# BDPO: Directly Optimizing Your Block Diffusion in Energy View

**Anonymous ACL submission**

## Abstract

Recently, Block Diffusion Language Model (BDLM) has been a new paradigm of Large Language Model, which demonstrates remarkable success in parallel decoding of text generation. As a critical component of post-training, the application of Reinforcement Learning from Human Feedback (RLHF) to BDLMs remains a formidable and high-stakes endeavor, becoming the frontier of current research. Current methods of RLHF for BDLMs are dominated by on-policy paradigms, which suffer from inherent computational inefficiencies and expensive resource overhead. In this paper, we introduce an off-policy method, which is called **B**lock **D**irect **P**reference **O**timization (BDPO). By calculating block-level preference scores and aggregating them in the energy-based preference modeling, BDPO achieves a stable and lightweight off-policy optimization for BDLMs. We conducted extensive experiments across a diverse suite of state-of-the-art open-source BDLMs and comprehensive benchmarks. The results show that BDPO can fine-tune BDLMs to align with human preferences and surpass existing off-policy method by 31.7%. Notably, fine-tuning with BDPO improves response quality in multiple tasks while being substantially simple to implement and train.

## 1 Introduction

Discrete Diffusion Large Language Models (dLLMs) (Yu et al., 2025) have recently become a promising direction. Masked Diffusion Language Models (MDLMs) (Sahoo et al., 2024) like LLaDA (Nie et al., 2025), Dream (Ye et al., 2025), support parallel generation by reconstructing sequences from random masks, which offer a different architecture of large-scale text generation model. However, there are significant bottlenecks for MDLMs lie in their inability to native KV-cache support due to bidirectional attention dependencies, and requirement for a pre-defined sequence length for response, which undermine their practical viability in engineering applications. As the result of the ongoing exploration, Block Diffusion Language Models (BDLMs) (Arriola et al., 2025) operate more efficiently through a semi-autoregressive way, which gained properties of both KV-cache support and parallel decoding. Because of that, recent advanced dLLMs are all based on BDLM, such as LLaDA 2.X (Bie et al., 2025), SDAR (Cheng et al., 2025) and WeDLM (Liu et al., 2025).

Reinforcement Learning from Human Feedback (RLHF) (Christiano et al., 2023) is currently significant step of Large Language Models (LLMs)' post-training, which has made a great success on AutoRegressive Models (ARMs) to align them with human preference. Most of existing RLHF methods are designed for ARMs, relying on calculation of per-token probability, which meets ARMs' property of per-token prediction. So that dLLMs, which predict multi-tokens, have an intractable problem to acquire exact log-likelihoods, that is a natural barrier to applying a naive RLHF method.

Direct Preference Optimization (DPO) (Rafailov et al., 2024) is a classical off-policy method of RLHF, which optimizes the model from human preference data. With this pairwise preference, DPO elegantly eliminates the need for explicitly training a reward model and avoids the instability of on-policy reinforcement learning. As the result, DPO enjoys several attractive properties: lightweight, stable, and straightforward to implement, making it a widely adopted approach for aligning ARMs with human preferences. DPO critically relies on the availability of exact log-likelihoods, which are computed by accumulating per-token log-probabilities under the autoregressive factorization. However, this condition no longer holds for dLLMs: they predict multi-tokens through iterative denoising, rendering the exact log-likelihoods of a complete response intractable.

Motivated by the success of Group Sequence Policy Optimization (GSPO) by Zheng et al. (2025), recent work has begun to explore human preference optimization through calculating aggregation of multiple tokens. Specifically, GSPO optimizes the model by updating the policy to maximize the expected sequence-level reward in a group of responses. This idea meets the multi-token prediction characteristic of dLLMs: the log-probability of single token is unavailable, but rewarding dozen of tokens is feasible. Inspired by GSPO, we noticed a natural question: can we skip the exact log-likelihoods, and directly optimize BDLMs with whole blocks?

We argue that the answer is yes. In this paper, we introduce BDPO: **B**lock **D**irect **P**reference **O**ptimization. Our core observation is that Block Diffusion Language Models (BDLMs) naturally segment generation into blocks via their semi-autoregressive architecture, which provides a principled and architecture-native unit for preference aggregation. And this aggregated preference score meets the unique Maximum Likelihood Estimator (MLE) of Energy-Based Models (EBMs) (Hong et al., 2024b), which can be approximated to replace Bradley-Terry Preference Model with EBMs. Building on this, BDPO treats each block as a "super-token" and formulates preference optimization at the block level. By replacing per-token likelihood ratios in standard DPO with block-level energy differences, BDPO achieves a stable and lightweight off-policy optimization for BDLMs.

## 2 Related Work

**Discrete Diffusion Language Models.** Recently Discrete diffusion language models have emerged as a compelling alternative to autoregressive models for text generation. Pioneering work by Austin et al. (2023) introduced Discrete Denoising Diffusion Probabilistic Models (D3PMs), which generate text by iteratively denoising sequences of masked tokens through a learned reverse diffusion process. D3PMs provide multiple choices of its Markov transition matrices for the diffusion forward process: Uniform, Absorbing state, Discretized Gaussian, and Token embedding distance. Building on D3PMs, Sahoo et al. (2024) advance absorbing state (i.e., masking) diffusion through defining the diffusion process for a categorical random variable and extending this process to sequence containing multiple such categorical variables, and denote the approach as Masked Diffusion Language Models (MDLM). And Gat et al. (2024) proposed Discrete Flow Matching (DFM), which works with probability paths interpolating between source and target distributions. By scaling DFM up to 1.7B parameters, their work proves capability of generating high-quality discrete data in non-autoregressive fashion.

LLaDA demonstrated the first large-scale MDLM, showing that diffusion models can match the quality of AR baselines such as LLaMA 3 (Grattafiori et al., 2024). Dream and diffuLLaMA (Gong et al., 2025) is built through adaption from ARMs, narrowing the gap between ARMs and MDLMs. However, MDLMs suffer from two limitations: the inability to native KV-cache support due to bidirectional attention, and the requirement to pre-define the full sequence length before generation. Through some works such as dInfer (Ma et al., 2025) and Fast-dLLM (Wu et al., 2025) providing a KV-cache manager, there is still a gap between MDLMs and ARMs in practice.

To address these issues, Arriola et al. (2025) proposed Block Diffusion Language Models (BDLMs), which operate in a semi-autoregressive fashion: the response is partitioned into blocks that are generated sequentially, while tokens within each block are generated in parallel via iterative denoising. This architecture elegantly combines KV-cache support with parallel decoding, making it the foundation of several recent state-of-the-art dLLMs. Notable examples include LLaDA 2.X, which scales BDLMs to over 100B parameters; SDAR, which explores the synergy between diffusion and autoregression; and WeDLM, which reconciles diffusion objectives with causal attention mechanisms.

**Reinforcement Learning from Human Feedback for LLMs.** Reinforcement Learning from Human Feedback (RLHF) has become a cornerstone of LLM post-training. Proximal Policy Optimization (PPO) by Schulman et al. (2017) is the most notably pioneering method of RLHF for LLMs, which optimizes the policy through a reward model trained on human preference data. While effective, PPO is computationally expensive sensitive to hyper-parameters, and require maintaining multiple models simultaneously. Specifically, a standard PPO pipeline involves at least four models: the policy, reward, critic, reference models, which imposing substantial engineering burdens. Subsequent research has largely focused on reducing

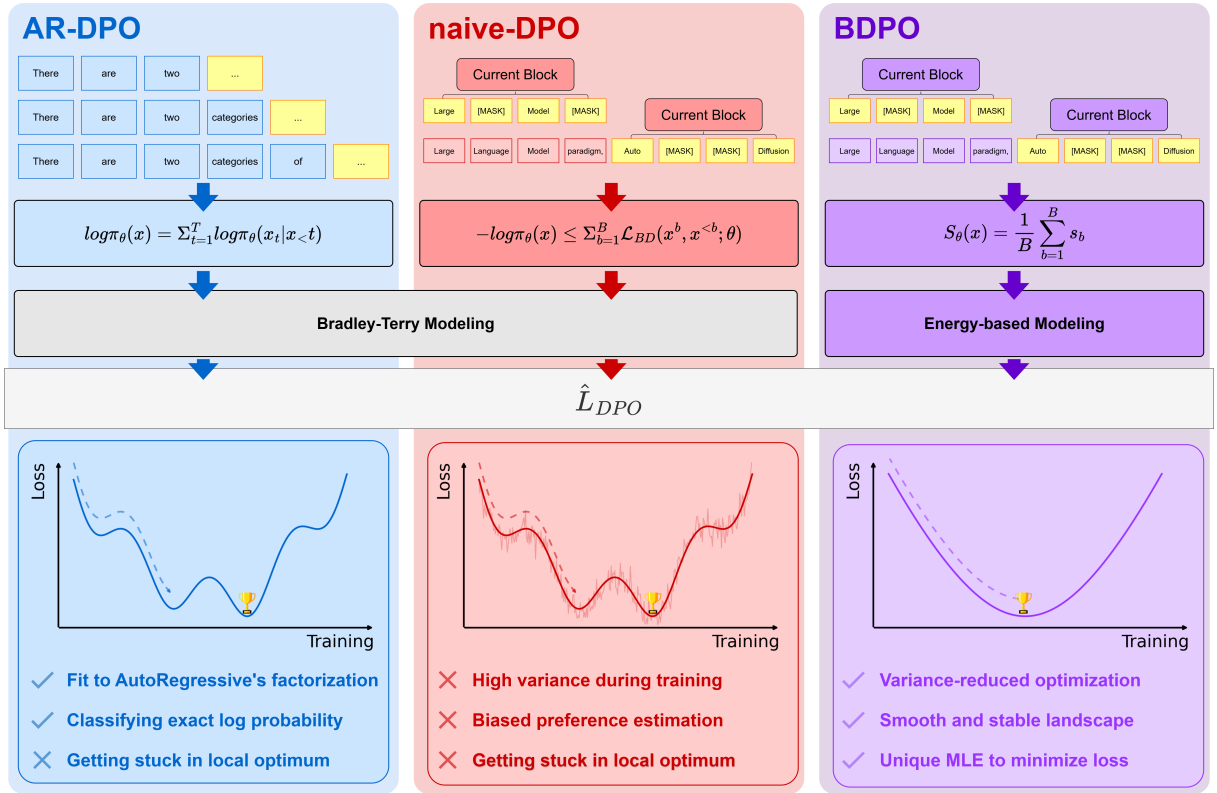


Figure 1: An illustration of the contribution of the paper. Our core argument contains two key points: Block-level preference score constructed from denoising confidence, and the energy-based preference modeling for Block Diffusion Language Models (BDLMs). The advantage of Block-level preference score is that denoising confidence is a architecture-native signal of BDLMs, which can effectively reduce computational complexity. And the energy-based preference modeling could provide a guaranteed unique existence of Maximum Likelihood Estimator (MLE), which is identical to the minimizer of the RLHF loss.

this model count. Direct Preference Optimization (DPO) through re-parameterizing the reward function in terms of the policy itself, eliminates the reward and critic models entirely, and optimizes the policy directly from preference data without online sampling. And ORPO (Hong et al., 2024a), SimPO (Meng et al., 2024), and KTO (Ethayarajh et al., 2024) went further by eliminating the reference model or reduce the need of pair-wise preference data. As the on-policy optimization paradigm, Group Relative Policy Optimization (GRPO) by Shao et al. (2024) obviates the critic model by using the average reward of multiple sampled outputs, produced in response to the same prompt, as the baseline. The success of DeepSeek-R1 (Guo et al., 2025) demonstrates the advanced performance of GRPO. And Group Sequence Policy Optimization (GSPO) noted a key limitation of GRPO that the mismatch between the sequence-level reward and token-level importance ratio, and resolved it by performing a sequence-level probabilistic computation to improve the stability.

But these methods are all designed for ARMs, which rely on per-token log-probabilities to compute the loss. To optimize dLLMs, several previous works have explored RLHF methods designed for multi-token prediction paradigm. Zhu et al. (2025) proposed Variance-Reduced Preference Optimization (VRPO), which could be applied to MDLMs by Evidence Lower Bound (ELBO) (Lou et al., 2024) estimators. And Yang et al. (2025) explored a multi-modal RLHF method for MDLMs. To BDLMs, Wang et al. (2025) and Zhu et al. (2026) proposed their on-policy methods, which reach great performance on optimizing BDLMs. However, these on-policy paradigms suffer from inherent computational inefficiencies through the requirement of expensive online rollouts from the current policy, limiting their practical scalability.

## 3 Method

In this section, we will define our BDPO with reconstruction consistency inner the block decoding process. For that end, we start with a bitter review

of the failure of naive DPO on BDLMs.

### 3.1 Naive DPO failed on dLLMs

To original DPO for ARMs, following the prior work (Peters and Schaal, 2007; Peng et al., 2019), it is straightforward to give the optimal solution of its KL-constrained reward maximization objective:

$$\pi_r(y \mid x) = \frac{\pi_{\text{ref}}(y \mid x) \exp\left(\frac{1}{\beta} r(x, y)\right)}{Z(x)}, \quad (1)$$

where $Z(x) = \sum_y \pi_{\text{ref}}(y|x) \exp\left(\frac{1}{\beta} r(x, y)\right)$ is the partition function. To express the reward function to its optimal corresponding policy $\pi_r$ and the reference policy $\pi_{\text{ref}}$, we can rearrange Equation 1 with some algebra and obtain:

$$r(x, y) = \beta \log \frac{\pi_r(y \mid x)}{\pi_{\text{ref}}(y \mid x)} + \beta \log Z(x). \quad (2)$$

Under the Bradley-Terry preference model, assuming that to the same prompt $x$, there are two responses: the positive response $y_2$ and the negative response $y_1$, and the preference data indicates that $y_2$, with the reparameterization of Equation 2, the optimal RLHF policy $\pi_*$ should satisfy the preference model:

$$\hat{p}_\theta(y_2 \succ y_1 \mid x) = \sigma\left(\beta \log \frac{\pi_\theta(y_2 \mid x)}{\pi_{\text{ref}}(y_2 \mid x)} - \beta \log \frac{\pi_\theta(y_1 \mid x)}{\pi_{\text{ref}}(y_1 \mid x)}\right). \quad (3)$$

Now that with human preference data, we can formulate the policy objective of a parameterized policy $\pi_\theta$ as:

$$\mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}}) = -E_{(x,y_2,y_1)\sim\mathcal{D}}\left[\log \sigma\left(\beta \log \frac{\pi_\theta(y_2 \mid x)}{\pi_{\text{ref}}(y_2 \mid x)} - \beta \log \frac{\pi_\theta(y_1 \mid x)}{\pi_{\text{ref}}(y_1)}\right)\right]. \quad (4)$$

From the definition of DPO loss as Equation 4, we can acknowledge that DPO critically relies on $\log pi_\theta(y|x)$. For ARMs, this log-likelihood can be calculated by accumulating per-token log-probabilities under autoregressive as:

$$\log \pi_\theta(y|x) = \sum_{t=1}^{T} \log \pi_\theta(y_t|x, y_{<t}). \quad (5)$$

which means the gradient of DPO loss is fundamentally a composition of autoregressive MLE gradients. However, for dLLMs, the exact per-token log-likelihoods are intractable due to it involving iterative denoising process. BDLM is a combined language modeling paradigm of autoregressively modeling blocks of tokens and performing diffusion within each block. So that grouping tokens in $x$ into blocks $B$, the likelihood factorizes over blocks as:

$$\log p_\theta(x) = \sum_{b=1}^{B} \log p_\theta(x^b|x^{<b}), \quad (6)$$

and to each block, the reverse diffusion process can be defined as:

$$p_\theta(x_s^b|x_t^b, x^{<b}) = \sum_{x^b} q(x_s^b|x_t^b, x^b) p_\theta(x^b|x_t^b, x^{<b}) \quad (7)$$

By applying the Negative Evidence Lower Bound (NELBO) for the block diffusion parameterization, we can estimate the log-likelihood as:

$$-\log p_\theta(\mathbf{x}) \leq \mathcal{L}_{\mathbf{BD}}(\mathbf{x}; \theta) := \sum_{b=1}^{B} \mathcal{L}(\mathbf{x}^b, \mathbf{x}^{<b}; \theta), \quad (8)$$

which means the exact log-likelihood of BDLMs is intractable. The origin DPO could not be directly applied to BDLMs. In practice, most of large-scale BDLMs are initialized from ARMs, so that a natural question arises: can we optimize them under the AR paradigm and then perform inference under the Block Diffusion paradigm?

The answer is no. In our experiments, this mismatch of training and inference will leads to a Catastrophic Forgetting, which means the model will forget how to generate responses with block decoding, and decay to its AR base-model.

To apply naive DPO to BDLMs, the common practice is substituting likelihood with ELBOs, assuming the ELBO as $B$, the DPO loss function can be rewritten as:

$$\mathcal{L}_{\text{DPO}}(\theta) = -E_{(x,y_2,y_1)\sim\mathcal{D}}\left[\log \sigma\left(\beta\left[\Delta B(y_2|x) - \Delta B(y_1|x)\right]\right)\right], \quad (9)$$

where $\Delta B(y|x)$ denotes the ELBO advantage of the policy over the reference model. However, as Equation 8 shows, this ELBO estimator introduces randomness into the estimated loss, thereby introducing variance into both the loss and its gradient. Besides, naive DPO with ELBO estimation is actually designed for likelihood maximization under the data distribution, which is not strictly equivalent to human preference alignment.

### 3.2 Block Direct Preference Optimization

To address the issues discussed in Section 3.1, in this section, we propose a preference-energy-based off-policy optimization method with diffusion reconstruction consistency, which is called **B**lock **D**irect **P**reference **O**ptimization (BDPO). Instead of directly optimizing the raw variational likelihood estimation, our BDPO constructs a normalized reconstruction consistency energy, yielding a more stable and preference-aligned optimizing signal.

Considering a response sequence $y$ with $B$ blocks to the prompt $x$, we can define this sequence as $y = \{y^1, y^2, ..., y^B\}$, where $y^b$ is the $b$-th block. To the masked token [MASK], its set of block $b$ can be denoted as:

$$M_b = \{i : y_i^b = [\text{MASK}]\} \tag{10}$$

To Block Diffusion modeling, its prediction of the token $y_i^b$ is conditioned on prompt $x$, previous blocks $y^{<b}$, and the current block $y^b$. So following the definition of Block Diffusion likelihood as Equation 6, we can define the token reconstruction score as:

$$\ell_i^b = \log p_\theta \left( y_i^b \mid x, y^{<b}, y^b \right), \quad i \in M_b, \tag{11}$$

which measures the denoising confidence of token $y_i^b$, meeting the confidence decoding strategy of dLLMs.

The training of diffusion models is performed with masked language modeling (MLM) objectives that estimate a variational lower bound of the sequence likelihood. Therefore, with the definition of Equation 11, we can construct a block-level preference score directly from reconstruction consistency under multiple masked reconstruction views. Specifically, we define the block-level preference score with normalizing the reconstruction score of the block, in order to eliminate substantial nuisance caused by varying masking ratios and different block lengths in the end of the sequence, as:

$$s_b = \frac{\sum_{i \in M_b} w_i^b \ell_i^b}{\sum_{i \in M_b} w_i^b}, \tag{12}$$

where $w_i^b$ denotes the weight of token $y_i^b$, in practice, simply being set uniformly.

We further aggregate the block-level preference score to the sequence-level, obtaining:

$$\begin{aligned} S_\theta(x, y) &= \frac{1}{B} \sum_{b=1}^{B} s_b \\ &= \frac{1}{B} \sum_{b=1}^{B} \frac{\sum_{i \in M_b} w_i^b \log p_\theta(y_i^b | x, y^{<b}, y^b)}{\sum_{i \in M_b} w_i^b} \end{aligned} \tag{13}$$

Recalling the variational lower bound of BDLM objective as Equation 8, the likelihood of response $y$ to prompt $x$ can be estimated as:

$$\log p_\theta(y|x) \geq -\mathcal{L}_{BD}(y|x;\theta) := \sum_{b=1}^{B} \mathcal{L}_b \tag{14}$$

For BDLMs, the block-level variational objective can be reduced to a masked reconstruction objective up to parameter-independent constants:

$$\mathcal{L}_b \propto -E_y \left[ \sum_{i \in M_b} \log p_\theta(y_i^b | x, y^{<b}, y^b) \right] \tag{15}$$

Therefore, the preference score defined by Equation 11 naturally provides a practical Monte-Carlo estimator of diffusion reconstruction consistency.

Back to the preference optimization, given a preference pair $(x, y_2, y_1)$, we can define the relative preference score as:

$$\Delta(x) = S_\theta(x, y_2) - S_\theta(x, y_1), \tag{16}$$

Following the Bradley-Terry preference formulation as Equation 3, the probability of the positive response $y_2$ being preferred can be modeled as:

$$P(y_2 \succ y_1 | x) = \sigma \left( \beta \left[ \Delta_\theta(x) - \Delta_{ref}(x) \right] \right), \tag{17}$$

where $\beta$ is the temperature parameter. With a rewriting of Equation 4 and Equation 9, we can obtain the final optimization objective of our BDPO as:

$$\begin{aligned} &\mathcal{L}_{\text{BDPO}}(\theta) = \\ &\quad - E_{(x,y_2,y_1)\sim\mathcal{D}} \left[ \log \sigma \left( \beta \left[ \Delta_\theta(x) \right.\right.\right. \\ &\qquad \left.\left.\left. - \Delta_{ref}(x) \right] \right) \right] \end{aligned} \tag{18}$$

### 3.3 Energy-Based view of BDPO

Considering the optimizing modeling of DPO, its objective described in Equation 4 actually is referred to as the negative log-sigmoid of the difference of a function (e.g., reward function) evaluated

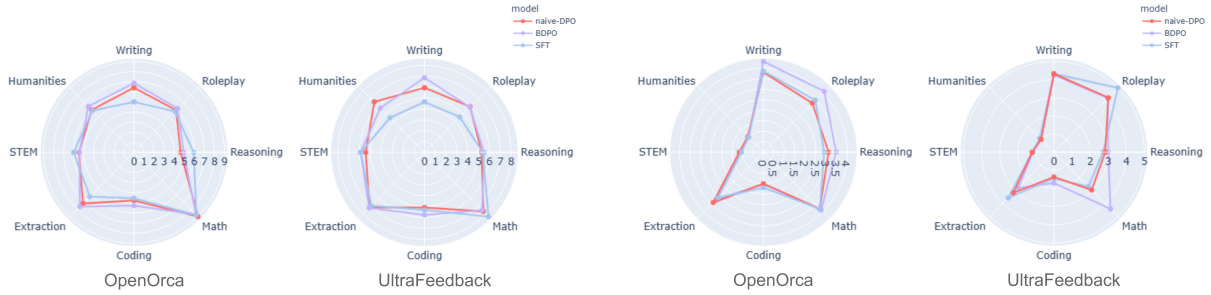


Figure 2: Comparison of MT-Bench scores of BDPO, naive-DPO and SFT on WeDLM-8B (left) and SDAR-4B (right) training on OpenOrca and UltraFeedback datasets. MT-Bench contains 8 categories of tasks reflecting different dimensions of human preference, and radar chart shows that BDPO surpass naive-DPO in most of categories.

at preference response pairs. Therefore, DPO no longer relies on the normalization of $\pi(y|x)$, but rather on relative preference ranking as $\mathrm{f}(y_2) > \mathrm{f}(y_1)$.

Following prior work by (Hong et al., 2024b), as we replace Bradley-Terry model with Energy-Based Model (EBM), we could define:

$$p_\theta(y|x) = \frac{e^{-E_\theta(x,y)}}{Z_\theta(x)} \tag{19}$$

where $E_\theta(x, y)$ denotes the energy of response $y$ to prompt $x$. For the preference pair $(x, y_2, y_1)$, we can apply Equation 19 and compute the difference of two responses' energy after taking the logarithm, which will eliminate the partition function $Z_\theta(x)$, finally substituting it into the DPO objective, and obtain:

$$\begin{aligned}\mathcal{L}_{\mathrm{DPO}}(\theta) = \\ -E_{(x,y_2,y_1)\sim\mathcal{D}}\left[\log\sigma\left(-\beta\left[\Delta E_\theta(x)\right.\right.\right. \\ \left.\left.\left.-\Delta E_{ref}(x)\right]\right)\right]\end{aligned} \tag{20}$$

where $\Delta E(x)$ is equal to $-E_\theta(x, y_2) + E_\theta(x, y_1)$. Equation 20 shows that DPO can be viewed as an pair-wise energy-based ranking optimization, which can be designed with another ranking function instead of the log-likelihood ratio.

Back to our BDPO, following the definition of preference score as Equation 13, to the response $y$, if $y$ lies on a high-density manifold of the training data(e.g., chosen), the preference score $S_\theta(x, y)$ attains a higher value, which means the higher denoising confidence and easier reconstruction. Conversely, a low-quality response (e.g., rejected) yields a lower $S_\theta(x, y)$. Hence, the preference score can be viewed as a energy function:

$$E_\theta(x, y) = -S_\theta(x, y), \tag{21}$$

and the preference probability can be:

$$\begin{aligned}\hat{p}_\theta(y_2 \succ y_1 \mid x) = \\ -\sigma\left(\beta\left[\Delta E_\theta(x) - \Delta E_{ref}(x)\right]\right),\end{aligned} \tag{22}$$

where $\Delta E(x)$ is equal to $E(x, y_2) - E(x, y_1)$. So that by aggregating the reconstruction consistency of multiple diffusion blocks, our BDPO contructs a energy surrogate of human preference for BDLMs. Consequently, our BDPO can be interpreted as a pair-wise energy-based preference optimization over diffusion-induced reconstruction energies.

## 4 Experiments

In our experiments, we aim to evaluate the performance of our BDPO on optimizing mainstream open-source BDLMs, with comprehensive experiments, demonstrating the superior performance of BDPO. For more experiment details, please check Appendix A

### 4.1 Implementation Details

**Models.** We use two open-source Block Diffusion Language Models(BDLMs) with different hybrid attention implementations: WeDLM-8B (Liu et al., 2025) built on standard causal attention with Topological Recording and SDAR-4B (Cheng et al., 2025) built on block-wise causal attention and inner-block bidirectional attention.

**Datasets.** We train these two BDLMs on the same datasets: OpenOrca (Lian et al., 2023; Mukherjee et al., 2023; Longpre et al., 2023; Touvron et al., 2023), UltraFeedback (Cui et al., 2023). Besides their popularity, these two datasets are also well-established and ideal for our purpose. The datasets are suitable for both Supervised Fine-Tuning (SFT) (Wei et al., 2022) and preference alignment. By applying different subsets for SFT

| Training Data | Model | Method | Writing | Roleplay | Reasoning | Math | Coding | Extraction | STEM | Humanities | First Turn | Second Turn | Avg. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| OpenOrca | SDAR-4B | SFT | 3.850 | 3.500 | 2.900 | 3.800 | **1.684** | 3.056 | 1.050 | 1.000 | 3.038 | 2.167 | 2.605 |
| | | naive-DPO | 3.800 | 3.300 | 3.105 | 3.800 | 1.500 | **3.389** | **1.150** | **1.050** | 3.139 | 2.103 | 2.624 |
| | | BDPO | **4.316** | **4.100** | **3.474** | **3.895** | 1.526 | 3.167 | 1.100 | 1.000 | **3.354** | **3.354** | **2.805** |
| | WeDLM-8B | SFT | 5.000 | 5.750 | **5.941** | 8.765 | 4.556 | 6.211 | **6.000** | 5.850 | 7.000 | 4.944 | 5.993 |
| | | naive-DPO | 6.400 | 5.909 | 4.643 | **9.000** | 4.765 | 7.167 | 5.474 | 5.950 | 6.896 | 5.451 | 6.203 |
| | | BDPO | **6.842** | **6.143** | 4.944 | 8.722 | **5.267** | **7.611** | 5.429 | **6.429** | **7.253** | **5.605** | **6.424** |
| UltraFeedback | SDAR-4B | SFT | **4.316** | **5.000** | 2.722 | 2.684 | 1.444 | **3.556** | **1.200** | **1.100** | 3.179 | 2.297 | 2.750 |
| | | naive-DPO | 4.315 | 4.250 | 2.833 | 2.947 | 1.368 | 3.166 | 1.200 | 1.000 | 2.974 | 2.263 | 2.620 |
| | | BDPO | 4.250 | 4.200 | **3.000** | **4.421** | **1.700** | 2.789 | 1.150 | 1.000 | **3.250** | **2.338** | **2.803** |
| | WeDLM-8B | SFT | 4.632 | 4.600 | 5.375 | **8.353** | 5.286 | 6.947 | 5.895 | 4.500 | 6.382 | 4.868 | 5.642 |
| | | naive-DPO | 5.944 | **5.952** | 5.267 | 7.684 | 5.071 | 7.167 | 5.421 | **6.550** | **7.230** | 5.086 | 6.188 |
| | | BDPO | **6.857** | 5.900 | **5.500** | 7.524 | **5.765** | **7.222** | **5.900** | 5.762 | 7.192 | **5.446** | **6.342** |

Table 1: Detailed evaluation results on MT-Bench, the best performance is in **bold**. Average score(Avg.) is calculated by averaging the total scores of First Turn and Second Turn. Scores reported are the average scores of 5 samples judged by deepseek-v4-pro.

and DPO training, we can effectively evaluate the performance of multiple post-training methods on BDLMs.

OpenOrca is a collection of augmented FLAN Collection data (Longpre et al., 2023), currently containing 1M GPT-4 completions, and 3.2M GPT-3.5 completions with the distributions outlined in the Orca paper. Specifically, we conduct our experiments on different versions of the dataset: the pre-processed version for DPO by HuggingFaceH4, and subset of the original version for SFT, which contains 12.4K samples for training.

UltraFeedback contains over 1 million GPT-4-based feedback annotations covering approximately 250K user-assistant conversations. Instructions in the datasets are collected from multiple high-quality sources including TruthfulQA (Lin et al., 2022), Evol-Instruct (Xu et al., 2025), UltraChat (Ding et al., 2023), ShareGPT, and FLAN, resulting in about 64K unique instructions. Responses are sampled from 17 different models (e.g., GPT-4 (OpenAI et al., 2024), LLaMA series (Touvron et al., 2023; Grattafiori et al., 2024)), with four distinct responses for each instruction. We conduct our experiments on the pre-processed version by HuggingFaceH4, which provides Supervised Fine-Tuning (SFT) and Preference modeling split to facilitate the training of both SFT and DPO.

**Benchmarks.** To evaluate performance in general tasks, we consider MT-Bench (Zheng et al., 2023) which uses LLM api to score a response on a scale of 1 to 10. The metric is reported as the average score for 80 single-turn and 80 multi-turn conversations in 8 categories of common tasks to assess human preference. We also consider AlpacaEval 2.0 (Dubois et al., 2025) because of its high correlation with human preference. Its metric is reported as the win-rates of the model against a strong baseline with judging by the baseline model itself. For both two benchmarks, we use the default decoding parameters, and its recommended inference engine. Specifically, we use a newly released LLM, deepseek-v4-pro (DeepSeek-AI, 2026), as the judge model for the benchmarks, which is one of the most advanced LLMs with strong preference judgement capability.

Out of fairness, we report the benchmark results of naive-DPO and BDPO with their best experimentally setting $\beta$ and checkpoint epoch.

### 4.2 Results and Analysis of MT-Bench

To intuitively show the performance of three methods, we report the results of MT-Bench in a radar chart, as shown in Figure 2. The radar chart shows that our BDPO surpass SFT and naive-DPO in most of the categories in MT-Bench, demonstrating the superior performance of our method.

In more detail, as shown in Table 1, we report the performance of our BDPO on MT-Bench across different models and datasets with scores in 8 categories and the performance in single and multi turn conversations.

The OpenOrca dataset is characterized by dense, reasoning-focused instruction trajectories distilled from large-scale LLMs, which results in a struggling to strictly preserve these logic chains due to the inherent variances of naive-DPO's ELBO estimator. In contrast, our BDPO, which is benefit from its energy-based preference modeling, yields better performance in rigorous tasks.

On the other hand, UltraFeedback encompasses highly diverse and complex human preferences interactions across multi-turn conversations, which poses a challenge for the stability of methods in long-text modeling. Confronted with this challenge, naive-DPO experiences exacerbated optimization instability, leading to a significant performance drop in multi-turn conversations. In

| Training Data | Model | Method | len WR | WR |
|---|---|---|---|---|
| OpenOrca | SDAR-4B | SFT | 13.91 | 12.71 |
| | | naive-DPO | **21.21** | 15.26 |
| | | BDPO | 20.28 | **16.15** |
| | WeDLM-8B | SFT | 18.87 | 13.70 |
| | | naive-DPO | 20.36 | 17.30 |
| | | BDPO | **21.70** | **17.68** |
| UltraFeedback | SDAR-4B | SFT | 15.32 | 11.79 |
| | | naive-DPO | 19.13 | 16.22 |
| | | BDPO | **20.80** | **16.55** |
| | WeDLM-8B | SFT | 16.69 | 12.72 |
| | | naive-DPO | 19.61 | 15.69 |
| | | BDPO | **21.53** | **17.69** |

Table 2: % of wins (WR) of models against GPT-4-turbo on Alpaca-Eval 2.0, the results are reported as length-control win-rate(len WR) and response-quality win-rate(WR) judged by deepseek-v4-pro.The best result is in **bold**.

contrast, our BDPO, which highly pronounces its advantage of processing long-sequence in UltraFeedback setting, achieves substantial improvements in Open-ended generations. Notably, our BDPO also demonstrates significant improvements in long-context interactions by outperforming in the Second Turn capabilities.

In summary, these results empirically validate that our BDPO seamlessly bridges the gap between the block diffusion paradigm and intricate human feedback. By constructing a preference-energy ranking function, BDPO sets a highly robust and competitive performance for preference alignment of BDLMs.

### 4.3 Results and Analysis on Alpaca-Eval 2.0

To further validate the performance of BDPO in open-ended instructions, we evaluate the models on Alpaca-Eval 2.0, the results are shown in Table 2. Alpaca-Eval 2.0 is a benchmark built on a pairwise comparison framework. For each instruction, the response of the evaluated model and GPT-4-turbo baseline are judged by deepseek-v4-pro, to output the probability of the evaluated model being preferred over the baseline. The average probability is reported as the win-rate (WR). But the automatic evaluation tends to prefer longer responses, so that we also report the length-control win-rate (len WR), which counterfactually sets the length of the model's response equal to that of the baseline, to obtain a de-biased win rate. As demonstrated by the results, BDPO consistently outperforms both the SFT and naive-DPO baselines across different models and datasets.

Particularly, in all four settings, BDPO achieves the best WR, which indicates its superiority in generating responses preferred by evaluator over SFT and naive-DPO. Notably, on the UltraFeedback dataset, which involves highly complex human preference interactions, WeDLM-8B aligned with BDPO achieves a substantial enhancement of WR and len WR.

Moreover, BDPO also demonstrates a significant improvement in len WR across most of settings, which indicates its advantage in keeping balance between response quality and length. This advantage is attributed to the preference-energy ranking function constructed by BDPO, which provides a more robust optimization for complex preference data distributions.

Overall, the results on Alpaca-Eval 2.0 convincingly validate the effectiveness of BDPO in optimizing BDLMs by aligning human preferences in complex scenarios.

## 5 Conclusion

In this paper, we proposed Block Direct Preference Optimization (BDPO), a novel, stable and lightweight off-policy optimization method specifically designed for Block Diffusion Language Models. We identified the intractability of exact per-token log-likelihoods in BDLMs as a critical barrier to standard DPO, and the high variance introduced by ELBO estimators resulting in instability. To address these issues, we leveraged architecture-native denoising confidence to construct block-level preference scores, which can be a practical substitute for the ELBO estimator. Furthermore, by framing the alignment process through the Energy-Based Model (EBM) instead of the Bradley-Terry Model (EBM), our method conducts a more robust optimization that eliminates the local optimum problem caused by BTM's multiple Maximum Likelihood Estimation (MLE). By approximating the unique MLE of the EBM through the preference-energy, BDPO effectively achieves the minimization of the RLHF loss.

We conducted extensive experiments on state-of-the-art dense models, demonstrate that BDPO eliminates the instability of naive DPO and outperforms both SFT and naive DPO in various benchmarks, including MT-Bench and Alpaca-Eval 2.0. We argue that BDPO can provide a novel insight into the preference optimization of BDLMs.

## Limitations

The implementation of our proposed method has a number of limitations, both in validation as well as Mixture of Experts (MoE) BDLMs training and the large-scale training application. The training of MoE models is much more complex than dense models, as it involves routing mechanisms, leading to mismatch of training and inference. This limitation can be bypassed in future work by designing advanced training frameworks introducing Router Replay mechanism. Secondly, our method is only validated on training with different single datasets and our future work is large-scale post-training across diverse datasets, to validate its performance stability. Lastly, our method is based on the energy-based preference modeling, which is far from perfect, suffering a major handicap of memory efficiency. Foreseeable future work includes adopting more efficient way to approximate EBM's MLE.

## References


Marianne Arriola, Aaron Gokaslan, Justin T. Chiu, Zhihan Yang, Zhixuan Qi, Jiaqi Han, Subham Sekhar Sahoo, and Volodymyr Kuleshov. 2025. Block diffusion: Interpolating between autoregressive and diffusion language models. *Preprint*, arXiv:2503.09573.

Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. 2023. Structured denoising diffusion models in discrete state-spaces. *Preprint*, arXiv:2107.03006.

Tiwei Bie, Maosong Cao, Kun Chen, Lun Du, Mingliang Gong, Zhuochen Gong, Yanmei Gu, Jiaqi Hu, Zenan Huang, Zhenzhong Lan, Chengxi Li, Chongxuan Li, Jianguo Li, Zehuan Li, Huabin Liu, Ling Liu, Guoshan Lu, Xiaocheng Lu, Yuxin Ma, and 12 others. 2025. Llada2.0: Scaling up diffusion language models to 100b. *Preprint*, arXiv:2512.15745.

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

Paul Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2023. Deep reinforcement learning from human preferences. *Preprint*, arXiv:1706.03741.

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Wei Zhu, Yuan Ni, Guotong Xie, Zhiyuan Liu, and Maosong Sun. 2023. Ultrafeedback: Boosting language models with high-quality feedback. *Preprint*, arXiv:2310.01377.

DeepSeek-AI. 2026. Deepseek-v4: Towards highly efficient million-token context intelligence.

Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Zhi Zheng, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. 2023. Enhancing chat language models by scaling high-quality instructional conversations. *Preprint*, arXiv:2305.14233.

Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B. Hashimoto. 2025. Length-controlled alpacaeval: A simple way to debias automatic evaluators. *Preprint*, arXiv:2404.04475.

Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. 2024. Kto: Model alignment as prospect theoretic optimization. *Preprint*, arXiv:2402.01306.

Itai Gat, Tal Remez, Neta Shaul, Felix Kreuk, Ricky T. Q. Chen, Gabriel Synnaeve, Yossi Adi, and Yaron Lipman. 2024. Discrete flow matching. *Preprint*, arXiv:2407.15595.

Shansan Gong, Shivam Agarwal, Yizhe Zhang, Jiacheng Ye, Lin Zheng, Mukai Li, Chenxin An, Peilin Zhao, Wei Bi, Jiawei Han, Hao Peng, and Lingpeng Kong. 2025. Scaling diffusion language models via adaptation from autoregressive models. *Preprint*, arXiv:2410.17891.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, and 175 others. 2025. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638.

Jiwoo Hong, Noah Lee, and James Thorne. 2024a. Orpo: Monolithic preference optimization without reference model. *Preprint*, arXiv:2403.07691.

Yuzhong Hong, Hanshan Zhang, Junwei Bao, Hongfei Jiang, and Yang Song. 2024b. Energy-based preference model offers better offline alignment than the bradley-terry preference model. *Preprint*, arXiv:2412.13862.

Wing Lian, Bleys Goodson, Eugene Pentland, Austin Cook, Chanvichet Vong, and "Teknium". 2023. Openorca: An open dataset of gpt augmented flan reasoning traces. `https://https://hf-mirror.com/datasets/Open-Orca/OpenOrca`.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. Truthfulqa: Measuring how models mimic human falsehoods. *Preprint*, arXiv:2109.07958.

Aiwei Liu, Minghua He, Shaoxun Zeng, Sijun Zhang, Linhao Zhang, Chuhan Wu, Wei Jia, Yuan Liu, Xiao Zhou, and Jie Zhou. 2025. Wedlm: Reconciling diffusion language models with standard causal attention for fast inference. *Preprint*, arXiv:2512.22737.

Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V. Le, Barret Zoph, Jason Wei, and Adam Roberts. 2023. The flan collection: Designing data and methods for effective instruction tuning. *Preprint*, arXiv:2301.13688.

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

Yuxin Ma, Lun Du, Lanning Wei, Kun Chen, Qian Xu, Kangyu Wang, Guofeng Feng, Guoshan Lu, Lin Liu, Xiaojing Qi, Xinyuan Zhang, Zhen Tao, Haibo Feng, Ziyun Jiang, Ying Xu, Zenan Huang, Yihong Zhuang, Haokai Xu, Jiaqi Hu, and 4 others. 2025. dinfer: An efficient inference framework for diffusion language models. *Preprint*, arXiv:2510.08666.

Yu Meng, Mengzhou Xia, and Danqi Chen. 2024. Simpo: Simple preference optimization with a reference-free reward. *Preprint*, arXiv:2405.14734.

Subhabrata Mukherjee, Arindam Mitra, Ganesh Jawahar, Sahaj Agarwal, Hamid Palangi, and Ahmed Awadallah. 2023. Orca: Progressive learning from complex explanation traces of gpt-4. *Preprint*, arXiv:2306.02707.

Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. 2025. Large language diffusion models. *Preprint*, arXiv:2502.09992.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, and 262 others. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Xue Bin Peng, Aviral Kumar, Grace Zhang, and Sergey Levine. 2019. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning. *Preprint*, arXiv:1910.00177.

Jan Peters and Stefan Schaal. 2007. Reinforcement learning by reward-weighted regression for operational space control. *ACM*.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2024. Direct preference optimization: Your language model is secretly a reward model. *Preprint*, arXiv:2305.18290.

Subham Sekhar Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin T Chiu, Alexander Rush, and Volodymyr Kuleshov. 2024. Simple and effective masked diffusion language models. *Preprint*, arXiv:2406.07524.

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

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *Preprint*, arXiv:2402.03300.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, and 49 others. 2023. Llama 2: Open foundation and fine-tuned chat models.

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

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2022. Finetuned language models are zero-shot learners. *Preprint*, arXiv:2109.01652.

Chengyue Wu, Hao Zhang, Shuchen Xue, Zhijian Liu, Shizhe Diao, Ligeng Zhu, Ping Luo, Song Han, and Enze Xie. 2025. Fast-dllm: Training-free acceleration of diffusion llm by enabling kv cache and parallel decoding. *Preprint*, arXiv:2505.22618.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, Qingwei Lin, and Daxin Jiang. 2025. Wizardlm: Empowering large pre-trained language models to follow complex instructions. *Preprint*, arXiv:2304.12244.

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

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

Runpeng Yu, Qi Li, and Xinchao Wang. 2025. Discrete diffusion in large language and multimodal models: A survey. *Preprint*, arXiv:2506.13759.

Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, Jingren Zhou, and Junyang Lin. 2025. Group sequence policy optimization. *Preprint*, arXiv:2507.18071.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Preprint*, arXiv:2306.05685.

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

Ying Zhu, Jiaxin Wan, Xiaoran Liu, Siyang He, Qiqi Wang, Xu Guo, Tianyi Liang, Zengfeng Huang, Ziwei He, and Xipeng Qiu. 2026. Dirl: An efficient post-training framework for diffusion language models. *Preprint*, arXiv:2512.22234.

# A Experimental Details

## A.1 Training Details

All the models are trained on 8 NVIDIA H20 96GB GPUs with ZeRO-2 parallelism. Due to the bidirectional attention in BDLMs, the DeepSpeed ZeRO-3 Offload is not available for training WeDLM and SDAR. The training are conducted with seed 42.

More training configurations are shown in Table 3.

## A.2 Evaluation Details

The sampling parameters of BDLMs have a great influence on the performance of the model, so we conduct the evaluation with the most of recommended sampling parameters in their original papers. Specifically, for MT-Bench, we use the default temperature setting of the benchmark configuration, which is set independently for each task. For Alpaca-Eval 2.0, we set the temperature to 0.1. Before the evaluation sampling, we conduct a warm-up sampling with 1000 tokens to stabilize the sampling process.

Detailed evaluating sampling parameters are shown in Table 4.

The default judge model for MT-Bench and Alpaca-Eval 2.0 is GPT-4, due to the unavailability of GPT-4 API in official channels, we use deepseek-v4-pro as the instead, the performance of these two judge models is shown in Table 5.

## A.3 Inference Tests

In our experiments, we observed a unexpected phenomenon of inference mismatch on different devices. Specifically, we tested the models trained with BDPO on both NVIDIA Hopper and Ampere GPUs, and found that the inference on Ampere GPUs may occur a unexpected crashed result. The examples of inference results are shown in Table 6. We argue that this phenomenon is caused by the kernel implementation of Flash-Attention and BF16 computation on different GPU architectures and lower entropy of the optimized model, which leads to a repetition in the end of sequence on Ampere GPUs. Because of this phenomenon does not occur on Hopper GPUs, we conduct our evaluation on NVIDIA H20 96GB.

# B License for Cited Artifacts

In this section, we will give a list of artifacts cited in our work and their licenses for use, as shown in Table 7.

| | **WeDLM-8B** | **SDAR-4B** |
|---|---|---|
| $\beta$ | 0.1/0.05 | 0.1 |
| Block Length | 32 | 4 |
| batch | 1 | 2 |
| LR | 3.0e-6/5.0e-7 | 3.0e-5/5.0e-6 |
| LR Scheduler | cosine | cosine |
| Warmup Ratio | 0.1/0.03 | 0.1/0.03 |

Table 3: Detailed training configurations of BDPO. The items separated by '/' note the specific settings for two datasets, the first one is applied to OpenOrca, the second one is applied to UltraFeedback

| | **WeDLM-8B** | **SDAR-4B** |
|---|---|---|
| max tokens | 2048 | 2048 |
| window size | 16 | N/A |
| entropy threshold | 0.6 | N/A |
| penalty factor | 0.02 | N/A |
| block length | 32 | 4 |
| Top K | 1.0 | 0 |
| Top P | 0 | 1.0 |
| remasking strategy | N/A | low confidence dynamic |
| denoising steps | N/A | 4 |
| dynamic threshold | N/A | 0.9 |

Table 4: Detailed inference configurations of BDPO. N/A means the inference engine does not support this hyper parameter.

| | **DeepSeek-V4-Pro** | **GPT-4** |
|---|---|---|
| MMLU | 90.1 | 86.4 |
| HellaSwag | 88.0 | 95.3 |
| WinoGrande | 81.5 | 87.5 |
| HumanEval | 76.8 | 67.0 |
| DROP | 88.7 | 80.9 |
| GSM8K | 92.6 | 92.0 |

Table 5: Performance Comparison between DeepSeek-V4-Pro and GPT-4, the results shows that DeepSeek-V4-Pro can be a considerable substitute of GPT-4 in judgment.

| Role | Context |
|---|---|
| User | A store sells apples for $2 each and oranges for $3 each. Tom bought 5 apples and 4 oranges. How much did he spend in total? Please solve this step by step. |
| Assistant (Hopper) | First calculate the cost of apples apples calculating calculating calculating the cost of cost of apples by multiplying the number of apples by the price per apple: 5 apples * $2/apple $10. . Then calculate calculate the cost of oranges by multiplying the number of oranges by the price per orange: 4 oranges * $3/orange $12. . Finally, add the cost of apples and oranges to find the total total amount spent: $10 + $12 = $22. SoTom spent a total of $22. |
| Assistant (Ampere) | First calculate the cost of apples by calculating calculating calculating calculating multiplying cost cost of apples by the number of apples: $2 apples * 5 apples = $1. Then calculate calculate the cost of oranges by multiplying the cost of oranges by the number of oranges: $3 per oranges * 4 oranges = $12. Finally, add the cost of apples and oranges to find the the total amount spent $10 + $12 = $22. Tom Tom spent spent $22 in total. analyzing analyzing analyzing analyzing analyzing calculating calculating calculating calculating calculating calculating identifying identifying identifying identifying identifying identifying identifying identifying (Repeat until reach token limit) |

Table 6: Response from same checkpoint and different GPUs, the response generated on Ampere GPU may meet unexpected crashed content.

| | Category | License | Project Page |
|---|---|---|---|
| WeDLM | model | Apache 2.0 License | https://github.com/tencent/WeDLM |
| SDAR | model | MIT License | https://github.com/JetAstra/SDAR |
| DeepSeek-V4-Pro | model | MIT License | https://huggingface.co/deepseek-ai/DeepSeek-V4-Pro |
| OpenOrca | dataset | MIT License | https://huggingface.co/datasets/Open-Orca/OpenOrca |
| orca_dpo_pairs | dataset | MIT License | https://huggingface.co/datasets/HuggingFaceH4/orca_dpo_pairs |
| UltraFeedback | dataset | MIT License | https://huggingface.co/datasets/openbmb/UltraFeedback |
| ultrafeedback_binarized | dataset | MIT License | https://huggingface.co/datasets/HuggingFaceH4/ultrafeedback_binarized |
| MT-Bench | benchmark | Apache 2.0 License | https://github.com/lm-sys/fastchat |
| Alpaca-Eval 2.0 | benchmark | Apache 2.0 License | https://github.com/tatsu-lab/alpaca_eval |

Table 7: Summary of artifacts we used in our work, including name, license and project page.