Sparse but Critical: A Token-Level Analysis of Distributional Shifts in RLVR Fine-Tuning of LLMs

Anonymous authors
Paper under double-blind review

Abstract

Reinforcement learning with verifiable rewards (RLVR) has significantly improved reasoning in large language models (LLMs), yet the token-level mechanisms through which they reshape model behavior remain unclear. We present a systematic empirical study of RLVR's distributional effects across three complementary axes: (1) token-level distributional shifts, (2) functional validation via cross-sampling interventions, and (3) exploratory investigations of advantage signal modulation based on token divergence. We find that RL fine-tuning induces sparse, targeted changes, with only a small fraction of tokens exhibiting significant distributional divergence, and we further analyze the nature of these shifts. These divergent distributions are not uniformly predicted by entropy, indicating that RLVR can modify both initially high and low entropy distributions under different settings. Cross-sampling experiments reveal that inserting just a small fraction of RL-sampled tokens into base model generations recovers most RL performance gains, while injecting a small portion of base-sampled tokens into RL generations collapses performance to base levels, functionally isolating the critical role of divergent tokens. Finally, we explore divergence-weighted variants of the advantage signal, finding that they can amplify improvements in baselines. Our work sheds light on the distributional changes induced by RLVR and provides a granular, token-level lens for understanding and improving RL fine-tuning in LLMs.

1 Introduction

Recent advances in reinforcement learning with verifiable rewards (RLVR) (Lambert et al., 2024) for reasoning in large language models (LLMs), such as Group Relative Policy Optimization (GRPO) (Shao et al., 2024), have enabled models to achieve strong performance on challenging reasoning and math benchmarks. Despite their empirical success, the mechanisms through which RL modifies model behavior remain unclear.

Many evaluations focus on aggregate metrics such as accuracy, rewards, and response lengths. While valuable, these metrics provide only a coarse view of model improvement and offer limited insight into the fine-grained changes induced by RL fine-tuning. In particular, it remains unclear: *How does RL reshape the token-level output distributions of the base model?* A growing line of work has begun to examine RL-finetuning of LLMs from the perspective of token entropy (Wang et al., 2025; Cheng et al., 2025; Cui et al., 2025), highlighting the role of high-entropy tokens. However, a more granular understanding of how RLVR redistributes probability mass across tokens, and what structure these shifts exhibit, remains unclear.

In this paper, we investigate the fine-grained dynamics of **distributional change** induced by RLVR on the token-level. Our contributions are organized into three interconnected themes:

- **Token-Level Distribution Analysis:** We quantify how RL fine-tuning reshapes token distributions, showing that changes are sparse and targeted, with only a small fraction of tokens exhibit significant divergence. We characterize these shifts using JS divergence, entropy, and positional trends, and compare DAPO and GRPO, revealing differences in exploration behavior and refinement strategies across training dynamics.

- **Functional Interventions:** Through forward and reverse cross-sampling experiments, we demonstrate that high-divergence tokens are functionally critical: injecting RL choices at these positions progressively recovers most of the performance gains, while reverting them in RL generations collapses performance.
- **Divergence-Weighted Advantage:** Motivated by our findings, we experiment with an intervention to the RL objective that modulates advantages by per-token KL divergence.

Together, these results reveal that RL improves model behavior not through widespread changes, but via sparse, structured, and high-leverage interventions, providing insights that deepen our understanding of RLVR and potentially inform more effective algorithm design.

2 Related Work

RLVR for Reasoning in LLMs. Reinforcement learning with verifiable rewards (RLVR) has become a central paradigm for enhancing reasoning in large language models, with many works extending the method or deepening its understanding (Chen et al., 2025; Wen et al., 2025; Yue et al., 2025; Liu et al., 2025; Shao et al., 2025). Recent studies further suggest that RL fine-tuning often acts as a *scalpel* rather than a hammer—amplifying existing capabilities through localized changes, in contrast to the broader changes induced by supervised finetuning (Rajani et al., 2025; Chu et al., 2025; Shenfeld et al., 2025). Our work focuses on analyzing token-level distributional shifts and their properties.

Token-Level and Entropy Analyses. A growing body of work investigates RLVR at the token level. Wang et al. (2025) find that high-entropy minority tokens account for much of RL's gains, while Cheng et al. (2025) associate them with exploratory reasoning steps and propose entropy-augmented rewards. Cui et al. (2025) warn of entropy collapse and introduce token-level clipping and KL penalty to stabilize training. Other studies (Vassoyan et al., 2025; Lin et al., 2025) point to the importance of critical tokens that disproportionately shape final responses. Karan & Du (2025) explore how base models can exhibit stronger reasoning capabilities through sampling strategies. Our analysis aligns with these perspectives, showing that distributional shifts are highly targeted, though not fully predictable from entropy alone. Finally, Huan et al. (2025) adopt a general token-level KL view, whereas we provide a more fine-grained quantification of these distribution shifts on the token-level.

Advantage Modulation. Several recent methods adjust the advantage signal to better focus updates on impactful tokens. Cheng et al. (2025) introduce entropy-based bonuses, while Cui et al. (2025) apply token-level gradient clipping to prevent overdominance. Yang et al. (2025) downweight low-probability tokens, isolating their influence, and Deng et al. (2025) reweight advantages based on perplexity and positional information. Wang et al. (2025) implicitly reweight updates by emphasizing forking tokens. Building on this line of work, we explore divergence-weighted advantages, explicitly scaling advantage weights by the magnitude of distributional change.

3 Token Distribution Analysis between Base and RL Models

We begin by analyzing the distributional shifts introduced by RLVR. To characterize token-level differences between the base model and the RL-finetuned model, we compare their next-token distributions under the same sequence contexts. Specifically, we take sequences generated by the RL policy and evaluate both models' conditional distributions at each position. This treats the RL output as a target trajectory, allowing us to measure how the base model would need to adapt to emulate it. Formally, for each token position t, let $\pi_{\text{base}}(\cdot \mid x_{<t})$ and $\pi_{\text{RL}}(\cdot \mid x_{<t})$ denote the respective distributions. A natural choice of discrepancy is the Kullback–Leibler (KL) divergence $D_{\text{KL}}\left(\pi_{\text{base}}(\cdot \mid x_{<t}) \parallel \pi_{\text{RL}}(\cdot \mid x_{<t})\right)$, or its reverse. However, due to practical limitations, such as memory constraints that prevent full distribution retrieval, there is the potential for the distributions to lack absolute continuity with respect to the other, in which case the KL divergence would be undefined. In addition, JS divergence is bounded, providing a normalized measurement that avoids the unboundedness of KL divergence that may skew our results.

We therefore adopt the Jensen–Shannon divergence (JSD), which is symmetric and bounded. For each token t, it is defined by

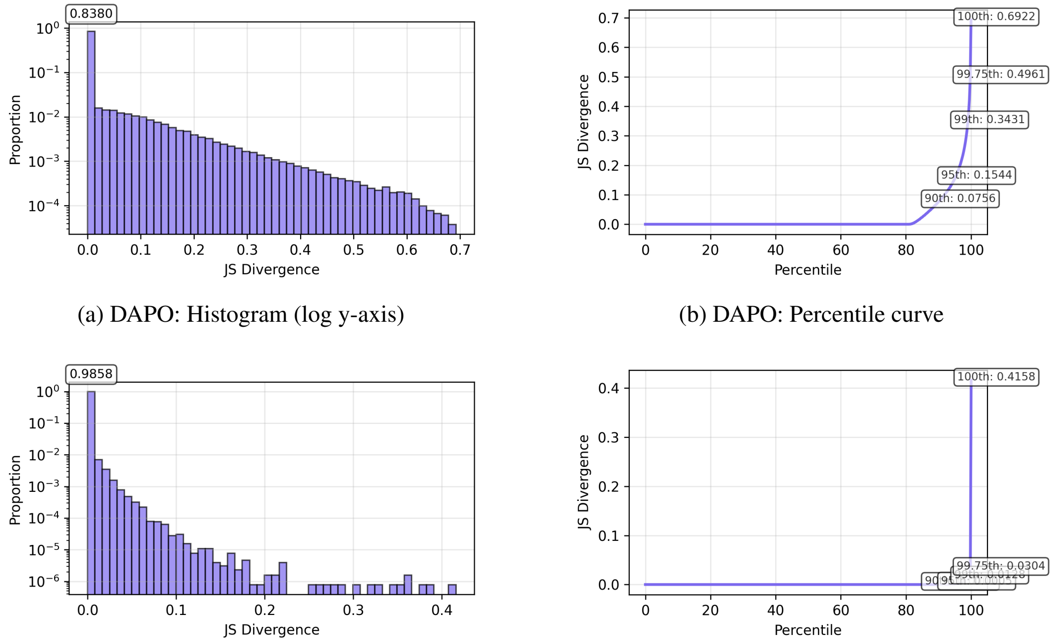
$$\mathrm{JS}_t = \tfrac{1}{2} D_{\mathrm{KL}}(\pi_{\mathrm{base}}(\cdot \mid x_{<t}) \parallel M_t) + \tfrac{1}{2} D_{\mathrm{KL}}(\pi_{\mathrm{RL}}(\cdot \mid x_{<t}) \parallel M_t)\,,$$

where $M_t = \frac{1}{2}\big(\pi_{\mathrm{base}}(\cdot \mid x_{<t}) + \pi_{\mathrm{RL}}(\cdot \mid x_{<t})\big)$. JSD is bounded in $[0, \log 2]$, enabling consistent comparison across positions and sequences.

In this section, we primarily use Qwen2.5-32B (Qwen et al., 2025) as the base model, and consider RLVR variants trained with DAPO (Yu et al., 2025) and GRPO, the latter paired with the corresponding SimpleRL model (Zeng et al., 2025). For evaluation on the AIME24 and AIME25 datasets, we sample 32 responses per problem. We also analyze additional models (Qwen2.5-Math-7B (Yang et al., 2024), Mistral-Small-24B (MistralAI, 2025)) and datasets (AIME 2025, GPQA (Rein et al., 2023), fine-tuning data), as well as comparisons with supervised fine-tuning approaches (see Appendix A.2 and Appendix A.4).

3.1 Distribution Shifts Are Highly Targeted and Sparse

A natural starting point is to ask: *how widely are distributional shifts spread across the output tokens?* To answer this, we examine the token-level Jensen–Shannon (JS) divergence between base and RL-trained models. Figure 1 shows histograms (log-scaled) and percentile curves for DAPO and SimpleRL.



(a) DAPO: Histogram (log y-axis)

(b) DAPO: Percentile curve

(c) SimpleRL: Histogram (log y-axis)

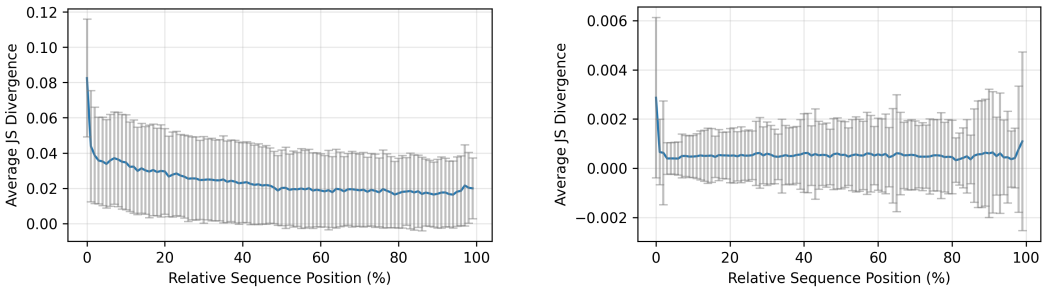
(d) SimpleRL: Percentile curve

Figure 1: JS divergence distributions for Qwen2.5 32B DAPO and SimpleRL on AIME 2024.

The distributions reveal that RL refinement is highly sparse. Under DAPO, over 83% of tokens show near-zero divergence, and this proportion rises to more than 98% under SimpleRL. The sharp peaks at zero and the steep ascent of the percentile curves indicate that only a small fraction of tokens are meaningfully altered. Comparing the two approaches, DAPO exhibits a heavier-tailed divergence distribution and a more gradual percentile curve, likely reflecting its clip-higher mechanism that encourages broader exploration. SimpleRL, by contrast, imposes stricter policy constraints and explicit KL regularization, leading to more concentrated updates. While differences in training data may also contribute, both methods ultimately achieve their gains through sparse and targeted modifications.

3.2 Positional Concentration

Beyond sparsity, we next ask: *where within a response, on average, do distributional shifts tend to occur?* Figure 2 plots the mean JS divergence by relative token position, with standard deviation, for DAPO and SimpleRL.



(a) DAPO

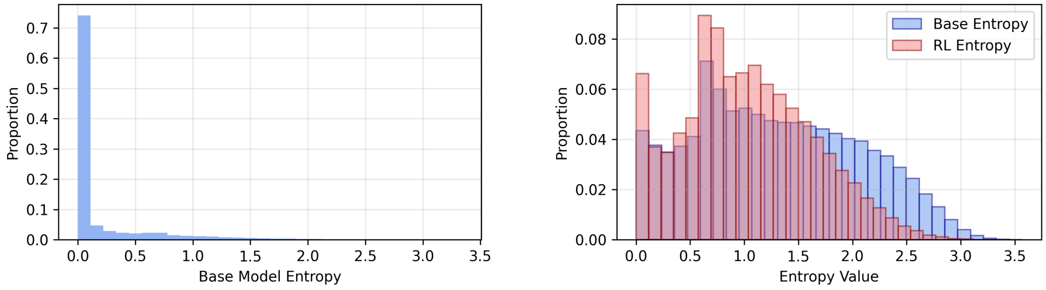
(b) SimpleRL

Figure 2: Mean JS divergence by normalized token position, with standard deviation. Both methods concentrate updates at the start and, to a lesser degree, at the end of responses.

Both methods show a clear positional structure: divergences are consistently high at the start of responses, decline in the middle, and modestly increase again toward the end. The early spikes likely reflect adjustments in initial planning and high-stakes decision points, while the end-of-sequence divergence corresponds to formatting and answer-boxing tokens. However, individual sequences still exhibit high divergences throughout the sequence, as indicated by the standard deviation bars. Overall, RL refinements are consistently concentrated at the boundaries of reasoning, but occur across the sequence for individual responses.

3.3 Divergence–Entropy Relationship

Next, we examine the relationship between divergence and predictive token-level entropy $H_t = -\sum_i \pi(i|x_{<t}) \log \pi(i|x_{<t})$., defined for each token t in the sequence x. While previous work suggests that RL may primarily adjust high-entropy (uncertain) predictions, leaving low-entropy (confident) predictions unchanged (Wang et al., 2025), we investigate this relationship by comparing the entropy distributions of base and RL models across token distributions with high and low divergence. Token distributions are grouped into low- and high-divergence bins (< 0.1 vs. > 0.1 JS), and we compare the entropy distributions of both base and RL models within each bin (Figures 3 and 15).



(a) Low JS bin (< 0.1).

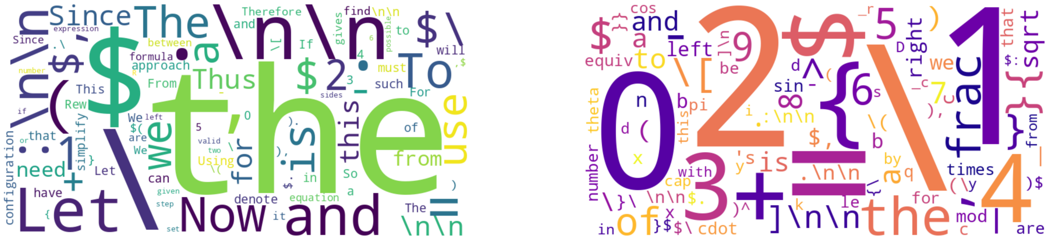
(b) High JS bin (> 0.1).

Figure 3: Entropy distributions across divergence bins for **DAPO**. Low-divergence tokens are generally low-entropy, while high-divergence tokens span both high- and low-entropy regions, indicating that DAPO can modify even confident predictions.

The results confirm that low-divergence tokens are almost always low-entropy. High-divergence tokens, however, span a broad entropy spectrum. DAPO modifies both high- and low-entropy predictions, indicating willingness to override even confident base outputs. By contrast, SimpleRL focuses changes in higher-entropy regions (though still with some lower-entropy distributions), following a more conservative approach that avoids modifying confident predictions.

3.4 Semantic Identity of Divergent Tokens

We next examine which types of token distributions tend to exhibit high versus low distributional divergence. Figure 4 visualizes representative examples using word clouds, with high-divergence tokens including common words, reasoning/problem-solving related terms, as well as equation fragments. On the other hand, low-divergence tokens are mainly numerals, operators, and equation components.



(a) Tokens with high JS divergence (JS > 0.1).

(b) Tokens with low JS divergence (JS < 0.01).

Figure 4: Word clouds of high and low divergence tokens under DAPO.

However, Figure 18 shows the full JS divergence distributions of frequently high and low divergence tokens, demonstrating that many tokens are not inherently divergent, and their behavior is context-dependent. For instance, the word “the” is among the highest frequency high divergence tokens, yet its full distribution of divergence values is mostly in the low regime. This suggests that token identity alone is insufficient to characterize divergence, and that a contextual perspective is essential.

4 Cross-Sampling: Functional Importance of Divergent Distributions

In the previous section, we showed that, conditioned on RL-generated sequences, only a small fraction of token distributions exhibit substantial shifts between the base and RL models. This observation motivates a fundamental question: are these sparse divergent token distributions directly responsible for the performance improvements of RLVR? Specifically, can we recover the RL model’s gains by generating primarily with π_{base} while introducing only a small number of tokens sampled from π_{RL}? On the other hand, does the RL model’s performance degrade when a small number of its tokens are replaced with tokens sampled from π_{base}?

To investigate this, we conduct controlled cross-sampling experiments that selectively swap token choices between the base model π_{base} and the RL-trained model π_{RL}. We employ two complementary interventions: (1) injecting RL tokens into base generations (*forward cross-sampling*), and (2) replacing RL tokens with base tokens (*reverse cross-sampling*). The general procedure for this is provided in Algorithm 1.

4.1 Experimental Setup

Forward Cross-Sampling. We investigate whether augmenting base model generations with small amounts of RL-sampled tokens can recover RL-level performance. Specifically, we generate sequences under π_{base} but sequentially replace tokens at positions where the divergence between π_{base} and π_{RL} exceeds a fixed threshold. At each such position, a token is sampled from π_{RL} instead, and we measure the resulting accuracy after completing the response with π_{base}.

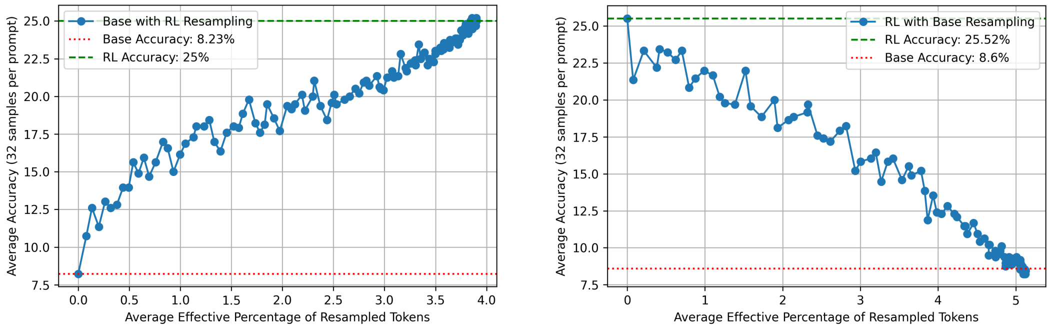
Reverse Cross-Sampling. To assess the robustness of the RL model, we invert the intervention: sequences are generated under π_{RL}, but at high-divergence positions, tokens are sampled from π_{base}.

This enables us to quantify the performance degradation when small amounts of RL tokens are replaced with base tokens in RL generations.

4.2 Results and Findings

Forward Cross-Sampling: A small proportion of RL-sampled tokens in base generations steadily increases performance to RL-levels. Figure 5a presents results for Qwen2.5-32B fine-tuned using SimpleRL on AIME24. Injecting fewer than 4% of RL-sampled tokens suffices to recover RL-level performance. Performance improves progressively with additional interventions, indicating that on average, each RL token systematically contributes to improved reasoning performance.

Reverse Cross-Sampling: A small amount of base-sampled tokens in RL generations progressively collapses performance to base-levels. As shown in Figure 5b, substituting even a small fraction of base tokens into RL generations rapidly collapses accuracy. Reverting as little as 5% of high-divergence tokens reduces performance to near base levels. The degradation is consistent, demonstrating that RL's gains depend critically on preserving its edits at these positions. Although substituted base tokens remain semantically valid (Figure 41), they progressively derail the reasoning trajectory.



(a) Forward cross-sampling

(b) Reverse cross-sampling

Figure 5: Cross-sampling results (Qwen2.5 32B SimpleRL on AIME24): injecting RL tokens into base generations progressively recovers RL accuracy, while reverting RL tokens with base tokens causes near-monotonic degradation toward base performance.

Table 1: Summary of cross-sampled tokens required to achieve approximate RL (forward) or Base (reverse) performance levels for Qwen2.5-32B on AIME24 and AIME25. Effective token counts exclude identity swaps during cross-sampling. Percentages are computed at the sequence level.

Dataset	**Method**	**Eff. % Tok.**	**% Tok.**	**Eff. # Tok.**	**# Tok.**	**Start Acc.**	**End Acc.**
AIME24	SimpleRL	3.86%	7.58%	38	75	8.23	> 25
	SimpleRL Rev.	5%	8.3%	29	51	25.52	< 8.3
	DAPO	7.8%	11.9%	280	410	8.23	> 44
	DAPO Rev.	10.1%	14.9%	173	258	44.8	< 8.5
AIME25	SimpleRL	1.53%	2.97%	13	26	5.3	> 14
	SimpleRL Rev.	4.73%	7.87%	31	53	12.71	< 4
	DAPO	6.47%	9.18%	230	326	4.8	> 33
	DAPO Rev.	9.89%	14.19%	181	261	32	< 4.5

Forward and reverse cross-sampling yield a consistent conclusion: the improvements from RL fine-tuning are concentrated in a sparse set of high-divergence tokens. The functional leverage of these positions demonstrates that RL refinement operates in a highly targeted manner, with performance gains critically dependent on preserving edits at these specific locations. Additional cross-sampling results, including detailed analyses and results for other model configurations, are provided in Appendix A.5.

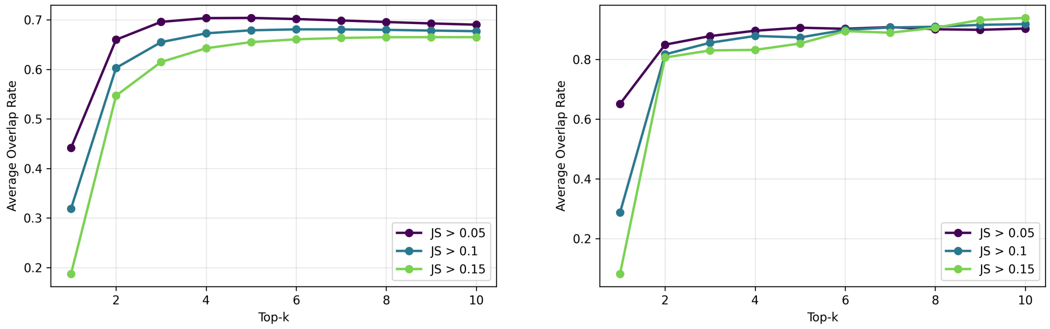
5 Fine-Grained Dynamics of Distribution Shifts

Having established the overall distributional changes, including their sparsity, positional concentration, and relationship to entropy, we now analyze *how* token-level distributions change at high-divergence positions. While our previous analyses quantified the extent and spread of these changes, as well as their relationships with entropy regimes, they did not reveal the detailed mechanics of how probability mass is redistributed within these critical positions. This finer-grained perspective reveals the mechanisms of RL-induced refinements, including overlap in candidate sets, rank shifts, and systematic probability adjustments.

5.1 Top-k Overlap and Rank Reordering

We study how RLVR modifies token predictions by analyzing overlap in top-k candidates and changes in their relative ranking. The central questions first consider (1) Do base and RL models consider similar candidate sets? (2) How much does their ordering shift?

Figure 6 reports the fraction of shared tokens between the top-k lists of the two models, restricted to positions with high JS divergence. Despite distributional shifts, overlap remains high for $k \geq 2$. SimpleRL exceeds 80% average overlap (often > 0.85), indicating that updates mostly reshuffle probabilities within a common set. DAPO shows slightly lower but still substantial overlap. Both methods exhibit a sharp jump from $k = 1$ to $k = 2$, suggesting that while the top-1 token often changes, the replacement was typically already among the base model's top-3.



(a) DAPO: Top-k overlap across thresholds.

(b) SimpleRL: Top-k overlap across thresholds.

Figure 6: Top-k token overlap between base and RL models at divergent positions (with $\mathrm{JS}_t > 0.1$). Computed as the size of the intersection divided by k. High overlap for $k \geq 2$ shows that distributional shifts occur mostly within shared candidate sets.

Re-ranking dynamics are shown in Figure 17, which locates the RL model's top-3 tokens in the base distribution (at high-divergence positions). About 30% of RL top-1 tokens were already ranked first in the base model; over 80% (DAPO) and 90% (SimpleRL) were in the base top-3. RL top-2 tokens generally fell within the base top-3–4, with SimpleRL showing stronger alignment.

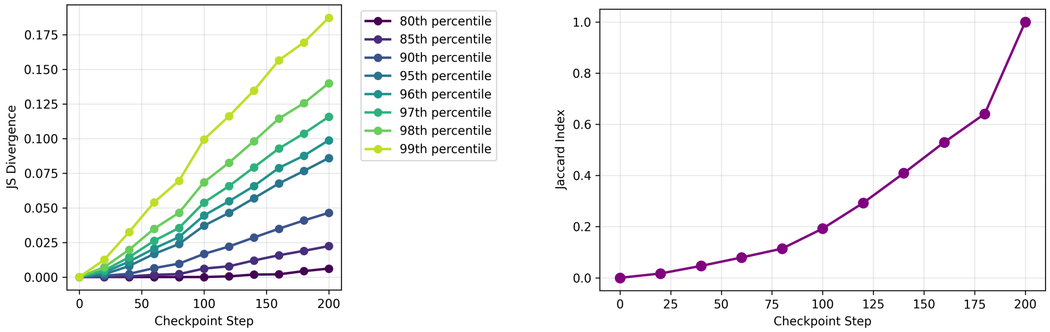
Overall, RL refinement acts primarily as a *re-ranking process*, elevating plausible alternatives already in the base shortlist. The main difference is selectivity: SimpleRL restricts updates to a narrow consensus of high-priority candidates, while DAPO permits broader rank shifts, consistent with its exploratory behavior.

5.2 Low Probability Behavior: Does RL Invent or Select?

We next ask whether RL encourages tokens that were highly unlikely under the base model. Concretely, for each divergent token distribution, we examine the top-1 token chosen by the RL-trained model and record its probability under the base distribution. We then measure the proportion of these tokens that fall below a given base-probability threshold. Results show that under DAPO, roughly 5% of divergent top-1 tokens have base probability < 0.01 (Figure 16), whereas under GRPO, this fraction is nearly zero (Figure 19). Thus, even in DAPO, which is designed to encourage broader exploration, RL still rarely promotes tokens that were very unlikely under the base policy.

5.3 Evolution Across Training

We analyze intermediate checkpoints of Qwen2.5-Math-7B (Yang et al., 2024) when trained with DAPO, conditioning all distributions on the final model's outputs. This alignment enables tracking of distributional changes over time for a fixed sequence of tokens. Figure 7 shows how JS divergence and divergent token sets evolve during training. JS divergence increases monotonically, with higher percentiles (95th, 99th) rising faster than lower ones (e.g., 80th), indicating that shifts are sparse and intensify over time. A small subset of tokens undergoes progressively stronger refinement, while most remain stable. On the other hand, the Jaccard index between each checkpoint's set of divergent tokens and the final set rises steadily, then jumps sharply near the end (Figure 7b).



(a) JS divergence percentiles.

(b) Jaccard index with final divergent set ($\text{JS}_t > 0.1$).

Figure 7: Distributional shifts grow increasingly focused and stable. Most tokens remain unchanged; updates concentrate in a sparse set late in training.

6 Exploratory Investigation: Divergence-Weighted Advantages

Our earlier analyses reveal that RL refinements are *sparse and targeted*, with only a small subset of tokens exhibiting meaningful distributional change. Moreover, cross-sampling experiments demonstrate that these high-divergence tokens are functionally critical, with performance gains hinging on precisely these positions. This raises a natural question: if only a small fraction of tokens drive improvements, can training be more effectively guided by modulating token-level learning signals according to these divergences? Intuitively, if certain tokens undergo substantial distributional shifts and are functionally important, explicitly weighting their advantages during training might amplify learning efficiency or stability. To investigate this possibility, we conduct a preliminary exploration of *divergence-weighted advantages* as a diagnostic intervention, where token-level advantages are reweighted by distributional divergence. We explore two different approaches: *high-KL boost*, which concentrates updates towards token distributions that are already changing substantially, and *low-KL boost*, which focuses updates on distributions that have changed less, potentially encouraging updates in previously stable regions.

6.1 GRPO-based Methods

GRPO in brief. GRPO (Shao et al., 2024) samples G responses $\{o_i\}_{i=1}^{G}$ from a policy $\pi_{\theta_{\text{old}}}(\cdot \mid q)$ for a prompt q with ground-truth answer a, assigns sequence-level rewards $\{R_i\}_{i=1}^{G}$, and computes a *group-normalized* advantage for each sample. GRPO then applies a PPO-style (Schulman et al., 2017) clipped surrogate objective at the *token* level, typically with an explicit KL penalty to a reference model.

DAPO. DAPO (Yu et al., 2025) modifies GRPO with an asymmetric clip-higher mechanism, dynamic sampling of correct/incorrect completions, token-level averaging, and removal of the explicit KL penalty term. Its objective is

$$\begin{aligned} J_{\text{DAPO}}(\theta) = \; & \mathbb{E}_{(q,a)\sim\mathcal{D},\, \{o_i\}_{i=1}^{G}\sim\pi_{\theta_{\text{old}}}(\cdot|q)} \\ & \left[\frac{1}{\sum_{i=1}^{G}|o_i|} \sum_{i=1}^{G}\sum_{t=1}^{|o_i|} \min\Big(r_{i,t}(\theta)\,\hat{A}_{i,t},\; \text{clip}\big(r_{i,t}(\theta),\, 1-\epsilon_{\text{low}},\, 1+\epsilon_{\text{high}}\big)\,\hat{A}_{i,t} \Big) \right] \\ & \text{s.t.} \quad 0 \,<\, \Big|\big\{\, o_i \;\big|\; \texttt{is\_equivalent}(a, o_i) \,\big\}\Big| \,<\, G, \end{aligned} \tag{1}$$

with

$$r_{i,t}(\theta) = \frac{\pi_\theta(o_{i,t} \mid q, o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t} \mid q, o_{i,<t})}, \quad \hat{A}_{i,t} = \frac{R_i - \text{mean}\big(\{R_j\}_{j=1}^{G}\big)}{\text{std}\big(\{R_j\}_{j=1}^{G}\big)}. \tag{2}$$

6.2 Exploratory Interventions: Divergence-Weighted Advantage

Standard RLVR objectives treat all tokens within a sequence uniformly in terms of their advantages. Motivated by our observation that distributional shifts are sparse and concentrated, we investigate whether modulating token-level advantages according to divergence magnitude can help improve or control aspects of training. We explore modifications where advantages are rescaled depending on the per-token divergences.

General formulation. We define a divergence-weighted advantage:

$$\tilde{A}_t = w_t \cdot \hat{A}_t, \tag{3}$$

where $\hat{A}_t$ denotes the standard group-normalized advantage and w_t is a per-token weight based on divergence. To ensure that the introduced divergence weight influences only the weighting and not the gradient computation, divergence values are detached from the computation graph.

Choice of divergence. We employ KL divergence with respect to the old policy as our primary divergence measure:

$$\text{KL}_t^{\text{old}} = D_{\text{KL}}(\pi_{\theta_{\text{old}}}(\cdot \mid x_{<t}) \parallel \pi_\theta(\cdot \mid x_{<t}))\,, \tag{4}$$

where $\pi_{\theta_{\text{old}}}$ denotes the policy from the previous update iteration, as in PPO/GRPO. This old-policy KL quantifies the magnitude of recent policy updates at each token position, serving as a proxy for the extent of local distributional change. For computational efficiency and compatibility with existing training frameworks such as verl (Sheng et al., 2024), we estimate these quantities using KL estimators computed over sampled tokens only, which may not capture the full distributional structure. Alternative divergences signals, including reference-based KL, are discussed in Appendix A.6.

Weighting schemes. We adopt a simple sigmoid weighting scheme (to ensure bounded weights), which transforms divergence into weights through:

$$w_t = 1 + s\,(\sigma(\alpha \cdot \text{KL}_t) - 0.5)\,, \quad \sigma(x) = \tfrac{1}{1+e^{-x}}\,. \tag{5}$$

The parameter α controls the direction and magnitude of emphasis: $\alpha > 0$ amplifies high-divergence tokens, whereas $\alpha < 0$ emphasizes low-divergence ones. The sigmoid function provides a smooth, bounded nonlinear transformation that enables selective focus on either high- or low-divergence regions depending on the sign of α. This formulation allows us to investigate whether concentrating the learning signal on regions that have already changed or those that remain unchanged yields more effective training dynamics. Alternative weighting schemes, including linear relative weighting, are dicussed in Appendix A.6.

Evaluation. We evaluate divergence-weighted advantages using the DAPO training recipe and data on Qwen2.5-Math-7B. Results are presented in Table 2. In the main text, we focus on configurations employing KL divergence with respect to $\pi_{\theta_{\text{old}}}$ and the sigmoid weighting scheme. Additional possible configurations are discussed in Appendix A.6. Detailed training hyperparameters and implementation details are documented in Appendix A.3.3.

Table 2: Accuracy (%) under divergence-weighted configurations on Qwen2.5-Math-7B. Results shown for KL divergence with $\pi_{\theta_{\text{old}}}$ and sigmoid weighting scheme across AIME24, AIME25, and AMC datasets. The results displayed are the avg@32 scores (or the pass@1 scores computed using 32 samples). Results are each averaged over 3 runs.

Configuration	**AIME24**	**AIME25**	**AMC**	**Overall Avg**
Baseline DAPO	33.61	18.75	75.08	42.48 ± 1.35
Low-KL boost	35.90	19.90	78.97	44.92 ± 0.05
High-KL boost	36.74	20.00	78.40	45.05 ± 0.79

These results demonstrate that weighting token-level updates by divergence can amplify performance gains, providing empirical support for the hypothesis that targeted tokens disproportionately drive improvements. Both low-KL and high-KL boost configurations yield improvements over the baseline, suggesting that different divergence weighting strategies can be effective. However, the optimal choice between these approaches, and indeed whether divergence weighting provides benefits at all, may depend on the specific models and training methods used. Effective divergence weighting across training configurations may require model-specific paradigms or adaptive scheduling mechanisms to stabilize learning dynamics. We present this approach as a complementary diagnostic tool that may inform future refinements of token-level training strategies.

7 Conclusion

Our study reveals that reinforcement learning with verifiable rewards (RLVR) reshapes LLMs in a manner that is sparse, targeted, and structured rather than uniformly diffused across tokens. By analyzing token-level distributional shifts, we show that only a small subset of tokens undergo meaningful divergence, and that these divergences carry disproportionate functional importance: cross-sampling interventions confirm that performance gains hinge on precisely these positions. To complement these analyses, we explored divergence-weighted advantage, a simple modification that scales token-level advantages by per-token divergence. These results suggest that weighting strategies can influence learning dynamics, though stabilizing performance may require model-specific choices or schedulers.

Together, these findings advance a token-level understanding of RL fine-tuning. They highlight that the essence of RLVR's success lies not in widespread distributional changes, but in selective refinements aligned with varying entropy levels. Beyond clarifying the mechanics of existing methods, our work offers a perspective for designing future RL objectives that explicitly incorporate distributional structure, opening avenues for more effective, interpretable, and controllable LLM post-training.

References

Zhipeng Chen, Xiaobo Qin, Youbin Wu, Yue Ling, Qinghao Ye, Wayne Xin Zhao, and Guang Shi. Pass@k training for adaptively balancing exploration and exploitation of large reasoning models, 2025. URL https://arxiv.org/abs/2508.10751.

Daixuan Cheng, Shaohan Huang, Xuekai Zhu, Bo Dai, Wayne Xin Zhao, Zhenliang Zhang, and Furu Wei. Reasoning with exploration: An entropy perspective on reinforcement learning for llms, 2025. URL https://arxiv.org/abs/2506.14758.

Tianzhe Chu, Shengbang Tong, Jihan Yang, Tianzhe Chu, Yuexiang Zhai, Yi Ma, Saining Xie, Dale Schuurmans, Quoc V. Le, and Sergey Levine. Sft memorizes, rl generalizes: A comparative study of foundation model post-training, 2025. URL https://arxiv.org/abs/2501.17161.

Ganqu Cui, Yuchen Zhang, Jiacheng Chen, Lifan Yuan, Zhi Wang, Yuxin Zuo, Haozhan Li, Yuchen Fan, Huayu Chen, Weize Chen, Zhiyuan Liu, Hao Peng, Lei Bai, Wanli Ouyang, Yu Cheng, Bowen Zhou, and Ning Ding. The entropy mechanism of reinforcement learning for reasoning language models, 2025. URL https://arxiv.org/abs/2505.22617.

Jia Deng, Jie Chen, Zhipeng Chen, Daixuan Cheng, Fei Bai, Beichen Zhang, Yinqian Min, Yanzipeng Gao, Wayne Xin Zhao, and Ji-Rong Wen. From trial-and-error to improvement: A systematic analysis of llm exploration mechanisms in rlvr, 2025. URL https://arxiv.org/abs/2508.07534.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration, 2020. URL https://arxiv.org/abs/1904.09751.

Maggie Huan, Yuetai Li, Tuney Zheng, Xiaoyu Xu, Seungone Kim, Minxin Du, Radha Poovendran, Graham Neubig, and Xiang Yue. Does math reasoning improve general llm capabilities? understanding transferability of llm reasoning, 2025. URL https://arxiv.org/abs/2507.00432.

Aayush Karan and Yilun Du. Reasoning with sampling: Your base model is smarter than you think, 2025. URL https://arxiv.org/abs/2510.14901.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, et al. T\" ulu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*, 2024.

Zicheng Lin, Tian Liang, Jiahao Xu, Qiuzhi Lin, 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, 2025. URL https://arxiv.org/abs/2411.19943.

Mingjie Liu, Shizhe Diao, Ximing Lu, Jian Hu, Xin Dong, Yejin Choi, Jan Kautz, and Yi Dong. Prorl: Prolonged reinforcement learning expands reasoning boundaries in large language models, 2025. URL https://arxiv.org/abs/2505.24864.

MistralAI. Mistral small 3, January 2025. URL https://mistral.ai/news/mistral-small-3.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025. URL https://arxiv.org/abs/2412.15115.

Neel Rajani, Aryo Pradipta Gema, Seraphina Goldfarb-Tarrant, and Ivan Titov. Scalpel vs. hammer: Grpo amplifies existing capabilities, sft replaces them, 2025. URL https://arxiv.org/abs/2507.10616.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. Gpqa: A graduate-level google-proof q&a benchmark, 2023. URL https://arxiv.org/abs/2311.12022.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv*, abs/1707.06347, 2017. doi: 10.48550/arXiv.1707.06347. URL https://arxiv.org/abs/1707.06347.

Rulin Shao, Shuyue Stella Li, Rui Xin, Scott Geng, Yiping Wang, Sewoong Oh, Simon Shaolei Du, Nathan Lambert, Sewon Min, Ranjay Krishna, Yulia Tsvetkov, Hannaneh Hajishirzi, Pang Wei Koh, and Luke Zettlemoyer. Spurious rewards: Rethinking training signals in rlvr, 2025. URL https://arxiv.org/abs/2506.10947.

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

Idan Shenfeld, Jyothish Pari, and Pulkit Agrawal. Rl's razor: Why online reinforcement learning forgets less, 2025. URL https://arxiv.org/abs/2509.04259.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. *arXiv preprint arXiv: 2409.19256*, 2024.

Jean Vassoyan, Nathanaël Beau, and Roman Plaud. Ignore the kl penalty! boosting exploration on critical tokens to enhance rl fine-tuning, 2025. URL https://arxiv.org/abs/2502.06533.

Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xionghui Chen, Jianxin Yang, Zhenru Zhang, Yuqiong Liu, An Yang, Andrew Zhao, Yang Yue, Shiji Song, Bowen Yu, Gao Huang, and Junyang Lin. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for llm reasoning, 2025. URL https://arxiv.org/abs/2506.01939.

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

Taiqiang Wu, Runming Yang, Jiayi Li, Pengfei Hu, Ngai Wong, and Yujiu Yang. Shadow-ft: Tuning instruct via base, 2025. URL https://arxiv.org/abs/2505.12716.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, Keming Lu, Mingfeng Xue, Runji Lin, Tianyu Liu, Xingzhang Ren, and Zhenru Zhang. Qwen2.5-math technical report: Toward mathematical expert model via self-improvement, 2024. URL https://arxiv.org/abs/2409.12122.

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, 2025. URL https://arxiv.org/abs/2505.12929.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Mu Qiao, Yonghui Wu, and Mingxuan Wang. Dapo: An open-source llm reinforcement learning system at scale, 2025. URL https://arxiv.org/abs/2503.14476.

Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model?, 2025. URL https://arxiv.org/abs/2504.13837.

Weihao Zeng, Yuzhen Huang, Qian Liu, Wei Liu, Keqing He, Zejun Ma, and Junxian He. Simplerl-zoo: Investigating and taming zero reinforcement learning for open base models in the wild, 2025. URL https://arxiv.org/abs/2503.18892.

A Appendix

A.1 Weight-Level analysis of Changes

Orthogonal to the analysis done in the main text, we also investigate the degree of modifications induced by RLVR at the parameter level. More specifically, we employ the relative gap ratio (Wu et al., 2025), denoted as σ, to quantify the magnitude of weight divergence pre- and post-fine-tuning. This ratio is formulated as:

$$\sigma = \frac{\sum |W_{\text{original}} - W_{\text{tuned}}|}{\sum |W_{\text{original}}| + \sum |W_{\text{tuned}}|}$$

where W_{original} and W_{tuned} represent the model parameters before and after fine-tuning, respectively. A lower σ value signifies greater similarity between the parameter sets, indicating a smaller overall modification from the fine-tuning process.

In our experiment, we utilized the Qwen2.5-32B and Qwen2.5-Math-7B models as foundations. Each model was independently fine-tuned via two distinct methodologies: RL and Supervised Fine-Tuning (SFT). To ensure a controlled and equitable comparison, the training regimen for both methods was standardized, employing an identical dataset size and the same number of training steps. Subsequently, the σ was computed between each original model and its corresponding tuned counterparts. The results are presented in the following table.

Table 3: Relative gap ratio (σ) after RL and SFT fine-tuning.

Model	**Qwen2.5-32B**	**Qwen2.5-Math-7B**
σ after RL	0.00143	0.00136
σ after SFT	0.00347	0.00944

The results presented in the table demonstrate a consistent trend across both models: the σ values corresponding to RL fine-tuning are substantially lower than those from SFT. This quantitative analysis at the parameter level suggests that the cumulative weight modifications induced by RL are significantly less extensive than those resulting from SFT. This finding provides empirical support for the hypothesis that RL achieves performance gains through sparse and targeted parameter adjustments, contrasting with the more distributed updates characteristic of SFT.

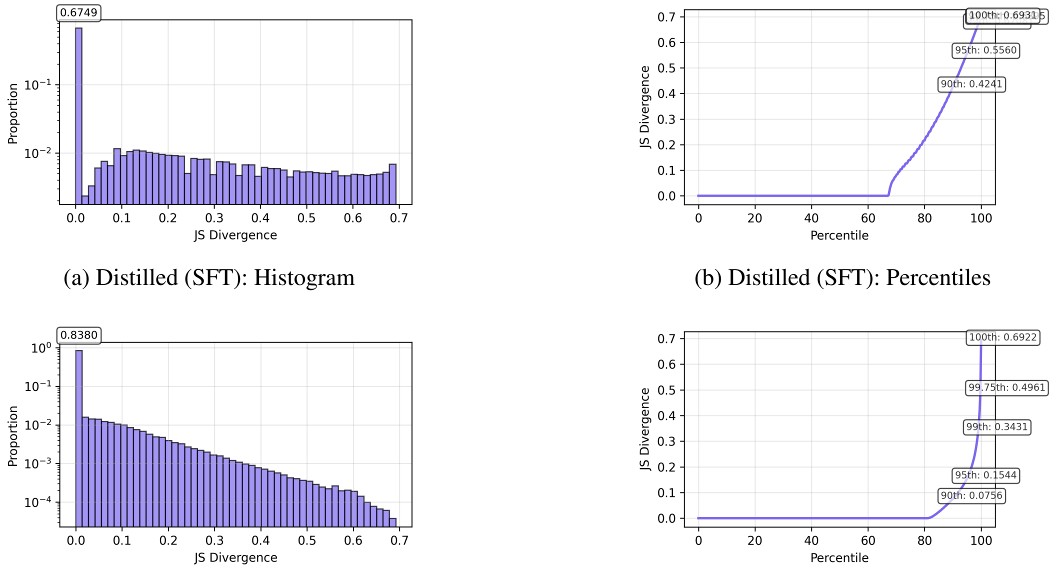
A.2 RLVR vs. Supervised Fine-Tuning: Contrasting Distributional Patterns

A natural question is whether the sparse, targeted distributional shifts we observe are specific to RLVR, or if they also characterize other fine-tuning approaches. To address this, we compare RLVR-trained models with models refined through supervised fine-tuning (SFT). We analyze Qwen2.5-32B trained with SFT alongside Qwen2.5-32B DAPO.

Figure 8 shows JS divergence distributions for both approaches. SFT produces a noticeably larger high-divergence set, whereas RLVR concentrates almost all token distributions below very small JS values. This directly reflects RLVR's extreme selectivity and the broader edits introduced by SFT. The top-k overlap analysis (Figure 11) highlights that SFT consistently achieves lower overlap with the base model, indicating more aggressive re-ranking, while RLVR largely stays within the base model's existing candidate set. The rank reordering analysis (Figure 12) further shows that SFT promotes many more tokens far outside the base model's top-3, whereas RLVR mainly promotes candidates that were already high-ranked.

Taken together, the metrics highlight that SFT diverges from RLVR along several axes. The SFT model exhibits higher median and tail JS divergence as well as a larger mass of high-divergence tokens (Figure 8), and attains lower top-k overlap with the base model (Figure 11) alongside larger rank shifts (Figure 12). Moreover, SFT's divergent tokens concentrate on low-entropy regions and more frequently elevate low base-probability choices (Figures 13 and 9), whereas RLVR keeps most divergent tokens within higher-entropy, already plausible candidates. These differences reinforce that RLVR acts as a targeted editor, while SFT drives broader, less selective reshaping of the distribution.

These findings align with recent work suggesting that RL fine-tuning acts as a *scalpel* rather than a hammer, making sparse, targeted changes compared to the broader modifications induced by supervised fine-tuning (Rajani et al., 2025; Chu et al., 2025). The key difference lies in the *token-level distributional changes*: RLVR modifies far fewer token positions (as measured by JS divergence), and at those positions, the changes are more likely to be re-ranking within the base model's top candidates rather than introducing entirely new token probabilities. In contrast, SFT-based distillation exhibits more widespread token-level distributional shifts across a larger fraction of positions, as it learns to mimic provided outputs by adjusting token probabilities more broadly across the vocabulary space.



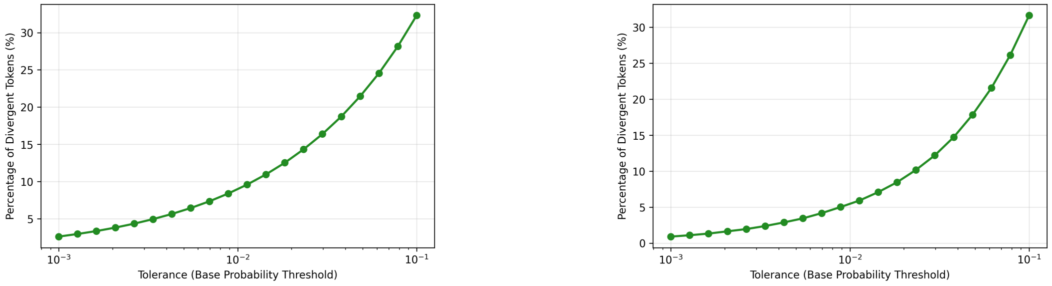
(a) Distilled (SFT): Histogram

(b) Distilled (SFT): Percentiles

(c) RLVR (DAPO): Histogram

(d) RLVR (DAPO): Percentiles

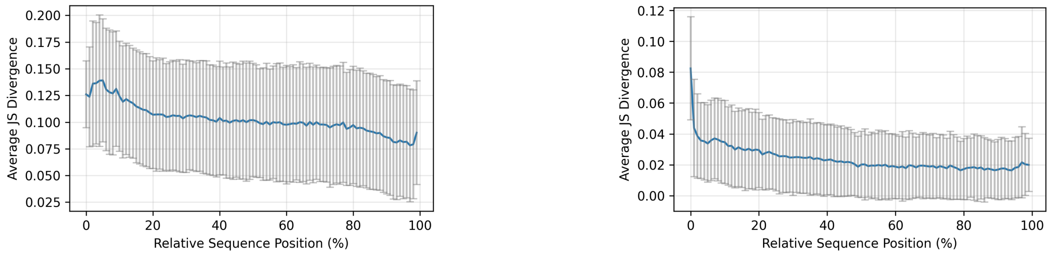
Figure 8: JS divergence distributions comparing supervised fine-tuning (distillation) and RLVR on AIME 2024. RLVR exhibits even sparser distributional shifts than SFT-based distillation, suggesting more targeted refinement.



(a) Qwen2.5-32B SFT

(b) Qwen2.5-32B+DAPO

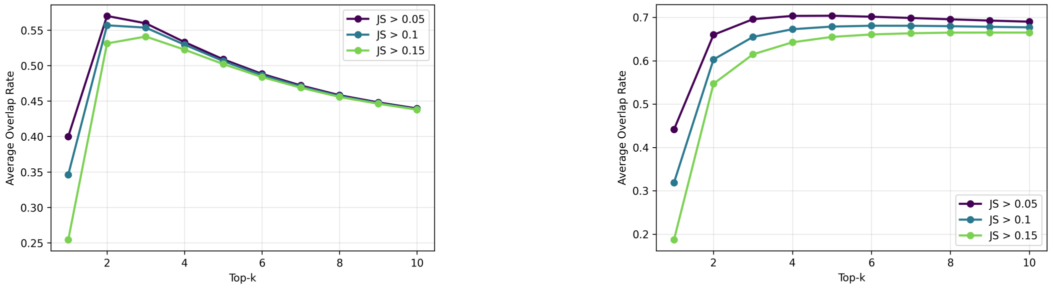
Figure 9: Percentage of divergent tokens whose RL top-1 choice had base probability below a given threshold comparing distilled and RLVR-trained models on AIME 2024.



(a) Qwen2.5-32B SFT

(b) Qwen2.5-32B+DAPO

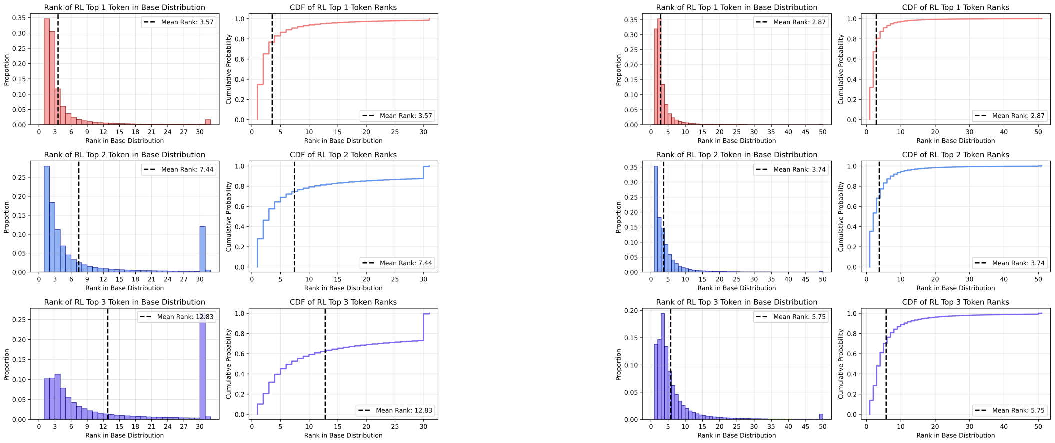
Figure 10: Mean JS divergence by normalized token position comparing distilled and RLVR-trained models on AIME 2024. The positional patterns reveal differences in how distillation and RLVR concentrate their updates.



(a) Qwen2.5-32B SFT

(b) Qwen2.5-32B+DAPO

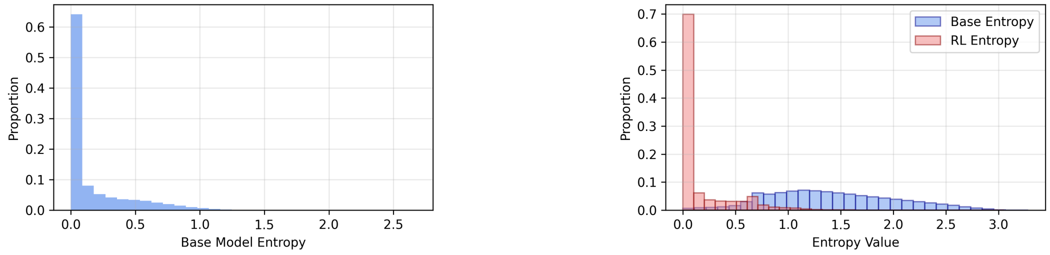
Figure 11: Top-k token overlap between base and refined models at divergent positions ($\text{JS}_t > 0.1$) comparing distilled and RLVR-trained models on AIME 2024.



(a) Qwen2.5-32B SFT

(b) Qwen2.5-32B+DAPO

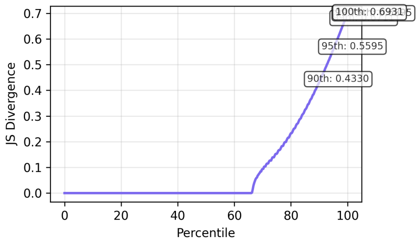
Figure 12: Distribution of base-model ranks for refined models' top-3 tokens at high-divergence positions ($\text{JS} > 0.1$) comparing distilled and RLVR-trained models on AIME 2024.



(a) Qwen2.5-32B SFT Low JS bin (< 0.1)

(b) Qwen2.5-32B SFT High JS bin (> 0.1)

Figure 13: Entropy distributions across divergence bins using full vocabulary for Qwen2.5-32B-distill on AIME 2025. Patterns are consistent with those observed in the main text, confirming the relationship between entropy and divergence for SFT-based distillation.



(a) Qwen2.5-32B SFT AIME 2025: Percentiles

Figure 14: JS divergence distributions for Qwen2.5-32B-distill on AIME 2025. Consistent patterns with AIME 2024 demonstrate robustness across datasets.

A.3 Experimental Details

Code will be released publicly upon acceptance.

A.3.1 Token Analysis

We run model inference using `vllm` (Kwon et al., 2023). On AIME, we apply nucleus sampling (Holtzman et al., 2020) with topp $= 0.7$ and temperature $= 1$. For divergence calculations on AIME, we use the top-p truncated distribution to reflect the effective sampling distribution, to provide a more accurate estimate for our cross-sampling experiments. We also look at the distribution of JS divergence values of the distribution without truncation to ensure the results are not affected much by the truncation. For experiments on the fine-training data, we use topp $= 1$ to reflect the training sampling distribution.

For token-level distributional analysis, we evaluate multiple model configurations across different datasets. On **Qwen2.5-32B**, we analyze distributional shifts on AIME 2024 and AIME 2025 for DAPO, SimpleRL, and SFT (we train the SFT model as outlined in Section A.3.3). On **Mistral-Small-24B**, we perform token analysis on AIME 2024 and AIME 2025 using SimpleRL. For **Qwen2.5-Math-7B** (trained as outlined in Section A.3.3), we analyze distributional shifts on AIME 2024, AIME 2025, and post-training data using both DAPO with the default clip-higher setting and with clip-higher=0.2. We also

A.3.2 Cross-Sampling

For cross-sampling experiments, we use the same inference setup as token analysis. Cross-sampling experiments selectively swap tokens between base and RL models at positions where JS divergence exceeds a threshold, allowing us to measure the functional importance of divergent token distributions.

We perform forward and reverse cross-sampling experiments on the following model-dataset combinations. For forward cross-sampling, we inject RL-sampled tokens into base generations at positions where JS divergence exceeds the specified threshold. For reverse cross-sampling, we replace RL tokens with base tokens at high-divergence positions on the RL generations. The divergence thresholds used for each configuration are as follows:

- **Qwen2.5-32B + SimpleRL:**
  - AIME 2024: Forward threshold JS > 0.03, Reverse threshold JS > 0.05
  - AIME 2025: Forward threshold JS > 0.05, Reverse threshold JS > 0.05
- **Qwen2.5-32B + DAPO:**
  - AIME 2024: Forward threshold JS > 0.08, Reverse threshold JS > 0.06
  - AIME 2025: Forward threshold JS > 0.1, Reverse threshold JS > 0.08
- **Mistral-Small-24B + SimpleRL:**
  - AIME 2024: Forward threshold JS > 0.002, Reverse threshold JS > 0.02

A.3.3 Additional Training Details

We implement RLVR training experiments using `verl` (Sheng et al., 2024) with the standard DAPO recipe (Yu et al., 2025).

Qwen2.5-Math-7B DAPO Training. We follow the public DAPO recipe, namely with clip ratios $\epsilon_{\text{low}} = 0.2$ and $\epsilon_{\text{high}} = 0.28$. However, for token analysis, we also train a variant with $\epsilon_{\text{high}} = 0.2$ for comparison. We optimize with learning rate 1×10^{-6}, a 10-step warmup using AdamW, and no explicit reference-KL penalty. Each RLVR step processes 512 prompts with 16 sampled responses per prompt; these are split into mini-batches of 32 prompts, yielding 16 gradient updates per RLVR step. Maximum generation length and the overlong-penalty threshold are set to 8k and 4k tokens.

Supervised Fine-Tuning (SFT) Training. For the SFT model based on on Qwen2.5 32B, we sampled 42k instances from the `AM-DeepSeek-R1-Distilled-1.4M` dataset. The model underwent full parameter fine-tuning for 5 epochs, employing DeepSpeed ZeRO-3 optimization.

For the divergence-weighted advantage experiments on **Qwen2.5-Math-7B**, under the **high-KL** setting we use $s = 0.3$ and set α to increase linearly from 0 to 50 starting at step 100. In the **low-KL** setting, we use $s = 0.3$ and set α to increase linearly from 0 to 50, which we linearly increase beginning at step 150.

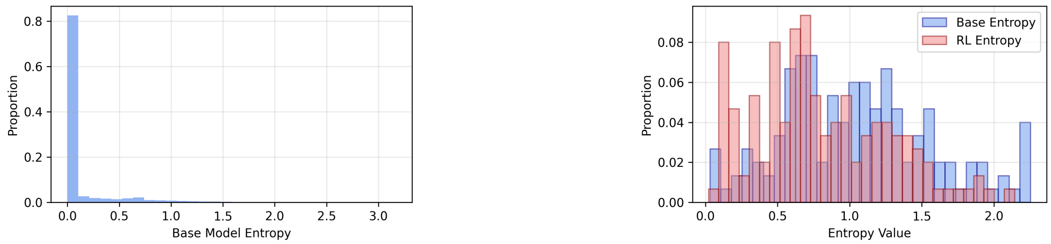
For **Qwen2.5-7B**, in the high-KL relative setting we set $\alpha = 4$. In the configuration with an additional scheduler, we initialize $\alpha = 2$ and linearly increase it to 3 from step 80 onward.

A.4 Additional Token Distribution Analyses

This section provides supplementary and extended token distribution analyses. We first present supplementary figures for the main models (Qwen2.5-32B with DAPO and SimpleRL on AIME 2024), then extend the analysis to additional models and datasets to demonstrate the generalizability of our findings.

A.4.1 Supplementary Figures for Main Models

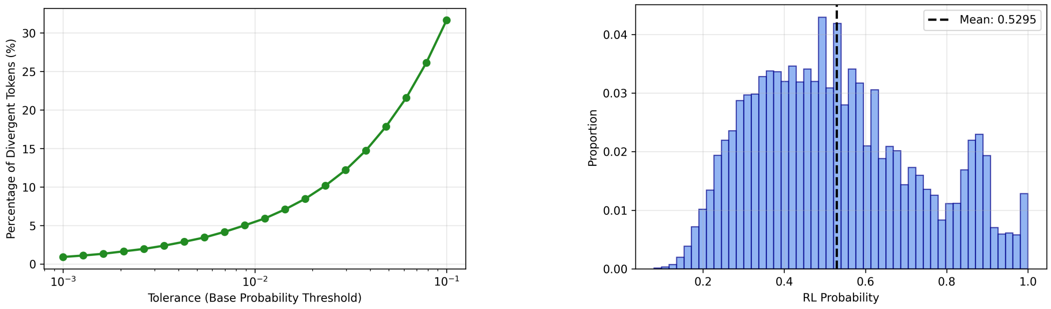
We provide additional figures for Qwen2.5-32B with DAPO and SimpleRL that complement the analyses in the main text.



(a) Low JS bin (< 0.1).

(b) High JS bin (> 0.1).

Figure 15: Entropy distributions across divergence bins for **SimpleRL**. Low-divergence tokens are mostly low-entropy, while high-divergence tokens are concentrated in higher-entropy regions, reflecting a more conservative update strategy.



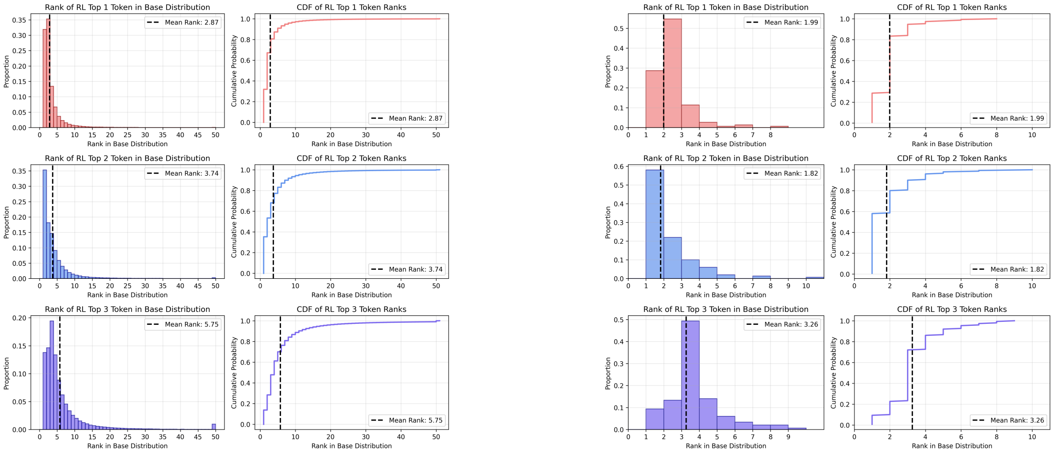
(a) Percentage of divergent tokens with low base probability.

(b) Histogram of RL probabilities for low base-probability tokens.

Figure 16: Analysis of tail behavior under DAPO for divergent token distributions ($\text{JS} > 0.1$). **(a)** shows the fraction of divergent tokens whose RL top-1 choice had base probability below a given threshold. **(b)** shows the distribution of RL probabilities for the subset with base probability < 0.01.

To supplement the positional analysis in the main text, we also examine localized averages of JS divergence near the start of the generation and near the final answer span.

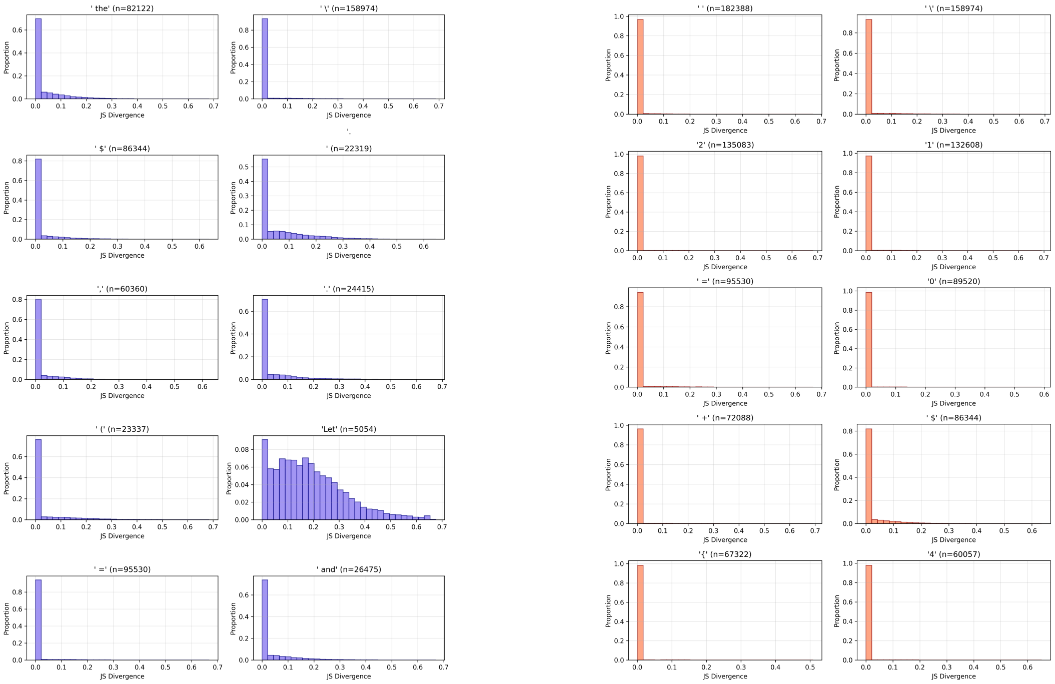
Results on GPQA-Diamond. We extend our analysis to GPQA-Diamond to demonstrate the generalizability of our findings across different reasoning benchmarks. Figure 23 shows JS divergence percentile curves and positional concentration for Qwen2.5-32B with DAPO on GPQA-Diamond, revealing consistent sparsity patterns. Figure 24 shows entropy distributions across divergence bins.



(a) Qwen2.5 32B DAPO.

(b) Qwen2.5 32B SimpleRL.

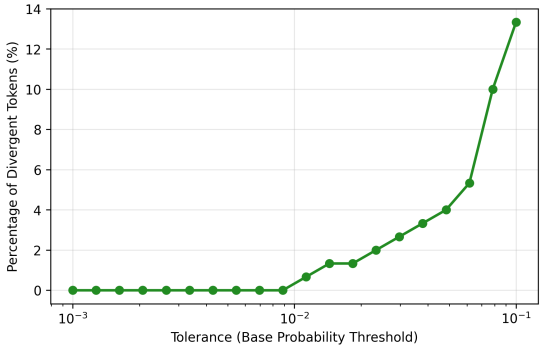
Figure 17: Distribution of base-model ranks for RL's top-3 tokens at high-divergence positions (JS > 0.1). Most RL-selected tokens were already highly ranked in the base model, especially under SimpleRL.



(a) Frequent high JS tokens.

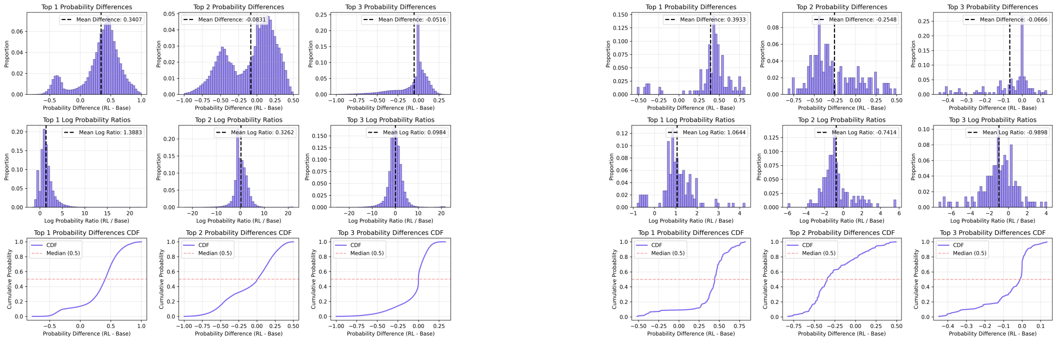
(b) Frequent low JS tokens.

Figure 18: Histogram of divergences for frequent high JS tokens and frequent low JS tokens (Qwen2.5 32B with DAPO).



(a) SimpleRL

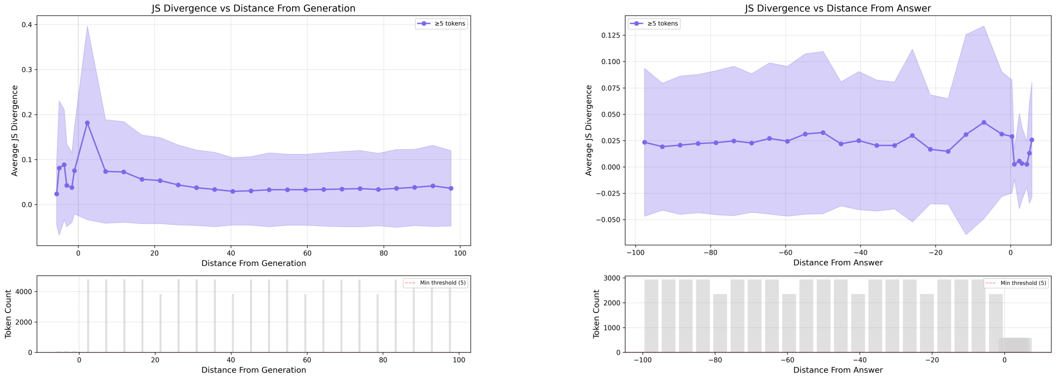
Figure 19: Percentage of divergent tokens whose RL top-1 choice had base probability below a given threshold.



(a) DAPO.

(b) SimpleRL.

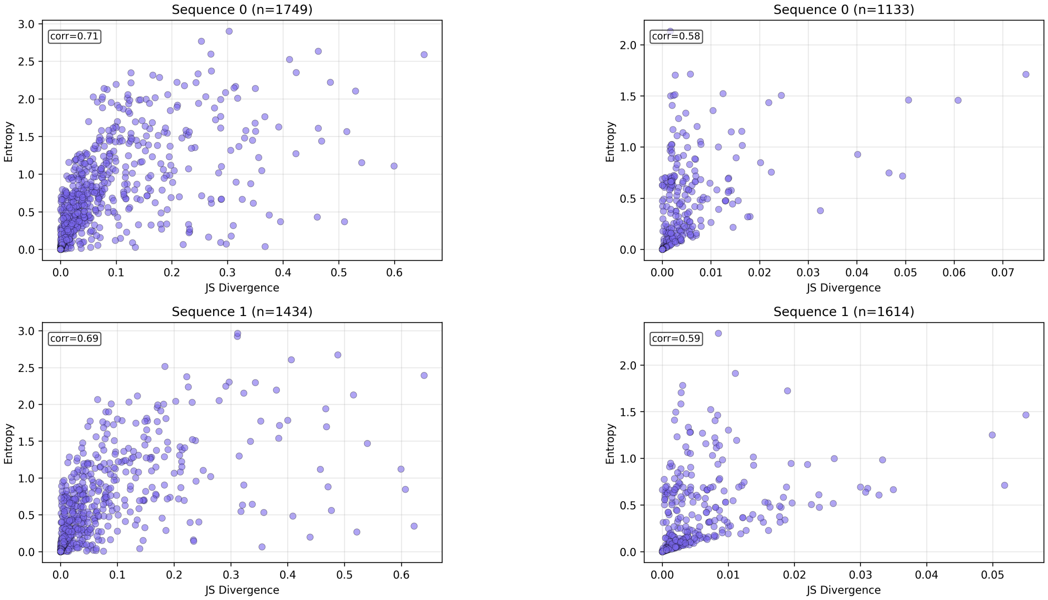
Figure 20: Probability differences and ratios for top-3 tokens under DAPO and SimpleRL among divergent distributions (JS > 0.1).



(a) Near the start of generation

(b) Near the answer span

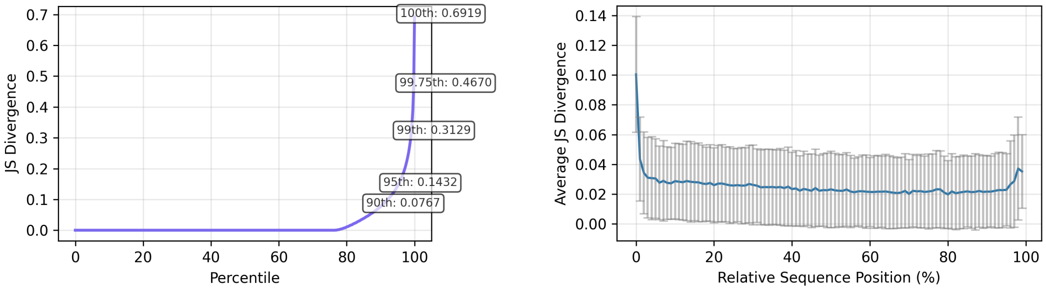
Figure 21: Local averages of JS divergence as a function of distance from key regions (prompt beginning and answer) for Qwen2.5-32B models on AIME 2024. Divergence peaks occur in the same early and late windows highlighted by the positional analysis.



(a) Qwen2.5-32B DAPO

(b) Qwen2.5-32B SimpleRL

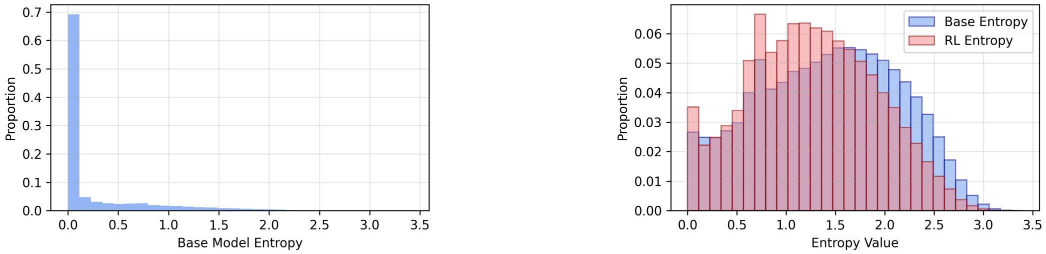
Figure 22: Per-sequence scatter plots relating entropy to JS divergence for Qwen2.5-32B DAPO and SimpleRL on AIME 2024. DAPO exhibits a broader entropy spread among divergent tokens, whereas SimpleRL concentrates divergence in higher-entropy regions.



(a) JS divergence percentiles

(b) Positional concentration

Figure 23: JS divergence analysis for Qwen2.5-32B with DAPO on GPQA-Diamond. The sparsity patterns and positional concentration are consistent with findings on AIME datasets.

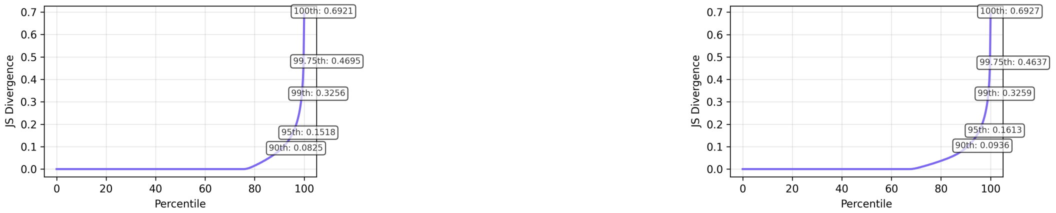


(a) Low JS bin (< 0.1).

(b) High JS bin (> 0.1).

Figure 24: Entropy distributions across divergence bins for Qwen2.5-32B with DAPO on GPQA-Diamond. Patterns are consistent with those observed on AIME datasets.

Effect of Top-p Sampling on JS Divergence. To verify that our findings are robust to different top-p sampling settings, we compare JS divergence distributions across different sampling configurations. The default setting uses top-$p = 0.7$ for sampling. We also evaluate configurations where sampling is performed with top-$p = 0.8$ and top-$p = 0.9$. Figure 25 shows that the sparsity patterns remain consistent across different sampling top-p values, confirming that our results are not sensitive to the specific sampling top-p value used.

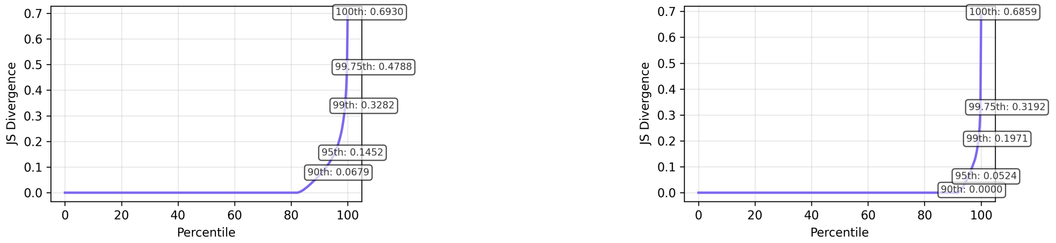


(a) Sampling top-$p = 0.8$

(b) Sampling top-$p = 0.9$

Figure 25: JS divergence percentile curves for Qwen2.5-32B with DAPO on AIME 2024 under different top-p sampling settings. The sparsity patterns remain consistent across different sampling top-p values, indicating robustness to the specific sampling configuration.

JS Divergence on AIME 2025. Figure 26 shows JS divergence percentile curves for Qwen2.5-32B with DAPO and SimpleRL on AIME 2025, demonstrating consistent sparsity patterns across datasets.



(a) DAPO: Percentile curve

(b) SimpleRL: Percentile curve

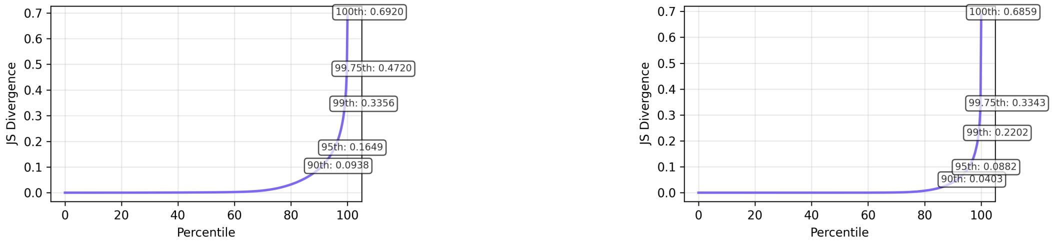
Figure 26: JS divergence distributions for Qwen2.5-32B with DAPO and SimpleRL on AIME 2025. The sparsity patterns are consistent with those observed on AIME 2024, confirming the robustness of our findings across datasets.

Effect of Top-p Truncation on JS Divergence. To verify that our use of top-p truncated distributions (with topp $= 0.7$) does not significantly impact our findings, we compare JS divergence distributions computed using the estimated full distribution (top-$p = 1$) with those using truncated distributions. Figure 27 shows that the patterns remain consistent: distributional shifts are highly sparse regardless of truncation, with the vast majority of tokens showing near-zero divergence.

A.4.2 Comparison of DAPO Variants: Clip-Higher Settings

DAPO's clip-higher mechanism controls the degree of exploration during training. We compare two Qwen2.5-Math-7B models trained with DAPO: one with the default clip-higher setting (0.28) and another with a more restrictive setting (0.2). Figure 28 shows their JS divergence distributions on AIME 2024 and AIME 2025, revealing how the clip-higher parameter affects distributional shifts across datasets.

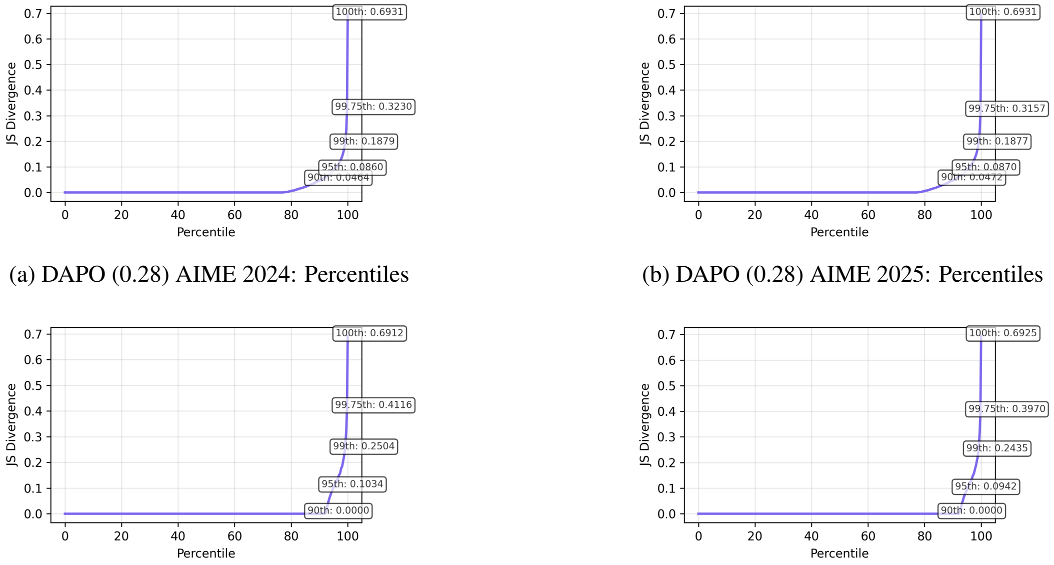
Figure 29 compares positional concentration patterns on AIME 2024 and AIME 2025, while Figure 30 and Figure 31 examine top-k overlap and rank reordering, respectively. Figure 32 shows the percentage of divergent tokens whose RL top-1 choice had base probability below a given threshold for both DAPO variants across different datasets. Figure 33 shows entropy distributions across divergence bins for both DAPO variants.



(a) DAPO: Percentile curve (topp1)

(b) SimpleRL: Percentile curve (topp1)

Figure 27: JS divergence distributions computed using top-$p = 1$ for Qwen2.5-32B with DAPO and SimpleRL on AIME 2025. The sparsity patterns are consistent with those observed using top-p truncated distributions, confirming that truncation does not significantly impact our findings.



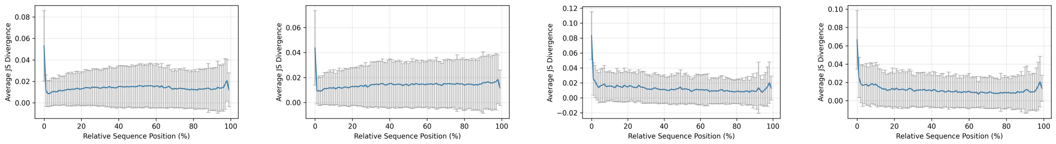
(a) DAPO (0.28) AIME 2024: Percentiles

(b) DAPO (0.28) AIME 2025: Percentiles

(c) DAPO (0.2) AIME 2024: Percentiles

(d) DAPO (0.2) AIME 2025: Percentiles

Figure 28: JS divergence distributions for Qwen2.5-Math-7B trained with DAPO under different clip-higher settings on AIME 2024 and AIME 2025. The more restrictive clip-higher=0.2 setting leads to sparser distributional shifts compared to the default 0.28 setting across both datasets, with a smaller proportion of tokens exhibiting nonnegligible divergence. However, on its divergent token set, the JS values are higher as indicated by the higher upper percentiles.



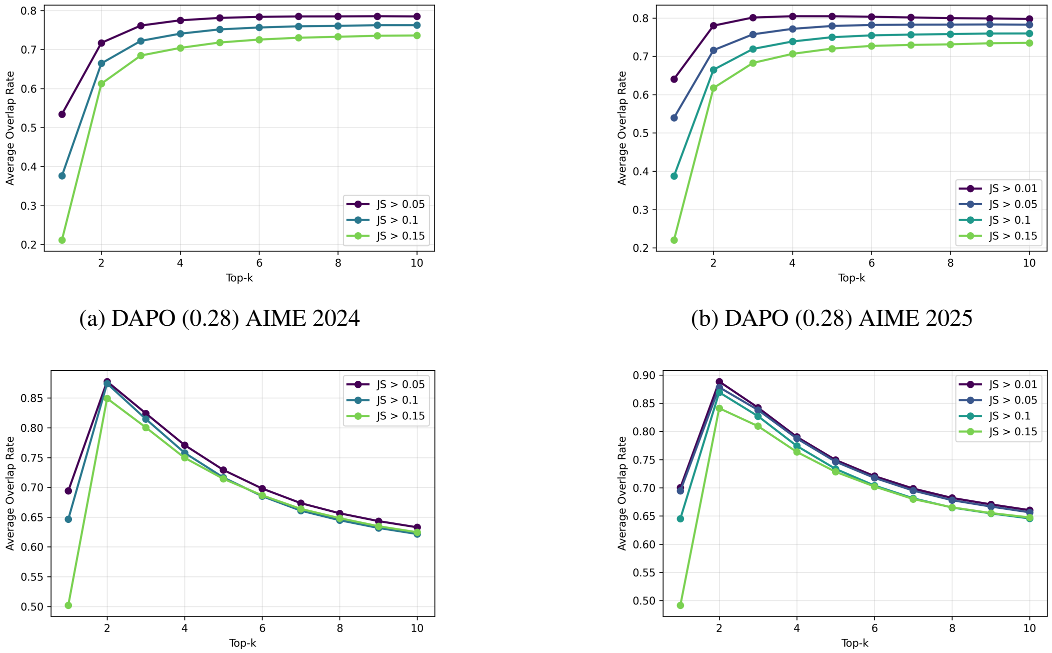
(a) DAPO (0.28) AIME 2024

(b) DAPO (0.28) AIME 2025

(c) DAPO (0.2) AIME 2024

(d) DAPO (0.2) AIME 2025

Figure 29: Mean JS divergence by normalized token position for DAPO variants with different clip-higher settings on AIME 2024 and AIME 2025.



(a) DAPO (0.28) AIME 2024

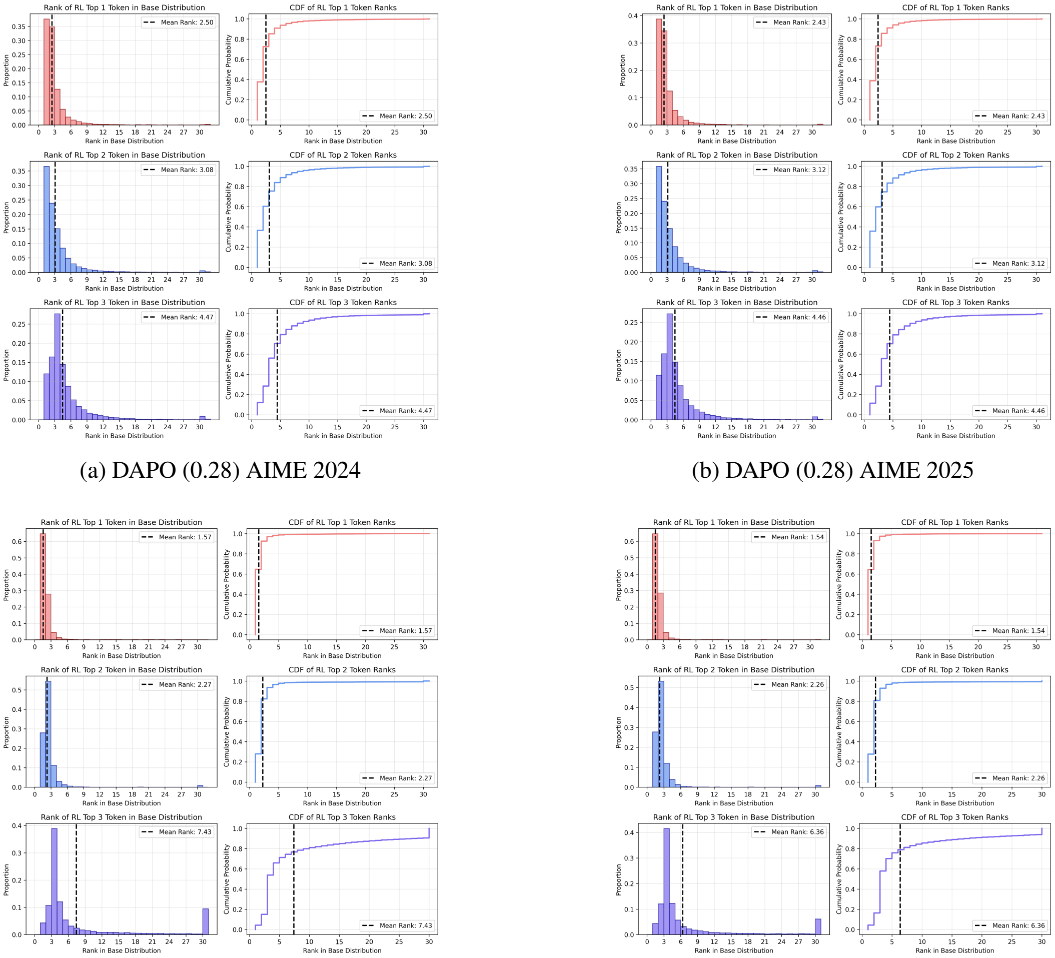
(b) DAPO (0.28) AIME 2025

(c) DAPO (0.2) AIME 2024

(d) DAPO (0.2) AIME 2025

Figure 30: Top-k token overlap between base and RL models at divergent positions ($\text{JS}_t > 0.1$) for DAPO variants on AIME 2024 and AIME 2025.

Fine-tuning Data Results. We also analyze distributional shifts on the fine-tuning data to examine how models behave on data they were fine-tuned on. Figure 34 shows JS divergence distributions, while Figures 35, 36 show additional analyses.



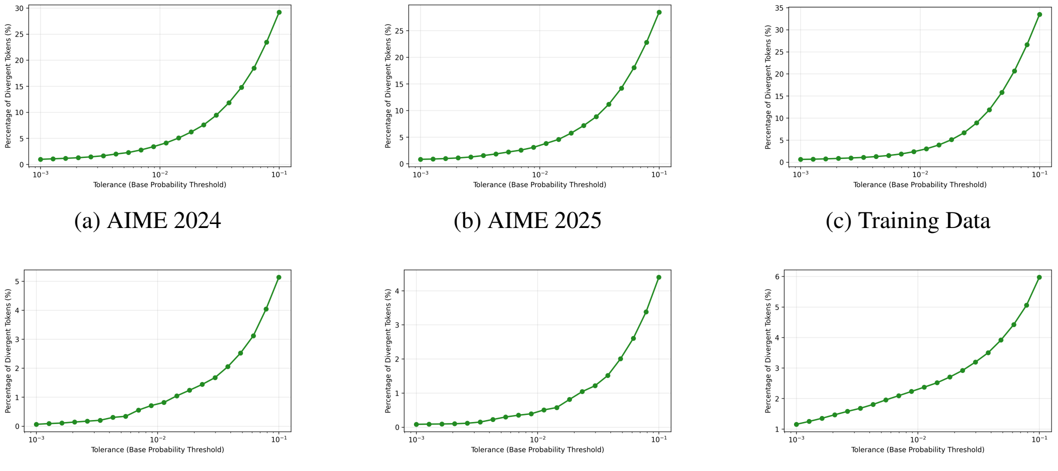
(a) DAPO (0.28) AIME 2024

(b) DAPO (0.28) AIME 2025

(c) DAPO (0.2) AIME 2024

(d) DAPO (0.2) AIME 2025

Figure 31: Distribution of base-model ranks for RL's top-3 tokens at high-divergence positions (JS > 0.1) for DAPO variants on AIME 2024 and AIME 2025.



(a) AIME 2024

(b) AIME 2025

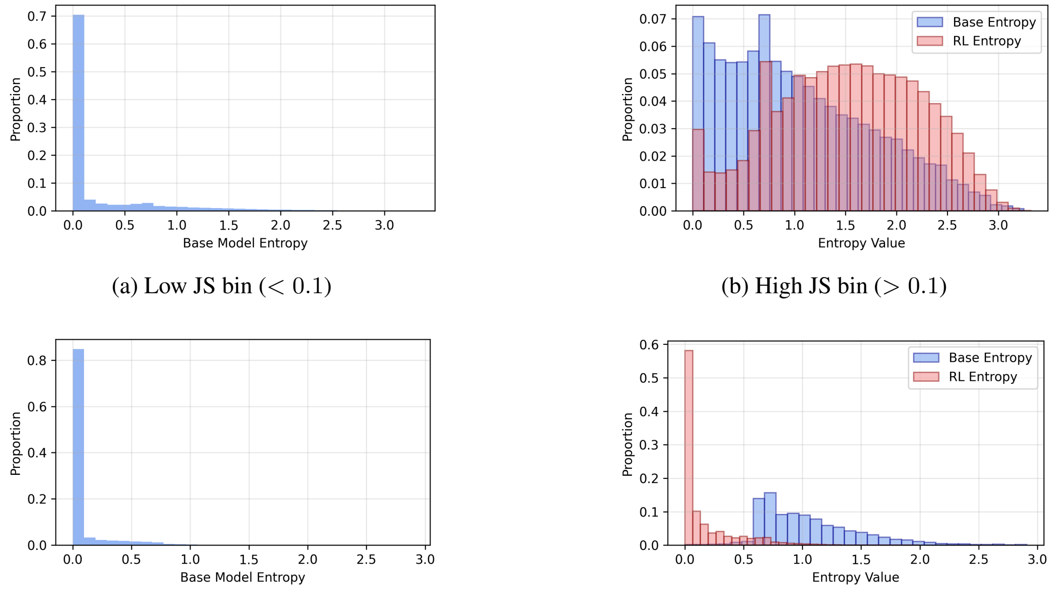
(c) Training Data

(d) AIME 2024

(e) AIME 2025

(f) Training Data

Figure 32: Percentage of divergent tokens whose RL top-1 choice had base probability below a given threshold for Qwen2.5-Math-7B with DAPO variants. Top row: DAPO (clip-higher=0.28); bottom row: DAPO (clip-higher=0.2). Consistent with findings in the main text, RL rarely promotes tokens with very low base probability, even under more exploratory settings like DAPO. We further observe a distinction between the two clip-high settings, with the more restrictive setting (0.2) promoting fewer tokens with very low base probability.



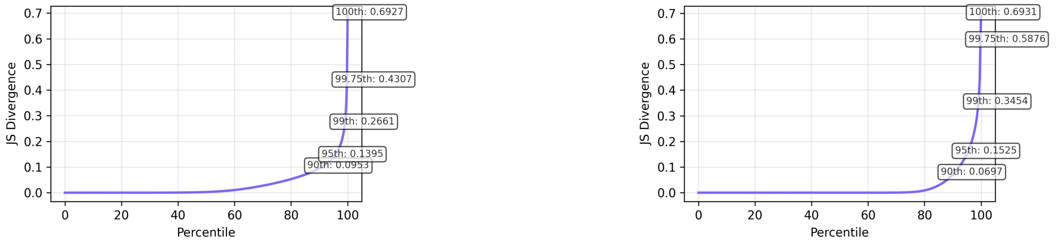
(a) Low JS bin (< 0.1)

(b) High JS bin (> 0.1)

(c) Low JS bin (< 0.1)

(d) High JS bin (> 0.1)

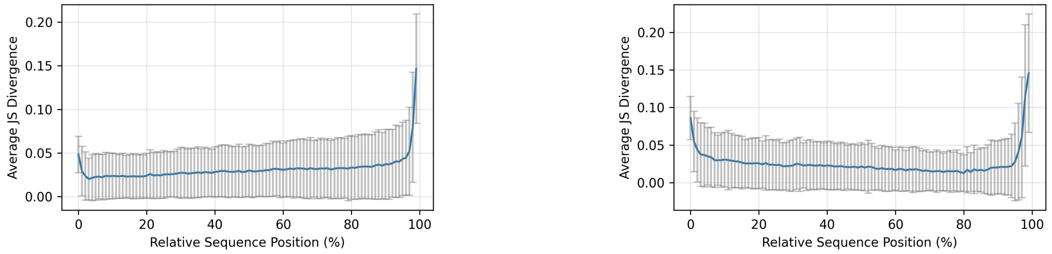
Figure 33: Entropy distributions across divergence bins for Qwen2.5-Math-7B with DAPO variants on AIME 2025. Top row: DAPO (clip-higher=0.28); bottom row: DAPO (clip-higher=0.2). Patterns are consistent with those observed in the main text, confirming the relationship between entropy and divergence across different clip-higher settings.



(a) DAPO (clip-higher=0.28): Percentiles

(b) DAPO (clip-higher=0.2): Percentiles

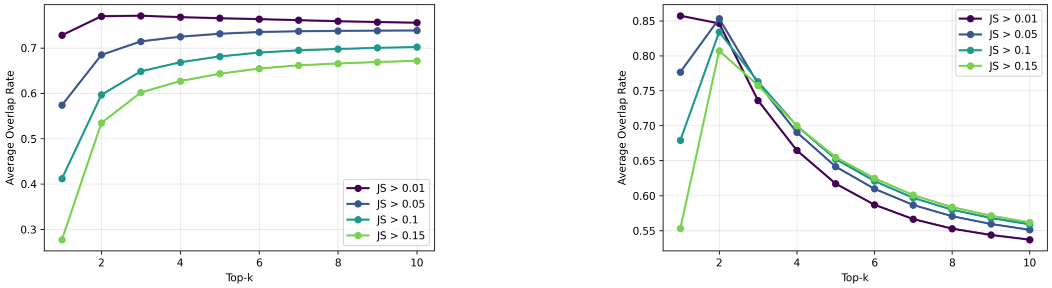
Figure 34: JS divergence distributions for DAPO variants on fine-tuning data. Distributional shifts on training data may differ from those on evaluation sets.



(a) DAPO (clip-higher=0.28)

(b) DAPO (clip-higher=0.2)

Figure 35: Mean JS divergence by normalized token position for DAPO variants on post-training data.



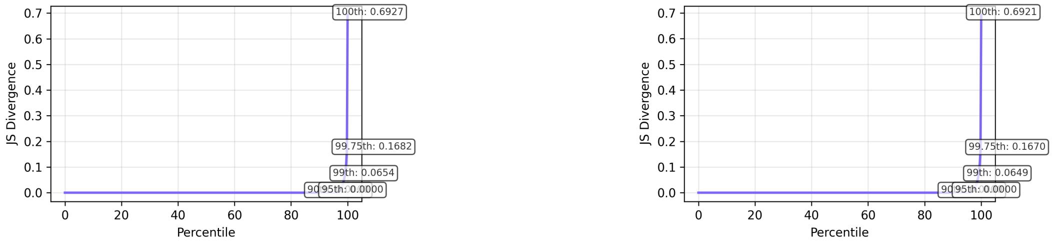
(a) DAPO (clip-higher=0.28)

(b) DAPO (clip-higher=0.2)

Figure 36: Top-k token overlap between base and RL models at divergent positions ($\text{JS}_t > 0.1$) for DAPO variants on training data.

A.4.3 MISTRAL-SMALL-24B WITH SIMPLERL

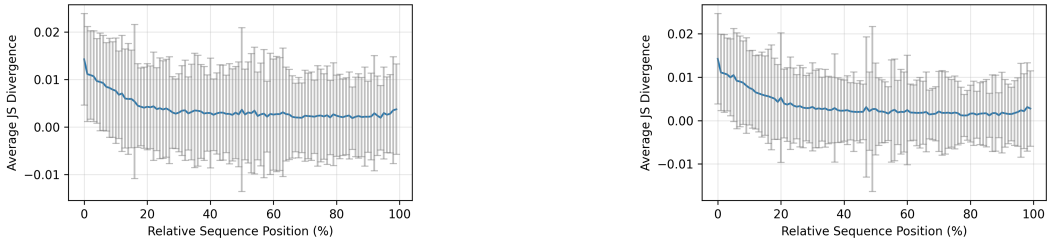
We analyze Mistral-Small-24B with SimpleRL on AIME 2024 and AIME 2025 to demonstrate the generalizability of our findings across different model architectures. Figure 37 shows JS divergence percentile curves, revealing consistent sparsity patterns. Figure 38 shows positional concentration, Figure 39 shows entropy distributions across divergence bins, and Figure 40 shows tail behavior analysis.



(a) AIME 2024: Percentiles

(b) AIME 2025: Percentiles

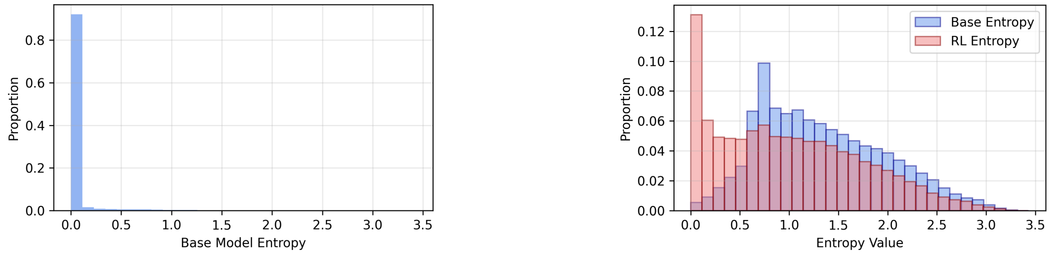
Figure 37: JS divergence distributions for Mistral-Small-24B with SimpleRL on AIME 2024 and AIME 2025. Sparse distributional shifts are consistent with findings in the main text across both datasets.



(a) AIME 2024

(b) AIME 2025

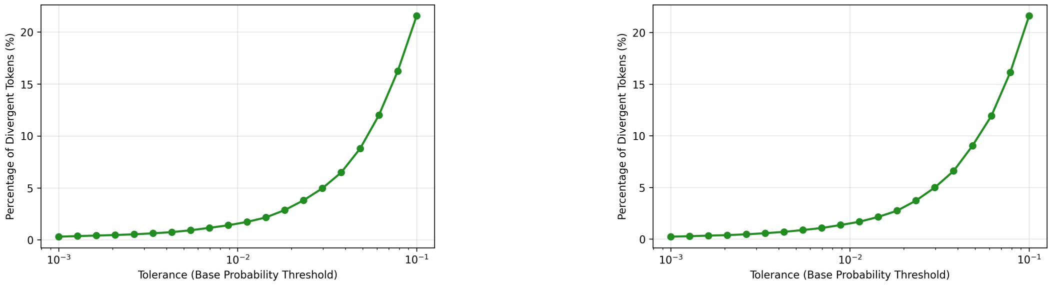
Figure 38: Mean JS divergence by normalized token position for Mistral-Small-24B with SimpleRL on AIME 2024 and AIME 2025. Consistent with findings for other models, divergences are concentrated at the start and end of responses.



(a) Mistral-24B+SimpleRL Low JS bin (< 0.1)

(b) Mistral-24B+SimpleRL High JS bin (> 0.1)

Figure 39: Entropy distributions across divergence bins using full vocabulary for Mistral-Small-24B with SimpleRL on AIME 2024. Patterns are consistent with those observed in the main text, confirming the relationship between entropy and divergence across different model architectures.



(a) Mistral-24B+SimpleRL AIME 2024

(b) Mistral-24B+SimpleRL AIME 2025

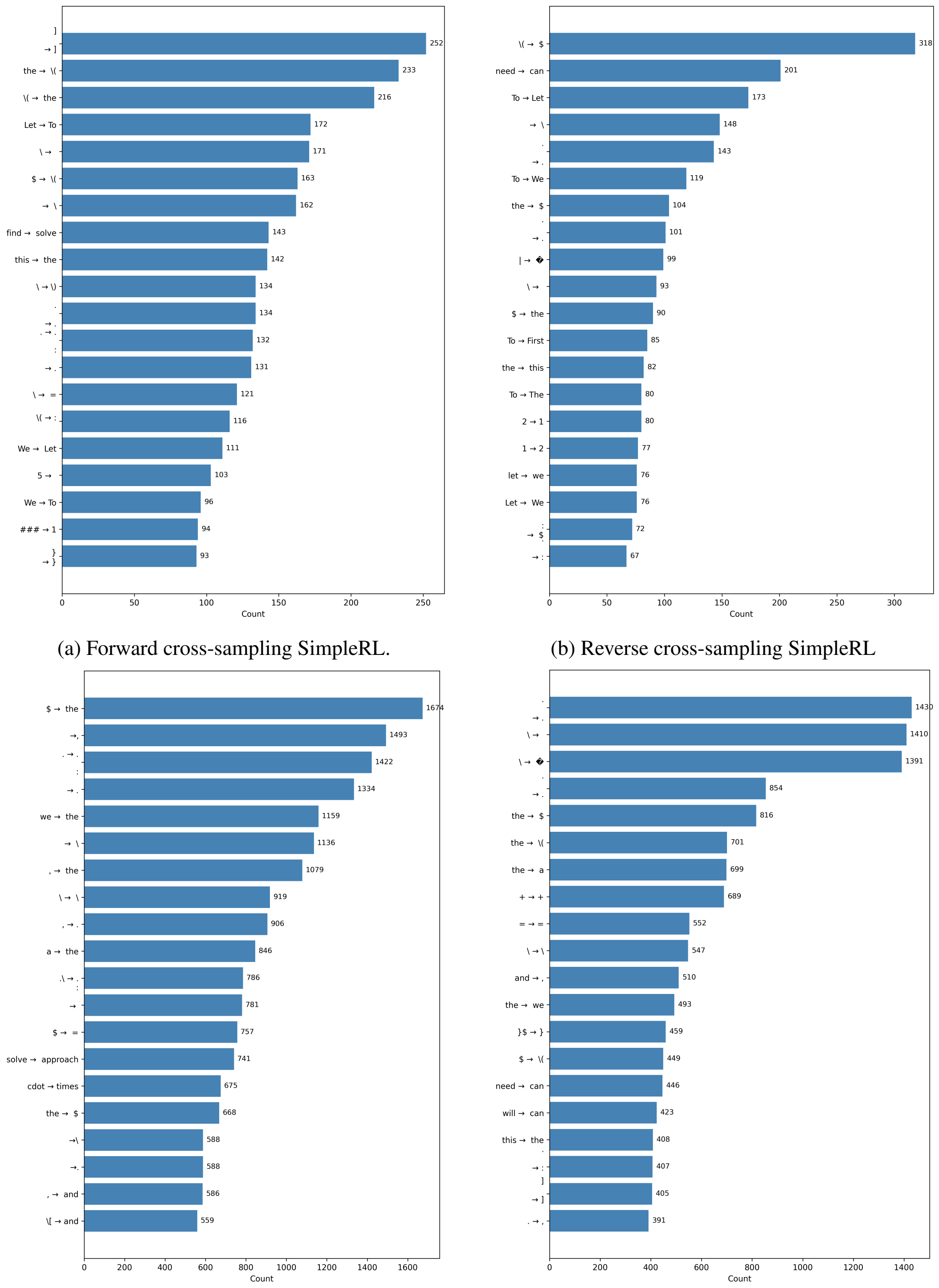
Figure 40: Percentage of divergent tokens whose RL top-1 choice had base probability below a given threshold for Mistral-Small-24B with SimpleRL on AIME 2024 and AIME 2025.

A.5 Additional Cross-Sampling Results

This section provides supplementary cross-sampling results and the algorithm used for cross-sampling experiments. Algorithm 1 describes the general procedure for cross-sampling, which generates sequences with one model, and selectively swaps tokens with another model at positions where divergence exceeds a threshold.

Algorithm 1 Cross-Sampling for a single prompt

```
Require: Prompt prefix $x_{<1}$, primary policy $\pi_{\text{primary}}$, alternate policy $\pi_{\text{alt}}$, threshold $\tau$, max steps $T$
Ensure: Generated sequence $x_{1:t}$, swap count $k$
$k \leftarrow 0$
Initialize prefix $x_{<1}$
for $t = 1 \ldots T$ do
    Compute $d_t = D(\pi_{\text{primary}}(\cdot \mid x_{<t}) \parallel \pi_{\text{alt}}(\cdot \mid x_{<t}))$
    if $d_t > \tau$ then
        Sample $x_t \sim \pi_{\text{alt}}(\cdot \mid x_{<t})$
        $k \leftarrow k + 1$
    else
        Sample $x_t \sim \pi_{\text{primary}}(\cdot \mid x_{<t})$
    end if
    Append $x_t$ to prefix $x_{<t+1}$
    if $x_t =$ EOS then
        break
    end if
end for
return generated tokens $x_{1:t}$ and swap count $k$
```



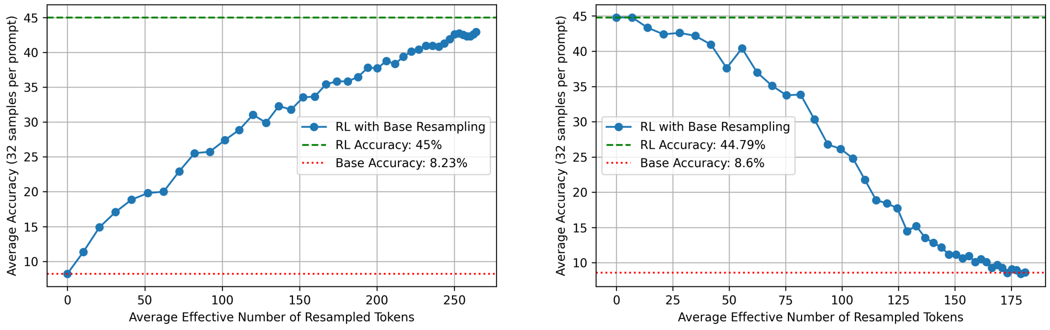
(a) Forward cross-sampling SimpleRL.

(b) Reverse cross-sampling SimpleRL

(c) Forward cross-sampling DAPO

(d) Reverse cross-sampling DAPO

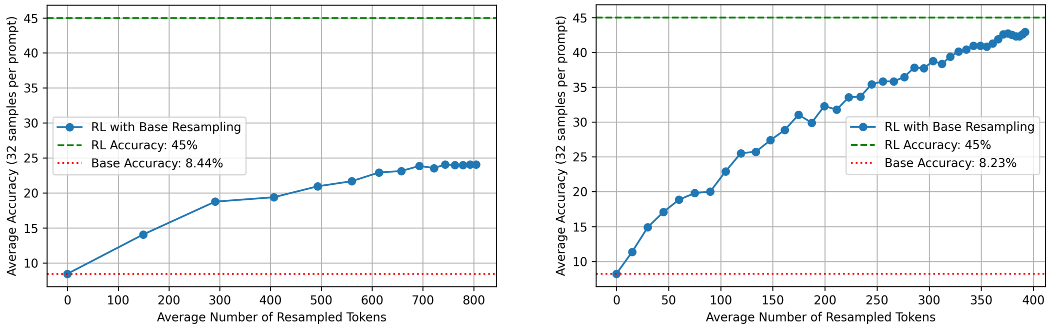
Figure 41: Cross-sampling token pair histograms.



(a) Forward cross-sampling

(b) Reverse cross-sampling

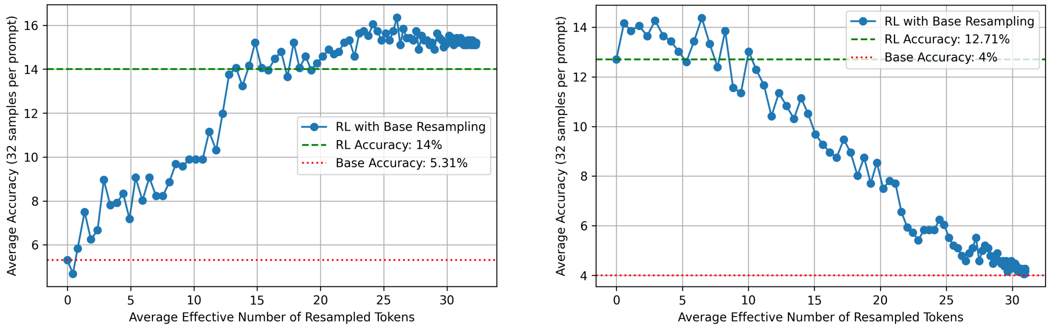
Figure 42: Cross-sampling results (DAPO on AIME24): injecting RL tokens into base generations progressively recovers RL accuracy, while reverting RL tokens with base tokens causes near-monotonic degradation toward base performance.



(a) Random baseline

(b) DAPO

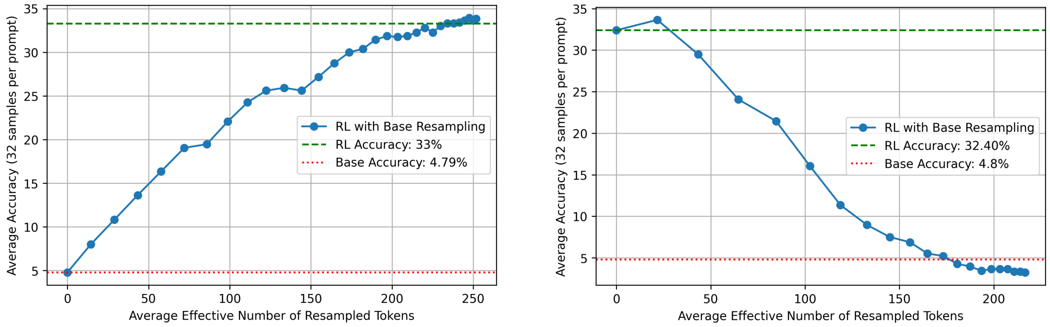
Figure 43: Comparison of random baseline and DAPO cross-sampling on AIME24: average number of tokens (including identity swaps) replaced versus accuracy. The random baseline shows minimal performance improvement, demonstrating that targeted RL token selection is critical for performance gains.



(a) Forward cross-sampling

(b) Reverse cross-sampling

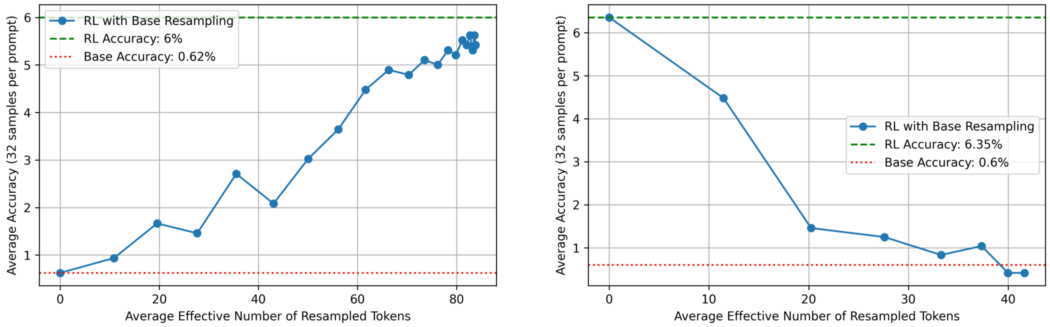
Figure 44: Cross-sampling results (SimpleRL on AIME25): injecting RL tokens into base generations progressively recovers RL accuracy, while reverting RL tokens with base tokens causes near-monotonic degradation toward base performance. Interestingly, in the base with RL resampling case, the performance actually meaningfully exceeds the RL model's performance.



(a) Forward cross-sampling

(b) Reverse cross-sampling

Figure 45: Cross-sampling results (DAPO on AIME25): injecting RL tokens into base generations progressively recovers RL accuracy, while reverting RL tokens with base tokens causes near-monotonic degradation toward base performance.



(a) Forward cross-sampling

(b) Reverse cross-sampling

Figure 46: Cross-sampling results (Mistral-Small-24B + SimpleRL on AIME 2024): injecting RL tokens into base generations progressively recovers RL accuracy, while reverting RL tokens with base tokens causes near-monotonic degradation toward base performance.

A.6 Additional Divergence-Weighted Advantage Discussion/Results

This section presents supplementary results for divergence-weighted advantages, including alternative configurations and evaluations on Qwen2.5-7B. The main text focuses on Qwen2.5-Math-7B with KL divergence computed with respect to $\pi_{\theta_{\text{old}}}$ and sigmoid weighting. Here we discuss alternative schemes, along with additional results.

A.6.1 Alternative Divergence Choices

Beyond the old-policy KL divergence presented in the main text, one could also use the reference-based KL divergence:

$$\mathrm{KL}_t^{\text{ref}} = D_{\mathrm{KL}}(\pi_{\theta_{\text{old}}}(\cdot \mid x_{<t}) \parallel \pi_{\text{ref}}(\cdot \mid x_{<t}))\,, \tag{6}$$

where π_{ref} denotes the base reference model. The reference-based KL quantifies the alignment between the current policy and the original base model, measuring the cumulative divergence from the initial model at each token position. This contrasts with the old-policy KL, which captures only the magnitude of recent policy updates within a single training iteration. In our experiments we mainly just use the old-policy KL as this does not require additional pass through the base model.

A.6.2 Alternative Weighting Schemes

In addition to the sigmoid weighting scheme presented in the main text, we examine linear relative weighting:

$$w_t = 1 + \alpha\big(\mathrm{KL}_t - \mu_{\mathrm{KL}}\big), \quad \mu_{\mathrm{KL}} = \tfrac{1}{T}\sum_{t=1}^{T}\mathrm{KL}_t. \tag{7}$$

The linear relative scheme scales weights linearly with the deviation from the mean KL divergence across the sequence, offering a simpler alternative to the sigmoid transformation. As in the sigmoid case, $\alpha > 0$ amplifies high-divergence tokens, while $\alpha < 0$ emphasizes low-divergence ones.

A.6.3 Results on Additional Configurations

Table 4 summarizes the performance of these alternative configurations, including evaluations using linear relative weighting on Qwen2.5-7B.

Table 4: Accuracy (%) under additional divergence-weighted configurations on Qwen2.5-7B. Results shown across AIME24, AIME25, and AMC datasets. The results displayed are the avg@32 scores.

Configuration	**AIME24**	**AIME25**	**AMC**	**Overall Avg**
Baseline DAPO	16.77	8.12	70.78	31.89
High-KL Lin. Rel. (sched.)	19.58	12.40	71.12	34.37
High-KL Lin. Rel.	20.00	12.29	73.31	35.20

Table 5: Accuracy (%) for 80/20 clip entropy configuration on Qwen2.5-Math-7B. Results shown across AIME24, AIME25, and AMC datasets. The results displayed are the avg@32 scores.

Configuration	**AIME24**	**AIME25**	**AMC**	**Overall Avg**
80/20 clip entropy	35.26	17.03	72.68	41.66

A.7 LLM USAGE

LLMs were used to assist with minor polishing of the writing.