Aligning Diffusion Language Models via Unpaired Preference Optimization

Vaibhav Jindal **Hejian Sang** **Chun-Mao Lai** **Yanning Chen** **Zhipeng Wang**

ReALM-Gen Workshop at ICLR 2026 · April 27, 2026 · Rio de Janeiro

Motivation

Diffusion language models (dLLMs) are an emerging alternative to autoregressive generators, refining sequences in parallel rather than decoding left-to-right, enabling significantly faster inference.

Aligning dLLMs with human preferences is important, but challenging:

→ Standard alignment methods (DPO, KTO) require log-likelihood ratios, which are **intractable** for dLLMs
→ VRPO (Zhu et al., 2025) addresses intractability using ELBO surrogates, but still requires **paired** preference data (chosen vs. rejected)
→ In practice, binary "good/bad" feedback is far cheaper and more abundant than paired annotations

Can we align dLLMs from unpaired binary feedback alone?

KTO: Kahneman-Tversky Optimization

KTO (Ethayarajh et al., 2024) aligns LLMs from **unpaired** binary feedback, with asymmetric weights for desirable (λ_D) and undesirable (λ_U) examples.

Reward: $r_\theta(x, y) = \log \pi_\theta(y \mid x) / \pi_{\text{ref}}(y \mid x)$

Value function (asymmetric weights):

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

Baseline: $z_0(x) = \mathrm{KL}(\pi_\theta(\cdot \mid x) \,\|\, \pi_{\text{ref}}(\cdot \mid x))$

Loss: $\mathcal{L}_{\text{KTO}}(\theta) = \mathbb{E}_{x,y}\left[\lambda_y - v(x, y)\right]$

For AR LLMs, $\log \pi_\theta(y|x)$ factorizes autoregressively:

$$\log \pi_\theta(y|x) = \sum_{t=1}^{T} \log \pi_\theta(y_t \mid y_{<t}, x)$$

This factorization does not exist for dLLMs, making both r_θ and z_0 intractable.

We need to adapt KTO for the diffusion setting.

ELBO-KTO: Our Method

Solving intractability. We replace the intractable $\log \pi_\theta(y|x)$ with an MC ELBO estimate:

$$\hat{\mathcal{B}}_\pi(y|x) = \frac{1}{n_t} \sum_{j=1}^{n_t} \frac{1}{n_{y_t}} \sum_{i=1}^{n_{y_t}} \ell_\pi\left(y_{t_j}^{(i)}, t_j, y|x\right)$$

where ℓ_π is the per-step mask prediction loss.

We define the **ELBO margin** as a proxy for the reward:

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

A new baseline for dLLMs. The KTO baseline $z_0 = \mathrm{KL}(\pi_\theta \,\|\, \pi_{\text{ref}})$ is intractable. Rather than approximating it, we propose a **different** baseline: the global per-batch mean of ELBO margins:

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

→ Stop-gradient, zero additional compute
→ **Control variate** that reduces gradient variance
→ Recenters scores to high-slope sigmoid region, avoiding loss saturation

ELBO-KTO Loss:

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

$g(u) = \sigma(\beta u)$. Only $\hat{\mathcal{B}}_{\pi_\theta}$ receives gradients.

Theoretical Guarantees

Replacing log-likelihoods with ELBO introduces MC noise. **How bad is it?**

A key quantity governing all bounds: the **centered-margin variance aggregator**

$$\Psi(S) := \frac{m-1}{m}(v(S) - c(S))$$

$v(S) = \mathrm{Var}_{\mathrm{MC}}(\hat{r}_i), \quad c(S) = \mathrm{Cov}_{\mathrm{MC}}(\hat{r}_i, \hat{r}_j)$ for $i \neq j$

Key results:

→ Loss bias: $O\left(\sqrt{\Psi(S)}\right)$, variance: $O(\Psi(S))$
→ Gradient bias and variance similarly controlled
→ Shrinking $\Psi(S)$ tightens all bounds

How to shrink $\Psi(S)$:

✓ Decrease $v(S)$: more MC samples, shared draws between policy/ref
✓ Increase $c(S)$: shared draws across batch items

Baseline Optimality:

$\hat{b}_0$ minimizes Ψ among all constant baselines at zero extra cost:

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

The noise is bounded, controllable, and shrinkable.

Main Results

Setup: Fine-tune LLaDA-8B-Instruct for a **single epoch**. Datasets: kto-mix-14k (13.5k unpaired prompts) and UltraFeedback-Binary (61.1k).

Adjusted win rates (AWR, %) vs. base model, judged by gpt-4o-mini:

Method	kto-mix-14k	UFB
ELBO-KTO (Ours)	**65.9**	**62.3**
Chosen (label=True)	70.1	61.6
Rejected (label=False)	47.3	40.0

→ ELBO-KTO wins in a **clear majority** using only **unpaired** binary labels

→ On UFB, **exceeds** the chosen-target win rate (62.3 vs 61.6)

Judge robustness: Consistent across two judges released one year apart (Cohen's κ = 0.56–0.61).

Downstream Generalization & Class Imbalance

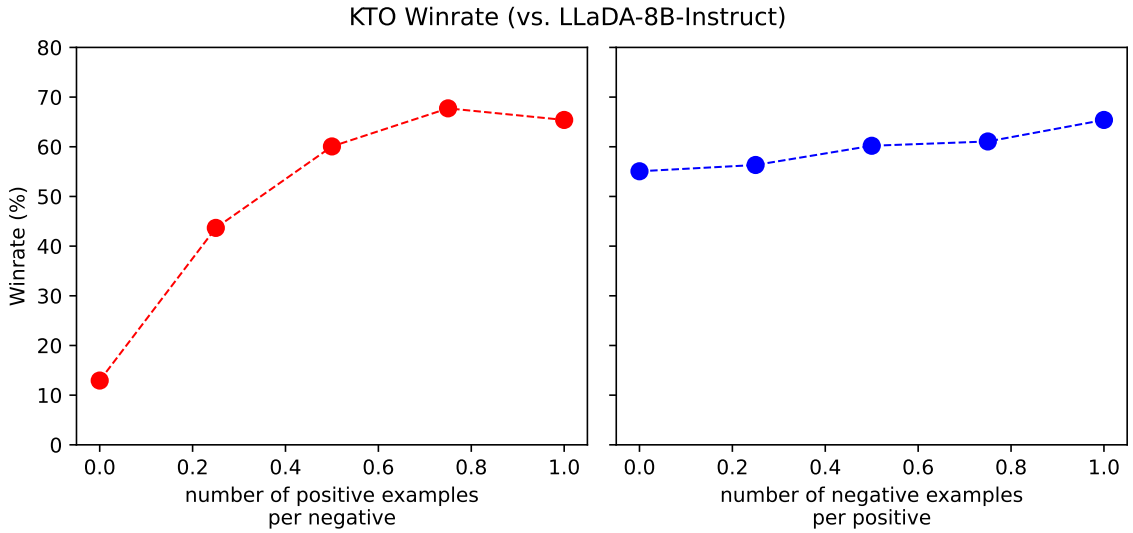
Downstream Generalization

We evaluate ELBO-KTO (trained on UltraFeedback-Binary) on downstream benchmarks:

Task	Base	ELBO-KTO
GSM8K (5)	79.53	**82.79**
MMLU (5)	63.85	**64.43**
GPQA (5)	29.02	**29.69**
HumanEval (0)	42.68	42.68
HellaSwag (0)	**78.03**	77.28

Alignment preserves downstream capabilities.

Class Imbalance Analysis



Reducing desirable examples hurts far more, consistent with **gain sensitivity** from prospect theory.

Ablations: Validating Design Choices

Effect of Global Per-Batch Baseline

β	LR	No baseline	Global
0.2	5×10^{-6}	55.46	**64.80 (+9.34)**
0.1	1×10^{-6}	60.63	**65.90 (+5.27)**
0.2	1×10^{-6}	57.40	**64.73 (+7.33)**

+5 to +9 pp improvement with zero extra compute.

MC Estimator Design

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 ($n = 8$)	
$n_t = 1, n_{y_t} = 8$	**66.0**
$n_t = 4, n_{y_t} = 2$	64.6
$n_t = 8, n_{y_t} = 1$	65.9
(c) Shared random draws	
CRN = False	61.1
CRN = True	**65.9 (+4.8)**

More MC samples and shared draws both reduce $\Psi(S)$, confirming theory.

Conclusion

ELBO-KTO: first unpaired preference method for diffusion LLMs

- Replaces intractable log-likelihoods in KTO with MC ELBO surrogates
- Proposes a variance-optimal per-batch baseline for dLLMs
- 65.9% and 62.3% AWR vs. base model on two benchmarks using only binary feedback
- Theoretical guarantees bounding bias and variance, with practical strategies to reduce them
- Preserves downstream reasoning, knowledge, and code performance

Code: github.com/vaibhavjindal/elbo-kto
Paper: arxiv.org/abs/2510.23658