wd1: Weighted Policy Optimization for Reasoning in Diffusion Language Models

Xiaohang Tang[1,2,*], **Rares Dolga**[1,3,*], **Sangwoong Yoon**[1] **and Ilija Bogunovic**[1]
[1]University College London (UCL) AI Centre, [2]UCL Department of Statistical Science, [3]UiPath

Improving the reasoning capabilities of diffusion-based large language models (dLLMs) through reinforcement learning (RL) remains an open problem. The intractability of dLLMs likelihood function necessitates approximating the current, old, and reference policy likelihoods at each policy optimization step. This reliance introduces additional computational overhead and lead to potentially large bias – particularly when approximation errors occur in the denominator of policy ratios used for importance sampling. To mitigate these issues, we introduce `wd1`, a novel policy optimization approach that reformulates the objective as a weighted likelihood, requiring only a single approximation for the current parametrized policy likelihood. Experiments on widely used reasoning benchmarks demonstrate that `wd1`, without supervised fine-tuning (SFT) or any supervised data, outperforms existing RL methods for dLLMs, achieving up to 16% higher accuracy. `wd1` delivers additional computational gains, including reduced training time and fewer function evaluations (NFEs) per gradient step. These findings, combined with the simplicity of method's implementation and R1-Zero-like training (no SFT), position `wd1` as a more effective and efficient method for applying RL to dLLMs reasoning.
Code: https://github.com/xiaohangt/wd1

1. Introduction

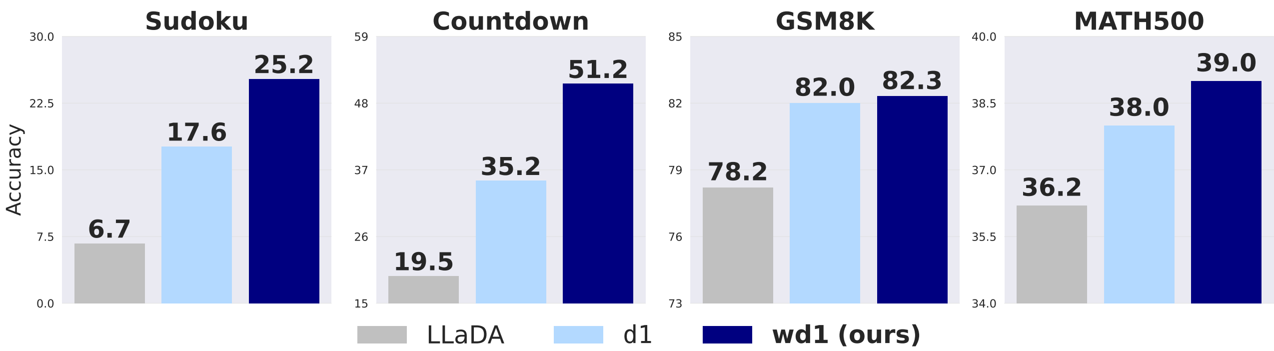

Figure 1 Performance on popular reasoning and planning benchmarks with the same base model LLaDA. For all models, we evaluate with maximum length 256 and 512, and report the best results. Our method **`wd1`**, outperforms both our reproduction of `d1` and accuracies of LLaDA (Zhao et al., 2025).

Diffusion-based large language models (dLLMs) have recently gained attention as promising alternatives to autoregressive (AR) models for language modelling tasks (Nie et al., 2025b; Ou et al., 2025; Yang et al., 2025). Unlike AR models, which generate tokens sequentially, dLLMs iteratively refine entire response sequences through a denoising process. A primary advantage of this approach is the significantly improved inference efficiency. Notably, recent closed models such as Mercury (Labs et al., 2025) and Gemini Diffusion achieve over an order of magnitude speed-up in generation compared to AR models,

*Equal Contribution. Corresponding author(s): i.bogunovic@ucl.ac.uk while maintaining comparable generation quality. Furthermore, open-weight dLLMs demonstrate competitive performance on standard language benchmarks, with smaller models (Lou et al., 2024; Ou et al., 2025; Nie et al., 2024) achieving parity with equivalently sized AR baselines, and larger-scale models such as LLaDA-8B (Zhu et al., 2025a) and Dream-7B (Ye et al., 2025) extending this trend at scale. While dLLMs demonstrate strong performance in text generation, it remains an open and important question how best to fine-tune dLLMs using RL – a paradigm that has proven highly effective in alignment and improving reasoning capabilities of AR models (Ouyang et al., 2022; Shao et al., 2024).

A key challenge in applying reinforcement learning (RL) to dLLMs is the intractability of their likelihood functions (Zhao et al., 2025; Yang et al., 2025), which necessitates approximating the sequence-level likelihoods for policy optimization. Methods such as GRPO (Shao et al., 2024) require estimating the likelihoods of the current, old, and reference policies at every update step, leading to significant computational overhead – especially when large sample sizes are needed for low-error approximations. Moreover, small errors in likelihood estimation can still induce large bias in the objective due to the use of importance sampling, where the importance weights are policy ratios. The inaccuracies in the denominator of policy ratios can severely distort the value of ratio. These issues can be further exacerbated as the completion length and the number of diffusion steps increase.

To address these challenges, we propose **wd1**, a policy optimization approach with **w**eighted log-likelihood objective for **d**LLMs. Crucially, this objective dispenses with explicit policy ratios and relies on a single likelihood estimate, thereby avoiding the potentially large bias from policy ratio and reducing the computational overhead. The principal contributions of this work are:

- We first propose to approximate the closed-form solution of single-iterate *reverse*-KL-constrained policy optimization in each iteration by optimizing a weighted log-likelihood objective. Crucially, this objective removes the need for policy ratios and depends only on the current policy's likelihood. We further prove that the resulting iterative updates guarantee monotonic policy improvement.
- Based on this insight, we introduce **wd1**, a simple weighted policy optimization method for diffusion language models, where the weights are dependent on the group-relative advantage $\hat{A}_i$:

$$\mathcal{L}_{\texttt{wd1}}(\theta) = \mathbb{E}_{q\sim\mathcal{D},\{o_i\}_{i=1}^{G}\sim\pi_{\text{old}}^{\text{ref}}(\cdot|q)}\Big[\sum_{i=1}^{G}\big(-\text{Softmax}(\hat{A}_i)+\text{Softmax}(-\hat{A}_i)\big)\cdot\log\pi_\theta(o_i|q)\Big].$$

  Our method aims to increase the probability of completions with higher $\hat{A}_i$ values and decrease it for lower ones, reinforcing beneficial outcomes and discouraging detrimental ones.
- In our experiments, we demonstrate that wd1, **even without supervised fine-tuning (SFT)**, outperforms the existing diffusion-based RL method d1 across a suite of reasoning benchmarks, **achieving up to a 16% improvement in test accuracy** (see Figure 1). wd1 delivers additional speed-ups in RL training, underscoring its effectiveness and establishing it as a substantially more efficient approach for reasoning dLLMs.

2. Preliminaries

We denote the generation distribution of diffusion-based Large Language Models (dLLMs) by π_θ. Denote prompt $q \in \mathcal{D}$, and completions o. Notably, the reward function denoted by $R(q, o)$ in this report is not limited to verifier. We use superscript k to indicate the k-th token in the sequence of completion: o^k.

2.1. Diffusion Large Language Models

The most promising discrete diffusion for language modeling is masked diffusion (MD), which gradually corrupt text sequence with `mask` token (Sahoo et al., 2024; Ou et al., 2025; Shi et al., 2025; Lou et al., 2024). Let t denote the diffusion timestep, and x_t the intermediate noised sequence at step t. The fully denoised sequence (i.e., the model output) is represented by x_0, and the forward process is a continuous-time Markov chain with transition denoted by $p_{t|0}(x_t \mid x_0)$.

This work aims to apply RL to LLaDA (Ou et al., 2025; Zhu et al., 2025a), which directly models the clean data distribution conditional on masked sequence $\pi_\theta(x_0^k \mid x_t)$. By assuming absorbing transition (Austin et al., 2023; Campbell et al., 2022), the Evidence Lower Bound (ELBO) is reduced to a simple objective that has been used for pretraining (Nie et al., 2025b; Ou et al., 2025):

$$\max_\theta \mathbb{E}_{t\sim\mathcal{U}[0,1],\ x_0\sim p_{\text{data}},\ x_t\sim p_{t|0}(x_t|x_0)} \left[\frac{1}{t} \sum_{k=1}^{|o_i|} \mathbf{1}[x_t^k = \texttt{mask}] \log \pi_\theta(x_0^k \mid x_t) \right], \tag{1}$$

where L is the length of the sequence and x_0^k is the k-th token of x_0. Specifically, intermediate steps t are sampled from uniform distribution, and masked sequence is sampled following the predefined forward process $p_{t|0}(x_t \mid x_0)$. The ELBO can be estimated by the sum of the conditional log-probabilities $\log \pi_\theta(x_0^k \mid x_t)$ of the tokens that are masked at t. This ELBO has also been used to approximate the marginal likelihood $\log \pi_\theta(x_0)$ for fine-tuning (Nie et al., 2025a; Yang et al., 2025; Zhu et al., 2025a).

2.2. Existing Policy Optimization Methods

The base method of current prevailing RL fine-tuning algorithms are Trust Region Policy Optimization (TRPO) (Schulman et al., 2015) or called Proximal Policy Optimization (PPO) with fixed KL penalty (Schulman et al., 2017), in which *forward* KL (FKL) divergence is applied to restrict the policy update:

$$\max_\theta\ \mathbb{E}_{q\sim\mathcal{D},\ o\sim\pi_\theta(\cdot|q)} \left[A^{\pi_{\text{old}}}(q,o) - \lambda D_{\text{KL}}(\ \pi_{\text{old}}(\cdot|q)\ \|\ \pi_\theta(\cdot|q)\ ) \right], \tag{2}$$

where $A^{\pi_{\text{old}}}$ is the advantage function, q and o are denoted as the prompt and (clean) response, respectively. Theorem 1 (see Appendix A) demonstrates the monotonic policy improvement of FKL regularized policy optimization.

PPO then extends the soft constraint (KL penalty) to clipping policy ratio $\pi_\theta(\cdot|q)/\pi_{\text{old}}(\cdot|q)$ for restricting the policy update, further employed in fine-tuning (Ouyang et al., 2022) with additional reverse-KL regularization w.r.t. the reference policy π_{ref}. Group Relative Policy Optimization (GRPO) (Shao et al., 2024) simplifies PPO by sampling a group of completions $o_{1:G}$ and approximating their advantage with their normalized rewards. This advantage is corrected by removing the standard deviation (Liu et al., 2025) $\hat{A}_i = R(q,o_i) - \texttt{mean}(R(q,o_{1:G}))$, which is called group-relative advantage. Denote r_i^k to be policy ratio $\pi_\theta(o_i^k)/\pi_{\text{old}}(o_i^k)$, and for conditional distribution a, b, $D_{\text{KL}}^{\mathcal{D}}(a\|b) = \mathbb{E}_{q\sim\mathcal{D}}[D_{\text{KL}}(a(\cdot|q)\|b(\cdot|q))]$, the GRPO objective is

$$\mathcal{L}_{\text{GRPO}}(\theta) = \mathbb{E}_{\substack{q\sim\mathcal{D}, \\ o_{1:G}\sim\pi_\theta(\cdot|q)}} \left[\frac{1}{G} \sum_{i=1}^{G} \frac{1}{|o_i|} \sum_{k=1}^{|o_i|} \min\left(r_i^k \hat{A}_i,\ \text{clip}(r_i^k, 1-\varepsilon, 1+\varepsilon)\hat{A}_i \right) \right] - \beta D_{\text{KL}}^{\mathcal{D}}\left[\pi_\theta \| \pi_{\text{ref}}\right]. \tag{3}$$

2.3. Group Relative Policy Optimization for Masked dLLMs

Adapting GRPO to diffusion-based large language models (dLLMs) presents notable challenges, primarily due to the method's dependence on sequence-level log-likelihoods. Unlike autoregressive models, dLLMs generate outputs via a non-autoregressive, iterative denoising process, making the computation of $\log \pi_\theta(o|q)$ intractable and necessitating approximation.

Existing works LLaDA (Nie et al., 2025a) and MMaDA (Yang et al., 2025) employ ELBO as the per-token log-likelihood approximation $\phi^\pi(o^k) = \phi^\pi(x_0^k) = \exp(\mathbb{E}_{t\in\mathcal{U}[0,1]}[\mathbf{1}[x_t^k = \texttt{mask}] \log \pi(x_0^k|x_t, q)])$. However, an accurate estimation requires a large sample size, resulting in inefficiency for online RL. An alternative method introduced in d1 (Zhao et al., 2025) is $\phi^{\pi}(o^k) = \phi^{\pi}(x_0^k) = \pi(x_0^k|q')$, where q' is randomly masked prompt. To obtain the sequence-level log-likelihood, all the current approaches are based on the mean-field approximation $\log \pi(o|q) \approx \sum_k \log \pi(x_0^k|q)$.

Diffusion-based GRPO (Zhao et al., 2025; Yang et al., 2025) can then be written as:

$$\mathbb{E}_{q\sim\mathcal{D},\, o_{1:G}\sim\pi_{\text{old}}(\cdot|q)}\left[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{k=1}^{|o_i|}\min\left(\frac{\phi^{\pi_\theta}(o_i^k)}{\phi^{\pi_{\text{old}}}(o_i^k)}, 1-\epsilon, 1+\epsilon\right)\hat{A}_i - \beta D_{\text{KL}}\left[\phi^{\pi_\theta}(\cdot)\,\|\,\phi^{\pi_{\text{ref}}}(\cdot)\right]\right]. \tag{4}$$

However, regardless of the specific approximation method employed, the approximated likelihood appears in the denominator of the policy ratio, where even minor estimation errors can introduce disproportionately large bias when $\phi^{\pi_\theta}(o_i^k) > \phi^{\pi_{\text{old}}}(o_i^k)$, and thus lead to instability during training. Moreover, the approximation function ϕ is typically applied to three separate policies π_θ, π_{old}, and π_{ref} – resulting in additional computational cost compared to RL for AR models. In this report, we seek a policy optimization method that minimizes reliance on approximated likelihoods, thereby improving both fine-tuning stability and efficiency for *diffusion*-based large language models.

3. wd1: Weighted Policy Optimization for dLLMs

3.1. Objective Derivation

As shown in previous works (Peng et al., 2019; Rafailov et al., 2023), policy optimization with reverse-KL regularization has a closed form solution. Approximating the target solution does not rely on importance sampling, and thus eliminates the dependence on policy ratio, i.e., $\pi_\theta(\cdot|q)/\pi_{\text{old}}(\cdot|q)$, and extensive likelihood approximation. Therefore, we propose to use reverse-KL regularization with respect to the old policy, along with standard reference policy regularization:

$$\max_\theta\ \mathbb{E}_{q\in\mathcal{D},o\sim\pi_\theta(\cdot|q)}\left[A^{\pi_{\text{old}}}(q,o) - \lambda D_{\text{KL}}\Big(\pi_\theta(\cdot|q)\,\|\,\pi_{\text{old}}(\cdot|q)\Big) - \beta D_{\text{KL}}\Big(\pi_\theta(\cdot|q)\,\|\,\pi_{\text{ref}}(\cdot|q)\Big)\right]. \tag{5}$$

Notably, reverse-KL regularized policy optimization objective in Equation (5) also enjoys monotonic improvement guarantees, as established in Theorem 3.

We note that the two reverse KL divergences in Equation (5) can be merged as $(\lambda+\beta)D_{\text{KL}}\big(\pi_\theta(\cdot|q)\,\|\,\pi_{\text{old}}^{\text{ref}}(\cdot|q)\big)$. Here $\pi_{\text{old}}^{\text{ref}}(\cdot|q) \propto \pi_{\text{old}}(\cdot|q)^{\lambda/(\lambda+\beta)} \cdot \pi_{\text{ref}}(\cdot|q)^{\beta/(\lambda+\beta)}$, is a geometric mixture between the reference and the old policy. We further recognize the solution to Equation (5) has the following form (Peng et al., 2019):

$$\pi^*(\cdot|q) \propto \pi_{\text{old}}^{\text{ref}}(\cdot|q)\cdot\exp\left(\psi A^{\pi_{\text{old}}}(q,\cdot)\right), \tag{6}$$

where we let $\psi := 1/(\lambda+\beta)$. Then, we can instead optimize the KL divergence $D_{\text{KL}}(\pi^*(\cdot|q)\,\|\,\pi_\theta(\cdot|q))$, resulting in the negative log-likelihood (NLL) loss $\mathcal{L}_{\text{NLL}}(\theta) := -\mathbb{E}_{q\sim\mathcal{D},o\sim\pi^*}[\log\pi_\theta(o|q)]$. We apply importance sampling with the proposal distribution $\pi_{\text{old}}^{\text{ref}}(o|q)$ and importance weights proportional to $\exp\left(\psi A^{\pi_{\text{old}}}(q,o)\right)$ to approximate the NLL loss (see Theorem 2):

$$\mathcal{L}_{\text{NLL}}(\theta) \approx -\mathbb{E}_{q\sim\mathcal{D},\{o_i\}_{i=1}^{\text{G}}\sim\pi_{\text{old}}^{\text{ref}}(\cdot|q)}\left[\sum_{i=1}^{\text{G}}\frac{\exp\left(\psi A^{\pi_{\text{old}}}(q,o_i)\right)}{\sum_{j=1}^{\text{G}}\exp\left(\psi A^{\pi_{\text{old}}}(q,o_j)\right)}\cdot\log\pi_\theta(o_i|q)\right] \tag{7}$$

Therefore, we obtain an alternative objective in Equation (7) that does not involve the policy ratio $\pi_\theta(\cdot|q)/\pi_{\text{old}}(\cdot|q)$ or $\pi_\theta(\cdot|q)/\pi_{\text{ref}}(\cdot|q)$. In contrast to the approaches of Zhao et al. (2025) and Zhu et al. (2025a), our algorithm only needs to approximate $\pi_\theta(\cdot\mid q)$.

Algorithm 1 `wd1`: Weighted Policy Optimization for **d**LLMs

Require: Reference model π_{ref}, prompt distribution $\mathcal{D}$, group size G, reward function R, dLLM π_θ, regularization hyperparameters λ and β

```
1:  Initialize π_θ ← π_ref
2:  while not converged do
3:      π_old ← π_θ
4:      Sample prompt q ~ D
5:      Sample G completions o_i ~ π_old^ref(· | q), ∀i ∈ [G]
6:      Compute group-relative advantage Â_i = R(q, o_i) − mean(R(q, o_{1:G})), ∀i ∈ [G]
7:      Compute weights w^+ and w^− in Equation (9), ∀i ∈ [G]
8:      for gradient update iterations n = 1, 2, . . . , μ  do
9:          Compute approximation of log-likelihood log π_θ(o_i|q)
10:         Compute objective L_wd1(θ) in Equation (8)
11:         Update π_θ by gradient descent
12:     end for
13: end while
14: return π_θ
```

3.2. Method

The objective in Equation (7) aims to amplify the likelihood of completions with high advantage. Crucially, due to the exponential form of the weighting in $\mathcal{L}_{NLL}$, completions with relatively low advantage – referred to as negative samples – may receive vanishingly small weights. To address this imbalance and make full use of the samples, we propose a complementary penalty term that minimizes the likelihood of low-advantage completions. Building on the weighted negative log-likelihood in Equation (7), we propose a novel weighted log-likelihood objective for dLLM policy optimization, where the weights are determined by group-relative advantage. Specifically, we define the original weights as w^+ and introduce negative weights w^- to downweight the likelihood of completions with low advantage:

$$\boxed{\mathcal{L}_{\texttt{wd1}}(\theta) = \mathbb{E}_{q\sim\mathcal{D},\{o_i\}_{i=1}^G\sim\pi_{\text{old}}^{\text{ref}}(\cdot|q)}\left[\sum_{i=1}^{G}\big(-w^+(q,o_i)+w^-(q,o_i)\big)\cdot\log\pi_\theta(o_i|q)\right],} \tag{8}$$

where the weights are defined with group-relative (GRPO) advantage $\hat{A}_i = R(q, o_i) - \texttt{mean}(R(q, o_{1:G}))$:

$$w^+(q,o_i) = \frac{\exp\big(\psi\hat{A}_i\big)}{\sum_{j=1}^G \exp\big(\psi\hat{A}_j\big)}, \qquad w^-(q,o_i) = \frac{\exp\big(-\psi\hat{A}_i\big)}{\sum_{j=1}^G \exp\big(-\psi\hat{A}_j\big)}. \tag{9}$$

Our method `wd1`, the **first w**eighted policy optimization for **d**LLMs, is formally presented in Algorithm 1. We first obtain G completions $\{o\}_{i=1}^G$ sampled from geometric mixture $\pi_{\text{old}}^{\text{ref}}(\cdot|q) \propto \pi_{\text{old}}(\cdot|q)^{\lambda/(\lambda+\beta)} \cdot \pi_{\text{ref}}(\cdot|q)^{\beta/(\lambda+\beta)}$ (line 5). Since our base model LLaDA parametrizes the clean token prediction $\pi_{\text{old}}^{\text{ref}}(x_0^k|x_t, q)$ for denoising, we approximate $\log \pi_{\text{old}}^{\text{ref}}(x_0^k|x_t, q) \approx \lambda \log \pi_{\text{old}}(x_0^k|x_t, q) + \beta \log \pi_{\text{ref}}(x_0^k|x_t, q)$ as the logits of the denoising distribution at each step t. We then use the samples to compute weights in Equation (9) (line 6). In weights computing, we leverage estimated partition functions $\sum_{j=1}^G \exp\big(A(q, o_j)\big)$ and $\sum_{j=1}^G \exp\big(-A(q, o_j)\big)$ for normalization, in order to restrict the absolute value of loss and thus restricting the gradient norm and stabilizing the training. Finally, we leverage the likelihood approximation introduced in `d1`: $\log \pi_\theta(o_i|q) \approx \sum_k \log \pi_\theta(o_i^k|q')$ (line 8), where q' is randomly masked from q at every gradient step n.

We simplify policy optimization to a weighted likelihood formulation, such that, during reasoning with dLLMs, the only required likelihood approximation is that of the current policy π_θ. This design significantly improves computational efficiency while reducing training bias and instability. In contrast to prior methods that rely extensively on multiple likelihood approximations across several policies, `wd1` offers a more streamlined and scalable reinforcement learning approach. As a result, `wd1` serves as a simple yet effective framework for scaling reasoning with diffusion-based language models[1].

4. Experiments

In this section, we empirically validate the following key advantages of our approach:

i) Improved reasoning capabilities and higher accuracy than existing method on popular reasoning benchmarks **without SFT**;
ii) reduced computational burden, as reflected by decreased runtime, lower FLOPs and numbers of function evaluations (NFEs) of the model; and
iii) marked performance gains attributable to the incorporation of samples with low-advantage.

To evaluate our approach, we next detail the experimental setup and implementation.

Experimental Setup. We perform reinforcement learning (RL) fine-tuning on the LLaDA-8B-Instruct model (Nie et al., 2025a), on four different benchmarks: GSM8k (Cobbe et al., 2021), MATH (Lightman et al., 2023), Sudoku (Arel, 2025), and Countdown (Pan et al., 2025). Our main baseline is `d1` (Zhao et al., 2025), the *first* RL method developed for masked diffusion LLMs (dLLMs). To ensure a fair comparison with our main baseline models, we use the same training dataset as in `d1` (Zhao et al., 2025).[2] Specifically, for GSM8k and MATH, we use the respective training splits, while for Sudoku and Countdown, we adopt the dataset splits provided by Zhao et al. (2025). We use s1K (Muennighoff et al., 2025) data for SFT in `d1`. We report results using zero-shot evaluation on sequence lengths of 256 and 512 tokens, with performance recorded at significantly less gradient steps than `d1`. Following `d1`, we report results from the best-performing checkpoint (more details provided in Appendix B.1).

Implementation. For a fair comparison, we reproduce the baseline methods *Diffu*-GRPO, which applies diffusion-based GRPO training directly to the LLaDA base model, and `d1`, which performs SFT before applying *Diffu*-GRPO. In our implementation of `wd1`, we apply the same likelihood approximation method as `d1`. Since previous works (Yu et al., 2025) have demonstrated that the reference policy is empirically unnecessary, we set $\beta = 0$ and $\lambda = 1$ to eliminate π_{ref} in practice. The hyperparameters used in our method and our reproduction of `d1` are listed in Table 5. The hyperparameters of SFT are provided in Table 4. For both methods, RL training is conducted on four NVIDIA A100 GPUs (80GB), and SFT is performed on four A6000 GPUs (48GB).

4.1. Main Results

Superior Reasoning Ability. In Table 1, we observe that `wd1`, even without supervised fine-tuning or using any supervised data, consistently outperforms our reproduced implementation of `d1`. Notably, `wd1` surpasses `d1` by 8% in test accuracy on the Sudoku task, and achieves up to a 25% improvement on Countdown with maximum length 256. Relative to the base LLaDA model, the performance gain reaches as high as 38%. On math problem-solving benchmarks such as GSM8K and MATH500, our approach attains slightly higher accuracy, further underscoring the effectiveness of `wd1` across tasks.

Reduced Training Cost. Table 2 demonstrates that the training cost required by `wd1` is substantially lower

[1] In `wd1` we include an additional term dependent on w^- in our objective. This make our objective different from Equation (6) and may alter the monotonic guarantee. We leave the theoretical analysis in the future work.

[2] We note that other concurrent works use larger datasets, which may lead to different performance outcomes on the same benchmarks (Yang et al., 2025; Zhu et al., 2025a; Huang et al., 2025).

Model	Sudoku		Countdown		GSM8K		MATH500	
	256	512	256	512	256	512	256	512
LLaDA-8B-Instruct	6.7	5.5	19.5	16.0	76.7	78.2	32.4	36.2
+ *diffu*-GRPO	16.1	9.3	27.0	34.0	80.7	79.1	34.4	39.0
+ SFT + *diffu*-GRPO (d1)	17.6	16.2	25.8	35.2	78.2	82.0	34.4	38.0
wd1	**25.2**	**24.2**	**51.2**	**46.1**	**80.8**	**82.3**	**34.4**	**39.0**

Table 1 Test Accuracy (%) across Different Tasks. wd1, **our approach without SFT, demonstrates consistently higher accuracy, particularly significant on Sudoku and Countdown.**

Model	SFT	RL Training		
	Time Cost	Time Cost	FLOPs	NFEs for Likelihood
d1	2.01 hrs	103.5 sec/step	9.922×10^{15}/step	$(\mu + 2)$/step
wd1	0 hrs	82.16 sec/step	8.887×10^{15}/step	μ/step

Table 2 Comparison of Training Cost on 4×A100. We show SFT cost, average training time, FLOPs evaluated by DeepSpeed Flops Profiler, and theoretical NFEs *per global step* which includes $\mu = 8$ gradient steps. wd1 eliminates the need for SFT and has less cost per-global-step in RL compared to d1.

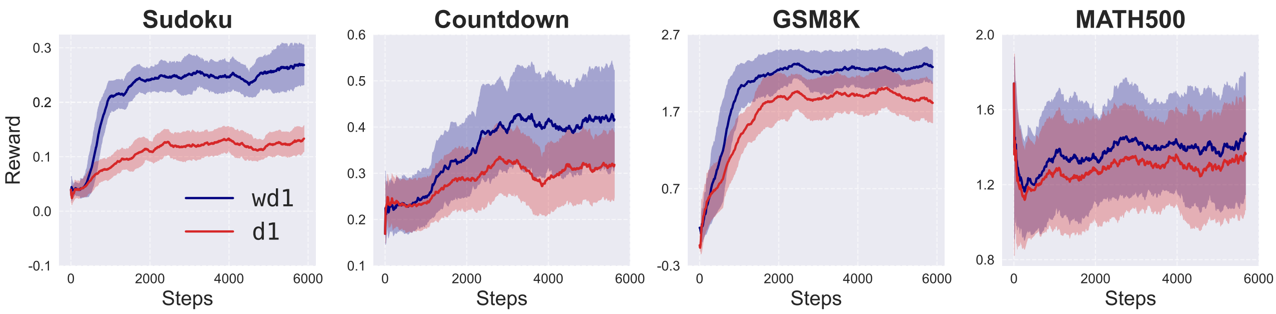

Figure 2 Training Rewards Dynamics of wd1 and d1. Standard deviation reported over a rolling window of 50 steps. The average reward of the samples generated by wd1 increase faster than the baseline d1.

than that of d1. Unlike d1, wd1 does not require a SFT stage, which alone accounts for approximately two hours of training in d1. wd1 achieves additional speedup during the RL phase, where runtime is measured by averaging over $\mu = 8$ inner gradient steps per global step. Notably, the time efficiency gap is expected to widen further under settings with larger maximum sequence lengths and more diffusion steps. This efficiency gain is further supported by a reduced FLOPs and number of function evaluations (NFEs) per step, as wd1 bypasses the need to approximate the likelihood of the old policy. We exclude NFEs associated with sampling, since both methods share identical sampling costs as wd1 removes the reference policy regularization.

4.2. Training Dynamics

Figure 2 presents the reward dynamics over gradient steps during training. wd1 exhibits a notably faster reward increase compared to d1, highlighting its superior sample efficiency–effectively leveraging the reward signal to accelerate policy optimization. In addition, Figure 4 shows the average length of generated completions during training. On math reasoning benchmarks such as GSM8K and MATH500, wd1 converges to shorter output sequences than d1, suggesting improved token efficiency

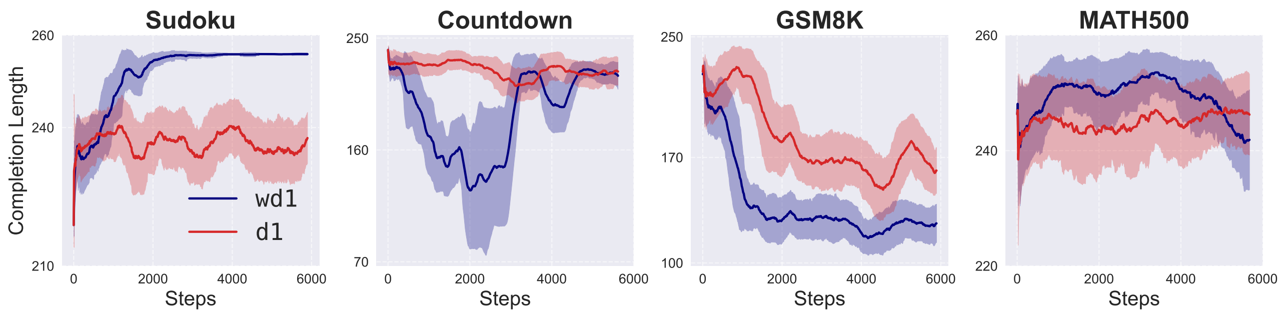

Figure 3 Completion lengths dynamics of wd1 and d1. In math problem-solving tasks (GSM8K and MATH500), our method demonstrates smaller completion lengths and better token efficiency.

Model	Sudoku		Countdown		GSM8K		MATH500	
	256	512	256	512	256	512	256	512
wd1	25.2	**24.2**	**51.2**	**46.1**	**80.8**	**82.3**	34.4	**39.0**
wd1-SFT	**26.5**	24.2	43.4	43.4	80.7	82.0	**36.4**	**39.0**
wd1-P	6.69	6.84	13.67	4.69	65.66	78.17	29.40	22.80

Table 3 Ablation on SFT and Negative Samples Weight (w^-), denoted by wd1-SFT and wd1-P (positive only), respectively. Results show that wd1 performs well without SFT, supporting R1-Zero-style training. Removing negative sample reinforcement (w^-) significantly hurts performance, highlighting its importance.

while maintaining or improving performance.

4.3. Ablation Study

We present an ablation study in Figure 3. Notably, we observe that supervised fine-tuning (SFT) yields only marginal improvements within our approach, with a slight gain in the Sudoku task. This contrasts with d1, where SFT plays a significant role in improving performance. These findings indicate that wd1 can eliminate the need for an SFT phase, thereby simplifying the training pipeline and substantially reducing computational cost. Additionally, we evaluate the impact of removing the negative-weighted term by setting $w^- = 0$, thus relying solely on the positive advantage weights w^+. The resulting degradation in performance highlights the importance of explicitly penalizing the likelihood of completions with low advantage, reinforcing the contribution of negative samples in effective policy optimization.

5. Related Work

RL for Diffusion-based LLM. Reinforcement learning for discrete diffusion models has been explored through several approaches. One line of work, exemplified by DRAKES (Wang et al., 2024), leverages reward backpropagation along the denoising trajectory. This approach requires computing a critic and propagating gradients through each denoising step, which is computationally intensive and prone to vanishing gradients. Alternatively, methods such as MMaDA (Yang et al., 2025) and d1 (Zhao et al., 2025) adopt direct RL formulations like GRPO, approximating missing diffusion components—such as per-token likelihoods—for policy optimization. Zhu et al. (2025a) applies Direct Preference Optimization (DPO) to fine-tune the LLaDA base model (Nie et al., 2025a), achieving notable gains in reasoning tasks. However, these approaches all depend on likelihood ratios, which can introduce bias and instability due to likelihood approximation errors. In contrast, our method derives a weighted policy optimization approach that eliminates the need for explicit policy ratios. Importantly, similar to prior works, our method directly optimizes the predictive distribution over clean data. A complementary line of research formulates policy optimization in terms of concrete scores (Lou et al., 2024; Meng et al., 2022). SEPO (Zekri and Boullé, 2025), for instance, introduces a policy optimization objective that only depends on concrete score estimation, thereby circumventing likelihood approximation altogether.

RL for AR Models. The connection between GRPO and weighted regression has recently been explored in the context of RL with verifier reward (Mroueh, 2025), where binary rewards simplify policy optimization into likelihood-based objectives. Other closely related approaches are Rejection Sampling Fine-Tuning (RAFT), which maximizes the likelihood of positive-reward samples (Xiong et al., 2025). Extensions of this idea incorporate negative samples to actively penalize the likelihood of negative-reward completions while enhancing that of high-reward ones (Zhu et al., 2025b; Chen et al., 2025). Other works introduce negative penalization through contrastive methods, such as Noise Contrastive Estimation (NCE) (Gutmann and Hyvärinen, 2012; van den Oord et al., 2019; Chen et al., 2024). Beyond binary rewards, preference-based learning has been widely studied using the Bradley–Terry model (Bradley and Terry, 1952; Ouyang et al., 2022; Rafailov et al., 2024; Azar et al., 2023; Ethayarajh et al., 2024; Wang et al., 2023; Hong et al., 2024). In contrast to these approaches, our method accommodates general reward signals and can be interpreted as a form of soft rejection sampling, enabling efficient and stable policy optimization for dLLMs.

RL via Weighted Regression. RL via weighted regression has been explored in earlier works advantage-weighted regression (AWR) (Peng et al., 2019; Peters et al., 2010), and more recently in the context of continuous control with diffusion policies (Ding et al., 2024; Zhang et al., 2025). Weighted likelihood-based approaches have also been proposed for fine-tuning autoregressive (AR) language models using general reward functions (Du et al., 2025; Baheti et al., 2024; Zhu et al., 2023). However, for AR models, where likelihoods are tractable, the necessity of such approaches remains unclear. In contrast, dLLMs suffer from intractable likelihoods, making weighted likelihood formulations particularly advantageous by reducing the number of required likelihood approximations. As such, RL via weighted likelihood provides a natural and efficient fit for optimizing dLLMs. In addition, we demonstrate in ablation study that merely optimizing policy with AWR (`wd1-P`) is ineffective.

6. Conclusion

In this work, we introduce `wd1`, a weighted policy optimization method for reasoning with diffusion-based large language models (dLLMs). `wd1` is designed to minimize reliance on likelihood approximation in policy optimization, thereby mitigating the potentially substantial bias that can arise from approximation errors in policy ratios. Our method is grounded in a weighted log-likelihood objective, derived to approximate the closed-form solution to the reverse-KL-constrained policy optimization. Empirically, we show that `wd1`, even without supervised fine-tuning, surpasses the existing method `d1` by up to 16% in accuracy on reasoning benchmarks, while also delivering notable improvements in computational efficiency during RL training. These results highlight the effectiveness of `wd1` and establish it as a more scalable and efficient approach for fine-tuning dLLMs.

References

Arel. Arel's sudoku generator. https://www.ocf.berkeley.edu/~arel/sudoku/main.html, 2025. Accessed: 2025-04-08.

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, Mark Rowland, Bilal Piot, Daniel Guo, Daniele Calandriello, Michal Valko, and

Rémi Munos. A General Theoretical Paradigm to Understand Learning from Human Preferences, 2023. URL https://arxiv.org/abs/2310.12036.

Ashutosh Baheti, Ximing Lu, Faeze Brahman, Ronan Le Bras, Maarten Sap, and Mark Riedl. Leftover Lunch: Advantage-Based Offline Reinforcement Learning for Language Models, 2024. URL https://arxiv.org/abs/2305.14718.

Ralph Allan Bradley and Milton E Terry. Rank Analysis of Incomplete Block Designs: I. The Method of Paired Comparisons. *Biometrika*, 39(3/4):324–345, 1952.

Andrew Campbell, Joe Benton, Valentin De Bortoli, Thomas Rainforth, George Deligiannidis, and Arnaud Doucet. A Continuous Time Framework for Discrete Denoising Models. *Advances in Neural Information Processing Systems*, 35:28266–28279, 2022.

Huayu Chen, Guande He, Lifan Yuan, Ganqu Cui, Hang Su, and Jun Zhu. Noise Contrastive Alignment of Language Models with Explicit Rewards, 2024. URL https://arxiv.org/abs/2402.05369.

Huayu Chen, Kaiwen Zheng, Qinsheng Zhang, Ganqu Cui, Yin Cui, Haotian Ye, Tsung-Yi Lin, Ming-Yu Liu, Jun Zhu, and Haoxiang Wang. Bridging Supervised Learning and Reinforcement Learning in Math Reasoning. *arXiv preprint arXiv:2505.18116*, 2025.

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

Shutong Ding, Ke Hu, Zhenhao Zhang, Kan Ren, Weinan Zhang, Jingyi Yu, Jingya Wang, and Ye Shi. Diffusion-Based Reinforcement Learning via Q-Weighted Variational Policy Optimization. *arXiv preprint arXiv:2405.16173*, 2024.

Yuhao Du, Zhuo Li, Pengyu Cheng, Zhihong Chen, Yuejiao Xie, Xiang Wan, and Anningzhe Gao. Simplify RLHF as Reward-Weighted SFT: A Variational Method, 2025. URL https://arxiv.org/abs/2502.11026.

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.

Michael U. Gutmann and Aapo Hyvärinen. Noise-Contrastive Estimation of Unnormalized Statistical Models, with Applications to Natural Image Statistics. *Journal of Machine Learning Research*, 13(11):307–361, 2012. URL http://jmlr.org/papers/v13/gutmann12a.html.

Jiwoo Hong, Noah Lee, and James Thorne. ORPO: Monolithic Preference Optimization without Reference Model, 2024. URL https://arxiv.org/abs/2403.07691.

Zemin Huang, Zhiyang Chen, Zijun Wang, Tiancheng Li, and Guo-Jun Qi. Reinforcing the Diffusion Chain of Lateral Thought with Diffusion Language Models. *arXiv preprint arXiv:2505.10446*, 2025.

Sham Kakade and John Langford. Approximately Optimal Approximate Reinforcement Learning. In *Proceedings of the nineteenth international conference on machine learning*, pages 267–274, 2002.

Inception Labs, Samar Khanna, Siddhant Kharbanda, Shufan Li, Harshit Varma, Eric Wang, Sawyer Birnbaum, Ziyang Luo, Yanis Miraoui, Akash Palrecha, Stefano Ermon, Aditya Grover, and Volodymyr Kuleshov. Mercury: Ultra-Fast Language Models Based on Diffusion, 2025. URL https://arxiv.org/abs/2506.17298.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*, 2023.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding R1-Zero-Like Training: A Critical Perspective. *arXiv preprint arXiv:2503.20783*, 2025.

Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete Diffusion Modeling by Estimating the Ratios of the Data Distribution, 2024. URL https://arxiv.org/abs/2310.16834.

Chenlin Meng, Kristy Choi, Jiaming Song, and Stefano Ermon. Concrete Score Matching: Generalized Score Matching for Discrete Data. *Advances in Neural Information Processing Systems*, 35:34532–34545, 2022.

Youssef Mroueh. Reinforcement Learning with Verifiable Rewards: GRPO's Effective Loss, Dynamics, and Success Amplification. *arXiv preprint arXiv:2503.06639*, 2025.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. s1: Simple test-time scaling, 2025. URL https://arxiv.org/abs/2501.19393.

Shen Nie, Fengqi Zhu, Chao Du, Tianyu Pang, Qian Liu, Guangtao Zeng, Min Lin, and Chongxuan Li. Scaling Up Masked Diffusion Models on Text. *arXiv preprint arXiv:2410.18514*, 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. *arXiv preprint arXiv:2502.09992*, 2025a.

Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. Large Language Diffusion Models, 2025b. 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, 2025. URL https://arxiv.org/abs/2406.03736.

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.

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

Xue Bin Peng, Aviral Kumar, Grace Zhang, and Sergey Levine. Advantage-Weighted Regression: Simple and Scalable Off-Policy Reinforcement Learning. *arXiv preprint arXiv:1910.00177*, 2019.

Jan Peters, Katharina Mulling, and Yasemin Altun. Relative Entropy Policy Search. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 24, pages 1607–1612, 2010.

David Pollard. Asymptopia: An Exposition of Statistical Asymptotic Theory. *URL http://www. stat. yale. edu/pollard/Books/Asymptopia*, 2000.

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.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct Preference Optimization: Your Language Model is Secretly a Reward Model, 2024. URL https://arxiv.org/abs/2305.18290.

Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 3505–3506, 2020.

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, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust Region Policy Optimization. In *International conference on machine learning*, pages 1889–1897. PMLR, 2015.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal Policy Optimization Algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, et al. Deepseekmath: Pushing the Limits of Mathematical Reasoning in Open Language Models. *arXiv preprint arXiv:2402.03300*, 2024.

Jiaxin Shi, Kehang Han, Zhe Wang, Arnaud Doucet, and Michalis K. Titsias. Simplified and Generalized Masked Diffusion for Discrete Data, 2025. URL https://arxiv.org/abs/2406.04329.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation Learning with Contrastive Predictive Coding, 2019. URL https://arxiv.org/abs/1807.03748.

Chaoqi Wang, Yibo Jiang, Chenghao Yang, Han Liu, and Yuxin Chen. Beyond Reverse KL: Generalizing Direct Preference Optimization with Diverse Divergence Constraints, 2023. URL https://arxiv.org/abs/2309.16240.

Chenyu Wang, Masatoshi Uehara, Yichun He, Amy Wang, Tommaso Biancalani, Avantika Lal, Tommi Jaakkola, Sergey Levine, Hanchen Wang, and Aviv Regev. Fine-Tuning Discrete Diffusion Models via Reward Optimization with Applications to DNA and Protein Design. *arXiv preprint arXiv:2410.13643*, 2024.

Wei Xiong, Jiarui Yao, Yuhui Xu, Bo Pang, Lei Wang, Doyen Sahoo, Junnan Li, Nan Jiang, Tong Zhang, Caiming Xiong, and Hanze Dong. A Minimalist Approach to LLM Reasoning: From Rejection Sampling to Reinforce, 2025. URL https://arxiv.org/abs/2504.11343.

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, 2025. URL https://hkunlp.github.io/blog/2025/dream.

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

Oussama Zekri and Nicolas Boullé. Fine-Tuning Discrete Diffusion Models with Policy Gradient Methods. *arXiv preprint arXiv:2502.01384*, 2025.

Shiyuan Zhang, Weitong Zhang, and Quanquan Gu. Energy-Weighted Flow Matching for Offline Reinforcement Learning. *arXiv preprint arXiv:2503.04975*, 2025.

Siyan Zhao, Devaansh Gupta, Qinqing Zheng, and Aditya Grover. d1: Scaling Reasoning in Diffusion Large Language Models via Reinforcement Learning. *arXiv preprint arXiv:2504.12216*, 2025.

Banghua Zhu, Hiteshi Sharma, Felipe Vieira Frujeri, Shi Dong, Chenguang Zhu, Michael I Jordan, and Jiantao Jiao. Fine-Tuning Language Models with Advantage-Induced Policy Alignment. *arXiv preprint arXiv:2306.02231*, 2023.

Fengqi Zhu, Rongzhen Wang, Shen Nie, Xiaolu Zhang, Chunwei Wu, Jun Hu, Jun Zhou, Jianfei Chen, Yankai Lin, Ji-Rong Wen, et al. LLaDA 1.5: Variance-Reduced Preference Optimization for Large Language Diffusion Models. *arXiv preprint arXiv:2505.19223*, 2025a.

Xinyu Zhu, Mengzhou Xia, Zhepei Wei, Wei-Lin Chen, Danqi Chen, and Yu Meng. The Surprising Effectiveness of Negative Reinforcement in LLM Reasoning. *arXiv preprint arXiv:2506.01347*, 2025b.

A. Theoretical Insights

Reinforcement Learning Formulation. We first introduce the reinforcement learning notations and then extend it to the setting of LLM post-training. Denote τ as a trajectory ($\tau = (s_0, a_0, s_1, \dots) \sim \pi$) sampled following policy π. Specifically, $s_0 \sim \mu$, $a_t \sim \pi(\cdot|s_t)$, $s_{t+1} \sim P(\cdot|q_t, a_t)$. The objective of Reinforcement Learning aims to find policy π, which maximizes a discounted total return,

$$\eta(\pi) = \mathbb{E}_{\tau\sim\pi}\Big[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t, s_{t+1})\Big].$$

Let the discounted return of a trajectory be $R(\tau) = \sum_{t=0}^{\infty} \gamma^t r(s_t, a_t, s_{t+1})$. The advantage function is defined as $A^\pi(s,a) = Q^\pi(s,a) - V^\pi(s)$, where $V^\pi(s) = \mathbb{E}_{\tau\sim\pi}[R(\tau)|s_0 = s]$ is state value function, and $Q^\pi(s,a) = \mathbb{E}_{\tau\sim\pi}[R(\tau)|s_0 = s, a_0 = a]$ is state-action value function. Denote $\rho_{\pi_{\text{old}}}$ as the marginal state distribution. Denote the total variation of two discrete probability distributions a, b by $D_{TV}(a,b) := \frac{1}{2}\sum_i |a_i - b_i|$ and $D_{TV}(a,b)^2 \le D_{\text{KL}}(a \parallel b)$ (Pollard, 2000; Schulman et al., 2015). When a and b are conditional probability distribution, denote $D_{TV}^{\max}(a,b) = \max_q D_{\text{TV}}(a(\cdot|q)\|b(\cdot|q))$ and $D_{\text{KL}}^{\max}(a \parallel b) = \max_q D_{\text{KL}}(a(\cdot|q)\|b(\cdot|q))$.

We then extend RL for LLM post-training. In this paper we only consider the sequence-level reward and loss objective, so we directly replace s with q and a with completion o. Then the horizon of the RL for post-training becomes only 1. The following theorem provides a monotonic (non-decreasing) guarantee of existing prevailing RL methods.

Theorem 1 (Policy Improvement Bound (Kakade and Langford, 2002; Schulman et al., 2015)). *Let surrogate objective* $L_{\pi_{old}}(\pi) = \eta(\pi_{old}) + \mathbb{E}_{s\sim\rho_{\pi_{old}}(\cdot),\ a\sim\pi(\cdot|s)}\left[A^{\pi_{old}}(s,a)\right]$, *and* $C = 4\max_{s,a,\pi}|A^\pi(s,a)|\gamma/(1-\gamma)^2$, *then* $\forall k \in \mathbb{N}$*:*

$$\eta(\pi_{\theta^*}) \ge L_{\pi_{old}}(\pi_{\theta^*}) - CD_{TV}^{\max}(\pi_{old}, \pi_{\theta^*})^2.$$

Remark 1. *Based on Theorem 1, due to* $D_{TV}^{\max}(a||b)^2 \le D_{KL}^{\max}(a||b)$ *(Pollard, 2000; Schulman et al., 2015), TRPO and PPO with fixed FKL regularization have the monotonic improvement guarantees. In other words,* $\eta(\pi_{\theta^*}) \ge L_{\pi_{old}}(\pi_{\theta^*}) - CD_{TV}^{\max}(\pi_{old}, \pi_{\theta^*})^2 \ge L_{\pi_{old}}(\pi_{\theta^*}) - C\mathbb{E}[D_{KL}(\pi_{old}\|\pi_{\theta^*})] \ge L_{\pi_{old}}(\pi_{old}) = \eta(\pi_{old})$.

Theorem 2. *The NLL objective satisfies that*

$$\mathcal{L}_{NLL} := -\mathbb{E}_{q\sim\mathcal{D},o\sim\pi_{\theta^*}}[\log \pi_\theta(o|q)] \approx -\mathbb{E}_{q\sim\mathcal{D},\{o_i\}_{i=1}^{G}\sim\pi_{old}^{ref}(\cdot|q)}\Big[\sum_{i=1}^{G} \frac{\exp\left(\psi A^{\pi_{\text{old}}}(q,o_i)\right)}{\sum_{j=1}^{G}\exp\left(\psi A^{\pi_{\text{old}}}(q,o_j)\right)} \cdot \log \pi_\theta(o_i|q)\Big]. \tag{10}$$

Proof. To obtain the practical objective in Equation (7), we first start from the cross-entropy loss, and obtain the following:

$$\begin{aligned}
\mathcal{L}_{\text{NLL}} &= -\mathbb{E}_{q\sim\mathcal{D},\ o\sim\pi_{\theta^*}(\cdot|q)}\Big[\log \pi_\theta(o|q)\Big] && (11)\\
&= -\mathbb{E}_{q\sim\mathcal{D}}\Big[\sum_o \pi_{\theta^*}(o|q)\cdot\log\pi_\theta(o|q)\Big] && (12)\\
&= -\mathbb{E}_{q\sim\mathcal{D}}\Big[\sum_o \frac{\pi_{\text{old}}^{\text{ref}}(o|q)\exp\left(\psi A^{\pi_{\text{old}}}(q,o)\right)}{\sum_{o'}\pi_{\text{old}}^{\text{ref}}(o'|q)[\exp\left(\psi A^{\pi_{\text{old}}}(q,o')\right)]}\cdot\log\pi_\theta(o|q)\Big] && (13)\\
&= -\mathbb{E}_{q\sim\mathcal{D},o\sim\pi_{\text{old}}^{\text{ref}}(o|q)}\Big[\frac{\exp\left(\psi A^{\pi_{\text{old}}}(q,o)\right)}{\mathbb{E}_{o\sim\pi_{\text{old}}^{\text{ref}}}[\exp\left(\psi A^{\pi_{\text{old}}}(q,o)\right)]}\cdot\log\pi_\theta(o|q)\Big] && (14)\\
&\approx -\mathbb{E}_{q\sim\mathcal{D},\{o_i\}_{i=1}^{G}\sim\pi_{\text{old}}^{\text{ref}}(o|q)}\Big[\frac{1}{G}\sum_{i=1}^{G}\frac{\exp\left(\psi A^{\pi_{\text{old}}}(q,o_i)\right)}{\mathbb{E}_{o\sim\pi_{\text{old}}^{\text{ref}}}[\exp\left(\psi A^{\pi_{\text{old}}}(q,o)\right)]}\cdot\log\pi_\theta(o_i|q)\Big]. && (15)
\end{aligned}$$

In the above we went from Equation (12) to Equation (13) by simply using the know form of the optimal policy $\pi^*(\cdot|q) \propto \pi_{\text{old}}^{\text{ref}}(\cdot|q) \cdot \exp\left(\psi \hat{A}(q,\cdot)\right)$. We go from Equation (13) to (14) by using the definition of expectation and from Equation (14) to Equation (15) by approximating through G samples $\{o_i\}_{i=1}^{G} \sim \pi_{\text{old}}^{\text{ref}}(o|q)$.

Finally, we can approximate the normalization constant $\mathbb{E}_{o\sim\pi_{\text{old}}^{\text{ref}}}\left[\exp\left(\psi A^{\pi_{\text{old}}}(q,o)\right)\right]$ using sampling, thus:

$$\mathcal{L}_{\text{NLL}} \approx -\mathbb{E}_{q\sim\mathcal{D},\{o_i\}_{i=1}^{G}\sim\pi_{\text{old}}^{\text{ref}}(\cdot|q)}\left[\frac{1}{G}\sum_{i=1}^{G}\frac{\exp\left(\psi A^{\pi_{\text{old}}}(q,o_i)\right)}{\frac{1}{G}\sum_{j=1}^{G}\exp\left(\psi A^{\pi_{\text{old}}}(q,o_j)\right)}\cdot\log\pi_\theta(o_i|q)\right] \tag{16}$$

$$= -\mathbb{E}_{q\sim\mathcal{D},\{o_i\}_{i=1}^{G}\sim\pi_{\text{old}}^{\text{ref}}(\cdot|q)}\left[\sum_{i=1}^{G} w(q,o_i)\cdot\log\pi_\theta(o_i|q)\right], \tag{17}$$

where $w(q,o_i) = \exp\left(\psi A^{\pi_{\text{old}}}(q,o_i)\right)/\sum_{j=1}^{G}\exp\left(\psi A^{\pi_{\text{old}}}(q,o_j)\right)$. □

Theorem 3. *Reverse-KL-regularized Policy Optimization defined in the following objective has monotonic improvement guarantees. Specifically, denote regularized objective* $\eta'(\pi) = \eta(\pi) - \mathbb{E}_{q\in\mathcal{D}}\left[\beta D_{KL}(\pi(\cdot|q) \,\|\, \pi_{ref}(\cdot|q))\right]$ *and denote*

$$M(\pi) = L(\pi) - \mathbb{E}_{q\in\mathcal{D}}\Big[\lambda D_{KL}(\pi(\cdot|q)\|\pi_{old}(\cdot|q)) + \beta D_{KL}\Big(\pi(\cdot|q)\,\|\,\pi_{ref}(\cdot|q)\Big)\Big], \tag{18}$$

where $L(\pi) = \eta(\pi_{old}) + \mathbb{E}_{q\sim\mathcal{D},\ o\sim\pi(\cdot|q)}\left[A^{\pi_{old}}(q,o)\right]$. *Let* θ^* *be the solution to the objective* $\max_\theta M(\pi_\theta)$*:*

$$\theta^* = \arg\max_\theta\ \mathbb{E}_{q\in\mathcal{D},o\sim\pi_\theta(\cdot|q)}\Big[A^{\pi_{old}}(q,o) - \lambda D_{KL}(\ \pi_\theta(\cdot|q)\ \|\ \pi_{old}(\cdot|q)\ ) - \beta D_{KL}\Big(\pi_\theta(\cdot|q)\,\|\,\pi_{ref}(\cdot|q)\Big)\Big] \tag{19}$$

then $\eta'(\pi_{\theta^*}) \geq \eta'(\pi_{old})$.

Proof. Based on Theorem 1, we have

$$\begin{aligned}
\eta'(\pi_{\theta^*}) =& \eta(\pi_{\theta^*}) - \mathbb{E}_{q\in\mathcal{D}}\Big[\beta D_{\text{KL}}\Big(\pi_{\theta^*}(\cdot|q)\,\|\,\pi_{\text{ref}}(\cdot|q)\Big)\Big] \\
\geq& L(\pi_{\theta^*}) - C D_{\text{TV}}^{\max}(\pi_{\text{old}},\pi_{\theta^*})^2 - \mathbb{E}_{q\in\mathcal{D}}\Big[\beta D_{\text{KL}}\Big(\pi_{\theta^*}(\cdot|q)\,\|\,\pi_{\text{ref}}(\cdot|q)\Big)\Big] && (20)\\
\geq& L(\pi_{\theta^*}) - C D_{\text{KL}}^{\max}(\pi_{\theta^*}\|\pi_{\text{old}}) - \mathbb{E}_{q\in\mathcal{D}}\Big[\beta D_{\text{KL}}\Big(\pi_{\theta^*}(\cdot|q)\,\|\,\pi_{\text{ref}}(\cdot|q)\Big)\Big] && (21)\\
\geq& L(\pi_{\theta^*}) - \mathbb{E}_{q\in\mathcal{D}}\Big[\lambda D_{\text{KL}}(\pi_{\theta^*}(\cdot|q)\|\pi_{\text{old}}(\cdot|q)) + \beta D_{\text{KL}}\Big(\pi_{\theta^*}(\cdot|q)\,\|\,\pi_{\text{ref}}(\cdot|q)\Big)\Big] && (22)\\
=& M(\pi_{\theta^*}) && (23)\\
\geq& M(\pi_{\text{old}}) && (24)\\
=& L(\pi_{\text{old}}) - \mathbb{E}_{q\in\mathcal{D}}\Big[\lambda D_{\text{KL}}(\pi_{\text{old}}(\cdot|q)\|\pi_{\text{old}}(\cdot|q)) + \beta D_{\text{KL}}\Big(\pi_{\text{old}}(\cdot|q)\,\|\,\pi_{\text{ref}}(\cdot|q)\Big)\Big] && (25)\\
\geq& L(\pi_{\text{old}}) - \mathbb{E}_{q\in\mathcal{D}}\Big[\beta D_{\text{KL}}\Big(\pi_{\text{old}}(\cdot|q)\,\|\,\pi_{\text{ref}}(\cdot|q)\Big)\Big] && (26)\\
=& \eta(\pi_{\text{old}}) - \mathbb{E}_{q\in\mathcal{D}}\Big[\beta D_{\text{KL}}\Big(\pi_{\text{old}}(\cdot|q)\,\|\,\pi_{\text{ref}}(\cdot|q)\Big)\Big] && (27)\\
=& \eta'(\pi_{\text{old}}) && (28)
\end{aligned}$$

Equation (20) holds due to Theorem 1. Equation (21) holds due to $D_{\text{TV}}^{\max}(p||q)^2 \leq D_{\text{KL}}^{\max}(p||q)$ (Pollard, 2000). Equation (22) holds due to the definition of $D_{\text{KL}}^{\max}$. Equation (23) is according to the definition of $M(\cdot)$. The key inequality Equation (24) holds since π_{θ^*} is the maximizer of function $L(\cdot)$. Equation (25) holds due to the definition of $M(\cdot)$. Equation (26) holds since $D_{\text{KL}}(\pi_{\text{old}}(\cdot|q)\|\pi_{\text{old}}(\cdot|q)) = 0$. Equation (27) holds since $L(\pi_{\text{old}}) = \eta(\pi_{\text{old}}) + \mathbb{E}_{q\sim\mathcal{D},\ o\sim\pi_{\text{old}}(\cdot|q)}\left[A^{\pi_{\text{old}}}(q,o)\right] = \eta(\pi_{\text{old}})$. Equation (28) is from the definition of η'. □

B. Additional Details

B.1. Dataset, Training and Evaluation Protocol

We reproduce d1 by running the official code[3] without and change, and train our method wd1 evaluated for accuracy of the test datasets at steps 1000, 2500, 5000, 7500 in both GSM8k and MATH; at steps 1000, 2500, 4000, 5000 in Sudoku; and at 1000, 2500, 4000 in Countdown. We evaluate less checkpoints compared to d1.

On the GSM8K, we train models on the train split[4] and evaluate on the test split. On Countdown, we train on the 3-number subset of the dataset[5] from TinyZero (Pan et al., 2025), and evaluate on 256 synthetic 3-number questions provided by Zhao et al. (2025). On Sudoku we use the 4×4 dataset[6] generated by Arel (2025). We train on 1M unique puzzles and evaluate on 256 synthetic ones provided by Zhao et al. (2025). On MATH500, we train models on the train split[7].

B.2. Reward Function

We use the same reward function as in d1 (Zhao et al., 2025). For completion, we provide the details as following.

GSM8K. Following the Unsloth reward setup[8], we apply five addtive components: XML Structure Reward: +0.125 per correct tag; small penalties for extra content post-tags. Soft Format Reward: +0.5 for matching the pattern

<reasoning>...</reasoning><answer>...</answer>.

Strict Format Reward: +0.5 for exact formatting with correct line breaks. Integer Answer Reward: +0.5 if the answer is a valid integer. Correctness Reward: +2.0 if the answer matches ground truth.

Countdown. We include three cases: +1.0 if the expression reaches the target using the exact numbers. +0.1 if numbers are correct but target is missed. 0 otherwise.

Sudoku. The reward is the fraction of correctly filled empty cells, focusing on solving rather than copying.

MATH500. We include two additive subrewards. Format Reward is +1.00 for <answer> with `\boxed` inside; +0.75 for <answer> without `\boxed`; +0.50 for `\boxed` only. +0.25 for neither. Correctness Reward: +2.0 if the correct answer is in `\boxed{}`.

B.3. Sampling from Geometric Mixture

Although the approximate sampling strategy introduced in Section 3.2 eliminates the need to approximate the reference policy's likelihood, it incurs computational overhead, as generating a full completion requires multiple forward passes through the dLLM—compared to a single pass for likelihood estimation. An alternative is to sample from π_{old} and shift the advantage to $\hat{A}_i = A^{\pi_{\text{old}}}(q, o_i) + \beta \log \hat{\pi_{\text{ref}}}/(\lambda + \beta)$, which reintroduces the need for reference policy likelihood approximation. However, our method disables reference policy regularization, following prior works, thereby avoiding this issue entirely.

[3] https://github.com/dllm-reasoning/d1
[4] https://huggingface.co/datasets/openai/gsm8k
[5] https://huggingface.co/datasets/Jiayi-Pan/Countdown-Tasks-3to4
[6] https://github.com/Black-Phoenix/4x4-Sudoku-Dataset
[7] https://huggingface.co/datasets/ankner/math-500
[8] https://unsloth.ai/blog/r1-reasoning

B.4. Hyperparameters

We provide the hyperparameters of SFT in Table 4 and for wd1 in Table 5.

	bacth_size	max_length	learning_rate	grad_accum_steps
Value	1	4096	1e-5	4

Table 4 Hyperparameters of SFT.

Parameter	wd1	d1
Model and Precision		
use_peft	true	true
torch_dtype	bfloat16	bfloat16
load_in_4bit	true	true
attn_implementation	flash_attention_2	flash_attention_2
lora_r	128	128
lora_alpha	64	64
lora_dropout	0.05	0.05
peft_task_type	CAUSAL_LM	CAUSAL_LM
Training Configuration		
seed	42	42
bf16	true	true
sync_ref_model	True	True
ref_model_sync_steps	64	64
adam_beta1	0.9	0.9
adam_beta2	0.99	0.99
weight_decay	0.1	0.1
max_grad_norm	0.2	0.2
warmup_ratio	0.0001	0.0001
learning_rate	3e-6	3e-6
lr_scheduler_type	constant_with_warmup	constant_with_warmup
Batching and Evaluation		
per_device_train_batch_size	6	6
per_device_eval_batch_size	1	1
gradient_accumulation_steps	2	2
RL		
num_generations	6	6
max_completion_length	256	256
max_prompt_length	200	200
block_length	32	32
diffusion_steps	128	128
generation_batch_size	6	6
remasking	low_confidence	low_confidence
random_masking	True	True
p_mask_prompt	0.15	0.15
beta	0.00	0.04
epsilon	–	0.5
num_iterations	12	12

Table 5 Comparison of hyperparameters between wd1 and d1.

B.5. Training Cost Estimation

For the runtime measurements reported in Table 2, we set $\mu = 8$ and train for a total of 6 global steps, corresponding to 48 gradient update steps. We use a batch size of 4 and the rest of the hyperparameters are the same as in Table 5. To estimate the number of function evaluations (NFEs) involved in computing likelihood approximations, we count only the forward passes, as the number of backward passes remains consistent across methods. The additional NFEs observed in the d1 model arise from evaluating the likelihood under both the old and reference models, which are used for regularization. These extra evaluations are required only when new samples are drawn, as their outputs can be cached and reused across all gradient updates for μ. We additionally report the number of floating-point operations (FLOPs) per global training step, measured using the Flops Profiler from Rasley et al. (2020).

C. Additional Experiments

We additionally report results for comparison to the results of the baseline d1 reported in the paper (Zhao et al., 2025). As shown in Table 6, our method wd1 evaluated and selected from less checkpoints, can outperform d1 with a large margin in Sudoku and Countdown, achieving comparable performance in math problem-solving tasks.

Model	**Sudoku**		**Countdown**		**GSM8K**		**MATH500**	
	256	512	256	512	256	512	256	512
LLaDA-8B-Instruct	6.7	5.5	19.5	16.0	76.7	78.2	32.4	36.2
+ diffu-GRPO *(reported)*	12.9	11.0	31.3	37.1	79.8	81.9	37.2	39.2
+ diffu-GRPO *(reproduced)*	16.1	9.3	27.0	34.0	80.7	79.1	34.4	39.0
d1 *(reported)*	16.7	9.5	32.0	42.2	**81.1**	82.1	**38.6**	**40.2**
d1 *(reproduced)*	17.6	16.2	25.8	35.2	78.2	82.0	34.4	38.0
wd1	25.2	**24.2**	**51.2**	**46.1**	80.8	**82.3**	34.4	39.0
wd1-SFT	**26.5**	24.2	43.4	43.4	80.7	82.0	36.4	39.0
wd1-P	6.69	6.84	13.67	4.69	65.66	78.17	29.40	22.80

Table 6 Test accuracy (↑) across different tasks. Our method demonstrates higher accuracy, especially significant in Sudoku and Countdown. The shaded numbers indicate where our method performs the best.

We additionally provide reward dynamics in comparison to wd1-SFT in training. In Sudoku and Countdown, directly training with wd1 without SFT demonstrate significantly more efficient and stable learning process. In GSM8k and MATH500, the different is negligible.

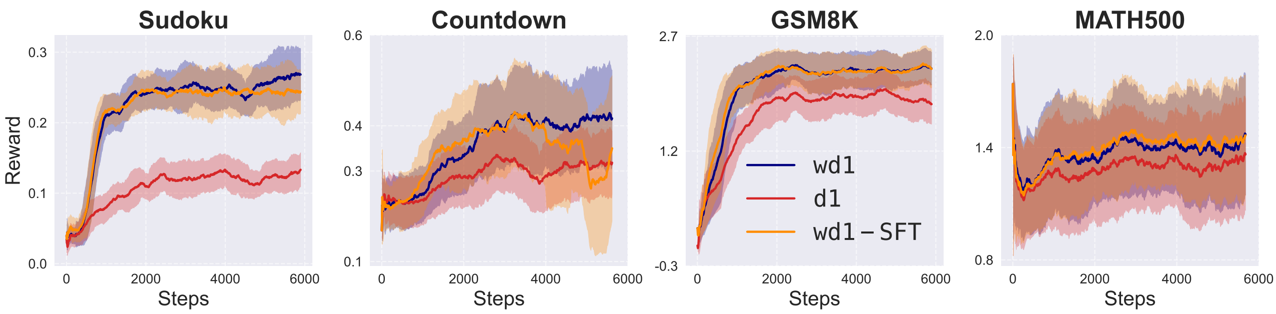

Figure 4 Reward Dynamics. wd1 without SFT demonstrates better rewards in Sudoku and Countdown.