RETHINKING TRAINING–INFERENCE MISMATCH IN LLM REINFORCEMENT LEARNING: WHERE IT ARISES AND HOW TO CORRECT IT

Anonymous authors
Paper under double-blind review

ABSTRACT

We study training–inference mismatch in reinforcement learning with verifiable rewards (RLVR) for large language models, where rollouts are sampled by an inference engine while gradients are computed by a training engine, and the two engines assign different probabilities to the same tokens. To account for this discrepancy in policy updates, we introduce calibrated importance sampling (CIS). CIS is motivated by an empirically supported logit-displacement characterization that expresses the mismatch as an additive displacement ε_t in log-odds, determined by the per-logit perturbation before the softmax, whose distribution is approximately invariant to token confidence. This characterization motivates a confidence-aware truncation: large positive displacements are truncated at a single constant threshold, which maps back to an importance-ratio cap that tightens as token confidence increases. Theoretically, we show that CIS replaces the unbounded second moment that governs the error of exact importance sampling with a term bounded by a constant, at the cost of a bias controlled by the truncated excess. In evaluation across three mixture-of-experts models and five mathematical reasoning benchmarks, CIS achieves the highest five-benchmark average on all three models among the evaluated baselines. Diagnostic analyses show that CIS places less truncation bias on low-confidence tokens than truncated importance sampling, while upward clipping of small importance weights reduces held-out accuracy.

1 INTRODUCTION

Reinforcement learning with verifiable rewards (RLVR) has become a dominant paradigm for improving the reasoning abilities of large language models (LLMs) (Shao et al., 2024; DeepSeek-AI, 2025; Yu et al., 2025). To improve throughput, current RL post-training frameworks decouple rollout generation from gradient computation and assign them to an inference engine and a training engine, respectively (Sheng et al., 2025; Hu et al., 2025; Fu et al., 2025). Although the two engines load the same parameters, differences in their numerical implementations lead them to assign different probabilities to the same token. As a result, training that is nominally on-policy becomes off-policy (Yao et al., 2025; Qi et al., 2025; Zheng et al., 2025a). This *training–inference mismatch* is particularly severe for mixture-of-experts models, where it can cause training instability or even collapse (Ling Team & Inclusion AI, 2025; Zheng et al., 2025b; Ma et al., 2025; Zheng et al., 2025a).

The standard remedy for this mismatch is importance sampling, which reweights the gradient of each token by the ratio between the probabilities that the training and inference engines assign to the sampled token (Yao et al., 2025; Liu et al., 2025; Zheng et al., 2025a). However, occasional sharp disagreements between the two engines on individual tokens can make this importance ratio extremely large (Zhao et al., 2025; Ma et al., 2025). Exact correction can therefore yield gradient estimates with uncontrolled variance (Ionides, 2008; Metelli et al., 2018; Liu et al., 2025). Truncating the importance sampling factor controls this variance by introducing the bias–variance trade-off. Understanding this bias–variance trade-off and designing better truncation methods has emerged as a new question to answer (Ionides, 2008; Liu et al., 2025).

Existing methods largely rely on heuristic rules to truncate or mask this ratio. At the token level, TIS caps the ratio at a fixed upper bound (Yao et al., 2025; Qi et al., 2025), IcePop masks tokens

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

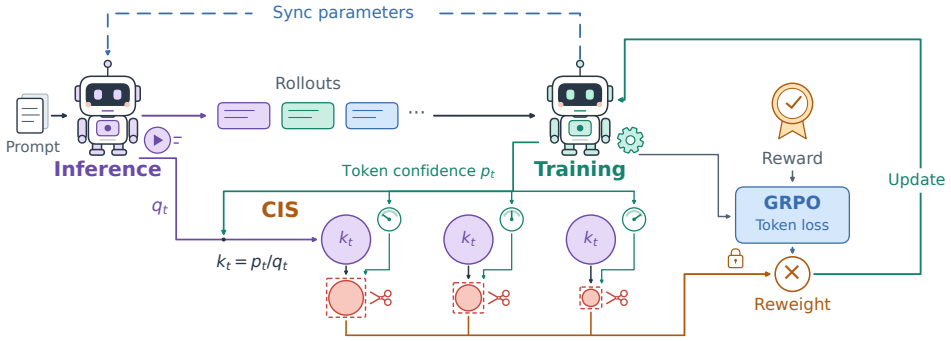


Figure 1: The inference and training engines assign different probabilities to the same tokens. CIS truncates the importance ratios with confidence-dependent caps, which are tighter for more confident tokens. The truncated ratios reweight the token-level GRPO loss (Shao et al., 2024).

whose ratio falls outside a fixed interval (Zhao et al., 2025; Ling Team & Inclusion AI, 2025), and KPop masks tokens by thresholding the KL divergence between the probabilities of the inference and training engines (Guo et al., 2026; Ling Team, 2026). At the sequence level, existing methods truncate or reject the ratio of the entire response (Liu et al., 2025), or replace the token-level objective with a sequence-level one (Zheng et al., 2025b). Other work addresses the problem through numerical precision and runs the entire pipeline in FP16 (Qi et al., 2025). Although these methods are effective to varying degrees, all of their rules act directly on the importance ratio or on quantities computed directly from the probabilities of the two engines. This ratio, however, mixes two factors: the actual discrepancy between the two engines and the confidence of the token itself. A discrepancy of the same size moves the ratio far from one on low-confidence tokens but barely changes it on high-confidence tokens. As a result, fixed thresholds act mainly on low-confidence tokens (Guo et al., 2026; Ling Team, 2026), and the truncation bias concentrates on these tokens. This motivates our question:

Can we design a principled mismatch correction grounded in the statistical structure of the training–inference mismatch itself?

We address this question by proposing **CIS** (**C**alibrated **I**mportance **S**ampling), which is illustrated in Figure 1. CIS first separates the two factors in the importance ratio, which can be written exactly as a function of the token confidence and a component that reflects only the discrepancy between the two engines. Our measurements show that the distribution of this discrepancy component varies only slightly with token confidence, so anomalous discrepancies can be identified with the same threshold at every confidence level. CIS applies a single constant threshold to this component, which truncates anomalously large discrepancies at their source. When mapped back to the ratio, this threshold imposes a cap on the importance weight of each token, which depends on the token confidence and becomes tighter as confidence increases. This design determines where the truncation bias falls. Discrepancies of the same size are treated in the same way at every confidence level, so the truncation bias no longer concentrates on low-confidence tokens. Our contributions are as follows:

- We decompose the importance ratio into token confidence and a discrepancy component, and find in large-scale analyses that the distribution of the discrepancy component barely varies with token confidence. With mixture-of-experts models, routing disagreement between engines substantially amplifies the heavy tail of this distribution.
- Building on this characterization, we propose CIS, which truncates the discrepancy component rather than the ratio, yielding a ratio cap that tightens with token confidence. Theoretically, we prove that CIS bounds the potentially explosive second-moment term of exact correction, at the cost of a bias controlled by the truncated excess.
- Experiments on three mixture-of-experts models and five mathematical reasoning benchmarks validate the effectiveness of CIS. Diagnostic analyses show that, unlike a fixed ratio cap, CIS does not concentrate its truncation bias on low-confidence tokens and attains a lower overall bias.

2 PRELIMINARIES

2.1 TRAINING–INFERENCE MISMATCH

In current RL post-training frameworks, rollouts are generated by an inference engine (e.g., vLLM and SGLang), while gradients are computed by a separate training engine (e.g., FSDP and Megatron). Both engines load the same parameters θ, but differences in kernel implementations, reduction orders, and numerical precision lead them to assign different probabilities to the same token. We write $h_t := (x, y_{<t})$ for the prefix at position t. Then $\pi_{\text{train}}(y_t \mid h_t; \theta) \neq \pi_{\text{infer}}(y_t \mid h_t; \theta)$, so rollouts that are nominally on-policy are in fact off-policy with respect to the training engine. The problem is more severe for mixture-of-experts models, where routing is a discrete top-k decision and a small perturbation can change which experts are activated.

For the token-level GRPO surrogate evaluated on the observed rollout prefixes, correcting the conditional distribution of each sampled token introduces an importance ratio:

$$\mathcal{J}(\theta) = \mathbb{E}_{x\sim\mathcal{D},\, \{y_i\}_{i=1}^{G}\sim\pi_{\text{infer}}(\cdot|x;\theta_{\text{old}})} \left[\frac{1}{G}\sum_{i=1}^{G} \frac{1}{|y_i|} \sum_{t=1}^{|y_i|} k_{i,t}\, \min\left(\rho_{i,t}\hat{A}_{i,t},\ \text{clip}\left(\rho_{i,t}\right)\hat{A}_{i,t}\right) \right], \quad (1)$$

where $h_{i,t} = (x, y_{i,<t})$, θ_{old} is the parameter snapshot at rollout time, $\hat{A}_{i,t}$ is the group-normalized advantage, and clip restricts $\rho_{i,t}$ to $[1 - c_{\text{low}},\, 1 + c_{\text{high}}]$. Let $p_{i,t} = \pi_{\text{train}}(y_{i,t} \mid h_{i,t}; \theta_{\text{old}})$ and $q_{i,t} = \pi_{\text{infer}}(y_{i,t} \mid h_{i,t}; \theta_{\text{old}})$ be the probabilities that the two engines assign to the sampled token. The update ratio is $\rho_{i,t} = \pi_{\text{train}}(y_{i,t} \mid h_{i,t}; \theta)/p_{i,t}$, and the mismatch ratio is $k_{i,t} = p_{i,t}/q_{i,t}$. The mismatch ratio $k_{i,t}$ corrects only the next-token distribution given the observed prefix $h_{i,t}$; it does not reweight the distribution of the prefix itself. The update ratio $\rho_{i,t}$ is handled by the standard PPO clipping, while the mismatch ratio $k_{i,t}$ carries the discrepancy between the two engines and is the focus of the rest of this paper.

2.2 CORRECTING THE MISMATCH

The most direct way to correct the training–inference mismatch is to use the exact ratio $k_{i,t}$ in Equation (1). This correction is unbiased, but on the few tokens where the two engines disagree sharply, the ratio becomes very large and the gradient estimate has uncontrolled variance. Existing methods therefore truncate or mask $k_{i,t}$ and accept some bias in exchange for lower variance.

At the token level, TIS replaces $k_{i,t}$ with $\min(k_{i,t}, C)$ for a fixed cap C (Yao et al., 2025). IcePop instead uses $k_{i,t}\,\mathbf{1}\{\alpha \le k_{i,t} \le \beta\}$, which masks any token whose ratio falls outside a fixed interval (Zhao et al., 2025; Ling Team & Inclusion AI, 2025). KPop masks a token when the binary KL divergence between the probabilities of the two engines exceeds a fixed threshold (Guo et al., 2026; Ling Team, 2026). At the sequence level, Seq-TIS and Seq-MIS apply the same cap or mask to the product of the ratios over a response (Liu et al., 2025), and GSPO optimizes a sequence-level objective instead (Zheng et al., 2025b). A different line of work leaves the reweighting unchanged and reduces the discrepancy itself: FP16 training raises numerical precision (Qi et al., 2025), and routing replay makes the two engines select the same experts (Ma et al., 2025). All of these approaches are effective to some extent. However, the thresholds of the reweighting methods either act on $k_{i,t}$ directly and treat every token in the same way, or are not derived from the statistics of the mismatch itself. In Section 3, we start from the logit perturbation between the two engines, characterize the structure of the mismatch, and build our correction on this characterization.

3 CALIBRATED IMPORTANCE SAMPLING

3.1 FROM LOGIT PERTURBATION TO LOG-ODDS DISPLACEMENT

We measure, on large-scale rollouts, the probabilities that the two engines assign to the same sampled token (experimental details in Section A.1). On mixture-of-experts models, the top-k selection of the gating network is a discrete decision, so a small numerical difference can make the two engines activate different experts at some layer, and this difference propagates through all subsequent layers. These differences accumulate layer by layer and ultimately appear as different logits output by

the two engines under the same parameters and the same prefix. Write $\mathbf{z} = (z_1, \ldots, z_V) \in \mathbb{R}^V$ for the logit vector of the training engine at the prefix h_t, and $\mathbf{z} + \boldsymbol{\delta}$ for that of the inference engine, where $\boldsymbol{\delta} = (\delta_1, \ldots, \delta_V) \in \mathbb{R}^V$ is the per-logit perturbation between the two engines. Then $p_t = \pi_{\text{train}}(y_t \mid h_t; \theta_{\text{old}}) = \text{softmax}(\mathbf{z})_{y_t}$ and $q_t = \pi_{\text{infer}}(y_t \mid h_t; \theta_{\text{old}}) = \text{softmax}(\mathbf{z} + \boldsymbol{\delta})_{y_t}$. Substituting the two softmaxes into the mismatch ratio $k_t := p_t/q_t$ yields the following identity.

Lemma 3.1 (Ratio–displacement identity). *Let* $w_j := e^{z_j} / \sum_{j' \neq y_t} e^{z_{j'}}$ *for* $j \neq y_t$. *Then the mismatch ratio satisfies*

$$k_t = \frac{\text{softmax}(\mathbf{z})_{y_t}}{\text{softmax}(\mathbf{z} + \boldsymbol{\delta})_{y_t}} = p_t + (1 - p_t)\exp(\varepsilon_t), \qquad k_t - 1 = (1 - p_t)\big(\exp(\varepsilon_t) - 1\big), \tag{2}$$

where $\varepsilon_t := -\delta_{y_t} + \log \sum_{j \neq y_t} w_j \exp(\delta_j)$.

The proof is given in Section B.1. We call ε_t the log-odds displacement of the inference engine relative to the training engine on the current token. From Equation (2), $\varepsilon_t = \log \frac{p_t}{1-p_t} - \log \frac{q_t}{1-q_t}$, that is, ε_t is exactly the difference between the log-odds that the two engines assign to the sampled token. In the log-odds coordinate, the discrepancy between the two engines is therefore an additive displacement determined entirely by the perturbation $\boldsymbol{\delta}$, and it can be computed directly from the log-probabilities recorded by the two engines. Figure 2 shows the measured $\boldsymbol{\delta}$ and ε_t. On the dense control, $\boldsymbol{\delta}$ is concentrated near zero and bounded, consistent with bf16 rounding error, and ε_t is correspondingly concentrated tightly around zero. On the MoE, the body of $\boldsymbol{\delta}$ is likewise concentrated near zero, but is distinguished by a pronounced heavy tail. We analyze the source of this tail in Sections A.3 and A.4: rounding error accounts only for the body, while routing disagreement between the two engines is a major amplifier of the tail, and the tail of $\boldsymbol{\delta}$ becomes heavier as the two engines select different experts at more layers. Since $\varepsilon_t := -\delta_{y_t} + \log \sum_{j \neq y_t} w_j \exp(\delta_j)$, the heavy tail of $\boldsymbol{\delta}$ carries over to ε_t, so the log-odds displacement on the MoE is heavy-tailed as well. Given p_t, by Equation (2), the ratio is a deterministic, increasing function of ε_t: the mismatch enters the importance weight through the displacement factor e^{ε_t}, rescaled by $1 - p_t$. The same displacement therefore moves the ratio far from one on low-confidence tokens, whereas on high-confidence tokens the factor $1 - p_t$ compresses it and the ratio barely moves.

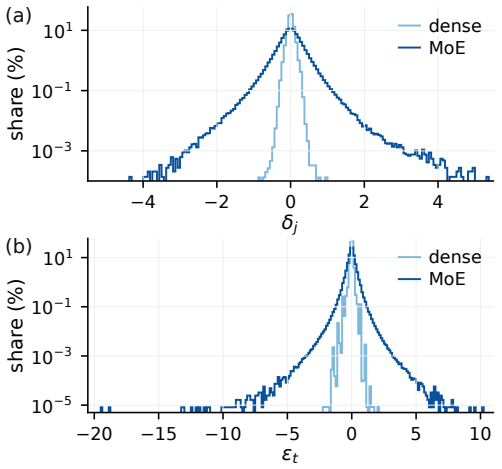


Figure 2: (a) Per-logit perturbation δ_j and (b) log-odds displacement ε_t, MoE vs. dense.

3.2 Why exact correction is unstable

A gradient update uses n tokens, indexed by m. Write the payload of the m-th token as $\ell'_m = \nabla_\theta \min\big(\rho_m \hat{A}_m, \text{clip}(\rho_m)\hat{A}_m\big)$; the mismatch ratio k_m is the importance sampling factor used to scale this gradient. Since k_m does not depend on θ, the gradient of the token-level surrogate objective and its estimator are

$$G_0 := \nabla_\theta \mathcal{J} = \mathbb{E}[k\,\ell'], \qquad \widehat{G} := \frac{1}{n}\sum_{m=1}^{n} k_m\,\ell'_m,$$

where the expectation is over the prompt, rollout, and numerical discrepancy between the two engines for each token. G_0 corrects the conditional distribution of each token given its prefix, not the distribution of the prefix itself. We measure the quality of the estimate by the mean-squared error $\mathbb{E}\big\|\widehat{G} - G_0\big\|_2^2$.

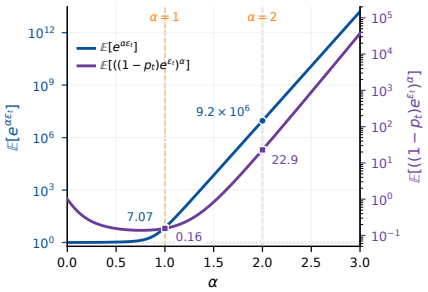


Figure 3: $\mathbb{E}[e^{\alpha\varepsilon_t}]$ (left axis) and $\mathbb{E}[((1-p_t)e^{\varepsilon_t})^\alpha]$ (right axis) on the measured displacements of the trained MoE.

Theorem 3.2 (Informal). *Assume $\|\ell'_t\|_2 \le b$ and that the contributions of different tokens are independent (for correlated contributions, replace n by n_{eff}; see Section B). Then*

$$\mathbb{E}\big\|\widehat{G} - G_0\big\|_2^2 \;\le\; \frac{b^2}{n}\,\mathbb{E}\Big[p_t^2 + 2p_t(1-p_t)\exp(\varepsilon_t) + (1-p_t)^2\, e^{2\varepsilon_t}\Big]. \tag{3}$$

Figure 3 shows both moments on the displacements of the MoE after RL training (Section A.7). The raw moment $\mathbb{E}[e^{\alpha\varepsilon_t}]$ is only 7.07 at $\alpha = 1$ but reaches 9.2×10^6 at $\alpha = 2$. The weighted moment $\mathbb{E}[((1-p_t)e^{\varepsilon_t})^\alpha]$, which is the quantity that enters Equation (3), is compressed by the factor $(1-p_t)^\alpha$ but follows the same pattern: it rises from 0.16 at $\alpha = 1$ to 22.9 at $\alpha = 2$. Since $p_t^2 \le 1$ and the cross term is at most 2×0.16, the bound (Equation (3)) is dominated by the second-moment term and is more than twenty times the corresponding bound without correction, for which the expectation equals one. Exact correction is therefore unbiased, but its variance is driven by the upper tail of ε_t.

This changes the problem from whether the mismatch should be corrected to how the exact correction should be truncated. Some bias must be accepted in exchange for controlling its second moment. The next section constructs such a truncation directly in the coordinate from which the mismatch arises.

3.3 Correction in the displacement coordinate

Given p_t, the probability of generating a token, the mismatch enters the ratio only through e^{ε_t}, by Equation (2):

$$k_t = p_t + (1-p_t)\exp(\varepsilon_t).$$

The variance problem is intrinsically one-sided. If $e^{\varepsilon_t} \le 1$, then

$$p_t < k_t \le 1,$$

so the importance weight on this side satisfies $k_t^2 \le 1$. In contrast, when $\exp(\varepsilon_t) > 1$, the ratio exceeds one and can grow without bound as the positive displacement increases. Thus the second-moment explosion identified in Section 3.2 can only arise from the upper tail. Modifying the lower side does not address this source of unbounded variance and instead introduces additional deviation from the exact correction. We therefore truncate only large positive displacements. It remains to choose where to truncate the upper tail.

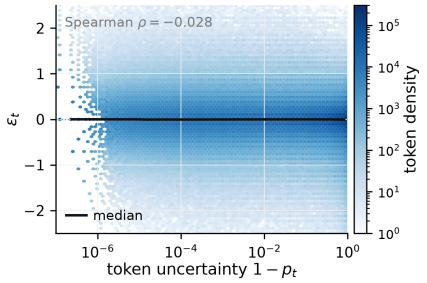


Figure 4: ε_t against token uncertainty $1 - p_t$ on the base MoE; the black line is the median.

The displacement itself is only weakly related to confidence. Figure 4 plots ε_t against $1 - p_t$ on the base MoE (Section A.6): across six orders of magnitude of uncertainty, the median stays at zero and the spread changes by less than a factor of two with a Spearman rank correlation of only -0.028. A displacement of a given amount is therefore approximately equally anomalous across confidence levels. This suggests placing a single upper threshold directly in the displacement coordinate with $\exp(\varepsilon_t) \le 1 + \lambda$ where threshold $\lambda \ge 0$.

Truncating e^{ε_t} at $1 + \lambda$ and mapping back through Equation (2) gives

$$f_\lambda(k_t, p_t) \;=\; p_t + (1-p_t)\min\big\{e^{\varepsilon_t},\, 1+\lambda\big\} \;=\; \min\big\{k_t,\; 1 + \lambda(1-p_t)\big\}. \tag{4}$$

Equation (4) introduces *calibrated importance sampling* (CIS), our proposed correction for training–inference mismatch. The factor $1 - p_t$ in the cap is not a design choice: it is the factor by which Equation (2) maps displacements to ratios. The two forms in Equation (4) are identical, token by token; we implement the right-hand form as it only requires the difference of the two log-probabilities, which remains finite when a probability rounds to one in bf16, whereas $\operatorname{logit} p_t$ and $\operatorname{logit} q_t$ do not.

For comparison, a fixed ratio cap $k_t \le C$ corresponds through Equation (2) to

$$\exp(\varepsilon_t) \le 1 + \frac{C-1}{1-p_t}, \tag{5}$$

whose displacement threshold becomes increasingly permissive as confidence grows. CIS instead applies the same upper displacement threshold at every confidence level, and the confidence-dependent cap in Equation (4) follows automatically from the mapping back to ratio space.

Theorem 3.3 (Informal). *Assume $\|\ell'_t\|_2 \le b$ and that the contributions of different tokens are independent. Then for any finite $\lambda \ge 0$,*

$$\mathbb{E}\big\|\widehat{G}_{f_\lambda} - G_0\big\|_2^2 \le b^2\Big(\mathbb{E}\big[(1-p_t)(e^{\varepsilon_t}-1-\lambda)_+\big]\Big)^2 + \frac{b^2}{n}\,\mathbb{E}\Big[\big(p_t + (1-p_t)\min\{e^{\varepsilon_t}, 1+\lambda\}\big)^2\Big]. \tag{6}$$

Compared with Equation (3), the unbounded second moment $\mathbb{E}[e^{2\varepsilon_t}]$ is replaced by $\mathbb{E}\big[\min(e^{\varepsilon_t}, 1+\lambda)^2\big] \le (1+\lambda)^2$, at the price of a bias from the truncated excess; λ controls this trade-off, and $\lambda \to \infty$ recovers exact correction. Any cap with a finite maximum admits a similar bound, so the advantage of CIS over a fixed ratio cap lies not in this bound but in where the bias falls: CIS spreads the truncation bias more evenly across confidence levels and attains a lower overall bias (Section 4.3).

Algorithm. We treat λ as a hyperparameter; Section 4.4 shows that every positive threshold outperforms the uncorrected run. Applying Equation (4) in practice raises one issue: as $1-p_t$ becomes very small, the cap $1+\lambda(1-p_t)$ approaches one and becomes narrower than the storage resolution of the log-probabilities. The truncated tokens then have $|\log k_t|$ at the level of that resolution, so the truncation acts on rounding error rather than on the mismatch; without a remedy, the average accuracy drops from 34.78 to 29.02 (Section 4.3). We therefore floor $1-p_t$ at the storage resolution $\kappa = 5\times10^{-3}$, replacing it by $\varphi_t = \max(1-p_t, \kappa)$, which gives Algorithm 1. For tokens with $1-p_t \ge \kappa$, Algorithm 1 applies exactly the displacement threshold $\exp(\varepsilon_t) \le 1+\lambda$; below the floor, the threshold becomes $e^{\varepsilon_t} \le 1+\lambda\kappa/(1-p_t)$, which is looser. The correction adds one elementwise pass and no forward or backward computation. The inference-side log-probability must be the raw value recorded at the sampling step, and the weight must be detached.

Algorithm 1 Calibrated importance sampling (CIS)

Require: ℓ_m^{train}, ℓ_m^{infer}, ℓ'_m for the n tokens of the batch; threshold λ; floor κ

for $m = 1, \dots, n$ **do**
  $p_m \leftarrow \exp(\ell_m^{\text{train}})$,
  $k_m \leftarrow \exp(\ell_m^{\text{train}} - \ell_m^{\text{infer}})$
  $\varphi_m \leftarrow \max(1-p_m,\ \kappa)$
  $f_m \leftarrow \min\{k_m,\ 1+\lambda\varphi_m\}$
return $\widehat{G}_{\text{CIS}} = \frac{1}{n}\sum_{m=1}^{n} \text{stopgrad}(f_m)\,\ell'_m$

4 Experiments

We organize the experiments around three questions. **RQ1**: Does CIS improve held-out performance over existing training–inference mismatch corrections? **RQ2**: Why does CIS work, and which design choices matter? **RQ3**: How sensitive is CIS to its hyperparameters? Section 4.1 describes the shared setup. Implementation details, training dynamics, additional diagnostics, and out-of-domain evaluations are provided in Sections C and D.

4.1 Setup

We evaluate RLVR on three MoE models: Qwen1.5-MoE-A2.7B-Chat (Qwen Team, 2024), DeepSeek-V2-Lite (DeepSeek-AI, 2024), and Qwen3-30B-A3B (Qwen Team, 2025). All models are trained on GSM8K (Cobbe et al., 2021) and evaluated with greedy pass@1 on five mathematical reasoning benchmarks: GSM8K (Cobbe et al., 2021), MATH500 (Hendrycks et al., 2021; Lightman et al., 2024), SVAMP (Patel et al., 2021), Minerva-Math (Lewkowycz et al., 2022), and OlympiadBench (He et al., 2024). All methods share the same training recipe and code path and differ only in how they handle the training–inference mismatch. Unless stated otherwise, we report mean and standard deviation over three training seeds. We group the baselines by where they act: *token-level* (Exact Ratio (Yao et al., 2025; Liu et al., 2025), TIS (Yao et al., 2025), IcePop (Zhao et al., 2025; Ling Team & Inclusion AI, 2025), and KPop† (Guo et al., 2026; Ling Team, 2026)), *sequence-level* (Seq-TIS and Seq-MIS (Liu et al., 2025), and GSPO (Zheng et al., 2025b)), and *numerical* (FP16 (Qi et al., 2025)). The baseline "No Correction" provides the uncorrected reference, while CIS belongs to the token-level class. Full implementation details, hyperparameters, and evaluation protocols are given in Appendix C.

Table 1: Held-out accuracy (%) on five math benchmarks after RL on GSM8K. Mean ± standard deviation over three seeds; bold marks the best mean per column within each model.

Model	Method	GSM8K	MATH500	SVAMP	Minerva	Olympiad	Avg
Qwen1.5-MoE-A2.7B	Base model (no RL)	55.50	21.20	59.00	4.04	4.90	28.93
	No correction	$61.21_{\pm 1.17}$	$21.33_{\pm 1.40}$	$64.33_{\pm 1.73}$	$3.68_{\pm 1.27}$	$4.40_{\pm 0.45}$	$30.99_{\pm 0.22}$
	Exact ratio	$63.18_{\pm 2.97}$	$22.00_{\pm 1.22}$	$67.11_{\pm 3.75}$	$5.51_{\pm 0.64}$	$4.60_{\pm 0.15}$	$32.48_{\pm 1.21}$
	KPop†	$64.80_{\pm 1.29}$	$23.67_{\pm 1.29}$	$68.67_{\pm 3.71}$	$5.15_{\pm 1.47}$	$5.24_{\pm 0.67}$	$33.50_{\pm 1.04}$
	IcePop	$66.16_{\pm 0.96}$	$24.33_{\pm 1.03}$	$69.78_{\pm 1.02}$	$5.51_{\pm 0.64}$	$5.09_{\pm 1.11}$	$34.18_{\pm 0.55}$
	TIS	$61.03_{\pm 5.11}$	$20.73_{\pm 3.92}$	$66.78_{\pm 5.09}$	$4.54_{\pm 0.76}$	$3.91_{\pm 0.31}$	$31.40_{\pm 2.47}$
	Seq-TIS	$65.93_{\pm 0.19}$	$22.93_{\pm 0.61}$	$63.56_{\pm 1.39}$	$3.68_{\pm 1.60}$	$5.29_{\pm 0.89}$	$32.28_{\pm 0.86}$
	Seq-MIS	$60.98_{\pm 1.50}$	$22.07_{\pm 0.50}$	$63.56_{\pm 1.17}$	$4.41_{\pm 0.64}$	$4.75_{\pm 0.26}$	$31.15_{\pm 0.67}$
	GSPO	$60.10_{\pm 2.77}$	$17.47_{\pm 0.50}$	$58.56_{\pm 7.73}$	$5.02_{\pm 0.56}$	$3.36_{\pm 0.31}$	$28.90_{\pm 1.25}$
	FP16	$60.02_{\pm 1.16}$	$22.67_{\pm 0.46}$	$62.78_{\pm 1.54}$	$5.02_{\pm 0.21}$	$4.60_{\pm 0.45}$	$31.02_{\pm 0.39}$
	CIS (ours)	$\mathbf{67.88}_{\pm 1.64}$	$\mathbf{24.53}_{\pm 1.03}$	$\mathbf{70.22}_{\pm 1.35}$	$\mathbf{5.88}_{\pm 0.64}$	$\mathbf{5.39}_{\pm 0.62}$	$\mathbf{34.78}_{\pm 0.25}$
DeepSeek-V2-Lite	Base model (no RL)	70.66	25.40	67.67	6.25	6.08	35.21
	No correction	$71.06_{\pm 0.83}$	$23.47_{\pm 0.64}$	$70.89_{\pm 2.91}$	$7.23_{\pm 1.49}$	$6.48_{\pm 0.56}$	$35.83_{\pm 1.08}$
	Exact ratio	$71.17_{\pm 0.27}$	$24.67_{\pm 0.64}$	$70.22_{\pm 1.84}$	$6.37_{\pm 1.49}$	$4.95_{\pm 0.56}$	$35.47_{\pm 0.34}$
	KPop†	$71.14_{\pm 0.59}$	$26.33_{\pm 0.64}$	$70.89_{\pm 2.55}$	$6.62_{\pm 1.10}$	$6.13_{\pm 0.82}$	$36.22_{\pm 0.74}$
	IcePop	$72.48_{\pm 0.69}$	$25.60_{\pm 0.92}$	$71.44_{\pm 1.02}$	$7.72_{\pm 1.60}$	$6.43_{\pm 0.76}$	$36.73_{\pm 0.73}$
	TIS	$72.55_{\pm 0.84}$	$24.80_{\pm 1.00}$	$69.33_{\pm 1.76}$	$7.72_{\pm 0.97}$	$5.69_{\pm 1.12}$	$36.02_{\pm 0.51}$
	Seq-TIS	$71.87_{\pm 0.50}$	$24.53_{\pm 0.70}$	$70.89_{\pm 0.84}$	$6.74_{\pm 1.18}$	$5.74_{\pm 0.45}$	$35.95_{\pm 0.29}$
	Seq-MIS	$70.61_{\pm 0.46}$	$25.93_{\pm 1.01}$	$70.78_{\pm 2.59}$	$6.86_{\pm 0.56}$	$6.33_{\pm 0.45}$	$36.10_{\pm 0.71}$
	GSPO	$72.50_{\pm 0.86}$	$25.93_{\pm 0.70}$	$71.33_{\pm 1.00}$	$6.37_{\pm 1.66}$	$6.18_{\pm 0.52}$	$36.47_{\pm 0.43}$
	FP16	$71.80_{\pm 0.60}$	$24.47_{\pm 0.70}$	$69.89_{\pm 2.17}$	$7.97_{\pm 1.66}$	$6.23_{\pm 0.68}$	$36.07_{\pm 0.61}$
	CIS (ours)	$\mathbf{72.58}_{\pm 0.76}$	$\mathbf{26.47}_{\pm 0.99}$	$\mathbf{71.89}_{\pm 2.04}$	$\mathbf{8.21}_{\pm 1.66}$	$\mathbf{6.68}_{\pm 0.51}$	$\mathbf{37.16}_{\pm 0.73}$
Qwen3-30B-A3B	Base model (no RL)	95.22	74.00	93.33	31.25	45.40	67.84
	No correction	$94.79_{\pm 0.32}$	$72.40_{\pm 0.72}$	$94.56_{\pm 1.17}$	$33.33_{\pm 0.93}$	$46.24_{\pm 0.52}$	$68.26_{\pm 0.27}$
	Exact ratio	$95.35_{\pm 0.32}$	$74.40_{\pm 0.53}$	$93.22_{\pm 1.07}$	$33.33_{\pm 0.21}$	$47.28_{\pm 1.09}$	$68.72_{\pm 0.30}$
	KPop†	$95.40_{\pm 0.23}$	$74.80_{\pm 0.53}$	$94.56_{\pm 0.69}$	$33.21_{\pm 0.93}$	$47.33_{\pm 0.39}$	$69.06_{\pm 0.29}$
	IcePop	$95.43_{\pm 0.49}$	$74.93_{\pm 0.64}$	$94.89_{\pm 1.17}$	$32.11_{\pm 1.29}$	$47.58_{\pm 0.70}$	$68.99_{\pm 0.51}$
	TIS	$95.15_{\pm 0.20}$	$74.80_{\pm 0.72}$	$94.56_{\pm 0.84}$	$33.58_{\pm 1.18}$	$47.63_{\pm 0.45}$	$69.14_{\pm 0.30}$
	Seq-TIS	$95.12_{\pm 0.16}$	$74.67_{\pm 0.61}$	$93.78_{\pm 1.02}$	$33.33_{\pm 1.29}$	$48.07_{\pm 0.45}$	$68.99_{\pm 0.61}$
	Seq-MIS	$95.15_{\pm 0.35}$	$73.93_{\pm 0.42}$	$93.89_{\pm 1.02}$	$32.72_{\pm 1.10}$	$47.73_{\pm 0.45}$	$68.68_{\pm 0.15}$
	GSPO	$95.55_{\pm 0.24}$	$74.47_{\pm 0.50}$	$94.44_{\pm 1.17}$	$33.21_{\pm 0.56}$	$48.47_{\pm 0.73}$	$69.23_{\pm 0.16}$
	FP16	$94.79_{\pm 0.27}$	$74.13_{\pm 0.42}$	$93.67_{\pm 1.45}$	$33.21_{\pm 1.66}$	$46.83_{\pm 0.91}$	$68.53_{\pm 0.31}$
	CIS (ours)	$\mathbf{95.60}_{\pm 0.33}$	$\mathbf{75.00}_{\pm 0.92}$	$\mathbf{95.89}_{\pm 1.50}$	$\mathbf{34.31}_{\pm 0.56}$	$\mathbf{48.62}_{\pm 0.45}$	$\mathbf{69.88}_{\pm 0.55}$

4.2 RQ1: Does CIS improve held-out performance?

Table 1 compares all methods under the same training recipe. CIS achieves the highest five-benchmark average on all three MoE models. On Qwen1.5-MoE-A2.7B, the average improves from 30.99 without correction to 34.78 with CIS, compared with 32.48, 33.50, 34.18, and 31.40 for Exact Ratio, KPop†, IcePop, and TIS, respectively. CIS also achieves 67.88 on the training-domain GSM8K benchmark. Its margins over IcePop on several transfer benchmarks are smaller than the across-seed variation, however, so we do not interpret these individual columns as statistically separated.

The same trend holds across model families. CIS reaches an average of 37.16 on DeepSeek-V2-Lite and 69.88 on Qwen3-30B-A3B, the highest mean in both blocks. For Qwen3-30B-A3B, the base model already reaches 95.22% on GSM8K, leaving little headroom on the training task; the five-benchmark average therefore provides the more informative comparison.

Methods that act at other levels show different limitations. Sequence-level correction accumulates mismatch over the whole response, while changing numerical precision reduces but does not eliminate the residual engine discrepancy. We also evaluate the Qwen1.5-MoE checkpoints on three knowledge and two code benchmarks that the RL stage never uses (Table 13 in Section D.3); CIS has the highest pooled accuracy there, but its margin over IcePop is within the seed spread. Training dynamics and computational cost are reported in Sections D.1 and D.4.

4.3 RQ2: Why does CIS work, and which design choices matter?

Why truncate only the upper side? The variance is almost entirely carried by the upper tail: on the sampled tokens of the trained MoE at step 86, tokens with $k_t > 1$ account for 99.96% of $\mathrm{Var}(k_t)$, whereas those with $k_t < 1$ contribute only 0.04% (Figure 5a). The training ablation shows the same asymmetry: additionally lifting small displacements to $e^{\varepsilon_t} \geq 1/2$, which gives the two-sided weight $w_t = \max\{\min\{k_t, 1 + \lambda\varphi_t\}, (1 + p_t)/2\}$, reduces the average accuracy from 34.78 to 6.01 (see Table 2); CIS applies no lower threshold. CIS therefore truncates only the upper side and leaves the bounded lower side unchanged.

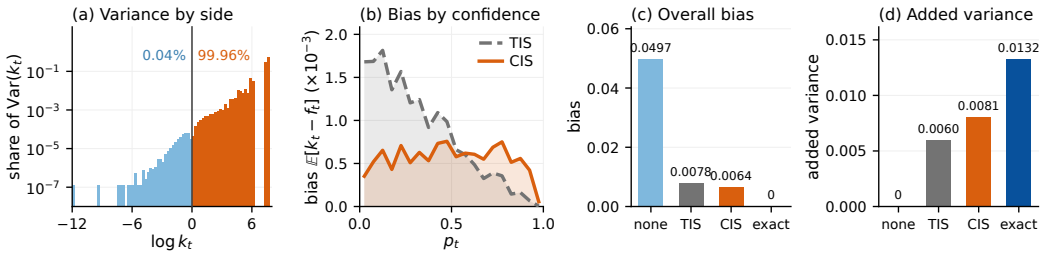


Figure 5: **(a)** Share of $\mathrm{Var}(k_t)$ on each side of $\log k_t = 0$. **(b)** Truncation bias across confidence bins. **(c)** Overall bias relative to exact correction. **(d)** Added variance relative to no correction.

Table 2: Ablations of CIS on Qwen1.5-MoE-A2.7B. Held-out accuracy (%) under the protocol of Table 1; mean ± standard deviation over three seeds.

Variant	GSM8K	MATH500	SVAMP	Minerva	Olympiad	Avg
CIS ($\lambda = 2.3$, $\kappa = 5 \times 10^{-3}$)	$\mathbf{67.88}_{\pm 1.64}$	$\mathbf{24.53}_{\pm 1.03}$	$\mathbf{70.22}_{\pm 1.35}$	$\mathbf{5.88}_{\pm 0.64}$	$\mathbf{5.39}_{\pm 0.62}$	$\mathbf{34.78}_{\pm 0.25}$
Fixed ratio cap (TIS, $C = 2$)	$61.03_{\pm 5.11}$	$20.73_{\pm 3.92}$	$66.78_{\pm 5.09}$	$4.54_{\pm 0.76}$	$3.91_{\pm 0.31}$	$31.40_{\pm 2.47}$
Fixed ratio cap (TIS, $C = 1.13$)	$14.99_{\pm 17.04}$	$5.73_{\pm 5.06}$	$20.33_{\pm 20.53}$	$1.47_{\pm 1.91}$	$1.88_{\pm 1.86}$	$8.88_{\pm 9.24}$
Two-sided band ($w_t \geq (1 + p_t)/2$)	$5.31_{\pm 0.33}$	$4.40_{\pm 0.40}$	$15.67_{\pm 1.86}$	$1.47_{\pm 0.64}$	$3.21_{\pm 0.45}$	$6.01_{\pm 0.50}$
Raised floor ($\kappa = 0.1$)	$64.42_{\pm 1.67}$	$22.60_{\pm 4.06}$	$64.44_{\pm 1.95}$	$4.90_{\pm 0.85}$	$4.50_{\pm 0.37}$	$32.17_{\pm 1.28}$
No floor ($\kappa = 0$)	$55.62_{\pm 2.69}$	$20.87_{\pm 1.10}$	$59.78_{\pm 2.83}$	$5.88_{\pm 0.37}$	$2.97_{\pm 0.15}$	$29.02_{\pm 1.43}$

Why truncate at the source of the mismatch? TIS caps the final importance ratio k_t, whereas CIS caps the displacement factor e^{ε_t}, through which the mismatch enters the ratio, at the common threshold $1 + \lambda$ of Equation (4) and then maps this bound back to ratio space. Because k_t mixes the displacement with token confidence, a fixed cap on k_t does not represent the same amount of mismatch across tokens. In contrast, truncating e^{ε_t} applies the same criterion to the mismatch itself, while the induced ratio bound automatically adapts to confidence.

This changes where truncation bias is spent. Figure 5b bins the sampled tokens of the base MoE by p_t and reports $\mathbb{E}[k_t - f_t]$ in each bin. TIS concentrates its bias on low-confidence tokens: below $p_t = 0.3$, the bias induced by CIS is less than half that of TIS, whereas CIS takes on more bias at high confidence. CIS truncates more high-confidence tokens by count, but each such truncation removes little weight, because the factor $1-p_t$ compresses the ratio on these tokens; more than half of the weight that CIS removes comes from tokens with $p_t < 0.9$. The two operators differ most in gradient bias at low confidence (Figure 5b): for $p_t < 0.565$, the cap of CIS is looser than $C = 2$, so CIS removes less weight there than TIS. Section D.5 shows the tokens that each operator acts on during training.

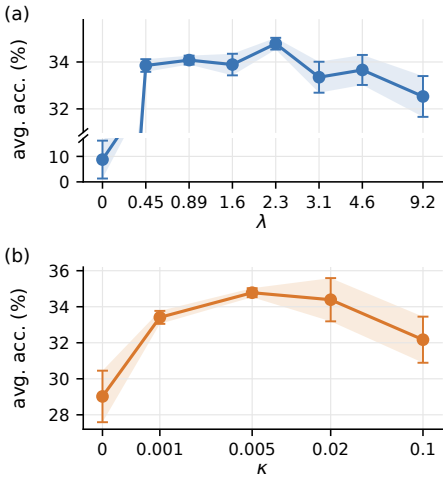


Figure 6: (a) Threshold λ and (b) floor κ on Qwen1.5-MoE-A2.7B.

Figure 5c,d correspond to the two terms of the mean-squared error in Theorem 3.3: the bias $\|\mathbb{E}[(f_t - k_t)\ell'_t]\|_2$ and the increase in the variance of $f_t\ell'_t$ over $f_t \equiv 1$. We compute both with the gradient of $\log p_t$ with respect to the logits in place of ℓ'_t, taking expectations over the next-token distributions at the 6,000 base-model positions of Section A.1; exact correction is unbiased for the conditional target G_0 and therefore has zero bias. Although CIS truncates more probability mass overall, its overall bias is lower (0.0064 versus 0.0078 for TIS), while its added variance remains well below that of Exact Ratio (0.0081 versus 0.0132). Thus, truncating at the mismatch source changes *where* the clipping occurs, not only how much is clipped, which yields a more favorable bias–variance trade-off.

Ablations. Table 2 supports these design choices. Replacing the confidence-dependent CIS boundary with the fixed TIS cap reduces the average accuracy from 34.78 to 31.40, while the two-sided band drops it to 6.01, below the base model (28.93). With $C = 1.13$, which truncates a similar share of

tokens as CIS (4.3% versus 3.8%, Table 10), the average falls to 8.88: the same amount of truncation harms training when a fixed cap places it on low-confidence tokens. TIS with $C = 1$, which is CIS with $\lambda = 0$, collapses as well (Figure 6a). The numerical floor is also necessary: removing it lowers the average accuracy to 29.02, and raising it to $\kappa = 0.1$ lowers it to 32.17. Together, the results support the three components of CIS: upper-side truncation, truncation at the mismatch source, and a small numerical floor.

4.4 RQ3: How sensitive is CIS to its hyperparameters?

Figure 6 varies one hyperparameter of CIS at a time on Qwen1.5-MoE-A2.7B, with the other fixed at its default; per-benchmark results are in Section D.2. For the threshold, $\lambda = 0$ truncates every ratio above one, which is TIS with $C = 1$, and the average accuracy collapses to 8.75. Every positive threshold stays above the uncorrected run (30.99), and the default $\lambda = 2.3$ gives the highest accuracy (34.78). The other positive thresholds lie between 32.53 and 34.08, so once $\lambda > 0$ the average changes by at most 2.3 points. For the floor, accuracy peaks near the storage resolution of the log-probabilities, where $\kappa = 5 \times 10^{-3}$ and $\kappa = 2 \times 10^{-2}$ reach 34.78 and 34.39. A smaller floor of 10^{-3} loses 1.4 points, and removing the floor lowers the average to 29.02, below the uncorrected run, which is consistent with the cap then truncating rounding error on high-confidence tokens. Raising the floor to 0.1 lowers the average to 32.17; the cap then stays constant on every token with $1 - p_t < 0.1$ and no longer tightens with confidence.

5 Related Work

RL frameworks separate rollout generation from gradient computation to improve efficiency (Sheng et al., 2025; Hu et al., 2025; Fu et al., 2025), but numerical differences between the engines can yield different token probabilities even with identical parameters (Yao et al., 2025). In mixture-of-experts models, small perturbations can change discrete expert selections and amplify the mismatch (Ma et al., 2025; Ling Team & Inclusion AI, 2025). FP16 reduces numerical error (Qi et al., 2025), routing replay aligns expert selections (Ma et al., 2025), and adaptive learning-rate decay stabilizes optimization (Zhang et al., 2026b). These interventions address different mechanisms: improved precision does not enforce routing agreement, while replay requires routing traces. RSPO instead addresses routing drift across policy updates (Zhang et al., 2026a), which differs from engine mismatch at identical parameters. CIS complements these approaches by controlling the gradient impact of residual mismatch without changing numerical precision or recording routes.

Importance sampling corrects distribution mismatch, while truncation trades bias for reduced variance (Ionides, 2008; Espeholt et al., 2018). For engine mismatch, TIS caps token ratios (Yao et al., 2025), and IcePop masks ratios outside a fixed interval (Zhao et al., 2025; Ling Team & Inclusion AI, 2025). These thresholds do not account for how token confidence affects the ratio, whereas KPop addresses this heterogeneity through binary-KL masking (Guo et al., 2026; Ling Team, 2026). Sequence-level alternatives truncate response weights (Liu et al., 2025), optimize sequence ratios (Zheng et al., 2025b), or reject responses that violate a trust-region criterion (Li et al., 2025). CISPO also clips weights to preserve token contributions, but uses fixed bounds on the current-to-old policy ratio (Chen et al., 2025). CIS instead derives its training–inference ratio cap from a common threshold on log-odds displacement. The resulting cap tightens with token confidence, and it keeps a nonzero weight for truncated tokens.

6 Conclusion

We study training–inference mismatch in RLVR and show that it enters the importance ratio as an additive displacement in log-odds, whose distribution varies far less with token confidence than the ratio and has a heavy upper tail on mixture-of-experts models. Based on this characterization, calibrated importance sampling (CIS) truncates the upper tail of the displacement at a single threshold, which yields a ratio cap that tightens as token confidence increases. We prove that CIS bounds the second moment that makes exact correction unstable, at the cost of a bias controlled by the truncated excess. On three mixture-of-experts models, CIS achieves the highest five-benchmark average among the evaluated baselines. Diagnostic analyses show that CIS places less truncation bias on low-confidence tokens than TIS, and that truncating the lower side reduces held-out accuracy.

REPRODUCIBILITY STATEMENT

To support reproducibility, we describe the models, training data, evaluation benchmarks, and baselines in Section 4.1, and summarize the correction in Algorithm 1. Section C reports the training recipe and hardware, the operator and hyperparameters of every method, the datasets and evaluation protocol, the prompts, the measurement and logging procedures, and the evaluation choices that were fixed before training. Anonymized source code is available at `https://anonymous.4open.science/r/CIS-BDD8`.

ETHICS STATEMENT

This work studies how to correct the training–inference mismatch in reinforcement learning for language models. All experiments use publicly available pretrained models and established benchmark datasets, and the study does not involve human subjects or private user data. CIS changes only how token-level importance weights are computed during training. Although CIS improves training stability and performance on mathematical reasoning benchmarks, these results should not be interpreted as guarantees of safety, fairness, or factual reliability. Downstream applications should therefore undergo safety evaluation suited to their intended use. Use and redistribution of the underlying models, datasets, and code should respect their applicable licenses and terms of use.

AI USE STATEMENT

In this work, we used generative AI tools only for language editing to improve the readability of the manuscript. All AI-assisted edits were reviewed by the authors to ensure that they preserve the original technical meaning. The authors take full responsibility for the final content of this work.

REFERENCES

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and Charles Sutton. Program synthesis with large language models, 2021. URL `https://arxiv.org/abs/2108.07732`.

Aili Chen, Aonian Li, Bangwei Gong, Binyang Jiang, Bo Fei, Bo Yang, Boji Shan, Changqing Yu, Chao Wang, Cheng Zhu, et al. Minimax-m1: Scaling test-time compute efficiently with lightning attention. *arXiv preprint arXiv:2506.13585*, 2025.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating large language models trained on code, 2021. URL `https://arxiv.org/abs/2107.03374`.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge, 2018. URL `https://arxiv.org/abs/1803.05457`.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems, 2021. URL `https://arxiv.org/abs/2110.14168`.

DeepSeek-AI. DeepSeek-V2: A strong, economical, and efficient mixture-of-experts language model. *CoRR*, abs/2405.04434, 2024. doi: 10.48550/arXiv.2405.04434. URL https://arxiv.org/abs/2405.04434.

DeepSeek-AI. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *CoRR*, abs/2501.12948, 2025. doi: 10.48550/arXiv.2501.12948. URL https://arxiv.org/abs/2501.12948.

Lasse Espeholt, Hubert Soyer, Rémi Munos, Karen Simonyan, Volodymyr Mnih, Tom Ward, Yotam Doron, Vlad Firoiu, Tim Harley, Iain Dunning, Shane Legg, and Koray Kavukcuoglu. IMPALA: Scalable distributed deep-RL with importance weighted actor-learner architectures. In *Proceedings of the 35th International Conference on Machine Learning*, 2018.

Wei Fu, Jiaxuan Gao, Xujie Shen, Chen Zhu, Zhiyu Mei, Chuyi He, Shusheng Xu, Guo Wei, Jun Mei, Jiashu Wang, Tongkai Yu, Binhang Yuan, and Yi Wu. AReaL: A large-scale asynchronous reinforcement learning system for language reasoning. *arXiv preprint arXiv:2505.24298*, 2025.

Jia Guo, Yan Sun, Zhenyu Huang, Zihao Wang, Zujie Wen, Zhiqiang Zhang, Jun Zhou, and Stanley Kok. KPop: Taming training–inference mismatch in reinforcement learning with adaptive masking regions. Technical blog, May 2026. URL https://ringtech.notion.site/kpop.

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

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

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.

Jian Hu, Xibin Wu, Wei Shen, Jason Klein Liu, Weixun Wang, Songlin Jiang, Haoran Wang, Hao Chen, Bin Chen, Wenkai Fang, Xianyu, Yu Cao, Haotian Xu, and Yiming Liu. OpenRLHF: A ray-based easy-to-use, scalable and high-performance RLHF framework. In Ivan Habernal, Peter Schulam, and Jörg Tiedemann (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 656–666, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-334-0. doi: 10.18653/v1/2025.emnlp-demos.48. URL https://aclanthology.org/2025.emnlp-demos.48/.

Edward L. Ionides. Truncated importance sampling. *Journal of Computational and Graphical Statistics*, 17(2):295–311, 2008.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. Solving quantitative reasoning problems with language models, 2022. URL https://arxiv.org/abs/2206.14858.

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

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations*, volume 2024, pp. 39578–39601, 2024.

Ling Team. Ling and ring 2.6 technical report: Efficient and instant agentic intelligence at trillion-parameter scale. *CoRR*, abs/2606.15079, 2026. doi: 10.48550/arXiv.2606.15079. URL https://arxiv.org/abs/2606.15079.

Ling Team and Inclusion AI. Every step evolves: Scaling reinforcement learning for trillion-scale thinking model. *CoRR*, abs/2510.18855, 2025. doi: 10.48550/arXiv.2510.18855. URL https://arxiv.org/abs/2510.18855.

Jiacai Liu, Yingru Li, Yuqian Fu, Jiawei Wang, Qian Liu, and Yu Shen. When speed kills stability: Demystifying RL collapse from the training-inference mismatch, September 2025. URL https://richardli.xyz/rl-collapse.

Wenhan Ma, Hailin Zhang, Liang Zhao, Yifan Song, Yudong Wang, Zhifang Sui, and Fuli Luo. Stabilizing MoE reinforcement learning by aligning training and inference routers. *arXiv preprint arXiv:2510.11370*, 2025. Introduces Rollout Routing Replay (R3).

Alberto Maria Metelli, Matteo Papini, Francesco Faccio, and Marcello Restelli. Policy optimization via importance sampling. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/6aed000af86a084f9cb0264161e29dd3-Paper.pdf.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii (eds.), *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2381–2391, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1260. URL https://aclanthology.org/D18-1260/.

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. Are NLP models really able to solve simple math word problems? In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou (eds.), *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 2080–2094, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.168. URL https://aclanthology.org/2021.naacl-main.168/.

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.

Qwen Team. Qwen1.5-MoE: Matching 7B model performance with 1/3 activated parameters. https://qwenlm.github.io/blog/qwen-moe/, March 2024.

Qwen Team. Qwen3 technical report. *CoRR*, abs/2505.09388, 2025. doi: 10.48550/arXiv.2505.09388. URL https://arxiv.org/abs/2505.09388.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Yu Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024. Introduces GRPO.

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. In *Proceedings of the Twentieth European Conference on Computer Systems*, pp. 1279–1297. ACM, 2025. URL https://dblp.org/rec/conf/eurosys/ShengZYWZZPL025.

Feng Yao, Liyuan Liu, Dinghuai Zhang, Chengyu Dong, Jingbo Shang, and Jianfeng Gao. Your efficient rl framework secretly brings you off-policy rl training, August 2025. URL https://fengyao.notion.site/off-policy-rl.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, juncai liu, LingJun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Ru 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, Yonghui Wu, and Mingxuan Wang. Dapo: An open-source llm reinforcement learning system at scale. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu,

P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 113222–113244. Curran Associates, Inc., 2025. doi: 10.52202/085713-3775. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/a4277440d50f1f15d2cb4c14f7e0c0d2-Paper-Conference.pdf.

Di Zhang, Xun Wu, Shaohan Huang, Lingjie Jiang, Yaru Hao, Li Dong, Zewen Chi, Zhifang Sui, and Furu Wei. Towards stable and effective reinforcement learning for mixture-of-experts. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 25446–25457, 2026a.

Yaxiang Zhang, Yingru Li, Jiacai Liu, Jiawei Xu, Ziniu Li, Qian Liu, and Haoyuan Li. Beyond precision: Training-inference mismatch is an optimization problem and simple lr scheduling fixes it. *arXiv preprint arXiv:2602.01826*, 2026b.

Xin Zhao, Yongkang Liu, Kuan Xu, Jia Guo, Zihao Wang, Yan Sun, Xinyu Kong, Qianggang Cao, Liang Jiang, Zujie Wen, Zhiqiang Zhang, and Jun Zhou. Small leak can sink a great ship—boost RL training on MoE with IcePop! Technical blog, September 2025. URL https://ringtech.notion.site/icepop.

Chujie Zheng, Kai Dang, Bowen Yu, Mingze Li, Huiqiang Jiang, Junrong Lin, Yuqiong Liu, Hao Lin, Chencan Wu, Feng Hu, An Yang, Jingren Zhou, and Junyang Lin. Stabilizing reinforcement learning with LLMs: Formulation and practices. *CoRR*, abs/2512.01374, 2025a. doi: 10.48550/arXiv.2512.01374. URL https://arxiv.org/abs/2512.01374.

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*, 2025b.

APPENDIX

A Empirical Anatomy of the Training–Inference Mismatch **14**

A.1 Measurement protocol 15

A.2 Dense versus MoE: a heavy tail in the per-logit perturbation 15

A.3 Routing disagreement and the MoE tail 16

A.4 Conditional routing control 17

A.5 The heavy tail propagates from δ to ε 17

A.6 The displacement is approximately invariant to token confidence 18

A.7 The second moment on the trained model 19

B Proofs and Technical Details for Section 3 **19**

B.1 Preliminaries and Key Lemmas 19

B.2 Proof of Theorem 3.2 22

B.3 Proof of Theorem 3.3 23

B.4 Numerical Floor in Algorithm 1 24

C Experimental Details **25**

C.1 Implementation and Hardware 25

C.2 Datasets and Evaluation Protocol 25

C.3 Prompts 26

C.4 Diagnostics 27

C.5 Preregistered Evaluation Contract 27

D Additional Results and Limitations **28**

D.1 Training Dynamics 28

D.2 Hyperparameter Sweeps 29

D.3 Out-of-Domain Evaluation 29

D.4 Computational Cost 30

D.5 Where Each Operator Acts 30

D.6 Limitations 32

A EMPIRICAL ANATOMY OF THE TRAINING–INFERENCE MISMATCH

Section 3.1 derives the ratio–displacement identity

$$k_t = p_t + (1 - p_t)e^{\varepsilon_t},$$

starting from the per-logit perturbation between the training and inference engines. This appendix provides the measurements behind the empirical claims used in that derivation and in Sections 3.2 and 3.3. We first describe how the per-logit perturbation is measured (Section A.1). We then compare dense and mixture-of-experts architectures (Section A.2), examine how routing disagreement is associated with the MoE tail (Sections A.3 and A.4), and verify that the heavy tail propagates from the per-logit perturbation δ_j to the log-odds displacement ε_t (Section A.5). Finally, we examine the

dependence of ε_t on token confidence (Section A.6) and show how the extreme upper tail dominates the second moment that enters the importance-sampling variance bound (Section A.7).

A.1 Measurement protocol

Models and engine configuration. We measure Qwen1.5-MoE-A2.7B-Chat and use Qwen1.5-14B-Chat as a dense control. All measurements in Sections A.2 to A.4 use the base checkpoints, with the training and inference engines loading exactly the same model parameters. The training engine is HuggingFace Transformers in bf16 with teacher forcing. The inference engine is vLLM 0.26 in bf16 with eager execution and prefix caching disabled.

Per-logit measurement. For each model, we randomly select 250 trajectories from the first shard of the static mismatch measurement (Section C.4) and sample 24 generation positions from each trajectory, giving 6,000 positions. At each position, the training engine records its top-256 pre-softmax logits together with the corresponding log-normalizer. On the inference side, we truncate the trajectory immediately before that position and generate one token with `logprobs=256`, which returns the next-token distribution at the same prefix. The trajectory and position identifiers align on all sampled positions. The two top-256 sets have mean intersection rates of 0.991 on the dense model and 0.949 on the MoE, yielding $1{,}521{,}672$ and $1{,}457{,}477$ paired logit entries, respectively. The complete measurement takes approximately eight minutes on four A100 GPUs.

Gauge convention. In Section 3.1, the training-engine logits are written as $\mathbf{z}$ and the inference-engine logits as $\mathbf{z}+\boldsymbol{\delta}$. Raw logits are defined only up to an additive constant at each position. Because the displacement depends only on logit contrasts, adding the same constant to every coordinate of $\boldsymbol{\delta}$ does not change ε_t. We therefore use the normalized log-probability gauge

$$\delta_j = \log \pi_{\text{infer}}(j \mid h_t) - \log \pi_{\text{train}}(j \mid h_t). \tag{7}$$

This differs from the raw logit perturbation only by a position-specific constant and is uniquely determined by the two observed distributions. Under this convention,

$$\varepsilon_t = \operatorname{logit} p_t - \operatorname{logit} q_t,$$

consistent with the convention used throughout the main text.

For the MoE model, the same measurement additionally records the top-4 expert identities selected by each engine at every layer, together with the routing margin between the fourth- and fifth-ranked experts. These records are used in Sections A.3 and A.4.

A.2 Dense versus MoE: a heavy tail in the per-logit perturbation

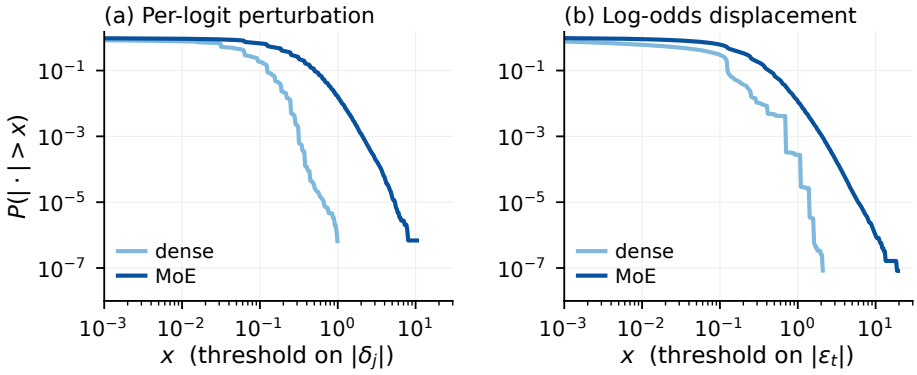


Figure 7: Complementary cumulative distributions of (a) $|\delta_j|$ and (b) $|\varepsilon_t|$ on the dense and MoE models.

Figure 7(a) compares the full tails. The two distributions are both concentrated near zero, but they separate sharply beyond the central region. On the dense control, only 0.003% of paired entries

Table 3: Per-logit perturbation δ_j on the base checkpoints.

	Dense	MoE
Paired entries	1,521,672	1,457,477
Median	0.0000	0.0000
Mean	-0.0013	-0.0036
$\lvert\text{mean}\rvert/\text{sd}$	1.6%	1.1%
$\text{sd}(\delta_j)$	0.079	0.336
$q_{99}(\lvert\delta_j\rvert)$	0.250	1.144
$q_{99.9}(\lvert\delta_j\rvert)$	0.312	2.062
$\Pr(\lvert\delta_j\rvert > 0.5)$	0.003%	9.87%
$\Pr(\lvert\delta_j\rvert > 1)$	0.0001%	1.52%
$\max_j \lvert\delta_j\rvert$	1.00	10.47
Range	$[-1.00, 0.91]$	$[-7.70, 10.47]$

satisfy $|\delta_j| > 0.5$, and essentially none exceed one. On the MoE, 9.87% exceed 0.5, 1.52% exceed one, and the largest perturbation reaches 10.47. The standard deviation is correspondingly 0.336, more than four times the dense value of 0.079.

The difference is especially visible in the extreme quantiles: $q_{99.9}(|\delta_j|)$ grows from 0.312 on the dense model to 2.062 on the MoE. Thus the distinction between the two architectures is not merely a uniform rescaling of a narrow numerical perturbation; the MoE develops a pronounced tail that is absent from the dense control.

A.3 Routing disagreement and the MoE tail

To examine the source of this additional tail, we use the expert identities recorded on the MoE. For each sampled position, let N_{flip} denote the number of layers, out of 24, for which the two engines select different top-4 expert sets. Routing disagreement is common: the mean is 3.69 disagreeing layers per position, and only 8.0% of positions select the same expert set at all 24 layers.

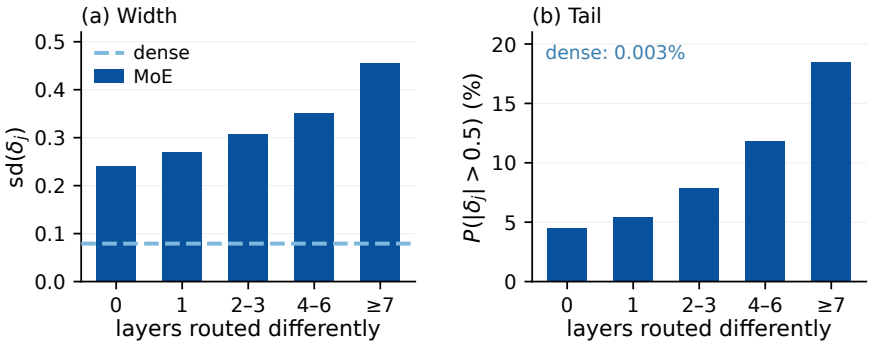


Figure 8: (a) Standard deviation of δ_j and (b) share of entries with $|\delta_j| > 0.5$ against the number of layers with routing disagreement; dashed: dense model.

Table 4: Per-logit perturbation against the number of layers N_{flip} with routing disagreement.

N_{flip}	0	1	2–3	4–6	≥ 7
Positions (%)	8.0	15.6	33.1	28.2	15.1
$\text{sd}(\delta_j)$	0.240	0.269	0.307	0.352	0.457
$\Pr(\lvert\delta_j\rvert > 0.5)$ (%)	4.51	5.42	7.85	11.86	18.48
$\Pr(\lvert\delta_j\rvert > 1)$ (%)	0.47	0.68	1.11	1.65	3.73

The two direct perturbation statistics show a clear monotonic trend. From positions with no routing disagreement to positions with disagreement in at least seven layers, $\text{sd}(\delta_j)$ increases from 0.240 to 0.457, a 90% increase, while $\Pr(|\delta_j| > 0.5)$ rises from 4.51% to 18.48%, a factor of 4.1.

The conclusion is unchanged under a stricter definition that counts only routing differences whose gate margin exceeds 10^{-3}. Under this "hard-flip" definition, the conditional standard deviations across the five bins are 0.284, 0.320, 0.405, 0.460, and 0.597, while $\Pr(|\delta_j| > 0.5)$ increases from 6.40% to 9.42%, 14.72%, 20.52%, and 30.78%. The association between routing disagreement and the width of the perturbation is therefore robust to the definition of a route change.

A.4 Conditional routing control

The previous analysis is observational: it stratifies positions by their realized routing disagreement rather than intervening on the routes. To separate the architecture-level difference from the additional effect associated with routing disagreement at the current position, we compare the dense control with MoE positions whose expert selections either agree at all 24 layers or disagree in at least one layer.

Table 5: Per-logit perturbation on the dense model and on MoE positions with and without routing disagreement.

	$\mathrm{sd}(\delta_j)$	$\Pr(\lvert\delta_j\rvert > 0.5)$	$\Pr(\lvert\delta_j\rvert > 1)$
Dense	0.079	0.003%	0.0001%
MoE, $N_{\text{flip}} = 0$	0.240	4.51%	0.47%
MoE, $N_{\text{flip}} > 0$	0.343	10.35%	1.62%

Routing disagreement at the current position therefore does not explain the entire architecture gap. Even at $N_{\text{flip}} = 0$, the MoE standard deviation is approximately three times that of the dense control. One possible source is routing disagreement earlier in the prefix, whose effect propagates through the hidden state to subsequent positions. The existing measurement cannot isolate this contribution: only 0.25% of tokens have prefixes for which the two engines agree on every recorded route.

Accordingly, we interpret the results of Figure 8 and Table 5 as evidence that *routing disagreement is a major amplifier of the tail, rather than the sole source of the mismatch*. A causal decomposition would require an intervention that explicitly replays identical routes across the two engines, which is outside the scope of the measurements reported here.

A.5 The heavy tail propagates from δ to ε

The identity in Section 3.1 maps the full perturbation vector $\boldsymbol{\delta}$ into the sampled token's log-odds displacement,

$$\varepsilon_t = \operatorname{logit} p_t - \operatorname{logit} q_t.$$

We next verify that the architectural difference observed directly in $\boldsymbol{\delta}$ remains visible after this mapping.

Unlike Section A.2, which uses the 6,000-position logit dump, this analysis uses the full static sampled-token measurement: approximately 12.4M dense tokens and 12.3M MoE tokens on the corresponding base checkpoints. We compute ε_t whenever neither p_t nor q_t rounds exactly to one.

Table 6: Log-odds displacement ε_t in the static measurement on the base checkpoints.

	Dense	MoE
Sampled tokens	12,389,115	12,293,115
Finite ε_t	11,934,872	12,226,092
Finite fraction	96.3%	99.5%
$\mathrm{sd}(\varepsilon_t)$	0.0997	0.2947
$q_{95}(\lvert\varepsilon_t\rvert)$	0.165	0.599
$q_{99}(\lvert\varepsilon_t\rvert)$	0.336	1.055
$q_{99.9}(\lvert\varepsilon_t\rvert)$	0.693	2.044
$\Pr(\lvert\varepsilon_t\rvert > 1)$	0.028%	1.13%
$\Pr(\lvert\varepsilon_t\rvert > 2)$	0	0.106%
$\max_t \lvert\varepsilon_t\rvert$	2.13	19.38

As shown in Figure 7(b), the dense displacement decays rapidly, whereas the MoE retains a long tail. The 99.9th percentile of $|\varepsilon_t|$ is 0.693 on the dense model and 2.044 on the MoE, and the maximum increases from 2.13 to 19.38. Thus the heavy tail identified at the level of individual logit perturbations survives the nonlinear mapping to the log-odds displacement that enters the importance ratio.

The dense CCDF contains several visible steps at high confidence. These arise from the discrete numerical structure of probabilities close to one and should not be interpreted as a continuous heavy tail. They do not affect the qualitative dense–MoE separation.

A.6 The displacement is approximately invariant to token confidence

The central design choice of CIS is to place a common upper threshold in the displacement coordinate rather than a common threshold on the importance ratio. This choice is motivated by the empirical observation that the scale of ε_t changes only mildly with token confidence, whereas the scale of the ratio deviation changes by orders of magnitude.

We examine the static base-MoE measurement and retain tokens with $1 - p_t \geq 10^{-7}$. Figure 9(a) plots conditional quantiles of ε_t against token uncertainty $1 - p_t$. Figure 9(b) compares the median absolute deviation of ε_t with that of $\log k_t$.

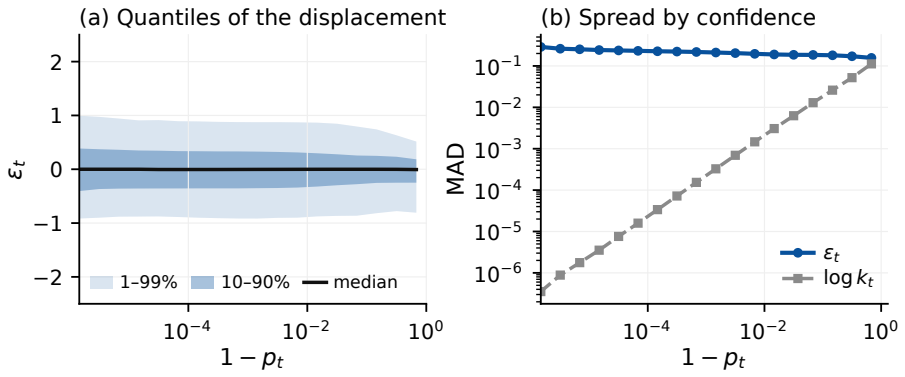


Figure 9: (a) Median, 10–90% and 1–99% intervals of ε_t and (b) median absolute deviation of ε_t and $\log k_t$ against $1 - p_t$ on the base MoE.

Table 7: Conditional statistics of ε_t and $\log k_t$ on the base MoE.

p_t range	Tokens	Med. ε	MAD(ε)	$q_{05}(\varepsilon)$	$q_{95}(\varepsilon)$	MAD($\log k$)
$p_t < 0.5$	2,824,021	−0.0059	0.156	−0.375	0.259	1.15×10^{-1}
0.5–0.9	2,337,554	−0.0011	0.175	−0.376	0.355	3.96×10^{-2}
0.9–0.99	1,951,864	−0.0005	0.186	−0.453	0.420	6.43×10^{-3}
0.99–0.999	1,549,915	−0.0019	0.203	−0.492	0.472	6.78×10^{-4}
$p_t > 0.999$	3,562,738	0.0000	0.237	−0.504	0.501	9.90×10^{-6}

Across 18 logarithmic bins spanning $1 - p_t \approx 1.5 \times 10^{-6}$ to 0.68, the conditional median of ε_t remains within 0.006 of zero. Its median absolute deviation ranges from 0.157 to 0.288, a factor of only 1.8. The displacement distribution is therefore not exactly confidence-invariant, and in particular becomes moderately wider toward the highest-confidence region, but the variation is small relative to the ratio coordinate.

Indeed, over the same bins, the median absolute deviation of $\log k_t$ changes by approximately 3.2×10^5. This contrast follows the mapping

$$k_t - 1 = (1 - p_t)(e^{\varepsilon_t} - 1) :$$

the same displacement is strongly compressed in ratio space when $p_t \to 1$. Consistently, on a random subsample of 300,000 tokens the Spearman correlation between ε_t and $\log(1 - p_t)$ is only -0.028.

Below the floor κ of Algorithm 1, ε_t is dominated by the storage resolution of the log-probabilities, so we repeat the comparison on the 7.6M tokens with $1 - p_t \geq \kappa$, which is the region where CIS acts. There, the median absolute deviation of ε_t varies by a factor of 1.3 across 18 logarithmic bins, whereas that of $\log k_t$ varies by a factor of 115, and the Spearman correlation between $|\varepsilon_t|$ and $\log(1 - p_t)$ is -0.12. The same contrast holds between the SGLang and FSDP engines used for training. On the tokens with zero staleness in the training logs of the exact-ratio, TIS, and IcePop runs, the median absolute deviation of ε_t varies by a factor of 1.8 to 2.7 across confidence bins above the floor, against 150 to 220 for $\log k_t$. These measurements support treating the displacement scale as approximately common across confidence levels, while making explicit that the approximation is not exact.

A.7 The second moment on the trained model

Figure 3 and Figure 5a use the MoE after RL training. We load the step-86 checkpoint of a CIS run on Qwen1.5-MoE-A2.7B into both engines of Section A.1 and score the 510,915 sampled tokens of 4,000 GSM8K rollouts, which are generated by the step-57 snapshot of the same run. The training engine scores them with teacher forcing and the inference engine by prefill, so the two engines score the same tokens with the same parameters. We compute ε_t whenever neither p_t nor q_t rounds to one, as in Section A.5. The moments of the displacement grow quickly with the order: $\mathbb{E}[e^{\varepsilon_t}] = 7.07$ and $\mathbb{E}[e^{2\varepsilon_t}] = 9.2 \times 10^6$, and the weighted moments $\mathbb{E}[((1 - p_t)e^{\varepsilon_t})^\alpha]$ that enter Theorem 3.2 are 0.16 at $\alpha = 1$ and 22.9 at $\alpha = 2$. The second moment is therefore determined by the upper tail of ε_t, which is the regime in which the truncated moment of CIS differs from exact correction.

B Proofs and Technical Details for Section 3

B.1 Preliminaries and Key Lemmas

In this subsection, we establish the minimal set of identities required for the proofs of Theorems 3.2 and 3.3. We adopt all notation and settings from Section 3. Throughout, $p_t, q_t \in (0, 1)$, $k_t := p_t/q_t$, and the n token contributions $(k_j, p_j, \ell'_j)_{j=1}^n$ are independent copies of (k_t, p_t, ℓ'_t). For a measurable correction rule $f : (0, \infty) \times (0, 1) \to [0, \infty)$, define

$$\widehat{G}_f := \frac{1}{n} \sum_{j=1}^{n} f(k_j, p_j)\, \ell'_j, \qquad G_0 := \mathbb{E}[k_t \ell'_t]. \tag{8}$$

For $\lambda \geq 0$, we write

$$f_\lambda(k, p) := \min\big\{k,\ 1 + \lambda(1 - p)\big\}. \tag{9}$$

Lemma B.1 (From logit perturbation to log-odds displacement)**.** *Fix the sampled token $y = y_t$. Let $p_t = \mathrm{softmax}(z)_y$ and $q_t = \mathrm{softmax}(z + \delta)_y$, and define*

$$w_j := \frac{e^{z_j}}{\sum_{j' \neq y} e^{z_{j'}}}, \qquad j \neq y, \tag{10}$$

and

$$\varepsilon_t := -\delta_y + \log \sum_{j \neq y} w_j e^{\delta_j}. \tag{11}$$

Then

$$k_t = \frac{p_t}{q_t} = p_t + (1 - p_t)e^{\varepsilon_t}, \qquad k_t - 1 = (1 - p_t)\big(e^{\varepsilon_t} - 1\big), \tag{12}$$

and

$$\varepsilon_t = \log \frac{p_t}{1 - p_t} - \log \frac{q_t}{1 - q_t}. \tag{13}$$

Proof. Substituting the two softmax probabilities directly into the ratio gives

$$\begin{aligned}
k_t &= \frac{e^{z_y}/\big(e^{z_y}+\sum_{j\neq y}e^{z_j}\big)}{e^{z_y+\delta_y}/\big(e^{z_y+\delta_y}+\sum_{j\neq y}e^{z_j+\delta_j}\big)} \\
&\overset{(a)}{=} \frac{e^{z_y}}{e^{z_y}+\sum_{j\neq y}e^{z_j}}\left(1+e^{-\delta_y}\frac{\sum_{j\neq y}e^{z_j+\delta_j}}{e^{z_y}}\right) \\
&\overset{(b)}{=} p_t + \frac{e^{-\delta_y}\sum_{j\neq y}e^{z_j+\delta_j}}{e^{z_y}+\sum_{j\neq y}e^{z_j}} \\
&\overset{(c)}{=} p_t + \frac{\sum_{j\neq y}e^{z_j}}{e^{z_y}+\sum_{j\neq y}e^{z_j}}e^{-\delta_y}\sum_{j\neq y}\frac{e^{z_j}}{\sum_{j'\neq y}e^{z_{j'}}}e^{\delta_j} \\
&\overset{(d)}{=} p_t + (1-p_t)e^{-\delta_y}\sum_{j\neq y}w_j e^{\delta_j} \\
&\overset{(e)}{=} p_t + (1-p_t)e^{\varepsilon_t},
\end{aligned}$$

where (a) expands the reciprocal of the inference probability, (b) uses the definition of p_t, (c) multiplies and divides by $\sum_{j\neq y}e^{z_j}$, (d) uses

$$1-p_t = \frac{\sum_{j\neq y}e^{z_j}}{e^{z_y}+\sum_{j\neq y}e^{z_j}}$$

together with (10), and (e) substitutes (11). Subtracting $1 = p_t + (1-p_t)$ immediately gives

$$\begin{aligned}
k_t - 1 &= p_t + (1-p_t)e^{\varepsilon_t} - p_t - (1-p_t) \\
&= (1-p_t)\big(e^{\varepsilon_t}-1\big).
\end{aligned}$$

To obtain the log-odds form, note that

$$\begin{aligned}
\frac{q_t}{1-q_t} &= \frac{e^{z_y+\delta_y}}{\sum_{j\neq y}e^{z_j+\delta_j}} \\
&\overset{(f)}{=} \frac{e^{z_y}}{\sum_{j\neq y}e^{z_j}}\frac{e^{\delta_y}}{\sum_{j\neq y}w_j e^{\delta_j}} \\
&\overset{(g)}{=} \frac{p_t}{1-p_t}\frac{e^{\delta_y}}{\sum_{j\neq y}w_j e^{\delta_j}},
\end{aligned}$$

where (f) factors $\sum_{j\neq y}e^{z_j+\delta_j}$ using (10), and (g) uses $p_t/(1-p_t) = e^{z_y}/\sum_{j\neq y}e^{z_j}$. Taking logarithms and rearranging yields

$$\begin{aligned}
\log\frac{p_t}{1-p_t} - \log\frac{q_t}{1-q_t} &= -\delta_y + \log\sum_{j\neq y}w_j e^{\delta_j} \\
&= \varepsilon_t,
\end{aligned}$$

which proves (13). □

Lemma B.2 (Displacement-coordinate truncation identities)**.** *Let $p \in (0,1)$ and suppose $k = p + (1-p)e^{\varepsilon}$. Then*

$$\varepsilon \le 0 \quad\Longrightarrow\quad p < k \le 1, \qquad \varepsilon > 0 \quad\Longrightarrow\quad k > 1. \tag{14}$$

Moreover, for every $\lambda \ge 0$,

$$f_\lambda(k,p) = p + (1-p)\min\big\{e^{\varepsilon},\, 1+\lambda\big\}, \tag{15}$$

and

$$k - f_\lambda(k,p) = (1-p)\big(e^{\varepsilon}-1-\lambda\big)_+. \tag{16}$$

Finally, for any fixed ratio cap $C \ge 1$,

$$k \le C \quad\Longleftrightarrow\quad e^{\varepsilon} \le 1 + \frac{C-1}{1-p}. \tag{17}$$

Proof. If $\varepsilon \le 0$, then $e^{\varepsilon} \le 1$, so

$$\begin{aligned} k &= p + (1-p)e^{\varepsilon} \\ &\le p + (1-p) = 1. \end{aligned}$$

At the same time, $k - p = (1-p)e^{\varepsilon} > 0$, and hence $p < k \le 1$. If $\varepsilon > 0$, then

$$k - 1 = (1-p)(e^{\varepsilon} - 1) > 0,$$

which gives $k > 1$.

For the clipped ratio,

$$\begin{aligned} f_\lambda(k,p) &\overset{(a)}{=} \min\Big\{p + (1-p)e^{\varepsilon},\ 1 + \lambda(1-p)\Big\} \\ &\overset{(b)}{=} \min\Big\{p + (1-p)e^{\varepsilon},\ p + (1-p)(1+\lambda)\Big\} \\ &\overset{(c)}{=} p + (1-p)\min\big\{e^{\varepsilon}, 1+\lambda\big\}, \end{aligned}$$

where (a) substitutes the ratio–displacement identity, (b) rewrites $1 + \lambda(1-p) = p + (1-p)(1+\lambda)$, and (c) uses that $x \mapsto p + (1-p)x$ is strictly increasing. Therefore

$$\begin{aligned} k - f_\lambda(k,p) &= (1-p)\left(e^{\varepsilon} - \min\{e^{\varepsilon}, 1+\lambda\}\right) \\ &= (1-p)(e^{\varepsilon} - 1 - \lambda)_+, \end{aligned}$$

using the pointwise identity $x - \min\{x, c\} = (x-c)_+$.

For a fixed cap $C \ge 1$,

$$\begin{aligned} k \le C &\iff p + (1-p)e^{\varepsilon} \le C \\ &\iff e^{\varepsilon} \le \frac{C-p}{1-p} \\ &\iff e^{\varepsilon} \le 1 + \frac{C-1}{1-p}, \end{aligned}$$

where the last equality follows from $C - p = (1-p) + (C-1)$. □

Lemma B.3 (Bias–variance decomposition)**.** *Let $f : (0,\infty) \times (0,1) \to [0,\infty)$ be measurable with $\mathbb{E}[f(k_t,p_t)^2 \|\ell'_t\|_2^2] < \infty$, and abbreviate $f_t := f(k_t,p_t)$. Then*

$$\mathbb{E}\big\|\widehat{G}_f - G_0\big\|_2^2 = \big\|\mathbb{E}\big[(f_t - k_t)\ell'_t\big]\big\|_2^2 + \frac{1}{n}\mathbb{E}\big\|f_t\ell'_t - \mathbb{E}[f_t\ell'_t]\big\|_2^2 \tag{18}$$

$$\le \big\|\mathbb{E}\big[(f_t - k_t)\ell'_t\big]\big\|_2^2 + \frac{1}{n}\mathbb{E}\big[f_t^2\|\ell'_t\|_2^2\big]. \tag{19}$$

Proof. Write $X_j := f(k_j,p_j)\ell'_j$ and $G_f := \mathbb{E}[X_j]$. Then

$$\begin{aligned} \mathbb{E}\big\|\widehat{G}_f - G_0\big\|_2^2 &\overset{(a)}{=} \mathbb{E}\big\|(\widehat{G}_f - G_f) + (G_f - G_0)\big\|_2^2 \\ &\overset{(b)}{=} \|G_f - G_0\|_2^2 + \mathbb{E}\big\|\widehat{G}_f - G_f\big\|_2^2 \\ &\overset{(c)}{=} \|G_f - G_0\|_2^2 + \frac{1}{n^2}\sum_{i,j=1}^{n}\mathbb{E}\big\langle X_i - G_f,\ X_j - G_f\big\rangle \\ &\overset{(d)}{=} \|G_f - G_0\|_2^2 + \frac{1}{n}\mathbb{E}\|X_1 - G_f\|_2^2 \\ &\overset{(e)}{=} \big\|\mathbb{E}[(f_t - k_t)\ell'_t]\big\|_2^2 + \frac{1}{n}\mathbb{E}\big\|f_t\ell'_t - \mathbb{E}[f_t\ell'_t]\big\|_2^2. \end{aligned}$$

Here (a) inserts and subtracts G_f. The cross term vanishes in (b) because $\mathbb{E}[\widehat{G}_f] = G_f$. Step (c) expands $\widehat{G}_f - G_f = n^{-1}\sum_j (X_j - G_f)$, while (d) uses independence: for $i \ne j$,

$$\mathbb{E}\langle X_i - G_f, X_j - G_f\rangle = 0,$$

and the n diagonal terms are identical. Finally,

$$G_f - G_0 = \mathbb{E}[f_t \ell'_t] - \mathbb{E}[k_t \ell'_t] = \mathbb{E}[(f_t - k_t)\ell'_t],$$

which gives (e) and proves (18).

For the variance term,

$$\begin{aligned}\frac{1}{n}\mathbb{E}\|X_1 - G_f\|_2^2 &= \frac{1}{n}\left(\mathbb{E}\|X_1\|_2^2 - \|G_f\|_2^2\right) \\ &\le \frac{1}{n}\mathbb{E}\|X_1\|_2^2 \\ &= \frac{1}{n}\mathbb{E}\big[f_t^2 \|\ell'_t\|_2^2\big],\end{aligned}$$

where the inequality simply drops the nonpositive term $-\|G_f\|_2^2/n$. This gives (19). □

Remark (correlated token contributions). Independence enters only when the off-diagonal covariance terms are removed. If the contributions are identically distributed but correlated, define

$$\bar{\rho}_f := \frac{\frac{1}{n(n-1)}\sum_{i\neq j}\mathbb{E}\langle X_i - G_f,\ X_j - G_f\rangle}{\mathbb{E}\|X_1 - G_f\|_2^2}. \tag{20}$$

If $\bar{\rho}_f \le \bar{\rho}$, then

$$\begin{aligned}\frac{1}{n^2}\sum_{i,j=1}^{n}\mathbb{E}\big\langle X_i - G_f,\ X_j - G_f\big\rangle &= \frac{1}{n^2}\left(n\mathbb{E}\|X_1 - G_f\|_2^2 + \sum_{i\neq j}\mathbb{E}\langle X_i - G_f, X_j - G_f\rangle\right) \\ &= \frac{1 + (n-1)\bar{\rho}_f}{n}\mathbb{E}\|X_1 - G_f\|_2^2 \\ &\le \frac{1}{n_{\text{eff}}}\mathbb{E}\|X_1 - G_f\|_2^2,\end{aligned}$$

where

$$n_{\text{eff}} := \frac{n}{1 + (n-1)\bar{\rho}}. \tag{21}$$

Hence the variance bounds below remain valid with n replaced by n_{eff}. Appendix D reports the corresponding empirical dependence among the token contributions.

B.2 Proof of Theorem 3.2

We first state the formal version of Theorem 3.2.

Theorem B.4 (Formal version of Theorem 3.2)**.** *Under the conventions of Section B.1, assume* $\|\ell'_t\|_2 \le b$*. Then*

$$\mathbb{E}\big\|\widehat{G} - G_0\big\|_2^2 \le \frac{b^2}{n}\mathbb{E}\Big[p_t^2 + 2p_t(1-p_t)e^{\varepsilon_t} + (1-p_t)^2 e^{2\varepsilon_t}\Big], \tag{22}$$

where $\widehat{G} := n^{-1}\sum_{j=1}^{n} k_j \ell'_j$.

Proof of Theorem 3.2. By Lemma B.1, the right-hand side of (22) equals $\frac{b^2}{n}\mathbb{E}[k_t^2]$. If $\mathbb{E}[k_t^2] = \infty$, there is nothing to prove. Assume therefore that $\mathbb{E}[k_t^2] < \infty$. Since $\|\ell'_t\|_2 \le b$,

$$\begin{aligned}\mathbb{E}\big[k_t^2\|\ell'_t\|_2^2\big] &\le b^2\mathbb{E}\big[k_t^2\big] \\ &\infty,\end{aligned}$$

so Lemma B.3 applies with $f_t = k_t$. Because exact correction has no truncation bias,

$$\begin{aligned}\mathbb{E}\big\|\widehat{G} - G_0\big\|_2^2 &= \frac{1}{n}\mathbb{E}\big\|k_t\ell'_t - \mathbb{E}[k_t\ell'_t]\big\|_2^2 \\ &\overset{(a)}{\le} \frac{1}{n}\mathbb{E}\big[k_t^2\|\ell'_t\|_2^2\big] \\ &\overset{(b)}{\le} \frac{b^2}{n}\mathbb{E}[k_t^2] \\ &\overset{(c)}{=} \frac{b^2}{n}\mathbb{E}\left[\big(p_t + (1-p_t)e^{\varepsilon_t}\big)^2\right] \\ &= \frac{b^2}{n}\mathbb{E}\left[p_t^2 + 2p_t(1-p_t)e^{\varepsilon_t} + (1-p_t)^2e^{2\varepsilon_t}\right].\end{aligned}$$

Here (a) drops the nonpositive centered-mean term in the variance, (b) uses $\|\ell'_t\|_2 \le b$, and (c) applies Lemma B.1. □

Remark (where the instability enters). The proof imposes no structural assumption on the law of ε_t beyond the moments that appear explicitly in the bound. Exact correction has zero truncation bias, but its variance depends on the untruncated second-moment term $\mathbb{E}[(1-p_t)^2e^{2\varepsilon_t}] \le \mathbb{E}[e^{2\varepsilon_t}]$.

Remark (correlated token contributions). Under correlated token contributions, the same argument holds with n replaced by n_{eff} from (21). The comparison below between the untruncated and truncated second moments is otherwise unchanged.

B.3 Proof of Theorem 3.3

We first state the formal version of Theorem 3.3. Throughout this subsection, let $m_t := \min\{e^{\varepsilon_t}, 1+\lambda\}$.

Theorem B.5 (Formal version of Theorem 3.3)**.** *Under the conventions of Section B.1, assume $\|\ell'_t\|_2 \le b$. Then for every finite $\lambda \ge 0$,*

$$\begin{aligned}\mathbb{E}\big\|\widehat{G}_{f_\lambda} - G_0\big\|_2^2 \le b^2\left(\mathbb{E}\big[(1-p_t)(e^{\varepsilon_t} - 1 - \lambda)_+\big]\right)^2 \\ + \frac{b^2}{n}\mathbb{E}\Big[\big(p_t + (1-p_t)\min\{e^{\varepsilon_t}, 1+\lambda\}\big)^2\Big].\end{aligned} \tag{23}$$

Proof of Theorem 3.3. Since

$$f_\lambda(k_t, p_t) \le 1 + \lambda(1-p_t) \le 1 + \lambda,$$

we have $\mathbb{E}[f_\lambda(k_t,p_t)^2\|\ell'_t\|_2^2] \le b^2(1+\lambda)^2 < \infty$, so Lemma B.3 applies:

$$\begin{aligned}\mathbb{E}\big\|\widehat{G}_{f_\lambda} - G_0\big\|_2^2 &\le \big\|\mathbb{E}[(f_\lambda(k_t,p_t) - k_t)\ell'_t]\big\|_2^2 \\ &\quad + \frac{1}{n}\mathbb{E}\big[f_\lambda(k_t,p_t)^2\|\ell'_t\|_2^2\big].\end{aligned}$$

For the bias term, $f_\lambda(k_t,p_t) \le k_t$, and hence

$$\begin{aligned}\big\|\mathbb{E}[(f_\lambda(k_t,p_t) - k_t)\ell'_t]\big\|_2 &\le \mathbb{E}\left[\big(k_t - f_\lambda(k_t,p_t)\big)\|\ell'_t\|_2\right] \\ &\le b\,\mathbb{E}\left[k_t - f_\lambda(k_t,p_t)\right] \\ &\overset{(a)}{=} b\,\mathbb{E}\left[(1-p_t)(e^{\varepsilon_t} - 1 - \lambda)_+\right],\end{aligned}$$

where (a) uses (16).

For the variance term, using (15),

$$\begin{aligned}\mathbb{E}\big[f_\lambda(k_t,p_t)^2\|\ell'_t\|_2^2\big] &\le b^2\mathbb{E}\big[f_\lambda(k_t,p_t)^2\big] \\ &\overset{(b)}{=} b^2\mathbb{E}\left[\big(p_t + (1-p_t)m_t\big)^2\right],\end{aligned}$$

where (b) substitutes the displacement-coordinate form of the clipped ratio.

Combining the two bounds gives (23). □

Remark (comparison with exact correction). The difference between Theorem B.4 and Theorem B.5 is the replacement of the untruncated displacement factor e^{ε_t} by its truncated counterpart m_t, together with the corresponding truncation bias. Pointwise,

$$\min\left(e^{\varepsilon_t}, 1+\lambda\right)^2 \le e^{2\varepsilon_t},$$
$$\min\left(e^{\varepsilon_t}, 1+\lambda\right)^2 \le (1+\lambda)^2,$$

and therefore

$$\mathbb{E}\big[\min(e^{\varepsilon_t}, 1+\lambda)^2\big] \le \min\left\{\mathbb{E}[e^{2\varepsilon_t}],\ (1+\lambda)^2\right\}. \tag{24}$$

Moreover, $p_t + (1-p_t)m_t \le \min\{k_t,\ 1+\lambda\}$, so the variance term of (23) satisfies $\mathbb{E}[(p_t + (1-p_t)m_t)^2] \le \min\{\mathbb{E}[k_t^2],\ (1+\lambda)^2\}$. Thus, regardless of how heavy the positive tail of ε_t becomes, the second-moment term entering the CIS variance bound is at most $(1+\lambda)^2$.

The corresponding bias remains controlled by the first moment of the truncated excess, since

$$\begin{aligned}(1-p_t)(e^{\varepsilon_t} - 1 - \lambda)_+ &\le (e^{\varepsilon_t} - 1 - \lambda)_+ \\ &\le e^{\varepsilon_t}\mathbf{1}\{e^{\varepsilon_t} > 1+\lambda\} \\ &\le e^{\varepsilon_t}.\end{aligned}$$

CIS therefore trades a finite upper-tail bias for a uniformly bounded second-moment contribution, which is the bias–variance trade-off described in Section 3.3.

Remark (correlated token contributions). The same result holds with n replaced by n_{eff} under the correlated-token setting above. The truncation term and the bound (24) are pointwise and therefore do not depend on the effective sample size.

B.4 Numerical Floor in Algorithm 1

The theoretical operator in (9) uses the uncertainty $1 - p_t$. Algorithm 1 replaces it by $\phi_\kappa(p) := \max\{1-p, \kappa\}$, where $0 < \kappa \le 1$, and applies

$$f_{\lambda,\kappa}(k,p) := \min\left\{k,\ 1 + \lambda\phi_\kappa(p)\right\}. \tag{25}$$

This modification prevents the confidence-dependent cap from becoming narrower than the numerical resolution of the stored log-probabilities.

Corollary B.6 (Effect of the numerical floor)**.** *For every $\lambda \ge 0$ and $\kappa \in (0,1]$,*

$$f_\lambda(k,p) \le f_{\lambda,\kappa}(k,p) \le k, \tag{26}$$

and consequently

$$0 \le k - f_{\lambda,\kappa}(k,p) \le k - f_\lambda(k,p). \tag{27}$$

Moreover,

$$f_{\lambda,\kappa}(k,p) \le 1 + \lambda. \tag{28}$$

Hence, if $\|\ell'_t\|_2 \le b$,

$$\mathbb{E}\big[f_{\lambda,\kappa}(k_t,p_t)^2\|\ell'_t\|_2^2\big] \le b^2(1+\lambda)^2. \tag{29}$$

Proof. By definition, $\phi_\kappa(p) = \max\{1-p,\kappa\} \ge 1-p$. Since $\lambda \ge 0$,

$$1 + \lambda\phi_\kappa(p) \ge 1 + \lambda(1-p).$$

The map $c \mapsto \min\{k, c\}$ is nondecreasing, so

$$\begin{aligned}f_{\lambda,\kappa}(k,p) &= \min\{k, 1+\lambda\phi_\kappa(p)\} \\ &\ge \min\{k, 1+\lambda(1-p)\} \\ &= f_\lambda(k,p).\end{aligned}$$

The definition of the minimum also gives $f_{\lambda,\kappa}(k,p) \le k$, which proves (26). Subtracting both sides from k yields (27).

Since both $1-p$ and κ lie in $(0,1]$, $\phi_\kappa(p) \le 1$. Therefore

$$\begin{aligned} f_{\lambda,\kappa}(k,p) &\le 1 + \lambda\phi_\kappa(p) \\ &\le 1 + \lambda, \end{aligned}$$

which gives (28). If $\|\ell'_t\|_2 \le b$, it follows immediately that

$$\begin{aligned} \mathbb{E}\big[f_{\lambda,\kappa}(k_t,p_t)^2\|\ell'_t\|_2^2\big] &\le b^2\mathbb{E}\big[f_{\lambda,\kappa}(k_t,p_t)^2\big] \\ &\le b^2(1+\lambda)^2, \end{aligned}$$

proving (29). □

Remark. The floor raises the cap only at the extreme high-confidence end, so $f_{\lambda,\kappa}$ removes no more of the exact importance weight than the idealized operator f_λ. At the same time, (28) keeps the applied weight uniformly bounded by $1+\lambda$. The numerical safeguard in Algorithm 1 therefore preserves the central variance-control property of CIS while preventing the theoretical band from becoming narrower than the storage resolution.

C Experimental Details

C.1 Implementation and Hardware

Recipe. Every arm uses the same recipe: learning rate 3×10^{-6}, weight decay 0.01, PPO clip range $(0.2, 0.28)$, no KL penalty, four samples per prompt at temperature one, at most 1,024 generated tokens, at most two versions of rollout staleness, three epochs over the 7,473 GSM8K training problems (87 optimizer steps), and one node with eight A100-80GB GPUs per arm, four serving SGLang and four training with FSDP and flash attention.

Operators. Training uses AReaL v2.0.0 with its decoupled loss, in which the rollout-side and training-side log-probabilities are both available per token. All operators are implemented in the same token-level branch that AReaL uses for TIS and IcePop, so that every arm differs only in the weight on the token loss; CIS multiplies the loss by the weight $\min\{k_t, 1+\lambda\varphi_t\}$ of Algorithm 1, with $\varphi_t = \max(1-p_t, \kappa)$ and p_t taken from the training-side log-probability, and the weight carries no gradient. The two-sided band of Table 2 additionally lifts the displacement from below to $e^{\varepsilon_t} \ge 1/2$, which maps through Equation (2) to the weight $\max\{\min\{k_t, 1+\lambda\varphi_t\},\ (1+p_t)/2\}$; CIS applies no lower threshold. Table 8 lists the operator and the hyperparameters of every arm. Beyond the operators, the changes to the framework are longer worker-readiness timeouts, a numerically corrected binary-KL metric for KPop†, product weights over a response for Seq-TIS and Seq-MIS, static loss scaling for the fp16 arm, and a token-level dump hook. The training engine must use flash attention; with packed sequences the scaled-dot-product fallback attends across sequence boundaries and corrupts the training-side log-probabilities, which we detected as a median $\log k_t$ far from zero on every method. Weights are sent from the trainer to the rollout engine by collective communication after every optimizer step. On Qwen1.5-MoE each arm takes about 1.0 to 1.4 hours for the 87 steps, and Appendix D.4 breaks this time down by phase. DeepSeek-V2-Lite (DeepSeek-AI, 2024) and Qwen3-30B-A3B (Qwen Team, 2025) use the same recipe with vLLM rollouts, weights reloaded from disk after every step, and optimizer states in bf16; the thinking mode of Qwen3-30B-A3B is disabled in both training and evaluation.

C.2 Datasets and Evaluation Protocol

Training data. Every RL run trains on the 7,473 problems of the GSM8K training split (Cobbe et al., 2021). The prompt is the problem followed by an instruction to put the final answer in `\boxed{}` (Appendix C.3), and the reward is one when math-verify judges the final answer equal to the reference answer and zero otherwise. The framework evaluates the GSM8K test split once per epoch for logging; these numbers are not used to select a checkpoint, and every number that we report comes from the final checkpoint.

Measurement data. The static measurement of the mismatch uses 2,500 problems drawn at random from DAPO-Math-17k (Yu et al., 2025). This set is not used for training.

Table 8: Operators and hyperparameters of the arms in Table 1. $K = \prod_t k_t$ is the product over a response and $\varphi_t = \max(1 - p_t, \kappa)$.

Method	Weight on the token loss	Hyperparameters
No correction	1 (decoupled loss disabled)	—
Exact ratio	k_t	clamp to $[10^{-6}, 10^{6}]$
TIS	$\min(k_t, C)$	$C = 2$
IcePop	$k_t \, \mathbf{1}\{k_t \in [\alpha, \beta]\}$	$[\alpha, \beta] = [0.5, 5]$
KPop[†]	$k_t \, \mathbf{1}\{\max(\mathrm{KL_B}(p\|q), \mathrm{KL_B}(q\|p)) \le \phi\}$, float64	$\phi = 2$
Seq-TIS	$\min(K, C)$ for every token of the response	$C = 2$
Seq-MIS	$K \, \mathbf{1}\{K \in [\alpha, \beta]\}$ for every token of the response	$[\alpha, \beta] = [0.5, 2]$
GSPO	sequence-level surrogate, k_t uncorrected	clip $(3 \times 10^{-4}, 4 \times 10^{-4})$
FP16	1, rollout and training in fp16	loss scale 1024
CIS (ours)	$\min\{k_t, 1 + \lambda\varphi_t\}$	$\lambda = 2.3$, $\kappa = 5 \times 10^{-3}$

Protocols. Table 9 lists the evaluation benchmarks. Tables 1 and 2 use the chat template of each model with the instruction to put the final answer in `\boxed{}`, greedy decoding with vLLM, at most 1,536 new tokens, and the math-verify judge. The out-of-domain evaluation of Table 13 uses the same template and greedy decoding; a multiple-choice item is scored by the option letter that the response states, with at most 512 new tokens, and a code item is scored by running the completion against the benchmark tests with a limit of 15 seconds, with at most 768 new tokens. Changing the tensor-parallel degree of the evaluation engine changes a few greedy decisions (two HumanEval and six MBPP items on one checkpoint), so every checkpoint of a table is evaluated with the same degree.

Table 9: Evaluation benchmarks.

Benchmark	Domain	Split	Scoring	Items
GSM8K (Cobbe et al., 2021)	math	test	math-verify	1,319
MATH-500 (Hendrycks et al., 2021; Lightman et al., 2024)	math	test subset	math-verify	500
SVAMP (Patel et al., 2021)	math	test	math-verify	300
Minerva-Math (Lewkowycz et al., 2022)	math	test	math-verify	272
OlympiadBench (He et al., 2024)	math	English, text-only, open-ended	math-verify	674
MMLU-STEM (Hendrycks et al., 2020)	knowledge	test, 19 STEM subjects	option letter	3,153
ARC-Challenge (Clark et al., 2018)	knowledge	test	option letter	1,172
OpenBookQA (Mihaylov et al., 2018)	knowledge	test	option letter	500
HumanEval (Chen et al., 2021)	code	test	unit tests	164
MBPP (Austin et al., 2021)	code	test	unit tests	500

C.3 Prompts

Every prompt below is the user message, formatted with the chat template of the model and no additional system message; the thinking mode of Qwen3-30B-A3B is disabled. Placeholders in braces are filled with the fields of each item.

Prompt: RL training on GSM8K

`{question}`
Please put your final answer within `\boxed{}`.

Prompt: evaluation on the five math benchmarks (Table 1)

`{question}`
Please reason step by step, and put your final answer within `\boxed{}`.

Prompt: out-of-domain multiple choice (MMLU-STEM, ARC-Challenge, OpenBookQA)

{question}

A. {option A}
B. {option B}
C. {option C}
D. {option D}

Answer with the letter of the correct option. Put the single letter within \boxed{}.

Prompt: out-of-domain code (HumanEval)

Complete the following Python function.

```python
{function signature and docstring}
```

Return the complete function in a single ```python code block. Do not include example usage or tests.

Prompt: out-of-domain code (MBPP)

{problem description}

Your code must pass these tests:

```python
{three assert statements}
```

Return the complete function in a single ```python code block. Do not include example usage or tests.

Prompt: measurement of the mismatch (DAPO-Math-17k)

{problem}

C.4 Diagnostics

Static measurement. Prompts are 2,500 problems drawn at random from DAPO-Math-17k. Generation uses vLLM 0.26 in bf16 with eager execution and without prefix caching; each prompt receives eight samples at temperature one and top-p one, at most 1,024 tokens per response, and the log-probability of every sampled token is taken from the sampling step itself rather than from a later prefill pass. Rescoring uses HuggingFace Transformers in bf16 with teacher forcing on the exact sampled sequence and the same parameter snapshot; the token at position t is scored from the logits at position $t-1$ on both sides. For the MoE model we record the top-4 expert identities of every layer on both sides and the gate margin between the fourth and fifth expert; the routing read from forward hooks agrees with the identities returned by the fused kernel on 100% of the (token, layer) pairs whose margin is positive. Two controls fix the measurement floor: rescoring a sequence twice on the training engine returns exactly zero difference, and the empirical mean of k_t is 1.0000 on both architectures, which verifies the per-step calibration identity $\mathbb{E}[k_t \mid h_t] = 1$. Two further controls accompany the main measurement: rescoring in fp32, and rescoring by a prefill pass on the inference engine, which reproduces the decode-time bulk. The displacement is computed as $\varepsilon_t = \operatorname{logit} p_t - \operatorname{logit} q_t$ from the two log-probabilities.

Training-time logging. The training runs dump, every ten optimizer calls, the rollout-side log-probability, the training-side log-probability at the start of the update, and the parameter version of the rollout for a sample of tokens. Version staleness is the difference between the current version and the rollout version.

C.5 Preregistered Evaluation Contract

Held-out accuracy at the final checkpoint is the arbiter of every comparison; training reward is reported but is not used to choose between methods. The following choices were fixed before the corresponding runs. The floor is $\kappa = 5 \times 10^{-3}$, set from the storage resolution of the log-probabilities

before any training run. The slope grid is $\lambda \in \{1.6, 2.3, 3.1\}$. All arms use the recipe of Section 4.1 without per-arm tuning. Every ablation changes one element of the operator relative to the deployed configuration. The main comparison and the ablations of Table 2 use three seeds.

D Additional Results and Limitations

D.1 Training Dynamics

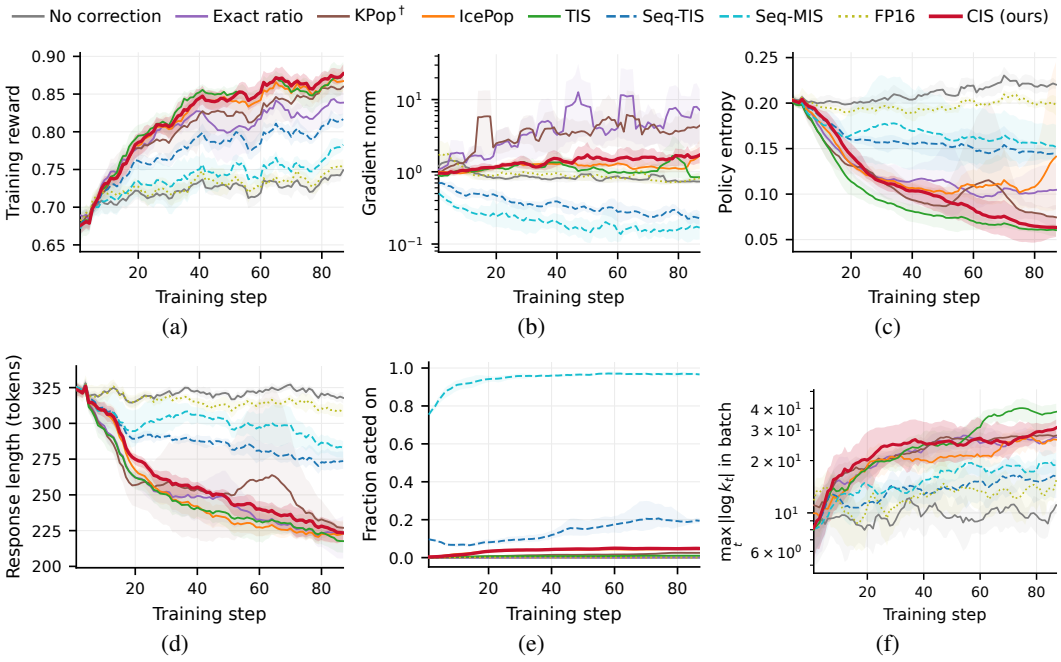


Figure 10: Training dynamics on Qwen1.5-MoE-A2.7B (mean over seeds, band spanning the seeds, five-step smoothing). (a) Reward. (b) Gradient norm. (c) Entropy. (d) Response length. (e) Fraction of tokens acted on, counting whole responses for Seq-TIS and Seq-MIS. (f) Largest $|\log k_t|$ in the batch.

Table 10: Training statistics on Qwen1.5-MoE-A2.7B, mean $\pm$ standard deviation over three seeds: third-epoch mean (median for the gradient norm and $\max_t |\log k_t|$), and peak gradient norm and fraction acted on over the whole run.

Method	Reward	Grad. norm	Peak grad.	Entropy	Length	Acted on (%)	$\max_t \lvert\log k_t\rvert$
No correction	$0.732_{\pm 0.004}$	$0.75_{\pm 0.02}$	$2.5_{\pm 0.4}$	$0.220_{\pm 0.005}$	$322_{\pm 2}$	$0.00_{\pm 0.00}$	$9.5_{\pm 0.5}$
Exact ratio	$0.829_{\pm 0.012}$	$3.58_{\pm 0.75}$	$69.5_{\pm 41.4}$	$0.102_{\pm 0.013}$	$227_{\pm 4}$	$0.06_{\pm 0.03}$	$25.9_{\pm 3.4}$
KPop†	$0.847_{\pm 0.016}$	$3.09_{\pm 0.76}$	$29.2_{\pm 23.5}$	$0.093_{\pm 0.033}$	$245_{\pm 29}$	$1.33_{\pm 0.05}$	$27.3_{\pm 1.7}$
IcePop	$0.859_{\pm 0.005}$	$1.12_{\pm 0.04}$	$2.8_{\pm 1.5}$	$0.113_{\pm 0.035}$	$224_{\pm 7}$	$3.64_{\pm 0.40}$	$24.0_{\pm 2.3}$
TIS ($C = 2$)	$0.861_{\pm 0.014}$	$0.97_{\pm 0.11}$	$3.6_{\pm 1.6}$	$0.064_{\pm 0.004}$	$226_{\pm 9}$	$0.76_{\pm 0.07}$	$37.7_{\pm 4.4}$
TIS ($C = 1.13$)	$0.780_{\pm 0.065}$	$0.91_{\pm 0.08}$	$8.5_{\pm 11.7}$	$0.200_{\pm 0.142}$	$224_{\pm 23}$	$4.32_{\pm 0.64}$	$29.7_{\pm 4.4}$
Seq-TIS	$0.803_{\pm 0.004}$	$0.25_{\pm 0.02}$	$0.8_{\pm 0.0}$	$0.147_{\pm 0.016}$	$276_{\pm 7}$	$13.54_{\pm 2.83}$	$15.6_{\pm 2.9}$
Seq-MIS	$0.759_{\pm 0.002}$	$0.15_{\pm 0.01}$	$0.6_{\pm 0.2}$	$0.158_{\pm 0.025}$	$292_{\pm 9}$	$94.56_{\pm 0.50}$	$18.2_{\pm 1.7}$
FP16	$0.740_{\pm 0.002}$	$0.74_{\pm 0.01}$	$3.2_{\pm 1.3}$	$0.202_{\pm 0.002}$	$313_{\pm 3}$	$0.00_{\pm 0.00}$	$12.7_{\pm 2.1}$
CIS (ours)	$\mathbf{0.867}_{\pm 0.009}$	$1.47_{\pm 0.31}$	$3.0_{\pm 1.1}$	$0.071_{\pm 0.017}$	$232_{\pm 8}$	$3.81_{\pm 0.42}$	$26.9_{\pm 5.1}$

Figure 10 and Table 10 show how the arms of Table 1 train on Qwen1.5-MoE. The token-level operators that bound the ratio (TIS, IcePop, and CIS) reach a training reward between 0.859 and 0.867 with a peak gradient norm below four, whereas the exact ratio and KPop† reach a lower reward with peak gradient norms of 70 and 29. The uncorrected run and the fp16 run stay near a reward of 0.74, and their entropy and response length stay close to their initial values. Seq-MIS removes 95% of the responses, and its gradient norm is the smallest of all arms because most of the batch carries no gradient. The largest $|\log k_t|$ in the batch stays near 10 for the uncorrected and fp16 runs and grows

to between 16 and 38 for the other arms as the policy sharpens. The fraction of tokens that CIS acts on rises during the first third of training and then stays between 4% and 5% (panel e).

D.2 Hyperparameter Sweeps

Tables 11 and 12 list the per-benchmark accuracy behind Figure 6. Each sweep varies one hyperparameter of CIS on Qwen1.5-MoE-A2.7B with the other fixed at its default, and uses the recipe and evaluation protocol of Table 1; the default rows are the CIS entries of Table 1. Setting $\lambda = 0$ truncates every ratio above one, which is TIS with $C = 1$, and the average accuracy collapses to 8.75 with a large spread across seeds. Every positive threshold stays above the uncorrected run (30.99), and the default $\lambda = 2.3$ gives the highest average (34.78), while the other positive thresholds lie between 32.53 and 34.08.

Table 11: Accuracy (%) of CIS on Qwen1.5-MoE-A2.7B for different thresholds λ with $\kappa = 5 \times 10^{-3}$. Mean $\pm$ standard deviation over three seeds.

λ	GSM8K	MATH500	SVAMP	Minerva	Olympiad	Avg
0	$14.71_{\pm 13.47}$	$6.80_{\pm 5.67}$	$19.33_{\pm 17.17}$	$0.98_{\pm 1.39}$	$1.93_{\pm 1.12}$	$8.75_{\pm 7.44}$
0.45	$64.82_{\pm 1.88}$	$23.33_{\pm 2.44}$	$70.33_{\pm 2.08}$	$7.11_{\pm 0.21}$	$3.66_{\pm 0.17}$	$33.85_{\pm 0.27}$
0.89	$66.59_{\pm 1.95}$	$22.87_{\pm 1.75}$	$70.22_{\pm 1.90}$	$6.25_{\pm 1.33}$	$4.45_{\pm 0.83}$	$34.08_{\pm 0.19}$
1.6	$65.68_{\pm 0.46}$	$23.33_{\pm 1.63}$	$68.78_{\pm 2.17}$	$6.74_{\pm 0.56}$	$4.90_{\pm 0.59}$	$33.89_{\pm 0.46}$
2.3 (default)	$67.88_{\pm 1.64}$	$24.53_{\pm 1.03}$	$70.22_{\pm 1.35}$	$5.88_{\pm 0.64}$	$5.39_{\pm 0.62}$	$34.78_{\pm 0.25}$
3.1	$65.07_{\pm 0.70}$	$23.67_{\pm 0.92}$	$67.89_{\pm 2.50}$	$6.13_{\pm 1.49}$	$4.01_{\pm 0.90}$	$33.35_{\pm 0.66}$
4.6	$64.54_{\pm 1.19}$	$24.20_{\pm 0.53}$	$69.22_{\pm 1.26}$	$6.86_{\pm 0.21}$	$3.46_{\pm 0.09}$	$33.66_{\pm 0.64}$
9.2	$62.27_{\pm 1.67}$	$23.40_{\pm 0.60}$	$66.89_{\pm 1.84}$	$6.74_{\pm 0.21}$	$3.36_{\pm 0.09}$	$32.53_{\pm 0.87}$

For the floor, the accuracy peaks near the storage resolution of the log-probabilities, where $\kappa = 5 \times 10^{-3}$ and $\kappa = 2 \times 10^{-2}$ reach 34.78 and 34.39; a smaller floor of 10^{-3} loses 1.4 points, and removing the floor or raising it to 0.1 lowers the average to 29.02 and 32.17. The gains of the default floor are largest on GSM8K and OlympiadBench.

Table 12: Accuracy (%) of CIS on Qwen1.5-MoE-A2.7B for different floors κ with $\lambda = 2.3$. Mean $\pm$ standard deviation over three seeds.

κ	GSM8K	MATH500	SVAMP	Minerva	Olympiad	Avg
0	$55.62_{\pm 2.69}$	$20.87_{\pm 1.10}$	$59.78_{\pm 2.83}$	$5.88_{\pm 0.37}$	$2.97_{\pm 0.15}$	$29.02_{\pm 1.43}$
10^{-3}	$64.01_{\pm 0.65}$	$24.00_{\pm 0.20}$	$68.67_{\pm 0.67}$	$6.86_{\pm 0.21}$	$3.51_{\pm 0.09}$	$33.41_{\pm 0.36}$
5×10^{-3} (default)	$67.88_{\pm 1.64}$	$24.53_{\pm 1.03}$	$70.22_{\pm 1.35}$	$5.88_{\pm 0.64}$	$5.39_{\pm 0.62}$	$34.78_{\pm 0.25}$
2×10^{-2}	$65.93_{\pm 2.31}$	$24.73_{\pm 0.95}$	$70.56_{\pm 2.46}$	$7.11_{\pm 0.21}$	$3.61_{\pm 0.09}$	$34.39_{\pm 1.20}$
0.1	$64.42_{\pm 1.67}$	$22.60_{\pm 4.06}$	$64.44_{\pm 1.95}$	$4.90_{\pm 0.85}$	$4.50_{\pm 0.37}$	$32.17_{\pm 1.28}$

D.3 Out-of-Domain Evaluation

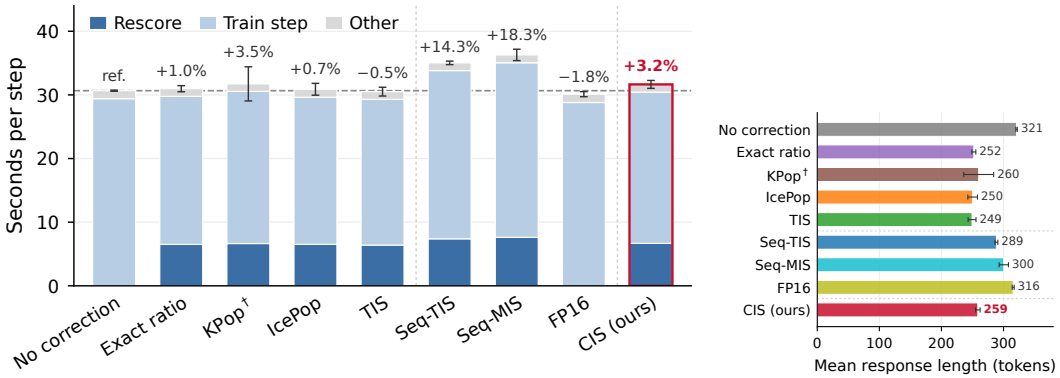
Table 13: Out-of-domain accuracy (%) on Qwen1.5-MoE-A2.7B after RL on GSM8K. Mean $\pm$ standard deviation over three seeds; Science, Code, and All are item-weighted over the knowledge, code, and all benchmarks.

Method	MMLU	ARC-C	OBQA	HumanEval	MBPP	Science	Code	All
Base model (no RL)	47.95	72.61	66.60	44.51	33.40	55.88	36.14	53.49
No correction	$49.96_{\pm 0.36}$	$72.90_{\pm 0.65}$	$69.67_{\pm 0.42}$	$48.37_{\pm 1.86}$	$33.67_{\pm 1.40}$	$57.58_{\pm 0.23}$	$37.30_{\pm 1.52}$	$55.12_{\pm 0.34}$
Exact ratio	$50.49_{\pm 2.21}$	$67.49_{\pm 5.67}$	$69.20_{\pm 2.11}$	$50.00_{\pm 1.22}$	$33.27_{\pm 1.21}$	$56.56_{\pm 3.00}$	$37.40_{\pm 1.11}$	$54.24_{\pm 2.56}$
KPop[†]	$51.33_{\pm 0.23}$	$71.87_{\pm 1.08}$	$71.07_{\pm 1.21}$	$50.20_{\pm 3.57}$	$33.33_{\pm 1.89}$	$58.36_{\pm 0.33}$	$37.50_{\pm 2.19}$	$55.84_{\pm 0.25}$
IcePop	$51.79_{\pm 0.40}$	$72.81_{\pm 1.44}$	$71.40_{\pm 2.42}$	$52.64_{\pm 0.35}$	$35.40_{\pm 1.64}$	$58.93_{\pm 0.10}$	$39.66_{\pm 1.30}$	$56.60_{\pm 0.09}$
TIS	$48.76_{\pm 1.81}$	$69.77_{\pm 6.27}$	$70.87_{\pm 1.79}$	$48.17_{\pm 1.22}$	$33.47_{\pm 1.60}$	$56.15_{\pm 2.85}$	$37.10_{\pm 1.51}$	$53.85_{\pm 2.32}$
Seq-TIS	$51.43_{\pm 0.65}$	$\mathbf{73.41}_{\pm 0.57}$	$71.00_{\pm 1.91}$	$50.20_{\pm 2.75}$	$\mathbf{36.13}_{\pm 1.33}$	$58.80_{\pm 0.56}$	$39.61_{\pm 0.45}$	$56.48_{\pm 0.44}$
Seq-MIS	$50.45_{\pm 1.57}$	$72.35_{\pm 0.92}$	$70.80_{\pm 1.59}$	$50.81_{\pm 1.86}$	$35.87_{\pm 0.83}$	$57.88_{\pm 1.28}$	$39.56_{\pm 0.92}$	$55.66_{\pm 1.02}$
GSPO	$50.32_{\pm 1.82}$	$72.61_{\pm 1.19}$	$70.67_{\pm 3.29}$	$44.31_{\pm 3.36}$	$32.53_{\pm 1.30}$	$57.84_{\pm 1.72}$	$35.44_{\pm 1.76}$	$55.13_{\pm 1.69}$
FP16	$49.19_{\pm 0.70}$	$73.27_{\pm 0.26}$	$68.67_{\pm 0.42}$	$48.78_{\pm 3.23}$	$34.47_{\pm 1.17}$	$57.06_{\pm 0.49}$	$38.00_{\pm 0.86}$	$54.75_{\pm 0.44}$
CIS (ours)	$\mathbf{51.87}_{\pm 0.25}$	$72.92_{\pm 1.28}$	$\mathbf{72.13}_{\pm 1.75}$	$\mathbf{53.25}_{\pm 2.46}$	$35.80_{\pm 2.40}$	$\mathbf{59.08}_{\pm 0.52}$	$\mathbf{40.11}_{\pm 2.24}$	$\mathbf{56.79}_{\pm 0.71}$

Table 13 evaluates the checkpoints of Table 1 on Qwen1.5-MoE-A2.7B on three knowledge benchmarks (MMLU-STEM, ARC-Challenge, and OpenBookQA) and two code benchmarks (HumanEval and MBPP), none of which the RL stage uses. Every trained arm ends above the untrained model

on the 5,489 pooled items, so RL on GSM8K does not erase knowledge or coding ability. CIS has the highest pooled accuracy (56.79), but its margin over IcePop (56.60) is smaller than the standard deviations, so we do not claim a separation between the two. The exact ratio and TIS have the widest spreads on the pooled accuracy (± 2.56 and ± 2.32), as they do in Table 1.

D.4 Computational Cost



(a) (b)

Figure 11: Compute time per step on Qwen1.5-MoE-A2.7B, excluding checkpointing and evaluation. (a) Time by phase; dashed: no correction. (b) Mean response length.

Table 14: Compute time per optimizer step (seconds) on eight A100 GPUs, mean $\pm$ standard deviation over three seeds. Rescore is the training-engine forward pass that produces the training-side log-probabilities.

Method	Rollout	Rescore	Train step	Advantage	Total	vs. no corr.	Length
No correction	$0.44_{\pm 0.02}$	$0.00_{\pm 0.00}$	$29.41_{\pm 0.08}$	$0.82_{\pm 0.02}$	$30.66_{\pm 0.09}$	—	$321_{\pm 1}$
Exact ratio	$0.48_{\pm 0.03}$	$6.50_{\pm 0.08}$	$23.26_{\pm 0.40}$	$0.72_{\pm 0.01}$	$30.97_{\pm 0.48}$	$+1.0\%$	$252_{\pm 3}$
KPop†	$0.45_{\pm 0.01}$	$6.64_{\pm 0.58}$	$23.90_{\pm 2.08}$	$0.74_{\pm 0.03}$	$31.73_{\pm 2.68}$	$+3.5\%$	$260_{\pm 24}$
IcePop	$0.49_{\pm 0.03}$	$6.51_{\pm 0.23}$	$23.12_{\pm 0.67}$	$0.77_{\pm 0.06}$	$30.88_{\pm 0.93}$	$+0.7\%$	$250_{\pm 8}$
TIS	$0.48_{\pm 0.03}$	$6.40_{\pm 0.13}$	$22.89_{\pm 0.56}$	$0.74_{\pm 0.01}$	$30.51_{\pm 0.69}$	-0.5%	$249_{\pm 6}$
Seq-TIS	$0.49_{\pm 0.02}$	$7.39_{\pm 0.04}$	$26.44_{\pm 0.24}$	$0.72_{\pm 0.01}$	$35.03_{\pm 0.28}$	$+14.3\%$	$289_{\pm 2}$
Seq-MIS	$0.49_{\pm 0.00}$	$7.61_{\pm 0.23}$	$27.44_{\pm 0.66}$	$0.74_{\pm 0.01}$	$36.28_{\pm 0.89}$	$+18.3\%$	$300_{\pm 7}$
FP16	$0.46_{\pm 0.03}$	$0.00_{\pm 0.00}$	$28.81_{\pm 0.34}$	$0.84_{\pm 0.05}$	$30.11_{\pm 0.40}$	-1.8%	$316_{\pm 2}$
CIS (ours)	$0.49_{\pm 0.03}$	$6.67_{\pm 0.13}$	$23.77_{\pm 0.48}$	$0.72_{\pm 0.01}$	$31.65_{\pm 0.62}$	$+3.2\%$	$259_{\pm 4}$

Every operator is an elementwise map of the rollout-side and training-side log-probabilities, so no operator adds a forward or backward pass relative to the others. The arms that use the decoupled loss run one rescoring pass of the training engine per step (6.4 to 7.6 seconds), and the train step of these arms is shorter by a similar amount; the uncorrected and fp16 arms do not rescore. Figure 11 and Table 14 report the resulting compute time per step. The token-level operators lie between -0.5% and $+3.5\%$ of the uncorrected run, and CIS costs 3.2% more. Seq-TIS and Seq-MIS cost 14% and 18% more, but they also generate longer responses (289 and 300 tokens against 249 to 260 for the token-level operators), and response length changes the step time independently of the operator. Normalizing by the number of tokens does not remove this confound either, because the uncorrected run processes the most tokens per step and is therefore the fastest per token. The wall-clock time between steps also includes checkpointing and evaluation, which differ between runs by up to 25 seconds per step because of how often each run was scheduled to evaluate, not because of the operator; we exclude them here.

D.5 Where Each Operator Acts

Figure 12 shows which tokens each operator removes or down-weights during training. Early, middle, and late are the first, middle, and last twentieth of training; kept tokens are subsampled to 25,000 per panel, and between 68% and 85% of all tokens have $\pi^{\text{trn}}(y_t) > 0.99$ and overlap in the upper right corner. Early in training the two probabilities agree closely and every operator acts on less than

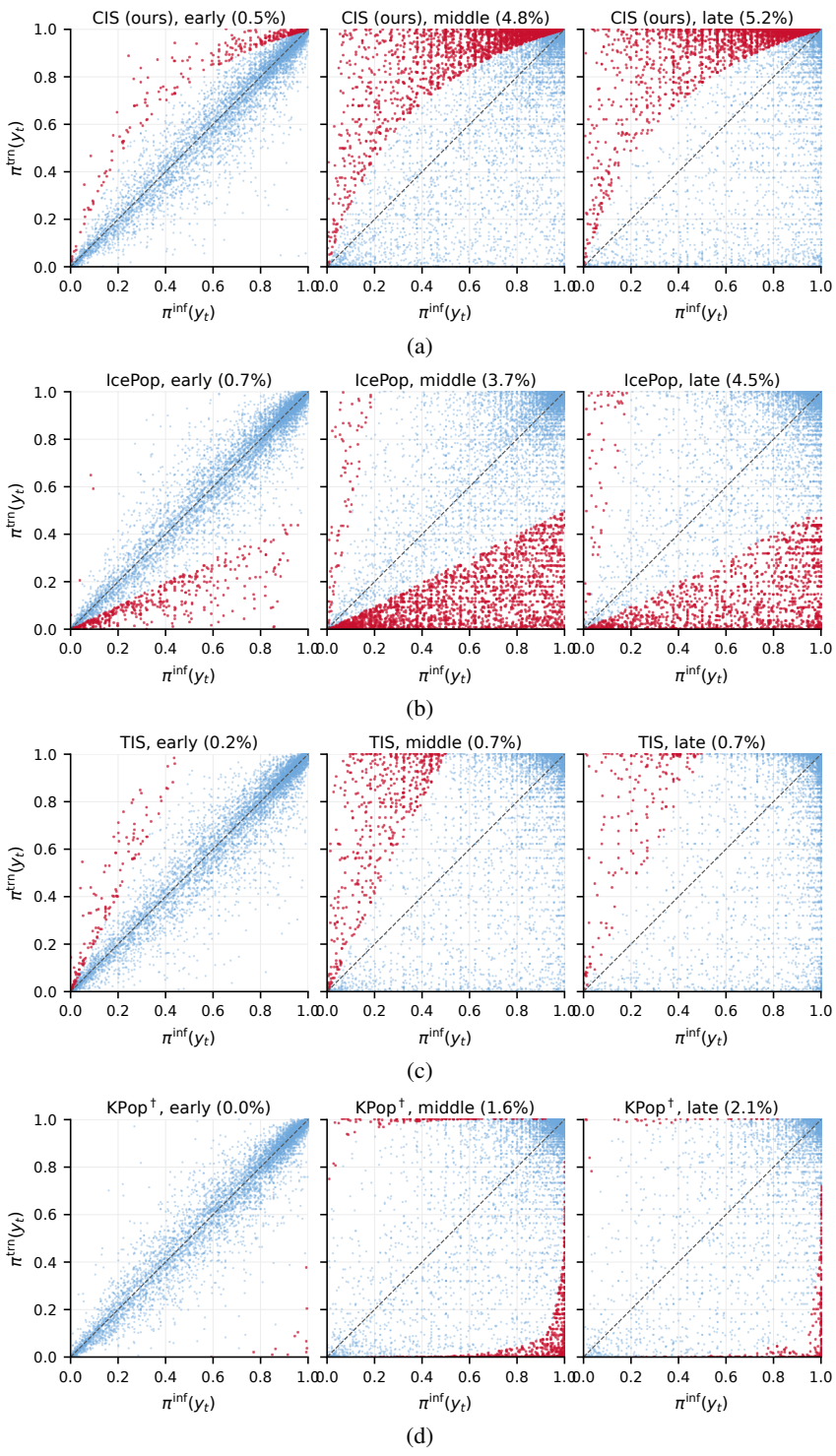


Figure 12: Tokens kept (blue) and truncated or masked (red) in the run of each operator (seed two), early, middle, and late in training; panel titles give the share acted on. (a) CIS. (b) IcePop. (c) TIS. (d) KPop†.

1% of tokens. As the policy sharpens, the spread around the diagonal grows. The operators then act on different regions of the plane. CIS truncates only above the diagonal, and its truncated tokens concentrate at $\pi^{\text{trn}}(y_t) \to 1$, where the training engine is confident and the displacement exceeds the threshold. This concentration does not mean that the floor does most of the work. Applying Algorithm 1 to the logged tokens of the three CIS runs, about one third of the truncated tokens lie below the floor, where the cap equals $1 + \lambda\kappa \approx 1.01$, and these tokens account for only about 10% of the total truncated excess $\sum_t (k_t - f_t)$; more than half of it comes from tokens with $1 - p_t \geq 0.1$. IcePop masks mainly below the diagonal, on tokens whose ratio falls under 0.5, which the one-sided band of CIS leaves untouched. TIS caps the tokens whose training-side probability exceeds twice the inference-side probability, which lie mostly at low $\pi^{\text{inf}}(y_t)$. KPop† masks tokens near the edges of the plane, where one engine assigns a probability close to one and the other does not, at a rate that grows to 2.1% late in training. CIS and IcePop therefore act on complementary sets of tokens.

The KL mask and probability saturation. The binary-KL metric of the KPop baseline is $\mathrm{KL}(\mathrm{Bern}(p) \,\|\, \mathrm{Bern}(q)) = p \log(p/q) + (1-p)\log((1-p)/(1-q))$ evaluated in both directions, with p and q clamped to $[10^{-8}, 1 - 10^{-8}]$. In single precision the upper clamp is the identity, so a token whose probability is exactly one on both engines yields $0 \cdot \log(0/0)$, and a token whose probability is exactly one on one engine yields $+\infty$; the bounds check treats both as out of bounds. On the tokens logged in the KPop run, 16.5% have probability exactly one on both engines and 2.8% on one engine, which together with the 0.45% of tokens whose finite metric exceeds the threshold reproduces the 22% intervention rate that the framework reports. Recomputing the metric in double precision with a representable clamp and mapping the undefined case to zero gives a mask rate of 0.45%. The main text reports this corrected variant as KPop†, so that the comparison is not driven by probability saturation.Figure 12 counts tokens, so the truncations of CIS appear concentrated at high confidence: 83% of the tokens that CIS truncates have $p_t \geq 0.9$, compared with 25% under TIS. By removed weight, however, more than half of the truncation of CIS comes from tokens with $p_t < 0.9$.

D.6 Limitations

Models and engines. We train only mixture-of-experts models, on which the mismatch has a heavy tail (Section A). On the dense control the displacement is concentrated near zero, so we expect every correction to matter less there, but we do not train a dense model. The measurements and the training runs use bf16 and specific inference and training engines; other kernels or numerical formats can change the size of the mismatch, in which case λ may need to be retuned. In asynchronous training, k_t also contains the policy lag of stale rollouts, which our analysis does not model separately; stale tokens make up less than 1% of the tokens in our runs.

Hyperparameters and baselines. The threshold λ and the floor κ are chosen from a preregistered grid on Qwen1.5-MoE-A2.7B and kept fixed for the other two models, whereas every baseline uses the default hyperparameters of its reference implementation. A comparable sweep for each baseline, and combinations of CIS with methods that reduce the mismatch at its source, such as routing replay (Ma et al., 2025) or FP16 training (Qi et al., 2025), are left to future work.

Scope of the theory. Theorems 3.2 and 3.3 give upper bounds under bounded per-token gradients and independent token contributions; for correlated contributions, the sample size n is replaced by $n_{\text{eff}} = n/(1 + (n-1)\bar{\rho})$ (Section B). Any cap with a finite maximum admits a bound similar to Theorem 3.3, so the theory explains why truncation is needed but does not by itself rank CIS against other bounded caps; that comparison rests on the experiments of Section 4.

What the token-level correction does not correct. The target gradient G_0 corrects the conditional distribution of each token given its prefix and not the distribution of the prefix itself. Neither the exact ratio nor CIS removes the shift in the prefix distribution. Seq-TIS and Seq-MIS act on the product of the ratios over a response, which addresses the prefix, but the product leaves any fixed interval for most responses of a few hundred tokens: Seq-MIS rejects 95% of the responses in our runs (Table 10), and Seq-TIS caps 12 to 17% of them.