# Aligning Diffusion Language Models via Unpaired Preference Optimization

**Anonymous authors**
**Paper under double-blind review**

## Abstract

Diffusion language models (dLLMs) are an emerging alternative to autoregressive (AR) generators, but aligning them to human preferences is challenging because sequence log-likelihoods are intractable and pairwise preference data are costly to collect. We introduce ELBO-KTO, which combines an Evidence Lower Bound (ELBO) surrogate for diffusion log-likelihoods with a prospect-theoretic, unpaired preference objective (Kahneman–Tversky Optimization, KTO). We analyze the bias and variance induced by the ELBO substitution and employ variance-reduction practices that stabilize gradients during training. Applied to LLaDA-8B-Instruct, ELBO-KTO yields **65.9%** and **62.3%** adjusted win rates on kto-mix-14k and UltraFeedback-Binary, respectively, versus the base model under an automatic LLM judge. Across downstream tasks, including GSM8K, MMLU, and additional reasoning/knowledge benchmarks, ELBO-KTO trained on UltraFeedback-Binary performs on par with or better than the base model under identical decoding. This establishes unpaired preference optimization as a viable alternative to pairwise alignment in diffusion LLMs. Code is available in the supplementary material and will be released publicly upon publication.

## 1 Introduction

Aligning large language models (LLMs) with human preferences is central to building helpful systems. In autoregressive LLMs, methods such as Reinforcement learning from human feedback (RLHF) (Christiano et al., 2017; Ouyang et al., 2022; Stiennon et al., 2020; Schulman et al., 2017) and Direct Preference Optimization (DPO) (Rafailov et al., 2023) and its variants (Meng et al., 2024; Yuan et al., 2023; Azar et al., 2024) are standard and rely on curated *paired* comparisons for each prompt. Diffusion-style large language models (dLLMs) extend text generation beyond left-to-right decoding by iteratively refining a sequence in parallel (Ye et al., 2025; Ou et al., 2024; Lou et al., 2024; Nie et al., 2025; Khanna et al., 2025) which makes sentence-level likelihoods intractable and complicates preference optimization.

Recent approaches to align dLLMs have replaced log-likelihoods with Evidence Lower Bound (ELBO) based surrogates and incorporated variance reduction techniques to stabilize training, but remain constrained by the assumption of paired preference data (Zhu et al., 2025). This assumption limits scalability and applicability in practical settings where feedback is abundant but inherently unpaired—as in binary "good/bad" user ratings or safety filter signals.

In this work, we take a fundamentally different direction by proposing ELBO-KTO as a framework to train dLLMs directly from unpaired preference signals. Our approach integrates ELBO estimates into KTO's value computation via an ELBO margin (policy vs. reference) and a baseline for variance control. This formulation unlocks learning from a broader class of real-world "good/bad" feedback signals while preserving training stability through lightweight variance reduction techniques.

The main contributions are summarized as follows:

- We present ELBO-KTO, a principled preference optimization framework that enables alignment of diffusion language models (dLLMs) from unpaired feedback.

- We provide a theoretical analysis of the bias and variance tradeoff introduced by ELBO substitution, and prove that our estimator enjoys bounded bias with controlled variance.
- We empirically validate our proposed method ELBO-KTO on kto-mix-14k (TRL contributors, 2024) and UltraFeedback-Binary (Cui et al., 2023), where it achieves **65.9%** and **62.3%** adjusted win rates over the base model, demonstrating consistent improvements. Extensive analysis confirms that our method provides a simple yet effective way to align dLLMs with stronger models. On downstream reasoning/knowledge tasks such as GSM8K and MMLU, ELBO-KTO trained on UltraFeedback-Binary performs on par with or better than the LLaDA-8B-Instruct.

## 2 Preliminaries

### 2.1 LLaDA and Variance-Reduced Preference Optimization (VRPO)

Preference alignment is essential for improving diffusion language models (dLLMs). In this work, we use LLaDA-8B-Instruct (Nie et al., 2025) as the base model, a masked diffusion LM pretrained on 2.3T tokens and instruction-tuned on 4.5M examples.

A key challenge is that DPO-style methods rely on sequence-level log-likelihoods, which are intractable in masked diffusion LMs. VRPO addresses this by replacing log-likelihood terms with ELBO surrogates defined over diffusion denoising steps (Zhu et al., 2025). Given a prompt $x$ and response $y$, a diffusion timestep $t \sim \mathcal{U}[0,1]$ is sampled and a noised intermediate $y_t$ is drawn from the forward process $q(y_t \mid t, y, x)$. Averaging over $t$ and $y_t$ yields the ELBO surrogate

$$\mathcal{B}_\pi(y \mid x) = \mathbb{E}_{t\sim\mathcal{U}[0,1]}\ \mathbb{E}_{y_t\sim q(\cdot|t,y,x)}\big[\ell_\pi(y_t, t, y \mid x)\big], \tag{1}$$

where $\ell_\pi$ denotes the per-step loss of the mask prediction model (Appendix A), and $\mathcal{B}_\pi$ lower-bounds $\log \pi(y \mid x)$ (Lou et al., 2024; Ou et al., 2024).

In practice, $\mathcal{B}_\pi$ is estimated via Monte Carlo (MC) sampling. Because optimization is sensitive to estimator variance, VRPO emphasizes variance-reduction strategies such as increasing the sampling budget, reallocating samples across diffusion steps, and sharing random draws between policy and reference.

### 2.2 Kahneman–Tversky Optimization (KTO)

KTO (Ethayarajh et al., 2024) is a preference optimization algorithm inspired by prospect theory in behavioral economics, which models human perception of gains and losses as asymmetric.

Let $\pi_\theta$ be the policy, $\pi_{\text{ref}}$ be a frozen reference, and $\sigma(\cdot)$ the sigmoid function. KTO defines the *reference–adjusted reward* as $r_\theta(x,y) = \log \frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)}$. This reward is mapped through a prospect-theoretic *value function* with separate loss-aversion weights for desirable and undesirable examples:

$$v(x,y) = \begin{cases} \lambda_D \sigma(\beta\,(r_\theta(x,y) - z_0)), & y \sim y_{\text{desirable}}|x, \\ \lambda_U \sigma(\beta\,(z_0 - r_\theta(x,y))), & y \sim y_{\text{undesirable}}|x. \end{cases} \tag{2}$$

The baseline $z_0 = \text{KL}(\pi_\theta(\cdot|x) \,\|\, \pi_{\text{ref}}(\cdot|x))$ acts as a reference point to control loss saturation. Here, $\beta > 0$ controls risk aversion, while $\lambda_D$ and $\lambda_U$ govern asymmetric loss aversion for desirable and undesirable samples. Assigning $\lambda_y = \lambda_D$ for desirable and $\lambda_y = \lambda_U$ for undesirable samples, the KTO objective is $\mathcal{L}_{\text{KTO}}(\theta) = \mathbb{E}_{(x,y)\sim\mathcal{D}}[\lambda_y - v(x,y)]$.

**Adapting KTO to dLLMs is non-trivial.** A direct ELBO substitution in KTO is not immediate: the original KTO score and baseline are defined in log-likelihood and KL terms, while dLLMs require MC ELBO surrogates. This raises three practical issues: (1) the per-prompt KL baseline is intractable and expensive to estimate, (2) naive MC replacement can introduce high variance, and (3) variance-reduction techniques established for VRPO are not automatically guaranteed to transfer to KTO. These challenges motivate the method we present next.

# 3 Method

## 3.1 ELBO-KTO for Diffusion LLMs

KTO scores a response by the log-likelihood ratio $\log \frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)}$, which is intractable for dLLMs. To solve for this, we replace each log-likelihood with an MC estimator of the ELBO lower bound ($\mathcal{B}$) and work with the *MC ELBO margin*

$$\widehat{r}_\theta(x, y) \;=\; \widehat{\mathcal{B}}_{\pi_\theta}(y \mid x) \;-\; \widehat{\mathcal{B}}_{\pi_{\text{ref}}}(y \mid x), \tag{3}$$

where $\widehat{\mathcal{B}}_\pi$ is defined as $\widehat{\mathcal{B}}_\pi(y \mid x) = \frac{1}{n_t}\sum_{j=1}^{n_t} \frac{1}{n_{y_t}} \sum_{i=1}^{n_{y_t}} \ell_\pi\big(y_{t_j}^{(i)}, t_j, y \mid x\big)$, and $\ell_\pi$ represents the per-step loss of the mask prediction model (Appendix A). Here, we sample $n_t$ time steps, $\{t_j\}_{j=1}^{n_t} \sim \mathcal{U}[0, 1]$, and $n_{y_t}$ draws per timestep, $\{y_{t_j}^{(i)}\}_{i=1}^{n_{y_t}} \sim q(y_{t_j} \mid t_j, y, x)$ for the MC estimate. Here, $q$ is the forward diffusion process.

## 3.2 Global Per-Batch Baseline for Variance Control

Classical KTO introduces a per-prompt reference point

$$z_0(x) = \mathbb{E}_{y' \sim \pi_\theta(\cdot|x)} \left[ \log \frac{\pi_\theta(y' \mid x)}{\pi_{\text{ref}}(y' \mid x)} \right],$$

i.e. a KL term. This reference is intractable for dLLMs and estimating it with Monte Carlo methods is prohibitively expensive. Thus, for computational efficiency and stability, we use a *single scalar global baseline* computed per mini-batch $S$

$$\hat{b}_0(S) \;=\; \frac{1}{m}\sum_{i=1}^{m} \widehat{r}_\theta(x_i, y_i), \qquad S = \{(x_i, y_i)\}_{i=1}^{m}, \tag{4}$$

and treat $\hat{b}_0(S)$ as stop-gradient, i.e., we do not backpropagate through it. This is a standard control variate which recenters the scores entering the KTO sigmoid function. It reduces the gradient variance without any additional ELBO evaluations or KL baseline calculations using mismatched pairs as done in Ethayarajh et al. (2024).

## 3.3 Instantiated Loss for Diffusion Language Models

Let $s_i \in \{+1, -1\}$ encode desirable/undesirable and $g(u) = \sigma(\beta u)$ be the logistic link, where $\beta$ controls the KTO risk aversion. We instantiate the general KTO Loss for a mini-batch $S$ with the MC ELBO margin (3) and the global baseline (4):

$$\widehat{L}(S) = \frac{1}{m}\sum_{i=1}^{m} \lambda_{s_i} \left\{ 1 - g\Big( s_i \left[ \widehat{r}_\theta(x_i, y_i) - \hat{b}_0(S) \right] \Big) \right\}, \tag{5}$$

where only the $\widehat{\mathcal{B}}_{\pi_\theta}$ term backpropagates; the reference term, $\widehat{\mathcal{B}}_{\pi_{\text{ref}}}$ and $\hat{b}_0(S)$ are treated as constants. This technique centers the scores around the batch mean to improve stability and efficiency while preserving the ELBO-difference structure needed for diffusion LLMs.

Subtracting a constant baseline minimizes the variance of the centered scores; with logistic $g$, this keeps $s_i[\widehat{r}_\theta - \hat{b}_0]$ near the high slope region, avoiding loss saturation and reducing the gradient variance (Section 4.5). Using the batch mean (4) gives this benefit without additional compute. A detailed bias-variance discussion for the loss and the gradient bounds for (5) is provided in the Section 4.

**Variance Reduction.** We show theoretically (Section 4) and verify empirically (Section 5.3) that VRPO-style ELBO variance controls work for ELBO-KTO and we use these techniques in our implementation. These controls are not automatically guaranteed for KTO; in our method, the global per-batch baseline plays a key role in making variance control practical and stable.

# 4 Theoretical Analysis

We analyze ELBO-KTO and bound the bias and variance of its loss and gradient, motivating VRPO-style variance reduction for stable training. We also show that the global per-batch baseline $\hat{b}_0$ is optimal among constant baselines. Proofs and additional details are deferred to Appendix C.

## 4.1 Setup and Notation

**Dataset and Minibatch.** Let the finite dataset be $\mathcal{D}_N = \{(x_n, y_n)\}_{n=1}^N$, where $x_n$ is prompt and $y_n$ is response. A minibatch is $S = \{(x_i, y_i, s_i, \lambda_i)\}_{i=1}^m$, sampled uniformly. Here $s_i \in \{-1, +1\}$ denotes undesirable/desirable, and $\lambda_i \in \{\lambda_D, \lambda_U\}$ are class weights with $\lambda_{\max} = \max\{\lambda_D, \lambda_U\}$.

**ELBO Margins and Global Baseline.** Define the ELBO margin as $r_i := B_{\pi_\theta}(y_i \mid x_i) - B_{\pi_{\text{ref}}}(y_i \mid x_i)$, and its MC estimate as $\hat{r}_i := \widehat{B}_{\pi_\theta}(y_i \mid x_i) - \widehat{B}_{\pi_{\text{ref}}}(y_i \mid x_i)$. The global mini-batch baseline is $\hat{b}_0 = \frac{1}{m}\sum_{j=1}^m \hat{r}_j$, and the signed centered margin is $\hat{\delta}_i = s_i(\hat{r}_i - \hat{b}_0)$. For gradients, we treat $\hat{b}_0$ and the reference term as stop-grad, i.e., we don't backpropagate through them.

**Objective and Target.** With logistic link $g(u) = \sigma(\beta u)$, the per-item and batch losses are

$$\hat{\ell}_i \;=\; \lambda_i\big(1 - g(\hat{\delta}_i)\big), \qquad \hat{L}(S) \;=\; \tfrac{1}{m}\sum_{i=1}^m \hat{\ell}_i. \tag{6}$$

The corresponding noise-free target *global-baseline* replaces the MC estimates by expectations:

$$L_{\text{GB}}^{\text{sg}}(S;\theta) = \tfrac{1}{m}\sum_{i=1}^m \lambda_i\big(1 - g\big(s_i(r_i - b_0)\big)\big), \tag{7}$$

where $b_0 = \mathbb{E}_{\text{MC}}[\hat{b}_0] = \frac{1}{m}\sum_{j=1}^m r_j$. Also, for $g = \sigma(\beta u)$, we get $L_g = \beta/4$ and $L_{g'} = \beta^2/(6\sqrt{3})$ as the Lipschitz constants of $g$ and $g'$ respectively (Lemma 2).

**Centered-Margin Variance Aggregator.** Conditioning on $S$, we can write $\mathbb{E}_{\text{MC}}[\hat{r}_i] = r_i$. Assuming the exchangeability of the indices under the MC design (Appendix C), let $v(S) := \text{Var}_{\text{MC}}(\widehat{r}_i)$ and $c(S) := \text{Cov}(\widehat{r}_i, \widehat{r}_j)$ for $i \neq j$. Under these assumptions, we get

$$\Psi(S) \;:=\; \tfrac{1}{m}\sum_{i=1}^m \mathbb{E}_{\text{MC}}\big[\text{Var}_{\text{MC}}(\hat{\delta}_i)\big] \;=\; \tfrac{m-1}{m}\,\big(v(S) - c(S)\big), \tag{8}$$

All our theoretical bounds depend on this scalar quantity $\Psi(S)$.

## 4.2 Bias and Variance Analysis for Loss

**Theorem 1** (Loss bias bound)**.** *The minibatch loss bias relative to the global-baseline target satisfies*

$$\Big|\mathbb{E}_{\mathcal{D}}\mathbb{E}_{\text{MC}}[\hat{L}(S)] - \mathbb{E}_{\mathcal{D}}[L_{\text{GB}}^{\text{sg}}(S;\theta)]\Big| \;\le\; \lambda_{\max} L_g\, \mathbb{E}_{\mathcal{D}}\big[\sqrt{\Psi(S)}\big],$$

*where $\Psi(S)$ is the centered-margin variance aggregator defined in equation 8.*

**Theorem 2** (Loss variance bound)**.** *For the minibatch loss $\hat{L}(S) = \frac{1}{m}\sum_{i=1}^m \lambda_i\big(1 - g(\hat{\delta}_i)\big)$ with $g(u) = \sigma(\beta u)$, the MC-induced variance satisfies*

$$\text{Var}_{\text{MC}}(\hat{L}(S)) \;\le\; (\lambda_{\max} L_g)^2\, \mathbb{E}_{\mathcal{D}}\big[\Psi(S)\big].$$

These results show that loss bias and variance arise from applying $g$ to the noisy centered margin $\hat{\delta}_i$ and are jointly controlled by $\Psi(S)$. Shrinking $\Psi(S)$ tightens the bias gap and reduces variance.

### 4.3 Bias and Variance Analysis for Gradient

Assuming unbiased ELBO gradients $\mathbb{E}_{\mathrm{MC}}[\nabla_\theta \hat{r}_i] = \nabla_\theta r_i$, we define the stochastic gradient for a minibatch $S$ as

$$\hat{G}(S) = \frac{1}{m} \sum_{i=1}^{m} \hat{a}_i \, \nabla_\theta \hat{r}_i, \tag{9}$$

where $\hat{a}_i = -\lambda_i s_i g'(\hat{\delta}_i)$ and $\hat{\delta}_i = s_i(\hat{r}_i - \hat{b}_0)$. Also, let $G = \frac{1}{m}\sum_{i=1}^{m} a_i \, \nabla_\theta r_i$, where $a_i = -\lambda_i s_i g'(\delta_i)$ and $\delta_i = s_i(r_i - b_0)$.

We further define

$$\tilde{G}^2(S) = \tfrac{1}{m} \sum_{i=1}^{m} \mathbb{E}_{\mathrm{MC}}\big[\|\nabla_\theta \hat{r}_i\|^2\big],$$

$$\bar{c}_\nabla(S) = \tfrac{1}{m(m-1)} \sum_{i \neq j} \mathbb{E}_{\mathrm{MC}}[\langle \xi_i, \xi_j \rangle],$$

where $\xi_i = \nabla_\theta \hat{r}_i - \nabla_\theta r_i$.

**Theorem 3** (Gradient bias bound)**.** *Let $\|\cdot\|$ be the $L_2$-norm. If $\|\nabla_\theta r_i\| \leq \bar{G}$ for all items, then*

$$\big\|\mathbb{E}_{\mathcal{D}}\mathbb{E}_{\mathrm{MC}}[\hat{G}(S)] - \mathbb{E}_{\mathcal{D}}[G(S)]\big\| \leq \lambda_{\max} L_{g'} \, \mathbb{E}_{\mathcal{D}}\big[\sqrt{\Psi(S)}\big] \, \bar{G} + \lambda_{\max} L_g \, \mathbb{E}_{\mathcal{D}}\big[\sqrt{\bar{v}_\nabla(S)}\big],$$

*where $\bar{v}_\nabla(S) = \frac{1}{m}\sum_{i=1}^{m} \mathbb{E}_{\mathrm{MC}}\big[\|\nabla_\theta \hat{r}_i - \nabla_\theta r_i\|^2\big]$.*

**Theorem 4** (Gradient variance bound)**.** *For mini-batch $S$, let $U(S) = \frac{1}{m}\sum_{i=1}^{m} a_i \, \nabla_\theta \hat{r}_i$. Then*

$$\mathrm{Var}_{\mathrm{MC}}\big(\hat{G}(S)\big) \leq \Big(\sqrt{\mathrm{Var}_{\mathrm{MC}}\big(U(S)\big)} \; + \; \lambda_{\max} L_{g'} \; \sqrt{\Psi(S)} \, \tilde{G}(S)\Big)^2,$$

*where*

$$\mathrm{Var}_{\mathrm{MC}}(U) \; \leq \; (\lambda_{\max} L_g)^2 \Big(\frac{\bar{v}_\nabla(S)}{m} \; + \; \frac{m-1}{m} \; \bar{c}_\nabla(S)\Big).$$

For the special case of independent per–item MC seeds, we further get

$$\mathrm{Var}_{\mathrm{MC}}\big(\hat{G}(S)\big) \leq \Big((\lambda_{\max} L_g) \sqrt{\tfrac{\bar{v}_\nabla(S)}{m}} \; + \; \lambda_{\max} L_{g'} \; \sqrt{\Psi(S)} \, \tilde{G}(S)\Big)^2,$$

where $\bar{v}_\nabla(S) = \frac{1}{m}\sum_{i=1}^{m} \mathbb{E}_{\mathrm{MC}}\big[\|\xi_i\|^2\big]$ (see Appendix C.5).

Both gradient bias and variance decompose into weight noise and score-gradient noise. Weight noise arises from applying the link function $g$ to the noisy centered margin $\hat{\delta}_i$ and is controlled by $\Psi(S)$. Score-gradient noise originates from stochastic ELBO gradients, is governed by $\bar{v}_\nabla(S)$, and can be reduced by increasing the MC budget.

### 4.4 Estimator Design Strategies

We now make the dependence on the estimator design explicit. Throughout this section we assume the exchangeable MC setting of Section 4.1, so that $v(S)$ and $c(S)$ are well-defined. Our goal is to strategically decrease $v(S)$ and increase $c(S)$, thereby shrinking the common driver $\Psi(S)$.

**Decreasing $v(S)$.** Expanding the formula for $v(S)$, we get

$$\begin{aligned} v(S) &= \mathrm{Var}_{\mathrm{MC}}(\hat{r}_i) \\ &= \mathrm{Var}_{\mathrm{MC}}(\widehat{B}_{\pi_\theta}(y_i \mid x_i) - \widehat{B}_{\pi_{\mathrm{ref}}}(y_i \mid x_i)) \\ &= \mathrm{Var}_{\mathrm{MC}}(\widehat{B}_{\pi_\theta}(y_i \mid x_i)) + \mathrm{Var}_{\mathrm{MC}}(\widehat{B}_{\pi_{\mathrm{ref}}}(y_i \mid x_i)) - 2\mathrm{Cov}(\widehat{B}_{\pi_\theta}(y_i \mid x_i), \widehat{B}_{\pi_{\mathrm{ref}}}(y_i \mid x_i)) \end{aligned}$$

Previous variance-reduction analysis for diffusion language models (Zhu et al., 2025) shows that the MC variance $\mathrm{Var}_{\mathrm{MC}}(\widehat{B})$ can be reduced by increasing the sampling budget and by allocating samples across diffusion timesteps. Moreover, using shared random numbers between the policy and reference ELBO estimators introduces positive covariance, which further decreases $v(S)$.

**Increasing $c(S)$.** Recall $c(S) = \mathrm{Cov}(\hat{r}_i, \hat{r}_j)$ under the exchangeability of indices. Intuitively, sharing random numbers across items in the minibatch $S$ will lead to a positive co-movement of $\hat{r}_i$ and $\hat{r}_j$, thereby increasing $c(S)$. This can further reduce $\Psi(S)$ by raising $c(S)$. For now, we do not use this technique in our experiments due to time constraints and leave this for future work.

### 4.5 Global Baseline Optimality

**Lemma 1** (Global Baseline Optimality)**.** *For any baseline $b$ that is constant across items in a batch and may depend on MC randomness,*

$$\Psi_b(S) - \Psi_{\hat{b}_0}(S) \;=\; \mathrm{Var}_{\mathrm{MC}}\big(b - \hat{b}_0\big) \;\geq\; 0,$$

*where $\hat{b}_0 = \frac{1}{m}\sum_j \hat{r}_j$. Hence every baseline of the form $b = \hat{b}_0 + K$ with deterministic constant $K$ attains the same minimum value $\Psi_b(S) = \Psi_{\hat{b}_0}(S)$. If, in addition, $\mathbb{E}_{\mathrm{MC}}[b] = b_0$, then $K = 0$ and the unique minimizer is $b = \hat{b}_0$.*

This result says that $\hat{b}_0$ is variance-optimal for all possible values of $b$. It requires no additional compute and is thus the principled default for ELBO-KTO.

## 5 Experiments

### 5.1 ELBO-KTO Training Recipe

We evaluate on two public preference datasets. kto-mix-14k contains ∼13.5k training prompts with desirable/undesirable labels and a 1.5k test split (750/750). It is the KTO counterpart of dpo-mix-7k (Argilla, 2024). UltraFeedback-Binary provides 61.1k train and 2k test pairs; we convert pairs to unpaired labels for ELBO-KTO. All models are trained on the respective train split and evaluated on the held-out test split.

We train for one epoch with batch size 8 using AdamW (weight decay 0.01, $\beta_1$=0.9, $\beta_2$=0.95) and a 3% linear warmup followed by cosine decay. For kto-mix-14k, the peak learning rate is $1\times10^{-6}$ and we draw 8 MC samples per example. For UltraFeedback-Binary, the peak learning rate is $5\times10^{-7}$ with 4 MC samples per example. Other experimental details can be found in Appendix D.

We will release training code, evaluation scripts, and configuration files to support reproducibility upon publication.

### 5.2 Overall test-set results

Table 1 reports adjusted win rates on the same test prompts for comparisons against the base model, LLaDA-8B-Instruct. Our ELBO-KTO completions win in a clear majority of cases with an adjusted win rate of 65.9% on kto-mix-14k and 62.3% on UltraFeedback-Binary.

For context, we report results for the publicly released LLaDA-1.5 trained with VRPO. This is not an apples-to-apples comparison as the data quantity and distribution used to train LLaDa-1.5 differ from our method. LLaDA-1.5 was trained on $\sim$ 350k paired preferences from a different distribution. As shown in Table 1, our method shows better performance than LLaDA-1.5. This highlights that targeted preference data and unpaired alignment can be more sample-efficient than scaling paired data from a broader mix.

For further evaluation, we calculate the adjusted win rates for the chosen and the rejected targets from the test dataset. We observe that the win rate of ELBO-KTO approaches the chosen-target win rate on kto-mix-14k and exceeds that of UltraFeedback-Binary. The performance of rejected targets fall well below 50%, confirming that the judge consistently disfavors undesirable responses.

To ensure our results are not an artifact of a particular LLM judge, we evaluated with two OpenAI models: `gpt-4o-mini` and `gpt-4.1-mini`, released about a year apart and trained under different alignment regimes. As shown in Table 2, both judges report nearly the same adjusted win rates. The majority-vote rate sits close to the two judges, confirming consistency. Importantly, to measure the inter-judge agreement, we

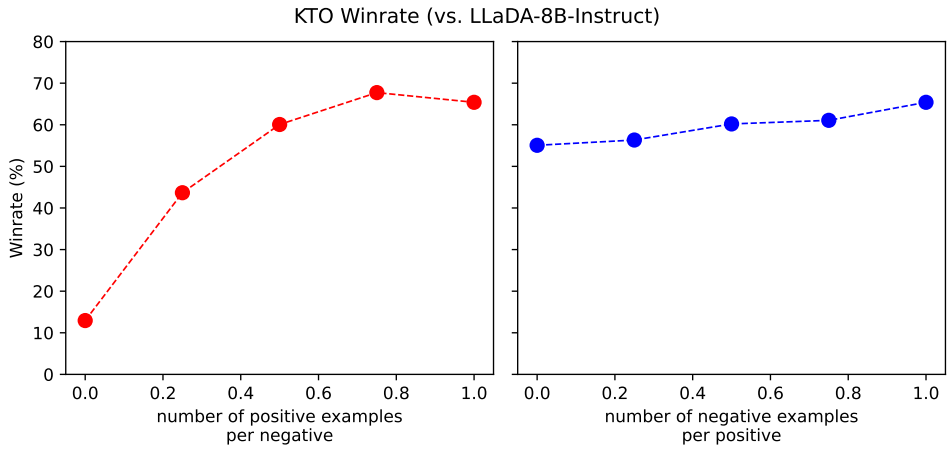


Figure 1: Adjusted win rate vs. LLaDA-8B-Instruct on kto-mix-14k when varying the ratio of desirable to undesirable examples. Left: subsampling desirable examples; Right: subsampling undesirable examples. ELBO-KTO benefits more from desirable examples, consistent with gain sensitivity.

Table 1: Adjusted win rates (AWR) (%) vs. LLaDA-8B-Instruct under FastChat `lm_judge` evaluated using `gpt-4o-mini` on kto-mix-14k and UltraFeedback-Binary (UFB). We use $\beta = 0.1$ for kto-mix-14k and $\beta = 0.2$ for UFB. For both datasets, we calculate AWR for (i) our ELBO-KTO method, (ii) LLaDA-1.5, (iii) the dataset's chosen target, and (iv) the dataset's rejected target. Both output orderings are judged and ties are split equally.

| | kto-mix-14k | UFB |
|---|---|---|
| ELBO-KTO | **65.9** | **62.3** |
| LLaDA-1.5 | 57.2 | 60.3 |
| Chosen (label=True) | 70.1 | 61.6 |
| Rejected (label=False) | 47.3 | 40.0 |

calculate Cohen's $\kappa$, a statistical measure of agreement between two annotators beyond chance. We obtain $\kappa = 0.56$ on kto-mix-14k and $\kappa = 0.61$ on UltraFeedback-Binary, indicating moderate-substantial agreement. These results demonstrate that our conclusions are robust to judge choice, even when the judges come from different generations of OpenAI models released one year apart.

### 5.3 Variance Reduction Ablations

This section empirically evaluates the variance-reduction components of ELBO-KTO by varying only the estimator design choices in Sections 4.4 and 4.5, while fixing all other experimental settings. Results are reported on kto-mix-14k using adjusted win rate (AWR) against LLaDA-8B-Instruct.

#### 5.3.1 Effect of the Global Per-Batch Baseline

We run an ablation comparing no baseline vs. the global per-batch baseline across three ($\beta$, learning-rate) settings under identical training and evaluation budgets. The global baseline consistently improves AWR by 5.27–9.34 percentage points (Table 3), indicating that centering margins with a stop-gradient batch mean stabilizes updates and yields better preference optimization.

#### 5.3.2 Estimator-design ablations for Monte Carlo sampling

Estimator-design ablations (Table 5) confirm the variance analysis from Section 4. Increasing the Monte Carlo budget by sampling more diffusion timesteps monotonically improves alignment performance, yielding

Table 2: Cross-generation judge comparison. Adjusted win rate (AWR, %) with 90% bootstrap CIs, majority-vote (MV), and Cohen's $\kappa$.

(a) kto-mix-14k

| Judge | AWR (%) | 90% CI |
|---|---|---|
| `gpt-4o-mini` | 65.87 | [63.73, 68.00] |
| `gpt-4.1-mini` | 63.80 | [61.67, 66.07] |
| MV (two judges) | 63.07 | [61.20, 64.93] |
| Cohen's $\kappa$ | 0.562 | [0.518, 0.604] |

(b) UltraFeedback-Binary

| Judge | AWR (%) | 90% CI |
|---|---|---|
| `gpt-4o-mini` | 62.28 | [60.88, 63.73] |
| `gpt-4.1-mini` | 62.21 | [60.71, 63.68] |
| MV (two judges) | 61.38 | [60.11, 62.68] |
| Cohen's $\kappa$ | 0.611 | [0.586, 0.636] |

Table 3: Adjusted win rate (AWR, %) with and without the global per-batch baseline across three $(\beta, \mathrm{lr})$ settings. Brackets show absolute gain (pp) vs. *No baseline*.

| $\beta$ | LR | No baseline | Global baseline |
|---|---|---|---|
| 0.2 | $5 \times 10^{-6}$ | 55.46 | **64.80** (+9.34) |
| 0.1 | $1 \times 10^{-6}$ | 60.63 | **65.90** (+5.27) |
| 0.2 | $1 \times 10^{-6}$ | 57.40 | **64.73** (+7.33) |

nearly a 9-point AWR gain from 2 to 8 samples. Holding the total budget fixed, different allocations across timesteps and per-timestep samples perform similarly. Finally, sharing random draws between the policy and reference improves AWR by +4.8 points, consistent with reduced ELBO margin variance. See Appendix B.1 for detailed results.

### 5.4 KTO under Class Imbalance

To test robustness under class imbalance, we subsample the kto-mix-14k dataset to vary the ratio of desirable to undesirable samples (Figure 1). We set $\lambda_D n_D = \lambda_U n_U$ to balance the effective contribution of each class in the loss, where $n_D$ and $n_U$ are the numbers of desired and undesired samples.

Overall, we find that performance increases with more data for either class, consistent with KTO scaling more favorably when trained on larger sample sizes. However, the effect of imbalance is asymmetric. Reducing the number of desirable samples hurts performance much more severely than reducing the number of undesirable samples. This pattern mirrors the *gain sensitivity* reported in Ethayarajh et al. (2024), where desirable examples contribute more strongly to alignment.

### 5.5 Downstream generalization

We further evaluate ELBO-KTO by fine-tuning LLaDA-8B-Instruct on UltraFeedback-Binary and testing under identical decoding on GSM8K, MMLU, HellaSwag, HumanEval, and GPQA. As shown in Table 4, ELBO-KTO improves GSM8K (+3.26), slightly lifts MMLU (+0.58) and GPQA (+0.67), keeps HumanEval unchanged, and slightly drops HellaSwag (−0.75). Overall, ELBO-KTO preserves downstream performance while yielding modest gains on reasoning/knowledge benchmarks. Further evaluation details are provided in Appendix D.

## 6 Related Work

**Image Diffusion Model Alignment** Preference optimization for diffusion models on image data has received significant attention. Wallace et al. (2024) propose *Diffusion-DPO*, which adapts DPO to diffusion models by defining a diffusion-based likelihood and directly optimizing over paired human preferences. Hong et al. (2024) introduce a margin-aware preference optimization method that removes the need for a reference model, while Lee et al. (2025) (CaPO) further improve alignment through reward calibration and Pareto-frontier-based pair selection. Similarly, Li et al. (2024) present *Diffusion-KTO*, which aligns text-to-image

Table 4: Performance comparison of LLaDA-8B-Instruct and ELBO-KTO on UltraFeedback-Binary on downstream tasks. The numbers in parentheses represent the number of shots used for evaluation.

| Task | LLaDA-8B-Instruct | ELBO-KTO |
|---|---|---|
| GSM8K (5) | 79.53 | **82.79** |
| MMLU (5) | 63.85 | **64.43** |
| HellaSwag (0) | **78.03** | 77.28 |
| HumanEval (0) | 42.68 | 42.68 |
| GPQA (5) | 29.02 | **29.69** |

diffusion models by maximizing expected human utility. Together, these approaches advance preference alignment in text-to-image diffusion models, whereas our proposed method specifically targets diffusion *language* models.

**Alignment for Diffusion Language Models** Diffusion language models are emerging as promising alternatives to autoregressive LLMs, including architectures such as DiffuLLaMA (Gong et al., 2024), Dream (Ye et al., 2025), LLaDA (Nie et al., 2025), and Mercury (Khanna et al., 2025). Recent work has begun exploring how to align dLLMs with preferences or reasoning objectives. The most closely related method is VRPO in LLaDA 1.5 (Zhu et al., 2025), which still operates under the paired preference regime and uses variance reduction to stabilize learning. Other diffusion-LM works emphasize reasoning or model structure rather than alignment: Zhao et al. (2025) scale reasoning in dLLMs via reinforcement learning, Tang et al. (2025) apply weighted policy optimization for reasoning in dLLMs, Yang et al. (2025) explore masked diffusion models in code generation or multimodal settings (focusing on model performance rather than feedback alignment), and Han et al. (2025) propose trajectory-level alignment via stepwise decomposition of discrete diffusion processes. While these advance reasoning, architecture, or decoding, they generally do not support preference alignment from unpaired feedback. Our proposed ELBO-KTO method fills in this gap.

## 7 Conclusion

We introduced a simple and effective recipe for aligning diffusion LLMs with unpaired feedback by plugging ELBO surrogates into KTO's value formulation: an ELBO margin for the policy–reference contrast and a mini-batch ELBO baseline, with VRPO-style variance reduction for stability. On LLaDA-8B-Instruct, the approach achieves strong automatic-judge gains on two test sets (65.9% on kto-mix-14k, 62.3% on UltraFeedback-Binary) and shows modest improvements on downstream tasks. Together, these results indicate that unpaired preference optimization is a viable path for diffusion LLM alignment and a practical complement to paired alignment methods.

Limitations include reliance on an ELBO surrogate (with potential bias/variance). Future work includes human evaluation, broader downstream tasks, principled choices for class weights $(\lambda_D, \lambda_U)$, stronger variance-reduction schemes, and combining unpaired ELBO-KTO with paired or reward-model objectives in dLLM settings.

## References


Argilla. argilla/dpo-mix-7k. https://huggingface.co/datasets/argilla/dpo-mix-7k, 2024.

Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces, 2023. URL https://arxiv.org/abs/2107.03006.

Mohammad Gheshlaghi Azar, Zhaohan Daniel Guo, Bilal Piot, Remi Munos, Mark Rowland, Michal Valko, and Daniele Calandriello. A general theoretical paradigm to understand learning from human preferences. In *International Conference on Artificial Intelligence and Statistics*, pp. 4447–4455. PMLR, 2024.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017.

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Wei Zhu, Yuan Ni, Guotong Xie, Zhiyuan Liu, and Maosong Sun. Ultrafeedback: Boosting language models with high-quality feedback, 2023.

Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. Kto: Model alignment as prospect theoretic optimization, 2024. URL https://arxiv.org/abs/2402.01306.

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. *arXiv preprint arXiv:2410.17891*, 2024.

Jiaqi Han, Austin Wang, Minkai Xu, Wenda Chu, Meihua Dang, Yisong Yue, and Stefano Ermon. Discrete diffusion trajectory alignment via stepwise decomposition. *arXiv preprint arXiv:2507.04832*, 2025.

Jiwoo Hong, Sayak Paul, Noah Lee, Kashif Rasul, James Thorne, and Jongheon Jeong. Margin-aware preference optimization for aligning diffusion models without reference. In *First Workshop on Scalable Optimization for Efficient and Adaptive Foundation Models*, 2024.

Samar Khanna, Siddhant Kharbanda, Shufan Li, Harshit Varma, Eric Wang, Sawyer Birnbaum, Ziyang Luo, Yanis Miraoui, Akash Palrecha, Stefano Ermon, et al. Mercury: Ultra-fast language models based on diffusion. *arXiv preprint arXiv:2506.17298*, 2025.

Kyungmin Lee, Xiahong Li, Qifei Wang, Junfeng He, Junjie Ke, Ming-Hsuan Yang, Irfan Essa, Jinwoo Shin, Feng Yang, and Yinxiao Li. Calibrated multi-preference optimization for aligning diffusion models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 18465–18475, 2025.

Shufan Li, Konstantinos Kallidromitis, Akash Gokul, Yusuke Kato, and Kazuki Kozuka. Aligning diffusion models by optimizing human utility. *Advances in Neural Information Processing Systems*, 37:24897–24925, 2024.

Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution. *arXiv preprint arXiv:2310.16834*, 2024.

Yu Meng, Mengzhou Xia, and Danqi Chen. Simpo: Simple preference optimization with a reference-free reward. *Advances in Neural Information Processing Systems*, 37:124198–124235, 2024.

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, 2025. URL https://arxiv.org/abs/2502.09992.

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. *arXiv preprint arXiv:2406.03736*, 2024.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741, 2023.

Subham Sekhar Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin T Chiu, Alexander Rush, and Volodymyr Kuleshov. Simple and effective masked diffusion language models, 2024. URL `https://arxiv.org/abs/2406.07524`.

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

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. Learning to summarize with human feedback. *Advances in neural information processing systems*, 33:3008–3021, 2020.

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.

TRL contributors. trl-lib/kto-mix-14k. `https://huggingface.co/datasets/trl-lib/kto-mix-14k`, 2024.

Bram Wallace, Meihua Dang, Rafael Rafailov, Linqi Zhou, Aaron Lou, Senthil Purushwalkam, Stefano Ermon, Caiming Xiong, Shafiq Joty, and Nikhil Naik. Diffusion model alignment using direct preference optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8228–8238, 2024.

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.

Hongyi Yuan, Zheng Yuan, Chuanqi Tan, Wei Wang, Songfang Huang, and Fei Huang. Rrhf: Rank responses to align language models with human feedback. *Advances in Neural Information Processing Systems*, 36: 10935–10950, 2023.

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.

Fengqi Zhu, Rongzhen Wang, Shen Nie, Xiaolu Zhang, Chunwei Wu, Jun Hu, Jun Zhou, Jianfei Chen, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. Llada 1.5: Variance-reduced preference optimization for large language diffusion models, 2025. URL `https://arxiv.org/abs/2505.19223`.

# A Additional Formulation

Diffusion Language Models (dLLMs) are inherently Masked Diffusion Models (MDMs), which incorporate a discrete random masking forward process and train a mask predictor to approximate the reverse unmasking process (Sahoo et al., 2024; Ou et al., 2024; Austin et al., 2023; Lou et al., 2024; Nie et al., 2025).

## A.1 Forward Process for Diffusion Language Models

During the forward process of MDMs, an original sequence $y$ is progressively corrupted by masking tokens independently at a noise level $t \in [0, 1]$. Let $x$ be the prompt, $y$ be the original response to the prompt, $y^i$ denote the $i-$th token of $y$, $L$ denote the total number of tokens in $y$, $K$ denote the vocabulary size, and $\mathbf{M}$ denote the mask token. The forward process $q$ to obtain the masked response $y_t$ at time $t$ can be formulated as

$$q(y_t \mid y, x, t) = \prod_{i=1}^{L} q(y_t^i \mid y^i, x, t), \tag{10}$$

where

$$q(y_t^i \mid y^i, x, t) = \begin{cases} t, & y_t^i = \mathbf{M}, \\ 1 - t, & y_t^i = y^i \end{cases}.$$

## A.2 Reverse Process for Diffusion Language Models

The reverse process starts at $t = 1$ from a fully masked sequence and gradually unmasks tokens till $t = 0$ to recover a fully unmasked language sequence. Let $p_\theta$ be the mask prediction models. Then, for timesteps $0 \leq s < t \leq 1$, the conditional distribution for the reverse process can be defined as

$$q(y_s \mid s, t, y_t, x) = \prod_{i=1}^{L} q\big(y_s^i \mid s, t, y_t, x\big) \tag{11}$$

$$q\big(y_s^i \mid s, t, y_t, x\big) = \begin{cases} \dfrac{t-s}{t}\, p_\theta\big(y^i \mid y_t, x\big), & y_t^i = \mathbf{M} \ \wedge \ y_s^i \neq \mathbf{M}, \\ \dfrac{s}{t}, & y_t^i = \mathbf{M} \ \wedge \ y_s^i = \mathbf{M}, \\ 1, & y_t^i \neq \mathbf{M} \ \wedge \ y_s^i = y_t^i, \\ 0, & \text{otherwise.} \end{cases} \tag{12}$$

## A.3 Log-Likelihood ELBO

The exact log-likelihood $\log \pi(y \mid x)$ intractable for dLLMs because of the nature of forward and reverse process. To tackle this, the log-likelihood is usually approximated by its ELBO (Lou et al., 2024; Ou et al., 2024; Nie et al., 2025):

$$\mathcal{B}_\pi(y \mid x) \triangleq \mathbb{E}_{t \sim \mathcal{U}[0,1]}\, \mathbb{E}_{y_t \sim q(y_t \mid t, y, x)}\, \ell_\pi(y_t, t, y \mid x), \tag{13}$$

where

$$\ell_\pi(y_t, t, y \mid x) \ \triangleq \ \left[ \frac{1}{t} \sum_{i=1}^{L} \mathbf{1}\big[y_t^i = \mathbf{M}\big] \ \log p_\theta\big(y^i \mid y_t, x\big) \right] \tag{14}$$

is the per-step loss of the mask prediction model.

Let $l$ be uniformly sampled from $\{1, 2, \ldots, L\}$ and $y_l$ denote the sequence obtained by masking $l$ tokens without replacement. Using this terminology, Zhu et al. (2025); Ou et al. (2024) define another equivalent formulation for the ELBO approximation:

$$\mathcal{B}'_\pi(y \mid x) \ \triangleq \ \mathbb{E}_{l \sim \mathcal{U}(\{1,2,\ldots,L\})}\, \mathbb{E}_{y_l \sim q(y_l \mid l, y, x)}\, \ell'_\pi(y_l, l, y \mid x), \tag{15}$$

Table 5: MC estimator ablations: (a) increasing total MC budget, (b) allocating a fixed MC budget ($n = 8$), and (c) independent noise vs. common random numbers (CRN).

| **Ablation setting** | **AWR (%)** |
|---|---|
| *(a) Increasing MC budget* | |
| $n_t = 2,\ n_{y_t} = 1$ | 57.0 |
| $n_t = 4,\ n_{y_t} = 1$ | 62.5 |
| $n_t = 8,\ n_{y_t} = 1$ | **65.9** |
| *(b) Fixed budget allocation ($n = 8$)* | |
| $n_t = 1,\ n_{y_t} = 8$ | **66.0** |
| $n_t = 2,\ n_{y_t} = 4$ | 65.5 |
| $n_t = 4,\ n_{y_t} = 2$ | 64.6 |
| $n_t = 8,\ n_{y_t} = 1$ | 65.9 |
| *(c) Common random numbers (CRN)* | |
| CRN= False | 61.1 |
| CRN= True | **65.9** |

where

$$\ell'_\pi(y_l, l, y \mid x) \triangleq \left[\frac{L}{l}\sum_{i=1}^{L}\mathbf{1}\big[y_l^i = \mathbf{M}\big]\ \log p_\theta\big(y^i \mid y_l, x\big)\right]. \tag{16}$$

Following Zhu et al. (2025), we adopt the $\mathcal{B}'_\pi(y \mid x)$ formulation for our experiments. Although $\mathcal{B}_\pi(y \mid x)$ and $\mathcal{B}'_\pi(y \mid x)$ are equivalent in expectation, the latter generally exhibits lower variance during estimation. This improvement arises because Eq. equation 15 deterministically masks exactly $l$ out of $L$ tokens in each sequence, producing more consistent samples, whereas Eq. equation 13 masks an expected fraction $t$ of tokens, introducing greater stochasticity. Consequently, we also employ the $\mathcal{B}'$ formulation in Eq. equation 15 as our log-likelihood estimator.

# B Additional Results

## B.1 Estimator-design ablations for Monte Carlo sampling

In Table 5, we show how different MC design choices affect performance. Let $n_t$ denote the number of diffusion timesteps sampled per example, and $n_{y_t}$ the number of forward-process samples per timestep. The total per-example MC budget is $n = n_t \times n_{y_t}$. We also study the effect of *common random numbers* (CRN), i.e., whether the policy and reference share the same noise draws.

**Increasing MC budget improves alignment.** We fix $n_{y_t} = 1$ and increase the number of sampled timesteps $n_t \in \{2, 4, 8\}$, thereby increasing the MC budget. As shown in Table 5, AWR improves monotonically, with nearly a 9-point gain from 2 to 8 samples. This trend matches our variance analysis, where larger MC budgets reduce estimator variance and yield more stable gradients.

**Allocating a fixed MC budget across timesteps.** With a fixed budget of $n = 8$, varying the allocation between $n_t$ and $n_{y_t}$ yields similar performance within $\approx 1.5$ AWR. Our default ($n_t = 8, n_{y_t} = 1$) is within 0.1 points of the best split.

**Common random numbers between policy and reference.** With the MC budget fixed at $n = 8$ and $(n_t, n_{y_t}) = (8, 1)$, we compare independent noise with shared noise (CRN) between the policy and reference. CRN yields a +4.8 AWR gain, confirming that increased covariance reduces ELBO margin variance and improves optimization.

# C Theoretical Analysis and Proofs

## C.1 Setup and Notation

**Dataset and Minibatch.** Let the finite dataset be $\mathcal{D}_N = \{(x_n, y_n)\}_{n=1}^N$, where $x_n$ is prompt and $y_n$ is response. A minibatch is $S = \{(x_i, y_i, s_i, \lambda_i)\}_{i=1}^m$, sampled uniformly. Here $s_i \in \{-1, +1\}$ denotes undesirable/desirable, and $\lambda_i \in \{\lambda_D, \lambda_U\}$ are class weights with $\lambda_{\max} = \max\{\lambda_D, \lambda_U\}$.

**Policies and ELBO Plug-In.** Let $\pi_\theta(\cdot \mid x)$ be the current policy and $\pi_{\text{ref}}(\cdot \mid x)$ be the frozen reference. We use an ELBO surrogate $B_\pi(y \mid x)$ and an unbiased MC estimator $\widehat{B}_\pi(y \mid x)$ satisfying $\mathbb{E}_{\text{MC}}[\widehat{B}_\pi(y \mid x)] = B_\pi(y \mid x)$.

**ELBO Margins and Global Baseline.** Define the ELBO margin as

$$r_i \;:=\; B_{\pi_\theta}(y_i \mid x_i) - B_{\pi_{\text{ref}}}(y_i \mid x_i), \tag{17}$$

and its MC estimate as

$$\hat{r}_i \;:=\; \widehat{B}_{\pi_\theta}(y_i \mid x_i) - \widehat{B}_{\pi_{\text{ref}}}(y_i \mid x_i). \tag{18}$$

The global mini-batch baseline is $\hat{b}_0 = \frac{1}{m}\sum_{j=1}^m \hat{r}_j$, and the signed centered margin is $\hat{\delta}_i = s_i(\hat{r}_i - \hat{b}_0)$. For gradients, we treat $\hat{b}_0$ and the reference term as stop-grad.

**Objective and Target.** With logistic link $g(u) = \sigma(\beta u)$, where $\sigma(x) = \frac{1}{1+e^{-x}}$, the per-item and batch losses are

$$\hat{\ell}_i \;=\; \lambda_i\big(1 - g(\hat{\delta}_i)\big), \qquad \hat{L}(S) \;=\; \tfrac{1}{m}\sum_{i=1}^m \hat{\ell}_i. \tag{19}$$

The corresponding noise-free target *global-baseline* replaces the MC estimates by expectations:

$$L_{\text{GB}}^{\text{sg}}(S;\theta) = \tfrac{1}{m}\sum_{i=1}^m \lambda_i\big(1 - g\big(s_i(r_i - b_0)\big)\big), \tag{20}$$

where $b_0 = \mathbb{E}_{\text{MC}}[\hat{b}_0] = \frac{1}{m}\sum_{j=1}^m r_j$.

**Lipschitz Constants.** Let $L_g$ and $L_{g'}$ denote the Lipschitz constants of $g$ and $g'$. For $g = \sigma(\beta u)$, we have $L_g = \beta/4$ and $L_{g'} = \beta^2/(6\sqrt{3})$.

**Lemma 2** (Lipschitz constants of the scaled sigmoid)**.** *Let $g(u) = \sigma(\beta u)$ with $\sigma(z) = \frac{1}{1+e^{-z}}$ and $\beta > 0$. Then $g$ is $L_g$-Lipschitz with $L_g = \beta/4$, and its derivative $g'$ is $L_{g'}$-Lipschitz with $L_{g'} = \beta^2/(6\sqrt{3})$.*

*Proof.* Since $g'(u) = \beta\,\sigma(\beta u)(1 - \sigma(\beta u))$ and $\sigma(z)(1 - \sigma(z)) \le \frac{1}{4}$ for all $z$, we have $\sup_u |g'(u)| = \beta/4$, so $g$ is $\beta/4$-Lipschitz.

For $g'$, differentiate once more:

$$g''(u) = \beta^2\,\sigma(\beta u)(1 - \sigma(\beta u))(1 - 2\sigma(\beta u)).$$

Let $p = \sigma(\beta u) \in (0, 1)$ and define $h(p) = p(1 - p)(1 - 2p)$. The extrema of $|h(p)|$ satisfy $h'(p) = 0$, where $h'(p) = 1 - 6p(1 - p) = 0 \implies p_\star = \frac{1}{2}(1 \pm 1/\sqrt{3})$. At these points, $|h(p_\star)| = 1/(6\sqrt{3})$, so $\sup_u |g''(u)| = \beta^2/(6\sqrt{3})$. By the mean-value theorem, $|g'(u) - g'(v)| \le (\sup_t |g''(t)|)\,|u - v|$, giving $L_{g'} = \beta^2/(6\sqrt{3})$. □

**Lemma 3** (Centered-Margin Variance Aggregator)**.** *Fix a batch $S = \{(x_i, y_i, s_i, \lambda_i)\}_{i=1}^m$ and assume the MC design renders $(\widehat{r}_1, \ldots, \widehat{r}_m)$ exchangeable when conditioning on $S$. Let $v(S) := \text{Var}_{\text{MC}}(\widehat{r}_i)$ and $c(S) := \text{Cov}_{\text{MC}}(\widehat{r}_i, \widehat{r}_j)$ for $i \ne j$. With $\widehat{b}_0 = \frac{1}{m}\sum_{j=1}^m \widehat{r}_j$ and $\widehat{\delta}_i = s_i\big(\widehat{r}_i - \widehat{b}_0\big)$, we have*

$$\text{Var}_{\text{MC}}(\widehat{\delta}_i) = \frac{m-1}{m}\big(v(S) - c(S)\big),$$

*and consequently,*

$$\Psi(S) := \tfrac{1}{m}\sum_{i=1}^{m}\mathbb{E}_{\mathrm{MC}}\big[\mathrm{Var}_{\mathrm{MC}}(\widehat{\delta}_i)\big] = \frac{m-1}{m}\big(v(S)-c(S)\big).$$

*Proof.* Since $s_i^2 = 1$, $\mathrm{Var}_{\mathrm{MC}}(\widehat{\delta}_i) = \mathrm{Var}_{\mathrm{MC}}(\widehat{r}_i - \widehat{b}_0)$. Write

$$\widehat{r}_i - \widehat{b}_0 = \Big(1 - \tfrac{1}{m}\Big)\widehat{r}_i - \tfrac{1}{m}\sum_{j\neq i}\widehat{r}_j =: a\,\widehat{r}_i + b\sum_{j\neq i}\widehat{r}_j,$$

with $a = \frac{m-1}{m}$ and $b = -\frac{1}{m}$. By bilinearity of covariance,

$$\mathrm{Var}_{\mathrm{MC}}(\widehat{r}_i - \widehat{b}_0) = a^2\,\mathrm{Var}_{\mathrm{MC}}(\widehat{r}_i) + b^2\,\mathrm{Var}_{\mathrm{MC}}\Big(\sum_{j\neq i}\widehat{r}_j\Big) + 2ab\,\mathrm{Cov}_{\mathrm{MC}}\Big(\widehat{r}_i, \sum_{j\neq i}\widehat{r}_j\Big).$$

Exchangeability of $(\widehat{r}_1, \ldots, \widehat{r}_m)$ implies

$$\mathrm{Var}_{\mathrm{MC}}\Big(\sum_{j\neq i}\widehat{r}_j\Big) = (m-1)v(S) + (m-1)(m-2)c(S), \qquad \mathrm{Cov}_{\mathrm{MC}}\Big(\widehat{r}_i, \sum_{j\neq i}\widehat{r}_j\Big) = (m-1)c(S).$$

Substituting and simplifying with $a = \frac{m-1}{m}$ and $b = -\frac{1}{m}$ yields

$$\mathrm{Var}_{\mathrm{MC}}(\widehat{r}_i - \widehat{b}_0) = \frac{m-1}{m}\big(v(S)-c(S)\big).$$

Conditioned on $S$, $\widehat{\delta}_i$ depends only on MC randomness, so $\mathbb{E}_{\mathrm{MC}}\big[\mathrm{Var}_{\mathrm{MC}}(\widehat{\delta}_i)\big] = \mathrm{Var}_{\mathrm{MC}}(\widehat{\delta}_i)$. Since this value is identical for all $i$, averaging gives

$$\Psi(S) = \tfrac{1}{m}\sum_{i=1}^{m}\mathrm{Var}_{\mathrm{MC}}(\widehat{\delta}_i) = \frac{m-1}{m}\big(v(S)-c(S)\big).$$

□

The result relies only on exchangeability (common variance $v(S)$ and common covariance $c(S)$) and centering by the batch mean $\widehat{b}_0$. The signs $s_i \in \{-1, +1\}$ do not affect the variance since $s_i^2 = 1$.

### C.2 Loss Bias

**Theorem 1 (Restated).** *The minibatch loss bias relative to the global-baseline target satisfies*

$$\Big|\mathbb{E}_{\mathcal{D}}\mathbb{E}_{\mathrm{MC}}[\hat{L}(S)] - \mathbb{E}_{\mathcal{D}}[L_{\mathrm{GB}}^{\mathrm{sg}}(S;\theta)]\Big| \;\le\; \lambda_{\max} L_g\; \mathbb{E}_{\mathcal{D}}\big[\sqrt{\Psi(S)}\big],$$

*where $\Psi(S)$ is the centered-margin variance aggregator defined in equation 8. Here $|\cdot|$ refers to the absolute value.*

*Proof.* We have

$$\widehat{L}(S) = \tfrac{1}{m}\sum_{i=1}^{m}\lambda_i\big(1 - g(\widehat{\delta}_i)\big), \qquad L_{\mathrm{GB}}^{\mathrm{sg}}(S;\theta) = \tfrac{1}{m}\sum_{i=1}^{m}\lambda_i\big(1 - g(\delta_i^{\mathrm{gb}})\big),$$

with $\widehat{\delta}_i = s_i(\widehat{r}_i - \widehat{b}_0)$, $\delta_i^{\mathrm{gb}} = s_i(r_i - b_0)$, $\widehat{b}_0 = \frac{1}{m}\sum_{j=1}^{m}\widehat{r}_j$, and $b_0 = \frac{1}{m}\sum_{j=1}^{m} r_j$. Since $L_{\mathrm{GB}}^{\mathrm{sg}}(S;\theta)$ is deterministic under MC randomness for fixed $S$,

$$\Big|\mathbb{E}_{\mathcal{D}}\mathbb{E}_{\mathrm{MC}}[\widehat{L}(S)] - \mathbb{E}_{\mathcal{D}}[L_{\mathrm{GB}}^{\mathrm{sg}}(S;\theta)]\Big| = \Big|\mathbb{E}_{\mathcal{D}}\big[\mathbb{E}_{\mathrm{MC}}[\widehat{L}(S)] - L_{\mathrm{GB}}^{\mathrm{sg}}(S;\theta)\big]\Big| \;\le\; \mathbb{E}_{\mathcal{D}}\Big[\big|\mathbb{E}_{\mathrm{MC}}[\widehat{L}(S)] - L_{\mathrm{GB}}^{\mathrm{sg}}(S;\theta)\big|\Big],$$

where the inequality uses $|\mathbb{E}[Z]| \le \mathbb{E}[|Z|]$ (triangle inequality for $|\cdot|$). Expanding $\widehat{L}$ and $L_{\mathrm{GB}}^{\mathrm{sg}}$ and applying $|g(u) - g(v)| \le L_g|u - v|$ then yields

$$\left|\mathbb{E}_{\mathcal{D}}\mathbb{E}_{\mathrm{MC}}[\widehat{L}] - \mathbb{E}_{\mathcal{D}}[L_{\mathrm{GB}}^{\mathrm{sg}}]\right| \le L_g\ \mathbb{E}_{\mathcal{D}}\mathbb{E}_{\mathrm{MC}}\left[\frac{1}{m}\sum_{i=1}^{m}\lambda_i\left|\widehat{\delta}_i - \delta_i^{\mathrm{gb}}\right|\right] \le \lambda_{\max}L_g\ \mathbb{E}_{\mathcal{D}}\mathbb{E}_{\mathrm{MC}}\left[\frac{1}{m}\sum_{i=1}^{m}\left|\widehat{\delta}_i - \delta_i^{\mathrm{gb}}\right|\right].$$

Conditioned on $S$, $\mathbb{E}_{\mathrm{MC}}[\widehat{r}_i] = r_i$ and $\mathbb{E}_{\mathrm{MC}}[\widehat{b}_0] = b_0$, so $\mathbb{E}_{\mathrm{MC}}[\widehat{\delta}_i] = \delta_i^{\mathrm{gb}}$ and $\left|\widehat{\delta}_i - \delta_i^{\mathrm{gb}}\right| = \left|\widehat{\delta}_i - \mathbb{E}_{\mathrm{MC}}[\widehat{\delta}_i]\right|$. By Cauchy–Schwarz,

$$\mathbb{E}_{\mathrm{MC}}\left|\widehat{\delta}_i - \mathbb{E}_{\mathrm{MC}}[\widehat{\delta}_i]\right| \le \sqrt{\mathbb{E}_{\mathrm{MC}}\big[(\widehat{\delta}_i - \mathbb{E}_{\mathrm{MC}}[\widehat{\delta}_i])^2\big]} = \sqrt{\mathrm{Var}_{\mathrm{MC}}(\widehat{\delta}_i)},$$

since $\big(\mathbb{E}_{\mathrm{MC}}|X|\big)^2 \le \mathbb{E}_{\mathrm{MC}}[X^2]\mathbb{E}_{\mathrm{MC}}[1] = \mathbb{E}_{\mathrm{MC}}[X^2]$ for $X = \widehat{\delta}_i - \mathbb{E}_{\mathrm{MC}}[\widehat{\delta}_i]$. Therefore,

$$\left|\mathbb{E}_{\mathcal{D}}\mathbb{E}_{\mathrm{MC}}[\widehat{L}] - \mathbb{E}_{\mathcal{D}}[L_{\mathrm{GB}}^{\mathrm{sg}}]\right| \le \lambda_{\max}L_g\ \mathbb{E}_{\mathcal{D}}\left[\frac{1}{m}\sum_{i=1}^{m}\sqrt{\mathrm{Var}_{\mathrm{MC}}(\widehat{\delta}_i)}\right].$$

Using concavity of $\sqrt{\cdot}$ for fixed $S$,

$$\frac{1}{m}\sum_{i=1}^{m}\sqrt{\mathrm{Var}_{\mathrm{MC}}(\widehat{\delta}_i)} \le \sqrt{\frac{1}{m}\sum_{i=1}^{m}\mathrm{Var}_{\mathrm{MC}}(\widehat{\delta}_i)},$$

and noting that the variances are deterministic under MC conditioning, we identify

$$\frac{1}{m}\sum_{i=1}^{m}\mathrm{Var}_{\mathrm{MC}}(\widehat{\delta}_i) = \frac{1}{m}\sum_{i=1}^{m}\mathbb{E}_{\mathrm{MC}}\big[\mathrm{Var}_{\mathrm{MC}}(\widehat{\delta}_i)\big] = \Psi(S).$$

Taking $\mathbb{E}_{\mathcal{D}}$ completes the claim:

$$\left|\mathbb{E}_{\mathcal{D}}\mathbb{E}_{\mathrm{MC}}[\widehat{L}(S)] - \mathbb{E}_{\mathcal{D}}[L_{\mathrm{GB}}^{\mathrm{sg}}(S;\theta)]\right| \le \lambda_{\max}L_g\ \mathbb{E}_{\mathcal{D}}\big[\sqrt{\Psi(S)}\big].$$

□

### C.3 Loss Variance

**Theorem 2 (Restated).** *For the minibatch loss $\hat{L}(S) = \frac{1}{m}\sum_{i=1}^{m}\lambda_i\big(1 - g(\hat{\delta}_i)\big)$ with $g(u) = \sigma(\beta u)$, the MC-induced variance satisfies*

$$\mathrm{Var}_{\mathrm{MC}}(\hat{L}(S)) \ \le\ (\lambda_{\max}L_g)^2\ \mathbb{E}_{\mathcal{D}}\big[\Psi(S)\big].$$

*Proof.* Fix a batch $S$ and write $Z_i := \lambda_i\, g(\widehat{\delta}_i)$ so that

$$\widehat{L}(S) \ =\ \tfrac{1}{m}\sum_{i=1}^{m}\lambda_i\big(1 - g(\widehat{\delta}_i)\big) \qquad \Longrightarrow \qquad \mathrm{Var}_{\mathrm{MC}}(\widehat{L}(S)) = \mathrm{Var}_{\mathrm{MC}}\Big(\tfrac{1}{m}\sum_{i=1}^{m}Z_i\Big),$$

since subtracting the constant $\frac{1}{m}\sum_i\lambda_i$ does not affect variance. Using $\mathrm{Cov}_{\mathrm{MC}}(Z_i, Z_j) \le \sqrt{\mathrm{Var}_{\mathrm{MC}}(Z_i)\mathrm{Var}_{\mathrm{MC}}(Z_j)}$,

$$\begin{aligned}
\mathrm{Var}_{\mathrm{MC}}\Big(\tfrac{1}{m}\sum_{i=1}^{m}Z_i\Big) &= \frac{1}{m^2}\Big(\sum_{i=1}^{m}\mathrm{Var}_{\mathrm{MC}}(Z_i) + 2\sum_{1\le i<j\le m}\mathrm{Cov}_{\mathrm{MC}}(Z_i, Z_j)\Big)\\
&\le \frac{1}{m^2}\Big(\sum_{i=1}^{m}\mathrm{Var}_{\mathrm{MC}}(Z_i) + 2\sum_{1\le i<j\le m}\sqrt{\mathrm{Var}_{\mathrm{MC}}(Z_i)\mathrm{Var}_{\mathrm{MC}}(Z_j)}\Big)\\
&= \frac{1}{m^2}\Big(\sum_{i=1}^{m}\sqrt{\mathrm{Var}_{\mathrm{MC}}(Z_i)}\Big)^2.
\end{aligned}$$

Applying Cauchy–Schwarz again in the form $(\sum_i a_i)^2 \le m \sum_i a_i^2$ with $a_i = \sqrt{\mathrm{Var}_{\mathrm{MC}}(Z_i)}$ gives

$$\mathrm{Var}_{\mathrm{MC}}\Big(\tfrac{1}{m}\sum_{i=1}^{m} Z_i\Big) \;\le\; \frac{1}{m}\sum_{i=1}^{m} \mathrm{Var}_{\mathrm{MC}}(Z_i).$$

Next, use Lipschitz contraction for each $i$: since $u \mapsto \lambda_i g(u)$ is $(\lambda_i L_g)$-Lipschitz,

$$\mathrm{Var}_{\mathrm{MC}}(Z_i) = \mathrm{Var}_{\mathrm{MC}}(\lambda_i g(\widehat{\delta}_i)) \;\le\; (\lambda_i L_g)^2\, \mathrm{Var}_{\mathrm{MC}}(\widehat{\delta}_i),$$

where we used $\mathrm{Var}(f(X)) = \min_a \mathbb{E}_{\mathrm{MC}}[(f(X)-a)^2] \le \mathbb{E}_{\mathrm{MC}}[(f(X)-f(\mathbb{E}_{\mathrm{MC}}X))^2] \le L^2\mathbb{E}_{\mathrm{MC}}[(X-\mathbb{E}_{\mathrm{MC}}X)^2] = L^2\mathrm{Var}(X)$ for an $L$-Lipschitz $f$. Hence,

$$\mathrm{Var}_{\mathrm{MC}}(\widehat{L}(S)) \;\le\; \frac{(\lambda_{\max}L_g)^2}{m}\sum_{i=1}^{m} \mathrm{Var}_{\mathrm{MC}}(\widehat{\delta}_i) \;=\; (\lambda_{\max}L_g)^2\, \Psi(S),$$

using the definition (and MC-determinism) of $\Psi(S) = \frac{1}{m}\sum_{i=1}^{m} \mathrm{Var}_{\mathrm{MC}}(\widehat{\delta}_i)$. Finally, taking $\mathbb{E}_{\mathcal{D}}[\cdot]$ over batches yields

$$\mathbb{E}_{\mathcal{D}}\big[\mathrm{Var}_{\mathrm{MC}}(\widehat{L}(S))\big] \;\le\; (\lambda_{\max}L_g)^2\, \mathbb{E}_{\mathcal{D}}[\Psi(S)].$$

□

### C.4 Gradient Bias

**Theorem 3 (Restated).** *Let $\|\cdot\|$ be the $L_2$-norm. If $\|\nabla_\theta r_i\| \le \bar{G}$ for all items, then*

$$\big\|\mathbb{E}_{\mathcal{D}}\mathbb{E}_{\mathrm{MC}}[\hat{G}(S)] - \mathbb{E}_{\mathcal{D}}[G(S)]\big\| \le \\ \lambda_{\max}L_{g'}\, \mathbb{E}_{\mathcal{D}}\big[\sqrt{\Psi(S)}\big]\, \bar{G} + \lambda_{\max}L_g\, \mathbb{E}_{\mathcal{D}}\big[\sqrt{\bar{v}_\nabla(S)}\big],$$

*where $\bar{v}_\nabla(S) = \frac{1}{m}\sum_{i=1}^{m} \mathbb{E}_{\mathrm{MC}}\big[\|\nabla_\theta \hat{r}_i - \nabla_\theta r_i\|^2\big]$, $\hat{G}(S)$ is the stochastic gradient for the minibatch $S$ as defined in equation 9, and $G$ is defined in Section 4.3.*

*Proof.* We start by expanding the difference of the expectations and splitting it into two parts:

$$\begin{aligned}\mathbb{E}_{\mathcal{D}}\mathbb{E}_{\mathrm{MC}}[\hat{G}(S)] - \mathbb{E}_{\mathcal{D}}[G(S)] &= \frac{1}{m}\sum_{i=1}^{m} \mathbb{E}_{\mathcal{D}}\mathbb{E}_{\mathrm{MC}}\big[\hat{a}_i \nabla_\theta \hat{r}_i - a_i \nabla_\theta r_i\big] \\ &= \frac{1}{m}\sum_{i=1}^{m} \mathbb{E}_{\mathcal{D}}\Big(\underbrace{\mathbb{E}_{\mathrm{MC}}[\hat{a}_i - a_i]}_{\text{(A) weight noise}}\; \nabla_\theta r_i \;+\; \underbrace{\mathbb{E}_{\mathrm{MC}}\big[\hat{a}_i(\nabla_\theta \hat{r}_i - \nabla_\theta r_i)\big]}_{\text{(B) score-gradient noise}}\Big). \end{aligned} \tag{21}$$

**(A) Weight-noise term.** Using $a_i = -\lambda_i s_i g'(\delta_i)$ and $\hat{a}_i = -\lambda_i s_i g'(\hat{\delta}_i)$ with $g'$ $L_{g'}$-Lipschitz,

$$\begin{aligned}\big|\mathbb{E}_{\mathrm{MC}}[\hat{a}_i - a_i]\big| &\le \lambda_i\, \mathbb{E}_{\mathrm{MC}}\big[\,|g'(\hat{\delta}_i) - g'(\delta_i)|\,\big] \le \lambda_i L_{g'}\, \mathbb{E}_{\mathrm{MC}}\big[\,|\hat{\delta}_i - \delta_i|\,\big] \\ &\le \lambda_i L_{g'} \sqrt{\mathbb{E}_{\mathrm{MC}}\big[(\hat{\delta}_i - \delta_i)^2\big]} = \lambda_i L_{g'} \sqrt{\mathbb{E}_{\mathrm{MC}}\big[\mathrm{Var}_{\mathrm{MC}}(\hat{\delta}_i)\big]},\end{aligned}$$

where we used Cauchy–Schwarz $\big(\mathbb{E}_{\mathrm{MC}}\big[|\hat{\delta}_i - \delta_i|.|1|\big] \le \sqrt{\mathbb{E}_{\mathrm{MC}}\big[(\hat{\delta}_i - \delta_i)^2\big]\mathbb{E}_{\mathrm{MC}}\big[1^2\big]}\big)$ and $\mathbb{E}_{\mathrm{MC}}[\hat{\delta}_i] = \delta_i$. Hence,

$$\begin{aligned}\frac{1}{m}\sum_{i=1}^{m} \mathbb{E}_{\mathcal{D}}\Big(\mathbb{E}_{\mathrm{MC}}[\hat{a}_i - a_i]\; \nabla_\theta r_i\Big) &\le \frac{1}{m}\sum_{i=1}^{m} \mathbb{E}_{\mathcal{D}}\big[\,|\mathbb{E}_{\mathrm{MC}}[\hat{a}_i - a_i]|\; \|\nabla_\theta r_i\|\,\big] \\ &\le \frac{\lambda_{\max}L_{g'}}{m}\sum_{i=1}^{m} \mathbb{E}_{\mathcal{D}}\Big[\sqrt{\mathbb{E}_{\mathrm{MC}}\mathrm{Var}_{\mathrm{MC}}(\hat{\delta}_i)}\, \|\nabla_\theta r_i\|\Big] \\ &\le \lambda_{\max}L_{g'}\, \bar{G}\, \mathbb{E}_{\mathcal{D}}\Big[\frac{1}{m}\sum_{i=1}^{m} \sqrt{\mathbb{E}_{\mathrm{MC}}\mathrm{Var}_{\mathrm{MC}}(\hat{\delta}_i)}\Big] \\ &\le \lambda_{\max}L_{g'}\, \bar{G}\, \mathbb{E}_{\mathcal{D}}\big[\sqrt{\Psi(S)}\big],\end{aligned} \tag{22}$$

using the concavity of $x \mapsto \sqrt{x}$, and where we (re)use $\Psi(S) := \frac{1}{m}\sum_{i=1}^{m} \mathbb{E}_{\mathrm{MC}}\mathrm{Var}(\hat{\delta}_i)$.

**(B) Score-gradient noise term.** By Cauchy–Schwarz and the uniform bound $|g'(u)| \le L_g$,

$$\begin{aligned}\big\|\mathbb{E}_{\mathrm{MC}}[\hat{a}_i(\nabla_\theta \hat{r}_i - \nabla_\theta r_i)]\big\| &\le \sqrt{\mathbb{E}_{\mathrm{MC}}[\hat{a}_i^2]}\,\sqrt{\mathbb{E}_{\mathrm{MC}}\big[\|\nabla_\theta \hat{r}_i - \nabla_\theta r_i\|^2\big]} \\ &\le \lambda_{\max} L_g\,\sqrt{\mathbb{E}_{\mathrm{MC}}\big[\|\nabla_\theta \hat{r}_i - \nabla_\theta r_i\|^2\big]}.\end{aligned}$$

Averaging over $i$ and applying the same concavity argument gives

$$\frac{1}{m}\sum_{i=1}^{m} \mathbb{E}_{\mathcal{D}}\Big[\big\|\mathbb{E}_{\mathrm{MC}}[\hat{a}_i(\nabla_\theta \hat{r}_i - \nabla_\theta r_i)]\big\|\Big] \;\le\; \lambda_{\max} L_g\, \mathbb{E}_{\mathcal{D}}\big[\sqrt{\bar{v}_\nabla(S)}\big], \tag{23}$$

with $\bar{v}_\nabla(S) = \frac{1}{m}\sum_{i=1}^{m} \mathbb{E}_{\mathrm{MC}}\big[\|\nabla_\theta \hat{r}_i - \nabla_\theta r_i\|^2\big]$.

**Conclusion.** Combining equation 22 and equation 23 in equation 21 and applying the triangle inequality yields

$$\Big\|\mathbb{E}_{\mathcal{D}}\mathbb{E}_{\mathrm{MC}}[\hat{G}(S)] - \mathbb{E}_{\mathcal{D}}[G(S)]\Big\| \le \lambda_{\max} L_{g'}\, \mathbb{E}_{\mathcal{D}}\big[\sqrt{\Psi(S)}\big]\,\bar{G} \;+\; \lambda_{\max} L_g\, \mathbb{E}_{\mathcal{D}}\big[\sqrt{\bar{v}_\nabla(S)}\big],$$

which is the desired bound. □

This result shows that gradient bias has two contributions: weight noise and score-gradient noise. Weight noise arises from applying $g$ to noisy centered margin $\hat{\delta}_i$, and is controlled by $\Psi(S)$. Score-gradient noise arises from stochastic ELBO gradients controlled by $\bar{v}_\nabla(S)$, which can be reduced by increasing the MC budget.

### C.5 Gradient Variance

Let

$$\begin{aligned}\tilde{G}^2(S) &= \tfrac{1}{m}\sum_{i=1}^{m} \mathbb{E}_{\mathrm{MC}}\big[\|\nabla_\theta \hat{r}_i\|^2\big], \\ \bar{c}_\nabla(S) &= \tfrac{1}{m(m-1)}\sum_{i\neq j} \mathbb{E}_{\mathrm{MC}}[\langle \xi_i, \xi_j\rangle],\end{aligned}$$

where $\xi_i = \nabla_\theta \hat{r}_i - \nabla_\theta r_i$, and assume unbiased ELBO gradients $\mathbb{E}_{\mathrm{MC}}[\nabla_\theta \hat{r}_i] = \nabla_\theta r_i$.

**Theorem 4 (Restated).** *For mini-batch $S$, let $U(S) = \frac{1}{m}\sum_{i=1}^{m} a_i\,\nabla_\theta \hat{r}_i$. Then*

$$\begin{aligned}&\mathrm{Var}_{\mathrm{MC}}\big(\hat{G}(S)\big) \le \\ &\qquad \Big(\sqrt{\mathrm{Var}_{\mathrm{MC}}\big(U(S)\big)} \;+\; \lambda_{\max} L_{g'}\,\sqrt{\Psi(S)}\,\tilde{G}(S)\Big)^2,\end{aligned}$$

*where*

$$\mathrm{Var}_{\mathrm{MC}}(U) \;\le\; (\lambda_{\max} L_g)^2\Big(\frac{\bar{v}_\nabla(S)}{m} \;+\; \frac{m-1}{m}\,\bar{c}_\nabla(S)\Big),$$

*and $\hat{G}(S) := \frac{1}{m}\sum_{i=1}^{m} \hat{a}_i\,\nabla_\theta \hat{r}_i$ is the stochastic gradient of the mini-batch $S$.*

*Proof.* Let

$$U(S) := \frac{1}{m}\sum_{i=1}^{m} a_i\,\nabla_\theta \hat{r}_i, \qquad V(S) := \frac{1}{m}\sum_{i=1}^{m} (\hat{a}_i - a_i)\,\nabla_\theta \hat{r}_i,$$

so that $\hat{G} = U + V$. We also use $\xi_i := \nabla_\theta \hat{r}_i - \nabla_\theta r_i$, so $\mathbb{E}_{\mathrm{MC}}[\xi_i] = 0$.

**(A) Relating $\mathrm{Var}_{\mathrm{MC}}(\hat{G})$ to $\mathrm{Var}_{\mathrm{MC}}(U)$ and $\mathrm{Var}_{\mathrm{MC}}(V)$.** By definition of vector variance,

$$\mathrm{Var}_{\mathrm{MC}}(\hat{G}) = \mathbb{E}_{\mathrm{MC}}\left[\left\|\hat{G} - \mathbb{E}_{\mathrm{MC}}[\hat{G}]\right\|_2^2\right].$$

Since $\hat{G} = U + V$, write

$$\hat{G} - \mathbb{E}_{\mathrm{MC}}[\hat{G}] = (U - \mathbb{E}_{\mathrm{MC}}[U]) + (V - \mathbb{E}_{\mathrm{MC}}[V]) =: A + B.$$

Using the pointwise triangle inequality $\|A+B\|_2 \le \|A\|_2 + \|B\|_2$ and then squaring and taking expectation, we get

$$\mathbb{E}_{\mathrm{MC}}\left[\|A+B\|_2^2\right] \le \mathbb{E}_{\mathrm{MC}}\left[(\|A\|_2 + \|B\|_2)^2\right] = \mathbb{E}_{\mathrm{MC}}\left[\|A\|_2^2\right] + 2\,\mathbb{E}_{\mathrm{MC}}[\|A\|_2\|B\|_2] + \mathbb{E}_{\mathrm{MC}}\left[\|B\|_2^2\right].$$

Apply Cauchy–Schwarz to the cross term:

$$\mathbb{E}_{\mathrm{MC}}[\|A\|_2\|B\|_2] \;\le\; \sqrt{\mathbb{E}_{\mathrm{MC}}[\|A\|_2^2]}\;\sqrt{\mathbb{E}_{\mathrm{MC}}[\|B\|_2^2]}.$$

Combining,

$$\mathbb{E}_{\mathrm{MC}}\left[\|A+B\|_2^2\right] \le \left(\sqrt{\mathbb{E}_{\mathrm{MC}}[\|A\|_2^2]} + \sqrt{\mathbb{E}_{\mathrm{MC}}[\|B\|_2^2]}\right)^2.$$

Recognizing $\mathbb{E}_{\mathrm{MC}}[\|A\|_2^2] = \mathrm{Var}_{\mathrm{MC}}(U)$ and $\mathbb{E}_{\mathrm{MC}}[\|B\|_2^2] = \mathrm{Var}_{\mathrm{MC}}(V)$, we conclude

$$\sqrt{\mathrm{Var}_{\mathrm{MC}}(\hat{G})} \;\le\; \sqrt{\mathrm{Var}_{\mathrm{MC}}(U)} + \sqrt{\mathrm{Var}_{\mathrm{MC}}(V)}. \tag{24}$$

**(B) Bounding $\mathrm{Var}_{\mathrm{MC}}(V)$.** By the Lipschitz property of $g'$,

$$|\hat{a}_i - a_i| = \lambda_i\,|g'(\hat{\delta}_i) - g'(\delta_i)| \le \lambda_i L_{g'}\,|\hat{\delta}_i - \delta_i|.$$

Therefore,

$$\begin{aligned}
\mathrm{Var}_{\mathrm{MC}}(V) &= \mathrm{Var}_{\mathrm{MC}}\Big(\frac{1}{m}\sum_i (\hat{a}_i - a_i)\,\nabla_\theta \hat{r}_i\Big) \;\le\; \mathbb{E}_{\mathrm{MC}}\Big\|\frac{1}{m}\sum_i (\hat{a}_i - a_i)\,\nabla_\theta \hat{r}_i\Big\|_2^2 \\
&= \frac{1}{m^2}\sum_{i,j} \mathbb{E}_{\mathrm{MC}}\big[(\hat{a}_i - a_i)(\hat{a}_j - a_j)\,\langle \nabla_\theta \hat{r}_i, \nabla_\theta \hat{r}_j\rangle\big] \\
&\le \frac{1}{m^2}\Big(\sum_i \sqrt{\mathbb{E}_{\mathrm{MC}}\big[(\hat{a}_i - a_i)^2\,\|\nabla_\theta \hat{r}_i\|_2^2\big]}\Big)^2 \quad \text{(Cauchy–Schwarz across } i\text{)} \\
&\le \frac{1}{m^2}\Big(\sum_i \sqrt{\mathbb{E}_{\mathrm{MC}}[(\hat{a}_i - a_i)^2]}\;\sqrt{\mathbb{E}_{\mathrm{MC}}\|\nabla_\theta \hat{r}_i\|_2^2}\Big)^2 \quad \text{(Cauchy–Schwarz in expectation)} \\
&\le \lambda_{\max}^2 L_{g'}^2 \Big(\frac{1}{m}\sum_i \sqrt{\mathbb{E}_{\mathrm{MC}}[(\hat{\delta}_i - \delta_i)^2]}\;\sqrt{\mathbb{E}_{\mathrm{MC}}\|\nabla_\theta \hat{r}_i\|_2^2}\Big)^2 \\
&\le \lambda_{\max}^2 L_{g'}^2 \Big(\frac{1}{m}\sum_i \mathbb{E}_{\mathrm{MC}}[(\hat{\delta}_i - \delta_i)^2]\Big)\Big(\frac{1}{m}\sum_i \mathbb{E}_{\mathrm{MC}}\|\nabla_\theta \hat{r}_i\|_2^2\Big) \quad \text{(Cauchy–Schwarz on the index } i\text{)} \\
&= \lambda_{\max}^2 L_{g'}^2\;\Psi(S)\;\tilde{G}^2(S).
\end{aligned}$$

Taking square roots yields

$$\sqrt{\mathrm{Var}_{\mathrm{MC}}(V)} \;\le\; \lambda_{\max} L_{g'}\;\sqrt{\Psi(S)}\;\tilde{G}(S). \tag{25}$$

**(C) Bounding $\mathrm{Var}_{\mathrm{MC}}(U)$.** Since $\mathbb{E}_{\mathrm{MC}}[\xi_i] = 0$ and $U - \mathbb{E}_{\mathrm{MC}}[U] = \frac{1}{m}\sum_i a_i\,\xi_i$ with deterministic $a_i$,

$$\mathrm{Var}_{\mathrm{MC}}(U) = \mathbb{E}_{\mathrm{MC}}\Big\|\frac{1}{m}\sum_{i=1}^m a_i\,\xi_i\Big\|_2^2 = \frac{1}{m^2}\sum_{i,j=1}^m a_i a_j\,\mathbb{E}_{\mathrm{MC}}\big[\langle \xi_i, \xi_j\rangle\big]. \tag{26}$$

Using $|a_i| \le \lambda_{\max}|g'(\delta_i)| \le \lambda_{\max} L_g$ and splitting diagonal/off-diagonal terms,

$$\begin{aligned}
\mathrm{Var}_{\mathrm{MC}}(U) &\le \frac{(\lambda_{\max} L_g)^2}{m^2}\Big(\sum_{i=1}^{m} \mathbb{E}_{\mathrm{MC}}\|\xi_i\|_2^2 + \sum_{i\neq j} \mathbb{E}_{\mathrm{MC}}\langle \xi_i, \xi_j\rangle\Big) \\
&= (\lambda_{\max} L_g)^2\Big(\frac{1}{m}\cdot\frac{1}{m}\sum_{i=1}^{m} \mathbb{E}_{\mathrm{MC}}\|\xi_i\|_2^2 + \frac{m-1}{m}\cdot\frac{1}{m(m-1)}\sum_{i\neq j} \mathbb{E}_{\mathrm{MC}}\langle \xi_i, \xi_j\rangle\Big) \\
&= (\lambda_{\max} L_g)^2\Big(\frac{\bar{v}_\nabla(S)}{m} + \frac{m-1}{m}\ \bar{c}_\nabla(S)\Big),
\end{aligned}$$

which is the claimed proxy bound.

**(D) Combine.** From equation 24 and equation 25,

$$\sqrt{\mathrm{Var}_{\mathrm{MC}}(\hat{G}(S))}\ \le\ \sqrt{\mathrm{Var}_{\mathrm{MC}}(U(S))}\ +\ \lambda_{\max} L_{g'}\ \sqrt{\Psi(S)}\ \tilde{G}(S).$$

Squaring both sides gives

$$\mathrm{Var}_{\mathrm{MC}}(\hat{G}(S))\ \le\ \Big(\sqrt{\mathrm{Var}_{\mathrm{MC}}(U(S))} + \lambda_{\max} L_{g'}\ \sqrt{\Psi(S)}\,\tilde{G}(S)\Big)^2.$$

Together with the bound on $\mathrm{Var}_{\mathrm{MC}}(U)$ above, this establishes the theorem. □

For the special case of independent per–item MC seeds, we can assume that the per–item MC randomness is conditionally independent across items in the batch, so that $\mathbb{E}_{\mathrm{MC}}[\langle \xi_i, \xi_j\rangle] = 0$ for $i \neq j$. Then the variance of $U(S)$ collapses to the diagonal:

$$\mathrm{Var}_{\mathrm{MC}}\big(U(S)\big) = \frac{1}{m^2}\sum_{i=1}^{m} a_i^2\, \mathbb{E}_{\mathrm{MC}}\big[\|\xi_i\|^2\big]\ \le\ (\lambda_{\max} L_g)^2\ \frac{\bar{v}_\nabla(S)}{m},$$

where $\bar{v}_\nabla(S) = \frac{1}{m}\sum_{i=1}^{m} \mathbb{E}_{\mathrm{MC}}\big[\|\xi_i\|^2\big]$. Substituting into Theorem 4 gives

$$\mathrm{Var}_{\mathrm{MC}}\big(\hat{G}(S)\big) \le \Big((\lambda_{\max} L_g)\sqrt{\tfrac{\bar{v}_\nabla(S)}{m}}\ +\ \lambda_{\max} L_{g'}\ \sqrt{\Psi(S)}\ \tilde{G}(S)\Big)^2.$$

### C.6 Global Baseline Optimality

**Lemma 1 (Restated).** *For any baseline $b$ that is constant across items in a batch and may depend on MC randomness,*

$$\Psi_b(S) - \Psi_{\hat{b}_0}(S)\ =\ \mathrm{Var}_{\mathrm{MC}}\big(b - \hat{b}_0\big)\ \ge\ 0,$$

*where $\hat{b}_0 = \frac{1}{m}\sum_j \hat{r}_j$. Hence every baseline of the form $b = \hat{b}_0 + K$ with deterministic constant $K$ attains the same minimum value $\Psi_b(S) = \Psi_{\hat{b}_0}(S)$. If, in addition, $\mathbb{E}_{\mathrm{MC}}[b] = b_0$, then $K = 0$ and the unique minimizer is $b = \hat{b}_0$.*

*Proof.* Let $\Delta := b - \hat{b}_0$. Using $\mathrm{Var}(X - Y) = \mathrm{Var}X + \mathrm{Var}Y - 2\mathrm{Cov}(X, Y)$ and $\Psi(S)$ from Lemma 3,

$$\Psi_b - \Psi_{\hat{b}_0} = \frac{1}{m}\sum_i \mathbb{E}_{\mathrm{MC}}\big[\mathrm{Var}((\hat{r}_i - \hat{b}_0) - \Delta)\big] - \frac{1}{m}\sum_i \mathbb{E}_{\mathrm{MC}}\big[\mathrm{Var}(\hat{r}_i - \hat{b}_0)\big].$$

This equals

$$\mathrm{Var}(\Delta) - \frac{2}{m}\sum_{i=1}^{m}\mathrm{Cov}(\hat{r}_i - \hat{b}_0, \Delta) = \mathrm{Var}(\Delta) - 2\,\mathrm{Cov}(\hat{b}_0 - \hat{b}_0, \Delta) = \mathrm{Var}(\Delta) \ge 0,$$

since $\frac{1}{m}\sum_i \mathrm{Cov}(\hat{r}_i, \Delta) = \mathrm{Cov}(\hat{b}_0, \Delta)$. Equality holds iff $\mathrm{Var}(\Delta) = 0$, i.e. $b - \hat{b}_0$ is a constant. If moreover $\mathbb{E}_{\mathrm{MC}}[b] = b_0$, then that constant must be 0, so $b = \hat{b}_0$. □

This result says that $\hat{b}_0$ is variance-optimal for all possible values of $b$. It requires no additional compute and is thus chosen as the principled default for ELBO-KTO.

Table 6: Licenses for datasets used in this work.

| Dataset | License |
|---|---|
| kto-mix-14k | MIT |
| UltraFeedback-Binary | MIT |
| GSM8K | MIT |
| HumanEval | MIT |
| MMLU | MIT |
| GPQA | CC-BY-4.0 |
| HellaSwag | MIT |

# D Experimental Details

This section specifies datasets, models, preprocessing, optimization, compute, decoding, and evaluation protocols used in our experiments.

## D.1 Models and Tokenization

We fine-tune LLaDA-8B-Instruct as the policy and use a frozen copy of the same checkpoint as the reference $\pi_{\text{ref}}$ for ELBO-KTO. We apply the official chat template during both training and inference for consistency with the base model.[1]

## D.2 Data and Preprocessing

**Preference Data.** We use kto-mix-14k and UltraFeedback-Binary for training. Each example is $(x, y, s)$ with $s \in \{+1, -1\}$ indicating desirable/undesirable. For UltraFeedback-Binary, we convert pairs to unpaired labels by taking the *chosen* response as desirable and the *rejected* response as undesirable. Unless noted, we train on the official train split and report on the test split.

**Length and Formatting.** We cap the concatenated prompt+response length at $L$=4096 tokens, pad with `|EOS|` to length 4096, and *only* mask completion tokens during training. Because we fine-tune from LLaDA-8B-Instruct, the chat template is applied at train and inference time.

**Licenses.** Dataset licenses used in the paper are summarized in Table 6.

## D.3 Training

**Optimizer and Schedule.** We train for one epoch with batch size 8 using AdamW (weight decay 0.01, $\beta_1$=0.9, $\beta_2$=0.95), a 3% linear warmup, and cosine decay.

**Learning Rate and MC Budget.** For kto-mix-14k: peak learning-rate $1\times10^{-6}$ with 8 MC samples per example. For UltraFeedback-Binary (UFB): peak learning-rate $5\times10^{-7}$ with 4 MC samples per example. The same UFB-trained checkpoint is used for downstream generalization experiments in Section 5.5.

**Hyperparameters.** For the results reported in Tables 1-3 we set $\lambda_D = \lambda_U = 1$. For results in Tables 1-2, we set $\beta = 0.1$ for kto-mix-14k and $\beta = 0.2$ for UltraFeedback-Binary. For the ablation study in Table 3, we use kto-mix-14k and report the values of $\beta$ and learning-rate in the table. For downstream results in Section 5.5, we use the ELBO-KTO model trained on UltraFeedback-Binary used in Section 5.2.

**Reference Precompute.** For memory efficiency during training we precompute $\widehat{B}_{\pi_{\text{ref}}}$ once and cache the mask metadata, so we do not keep both the policy and reference models resident simultaneously. This

[1] `https://huggingface.co/GSAI-ML/LLaDA-8B-Instruct`

choice does not alter the ELBO-KTO objective or estimator; it only reduces memory/compute overhead. Implementations with sufficient memory can compute $\widehat{B}_{\pi_{\text{ref}}}$ on-the-fly by holding both models at once and will obtain identical updates (we share the same masks via cached metadata either way).

**Compute.** Unless specified, experiments run on 8× NVIDIA H100 80GB with FSDP (no offload) and per-GPU microbatch = 1 (global batch = 8). Typical wall-clock: precompute $\widehat{B}_{\pi_{\text{ref}}}$ on ∼13.5k samples ≈ 37 min; training after precompute ≈ 172 min. We include precompute timings only for reproducibility/accounting, not as a core component of the method.

### D.4 Decoding and Evaluation Protocol

Unless noted otherwise, we use temperature 0.0, classifier-free guidance 0.0, and low-confidence remasking strategy for all evaluation purposes as recommended by official LLaDA evaluation script[2]. For Sections 5.2-5.4, we use generation length 512, block length 32, 512 diffusion steps. For evaluation on downstream tasks in Section 5.5, task-specific parameters are in Table 7. We evaluate LLaDA-based instruction-tuned models with conditional generation, following Nie et al. (2025). Because their public repo lacks scripts for conditional evaluation of LLaDA-8B-Instruct, we implement an evaluation script on top of `lm-evaluation-harness`[3].

Table 7: Evaluation configuration per task (gen length, block length, diffusion steps, #few-shot).

| Task | Gen Len | Block Len | Steps | Few-shot |
|---|---|---|---|---|
| GSM8K | 256 | 8 | 256 | 5 |
| MMLU | 3 | 3 | 3 | 5 |
| HellaSwag | 3 | 3 | 3 | 0 |
| HumanEval | 512 | 512 | 32 | 0 |
| GPQA | 128 | 128 | 64 | 5 |

### D.5 Win-Rate, Majority-Vote and Cohen's $\kappa$

**Win-rate protocol.** We report *Adjusted Win Rate* (AWR) of the tuned model versus the base model using an open-source LLM-as-a-judge (FastChat `lm_judge`).[4] For each prompt $x$, we generate two completions under identical decoding: $y^{(A)}$ from the tuned model and $y^{(B)}$ from the base model. The judge receives $(x, y^{(A)}, y^{(B)})$ with a standardized instruction template and returns a label in $\{\text{A wins}, \text{B wins}, \text{tie}\}$. To mitigate position bias, we evaluate *both* orderings, $(x, y^{(A)}, y^{(B)})$ and $(x, y^{(B)}, y^{(A)})$, and declare a win for a model only if *both* orderings favor the same model; otherwise the outcome is counted as a tie. We then compute

$$\text{AWR} \;=\; \frac{\#\text{A wins} + 0.5\,\#\text{ties}}{\#\text{A wins} + \#\text{B wins} + \#\text{ties}}\,,$$

and report the mean AWR over the full test set.

**Two-judge majority vote and agreement.** We also report AWR under a majority vote of two independent judges. The same two-ordering rule is applied per judge; the ensemble outcome is a win for a model only if *both* judges, under *both* orderings, favor that model; otherwise it is treated as a tie. Inter-judge agreement is measured with Cohen's $\kappa$, treating win, loss, and tie as three distinct classes.

**Uncertainty.** We report 90% confidence intervals via nonparametric bootstrapping with 5,000 resamples of the test set (sampling prompts with replacement and recomputing metrics per resample); intervals are taken from the $5^{\text{th}}$ and $95^{\text{th}}$ percentiles.

[2]https://github.com/ML-GSAI/LLaDA
[3]https://github.com/EleutherAI/lm-evaluation-harness
[4]https://github.com/lm-sys/FastChat