Energy Matching based Preference Learning for Diffusion Language Models

Shiv Shankar
University of Massachusetts
sshankar@cics.umass.edu

Abstract

Policy-gradient reinforcement learning (RL) is widely used to improve language model reasoning, but existing methods are not compatible with diffusion language models. The primary reason for this is the difficulty of likelihood estimation with such models. We propose EMBR , a scalable off-policy framework that reformulates KL-regularized RL as an energy-based distribution matching problem. By aligning policy updates with reward signals through energy matching, EMBR avoids the overhead of on-policy learning and the variance of importance weighting. We further derive a principled upper bound for the energy matching objective which can be used to fine-tune dLLMs. Experiments on multiple benchmarks in both online and offline setting show that EMBR matches or surpasses the performance of GRPO and related baselines in the online case, and of DPO in the offline case. Our approach provides a practical alternative for post-training of diffusion LMs.

1 Introduction

Large Language Models (LLMs) have powered remarkable progress in code generation (Gehring et al., 2024), autonomous agents (Deng et al., 2023) and many language-based tasks (Ouyang et al., 2022). Most current applications use Reinforcement learning (RL) to post-train the LLMs' reasoning and generation abilities (Luong et al., 2024) for the task. Standard RL-based methods for tuning LLMs are based upon the policy gradient methods (Sutton and Barto, 2018) or PGRL. Based upon the classic REINFORCE estimator (Williams and Peng, 1990), PGRL uses direct stochastic gradient based optimization of a policy to maximize task-specific rewards. PGRL-based gradient estimation requires the ability to compute likelihoods of the generations. Most current LLMs are based on autoregressive (AR) transformer models, which have a naturally efficient way to compute the requisite likelihoods for the gradient update, and thus mesh well with PGRL.

Recently, diffusion-based language models (dLLMs) have emerged as an equally powerful way to train language models (Nie et al., 2025; Shi et al., 2024). dLLMs (also sometimes called Masked Diffusion Language Models) model sequence generation as an iterative denoising process, allowing them to break the sequentiality of autoregressive models. This allows dLLMs to significantly outperform autoregressive (AR) models in inference speed, especially when handling long sequences. Unfortunately, the policy-gradient methods that underpin the success of standard LLMs cannot directly be applied to dLLMs. Mainstream PGRL algorithms rely on a factorization of the sequence likelihood to efficiently compute gradients using conditional likelihoods. In contrast, dLLMs generate sequences non-autoregressively, making such conditionals computationally intractable.

Policy-gradient methods can be divided into two groups: a) on-policy methods and b) off-policy methods. A primary issue for on-policy methods is the need for continuous rollouts, making such training resource-intensive and slow (Sutton and Barto, 2018). Even when feasible, online policy-gradient-based alignment methods are shown to disproportionately favour a few tokens (Lin et al., 2024).

Off-policy learning offers a promising alternative by enabling the reuse of past trajectories, improving sample efficiency. However, this approach introduces its own difficulties. Specifically, it typically relies on importance weighting (IW) (Horvitz and Thompson, 1952) to correct for distribution mismatch between the training data and the current policy. Since reasoning tasks often involve long trajectories such as chain-of-thought (Wei et al., 2022) explanations or multi-step solutions (de Winter et al., 2024), importance weights often become unstable across lengthy token sequences. As a re-

Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics
Volume 4: Student Research Workshop, pages 776–786

sult, off-policy updates without careful correction risk severe bias, while exact IW can lead to impractically high variance. Additionally, when we consider off-policy/off-line methods for dLLMs, the problem of incorrect likelihood comes back two-fold. First, the gradients of the likelihood as required in PGRL are biased. Secondly, since policy updates require importance weights, the usage of incorrect likelihood makes the gradient update further inconsistent compared to the true objective.

Fortunately, the standard KL-regularized RL perspective of LLMs training has a natural interpretation to bayesian inference, where the KL term acts as a regularizer balancing the model prior with new reward-driven evidence. This probabilistic perspective leads to a broader divergence-minimization approaches in LLM fine-tuning (e.g. (Khalifa et al., 2020; Rafailov et al., 2023)).

Contributions Building on this insight, we leverage the implied distribution matching in KL-regularized RL to formulate a new method for fine-tuning dLLMs which does not involve importance weights (IW). Our approach, dubbed EMBR (Energy Matching Based Realignment), is inspired by energy matching (Chopra et al., 2006) and avoids the inefficiencies of on-policy rollouts and the instability of long-horizon importance weighting. The lack of importance weights makes our approach 'supervision-friendly' as one can use a general dataset of preference pairs for post-training the dLLM (similar to DPO (Rafailov et al., 2023)). Additionally this lack of IW, removes one of the errors used by biased likelihood approximations in dLLMs. Finally, we also describe principled alternatives to ELBO based DPO for fine-tuning dLLMs. This enables scalable, stable, and practical RL-based fine-tuning for reasoning-intensive tasks in dLLMs.

2 Preliminaries

Diffusion Language Model Diffusion language models (dLLMs) are conceptually analogous to continuous diffusion models in generative modeling. The basic idea is to systematically corrupt a given clean token sequence and subsequently learning to reverse this corruption to recover the original input. This framework is structured around two stochastic processes: a forward or noising process and a reverse or generative process.

The forward process begins with a clean text sequence $\mathbf{x} = x_{1:n}$ and progressively corrupts it into a noisy sequence $\mathbf{z}_t$ over timestep t. Corruption is implemented by independently replacing tokens with a special `[MASK]` token according to a noise schedule. At $t = 0$, $\mathbf{z}_0 = \mathbf{x}$, representing the original sequence, and at $t = 1$, $\mathbf{z}_1$ consists entirely of `[MASK]` tokens. The corruption for each token is governed by a forward transition kernel $q_{t|0}(z_{t,i} \mid x_i)$, defined as a categorical distribution that mixes the original token x_i and the `[MASK]` token.

$$q_{t|0}(z_{t,i} \mid x_i) = \text{Cat}\left(z_{t,i}; \alpha_t x_i + (1 - \alpha_t)\texttt{[MASK]}\right). \tag{FWD}$$

The reverse generation process is parameterized by a neural network policy π_θ, which is trained to denoise the corrupted sequence. It learns to predict the original tokens $\mathbf{x}$ from any intermediate corrupted state $\mathbf{z}_t$ by modeling the reverse transition from $\mathbf{z}_t$ to a less noisy state $\mathbf{z}_s$ (where $s < t$). Since the exact log-likelihood for the reverse paths is intractable, training objective for π_θ is derived from maximizing the Evidence Lower Bound (ELBO) on the log-likelihood of the clean data. The resulting objective can be written as

$$L_\theta(x) = \mathbb{E}_{t,z_t}\left[w(t)\sum_{i=1}^{L}\mathbb{I}[z_{t,i} = \texttt{[MASK]}] \cdot \log \pi_\theta(x_i|z_t)\right]. \tag{ELBO}$$

$L_\theta(\mathbf{x};\theta)$ involves an expectation over a random timestep $t \sim \mathcal{U}[0,1]$ and the corrupted sequence $\mathbf{z}_t$. The loss is computed only over tokens that are masked at timestep t (indicated by the indicator function $\mathbb{1}$) and is determined by the network's (π_θ) probability of predicting the original token x_i. While the exact ELBO would use a weighting dependent on the noising sequence α_t, in practice it is replaced by a time-dependent loss weight $w(t)$.Nie et al. (2025) set $w(t)$ to be $1/t$.

Group Relative Policy Optimization GRPO (Shao et al., 2024) is a PPO (Schulman et al., 2017b) based method for finetuning LLMs. GRPO usually samples multiple responses $y^{(i)}$ for each prompt x, uses a verifier (for math-like problems) or other reward functions to rate these samples, and computes advantages by normalizing rewards within each prompt group. The advantage for the i-th response $y^{(i)}$ is computed as:

$$\widehat{A}^{(i)} = \frac{r(x,y^{(i)}) - \text{mean}(r(x,y^{(1)}),..,r(x,y^{(G)}))}{\text{stdev}(r(x,y^{(1)}),..,r(x,y^{(G)}))}, \tag{1}$$

where $r(x, y^{(i)})$ is the outcome for response $y^{(i)}$ to prompt x as we defined above. In general for fine-tuning dLLMs the normalization is skipped (Nie et al., 2025).

GRPO uses the response-level advantage $\widehat{A}^{(i)}$ in the PPO objective (Schulman et al., 2017b), along with KL-regularization to give the following objective:

$$J^{\text{GRPO}}(\pi) = \sum_{k=1}^{G} \sum_{t=1}^{|y^{(i)}|} \min \left[\frac{\pi(y_k^{(i)} \mid s_k^{(i)})}{\pi_{\theta-}(y_k^{(i)} \mid s_k^{(i)})} \widehat{A}^{(i)}, \right.$$
$$\left. \text{clip} \left(\frac{\pi(y_k^{(i)} \mid s_k^{(i)})}{\pi_{\theta-}(y_t^{(i)} \mid s_t^{(i)})}, \epsilon \right) \widehat{A}^{(i)} \right]$$
$$- \beta D_{\text{KL}}(\pi \| \pi_{ref}),$$

where $y_k^{(i)}$ is the k^{th} token in the sequence y^i, and $s_k^i = (y_{<k}^{(i)}, x)$ is the concatenation of all processed tokens. clip is a function which clamps its input in the range $[1-\varepsilon, 1+\varepsilon]$. Effectively, instead of a per-step/action reward as in PPO, GRPO uses the entire trajectory reward in the objective, implicitly assigning each token in the response its corresponding reward. The validity of this objective however relies on the sequential factorization of the likelihood as evidenced by the term $\frac{\pi(y_k^{(i)}|s_k^{(i)})}{\pi_{\theta-}(y_k^{(i)}|s_k^{(i)})}$ which is the importance ratio for only the k-th token conditional on the history.

3 Related Work

Reinforcement learning (RL) with Kullback-Leibler (KL) regularization KL regularized learning has its roots in maximum-entropy RL Ziebart et al. (2008); Neu et al. (2017), where a KL penalty ensures that learned policies remain close to a reference distribution. This framework has been well studied in RL literature (Schulman et al., 2017a; Nachum et al., 2017; Haarnoja et al., 2018). Furthermore, the connection between KL-regularized control and KL-divergence-minimization is known since the seminal work of Jaynes (1979). Others have also noted the relation between bayesian inference and optimal control (Ziebart et al., 2008; Levine, 2018). Based on the form of the optimal policy of such a procedure (Ziebart et al., 2008) various direct alignment algorithms like DPO (Rafailov et al., 2023), IPO (Azar et al., 2024), KTO (Ethayarajh et al., 2024) have been proposed.

Post-Training of Diffusion Language Models Several methods have been proposed for post-training dLLMs. Nie et al. (2025) estimate the log-likelihood $\log \pi_\theta(y|x)$ using a Monte Carlo estimate of the ELBO. To reduce the computational cost, Zhao et al. (2025) utilize a mean-field variant of the output likelihood, which approximates the likelihood by performing a single denoising step for each token position independently. This approach results in biased optimization, and in practice, they must randomly mask different portions of the output. Despite the success of such heuristics, the use of biased gradients remains a fundamental issue, and even under ideal conditions, the method does not guarantee reward optimization. Zhu et al. (2025) also note the challenges of Monte Carlo ELBO approximation, particularly the variance of the estimate, and propose an antithetic sampling method to reduce this variance. Other approximations which adopt GRPO-style training and use other likelihood approximations have also been proposed (Shankar, 2025; Tang et al., 2025). Wang et al. (2025) have recently proposed using the evidence upper bound (EUBO) based to improve GRPO-style training of dLLMs by penalizing EUBO of negative samples.

Policy Gradient Methods Policy gradient methods (Williams and Peng, 1990) have been foundational in modern RL. Recent advancements for language model training (Ouyang et al., 2022; Shao et al., 2024) have been based on PPO (Schulman et al., 2017b) and its variants (Wu et al., 2023). However these methods rely on importance sampling and clipping mechanisms to ensure stable training (Wu et al., 2023). In contrast, EMBR avoids these complexities by adopting an off-policy approach, eliminating the need for importance sampling altogether. This design choice enhances EMBR 's applicability, particularly in offline settings where dataset densities are unknown, making it a more flexible alternative to PPO-based methods.

Reinforcement Learning for LLM Reasoning Ouyang et al. (2022) opened the floodgates for research on the application of MDP-based formulations for reasoning in large language models (LLMs). This has led to has seen significant progress, as seen in models like OpenAI's O1 and DeepSeek's R1. While policy-based RL methods such as GRPO (Guo et al., 2025), and their variants (e.g., DAPO (Yu et al., 2025), Dr. GRPO (Liu et al., 2025)) dominate this space, some other approaches like ReMax (Li et al., 2023) and RAFT (Dong et al., 2023) have also been explored. Re-

cently value based methods (Jia et al., 2025) have also been proposed for post-training LLMs. However, these methods are a) for AR models and b) for online learning. Unlike these methods EMBR supports an off-policy offline paradigm, offering potential advantages in sample efficiency.

4 Distribution Matching for Language Models

Ziebart et al. (2008) had reframed max-entropy RL as a problem of probabilistic inference. This connection provides the theoretical grounding of our proposal. Hence we first, describe this connection which will naturally lead to our proposed method.

Consider the unnormalized target distribution $\tilde{q}(\tau)$ over the space of trajectories given as:

$$\tilde{q}(\tau) = \pi_{\text{ref}}(\tau) \exp\left(r(\tau)/\beta\right), \tag{2}$$

where $\beta > 0$ is a temperature parameter controlling the deviation from the reference policy. Normalizing this yields the *Boltzmann* (or Gibbs) distribution (Jaynes, 1979):

$$q(\tau) = \frac{1}{Z}\pi_{\text{ref}}(\tau) \exp\left(r(\tau)/\beta\right), \tag{3}$$
$$\text{with } Z = \sum_{\tau} \pi_{\text{ref}}(\tau) \exp\left(r(\tau)/\beta\right).$$

Under ideal optimization, the standard RLHF objective $J_\beta(\theta) = \mathbb{E}_{\tau\sim\pi_\theta}[r(\tau)] - \beta D_{\text{KL}}(\pi_\theta \| \pi_{\text{ref}})$ leads to this target distribution. Expanding the KL divergence between π_θ and π_{ref} reveals the equivalence:

$$J_\beta = \mathbb{E}_{\tau\sim\pi_\theta}[r(\tau)] - \beta \mathbb{E}_{\tau\sim\pi_\theta}\left[\log \pi_\theta(\tau)/\pi_{\text{ref}}(\tau)\right] \tag{4}$$

$$= -\beta\, \mathbb{E}_{\tau\sim\pi_\theta}\left[\frac{\log \pi_\theta(\tau)}{\pi_{\text{ref}}(\tau) \exp r(\tau)/\beta}\right] \tag{5}$$

$$= -\beta \left(D_{\text{KL}}(\pi_\theta \| q) + \log Z\right). \tag{6}$$

Since Z is a constant independent of the policy parameters θ, **maximizing $J_\beta(\theta)$ is equivalent to minimizing the reverse Kullback-Leibler divergence** $D_{\text{KL}}(\pi_\theta \| q)$ between the learned policy and the target Boltzmann distribution.

This equivalence suggests an alternative fundamental goal of *distribution matching*: aligning π_θ with the unnormalized target $\tilde{q}(\tau)$ (or normalized target q). The canonical RLHF approach implicitly optimizes the reverse KL divergence. However, this naturally invites considering alternative divergence measures. While the optimal policy is invariant to the choice of divergence under ideal conditions of infinite model capacity and perfect optimization, practical considerations can lead to significantly different empirical behavior. Exploring this broader family of distribution matching objectives thus opens new pathways for the fine-tuning of large language models.

One desired property is to use previously logged data i.e. use off-policy learning. RLHF style KL minimization naturally leads towards on-policy learning as it computes expectations under π_θ. Another alternative is the forward KL, $D_{KL}(q||\pi)$, however this is difficult as it requires sampling from the unnormalized energy model $\tilde{q}$. Additionally, when the divergence does not go down to 0, forward KL can lead to mode-covering and overly diffuse models.

From this work's perspective, an ideal divergence should satisfy three key criteria: a) avoid requiring the partition function Z of q (efficiency), b) need not require sampling from π_θ (off-policy), and c) can directly use a preference dataset (supervision-friendly/offline friendly) . In the next section, we discuss energy matching, a candidate objective with such properties. Based on this objective, we call our proposed method EMBR , short for Energy Matching Based Realignment.

4.1 Energy Matching

The energy matching objective (Chopra et al., 2006) is designed to align the energy landscape of p_θ and q:

$$\mathcal{L}_{\text{EM}} = \min_c \mathbb{E}_{\tau\sim\mu}\left[\left(\log \pi_\theta(\tau) - \log \tilde{q}(\tau) + c\right)^2\right]. \tag{7}$$

Here μ is an arbitrary distribution to draw samples from. By inspection, one can see that this loss is 0 if π and q match over the support of μ. Thus if μ has full support, then this objective has a unique global minimum which matches π and q.

Note that $\log q$ is just $\log \tilde{q}$ shifted by the log-partition function, which can be absorbed into the parameter c. Thus Eq 7 can equivalently be written as minimizing the MSE between the log probabilities. However this version of the loss removes dependence to the unknown $\log Z$ by minimizing the variance of the energy difference. It encourages π_θ to match $\tilde{q}$ up to a constant shift: Thus, at

optimum, the energy difference is constant:

$$\log \pi_\theta(\tau) - \log \pi_{\text{ref}}(\tau) - r(\tau)/\beta = \text{const}, \quad (8)$$

which implies $\pi_\theta(\tau) \propto \pi_{\text{ref}}(\tau) e^{r(\tau)/\beta} = q(\tau)$.

Therefore, minimizing $\mathcal{L}_{\text{EM}}$ achieves the same stationary point as maximizing $J(\theta)$: $p_\theta \propto \tilde{q}$, which matches the goal of minimizing $D_{\text{KL}}(\pi_\theta \| q)$.

Notice that *we do not need to restrict the trajectory sampling* μ to a specific model or distribution as long as its positive wherever π_θ, q are positive [1]. As such this objective can be used for off-policy learning, but unlike standard off-policy methods, we do not need to compute importance weights/ density ratios. We can set $\mu = \pi$ (for on-policy learning), or any other distribution with required support over data. Only the energy difference $\log \pi_\theta$ and $\log q$ needs to be computable per sample.

Incorporating Conditional Generation The previous objective was written directly in terms of general samples τ. For the case of model alignment or math proving, we have outputs conditioned on the prompts. Thus we write the corresponding conditional objective:

$$\mathcal{L}_{EM} = \mathbb{E}_{y\sim\mu(x)}\Big[\Big(\log \pi_\theta(y;x) - \log \tilde{q}(y;x) + c(x)\Big)^2\Big] \quad (9)$$

which requires a prompt/context dependent c function. Unlike unconditional models, where c can be optimized as a free parameter, one now requires a model for c. However, when the model π_θ is conditioned on the prompt, as is usual in language models, we can use the layers of the same model as input to an MLP to predict c(x) as well.

Relation to RL Objective EMBR is related to policy gradient (Williams and Peng, 1990) as an instantiation of the off-policy policy gradient, but unlike standard PG methods, it does not rely on importance sampling or trust-regions. Furthermore, it can be used even with offline data collected from unknown densities. The relation between EMBR and standard RLHF can be formalized by evaluating $\mathcal{L}_{\text{EM}}$ over samples from $\mu = \pi_\theta$. Let:

$$\begin{aligned} f(\tau) &= [\log \pi_\theta(\tau) - \log \pi_{\text{ref}}(\tau)] - r(\tau)/\beta \quad (10)\\ &= \log \pi_\theta(\tau) - \log \tilde{q}(\tau) \end{aligned}$$

Then we can write the gradient of L_{EM} as:

$$\begin{aligned} \nabla_\theta \mathcal{L}_{\text{EM}}(\theta) &= 2\mathbb{E}\Big[(\log \pi_\theta(y;x) - \log \tilde{q}(y;x) - c(x)) \\ &\qquad \nabla_\theta \log \pi_\theta(y;x)\Big] \quad (11)\\ &= 2\,\text{Cov}_{\tau\sim\pi_\theta}\left(f(\tau), \nabla_\theta \log \pi_\theta(\tau)\right). \quad (12) \end{aligned}$$

which is upto scaling factors the same as the *on-policy policy gradient* for $J(\theta)$ (Sutton and Barto, 2018). Thus, not only the optimum policy but even the gradient of energy matching coincides with policy gradient dynamics under a KL-regularized objective.

4.2 Contrastive Energy Matching

An alternative to optimizing for the function c in Equation (9) is to note that c is purely a function of the prompt x (and in fact is related to the normalization constant Z of $\tilde{q}$), and does not depend on the generations y. Thus, we can eliminate c from the objective by using another generation y'. This gives the following pairwise or contrastive objective

$$\begin{aligned} \mathcal{L}_{\text{CM}} = \mathbb{E}_{y\sim\mu(x),y'\sim\mu'(x)}\Big[\Big(&\log \pi_\theta(y;x) - \log \tilde{q}(y;x) \\ &- \log \pi_\theta(y';x) + \log \tilde{q}(y';x)\Big)^2\Big] \end{aligned} \quad (13)$$

Note that the sample y' need not come from the same distribution as μ. As long as μ' also has full support, its relation to μ has no impact on the optimality. Thus this is a "supervised-friendly" loss (Flet-Berliac et al., 2024) as it does not involve a) any additional model or architecture changes and b) sampling from trained policies. Instead like in DPO one can use a dataset of preference pairs [2].

4.3 Principled Upper Bound

While theoretically sound and seemingly easy to implement the objective of Equation 14, requires likelihood under π . However when π is given by a dLLM, this objective is still not tractable. One natural idea is to use the ELBO value itself as the likelihood; and in fact many existing methods directly use the ELBO as the likelihood (Zhu et al., 2025; Tang et al., 2025).

When it comes to training a model via maximum-likelihood, an ELBO style loss has a principled nature by being a lower bound to the true objective. However, both the $\mathcal{L}_{EM}$ and the $\mathcal{L}_{CM}$ objectives involve the difference of likelihood terms. If we replace them with the ELBO, the resultant objective is neither a lower or upper bound to the original objective; and thus it is unclear how optimizing them improves the underlying expected reward objective.

[1] Since q is just re-weighted π_{ref}, and π_θ starts from π_{ref}, this effectively just means support over the reference model.

[2] L_{EM} is also offline friendly, but since it requires an additional network to compute the function $c(x)$ it requires greater access into the model architecture

To bypass this, we propose to use the variational EUBO or Evidence Upper Bound (Ji and Shen, 2019). For the specific case of dLLM loss, the EUBO is given by:

$$U_\theta(x;\gamma) = \frac{1}{\gamma}\sum_{i=1}^{L} \log \mathbb{E}_{t,z_t}\Big[w(t)\mathbb{1}[z_{t,i} = \texttt{[MASK]}] \cdot \log \pi_\theta^\gamma(x_i|z_t)\Big],$$

where γ is a constant ≥ 1 that controls how close U is to the true likelihood (with γ closer to 1 being tighter). Similar to the ELBO, the expectation is taken over sampling time t and noised sequences z_t. In practice, the expectations is computed via monte-carlo sampling.

Using the EUBO U and the ELBO L we can derive a principled objective that is always an upper-bound to the contrastive loss . To see this, consider the differences

$$\Delta_\theta := \log \pi_\theta(\tau) - \log \pi_\theta(\tau'), \tag{14}$$
$$\Delta_{\text{ref}} := \log \pi_{\text{ref}}(\tau) - \log \pi_{\text{ref}}(\tau'), \tag{15}$$
$$\Delta_r := r(\tau) - r(\tau'). \tag{16}$$

Then the contrastive loss can be written as

$$\mathcal{L}_{\text{CM}}(\theta) = \big(\Delta_\theta - \Delta_{\text{ref}} - \Delta_r\big)^2. \tag{17}$$

Since we have lower and upper bounds on the model log-likelihood:

$$L_\theta(\tau) \leq \log p_\theta(\tau) \leq U_\theta(\tau),$$
$$L_\theta(\tau') \leq \log p_\theta(\tau') \leq U_\theta(\tau'),$$

the difference Δ_θ lies in the interval

$$\Delta_\theta \in \big[L_\theta(\tau) - U_\theta(\tau'),\ U_\theta(\tau) - L_\theta(\tau')\big]. \tag{18}$$

An analogous bound can be written for the reference model likelihoods. Combining these we get that the full scalar $S = \Delta_\theta - \Delta_{\text{ref}} - \Delta_r$, lies in an interval $[S_{\min}, S_{\max}]$, where

$$S_{\min} = \big(L_\theta(\tau) - U_\theta(\tau')\big) - \big(U_{\text{ref}}(\tau) - L_{\text{ref}}(\tau')\big) - \Delta_r \tag{19}$$

$$S_{\max} = \big(U_\theta(\tau) - L_\theta(\tau')\big) - \big(L_{\text{ref}}(\tau) - U_{\text{ref}}(\tau')\big) - \Delta_r. \tag{20}$$

Thus, we have a strict upper bound on the true contrastive squared loss given by:

$$\mathcal{L}_{\texttt{UCM}} = \max\{S_{\min}^2, S_{\max}^2\}. \tag{21}$$

Algorithm 1 Training Algorithm

Initialize $\pi_\theta \leftarrow \pi_{\text{ref}}$
$\mathcal{D} = \phi$ i.e. empty set for online learning
$\mathcal{D}$ is the preference dataset $\mathcal{D}_{pref}$ if doing offline learning
while not converged **do**
  Sample a prompt $x \sim \mathcal{D}_{task}$
  Sample G completions $y_i \sim \pi_\theta(\cdot \mid x),\ i \in [G]$
  Standard offline learning does not usually produce new completions
  For each y, compute reward r
  Add (x, y, r) tuples to $\mathcal{D}$
  for gradient update iterations $n \in [N]$ **do**
    Sample M tuples $\mathcal{D}_{batch}$ from $\mathcal{D}$
    If using contrastive variant, sample contrastive pairs y' for each x in $\mathcal{D}_{batch}$
    Sample random time-steps t for each tuple in $\mathcal{D}_{batch}$
    From t, y sample z_t by randomly masking tokens (see eq. FWD)
    Compute loss (Eq 9 or Eq 13) by using ELBO L_θ, L_{ref} instead of $\log \pi_\theta, \log_{\pi_{ref}}$
    If using the upper bound approach compute EUBO U_θ, U_{ref} and use Eq 21.
    Update π_θ by gradient descent
  end for
end while
  return π_θ

We present an algorithmic description of the training procedure in Algorithm 1. We note that EMBR training can be used in an online fashion (with an experience replay buffer $\mathcal{D}$; or in an offline fashion with a labeled preference dataset $\mathcal{D}_{pref}$. Unlike standard online (off-policy or on-policy learning) EMBR does not need importance weights. This allows the same algorithm/code to be used in either fashion with minimal adjustments. We specifically highlight in red the difference when doing offline training in the algorithm. Furthermore we have presented three different losses viz. vanilla energy matching $\mathcal{L}_{EM}$, the contrastive loss $\mathcal{L}_{CM}$ and the upper bound loss $\mathcal{L}_{UCM}$.

5 Experiments

We experiment with our method under two different settings. First is the standard setting for training most dLLMs. Under this setting, as the model gets updated, one keeps producing new generations from the model. In RL terminology, this is usu-

ally called online learning. To improve efficiency, one typically uses a replay buffer to keep a history of generations, with old samples being discarded. Most works in this direction use the GRPO update, but work with different methods of estimating the log-likelihood using π_θ (Zhao et al., 2025; Shankar, 2025; Tang et al., 2025).

The second setting is of offline learning from a preference dataset. In this version, instead of a dynamic replay buffer of recent generations, we have a static dataset of generations pertinent to the task at hand. Most current works on dLLM ignore the offline learning setting, with (to the best of our knowledge) the exception of VRPO (Zhu et al., 2025).

For either setting, we will use the recent dLLM LLaDA-8B (Nie et al., 2025) as the baseline model, which we then fine-tune based on different alignment methods.

5.1 Online Learning

Datasets We focus on tasks and datasets commonly used in the dLLM literature (Tang et al., 2025; Zhao et al., 2025). These include a) GSM8K (Cobbe et al., 2021), a dataset of multi-step grade school math problems, b) (Lightman et al., 2023), a curated subset of high-level math problems, and c) HumanEval, a coding benchmark. For mathematical tasks, we follow the same train-test splitting, reward functions, and evaluation protocol as (Zhao et al., 2025). For coding tasks, we follow the protocol in Gong et al. (2025) and train on a subset of AceCoder-87K (Zeng et al.).

Models We train the LLada model (Nie et al., 2025) with the different EMBR algorithms, labeled EMBR -E, EMBR -C and EMBR -U corresponding to the vanilla energy matching $\mathcal{L}_{EM}$, the contrastive loss $\mathcal{L}_{CM}$ and the upper bound loss $\mathcal{L}_{UCM}$ respectively. As baselines we consider recent dLLM training methods like diffu-GRPO (Zhao et al., 2025), wd1 (Tang et al., 2025), and UniGRPO (Yang et al., 2025) . We did not run the baseline methods and instead report results from existing literature.

For both RL rollouts and evaluation, we use the confidence-based decoding strategy common in earlier works (Nie et al., 2025; Zhao et al., 2025). During training, exploration is encouraged by using a higher sampling temperature of 0.9. During evaluation, the sampling temperature is set to 0.0. We report results for test accuracy across generation lengths of 128, 256, and 512. Our experiments are based on the code and hyperparameters provided in Zhao et al. (2025) [3].

Results are presented in Table 1, where we see that EMBR consistently achieves competitive or superior accuracy compared to diffu-GRPO/d1 (Zhao et al., 2025). In general we see that EMBR -E is worse than the other variants. This is not surprising, as in some sense the c function of Equation 9 needs to be learnt which based on chosen parameterization may not be optimal. We also see that the principled upper bound EMBR -U in general works better than the others. This may not be surprising as the training objective puts a ceiling on how far the model might be from the target boltzmann model. Example reward dynamics on GSM are presented in the Appendix.

5.2 Offline Learning

Next we consider the case of offline learning from a static dataset of generations and rewards. This setting closely matches the setting of DPO (Rafailov et al., 2023), where instead of learning a reward model and subsequent optimization of a LLM by RLHF, one optimizes the model directly. While well explored for standard AR-LLMs this setting has not been considered for dLLMs.

Offline learning can be sensitive to the choice of dataset. For these experiments we try to follow the procedure in Zhu et al. (2025). In that work, authors post-train the LLaDA model on a large collection of 350k generations across a wide range of topics such as Q&A, reasoning, mathematics, and coding. However the corresponding preference dataset has not been released. As such a direct comparison in the offline setting with their model is not feasible.

To best approximate a high quality preference data for the offline setting, we used the data generated from online learning. For fair comparisons, we create a static dataset for each task from the d1 model. Specifically for each task we generated samples from the actively optimized d1 model as it gets trained on the task. These outputs were then evaluated and correponding normalized rewards obtained. This constitutes the static dataset that is then used as $\mathcal{D}$ in Algorithm 1 .

Models We use the LLaDA model and focus on the methods described in Zhu et al. (2025). These include DPO and VRPO based optimization of the

[3] available at https://github.com/dllm-reasoning/d1

Task	GSM8K			MATH500			HumanEval		
Sequence Length	128	256	512	128	256	512	128	256	512
LLaDA	68.7	76.7	78.2	26.0	32.4	36.2	26.8	37.8	45.8
UniGRPO	74.9	82.5	82.7	32.4	37.4	39.4	-	-	-
d1	73.2	81.1	82.1	**33.8**	**38.6**	**40.2**	25.6	36.0	47.1
wd1	73.8	80.8	82.3	33.5	34.4	39.0	**34.7**	38.4	38.4
EMBR -E	72.5	82.3	83.0	32.1	36.5	39.4	28.1	39.3	45.3
EMBR -C	74.3	81.2	82.9	31.4	38.1	36.6	30.6	35.1	48.0
EMBR -U	**75.4**	**83.1**	**83.8**	33.4	37.1	39.4	30.9	**40.1**	**48.3**

Table 1: Model performances on different benchmarks tasks across different generation lengths for online learning.

base model. Note that the other methods like d1, UNIGRPO etc. are online learning methods and cannot be applied to offline setting. DPO is a variant of the original DPO loss (Rafailov et al., 2023) adapted to dLLMs by changing the log-likelihood used in DPO to the ELBO. Furthermore, our preference data is different from the one in Zhu et al. (2025), the numbers are not directly comparable to their results. We tried to achieve as close as possible to the reported results using the resources available to us.

Results are presented in Table 2. We see that in general offline methods competitive with online methods, though there does seem to be a shortfall. We attribute it to both the preference data size as well as the better exploration online methods can achieve when they consistently sample model dependent high likelihood trajectories.

We also see in the results the general pattern that EMBR -U outperforms other methods. On average EMBR -U improves by 2 points over other variants. Furthermore, all EMBR variants improve over DPO. One possibility which was not explored in this work but which makes VRPO (Zhu et al., 2025) better is the variance reduction tricks they apply in estimating the ELBO. Exploring such variance reduction in context of contrastive methods is an interesting future direction for exploration.

6 Conclusion

We have introduced EMBR , a novel approach to post-train dLLMs for reasoning on entire reasoning trajectories without using process models, on-policy learning, or importance sampling. The objective does not rely on sampling from the same model, and hence allows learning in a pure offline setting. Our method is a version of the energy matching objective from classical probabilistic inference (Chopra et al., 2006). Furthermore we propose two other variants of the energy matching objective: a version based on contrastive matching (Flet-Berliac et al., 2024) and another which provides a strict upper bound to the matching loss for dLLMs. Our experiments show the efficacy of these methods in both the online and offline setting.

Method	GSM8K	MATH500	HumanEval
DPO	58.2	31.6	35.1
VRPO†	63.9	35.2	40.0
EMBR -E	65.5	34.0	36.3
EMBR -C	63.1	36.4	35.0
EMBR -U	68.5	37.0	39.8

Table 2: Performance of various methods on benchmarks tasks for offline learning. † indicates this result is from Zhu et al. (2025) which has a different preference data used for training making the exact numbers incomparable.

Limitations

Previous works have shown that dLLM methods have generally been worse at solving tasks which rely on long horizon planning. Theoretically, the lack of importance weights in our method should help with such tasks; however our experiments do not cover such tasks, and any conclusion we draw are based on the limited experiments conducted here. Additionally, our approach focuses on utilizing rewards at the sequence levels. However, intermediate levels such as span-level rewards, also provide useful information for alignment tasks.

The current method cannot account for these intermediate reward levels. Future research could explore methods that incorporate multiple levels of rewards, potentially enhancing the flexibility and effectiveness of post-training. Another natural direction is to look at alternative surrogates for the upper bound loss in terms of hinge, softplus and Huber style losses. Finally, improving offline training using online samples and variance reduction techniques is another future direction of research.

References

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

Sumit Chopra, Raia Hadsell, M Ranzato, Fujie Huang, et al. 2006. A tutorial on energy-based learning. *Predicting structured data*, 1(0).

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

Joost CF de Winter, Dimitra Dodou, and Yke Bauke Eisma. 2024. System 2 thinking in openai's o1-preview model: Near-perfect performance on a mathematics exam. *Computers*, 13(11):278.

Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Sam Stevens, Boshi Wang, Huan Sun, and Yu Su. 2023. Mind2web: Towards a generalist agent for the web. *Advances in Neural Information Processing Systems*, 36:28091–28114.

Hanze Dong, Wei Xiong, Deepanshu Goyal, Yihan Zhang, Winnie Chow, Rui Pan, Shizhe Diao, Jipeng Zhang, Kashun Shum, and Tong Zhang. 2023. Raft: Reward ranked finetuning for generative foundation model alignment. *arXiv preprint arXiv:2304.06767*.

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

Yannis Flet-Berliac, Nathan Grinsztajn, Florian Strub, Bill Wu, Eugene Choi, Chris Cremer, Arash Ahmadian, Yash Chandak, Mohammad Gheshlaghi Azar, Olivier Pietquin, et al. 2024. Contrastive policy gradient: Aligning llms on sequence-level scores in a supervised-friendly fashion. *arXiv preprint arXiv:2406.19185*.

Jonas Gehring, Kunhao Zheng, Jade Copet, Vegard Mella, Quentin Carbonneaux, Taco Cohen, and Gabriel Synnaeve. 2024. Rlef: Grounding code llms in execution feedback with reinforcement learning. *arXiv preprint arXiv:2410.02089*.

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

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. 2018. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning*, pages 1861–1870. PMLR.

Daniel G Horvitz and Donovan J Thompson. 1952. A generalization of sampling without replacement from a finite universe. *Journal of the American statistical Association*, 47(260):663–685.

Edwin T Jaynes. 1979. Concentration of distributions at entropy maxima. *ET Jaynes: Papers on probability, statistics and statistical physics*, page 315.

Chunlin Ji and Haige Shen. 2019. Stochastic variational inference via upper bound. *Preprint*, arXiv:1912.00650.

Zeyu Jia, Alexander Rakhlin, and Tengyang Xie. 2025. Do we need to verify step by step? rethinking process supervision from a theoretical perspective. *arXiv preprint arXiv:2502.10581*.

Muhammad Khalifa, Hady Elsahar, and Marc Dymetman. 2020. A distributional approach to controlled text generation. *arXiv preprint arXiv:2012.11635*.

Sergey Levine. 2018. Reinforcement learning and control as probabilistic inference: Tutorial and review. *arXiv preprint arXiv:1805.00909*.

Ziniu Li, Tian Xu, Yushun Zhang, Zhihang Lin, Yang Yu, Ruoyu Sun, and Zhi-Quan Luo. 2023. Remax: A simple, effective, and efficient reinforcement learning method for aligning large language models. *arXiv preprint arXiv:2310.10505*.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let's verify step by step. *arXiv preprint arXiv:2305.20050*.

Zicheng Lin, Tian Liang, Jiahao Xu, Xing Wang, Ruilin Luo, Chufan Shi, Siheng Li, Yujiu Yang, and Zhaopeng Tu. 2024. Critical tokens matter: Token-level contrastive estimation enhence llm's reasoning capability. *arXiv preprint arXiv:2411.19943*.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. 2025. Understanding r1-zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*.

Trung Quoc Luong, Xinbo Zhang, Zhanming Jie, Peng Sun, Xiaoran Jin, and Hang Li. 2024. Reft: Reasoning with reinforced fine-tuning. *arXiv preprint arXiv:2401.08967*, 3.

Ofir Nachum, Mohammad Norouzi, Kelvin Xu, and Dale Schuurmans. 2017. Bridging the gap between value and policy based reinforcement learning. *Advances in Neural Information Processing Systems*, 30.

Gergely Neu, Anders Jonsson, and Vicenç Gómez. 2017. A unified view of entropy-regularized markov decision processes. *arXiv preprint arXiv:1705.07798*.

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.

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

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

John Schulman, Xi Chen, and Pieter Abbeel. 2017a. Equivalence between policy gradients and soft q-learning. *arXiv preprint arXiv:1704.06440*.

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

Shiv Shankar. 2025. Padre: Pseudo-likelihood based alignment of diffusion language models. In *2nd AI for Math @ ICML 2025*.

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

Jiaxin Shi, Kehang Han, Zhe Wang, Arnaud Doucet, and Michalis Titsias. 2024. Simplified and generalized masked diffusion for discrete data. *Advances in neural information processing systems*, 37:103131–103167.

Richard S Sutton and Andrew G Barto. 2018. *Reinforcement learning: An introduction*. MIT press.

Xiaohang Tang, Rares Dolga, Sangwoong Yoon, and Ilija Bogunovic. 2025. wd1: Weighted policy optimization for reasoning in diffusion language models. *Preprint*, arXiv:2507.08838.

Chenyu Wang, Paria Rashidinejad, DiJia Su, Song Jiang, Sid Wang, Siyan Zhao, Cai Zhou, Shannon Zejiang Shen, Feiyu Chen, Tommi Jaakkola, et al. 2025. Spg: Sandwiched policy gradient for masked diffusion language models. *arXiv preprint arXiv:2510.09541*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Ronald J Williams and Jing Peng. 1990. An efficient gradient-based algorithm for on-line training of recurrent network trajectories. *Neural computation*, 2(4):490–501.

Tianhao Wu, Banghua Zhu, Ruoyu Zhang, Zhaojin Wen, Kannan Ramchandran, and Jiantao Jiao. 2023. Pairwise proximal policy optimization: Harnessing relative feedback for llm alignment. *arXiv preprint arXiv:2310.00212*.

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

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

Huaye Zeng, Dongfu Jiang, Haozhe Wang, Ping Nie, Xiaotong Chen, and Wenhu Chen. Acecoder: Acing coder rl via automated test-case synthesis, 2025a. *URL https://arxiv. org/abs/2502.01718*.

Siyan Zhao, Devaansh Gupta, Qinqing Zheng, and Aditya Grover. 2025. d1: Scaling reasoning in diffusion large language models via reinforcement learning. *arXiv preprint arXiv:2504.12216*.

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.

Brian D. Ziebart, Andrew L. Maas, J. Andrew Bagnell, and Anind K. Dey. 2008. Maximum entropy inverse reinforcement learning. In *AAAI Conference on Artificial Intelligence*.

A Additional Information

Training Dynamics In Figure 1 we plot the training dynamics of different methods on the GSM (top) and MATH (bottom). We can see that EMBR learns faster than other post-training methods.

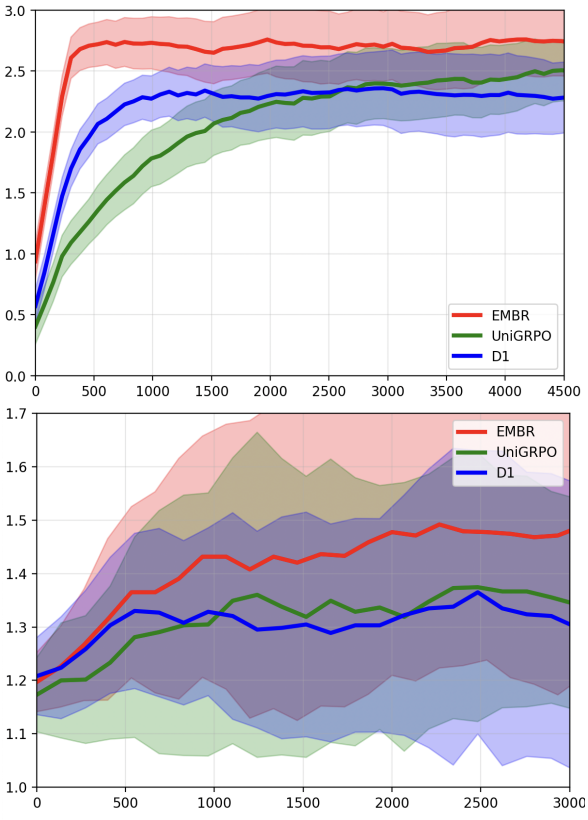

Figure 1: Reward dynamics of EMBR-U with standard error during online training compared with other methods on GSM and MATH.