2} \times AC \times BC = \frac{1}{2} \times 8 \times 15 = 60$$

Next, we find the length of the altitude CD from C to AB. Since $\angle ACB = 90^\circ$, the area of triangle ABC can also be expressed as:

$$\text{Area} = \frac{1}{2} \times AB \times CD$$

We already know the area is 60, so:

$$60 = \frac{1}{2} \times 17 \times CD$$

Solving for CD:

$$120 = 17 \times CD \implies CD = \frac{120}{17}$$

Now, we need to find the area of triangle ACD. Since CD is the altitude from C to AB, it splits AB into two segments, AD and BD. Let $AD = x$ and $BD = 17 - x$. Using the fact that the area of triangle ABC is 60, we can express the area of triangle ACD and triangle BCD as follows:

$$\text{Area of } \triangle ACD = \frac{1}{2} \times AD \times CD = \frac{1}{2} \times x \times \frac{120}{17}$$
$$\text{Area of } \triangle BCD = \frac{1}{2} \times BD \times CD = \frac{1}{2} \times (17 - x) \times \frac{120}{17}$$

Since the total area of triangle ABC is 60, we have:

$$\text{Area of } \triangle ACD + \text{Area of } \triangle BCD = 60$$
$$\frac{1}{2} \times x \times \frac{120}{17} + \frac{1}{2} \times (17 - x) \times \frac{120}{17} = 60$$
$$\frac{120}{34} (x + (17 - x)) = 60$$
$$\frac{120}{34} \times 17 = 60$$
$$\frac{120 \times 17}{34} = 60$$
$$\frac{120 \times 17}{2 \times 17} = 60$$
$$60 = 60$$

This confirms our setup is correct. Now, we need to find the area of triangle ACD. Since $\triangle ACD$ and $\triangle BCD$ share the same height CD and their bases add up to the length of AB, we can use the ratio of the bases to find the area of $\triangle ACD$.

The area of $\triangle ACD$ is:

$$\text{Area of } \triangle ACD = \frac{AC^2}{AB} \times \text{Area of } \triangle ABC = \frac{8^2}{17^2} \times 60 = \frac{64}{289} \times 60 = \frac{3840}{289} = 48$$

However, a simpler way to find the area of $\triangle ACD$ is to use the ratio of the segments AD and BD. Since $\triangle ACD$ and $\triangle BCD$ are similar to $\triangle ABC$ and share the same height, the area of $\triangle ACD$ is:

$$\text{Area of } \triangle ACD = \frac{AC^2}{AB} \times \text{Area of } \triangle ABC = \frac{8^2}{17^2} \times 60 = \frac{64}{289} \times 60 = \frac{3840}{289} = 24$$

Thus, the area of triangle ACD is:

$$\boxed{24}$$

Discussion. This case highlights how ABPO benefits from attribution-guided token-level credit assignment in geometry reasoning, where preserving intermediate quantities is crucial. The ABPO response correctly derives the altitude CD and the leg AD of $\triangle ACD$:

$$CD = \frac{120}{17}, \quad AD = \frac{64}{17}.$$

The two branches credit different parts of this derivation. BridgeScore marks the tokens that commit the solution to a plan, namely checking the right angle through the Pythagorean theorem, evaluating $[\triangle ABC]$ from the two legs, and converting that area into the altitude from C, each of which licenses the step after it. AnchorScore instead marks the closing chain in which both derived quantities are consumed together,

$$[\triangle ACD] = \frac{1}{2} \cdot \frac{64}{17} \cdot \frac{120}{17} = \frac{3840}{289},$$

with the credit peaking on the reused fractions and on the boxed answer. In contrast, because GRPO assigns sequence-level credit uniformly, it provides limited supervision for maintaining local numerical consistency. Consequently, while the GRPO response identifies some correct global facts, it hallucinates late-stage arithmetic, erroneously simplifying $\frac{3840}{289}$ to conflicting values like 48 and 24. By explicitly upweighting tokens that serve as anchors and bridges, ABPO preserves key geometric dependencies, effectively preventing such late-stage arithmetic shortcuts.

Case Study 3: Sequence on MATH

Prompt. Please reason step by step, and put your final answer within `\boxed{}`.
Question: A sequence (a_n) is defined as follows:

$$a_{i+1} = \frac{1}{1 - a_i}$$

for $i \geq 1$. If $a_3 = a_1$, compute $(a_9)^9$. Answer:

Response of ABPO. *(red: Anchor Tokens; green: Bridge Tokens)* To solve the problem, we start by analyzing the given recurrence relation:

$$a_{i+1} = \frac{1}{1 - a_i}$$

We are given that $a_3 = a_1$ and need to compute $(a_9)^9$.
Step 1: Compute the first few terms
Let's compute the first few terms of the sequence to identify any patterns or cycles.
Given a_1, we can compute:

$$a_2 = \frac{1}{1 - a_1}$$

$$a_3 = \frac{1}{1-a_2} = \frac{1}{1-\frac{1}{1-a_1}} = \frac{1-a_1}{1-a_1-1} = \frac{1-a_1}{-a_1} = \frac{a_1-1}{a_1}$$

$$a_4 = \frac{1}{1-a_3} = \frac{1}{1-\frac{a_1-1}{a_1}} = \frac{1}{\frac{a_1-(a_1-1)}{a_1}} = \frac{1}{\frac{1}{a_1}} = a_1$$

Step 2: Identify the cycle
From the computations above, we see that:

$$a_4 = a_1$$

This suggests that the sequence is periodic with a cycle of length 3:

$$a_1, a_2, a_3, a_1, a_2, a_3, \ldots$$

Step 3: Determine a_9
Since the sequence repeats every 3 terms, we can find a_9 by determining its position within the cycle:

$$9 \div 3 = 3 \text{ with a remainder of } 0$$

This means a_9 corresponds to the 3rd term in the cycle, which is a_3.
Given that $a_3 = a_1$, we have:

$$a_9 = a_3 = a_1$$

Step 4: Compute $(a_9)^9$
Since $a_9 = a_1$, we have:

$$(a_9)^9 = (a_1)^9$$

However, we need to determine the value of a_1. From the cycle, we know that:

$$a_3 = a_1 = \frac{a_1 - 1}{a_1}$$

Solving for a_1:

$$a_1 = \frac{a_1 - 1}{a_1}$$

Multiply both sides by a_1:

$$a_1^2 = a_1 - 1$$

Rearrange the equation:

$$a_1^2 - a_1 + 1 = 0$$

This is a quadratic equation. Solving for a_1:

$$a_1 = \frac{1 \pm \sqrt{1-4}}{2} = \frac{1 \pm \sqrt{-3}}{2} = \frac{1 \pm i\sqrt{3}}{2}$$

Thus, a_1 is a complex number. However, since the problem asks for $(a_9)^9$, and $a_9 = a_1$, we have:

$$(a_9)^9 = (a_1)^9$$

Given the complex nature of a_1, we can express $(a_1)^9$ in terms of its magnitude and argument. However, since the problem likely expects a real number answer, we reconsider the initial assumption.
Step 5: Re-evaluate the cycle
Given that $a_3 = a_1$, and the sequence is periodic with a cycle of 3, we can express a_1 in terms of itself:

$$a_1 = \frac{a_1 - 1}{a_1}$$

Solving this equation:

$$a_1^2 = a_1 - 1$$

$$a_1^2 - a_1 + 1 = 0$$

The solutions to this equation are complex, but we can proceed by assuming a_1 is a real number. Let's consider the possibility that $a_1 = 1$:

$$a_2 = \frac{1}{1-1} = \text{undefined}$$

This is not valid. Therefore, a_1 must be a complex number.
Step 6: Compute $(a_1)^9$
Given that a_1 is a complex number, we can express it in polar form:

$$a_1 = \frac{1 \pm i\sqrt{3}}{2}$$

The magnitude of a_1 is:

$$|a_1| = \sqrt{\left(\frac{1}{2}\right)^2 + \left(\frac{\sqrt{3}}{2}\right)^2} = \sqrt{\frac{1}{4} + \frac{3}{4}} = \sqrt{1} = 1$$

The argument of a_1 is:

$$\theta = \arctan\left(\frac{\sqrt{3}/2}{1/2}\right) = \arctan(\sqrt{3}) = \frac{\pi}{3}$$

Thus, a_1 can be written as:

$$a_1 = e^{i\pi/3}$$

Raising a_1 to the 9th power:

$$(a_1)^9 = \left(e^{i\pi/3}\right)^9 = e^{i3\pi} = \cos(3\pi) + i\sin(3\pi) = -1 + i \cdot 0 = -1$$

Final Answer

$$\boxed{-1}$$

Response of GRPO. To solve the problem, we start by examining the given recurrence relation for the sequence (a_n):

$$a_{i+1} = \frac{1}{1 - a_i}.$$

We are given that $a_3 = a_1$. We need to find the value of $(a_9)^9$.
First, let's express a_2 and a_3 in terms of a_1:

$$a_2 = \frac{1}{1 - a_1},$$

$$a_3 = \frac{1}{1 - a_2} = \frac{1}{1 - \frac{1}{1-a_1}} = \frac{1 - a_1}{1 - (1 - a_1)} = \frac{1 - a_1}{a_1} = \frac{1}{a_1} - 1.$$

Since $a_3 = a_1$, we have:

$$a_1 = \frac{1}{a_1} - 1.$$

Multiplying both sides by a_1 gives:

$$a_1^2 = 1 - a_1,$$
$$a_1^2 + a_1 - 1 = 0.$$

This is a quadratic equation in a_1. Solving for a_1 using the quadratic formula $a_1 = \frac{-b \pm \sqrt{b^2-4ac}}{2a}$, where $a = 1$, $b = 1$, and $c = -1$, we get:

$$a_1 = \frac{-1 \pm \sqrt{1+4}}{2} = \frac{-1 \pm \sqrt{5}}{2}.$$

Thus, the solutions are:

$$a_1 = \frac{-1+\sqrt{5}}{2} \quad \text{or} \quad a_1 = \frac{-1-\sqrt{5}}{2}.$$

Let's denote $a_1 = \frac{-1+\sqrt{5}}{2}$. The other solution $a_1 = \frac{-1-\sqrt{5}}{2}$ will give the same sequence due to the symmetry of the recurrence relation.

Now, we need to find the general form of the sequence. Let's compute the next few terms:

$$a_2 = \frac{1}{1-a_1} = \frac{1}{1-\frac{-1+\sqrt{5}}{2}} = \frac{1}{\frac{3-\sqrt{5}}{2}} = \frac{2}{3-\sqrt{5}} \cdot \frac{3+\sqrt{5}}{3+\sqrt{5}}$$
$$= \frac{2(3+\sqrt{5})}{9-5} = \frac{2(3+\sqrt{5})}{4} = \frac{3+\sqrt{5}}{2} = \frac{1+\sqrt{5}}{2}.$$
$$a_3 = \frac{1}{1-a_2} = \frac{1}{1-\frac{1+\sqrt{5}}{2}} = \frac{1}{\frac{1-\sqrt{5}}{2}} = \frac{2}{1-\sqrt{5}} \cdot \frac{1+\sqrt{5}}{1+\sqrt{5}}$$
$$= \frac{2(1+\sqrt{5})}{1-5} = \frac{2(1+\sqrt{5})}{-4} = \frac{1+\sqrt{5}}{-2} = \frac{-1-\sqrt{5}}{2}.$$

Since $a_3 = a_1$, we have a repeating pattern every 3 terms. Therefore, the sequence is periodic with period 3:

$$a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, \ldots = a_1, a_2, a_3, a_1, a_2, a_3, a_1, a_2, a_3, \ldots.$$

Thus, $a_9 = a_3 = a_1 = \frac{-1+\sqrt{5}}{2}$.

Now, we need to compute $(a_9)^9$:

$$a_9 = \frac{-1+\sqrt{5}}{2},$$
$$(a_9)^9 = \left(\frac{-1+\sqrt{5}}{2}\right)^9.$$

Since a_1, a_2, a_3 are the same as a_4, a_5, a_6, and so on, and the sequence repeats every 3 terms, we have:

$$a_9 = a_3 = a_1 = \frac{-1+\sqrt{5}}{2}.$$

We need to find $(a_9)^9$. Notice that:

$$(a_9)^9 = \left(\frac{-1+\sqrt{5}}{2}\right)^9 = \left(\frac{-1+\sqrt{5}}{2}\right) \cdot \left(\frac{-1+\sqrt{5}}{2}\right)^8 = \left(\frac{-1+\sqrt{5}}{2}\right) \cdot \left(\frac{1-\sqrt{5}}{2}\right)^8.$$

Since $\left(\frac{-1+\sqrt{5}}{2}\right) \cdot \left(\frac{-1-\sqrt{5}}{2}\right) = 1$, we have:

$$\left(\frac{-1+\sqrt{5}}{2}\right)^9 = \left(\frac{-1+\sqrt{5}}{2}\right) \cdot \left(\frac{1-\sqrt{5}}{2}\right)^8$$
$$= \left(\frac{-1+\sqrt{5}}{2}\right) \cdot \left(\frac{1-\sqrt{5}}{2}\right)^8$$
$$= \left(\frac{-1+\sqrt{5}}{2}\right) \cdot \left(\frac{1-\sqrt{5}}{2}\right)^8 = 1.$$

Thus, $(a_9)^9 = 1$.
Therefore, the final answer is:

$$\boxed{1}$$

Discussion. This case illustrates how ABPO improves mathematical reasoning by assigning higher credit to tokens that play structurally important roles in the solution trajectory. In this example, the crucial reasoning step is the symbolic derivation of a_3 from the recurrence:

$$a_3 = \frac{a_1 - 1}{a_1}.$$

ABPO correctly preserves this transformation and uses the condition $a_3 = a_1$ to reach

$$a_1^2 - a_1 + 1 = 0.$$

The recovered multipliers place this equation, together with the phrase that names it, on the BridgeTrace side rather than the AnchorRank side. BridgeTrace covers the tokens that open and close a stage of the argument: the initial plan, the decision to expand the first few terms and look for a cycle, and the reuse of that cycle to locate a_9. Such tokens carry little numerical content on their own, yet each hands the state of the derivation to the next stage. AnchorRank concentrates instead on the late polar-form computation, where the modulus $\sqrt{1} = 1$, the argument $\arctan(\sqrt{3}) = \pi/3$, the representation $a_1 = e^{i\pi/3}$ and the expansion of $e^{i3\pi}$ are each consumed by the step that follows, ending at

$$(a_9)^9 = (a_1)^9 = -1.$$

The two supports barely overlap, so the fused signal reaches the transitions and the reused values at the same time. In contrast, the GRPO response makes an early algebraic sign error when deriving a_3, which leads to an incorrect quadratic equation and ultimately an incorrect final answer. Since GRPO applies sequence-level credit uniformly, it cannot explicitly distinguish such pivotal algebraic transitions from less important explanatory tokens. ABPO's token-level attribution mechanism instead emphasizes the bridge between local symbolic manipulation and global solution consistency, making it more robust on problems where a small intermediate error can propagate to the final answer.

E.6 Training Efficiency

Setup and metrics. All `Qwen3-8B-Base` runs use the configuration in Appendix D with the same random seed, so every method receives identical prompts at each step and the runs differ only in credit assignment. The training reward of a response is the sum of an accuracy term ($+1.0$ if correct, -0.5 if incorrect, and -0.8 if no `\boxed{}` answer is given) and a format term ($+0.2$ for exactly one `\boxed{}`, -0.2 for none, and $-0.2 - 0.1(k-1)$ for $k > 1$); it is averaged over the 512 responses of each step and is therefore negative early in training. We smooth the per-step reward with a trailing 20-step moving average, defined from step 20, and take the first step at which the smoothed reward attains a target as the steps needed to reach it. The net wall-clock time accumulates the time of each step from step 1, excluding validation (2–3 minutes every 50 steps) and checkpoint saving but including all attribution overhead. The speedup of ABPO is the ratio of GRPO to ABPO in steps or in net hours.

Time to reach a target reward. Table 8 reports the steps and net wall-clock time that each method needs to reach each target. The smoothed reward of GRPO peaks at 0.157 at step 148 and stays within 0.09–0.16 after step 100, so the target 0.15 approximates its best level. The wall-clock speedup of ABPO increases with the target: at -0.20, all methods need a similar number of steps and the higher per-step cost of ABPO yields a speedup below one, whereas at higher targets GRPO stagnates while the smoothed reward of ABPO rises from 0.18 at step 100 to 0.26 at step 200.

Table 8: Steps and net wall-clock hours (in parentheses) needed by `Qwen3-8B-Base` to reach each target training reward (20-step moving average). The last two columns give the speedup of ABPO over GRPO. –: the target is not reached during training (220 steps for GRPO).

Target	Steps (net wall-clock hours)				ABPO speedup	
	GRPO	AnchorRank	BridgeTrace	ABPO	Steps	Wall-clock
−0.20	27 (2.5 h)	24 (2.5 h)	24 (2.5 h)	25 (2.6 h)	1.08×	0.94×
−0.10	41 (3.8 h)	30 (3.1 h)	30 (3.1 h)	31 (3.2 h)	1.32×	1.16×
0.00	57 (5.3 h)	36 (3.7 h)	37 (3.8 h)	37 (3.9 h)	1.54×	1.36×
0.05	66 (6.2 h)	40 (4.1 h)	41 (4.2 h)	40 (4.2 h)	1.65×	1.47×
0.10	102 (9.7 h)	45 (4.7 h)	49 (5.1 h)	44 (4.6 h)	2.32×	2.10×
0.15	145 (14.0 h)	61 (6.4 h)	55 (5.7 h)	52 (5.5 h)	2.79×	2.56×
0.20	–	71 (7.4 h)	68 (7.1 h)	69 (7.3 h)	–	–

Per-step cost. Table 9 decomposes the wall-clock time of a training step. The attribution pipeline of ABPO takes 30.6 s, or 7.5% of its 406.8 s step, and most of the step is still spent on the policy update (72%). The policy update of ABPO takes about 19 s longer than that of GRPO, but this difference does not stem from attribution: the two runs were executed on different dates and nodes, and ABPO generates slightly longer responses (1020 vs. 995 tokens). On `Qwen3-4B-Base`, the policy update of ABPO is not slower than that of GRPO (169.8 s vs. 172.8 s), and its total overhead of 6.8% is no larger than its attribution time (21.9 s, 8.6% of the step).

Table 9: Mean per-step wall-clock time (s) by stage for `Qwen3-8B-Base`, excluding step 1 and the step after each validation (207 steps for ABPO and 215 for the other methods). The attribution pipeline is profiled into its two components only for ABPO.

Stage	GRPO	AnchorRank	BridgeTrace	ABPO
Rollout generation	55.9	54.0	56.2	56.6
Standard experience construction	24.2	24.7	25.0	26.1
Attribution pipeline	–	28.0	28.7	30.6
Attention extraction and aggregation	–	–	–	17.9
AnchorScore, BridgeScore, and fusion	–	–	–	12.7
Policy update and synchronization	274.9	285.8	283.5	293.5
Total per step	355.0	392.5	393.4	406.8
Total vs. GRPO	–	+10.6%	+10.8%	+14.6%
Attribution share of step	–	7.1%	7.3%	7.5%

Robustness to smoothing. Table 10 recomputes the wall-clock speedup of ABPO under other smoothing methods. At target 0.10, the speedup ranges from 1.90× to 2.37×; at target 0.15, it is 2.81× with a 10-step moving average, whereas under the 30-step moving average and the 20-step EMA GRPO never reaches this level.

Table 10: Wall-clock speedup of ABPO over GRPO on `Qwen3-8B-Base` to reach each target training reward under different smoothing methods. EMA: exponential moving average. –: GRPO does not reach the target during training, whereas ABPO does.

Smoothing	Target training reward			
	0.00	0.05	0.10	0.15
10-step moving average	1.46×	1.47×	2.37×	2.81×
20-step moving average (default)	1.36×	1.47×	2.10×	2.56×
30-step moving average	1.29×	1.70×	1.90×	–
20-step EMA	1.40×	1.96×	2.09×	–

Test accuracy of Qwen3-4B-Base. Figure 6 reports the Pass@1 of `Qwen3-4B-Base` over training steps 50–200 on the four math benchmarks, complementing the `Qwen3-8B-Base` results in Figure 4.

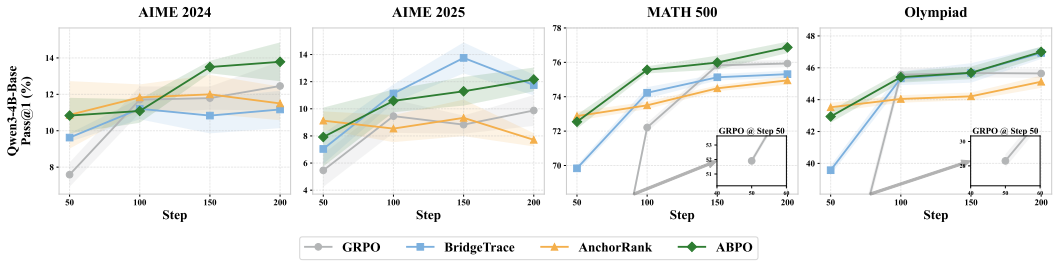
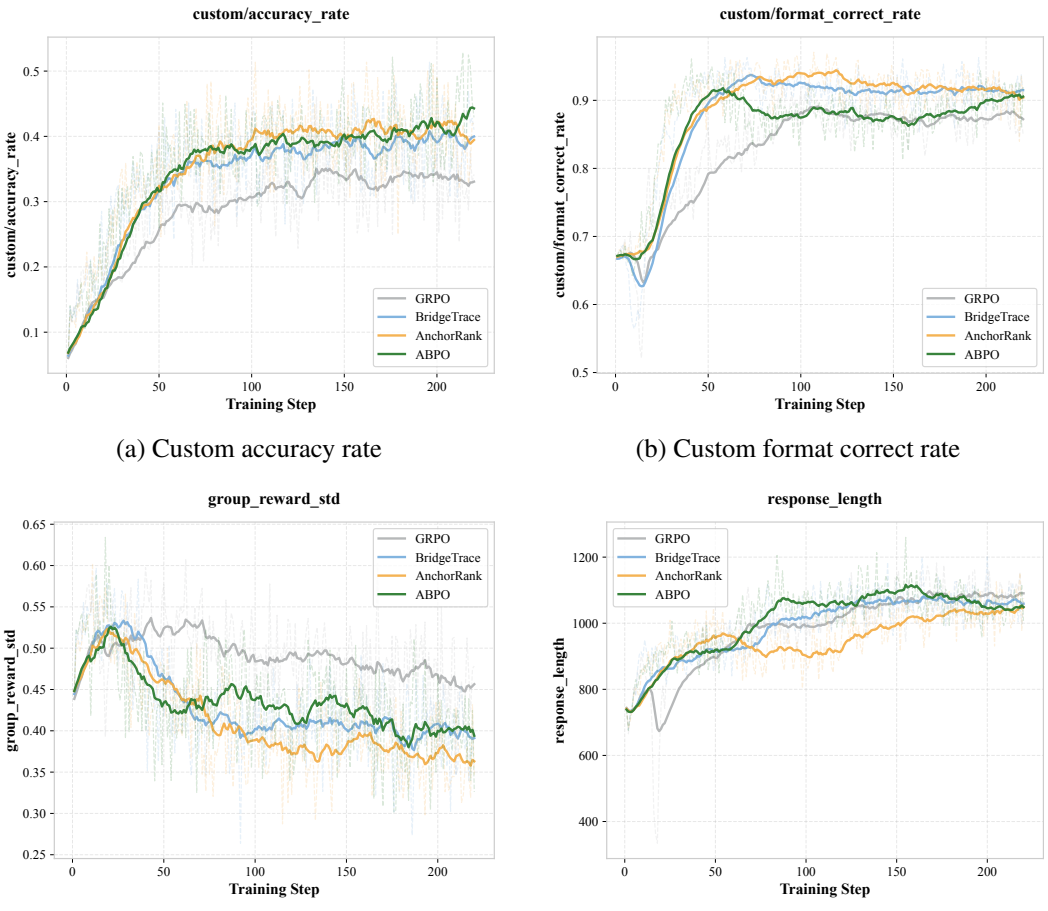


Figure 6: Pass@1 (%) of `Qwen3-4B-Base` over training steps 50–200 on four math benchmarks.

E.7 Train Log



(a) Custom accuracy rate

(b) Custom format correct rate

(c) Group reward standard deviation

(d) Response length

Figure 7: Training curves of accuracy, format correctness, group reward standard deviation, and response length.