RELATIVE SCORE POLICY OPTIMIZATION FOR DIFFUSION LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Diffusion large language models (dLLMs) offer a promising route to parallel and efficient text generation, but improving their reasoning ability requires effective post-training. Reinforcement learning with verifiable rewards (RLVR) is a natural choice for this purpose, yet its application to dLLMs is hindered by the absence of tractable sequence-level log-ratios, which are central to standard policy optimization. The lack of tractable sequence-level log-ratios forces existing methods to rely on high-variance ELBO-based approximations, where high verifier rewards can amplify inaccurate score estimates and destabilize RL training. To overcome this issue, we propose **R**elative **S**core **P**olicy **O**ptimization (RSPO), a simple RLVR method that uses verifiable rewards to calibrate noisy likelihood estimates in dLLMs. The core of our algorithm relies on a key observation: a reward advantage can be interpreted not only as an update direction, but also as a target for the relative log-ratio between the current and reference policies. Accordingly, RSPO calibrates this noisy relative log-ratio estimate by comparing its reward advantage with the reward-implied target relative log-ratio, updating the policy according to the gap between the current estimate and the target rather than the raw advantage alone. Experiments on mathematical reasoning and planning benchmarks show that RSPO yields especially strong gains on planning tasks and competitive mathematical-reasoning performance.

1 INTRODUCTION

Reinforcement learning with verifiable rewards (RLVR) (He et al., 2025; Shao et al., 2024; Yu et al., 2025; Zheng et al., 2025a) has become an effective post-training paradigm for improving language models on reasoning tasks: sample candidate answers, verify their correctness, and update the model so that correct answers become more likely (Sutton et al., 1998; Ouyang et al., 2022; Jaech et al., 2024; Guo et al., 2025). For autoregressive language models, this update is naturally expressed through policy log-ratios, namely the change in log-likelihood of a sampled response under the current policy relative to a reference or previous policy. Because autoregressive models admit an explicit left-to-right factorization, these log-ratios are tractable, enabling methods such as Proximal Policy Optimization (PPO) (Schulman et al., 2017) and Group Relative Policy Optimization (GRPO) (Shao et al., 2024) to combine reward improvement with Kullback–Leibler (KL) control.

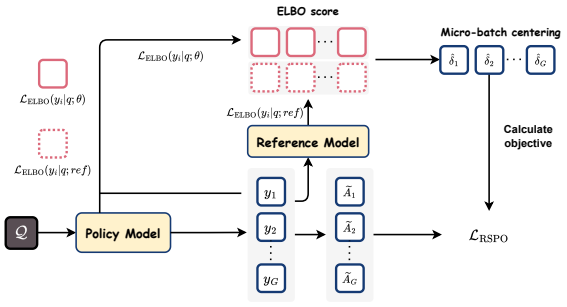


Figure 1: Overview of RSPO: verifier rewards define centered relative-score targets, and the residual to the current ELBO-based score provides policy-update feedback.

Diffusion large language models (dLLMs) (Sahoo et al., 2024; Shi et al., 2024; Nie et al.; Yang et al., 2025; Bie et al., 2025) break this convenient interface. Their generation process proceeds through iterative denoising rather than left-to-right prediction, so the sequence likelihood of a completed response is not directly available. As a result, the sequence-level policy log-ratios used by standard RL objectives become difficult to compute. Recent RL methods for dLLMs therefore replace exact

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

log-ratios with tractable surrogates, such as one-step mean-field estimates (Zhao et al., 2025), ELBO-based token or sequence scores (Ou et al., 2025), trajectory-level formulations (Wang et al., 2025b), and evidence-bound objectives (Wang et al., 2025a; Lin et al., 2025). These substitutes make policy optimization possible, but they also introduce a new source of uncertainty: in RLVR, the verifier reward is a reliable task signal, whereas the model-side relative score used to apply that reward is only an approximate and noisy estimate of the true likelihood change.

This mismatch raises a basic question: how should dLLM RLVR combine a reliable verifier reward with a noisy likelihood-ratio surrogate? Standard advantage-weighted objectives address only part of this question. They multiply a model-side score by a reward or advantage coefficient, thereby specifying which responses should be encouraged. When the score is an ELBO-based log-ratio surrogate, however, the coefficient does not depend on the current relative score. A high-advantage response can therefore continue to amplify a noisy likelihood estimate even after its estimated relative likelihood has moved far from the reference. Thus, the reward provides a preference direction, but the objective lacks a calibrated target for the relative score itself.

Our key observation is that KL-regularized policy improvement provides exactly this target. The optimal KL-regularized policy assigns each response a current-reference log-ratio determined by its reward, up to a prompt-level normalization term. Within a comparison group, this normalization term is shared and disappears after centering. The centered verifier rewards therefore specify the relative likelihood changes that the improved policy should assign to sampled responses. Based on this principle, we propose **R**elative **S**core **P**olicy **O**ptimization (RSPO), a simple RLVR objective for diffusion language models. RSPO estimates a centered current-reference score with ELBO differences, derives a centered relative-score target from verifier rewards, and uses the residual gap as feedback in the policy update. In this way, reliable rewards calibrate noisy likelihood-ratio estimates instead of serving only as fixed multiplicative weights. Figure 1 gives an overview of this feedback loop: the policy generates a group of responses for each prompt, the reference model is used to form ELBO-based log-ratio scores, and micro-batch centering turns these scores into relative feedback. RSPO compares this current centered score with the verifier-induced target and optimizes the resulting residual objective; details are deferred to Section 3. Experiments on mathematical reasoning and planning benchmarks show that RSPO yields large planning gains and competitive mathematical-reasoning performance against existing RL baselines.

Our contributions are summarized as follows:

- We identify a calibration problem in dLLM RLVR. Verifier rewards provide reliable task-level feedback, but the ELBO-based log-ratios used for policy optimization are noisy likelihood-ratio surrogates. Standard advantage-weighted objectives use these rewards as fixed multiplicative coefficients, which can keep pushing already high-scoring responses and over-amplify noisy relative-score estimates.
- We derive a principled relative-score target from KL-regularized policy improvement. Although the optimal policy log-ratio contains an unknown normalization term, this term is shared within a comparison group and can be removed by centering. The resulting centered form shows that verifier rewards specify not only which responses should be preferred, but also the relative likelihood changes an improved policy should assign to them.
- We propose RSPO, a simple stop-gradient objective for diffusion language models. Instead of using rewards only as fixed advantages, RSPO compares the current centered ELBO-based relative score with the reward-implied target and updates according to the remaining gap. We further provide a local first-order analysis showing that the RSPO update admits a finite centered-score target and is first-order equivalent to a matched quadratic target objective.
- We validate RSPO on mathematical reasoning and planning benchmarks. Across these settings, RSPO improves diffusion language models over existing dLLM RL baselines, supporting our key observations and demonstrating the superior performance of the proposed algorithm.

2 PRELIMINARIES

Masked Diffusion Language Models. Masked diffusion models (MDMs) (Sahoo et al., 2024) generate sequences by reversing a gradual masking process. Given a prompt $\boldsymbol{q}$ and a clean response $\boldsymbol{y} = (y^1, \ldots, y^L)$, the forward process corrupts $\boldsymbol{y}$ into $\boldsymbol{z}_t = (z_t^1, \ldots, z_t^L)$ at timestep $t \in [0, 1]$, while keeping $\boldsymbol{q}$ fixed. Each token is independently replaced by the mask token $[\text{MASK}]$, whose one-hot

representation is denoted by $\boldsymbol{m}$: $q_{t|0}(z_t^i \mid y^i) = \mathrm{Cat}\left(z_t^i;\, \alpha_t y^i + (1-\alpha_t)\boldsymbol{m}\right)$, where Cat denotes a categorical distribution and α_t is a decreasing noise schedule with $\alpha_0 = 1$ and $\alpha_1 = 0$. Thus $\boldsymbol{z}_0 = \boldsymbol{y}$ and $\boldsymbol{z}_1$ is fully masked; for the linear schedule $\alpha_t = 1 - t$, each token is masked with probability t. The reverse process is parameterized by a denoising model π_θ. For $0 \le s < t \le 1$, the transition from $\boldsymbol{z}_t$ to $\boldsymbol{z}_s$ keeps unmasked tokens fixed and predicts masked tokens from the corrupted context:

$$p_\theta(z_s^i \mid \boldsymbol{z}_t, \boldsymbol{q}) = \begin{cases} \mathrm{Cat}(z_s^i;\, z_t^i), & z_t^i \neq \boldsymbol{m}, \\ \mathrm{Cat}\left(z_s^i;\, \dfrac{(1-\alpha_s)\boldsymbol{m} + (\alpha_s - \alpha_t)\pi_\theta(\cdot \mid \boldsymbol{z}_t, \boldsymbol{q})}{1-\alpha_t}\right), & z_t^i = \boldsymbol{m}. \end{cases}$$

Generation starts from a fully masked response and iteratively denoises masked positions, often in parallel. Since MDMs do not use an autoregressive factorization, their likelihood is commonly optimized through an evidence lower bound (ELBO) (Ou et al.; Shi et al., 2024):

$$\mathcal{L}_{\mathrm{ELBO}}(\boldsymbol{y} \mid \boldsymbol{q}; \theta) = \mathbb{E}_{t\sim\mathcal{U}[0,1],\, \boldsymbol{z}_t\sim q_{t|0}(\cdot|\boldsymbol{y})}\left[\sum_{i=1}^{L} w(t)\, \mathbf{1}[z_t^i = \boldsymbol{m}]\, \log \pi_\theta(y^i \mid \boldsymbol{z}_t, \boldsymbol{q})\right] \le \log \pi_\theta(\boldsymbol{y} \mid \boldsymbol{q}).$$

Here the model is trained to recover clean tokens only at masked positions. The time-dependent weight $w(t) = \frac{\alpha_t'}{\alpha_t - 1}$ is determined by the noise schedule, which reduces to $w(t) = 1/t$ under the linear schedule $\alpha_t = 1 - t$.

Reinforcement Learning with Verifiable Rewards. In reinforcement learning with verifiable rewards (RLVR), a language model is treated as a policy π_θ that generates a response $\boldsymbol{y}$ conditioned on a prompt $\boldsymbol{q}$. A verifier assigns a scalar reward $r(\boldsymbol{y}, \boldsymbol{q})$ to the completed response, and the goal is to improve the policy toward high-reward outputs while controlling its deviation from a reference policy π_{ref}. The standard reward-maximization objective is $J(\theta) = \mathbb{E}_{\boldsymbol{y}\sim\pi_\theta(\cdot|\boldsymbol{q})}\left[r(\boldsymbol{y}, \boldsymbol{q})\right]$, whose policy-gradient form is

$$\nabla_\theta J(\theta) = \mathbb{E}_{\boldsymbol{y}\sim\pi_\theta(\cdot|\boldsymbol{q})}\left[r(\boldsymbol{y}, \boldsymbol{q})\, \nabla_\theta \log \pi_\theta(\boldsymbol{y} \mid \boldsymbol{q})\right].$$

For autoregressive language models, the sequence log-likelihood $\log \pi_\theta(\boldsymbol{y} \mid \boldsymbol{q})$ can be computed by summing token-level log-probabilities. However, this is no longer tractable for dLLMs, whose generation marginalizes over iterative denoising paths rather than following a fixed left-to-right factorization. Consequently, recent dLLM RL methods use an ELBO-based score derived from the denoising objective as a practical substitute (Zhao et al., 2025; Ou et al., 2025; Wang et al., 2025b;a; Lin et al., 2025) $\log \pi_\theta(\boldsymbol{y} \mid \boldsymbol{q}) \approx \mathcal{L}_{\mathrm{ELBO}}(\boldsymbol{y} \mid \boldsymbol{q}; \theta)$. Here the ELBO score is tractable because it only requires evaluating denoising probabilities at sampled timesteps and masked positions, rather than summing over all possible denoising trajectories. **The Intractability Challenge.** This substitution changes what the RL objective can reliably measure. The verifier reward remains a direct task-level signal: it tells us whether a completed response is correct or useful. By contrast, the ELBO-based score is only a tractable proxy for the sequence log-likelihood. It is a lower-bound-style estimate, and in practice it is evaluated through sampled timesteps, sampled corruptions, and token-level denoising predictions. Thus, the reward signal and the model-side score have different reliability: the former can be exact for verifiable tasks, while the latter is approximate and can be noisy.

This distinction matters when the ELBO score is used inside a reward-weighted policy objective. A high reward can increase the weight placed on a sampled response, but the objective does not know whether the corresponding ELBO-based relative score is still too small, already sufficiently large, or large mainly because of estimation noise. As a result, reward weighting provides a direction of preference but not a calibrated target for the relative score itself. The goal of this work is to develop an RLVR objective for dLLMs that uses reliable verifier rewards to calibrate ELBO-based relative scores, rather than merely using rewards as fixed weights on approximate likelihood surrogates.

3 Method

3.1 Motivation

Section 2 motivates the following objective-design issue: when dLLM RLVR relies on an ELBO-based likelihood surrogate, the surrogate requires a well-defined optimization target. We derive this target from the standard KL-regularized policy-improvement problem. In particular, for a fixed

prompt $\boldsymbol{q}$ and reference policy π_{ref}, the solution of the standard KL-regularized policy-improvement problem is

$$\pi^\star(\boldsymbol{y} \mid \boldsymbol{q}) = Z(\boldsymbol{q})^{-1}\pi_{\text{ref}}(\boldsymbol{y} \mid \boldsymbol{q}) \exp(r(\boldsymbol{y}, \boldsymbol{q})/\beta)\,.$$

where $Z(\boldsymbol{q})$ is the prompt-dependent normalizer. Therefore, the ideal current-reference log-ratio is $\Delta^\star(\boldsymbol{y}, \boldsymbol{q}) := r(\boldsymbol{y}, \boldsymbol{q})/\beta - \log Z(\boldsymbol{q})$. For a group $\{\boldsymbol{y}_i\}_{i=1}^G$ sampled from the same prompt, let $r_i = r(\boldsymbol{y}_i, \boldsymbol{q})$ and $\Delta_i^\star = \Delta^\star(\boldsymbol{y}_i, \boldsymbol{q})$. Then define $\bar{\Delta}^\star$ by the mean of $\Delta_i^\star$ over all i's, then the normalization factor $Z(\boldsymbol{q})$ eliminates under centering:

$$\Delta_i^\star - \bar{\Delta}^\star = \frac{1}{\beta}(r_i - \bar{r}). \tag{1}$$

In practice, we use a group-relative advantage $\widetilde{A}_i$ as the normalized version of the centered reward signal. The scaling by $1/\beta$ and any group-shared reward normalization are absorbed into $\widetilde{A}_i$, with $\sum_{i=1}^G \widetilde{A}_i = 0$ for a complete prompt group.[1] Moreover, we introduce a fixed feedback coefficient $\lambda > 0$ and use $\tau_i := \widetilde{A}_i/\lambda$ as the reward-implied target for the centered relative score. The coefficient λ controls the scale of the target: larger λ makes the target more conservative, while smaller λ asks the model to move farther from the reference for the same group-relative advantage.

It remains to define a tractable relative score for dLLMs. Since exact sequence likelihoods are unavailable, RSPO uses current-reference differences of ELBO scores. Let $\mathcal{E}_i^\theta = \mathcal{L}_{\text{ELBO}}(\boldsymbol{y}_i \mid \boldsymbol{q}; \theta)$ and $\mathcal{E}_i^{\text{ref}} = \mathcal{L}_{\text{ELBO}}(\boldsymbol{y}_i \mid \boldsymbol{q}; \text{ref})$. For completion length L_c,

$$\delta_i(\theta) = \frac{\mathcal{E}_i^\theta - \mathcal{E}_i^{\text{ref}}}{L_c}. \tag{2}$$

Positive $\delta_i(\theta)$ means that the current model assigns a higher ELBO score to response $\boldsymbol{y}_i$ than the reference model. If an implementation stores cross-entropy or negative ELBO rather than ELBO, the current-reference difference must be reversed.

Remark 1 (Sign convention)**.** *Eq. equation 2 must increase with completion likelihood; otherwise high-reward samples receive the wrong update direction.*

We then center this score within the same prompt group. For a group $\mathcal{B}$ of size G, define $\bar{\delta}_{\mathcal{B}}(\theta) = N^{-1}\sum_{j\in\mathcal{B}} \delta_j(\theta)$ and $\widehat{\delta}_i(\theta) = \delta_i(\theta) - \text{sg}(\bar{\delta}_{\mathcal{B}}(\theta))$, where $\text{sg}(\cdot)$ denotes detachment. Then, The centered score $\widehat{\delta}_i(\theta)$ is RSPO's tractable counterpart of the centered log-ratio in Eq. equation 1. When the micro-batch advantages sum to zero, the verifier supplies the target $\widehat{\delta}_i(\theta) \to \widetilde{A}_i/\lambda$, so the update should follow the residual error between the current relative score and this reward-implied target.

3.2 Relative Score Policy Optimization

The name *Relative Score Policy Optimization* reflects the two ingredients above. First, the quantity being controlled is the centered current-reference relative score $\widehat{\delta}_i(\theta)$. Second, the objective keeps the policy-optimization form: a differentiable model-side score is multiplied by a scalar feedback coefficient.

To see the difference from standard advantage weighting, consider a loss micro-batch $\mathcal{B}$ with $N = |\mathcal{B}|$. The advantage-weighted (AW) surrogate uses the group-relative advantage as a fixed coefficient, $\ell_{\text{AW}}(\theta; \mathcal{B}) = -N^{-1}\sum_{i\in\mathcal{B}} \widetilde{A}_i\, \widehat{\delta}_i(\theta)$. This objective always pushes a positive-advantage response upward, regardless of its current centered relative score. It therefore encodes a preference direction, but not a stopping rule tied to the reward-implied target.

RSPO replaces this fixed coefficient with the remaining calibration gap. Recall that the target relative score is $\widetilde{A}_i/\lambda$, or equivalently that the scaled residual is $e_i(\theta) = \widetilde{A}_i - \lambda\widehat{\delta}_i(\theta)$. RSPO uses a detached version of this residual as the feedback coefficient, $w_i = \widetilde{A}_i - \lambda\ \text{sg}(\widehat{\delta}_i(\theta))$, and defines

$$\ell_{\text{RSPO}}(\theta; \mathcal{B}) = -\frac{1}{N}\sum_{i\in\mathcal{B}} w_i\, \widehat{\delta}_i(\theta). \tag{3}$$

[1]The analysis only requires the advantages in a complete prompt group to sum to zero; any group-shared reward scaling preserves this property.

This coefficient adapts to the current relative score. If $\widehat{\delta}_i(\theta) < \widetilde{A}_i/\lambda$, then $w_i > 0$ and the update increases the score. If $\widehat{\delta}_i(\theta) > \widetilde{A}_i/\lambda$, then $w_i < 0$ and the update decreases the score. Near the target, the coefficient becomes small, so the update naturally weakens. Thus RSPO keeps the familiar policy-gradient-style objective while turning verifier rewards into calibrated feedback toward a finite relative-score target. The exact first-order gradient of Eq. equation 3 is analyzed in Section 4.

Remark 2. *A direct squared calibration loss between $\widehat{\delta}_i(\theta)$ and $\widetilde{A}_i/\lambda$ would induce the same local first-order coefficient. RSPO deliberately keeps a policy-optimization form instead: the calibration gap is detached and used as a scalar coefficient, while differentiation passes through a single current relative-score factor.*

3.3 Training Procedure

Operationally, RSPO can be implemented with the same outer loop as standard group-relative RLVR, as illustrated in Figure 1. For each prompt, we sample a group of completions, evaluate them with a verifier, and convert the resulting rewards into group-relative advantages. We then estimate the current-reference ELBO difference for each completion, center these relative scores within the loss micro-batch, and minimize the RSPO loss in Eq. equation 3. When computing the current and reference ELBO scores for the same completion, we use the same diffusion masks for both models to reduce Monte Carlo variance in their difference. The reference model is kept fixed throughout training, and λ is treated as a fixed feedback-scale hyperparameter. Full score-estimation and algorithmic details are deferred to Appendix B.

4 Theory

The previous section introduced RSPO as a simple change to the usual group-relative RLVR update: keep the policy-score form, but replace the fixed advantage coefficient with a residual to a reward-implied relative-score target. This section explains what that change means mathematically by analyzing the local update RSPO applies in one backward pass. The point is not to prove global convergence for dLLMs, and it is not to assume that we can recover the exact sequence likelihood that Section 2 identified as intractable. Instead, we fix the sampled micro-batch: completions, verifier rewards, group-relative advantages, reference ELBO scores, diffusion masks, and the feedback scale λ are all treated as constants. The batch center is detached, as in Eq. equation 3, and the only differentiated quantity is the current-model score through $\delta_i(\theta)$. Under this local view, the theory answers three questions that mirror the design choices in Section 3: what centering removes, what coefficient multiplies the current score derivative, and what finite target the update is trying to reach. Estimator-level details and proofs are deferred to Appendices D and E.

4.1 Relative-Score Feedback and Fixed Point

We first make precise why RSPO is a *relative-score* method. The ELBO values used by dLLMs are only tractable surrogates for sequence log-likelihoods, and their absolute level can contain shifts shared by many samples in a batch. Such shared shifts should not determine which completion is preferred within a prompt group. Detached centering removes them in the forward pass, while preserving the per-sample derivative of the current score: $\sum_{i\in\mathcal{B}} \widehat{\delta}_i = 0$, and $\nabla_\theta \widehat{\delta}_i = \nabla_\theta \delta_i(\theta)$ because the center is not differentiated.

Lemma 1 (Forward centering and zero-sum weights)**.** *For a fixed sampled micro-batch, $\sum_{i\in\mathcal{B}} \widehat{\delta}_i = 0$ and $\sum_{i\in\mathcal{B}} w_i = \sum_{i\in\mathcal{B}} \widetilde{A}_i$. If $\mathcal{B}$ is a union of complete prompt groups and $\widetilde{A}_i$ is a zero-sum group-relative advantage, optionally divided by a group-shared reward scale, then the RSPO weights are also zero-sum.*

The lemma is a useful consistency check. When the advantages are group-relative, the feedback term does not introduce a new batch-level push. It continues to compare responses within the same prompt group, which is exactly the comparison structure used to derive the centered target in Section 3.

We next look at the actual gradient of the RSPO loss. Since the residual coefficient is detached, differentiation passes through only one copy of the current relative score.

Proposition 1 (RSPO relative-score feedback gradient)**.** *For fixed λ and a fixed sampled micro-batch,*

$$\nabla_\theta \ell_{\mathrm{RSPO}}(\theta; \mathcal{B}) = -\frac{1}{N} \sum_{i \in \mathcal{B}} (\widetilde{A}_i - \lambda \widehat{\delta}_i) \nabla_\theta \delta_i(\theta). \tag{4}$$

The policy-gradient coefficient is the relative-score feedback error, rather than the advantage alone.

This formula is the core local interpretation of RSPO. The usual advantage-weighted update, as used in policy-gradient-style RL objectives such as PPO or GRPO (Schulman et al., 2017; Shao et al., 2024), multiplies the score derivative by a fixed advantage. RSPO instead multiplies it by $\widetilde{A}_i - \lambda \widehat{\delta}_i$. If the centered score is still below its reward-implied target $\widetilde{A}_i/\lambda$, the coefficient is positive and the update increases the score. If the score has already passed that target, the coefficient becomes negative and pulls it back. Near the target, the coefficient is small. So the update is still a policy-score update, but the coefficient now measures how much target error remains.

Corollary 1 (Standard advantage-weighted objective at the reference point)**.** *If the coupled current and reference ELBO scores match for every sampled completion in the micro-batch, then $\widehat{\delta}_i = 0$ for all $i \in \mathcal{B}$ and $\nabla_\theta \ell_{\mathrm{RSPO}}(\theta; \mathcal{B}) = \nabla_\theta \ell_{\mathrm{AW}}(\theta; \mathcal{B})$. If additionally $\sum_{i \in \mathcal{B}} \widetilde{A}_i = 0$, then centering does not change the forward value of this standard surrogate.*

This corollary explains why RSPO should be viewed as a modification of advantage-weighted RLVR, not a replacement for the policy-improvement direction. At the reference point, before the current model has moved away from the reference on the sampled completions, RSPO gives the same first-order update as the standard advantage-weighted surrogate. The distinction appears after the relative scores start moving: advantage weighting keeps applying the same coefficient, whereas RSPO asks whether the current score is still below, near, or above its target.

Proposition 2 (Fixed point under zero-sum advantages)**.** *Assume $\lambda > 0$ and $\sum_{i \in \mathcal{B}} \widetilde{A}_i = 0$. If*

$$\widehat{\delta}_i = \widetilde{A}_i/\lambda, \qquad i \in \mathcal{B}, \tag{5}$$

then every sample weight is zero and $\nabla_\theta \ell_{\mathrm{RSPO}}(\theta; \mathcal{B}) = 0$. Conversely, if the Jacobian of $\delta(\theta)$ has full row rank on the micro-batch, every first-order stationary point satisfies Eq. equation 5.

The fixed point is the formal version of the target-tracking story. A reward advantage is not treated as a coefficient to apply indefinitely. It specifies a finite centered current-reference score, with λ setting the scale of that target. The full-row-rank assumption is only needed for the converse direction; even without it, Eq. equation 5 is the condition under which all RSPO feedback weights vanish.

4.2 First-Order Equivalence of RSPO

The preceding results describe the coefficient used by RSPO. There is still a natural question: should RSPO be understood as a calibration objective, or as a policy-optimization update? The answer is that, locally, it is both. It tracks the same target as a simple quadratic calibration loss, while keeping the implementation in the familiar form of a detached coefficient times a differentiable policy score.

Let δ and $\widehat{\delta}$ denote the micro-batch vectors, and write $\langle a, b \rangle_{\mathcal{B}} = N^{-1} \sum_{i \in \mathcal{B}} a_i b_i$ and $\|a\|_{\mathcal{B}}^2 = \langle a, a \rangle_{\mathcal{B}}$. Consider the matched quadratic objective

$$\ell_{\mathrm{quad},\lambda}(\theta; \mathcal{B}) = -\langle \widetilde{A}, \delta \rangle_{\mathcal{B}} + \frac{\lambda}{2} \|\widehat{\delta}\|_{\mathcal{B}}^2. \tag{6}$$

When the advantages are zero-sum, completing the square shows that Eq. equation 6 has the same target $\widehat{\delta}_i = \widetilde{A}_i/\lambda$ as Proposition 2, up to terms whose derivative vanishes under detached centering; see Appendix D.1. This comparison separates the target being tracked from the computational graph used to track it.

Theorem 1 (First-order equivalence to a matched quadratic objective)**.** *For fixed λ and a fixed sampled micro-batch,*

$$\nabla_\theta \ell_{\mathrm{RSPO}}(\theta; \mathcal{B}) = \nabla_\theta \ell_{\mathrm{quad},\lambda}(\theta; \mathcal{B}). \tag{7}$$

Thus RSPO follows the same first-order update vector as Eq. equation 6, while detaching one copy of $\widehat{\delta}$ in the feedback term.

The theorem clarifies the role of the stop-gradient construction. The finite target can be written as a quadratic calibration target, but RSPO implements the corresponding first-order direction as a policy-score update using the quantities already computed during training: a centered current-reference ELBO score and a detached residual coefficient. The equivalence is only first-order. The scalar objectives differ for higher-order differentiation, and this section does not claim a general variance reduction or global convergence guarantee. Instead, it gives the local mechanism that the experiments probe next: whether residual feedback, reference subtraction, and centering improve dLLM RLVR when the likelihood signal is an ELBO-based surrogate.

5 EXPERIMENTS

The experiments are designed to test the two claims suggested by the method and theory. First, if verifier rewards can serve as targets for centered relative scores, then RSPO should be especially useful in settings where the reward is reliable but sparse: the verifier can judge a completed answer, but it does not provide token-level or denoising-step supervision. Second, if the gains come from the relative-score feedback mechanism, then removing feedback, reference subtraction, or centering should change the optimization behavior in predictable ways. We therefore evaluate RSPO on both planning and mathematical reasoning benchmarks, and then use ablations to isolate the roles of feedback, current-reference scoring, and centering.

5.1 EXPERIMENTAL SETUP

Models and Datasets. We follow the experimental protocols of d1 (Zhao et al., 2025) and wd1 (Tang et al., 2025). We use LLaDA-8B-Instruct (Nie et al.) as the base diffusion language models. Our evaluation covers two categories of reasoning tasks: mathematical reasoning, including GSM8K (Cobbe et al., 2021) and MATH500 (Lightman et al., 2023), and planning, including Countdown (Pan et al., 2025a) and Sudoku (cor). We follow prior work (Zhao et al., 2025; Tang et al., 2025) for train-test splits, reward functions, and evaluation protocols, except for Sudoku. For Sudoku, to avoid train-test leakage, we split the data by puzzle solutions so that the test set contains unseen answers, preventing models from solving test instances through memorization.

Baselines. We compare RSPO with representative RL fine-tuning methods for diffusion language models, including d1/Diffu-GRPO (Zhao et al., 2025), VRPO on LLaDA-1.5 (Zhu et al., 2025), wd1 (Tang et al., 2025), SAPO (Xie et al., 2025), and TraceRL (Wang et al., 2025b). These baselines cover both GRPO-style adaptations and recent dLLM-specific policy optimization methods, providing a broad comparison across mathematical and planning benchmarks.

Training Details. For RSPO training, we apply Low-Rank Adaptation (LoRA) with rank $r = 128$ and scaling factor $\alpha = 64$, and set $\lambda = 0.01$. All experiments are conducted in the zero-shot setting. For both RL rollouts and evaluation, we use the semi-autoregressive confidence-based decoding strategy following LLaDA, d1, and wd1. We apply the same generation setup as d1: the denoising timestep is set to half of the total sequence length, and the sequence is divided into blocks of 32 tokens. At each diffusion step, we unmask the two tokens with the highest confidence, measured by the probability of the sampled token, within the current incomplete block. During RL rollouts, we use a generation length of 256 and a sampling temperature of 0.9 across all benchmarks, except for Sudoku, where the temperature is set to 0.3 following d1. During evaluation, the sampling temperature is set to 0.0. We evaluate checkpoints every 100 training steps and report results from the checkpoint with the highest average test accuracy across generation lengths of 256 and 512.

5.2 MAIN RESULTS

Table 1 reports final accuracy on planning and mathematical reasoning benchmarks. RSPO gives its clearest gains on sparse-reward planning tasks: on Sudoku, it reaches 92.1/90.8 accuracy at generation lengths 256/512, improving over the strongest non-RSPO result by 66.5/65.4 points; on Countdown, it reaches 78.83/73.83, improving by 26.83/17.53 points. These gains are consistent across generation budgets and align with the motivation of RSPO: reliable response-level rewards can serve as calibrated targets for noisy ELBO-based current-reference scores. On mathematical reasoning, the pattern is more mixed but still favorable. RSPO obtains the best length-256 results on both GSM8K and MATH500, and the best length-512 result on MATH500. On GSM8K at length

Table 1: Benchmark accuracy of RSPO and dLLM RL baselines across planning and mathematical reasoning tasks. Columns 256 and 512 denote generation length; best results are in **bold**.

Methods	Sudoku		Countdown		GSM8K		Math500	
	256	512	256	512	256	512	256	512
LLaDA-8B-Instruct (Nie et al.)	16.2	6.0	19.5	16.0	76.7	78.2	32.4	36.2
+ d1 (Zhao et al., 2025)	24.1	15.9	31.3	37.1	78.1	81.2	34.1	39.0
+ VRPO (Zhu et al., 2025)	12.8	9.6	22.3	18.0	80.1	81.5	35.6	34.8
+ wd1 (Tang et al., 2025)	22.0	24.6	51.2	46.1	80.8	82.3	34.4	39.0
+ SAPO (Xie et al., 2025)	20.3	16.1	52.0	56.3	80.6	82.1	33.8	38.4
+ TraceRL (Wang et al., 2025b)	25.6	25.4	50.4	52.6	81.3	**82.4**	35.6	39.1
+ RSPO (ours)	**92.1**	**90.8**	**78.83**	**73.83**	**82.63**	80.55	**42.07**	**40.25**

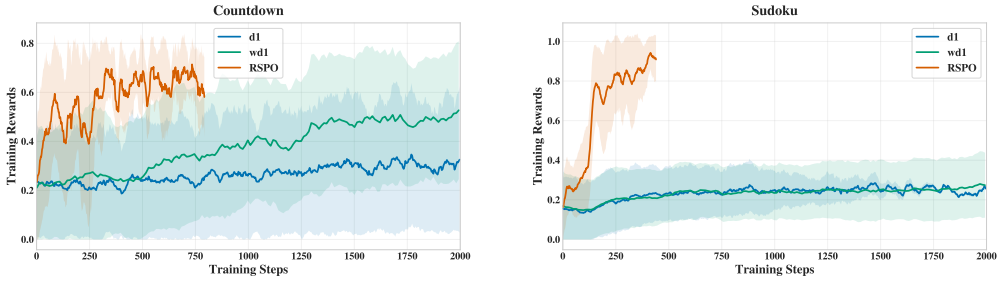


Figure 2: Training reward dynamics of RSPO and baselines on planning benchmarks. Shaded regions indicate variation across runs.

512, RSPO is below the strongest baseline, indicating that the benefit is task-dependent in high-performing mathematical-reasoning regimes. *Overall, RSPO is strongest on planning benchmarks while remaining competitive on standard mathematical reasoning tasks.*

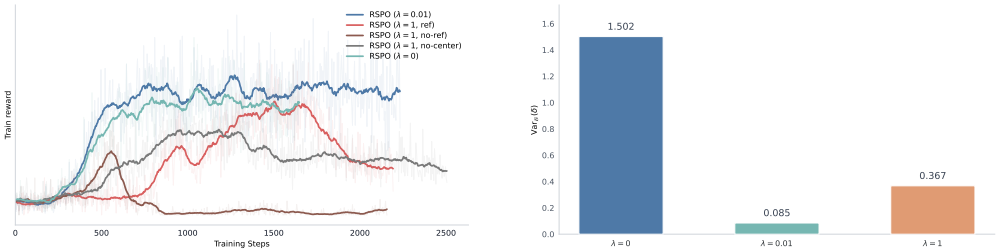
Figure 2 reports training reward trajectories on the planning benchmarks. On both Countdown and Sudoku, RSPO improves verifier rewards faster than d1 and wd1 and reaches higher reward levels within the plotted training window. This trajectory evidence suggests that RSPO changes the optimization path during training, rather than merely selecting a better final checkpoint. It also matches the local mechanism in Section 4: as a response approaches its reward-implied relative-score target, RSPO reduces or reverses the update coefficient instead of repeatedly applying the same fixed advantage. Additional reasoning reward trajectories are provided in Figure 4 in Appendix G.

5.3 Ablation Studies

The ablations focus on Sudoku_new (Wang et al., 2025a), a sparse correctness-reward planning task, to isolate the three RSPO components introduced in Sections 3 and 4: residual feedback through λ, current-reference subtraction in δ_i, and centering of the relative scores.

Relative-Score Feedback. We first ablate the feedback coefficient. Setting $\lambda = 0$ removes the residual feedback term and reduces the update to the standard advantage-weighted form over the centered current-reference score. Figure 3a shows that this variant improves early but converges to a lower reward level, supporting Section 3: fixed advantages rank samples but do not define finite relative-score targets. RSPO with $\lambda = 0.01$ gives the best reward dynamics, whereas stronger feedback at $\lambda = 1$ is less effective and removing reference-model subtraction at $\lambda = 1$ severely degrades reward. *Thus RSPO works best when feedback calibrates current-reference scores without letting noisy ELBO estimates dominate reward-driven improvement.*

Relative-Score Variance Diagnostics. We next examine the spread of the centered current-reference scores. Figure 3b reports $\mathrm{Var}_{\mathcal{B}}(\delta)$, the batch variance of the ELBO-based relative scores after centering. Because centering removes shared score offsets, this quantity measures relative spread within the response group rather than a batch-level shift. Without feedback ($\lambda = 0$), $\mathrm{Var}_{\mathcal{B}}(\delta)$ is large; at $\lambda = 1$, the spread is reduced but rewards are worse. The moderate setting $\lambda = 0.01$ gives both the strongest reward dynamics and the smallest measured $\mathrm{Var}_{\mathcal{B}}(\delta)$, consistent with Proposition 1. *Thus*



(a) Training reward across feedback variants (b) Variance of centered relative scores $\mathrm{Var}_{\mathcal{B}}(\delta)$

Figure 3: Sudoku ablations of RSPO feedback components. The two panels compare reward dynamics and centered relative-score variance under different feedback settings.

Table 2: Batch-mean relative-score offset in Sudoku ablations. The table checks whether centering removes shared ELBO-score shifts across feedback and reference settings.

λ	Centering	Reference	Mean $\lvert\text{offset}\rvert$	Std.	Range
0	✓	✓	1.80×10^{-8}	2.40×10^{-8}	$[-1.14,\ 1.23]\times10^{-7}$
0.01	✓	✓	1.32×10^{-9}	2.41×10^{-9}	$[-1.68,\ 1.49]\times10^{-8}$
1	✓	✓	9.91×10^{-9}	1.28×10^{-8}	$[-5.59,\ 5.59]\times10^{-8}$
1	✓	–	3.94×10^{-8}	4.94×10^{-8}	$[-1.49,\ 1.49]\times10^{-7}$
1	–	✓	1.14×10^{-1}	1.42×10^{-1}	$[-4.11,\ 5.08]\times10^{-1}$

the diagnostic supports feedback calibration, but remains only a proxy related to Appendix D.2 and is not used to adapt λ.

Batch-Mean Score-Offset Diagnostics. Finally, we check the centering operation itself. RSPO applies feedback to $\widehat{\delta}_i$, not to the raw ELBO difference δ_i, so shared ELBO-score shifts do not affect within-group preferences. Table 2 reports that the mean absolute batch-mean offset stays near zero with centering, including 1.32×10^{-9} at $\lambda = 0.01$ and 9.91×10^{-9} at $\lambda = 1$, while removing centering raises it to 1.14×10^{-1} with much larger variation. *This confirms the centering condition in Lemma 1: RSPO feedback should act on relative likelihood changes within the response group, rather than on a shared ELBO-score shift.*

6 Conclusion and Limitations

This paper introduced RSPO, a relative-score objective for RLVR post-training of diffusion language models. The central idea is to use group-relative verifier rewards not only to rank sampled completions, but also to define where their centered current-reference scores should move. RSPO implements this idea by converting the gap between the reward-implied target and the current centered ELBO score into a detached feedback coefficient for a policy-optimization update. This gives a direct way to use noisy dLLM likelihood surrogates without treating reward advantages as indefinitely applied update weights. Empirically, RSPO shows its clearest gains on sparse-reward planning tasks and remains competitive on mathematical reasoning benchmarks, with ablations supporting the roles of residual feedback, reference subtraction, and centering.

We also note several limitations that point to useful future directions. The current configurations of RSPO keep RSPO simple and close to standard group-relative RLVR, but they leave open how to adapt λ automatically or vary the reference policy during training. The theory in Section 4 is local and first-order: it explains the implemented update by matching RSPO to a quadratic target objective at the gradient level, but does not establish global convergence or higher-order equivalence. It is interesting and important to develop a global theory for dLLM RL with ELBO-based score estimates. Finally, the score-spread diagnostic $\mathrm{Var}_{\mathcal{B}}(\delta)$ is only a local proxy rather than an exact KL estimator, and implementations must respect the ELBO sign convention in Remark 1; more precise trust-region diagnostics and robust score-estimation procedures are promising directions for future work.

AI USE STATEMENT

We used generative AI tools, including an LLM-based coding assistant, in several parts of this work. For the manuscript, they drafted and revised text, which we then edited. For the experiments, they wrote, debugged, and ran analysis and plotting code, including the code for the swap-readout analyses, the feature-level intervention, and the figures, and they suggested some analyses and controls, which we evaluated before adopting. The research questions, the selection of experiments, and the conclusions are our own. We reviewed all AI-assisted text, checked the code and its outputs against the reported results, and take full responsibility for the content of this paper, including any errors.

REFERENCES

Arel's sudoku generator. https://www.ocf.berkeley.edu/~arel/sudoku/main.html. Accessed: 2026-05-03.

Inclusion AI, Tiwei Bie, Haoxing Chen, Tieyuan Chen, Zhenglin Cheng, Long Cui, Kai Gan, Zhicheng Huang, Zhenzhong Lan, Haoquan Li, et al. Llada2. 0-uni: Unifying multimodal understanding and generation with diffusion large language model. *arXiv preprint arXiv:2604.20796*, 2026.

Marianne Arriola, Aaron Gokaslan, Justin T Chiu, Zhihan Yang, Zhixuan Qi, Jiaqi Han, Subham Sekhar Sahoo, and Volodymyr Kuleshov. Block diffusion: Interpolating between autoregressive and diffusion language models. In *The Thirteenth International Conference on Learning Representations*.

Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne Van Den Berg. Structured denoising diffusion models in discrete state-spaces. *Advances in neural information processing systems*, 34:17981–17993, 2021.

Tiwei Bie, Maosong Cao, Kun Chen, Lun Du, Mingliang Gong, Zhuochen Gong, Yanmei Gu, Jiaqi Hu, Zenan Huang, Zhenzhong Lan, et al. Llada2. 0: Scaling up diffusion language models to 100b. *arXiv preprint arXiv:2512.15745*, 2025.

Andrew Campbell, Joe Benton, Valentin De Bortoli, Thomas Rainforth, George Deligiannidis, and Arnaud Doucet. A continuous time framework for discrete denoising models. *Advances in Neural Information Processing Systems*, 35:28266–28279, 2022.

Shuang Cheng, Yihan Bian, Dawei Liu, Linfeng Zhang, Qian Yao, Zhongbo Tian, Wenhai Wang, Qipeng Guo, Kai Chen, Biqing Qi, et al. Sdar: A synergistic diffusion-autoregression paradigm for scalable sequence generation. *arXiv preprint arXiv:2510.06303*, 2025.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

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

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Yoav HaCohen, Nisan Chiprut, Benny Brazowski, Daniel Shalem, Dudu Moshe, Eitan Richardson, Eran Levin, Guy Shiran, Nir Zabari, Ori Gordon, et al. Ltx-video: Realtime video latent diffusion. *arXiv preprint arXiv:2501.00103*, 2024.

Yoav HaCohen, Benny Brazowski, Nisan Chiprut, Yaki Bitterman, Andrew Kvochko, Avishai Berkowitz, Daniel Shalem, Daphna Lifschitz, Dudu Moshe, Eitan Porat, et al. Ltx-2: Efficient joint audio-visual foundation model. *arXiv preprint arXiv:2601.03233*, 2026.

Shenghua He, Tian Xia, Xuan Zhou, and Hui Wei. Response-level rewards are all you need for online reinforcement learning in llms: A mathematical perspective. *arXiv preprint arXiv:2506.02553*, 2025.

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

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Liang Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *Iclr*, 1(2):3, 2022.

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *The twelfth international conference on learning representations*, 2023.

Nianyi Lin, Jiajie Zhang, Lei Hou, and Juanzi Li. Boundary-guided policy optimization for memory-efficient rl of diffusion large language models. *arXiv preprint arXiv:2510.11683*, 2025.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution. In *Forty-first International Conference on Machine Learning*.

Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, JUN ZHOU, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. Large language diffusion models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Jingyang Ou, Shen Nie, Kaiwen Xue, Fengqi Zhu, Jiacheng Sun, Zhenguo Li, and Chongxuan Li. Your absorbing discrete diffusion secretly models the conditional distributions of clean data. In *The Thirteenth International Conference on Learning Representations*.

Jingyang Ou, Jiaqi Han, Minkai Xu, Shaoxuan Xu, Jianwen Xie, Stefano Ermon, Yi Wu, and Chongxuan Li. Principled rl for diffusion llms emerges from a sequence-level perspective. *arXiv preprint arXiv:2512.03759*, 2025.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, 2022.

Jiayi Pan, Junjie Zhang, Xingyao Wang, Lifan Yuan, Hao Peng, and Alane Suhr. Tinyzero. https://github.com/Jiayi-Pan/TinyZero, 2025a. Accessed: 2025-01-24.

Leyi Pan, Shuchang Tao, Yunpeng Zhai, Zheyu Fu, Liancheng Fang, Minghua He, Lingzhe Zhang, Zhaoyang Liu, Bolin Ding, Aiwei Liu, et al. d-treerpo: Towards more reliable policy optimization for diffusion language models. *arXiv preprint arXiv:2512.09675*, 2025b.

Subham S Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin T Chiu, Alexander Rush, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. *Advances in Neural Information Processing Systems*, 37:130136–130184, 2024.

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

Team Seedance, De Chen, Liyang Chen, Xin Chen, Ying Chen, Zhuo Chen, Zhuowei Chen, Feng Cheng, Tianheng Cheng, Yufeng Cheng, et al. Seedance 2.0: Advancing video generation for world complexity. *arXiv preprint arXiv:2604.14148*, 2026.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Jiaxin Shi, Kehang Han, Zhe Wang, Arnaud Doucet, and Michalis Titsias. Simplified and generalized masked diffusion for discrete data. *Advances in neural information processing systems*, 37: 103131–103167, 2024.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*.

Richard S Sutton, Andrew G Barto, et al. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

Xiaohang Tang, Rares Dolga, Sangwoong Yoon, and Ilija Bogunovic. wd1: Weighted policy optimization for reasoning in diffusion language models. *arXiv preprint arXiv:2507.08838*, 2025.

Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, Shengyi Huang, Kashif Rasul, and Quentin Gallouédec. TRL: Transformers Reinforcement Learning, 2020. URL `https://github.com/huggingface/trl`.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

Chenyu Wang, Paria Rashidinejad, DiJia Su, Song Jiang, Sid Wang, Siyan Zhao, Cai Zhou, Shannon Zejiang Shen, Feiyu Chen, Tommi Jaakkola, et al. Spg: Sandwiched policy gradient for masked diffusion language models. *arXiv preprint arXiv:2510.09541*, 2025a.

Yinjie Wang, Ling Yang, Bowen Li, Ye Tian, Ke Shen, and Mengdi Wang. Revolutionizing reinforcement learning framework for diffusion large language models. *arXiv preprint arXiv:2509.06949*, 2025b.

Shaoan Xie, Lingjing Kong, Xiangchen Song, Xinshuai Dong, Guangyi Chen, Eric P Xing, and Kun Zhang. Step-aware policy optimization for reasoning in diffusion large language models. *arXiv preprint arXiv:2510.01544*, 2025.

Ling Yang, Ye Tian, Bowen Li, Xinchen Zhang, Ke Shen, Yunhai Tong, and Mengdi Wang. Mmada: Multimodal large diffusion language models. *arXiv preprint arXiv:2505.15809*, 2025.

Jiacheng Ye, Zhihui Xie, Lin Zheng, Jiahui Gao, Zirui Wu, Xin Jiang, Zhenguo Li, and Lingpeng Kong. Dream 7b: Diffusion large language models. *arXiv preprint arXiv:2508.15487*, 2025.

Zebin You, Shen Nie, Xiaolu Zhang, Jun Hu, Jun Zhou, Zhiwu Lu, Ji-Rong Wen, and Chongxuan Li. Llada-v: Large language diffusion models with visual instruction tuning. *arXiv preprint arXiv:2505.16933*, 2025.

Zebin You, Xiaolu Zhang, Jun Zhou, Chongxuan Li, and Ji-Rong Wen. Llada-o: An effective and length-adaptive omni diffusion model. *arXiv preprint arXiv:2603.01068*, 2026.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025.

Siyan Zhao, Devaansh Gupta, Qinqing Zheng, and Aditya Grover. d1: Scaling reasoning in diffusion large language models via reinforcement learning. *arXiv preprint arXiv:2504.12216*, 2025.

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

Zangwei Zheng, Xiangyu Peng, Yuxuan Lou, Chenhui Shen, Tom Young, Xinying Guo, Binluo Wang, Hang Xu, Hongxin Liu, Mingyan Jiang, et al. Open-sora 2.0: Training a commercial-level video generation model in $200 k. *arXiv preprint arXiv:2503.09642*, 2025b.

Jianyuan Zhong, Kaibo Wang, Ding Ding, Zijin Feng, Haoli Bai, Yang Xiang, Jiacheng Sun, and Qiang Xu. Stabilizing reinforcement learning for diffusion language models. *arXiv preprint arXiv:2603.06743*, 2026.

Fengqi Zhu, Rongzhen Wang, Shen Nie, Xiaolu Zhang, Chunwei Wu, Jun Hu, Jun Zhou, Jianfei Chen, Yankai Lin, Ji-Rong Wen, et al. Llada 1.5: Variance-reduced preference optimization for large language diffusion models. *arXiv preprint arXiv:2505.19223*, 2025.

CONTENTS

1 Introduction **1**

2 Preliminaries **2**

3 Method **3**

3.1 Motivation 3

3.2 Relative Score Policy Optimization 4

3.3 Training Procedure 5

4 Theory **5**

4.1 Relative-Score Feedback and Fixed Point 5

4.2 First-Order Equivalence of RSPO 6

5 Experiments **7**

5.1 Experimental Setup 7

5.2 Main Results 7

5.3 Ablation Studies 8

6 Conclusion and Limitations **9**

A Notation Summary **15**

B Training Procedure and Score Estimation **15**

B.1 Masked diffusion ELBO estimator 15

B.2 Algorithm 16

C Related Work **16**

D Additional Theory Details **17**

D.1 Coefficient convention for the matched quadratic objective 17

D.2 Local KL interpretation of the monitored variance 17

E Additional Proof Details **18**

E.1 Proof of Lemma 1 18

E.2 Proof of Proposition 1 18

E.3 Proof of Corollary 1 19

E.4 Proof of Proposition 2 19

E.5 Proof of Theorem 1 20

E.6 Proof of Proposition 3 20

E.7 Perturbation from surrogate error 21

F Implementation Notes **22**

G Additional Experimental Details **22**

A Notation Summary

Table 3: Notation used in the RSPO formulation.

Symbol	Meaning
q	Prompt
y	Generic completion used in policy-improvement discussion
o_i	The i-th sampled completion in a group
G	Group size
K	Number of ELBO Monte Carlo mask samples per completion
$\mathcal{B}$	Current loss micro-batch
N	Micro-batch size, $\lvert\mathcal{B}\rvert$
π_θ	Current model or policy parameterized by θ
π_{ref}	Frozen reference model or policy
$\pi^\star$	Ideal KL-regularized improved policy in the motivating calculation
β	KL temperature in the motivating policy-improvement calculation
$Z(q)$	Prompt-level normalizer in the motivating policy-improvement calculation
r_i	Reward of completion o_i
$\bar{r}$	Group mean reward, $G^{-1}\sum_i r_i$
σ_r	Group reward standard deviation for the normalized advantage variant
s_q	Prompt-level scale used by the normalized advantage variant
$\widetilde{A}_i$	Zero-sum group-relative advantage passed to the loss, optionally divided by $\sigma_r + 10^{-4}$
L_c	Number of completion tokens used to normalize score differences
$\widehat{\mathcal{E}}_\theta$	Likelihood-oriented ELBO or score estimate for the current model
$\widehat{\mathcal{E}}_{\text{ref}}$	Matched reference-model ELBO or score estimate
$M_{i,k}$	The k-th sampled diffusion mask set for completion o_i
δ_i	Per-token ELBO log-ratio surrogate, $(\widehat{\mathcal{E}}_\theta(o_i;q) - \widehat{\mathcal{E}}_{\text{ref}}(o_i;q))/L_c$
$\bar{\delta}_{\mathcal{B}}$	Current micro-batch mean of δ
$\widehat{\delta}_i$	Centered relative-score surrogate, $\delta_i - \text{sg}(\bar{\delta}_{\mathcal{B}})$
λ	Fixed feedback coefficient; the main RSPO experiments set $\lambda = 0.01$ unless otherwise stated
$\text{sg}(\cdot)$	Detachment operator
e_i	Remaining calibration gap, $\widetilde{A}_i - \lambda\widehat{\delta}_i$
w_i	RSPO weight, $\widetilde{A}_i - \lambda\,\text{sg}(\widehat{\delta}_i)$
ℓ_{AW}	Standard group-relative advantage-weighted surrogate
ℓ_{RSPO}	RSPO objective
$\ell_{\text{quad},\lambda}$	Matched quadratic objective with RSPO's relative-score target
$\langle a,b\rangle_{\mathcal{B}}$	Micro-batch inner product, $N^{-1}\sum_{i\in\mathcal{B}} a_i b_i$
$\lVert a\rVert_{\mathcal{B}}$	Micro-batch norm, with $\lVert a\rVert_{\mathcal{B}}^2 = \langle a,a\rangle_{\mathcal{B}}$
$\text{Var}_{\mathcal{B}}(\delta)$	Within-micro-batch variance of the relative-score surrogate

B Training Procedure and Score Estimation

This appendix specifies the score-estimation and training details used by RSPO. The main text abstracts these details into an ELBO-based current-reference relative score δ_i for each completion, micro-batch centering, and the relative-score feedback loss in Eq. equation 3.

B.1 Masked diffusion ELBO estimator

For a prompt-completion pair (q, o_i), let L_c be the number of completion tokens. We estimate a likelihood-oriented sequence score by sampling K masks and averaging denoising log-probabilities

on completion tokens:

$$\widehat{\mathcal{E}}_\theta(o_i; q) = \frac{1}{K}\sum_{k=1}^{K} \frac{L_c}{|M_{i,k} \cap o_i|} \sum_{t \in M_{i,k} \cap o_i} \log p_\theta(x_t \mid x_{\setminus M_{i,k}}, q, M_{i,k}). \tag{8}$$

Here $M_{i,k}$ is the k-th sampled mask set for completion o_i, x_t is the token at position t, $x_{\setminus M_{i,k}}$ denotes the sequence context outside the masked positions, and p_θ is the current model's denoising distribution. Masks with no completion token are resampled or omitted from the Monte Carlo average. For each completion, the same mask set is reused for the current model and the reference model, giving the coupled per-token score difference

$$\delta_i = \frac{\widehat{\mathcal{E}}_\theta(o_i; q) - \widehat{\mathcal{E}}_{\mathrm{ref}}(o_i; q)}{L_c}.$$

The factor $L_c/|M_{i,k} \cap o_i|$ makes $\widehat{\mathcal{E}}_\theta$ a sequence-level completion score, so the division by L_c in δ_i gives the per-token score difference used in Eq. equation 2. If an implementation stores a negative ELBO or cross-entropy instead, the sign must follow Remark 1.

B.2 Algorithm

Algorithm 1 RSPO for diffusion language model reinforcement learning

Require: Current dLLM π_θ, fixed reference dLLM π_{ref}, reward function R, group size G, ELBO samples K, feedback coefficient $\lambda > 0$.

1: **for** each training step **do**
2: Sample prompts and generate G completions $\{o_i\}_{i=1}^{G}$ per prompt from π_θ.
3: Compute rewards $r_i = R(q, o_i)$ and group-relative advantages within each prompt group.
4: Optionally normalize the advantages within the group to obtain $\widetilde{A}_i$.
5: Estimate $\widehat{\mathcal{E}}_\theta(o_i; q)$ and $\widehat{\mathcal{E}}_{\mathrm{ref}}(o_i; q)$ with the shared masks in Eq. equation 8.
6: Form $\delta_i = (\widehat{\mathcal{E}}_\theta(o_i; q) - \widehat{\mathcal{E}}_{\mathrm{ref}}(o_i; q))/L_c$ and center it within the loss micro-batch: $\widehat{\delta}_i = \delta_i - \mathrm{sg}(\bar{\delta}_{\mathcal{B}})$.
7: Set the relative-score feedback weight $w_i = \widetilde{A}_i - \lambda\,\mathrm{sg}(\widehat{\delta}_i)$ and descend $\ell_{\mathrm{RSPO}}(\theta; \mathcal{B}) = -N^{-1}\sum_{i \in \mathcal{B}} w_i \widehat{\delta}_i$.
8: Record $\mathrm{Var}_{\mathcal{B}}(\delta)$ as a stability diagnostic.
9: **end for**

C Related Work

Diffusion Language Models. Diffusion models were originally developed as a powerful generative framework for continuous data, especially high-fidelity image generation (Song et al.; Ho et al., 2020; Dhariwal & Nichol, 2021) and video generation (Wan et al., 2025; Seedance et al., 2026; HaCohen et al., 2024; 2026; Zheng et al., 2025b). More recently, this paradigm has been extended to discrete text generation, giving rise to diffusion language models that generate token sequences through iterative denoising (Austin et al., 2021; Campbell et al., 2022; Lou et al.; Sahoo et al., 2024; Shi et al., 2024). Unlike autoregressive models, dLLMs are not tied to a fixed left-to-right generation order, enabling bidirectional context modeling, flexible-order refinement, and parallel decoding. Recent large-scale dLLMs, such as LLaDA (Nie et al.; Bie et al., 2025), Dream (Ye et al., 2025), multimodal diffusion language models (Yang et al., 2025; You et al., 2025; 2026; AI et al., 2026), and block-diffusion variants (Arriola et al.; Cheng et al., 2025), have demonstrated competitive generation quality and promising efficiency compared with autoregressive models. These advances motivate post-training methods that can further improve dLLMs on reasoning, alignment, and task-specific objectives.

Reinforcement Learning for Diffusion Language Models. RL for dLLMs is challenging because PPO/GRPO-style objectives rely on policy log-probabilities or importance ratios that are tractable for autoregressive models but unavailable for diffusion generation. Existing methods mainly differ in how they approximate or avoid these ratios. diffu-GRPO (Zhao et al., 2025) uses one-step mean-field

log-probability estimates to adapt GRPO to masked dLLMs, while wd1 (Tang et al., 2025) removes explicit importance ratios by reformulating policy optimization as a weighted log-likelihood objective. ELBO-based methods approximate likelihoods or log-ratios with denoising objectives, including VRPO in LLaDA 1.5 (Zhu et al., 2025), sequence-level ELBO policy optimization (Ou et al., 2025), sandwiched evidence-bound policy gradients (Wang et al., 2025a), and memory-efficient large Monte Carlo (large-MC) ELBO optimization (Lin et al., 2025). Other works exploit diffusion trajectories or step structure, such as TraceRL (Wang et al., 2025b) and d-TreeRPO (Pan et al., 2025b), or stabilize noisy ratio-based training with clipping and self-normalization (Zhong et al., 2026). Our work is complementary: instead of changing the likelihood surrogate, trajectory estimator, or clipping rule, we derive the centered log-ratio target implied by KL-regularized group-relative policy improvement and optimize noisy ELBO log-ratios toward this target with relative-score feedback.

D Additional Theory Details

D.1 Coefficient convention for the matched quadratic objective

Expanding the RSPO loss makes the coefficient convention explicit:

$$\ell_{\mathrm{RSPO}}(\theta;\mathcal{B}) = -\langle \widetilde{A}, \widehat{\delta}\rangle_{\mathcal{B}} + \lambda \langle \mathrm{sg}(\widehat{\delta}), \widehat{\delta}\rangle_{\mathcal{B}}. \tag{9}$$

The forward value of the detached feedback term in Eq. equation 9 is $\lambda\|\widehat{\delta}\|^2_{\mathcal{B}}$, but its first-order gradient is the gradient of $\frac{\lambda}{2}\|\widehat{\delta}\|^2_{\mathcal{B}}$. If one removes the detachment operator from the written scalar expression $\lambda\langle \mathrm{sg}(\widehat{\delta}), \widehat{\delta}\rangle_{\mathcal{B}}$ without changing the coefficient, the penalty gradient doubles. Therefore, comparisons with a fully differentiable matched quadratic objective use the first-order matched coefficient, i.e., $\frac{\lambda}{2}\|\widehat{\delta}\|^2_{\mathcal{B}}$.

Let $c_{\mathcal{B}} = \mathrm{sg}(N^{-1}\sum_{j\in\mathcal{B}}\delta_j)$ be the detached micro-batch center and let $\mathbf{1}$ be the all-ones vector, so $\widehat{\delta} = \delta - c_{\mathcal{B}}\mathbf{1}$. Completing the square in Eq. equation 6 gives the local relative-score target form

$$\ell_{\mathrm{quad},\lambda}(\theta;\mathcal{B}) = \frac{\lambda}{2}\|\widehat{\delta} - \frac{1}{\lambda}\widetilde{A}\|^2_{\mathcal{B}} - \frac{1}{2\lambda}\|\widetilde{A}\|^2_{\mathcal{B}} - c_{\mathcal{B}}\langle \widetilde{A}, \mathbf{1}\rangle_{\mathcal{B}}. \tag{10}$$

When the micro-batch advantages sum to zero, the final term vanishes and the visible target is $\widehat{\delta} = \widetilde{A}/\lambda$. If the advantages are not zero-sum, the final term is not a forward-value constant, but it has zero derivative in the local backward pass because $c_{\mathcal{B}}$ is detached.

Theorem 1 is first-order only. RSPO is not the same scalar objective as Eq. equation 6 for higher-order differentiation, and the theorem does not by itself establish lower gradient variance. The precise claim is that the update uses the relative-score feedback coefficient $(\widetilde{A}_i - \lambda\widehat{\delta}_i)$ while preventing one copy of the noisy ELBO difference from contributing a differentiable path in the backward graph. Variance reduction is therefore an empirical property to be evaluated in experiments.

D.2 Local KL interpretation of the monitored variance

The centered log-ratio variance is interpreted as a local divergence proxy rather than as an exact KL estimator. The following second-order calculation gives its scale.

Proposition 3 (Second-order KL proxy)**.** *Let P and Q be two nearby completion distributions for the same prompt with common support, let Y denote a completion sampled from the distribution indicated in the expectation, and define*

$$\delta(y) = \log\frac{Q(y)}{P(y)}.$$

Assume δ is uniformly small, where $\|\delta\|_\infty$ denotes the supremum norm. Then

$$\mathrm{KL}(P\|Q) = \frac{1}{2}\,\mathrm{Var}_{Y\sim P}(\delta(Y)) + O(\|\delta\|^3_\infty),$$

and, to the same second order,

$$\mathrm{KL}(Q\|P) = \frac{1}{2}\,\mathrm{Var}_{Y\sim P}(\delta(Y)) + O(\|\delta\|^3_\infty).$$

Consequently, when the current and reference policies are close and the sampled micro-batch is representative of the local completion distribution, the monitored quantity $\mathrm{Var}_{\mathcal{B}}(\delta)$ *is a second-order proxy for approximately twice a local KL divergence.*

Proposition 3 supports monitoring $\mathrm{Var}_{\mathcal{B}}(\delta)$ as a trust-region-like diagnostic, but the evaluated method does not use it to update λ. It does not imply that a finite sampled micro-batch variance is an unbiased or exact estimator of either forward or reverse KL. In particular, the approximation depends on the sampling distribution, the sign convention for δ, and the current policy being close to the reference policy.

E Additional Proof Details

E.1 Proof of Lemma 1

The forward-centered surrogate is

$$\widehat{\delta}_i = \delta_i - c_{\mathcal{B}}, \qquad c_{\mathcal{B}} = \mathrm{sg}(\frac{1}{N}\sum_{j\in\mathcal{B}}\delta_j).$$

The detachment operator changes derivatives but not forward values, so in the forward pass

$$\begin{aligned}\sum_{i\in\mathcal{B}}\widehat{\delta}_i &= \sum_{i\in\mathcal{B}}\delta_i - \sum_{i\in\mathcal{B}}\frac{1}{N}\sum_{j\in\mathcal{B}}\delta_j \\ &= \sum_{i\in\mathcal{B}}\delta_i - N\cdot\frac{1}{N}\sum_{j\in\mathcal{B}}\delta_j = 0.\end{aligned}$$

Using $w_i = \widetilde{A}_i - \lambda\,\mathrm{sg}(\widehat{\delta}_i)$ and again using that detachment preserves forward values,

$$\begin{aligned}\sum_{i\in\mathcal{B}} w_i &= \sum_{i\in\mathcal{B}}\widetilde{A}_i - \lambda\sum_{i\in\mathcal{B}}\mathrm{sg}(\widehat{\delta}_i) \\ &= \sum_{i\in\mathcal{B}}\widetilde{A}_i - \lambda\sum_{i\in\mathcal{B}}\widehat{\delta}_i = \sum_{i\in\mathcal{B}}\widetilde{A}_i.\end{aligned}$$

If the group-relative advantages sum to zero within each complete prompt group, then any micro-batch that is a union of complete prompt groups has $\sum_{i\in\mathcal{B}}\widetilde{A}_i = 0$. Any group-shared reward scaling preserves this zero-sum property, and the weights are zero-sum.

E.2 Proof of Proposition 1

The implemented loss is

$$\ell_{\mathrm{RSPO}}(\theta;\mathcal{B}) = -\frac{1}{N}\sum_{i\in\mathcal{B}}(\widetilde{A}_i - \lambda\ \mathrm{sg}(\widehat{\delta}_i))\widehat{\delta}_i.$$

Rewards, advantages, the reference scores, λ, $\mathrm{sg}(\widehat{\delta}_i)$, and the batch center $c_{\mathcal{B}}$ are constants with respect to the derivative. Since

$$\widehat{\delta}_i = \delta_i - c_{\mathcal{B}}, \qquad \nabla_\theta c_{\mathcal{B}} = 0,$$

we have $\nabla_\theta\widehat{\delta}_i = \nabla_\theta\delta_i$. Therefore

$$\begin{aligned}\nabla_\theta\ell_{\mathrm{RSPO}}(\theta;\mathcal{B}) &= -\frac{1}{N}\sum_{i\in\mathcal{B}}(\widetilde{A}_i - \lambda\ \mathrm{sg}(\widehat{\delta}_i))\nabla_\theta\widehat{\delta}_i \\ &= -\frac{1}{N}\sum_{i\in\mathcal{B}}(\widetilde{A}_i - \lambda\widehat{\delta}_i)\nabla_\theta\delta_i(\theta),\end{aligned}$$

which is the stated formula.

E.3 Proof of Corollary 1

If the current model and reference model have identical coupled ELBO scores on the sampled completions, then

$$\widehat{\mathcal{E}}_\theta(o_i;q) = \widehat{\mathcal{E}}_{\text{ref}}(o_i;q) \qquad \text{for all } i \in \mathcal{B}.$$

The per-token surrogate definition gives

$$\delta_i = \frac{\widehat{\mathcal{E}}_\theta(o_i;q) - \widehat{\mathcal{E}}_{\text{ref}}(o_i;q)}{L_c} = 0.$$

Therefore the detached batch center is also zero,

$$c_{\mathcal{B}} = \text{sg}(\frac{1}{N}\sum_{j\in\mathcal{B}} \delta_j) = 0,$$

and hence $\widehat{\delta}_i = \delta_i - c_{\mathcal{B}} = 0$ for every $i \in \mathcal{B}$. The RSPO weight becomes

$$w_i = \widetilde{A}_i - \lambda\ \text{sg}(\widehat{\delta}_i) = \widetilde{A}_i.$$

Using Proposition 1,

$$\nabla_\theta \ell_{\text{RSPO}}(\theta;\mathcal{B}) = -\frac{1}{N}\sum_{i\in\mathcal{B}} \widetilde{A}_i \nabla_\theta \delta_i(\theta).$$

For the standard group-relative advantage-weighted surrogate,

$$\ell_{\text{AW}}(\theta;\mathcal{B}) = -\frac{1}{N}\sum_{i\in\mathcal{B}} \widetilde{A}_i \widehat{\delta}_i(\theta),$$

and $\nabla_\theta \widehat{\delta}_i = \nabla_\theta \delta_i$ because the center is detached. Hence

$$\nabla_\theta \ell_{\text{AW}}(\theta;\mathcal{B}) = -\frac{1}{N}\sum_{i\in\mathcal{B}} \widetilde{A}_i \nabla_\theta \delta_i(\theta) = \nabla_\theta \ell_{\text{RSPO}}(\theta;\mathcal{B}).$$

If $\sum_{i\in\mathcal{B}} \widetilde{A}_i = 0$, then

$$\begin{aligned} -\frac{1}{N}\sum_{i\in\mathcal{B}} \widetilde{A}_i \widehat{\delta}_i &= -\frac{1}{N}\sum_{i\in\mathcal{B}} \widetilde{A}_i (\delta_i - c_{\mathcal{B}}) \\ &= -\frac{1}{N}\sum_{i\in\mathcal{B}} \widetilde{A}_i \delta_i + \frac{c_{\mathcal{B}}}{N}\sum_{i\in\mathcal{B}} \widetilde{A}_i \\ &= -\frac{1}{N}\sum_{i\in\mathcal{B}} \widetilde{A}_i \delta_i. \end{aligned}$$

This proves the forward-value identity.

E.4 Proof of Proposition 2

If $\widehat{\delta}_i = \widetilde{A}_i/\lambda$ for all $i \in \mathcal{B}$, then

$$w_i = \widetilde{A}_i - \lambda\,\text{sg}(\widehat{\delta}_i) = \widetilde{A}_i - \lambda\widehat{\delta}_i = 0$$

for every sample in the micro-batch. Substituting $w_i = 0$ into the loss gradient in Proposition 1 gives

$$\nabla_\theta \ell_{\text{RSPO}}(\theta;\mathcal{B}) = -\frac{1}{N}\sum_{i\in\mathcal{B}} 0 \cdot \nabla_\theta \delta_i(\theta) = 0.$$

The zero-sum assumption $\sum_i \widetilde{A}_i = 0$ is needed for this target to be compatible with the forward centering identity $\sum_i \widehat{\delta}_i = 0$:

$$\sum_{i\in\mathcal{B}} \frac{\widetilde{A}_i}{\lambda} = \frac{1}{\lambda}\sum_{i\in\mathcal{B}} \widetilde{A}_i = 0.$$

For the converse, let $J_\theta = \nabla_\theta \delta(\theta) \in \mathbb{R}^{N \times d}$ be the Jacobian of the micro-batch relative-score vector, where d is the number of trainable parameters. Proposition 1 can be written as

$$\nabla_\theta \ell_{\mathrm{RSPO}}(\theta; \mathcal{B}) = -\frac{1}{N} J_\theta^\top (\widetilde{A} - \lambda \widehat{\delta}).$$

At a stationary point this gradient is zero, so

$$J_\theta^\top (\widetilde{A} - \lambda \widehat{\delta}) = 0.$$

If the Jacobian can realize arbitrary first-order changes in the sampled relative-score vector, equivalently if J_θ has full row rank on these sampled directions, then the nullspace of $J_\theta^\top$ is trivial. Hence

$$\widetilde{A} - \lambda \widehat{\delta} = 0,$$

which is exactly $\widehat{\delta}_i = \widetilde{A}_i / \lambda$ for all $i \in \mathcal{B}$.

E.5 Proof of Theorem 1

Recall the matched quadratic objective:

$$\ell_{\mathrm{quad},\lambda}(\theta; \mathcal{B}) = -\langle \widetilde{A}, \delta \rangle_{\mathcal{B}} + \frac{\lambda}{2} \|\widehat{\delta}\|_{\mathcal{B}}^2, \qquad \widehat{\delta}_i = \delta_i - c_{\mathcal{B}}, \qquad \nabla_\theta c_{\mathcal{B}} = 0.$$

Its gradient is

$$\begin{aligned}
\nabla_\theta \ell_{\mathrm{quad},\lambda} &= -\frac{1}{N} \sum_{i \in \mathcal{B}} \widetilde{A}_i \nabla_\theta \delta_i + \frac{\lambda}{2} \nabla_\theta (\frac{1}{N} \sum_{i \in \mathcal{B}} \widehat{\delta}_i^2) \\
&= -\frac{1}{N} \sum_{i \in \mathcal{B}} \widetilde{A}_i \nabla_\theta \delta_i + \lambda \frac{1}{N} \sum_{i \in \mathcal{B}} \widehat{\delta}_i \nabla_\theta \widehat{\delta}_i \\
&= -\frac{1}{N} \sum_{i \in \mathcal{B}} \widetilde{A}_i \nabla_\theta \delta_i + \lambda \frac{1}{N} \sum_{i \in \mathcal{B}} \widehat{\delta}_i \nabla_\theta \delta_i \\
&= -\frac{1}{N} \sum_{i \in \mathcal{B}} (\widetilde{A}_i - \lambda \widehat{\delta}_i) \nabla_\theta \delta_i.
\end{aligned}$$

By Proposition 1, this equals $\nabla_\theta \ell_{\mathrm{RSPO}}(\theta; \mathcal{B})$. Therefore the two objectives induce the same first-order update vector for fixed λ and fixed sampled data.

The rewriting in Eq. equation 10 follows by completing the square:

$$\begin{aligned}
-\langle \widetilde{A}, \delta \rangle_{\mathcal{B}} + \frac{\lambda}{2} \|\widehat{\delta}\|_{\mathcal{B}}^2 &= -\langle \widetilde{A}, \widehat{\delta} + c_{\mathcal{B}} \mathbf{1} \rangle_{\mathcal{B}} + \frac{\lambda}{2} \|\widehat{\delta}\|_{\mathcal{B}}^2 \\
&= -\langle \widetilde{A}, \widehat{\delta} \rangle_{\mathcal{B}} - c_{\mathcal{B}} \langle \widetilde{A}, \mathbf{1} \rangle_{\mathcal{B}} + \frac{\lambda}{2} \|\widehat{\delta}\|_{\mathcal{B}}^2 \\
&= \frac{\lambda}{2} \|\widehat{\delta} - \frac{1}{\lambda} \widetilde{A}\|_{\mathcal{B}}^2 - \frac{1}{2\lambda} \|\widetilde{A}\|_{\mathcal{B}}^2 - c_{\mathcal{B}} \langle \widetilde{A}, \mathbf{1} \rangle_{\mathcal{B}}.
\end{aligned}$$

E.6 Proof of Proposition 3

Because Q is normalized and $\delta(y) = \log(Q(y)/P(y))$, we have

$$\mathbb{E}_{Y \sim P}[e^{\delta(Y)}] = \sum_y P(y) \frac{Q(y)}{P(y)} = \sum_y Q(y) = 1,$$

with the same argument written as an integral for continuous spaces. Since δ is uniformly small, Taylor expansion gives

$$e^\delta = 1 + \delta + \frac{1}{2} \delta^2 + O(\|\delta\|_\infty^3).$$

Taking expectation under P and using $\mathbb{E}_P[e^\delta] = 1$ yields

$$0 = \mathbb{E}_P[\delta] + \frac{1}{2} \mathbb{E}_P[\delta^2] + O(\|\delta\|_\infty^3).$$

Thus

$$\mathbb{E}_P[\delta] = -\frac{1}{2}\mathbb{E}_P[\delta^2] + O(\|\delta\|_\infty^3).$$

The forward KL is

$$\mathrm{KL}(P\|Q) = \mathbb{E}_P[\log \frac{P(Y)}{Q(Y)}] = -\mathbb{E}_P[\delta(Y)] = \frac{1}{2}\mathbb{E}_P[\delta(Y)^2] + O(\|\delta\|_\infty^3).$$

Moreover $\mathbb{E}_P[\delta] = O(\|\delta\|_\infty^2)$, so

$$\mathrm{Var}_{Y\sim P}(\delta(Y)) = \mathbb{E}_P[\delta(Y)^2] - \mathbb{E}_P[\delta(Y)]^2 = \mathbb{E}_P[\delta(Y)^2] + O(\|\delta\|_\infty^4).$$

Substituting this into the previous display gives

$$\mathrm{KL}(P\|Q) = \frac{1}{2}\,\mathrm{Var}_{Y\sim P}(\delta(Y)) + O(\|\delta\|_\infty^3).$$

For the reverse KL,

$$\begin{aligned}
\mathrm{KL}(Q\|P) &= \mathbb{E}_Q[\log \frac{Q(Y)}{P(Y)}] = \mathbb{E}_Q[\delta(Y)] \\
&= \mathbb{E}_P[e^{\delta(Y)}\delta(Y)] \\
&= \mathbb{E}_P[(1 + \delta(Y) + O(\|\delta\|_\infty^2))\delta(Y)] \\
&= \mathbb{E}_P[\delta(Y)] + \mathbb{E}_P[\delta(Y)^2] + O(\|\delta\|_\infty^3) \\
&= \frac{1}{2}\mathbb{E}_P[\delta(Y)^2] + O(\|\delta\|_\infty^3) \\
&= \frac{1}{2}\,\mathrm{Var}_{Y\sim P}(\delta(Y)) + O(\|\delta\|_\infty^3).
\end{aligned}$$

This proves both claims.

E.7 Perturbation from surrogate error

The theory above treats δ_i as a log-ratio surrogate. Let R_i denote an ideal uncentered log-ratio value for sample i. Suppose the corresponding ideal centered relative-score vector on the current micro-batch is $\widehat{R}_i = R_i - N^{-1}\sum_{j\in\mathcal{B}} R_j$, and the implemented centered relative-score surrogate is

$$\widehat{\delta}_i = \widehat{R}_i + \widehat{\xi}_i,$$

where ξ_i is the uncentered surrogate error, $\widehat{\xi}_i$ is the same error after centering, and ϵ is a uniform error bound satisfying $|\xi_i| \le \epsilon$ before centering. Then $|\widehat{\xi}_i| \le 2\epsilon$ and $\|\widehat{\xi}\|_\mathcal{B} \le 2\epsilon$.

Let $\ell^\delta_{\mathrm{AW}}$ and ℓ^R_{AW} denote the standard group-relative advantage-weighted surrogate evaluated with the implemented surrogate $\widehat{\delta}$ and the ideal centered relative score $\widehat{R}$, respectively. Define $\ell^\delta_{\mathrm{RSPO}}$ and ℓ^R_{RSPO} analogously for the RSPO objective.

For the standard group-relative advantage-weighted surrogate,

$$|\ell^\delta_{\mathrm{AW}} - \ell^R_{\mathrm{AW}}| = |\langle \widetilde{A}, \widehat{\xi}\rangle_\mathcal{B}| \le \|\widetilde{A}\|_\mathcal{B}\,\|\widehat{\xi}\|_\mathcal{B} \le 2\epsilon\|\widetilde{A}\|_\mathcal{B}.$$

For the variance term,

$$\begin{aligned}
|\|\widehat{\delta}\|_\mathcal{B}^2 - \|\widehat{R}\|_\mathcal{B}^2| &\le 2\|\widehat{R}\|_\mathcal{B}\|\widehat{\xi}\|_\mathcal{B} + \|\widehat{\xi}\|_\mathcal{B}^2 \\
&\le 4\epsilon\|\widehat{R}\|_\mathcal{B} + 4\epsilon^2.
\end{aligned}$$

Consequently, for a fixed λ, the forward RSPO loss values satisfy

$$|\ell^\delta_{\mathrm{RSPO}} - \ell^R_{\mathrm{RSPO}}| \le 2\epsilon\|\widetilde{A}\|_\mathcal{B} + \lambda(4\epsilon\|\widehat{R}\|_\mathcal{B} + 4\epsilon^2).$$

This bound controls the scalar surrogate error. Gradient error additionally depends on the Jacobian error of the ELBO estimator and is therefore implementation-dependent.

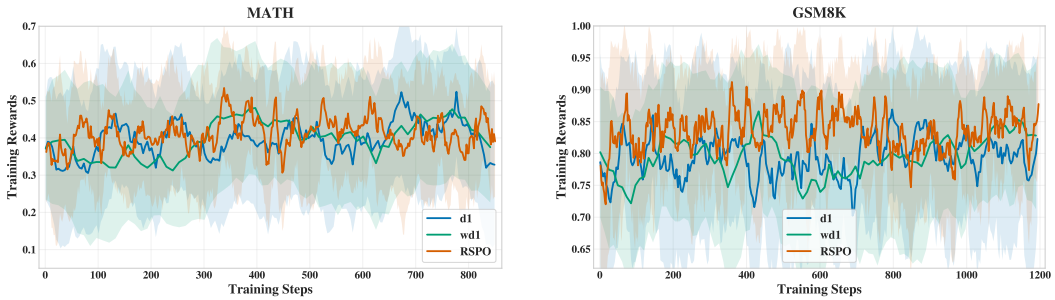


Figure 4: Additional training reward dynamics on mathematical reasoning benchmarks. Shaded regions indicate variation across runs.

F Implementation Notes

Advantage scaling. RSPO uses zero-sum group-relative advantages. Here s_q denotes the prompt-level advantage scale and σ_r denotes the reward standard deviation within the prompt group. The main experiments use the unscaled advantage $s_q = 1$, while the normalized ablation uses $s_q = \sigma_r + 10^{-4}$. When reward scaling is enabled, the sample standard deviation is computed using the training framework's standard convention.

Fixed feedback coefficient and inactive components. The value λ used in every backward pass is fixed rather than adaptively updated; the main experiments use $\lambda = 0.01$, while ablations also test other fixed values. Adaptive updates of λ, reference-model exponential-moving-average (EMA) updates, and removal of zero-standard-deviation reward groups are not used in the reported experiments.

Detached batch centering. The batch mean subtracted from δ is detached before subtraction. This makes the forward values centered but leaves $\nabla_\theta \widehat{\delta}_i = \nabla_\theta \delta_i$ in the backward pass.

Reference-model coupling. For each completion, the current and reference scores share the same mask time and mask set. Without this coupling, the difference δ_i includes unnecessary independent Monte Carlo noise.

Zero-variance reward groups. If the group reward standard deviation is numerically zero, then all rewards in the group are equal and the group has no relative preference signal. Such groups are retained in the reported training runs; the zero-standard-deviation ratio is recorded only as a diagnostic.

G Additional Experimental Details

All reinforcement learning training is implemented with the TRL library (von Werra et al., 2020). We conduct experiments on mathematical reasoning and planning benchmarks, including GSM8K, Math500, Countdown, and Sudoku. All experiments were conducted on 8 NVIDIA H100-80G GPUs

Parameter-Efficient Fine-Tuning. For all experiments, we apply Low-Rank Adaptation (LoRA) (Hu et al., 2022) to the base diffusion language model. The LoRA rank is set to $r = 128$ and the scaling factor is set to $\alpha = 64$. All reported results are obtained with LoRA fine-tuning rather than full-parameter fine-tuning.

Optimization. We optimize the policy with AdamW (Loshchilov & Hutter, 2017) using $\beta_1 = 0.9$ and $\beta_2 = 0.99$. A constant learning-rate schedule is used throughout training. Unless otherwise specified, the learning rate is set to 3×10^{-6}, weight decay to 0.01, and gradient clipping to 0.2. For estimating ELBO-based relative scores, we use $K = 2$ Monte Carlo samples for computational efficiency.

Batching. We set the group size according to task difficulty. For GSM8K, Countdown, and Sudoku, we use group size $G = 6$ and total batch size 96. For MATH500, we use group size $G = 16$ and total batch size 256. Gradient accumulation is applied to reach the target effective batch size.

Training Steps and Checkpoint Selection. Models are trained until the reward curve stabilizes. For mathematical reasoning, we train GSM8K and MATH500 for up to 1.5k steps. For planning tasks, we train models for up to 2k steps. During training, checkpoints are evaluated periodically, and we report the checkpoint with the highest average test accuracy across generation lengths 256 and 512.

Decoding and Evaluation. All experiments are conducted in the zero-shot setting. For both RL rollouts and evaluation, we use semi-autoregressive confidence-based decoding following prior dLLM work. The denoising timestep is set to half of the total generation length, and the sequence is divided into blocks of 32 tokens. At each diffusion step, the model unmasks the two tokens with the highest confidence within the current incomplete block. During RL rollouts, the generation length is set to 256 for all tasks. The sampling temperature is set to 0.9 for GSM8K, MATH500, and Countdown, and to 0.3 for Sudoku. During evaluation, the temperature is set to 0.

Special Notes on Sudoku. Sudoku evaluation follows the verifier-based setting used in prior dLLM planning work. A generated solution is considered correct only if it satisfies the Sudoku constraints and is consistent with the given puzzle. Although some prior work uses few-shot prompting for Sudoku, our reported experiments use zero-shot prompting consistently for both training and evaluation.