VESPO: Variational Sequence-level Soft Policy Optimization for Off-Policy LLM Training

Guobin Shen **Chenxiao Zhao** **Xiang Cheng** **Lei Huang** **Xing Yu***
Xiaohongshu Inc.

Abstract

Off-policy updates are inevitable in reinforcement learning (RL) for large language models (LLMs) due to rollout staleness from asynchronous training and mismatches between training and inference engines. Naive importance sampling gives an unbiased correction but suffers from high variance, which is amplified by unbounded ratios and autoregressive generation. Prior remedies either rely on scenario-specific engineering, or trade bias for variance via token-level clipping or sequence-level normalization, yet these approaches remain largely heuristic. We propose **V**ariational s**E**quence-level **S**oft **P**olicy **O**ptimization (**VESPO**). By explicitly incorporating variance reduction into a variational formulation, we derive a principled closed-form reshaping kernel that operates directly on sequence-level importance weights, avoids token-level approximation and length normalization, and admits an explicit second-moment bound under a moment constraint. Experiments on math reasoning and code generation show that VESPO maintains stable training under severe off-policy conditions (staleness up to $64\times$) and delivers consistent gains across both dense and Mixture-of-Experts (MoE) models, outperforming recent reshaping baselines under matched setup. Code is available at `https://github.com/FloyedShen/VESPO`.

1 Introduction

Reinforcement learning (RL) has become a key technique for tackling complex problem-solving tasks with large language models (LLMs), enabling capabilities such as multi-step mathematical reasoning and code generation [29, 2, 6, 41].

In practice, off-policy updates arise naturally in RL pipelines for LLMs. A common source is that systems split large rollout batches into mini-batches for sequential updates [41], causing later batches to become stale relative to the evolving policy. Asynchronous systems [10, 49, 28] amplify this by decoupling rollout from training entirely. Training-inference mismatches introduce further discrepancies, especially in MoE models where routing decisions compound through layers. To stabilize training under such distribution mismatch, existing works adopt truncated importance sampling (TIS) [23], mask out off-policy samples from training [15], or replay expert routing for target policy inference [46, 25]. While PPO [32] enforces trust region constraints via clipping, it does not directly address the variance challenge of sequence-level importance sampling (IS).

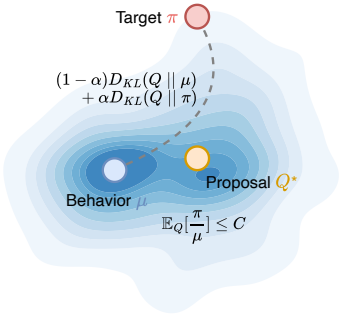


Figure 1: VESPO reformulates IS weight reshaping as finding a proposal Q^* that balances proximity to μ and π under a variance constraint.

*Correspondence to: `yuanshan2@xiaohongshu.com`, `floyed_shen@outlook.com`.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

Existing methods address this challenge through various *importance weight transformations*. Most operate at the token level: GRPO [33, 6] applies PPO-style clipping to per-token ratios; other methods such as SAPO [11] design heuristic transformations for off-policy importance weights [40, 47, 8, 31, 16]. However, token-level transformations are a compromise to avoid the multiplicative variance explosion of sequence-level weights, and have been shown to be merely a first-order approximation to their sequence-level counterparts [46, 45]. Sequence-level approaches [46, 44] introduce length normalization to control variance, but this normalization makes the IS estimator biased; hard clipping is still required on top. Despite these efforts, principled guidance for designing importance weight transformations remains limited.

We propose **V**ariational s**E**quence-level **S**oft **P**olicy **O**ptimization (**VESPO**), which builds on classical importance-sampling variance control and KL-regularized variational updates (Section 5): rather than designing reshaping heuristics directly, we incorporate variance reduction for off-policy importance sampling into a variational formulation over the proposal, yielding a closed-form solution (Figure 1). Additionally, we find that the resulting transformation is particularly friendly to sequence-level optimization: unlike previous methods that rely on length normalization to avoid variance explosion, VESPO operates directly on sequence-level importance weights without approximation or normalization.

Our contributions are summarized as follows:

- We recast importance weight reshaping as a measure change to an implicit proposal Q and formulate a constrained variational problem over Q that uses the sequence-level likelihood ratio W as the tilting statistic under the moment constraint $\mathbb{E}_Q[W] \leq C$. The KL-regularized objective and its exponential-tilting solution are classical; this choice of statistic and constraint yields a smooth closed-form reshaping kernel.
- Under this moment constraint we prove a second-moment certificate for the kernel that is non-vacuous iff the kernel's power exponent c_1 is at least 1 (Proposition 3.1), and show that the kernel is uniformly bounded. The resulting algorithm, VESPO, applies the kernel directly to sequence-level weights without length normalization, preserving inter-token dependencies.
- Experiments on verifier-based math reasoning and execution-based code generation show that, with one fixed coefficient setting, VESPO matches or outperforms recent reshaping baselines on dense and MoE models and remains stable under staleness up to $64\times$ on the MoE model.

2 Preliminaries

Notation. We consider an autoregressive language model parameterized by θ as a policy π_θ. Let x denote a query sampled from a dataset $\mathcal{D}$. In off-policy settings, responses $y = (y_1, \ldots, y_T)$ are sampled from a behavior policy μ (e.g., an earlier checkpoint or a different inference engine). The likelihood of generating y given x factorizes as:

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

We write $\tau = (x, y)$ for a query-response pair, and $R(\tau) \in \mathbb{R}$ for the sequence-level reward assigned to the complete response.

Policy Gradient with Off-Policy Correction. The goal is to maximize the expected reward under the current policy:

$$\mathcal{J}(\theta) = \mathbb{E}_{\tau\sim\pi_\theta}\big[R(\tau)\big]. \tag{2}$$

When samples are drawn from a behavior policy μ instead, importance sampling provides an unbiased correction. Taking the gradient yields the off-policy policy gradient:

$$\nabla_\theta \mathcal{J}(\theta) = \mathbb{E}_{\tau\sim\mu}\left[W(\tau) \cdot R(\tau) \cdot \nabla_\theta \log \pi_\theta(\tau)\right], \tag{3}$$

where $W(\tau) = \pi_\theta(\tau)/\mu(\tau)$ is the importance weight. This classical policy gradient view [37, 7] reveals that the importance weight W serves as a *gradient weighting factor*: it determines how much each sample contributes to the parameter update. More generally, any modification to the importance weight can be understood as defining a reshaping function $\phi(W)$ that reweights the gradient:

$$\nabla_\theta \tilde{\mathcal{J}}(\theta) = \mathbb{E}_{\tau\sim\mu}\left[\phi(W(\tau)) \cdot R(\tau) \cdot \nabla_\theta \log \pi_\theta(\tau)\right]. \tag{4}$$

This gradient-centric view will be central to our analysis: the practical effect of any weight transformation must ultimately be understood through how it reweights the policy gradient.

The Variance Challenge of Sequence-Level IS. Expanding the importance weight in terms of token-level ratios reveals a fundamental structural tension. Define the token-level importance ratio as $\rho_t = \frac{\pi_\theta(y_t|x,y_{<t})}{\mu(y_t|x,y_{<t})}$. The sequence-level weight is then a product:

$$W(\tau) = \prod_{t=1}^{T} \rho_t, \tag{5}$$

while the log-policy gradient is a sum:

$$\nabla_\theta \log \pi_\theta(\tau) = \sum_{t=1}^{T} \nabla_\theta \log \pi_\theta(y_t \mid x, y_{<t}). \tag{6}$$

This product-sum structure creates a tension: the gradient contribution of each token is weighted by a global factor $W(\tau)$ that compounds across all T positions. Even small per-token deviations accumulate multiplicatively, causing $W(\tau)$ to exhibit extreme values for long sequences. The variance of W grows exponentially with T, rendering naive importance sampling impractical.

To tame this variance, existing methods define specific reshaping functions ϕ that modify the gradient weighting. GRPO [33] operates at the token level with a PPO-style clipped surrogate. From the gradient perspective, the effective weight function depends on the sign of the advantage A:

$$\phi_{\text{GRPO}}(\rho_t; A) = \begin{cases} \rho_t, & \text{if } A > 0 \text{ and } \rho_t \leq 1+\varepsilon, \\ \rho_t, & \text{if } A < 0 \text{ and } \rho_t \geq 1-\varepsilon, \\ 0, & \text{otherwise (gradient zeroed).} \end{cases} \tag{7}$$

This breaks the product structure and treats each token update independently, yielding only a first-order approximation [45].

GSPO [46] operates at the sequence level, defining the gradient weight as the geometric mean of token-level ratios (i.e., normalizing by sequence length):

$$\phi_{\text{GSPO}}(W) = \left(\prod_{t=1}^{T} \rho_t\right)^{1/T} = \exp\left(\frac{1}{T}\sum_{t=1}^{T} \log \rho_t\right), \tag{8}$$

followed by a clipping mechanism similar to GRPO. This normalization introduces a length-dependent bias: the implicit proposal distribution varies with T, and sequences with identical per-token statistics but different lengths receive identical weights despite having different true importance weights (see Appendix H for a formal analysis). These approaches all define ϕ heuristically; the question of what constitutes a principled choice of ϕ motivates the variational framework we develop next.

3 VESPO: Variational Sequence-level Soft Policy Optimization

We develop a principled framework for designing importance weight transformations. We first show that any reshaping function $\phi(W)$ implicitly defines a proposal distribution, then formulate the design of ϕ as a variational problem with variance constraints, and finally derive a closed-form solution.

3.1 Weight Reshaping as Measure Change

Standard importance sampling performs an unbiased measure change $\mu \to \pi$, while any transformation $\phi(W)$ induces a different measure change $\mu \to Q$ for some implicit proposal Q. We now formalize this perspective. For any function $G(\tau)$:

$$\mathbb{E}_{\tau\sim\mu}\big[\phi(W(\tau)) \cdot G(\tau)\big] = Z \cdot \mathbb{E}_{\tau\sim Q}\big[G(\tau)\big], \tag{9}$$

where $Z = \mathbb{E}_\mu[\phi(W)]$ is a normalization constant, and Q is defined by

$$Q(\tau) = \frac{1}{Z}\mu(\tau) \cdot \phi(W(\tau)). \tag{10}$$

The reshaped gradient (Equation (4)) thus equals $Z \cdot \mathbb{E}_{\tau\sim Q}\big[R(\tau)\,\nabla_\theta \log \pi_\theta(\tau)\big]$: an expectation under the proposal Q retaining π_θ's score function (Q's own score differs unless $\phi \propto W$).

This is the key insight: any sequence-level reshaping function $\phi(W)$ implicitly defines a proposal distribution Q. This perspective provides a unified lens to analyze existing importance weight transformations (see Appendix A for detailed analysis of specific methods). Rather than handcrafting ϕ directly, we can specify desirable properties of the proposal Q and derive the corresponding ϕ. A good proposal should remain close to μ for sampling efficiency, incorporate π to guide optimization, and control variance. The following subsections formalize these desiderata as a variational objective and derive a closed-form ϕ^*.

3.2 Variational Objective

KL prior with importance tilt. We seek a proposal Q that retains sample efficiency from the available μ-samples while shifting mass toward trajectories favored by the target π. We formalize this as a KL-regularized linear objective:

$$\mathcal{J}(Q) \;=\; D_{\mathrm{KL}}(Q\|\mu) \;-\; \alpha\,\mathbb{E}_Q[\log W], \qquad \alpha \geq 0. \tag{11}$$

The KL term keeps Q close to the sampling distribution, ensuring that μ-samples remain informative for estimating expectations under Q. The log-importance term rewards Q for placing mass where π exceeds μ (i.e., $W > 1$), tilting Q toward the target policy and reducing bias in the gradient estimate. The coefficient α acts as an inverse temperature controlling the strength of this tilt: $\alpha = 0$ recovers $Q = \mu$, $\alpha = 1$ yields $Q = \pi$, and $\alpha > 1$ concentrates Q on increasingly high-importance trajectories. This KL-regularized linear form and its exponential-tilting (Gibbs) solution are classical, appearing in I-projection [4], REPS [30], the MPO E-step [1], and control-as-inference [21]. REPS and the MPO E-step tilt a base measure by reward or advantage to build a policy-update target; here the optimization variable is the proposal Q and the tilting statistic is the sequence-level ratio W.

Variance Constraint. Proximity alone is insufficient: finite sample sizes demand variance control. In importance sampling, the variance of the estimator scales with the second moment $\mathbb{E}_\mu[W^2]$, a classical result that also underlies the effective sample size (ESS) diagnostic.

Under the measure-change view, this second moment can be related to an expectation under Q. By Equation (10), we have $Q(\tau) \propto \mu(\tau)\phi(W(\tau))$, so

$$\mathbb{E}_{\tau\sim Q}[W(\tau)] \propto \mathbb{E}_{\tau\sim\mu}[\phi(W)\cdot W]. \tag{12}$$

When $\phi(W) \approx W$ (approaching unbiased IS), this recovers $\mathbb{E}_\mu[W^2]$; for general ϕ with $K := \sup_w \phi(w)/w$, the chain $\mathbb{E}_\mu[\phi(W)^2] = Z\cdot\mathbb{E}_Q[\phi(W)] \leq ZK\cdot\mathbb{E}_Q[W]$ (formalized in Proposition 3.1) shows that bounding $\mathbb{E}_Q[W]$ controls the second moment for any reshaping kernel. We therefore impose:

$$\mathbb{E}_{\tau\sim Q}[W(\tau)] \leq C. \tag{13}$$

Constrained Optimization. Combining the KL-regularized tilt objective with the variance constraint:

$$\begin{aligned} &\min_Q\; D_{\mathrm{KL}}(Q\|\mu) - \alpha\,\mathbb{E}_Q[\log W] \\ &\text{s.t. } \mathbb{E}_Q[W] \leq C, \quad \textstyle\int Q(\tau)\,d\tau = 1. \end{aligned} \tag{14}$$

Introducing Lagrange multipliers $\lambda \geq 0$ for the moment constraint and γ for normalization, we obtain the Lagrangian:

$$\begin{aligned} \mathcal{L}(Q,\lambda,\gamma) = D_{\mathrm{KL}}(Q\|\mu) - \alpha\,\mathbb{E}_Q[\log W] \\ + \lambda\big(\mathbb{E}_Q[W]-C\big) + \gamma\big(\textstyle\int Q\,d\tau - 1\big). \end{aligned} \tag{15}$$

3.3 Closed-Form Solution

Taking the functional derivative $\frac{\delta\mathcal{L}}{\delta Q} = 0$ yields (see Appendix C):

$$Q^*(\tau) \;\propto\; \mu(\tau)\cdot W(\tau)^\alpha \cdot \exp(-\lambda W(\tau)). \tag{16}$$

Comparing with Equation (10), we identify the reshaping function:

$$\phi(W) = W^\alpha \cdot \exp(-\lambda W). \tag{17}$$

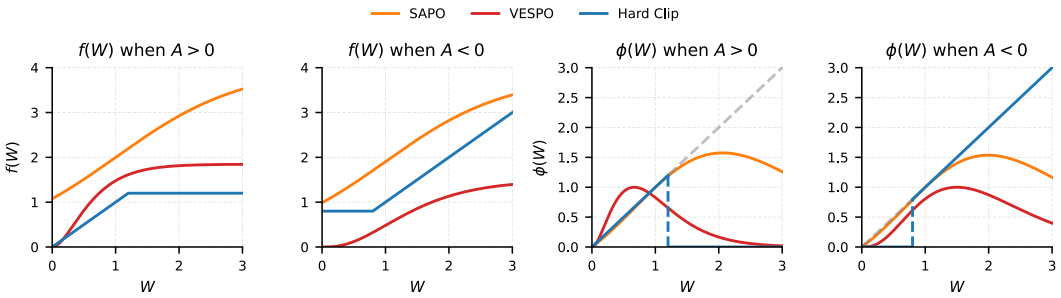


Figure 2: Surrogate objectives $f(W)$ and gradient scaling factors $\phi(W) = W \cdot f'(W)$ for positive and negative advantages. Hard clipping zeros ϕ abruptly at the boundary; VESPO peaks at $W{=}c_1/c_2$ and decays smoothly (VESPO curves scaled so that $\max \phi{=}1$).

This kernel has two components: the power term W^α and the exponential term $\exp(-\lambda W)$ for soft suppression. It is smooth and differentiable everywhere, avoiding the discontinuities of hard clipping.

Surrogate Objective. The reshaped gradient implicitly maximizes a smooth surrogate $\mathcal{J}_{\text{VESPO}}(\theta) = \mathbb{E}_\mu[f(W(\theta))A(\tau)]$ that saturates as $W \to \infty$ (Figure 2); f is the lower incomplete gamma function (full derivation in Appendix C).

Shifted form for practice. In what follows, we use the shifted form $\phi(W) = W^{c_1}\exp(c_2(1{-}W))$ with $c_1 = \alpha$, $c_2 = \lambda$ (objective coefficient and Lagrange multiplier, respectively) to ensure $\phi(1) = 1$, so that on-policy samples receive unit weight. This shift differs from Equation (17) only by the multiplicative constant e^{c_2}, equivalent to rescaling the learning rate, so the variational derivation carries over unchanged.

3.4 Variance Bound and Staleness Robustness

The variational derivation in Section 3.3 identifies the kernel form, but the constraint $\mathbb{E}_Q[W] \le C$ controls only the first moment under Q. We now establish a formal variance bound that translates this constraint into an explicit guarantee for the second moment under μ, which holds for the deployed kernel under the stated moment constraint.

Proposition 3.1 (Variance Bound)**.** *Let $\phi(W) = W^{c_1}\exp(c_2(1{-}W))$ with $c_1, c_2 > 0$, and define $K = \sup_{w>0}\phi(w)/w$. Under the moment constraint $\mathbb{E}_Q[W] \le C$,*

$$\mathbb{E}_\mu\left[\phi(W)^2\right] \;\le\; Z \cdot K \cdot C, \quad Z = \mathbb{E}_\mu[\phi(W)]. \tag{18}$$

Moreover, $K < \infty$ if and only if $c_1 \ge 1$. When finite, K admits the closed form

$$K = \left(\frac{c_1 - 1}{c_2}\right)^{c_1 - 1} \exp(c_2 - c_1 + 1), \tag{19}$$

attained at $w^* = (c_1{-}1)/c_2$ for $c_1 > 1$.

The proof, given in Appendix D, follows by setting $G = \phi(W)$ in the measure-change identity (Equation (9)) and applying $\phi(w) \le Kw$. The bound is non-trivial precisely when $c_1 \ge 1$: at our practical settings, $(c_1, c_2) = (2, 3)$ gives $K \approx 2.46$ and $(3, 2)$ gives $K = 1$, while for $c_1 < 1$ the bound becomes vacuous ($K = \infty$). In the unbiased-IS limit ($\phi \approx W$), the bound reduces to $\mathbb{E}_\mu[W^2] \lesssim Z \cdot K \cdot C$, i.e., the classical second-moment quantity controlling effective sample size (ESS) and IS variance, so Proposition 3.1 can be viewed as extending this classical control to general ϕ. POIS [26] controls the same second moment $\mathbb{E}_\mu[W^2]$ of the ordinary ratio in its high-confidence bound while retaining that ratio in its return term; Proposition 3.1 instead certifies a smooth reshaping kernel. The criterion $c_1 \ge 1$ identified by the bound coincides with the inverse-temperature regime of the variational objective (Section 3.2, $\alpha = c_1 \ge 1$). Theory therefore identifies a regime, not a unique pair: our deployed values $(2, 3)$ and $(3, 2)$ are simple integers in this regime (Section 3.5).

Formal constraint versus deployed coefficients. In the derivation, $c_1 = \alpha$ is an objective coefficient and $c_2 = \lambda$ is the Lagrange multiplier for $\mathbb{E}_Q[W] \le C$. In implementation, we fix branch-specific values of (c_1, c_2); in particular, c_2 is not obtained by solving the dual for a prescribed common C. The implementation therefore neither enforces nor records a known common C: the formal constraint

motivates the kernel form, the deployed coefficients are fixed hyperparameters, and Proposition 3.1 is a certificate under the stated moment constraint.

Uniform Boundedness. The smooth kernel is also uniformly bounded for any $c_1, c_2 > 0$ (proof and explicit $\phi_{\max}$ in Appendix D). Consequently, no single off-policy sample can produce an unbounded gradient contribution, no matter how stale (i.e., no matter how large its raw weight W). Hard-clipping methods achieve a similar property only through discontinuous truncation, while VESPO's smooth attenuation preserves differentiability throughout.

3.5 The VESPO Algorithm

We instantiate the shifted kernel from Section 3.3 with asymmetric hyperparameters (c_1, c_2) for positive and negative advantages, mirroring the asymmetric clipping in PPO. Recent work [38] has shown that positive and negative samples exhibit different gradient dynamics during training; accordingly, we apply stronger suppression for $A < 0$ with $W < 1$ to prevent excessive penalization of samples the policy already dislikes, as shown in Figure 2. We fix $(c_1^+, c_2^+) = (2, 3)$ and $(c_1^-, c_2^-) = (3, 2)$: simple integers with $c_1 \geq 1$ (so K is finite; Section 3.4), whose stronger suppression for $A < 0$ samples with $W < 1$ is motivated by the failure-mode analysis (Section 4.3, Appendix I). These values are used unchanged in every experiment, without per-condition retuning. The asymmetric assignment is an instance of the same kernel functional family (rather than a new family), so Proposition 3.1 applies to each branch since both (c_1^+, c_2^+) and (c_1^-, c_2^-) satisfy $c_1, c_2 > 0$. Substituting the shifted kernel into Equation (4), the VESPO gradient estimator becomes:

$$\nabla \mathcal{J}_{\text{VESPO}} = \mathbb{E}_{\tau \sim \mu}\big[W^{c_1} \exp(c_2(1-W)) \cdot A(\tau) \cdot \nabla \log \pi_\theta(\tau)\big], \tag{20}$$

where $A(\tau) = R(\tau) - b$ is the advantage with baseline b (mean reward within each prompt group, following GRPO [33]). For general $\phi \neq W$, the baseline acts as a control variate for the deployed surrogate $\mathbb{E}_\mu[\phi(W)A(\tau)]$ rather than preserving the unbiasedness identity that holds only for $\phi = W$.

The kernel adapts smoothly to the importance weight: $\phi(W) \approx 1$ near on-policy, the exponential term decays for $W \gg 1$, and the power term down-weights $W \ll 1$. The factorized form gives flexible control: c_1 governs $W < 1$ behavior and c_2 controls decay for $W > 1$.

Computational cost. VESPO incurs no additional forward passes or memory beyond GRPO/GSPO: it reuses the same per-token log-probabilities under π_θ and μ, computes $\log W$ as their masked sum, and applies ϕ as an elementwise reshaping detached from the computation graph. We evaluate $c_1 \log W + c_2(1 - W)$ in log-space, exponentiating only at the final step to avoid overflow under extreme staleness. This makes VESPO a drop-in replacement in standard RL pipelines (full pseudocode in Appendix L).

4 Experiments

We evaluate VESPO on **mathematical reasoning** (the primary setting) and **code generation** (cross-domain transfer; Section 4.5). For math, we examine two practical sources of off-policy distribution shift: (1) **policy staleness** from batched rollouts, where later mini-batches are updated using samples from an outdated policy; and (2) **train-inference mismatch**, where different implementations between training and inference engines produce different outputs for the same input, an effect exacerbated in MoE models due to routing inconsistencies.

4.1 Experimental Setup

We train on DAPO-Math [42] (verifier reward [17]) for math reasoning across three model scales: Llama-3.2-3B-Instruct [12], Qwen3-8B-Base, and Qwen3-30B-A3B-Base [41], on 32 H20 GPUs with veRL [34]. Math evaluation uses AIME 2024/2025, AMC 2023, and MATH-500 [13] (avg@k). For cross-domain transfer (Section 4.5), we additionally train Qwen3-30B-A3B-Base on PRIME-RL/Eurus-2-RL-Data [5] with execution reward and evaluate on HumanEval+, MBPP+ [24], and LiveCodeBench v6 [19] (pass@10). Best checkpoint per method is selected by average accuracy. Baselines GRPO [33], GSPO [46], SAPO [11], TOPR [31], CISPO [3], and BAPO [40] use their respective official hyperparameters. For the main Table 1 baselines, GRPO uses DAPO's asymmetric clipping $(0.2, 0.28)$, GSPO uses sequence-level clipping $(3{\times}10^{-4}, 4{\times}10^{-4})$, and SAPO uses $\tau_{\text{pos}}{=}1.0, \tau_{\text{neg}}{=}1.05$ (Appendix E). All methods use the same training data, reward, update budget, staleness setup, and infrastructure. VESPO uses the fixed setting $(c_1, c_2) = (2, 3)$ for $A{>}0$ and $(3, 2)$ for $A{<}0$ in every experiment (dense and MoE, all staleness levels, math and code), without

per-condition retuning; only the explicit coefficient ablations (Section 4.7, Appendix F) vary it. Staleness is simulated by fixing mini-batch size 256 and varying global batch size; primary experiments use staleness ratio N=gbs/mbs=8, ablations span $N \in \{4, 8, 16, 32, 64\}$. Full per-method hyperparameters and infrastructure details are in Appendix E.

4.2 Main Results

Table 1 presents the main results with gbs/mbs = 8 across three model scales. VESPO achieves the best (or tied-best) average accuracy on all three models, demonstrating the generality of our approach. To assess run-to-run variability, we repeat GRPO and VESPO on Qwen3-8B-Base (N=8) with three training seeds (Table 6, Appendix F); in this separate summary, distinct from the single-run Table 1 entries, the macro-averages of benchmark means are 54.5 (GRPO) and 61.1 (VESPO), and VESPO has the higher mean on all four benchmarks. Notably, the improvements are most pronounced on Qwen3-30B-A3B-Base, where VESPO outperforms the best Table 1 baseline by 10.9 percentage points in average accuracy (70.0 vs 59.1). This suggests that VESPO's soft suppression of extreme importance weights is particularly beneficial for MoE architectures, where routing inconsistencies amplify distribution shift and make training stability more challenging (we do not isolate the router-specific component; Section 4.6). Given these observations, we focus our detailed analysis on Qwen3-30B-A3B-Base in the following sections.

Table 1: Mathematical reasoning accuracy (%) with gbs/mbs = 8 across three model scales. Best results in **bold**.

Model	Method	AIME25	AIME24	AMC23	MATH500	Avg
Llama-3.2-3B-Instruct	GRPO	0.5	14.5	40.6	**51.6**	26.8
	GSPO	0.2	**14.7**	43.9	47.8	26.7
	SAPO	**0.7**	12.2	34.1	51.0	24.5
	VESPO	0.6	13.9	**47.3**	51.3	**28.3**
Qwen3-8B-Base	GRPO	27.4	40.0	74.5	76.4	54.6
	GSPO	28.8	37.7	80.8	78.4	56.4
	SAPO	**36.7**	49.0	80.0	78.0	**60.9**
	VESPO	33.5	**49.4**	**82.2**	**78.6**	**60.9**
Qwen3-30B-A3B-Base	GRPO	28.2	40.0	81.4	78.3	57.0
	GSPO	25.1	43.3	83.0	**84.8**	59.1
	SAPO	21.4	27.9	73.0	84.6	51.7
	VESPO	**44.3**	**59.6**	**91.4**	**84.8**	**70.0**

4.3 Robustness to Policy Staleness

We examine robustness to policy staleness by varying the staleness ratio N = gbs/mbs from 4 to 64; with the engine stack fixed, increasing N increases policy staleness (Table 8). The training-reward curves in Figure 3 show VESPO's remarkable consistency: all five curves (N=4 to N=64) follow nearly identical trajectories, converging to similar final rewards around 0.7. In contrast, GRPO saturates early; GSPO degrades as N grows, and additionally suffers a catastrophic collapse around step 1,200 at N=4; SAPO is competitive at N=4 until its training reward collapses near step 950, is unstable from N=8 onward, and downstream accuracy collapses at $N \geq 32$. These differences translate to downstream performance (Table 2): VESPO achieves the best average accuracy at every N, maintaining 61.8% at N=64, while GRPO and GSPO degrade to 48.6% and 47.6% respectively, and SAPO collapses entirely.

Table 2: Effect of staleness ratio N = gbs/mbs on Qwen3-30B-A3B-Base (avg@k, %). N=8 is in Table 1. SAPO's downstream accuracy collapses at $N \geq 32$.

$N = 4$						$N = 16$					
Method	AIME25	AIME24	AMC23	MATH500	Avg	Method	AIME25	AIME24	AMC23	MATH500	Avg
GRPO	22.1	33.1	76.4	84.5	54.0	GRPO	20.3	31.4	71.4	81.3	51.1
GSPO	27.6	43.3	83.1	85.7	59.9	GSPO	24.3	41.6	83.1	84.2	58.3
SAPO	38.4	51.4	85.2	**85.9**	65.2	SAPO	19.6	26.0	72.2	84.3	50.5
VESPO	**43.1**	**60.3**	**91.2**	85.4	**70.0**	**VESPO**	**40.2**	**53.2**	**90.8**	**84.8**	**67.3**
$N = 32$						$N = 64$					
Method	AIME25	AIME24	AMC23	MATH500	Avg	Method	AIME25	AIME24	AMC23	MATH500	Avg
GRPO	21.8	33.4	73.4	80.9	52.4	GRPO	14.6	28.0	69.4	82.4	48.6
GSPO	18.8	27.3	73.4	80.6	50.0	GSPO	15.4	24.1	73.9	76.8	47.6
SAPO	12.4	7.2	29.5	34.5	20.9	SAPO	3.3	7.3	23.8	31.4	16.5
VESPO	**37.7**	**51.4**	**85.2**	**83.7**	**64.5**	**VESPO**	**34.2**	**46.2**	**83.6**	**83.3**	**61.8**

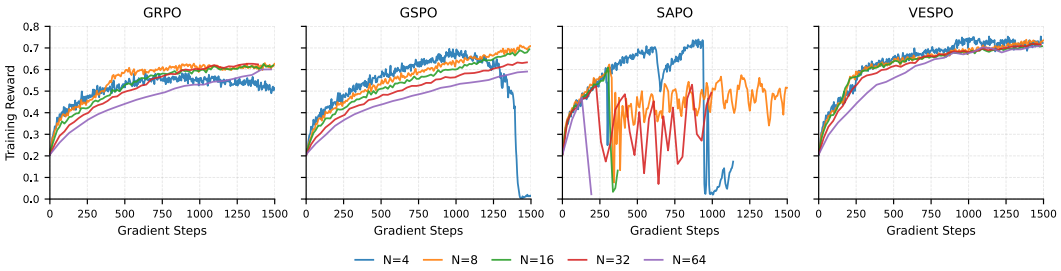


Figure 3: Training reward across staleness levels ($N \in \{4, 8, 16, 32, 64\}$) on Qwen3-30B-A3B-Base. Each panel shows one method; VESPO is the only one with stable, consistent curves across all N.

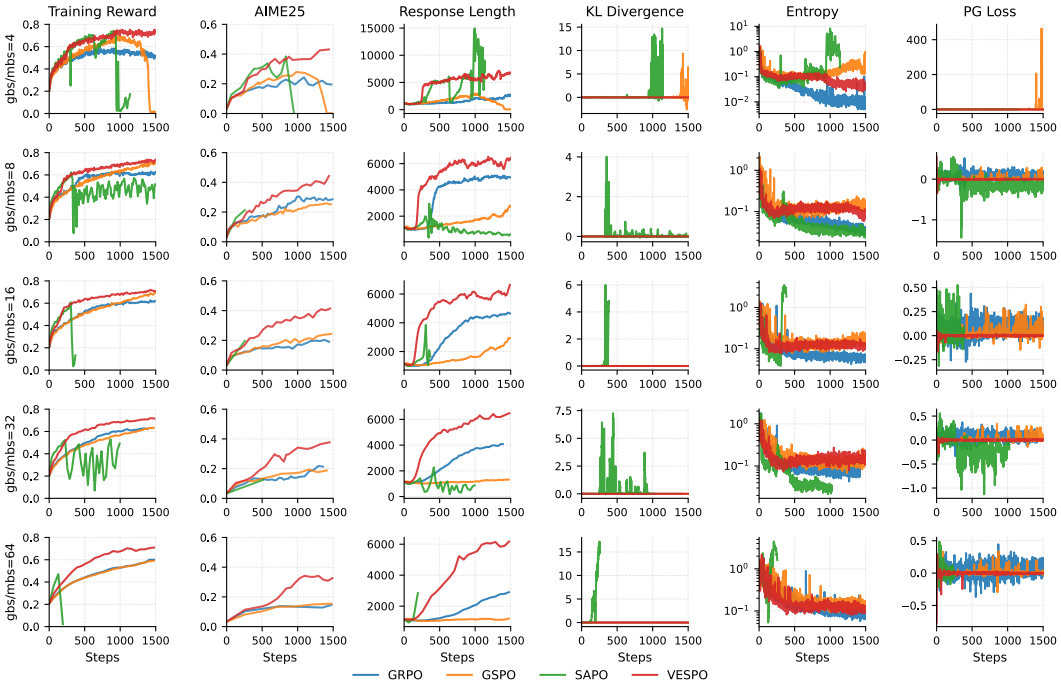


Figure 4: Training dynamics on Qwen3-30B-A3B-Base across staleness $N \in \{4, 8, 16, 32, 64\}$ (rows: N; columns: training reward, AIME25, response length, KL, entropy, PG loss). VESPO (red) is the only method with stable training across all N; baseline failure modes (GRPO entropy collapse, GSPO length-bias amplification, SAPO under-suppressed $A<0$) are detailed in Appendix I.

4.4 Comparison with Recent Importance Weight Methods

Recent works propose alternative importance weight transformations targeting specific instability sources. We compare VESPO directly against three representative methods on Qwen3-30B-A3B-Base under identical training setup (N=8): TOPR [31] hard-clamps sequence-level ratios to $[0, 1]$ for negative rewards (and uses weight 1 for positive rewards) with length normalization; CISPO [3] clips token-level ratios as stop-gradient REINFORCE weights; BAPO [40] adjusts PPO clipping bounds dynamically with a hard cap for negative advantages.

Table 3: Comparison with recent importance weight reshaping methods on Qwen3-30B-A3B-Base (N=8).

Method	AIME25	AIME24	AMC23	MATH500	Avg
TOPR	16.5	29.9	66.6	81.8	48.7
CISPO	16.0	25.5	70.2	83.0	48.7
BAPO	34.2	47.9	88.9	83.2	63.6
VESPO	**44.3**	**59.6**	**91.4**	**84.8**	**70.0**

Table 4: Code generation (pass@10) using the same fixed (c_1, c_2) as math.

Method	HE+	MBPP+	LCB v6	Avg
GRPO	84.8	74.9	23.1	60.9
GSPO	82.9	73.0	23.1	59.7
SAPO	86.6	74.6	24.2	61.8
VESPO	**88.4**	**75.4**	**25.3**	**63.0**

Table 3 shows VESPO substantially outperforms all three baselines: $+10$ on AIME25 over BAPO and $+28$ over TOPR/CISPO. The training dynamics (Figure 11 in Appendix J) reveal distinct failure modes: *CISPO* collapses around step 280 from insufficient sequence-level variance control; *BAPO* exhibits entropy explosion after step 1,100 as adaptive clip bounds over-relax; *TOPR* converges slowly with suppressed response length.

4.5 Code Generation

Table 4 reports pass@10 on HumanEval+, MBPP+, and LiveCodeBench v6 after training on code (setup in Section 4.1). With the same fixed coefficients as math, VESPO has the highest reported point estimate on all three benchmarks (88.4, 75.4, 25.3), averaging 63.0 pass@10 versus 61.8 for SAPO. This supports reuse of the fixed setting across the evaluated verifier-based math and execution-based code settings; preference-based rewards are not evaluated here. Training dynamics and detailed analysis are in Appendix K.

4.6 Robustness to Train-Inference Mismatch

A second source of off-policy shift is *train-inference mismatch*: different numerical implementations between training (FSDP/Megatron) and inference (vLLM/SGLang [48]) engines produce different outputs, amplified in MoE models. Engineering fixes include Truncated IS (TIS) [23] and Routing Replay (R2) [25]. Figure 5 shows vanilla GRPO plateaus at $\sim$0.60; R2 improves it, TIS only until a late collapse. VESPO *without fixes* matches GRPO+R2; VESPO+R2 attains the highest training reward, showing complementarity with these fixes; both VESPO runs end best on AIME25.

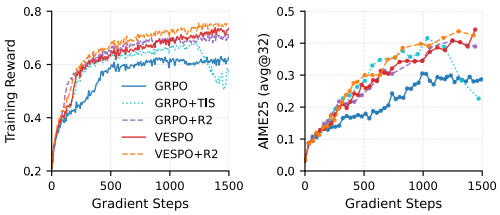


Figure 5: GRPO and VESPO with and without engineering mitigations (TIS, R2) under the combined off-policy discrepancy (train-inference mismatch and policy staleness) at N=8 on Qwen3-30B-A3B-Base.

Separating mismatch sources. Both policy staleness and train-inference mismatch are present in the MoE runs. Following Eq. 5 of Zheng et al. [45], $\pi_\theta/\mu_{\text{old}} = (\pi_{\text{old}}/\mu_{\text{old}}) \cdot (\pi_\theta/\pi_{\text{old}})$ (i.e., W with $\mu = \mu_{\text{old}}$), where the first factor is train-inference discrepancy (trainer vs. vLLM at the same checkpoint) and the second, r_s, is policy staleness between rollout-time and current parameters in the training engine. Table 2 varies N with the engine stack fixed, and the logged $-\log r_s$ statistic grows with N (Table 8), characterizing increasing policy staleness. Figure 5 compares engineering mitigations under the combined discrepancy at N=8. The dense Qwen3-8B-Base setting has no router-specific component, although numerical train-inference differences can still arise. Because a matched same-checkpoint trainer–vLLM measurement was not recorded, our runs do not separately quantify the train-inference component (Appendix M).

4.7 Ablations

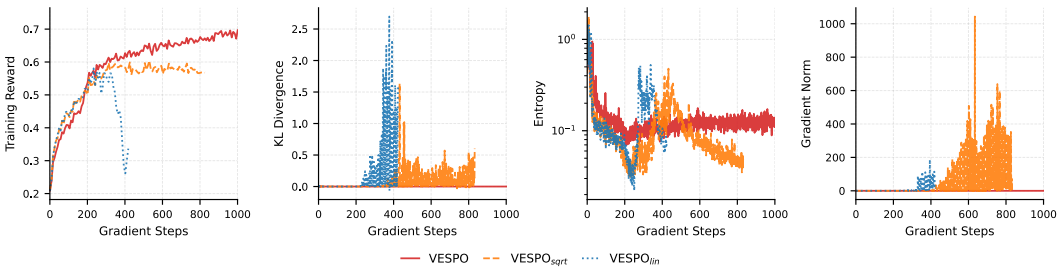


Figure 6: Ablation on length normalization. VESPO without normalization achieves stable training; adding $\sqrt{T}$ or T normalization causes instability and collapse.

Length normalization. A key design choice in VESPO is operating at the sequence level *without* length normalization. Methods like GSPO normalize by $1/T$ to reduce variance, but we hypothesize this creates length-dependent bias. We compare VESPO with two normalized variants: VESPO$_{\text{sqrt}}$ (normalize by $\sqrt{T}$) and VESPO$_{\text{lin}}$ (normalize by T). Figure 6 reveals striking differences: VESPO$_{\text{lin}}$ collapses around step 350 (KL spike + gradient explosion); VESPO$_{\text{sqrt}}$ shows periodic gradient spikes;

VESPO without normalization is stable. Length normalization causes longer sequences to dominate batch gradients, biasing toward even longer outputs until collapse, a failure mode VESPO avoids.

Asymmetric hyperparameters. VESPO uses asymmetric hyperparameters $c^+ = (2, 3)$ and $c^- = (3, 2)$ for positive and negative advantages. We compare against symmetric variants (Figure 7). Using $(2, 3)$ for both gives insufficient suppression for $A < 0$, leading to instability; using $(3, 2)$ for both over-suppresses positive samples, ending slightly lower. The asymmetric design balances these trade-offs. Complementing this ablation, a local one-at-a-time sensitivity study on Qwen3-8B-Base at N=8 (Table 7) gives four-benchmark averages from 60.0 to 61.0 across the tested settings (60.9 for the default Table 1 run).

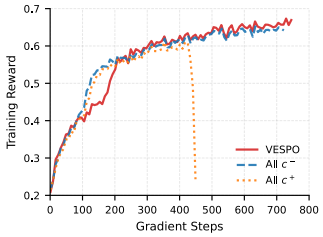


Figure 7: Asymmetric hyperparameter ablation.

5 Related Work

Classical off-policy correction and variational RL. VESPO builds on classical ingredients. Policy-gradient estimation rests on the policy-gradient theorem [37]; Off-PAC adds per-decision importance correction [7]; POIS studies trajectory-level importance sampling [26]. KL-regularized objectives with exponential-tilting solutions appear in I-projection [4], REPS [30], the MPO E-step [1], and control-as-inference [21]; REPS and MPO tilt a base measure by reward or advantage to construct a policy-update target, whereas VESPO uses W as the tilting statistic under $\mathbb{E}_Q[W] \leq C$, yielding the $\exp(-\lambda W)$ damping. Bias–variance control of importance weights includes truncated IS [18], the per-decision hard truncation of V-trace [9] and Retrace [27], and defensive mixtures [14], which modify the proposal before sampling; PSIS [39] fits a generalized Pareto tail to batch order statistics and is thus batch- and rank-dependent; Rényi-divergence VI [22] uses power tempering of ratios. VESPO instead applies a fixed, analytic, smooth pointwise kernel to the sequence-level ratio of fixed stale rollouts. We thus view VESPO as adapting and combining these ideas for sequence-level LLM RL; our claimed contribution is the constrained proposal formulation with W as the tilting statistic, the resulting kernel and certificate, and the sequence-level implementation (Appendix B).

Policy Gradient Methods for LLMs. PPO [32] stabilizes updates via a clipped surrogate objective, while value-free alternatives have gained traction for LLM fine-tuning: GRPO [33] normalizes rewards within sample groups and clips token-level ratios; GSPO [46] operates at the sequence level with geometric-mean (length) normalization; DAPO [42] introduces decoupled clipping and dynamic sampling. These methods control variance via heuristic clipping or normalization. Our measure-change view (Section 3.1) reveals each as a specific choice of implicit proposal distribution, and VESPO derives the weight function from a variational principle rather than manual design.

Importance Weight Reshaping. Recent alternatives to hard clipping can be categorized along three axes: granularity, boundary, and theoretical origin. Token-level hard-boundary methods include CISPO [3], BAPO [40], GPPO/KLEAR [36], and M2PO [47]; however, per-token reshaping does not compose into IS under any single sequence-level proposal (Appendix A). SAPO [11] uses token-level smooth gating, sharing VESPO's philosophy but inheriting the token-level limitation. TOPR [31] is sequence-level but with hard clamping to $[0, 1]$ for negative rewards and length normalization, suffering from boundary discontinuity and length bias (Appendix H). Among these LLM reshaping methods, VESPO combines sequence-level operation, a smooth kernel, and a variational derivation, outperforming TOPR/CISPO/BAPO under matched setup (Table 3). Engineering fixes such as Routing Replay [25] and truncated IS [23] target specific instability sources and are complementary to VESPO (Section 4.6).

6 Conclusion

We presented a measure-change view of importance weight reshaping and derived VESPO, a sequence-level smooth kernel with a second-moment certificate under the stated moment constraint. VESPO is stable under staleness up to $64\times$ on the MoE model, matches or outperforms baselines on dense and MoE models, and transfers to code generation without retuning. **Limitations.** Theory identifies the regime $c_1 \geq 1$ rather than unique coefficient values (we use one fixed setting outside the coefficient ablations); our evaluation covers verifier-based math and execution-based code, and preference-based rewards are not evaluated; our largest model is a 30B MoE; and in the MoE runs the train-inference component of the off-policy discrepancy is not separately quantified (Appendix M).

Acknowledgments and Disclosure of Funding

We thank the area chair and the reviewers for their constructive feedback, which improved the positioning, uncertainty reporting, and scope statements of this paper. This work was carried out at the authors' listed affiliation, Xiaohongshu Inc. The authors declare no competing interests.

References

[1] Abbas Abdolmaleki, Jost Tobias Springenberg, Yuval Tassa, Remi Munos, Nicolas Heess, and Martin Riedmiller. Maximum a posteriori policy optimisation. In *International Conference on Learning Representations*, 2018.

[2] Anthropic. Introducing Claude 4. `https://www.anthropic.com/news/claude-4`, 2025. Accessed: 2025-05-22.

[3] 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, 2025. URL `https://arxiv.org/abs/2506.13585`.

[4] Imre Csiszár. I-divergence geometry of probability distributions and minimization problems. *The Annals of Probability*, 3(1):146–158, 1975.

[5] Ganqu Cui, Lifan Yuan, Zefan Wang, Hanbin Wang, Yuchen Zhang, Jiacheng Chen, Wendi Li, Bingxiang He, Yuchen Fan, Tianyu Yu, Qixin Xu, Weize Chen, Jiarui Yuan, Huayu Chen, Kaiyan Zhang, Xingtai Lv, Shuo Wang, Yuan Yao, Xu Han, Hao Peng, Yu Cheng, Zhiyuan Liu, Maosong Sun, Bowen Zhou, and Ning Ding. Process reinforcement through implicit rewards, 2025. URL `https://arxiv.org/abs/2502.01456`.

[6] DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, et al. DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning. *Nature*, 645:633 – 638, 2025. URL `https://api.semanticscholar.org/CorpusID:275789950`.

[7] Thomas Degris, Martha White, and Richard S. Sutton. Off-policy actor-critic. In *Proceedings of the 29th International Conference on Machine Learning*, 2012.

[8] Madeleine Dwyer, Adam Sobey, and Adriane Chapman. It's not you, it's clipping: A soft trust-region via probability smoothing for LLM RL, 2025. URL `https://arxiv.org/abs/2509.21282`.

[9] Lasse Espeholt, Hubert Soyer, Remi Munos, Karen Simonyan, Vlad 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*, volume 80 of *Proceedings of Machine Learning Research*, pages 1407–1416, 2018.

[10] Wei Fu, Jiaxuan Gao, Xujie Shen, Chen Zhu, Zhiyu Mei, Chuyi He, Shusheng Xu, Guo Wei, Jun Mei, Jiashu Wang, Tongkai Yang, Binhang Yuan, and Yi Wu. AREAL: A large-scale asynchronous reinforcement learning system for language reasoning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL `https://openreview.net/forum?id=X9diEuva9R`.

[11] Chang Gao, Chujie Zheng, Xiong-Hui Chen, Kai Dang, Shixuan Liu, Bowen Yu, An Yang, Shuai Bai, Jingren Zhou, and Junyang Lin. Soft adaptive policy optimization, 2025. URL `https://arxiv.org/abs/2511.20347`.

[12] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, et al. The Llama 3 herd of models, 2024. URL `https://arxiv.org/abs/2407.21783`.

[13] 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.

[14] Tim Hesterberg. Weighted average importance sampling and defensive mixture distributions. *Technometrics*, 37(2):185–194, 1995.

[15] Jian Hu. Stabilizing MoE RL without router replay: The online IcePop solution. https://hijkzzz.notion.site/online-ice-pop, 12 2025. Accessed: 2025-12-16.

[16] Jian Hu, Jason Klein Liu, Haotian Xu, and Wei Shen. REINFORCE++: Stabilizing critic-free policy optimization with global advantage normalization, 2025. URL https://arxiv.org/abs/2501.03262.

[17] Hugging Face. Math-verify: A rule-based math verification library. https://github.com/huggingface/Math-Verify, 2025. Accessed: 2025-01-26.

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

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

[20] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th symposium on operating systems principles*, pages 611–626, 2023.

[21] Sergey Levine. Reinforcement learning and control as probabilistic inference: Tutorial and review, 2018. URL https://arxiv.org/abs/1805.00909.

[22] Yingzhen Li and Richard E. Turner. Rényi divergence variational inference. In *Advances in Neural Information Processing Systems*, volume 29, 2016.

[23] 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.

[24] Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. Is your code generated by ChatGPT really correct? rigorous evaluation of large language models for code generation, 2023. URL https://arxiv.org/abs/2305.01210.

[25] 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, 2025. URL https://arxiv.org/abs/2510.11370.

[26] Alberto Maria Metelli, Matteo Papini, Francesco Faccio, and Marcello Restelli. Policy optimization via importance sampling. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

[27] Rémi Munos, Tom Stepleton, Anna Harutyunyan, and Marc G. Bellemare. Safe and efficient off-policy reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 29, 2016.

[28] Michael Noukhovitch, Shengyi Huang, Sophie Xhonneux, Arian Hosseini, Rishabh Agarwal, and Aaron Courville. Asynchronous RLHF: Faster and more efficient off-policy RL for language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=FhTAG591Ve.

[29] OpenAI. Learning to reason with LLMs. https://openai.com/index/learning-to-reason-with-llms/, 2024. Accessed: 2024-09-12.

[30] Jan Peters, Katharina Mülling, and Yasemin Altun. Relative entropy policy search. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*, pages 1607–1612, 2010.

[31] Nicolas Le Roux, Marc G. Bellemare, Jonathan Lebensold, Arnaud Bergeron, Joshua Greaves, Alex Fréchette, Carolyne Pelletier, Eric Thibodeau-Laufer, Sándor Toth, and Sam Work. Tapered off-policy REINFORCE: Stable and efficient reinforcement learning for LLMs, 2025. URL https://arxiv.org/abs/2503.14286.

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

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

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

[35] Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. Megatron-LM: Training multi-billion parameter language models using model parallelism. *arXiv preprint arXiv:1909.08053*, 2019.

[36] Zhenpeng Su, Leiyu Pan, Xue Bai, Dening Liu, Guanting Dong, Jiaming Huang, Minxuan Lv, Wenping Hu, Fuzheng Zhang, Kun Gai, and Guorui Zhou. Klear-Reasoner: Advancing reasoning capability via gradient-preserving clipping policy optimization, 2025. URL https://arxiv.org/abs/2508.07629.

[37] Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems*, 12, 1999.

[38] Xinyu Tang, Yuliang Zhan, Zhixun Li, Wayne Xin Zhao, Zhenduo Zhang, Zujie Wen, Zhiqiang Zhang, and Jun Zhou. Rethinking sample polarity in reinforcement learning with verifiable rewards, 2025. URL https://arxiv.org/abs/2512.21625.

[39] Aki Vehtari, Daniel Simpson, Andrew Gelman, Yuling Yao, and Jonah Gabry. Pareto smoothed importance sampling. *Journal of Machine Learning Research*, 25(72):1–58, 2024.

[40] Zhiheng Xi, Xin Guo, Yang Nan, Enyu Zhou, Junrui Shen, Wenxiang Chen, Jiaqi Liu, Jixuan Huang, Zhihao Zhang, Honglin Guo, Xun Deng, Zhikai Lei, Miao Zheng, Guoteng Wang, Shuo Zhang, Peng Sun, Rui Zheng, Hang Yan, Tao Gui, Qi Zhang, and Xuanjing Huang. BAPO: Stabilizing off-policy reinforcement learning for LLMs via balanced policy optimization with adaptive clipping, 2025. URL https://arxiv.org/abs/2510.18927.

[41] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxin Yang, Jingren Zhou, Jingren Zhou, Junyan Lin, Kai Dang, Keqin Bao, Ke-Pei Yang, Le Yu, Li-Chun Deng, Mei Li, Min Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shi-Qiang Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yi-Chao Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report. *ArXiv*, abs/2505.09388, 2025. URL https://api.semanticscholar.org/CorpusID:278602855.

[42] Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, YuYue, 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 *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=2a36EMSSTp.

[43] Yanli Zhao, Andrew Gu, Rohan Varma, Liang Luo, Chien-Chin Huang, Min Xu, Less Wright, Hamid Shojanazeri, Myle Ott, Sam Shleifer, et al. PyTorch FSDP: experiences on scaling fully sharded data parallel. *arXiv preprint arXiv:2304.11277*, 2023.

[44] Yuzhong Zhao, Yue Liu, Junpeng Liu, Jingye Chen, Xun Wu, Yaru Hao, Tengchao Lv, Shaohan Huang, Lei Cui, Qixiang Ye, Fang Wan, and Furu Wei. Geometric-mean policy optimization, 2025. URL https://arxiv.org/abs/2507.20673.

[45] 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, 2025. URL https://arxiv.org/abs/2512.01374.

[46] Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, Jingren Zhou, and Junyang Lin. Group sequence policy optimization. *arXiv preprint arXiv:2507.18071*, 2025. URL https://arxiv.org/abs/2507.18071.

[47] Haizhong Zheng, Jiawei Zhao, and Beidi Chen. Prosperity before collapse: How far can off-policy RL reach with stale data on LLMs? In *NeurIPS 2025 Workshop on Efficient Reasoning*, 2025. URL https://openreview.net/forum?id=osEqHuHQWK.

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

[49] Yinmin Zhong, Zili Zhang, Xiaoniu Song, Hanpeng Hu, Chao Jin, Bingyang Wu, Nuo Chen, Yukun Chen, Yu Zhou, Changyi Wan, Hongyu Zhou, Yimin Jiang, Yibo Zhu, and Daxin Jiang. StreamRL: Scalable, heterogeneous, and elastic RL for LLMs with disaggregated stream generation. *arXiv preprint arXiv:2504.15930*, 2025. URL https://arxiv.org/abs/2504.15930.

VESPO: Variational Sequence-level Soft Policy Optimization Supplementary Material

Table of Contents

A Implicit Proposal Distributions of Existing Methods . . . 15
B Relation to Classical Off-Policy and Variational RL . . . 17
C Derivation of the Proposal Distribution . . . 17
D Proof of the Variance Bound and Uniform Boundedness . . . 19
E Experimental Details . . . 19
F Seed Variability and Coefficient Sensitivity . . . 20
G Policy-Staleness Diagnostic for the MoE Runs . . . 20
H Length Normalization Introduces Bias . . . 21
I Additional Analysis of Baseline Failure Modes . . . 21
J Training Dynamics for Recent Baseline Comparison . . . 22
K Code Generation Training Details and Dynamics . . . 22
L Algorithm Pseudocode . . . 24
M Limitations and Future Directions . . . 24

A Implicit Proposal Distributions of Existing Methods

We analyze the implicit proposal distributions induced by existing reshaping strategies under the measure-change framework of Section 3.1.

Token-level methods (GRPO). GRPO applies clipping independently at each token position, yielding a gradient of the form

$$\nabla J_{\text{GRPO}} = \sum_{t=1}^{T} \mathbb{E}_{\tau\sim\mu}\big[\phi_t(\rho_t) \cdot A(\tau) \cdot \nabla \log \pi(y_t|x, y_{<t})\big], \tag{21}$$

where $\phi_t(\rho_t)$ is the clipping function applied to the t-th token ratio. The key observation is that **different tokens within the same trajectory receive different weights** $\phi_t(\rho_t)$. In contrast, sequence-level methods assign a single weight $\phi(W)$ to the entire trajectory:

$$\nabla J_{\text{seq}} = \mathbb{E}_{\tau\sim\mu}\big[\phi(W) \cdot A(\tau) \cdot \sum_{t=1}^{T} \nabla \log \pi(y_t|x, y_{<t})\big]. \tag{22}$$

The token-level formulation cannot be expressed as importance sampling toward any single proposal distribution Q, because such a representation requires a uniform weight across all token gradients within each trajectory. This token-wise weighting breaks the coherence of sequence-level credit assignment: tokens in the same trajectory that share a common outcome receive inconsistent gradient signals.

Token-level IS as a First-Order Approximation. We now show that token-level IS is a first-order approximation to sequence-level IS, following Zheng et al. [45]. Consider the unclipped case where $\phi_t(\rho_t) = \rho_t$ and $\phi(W) = W$. The sequence-level gradient (without advantage for clarity) is

$$\nabla J_{\text{seq}} = \mathbb{E}_{\tau\sim\mu}\Big[W \cdot \sum_{t=1}^{T} \nabla \log \pi(y_t|\cdot)\Big] = \mathbb{E}_{\tau\sim\mu}\Big[\prod_{s=1}^{T} \rho_s \cdot \sum_{t=1}^{T} \nabla \log \pi(y_t|\cdot)\Big]. \tag{23}$$

When $\rho_t \approx 1$ for all t (near on-policy), we can expand $W = \prod_s \rho_s$ via Taylor series:

$$W = \prod_{s=1}^{T} \rho_s \approx 1 + \sum_{s=1}^{T} (\rho_s - 1) + \sum_{s<s'} (\rho_s - 1)(\rho_{s'} - 1) + \cdots \tag{24}$$

Substituting into the gradient and keeping only first-order terms in $(\rho_s - 1)$:

$$\nabla J_{\text{seq}} \approx \mathbb{E}_{\tau\sim\mu}\Big[\Big(1 + \sum_{s=1}^{T}(\rho_s - 1)\Big) \sum_{t=1}^{T} \nabla \log \pi(y_t|\cdot)\Big] \tag{25}$$

$$= \underbrace{\mathbb{E}_{\tau\sim\mu}\Big[\sum_{t=1}^{T} \nabla \log \pi(y_t|\cdot)\Big]}_{\text{REINFORCE (no IS)}} + \underbrace{\sum_{s,t} \mathbb{E}_{\tau\sim\mu}\big[(\rho_s - 1)\nabla \log \pi(y_t|\cdot)\big]}_{\text{first-order IS correction}}. \tag{26}$$

The token-level gradient is

$$\begin{aligned} \nabla J_{\text{tok}} &= \sum_{t=1}^{T} \mathbb{E}_{\tau\sim\mu}\big[\rho_t \cdot \nabla \log \pi(y_t|\cdot)\big] \\ &= \mathbb{E}_{\tau\sim\mu}\Big[\sum_{t=1}^{T} \nabla \log \pi(y_t|\cdot)\Big] + \sum_{t=1}^{T} \mathbb{E}_{\tau\sim\mu}\big[(\rho_t - 1)\nabla \log \pi(y_t|\cdot)\big]. \end{aligned} \tag{27}$$

Comparing, we see that token-level IS **only retains the diagonal terms** $(s = t)$ of the first-order correction, discarding cross-token interactions where $s \neq t$. The approximation error is

$$\nabla J_{\text{seq}} - \nabla J_{\text{tok}} \approx \sum_{s\neq t} \mathbb{E}_{\tau\sim\mu}\big[(\rho_s - 1)\nabla \log \pi(y_t|\cdot)\big] + O\big((\rho - 1)^2\big). \tag{28}$$

This error captures the fact that changing the policy at position s affects the importance of the gradient at position t, a cross-token dependency that token-level methods ignore.

Remark A.1 (When Token-Level Approximation is Reasonable). The approximation $\nabla J_{\text{tok}} \approx \nabla J_{\text{seq}}$ is reasonable when:

1. **Near on-policy**: $\rho_t \approx 1$ for all t, so higher-order terms and cross-token terms are small.
2. **Short sequences**: Fewer cross-token pairs (s, t) with $s \neq t$ means smaller accumulated error.

In practice, RL for LLMs often violates both conditions: off-policy updates are common, and reasoning tasks require long sequences. This motivates sequence-level methods that preserve the full product structure of importance weights.

Length normalization (GSPO). GSPO uses the geometric mean $\phi(W) = W^{1/T}$, which induces a proposal distribution that explicitly depends on sequence length:

$$Q_{\text{GSPO}}(\tau) \propto \mu(\tau)^{1-1/T} \cdot \pi(\tau)^{1/T}. \tag{29}$$

As $T \to \infty$, $Q_{\text{GSPO}} \to \mu$, meaning longer sequences receive vanishingly small corrections toward the target policy. See Appendix H for a detailed analysis of the length-dependent bias this introduces.

Sequence-level hard clipping. If one were to apply hard clipping at the sequence level (truncating W at threshold c), the reshaping function would be $\phi(W) = \min(W, c)$, inducing

$$Q_{\text{clip}}(\tau) \propto \min\big(\pi(\tau), c \cdot \mu(\tau)\big). \tag{30}$$

This is a *truncated distribution*: trajectories with $W > c$ are capped rather than weighted proportionally. The discontinuity at $W = c$ can cause optimization difficulties when trajectories cross the boundary during training. Note that standard PPO applies clipping at the token level rather than the sequence level, combining the issues of both token-level weighting and hard truncation.

Table 5: Classical components underlying VESPO and their relation to our contribution.

Component	Prior art	Our relation
Policy-gradient estimation with off-policy importance correction	Policy-gradient theorem [37]; Off-PAC [7]; trajectory-level IS in POIS [26]	Classical foundation. Sutton et al. establish the policy-gradient theorem, Off-PAC uses per-decision action-ratio correction, and POIS studies trajectory-level importance sampling; the sequence-level ratio itself is not our contribution.
KL-regularized variational objective; Gibbs form $Q^* \propto \mu\, e^{\alpha \log W - \lambda W}$	I-projection [4]; REPS [30]; MPO E-step [1]; control-as-inference [21]	The objective and exponential-tilting form are classical. MPO/REPS tilt a base measure by reward or advantage to construct a policy-update target; VESPO instead uses W as the tilting statistic and imposes $\mathbb{E}_Q[W] \leq C$, yielding the $\exp(-\lambda W)$ damping.
Variance control for IS weights	Truncated IS [18]; V-trace [9]; Retrace [27]; defensive-mixture IS [14]	The shared goal is bias–variance control. V-trace/Retrace use per-decision hard truncation, defensive mixtures modify the proposal before sampling, and VESPO applies a smooth transformation to the sequence-level ratio from fixed stale rollouts.
Smoothing extreme importance ratios	PSIS [39]	PSIS fits a generalized Pareto distribution to batch tail order statistics. It is batch- and rank-dependent, whereas VESPO uses a fixed analytic pointwise kernel for policy-gradient training.
Power tempering of importance ratios	Rényi-divergence VI [22]	Power tempering is not new in isolation; VESPO's W^{c_1} factor is an instance of it.

B Relation to Classical Off-Policy and Variational RL

VESPO adapts and combines classical importance-sampling variance control and KL-regularized variational ideas for sequence-level LLM RL. Table 5 separates the inherited components from what we claim as our contribution.

Claimed contribution. Our contribution is the combination of (i) formulating the constrained variational problem over an implicit proposal Q, with W as the tilting statistic (Section 3.2); (ii) identifying the induced smooth reshaping kernel and, under $\mathbb{E}_Q[W] \leq C$, proving the certificate $\mathbb{E}_\mu[\phi(W)^2] \leq ZKC$, with finite $K = \sup_w \phi(w)/w$ iff $c_1 \geq 1$ (Proposition 3.1); and (iii) applying the measure-change view to sequence-level LLM optimization (Section 3.5).

Relation to POIS. POIS [26] is a particularly close relative: its high-confidence bound is controlled by the second moment $\mathbb{E}_\mu[W^2]$ of the ordinary importance ratio, while its return term retains that ratio. VESPO instead derives a smooth reshaping kernel and certifies its second moment under the stated moment constraint.

Relation to V-trace. V-trace [9] applies per-decision hard truncation in both its multi-step value correction and actor update, and analyzes the value operator's convergence to an intermediate clipped policy. VESPO transforms the full sequence likelihood ratio with an analytically derived smooth kernel and provides a second-moment bound under the stated moment constraint. The commonality is variance control through an altered correction; the differences are granularity, kernel form, and guarantee.

C Derivation of the Proposal Distribution

We derive the closed-form solution to the constrained optimization problem in Equation (14).

Proposition C.1 (Solution to the Constrained Problem)**.** *The solution to*

$$\min_{Q} D_{\mathrm{KL}}(Q\|\mu) - \alpha\,\mathbb{E}_Q[\log W] \quad \text{s.t. } \mathbb{E}_Q[W] \le C, \quad \int Q(\tau)\,d\tau = 1 \tag{31}$$

is given by

$$Q^*(\tau) = \frac{1}{Z}\,\mu(\tau)\,W(\tau)^{\alpha}\,\exp(-\lambda W(\tau)), \tag{32}$$

where $\lambda \ge 0$ is the Lagrange multiplier for the moment constraint and Z is the normalization constant.

Proof. The Lagrangian, after expanding $D_{\mathrm{KL}}(Q\|\mu) = \int Q \log(Q/\mu)\,d\tau$, is

$$\mathcal{L}(Q,\lambda,\gamma) = \int Q(\tau)\big[\log Q(\tau) - \log\mu(\tau) - \alpha\log W(\tau) + \lambda W(\tau) + \gamma\big]d\tau \;-\; \gamma \;-\; \lambda C. \tag{33}$$

Taking the functional derivative with respect to Q and setting it to zero:

$$\frac{\delta\mathcal{L}}{\delta Q} = \log Q(\tau) + 1 - \log\mu(\tau) - \alpha\log W(\tau) + \lambda W(\tau) + \gamma = 0. \tag{34}$$

Solving for Q:

$$\log Q^*(\tau) = \log\mu(\tau) + \alpha\log W(\tau) - \lambda W(\tau) + \text{const} \tag{35}$$

$$Q^*(\tau) \propto \mu(\tau)\,W(\tau)^{\alpha}\,\exp(-\lambda W(\tau)). \tag{36}$$

Comparing with $Q(\tau) = \frac{1}{Z}\mu(\tau)\,\phi(W(\tau))$ from Equation (10), we identify

$$\phi(W) = W^{\alpha}\cdot\exp(-\lambda W). \tag{37}$$

□

Equivalent dual-KL form. Using the identity $D_{\mathrm{KL}}(Q\|\pi) = D_{\mathrm{KL}}(Q\|\mu) - \mathbb{E}_Q[\log W]$, the objective in Equation (11) can equivalently be written as

$$\mathcal{J}(Q) = (1-\alpha)\,D_{\mathrm{KL}}(Q\|\mu) + \alpha\,D_{\mathrm{KL}}(Q\|\pi), \tag{38}$$

which exposes the geometric interpretation: the proposal Q interpolates between μ (α=0) and π (α=1). Both forms yield the same Q^*; we use the single-KL form in the main text because $D_{\mathrm{KL}}(Q\|\mu)$ retains unit weight in the deployed regime $\alpha \ge 1$.

Surrogate Objective. The reshaped gradient (Equation (4)) corresponds to maximizing an implicit surrogate. Using $\nabla_\theta W = W\nabla_\theta \log\pi_\theta$, the estimator can be rewritten as

$$\mathbb{E}_{\tau\sim\mu}\big[\phi(W)\cdot A\cdot\nabla\log\pi_\theta\big] = \mathbb{E}_{\tau\sim\mu}\big[\tfrac{\phi(W)}{W}\cdot A\cdot\nabla W\big]. \tag{39}$$

Defining f by $f'(W) = \phi(W)/W$, this equals $\nabla_\theta\mathbb{E}_{\tau\sim\mu}[f(W)\,A(\tau)]$, since $A(\tau)$ does not depend on θ. Hence VESPO implicitly maximizes

$$\mathcal{J}_{\text{VESPO}}(\theta) = \mathbb{E}_{\tau\sim\mu}\big[f(W(\theta))\,A(\tau)\big]. \tag{40}$$

For $\phi(W) = W^{\alpha}\exp(-\lambda W)$, $f'(W) = W^{\alpha-1}\exp(-\lambda W)$ integrates to the *lower incomplete gamma function*:

$$f(W) = \frac{1}{\lambda^{\alpha}}\,\gamma(\alpha,\lambda W), \qquad \gamma(a,x) = \int_0^x t^{a-1}e^{-t}\,dt. \tag{41}$$

This form is smooth and infinitely differentiable, and saturates as $W\to\infty$, providing a principled soft alternative to hard clipping.

Shifted Form. In practice, we use the shifted form $\phi(W) = W^{c_1}\exp(c_2(1-W))$, which satisfies $\phi(1) = 1$. This ensures that on-policy samples receive unit weight. In implementation, (c_1, c_2) are fixed branch-specific hyperparameters, $(2,3)$ for $A>0$ and $(3,2)$ for $A<0$, rather than values obtained by solving the dual for a prescribed C (Section 3.4).

D Proof of the Variance Bound and Uniform Boundedness

We prove Proposition 3.1 from Section 3.4 and the supporting boundedness result for the deployed kernel.

Proof of Proposition 3.1. By Equation (10), $Q(\tau) = \mu(\tau)\phi(W(\tau))/Z$, so for any measurable $g : \mathbb{R}_{>0} \to \mathbb{R}$:

$$\mathbb{E}_\mu\big[\phi(W) \cdot g(W)\big] = Z \cdot \mathbb{E}_Q\big[g(W)\big]. \tag{42}$$

Setting $g = \phi$ gives $\mathbb{E}_\mu[\phi(W)^2] = Z \cdot \mathbb{E}_Q[\phi(W)]$. Since $\phi(w) = (\phi(w)/w) \cdot w \le K \cdot w$ for all $w > 0$ where $K = \sup_{w>0} \phi(w)/w$, we have

$$\mathbb{E}_Q[\phi(W)] \le K \cdot \mathbb{E}_Q[W] \le K \cdot C, \tag{43}$$

using the moment constraint $\mathbb{E}_Q[W] \le C$. Combining yields $\mathbb{E}_\mu[\phi(W)^2] \le Z \cdot K \cdot C$.

Finiteness and closed form of K. We have $\phi(w)/w = w^{c_1-1}\exp(c_2(1-w))$. As $w \to 0^+$, this tends to ∞ when $c_1 < 1$ (since $w^{c_1-1} \to \infty$), to $\exp(c_2)$ when $c_1 = 1$, and to 0 when $c_1 > 1$. As $w \to \infty$, the exponential dominates and the ratio tends to 0 for any $c_2 > 0$. Hence $K < \infty$ iff $c_1 \ge 1$.

For $c_1 > 1$, $h(w) := (c_1 - 1)\log w + c_2(1-w)$ has derivative $h'(w) = (c_1-1)/w - c_2$, vanishing at $w^* = (c_1-1)/c_2 > 0$. The second derivative $-(c_1-1)/w^2 < 0$ confirms an interior maximum. Substituting:

$$K = (w^*)^{c_1-1}\exp\big(c_2(1-w^*)\big) = \left(\frac{c_1-1}{c_2}\right)^{c_1-1}\exp(c_2 - c_1 + 1), \tag{44}$$

yielding the stated formula. For $c_1 = 1$ the supremum $K = \exp(c_2)$ is approached as $w \to 0^+$. □

Proposition D.1 (Uniform Boundedness of ϕ)**.** *For any $c_1, c_2 > 0$, the kernel $\phi(W) = W^{c_1}\exp(c_2(1-W))$ satisfies*

$$\sup_{W>0} \phi(W) \;=\; \phi_{\max} \;=\; \left(\frac{c_1}{c_2}\right)^{c_1}\exp(c_2 - c_1), \tag{45}$$

attained at $W^{**} = c_1/c_2$.

Proof. $\log\phi(W) = c_1 \log W + c_2(1-W)$ has derivative $c_1/W - c_2$, vanishing at $W^{**} = c_1/c_2$. The second derivative $-c_1/W^2 < 0$ confirms a maximum. Substituting yields $\phi_{\max} = (c_1/c_2)^{c_1}\exp(c_2 - c_1)$. □

Numerical values at practical settings. For $(c_1, c_2) = (2, 3)$: $K = (1/3)e^2 \approx 2.46$ and $\phi_{\max} = (2/3)^2 e \approx 1.21$. For $(c_1, c_2) = (3, 2)$: $K = 1$ and $\phi_{\max} = (3/2)^3 e^{-1} \approx 1.24$. The two propositions together establish that the deployed kernel offers (i) second-moment control under the moment constraint and (ii) uniformly bounded gradient contribution from any single off-policy sample, regardless of staleness.

E Experimental Details

Training infrastructure. All experiments run on 32 NVIDIA H20 GPUs with veRL [34], using vLLM 0.11.0 [20] for rollout inference, FSDP [43] for the dense models (Llama-3.2-3B-Instruct, Qwen3-8B-Base), and Megatron [35] for the MoE model (Qwen3-30B-A3B-Base). Each run has a budget of $1{,}500$ gradient steps with maximum sequence length $16{,}384$ tokens and 8 responses per query. Each training run takes approximately 48 hours on 32 H20 GPUs.

Reward and evaluation protocol. Math reward is verifier-based via Math-Verify [17]; code reward is execution-based (test case pass/fail). Math evaluation reports avg@k with k=32 for AIME, k=16 for AMC, and k=4 for MATH-500; code evaluation reports pass@10. The best checkpoint per method is selected by average accuracy across the corresponding evaluation suite.

Baseline hyperparameters. For the main Table 1 baselines we use the published configurations:

- **GRPO** [33]: token-level PPO clipping with asymmetric bounds $(0.2, 0.28)$ following DAPO [42].
- **GSPO** [46]: sequence-level with $1/T$ length normalization, clipping bounds (3e-4, 4e-4).
- **SAPO** [11]: token-level soft gating, τ_{pos}=1.0, τ_{neg}=1.05.

All methods share the same training data, reward, update budget (1,500 gradient steps), staleness setup, and infrastructure. VESPO uses $(c_1, c_2) = (2, 3)$ for $A>0$ and $(3, 2)$ for $A<0$ in every experiment, across models, staleness levels and tasks; only the explicit coefficient ablations (Figure 7, Table 7) use other values.

Staleness simulation. We fix mini-batch size 256 and vary global batch size to obtain staleness ratio $N = \text{gbs}/\text{mbs}$. Each rollout batch is split into N mini-batches for sequential gradient updates, so later mini-batches use parameters increasingly stale relative to the rollout policy. Primary experiments use N=8; staleness ablations span $N \in \{4, 8, 16, 32, 64\}$.

F Seed Variability and Coefficient Sensitivity

Three-seed replication. To assess run-to-run variability, we repeat GRPO and VESPO on Qwen3-8B-Base at N=8 with three training seeds (Table 6). The macro-averages of the benchmark means are 54.5 for GRPO and 61.1 for VESPO, and VESPO has the higher mean on all four benchmarks. This summary is distinct from the single-run entries of Table 1, including the VESPO run that serves as the default row of Table 7.

Table 6: Three-seed replication on Qwen3-8B-Base (N=8), GRPO and VESPO. Each benchmark cell reports mean ± std across the three training seeds; the final column is the macro-average of the benchmark means. This summary is distinct from the single-run Table 1 entries.

Method	AIME25	AIME24	AMC23	MATH500	Avg of means
GRPO	27.3 ± 1.5	40.5 ± 1.4	74.2 ± 1.5	75.9 ± 0.8	54.5
VESPO	33.7 ± 1.4	49.7 ± 1.3	82.1 ± 1.4	78.8 ± 0.8	61.1

Local coefficient sensitivity. We vary one displayed coefficient at a time on Qwen3-8B-Base (DAPO-Math, N=8), with all other choices matching Table 1 (Table 7). The averages remain close across the tested nearby configurations. This local grid complements the symmetric-versus-asymmetric ablation in Section 4.7 (Figure 7).

Table 7: Local one-at-a-time sensitivity of VESPO coefficients on Qwen3-8B-Base (DAPO-Math, N=8; all other choices as in Table 1). Each alternative changes one displayed coefficient while holding the others at their default values. Single-run point estimates; the default row is the Table 1 run, not the three-seed mean of Table 6.

$(c_1^+, c_2^+)/(c_1^-, c_2^-)$	AIME25	AIME24	AMC23	MATH500	Avg
$(2, 3)/(3, 2)$ (default)	33.5	49.4	82.2	78.6	60.9
$(1.5, 3)/(3, 2)$	31.7	48.8	81.4	78.2	60.0
$(2, 2)/(3, 2)$	32.8	49.2	81.9	78.5	60.6
$(2, 3)/(2.5, 2)$	34.2	48.6	82.3	78.8	61.0

G Policy-Staleness Diagnostic for the MoE Runs

Following Eq. 5 of Zheng et al. [45], the importance ratio factorizes as

$$\frac{\pi_\theta}{\mu_{\text{old}}} = \frac{\pi_{\text{old}}}{\mu_{\text{old}}} \cdot \frac{\pi_\theta}{\pi_{\text{old}}},$$

where the first factor is the train-inference discrepancy (trainer vs. vLLM at the same checkpoint) and the second, r_s, is policy staleness between rollout-time and current parameters in the training engine. Both sources are present in the MoE runs. Table 8 reports the logged diagnostic for the second factor in the fixed-engine N sweep of Table 2. As N increases, the $-\log r_s$ statistic grows, characterizing increasing policy staleness while the engine stack is fixed. Figure 5 evaluates engineering mitigations under the combined discrepancy at N=8. Because a matched measurement of the first factor was not recorded, these runs do not separately quantify the magnitude of the train-inference component (Appendix M).

Table 8: Policy-staleness diagnostic logged in the MoE runs of the fixed-engine N sweep (Table 2). $r_s = \pi_\theta/\pi_{\text{old}}$ is the staleness factor; the train-inference factor $\pi_{\text{old}}/\mu_{\text{old}}$ was not logged.

N	4	8	16	32	64
Median batch mean of $-\log r_s$	0.072	0.089	0.236	0.361	0.381

H Length Normalization Introduces Bias

We analyze how length normalization in sequence-level importance sampling introduces length-dependent bias that conflates distinct trajectories.

Length-Dependent Bias in GSPO. GSPO uses $\phi(W) = W^{1/T}$, which induces a proposal distribution that explicitly depends on the sequence length T:

$$Q_{\text{GSPO}}(\tau) \propto \mu(\tau)^{1-1/T} \cdot \pi(\tau)^{1/T}. \tag{46}$$

As $T \to \infty$, $Q_{\text{GSPO}} \to \mu$: the normalized weight converges to $\exp(\mathbb{E}[\log \rho_t])$, a constant independent of the specific trajectory. This causes two fundamental problems:

1. **Signal dissipation**: All weights collapse toward a constant, losing discriminative power.
2. **Conflation of distinct sequences**: Sequences with identical per-token statistics but different lengths receive identical weights, despite having different true importance weights.

Proposition H.1 (Conflation under Length Normalization)**.** *For any two sequences τ_1, τ_2 with lengths $T_1 \neq T_2$, if $\frac{\log W_1}{T_1} = \frac{\log W_2}{T_2} = \bar{r}$ with $\bar{r} \neq 0$, then GSPO assigns identical weights: $\phi_{GSPO}(W_1) = \phi_{GSPO}(W_2)$. However, their true importance weights differ: $W_1 = e^{T_1 \bar{r}} \neq e^{T_2 \bar{r}} = W_2$.*

Proof. By Equation (8), $\phi_{\text{GSPO}}(W_i) = W_i^{1/T_i} = \exp(\log W_i/T_i) = e^{\bar{r}}$ for $i = 1, 2$, so the two weights coincide. Since $\log W_i = T_i \bar{r}$, $T_1 \neq T_2$ and $\bar{r} \neq 0$ give $\log W_1 \neq \log W_2$, i.e., $W_1 \neq W_2$. □

This conflation is problematic: a short sequence that is moderately off-policy and a long sequence that is severely off-policy may receive identical gradient weights, even though their contributions to the policy gradient should differ substantially.

VESPO Avoids Length-Dependent Bias. VESPO uses $\phi(W) = W^{c_1} \exp(c_2(1-W))$ without any length normalization. The reshaping function depends only on the sequence-level importance weight W, not on the sequence length T. Two sequences with the same average per-token log-ratio $\bar{r}$ but different lengths receive different weights: the longer sequence has $W = e^{T\bar{r}}$, which is appropriately transformed by ϕ. This preserves the discriminative power of the importance weight while the soft-shaping kernel controls variance through exponential suppression of extreme weights.

I Additional Analysis of Baseline Failure Modes

This section provides supplementary visualizations comparing N=4 vs N=8 for each baseline (the comprehensive training dynamics figure across all N values is in the main text, Figure 4).

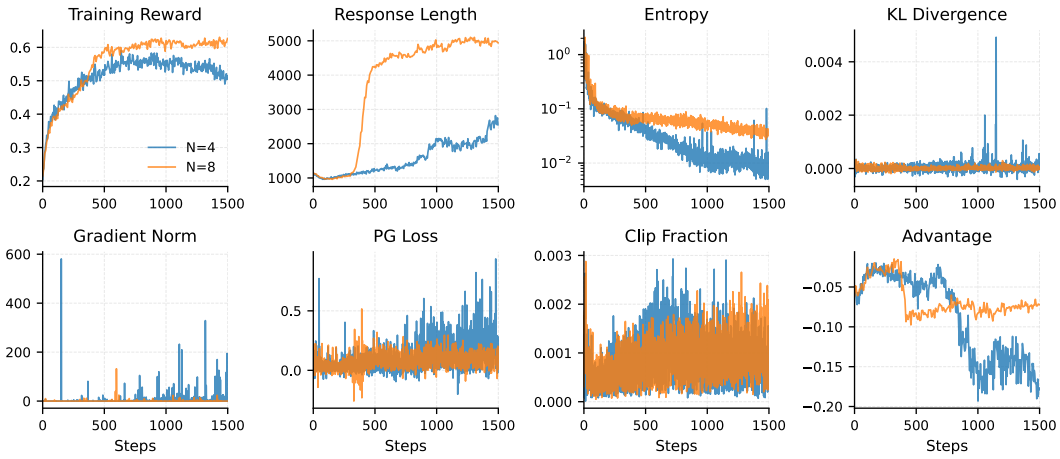


Figure 8: GRPO: $N = 4$ (blue) vs $N = 8$ (orange). Entropy decreases more rapidly at $N = 4$, limiting exploration.

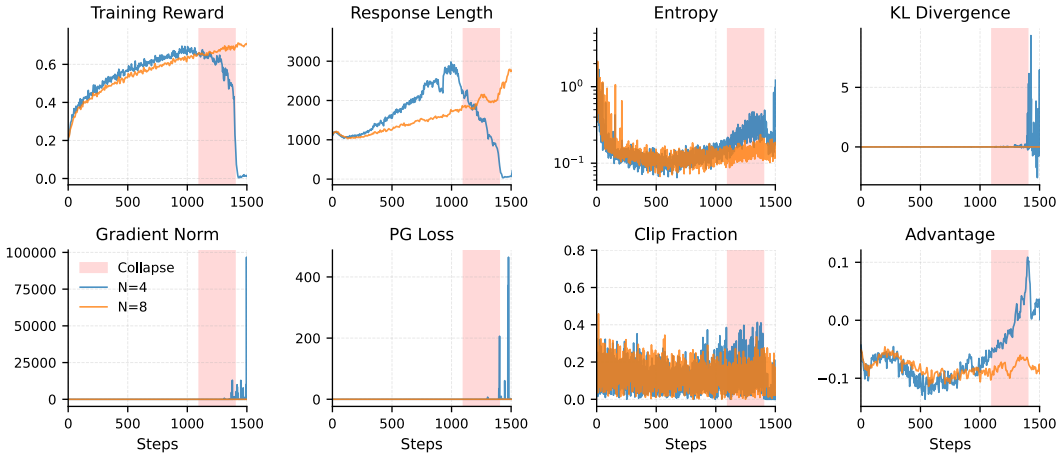


Figure 9: GSPO: $N = 4$ (blue) vs $N = 8$ (orange). Response length spikes above 3,000 tokens at $N = 4$ before collapse.

J Training Dynamics for Recent Baseline Comparison

Figure 11 shows the training dynamics of TOPR, CISPO, BAPO, and VESPO under the matched setup of Section 4.4. CISPO collapses around step 280 due to insufficient sequence-level variance control; BAPO exhibits entropy explosion after step 1,100 as adaptive clip bounds over-relax; TOPR converges slowly with suppressed response length. VESPO is the only method that maintains stable training across all metrics.

K Code Generation Training Details and Dynamics

Setup recap. We train Qwen3-30B-A3B-Base on PRIME-RL/Eurus-2-RL-Data [5] using the same VESPO hyperparameters $(c_1, c_2) = (2, 3)/(3, 2)$ as math, without any retuning. The reward is execution-based (test case pass/fail), structurally distinct from the verifier-based math reward. We evaluate on HumanEval+, MBPP+ [24], and LiveCodeBench v6 [19], reporting pass@10 (Table 4). VESPO has the highest reported point estimate on all three benchmarks (88.4, 75.4, 25.3), averaging 63.0 pass@10 versus 61.8 for SAPO. This supports reuse of the fixed setting across the evaluated verifier-based math and execution-based code settings; preference-based rewards are not evaluated here.

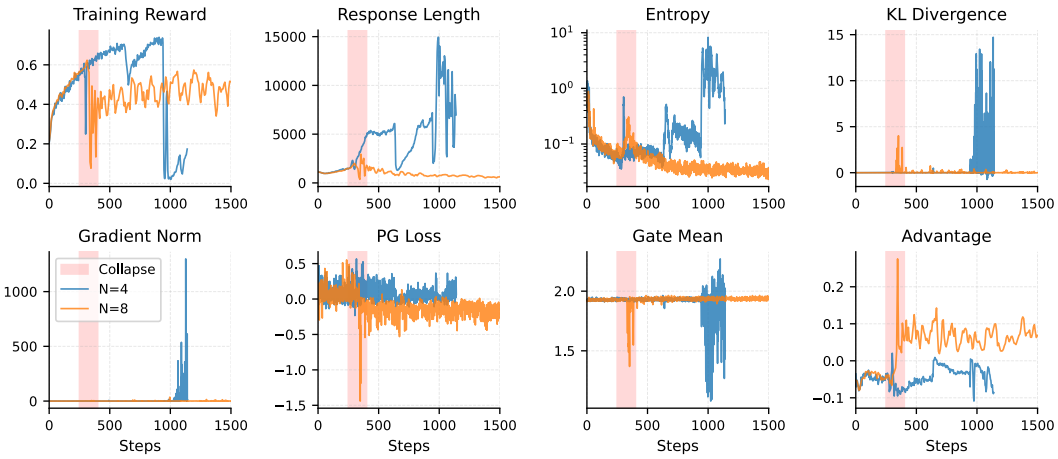


Figure 10: SAPO: $N = 4$ (blue) vs $N = 8$ (orange). Training reward collapses at $N = 8$ due to insufficient suppression for negative advantages.

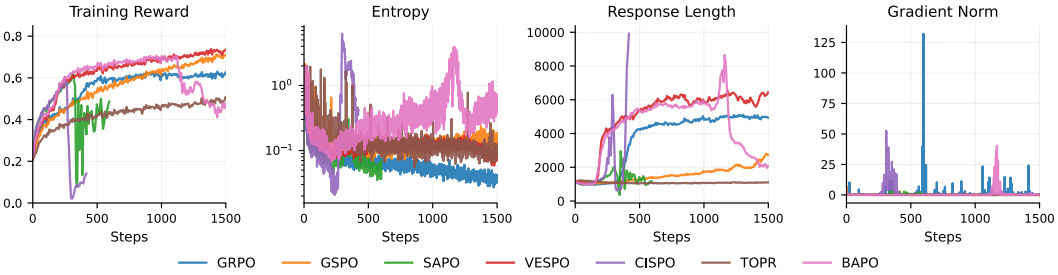


Figure 11: Training dynamics of recent importance weight reshaping methods on Qwen3-30B-A3B-Base (N=8). CISPO collapses early; BAPO experiences entropy explosion; TOPR converges slowly with suppressed length. VESPO is the only method maintaining stable training across all metrics.

Training dynamics. Figure 12 shows the training dynamics on PRIME-RL/Eurus-2-RL-Data. VESPO maintains the highest training reward with stable entropy and response length throughout 1,500 steps. SAPO trains well initially but collapses around step 1,250. GRPO and GSPO converge slowly to lower reward levels.

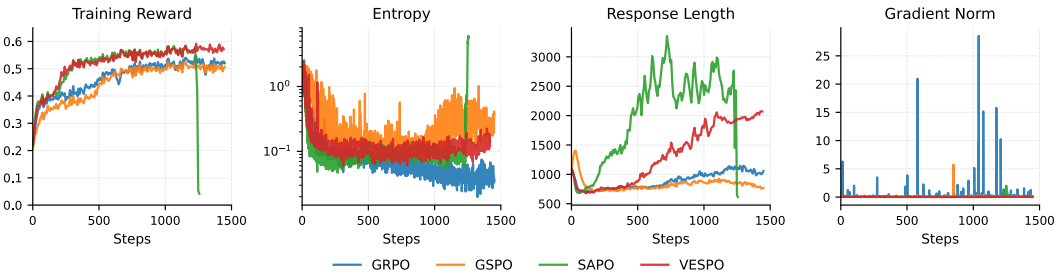


Figure 12: Training dynamics on code generation (PRIME-RL/Eurus-2-RL-Data, Qwen3-30B-A3B-Base, N=8). VESPO is the only method maintaining stable training; SAPO collapses around step 1,250.

L Algorithm Pseudocode

```python
def compute_policy_loss_vespo(log_pi, log_mu, advantages, mask, c_pos, c_neg):
    """
    Args:
        log_pi: log probs from current policy, shape (batch, seq_len)
        log_mu: log probs from behavior policy, shape (batch, seq_len)
        advantages: sequence-level advantages, shape (batch,)
        mask: response mask, shape (batch, seq_len)
        c_pos: (c1, c2) for positive advantages
        c_neg: (c1, c2) for negative advantages
    """
    # Sequence-level IS ratio in log-space (true product, no length norm)
    log_ratio = log_pi - log_mu
    seq_log_w = (log_ratio * mask).sum(dim=-1)  # (batch,)
    W = exp(seq_log_w)

    # Asymmetric hyperparameter selection
    c1 = where(advantages >= 0, c_pos[0], c_neg[0])
    c2 = where(advantages >= 0, c_pos[1], c_neg[1])

    # VESPO kernel in log-space, normalized so phi(1) = 1
    log_phi = c2 + c1 * seq_log_w - c2 * W
    phi = exp(log_phi).detach()  # gradient scaling only

    # Policy gradient loss: grad = -phi * A * grad_log_pi
    loss = -phi.unsqueeze(-1) * advantages.unsqueeze(-1) * log_pi
    return aggregate(loss, mask)
```

VESPO Policy Loss

M Limitations and Future Directions

Hyperparameter specification. While the variational framework provides the kernel form $\phi(W) = W^{c_1}\exp(c_2(1-W))$, theory does not determine a unique (c_1, c_2). It identifies $c_1 \geq 1$ as the regime in which K is finite and Proposition 3.1 is non-vacuous. The implementation fixes branch-specific coefficients and does not enforce a common C (Section 3.4). We selected simple integers in this regime, with stronger suppression for $A < 0$ samples with $W < 1$ as motivated by the failure-mode analysis, and used $(c_1^+, c_2^+) = (2, 3)$ and $(c_1^-, c_2^-) = (3, 2)$ in every experiment without retuning. A local one-at-a-time grid gives averages of 60.0–61.0 (Table 7). For a new domain, these considerations ($c_1 \geq 1$; stronger suppression for negative advantages with $W < 1$) are the guidance the analysis provides; their adequacy for preference-based rewards is untested. Adaptive schemes—scheduling (c_1, c_2) along training, automatic per-task search, or learning the hyperparameters from data—could further improve performance and remain an open direction.

Reward modality scope. Our experiments cover *verifiable-reward* settings: math reasoning with verifier-based reward (Math-Verify) and code generation with execution-based reward (test pass/fail). The fixed coefficients have been evaluated only with verifier-based math and execution-based code rewards; preference-based rewards (e.g., RLHF, RLAIF) are not evaluated. Effectiveness on such rewards, shaped/dense rewards, or open-ended creative generation remains to be explored. We expect VESPO's smooth variance control to remain beneficial in these regimes since the underlying off-policy IS challenge is the same, but direct empirical validation is needed.

Model scale. Our largest evaluated model is Qwen3-30B-A3B-Base (30B total parameters with MoE routing). Scaling to substantially larger models (>100B) is not directly evaluated due to compute constraints. Gains are largest on the MoE Qwen3-30B-A3B-Base, but model size, model family, and MoE routing are confounded across our three models, so we draw no scaling conclusion; verification at larger scales remains future work.

Separating mismatch sources. In the MoE runs, policy staleness and train-inference mismatch are both present; because a matched same-checkpoint trainer–vLLM measurement was not recorded, the current experiments do not separately quantify the train-inference component (Appendix G).

Statistical uncertainty. Multi-seed estimates are reported for the GRPO/VESPO comparison on Qwen3-8B-Base (Table 6); the other reported numbers are single-run point estimates (for Tables 1 to 4, the best checkpoint per method selected by average accuracy; Section 4.1). The math metrics (avg@k) average over evaluation samples and the code metric is pass@10; we do not report significance tests.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction state our contributions: the constrained variational formulation over an implicit proposal Q with the sequence-level ratio W as the tilting statistic; the resulting smooth kernel and its certificate under $\mathbb{E}_Q[W] \leq C$ (Section 3); the sequence-level implementation without length normalization (Section 3.5); and, with one fixed coefficient setting, gains on verifier-based math and execution-based code and stability under staleness up to $64\times$ on the MoE model in math (Section 4). The relation to classical work is stated in Section 5 and Appendix B.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Section 6 states the limitations, with details in Appendix M: (i) *hyperparameter specification* — theory identifies $c_1 \geq 1$ but not a unique (c_1, c_2), so we use one fixed setting outside the coefficient ablations and report a local sensitivity grid (Table 7); (ii) *reward modality* — verifier-based math and execution-based code only, preference-based rewards not evaluated; (iii) *model scale* — up to a 30B-parameter MoE; (iv) the train-inference component of the MoE off-policy discrepancy is not separately quantified; (v) multi-seed estimates cover only the GRPO/VESPO comparison on Qwen3-8B-Base.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.

- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: All theoretical results are stated as numbered Propositions (Propositions 3.1, C.1, D.1 and H.1) with full proofs in Appendices C, D and H. Assumptions are explicitly stated in each statement. The variance certificate (Proposition 3.1) holds under the stated moment constraint, which the implementation does not enforce (Section 3.4).

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Section 4.1 and Appendix E document the training framework (veRL, vLLM, FSDP/Megatron), datasets (DAPO-Math, PRIME-RL/Eurus-2-RL-Data), models, hyperparameters (including the published baseline configurations and the fixed VESPO coefficients), and evaluation protocol. Appendix L provides full pseudocode for the VESPO loss, and the implementation is released at `https://github.com/FloyedShen/VESPO`.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example

(a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
(b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: All datasets used (DAPO-Math, PRIME-RL/Eurus-2-RL-Data, AIME, AMC, MATH-500, HumanEval+, MBPP+, LiveCodeBench v6) are publicly available; all except the AIME and AMC competition problem sets are cited. The VESPO implementation (policy loss, training scripts and data preprocessing, with README instructions) is released at `https://github.com/FloyedShen/VESPO`; the loss is also given as pseudocode in Appendix L.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 4.1 and Appendix E specify the training details (1,500 gradient steps, 8 responses per query, max length 16k, staleness simulation), the published baseline configurations used for Table 1 (Appendix E), and the evaluation protocol (avg@k for math, pass@10 for code, best-checkpoint selection by average accuracy). VESPO uses the fixed $(c_1, c_2) = (2, 3)$ for $A > 0$ and $(3, 2)$ for $A < 0$, simple integers in the $c_1 \geq 1$ regime (Section 3.5); a local sensitivity grid is in Table 7.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Table 6 (Appendix F, referenced in Section 4.2) reports mean $\pm$ standard deviation across three training seeds for GRPO and VESPO on Qwen3-8B-Base at $N{=}8$. The variability captured is the training seed, and the final column is the macro-average of the benchmark means. Other tables report single-run point estimates (for Tables 1 to 4, the best checkpoint per method selected by average accuracy; Section 4.1); in particular, the Qwen3-30B-A3B-Base (MoE) results that support the staleness and MoE claims are single-run. Math avg@k metrics ($k \in \{4, 16, 32\}$) average over evaluation sampling; code uses pass@10. Each training run takes $\sim$48 hours on 32 H20 GPUs. We do not report significance tests.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Section 4.1 and Appendix E state that all experiments are conducted on 32 NVIDIA H20 GPUs using veRL with vLLM 0.11.0 for inference and FSDP/Megatron for training, with a budget of 1,500 gradient steps per run; each training run takes $\sim$48 hours on 32 H20 GPUs.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.

- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have reviewed and the work conforms to the NeurIPS Code of Ethics. The research uses publicly available datasets and pretrained models, involves no human subjects, and presents an algorithmic improvement to RL training stability.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: This is foundational research on the stability of RL training algorithms for LLMs. Improving training stability does not introduce direct societal impacts beyond those inherent to advancing LLM capabilities in general; we do not foresee specific positive or negative impacts that warrant separate discussion.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: We do not release any new models, datasets, or scraped data. The paper describes a training algorithm and reports results on publicly available benchmarks.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: All datasets (DAPO-Math, PRIME-RL/Eurus-2-RL-Data, MATH-500, HumanEval+, MBPP+, LiveCodeBench), models (Llama-3.2-3B-Instruct, Qwen3-8B-Base, Qwen3-30B-A3B-Base), and frameworks (veRL, vLLM, FSDP, Megatron, Math-Verify) are properly cited (AIME and AMC are public competition problem sets); we use them under their respective open-source licenses.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: We release the VESPO code (built on veRL), documented in the repository README and distributed under the Apache-2.0 license (per the repository LICENSE). The algorithm is documented in Section 3.5 and pseudocode is in Appendix L. We do not release new datasets or model checkpoints.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.

- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve human subjects research.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs are the subject of training in our experiments, but no LLM is used as an original or non-standard component of our method development; the variational derivation and algorithm are derived without LLM assistance.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.