Outcome-Grounded Advantage Reshaping for Fine-Grained Credit Assignment in Mathematical Reasoning

Anonymous ACL submission

Abstract

Group Relative Policy Optimization (GRPO) has emerged as a promising critic-free reinforcement learning paradigm for reasoning tasks. However, standard GRPO employs a coarse-grained credit assignment mechanism that propagates group-level rewards uniformly to to every token in a sequence, neglecting the varying contribution of individual reasoning steps. We address this limitation by introducing **O**utcome-grounded **A**dvantage **R**eshaping (**OAR**), a fine-grained credit assignment mechanism that redistributes advantages based on how much each token influences the model's final answer. We instantiate OAR via two complementary strategies: (1) **OAR-P**, which estimates outcome sensitivity through counterfactual token perturbations, serving as a high-fidelity attribution signal; (2) **OAR-G**, which uses an input-gradient sensitivity proxy to approximate the influence signal with a single backward pass. These importance signals are integrated with a conservative **Bi-Level** advantage reshaping scheme that suppresses low-impact tokens and boosts pivotal ones while preserving the overall advantage mass. Empirical results on extensive mathematical reasoning benchmarks demonstrate that while OAR-P sets the performance upper bound, OAR-G achieves comparable gains with negligible computational overhead, both significantly outperforming a strong GRPO baseline, pushing the boundaries of critic-free LLM reasoning [1].

1 Introduction

Large Language Models (LLMs) have recently demonstrated strong performance on complex reasoning tasks (Fedus et al., 2022; Achiam et al., 2023; Brown et al., 2020), driven in part by reinforcement learning with verifiable rewards (RLVR; Lambert et al., 2024). Among these approaches, Group Relative Policy Optimization (GRPO; Shao et al., 2024) has emerged as a practical critic-free algorithm for post-training reasoning models. By normalizing rewards over a group of sampled responses, GRPO avoids training a value function while remaining stable and scalable for long-horizon reasoning.

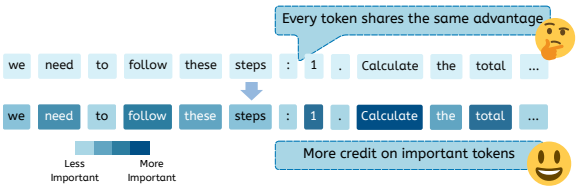


Figure 1: From broadcast credit to token reallocation.

However, GRPO relies on a coarse credit assignment: a single group-normalized advantage is broadcast to all tokens in a sampled response (Wei et al., 2023). This ignores the uneven structure of reasoning traces, where only a small set of tokens (e.g., key deductions or structural decisions) determines correctness, while many others are largely syntactic (Yang et al., 2025b). Figure 1 illustrates this mismatch between coarse sequence-level credit assignment and the desired token-level reallocation. Treating all positions equally increases gradient variance and can accelerate entropy collapse during extended training (Liu et al., 2024; Li et al., 2024).

A natural direction is to move from coarse sequence-level credit to token-level credit. Recent work uses token entropy as a proxy to reweight updates: Cheng et al. (2025) amplifies high-entropy tokens to encourage exploration, while Chen et al. (2025) shapes advantages over low-entropy segments in a correctness-aware way to consolidate useful structures and suppress recurring failure patterns. However, entropy primarily reflects uncertainty or stability(Gal and Ghahramani, 2016), not outcome-relevant importance, and may therefore misallocate credit to tokens that are salient to the

[1]We provide the anonymous code: https://anonymous.4open.science/r/OAR-592B

policy yet only weakly influential on the final answer outcome(Jain and Wallace, 2019).

In contrast, we argue that the ideal token-level credit in critic-free RL should be outcome-grounded, reflecting how much each token influences the model's final answer. To this end, we propose **O**utcome-grounded **A**dvantage **R**eshaping (**OAR**), a framework that enhances GRPO with outcome-sensitive token attribution. OAR scores each token by how much perturbing it shifts the model's final-answer distribution, inspired by perturbation-based feature attribution(Ribeiro et al., 2016). While the ideal attribution would measure reward changes under interventions, RLVR rewards are typically rule-based, sparse, and non-differentiable, making token–reward attribution prohibitively expensive. We therefore use the model's own answer distribution as a practical surrogate outcome to estimate token impact.

We instantiate OAR with two complementary attribution strategies that trade off fidelity and efficiency: **OAR-P** estimates outcome sensitivity via counterfactual token perturbations, providing a high-fidelity but costly signal, while **OAR-G** uses an efficient gradient-based proxy suitable for scalable online training(Simonyan et al., 2014). However, attribution scores alone do not specify *how* to inject token-level credit without destabilizing GRPO: naively multiplying or adding token weights can induce sample-dependent changes in the effective update scale.

Therefore, we introduce a conservative **Bi-Level** advantage reallocation mechanism that suppresses low-impact tokens to reduce gradient noise, boosts high-impact tokens to strengthen the learning signal, and renormalizes weights to preserve the overall advantage mass—ensuring OAR redistributes credit across tokens rather than globally amplifying or shrinking the verifier signal.

Our contributions can be summarized as follows:

- We propose **OAR**, an outcome-grounded framework for fine-grained credit assignment, which redistributes sequence-level advantages based on token influence on the model's final answer distribution, with two instantiations: **OAR-P** (perturbation-based attribution) and **OAR-G** (gradient-based approximation).
- We design a conservative **Bi-Level** advantage reallocation mechanism that sharpens learning signals by boosting pivotal tokens and suppressing low-influence noise, while preserving the overall advantage mass to maintain stable training.
- Extensive experiments on mathematical reasoning benchmarks demonstrate that OAR consistently outperforms strong GRPO baselines. While OAR-P sets the performance upper bound, OAR-G retains the majority of these gains with minimal computational cost.

2 Preliminaries

2.1 Group Relative Policy Optimization

A large language model defines an autoregressive policy π_θ that generates a token sequence $y = (y_1, \ldots, y_T)$. In RLVR, we maximize the expected sequence-level reward:

$$J(\theta) = \mathbb{E}_{x\sim\mathcal{D},\, y\sim\pi_\theta(\cdot|x)}\big[r(y)\big]. \tag{1}$$

GRPO eliminates the learned value function in PPO by using a group-normalized outcome reward as the advantage. Given a prompt x, we sample G completions $\{y^{(1)}, \ldots, y^{(G)}\}$ from the behavior policy $\pi_{\theta_{\text{old}}}$ and obtain rewards $\{r_1, \ldots, r_G\}$. The advantage for sample i is computed by within-group normalization:

$$\hat{A}_i = \frac{r_i - \frac{1}{G}\sum_{k=1}^{G} r_k}{\sqrt{\frac{1}{G}\sum_{k=1}^{G}\Big(r_k - \frac{1}{G}\sum_{j=1}^{G} r_j\Big)^2}}. \tag{2}$$

This scalar $\hat{A}_i$ is applied to all token-level log-probability terms in $y^{(i)}$.

GRPO then optimizes the PPO-style clipped surrogate objective:

$$\mathcal{L}_{\text{GRPO}} = \mathbb{E}_{i,t}\Big[\min\Big(\rho_t^{(i)}\hat{A}_i, \; \text{clip}(\rho_t^{(i)}, 1-\epsilon, 1+\epsilon)\hat{A}_i\Big)\Big]. \tag{3}$$

with importance ratio

$$\rho_t^{(i)} = \frac{\pi_\theta(y_t^{(i)} \mid x, y_{<t}^{(i)})}{\pi_{\theta_{\text{old}}}(y_t^{(i)} \mid x, y_{<t}^{(i)})}. \tag{4}$$

While GRPO avoids value-function training, using a single sequence-level advantage for all tokens can increase gradient variance due to reward aliasing; see Appendix A.1.1.

2.2 Intrinsic Signals for Token-Level Credit

Recent GRPO variants explore redistributing sequence-level advantages using intrinsic, model-derived signals as token-importance proxies (Li

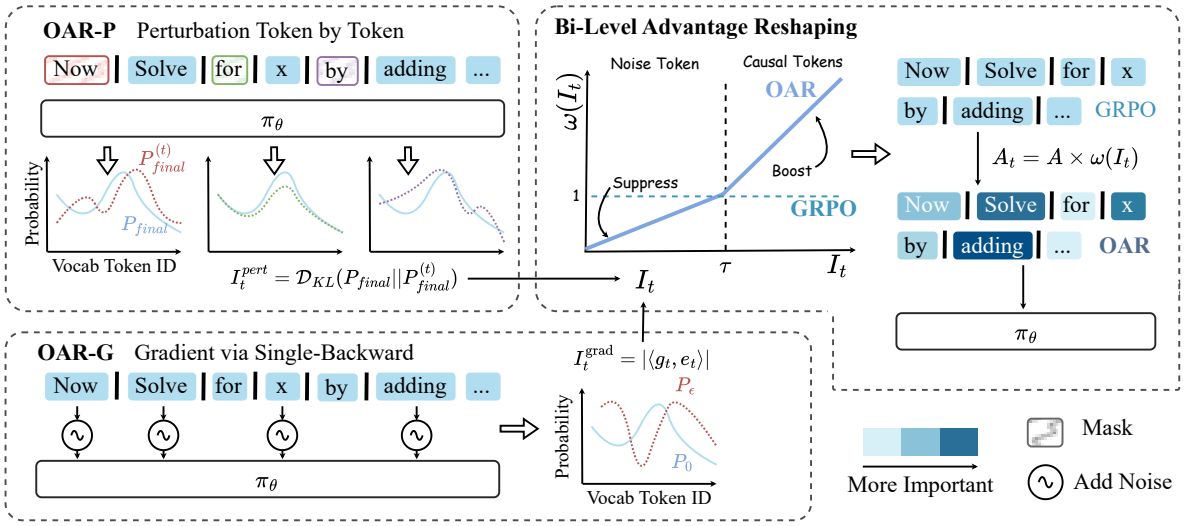


Figure 2: Overall architecture of the proposed OAR framework integrated into GRPO.

et al., 2025; Xie et al., 2025; Chen et al., 2025). Among them, entropy-based advantage shaping (Cheng et al., 2025) is a representative and easily reproducible instance. For token distribution $\pi_\theta(\cdot \mid x, y_{<t})$, the token entropy is

$$H_t = -\sum_v \pi_\theta(v \mid x, y_{<t}) \log \pi_\theta(v \mid x, y_{<t}). \tag{5}$$

A typical shaping rule augments the broadcast advantage A with an entropy-dependent bonus (Cheng et al., 2025):

$$\tilde{A}_t = A + \min\Big(\alpha \cdot \mathrm{sg}(H_t),\ |A|/\kappa\Big), \tag{6}$$

where $\mathrm{sg}(\cdot)$ stops gradients through the proxy for stability.

While simple, intrinsic proxies may not reflect *outcome-relevant* token importance: entropy measures predictive uncertainty and can be high for stylistic or lexically ambiguous tokens with little effect on correctness. Moreover, additive shaping alters the post-normalization advantage mass, inducing sequence-dependent update scales unless renormalization or re-tuning is applied (Theoretical Analysis is provided in Appendix A.1.2).

3 Method

We introduce **Outcome-grounded Advantage Reshaping (OAR)**, which equips GRPO with token-level credit assignment. OAR admits two instantiations: **OAR-P**, a perturbation-based attribution method, and **OAR-G**, an efficient gradient-based approximation. Figure 2 provides an overview of how OAR is integrated into GRPO.

3.1 OAR-P

To establish a rigorous causal baseline, OAR-P employs a counterfactual masking strategy. Let $y = (y_1, \ldots, y_T)$ be a reasoning chain generated by the policy π_θ given prompt x. We denote the model's final prediction distribution as $P_{\text{final}} = \pi_\theta(\cdot \mid x, y)$, which represents the probability distribution over the answer span. Here the answer span is extracted from the model output by a deterministic rule (e.g., regex matching `<answer>... </answer>`; see 6.2).

We construct a perturbed sequence $\tilde{y}^{(t)}$ by replacing the token at position t with a special token (e.g., `[PAD]`):

$$\tilde{y}^{(t)} = (y_1, \ldots, y_{t-1}, \texttt{[PAD]}, y_{t+1}, \ldots, y_T). \tag{7}$$

Then perform a forward pass to compute the perturbed final distribution $P^{(t)}_{\text{final}} = \pi_\theta(\cdot \mid x, \tilde{y}^{(t)})$.

The causal importance score I^{pert}_t for token y_t is defined as the KL divergence between the original and the perturbed distributions:

$$I^{\text{pert}}_t = D_{\text{KL}}\left(P_{\text{final}} \,||\, P^{(t)}_{\text{final}}\right). \tag{8}$$

Intuitively, I^{pert}_t measures the information loss incurred by removing y_t. If y_t is a logical pivot, masking it will significantly alter the final prediction ($I^{\text{pert}}_t \gg 0$); if y_t is syntactic filler, the prediction remains stable ($I^{\text{pert}}_t \approx 0$). A thereotical perspective is provided in Appendix A.1.3. While accurate, calculating I^{pert}_t for a sequence of length L requires L additional forward passes, creating a computational bottleneck.

3.2 OAR-G

To enable scalable fine-grained credit assignment during online training, we propose OAR-G, which approximates the perturbation-based attribution using a single backward pass. Instead of discrete masking, we inject small Gaussian noise into the input representations of reasoning tokens and measure how sensitive the final outcome distribution is to each token. Let $E = (e_1, \ldots, e_T)$ denote the embeddings of the reasoning tokens in the sampled trajectory. We first compute the *teacher* outcome distribution from the unperturbed input:

$$P_0 = P_{\text{final}}(x, E). \tag{9}$$

Next, we apply isotropic Gaussian noise to the embeddings and obtain a *student* outcome distribution under the perturbed representations:

$$\tilde{e}_t = e_t + \epsilon_t, \epsilon_t \sim \mathcal{N}(0, \sigma^2 I), P_\epsilon = P_{\text{final}}(x, \tilde{E}). \tag{10}$$

We then measure the induced distribution shift via a self-distillation objective:

$$\mathcal{J}(x, y) = D_{\text{KL}}\left(P_0 \,||\, P_\epsilon\right) = \sum_v P_0(v)\left(\log P_0(v) - \log P_\epsilon(v)\right). \tag{11}$$

Token-wise sensitivity is obtained by differentiating $\mathcal{J}$ with respect to each input embedding:

$$g_t = \nabla_{e_t} \mathcal{J}(x, y). \tag{12}$$

Following gradient-based attribution methods (Sundararajan et al., 2017), we define the importance score using *Gradient* $\times$ *Input*:

$$I_t^{\text{grad}} = \left|\langle g_t, e_t \rangle\right|. \tag{13}$$

This score corresponds to a first-order approximation of how strongly perturbing token t would affect the outcome distribution. Importantly, OAR-G replaces $O(L)$ counterfactual forward passes with a single backward pass, making it computationally negligible compared to the generation process.

Normalization For both OAR-P and OAR-G, the raw importance scores (I_t^{pert} or I_t^{grad}) are mapped to a unified scale for stability. We apply a log-transform followed by min-max normalization within each sequence:

$$\bar{I}_t = \log(1 + I_t), \quad \hat{I}_t = \frac{\bar{I}_t - \min_j \bar{I}_j}{\max_j \bar{I}_j - \min_j \bar{I}_j + \epsilon}. \tag{14}$$

These normalized scores $\hat{I}_t$ are then used to modulate the advantage estimates in the subsequent reshaping phase.

3.3 Bi-Level Advantage Reshaping

To concentrate credit on important positions without changing GRPO's overall update scale, we reallocate A_{seq} across tokens by suppressing low-importance tokens and boosting high-importance ones, followed by a sum-preserving renormalization. The token-level advantages are reshaped using the normalized importance $\hat{I}_t$. Specifically, we modulate the sequence advantage by a bi-level gating function:

$$A_t^{\text{OAR}} = A_{\text{seq}} \cdot \tilde{\omega}_t, \tag{15}$$

where $\tilde{\omega}_t$ suppresses low-importance tokens and boosts high-importance tokens. Given a threshold τ (e.g., the 70th percentile within a sequence), we define

$$\omega(\hat{I}_t) = \begin{cases} \underbrace{\dfrac{\hat{I}_t}{\tau + \epsilon}}_{\text{Noise Suppression}}, & \text{if } \hat{I}_t < \tau \\ \underbrace{1 + \beta \cdot \dfrac{\hat{I}_t - \tau}{1 - \tau + \epsilon}}_{\text{Signal Boosting}}, & \text{if } \hat{I}_t \geq \tau \end{cases} \tag{16}$$

with boosting coefficient $\beta \geq 0$ and smoothing constant ϵ. This function is continuous at $\hat{I}_t = \tau$ with $\omega(\tau) = 1$. To avoid changing the overall update scale, we apply sum-preserving renormalization:

$$\tilde{\omega}_t \;=\; \omega(\hat{I}_t) \cdot \frac{T}{\sum_{j=1}^{T} \omega(\hat{I}_j)}. \tag{17}$$

This conserves total advantage mass ($\sum_t \tilde{\omega}_t = T$), so OAR redistributes credit across tokens rather than globally rescaling advantages.

Finally, we replace the standard advantage in the clipped PPO objective with A_t^{OAR}:

$$\mathcal{L}_{\text{OAR}} = \mathbb{E}_{i,t}\Bigg[\min\Big(\rho_t^{(i)} A_t^{\text{OAR}(i)}, \; \text{clip}\big(\rho_t^{(i)}, 1-\epsilon, 1+\epsilon\big) A_t^{\text{OAR}(i)}\Big)\Bigg]. \tag{18}$$

4 Experiments

4.1 Experiment Setup

Datasets and Models Experiments are conducted on `Qwen2.5-7B-Base` and `Qwen2.5-Math-7B` (Yang et al., 2025a), trained

Method	AIME25		AIME24		AMC23		MATH500	GSM8K	Avg
	P@1	*P@32*	*P@1*	*P@32*	*P@1*	*P@32*	*P@1*	*P@1*	*Avg*
Qwen2.5-7B	2.5	26.4	4.5	43.6	29.5	81.4	52.7	82.0	45.2
+ GRPO	9.4	31.0	13.5	45.0	59.8	85.0	75.8	90.5	51.3
+ GRPO w/ Random Adv	9.2	31.0	13.2	44.3	60.1	85.5	75.2	90.3	51.1
+ GRPO w/ Entropy Adv	10.3	32.4	14.4	45.7	**62.2**	86.4	77.2	91.2	52.5
+ GRPO w/ OAR-G	11.8	33.2	14.8	47.8	61.4	86.9	**78.7**	91.4	53.2
+ GRPO w/ OAR-P	**12.2**	**33.9**	**15.2**	**48.2**	61.9	**87.7**	78.4	**92.0**	**53.7**
Δ (vs. GRPO)	+2.8	+2.9	+1.7	+3.2	+2.1	+2.7	+2.6	+1.5	+2.4
Qwen2.5-Math-7B	5.7	33.5	11.8	49.4	33.2	88.3	51.2	69.0	42.8
+ GRPO	12.2	39.1	31.2	48.9	64.5	90.6	80.8	91.2	57.3
+ GRPO w/ Random Adv	11.9	39.4	30.8	47.2	64.0	90.2	79.8	91.2	56.8
+ GRPO w/ Entropy Adv	13.0	39.6	32.4	49.0	65.4	90.6	81.7	91.4	57.9
+ GRPO w/ OAR-G	14.3	40.3	**33.5**	51.7	65.2	92.5	83.0	**92.9**	59.2
+ GRPO w/ OAR-P	**14.5**	**40.8**	**33.5**	**51.9**	**65.6**	**92.8**	**83.8**	92.7	**59.5**
Δ (vs. GRPO)	+2.3	+1.7	+2.3	+3.0	+1.1	+2.2	+3.0	+1.5	+2.2

Table 1: Main results on mathematical reasoning benchmarks. **bold** and underline indicate the best and second-best results, respectively. Δ denotes the performance gain between OAR-P and the vanilla GRPO baseline.

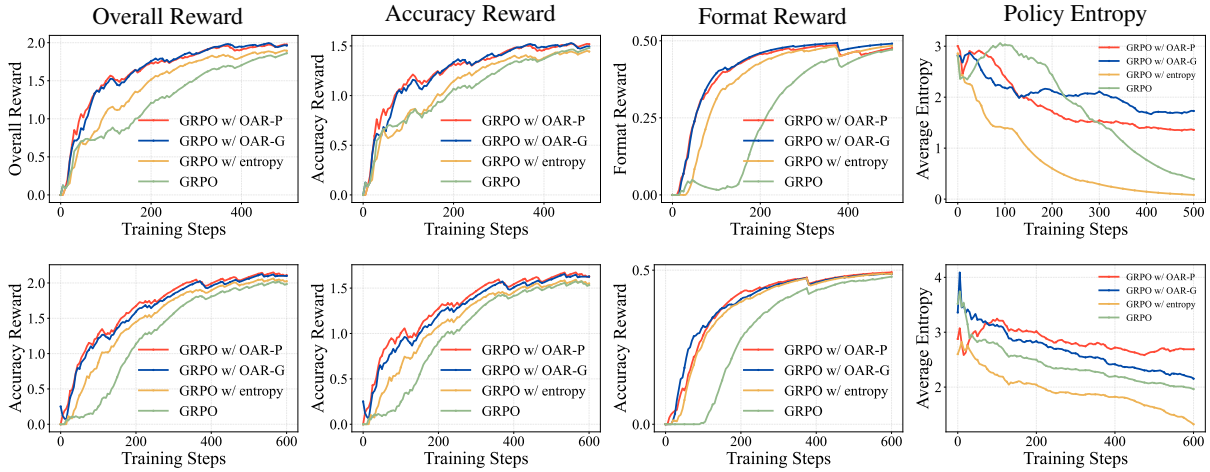


Figure 3: Training dynamics on Qwen2.5-7B-Base(Top Row) and Qwen2.5-Math-7B(Bottom Row).

with DAPO-MATH (Yu et al., 2025a) dataset, We evaluate the resulting performance on four widely used math reasoning benchmarks: AIME2024/2025 (LI et al., 2024), AMC23 (LI et al., 2024), MATH500 (Hendrycks et al., 2021), and GSM8K (Cobbe et al., 2021). For AIME and AMC, we report Pass@k where $k \in \{1, 32\}$; for others we report standard Pass@1(Chen et al., 2021).

Implementation Details We employ the GRPO framework with a group size of $G = 8$, learning rate of 1×10^{-6} and a global batch size of 64. For generation, We set the temperature to 1.0 with a max generation length of 1024. For our proposed OAR method, we set the bi-level threshold $\tau = 0.4$ (meaning the top 60% of tokens are boosted) and the boosting coefficient $\beta = 2.0$. To accelerate the counterfactual perturbation step during training, we implement a parallelized batch decoding strategy, which computes the perturbed logits for all masked positions in a single batched forward pass. All experiments are conducted on $8 \times$ H100 GPUs.

4.2 Baselines

To validate the effectiveness of our outcome-grounded credit assignment, we compare OAR against three baselines: (i) *Vanilla GRPO*, the standard Group Relative Policy Optimization algorithm equipped with the clip-higher technique from DAPO (Yu et al., 2025a) as a strong reference; (ii) *GRPO + Random Credit*, a diagnostic variant that assigns each token a random weight $w \sim \text{Uniform}(0, 1)$; and (iii) *GRPO + Entropy*, which follows Cheng et al. (2025) and uses token entropy as an importance proxy to reshape advantages.

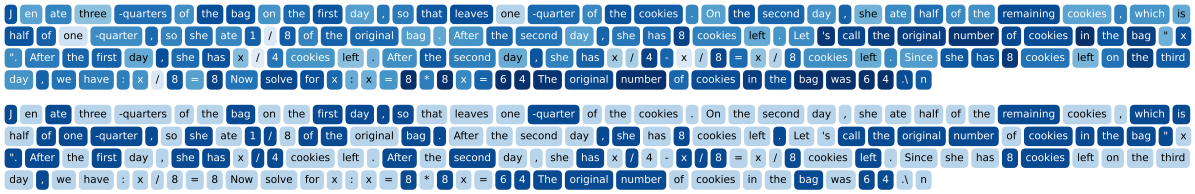
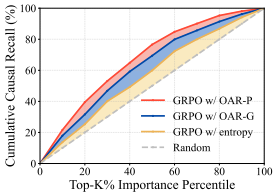
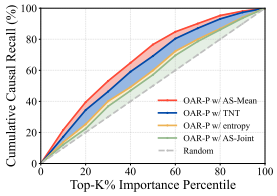


Figure 4: Token-importance visualization on a reasoning trace: OAR-P (top) vs. Oracle causal mask (bottom).



(a) Recall rate of different credit assignment signals.



(b) Recall rate under different outcome definitions.

Figure 5: Causal-token recall under the counterfactual Oracle as a function of the top-$K\%$ important tokens.

Method	AIME24	AMC23	MATH500	Avg
Qwen2.5-Math-1.5B	2.9	20.2	40.3	21.1
+ GRPO	13.8	56.7	73.0	47.8
w/ Random Adv	13.8	57.0	72.6	47.8
w/ Entropy Adv	14.2	58.2	74.5	48.9
w/ OAR-G	14.4	58.0	75.1	49.2
w/ OAR-P	**15.0**	**58.7**	**75.5**	**49.7**
Δ (vs. GRPO)	+1.2	+2.0	+2.5	+1.9

Table 2: Results on Qwen2.5-Math-1.5B.

4.3 Main Results

Reasoning Performance Table 1 summarizes results on five mathematical reasoning benchmarks using Qwen2.5-7B and Qwen2.5-Math-7B as backbones. Incorporating our OAR into GRPO consistently improves performance over vanilla GRPO and alternative token-weighting heuristics. For example, on Qwen2.5-7B, OAR-P improves AIME25 *P@1* from 9.0 to 11.8 and boosts the overall average from 51.1 to **53.5**; OAR-G achieves comparable performance (Avg 53.0). Similar trends hold for Qwen2.5-Math-7B, where OAR-P raises the average from 57.0 to **59.2**. Overall, OAR yields robust improvements across datasets.

Training Dynamics Figure 3 compares training dynamics under different credit assignment strategies. Across runs, OAR improves optimization stability and reaches higher rewards without inducing premature entropy collapse. Compared with entropy-based shaping, which tends to reduce policy entropy more aggressively, OAR maintains a healthier exploration–exploitation balance while steadily increasing both the accuracy-related reward and the format-related reward.

Scaling to Different Sizes We additionally conduct experiments on a smaller *Qwen2.5-Math-1.5B* model to verify the consistency and scalability of OAR. As shown in Table 2, OAR-G and OAR-P consistently improve over baselines.

5 Analysis

5.1 Is OAR Identifying Important Tokens?

A core premise of OAR is that the identified token importance score $\hat{I}_t$ correlates with a token's true causal contribution to the final outcome. We test this premise using a counterfactual Oracle and evaluate both OAR-P and OAR-G against entropy-based weighting in a quantitative correlation study. This experiment is conducted on Qwen2.5-7B-Base.

The Oracle Given a correct reasoning chain $y = (y_1, \ldots, y_T)$, following the counterfactual token-substitution test used to identify critical tokens in (Ruan et al., 2025), we construct a counterfactual sequence by replacing token y_t with the second most probable token under $\pi_\theta(\cdot \mid y_{<t})$, and then greedily decoding the remaining suffix. Let $\mathcal{R}(\cdot) \in \{0, 1\}$ denote final-answer correctness. The Oracle label is:

$$\mathcal{O}_t = \mathbb{I}\left(\mathcal{R}(y^{(t)}_{\text{replaced}}) \neq \mathcal{R}(y_{\text{original}})\right). \quad (19)$$

If replacement flips correctness, y_t is a **Causal Pivot** ($\mathcal{O}_t = 1$); otherwise it is **Non-Causal Noise** ($\mathcal{O}_t = 0$).

Alignment Visualization Figure 4 visualizes token importance for OAR-P (top) alongside the Oracle mask (bottom). OAR-P assigns high importance to outcome-critical symbols (e.g., numbers, operators, and key algebraic terms) while down-weighting stylistic or redundant tokens, qualita-

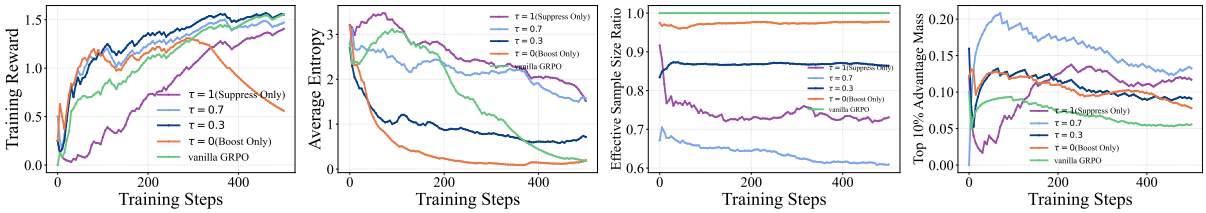


Figure 6: OAR-G ablations on Qwen2.5-7B-Base with different thresholds τ (left to right): reward, policy entropy, ESS ratio, and top-10% advantage mass.

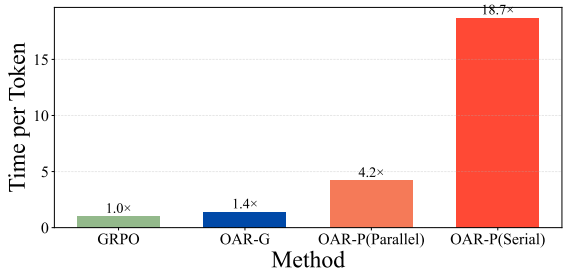


Figure 7: Normalized training time per token.

tively matching the sparse counterfactual supervision. We provide additional case studies for OAR-G in Appendix A.2.

Quantitative Correlation We sample 500 trajectories, rank tokens by $\hat{I}_t$, and compute the recall of Oracle causal tokens within the top-$K\%$ tokens. In Figure 5a, both OAR-P and OAR-G achieve substantially higher causal-token recall than entropy-based weighting across all percentiles, with OAR-P consistently performing best.

5.2 Computational Trade-off

OAR improves credit assignment at the cost of additional attribution computation. We quantify this overhead using wall-clock time per token during training. For each update iteration, we record the elapsed time Δt of one optimizer-update sweep over the buffered minibatches, and divide it by the number of action tokens.

Figure 7 shows the normalized cost (GRPO = $1.0\times$). OAR-G incurs only a modest overhead ($1.4\times$), consistent with requiring a single backward-based proxy for importance. In contrast, OAR-P is substantially more expensive due to counterfactual evaluation: even with our batched (parallel) implementation it costs $4.2\times$, while a naive serial implementation would be prohibitive ($18.7\times$). Overall, OAR-G offers the best accuracy–efficiency balance, while OAR-P serves as a higher-fidelity but costlier upper bound.

6 Ablations

6.1 Effect of the Bi-Level Gating Mechanism

Method	GSM8K	MATH500
Vanilla GRPO	90.5	75.8
OAR (Suppress-only, $\tau = 1.0$)	85.2	70.1
OAR (Boost-only, $\tau = 0.0$)	81.5†	61.3†
OAR (Balanced, $\tau = 0.7$)	89.8	74.2
OAR (Balanced, $\tau = 0.3$)	**91.2**	**78.1**

Table 3: Ablation results on Qwen2.5-7B-Base. † indicates runs that exhibited instability or collapse during training.

We ablate the bi-level gating in OAR by varying the threshold τ on Qwen2.5-7B-Base. This study is conducted based on OAR-G, for brevity we refer to it as OAR. Figure 6 reports training dynamics, and Table 3 summarizes final accuracy.

Credit concentration We evaluate whether OAR indeed reallocates token-level credit using two auxiliary metrics (full definitions in Appendix A.3). *ESS ratio* measures how uniform the token weights are (smaller means more concentrated), while *Top-10% advantage mass* measures what fraction of total token advantage lies in the top 10% tokens. Compared with vanilla GRPO, OAR yields markedly more concentrated credit assignment, and larger τ generally produces sparser allocations.

Trade-off More aggressive sparsification is not always beneficial. As shown by the reward and entropy curves in Figure 6, a moderate threshold performs best. *Boost-only* ($\tau = 0$) learns quickly but is prone to collapse, whereas *suppress-only* ($\tau = 1$) remains stable yet inefficient. Overall, the balanced setting achieves the best accuracy (Table 3).

6.2 Effect of Outcome Definition

OAR is outcome-grounded through an outcome probe $\phi(x, y)$ that summarizes the model's final-answer prediction. Because ϕ is not unique,

Outcome definition	GSM8K	MATH500
LT-LOGITS	91.5	78.0
AS-JOINT	91.0	77.3
AS-MEAN	**92.0**	**78.4**

Table 4: Accuracy ablation of outcome definitions.

different definitions may induce different token-importance rankings and thus different credit reallocation behaviors. We therefore ablate several outcome definitions to assess the robustness of causal-token identification and its impact on downstream RL performance.

Outcome definitions We consider three outcome operators $\phi(\cdot)$:

- **Last-Token Logits (LT-Logits).** $\phi_{\text{LT-LOGITS}}(x, y) = z_L$, i.e., the next-token logits at the last position.
- **Answer-Span Mean Logits (AS-Mean).** $\phi_{\text{AS-MEAN}}(x, y) = \frac{1}{|\mathcal{A}|} \sum_{\ell \in \mathcal{A}} z_\ell$, averaging logits over positions $\mathcal{A}$ that predict the final answer span, where the span is extracted by regex matching `<answer>...</answer>` in the model output.
- **Answer-Span Joint Likelihood (AS-Joint).** $\phi_{\text{AS-JOINT}}(x, y) = \frac{1}{|\mathcal{A}|} \sum_{\ell \in \mathcal{A}} \log \pi_\theta(a_{\ell+1} \mid x, y_{\le \ell})$, the mean token log-likelihood of the extracted answer span.

For LT-Logits and AS-Mean we compute importance via $D_{\text{KL}}(\text{softmax}(\phi) \,\|\, \text{softmax}(\tilde{\phi}))$ under token masking; for AS-Joint we use the score drop $\phi - \tilde{\phi}$.

Experiment setup We follow the quantitative correlation setup in Section 5.1, measure the recall rate under three methods. Notability, For AS-MEAN, both in this experiment and in our training runs, we initially apply LT-LOGITS as a warm-up, as in early training stage the model may not reliably produce the required *<answer>...</answer>* format. We further report the task accuracy to connect attribution quality with actual performance.

Results Figure 5b and Table 4 show that AS-MEAN achieves the highest Oracle top-$K\%$ recall and the best downstream training performance, followed by LT-LOGITS, while AS-JOINT performs worst. We attribute this to a representational mismatch: LT-LOGITS and AS-MEAN operate on distributions and thus can capture subtle shifts in the outcome distribution induced by token perturbations, whereas AS-JOINT collapses the outcome into a scalar likelihood and may miss substantial distributional changes.

7 Related Work

Reinforcement Learning with Verifiable Rewards RLVR post-trains LLMs using automatically checkable outcome signals, yielding strong gains in math and code via verifiers such as Math-Verify and sandboxed execution (Cui et al., 2025; Yu et al., 2025a; Luo et al., 2025; DeepSeek-AI et al., 2025). GRPO further stabilize large-scale training by group-normalizing verifier rewards (Shao et al., 2024). To move beyond domains with explicit checkers, recent studies learn generative reward/verifier models for model-based judging (Mahan et al., 2024; Ma et al., 2025; Liu et al., 2025), while verifier-free variants instead use intrinsic signals such as per-token likelihood on reference answers to enable RL-style post-training without dedicated verifiers (Yu et al., 2025b; Zhou et al., 2025).

Fine-Grained Credit Assignment To address the coarseness of group-level rewards, recent studies explored leveraging model-internal signals to redistribute credits at the token level. Li et al. (2025) identify a recurring *preplan-and-anchor* rhythm from attention dynamics and upweight advantages on the corresponding critical tokens. Xie et al. (2025) propose UCAS, which reshapes advantages using response-level confidence and a token-level logit-based certainty penalty to encourage exploration. Chen et al. (2025) introduce LESS, which segments generations by entropy and reweights advantages based on low-entropy segment overlap between correct and incorrect rollouts.

8 Conclusion

We propose Outcome-grounded Advantage Reshaping (OAR) to improve credit assignment in GRPO for long-horizon reasoning. OAR redistributes sequence-level advantages to tokens based on outcome contribution, with OAR-P using counterfactual attribution and OAR-G a lightweight gradient-based proxy. Experiments on mathematical reasoning benchmarks show consistent gains over strong GRPO baselines. Future work will extend OAR to open-ended tasks and deeper RL post-training integration.

Limitaions

OAR estimates token importance by measuring shifts in the model's own answer distribution, which serves as an efficient surrogate when verifier rewards are discrete and non-differentiable; nevertheless, this proxy can be imperfect when distributional changes are weakly coupled with verifier acceptance. Our perturbation-based attribution further relies on a masking intervention (e.g., `[PAD]`), which is a controlled counterfactual but may introduce some distribution shift compared to in-distribution edits. Empirically, we focus on mathematical reasoning with verifiable final answers, and extending OAR to open-ended or interactive settings where outcomes are less localized remains an important direction. Finally, beyond entropy-based shaping, several closely related recent methods are not yet fully reproducible, we will broaden comparisons as these baselines become easier to faithfully re-implement.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shayne An, and 1 others. 2023. GPT-4 Technical Report. *arXiv preprint arXiv:2303.08774*.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. *Preprint*, arXiv:2005.14165.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Pondé, Jared Kaplan, Harrison Edwards, Yura Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, and 34 others. 2021. Evaluating large language models trained on code. *ArXiv*, abs/2107.03374.

Xinzhu Chen, Xuesheng Li, Zhongxiang Sun, and Weijie Yu. 2025. Beyond high-entropy exploration: Correctness-aware low-entropy segment-based advantage shaping for reasoning llms. *Preprint*, arXiv:2512.00908.

Daixuan Cheng, Shaohan Huang, Xuekai Zhu, Bo Dai, Wayne Xin Zhao, Zhenliang Zhang, and Furu Wei. 2025. Reasoning with exploration: An entropy perspective. *Preprint*, arXiv:2506.14758.

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

Ganqu Cui, Lifan Yuan, Zefan Wang, Hanbin Wang, Yuchen Zhang, Jiacheng Chen, Wendi Li, Bingxiang He, Yuchen Fan, Tianyu Yu, Qixin Xu, Weize Chen, Jiarui Yuan, Huayu Chen, Kaiyan Zhang, Xingtai Lv, Shuo Wang, Yuan Yao, Xu Han, and 6 others. 2025. Process reinforcement through implicit rewards. *Preprint*, arXiv:2502.01456.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, and 181 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

William Fedus, Barret Zoph, and Noam Shazeer. 2022. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Preprint*, arXiv:2101.03961.

Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1050–1059, New York, New York, USA. PMLR.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, D. Song, and J. Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. *ArXiv*, abs/2103.03874.

Sarthak Jain and Byron C. Wallace. 2019. Attention is not explanation. *Preprint*, arXiv:1902.10186.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V. Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, David Havasi, Shrimai Prabhumoye, Pengfei Liu, Jesse Dodge, Yejin Choi, Noah A. Smith, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2024. Tulu 3: Pushing frontiers in open language model post-training. *Preprint*, arXiv:2411.15124.

Jia LI, Edward Beeching, Lewis Tunstall, Ben Lipkin, Roman Soletskyi, Shengyi Costa Huang, Kashif Rasul, Longhui Yu, Albert Jiang, Ziju Shen, Zihan Qin, Bin Dong, Li Zhou, Yann Fleureau, Guillaume Lample, and Stanislas Polu. 2024. Numinamath. `[https://huggingface.co/AI-MO/NuminaMath-CoT](https://github.com/project-numina/aimo-progress-prize/blob/main/report/numina_dataset.pdf)`.

Yang Li, Zhichen Dong, Yuhan Sun, Weixun Wang, Shaopan Xiong, Yijia Luo, Jiashun Liu, Han Lu, Jiamang Wang, Wenbo Su, Bo Zheng, and Junchi Yan. 2025. Attention illuminates LLM reasoning: The preplan-and-anchor rhythm enables fine-grained policy optimization. *Preprint*, arXiv:2510.13554.

Ziniu Li, Congliang Chen, Tian Xu, Zeyu Qin, Jiancong Xiao, Zhi-Quan Luo, and Ruoyu Sun. 2024. Preserving diversity in supervised fine-tuning of large language models. *arXiv preprint arXiv:2408.16673*.

Aiwei Liu, Haoping Bai, Zhiyun Lu, Yanchao Sun, Xiang Kong, Simon Wang, Jiulong Shan, Albin Madappally Jose, Xiaojiang Liu, Lijie Wen, and 1 others. 2024. Tis-dpo: Token-level importance sampling for direct preference optimization with estimated weights. *arXiv preprint arXiv:2410.04350*.

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

Michael Luo, Sijun Tan, Justin Wong, Xiaoxiang Shi, William Y. Tang, Manan Roongta, Colin Cai, Jeffrey Luo, Li Erran Li, Raluca Ada Popa, and Ion Stoica. 2025. DeepScaleR: Surpassing O1-Preview with a 1.5B Model by Scaling RL. https://pretty-radio-b75.notion.site/DeepScaleR-Surpassing-O1-Preview-with-a-1-5B-Model-by-Scaling-RL-19681902c1468005bed8ca303013a4e2. Notion Blog.

Xueguang Ma, Qian Liu, Dongfu Jiang, Ge Zhang, Zejun Ma, and Wenhu Chen. 2025. General-reasoner: Advancing llm reasoning across all domains. *Preprint*, arXiv:2505.14652.

Dakota Mahan, Duy Van Phung, Rafael Rafailov, Chase Blagden, Nathan Lile, Louis Castricato, Jan-Philipp Fränken, Chelsea Finn, and Alon Albalak. 2024. Generative reward models. *Preprint*, arXiv:2410.12832.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "why should i trust you?": Explaining the predictions of any classifier. *Preprint*, arXiv:1602.04938.

Zhiwen Ruan, Yixia Li, He Zhu, Yun Chen, Peng Li, Yang Liu, and Guanhua Chen. 2025. Enhancing large language model reasoning via selective critical token fine-tuning. *Preprint*, arXiv:2510.10974.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Rui Xu, Junmei Song, Mingchuan Zhang, Y.K. Li, Yuxiang Wu, and Daya Guo. 2024. DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models. *Preprint*, arXiv:2402.03300.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2014. Deep inside convolutional networks: Visualising image classification models and saliency maps. *Preprint*, arXiv:1312.6034.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. *Preprint*, arXiv:1703.01365.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models. *Preprint*, arXiv:2201.11903.

Can Xie, Ruotong Pan, Xiangyu Wu, Yunfei Zhang, Jiayi Fu, Tingting Gao, and Guorui Zhou. 2025. Unlocking exploration in rlvr: Uncertainty-aware advantage shaping for deeper reasoning. *Preprint*, arXiv:2510.10649.

An Yang, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoyan Huang, Jiandong Jiang, Jianhong Tu, Jianwei Zhang, Jingren Zhou, Junyang Lin, Kai Dang, Kexin Yang, Le Yu, Mei Li, Minmin Sun, Qin Zhu, Rui Men, Tao He, and 9 others. 2025a. Qwen2.5-1m technical report. *Preprint*, arXiv:2501.15383.

Zhihe Yang, Xufang Luo, Zilong Wang, Dongqi Han, Zhiyuan He, Dongsheng Li, and Yunjian Xu. 2025b. Do not let low-probability tokens over-dominate in rl for llms. *Preprint*, arXiv:2505.12929.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, and 16 others. 2025a. Dapo: An open-source llm reinforcement learning system at scale. *Preprint*, arXiv:2503.14476.

Tianyu Yu, Bo Ji, Shouli Wang, Shu Yao, Zefan Wang, Ganqu Cui, Lifan Yuan, Ning Ding, Yuan Yao, Zhiyuan Liu, Maosong Sun, and Tat-Seng Chua. 2025b. Rlpr: Extrapolating rlvr to general domains without verifiers. *Preprint*, arXiv:2506.18254.

Xiangxin Zhou, Zichen Liu, Anya Sims, Haonan Wang, Tianyu Pang, Chongxuan Li, Liang Wang, Min Lin, and Chao Du. 2025. Reinforcing general reasoning without verifiers. *Preprint*, arXiv:2505.21493.

A Appendix

A.1 Theoretical Analysis

A.1.1 Variance from Reward Alising in GRPO

Assume there exists a latent decomposition of the sequence advantage into token-level contributions a_t, such that $A^{(i)} = \sum_{\tau=1}^{T} a_\tau$, where a_t can be viewed as an unobserved per-token outcome contribution used only for analysis. The gradient contribution for a specific token at step t in GRPO can be rewritten as:

$$\hat{g}_t = \underbrace{a_t \nabla_\theta \log \pi_t}_{\text{Token-Specific Signal}} + \underbrace{\left(\sum_{\tau \neq t} a_\tau \right) \nabla_\theta \log \pi_t}_{\text{Aliased Credit (Noise)}}, \tag{20}$$

In an ideal token-credit setting, the update would depend primarily on a_t. However, GRPO effectively injects a term $\sum_{\tau \neq t} a_\tau$ into the update for token t, whose variance can be much larger than that of a_t in long-horizon sequences:

$$\mathrm{Var}\left(A^{(i)}\right) \gg \mathrm{Var}(a_t). \tag{21}$$

Consequently, for tokens with low causal influence, the optimizer receives a high-variance signal largely determined by other tokens' actions. This misalignment hinders the policy from distinguishing critical reasoning steps from syntactic fillers.

A.1.2 Why Additive Proxy Shaping may Induce Sequence-Dependent Update Scales

Additive proxy shaping (e.g., entropy bonuses) is applied after GRPO's group-relative normalization, but is typically not re-normalized at the sequence level. Consequently, the effective update scale can vary across samples as a function of sequence length and how often the proxy is activated.

Define the (non-negative) proxy bonus and shaped advantage:

$$\begin{aligned} \delta_t &= \min\left(\alpha \cdot \mathrm{sg}(H_t), \frac{|A|}{\kappa} \right), \\ \delta_t &\geq 0, \qquad \tilde{A}_t = A + \delta_t, \end{aligned} \tag{22}$$

where A is the GRPO group-normalized sequence advantage. The unclipped per-sequence policy gradient can be decomposed as

$$\begin{aligned} g_{\text{entropy}} &= \sum_{t=1}^{T} (A + \delta_t) \nabla_\theta \log \pi_\theta(y_t \mid x, y_{<t}) \\ &= g_{\text{GRPO}} + \sum_{t=1}^{T} \delta_t \nabla_\theta \log \pi_\theta(y_t \mid x, y_{<t}). \end{aligned} \tag{23}$$

The additional term scales with the bonus mass $M = \sum_{t=1}^{T} \delta_t$, which grows with sequence length T and the density of high-proxy tokens. Thus, even when two samples share the same normalized A, their gradients can have systematically different magnitudes, effectively inducing a sample-dependent step size.

In PPO-style updates, changing the effective step size shifts the distribution of importance ratios and the realized KL to the reference policy, which may harm stability unless one applies sequence-level renormalization or re-tunes hyperparameters.

A.1.3 Outcome-Grounded Token Importance Identification

Definition. Given a prompt x and a generated sequence $y_{1:T}$, we define an *outcome probe* Z as the model's predictive distribution at an outcome-bearing position (e.g., the terminal next-token distribution or the start of an <answer> block):

$$p \;:=\; p_\theta(\cdot \mid x, y_{1:T}).$$

Let $y^{(-t)}$ denote a counterfactual sequence where token y_t is masked (e.g., replaced by [PAD]), and let

$$q_t \;:=\; p_\theta(\cdot \mid x, y^{(-t)}).$$

We measure the influence of token y_t by the KL divergence between the factual and counterfactual outcome distributions:

$$I_t(y) \;:=\; D_{\mathrm{KL}}(p \,\|\, q_t). \tag{24}$$

Interpretation. $I_t(y)$ is *outcome-grounded*: it directly quantifies how much removing y_t changes the model's predicted outcome distribution, rather than relying on intrinsic signals such as token entropy. A useful sanity connection to RLVR is that, for any event S over outcomes (e.g., the verifier-accepted set of answers),

$$\begin{aligned} \big|p(S) - q_t(S)\big| \;&\leq\; \mathrm{TV}(p, q_t) \\ &\leq\; \sqrt{\tfrac{1}{2} D_{\mathrm{KL}}(p \| q_t)} \;=\; \sqrt{\tfrac{1}{2} I_t(y)}, \end{aligned} \tag{25}$$

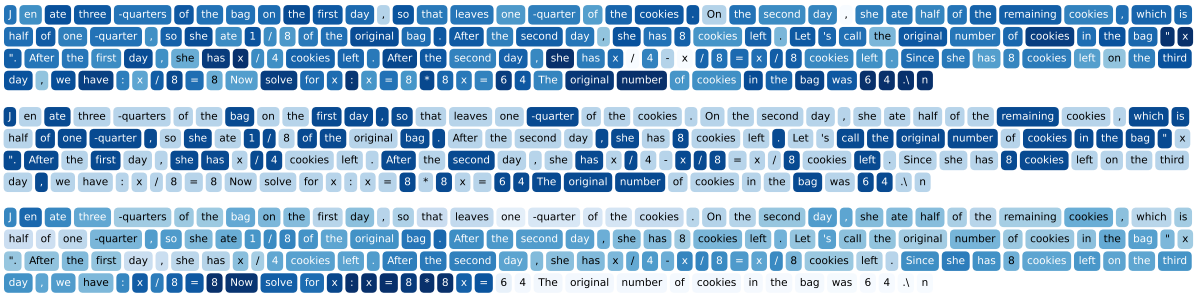


Figure 8: Token-importance visualization on a reasoning trace: OAR-G (top) vs. Oracle causal mask (Middle) vs. Entropy-Based(Bottom)

where the second inequality is Pinsker's inequality. Thus, if $I_t(y) \approx 0$, masking y_t cannot substantially change the probability mass of *any* verifier-defined correct set, suggesting y_t is counterfactually non-influential for the final outcome.

A.1.4 Gradient Noise Suppression

Standard GRPO broadcasts a trajectory-level advantage $\hat{A}$ to all tokens. For a nuisance token that causally contributes nothing to the reward, this broadcasting injects variance into the gradient estimate, as the token is updated based on a global signal it did not influence. We show how OAR mitigates this.

Claim (Noise Suppression) Consider a token position k that is counterfactually non-influential, i.e., $I_k(y) = 0$. Under OAR, the contribution of this position to the gradient variance is strictly suppressed compared to standard GRPO.

Proof Sketch. Let $s_k := \nabla_\theta \log \pi_\theta(y_k \mid x, y_{<k})$ be the score function at position k. In standard GRPO, the gradient estimator for this token is $\hat{g}^{(k)}_{\text{GRPO}} = \hat{A} \cdot s_k$. Even if token y_k is irrelevant to the task (nuisance), $\hat{A}$ varies across trajectories due to rewards earned by other tokens, and s_k is non-zero (as the model still predicts the token itself). This results in gradient noise: $\mathbb{E}[\|\hat{g}^{(k)}_{\text{GRPO}}\|^2] > 0$.

In OAR, the update is reweighted by a scalar $\omega(\hat{I}_k)$. Since $I_k(y) = 0$, our weighting scheme assigns $\omega(\hat{I}_k) \leq \varepsilon$ for some small $\varepsilon \ll 1$ (representing the suppression floor). The squared norm of the OAR gradient estimator becomes:

$$\begin{aligned}\mathbb{E}\big[\|\hat{g}^{(k)}_{\text{OAR}}\|^2\big] &= \mathbb{E}\big[\|\omega(\hat{I}_k) \cdot \hat{A} \cdot s_k\|^2\big] \\ &\leq \varepsilon^2 \cdot \mathbb{E}\big[\|\hat{A} \cdot s_k\|^2\big] \\ &= \varepsilon^2 \cdot \mathbb{E}\big[\|\hat{g}^{(k)}_{\text{GRPO}}\|^2\big].\end{aligned} \tag{26}$$

As $\varepsilon \to 0$, the gradient noise at position k vanishes. Thus, OAR acts as a *causal filter*, effectively freezing parameters associated with nuisance tokens and focusing the optimization on tokens that causally affect the outcome.

A.2 Additional Token-Importance Visualizations

This section provides additional qualitative evidence that the token-importance signals used for advantage reshaping are aligned with outcome-critical reasoning steps.Figure 8 shows a representative reasoning trace and compares OAR-G with an oracle causal mask and an entropy-based baseline.

A.3 Credit Concentration Metrics

We quantify how concentrated token-level credit becomes during training using two auxiliary metrics computed on each sequence with a valid-token mask.

ESS ratio Let $w_t \geq 0$ denote the token weight assigned by OAR, and let T be the number of valid (non-padding) tokens. Define

$$\begin{aligned}\text{ESS}(w) &= \frac{\left(\sum_{t=1}^{T} w_t\right)^2}{\sum_{t=1}^{T} w_t^2}, \\ \text{ESS-ratio}(w) &= \frac{\text{ESS}(w)}{T} = \frac{\left(\sum_{t=1}^{T} w_t\right)^2}{T \sum_{t=1}^{T} w_t^2}.\end{aligned} \tag{27}$$

$\text{ESS-ratio} \in (0, 1]$ measures the uniformity of weights: 1 corresponds to uniform assignment, and smaller values indicate more concentrated credit.

Top-10% advantage mass Let A_t denote the token-level advantage used in the policy-gradient update. We measure what fraction of the total absolute advantage mass is carried by the top 10%

tokens (by $|A_t|$):

$$m_{\text{top}} = \frac{\sum_{t \in \text{TopK}(|A|)} |A_t|}{\sum_{t=1}^{T} |A_t|}, \qquad K = \max\left(1, \lfloor 0.1T \rfloor\right). \tag{28}$$

where $\text{TopK}(|A|)$ returns the indices of the K largest $|A_t|$ among valid tokens. Larger m_{top} indicates that the update is dominated by a small subset of tokens.