ADAPTIVE LAYERWISE PERTURBATION FOR STABLE OFF-POLICY LLM REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Training large language models with reinforcement learning often reuses rollout batches and generates them with a separate inference engine. These practices create policy staleness and training–inference mismatch, which can destabilize importance-weighted updates. We introduce Adaptive Layerwise Perturbation (ALP), which learns a Gaussian perturbation scale for each layer and samples hidden-state perturbations during policy updates. The perturbed training policy forms the numerator of an importance ratio against the unperturbed rollout policy; rollout generation and evaluation remain unchanged. A smoothed-policy analysis relates perturbation to mismatch sensitivity and objective curvature. Across single-turn math and multi-turn tool-integrated reasoning with three model backbones, ALP achieves the highest average accuracy among the evaluated methods. In the five-seed multi-turn comparison, Seq-ALP reaches 50.36 average accuracy versus 48.92 for the strongest baseline, Token-MIS. ALP also shows controlled KL divergence, smaller importance-ratio tail excursions than Bypass, and higher multi-turn pass@k at fixed rollout budgets. Location ablations favor perturbing all layers over the tested partial-layer and logits-only variants.

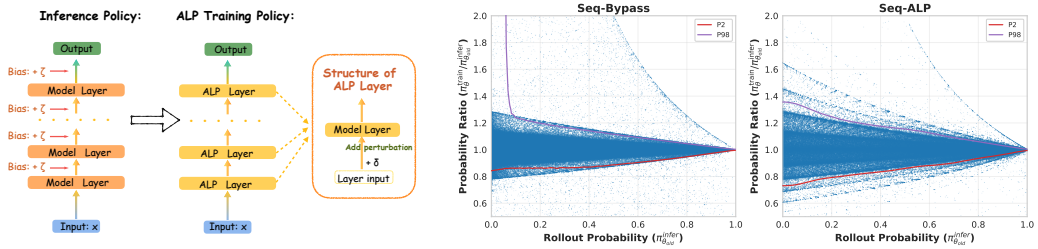


Figure 1: Left: ALP samples layerwise hidden-state perturbations during policy updates. Right: at the illustrated multi-turn checkpoint (step 140), ALP reduces the upper importance-ratio tail at low rollout probabilities relative to Bypass. P2 and P98 mark the 2nd and 98th ratio percentiles.

1 INTRODUCTION

The trade-off between training efficiency and stability is a central challenge in reinforcement learning (RL) for large language models (LLMs). To increase efficiency, practical systems often optimize a policy using rollouts generated by a different distribution, which is known as off-policy. Off-policy effects arise from multiple sources. First, the same batch of trajectories is used across several policy updates. Second, modern inference engines such as vLLM (Kwon et al., 2023) and SGLang (Zheng et al., 2024) introduce training–inference mismatch due to quantization, batching, and kernel-level differences, even when weights are nominally identical (Yao et al., 2025). In multi-turn agentic settings, this mismatch can be further amplified by inference distribution shifts induced by tool feedback and out-of-distribution observations (Jiacai et al., 2025). These factors lead to heavy-tailed importance ratios, KL spikes, and brittle optimization dynamics, making robust off-policy RL an urgent problem.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

A prominent line of work stabilizes training by modifying importance ratios. Sequence-level objectives aggregate token ratios over a trajectory (Zheng et al., 2025). Other approaches address training–inference mismatch. Bypass uses the ratio of updated and rollout policies (Jiacai et al., 2025; Yao et al., 2025). TIS/MIS (Yao et al., 2025; Jiacai et al., 2025) also use a ratio between the old training policy and the rollout policy, then truncate or mask large corrections. These methods can prevent collapse, but separate clipping or masking can discard usable updates and introduce additional bias. We defer a detailed discussion of related work to Section 1.1.

We argue that off-policy instability in LLM RL is not solely a bookkeeping issue of "which importance ratio to use", but also a **geometry** issue: the noisy update may push the policy toward sharp, brittle regions where small distribution shifts (from staleness or inference mismatch) cause large changes in action probabilities. This motivates a perturbation-sampling perspective from model smoothness and distributional learning approaches (Shen & Meinshausen, 2023; Hao et al., 2025): when the dominant failure mode is system-induced noise and staleness, a natural defense is *fighting noise with structured noise*.

Specifically, we propose **Adaptive Layerwise Perturbation (ALP)** by sampling Gaussian hidden-state perturbations with learned per-layer scales during policy updates, producing a perturbed training policy $\pi_{\theta,\sigma}^{\text{train}}$ (Figure 1, left). The objective uses one importance ratio between this sampled training policy and the unperturbed rollout policy; perturbation enters only the numerator, and rollout generation is unchanged.

Perturbing hidden states during policy updates can **smooth** the local objective and reduce sensitivity to mismatch. Figure 1 (right) shows one multi-turn checkpoint at which ALP has a smaller upper importance-ratio tail than Bypass for low-probability rollout tokens. Section 2.3 analyzes a simplified smoothing mechanism, while Sections 3.1 and 4 evaluate training behavior over longer runs.

Our contributions are three-fold:

1. **Layerwise perturbation for off-policy training.** We introduce ALP, which learns a perturbation scale for each layer and samples hidden-state perturbations during policy updates. Its objective compares the perturbed training policy with the unperturbed rollout policy through a single importance ratio, providing a simple approach to training with policy staleness and training–inference mismatch.
2. **Empirical stability and performance.** ALP leads the evaluated methods across three model backbones. In the five-seed multi-turn comparison, Seq-ALP improves mean average accuracy by 1.44 points over Token-MIS; the training diagnostics show controlled KL and smaller importance-ratio tails than Bypass. Location ablations favor perturbing all layers.
3. **Analysis of perturbation and mismatch.** We analyze a simplified smoothed-policy model to examine how perturbation can reduce sensitivity to an inference-time shift under explicit assumptions, and connect perturbation to objective smoothing. This provides a theoretical perspective on the empirical stability observed in the experiments.

1.1 Related Works

Perturbation Perturbation-based training is often motivated as a form of *local averaging*: optimizing the expected objective under small input/latent disturbances can reduce sensitivity to sharp, brittle regions of the loss landscape and improve robustness under distribution shift. This perspective has been used across a range of learning settings to stabilize optimization and enhance generalization. A line of work has focused on enhancing model smoothness via perturbation-based sampling across diverse settings. For instance, Moreno-Barea et al. (2018); Yu et al. (2023) inject Gaussian noise to augment the training data and improve generalization. Certified robustness methods (Cohen et al., 2019; Salman et al., 2019; Lecuyer et al., 2019; Yang et al., 2020) pursue stability against adversarial attacks by introducing perturbations drawn from specific distributions during training. Beyond robustness, Li (2022); Pereira et al. (2021) demonstrate that perturbation injection can also improve performance across a range of tasks. Another related line of research focuses on diffusion models (Ho et al., 2020; Song et al., 2020; Dhariwal & Nichol, 2021; Saharia et al., 2022; Rombach et al., 2022), which model sample distributions as convolutions of Gaussian distributions and

introduce Gaussian noise to the data during training. More recently, Shen & Meinshausen (2023); Hao et al. (2025) propose learnable perturbation mechanisms that adaptively enhance model performance. Related ideas also appear in diffusion models, where additional perturbations can mitigate train–test mismatch in iterative generation: Ning et al. (2023) reduce exposure bias by perturbing training inputs to better match the distribution encountered during sampling, while Li et al. (2023) alleviate exposure bias via sampling with shifted time steps.

2 Adaptive Layerwise Perturbation

2.1 Prior Approaches

For prompts $x \sim d_0$, the response a is generated from the rollout policy $\pi^{\text{infer}}_{\theta_{\text{old}}}(\cdot|x)$, where θ_{old} is the parameter from the last step. We then update θ using training probability $\pi^{\text{train}}_{\theta}$. The standard GRPO objective is

$$\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}_{x\sim d_0,\{a_i\}_{i=1}^{n}\sim\pi^{\text{infer}}_{\theta_{\text{old}}}(\cdot|x)}\Bigg[\frac{1}{n}\sum_{i=1}^{n}\sum_{t=1}^{|a_i|} \min\Big\{\rho_{i,t}(\theta)A_i, \text{clip}\left(\rho_{i,t}(\theta), 1-\epsilon, 1+\epsilon\right)A_i\Big\}\Bigg], \tag{1}$$

where $|a_i|$ denotes the length of a_i, A_i is the group advantage, and $\rho_{i,t}(\theta)$ is the token-level importance ratio at the t-th token: $\rho_{i,t}(\theta) = \pi^{\text{train}}_{\theta}(a_{i,t}|x, a_{i,<t})/\pi^{\text{train}}_{\theta_{\text{old}}}(a_{i,t}|x, a_{i,<t})$.

Sequence-level objective. However, this token-level objective is biased when the difference between $\pi^{\text{train}}_{\theta}$ and $\pi^{\text{train}}_{\theta_{\text{old}}}$ cannot be neglected. We can use the unbiased sequence-level objective (Zheng et al., 2025),

$$\rho_i(\theta) = \frac{\pi^{\text{train}}_{\theta}(a_i|x)}{\pi^{\text{train}}_{\theta_{\text{old}}}(a_i|x)} = \prod_{t=1}^{|a_i|}\frac{\pi^{\text{train}}_{\theta}(a_{i,t}|x, a_{i,<t})}{\pi^{\text{train}}_{\theta_{\text{old}}}(a_{i,t}|x, a_{i,<t})}.$$

To balance the bias and variance, Group Sequence Policy Optimization (GSPO) (Zheng et al., 2025) proposes using the geometric mean: $(\pi^{\text{train}}_{\theta}(a_i|x)/\pi^{\text{train}}_{\theta_{\text{old}}}(a_i|x))^{1/|a_i|}$.

Masked importance sampling (MIS). Since $\pi^{\text{infer}}_{\theta_{\text{old}}}$ and $\pi^{\text{train}}_{\theta_{\text{old}}}$ have a mismatch, to make algorithms robust to this mismatch, a standard approach is to multiply by a correcting ratio (Yao et al., 2025; Liu et al., 2025):

$$\mathcal{J}_{\text{MIS}}(\theta) = \mathbb{E}_{x\sim d_0,\{a_i\}_{i=1}^{n}\sim\pi^{\text{infer}}_{\theta_{\text{old}}}(\cdot|x)}\Bigg[\frac{1}{n}\sum_{i=1}^{n}\sum_{t=1}^{|a_i|}\text{Mask}\left\{\rho'_{i,t}, C\right\}\cdot \min\Big\{\rho_{i,t}(\theta)A_i, \text{clip}\left(\rho_{i,t}(\theta), 1-\epsilon, 1+\epsilon\right)A_i\Big\}\Bigg],$$

where C is a threshold parameter, the gradient is masked when $\rho'_{i,t} > C$, and the additional importance ratio $\rho'_{i,t} = \pi^{\text{train}}_{\theta_{\text{old}}}(a_{i,t}|x, a_{i,<t})/\pi^{\text{infer}}_{\theta_{\text{old}}}(a_{i,t}|x, a_{i,<t})$. This ratio can also be calculated at the sequence level: $\rho'_i = \prod_t \rho'_{i,t}$.

Bypass fails to stabilize training. A direct way to unify the importance ratio is to replace $\rho_{i,t}(\theta)$ with $\pi^{\text{train}}_{\theta}(a_{i,t}|x, a_{i,<t})/\pi^{\text{infer}}_{\theta_{\text{old}}}(a_{i,t}|x, a_{i,<t})$. This approach is a baseline in training–inference mismatch work (Yao et al., 2025; Jiacai et al., 2025). Figure 1 (right) shows an enlarged upper ratio tail for Bypass at the illustrated multi-turn checkpoint, especially among low-probability rollout tokens. The controlled comparison in Figure 2 further shows how perturbation narrows the conditional log-ratio envelope on the same checkpoint and rollout batch.

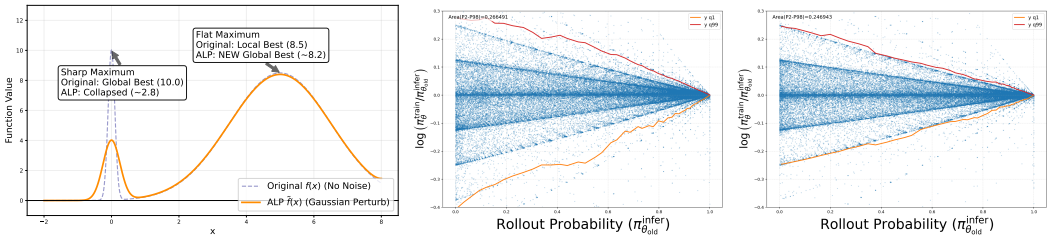


Figure 2: Effects of perturbation. Left: perturbation smooths a sharp objective into a flatter surrogate. Middle/right: in a controlled multi-turn comparison from the same checkpoint and rollout batch, perturbation shrinks the mismatch envelope, especially in the low-probability tail.

2.2 Our Approach

For layer $h \in [H]$ in the model, suppose the dimension of its input embedding is d_h. We add a Gaussian perturbation variable $\delta^h \sim \mathcal{N}(0, \sigma_h^2 I_{d_h})$ to the updated training policy. The per-token perturbation is $\delta_t = [(\delta_t^1)^T, \ldots, (\delta_t^H)^T]^T \sim \mathcal{N}(0, \Sigma_\sigma)$, where $\Sigma_\sigma = \text{diag}(\sigma_1^2 I_{d_1}, \ldots, \sigma_H^2 I_{d_H})$ and $d = \sum_{h=1}^H d_h$. Writing $\delta = (\delta_1, \ldots, \delta_{|a|})$, we define the perturbed policy $\pi_{\theta,\sigma}(a|x, \delta) = \prod_{t=1}^{|a|} \pi_{\theta,\sigma}(a_t|x, a_{<t}, \delta_t)$, where the δ_t are sampled independently across token positions. Each perturbed layer has one trainable scalar τ_h with $\sigma_h = \exp(\tau_h)$, and the perturbation is drawn as $\delta^h = \sigma_h \epsilon^h$ with $\epsilon^h \sim \mathcal{N}(0, I_{d_h})$, so σ receives gradients from the same clipped objective as θ and is updated by AdamW in a separate parameter group with its own learning rate. No explicit regularizer or constraint is placed on σ; Appendix E.1 gives the parameterization, optimizer settings, and resulting scale dynamics. The objective below depends on both θ and σ. Then, the loss function is

$$\mathcal{J}_{\text{ALP}}(\theta, \sigma) = \mathbb{E}_{x \sim d_0, \{a_i\}_{i=1}^n \sim \pi^{\text{infer}}_{\theta_{\text{old}}}(\cdot|x), \delta_{i,t} \overset{\text{iid}}{\sim} \mathcal{N}(0, \Sigma_\sigma)} \Bigg[\frac{1}{n} \sum_{i=1}^{n} \sum_{t=1}^{|a_i|} \min \left\{ \rho_{i,t}^{\text{ALP}}(\theta, \sigma) A_i, \text{clip}\left(\rho_{i,t}^{\text{ALP}}(\theta, \sigma), 1 - \epsilon_l, 1 + \epsilon_h\right) A_i \right\} \Bigg], \tag{2}$$

where the token-level importance ratio

$$\rho_{i,t}^{\text{ALP}}(\theta, \sigma) = \frac{\pi_{\theta,\sigma}(a_{i,t}|x, a_{i,<t}, \delta_{i,t})}{\pi^{\text{infer}}_{\theta_{\text{old}}}(a_{i,t}|x, a_{i,<t})},$$

and the sequence-level importance ratio

$$\rho_i^{\text{ALP}}(\theta, \sigma) = \prod_{t=1}^{|a_i|} \rho_{i,t}^{\text{ALP}}(\theta, \sigma) = \frac{\pi_{\theta,\sigma}(a_i|x, \delta_i)}{\pi^{\text{infer}}_{\theta_{\text{old}}}(a_i|x)} \tag{3}$$

defines the sequence-level variant of the objective. The numerator uses sampled perturbations, independently across token positions; rollout generation and evaluation use the unperturbed policy. The averaged policy $\widetilde{\pi}$ in Section 2.3 is an analytical device and is not computed in this objective.

2.3 A Simplified Analysis of Perturbation

We analyze mismatch sensitivity and objective curvature in a tractable smoothed-policy model. Here x is a continuous representation, and

$$\widetilde{\pi}_\theta(a|x) = \mathbb{E}_{\delta \sim \mathcal{N}(0, \sigma^2 I_d)} \pi_\theta(a|x + \delta).$$

This input-level averaging omits clipping and abstracts the sampled, multilayer ALP update. Gaussian noise is the smoothing kernel; we do not assume that the actual engine mismatch is Gaussian or globally additive. Formal statements are in Appendices C and D.

Controlling training-inference mismatch. Locally, we model a representation shift by $\pi^{\text{infer}}_{\theta_{\text{old}}}(a|x) = \pi_{\theta_{\text{old}}}(a|x + \zeta)$.

Theorem 1 (Informal). *For shifts whose line segments remain in a modeled neighborhood, the local domination and score conditions in Appendix C imply*

$$\mathbb{E}_{x\sim d_0,\zeta} D_{\mathrm{KL}}\left(\widetilde{\pi}_{\theta_{\mathrm{old}}}(\cdot|x)\,\|\,\pi^{\mathrm{infer}}_{\theta_{\mathrm{old}}}(\cdot|x)\right) \le -\ln\alpha_\sigma + L_\sigma\mathbb{E}\|\zeta\|_2 \le -\ln\alpha_\sigma + \frac{\sqrt{Cd}}{\sigma}\mathbb{E}\|\zeta\|_2. \tag{4}$$

Here α_σ is the worst-case local ratio between the original and smoothed policies, and L_σ is the local score envelope.

The proof integrates the score exactly and uses no Taylor approximation. The two terms also expose the scale trade-off: smoothing can reduce L_σ, while policy distortion can reduce α_σ. Thus the result is a local sensitivity bound, not a global guarantee or a prescription for the learned σ.

Smoothing a surrogate objective. Gaussian perturbation also averages an unclipped per-input surrogate over nearby representations.

Theorem 2 (Informal). *Let b bound the parameter-Hessian norm, and let the inputs with curvature at least $cb/2$ have finite volume V_c, for $c \in (0,1)$. Then*

$$\sigma \ge (2V_c/c)^{1/d}/\sqrt{2\pi} \quad\Longrightarrow\quad \sup_x \|\nabla^2_\theta \widetilde{J}_\sigma(x,\theta)\|_2 \le cb.$$

This is a sufficient result for a fixed-parameter, input-averaged surrogate; it does not assert that the sampled multilayer update or its learned scales satisfy the condition. Figure 2 illustrates the two mechanisms and shows a narrower conditional log-ratio envelope under perturbation.

3 Experiments

We repeat the main multi-turn TIR comparison across five training seeds (Table 3) to quantify training-run variability. Due to limited computational resources, we restrict this multi-seed evaluation to the main TIR experiment. Given the computational cost of full RL training, we use one run per method or variant in the complementary single-turn comparisons and perturbation-location ablations, which test additional backbones and perturbation choices.

3.1 Single-turn Reasoning

3.1.1 Experimental settings

Dataset. For *single-turn* reasoning, we study math tasks without tool use. The training set merges the math subset of Guru RL-92k[1](Cheng et al., 2025) and a 75K OpenR1[2](Hugging Face, 2025) subset, with correctness verified by **Math-Verify** (Kydlíček, n.d.). Additional preprocessing details are deferred to Appendix E.

Training. We implement all methods in verl (Sheng et al., 2024), following its default setup where possible. We use AdamW with learning rate 1×10^{-6} and the clip-higher trick (Yu et al., 2025) with asymmetric clipping $(1-\epsilon_l, 1+\epsilon_h)$. The models are trained from Qwen2.5-Math-1.5B-base (Yang et al., 2024); each iteration samples 512 prompts, rolls out $n = 8$ responses per prompt at temperature 1.0, and performs 16 policy updates. As provided in Table 6 in the appendix, token/seq denotes whether the ratio is aggregated at the token level or sequence level. The definitions of these notations are provided in Sec. 2.1. Unless otherwise specified, the main single-turn experiments use Qwen2.5-Math-1.5B-base; we additionally report a validation run on Qwen3-4B (Yang et al., 2025) to test whether the observed trend transfers to a different backbone.

Benchmarks. We evaluate on Math500 (Hendrycks et al., 2021), Minerva Math (Lewkowycz et al., 2022), Olympiad Bench (He et al., 2024), AIME2024[3], and AIME2025[4]. We report average@32 at temperature 1.0 with a generation limit of 4096 tokens. Baselines include Seq-Bypass, Seq-MIS, Token-MIS, and vanilla GRPO; Table 6 summarizes how each method forms and clips the importance ratios.

[1] https://huggingface.co/datasets/LLM360/guru-RL-92k

[2] https://huggingface.co/datasets/weqweasdas/from_default_filtered_openr1_with_scores

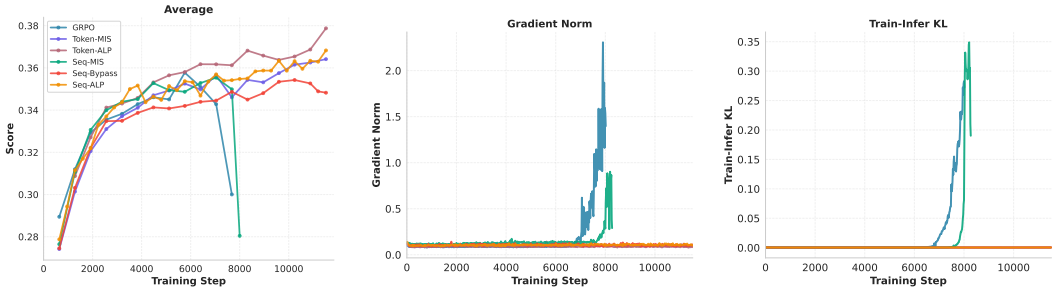
[3] https://huggingface.co/datasets/math-ai/aime24

[4] https://huggingface.co/datasets/math-ai/aime25

3.1.2 MAIN RESULTS

Method	Math500	Minerva Math	Olympiad Bench	AIME24	AIME25	Average
GRPO	75.91	36.43	38.82	16.77	10.94	35.77
Seq-Bypass	76.21	35.23	38.52	16.35	7.81	34.82
Seq-MIS	77.08	36.75	39.06	15.21	9.58	35.54
Token-MIS	77.84	35.94	40.06	17.40	10.83	36.41
Token-ALP	**78.10**	**37.27**	**40.77**	**21.46**	11.77	**37.87**
Seq-ALP	77.84	37.06	40.28	16.98	**11.98**	36.83

Table 1: Single-turn RL evaluation trained from Qwen2.5-Math-1.5B (average@32, temperature 1.0) across five math benchmarks. Token-ALP attains the highest reported average score.



(a) Average test accuracy (b) Gradient norm (c) Train–inference KL

Figure 3: Single-turn accuracy and training dynamics from Qwen2.5-Math-1.5B. GRPO and Seq-MIS show late accuracy drops together with spikes in gradient norm and train–inference KL; both ALP variants, Token-MIS, and Seq-Bypass continue without a comparable collapse in the displayed runs. Curves have different stopping steps, so the rightmost points are not a matched-step comparison.

Method	Math500	Minerva Math	Olympiad Bench	AIME24	AIME25	Average
GRPO	94.49	56.96	68.28	60.29	**51.17**	66.24
Token-MIS	94.28	53.66	66.36	62.35	50.00	65.33
Seq-Bypass	93.70	52.00	64.21	63.80	46.88	64.12
Seq-ALP	**95.18**	**57.44**	**69.49**	**64.32**	**51.17**	**67.52**

Table 2: Single-turn RL validation on Qwen3-4B under the same evaluation protocol. Seq-ALP achieves the best average score across benchmarks.

ALP achieves the highest average accuracy. Under single-turn evaluation in Table 1, Token-ALP attains the best overall average score (37.87), with Seq-ALP second (36.83). Token-MIS is the strongest baseline, while Seq-MIS, GRPO, and Seq-Bypass perform worse overall. To test whether the benefit of perturbation is specific to Qwen2.5-Math-1.5B, we further repeat the single-turn experiment on Qwen3-4B. Table 2 shows that Seq-ALP achieves the best overall average score (67.52), outperforming GRPO (66.24), Token-MIS (65.33), and Seq-Bypass (64.12). This suggests that the gain from ALP is not tied to one particular base model and persists at a larger model scale.

ALP combines strong accuracy with stable training in these runs. Figure 3 places average test accuracy beside gradient norm and train–inference KL. GRPO and Seq-MIS show late accuracy drops alongside sharp diagnostic spikes. Both ALP variants maintain their accuracy without comparable spikes in the displayed runs, and Token-ALP has the highest reported average in Table 1. Token-MIS and Seq-Bypass also avoid a comparable collapse.

4 MULTI-TURN AGENTIC REASONING

4.1 EXPERIMENTAL SETTINGS

We study multi-turn Tool-Integrated Reasoning (TIR) with a Python interpreter. We adapt the SimpleTIR codebase (Xue et al., 2025), mask void turns in the loss, and train on data merged from SimpleRL (Zeng et al., 2025), Deepscaler (Luo et al., 2025), and rStar2-Agent (Shang et al., 2025). ALP follows the zero-RL setting and starts from Qwen2.5-7B-base. The rollout batch size is 512, the mini-update size is 32, the maximum response length is 16K, and each episode allows up to 5 code-execution turns. We use the same baselines and evaluation benchmarks as in the single-turn setting.

4.2 MAIN RESULTS

Algorithm	Math500	Minerva Math	Olympiad Bench	AIME24	AIME25	Average
GRPO	80.89 ± 0.27	41.92 ± 0.57	49.64 ± 0.46	35.23 ± 1.50	24.17 ± 0.89	46.37 ± 0.58
Seq-Bypass	81.42 ± 0.35	40.83 ± 0.32	50.70 ± 0.19	36.12 ± 0.76	25.23 ± 1.23	46.86 ± 0.12
Token-MIS	83.39 ± 0.14	41.67 ± 0.25	50.90 ± 0.15	39.58 ± 0.19	29.04 ± 0.98	48.92 ± 0.20
Seq-MIS	80.80 ± 0.43	41.55 ± 0.49	48.83 ± 0.22	39.51 ± 0.98	24.69 ± 1.16	47.07 ± 0.36
Token-ALP	83.48 ± 0.25	$\mathbf{43.18 \pm 0.64}$	51.25 ± 0.26	38.98 ± 1.55	$\mathbf{29.69 \pm 1.22}$	49.32 ± 0.30
Seq-ALP	$\mathbf{84.13 \pm 0.15}$	42.64 ± 0.09	$\mathbf{52.75 \pm 0.13}$	$\mathbf{43.72 \pm 1.54}$	28.54 ± 0.72	$\mathbf{50.36 \pm 0.26}$

Table 3: TIR evaluation: average@32 accuracy (%) at temperature 1.0 across five benchmarks. Entries report the mean $\pm$ 95% Student-t confidence interval half-width over five training seeds. Average is computed across the five benchmarks within each seed. Bold indicates the highest mean in each column.

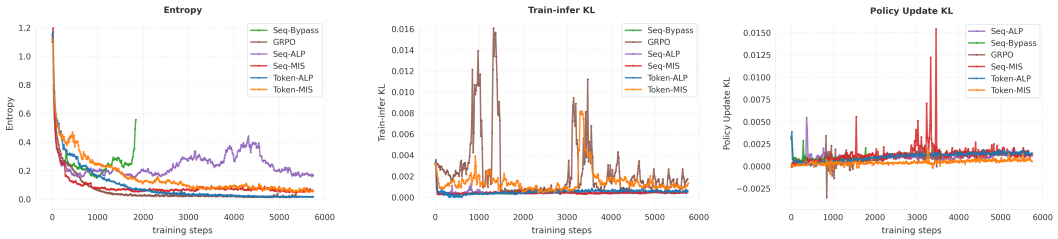


Figure 4: Multi-turn training dynamics: entropy, train–inference KL, and policy-update KL. Seq-ALP maintains relatively high entropy and mostly controlled KL, with an early policy-update KL spike. Token-ALP has a flatter policy-update KL trace, while Token-MIS shows larger train–inference KL excursions. Runs have different displayed lengths.

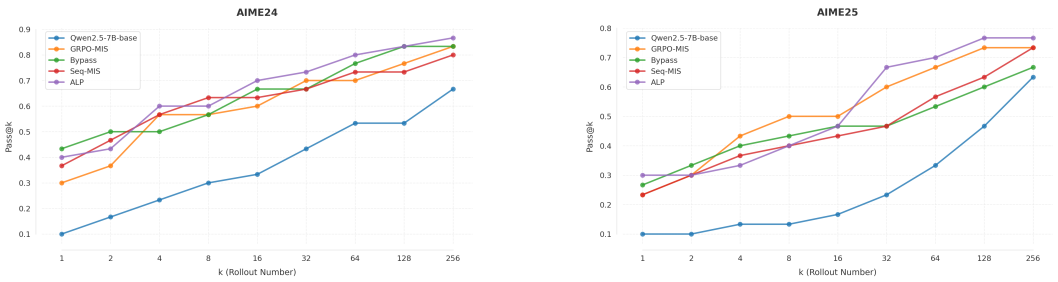


Figure 5: Pass@k on the 30-problem AIME 2024 and AIME 2025 sets in TIR. Seq-ALP (labeled ALP in the plots) attains the highest observed success rate from $k = 16$ on AIME 2024 and from $k = 32$ on AIME 2025 through $k = 256$. Each problem contributes about 3.3 percentage points. Plot labels GRPO-MIS and Bypass correspond to Token-MIS and Seq-Bypass.

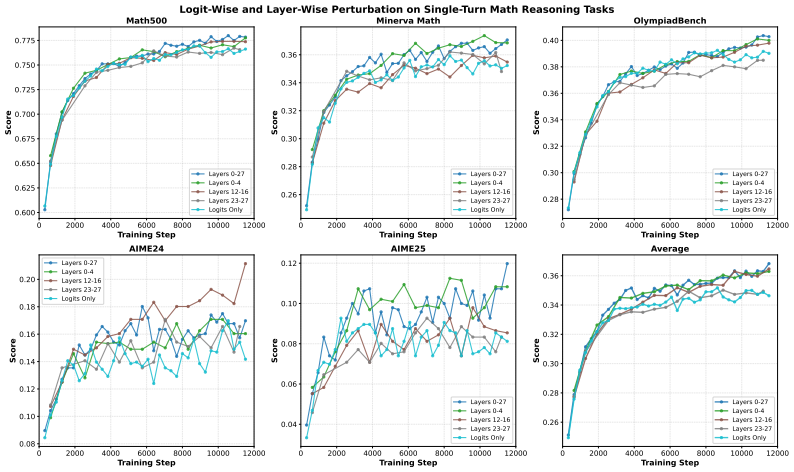


Figure 6: Perturbation-location ablation in the single-turn setting. All-layer, early-layer, and middle-layer perturbations have similar average trajectories; late-layer and logits-only perturbations generally trail.

ALP has the highest average performance. On the multi-turn TIR benchmarks in Table 3, Seq-ALP has the highest mean average accuracy (50.36), followed by Token-ALP (49.32) and the strongest baseline, Token-MIS (48.92). Seq-ALP has the highest mean on Math500, Olympiad Bench, and AIME24; Token-ALP leads on Minerva Math and AIME25. The reported intervals quantify variation across five training seeds.

ALP maintains controlled multi-turn diagnostics in the displayed runs. Figure 4 shows relatively high entropy for Seq-ALP and mostly controlled KL traces, alongside an early spike in policy-update KL. Token-ALP has the flattest displayed policy-update KL trace, while Token-MIS exhibits larger train–inference KL excursions.

Remark 1 (Token level and sequence level)**.** *Token-level ALP has the highest reported average in the single-turn setting, while sequence-level ALP has the highest mean average in the multi-turn setting. This pattern suggests a setting-dependent efficiency–bias trade-off: token-level clipping may preserve more learning signal, while sequence-level ratios account for policy drift across the trajectory.*

ALP improves success at fixed rollout budgets. In Figure 5, Seq-ALP has the highest observed Pass@k from $k = 16$ on AIME24 and from $k = 32$ on AIME25 through $k = 256$, producing a correct solution for more problems at the same sampling budget.

5 Ablations

5.1 Perturbation targets

We study how the *location* of Gaussian perturbations affects performance. Starting from the same base setup, we inject perturbations into (i) all transformer layers (0–27), (ii) early layers (0–4), (iii) middle layers (12–16), (iv) late layers (23–27), or (v) the output logits only (no hidden-state perturbation).

Single-turn location ablation. In Figure 6, all-layer perturbation reaches the highest final average, with the early- and middle-layer variants tracking it closely through training. Late-layer and logits-only variants generally have lower average accuracy.

Multi-turn Ablation. In multi-turn agentic evaluation (Table 4), all-layer perturbation reaches an average of 50.53, versus 48.25–48.66 for the partial bands. The late band has the highest average among the partial variants here, while early and middle layers track the all-layer variant most closely in single-turn training. Both settings favor broad hidden-state perturbation.

Perturbed layers	Math500	Minerva Math	Olympiad Bench	AIME24	AIME25	Average
Layers 0-5	83.58	43.62	49.88	38.85	25.31	48.25
Layers 12-17	82.19	41.70	51.69	39.69	27.29	48.51
Layers 23-27	82.77	**43.97**	49.60	40.10	26.87	48.66
Layers 0-27	**84.29**	43.10	**52.75**	**43.85**	**28.65**	**50.53**

Table 4: Perturbation-location ablation in multi-turn TIR. All-layer perturbation attains the highest average; the strongest partial band varies by benchmark. Table 3 reports the separate five-seed main comparison.

Interpreting the location effect. Perturbing every layer may provide more ways to vary the training policy than perturbing one narrow band. In Figure 8, all-layer perturbation generally has the lightest upper tail of the *absolute perturbation-induced change in token probability*, while the late-layer band has a heavier tail near the end of training. For low-probability tokens, even a small absolute change can produce a large relative change; the controlled one-step comparison in Figure 2 shows a narrower importance-ratio envelope in that region.

Perturbation-scale dynamics. Figure 8 (left) shows that learned scales stay within a factor of two of their 10^{-4} initialization across perturbation locations. Appendix E.1 analyzes the upward drift induced by weight decay on $\log \sigma_h$. The location study establishes the value of perturbing broadly. Table 5 further compares fixed and adaptive scales.

5.2 Fixed vs. adaptive perturbation scales

Scale	MATH-500	Minerva	OlympiadBench	AIME24	AIME25	Average
Fixed σ	77.86	35.87	39.67	17.19	10.31	36.18
Adaptive σ	**78.10**	**37.27**	**40.77**	**21.46**	**11.77**	**37.87**

Table 5: Token-ALP with fixed or adaptive scales: single-turn average@32 accuracy (%) at temperature 1.0.

To test whether adaptive scales help, we compare Qwen2.5-Math-1.5B Token-ALP with scale learning rates of 5×10^{-4} and 0, both initialized at $\sigma = 10^{-4}$. All other settings are unchanged. Fixed scales give lower accuracy on every benchmark: 36.18% on average versus 37.87% (Table 5). These results and the modest changes in Figure 8 (Section 5.1) suggest that small adjustments may help as model updates change the perturbation strengths suited to different layers. However, this comparison does not isolate gradient-driven changes from weight decay (Appendix E.1).

6 Conclusions

We introduced **Adaptive Layerwise Perturbation (ALP)**, which samples hidden-state perturbations with learned per-layer scales during policy updates and forms an importance ratio against the unperturbed rollout policy. ALP leads the evaluated methods across three backbones, including a 1.44-point gain over Token-MIS in the five-seed multi-turn comparison. The displayed training diagnostics show controlled KL, the matched-checkpoint comparison shows a narrower ratio tail, and the location ablations favor perturbing all layers. A smoothed-policy analysis explains how perturbation can reduce sensitivity to a modeled inference shift and smooth a surrogate objective.

AI USE STATEMENT

We used AI tools to assist with language polishing, code implementation, and literature research. The authors are responsible for the final manuscript, implementation, and references.

REPRODUCIBILITY STATEMENT

We provide the algorithm formulation and experimental settings in the main text, with formal assumptions and proofs in Appendices C and D, and additional data preparation procedures, training hyperparameters, and baseline settings in Appendix E. Our public code repository provides implementations and training scripts for ALP and the baselines in both single-turn and multi-turn reasoning settings, together with dataset preparation and evaluation instructions. The multi-turn documentation also describes the container environment and sandbox setup required for tool execution. The repository link is omitted here to preserve anonymity during double-blind review.

REFERENCES

Zhoujun Cheng, Shibo Hao, Tianyang Liu, Fan Zhou, Yutao Xie, Feng Yao, Yuexin Bian, Yonghao Zhuang, Nilabjo Dey, Yuheng Zha, Yi Gu, Kun Zhou, Yuqi Wang, Yuan Li, Richard Fan, Jianshu She, Chengqian Gao, Abulhair Saparov, Haonan Li, Taylor W. Killian, Mikhail Yurochkin, Zhengzhong Liu, Eric P. Xing, and Zhiting Hu. Revisiting reinforcement learning for llm reasoning from a cross-domain perspective, 2025. URL https://arxiv.org/abs/2506.14965.

Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. In *international conference on machine learning*, pp. 1310–1320. PMLR, 2019.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

Wei Fu, Jiaxuan Gao, Xujie Shen, Chen Zhu, Zhiyu Mei, Chuyi He, Shusheng Xu, Guo Wei, Jun Mei, Jiashu Wang, et al. Areal: A large-scale asynchronous reinforcement learning system for language reasoning. *arXiv preprint arXiv:2505.24298*, 2025.

Yifan Hao, Yanxin Lu, Hanning Zhang, Xinwei Shen, and Tong Zhang. Towards better generalization via distributional input projection network. *arXiv preprint arXiv:2506.04690*, 2025.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, et al. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. *arXiv preprint arXiv:2402.14008*, 2024.

Horace He and Thinking Machines Lab. Defeating nondeterminism in llm inference. *Thinking Machines Lab: Connectionism*, 2025. doi: 10.64434/tml.20250910. https://thinkingmachines.ai/blog/defeating-nondeterminism-in-llm-inference/.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Hugging Face. Open r1: A fully open reproduction of deepseek-r1, January 2025. URL https://github.com/huggingface/open-r1.

Liu* Jiacai, Li* Yingru, Fu Yuqian, Wang Jiawei, Qian Liu, and Zhuo Jiang. When speed kills stability: Demystifying RL collapse from the training-inference mismatch. https://yingru.notion.site/When-Speed-Kills-Stability-Demystifying-RL-Collapse-from-the-Training-Inference-Mismatch-271211a558b7808d8b12d403fd15edda, 2025. Accessed: 2026-01-28.

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.

Hynek Kydlíček. Math-Verify: Math Verification Library, n.d. URL https://github.com/huggingface/math-verify.

Mathias Lecuyer, Vaggelis Atlidakis, Roxana Geambasu, Daniel Hsu, and Suman Jana. Certified robustness to adversarial examples with differential privacy. In *2019 IEEE symposium on security and privacy (SP)*, pp. 656–672. IEEE, 2019.

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

Mingxiao Li, Tingyu Qu, Ruicong Yao, Wei Sun, and Marie-Francine Moens. Alleviating exposure bias in diffusion models through sampling with shifted time steps. *arXiv preprint arXiv:2305.15583*, 2023.

Xuelong Li. Positive-incentive noise. *IEEE Transactions on Neural Networks and Learning Systems*, 35(6):8708–8714, 2022.

Yingru Li, Jiacai Liu, Jiawei Xu, Yuxuan Tong, Ziniu Li, and Baoxiang Wang. Trust region masking for long-horizon llm reinforcement learning. *arXiv preprint arXiv:2512.23075*, 2025.

Liyuan Liu, Feng Yao, Dinghuai Zhang, Chengyu Dong, Jingbo Shang, and Jianfeng Gao. Flashrl: 8bit rollouts, full power rl, August 2025. URL https://fengyao.notion.site/flash-rl.

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. 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, 2025. Notion Blog.

Francisco J Moreno-Barea, Fiammetta Strazzera, José M Jerez, Daniel Urda, and Leonardo Franco. Forward noise adjustment scheme for data augmentation. In *2018 IEEE symposium series on computational intelligence (SSCI)*, pp. 728–734. IEEE, 2018.

Mang Ning, Mingxiao Li, Jianlin Su, Albert Ali Salah, and Itir Onal Ertugrul. Elucidating the exposure bias in diffusion models. *arXiv preprint arXiv:2308.15321*, 2023.

Lis Kanashiro Pereira, Yuki Taya, and Ichiro Kobayashi. Multi-layer random perturbation training for improving model generalization efficiently. In *Proceedings of the Fourth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pp. 303–310, 2021.

Penghui Qi, Zichen Liu, Xiangxin Zhou, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Defeating the training-inference mismatch via fp16. *arXiv preprint arXiv:2510.26788*, 2025.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

Chitwan Saharia, William Chan, Huiwen Chang, Chris Lee, Jonathan Ho, Tim Salimans, David Fleet, and Mohammad Norouzi. Palette: Image-to-image diffusion models. In *ACM SIGGRAPH 2022 conference proceedings*, pp. 1–10, 2022.

Hadi Salman, Jerry Li, Ilya Razenshteyn, Pengchuan Zhang, Huan Zhang, Sebastien Bubeck, and Greg Yang. Provably robust deep learning via adversarially trained smoothed classifiers. *Advances in neural information processing systems*, 32, 2019.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International conference on machine learning*, pp. 1889–1897. PMLR, 2015.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Ning Shang, Yifei Liu, Yi Zhu, Li Lyna Zhang, Weijiang Xu, Xinyu Guan, Buze Zhang, Bingcheng Dong, Xudong Zhou, Bowen Zhang, et al. rstar2-agent: Agentic reasoning technical report. *arXiv preprint arXiv:2508.20722*, 2025.

Xinwei Shen and Nicolai Meinshausen. Engression: Extrapolation for nonlinear regression? *arXiv preprint arXiv:2307.00835*, 2023.

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.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

Zhenghai Xue, Longtao Zheng, Qian Liu, Yingru Li, Xiaosen Zheng, Zejun Ma, and Bo An. Simpletir: End-to-end reinforcement learning for multi-turn tool-integrated reasoning. *arXiv preprint arXiv:2509.02479*, 2025.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, et al. Qwen2. 5-math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*, 2024.

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

Greg Yang, Tony Duan, J Edward Hu, Hadi Salman, Ilya Razenshteyn, and Jerry Li. Randomized smoothing of all shapes and sizes. In *International Conference on Machine Learning*, pp. 10693–10705. PMLR, 2020.

Feng Yao, Liyuan Liu, Dinghuai Zhang, Chengyu Dong, Jingbo Shang, and Jianfeng Gao. On the rollout-training mismatch in modern rl systems. OPT 2025: 17th Annual Workshop on Optimization for Machine Learning, 2025. URL https://opt-ml.org/papers/2025/paper116.pdf.

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, 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.

Xiaowei Yu, Zhe Huang, Minheng Chen, Yao Xue, Tianming Liu, and Dajiang Zhu. Noisynn: Exploring the impact of information entropy change in learning systems. *arXiv e-prints*, pp. arXiv–2309, 2023.

Weihao Zeng, Yuzhen Huang, Wei Liu, Keqing He, Qian Liu, Zejun Ma, and Junxian He. 7b model and 8k examples: Emerging reasoning with reinforcement learning is both effective and efficient. https://hkust-nlp.notion.site/simplerl-reason, 2025. Notion Blog.

Yanli Zhao, Andrew Gu, Rohan Varma, Liang Luo, Chien-Chin Huang, Min Xu, Less Wright, Hamid Shojanazeri, Myle Ott, Sam Shleifer, Alban Desmaison, Can Balioglu, Pritam Damania, Bernard Nguyen, Geeta Chauhan, Yuchen Hao, Ajit Mathews, and Shen Li. Pytorch fsdp: Experiences on scaling fully sharded data parallel, 2023. URL https://arxiv.org/abs/2304.11277.

Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, et al. Group sequence policy optimization. *arXiv preprint arXiv:2507.18071*, 2025.

Lianmin Zheng, Liangsheng Yin, Zhiqiang Xie, Chuyue Livia Sun, Jeff Huang, Cody Hao Yu, Shiyi Cao, Christos Kozyrakis, Ion Stoica, Joseph E Gonzalez, et al. Sglang: Efficient execution of structured language model programs. *Advances in neural information processing systems*, 37: 62557–62583, 2024.

A Additional Related Works

Off-policy Off-policy optimization is a challenging setting in reinforcement learning. Trust Region Policy Optimization (TRPO) (Schulman et al., 2015) constrains policy updates using KL divergence, while PPO (Schulman et al., 2017) uses a clipped surrogate. Long reasoning traces and multi-turn interactions can increase the effect of stale rollouts. Sequence-level importance ratios (Zheng et al., 2025) and masking (Li et al., 2025) offer ways to control such updates. Inference-engine quantization and batching can add training–inference mismatch. Recent methods (Yao et al., 2025; Jiacai et al., 2025) account for it with an additional ratio $\pi^{\text{train}}_{\theta_{\text{old}}}/\pi^{\text{infer}}_{\theta_{\text{old}}}$ and clip or mask large corrections.

Recent systems efforts aim to *eliminate* training-inference mismatch by enforcing bitwise-consistent inference, enabling closer-to on-policy RL in practice (He & Lab, 2025) or using higher precisions (Qi et al., 2025). These directions are promising, although strict consistency may require additional engineering or computation when execution engines or rollout precision differ. ALP is complementary: it uses one importance ratio with a sampled, perturbed training-policy numerator, and we evaluate it under rollout reuse and the mismatch of a vLLM–FSDP pipeline. Its behavior with other mismatch levels or fully asynchronous rollouts (Fu et al., 2025) remains to be tested.

B Technical Lemmas

Lemma 1 (Stam's inequality)**.** *For independent random vectors X and Y with smooth densities and finite Fisher information matrices, we have*

$$I(X+Y) \preceq I(X), \qquad I(X+Y) \preceq I(Y),$$

where $I(\cdot)$ is defined as the Fisher information matrix of the corresponding random variable.

Lemma 2. *If we assume $\delta \sim \mathcal{N}(0, \sigma^2 I_d)$, and define $\widetilde{\pi}_\theta(a|x) := \mathbb{E}_\delta \pi_\theta(a|x+\delta)$, then*

$$\nabla_x \ln \widetilde{\pi}_\theta(a|x) = \frac{1}{\sigma^2} \mathbb{E}_{\delta \sim q_\theta(\cdot|a,x)}[\delta],$$

where

$$q_\theta(\delta|a,x) = \frac{\pi_\theta(a|x+\delta)\phi(\delta)}{\widetilde{\pi}_\theta(a|x)}, \quad \phi(\delta) = \mathcal{N}(\delta; 0, \sigma^2 I_d).$$

Proof. First, we compute the gradient of $\widetilde{\pi}_\theta(a|x)$ with respect to x as:

$$\begin{aligned}
\nabla_x \widetilde{\pi}_\theta(a|x) &= \int \nabla_x \pi_\theta(a|x+\delta)\phi(\delta) d\delta \\
&= \int \nabla_\delta \pi_\theta(a|x+\delta)\phi(\delta) d\delta \\
&= -\int \pi_\theta(a|x+\delta)\nabla_\delta \phi(\delta) d\delta \\
&= \int \frac{\delta}{\sigma^2} \pi_\theta(a|x+\delta)\phi(\delta) d\delta \\
&= \frac{\widetilde{\pi}_\theta(a|x)}{\sigma^2} \int \delta \frac{\pi_\theta(a|x+\delta)\phi(\delta)}{\widetilde{\pi}_\theta(a|x)} d\delta \\
&= \frac{\widetilde{\pi}_\theta(a|x)}{\sigma^2} \mathbb{E}_{\delta \sim q_\theta(\cdot|a,x)} \delta,
\end{aligned}$$

The third equality uses integration by parts, with a vanishing boundary term. The fourth uses $\nabla_\delta \phi(\delta) = -\delta\phi(\delta)/\sigma^2$.

Substituting into $\nabla_x \ln \widetilde{\pi}_\theta(a|x)$ gives

$$\nabla_x \ln \widetilde{\pi}_\theta(a|x) = \frac{1}{\sigma^2} \mathbb{E}_{\delta \sim q_\theta(\cdot|a,x)}[\delta].$$

□

C Formal Statement and Proof for Theorem 1

Fix a neighborhood $\mathcal{U}$ in representation space. For $\widetilde{\pi}_\theta(a|y) > 0$, let $q_\theta(\delta|a, y)$ be the posterior in Lemma 2.

Condition 1 (Local score control)**.** *For every $y \in \mathcal{U}$ and action a,*

$$\left\|\mathbb{E}_{\delta\sim q_\theta(\cdot|a,y)}[\delta]\right\|_2^2 \le Cd\sigma^2$$

for a constant $C > 0$ uniform on $\mathcal{U}$.

Condition 2 (Local domination)**.** *The scale-dependent ratio*

$$\alpha_\sigma(\mathcal{U}) := \inf_{\substack{y\in\mathcal{U},\, a:\\ \widetilde{\pi}_\theta(a|y)>0}} \frac{\pi_\theta(a|y)}{\widetilde{\pi}_\theta(a|y)}$$

is strictly positive.

These are explicit local regularity assumptions. In particular, α_σ can decrease as smoothing changes the policy, and C need not be universal. They concern the one-input averaged model, not the sampled multilayer ALP objective.

For each (x, ζ) in the support of d_0 and the modeled shift, assume that the segment $[x, x + \zeta] \subset \mathcal{U}$ and define $\pi^{\text{infer}}_{\theta_{\text{old}}}(a|x) = \pi_{\theta_{\text{old}}}(a|x + \zeta)$. This is a local representation-shift abstraction, not a distributional assumption that real engine mismatch is Gaussian or additive at the system output.

Theorem 3 (Formal version of Theorem 1)**.** *Assume Conditions 1–2 and the differentiation conditions of Lemma 2. Define*

$$L_\sigma(\mathcal{U}) := \sup_{\substack{y\in\mathcal{U},\, a:\\ \widetilde{\pi}_{\theta_{\text{old}}}(a|y)>0}} \|\nabla_y \log \widetilde{\pi}_{\theta_{\text{old}}}(a|y)\|_2.$$

Then $L_\sigma(\mathcal{U}) \le \sqrt{Cd}/\sigma$ and

$$\begin{aligned} &\mathbb{E}_{x\sim d_0,\zeta} D_{\text{KL}}\big(\widetilde{\pi}_{\theta_{\text{old}}}(\cdot|x) \,\|\, \pi^{\text{infer}}_{\theta_{\text{old}}}(\cdot|x)\big) \\ &\quad\le -\log \alpha_\sigma(\mathcal{U}) + L_\sigma(\mathcal{U})\mathbb{E}\|\zeta\|_2 \\ &\quad\le -\log \alpha_\sigma(\mathcal{U}) + \frac{\sqrt{Cd}}{\sigma}\mathbb{E}\|\zeta\|_2. \end{aligned} \tag{5}$$

Proof. Lemma 2 and Condition 1 give $L_\sigma(\mathcal{U}) \le \sqrt{Cd}/\sigma$. Since $[x, x + \zeta] \subset \mathcal{U}$, the fundamental theorem of calculus yields, for every action,

$$\log \widetilde{\pi}_{\theta_{\text{old}}}(a|x) - \log \widetilde{\pi}_{\theta_{\text{old}}}(a|x + \zeta) \le L_\sigma(\mathcal{U})\|\zeta\|_2.$$

Condition 2 at $x + \zeta$ gives $\pi^{\text{infer}}_{\theta_{\text{old}}}(a|x) \ge \alpha_\sigma(\mathcal{U})\widetilde{\pi}_{\theta_{\text{old}}}(a|x + \zeta)$. Insert both inequalities into the KL definition, average first over $a \sim \widetilde{\pi}_{\theta_{\text{old}}}(\cdot|x)$, and then over (x, ζ). This proves Equation (5). □

D Proof for Theorem 2

Theorem 4. *Fix θ and a twice-differentiable, unclipped surrogate $J(x, \theta)$, $x \in \mathbb{R}^d$. Let $b := \sup_x \|\nabla^2_\theta J(x, \theta)\|_2 \in (0, \infty)$. For $c \in (0, 1)$, assume*

$$S_c := \{x : \|\nabla^2_\theta J(x, \theta)\|_2 \ge cb/2\}$$

has finite Lebesgue volume V_c. Define $\widetilde{J}_\sigma(x, \theta) := \mathbb{E}_{\delta\sim\mathcal{N}(0,\sigma^2 I_d)} J(x + \delta, \theta)$ and assume differentiation under the expectation is valid. If

$$\sigma \ge \frac{1}{\sqrt{2\pi}} \left(\frac{2V_c}{c}\right)^{1/d},$$

then $\sup_x \|\nabla^2_\theta \widetilde{J}_\sigma(x, \theta)\|_2 \le cb$.

Proof. The Gaussian density is at most $(2\pi\sigma^2)^{-d/2}$; hence, uniformly in x,

$$\Pr(x+\delta \in S_c) \le V_c(2\pi\sigma^2)^{-d/2} \le c/2.$$

Differentiating under the expectation and splitting over S_c and its complement gives

$$\begin{aligned}\|\nabla_\theta^2 \widetilde{J}_\sigma(x,\theta)\|_2 &\le \mathbb{E}_\delta \|\nabla_\theta^2 J(x+\delta,\theta)\|_2 \\ &\le b\,\Pr(x+\delta \in S_c) + (cb/2)\Pr(x+\delta \notin S_c) \\ &\le cb.\end{aligned}$$

Taking the supremum proves the claim. □

The finite-volume hypothesis is an explicit regularity condition excluding high curvature spread across almost all representation space. The theorem gives a sufficient scale for this simplified surrogate; it neither characterizes the learned ALP scale nor transfers automatically to sampled multilayer updates.

E Additional Experimental Details

Single-turn details. The single-turn training set is constructed from the math subset of Guru RL-92k and a 75K OpenR1 subset. Guru contains 91.9K high-quality samples spanning six reasoning-intensive domains, while OpenR1 is derived from a larger corpus of math problems with two to four reasoning traces generated by DeepSeek R1. For prompts in the single-turn task, we filter OpenR1 examples by their average score in $(0.2, 0.8)$, and then deduplicate the merged corpus by user-prompt content. Prompts are converted into a system-plus-user format with a fixed system instruction that requests a final `\boxed{}` answer. Rollouts are generated with vLLM(Kwon et al., 2023), while training is performed with Fully Sharded Data Parallel (FSDP)(Zhao et al., 2023). We set `max_prompt_length=2048` and `max_response_length=2048`. We use dual-clip PPO with `clip_ratio_c=10.0` throughout. Each step samples 512 prompts and generates $n = 8$ responses per prompt. The policy is optimized with AdamW (learning rate 10^{-6}; weight decay 0.01). Rewards are computed via rule-based math verification using the `math-verify` Python package, by extracting the final `\boxed{}` answer and matching it to the ground truth. We set the KL loss coefficient to 0.001 and entropy loss coefficient to 0.001. The models are trained with 8 H100 GPUs and 4 A100 GPUs. For the additional Qwen3-4B validation in Table 2, we keep the same training and evaluation protocol and only change the backbone model.

We further provide the clipping or masking details for the baselines in the following table. For

Baselines	$\pi_\theta^{\text{train}}/\pi_{\theta_{\text{old}}}^{\text{train}}$	$\pi_{\theta_{\text{old}}}^{\text{train}}/\pi_{\theta_{\text{old}}}^{\text{infer}}$	$\pi_\theta^{\text{train}}/\pi_{\theta_{\text{old}}}^{\text{infer}}$
GRPO	token	None	None
Token-MIS	token	seq	None
Seq-MIS	seq	seq	None
Seq-Bypass	None	None	seq

Table 6: Settings of parameters of importance ratios

the clipping or masking threshold parameters, they are chosen to achieve stable performance for different settings respectively. Specifically, for token-level importance ratio, the lower clipping ratio is 0.2, and the higher clipping ratio is 0.2 or 0.28; for sequence-level importance ratio, the lower clipping ratio is 0.5, and the higher one is 3.0. Besides, for MIS, the masking threshold is 2.0.

Multi-turn details. For multi-turn experiments, we use Tool-Integrated Reasoning (TIR) with a Python interpreter, allowing the model to execute code for intermediate verification and symbolic computation. We adapt the SimpleTIR codebase and mask void turns in the loss, where void turns are turns without tool calls and without final answers. The training data merges Math3-5 examples from SimpleRL, Deepscaler, and rStar2-Agent. Training starts from Qwen2.5-7B-base under the zero-RL setting, with rollout batch size 512, mini-update size 32, maximum response length 16K, and at most 5 code-execution turns per episode.



Figure 7: Per-benchmark single-turn test accuracy over training. GRPO and Seq-MIS curves end after their late accuracy drop; the remaining methods continue to later steps.

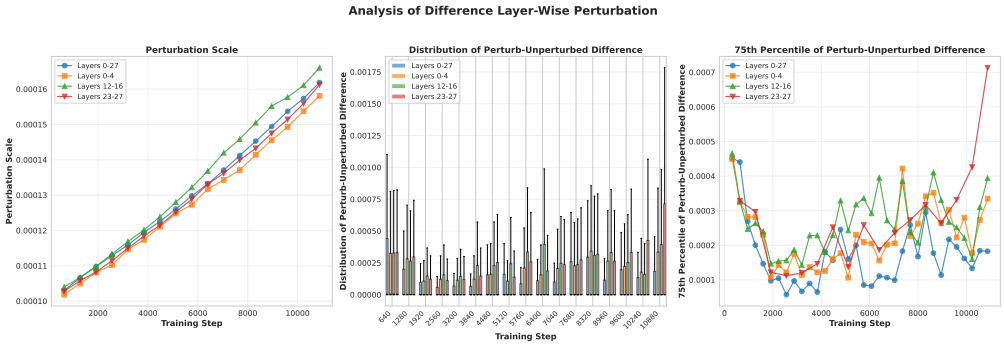


Figure 8: Layer-wise perturbation diagnostics across target choices: (left) mean learned noise scale σ, (middle) distribution of perturbation-induced probability differences $|\Delta p|$, and (right) tail behavior measured by the 75th percentile of $|\Delta p|$ over training steps.

E.1 Perturbation Parameterization and Update

Parameterization. Each decoder layer h in the perturbed range holds one scalar parameter τ_h and uses the scale $\sigma_h = \exp(\tau_h)$, which keeps σ_h positive without a constraint. All τ_h are initialized to $\log \sigma_0$ with $\sigma_0 = 10^{-4}$ (Appendix E.2). During a policy-update forward pass, the input hidden states of layer h receive $\sigma_h \epsilon$ with $\epsilon \sim \mathcal{N}(0, I)$ drawn independently for every token position and layer; the noise is generated from a per-layer seed so that gradient-checkpoint recomputation reproduces the same sample. Perturbation is active only in training mode, so rollout generation and evaluation use the unperturbed network.

Update. Because $\delta^h = \sigma_h \epsilon^h$ is a reparameterized sample, τ_h receives gradients from the same clipped objective in Equation (2) as θ; there is no separate loss, penalty, or clamp for σ. The τ_h form a separate AdamW parameter group with learning rate $\eta_\sigma = 5 \times 10^{-4}$, while the model parameters use 10^{-6}. The group inherits the actor's weight decay $\lambda = 0.01$, which acts on τ_h and therefore pulls σ_h toward 1.

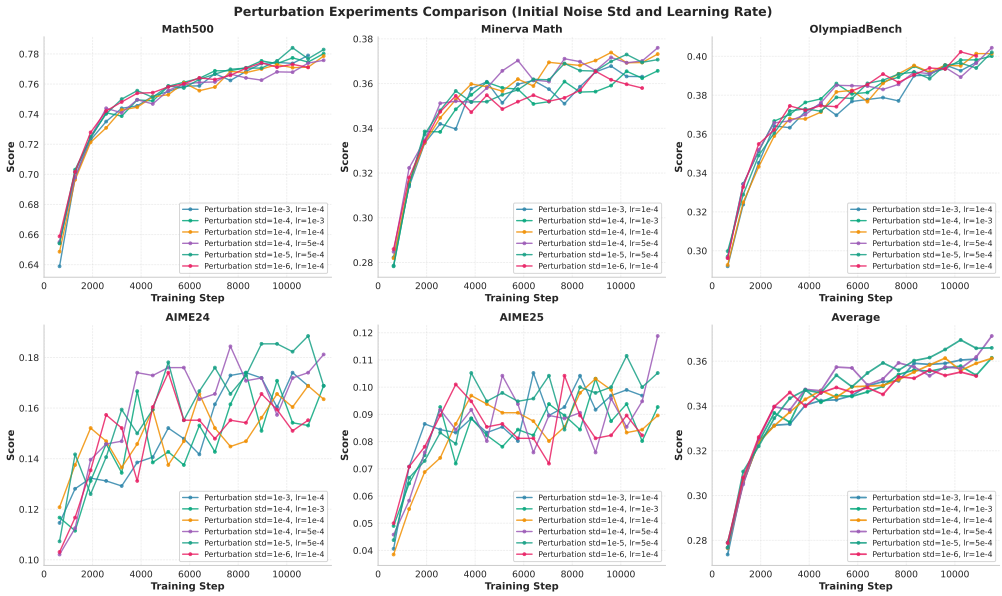


Figure 9: Hyperparameter sensitivity in the single-turn setting. We vary the initial Gaussian noise standard deviation (std) and its learning rate (lr), and report average@32 across Math500, Minerva Math, OlympiadBench, AIME24, and AIME25.

Scale dynamics. Decoupled weight decay multiplies τ_h by $(1 - \eta_\sigma \lambda)$ at every optimizer step, independently of the data. Starting from $\tau_h = \log 10^{-4} \approx -9.2$, this term alone moves σ_h from 1.0×10^{-4} to 1.6×10^{-4} within 1.1×10^4 optimizer steps, an upward drift at least as large as the one in Figure 8 (left). The near-identical trajectories across perturbation locations are consistent with this data-independent drift. The figure therefore shows that the learned scales remain within a factor of two of their initialization; it does not isolate the direction in which the policy gradient moves σ. The fixed-scale comparison in Table 5 shows a benefit from updating the scales, but does not separate policy-gradient effects from weight decay. Further controls, such as removing weight decay from the perturbation group, are needed to distinguish these effects.

E.2 Hyperparameter sensitivity

We study the sensitivity of ALP to two key hyperparameters: the initial Gaussian noise standard deviation (std) and the learning rate (lr) for the perturbation parameters. We evaluate the resulting policies on Math500, Minerva Math, OlympiadBench, AIME24, and AIME25, and report the average@32 score.

Figure 9 compares runs with learned scales initialized at different standard deviations and trained with different perturbation-parameter learning rates. Their performance curves are tightly clustered over the tested settings, suggesting limited sensitivity to these choices. All runs in this sweep update σ; the separate fixed-scale comparison is reported in Table 5. The inherited weight decay also changes $\log \sigma$, so neither comparison isolates the contribution of policy-gradient updates (Appendix E.1). Based on final-stage performance, we use std $= 1 \times 10^{-4}$ and lr $= 5 \times 10^{-4}$ as the default configuration in subsequent experiments.

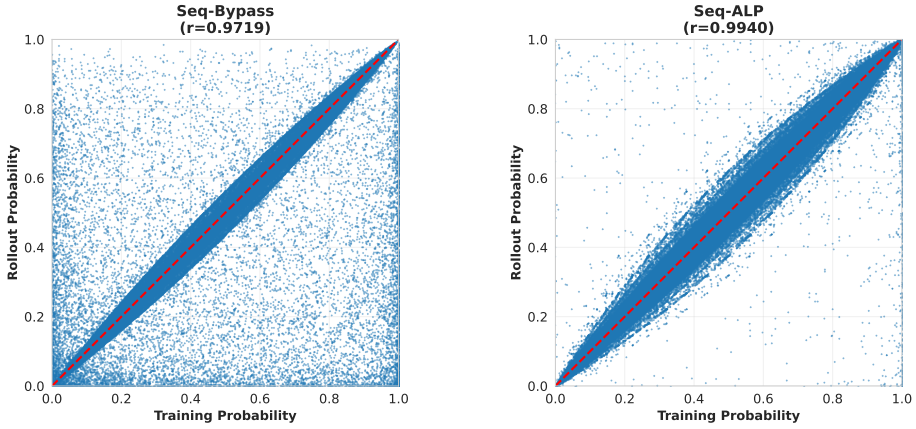


Figure 10: Token-level training-inference mismatch at step 140 in the multi-turn setting. Each point is one response token, with x-axis denoting the training probability $\pi_{\theta_{\text{old}}}^{\text{train}}$ and y-axis denoting the rollout-time token probability $\pi_{\theta_{\text{old}}}^{\text{infer}}$. The red dashed diagonal is $y = x$; tighter concentration around this line indicates smaller training-inference mismatch. The Pearson correlation coefficient r is reported in each panel title.