ONLINE TIMESTEP ALLOCATION WITH CONTROL VARIATES FOR MASKED DIFFUSION LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Masked diffusion language models exhibit substantial variation across corruption timesteps, mask patterns, and data samples, making uniform timestep sampling potentially inefficient. We propose **I**mportance **S**ampling with **C**ontrol **V**ariates (ISCV), an online, single-forward controller that adapts a full-support timestep proposal using confidence statistics already available in the standard model pass. We show that, with a control variate, the variance-optimal proposal is determined by the second moment of the residualized loss rather than the raw loss. This motivates ISCV-SG, which stop-gradients the correction so that the residual statistics guide future timestep allocation while preserving the expected gradient of the original objective under exact importance weighting. Across five LoRA adaptation tasks, ISCV-SG improves the three-seed mean over uniform training and remains competitive with strong variance-reduction baselines, while retaining runtime close to Standard and requiring approximately half the training time of the strongest baseline P-POTS+MIRROR. Matched ablations and controlled full-parameter pretraining further support the proposed allocation mechanism beyond downstream LoRA adaptation.

1 INTRODUCTION

Masked diffusion language models (MDLMs) (Austin et al., 2021; Lou et al., 2024; Sahoo et al., 2024; Nie et al., 2025b; Ye et al., 2025; von Rütte et al., 2025) have emerged as a promising non-autoregressive alternative to traditional autoregressive language models. Instead of generating tokens strictly from left to right, MDLMs generate text through iterative denoising: tokens are corrupted by masking, and the model learns to recover the missing content from partially observed context. This formulation offers attractive properties for large-scale text generation, including bidirectional context, parallelizable prediction, and flexible refinement. However, as MDLMs scale, their training efficiency is shaped not only by architecture or decoding, but also by the stochastic estimator used for each update. Each update samples a corruption timestep, a mask pattern, and data samples. Different timesteps induce qualitatively different prediction regimes, from local denoising at low noise to near-unconditional infilling at high noise. Consequently, uniform timestep sampling may allocate training inefficiently, as it assigns equal probabilities to regimes whose loss scales, variances, and optimization utility differ substantially.

Recent work addresses this problem from complementary angles. BD3-LMs (Arriola et al., 2025) identify gradient variance as a limiting factor for diffusion language models and reduce it through variance-aware noise schedules and timestep-window strategies. These strategies can emphasize useful corruption regimes, but they may also reduce coverage of other parts of the corruption trajectory. In parallel, Jia et al. (2026) decompose MDLM training variance into masking-rate noise, mask-pattern noise, and data noise. Their P-POTS method estimates a non-uniform timestep sampler from precomputed statistics, while their MIRROR method reduces mask-pattern noise using paired negatively correlated mask patterns. This design is effective, but its timestep sampler is tied to pre-estimated or pilot statistics and therefore cannot easily adapt as the model or loss signals change during training. Moreover, MIRROR introduces additional forward-pass cost relative to a single-forward training update. These approaches show that variance-aware allocation and mask-pattern variance reduction are both effective, but they also leave a practical estimator-design tradeoff: *Can we adapt timestep allocation during training, retain support over the full timestep range, and avoid the extra forward-pass cost of paired mask-pattern evaluations?*

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.

To achieve this goal, we propose **I**mportance **S**ampling with **C**ontrol **V**ariates (ISCV), an online estimator for MDLM training. ISCV adapts two components of the training estimator: a full-support timestep sampler and a lightweight control variate adjustment. The sampler reallocates probability mass across timesteps, while importance weighting preserves the uniform-timestep training objective. The adjustment uses confidence statistics computed from the same forward pass to reduce variation in the scalar training signal. Because this adjustment changes the scalar quantity whose second moment is estimated, it also changes the timestep statistics that should guide sampling. Accordingly, ISCV updates timestep probabilities from adjusted scalar-loss statistics, rather than from fixed pre-estimated statistics or raw timestep losses alone. In the idealized analysis, this coupling leads to a simple allocation rule: For fixed control variate coefficients, the timestep proposal should be governed by the second moment of the adjusted scalar loss at each timestep. In practice, ISCV approximates this quantity online using confidence statistics from the standard training forward pass, such as predictive entropy and probability margin. This allows ISCV to adapt timestep sampling and loss adjustment during training without explicit gradient-moment estimation or additional paired mask-pattern forward passes.

The same online controller supports two backward paths. **ISCV-S** retains the gradient through the parameter-dependent confidence correction and therefore optimizes a proxy-shaped surrogate. **ISCV-SG** detaches the complete correction. With full proposal support and exact importance weights, ISCV-SG preserves the expected parameter gradient of the original objective. The detached correction does not change the current gradient; it affects future updates by changing the proposal learned from adjusted statistics.

We evaluate both paths in compute-constrained LoRA adaptation of large MDLMs, where estimator efficiency directly affects how much useful learning can be obtained from a fixed training budget. We report ISCV-S results and complete three-seed ISCV-SG evaluations on GSM8K, MATH, OpenScience, HiTab, and MBPP. We also include gradient-allocation diagnostics and a controlled full-parameter pretraining experiment. The results are reported with the observed task and seed dependence rather than treating lower scalar variance as a sufficient predictor of final accuracy.

In summary, our work makes the following contributions:

- We formulate MDLM training as a timestep-stratified scalar estimation problem, separating cross-timestep allocation from within-timestep mask and data variability.
- We derive an idealized importance-sampling and control variate principle, showing that the second-moment-minimizing timestep proposal tracks the second moment of the control variate-adjusted scalar loss.
- We propose ISCV, a full-support, single-forward controller that reuses confidence statistics from the standard training pass to update the timestep proposal and construct the control-variate adjustment online. We distinguish its differentiable ISCV-S path from the target-preserving ISCV-SG path.
- We evaluate ISCV-S and ISCV-SG on reasoning and coding tasks, and provide gradient-allocation and full-parameter diagnostics while reporting the observed task and seed dependence.

2 Related Work

Masked and Discrete Diffusion Language Models. Diffusion-based language modeling extends denoising diffusion from continuous spaces to discrete token vocabularies. Foundational works established categorical and absorbing-state diffusion formulations (Hoogeboom et al., 2021; Austin et al., 2021), while later language-focused methods developed masked and discrete diffusion objectives for text modeling (Sahoo et al., 2024; Lou et al., 2024; Shi et al., 2024; Ou et al., 2025; Von Rütte et al., 2025). More recent work scales these ideas to large language models (Nie et al., 2025b; Gong et al., 2025a; Ye et al., 2025; Sahoo et al., 2026) and studies related generation settings such as blockwise generation (Arriola et al., 2025), reproducible dLLM training and evaluation frameworks (Zhou et al., 2026), code generation and diffusion-native RL (Gong et al., 2025b), task-specific fine-tuning for mathematical and logical patterns (Chen & Chen, 2025), and soft masked decoding variants (Hersche et al., 2025). Our work does not propose a new diffusion parameterization or decoding scheme. Instead, we study the stochastic *training estimator* induced by masked diffusion language model training.

Variance Reduction in Diffusion Training. Variance reduction is a recurring issue in diffusion training. In continuous diffusion, Nichol & Dhariwal (2021) observed that some variational training objectives induce substantially noisier gradients and showed that importance sampling can reduce this noise. Related work studies timestep-dependent gradient variance and adaptive timestep sampling in diffusion training (Kim et al., 2025), as well as variance-reduced score estimation and sampling in continuous diffusion models (Duston & Bui-Thanh, 2026). In language diffusion, Arriola et al. (2025) study objective and gradient variance and propose variance-aware noising schedules. Closest to our setting, Jia et al. (2026) decompose masked diffusion training variance into masking-rate noise, mask-pattern noise, and data noise, and propose source-specific remedies including the P-POTS timestep sampler and MIRROR for reducing mask-pattern noise. Our work is complementary to these methods. Rather than emphasizing selected corruption regimes through schedule or window design, using a pre-estimated timestep proposal, or evaluating paired complementary mask patterns, ISCV provides an online, full-support, single-forward estimator controller. Its timestep sampler tracks post-adjusted statistics rather than pre-adjustment timestep statistics, adapting the proposal using online proxies for the second moment of the control-variate-adjusted scalar loss.

Estimator Design, Proxy Signals, and Neighboring Methods. ISCV views MDLM training through the lens of estimator design rather than model reparameterization. It builds on classical importance sampling, control variates, and stratified Monte Carlo estimation (Shapiro, 2003; Glasserman, 2004), and relates to adaptive importance sampling and control variate methods (Leluc et al., 2022; He et al., 2024). In MDLM training, timesteps provide natural strata, while mask patterns and data samples induce substantial within-stratum variability. ISCV exploits this structure with online timestep sampling and control variate correction. Its confidence statistics, such as predictive entropy and probability margin, serve only as lightweight proxies for residual difficulty, connecting to work on token-level heterogeneity and uncertainty (Kapoor et al., 2024; Zhang et al., 2025; Li et al., 2025a) and entropy-based gradient allocation (Hu et al., 2026), but without requiring calibrated uncertainty estimates. ISCV is complementary to methods that intervene later in the modeling pipeline, including post-training, preference optimization, and reinforcement learning (Zhu et al., 2025; Gong et al., 2025b; Zhan, 2025; Rojas et al., 2026), generation-order learning (Fox et al., 2026), and inference-time guidance, sampling, or compute-allocation methods (Nie et al., 2025a; Schiff et al., 2025; Li et al., 2025b; Uehara et al., 2025; Luo et al., 2026; Liu et al., 2025). These directions target output quality, controllability, or sampling efficiency, whereas ISCV targets the sample and computational efficiency of the supervised MDLM training estimator.

3 METHOD

Loss variance versus gradient variance. We analyze ISCV at the level of the stochastic scalar loss estimator. Before automatic differentiation, masked diffusion training constructs a scalar objective by sampling a corruption timestep, a mask pattern, and data examples. The timestep proposal derived below is designed to minimize the second moment of this scalar estimator. This is not a direct gradient-variance optimality claim. Following the discussion of loss versus gradient variance in Jia et al. (2026), we focus on scalar loss-estimator variance because it is independent of the optimizer choice, avoids arbitrary scalarization of high-dimensional gradient vectors, and is the component most directly affected by timestep sampling. This choice is also practical in compute-constrained settings, since scalar losses and confidence proxies are available from the standard forward pass, whereas reliable gradient-moment estimates would require additional costly computation. We therefore treat scalar variance reduction as the primary theoretical objective and report gradient-side quantities only as supplementary diagnostics.

3.1 OBJECTIVE AND IDEAL ISCV PRINCIPLE

For a scalar random variable z and a zero-mean control c, the classical control-variate estimator $z_\beta = z - \beta c$ has

$$\beta^\star = \frac{\mathrm{Cov}(z,c)}{\mathrm{Var}(c)}, \qquad \mathrm{Var}(z_{\beta^\star}) = \mathrm{Var}(z)(1-\rho_{z,c}^2),$$

when $\mathrm{Var}(c) > 0$ (Glasserman, 2004; Shapiro, 2003). ISCV applies this standard identity within corruption-timestep strata. Its method-specific role is that the adjustment also changes the per-timestep second moment used by the online allocation rule.

Let $k \in \{1, \dots, K\}$ index a discretized corruption timestep, $x_0 \sim \mathcal{D}$ a clean token sequence, and $m \sim p(\cdot \mid x_0, k)$ a binary mask pattern. The corrupted input $x_k = C(x_0, m)$ is defined tokenwise by

$$(x_k)_j = \begin{cases} [\text{MASK}], & m_j = 1, \\ (x_0)_j, & m_j = 0. \end{cases}$$

We write $M = \{j : m_j = 1\}$ for the supervised masked positions. With model parameters θ, let $\ell_\theta(x_0, m; k)$ be the per-sample masked-token training loss. Consider the training objective:

$$\mathcal{L}(\theta) = \frac{1}{K} \sum_{k=1}^{K} \mathbb{E}[\ell_\theta(x_0, m; k) \mid k]. \tag{1}$$

We view this objective as timestep-stratified scalar estimation: Each timestep defines a stratum, the outer average defines the target weighting across strata, and the conditional expectation captures within-timestep variability.[1]

Given a proposal probability mass function q with full support (i.e., for all k $q(k) > 0$), the importance weight for this uniform target in (1) is $w(k) = 1/(Kq(k))$. Changing q changes how minibatch samples are allocated across timestep strata, while the importance weight preserves Eq. (1). ISCV does not change Eq. (1); it changes how this objective is estimated by adapting the timestep sampling distribution and adjusting the scalar loss estimator.

Let $h_\theta(x_0, m; k)$ be a scalar confidence statistic extracted from the forward pass in standard training. Define the centered control variate

$$c_\theta(x_0, m; k) = h_\theta(x_0, m; k) - \mu_{h,k},$$

with $\mu_{h,k} = \mathbb{E}[h_\theta(x_0, m; k) \mid k]$, so that $\mathbb{E}[c_\theta \mid k] = 0$. This is the standard regression control variate construction (Glasserman, 2004; Shapiro, 2003).

Let $\{\beta_k\}$ denote the control-variate coefficient for timestep k, and let $\beta = \{\beta_k\}_{k=1}^K$, the ideal adjusted loss and ISCV estimator of $\mathcal{L}(\theta)$ are

$$\tilde{\ell}_{\theta,\beta} = \ell_\theta - \beta_k c_\theta, \qquad \widehat{\mathcal{L}}_{\theta,q,\beta} = \frac{1}{Kq(k)} \tilde{\ell}_{\theta,\beta}, \qquad k \sim q.$$

Under exact centering and full importance weighting, $\widehat{\mathcal{L}}_{\theta,q,\beta}$ is unbiased for $\mathcal{L}(\theta)$, with expectation over $k \sim q$, $x_0 \sim \mathcal{D}$, and $m \sim p(\cdot \mid x_0, k)$. The proof and boundary conditions are given in Appendix A.2.

For fixed timestep-specific coefficients β, define the timestep-conditioned adjusted-loss second moment

$$R_k(\beta) = \mathbb{E}[\tilde{\ell}^2_{\theta,\beta} \mid k].$$

This is a second moment after control variate adjustment, not a centered variance. Since the estimator second moment is

$$\mathbb{E}[\widehat{\mathcal{L}}^2_{\theta,q,\beta}] = \frac{1}{K^2} \sum_{k=1}^{K} \frac{R_k(\beta)}{q(k)},$$

and the estimator is unbiased, subtracting $\mathcal{L}(\theta)^2$ gives the variance; this term is independent of q. The second-moment-minimizing proposal is therefore

$$q^\star(k) = \frac{\sqrt{R_k(\beta)}}{\sum_{j=1}^{K} \sqrt{R_j(\beta)}}.$$

The allocation rule is the standard second-moment importance-sampling rule, applied here to the control-variate-adjusted MDLM scalar estimator. The derivation and the variance decomposition are in Appendix A.3 and Appendix A.6, respectively. Thus, after a scalar control-variate adjustment, timestep allocation should track the adjusted-loss second moment rather than raw loss statistics before adjustment.

[1] Unless otherwise stated, conditional expectations and variances given k are over $x_0 \sim \mathcal{D}$ and $m \sim p(\cdot \mid x_0, k)$, with θ and estimator/controller quantities held fixed.

The ideal coefficient minimizing the within-timestep variance of the adjusted loss is the usual regression coefficient

$$\beta_k^\star = \frac{\mathrm{Cov}(\ell_\theta, h_\theta \mid k)}{\mathrm{Var}(h_\theta \mid k)}. \tag{2}$$

This is the standard regression control-variate coefficient; here it is used as a timestep-conditioned component of the MDLM estimator. The corresponding variance reduction term is $\mathrm{Cov}(\ell_\theta, h_\theta \mid k)^2/\mathrm{Var}(h_\theta \mid k)$; hence it vanishes when the proxy is uninformative. Standard boundary cases are summarized in Appendix A.

3.2 Online ISCV Controller

The ideal quantities above are population moments at a fixed model state, so ISCV approximates them online using detached controller state. During the standard training forward pass, the model already produces logits at supervised masked positions. Our default proxy is the average predictive entropy

$$h_{\mathrm{ent}}(x_0, m; k) = \frac{1}{|M|} \sum_{i \in M} \mathcal{H}(p_\theta(\cdot \mid x_k)_i)\,,$$

where $p_\theta(\cdot \mid x_k)_i$ is the predictive token distribution at masked position i. We also evaluate probability margin and one-minus-maximum-probability variants. Let $p_{(1),i}$ and $p_{(2),i}$ denote the largest and second-largest predicted token probabilities at masked position i. We use

$$h_{\mathrm{margin}} = \frac{1}{|M|} \sum_{i \in M} \big(1 - (p_{(1),i} - p_{(2),i})\big), \qquad h_{\mathrm{maxprob}} = \frac{1}{|M|} \sum_{i \in M} (1 - p_{(1),i}).$$

The proxy is not assumed to be calibrated; we use it as a lightweight signal intended to correlate with scalar loss variation within a timestep.

To stabilize estimation of the control-variate coefficient, adjacent timesteps are pooled into coefficient groups $g(k) \in \{1, \dots, G\}$. Importantly, proxy centering remains timestep-specific. ISCV maintains a separate EMA estimate $\widehat{\mu}_{h,k}$ of the proxy mean for each timestep k, while the moments used to estimate the regression coefficient are pooled within each coefficient group.

For $s \in \{\ell, h, \ell h, h^2\}$, let $\widehat{m}_{s,g}$ denote the corresponding grouped EMA moment and $\bar{s}_g$ its current observed mean. For each observed group g, the controller updates

$$\widehat{m}_{s,g} \leftarrow (1 - \eta_\beta)\widehat{m}_{s,g} + \eta_\beta \bar{s}_g, \qquad \eta_\beta \in (0, 1]. \tag{3}$$

It then forms

$$\widehat{\mathrm{Cov}}_g(\ell, h) = \widehat{m}_{\ell h, g} - \widehat{m}_{\ell, g}\widehat{m}_{h, g}, \tag{4}$$

$$\widehat{\mathrm{Var}}_g(h) = \max\big\{\widehat{m}_{h^2, g} - \widehat{m}_{h, g}^2, \epsilon_{\mathrm{var}}\big\}\,, \tag{5}$$

and sets

$$\beta_g = \frac{\widehat{\mathrm{Cov}}_g(\ell, h)}{\widehat{\mathrm{Var}}_g(h) + \lambda_{\mathrm{ridge}}}. \tag{6}$$

Here η_β is the regression-moment EMA update rate, $\epsilon_{\mathrm{var}} > 0$ is a variance floor, and $\lambda_{\mathrm{ridge}} > 0$ is a ridge stabilization constant. The reported configurations use $\eta_\beta = 0.02$, $\epsilon_{\mathrm{var}} = 10^{-4}$, and $\lambda_{\mathrm{ridge}} = 10^{-3}$.

Thus, grouping is used only to stabilize coefficient estimation; it does not change the timestep-specific centering required by the ideal control-variate construction. The practical coefficient $\beta_{g(k)}$ is shared among nearby timesteps and need not equal the population-optimal per-timestep coefficient $\beta_k^\star$.

For sample i, ISCV uses the timestep-specific centered proxy

$$c_i = h_i - \widehat{\mu}_{h, k_i}\,, \tag{7}$$

and forms the adjusted scalar used for proposal adaptation:

$$\widetilde{\ell}_i^{\mathrm{prop}} = \ell_i - \beta_{g(k_i)} c_i. \tag{8}$$

Given a sample i, ISCV centers the proxy and forms the adjusted scalar used for proposal adaptation:

$$c_i = h_i - \widehat{\mu}_{h,k_i}, \qquad \tilde{\ell}_i^{\text{prop}} = \ell_i - \beta_{g(k_i)} c_i.$$

The main proposal statistic is the squared adjusted scalar $(\tilde{\ell}_i^{\text{prop}})^2$. For each timestep k observed in the minibatch, let

$$\mathcal{B}_k = \{i : k_i = k\}, \qquad \bar{R}_k^{\text{obs}} = \frac{1}{|\mathcal{B}_k|} \sum_{i \in \mathcal{B}_k} \left(\tilde{\ell}_i^{\text{prop}}\right)^2.$$

For observed timesteps, ISCV updates the online second-moment estimate by

$$\widehat{R}_k \leftarrow (1 - \eta_R)\widehat{R}_k + \eta_R \bar{R}_k^{\text{obs}}, \qquad |\mathcal{B}_k| > 0.$$

Here $\eta_R \in (0, 1]$ is the EMA update rate for the timestep-conditioned second moment. Equivalently, for a configured decay ρ_R, we use $\eta_R = 1 - \rho_R$; the task-specific values are reported in the resolved configurations. This is the direct online analogue of $R_k(\beta)$. Raw-loss proposal statistics are used as pre-adjustment ablations, while importance-weighted adjusted-loss statistics are reported as diagnostics and, when used, as practical variants.

ISCV then forms

$$q_{\text{target}}(k) = \frac{\sqrt{\widehat{R}_k + \epsilon}}{\sum_{j=1}^{K} \sqrt{\widehat{R}_j + \epsilon}}, \qquad q_{\text{mix}}(k) = (1 - \alpha_q) q_{\text{target}}(k) + \alpha_q / K.$$

Here $\epsilon > 0$ stabilizes the square-root statistic when $\widehat{R}_k$ is small, while $\alpha_q \in [0, 1]$ mixes the adaptive target with uniform sampling to keep exploration over timesteps and limit proposal concentration. The mixed proposal is then floored and renormalized:

$$q_{\text{train}}(k) = \frac{\max(q_{\text{mix}}(k), q_{\min})}{\sum_{j=1}^{K} \max(q_{\text{mix}}(j), q_{\min})}.$$

We use q_{train} for subsequent timestep sampling. The floor $q_{\min}$ preserves full support and avoids extremely large importance weights caused by very small proposal probabilities.

3.3 Training Update and Stabilization

For each minibatch, ISCV samples timesteps from the current q_{train}, samples masks, and uses a single standard forward pass to compute both losses ℓ_i and proxies h_i. The proposal used for the current minibatch is computed before the batch, and detached statistics from the batch update the controller only for future minibatches.

Let

$$w_i = \frac{1}{K q_{\text{train}}(k_i)}, \qquad c_i = h_i - \widehat{\mu}_{h,k_i}. \tag{9}$$

The two variants share the controller above but pass different scalars to automatic differentiation.

ISCV-SG uses

$$s_i^{\text{SG}} = w_i \ell_i - \text{sg}\left(w_i \beta_{g(k_i)} c_i\right), \tag{10}$$

where $\text{sg}(\cdot)$ denotes stop-gradient. Conditional on the controller state available before the minibatch, full proposal support and exact importance weighting give

$$\mathbb{E}\left[\nabla_\theta s_i^{\text{SG}}\right] = \frac{1}{K} \sum_{k=1}^{K} \mathbb{E}\left[\nabla_\theta \ell_\theta \mid k\right]. \tag{11}$$

Thus, the detached correction does not directly change the current parameter gradient; it affects future updates through the proposal learned from the adjusted statistics.

ISCV-S instead uses

$$s_i^{\text{S}} = w_i \left(\ell_i - \beta_{g(k_i)} c_i\right). \tag{12}$$

Because h_θ is parameter dependent, while $\widehat{\mu}_{h,k}$, β_g, and the controller state are detached, ISCV-S includes the additional gradient term

$$-w_i \beta_{g(k_i)} \nabla_\theta h_\theta, \tag{13}$$

and therefore optimizes a proxy-shaped surrogate rather than the original objective.

For either path, the minibatch objective is

$$\widehat{\mathcal{L}}_{\text{batch}} = \frac{1}{B} \sum_{i=1}^{B} s_i.$$

All controller statistics and proposal parameters are detached. The reported ISCV-SG runs use exact, unclipped importance weights; task-specific correction safeguards are defined in Appendix A.9. The main update is summarized in Algorithm 1.

4 Experiments

We evaluate matched-budget task performance, compare the two ISCV backward paths, examine scalar and gradient-allocation diagnostics, and test full-parameter pretraining beyond LoRA. Detailed protocols, stability checks, and supplementary ablations appear in Appendix B.

4.1 Setup

We evaluate response-only LoRA adaptation of LLaDA-8B-Instruct (Nie et al., 2025b) with $K = 64$ timesteps on GSM8K, MATH, OpenScienceReasoning-2, HiTab, and MBPP. Main task results use seeds 42, 731, and 20231. Within each task, methods share the backbone, data splits, initialization protocol, optimizer schedule, and token/update budgets. Evaluation uses greedy generation and task-specific metrics; MATH uses the complete 5,000-example test set.

Baselines are **Standard** (uniform timesteps), **Clipped**, a clipped timestep-window baseline (Arriola et al., 2025), and **P-POTS+MIRROR** (paired-mask variance reduction). **ISCV-S** differentiates through the confidence correction; **ISCV-SG** fully detaches it and preserves the original expected gradient under exact importance weighting. Both use the controller in Section 3.2. We report measured runtime separately because matched token/update budgets do not imply equal model-evaluation cost. Appendix B provides adaptation settings, dataset protocols, and compute accounting; Section 4.4 additionally evaluates full-parameter pretraining.

4.2 Main Task Performance and Runtime

We first evaluate whether the target-preserving ISCV-SG variant improves downstream task performance under matched token and update budgets. Table 1 reports per-seed task performance and the available average training times for the main methods.

Across the five datasets with reported three-seed results, ISCV-SG improves over Standard by 4.02, 2.65, 5.24, 3.18, and 1.68 points on GSM8K, MATH, HiTab, OpenScience, and MBPP, respectively. Its corresponding margins over the strongest listed alternative are $+0.20$, $+0.94$, $+0.40$, -0.80, and $+0.65$ points, indicating competitive but task-dependent performance.

Across the four tasks with reported timings, ISCV-SG remains close to Standard in wall-clock training time and takes approximately half the time of P-POTS+MIRROR at matched token/update budgets.

Table 2 separates timestep allocation and control-variate adjustment on GSM8K. IS and CV each improve over Standard; their combination in ISCV-S gives the highest three-seed mean.

Table 3 gives a strict matched comparison of the differentiable ISCV-S and detached ISCV-SG paths. Within each pair, all training, controller, evaluation, and checkpoint-selection settings are fixed; only `cv_stopgrad` changes.

ISCV-S is higher on every GSM8K and HiTab seed; OpenScience and MBPP show mixed seed-wise comparisons. The strict matched ablation shows that differentiating through the correction can affect downstream outcomes, although its benefit is not uniform across tasks or seeds.

Table 1: Task performance (%) and average wall-clock training time. *Avg.* and *Time Avg.* denote averages over the available runs; "–" denotes unreported entries. Bold indicates the best value within each dataset/seed column. Per-run details and timing conventions are in Appendix B.

Dataset	Method	Seed 42	Seed 731	Seed 20231	Avg.	Time Avg. (h)
GSM8K	Standard	61.87	60.34	62.70	61.64	1.49
	Clipped	64.06	63.68	64.52	64.09	2.78
	P-POTS+MIRROR	65.73	64.14	**66.49**	65.45	3.35
	ISCV-SG	**67.63**	**65.05**	64.29	**65.66**	1.51
MATH	Standard	35.02	33.64	34.50	34.39	3.06
	Clipped	35.16	**36.40**	36.72	36.09	3.64
	P-POTS+MIRROR	35.28	34.82	35.24	35.11	6.14
	ISCV-SG	**38.54**	35.14	**37.42**	**37.03**	3.08
HiTab	Standard	52.52	60.10	57.08	56.57	3.82
	Clipped	**61.99**	61.04	58.90	60.64	4.15
	P-POTS+MIRROR	61.17	61.93	61.11	61.40	7.88
	ISCV-SG	61.50	**62.05**	**61.86**	**61.80**	3.84
OpenScience	Standard	64.95	63.60	62.90	63.82	6.02
	Clipped	63.87	65.90	64.60	64.79	7.37
	P-POTS+MIRROR	**68.30**	**67.40**	**67.70**	**67.80**	13.04
	ISCV-SG	67.10	67.30	66.60	67.00	6.08
MBPP	Standard	40.86	40.86	39.69	40.47	–
	Clipped	**41.25**	40.86	41.63	41.25	–
	P-POTS+MIRROR	40.86	42.02	41.63	41.50	–
	ISCV-SG	40.47	**42.41**	**43.58**	**42.15**	–

Table 2: GSM8K estimator components across three seeds. IS and CV each improve over Standard; ISCV-S achieves the highest score on every seed.

Method	Seed 42	Seed 731	Seed 20231	Avg.
Standard	61.87	60.34	62.70	61.64
IS	64.97	66.34	65.73	65.68
CV	68.01	67.17	66.34	67.17
ISCV-S	**68.39**	**67.24**	**67.32**	**67.65**

Table 3: Three-seed task performance of ISCV-S and ISCV-SG.

Dataset	Method	Seed 42	Seed 731	Seed 20231	Avg.
GSM8K	ISCV-S	68.39	67.24	67.32	67.65
	ISCV-SG	67.63	65.05	64.29	65.66
HiTab	ISCV-S	62.02	63.39	63.26	62.89
	ISCV-SG	61.50	62.05	61.86	61.80
OpenScience	ISCV-S	69.60	67.20	68.50	68.43
	ISCV-SG	67.10	67.30	66.60	67.00
MBPP	ISCV-S	43.19	42.41	42.80	42.80
	ISCV-SG	40.47	42.41	43.58	42.15

4.3 Estimator and Proposal Diagnostics

On GSM8K, ISCV-S has lower final weighted scalar variance than CV (0.0952 versus 0.1176), although CV has lower adjusted variance (0.1176 versus 0.1874). These loss-space diagnostics therefore do not monotonically predict downstream accuracy; complete results are reported in Appendix B.10.

Table 4: Scalar-estimator diagnostics on GSM8K (ISCV-S factorial). Raw, Adjusted, and Weighted denote loss-space variances; Weighted includes importance weighting. Correlations are shown for proxy-based methods. These are not gradient variances.

Method	Raw	Adjusted	Weighted	Loss corr.	Post-adj. corr.
Standard	0.2733	0.2733	0.2733	N/A	N/A
IS	0.2811	0.2811	0.1116	N/A	N/A
CV	0.2493	0.1176	0.1176	0.3937	0.3872
ISCV-S	0.2430	0.1874	0.0952	0.2291	0.1803

Gradient-allocation evidence. We probe all 6.29M trainable LoRA parameters at 27 checkpoints along one seed-42 ISCV-SG trajectory. The corrected-loss proposal is closer to the grouped gradient-second-moment-optimal target $q^{\star}_{\mathrm{grad}}$ than the raw-loss proposal (Table 5). Appendix B.8 defines the target and probe protocol.

Table 5: Checkpoint-local gradient-allocation diagnostic. The predicted importance-weighted gradient second moment is normalized by uniform sampling; lower ratios are better.

Proposal statistic	Median cosine with $q^{\star}_{\mathrm{grad}}$	Predicted moment ratio
Raw loss	0.743	0.837
Corrected loss	**0.942**	**0.586**

All 27 corrected-proposal moment ratios are below one. This is checkpoint-local allocation evidence along one AdamW trajectory; it does not establish trajectory-wide gradient-covariance reduction or determine downstream accuracy.Proposal-stability diagnostics are reported in Appendix **??**.

4.4 Beyond LoRA: Full-Parameter Pretraining

Beyond LoRA, we pretrain all parameters of a 169M-parameter DiT-backbone MDLM on OpenWebText. At matched token and update budgets, ISCV-SG achieves lower generative perplexity for both seeds at 500M, 750M, and 1B tokens, reaching a two-seed mean of 73.18 versus 74.97 for Standard at 1B (Table 6). The 250M result is mixed. These two trajectories extend the evidence beyond adapter training; wall-clock equivalence is not established. Appendix B.4 provides training-match details and checkpoint-level uncertainty.

Table 6: Generative perplexity during full-parameter pretraining at matched token budgets; lower is better. Bold marks the lower value within each same-seed checkpoint pair.

Budget	Standard 42	ISCV-SG 42	Standard 731	ISCV-SG 731
250M	100.09	**93.37**	**101.33**	108.85
500M	81.35	**79.34**	88.15	**87.84**
750M	75.75	**73.03**	84.15	**76.36**
1B	71.49	**71.35**	78.46	**75.01**

5 Conclusion

We framed masked diffusion language model training as a timestep-stratified scalar-estimation problem and showed that a control-variate adjustment changes the second moment that should determine the proposal. ISCV implements this principle online using confidence statistics from the standard forward pass. ISCV-SG preserves the original expected gradient under full-support sampling and exact importance weighting, whereas ISCV-S optimizes a proxy-shaped surrogate. Across five LoRA tasks, ISCV-SG improves over Standard on average, while comparisons with the strongest alternative remain task dependent. Gradient-allocation and full-parameter results provide complementary but seed- and budget-dependent evidence. We therefore treat scalar variance as an allocation diagnostic, not a monotone predictor of accuracy; limitations are detailed in Appendix C.

REPRODUCIBILITY STATEMENT

We provide complete derivations, implementation details, resolved configurations, per-seed results, evaluation protocols, and additional diagnostics in the appendices. An anonymized code artifact containing the training and evaluation implementation, configuration files, and commands needed to reproduce the principal experiments is provided with the submission.

AI USE STATEMENT

Generative AI tools were used to assist with generating outlines, checking paper formatting, polishing English prose, and polishing code. All AI-assisted text, formatting suggestions, and code edits were reviewed and verified by the authors. AI tools were not used to generate experimental results or statistical analyses. The authors take responsibility for the final content of this work, including all claims, code, and artifacts.

REFERENCES

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 *International Conference on Learning Representations*, 2025.

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.

Ranfei Chen and Ming Chen. Dsft: Inspiring diffusion large language models to comprehend mathematical and logical patterns. *arXiv preprint arXiv:2509.18164*, 2025.

Alois Duston and Tan Bui-Thanh. Variance-reduced diffusion sampling via conditional score expectation identity. *arXiv preprint arXiv:2601.01594*, 2026.

David Fox, Sam Bowyer, Song Liu, Laurence Aitchison, Raul Santos-Rodriguez, and Mengyue Yang. Learning generation orders for masked discrete diffusion models via variational inference. In *ICLR 2026 2nd Workshop on Deep Generative Model in Machine Learning: Theory, Principle and Efficacy*, 2026.

Paul Glasserman. *Monte Carlo methods in financial engineering*, volume 53. Springer, 2004.

Shansan Gong, Shivam Agarwal, Yizhe Zhang, Jiacheng Ye, Lin Zheng, Mukai Li, Chenxin An, Peilin Zhao, Wei Bi, Jiawei Han, et al. Scaling diffusion language models via adaptation from autoregressive models. *The Thirteenth International Conference on Learning Representations*, 2025a.

Shansan Gong, Ruixiang Zhang, Huangjie Zheng, Jiatao Gu, Navdeep Jaitly, Lingpeng Kong, and Yizhe Zhang. Diffucoder: Understanding and improving masked diffusion models for code generation. *arXiv preprint arXiv:2506.20639*, 2025b.

Shengyi He, Guangxin Jiang, Henry Lam, and Michael C Fu. Adaptive importance sampling for efficient stochastic root finding and quantile estimation. *Operations Research*, 72(6):2612–2630, 2024.

Michael Hersche, Samuel Moor-Smith, Thomas Hofmann, and Abbas Rahimi. Soft-masked diffusion language models. *arXiv preprint arXiv:2510.17206*, 2025.

Emiel Hoogeboom, Didrik Nielsen, Priyank Jaini, Patrick Forré, and Max Welling. Argmax flows and multinomial diffusion: Learning categorical distributions. *Advances in neural information processing systems*, 34:12454–12465, 2021.

Yuelin Hu, Zhengxue Cheng, Wei Liu, and Li Song. Entropy-gated selective policy optimization: Token-level gradient allocation for hybrid training of large language models. *arXiv preprint arXiv:2602.03309*, 2026.

Mengni Jia, Mengyu Zhou, Yihao Liu, Xiaoxi Jiang, and Guanjun Jiang. Bringing stability to diffusion: Decomposing and reducing variance of training masked diffusion models. In *International Conference on Learning Representations*, 2026.

Sanyam Kapoor, Nate Gruver, Manley Roberts, Katherine Collins, Arka Pal, Umang Bhatt, Adrian Weller, Samuel Dooley, Micah Goldblum, and Andrew G Wilson. Large language models must be taught to know what they don't know. *Advances in Neural Information Processing Systems*, 37: 85932–85972, 2024.

Myunsoo Kim, Donghyeon Ki, Seong Woong Shim, and Byung Jun Lee. Adaptive non-uniform timestep sampling for accelerating diffusion model training. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 2513–2522. IEEE Computer Society, 2025.

Rémi Leluc, François Portier, Johan Segers, and Aigerim Zhuman. A quadrature rule combining control variates and adaptive importance sampling. *Advances in neural information processing systems*, 35:11842–11853, 2022.

Rui Li, Jing Long, Muge Qi, Heming Xia, Lei Sha, Peiyi Wang, and Zhifang Sui. Towards harmonized uncertainty estimation for large language models. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 22938–22953, Vienna, Austria, July 2025a. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.1118. URL https://aclanthology.org/2025.acl-long.1118/.

Xiner Li, Yulai Zhao, Chenyu Wang, Gabriele Scalia, Gökcen Eraslan, Surag Nair, Tommaso Biancalani, Shuiwang Ji, Aviv Regev, Sergey Levine, et al. Derivative-free guidance in continuous and discrete diffusion models with soft value-based decoding. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025b.

Jiacheng Liu, Chang Zou, Yuanhuiyi Lyu, Junjie Chen, and Linfeng Zhang. From reusing to forecasting: Accelerating diffusion models with taylorseers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 15853–15863, 2025.

Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 32819–32848, 2024.

Ziwei Luo, Ziqi Jin, Lei Wang, Lidong Bing, and Thomas B Schön. Self-rewarding sequential monte carlo for masked diffusion language models. *arXiv preprint arXiv:2602.01849*, 2026.

Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International conference on machine learning*, pp. 8162–8171. PMLR, 2021.

Shen Nie, Fengqi Zhu, Chao Du, Tianyu Pang, Qian Liu, Guangtao Zeng, Min Lin, and Chongxuan Li. Scaling up masked diffusion models on text. *The Thirteenth International Conference on Learning Representations*, 2025a.

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. *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025b.

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*, 2025.

Kevin Rojas, Jiahe Lin, Kashif Rasul, Anderson Schneider, Yuriy Nevmyvaka, Molei Tao, and Wei Deng. Improving reasoning for diffusion language models via group diffusion policy optimization. In *International Conference on Learning Representations*, 2026.

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. In *Advances in Neural Information Processing Systems*, volume 37, pp. 130136–130184, 2024.

Subham Sekhar Sahoo, Jean-Marie Lemercier, Zhihan Yang, Justin Deschenaux, Jingyu Liu, John Thickstun, and Ante Jukic. Scaling beyond masked diffusion language models. *arXiv preprint arXiv:2602.15014*, 2026.

Yair Schiff, Subham Sekhar Sahoo, Hao Phung, Guanghan Wang, Alexander Rush, Volodymyr Kuleshov, Hugo Dalla-Torre, Sam Boshar, Bernardo P de Almeida, and Thomas Pierrot. Simple guidance mechanisms for discrete diffusion models. In ... *International Conference on Learning Representations*, volume 2025, pp. 44153, 2025.

Alexander Shapiro. Monte carlo sampling methods. *Handbooks in operations research and management science*, 10:353–425, 2003.

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.

Masatoshi Uehara, Yulai Zhao, Chenyu Wang, Xiner Li, Aviv Regev, Sergey Levine, and Tommaso Biancalani. Reward-guided controlled generation for inference-time alignment in diffusion models: Tutorial and review. *arXiv preprint arXiv:2501.09685*, 2025.

Dimitri Von Rütte, Janis Fluri, Yuhui Ding, Antonio Orvieto, Bernhard Schölkopf, and Thomas Hofmann. Generalized interpolating discrete diffusion. In *International Conference on Machine Learning*, pp. 61810–61843. PMLR, 2025.

Dimitri von Rütte, Janis Fluri, Omead Pooladzandi, Bernhard Schölkopf, Thomas Hofmann, and Antonio Orvieto. Scaling behavior of discrete diffusion language models. *arXiv preprint arXiv:2512.10858*, 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.

Anthony Zhan. Principled and tractable rl for reasoning with diffusion language models. *arXiv preprint arXiv:2510.04019*, 2025.

Tunyu Zhang, Haizhou Shi, Yibin Wang, Hengyi Wang, Xiaoxiao He, Zhuowei Li, Haoxian Chen, Ligong Han, Kai Xu, Huan Zhang, et al. Tokur: Token-level uncertainty estimation for large language model reasoning. *arXiv preprint arXiv:2505.11737*, 2025.

Zhanhui Zhou, Lingjie Chen, Hanghang Tong, and Dawn Song. dllm: Simple diffusion language modeling. *arXiv preprint arXiv:2602.22661*, 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.

A Theoretical Analysis

The main paper states the estimator analysis at the scalar-objective level. The gradient-level results below are supplementary analogues and diagnostics; they are not required for the scalar ISCV result used by the practical controller.

A.1 Definitions and Notation

We analyze the scalar estimator at a fixed model state θ. Let $k \in \{1, \dots, K\}$ index a discretized corruption level, which we call a timestep throughout. Let $x_0 \sim \mathcal{D}$ denote a clean training sample, and let $m \sim p(\cdot \mid x_0, k)$ denote the corruption mask sampled at timestep k. We write the per-sample masked diffusion loss as

$$\ell_\theta(x_0, m; k) \in \mathbb{R}.$$

Throughout this appendix, conditional expectations and variances given k are taken over $x_0 \sim \mathcal{D}$ and $m \sim p(\cdot \mid x_0, k)$, with θ, the proposal q, and the coefficients β held fixed.

Uniform-timestep objective. Define the conditional mean loss

$$\mu_k(\theta) \triangleq \mathbb{E}[\ell_\theta(x_0, m; k) \mid k]\,.$$

Because the analysis fixes θ, we write μ_k for $\mu_k(\theta)$ when no confusion can arise. The target scalar training objective is the uniform-timestep average

$$\mathcal{L}(\theta) \triangleq \frac{1}{K}\sum_{k=1}^{K} \mu_k = \mathbb{E}_{k\sim U(\{1,\dots,K\})}\mathbb{E}[\ell_\theta(x_0, m; k) \mid k]\,. \tag{14}$$

Proxy and centered control variate. Let $h_\theta(x_0, m; k)$ be a proxy statistic for within-timestep scalar loss. Define its conditional mean

$$\mu_{h,k} \triangleq \mathbb{E}[h_\theta(x_0, m; k) \mid k]\,, \tag{15}$$

and the centered proxy

$$c_\theta(x_0, m; k) \triangleq h_\theta(x_0, m; k) - \mu_{h,k}. \tag{16}$$

By construction under the exact conditional mean,

$$\mathbb{E}[c_\theta(x_0, m; k) \mid k] = 0. \tag{17}$$

Timestep proposal and importance weight. Let $q(k)$ be a proposal distribution over timesteps satisfying

$$q(k) > 0, \qquad \sum_{k=1}^{K} q(k) = 1.$$

Since the target timestep distribution is uniform, the corresponding importance weight is

$$w(k) \triangleq \frac{1/K}{q(k)} = \frac{1}{Kq(k)}. \tag{18}$$

Adjusted scalar loss and ISCV estimator. For timestep-specific coefficients $\beta = \{\beta_k\}_{k=1}^K$, define

$$\tilde{\ell}_{\theta,\beta}(x_0, m; k) \triangleq \ell_\theta(x_0, m; k) - \beta_k c_\theta(x_0, m; k). \tag{19}$$

Sampling $k \sim q$, $x_0 \sim \mathcal{D}$, and $m \sim p(\cdot \mid x_0, k)$, the ideal scalar ISCV estimator is

$$\widehat{\mathcal{L}}_{\theta,q,\beta} \triangleq w(k)\tilde{\ell}_{\theta,\beta}(x_0, m; k) = \frac{1}{Kq(k)}\left[\ell_\theta(x_0, m; k) - \beta_k c_\theta(x_0, m; k)\right]. \tag{20}$$

A.2 Unbiasedness of the Scalar ISCV Estimator

Theorem 1 (Unbiased scalar ISCV estimator)**.** *With exact centering by $\mu_{h,k}$, for any proposal $q(k) > 0$ and any fixed coefficients $\{\beta_k\}_{k=1}^K$, the estimator in (20) is an unbiased estimator of $\mathcal{L}(\theta)$:*

$$\mathbb{E}_{k\sim q,\, x_0\sim\mathcal{D},\, m\sim p(\cdot|x_0,k)}\left[\widehat{\mathcal{L}}_{\theta,q,\beta}\right] = \mathcal{L}(\theta). \tag{21}$$

Proof. Using the law of iterated expectations,

$$\mathbb{E}[\widehat{\mathcal{L}}_{\theta,q,\beta}] = \mathbb{E}_{k\sim q}\left[\mathbb{E}\left[w(k)\left(\ell_\theta(x_0,m;k) - \beta_k c_\theta(x_0,m;k)\right) \mid k\right]\right]. \tag{22}$$

For fixed k, $w(k)$ and β_k are constants. Therefore,

$$\mathbb{E}\left[w(k)\left(\ell_\theta - \beta_k c_\theta\right) \mid k\right] = w(k)\left(\mathbb{E}[\ell_\theta \mid k] - \beta_k \mathbb{E}[c_\theta \mid k]\right) \tag{23}$$

$$= w(k)\mu_k, \tag{24}$$

where we used $\mathbb{E}[c_\theta \mid k] = 0$. Substituting $w(k) = 1/(Kq(k))$,

$$\mathbb{E}[\widehat{\mathcal{L}}_{\theta,q,\beta}] = \sum_{k=1}^K q(k)\frac{1}{Kq(k)}\mu_k = \frac{1}{K}\sum_{k=1}^K \mu_k = \mathcal{L}(\theta). \tag{25}$$

□

A.3 Variance Formula and Coupled Roles of IS and CV

Let

$$Y \triangleq \widehat{\mathcal{L}}_{\theta,q,\beta}.$$

By the law of total variance with respect to $k \sim q$,

$$\mathrm{Var}(Y) = \mathbb{E}_{k\sim q}[\mathrm{Var}(Y \mid k)] + \mathrm{Var}_{k\sim q}(\mathbb{E}[Y \mid k]). \tag{26}$$

Define the conditional adjusted variance

$$r_k(\beta) \triangleq \mathrm{Var}\Big(\tilde{\ell}_{\theta,\beta}(x_0,m;k) \mid k\Big). \tag{27}$$

Since $Y = w(k)\tilde{\ell}_{\theta,\beta}$ and $w(k)$ is constant conditioned on k,

$$\mathbb{E}_{k\sim q}[\mathrm{Var}(Y \mid k)] = \sum_{k=1}^K q(k)w(k)^2 r_k(\beta) \tag{28}$$

$$= \frac{1}{K^2}\sum_{k=1}^K \frac{r_k(\beta)}{q(k)}. \tag{29}$$

Moreover, because the centered proxy has zero conditional mean,

$$\mathbb{E}[\tilde{\ell}_{\theta,\beta} \mid k] = \mathbb{E}[\ell_\theta \mid k] = \mu_k.$$

Thus,

$$\mathbb{E}[Y \mid k] = w(k)\mu_k = \frac{\mu_k}{Kq(k)}.$$

Therefore,

$$\mathrm{Var}_{k\sim q}(\mathbb{E}[Y \mid k]) = \mathbb{E}_{k\sim q}\left[\left(\frac{\mu_k}{Kq(k)}\right)^2\right] - \mathcal{L}(\theta)^2 \tag{30}$$

$$= \frac{1}{K^2}\sum_{k=1}^K \frac{\mu_k^2}{q(k)} - \mathcal{L}(\theta)^2. \tag{31}$$

Combining (26)–(31) gives the explicit scalar estimator variance formula:

$$\mathrm{Var}\Big(\widehat{\mathcal{L}}_{\theta,q,\beta}\Big) = \frac{1}{K^2}\sum_{k=1}^{K}\frac{r_k(\beta)+\mu_k^2}{q(k)} - \mathcal{L}(\theta)^2. \tag{32}$$

This formula separates the quantity affected by the control variate, $r_k(\beta)$, from the timestep proposal $q(k)$. The two roles are complementary but coupled: the control variate changes the conditional adjusted variance, and hence the adjusted second moment $r_k(\beta)+\mu_k^2$, while the proposal determines how these per-timestep second-moment terms are weighted across timesteps.

A.4 Optimal Control-Variate Coefficients

For a fixed timestep k, consider the conditional adjusted variance

$$f_k(\beta_k) \triangleq \mathrm{Var}(\ell_\theta(x_0,m;k) - \beta_k c_\theta(x_0,m;k) \mid k)\,. \tag{33}$$

Theorem 2 (Optimal control-variate coefficient)**.** *If* $\mathrm{Var}(h_\theta \mid k) > 0$*, the coefficient minimizing* $f_k(\beta_k)$ *is*

$$\beta_k^\star = \frac{\mathrm{Cov}(\ell_\theta, h_\theta \mid k)}{\mathrm{Var}(h_\theta \mid k)}. \tag{34}$$

The minimized conditional adjusted variance is

$$f_k(\beta_k^\star) = \mathrm{Var}(\ell_\theta \mid k) - \frac{\mathrm{Cov}(\ell_\theta, h_\theta \mid k)^2}{\mathrm{Var}(h_\theta \mid k)}. \tag{35}$$

If additionally $\mathrm{Var}(\ell_\theta \mid k) > 0$*, this can be written as*

$$f_k(\beta_k^\star) = \mathrm{Var}(\ell_\theta \mid k)(1-\rho_k^2), \tag{36}$$

where

$$\rho_k = \frac{\mathrm{Cov}(\ell_\theta, h_\theta \mid k)}{\sqrt{\mathrm{Var}(\ell_\theta \mid k)\mathrm{Var}(h_\theta \mid k)}}.$$

If $\mathrm{Var}(h_\theta \mid k) = 0$*, the proxy has no conditional variation at timestep* k*, and we take* $\beta_k^\star = 0$*.*

Proof. Since $c_\theta = h_\theta - \mu_{h,k}$, centering does not change the variance of h_θ or its covariance with ℓ_θ:

$$\mathrm{Var}(c_\theta \mid k) = \mathrm{Var}(h_\theta \mid k), \qquad \mathrm{Cov}(\ell_\theta, c_\theta \mid k) = \mathrm{Cov}(\ell_\theta, h_\theta \mid k).$$

Expanding the conditional variance gives

$$f_k(\beta_k) = \mathrm{Var}(\ell_\theta \mid k) + \beta_k^2 \mathrm{Var}(h_\theta \mid k) - 2\beta_k \mathrm{Cov}(\ell_\theta, h_\theta \mid k). \tag{37}$$

This is a convex quadratic in β_k when $\mathrm{Var}(h_\theta \mid k) > 0$. Differentiating and setting the derivative to zero yields

$$2\beta_k^\star \mathrm{Var}(h_\theta \mid k) - 2\mathrm{Cov}(\ell_\theta, h_\theta \mid k) = 0,$$

which gives (34). Substituting $\beta_k^\star$ into (37) gives (35). The correlation form (36) follows by the definition of ρ_k. □

A.5 Adjusted Second-Moment Decomposition

For fixed coefficients β, define the timestep-conditioned adjusted second moment

$$R_k(\beta) \triangleq \mathbb{E}\Big[\tilde{\ell}_{\theta,\beta}(x_0,m;k)^2 \mid k\Big]\,. \tag{38}$$

Lemma 3 (Adjusted second moment as variance plus mean square)**.** *For any* k *and any coefficient* β_k*,*

$$R_k(\beta) = r_k(\beta) + \mu_k^2. \tag{39}$$

Proof. For any random variable Z,

$$\mathbb{E}[Z^2] = \mathrm{Var}(Z) + (\mathbb{E}[Z])^2.$$

Apply this identity to $Z = \tilde{\ell}_{\theta,\beta}(x_0, m; k)$ conditioned on k. Since $\mathbb{E}[c_\theta \mid k] = 0$, we have

$$\mathbb{E}[\tilde{\ell}_{\theta,\beta} \mid k] = \mathbb{E}[\ell_\theta \mid k] = \mu_k.$$

Therefore,

$$R_k(\beta) = \mathrm{Var}(\tilde{\ell}_{\theta,\beta} \mid k) + \mu_k^2 = r_k(\beta) + \mu_k^2.$$

□

For timesteps with $\mathrm{Var}(h_\theta \mid k) > 0$, combining Lemma 3 with Theorem 2, the ideal coefficient $\beta_k^\star$ gives

$$R_k(\beta^\star) = \mu_k^2 + \mathrm{Var}(\ell_\theta \mid k) - \frac{\mathrm{Cov}(\ell_\theta, h_\theta \mid k)^2}{\mathrm{Var}(h_\theta \mid k)}. \tag{40}$$

When the correlation coefficient ρ_k is well-defined, this becomes

$$R_k(\beta^\star) = \mu_k^2 + \mathrm{Var}(\ell_\theta \mid k)(1 - \rho_k^2). \tag{41}$$

If $\mathrm{Var}(h_\theta \mid k) = 0$, the adjustment has no effect under the convention $\beta_k^\star = 0$, so $R_k(\beta^\star) = \mu_k^2 + \mathrm{Var}(\ell_\theta \mid k)$.

A.6 Optimal Timestep Proposal

For fixed coefficients β, the second moment of the scalar ISCV estimator is

$$\mathbb{E}\left[\widehat{\mathcal{L}}_{\theta,q,\beta}^2\right] = \sum_{k=1}^{K} q(k)\mathbb{E}\left[\left(\frac{1}{Kq(k)}\tilde{\ell}_{\theta,\beta}(x_0, m; k)\right)^2 \mid k\right] \tag{42}$$

$$= \frac{1}{K^2}\sum_{k=1}^{K}\frac{R_k(\beta)}{q(k)}. \tag{43}$$

Because $\mathbb{E}[\widehat{\mathcal{L}}_{\theta,q,\beta}] = \mathcal{L}(\theta)$, minimizing the variance over q is equivalent to minimizing this second moment.

Theorem 4 (Second-moment-minimizing timestep proposal)**.** *Fix β, and assume $R_k(\beta) > 0$ for all k. Among all proposal distributions satisfying $q(k) > 0$ and $\sum_{k=1}^K q(k) = 1$, the second moment in (43) is minimized by*

$$q^\star(k) = \frac{\sqrt{R_k(\beta)}}{\sum_{j=1}^K \sqrt{R_j(\beta)}}. \tag{44}$$

If some $R_k(\beta) = 0$, the same expression is understood as the limiting optimum with arbitrarily small positive mass on zero-second-moment timesteps.

Proof. For fixed β, minimizing $\mathbb{E}[\widehat{\mathcal{L}}_{\theta,q,\beta}^2]$ is equivalent to solving

$$\min_q \sum_{k=1}^{K}\frac{R_k(\beta)}{q(k)} \quad \text{s.t.} \quad q(k) > 0, \qquad \sum_{k=1}^{K} q(k) = 1.$$

The Lagrangian is

$$\mathcal{J}(q, \lambda) = \sum_{k=1}^{K}\frac{R_k(\beta)}{q(k)} + \lambda\left(\sum_{k=1}^{K} q(k) - 1\right).$$

Setting derivatives to zero gives

$$-\frac{R_k(\beta)}{q(k)^2} + \lambda = 0,$$

so

$$q(k) = \sqrt{\frac{R_k(\beta)}{\lambda}} \propto \sqrt{R_k(\beta)}.$$

Normalization over k yields (44). □

A.7 Idealized Dominance over the Uniform Baseline

This subsection gives an idealized comparison against uniform timestep sampling. It assumes exact moments, optimal coefficients, and unclipped importance weights. It should not be read as a guarantee for the stabilized online controller used in practice.

Let the uniform no-CV estimator be

$$\widehat{\mathcal{L}}_{\mathrm{uni}}(x_0, m; k) = \ell_\theta(x_0, m; k), \qquad k \sim U(1, K).$$

Define the raw timestep second moment

$$R_k(0) = \mathbb{E}[\ell_\theta(x_0, m; k)^2 \mid k]. \tag{45}$$

Then

$$\mathbb{E}[\widehat{\mathcal{L}}^2_{\mathrm{uni}}] = \frac{1}{K} \sum_{k=1}^{K} R_k(0). \tag{46}$$

Let $\beta^\star = \{\beta_k^\star\}_{k=1}^K$ be the per-timestep optimal control-variate coefficients from Theorem 2. Define

$$S_k^\star \triangleq \sqrt{R_k(\beta^\star)}.$$

Using the optimal proposal from Theorem 4, the ideal joint second moment is

$$\min_q \mathbb{E}[\widehat{\mathcal{L}}^2_{\theta,q,\beta^\star}] = \frac{1}{K^2} \left(\sum_{k=1}^{K} S_k^\star \right)^2. \tag{47}$$

Theorem 5 (Idealized dominance over uniform sampling)**.** *Under the idealized assumptions above,*

$$\min_q \mathbb{E}[\widehat{\mathcal{L}}^2_{\theta,q,\beta^\star}] \le \mathbb{E}[\widehat{\mathcal{L}}^2_{\mathrm{uni}}]. \tag{48}$$

Since both estimators are unbiased for $\mathcal{L}(\theta)$, the same ordering holds for their variances.

Proof. For each timestep k, $\beta_k^\star$ minimizes the conditional adjusted variance. Since the centered adjustment has zero conditional mean, the conditional mean of the adjusted loss remains μ_k for all β_k. Thus, minimizing the conditional adjusted variance is equivalent to minimizing the conditional adjusted second moment, and

$$R_k(\beta^\star) \le R_k(0) \qquad \text{for all } k.$$

By Cauchy–Schwarz,

$$\left(\sum_{k=1}^{K} S_k^\star \right)^2 \le K \sum_{k=1}^{K} (S_k^\star)^2 = K \sum_{k=1}^{K} R_k(\beta^\star).$$

Thus,

$$\frac{1}{K^2} \left(\sum_{k=1}^{K} S_k^\star \right)^2 \le \frac{1}{K} \sum_{k=1}^{K} R_k(\beta^\star) \le \frac{1}{K} \sum_{k=1}^{K} R_k(0).$$

The right-hand side is exactly $\mathbb{E}[\widehat{\mathcal{L}}^2_{\mathrm{uni}}]$, proving the result. Because both estimators are unbiased, subtracting the common $\mathcal{L}(\theta)^2$ gives the corresponding variance ordering. □

The inequality is strict unless both components provide no reduction: the proxy yields no per-timestep improvement, and the adjusted second moments are already uniform across timesteps. This statement is only an idealized oracle-style comparison, not a finite-time guarantee for the practical controller.

A.8 Gradient Second-Moment Analogue

This subsection is supplementary. It records the analogous second-moment calculation for an explicitly centered vector-valued estimator and a conservative cross-entropy upper bound for LoRA gradients. It is not a claim that every practical differentiable-proxy update is an exactly unbiased gradient estimator of the original objective.

Ideal vector-valued estimator. Let

$$g_\theta(x_0, m; k) \triangleq \nabla_\theta \ell_\theta(x_0, m; k) \in \mathbb{R}^d$$

denote the per-sample gradient at timestep k, and let

$$\bar{g}_k \triangleq \mathbb{E}[g_\theta(x_0, m; k) \mid k].$$

Under standard conditions allowing interchange of expectation and differentiation,

$$\nabla_\theta \mathcal{L}(\theta) = \frac{1}{K} \sum_{k=1}^{K} \bar{g}_k.$$

Let $\phi_\theta(x_0, m; k) \in \mathbb{R}^r$ be a vector-valued proxy with conditional mean

$$\mu_{\phi,k} = \mathbb{E}[\phi_\theta(x_0, m; k) \mid k].$$

Let $B_k \in \mathbb{R}^{d \times r}$. Define the ideal centered vector estimator

$$\widehat{g}_{q,B}(x_0, m; k) = w(k) \left[g_\theta(x_0, m; k) - B_k \big(\phi_\theta(x_0, m; k) - \mu_{\phi,k} \big) \right]. \tag{49}$$

If $\mu_{\phi,k}$ is exact and $q(k) > 0$, then

$$\mathbb{E}[\widehat{g}_{q,B}] = \nabla_\theta \mathcal{L}(\theta). \tag{50}$$

Optimal vector control variate. For fixed k, define the adjusted gradient

$$\widetilde{g}(x_0, m; k) = g_\theta(x_0, m; k) - B_k(\phi_\theta(x_0, m; k) - \mu_{\phi,k}).$$

Minimizing the conditional second moment

$$\mathbb{E}[\|\widetilde{g}(x_0, m; k)\|_2^2 \mid k]$$

over B_k gives the multivariate regression coefficient

$$B_k^\star = \mathrm{Cov}(g_\theta, \phi_\theta \mid k) \, \mathrm{Var}(\phi_\theta \mid k)^{-1}, \tag{51}$$

assuming the proxy covariance matrix is invertible. With a singular or ill-conditioned covariance, the corresponding pseudoinverse or a ridge-regularized inverse can be used.

Gradient second-moment proposal. For a fixed adjusted gradient $\widetilde{g}$, define

$$s_k \triangleq \mathbb{E}[\|\widetilde{g}(x_0, m; k)\|_2^2 \mid k].$$

Then

$$\mathbb{E}[\|\widehat{g}_{q,B}\|_2^2] = \frac{1}{K^2} \sum_{k=1}^{K} \frac{s_k}{q(k)}. \tag{52}$$

The same constrained minimization as in Appendix A.6 gives

$$q^\star(k) \propto \sqrt{s_k} = \sqrt{\mathbb{E}[\|\widetilde{g}(x_0, m; k)\|_2^2 \mid k]}.$$

This mirrors the scalar second-moment principle. We use it only as supplementary gradient-side intuition, not as the main theoretical claim.

A conservative cross-entropy link to LoRA gradient moments. The main paper does not claim a general equivalence between scalar objective variance and true gradient variance. The following proposition only provides a conservative upper-bound link in a common cross-entropy setting for the trainable LoRA parameters.

Proposition 6 (CE-based upper bound on LoRA gradient second moments)**.** *Consider a supervised masked-token cross-entropy term*

$$\ell_\psi(x, y, k) = -\log p_\psi(y \mid x, k),$$

where ψ denotes the trainable LoRA parameter vector, with all other model parameters fixed, and $p_\psi(\cdot \mid x, k) = \mathrm{softmax}(z_\psi(x, k))$. Let e_y denote the one-hot vector for target y, and define the logits-to-LoRA Jacobian

$$J_\psi(x, k) = \frac{\partial z_\psi(x, k)}{\partial \psi}.$$

Then

$$\nabla_\psi \ell_\psi(x, y, k) = J_\psi(x, k)^\top \big(p_\psi(\cdot \mid x, k) - e_y\big),$$

and

$$\|\nabla_\psi \ell_\psi(x, y, k)\|_2^2 \le 2\|J_\psi(x, k)\|_{\mathrm{op}}^2 \ell_\psi(x, y, k)^2. \tag{53}$$

Consequently, if a constant $C_k \ge 0$ satisfies

$$\|J_\psi(x, k)\|_{\mathrm{op}}^2 \le C_k$$

for all relevant x at timestep k, then

$$\mathbb{E}[\|\nabla_\psi \ell_\psi(x, y, k)\|_2^2 \mid k] \le 2C_k \mathbb{E}[\ell_\psi(x, y, k)^2 \mid k]. \tag{54}$$

Proof. Since

$$\nabla_\psi \ell_\psi = J_\psi^\top (p_\psi - e_y),$$

we have

$$\|\nabla_\psi \ell_\psi\|_2^2 \le \|J_\psi\|_{\mathrm{op}}^2 \|p_\psi - e_y\|_2^2.$$

For a one-hot target e_y,

$$\|p_\psi - e_y\|_2^2 = (1 - p_\psi(y \mid x, k))^2 + \sum_{j \ne y} p_\psi(j \mid x, k)^2 \le 2(1 - p_\psi(y \mid x, k))^2.$$

Using $1 - u \le -\log u$ for $u \in (0, 1]$,

$$\|p_\psi - e_y\|_2^2 \le 2\ell_\psi(x, y, k)^2.$$

Substituting gives (53). Taking conditional expectation given k and using $\|J_\psi(x, k)\|_{\mathrm{op}}^2 \le C_k$ yields (54). □

Proposition 6 is only an upper-bound relation. It does not establish a general equivalence between scalar objective variance and true gradient variance. We therefore keep the main analysis at the scalar-estimator level and use gradient-side evidence only as supplementary empirical support when feasible.

A.9 Practical Assumptions and Stabilization

The unbiasedness and optimality results above assume exact conditional moments, exact centering of the proxy, full importance weights, and proposal support $q(k) > 0$. The practical controller approximates these ideal quantities online.

In practice, $\mu_{h,k}$ is estimated using lagged exponential moving averages, while proposal probabilities, β coefficients, gates, and adjusted-moment estimates are maintained as detached controller state. ISCV-SG detaches the complete current-sample correction, so the proxy affects future proposal allocation but not the current parameter gradient. ISCV-S leaves the confidence correction differentiable and is therefore a proxy-adjusted surrogate.

The implementation clamps each proposal entry at the pre-normalization threshold $q_{\min}$ and then renormalizes, as defined in Section 3.2. Consequently, the realized lower bound is $q_{\min}/Z_{\mathrm{floor}}$, not $q_{\min}$ itself.These safeguards are essential for stable large-model training, but they mean the implemented controller is a stabilized online approximation to the unclipped ideal estimator analyzed above. Let $d_i^{(0)} = \beta_{g(k_i)} c_i$ denote the unscaled correction. The practical LoRA controller uses an effective multiplier $0 \le \lambda_t \le \lambda_{\max}$, including any configured warmup or diagnostic down-scaling, and optionally applies absolute and relative bounds:

$$d_i^{\mathrm{cap}} = \mathrm{sign}(\lambda_t d_i^{(0)}) \min\Big\{ |\lambda_t d_i^{(0)}|, \delta_{\mathrm{abs}}, \delta_{\mathrm{rel}} |\ell_i| \Big\}.$$

When a bound is disabled, its threshold is interpreted as $+\infty$. A diagnostic gate $a_i \in \{0, 1\}$ produces the applied correction $d_i = a_i d_i^{\text{cap}}$. When enabled, the gate sets $a_i = 0$ if the provisional corrected batch has nonpositive proxy correlation or if its variance ratio relative to the uncorrected batch is at least one.

Coefficient stabilization, when enabled, is

$$\beta_g^{\text{target}} = \text{clip}\left(\frac{\widehat{\text{Cov}}_g(\ell, h)}{\widehat{\text{Var}}_g(h) + \lambda_{\text{ridge}}}, \beta_{\min}, \beta_{\max}\right),$$
$$\beta_g \leftarrow \beta_g + \text{clip}\big(\beta_g^{\text{target}} - \beta_g, -\Delta_\beta, \Delta_\beta\big).$$

Exact task-specific values and disabled safeguards are reported in the resolved configurations.

Algorithm 1 ISCV training update

1: Maintain detached controller state: timestep-specific proxy means $\widehat{\mu}_{h,k}$, grouped regression moments $\widehat{m}_{s,g}$, grouped coefficients β_g, timestep-specific adjusted second moments $\widehat{R}_k$, and q_{train}.
2: Sample timesteps $k_i \sim q_{\text{train}}$ and masks $m_i \sim p(\cdot \mid x_{0,i}, k_i)$.
3: Run one standard MDLM forward pass to compute ℓ_i, logits, and proxy values h_i.
4: Compute $c_i = h_i - \widehat{\mu}_{h,k_i}$ and $\tilde{\ell}_i^{\text{prop}} = \ell_i - \beta_{g(k_i)} c_i$.
5: Set $w_i = 1/(K q_{\text{train}}(k_i))$. For ISCV-SG use $s_i = w_i \ell_i - \text{sg}(w_i \beta_{g(k_i)} c_i)$; for ISCV-S use $s_i = w_i(\ell_i - \beta_{g(k_i)} c_i)$.
6: Backpropagate $\widehat{\mathcal{L}}_{\text{batch}} = B^{-1} \sum_i s_i$.
7: Update detached EMA statistics for proxy centering and β_g; update $\widehat{R}_k$ from no-gradient squared adjusted scalars $(\tilde{\ell}_i^{\text{prop}})^2$; then update q_{train} for future minibatches.

B Experimental Protocol and Additional Diagnostics

This appendix records the experimental details and diagnostic quantities used to connect the practical implementation in Section 3 to the idealized scalar estimator. It provides reproducibility details, computation protocols, full per-run results, and supplementary diagnostics. Extended comparisons cover IS-only, CV-only, proxy and proposal-statistic variants, and safeguard ablations.

B.1 Dataset Sources and Evaluation Protocols

Table 7 summarizes the dataset sources and split protocols used by the training and evaluation pipeline. Metrics, prompt templates, generation settings, and answer extractors are recorded in the resolved run configurations.

Table 7: Datasets and split protocols. Metrics, prompt templates, generation settings, and answer extractors are recorded in the resolved run configurations.

Dataset	Source / config	Task type	Split protocol
GSM8K	gsm8k/main	Numeric QA	train[:90%] / train[90%:] / test
OpenScience	nvidia/OpenScienceReasoning-2	Multiple choice	paper split: 4500 / 500 / 1000
HiTab	microsoft/HiTab	Table QA	train / validation / test
MBPP	google-research-datasets/mbpp/full	Code generation	train / validation / test
MATH	MATH benchmark	Mathematical reasoning	train[:95%] / train[95%:] / test (5,000)

B.2 Experimental Details

Model and adaptation. We initialize from `GSAI-ML/LLaDA-8B-Instruct` (Nie et al., 2025b) and train a separate LoRA adapter for each downstream task. The main runs use response-only masked diffusion training with $K = 64$ timesteps and a log-linear corruption schedule. LoRA uses rank 8, alpha 16, and dropout 0.05, targeting the attention projection modules. Further run-specific hyperparameters and evaluation commands are recorded in the resolved configurations.

Comparison and evaluation protocol. We repeat training with seeds 42, 731, and 20231 when compute allows, holding the model, data processing, and hyperparameters fixed within each comparison. Within a task, comparisons share the pretrained backbone, split, initialization protocol, batch construction, optimizer schedule, token/data passes, and update budget. Evaluation uses greedy LLaDA generation under the dataset-specific protocol, so across-run differences primarily reflect training rather than inference sampling. MATH uses the complete 5,000-example test set for all three seeds.

Baseline implementations. Standard uses uniform timestep training. Clipped denotes a clipped timestep-window baseline inspired by the variance-aware noising schedules and timestep-window training used in BD3-LMs (Arriola et al., 2025). P-POTS+MIRROR uses paired negatively correlated mask-pattern computation where available. The S/SG comparison in Table 3 is a strict matched comparison of the differentiable ISCV-S and detached ISCV-SG paths. Within each pair, all training, controller, evaluation, and checkpoint-selection settings are fixed; only `cv_stopgrad` changes.

Dataset processing. All dataset processing is controlled by the saved resolved configuration for each run. Public datasets are loaded through the Hugging Face `datasets` interface when possible. HiTab is downloaded from the official `microsoft/HiTab` release and converted into the JSONL format used by our pipeline with repository scripts. GSM8K uses final-number extraction, MATH uses final-answer extraction, OpenScience uses final-option-letter extraction, HiTab uses normalized answer matching, and MBPP uses unit-test execution. Exact split strings, prompt templates, target fields, truncation policy, and answer extractors are recorded in the resolved configuration.

Prompt and target construction. Each sample is converted into a prompt and a target string. In paper protocol modes, the prompt style is forced to the dataset-specific template: GSM8K and OpenScience use a question-only prompt, HiTab serializes the table and question, and MBPP includes the programming problem plus an instruction to return only Python code. The target is the configured answer or code field with an EOS token appended before tokenization.

Masked diffusion supervision. Training samples concatenate the prompt and target tokens. The main runs use response-only diffusion eligibility, so masks and supervised loss terms are restricted to the target side while prompt tokens remain conditioning context. Strict filtering removes samples whose prompt-plus-target length exceeds the configured maximum sequence length, unless a dataset-specific target-preserving truncation policy is enabled.

B.3 Compute Accounting and Checkpoint Selection

Comparisons match the training horizon, tokens, and optimizer updates. When a baseline requires extra model evaluations per update, we report its measured wall-clock runtime rather than treating all optimizer steps as having the same cost; exact FLOPs and GPU-hours need not be matched.

All reported runs were conducted on NVIDIA RTX PRO 6000 GPUs with 96GB memory per GPU. The run manifests record the number of GPUs, wall-clock training time, selected checkpoint timestamp, and diagnostic logging window for each experiment.

We report training time as wall-clock time measured from the first optimizer step to the end of training, excluding post-hoc evaluation and artifact copying unless explicitly stated. When a run uses gradient accumulation, the step count refers to optimizer steps rather than microbatch forwards. If a method performs extra forward or backward passes inside each update, that overhead is included in the measured wall-clock time.

For a run using N_{gpu} GPUs, GPU-hours are computed as

$$H_{\mathrm{gpu}} = N_{\mathrm{gpu}} \frac{t_{\mathrm{end}} - t_{\mathrm{start}}}{3600}.$$

B.4 Full-Parameter Pretraining Protocol, Results, and Uncertainty

We pretrain all parameters of a 169M-parameter DiT-backbone MDLM from scratch on OpenWebText, using seeds 42 and 731. Standard and ISCV-SG share initialization, data order, optimizer, scheduler,

minibatches, and token/update budgets. Both use one model forward/backward pass per minibatch. This matches the dominant model evaluations but not exact wall-clock time or GPU-hours.

Table 8 reports the complete paired trajectories behind Section 4.4. Relative reductions are positive when ISCV-SG has lower generative perplexity.

Table 8: Full-parameter pretraining trajectories. Intervals resample the 200 paired generated evaluation sequences for each fixed checkpoint pair; they measure evaluation-example uncertainty, not uncertainty across independently trained seeds.

Budget	Seed	Standard PPL	ISCV-SG PPL	Relative PPL reduction (95% paired bootstrap CI)
250M	42	100.0859	93.3735	$+6.71\%$ $[3.86\%, 9.43\%]$
250M	731	101.3307	108.8543	-7.42% $[-11.14\%, -3.89\%]$
500M	42	81.3452	79.3441	$+2.46\%$ $[-0.51\%, 5.36\%]$
500M	731	88.1531	87.8404	$+0.36\%$ $[-2.90\%, 3.47\%]$
750M	42	75.7474	73.0317	$+3.58\%$ $[0.69\%, 6.40\%]$
750M	731	84.1471	76.3620	$+9.25\%$ $[6.66\%, 11.82\%]$
1B	42	71.4850	71.3531	$+0.18\%$ $[-2.90\%, 3.21\%]$
1B	731	78.4585	75.0107	$+4.39\%$ $[1.44\%, 7.26\%]$

At 250M tokens, ISCV-SG has lower generative perplexity for seed 42 but higher perplexity for seed 731; at 500M, 750M, and 1B tokens, it is lower for both seeds. The Standard/ISCV-SG two-seed means at these four budgets are $100.71/101.11$, $84.75/83.59$, $79.95/74.70$, and $74.97/73.18$, respectively. These two trajectories provide evidence beyond LoRA but remain seed- and budget-dependent; they do not establish broad generality or an exact compute-normalized advantage. The paired bootstrap intervals quantify evaluation-example uncertainty for each fixed checkpoint pair and do not replace uncertainty across independently trained seeds.

B.5 Full Per-Run Results

Table 9 reports the ISCV-S per-run results used in the component and S/SG comparisons. We include task performance, training time in hours where available, and the effective scalar-objective variance proxy for each run.

B.6 Seed-Wise Uncertainty for ISCV-SG Comparisons

For each task, Table 10 compares ISCV-SG with the external baseline having the highest three-seed mean. The exploratory t-intervals contain only three same-seed differences and are reported to show uncertainty rather than to support significance claims.

B.7 Scalar-Objective Variance Computation

For each logged sample or microbatch item i, define three scalar quantities:

$$z_i^{\mathrm{raw}} = \ell_i, \qquad z_i^{\mathrm{adj}} = \ell_i - \beta_{g(k_i)}\big(h_i - \widehat{\mu}_{h,k_i}\big), \qquad z_i^{\mathrm{w}} = w(k_i)\, z_i^{\mathrm{adj}}.$$

Here z_i^{raw} is the raw masked diffusion loss and z_i^{adj} is the control-variate-adjusted scalar loss. The numerical quantity z_i^{w} is used for loss-space diagnostics and must not be interpreted as a measurement of stochastic-gradient variance. For ISCV-S the adjustment remains on the autograd path, whereas for ISCV-SG the complete correction is detached as in Eq. (**??**). When clipping or weight normalization is enabled, $w(k_i)$ denotes the actual effective weight used by the practical training objective.

For a logging window $\mathcal{W}$ with n items, we compute the sample variance

$$\widehat{\mathrm{Var}}_{\mathcal{W}}(z) = \frac{1}{n-1}\sum_{i\in\mathcal{W}}\left(z_i - \bar{z}_{\mathcal{W}}\right)^2, \qquad \bar{z}_{\mathcal{W}} = \frac{1}{n}\sum_{i\in\mathcal{W}} z_i.$$

To align directly with the ideal proposal in Section 3, we also compute the per-timestep second moment of the adjusted scalar loss:

$$\widehat{R}_k = \frac{1}{|\mathcal{W}_k|}\sum_{i\in\mathcal{W}_k}\left(z_i^{\mathrm{adj}}\right)^2, \qquad \mathcal{W}_k = \{i \in \mathcal{W} : k_i = k\}.$$

Table 9: Full per-run results on GSM8K, HiTab, OpenScience, and MBPP. We report task performance, training time in hours where available, and the effective scalar-objective variance proxy for each run. Entries not reported in the table are marked as "–"; MBPP runtimes are omitted because hour-scale reporting is not informative for those shorter runs.

	Seed 42			Seed 731			Seed 20231		
Dataset / Method	Perf	Time (h)	Eff. Obj. Var.	Perf	Time (h)	Eff. Obj. Var.	Perf	Time (h)	Eff. Obj. Var.
GSM8K									
Standard	61.87	1.45	0.2733	60.34	1.52	0.2443	62.70	1.49	0.2479
Clipped	64.06	2.70	0.3939	63.68	2.86	0.6708	64.52	2.78	0.5487
P-POTS	64.59	1.58	0.0840	63.00	1.57	0.0895	67.10	1.57	0.0692
P-POTS+MIRROR	65.73	3.29	0.0311	64.14	3.36	0.0306	66.49	3.41	0.0302
IS	64.97	1.53	0.1116	66.34	1.58	0.1128	65.73	1.56	0.1188
CV	68.01	1.56	0.1176	67.17	1.58	0.1018	66.34	1.57	0.1115
ISCV-S	68.39	1.57	0.0952	67.24	1.63	0.0974	67.32	1.60	0.0865
HiTab									
Standard	52.52	3.83	0.2466	60.10	3.81	2.9249	57.08	3.83	0.2494
Clipped	61.99	4.16	0.1038	61.04	4.14	0.2688	58.90	4.15	1.0728
P-POTS	59.68	3.98	0.0971	59.34	3.97	0.6087	58.08	3.98	0.8665
P-POTS+MIRROR	61.17	7.76	0.0907	61.93	7.92	0.2955	61.11	7.95	0.0614
IS	57.64	3.85	0.1708	58.71	3.84	0.3040	60.10	3.82	0.1346
CV	58.96	3.86	0.1573	62.07	3.84	0.0811	62.12	3.83	0.1851
ISCV-S	62.02	3.84	0.0887	63.39	3.82	0.1692	63.26	3.85	0.0689
OpenScience									
Standard	64.95	6.44	1.0125	63.60	6.46	1.0552	62.90	5.15	2.0847
Clipped	63.87	7.36	1.4917	65.90	7.39	1.1800	64.60	7.36	0.5897
P-POTS	67.10	6.53	3.7958	66.90	6.55	2.4184	67.10	9.76	2.2366
P-POTS+MIRROR	68.30	13.05	2.2580	67.40	13.04	3.3577	67.70	13.02	2.2022
IS	66.40	8.42	2.9637	66.40	5.19	1.6298	67.00	6.46	3.8069
CV	66.90	6.47	1.0026	66.70	6.47	1.4266	66.30	5.16	2.6977
ISCV-S	69.60	6.85	1.0032	67.20	5.44	0.0011	68.50	6.84	0.0029
MBPP									
Standard	40.86	–	0.5435	40.86	–	1.5267	39.69	–	1.3591
Clipped	41.25	–	3.1936	40.86	–	0.9582	41.63	–	1.0278
P-POTS	42.41	–	0.7786	42.02	–	6.9451	41.25	–	8.5413
P-POTS+MIRROR	40.86	–	3.4101	42.02	–	3.5909	41.63	–	3.4179
IS	41.63	–	0.5886	40.86	–	1.6193	41.25	–	0.5060
CV	41.63	–	0.4440	40.86	–	1.0985	40.47	–	1.1754
ISCV-S	43.19	–	0.4973	42.41	–	1.2971	42.80	–	0.3993

Table 10: Same-seed descriptive differences for the ISCV-SG comparisons. All exploratory intervals include zero.

Dataset	External baseline	Seed-wise ISCV-SG minus baseline	Mean ± SD	Exploratory 95% t-CI
GSM8K	P-POTS+MIRROR	$1.90/0.91/-2.20$	0.20 ± 2.14	$[-5.11, 5.52]$
MATH	Clipped	$3.38/-1.26/0.70$	0.94 ± 2.33	$[-4.85, 6.73]$
HiTab	P-POTS+MIRROR	$0.33/0.12/0.75$	0.40 ± 0.32	$[-0.40, 1.20]$
OpenScience	P-POTS+MIRROR	$-1.20/-0.10/-1.10$	-0.80 ± 0.61	$[-2.31, 0.71]$
MBPP	P-POTS+MIRROR	$-0.39/0.39/1.95$	0.65 ± 1.19	$[-2.31, 3.61]$

The centered per-timestep variance $\widehat{\mathrm{Var}}_{\mathcal{W}_k}(z^{\mathrm{adj}})$ is reported as a stability diagnostic, while $\widehat{R}_k$ is the theorem-aligned statistic for proposal construction. Unless otherwise stated, summary tables report the average over the final diagnostic logging window.

B.8 Supplementary Gradient-Side Diagnostics and Stability Audits

The main analysis targets the scalar estimator before automatic differentiation. This diagnostic instead asks whether the proposal learned from scalar adjusted losses is aligned with parameter-gradient second moments. Along one seed-42 ISCV-SG trajectory, we evaluate 27 checkpoints at steps $448, 456, \ldots, 656$. Each common-random-number probe backpropagates the raw loss through 192 LoRA tensors comprising 6,291,456 trainable parameters, aggregates squared full-LoRA gradient norms into eight timestep groups, and compares two candidate proposal statistics.

For a grouped proposal, the gradient-second-moment-optimal target is

$$q^{\star}_{\mathrm{grad}}(k) \propto p(k)\sqrt{\mathbb{E}[\|g_\theta\|_2^2 \mid k]}.$$

Table 11: Gradient-allocation probe over 27 checkpoints. Cosine similarity is computed against the grouped gradient-second-moment-optimal target. Predicted moment ratios are normalized to uniform sampling; all 27 corrected-proposal ratios are below one.

Statistic	Cosine median	Cosine range	Moment-ratio median	Moment-ratio range
Raw loss	0.743	[0.639, 0.814]	0.837	[0.765, 0.987]
Corrected loss	**0.942**	[0.888, 0.974]	**0.586**	[0.566, 0.760]

This diagnostic is checkpoint-local. It does not estimate full minibatch-gradient covariance over training, establish that scalar variance determines accuracy, or justify extrapolation from AdamW to other optimizers.

B.9 Estimator-Component Comparisons

Table 2 further isolates the estimator components on GSM8K. Both IS and CV improve over standard training, and their combination achieves the best average score, consistent with the view that timestep allocation and within-timestep adjustment address complementary sources of estimator inefficiency.

B.10 Scalar-Estimator Diagnostics and Training Trajectories

The primary theorem-aligned quantity is scalar-objective variance, not full gradient variance. We therefore evaluate variance at the same level as the estimator analyzed in Section 3: raw scalar loss, adjusted scalar loss, and final weighted scalar objective. We also report proxy correlations, since the control-variate adjustment is expected to help only when the forward-pass proxy is informative within a timestep. Extended diagnostics across datasets, seeds, proxy variants, learned proposals, safeguards, and gradient-side quantities are reported in Appendix B.

Table 4 reports scalar-estimator diagnostics on GSM8K. ISCV-S obtains the lowest final weighted scalar-objective variance among the listed variants, and its adjusted scalar variance is lower than its raw scalar variance. The proxy correlations are positive but imperfect, suggesting that same-forward

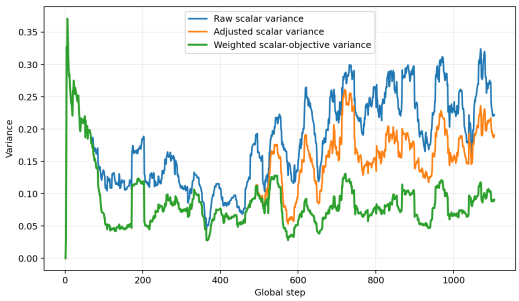

Figure 1: GSM8K scalar-estimator variance trajectories. The plot shows raw scalar variance, adjusted scalar variance, and final weighted scalar-objective variance over training. These are loss-space diagnostics aligned with the scalar-estimator analysis, not full gradient-variance measurements.

confidence statistics provide usable within-timestep signals without acting as oracle surrogates. We interpret these diagnostics at the scalar-estimator level, not as measurements of full gradient variance. This statistic is not a monotone predictor of downstream accuracy: P-POTS+MIRROR can have lower scalar-objective variance than ISCV-S while obtaining lower task accuracy. We therefore use scalar variance as a local controller diagnostic rather than a sufficient explanation of final accuracy.

Figure 1 visualizes the same scalar-estimator diagnostics over training. The trajectories summarize how the raw, adjusted, and final weighted scalar-objective variances evolve under the practical controller.

B.11 Proposal Stability and Safeguards

Table 12: Proposal stability diagnostics on GSM8K. ESS is reported over $K = 64$ timestep bins. The adaptive proposals remain high-entropy and avoid extremely large importance weights.

Method	ESS	Max weight	Entropy
IS	35.51	3.09	3.86
ISCV-S	38.50	2.50	3.91

Table 12 shows that the adaptive proposal remains well conditioned. ISCV-S maintains higher ESS, lower maximum weight, and higher proposal entropy than IS in these logs, indicating that the adjusted-scalar controller does not collapse onto a small set of timesteps.

B.12 Component Ablations

Table 13 reports targeted component ablations on GSM8K with seed 42. These ablations isolate proposal statistics, stabilization safeguards, and proxy choice while holding the training budget fixed. Because they are single-seed diagnostics from the ablation bundle, we use them to interpret mechanisms rather than as the primary downstream performance claim.

Table 13: Targeted GSM8K component ablations on seed 42. Test metrics are reported in percentage points, and scalar variance denotes the final weighted scalar-objective variance. These single-seed diagnostics isolate design choices and should be interpreted separately from the multi-seed main results.

Variant	Purpose	Test metric	Scalar var.
Raw-stat proposal	Proposal statistic	66.72	0.1059
Adjusted-stat proposal	Proposal statistic	68.39	0.0952
ISCV-S w/o gates/caps	Stabilization	66.19	0.1118
Entropy proxy	Proxy choice	68.39	0.0952
Margin proxy	Proxy choice	66.41	0.1042
Max-prob proxy	Proxy choice	65.20	0.1043

Adjusted-stat proposal yields both the lowest scalar variance and the highest task score among the proposal-statistic variants in this single-seed diagnostic. Removing gates/caps increases scalar variance relative to the entropy-proxy ISCV-S setting, supporting the role of stabilization. Among proxy variants, entropy gives the best task score and lowest scalar variance in this diagnostic. Margin and max-prob have similar scalar variances to each other but lower task scores, suggesting that proxy choice affects more than the scalar variance summary alone. This direct raw/adjusted comparison is specific to ISCV-S and seed 42.

Controller sensitivity. Seed-42 GSM8K runs with correction caps $\lambda_{\max} = 0.50/0.55/0.60$ obtain $66.11/66.64/64.90\%$ accuracy; exact paired McNemar tests do not resolve any pairwise difference ($p \geq 0.081$). Assigning $5/10/20\%$ of proposal mass adaptively gives $63.84/64.22/64.59\%$, with all pairwise $p \geq 0.51$. On OpenScience, otherwise identical seed-42 runs activating importance sampling at steps 480 and 600 both obtain 67.10%. Existing runs do not isolate EMA decay or timestep grouping, and the proposal floor and weight clip did not activate; these results support only local robustness to the tested settings.

B.13 Reproducibility Artifacts

Anonymous code and resolved configurations for the full-parameter comparison between Standard and ISCV-SG are available in an anonymous artifact. The artifact exposes the stop-gradient setting and exact importance weights and includes commands needed to reproduce the reported runs.

C LIMITATIONS

Our theory analyzes an idealized scalar estimator with exact conditional moments and centering, full-support proposals, and unstabilized importance weights. The implementation uses timestep-specific proxy centering but grouped regression-coefficient estimates, together with bounded adjustments, proposal floors, gates, and weight stabilization, and is therefore a stabilized online approximation to the ideal estimator.

More specifically, ISCV-S changes the expected gradient target through the differentiable proxy, whereas ISCV-SG preserves the original conditional expected gradient only under full support and exact, unclipped importance weighting. Neither property implies that lower scalar variance monotonically improves final accuracy.

The LoRA evidence uses one backbone and task-specific adapters. MATH uses the complete 5,000-example test set for all three seeds. Full-parameter evidence covers one 169M-parameter DiT backbone on OpenWebText and two seeds; the 250M-token result is seed dependent. Thus, the experiments do not establish universality across scales, architectures, optimizers, datasets, or long budgets. ISCV also requires an informative within-timestep proxy, whose correlation may vary across data and model states.

Finally, ISCV targets the supervised masked diffusion training estimator; it is not a downstream alignment method, inference-time controller, or safety filter. Deployment risks remain tied to the model, data, and application.

D BROADER IMPACTS

ISCV may reduce adaptation compute, but the same efficiency could lower the cost of producing harmful or misleading text. It is neither a deployed system nor a safety mechanism; risks depend on the model, data, release policy, and application context.