

# DISCRETE DIFFUSION TRAJECTORY ALIGNMENT VIA STEPWISE DECOMPOSITION

000  
001  
002  
003  
004  
005  
006  
007  
008  
009  
010  
011  
012  
013  
014  
015  
016  
017  
018  
019  
020  
021  
022  
023  
024  
025  
026  
027  
028  
029  
030  
031  
032  
033  
034  
035  
036  
037  
038  
039  
040  
041  
042  
043  
044  
045  
046  
047  
048  
049  
050  
051  
052  
053  
Anonymous authors  
Paper under double-blind review

## ABSTRACT

Discrete diffusion models have demonstrated great promise in modeling various sequence data, ranging from human language to biological sequences. Inspired by the success of RL in language models, there is growing interest in further improving the models by alignment with a certain reward. In this work, we propose an offline preference optimization method to approach trajectory alignment for discrete diffusion models. Instead of applying the reward on the final output and backpropagating the gradient to the entire denoising process, we decompose the problem into a set of stepwise alignment objectives by matching the per-step posterior. This framework enables efficient diffusion optimization, is compatible with arbitrary reward functions, and importantly, yields an equivalent optimal solution under additive factorization of the trajectory reward. Experiments across multiple domains including DNA sequence design, protein inverse folding, and language modeling consistently demonstrate the superiority of our approach. Notably, it achieves an up to 12% improvement over the most competitive RL-based baseline in terms of predicted activity on DNA sequence design, and further improves the GSM8K score from 78.6 to 81.2 on LLaDA-8B-Instruct for language modeling.

## 1 INTRODUCTION

Diffusion models (DMs) (Sohl-Dickstein et al., 2015; Ho et al., 2020; Song et al., 2021) have emerged as a powerful tool for modeling distributions and generating samples across an array of modalities such as visual contents (Rombach et al., 2021; Saharia et al., 2022; Ho et al., 2022), natural languages (Nie et al., 2025; Lou et al., 2023; Shi et al., 2024; Sahoo et al., 2024), and geometric structures (Xu et al., 2022; Han et al., 2024b; Hoogeboom et al., 2022b), to name a few. Among them, discrete diffusion models (Austin et al., 2021; Campbell et al., 2022; Lou et al., 2023; Sahoo et al., 2024; Shi et al., 2024; Hoogeboom et al., 2022a), those that are in particular grounded on masked discrete latent variables, have demonstrated remarkable promise for modeling sequence data in discrete space, achieving superior performance on tasks ranging from DNA sequence design (Wang et al., 2024; Gosai et al., 2023) and protein inverse folding (Campbell et al., 2024; Wang et al., 2024; Hsu et al., 2022) to even text generation (Lou et al., 2023; Sahoo et al., 2024; Shi et al., 2024; Zheng et al., 2023; Gong et al., 2025) and chatbot (Nie et al., 2025; Ye et al., 2025).

Despite the promise, a critical question still remains unrevealed for discrete DMs: *How to align pretrained discrete diffusion models towards a target distribution, usually defined in the presence of certain reward?* Such problem has been of core interest in finetuning modern Large Language Models (LLMs) (Brown et al., 2020; Achiam et al., 2023; Team et al., 2023), a paradigm usually referred to as Reinforcement Learning with Human Feedback (RLHF) (Christiano et al., 2017; Stiennon et al., 2020; Ouyang et al., 2022) or preference optimization (Rafailov et al., 2023; Ji et al., 2024). It is vital in enhancing the applicability of the pretrained model on downstream tasks by biasing its distribution towards that with higher rewards, *e.g.*, higher enhancer activity for DNA sequence (Wang et al., 2024) or helpfulness and harmlessness for chatbots (Rafailov et al., 2023; Ji et al., 2024; Bai et al., 2022).

Existing alignment literature is primarily based on the left-to-right autoregressive modeling of sequences (Rafailov et al., 2023; Han et al., 2024a), and performing preference optimization is particularly challenging for discrete DMs, which hold the fundamentally different factorization with a Markov chain of sequence-level discrete random variables through a large number of diffusion steps. Previous work explored using RL to fine-tune the model, but the inherent discrete representation

054 makes it challenging to efficiently backpropagate the gradient to the entire sampling process, with the  
 055 reward typically computed upon the final output. Furthermore, this nature also makes it prohibitive to  
 056 efficiently compute exact likelihood and evaluate rewards when aligning the joint of latent variables  
 057 on the chain, leading to suboptimal performance (Wallace et al., 2024; Zhu et al., 2025b). The chained  
 058 sampling of discrete diffusion also makes online RL (Zhao et al., 2025) computationally exhaustive.

059 In this work, we propose a principled approach for preference optimization of discrete diffusion  
 060 models via *stepwise decomposition*. Our key innovation is to decompose the alignment of the entire  
 061 diffusion trajectory  $p_\theta(\mathbf{x}_{0:T})$  into a set of subproblems, each of which is responsible for aligning the  
 062 per-step **factorized approximation of the posterior**  $\hat{p}_\theta(\mathbf{x}_0|\mathbf{x}_t)$ , where  $\mathbf{x}_0$  is clean sequence and  $\mathbf{x}_t$  is  
 063 the latent variable at diffusion step  $t$ . Our stepwise decomposition takes the advantage of leveraging  
 064  $\hat{p}_\theta(\mathbf{x}_0|\mathbf{x}_t)$  as the per-step alignment target, thus enabling both efficient and accurate likelihood  
 065 computation and reward evaluation defined on clean sequence  $\mathbf{x}_0$ . Furthermore, we also theoretically  
 066 reveal a novel connection between our stepwise decomposition alignment and the original problem  
 067 by showing that the optimally aligned posteriors  $\hat{p}^*(\mathbf{x}_0|\mathbf{x}_t)$  induce a joint  $p^*(\mathbf{x}_{0:T})$  that is also an  
 068 optimal solution of the diffusion trajectory alignment objective, when the reward of the trajectory  
 069 takes an additive factorization over certain stepwise reward. In addition, we also develop a general  
 070 form to align the stepwise posterior  $\hat{p}_\theta(\mathbf{x}_0|\mathbf{x}_t)$  that works with arbitrary reward models, as opposed to  
 071 previous preference optimization approaches (Rafailov et al., 2023; Wallace et al., 2024) specifically  
 072 tailored under certain simplified reward such as the Bradley-Terry model (Bradley & Terry, 1952).

073 **Contributions.** To sum up, we propose stepwise decomposition preference optimization (SDPO)  
 074 for offline finetuning of discrete diffusion models, with the following detailed contributions. **1.** We  
 075 decompose the diffusion trajectory alignment problem into a set of subproblems that align the posterior  
 076  $\hat{p}_\theta(\mathbf{x}_0|\mathbf{x}_t)$  for each diffusion step, allowing for efficient and exact likelihood and reward evaluation.  
 077 **2.** We theoretically demonstrate the equivalence of SDPO and diffusion trajectory alignment through  
 078 the bridge of certain stepwise reward. **3.** We derive a general loss function that jointly optimizes the  
 079 stepwise alignment problems under arbitrary reward functions. **4.** We conduct extensive experimental  
 080 evaluations on three different tasks, namely DNA sequence design, protein inverse folding, and  
 081 language modeling. Our approach exhibits consistent enhancements, outperforming baselines by  
 082 a significant margin across all benchmarks. Notably, we obtain a remarkable 12% gain in terms  
 083 of predicted activity on the DNA sequence design, compared with the most competitive RL-based  
 084 method (Wang et al., 2024; Borsø et al., 2025) tailored for finetuning discrete diffusion models.  
 085 Moreover, we adopt our approach to LLaDA-8B-Instruct (Nie et al., 2025), which further enhances  
 086 GSM8K 5-shot score from 78.6 to 81.2, further demonstrating its promise as large language models.

## 087 2 RELATED WORK

088 **Discrete diffusion models.** Discrete diffusion models, originally formulated in Austin et al. (2021);  
 089 Campbell et al. (2022); Hoogeboom et al. (2022a) and further extended by Lou et al. (2023); Sahoo  
 090 et al. (2024); Shi et al. (2024); Zhao et al. (2024), have attracted growing interest in particular for  
 091 modeling sequence data. Different from autoregressive models (Brown et al., 2020; Achiam et al.,  
 092 2023; Team et al., 2023), discrete diffusion models relax from the inherent left-to-right causal ordering,  
 093 allowing for more flexible modeling and parallel decoding (Xu et al., 2025; Zheng et al., 2025).  
 094 They have achieved remarkable performance on various tasks, ranging from biological sequence  
 095 design (Wang et al., 2024; Campbell et al., 2024) to human natural language modeling (Arriola et al.,  
 096 2025; Nie et al., 2025; Ye et al., 2025; Zheng et al., 2023). Despite the promise, how to perform  
 097 preference optimization on pretrained discrete diffusion models to align with certain reward still  
 098 remains a challenge, which we aim to address in this work.

099 **Preference optimization for language models.** Aligning language models with certain reward is a  
 100 core problem to enhance their utility (Ouyang et al., 2022). Initial approaches under the paradigm of  
 101 RLHF (Ouyang et al., 2022; Christiano et al., 2017) that employ RL-based algorithms (Schulman et al.,  
 102 2017) for alignment have been proposed and successfully adopted. Direct preference optimization  
 103 (DPO) (Rafailov et al., 2023) and subsequent works (Ethayarajh et al., 2024; Meng et al., 2024; Han  
 104 et al., 2024a; Ji et al., 2024; Lai et al., 2024) leverage pairwise or ranking-based preference dataset to  
 105 perform offline optimization that further address the optimization instability and complexity. Whilst  
 106 much progress have been made, they are developed upon autoregressive language models, while we  
 107 instead focus on discrete diffusion models with a substantially different probabilistic factorization.

108 **Diffusion alignment.** Preference optimization has also been explored for diffusion models. The  
 109 pioneer attempt of Wallace et al. (2024); Yang et al. (2024) extend DPO to Gaussian diffusion and  
 110 is able to promote image quality. Li et al. (2024); Gu et al. (2024) further improve the performance  
 111 by employing different human preference modeling while Zhu et al. (2025b) proposes to align the  
 112 score function. There are also works that resort to RL (Fan et al., 2023; Black et al., 2024) or  
 113 directly backpropagating through differentiable reward (Clark et al., 2024; Prabhudesai et al., 2024).  
 114 Differently, we develop a principled objective for discrete diffusion which pose unique challenges due  
 115 to the discrete nature. Wang et al. (2024) approaches this problem through RL by backpropagating the  
 116 gradient via the Gumbel trick, which leads to optimization overhead. Recent works also derive under  
 117 pairwise preference based on DPO (Borsig et al., 2025; Zhu et al., 2025a), resort to online sampling  
 118 and verification (Zhao et al., 2025; Yang et al., 2025), or perform optimization through posterior  
 119 matching (Rector-Brooks et al., 2025). **Besides these training-based approaches, inference-time**  
 120 **guidance has also been explored for discrete diffusion to align sampling distributions.** Nisonoff et al.  
 121 (2024) adapts classifier-guidance (Ho & Salimans, 2022) to discrete diffusion, while sequential Monte  
 122 Carlo (SMC)-based approaches (Wu et al., 2023; Phillips et al., 2024; Dou & Song, 2024) have also  
 123 been introduced for more effective guidance. However, these guidance-based methods usually induce  
 124 much higher sampling cost and easily suffer from suboptimal performance when the guidance signal  
 125 is insufficient. Critically, our approach instead offers a generalized optimization objective, does not  
 126 require online sampling at each iteration, and demonstrates enhanced performance.

### 3 PRELIMINARIES

127 **Discrete diffusion models.** Discrete diffusion models (Austin et al., 2021; Lou et al., 2023; Sahoo  
 128 et al., 2024; Shi et al., 2024) are a family of diffusion models with the latent variables residing in  
 129 the discrete space  $\mathcal{X}$  with dimensionality  $m$ . With input data point  $\mathbf{x}_0$ , discrete diffusion features a  
 130 forward diffusion process in the form of Markov chain  $q(\mathbf{x}_t|\mathbf{x}_0)$  with

$$q(\mathbf{x}_t|\mathbf{x}_0) := \text{Cat}(\mathbf{x}_t; \alpha_t \mathbf{x}_0 + (1 - \alpha_t) \boldsymbol{\pi}), \quad (1)$$

131 where  $\boldsymbol{\pi}$  is the vectorized representation of certain prior distribution  $\text{Cat}(\cdot; \boldsymbol{\pi})$ , and  $\alpha_t$ , usually  
 132 referred to as the noise schedule, is a decreasing function w.r.t.  $t$  satisfying that  $\alpha_0 = 1$  and  
 133  $\alpha_T = 0$ . The transition for any two timesteps  $0 \leq s \leq t \leq T$  that induces  $q(\mathbf{x}_t|\mathbf{x}_0)$  is specified as  
 134  $q(\mathbf{x}_t|\mathbf{x}_s) = \text{Cat}(\mathbf{x}_t; \alpha_{t|s} \mathbf{x}_s + (1 - \alpha_{t|s}) \boldsymbol{\pi})$  where  $\alpha_{t|s} = \alpha_t / \alpha_s$ .

135 **Masked discrete diffusion models.** Masked discrete diffusion models (Sahoo et al., 2024; Shi et al.,  
 136 2024; Lou et al., 2023; Austin et al., 2021) are discrete diffusion models when the prior  $\boldsymbol{\pi}$  is in  
 137 particular instantiated as the absorbing state  $\mathbf{m} := [0, \dots, 0, 1]$  where the last entry in  $\mathbf{m}$  corresponds  
 138 to a special MASK token. The posterior has a simplified form (Sahoo et al., 2024; Shi et al., 2024):

$$q(\mathbf{x}_s|\mathbf{x}_t, \mathbf{x}_0) = \begin{cases} \text{Cat}(\mathbf{x}_s; \mathbf{x}_t) & \mathbf{x}_t \neq \mathbf{m}, \\ \text{Cat}(\mathbf{x}_s; \frac{\alpha_s - \alpha_t}{1 - \alpha_t} \mathbf{x}_0 + \frac{1 - \alpha_s}{1 - \alpha_t} \mathbf{m}) & \mathbf{x}_t = \mathbf{m}. \end{cases} \quad (2)$$

139 The reversal  $p_\theta(\mathbf{x}_s|\mathbf{x}_t)$  is then parameterized by a neural network  $\mathbf{f}_\theta(\mathbf{x}_t, t)$  that predicts  $\mathbf{x}_0$  in Eq. 2,  
 140 which is optimized to approximate the posterior by minimizing the negative evidence lower bound  
 141  $-\log p(\mathbf{x}_0) \leq \mathcal{L}_{\text{NELBO}} := \mathbb{E}_{q(\mathbf{x}_t|\mathbf{x}_0)} \sum_{t=1}^{T-1} \frac{\alpha_t - \alpha_{t-1}}{1 - \alpha_t} \log (\mathbf{x}_0^\top \cdot \mathbf{f}_\theta(\mathbf{x}_t, t))$ .  
 142

143 **Reinforcement learning with human feedback.** At alignment stage, a pretrained model  $p_\theta(\mathbf{x}|\mathbf{c})$   
 144 is finetuned to maximize certain reward  $r(\mathbf{x}, \mathbf{c})$  subject to a Kullback–Leibler (KL) divergence  
 145 regularization w.r.t. the reference model  $p_{\text{ref}}(\mathbf{x}|\mathbf{c})$ , leading to the following objective:

$$\max_{p_\theta} \mathbb{E}_{\mathbf{x}, \mathbf{c}} [r(\mathbf{x}, \mathbf{c})] - \beta D_{\text{KL}} [p_\theta(\mathbf{x}|\mathbf{c}) \| p_{\text{ref}}(\mathbf{x}|\mathbf{c})], \quad (3)$$

146 where  $\mathbf{c}$  is some context such as a prompt and  $\beta$  is the balancing factor. The choice of the reward  
 147 model can be arbitrary, such as human or LLM-assisted preference labels (Ouyang et al., 2022;  
 148 Rafailov et al., 2023), or the predicted activity of the designed DNA sequence (Wang et al., 2024).  
 149 This KL-constrained optimization problem has the optimal solution (Peters & Schaal, 2007)

$$p^*(\mathbf{x}|\mathbf{c}) = \frac{1}{Z(\mathbf{c})} p_{\text{ref}}(\mathbf{x}|\mathbf{c}) \exp \left( \frac{1}{\beta} r(\mathbf{x}, \mathbf{c}) \right), \quad (4)$$

150 where  $Z(\mathbf{c}) = \sum_{\mathbf{x}} p_{\text{ref}}(\mathbf{x}|\mathbf{c}) \exp \left( \frac{1}{\beta} r(\mathbf{x}, \mathbf{c}) \right)$  is the partition function that is intractable to evaluate.

162 **Problem formulation.** In this work, we aim to develop an efficient offline alignment approach for  
 163 discrete diffusion models. Specifically, the algorithm directly operates on a pre-collected dataset  
 164  $\mathcal{D} = \{(\mathbf{x}_0, \mathbf{c}, r(\mathbf{x}_0, \mathbf{c}))\}$  on clean data  $\mathbf{x}_0$  without relying on on-policy generations during finetuning.  
 165

## 166 4 METHOD

168 In this section, we detail our approach for aligning discrete diffusion models through stepwise  
 169 optimization. In § 4.1, we first revisit the problem of discrete diffusion alignment and investigate  
 170 the challenges. In § 4.2, we propose a novel stepwise decomposition approach for discrete diffusion  
 171 alignment. In § 4.3, we introduce a principled way to solve the stepwise alignment objective through  
 172 distribution matching. We offer additional in-depth analyses and discussions in § 4.4.  
 173

### 174 4.1 ALIGNING DISCRETE DIFFUSION MODELS

176 Different from autoregressive models that can evaluate  $p_\theta(\mathbf{x}|\mathbf{c})$  efficiently in a single forward pass,  
 177 discrete diffusion models are grounded on a chain of random variables  $\mathbf{x}_{0:T} := [\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_T]$ ,  
 178 where the joint satisfies the Markovian factorization  $p_\theta(\mathbf{x}_{0:T}|\mathbf{c}) = p_\theta(\mathbf{x}_T|\mathbf{c}) \prod_{t=1}^T p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{c})$ .  
 179 The alignment objective in Eq. 3 is therefore extended to the entire chain (Wallace et al., 2024):

$$180 \max_{p_\theta} \mathbb{E}_{p_\theta(\mathbf{x}_{0:T}|\mathbf{c}), \mathbf{c}} [\hat{r}(\mathbf{x}_{0:T}, \mathbf{c})] - \beta D_{\text{KL}} [p_\theta(\mathbf{x}_{0:T}|\mathbf{c}) \| p_{\text{ref}}(\mathbf{x}_{0:T}|\mathbf{c})], \quad (5)$$

182 where the reward  $\hat{r}(\mathbf{x}_{0:T}, \mathbf{c})$  now considers the whole chain  $\mathbf{x}_{0:T}$ . We hence refer to the optimization  
 183 problem of Eq. 5 as *diffusion trajectory optimization*. Akin to Eq. 4, the optimal solution is

$$184 \quad 185 \quad p^*(\mathbf{x}_{0:T}|\mathbf{c}) = \frac{1}{Z(\mathbf{c})} p_{\text{ref}}(\mathbf{x}_{0:T}|\mathbf{c}) \exp \left( \frac{1}{\beta} \hat{r}(\mathbf{x}_{0:T}, \mathbf{c}) \right). \quad (6)$$

186 However, the optimization problem in Eq. 5 poses several challenges. First, the expectation is taken  
 187 over the entire chain  $p_\theta(\mathbf{x}_{0:T})$ , making it computationally expensive to estimate. Moreover, the  
 188 definition of the reward  $\hat{r}(\mathbf{x}_{0:T}, \mathbf{c})$  requires reconsideration as it is supposed to operate on the entire  
 189 chain, while empirical rewards  $r(\mathbf{x}_0, \mathbf{c})$ , e.g., human preference (Rafailov et al., 2023) or DNA  
 190 activity (Wang et al., 2024), are most commonly defined on the clean sequence  $\mathbf{x}_0$ . We will introduce  
 191 our stepwise decomposition approach that offers a simplified and tractable measure to solve Eq. 5.  
 192

### 193 4.2 DIFFUSION TRAJECTORY OPTIMIZATION THROUGH STEPWISE DECOMPOSITION

195 We propose a principled way to solve the problem by decomposing the trajectory optimization into a  
 196 set of subproblems, each of which corresponds to a *stepwise* alignment objective for the **factorized**  
 197 **approximation of the posterior**  $\hat{p}_\theta(\mathbf{x}_0|\mathbf{x}_t, \mathbf{c}) := \prod_{i=1}^L \hat{p}_\theta(\mathbf{x}_0^{(i)}|\mathbf{x}_t, \mathbf{c})$  (Shi et al., 2024; Austin et al.,  
 198 2021) at diffusion step  $1 \leq t \leq T$  (see Fig. 1), where  $i$  is the token index and  $L$  is the sequence  
 199 length. To be specific, the set of subproblems is

$$200 \quad 201 \quad \max_{\hat{p}_\theta} \mathbb{E}_{\hat{p}_\theta(\mathbf{x}_0|\mathbf{x}_t, \mathbf{c}), \mathbf{c}} [r(\mathbf{x}_0, \mathbf{c})] - \beta_t D_{\text{KL}} [\hat{p}_\theta(\mathbf{x}_0|\mathbf{x}_t, \mathbf{c}) \| \hat{p}_{\text{ref}}(\mathbf{x}_0|\mathbf{x}_t, \mathbf{c})], \quad \forall 1 \leq t \leq T, \quad (7)$$

202 where  $\beta_t = \beta/w(t)$  is the stepwise regularization  
 203 reweighted by certain scheduler  $w(t)$ . The optimal  
 204 solutions can be similarly derived as  $\hat{p}^*(\mathbf{x}_0|\mathbf{x}_t, \mathbf{c}) =$   
 205  $\frac{1}{Z(\mathbf{c})} \hat{p}_{\text{ref}}(\mathbf{x}_0|\mathbf{x}_t, \mathbf{c}) \exp \left( \frac{1}{\beta_t} r(\mathbf{x}_0, \mathbf{c}) \right)$  for any  $t$ .

207 Such a formulation enjoys several unique benefits  
 208 compared with the trajectory alignment objective in  
 209 Eq. 5. First, the expectation over the entire chain  
 210  $p_\theta(\mathbf{x}_{0:T}|\mathbf{c})$  has been decomposed into the stepwise  
 211 posterior  $\hat{p}_\theta(\mathbf{x}_0|\mathbf{x}_t, \mathbf{c})$ , which can be computed both  
 212 tractably and efficiently for discrete diffusion models.  
 213 Furthermore, by grounding on the clean data  $\mathbf{x}_0$  in-  
 214 stead of intermediate latent variables  $\mathbf{x}_t$ , we can readily reuse the reward model  $r(\mathbf{x}_0, \mathbf{c})$  without  
 215 resorting to its biased estimates (Lu et al., 2023; Chen et al., 2024). More interestingly, we reveal a  
 critical connection between the stepwise decomposition alignment objective (Eq. 7) and the trajectory  
 optimization objective (Eq. 5), as stated in the theorem below:

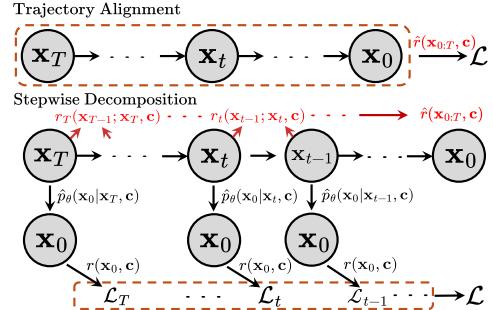


Figure 1: The flowchart of our SDPO.

216 **Theorem 4.1.** The joint  $p^*(\mathbf{x}_{0:T}|\mathbf{c})$  induced by the optimal solutions  $\{\hat{p}^*(\mathbf{x}_0|\mathbf{x}_t, \mathbf{c})\}_{t=1}^T$  of Eq. 7  
 217 is also the optimal solution of the trajectory alignment objective in Eq. 5, with the chain reward  
 218

$$219 \hat{r}(\mathbf{x}_{0:T}, \mathbf{c}) = \beta \sum_{t=1}^T r_t(\mathbf{x}_{t-1}; \mathbf{x}_t, \mathbf{c}) \text{ where } r_t(\mathbf{x}_{t-1}; \mathbf{x}_t, \mathbf{c}) = \log \frac{\mathbb{E}_{p'_{\text{ref}}(\mathbf{x}_0|\mathbf{x}_{t-1}, \mathbf{x}_t, \mathbf{c})} [\exp(\frac{1}{\beta_t} r(\mathbf{x}_0, \mathbf{c}))]}{\mathbb{E}_{p_{\text{ref}}(\mathbf{x}_0|\mathbf{x}_t, \mathbf{c})} [\exp(\frac{1}{\beta_t} r(\mathbf{x}_0, \mathbf{c}))]}.$$

220  
 221 Proof is in Appendix A.1. Here  $p'_{\text{ref}}(\mathbf{x}_0|\mathbf{x}_{t-1}, \mathbf{x}_t, \mathbf{c}) := \frac{\hat{p}_{\text{ref}}(\mathbf{x}_0|\mathbf{x}_t)q(\mathbf{x}_{t-1}|\mathbf{x}_0, \mathbf{x}_t)}{p_{\text{ref}}(\mathbf{x}_{t-1}|\mathbf{x}_t)}$  is the posterior of  $\mathbf{x}_0$   
 222 w.r.t. a specific choice of  $\mathbf{x}_{t-1}$ , given  $\mathbf{x}_t$ . In the case of masked diffusion models, the posterior refers  
 223 to the factorized conditional  $\hat{p}_{\text{ref}}(\mathbf{x}_0|\mathbf{x}_t, \mathbf{c})$  constrained on the set of all possible  $\mathbf{x}_0$  that share the  
 224 same decoded tokens with  $\mathbf{x}_{t-1}$ . Theorem 4.1 endorses our key finding that the intractable trajectory  
 225 optimization can be alternatively approached by jointly optimizing the *stepwise* alignment objectives,  
 226 under which the reward of the chain  $\hat{r}(\mathbf{x}_{0:T}, \mathbf{c})$  is effectively an additive factorization of the *stepwise*  
 227 *reward*  $r_t(\mathbf{x}_{t-1}; \mathbf{x}_t, \mathbf{c})$ . More interestingly, the stepwise reward also has intuitive implications. At  
 228 each diffusion step  $t$  with the sampled  $\mathbf{x}_t$ , the denominator inside log is a constant and  $r_t$  is therefore  
 229 distinguished fully by the numerator, a term that effectively assigns higher reward to those  $\mathbf{x}_{t-1}$  who  
 230 are more likely to be obtained from the  $\mathbf{x}_0$  with higher reward  $r(\mathbf{x}_0, \mathbf{c})$ . Furthermore, the stepwise  
 231 rewards also serve as more fine-grained supervision that enables tractable alignment of each diffusion  
 232 step, while previous works that operate fully on the trajectory-level confer no per-step guarantee.  
 233

### 234 4.3 GENERALIZED STEPWISE ALIGNMENT THROUGH DISTRIBUTION MATCHING

235 While the stepwise decomposition has introduced clear benefits, it is still yet unclear how to optimize  
 236  $\hat{p}_\theta(\mathbf{x}_0|\mathbf{x}_t, \mathbf{c})$  towards the optimal solution  $\hat{p}^*(\mathbf{x}_0|\mathbf{x}_t, \mathbf{c})$ , particularly under arbitrary reward  $r(\mathbf{x}_0, \mathbf{c})$ .  
 237 To this end, existing works seek to directly backpropagate the gradient from the reward model (Wang  
 238 et al., 2024), which inevitably incurs optimization overhead and instability, or to simplify the reward  
 239 into tractable forms such as the Bradley-Terry model (Wallace et al., 2024), which imposes additional  
 240 constraints. Differently, we propose to perform optimization based on the following objective:

$$241 \mathcal{L}_t(\theta) := \mathbb{E}_{\mathbf{x}_t, \mathbf{c}} [D_{\text{KL}} [\tilde{p}_r(\mathbf{x}_0|\mathbf{x}_t, \mathbf{c}) \| \tilde{p}_\theta(\mathbf{x}_0|\mathbf{x}_t, \mathbf{c})]], \quad (8)$$

242 where  $\tilde{p}_r(\mathbf{x}_0|\mathbf{x}_t, \mathbf{c}) \propto \hat{p}_{\text{ref}}(\mathbf{x}_0|\mathbf{x}_t, \mathbf{c}) \exp(r(\mathbf{x}_0, \mathbf{c}))$  is the Boltzmann policy (Laidlaw & Dragan,  
 243 2022; Peters & Schaal, 2007) induced by the reward then reweighted by  $p_{\text{ref}}$ , while  $\tilde{p}_\theta(\mathbf{x}_0|\mathbf{x}_t, \mathbf{c}) \propto$   
 244  $\hat{p}_{\text{ref}}(\mathbf{x}_0|\mathbf{x}_t, \mathbf{c})^{(1-\beta_t)} \hat{p}_\theta(\mathbf{x}_0|\mathbf{x}_t, \mathbf{c})^{\beta_t}$  is similarly the reweighted model policy. The rationale of Eq. 8  
 245 lies in that the minimizer of this KL-divergence distribution matching (Han et al., 2024a; Ji et al.,  
 246 2024) problem is also the optimal solution of stepwise alignment (proof in Appendix. A.2):

247 **Proposition 4.2.** Let  $\theta^* = \arg \min \mathcal{L}_t(\theta)$  defined in Eq. 8. Then  $\hat{p}_\theta^*(\mathbf{x}_0|\mathbf{x}_t, \mathbf{c}) = \hat{p}^*(\mathbf{x}_0|\mathbf{x}_t, \mathbf{c})$ , the  
 248 optimal solution of the stepwise alignment objective in Eq. 7.

249 Besides the guaranteed equivalence of the optimal solution, the definition of  $\tilde{p}_r$  also enables impor-  
 250 tance sampling by using  $p_{\text{ref}}$  as the proposal distribution, from which the offline preference datasets  
 251 are drawn. Expanding Eq. 8 with importance sampling (see Appendix. A.3), we have

$$253 \mathcal{L}_t(\theta) = -\mathbb{E}_{\mathbf{c}, \hat{p}_{\text{ref}}(\mathbf{x}_0, \mathbf{x}_t|\mathbf{c})} \left[ \frac{\exp(r(\mathbf{x}_0, \mathbf{c}))}{Z_r(\mathbf{c})} \log \frac{\exp(r_\theta(\mathbf{x}_0, \mathbf{x}_t, \mathbf{c}, \beta_t))}{Z_\theta^t(\mathbf{x}_t, \mathbf{c}, \beta_t)} \right] + C, \quad (9)$$

254 where  $r_\theta(\mathbf{x}_0, \mathbf{x}_t, \mathbf{c}, \beta_t) = \beta_t (\log \hat{p}_\theta(\mathbf{x}_0|\mathbf{x}_t, \mathbf{c}) - \log \hat{p}_{\text{ref}}(\mathbf{x}_0|\mathbf{x}_t, \mathbf{c}))$  refers to the implicit re-  
 255 ward (Rafailov et al., 2023; Cui et al., 2025),  $Z_r(\mathbf{c}) = \mathbb{E}_{p_{\text{ref}}(\mathbf{x}_0|\mathbf{c})} \exp(r(\mathbf{x}_0, \mathbf{c}))$  and  $Z_\theta^t(\mathbf{x}_t, \mathbf{c}, \beta_t) =$   
 256  $\mathbb{E}_{\hat{p}_{\text{ref}}(\mathbf{x}_0|\mathbf{x}_t, \mathbf{c})} \exp(r_\theta(\mathbf{x}_0, \mathbf{x}_t^{(i)}, \mathbf{c}, \beta_t))$  are the partition functions, and  $C$  is a constant irrelevant to  $\theta$ .

257 **Empirical form.** We leverage Monte-Carlo to estimate  $\mathcal{L}_t$  as well as the partitions using  $N$  samples  
 258  $\{(\mathbf{x}_0^{(i)}, \mathbf{x}_t^{(i)}, \mathbf{c})\}_{i=1}^N$  drawn from  $p_{\text{ref}}(\mathbf{x}_0, \mathbf{x}_t|\mathbf{c})$  for each  $\mathbf{c}$ . In form, we employ

$$259 \tilde{\mathcal{L}}_t^N(\theta) = -\mathbb{E}_{\mathbf{c}} \sum_{i=1}^N \left( \frac{\exp(r(\mathbf{x}_0^{(i)}, \mathbf{c}))}{\sum_{j=1}^N \exp(r(\mathbf{x}_0^{(j)}, \mathbf{c}))} \cdot \log \frac{\exp(\tilde{r}_\theta(\mathbf{x}_0^{(i)}, \mathbf{x}_t^{(i)}, \mathbf{c}, \beta_t))}{\sum_{j=1}^N \exp(\tilde{r}_\theta(\mathbf{x}_0^{(j)}, \mathbf{x}_t^{(j)}, \mathbf{c}, \beta_t))} \right). \quad (10)$$

260 Eq. 10 takes the form of cross-entropy loss (Ji et al., 2024; Lu et al., 2023) between the self-  
 261 normalized Boltzmann policies induced by  $r(\mathbf{x}_0, \mathbf{c})$  and  $\tilde{r}_\theta(\mathbf{x}_0, \mathbf{x}_t, \mathbf{c}, \beta_t)$ . As  $N \rightarrow \infty$ , the estimate  
 262 for the policy of  $r$  becomes unbiased, while an unbiased estimate of  $Z_\theta^t$  requires extensive sampling  
 263 from the posterior  $\hat{p}_{\text{ref}}(\mathbf{x}_0|\mathbf{x}_t, \mathbf{c})$  for each  $\mathbf{x}_t$ , which is highly prohibitive in the *offline* alignment  
 264 setup. In practice we still favor the simplified MC estimate in Eq. 10 which is efficient and performant.  
 265 We henceforth employ  $\tilde{\mathcal{L}}_t^N(\theta)$  to solve each subproblem of Eq. 7. For the sample size  $N$ , we view it  
 266 as a hyperparameter that trades off between efficiency and bias, depending on the task and dataset.

270 4.4 OVERALL OBJECTIVE  
271

272 Since the final objective (Eq. 7) requires to jointly optimize for the subproblems across all diffusion  
273 steps, at each iteration we randomly select a batch of diffusion steps, and optimize the corresponding  
274  $\mathcal{L}_t^N$  as per Eq. 10. Furthermore, since in offline settings the intermediate samples  $\mathbf{x}_t$  are not preserved,  
275 we instead keep track of the clean samples  $\mathbf{x}_0$  obtained from  $p_{\text{ref}}$  while approaching the corresponding  
276  $\mathbf{x}_t$  via the forward process  $q(\mathbf{x}_t|\mathbf{x}_0)$  at each training step. Putting all together we obtain our final loss

$$277 \mathcal{L}(\theta) = -\mathbb{E}_{t, \mathbf{c}, \mathbf{x}_0, q(\mathbf{x}_t|\mathbf{x}_0)} \sum_{i=1}^N \left( \frac{\exp(r(\mathbf{x}_0^{(i)}, \mathbf{c}))}{\sum_{j=1}^N \exp(r(\mathbf{x}_0^{(j)}, \mathbf{c}))} \cdot \log \frac{\exp(\tilde{r}_\theta(\mathbf{x}_0^{(i)}, \mathbf{x}_t^{(i)}, \mathbf{c}, \beta_t))}{\sum_{j=1}^N \exp(\tilde{r}_\theta(\mathbf{x}_0^{(j)}, \mathbf{x}_t^{(j)}, \mathbf{c}, \beta_t))} \right), \\ 278 \quad 279 \quad 280 \quad 281 \quad 282 \quad 283 \quad 284 \quad 285 \quad 286 \quad 287 \quad 288 \quad 289 \quad 290 \quad 291 \quad 292 \quad 293 \quad 294 \quad 295 \quad 296 \quad 297 \quad 298 \quad 299 \quad 300 \quad 301 \quad 302 \quad 303 \quad 304 \quad 305 \quad 306 \quad 307 \quad 308 \quad 309 \quad 310 \quad 311 \quad 312 \quad 313 \quad 314 \quad 315 \quad 316 \quad 317 \quad 318 \quad 319 \quad 320 \quad 321 \quad 322 \quad 323$$

where  $\tilde{r}_\theta$ , by further leveraging the reversal **factorized parameterization** of masked diffusion models (Shi et al., 2024; Sahoo et al., 2024) and the definition  $\beta_t = \beta/w(t)$ , takes the following form:

$$\tilde{r}_\theta(\mathbf{x}_0, \mathbf{x}_t, \mathbf{c}, \beta_t) = \beta \left( \frac{\log(\mathbf{x}_0^\top \mathbf{f}_\theta(\mathbf{x}_t, t, \mathbf{c}))}{w(t)} - \frac{\log(\mathbf{x}_0^\top \mathbf{f}_{\text{ref}}(\mathbf{x}_t, t, \mathbf{c}))}{w(t)} \right). \quad (12)$$

We note that our method applies to general discrete diffusion, but we choose to focus specifically on the masked variant. Our final loss has several implications, which we will analyze below.

**Pairwise preference data.** Our loss possesses a generalized form *w.r.t.* the reward model  $r(\mathbf{x}_0, \mathbf{c})$  and  $N$ , *i.e.*, the number of samples for each context or prompt  $\mathbf{c}$ . In particular, it subsumes the setting in DPO where each prompt is provided with a pair of winning and losing completions  $(\mathbf{x}_0^{(w)}, \mathbf{x}_0^{(l)})$ , by setting  $N = 2$  and leveraging Bradley-Terry (BT) model as the reward, *i.e.*,  $r(\mathbf{x}_0^{(w)}, \mathbf{c}) = 0$  and  $r(\mathbf{x}_0^{(l)}, \mathbf{c}) = -\infty$ . We provide detailed derivations of our loss in this special case in Appendix A.4.

**The role of  $w(t)$ .** The coefficient  $w(t)$  is initially introduced as the weight for the per-step reward  $\hat{r}_t$ . Interestingly, from Eq. 12 we can also interpret  $w(t)$  as a factor that controls the scale of  $\log(\mathbf{x}_0^\top \mathbf{f}(\mathbf{x}_t, t))$ , which is correlated to the number of masked tokens at step  $t$ . Therefore we set  $w(t) = 1 - \alpha_t$  to amortize the loss to each token, and empirically find this choice effective.

**The role of  $\beta$ .** Eq. 5 reveals that  $\beta$  controls the strength of the KL regularization *w.r.t.* the reference distribution, which is also widely reflected in literature (Rafailov et al., 2023; Wallace et al., 2024).

**Iterative labeling.** Empirically we have also explored a variant of our approach that updates the dataset with samples from the latest model and their corresponding rewards. We find this iterative labeling generally favorable since more useful rewards are progressively provided for samples of higher quality, as the training proceeds. We defer detailed justifications to § 5.4.

## 5 EXPERIMENTS

In this section, we perform empirical investigations of our approach on a wide suite of tasks and benchmarks, including DNA sequence design (§ 5.1), protein inverse folding (§ 5.2), and language modeling (§. 5.3). We provide ablation studies in § 5.4.

## 5.1 DNA SEQUENCE DESIGN

We aim to finetune our model to unconditionally generate DNA sequences that trigger gene expression in targeted cell types. This is a task commonly seen in cell and gene therapy (Taskiran et al., 2024).

**Experiment setup.** We use a publicly available dataset (Gosai et al., 2023) that contains the measured enhancer activity in  $\sim 700k$  DNA sequences, each 200 base-pairs in length. Cell line activity is measured for each sequence, quantified with massively parallel reporter assays (MPRAs) that record the expression each sequence drives. The pre-trained masked diffusion language model (Sahoo et al., 2024) is taken from Wang et al. (2024), trained on the entire enhancer dataset. The pre-trained finetuning and evaluation reward models predict the HepG2 cell line activity in a sequence, also taken from Wang et al. (2024) and trained on different splits of the dataset.

**Baselines.** We compare with the following baselines. *Pretrained*: the base pre-trained model (no finetuning). *Guidance methods*: classifier guidance (CG) (Nisonoff et al., 2024), classifier-free

324  
 325 Table 1: Model performance on DNA sequence design. Our approach generates sequences with  
 326 high activity measured by *Pred-Activity* and *ATAC-Acc*, while being natural-like by high 3-mer  
 327 and JASPAR correlations and likelihood. Results averaged across 3 random seeds with standard  
 328 deviations in parentheses. Numbers of baselines are taken from [Wang et al. \(2024\)](#).

	Pred-Activity (med) $\uparrow$	ATAC-Acc $\uparrow$ (%)	3-mer Corr $\uparrow$	JASPAR Corr $\uparrow$	App-Log-Lik (med) $\uparrow$	Entropy (med) $\uparrow$
Pretrained (Sahoo et al., 2024)	0.17 (0.04)	1.5 (0.2)	-0.061 (0.034)	0.249 (0.015)	-261 (0.6)	<b>390</b> (6.2)
CG (Nisonoff et al., 2024)	3.30 (0.00)	0.0 (0.0)	-0.065 (0.001)	0.212 (0.035)	-266 (0.6)	12 (4.1)
SMC (Wu et al., 2023)	4.15 (0.33)	39.9 (8.7)	0.840 (0.045)	0.756 (0.068)	-259 (2.5)	351 (6.5)
TDS (Wu et al., 2023)	4.64 (0.21)	45.3 (16.4)	0.848 (0.008)	0.846 (0.044)	-257 (1.5)	340 (5.4)
CFG (Ho & Salimans, 2022)	5.04 (0.06)	92.1 (0.9)	0.746 (0.001)	0.864 (0.011)	-265 (0.6)	363 (6.1)
D2-DPO (Borsø et al., 2025)	2.97 (0.03)	35.6 (0.9)	<b>0.944</b> (0.002)	0.883 (0.005)	-252 (0.4)	362 (4.9)
VRPO (Zhu et al., 2025a)	4.60 (0.01)	15.8 (0.2)	0.838 (0.002)	0.865 (0.005)	-255 (0.8)	289 (13.5)
<b>DDPP-IS</b> (Rector-Brooks et al., 2025)	<b>4.07 (0.02)</b>	<b>50.0 (0.3)</b>	<b>0.711 (0.001)</b>	<b>0.723 (0.004)</b>	<b>-253 (0.9)</b>	<b>378 (5.8)</b>
DRAKES (Wang et al., 2024)	5.61 (0.07)	92.5 (0.6)	0.887 (0.002)	0.911 (0.002)	-264 (0.6)	<b>375 (5.2)</b>
<b>diffu-GRPO</b> (Zhao et al., 2025)	<b>5.86 (0.04)</b>	<b>33.0 (0.8)</b>	<b>0.783 (0.001)</b>	<b>0.903 (0.004)</b>	<b>-245 (0.4)</b>	<b>310 (8.6)</b>
<b>SDPO</b>	<b>6.30 (0.003)</b>	<b>94.8 (0.01)</b>	<b>0.900 (0.003)</b>	<b>0.936 (0.003)</b>	<b>-246 (0.5)</b>	365 (4.4)

337 guidance (CFG) (Ho & Salimans, 2022) and two Sequential Monte Carlo-based methods (Wu et al.,  
 338 2023), namely *SMC*, where the proposal is the pretrained model, and *TDS*, where the proposal is  
 339 *CG*. *D2-DPO* (Borsø et al., 2025) and *VRPO* (Zhu et al., 2025a): offline preference optimization  
 340 algorithms that adapt DPO to discrete diffusion. *DRAKES* (Wang et al., 2024): an online RL  
 341 algorithm that backpropagates the reward through the generated trajectory with Gumbel-Softmax.  
 342 *diffu-GRPO* (Zhao et al., 2025): a policy gradient-based approach for discrete diffusion. *DDPP-  
 343 IS* (Rector-Brooks et al., 2025): an importance sampling method to match the reward-tilted posterior.

344 **Metrics.** We use the metrics following the protocol in Wang et al. (2024) to evaluate the model’s  
 345 enhancer generation. **1. Pred-Activity.** The enhancer activity level in the HepG2 cell line is predicted  
 346 by the evaluation reward model, trained on a held out evaluation set. **2. ATAC-Acc.** We measure  
 347 the proportion of generated sequences with high chromatin accessibility. This metric is typically  
 348 correlated with the enhancer activity. **3. 3-mer Corr.** We compute the 3-mer Pearson correlation  
 349 between the generated sequences and the sequences from the enhancer dataset with the top 0.1%  
 350 HepG2 activity. More natural, in-distribution sequences tend to have higher 3-mer Pearson correlation  
 351 values. **4. JASPAR-Corr.** We compute potential transcription factor binding motifs in the generated  
 352 sequences with JASPAR transcription factor binding profiles (Castro-Mondragon et al., 2022), and  
 353 calculate the Spearman correlation of motif frequency between the generated samples and the top  
 354 0.1% sequences in the dataset with the highest activity. **5. App-Log-Lik.** The approximated log-  
 355 likelihood of the generated sequences is computed with respect to the pre-trained model using the  
 356 discrete diffusion ELBO presented in Sahoo et al. (2024). This metric evaluates the naturalness of the  
 357 generations, as samples that over-optimize for the reward model tend to have worse log-likelihoods. **6.**  
 358 *Entropy.* Sequence entropy is computed following Wang et al. (2024) to measure the sample diversity.

359 **Results.** Our method generates sequences that are both natural-like and have high predicted enhancer  
 360 activity. Notably, we are able to significantly outperform all previous baselines in the predicted HepG2  
 361 activity, while also achieving strong 3-mer Pearson and JASPAR correlation numbers, demonstrating  
 362 our method’s robustness to over-optimizing for the reward model. In particular, we outperform the  
 363 RL-based approach DRAKES by a significant margin of 12.3% in terms of predicted activity. The  
 364 ATAC accuracy, another metric correlated with HepG2 activity, provides further validation of the high  
 365 quality of our generated samples, as we see that other baselines, such as the SMC-based methods,  
 366 may achieve relatively higher predicted enhancer activity but suffer poor ATAC accuracy numbers.

367 **Training efficiency.** Besides the superior performance, another feature that worth highlighting for  
 368 SDPO is its training efficiency, since it does not require on-policy sampling at each training iteration.  
 369 We report the average wallclock time per training step, where DRAKES takes 6.02 sec, diffu-GRPO  
 370 takes 1.51 sec, and SDPO only takes 0.77 sec, which verifies the superior training efficiency of SDPO.

## 371 5.2 PROTEIN INVERSE FOLDING

372 For the protein inverse folding task, we finetune a pre-trained model that predicts the protein sequence  
 373 from a 3D structure. We aim to optimize the stability of the protein sequences.

375 **Experiment setup.** The pre-trained diffusion model uses the ProteinMPNN (Dauparas et al., 2022)  
 376 architecture and is trained using the methodology from (Campbell et al., 2024) on the PDB training  
 377 dataset from Dauparas et al. (2022). The finetuning and evaluation reward models are trained on  
 378 different splits of the Megасale (Tsuboyama et al., 2023) dataset. We take all checkpoints directly

378 Table 2: Model performance on inverse protein folding. Our approach generates protein sequences  
 379 with high stability and desired structure. Results averaged across 3 random seeds with standard  
 380 deviations in parentheses. Numbers of baselines are taken from [Wang et al. \(2024\)](#).

	Pred-ddG (med) $\uparrow$	% (ddG > 0) (%) $\uparrow$	scRMSD (med) $\downarrow$	% (scRMSD < 2) (%) $\uparrow$	Success Rate (%) $\uparrow$	Entropy (med) $\uparrow$
Pretrained (Campbell et al., 2024)	-0.544 (0.037)	36.6 (1.0)	0.849 (0.013)	90.9 (0.6)	34.4 (0.5)	35.2 (8.1)
CG (Nisonoff et al., 2024)	-0.561 (0.045)	36.9 (1.1)	0.839 (0.012)	90.9 (0.6)	34.7 (0.9)	34.6 (7.1)
SMC (Wu et al., 2023)	0.659 (0.044)	68.5 (3.1)	0.841 (0.006)	93.8 (0.4)	63.6 (4.0)	24.9 (6.9)
TDS (Wu et al., 2023)	0.674 (0.086)	68.2 (2.4)	0.834 (0.001)	<b>94.4</b> (1.2)	62.9 (2.8)	24.9 (7.2)
CFG (Ho & Salimans, 2022)	-1.186 (0.035)	11.0 (0.4)	3.146 (0.062)	29.4 (1.0)	1.3 (0.4)	8.4 (5.9)
D2-DPO (Borsig et al., 2025)	0.500 (0.051)	66.4 (0.3)	0.909 (0.005)	93.6 (0.8)	61.0 (0.5)	<b>41.7</b> (7.4)
VRPO (Zhu et al., 2025a)	0.548 (0.032)	61.1 (0.1)	0.883 (0.004)	93.5 (0.7)	56.6 (0.3)	39.1 (9.3)
<b>DDPD-IS</b> (Zhu et al., 2025a)	<b>-0.130 (0.047)</b>	<b>46.7 (0.8)</b>	<b>0.829 (0.008)</b>	<b>89.3 (0.6)</b>	<b>43.3 (0.5)</b>	<b>24.3 (7.6)</b>
DRAKES (Wang et al., 2024)	1.095 (0.026)	86.4 (0.2)	0.918 (0.006)	91.8 (0.5)	<b>78.6</b> (0.7)	33.3 (6.4)
<b>diffu-GRPO</b> (Zhao et al., 2025)	<b>1.286 (0.021)</b>	<b>76.8 (0.3)</b>	<b>1.192 (0.005)</b>	<b>57.1 (0.8)</b>	<b>37.2 (1.4)</b>	<b>40.0 (7.8)</b>
SDPO	<b>1.400 (0.014)</b>	<b>87.1 (0.01)</b>	0.938 (0.005)	88.9 (0.3)	<u>75.5</u> (0.3)	<b>42.3</b> (6.5)

389 from [Wang et al. \(2024\)](#). For finetuning our model, we use the curated Megascle training dataset  
 390 from [Wang et al. \(2024\)](#), which consists of  $\sim 500k$  sequences with stability measurements.

391 **Metrics.** We use the following metrics ([Wang et al., 2024](#)) to evaluate the stability and naturalness of  
 392 the generated protein sequences. **1. Pred-ddG.** The evaluation reward model predicts the ddG (change  
 393 in Gibbs free energy) of a sequence, which is a measure of the sequence’s stability. The finetuning  
 394 dataset does not overlap with the evaluation dataset, so the model does not train on proteins used for  
 395 evaluation. **2. scRMSD.** The self-consistency root mean square deviation (scRMSD) measures the  
 396 ability of a sequence to fold into the desired structure. We use the pre-trained ESMFold ([Lin et al.,](#)  
 397 [2023](#)) model to compute the RMSD between the sequence’s predicted 3D structure and the original  
 398 backbone structure. **3. Success rate.** We compute the success rate as the proportion of generated  
 399 sequences with Pred-ddG  $> 0$  and scRMSD  $< 2$ . **4. Entropy.** The sequence entropy is computed to  
 400 measure the sample diversity.

401 **Results.** Our method is able to generate sequences with high stability that still remain in-distribution.  
 402 We significantly outperform all baselines in the predicted ddG for stability, showing strong reward  
 403 optimization ability, while still producing natural-like samples with scRMSD values and overall  
 404 success rate comparable to the state-of-the-art *DRAKES* method. **The policy-gradient based method**  
 405 **diffu-GRPO exhibits significant reward over-optimization with severe drop in metrics like Success**  
 406 **Rate.** Notably, the inverse folding problem is particularly difficult due to lack of labeled data in the  
 407 curated Megascle dataset (only several hundred distinct 3D structure backbones). During evaluation,  
 408 the model conditions on new backbone configurations not seen during training. Thus, our method is  
 409 still able to generate high reward samples without over-optimizing in a limited-data setting.

### 410 5.3 LANGUAGE MODELING

412 Crucially, we also apply our approach to a large-scale discrete diffusion for natural language modeling,  
 413 demonstrating its efficacy towards preference optimization of large language diffusion models.

414 **Experiment setup.** We employ LLaDA-8B-Instruct ([Nie et al., 2025](#)), a large-scale instruction-  
 415 tuned chat model based on the masked diffusion framework, as the reference model. We use  
 416 UltraFeedback ([Cui et al., 2023](#)) dataset annotated by [Meng et al. \(2024\)](#) as the preference dataset,  
 417 and finetune the model on 8 Nvidia A100 GPUs. Detailed hyperparameters are deferred to Appendix.

418 **Benchmarks and metrics.** We compare our  
 419 finetuned model against the reference model on  
 420 three important language model benchmarks. **1.**  
 421 *GSM8K* ([Cobbe et al., 2021](#)), which benchmarks  
 422 the math and reasoning capability of the model on  
 423 graduate school math problems. The metric is the  
 424 average accuracy of the answers. **2.** *IFEval* ([Zhou](#)  
 425 et al., 2023), which measures the model’s capability  
 426 of following human natural language instruc-  
 427 tions. We report IFEval score, the average of prompt and instruction-level strict-accuracy. **3.**  
 428 *AlpacaEval 2.0* ([Li et al., 2023; Dubois et al., 2024](#)) that evaluates the chat response quality by  
 429 comparing against certain baseline model on a suite of prompts. The metrics on this benchmark are  
 430 the winrate (WR) and length-controlled (LC) winrate against GPT-4-Preview-1106.

431 **Results.** The benchmark results are presented in Table 3. By finetuning LLaDA-8B-Instruct using our  
 432 proposed SDPO, we observe a consistent and remarkable enhancement across all three benchmarks,

Table 3: Results on finetuning LLaDA-8B-Instruct using dataset from [Cui et al. \(2023\)](#).

	Instruct	D2-DPO	diffu-GRPO	SDPO
Alpaca- LC (%)	10.6	12.1	<b>12.6</b>	<b>14.2</b>
Eval 2.0 WR (%)	6.8	7.5	<b>7.8</b>	<b>8.7</b>
GSM8K	78.6	78.1	<b>80.5</b>	<b>81.2</b>
IFEval	52.9	53.8	<b>53.5</b>	<b>55.1</b>

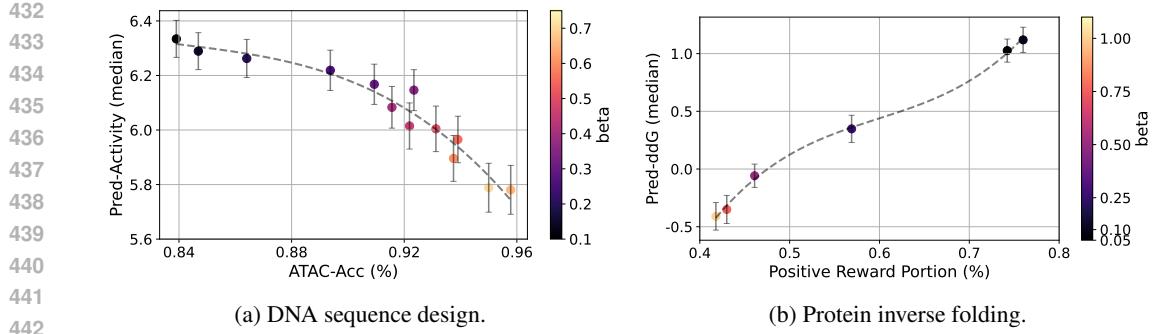


Figure 2: Ablation studies of  $\beta$  in (a) DNA design, and (b) protein inverse folding experiment.

which underscores the efficacy of SDPO towards promoting the capability of mathematical reasoning, instruction following, and chat quality of the discrete diffusion language model. Notably, our approach improves GSM8K score from 78.6 to 81.2, surpassing the score of LLaMA-3-8B post-trained with RL (c.f. Nie et al. (2025)). Furthermore, we obtain a relative improvement of 30.9% averaged across LC and WR on AlpacaEval 2.0 benchmark, demonstrating the applicability of SDPO for building helpful discrete diffusion-based chatbot. Our results on language modeling tasks open up new possibility towards building performant large language diffusion models through preference optimization.

#### 5.4 ABLATION STUDY

**The effect of  $\beta$ .** We study the effect of  $\beta$  in aligning models. As shown in Eq. 6, choosing a smaller  $\beta$  generally increases the weight of the reward function and tunes the model further away from the pretrained reference distribution. We verify this by two ablation studies on DNA sequence design and protein inverse folding, fixing all hyperparameters except  $\beta$ . Fig. 2 shows that a lower  $\beta$  value results in stronger reward guidance, resulting in greater *Pred-Activity* for DNA design, and greater *Pred-ddG* values for protein inverse folding. Conversely, a larger  $\beta$  poses more regularization to the model and thus the reward remains closer to the pretrained reference model. However, choosing too small a  $\beta$  may also steer the model too far away from the reference model and result in unnatural sequences. As shown in Fig. 2(a), the *ATAC-Acc* of the generated DNA sequences decreases as we over-optimizes *Pred-Activity* with a small  $\beta$ , despite their being positively correlated for natural DNA sequences.

**The effect of  $N$ .** We first investigate the effect of the sample size  $N$ . The results in protein inverse folding task without any iterative labeling are presented in Table 4. Notably, we observe that as the value of  $N$  gradually increases, we effectively reduces the variance in Monte-Carlo estimate performed by Eq. 11, which is further supported by the increasing trend in *Pred-ddG*. In particular, compared with  $N = 2$  which reflects the pairwise preference data setting adopted in DPO, leveraging a comparatively larger  $N$  is more beneficial. The performance plateaus as  $N$  further increases from 25 to 100, which is empirically not as favorable due to the memory overhead incurred.

**Iterative labeling.** In § 4.4 we additionally introduce a variant of our SDPO that leverages iterative labeling to enhance performance. Specifically, during training we iteratively generate 10,000 samples from the model and label them using the reward model in the DNA experiment. We then optimize the model on these labeled samples using the same objective. We demonstrate the advantage of such an approach in Fig. 3. Compared with the baseline that does not scale up the labeling on latest samples but always on samples from the original model, we observe consistent increment over 2 rounds of iterative labeling. In particular, the predicted DNA activity improves by a significant margin for SDPO with iterative labeling while the counterpart struggles in predicted

Table 4: Effect of sample size  $N$  in the inverse protein folding task.

$N$	<i>Pred-ddG</i>	Positive Reward Prop.
2	0.529	0.624
10	0.924	0.749
25	1.119	0.759
100	1.061	0.765

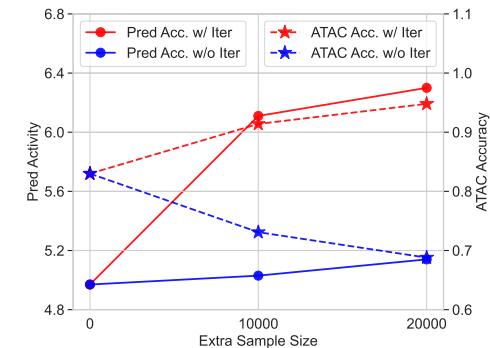
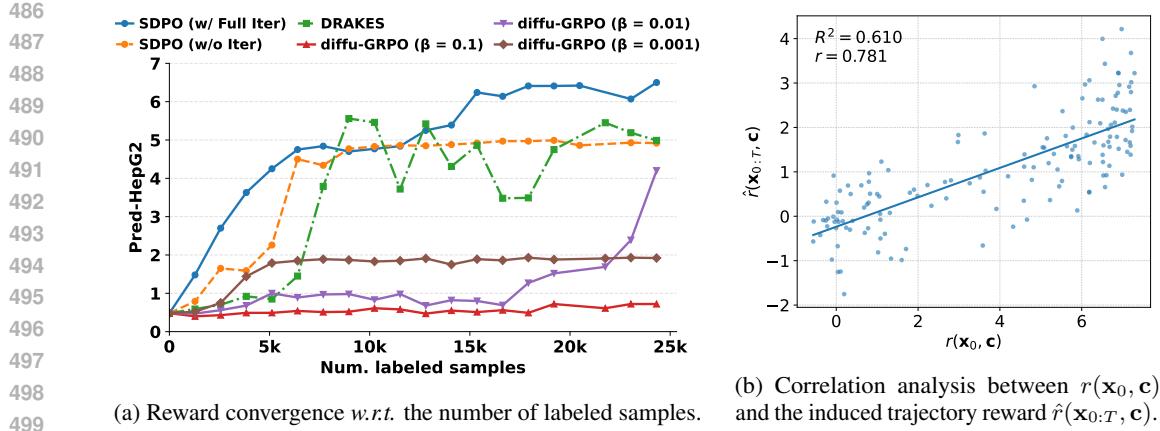


Figure 3: Ablation study of iterative labeling.

(a) Reward convergence *w.r.t.* the number of labeled samples.(b) Correlation analysis between  $r(\mathbf{x}_0, \mathbf{c})$  and the induced trajectory reward  $\hat{r}(\mathbf{x}_{0:T}, \mathbf{c})$ .Figure 4: (a) The reward curve *w.r.t.* the number of labeled samples throughout training. (b) The correlation analysis between the induced trajectory reward  $\hat{r}(\mathbf{x}_{0:T}, \mathbf{c})$  and the clean reward  $r(\mathbf{x}_0, \mathbf{c})$ .

activity while also encountering a drop in ATAC accuracy, possibly due to overfitting. Furthermore, our approach is also remarkably more labeling efficient compared with DRAKES that uses 128,000 additional labeling on the DNA task. The result implies that, on certain tasks when the reward model is available, performing SDPO in an iterative manner with reward labeling will lead to improved performance.

**Convergence rate comparison.** We also perform a systematic head-to-head comparison of the reward convergence speed in Fig. 4a, where we plot the reward curve *w.r.t.* the number of labeled samples throughout the training process. For a fair comparison with the on-policy baselines DRAKES and diffu-GRPO, we implement a variant of SDPO, *i.e.*, SDPO w/ Full Iter, that performs iterative labeling after *each* training step. Notably, our SDPO with full iterative labeling achieves 6.2 Pred-HepG2 using only 15k labeled samples, while DRAKES and diffu-GRPO only achieve 5.6 and 4.2 Pred-HepG2 with 25k labeled samples. SDPO without iterative labeling also exhibits fast convergence and high reward efficiency.

**Reward correlation analysis.** Here we provide more in-depth analysis on the DNA task regarding the relationship between the reward  $\hat{r}(\mathbf{x}_{0:T}, \mathbf{c})$  defined in Theorem 4.1 and the original reward  $r(\mathbf{x}_0, \mathbf{c})$ . Specifically, we sample 50 trajectories from the pretrained model, the model after first stage training, and the final model respectively, leading to 150 trajectories in total  $\{\mathbf{x}_{0:T}^{(i)}\}_{i=1}^{150}$ . For each trajectory  $\mathbf{x}_{0:T}^{(i)}$ , we evaluate its chain reward  $\hat{r}^{(i)} = \hat{r}(\mathbf{x}_{0:T}^{(i)}, \mathbf{c})$  using an unbiased MC estimator (see Appendix B.5) and the original reward  $r^{(i)} = r(\mathbf{x}_0^{(i)}, \mathbf{c})$ . We then perform linear correlation analysis for the set of datapoints  $\{(r^{(i)}, \hat{r}^{(i)})\}_{i=1}^{150}$  and show the plot in Fig. 4b. Despite the MC estimation, we observe a relatively strong positive correlation between the rewards with a Pearson correlation of 0.781. This indicates that the chain reward can be approximately viewed as  $\hat{r} \approx a \cdot r + b$ , which draws an interesting connection between the reward-tilted distributions:  $p_{\text{ref}}(\mathbf{x}_{0:T}|\mathbf{c}) \exp(\frac{1}{\beta} \hat{r}(\mathbf{x}_{0:T}, \mathbf{c})) \approx p_{\text{ref}}(\mathbf{x}_{0:T}|\mathbf{c}) \exp(\frac{a}{\beta} \cdot r(\mathbf{x}_0, \mathbf{c}) + \frac{b}{\beta}) \propto p_{\text{ref}}(\mathbf{x}_{0:T}|\mathbf{c}) \exp(\frac{1}{\beta/a} \cdot r(\mathbf{x}_0, \mathbf{c}))$ . Therefore, this empirical investigation interestingly reveals that our trajectory-level chain reward is an effective surrogate of the original reward on  $\mathbf{x}_0$  and, more importantly, our optimal reward-tilted distribution is also a decent approximation of the original optimal reward-tilted distribution without notable bias introduced.

## 6 CONCLUSION

We present SDPO for preference optimization of discrete diffusion models by decomposing diffusion trajectory alignment into a set of subproblems for each diffusion step. Crucially, we propose to align the posterior  $\hat{p}_\theta(\mathbf{x}_0|\mathbf{x}_t)$  for each step and draw an equivalence between the two objectives, with which we further derive a principled loss function. Experiments on a wide range of tasks including DNA sequence design, protein inverse folding, and language modeling consistently verify the efficacy of SDPO, showing its potential towards building performant and applicable discrete diffusion models.

540 ETHICS STATEMENT  
541

542 All authors have read and are committed to comply with the ICLR Code of Ethics (<https://iclr.cc/public/CodeOfEthics>). We present a principled approach for discrete diffusion model  
543 alignment via stepwise decomposition. The goal is to develop a fundamental algorithm for alignment  
544 of discrete diffusion models, where we do not find major ethical concerns that need to highlight.  
545

546  
547 REPRODUCIBILITY STATEMENT  
548

549 We have presented the detailed experimental setup in Sec. 5 and Appendix B. We also include the  
550 code in the supplementary material to ensure reproducibility.  
551

552 REFERENCES  
553

554 Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman,  
555 Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report.  
556 *arXiv preprint arXiv:2303.08774*, 2023. 1, 2

557 Marianne Arriola, Subham Sekhar Sahoo, Aaron Gokaslan, Zhihan Yang, Zhixuan Qi, Jiaqi Han,  
558 Justin T Chiu, and Volodymyr Kuleshov. Block diffusion: Interpolating between autoregres-  
559 sive and diffusion language models. In *The Thirteenth International Conference on Learning  
560 Representations*, 2025. URL <https://openreview.net/forum?id=tyEyYT267x>. 2

561 Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne Van Den Berg. Structured  
562 denoising diffusion models in discrete state-spaces. *Advances in neural information processing  
563 systems*, 34:17981–17993, 2021. 1, 2, 3, 4, 17

564 Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain,  
565 Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a helpful and harmless assistant with  
566 reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022. 1

567 Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models  
568 with reinforcement learning. In *The Twelfth International Conference on Learning Representations*,  
569 2024. URL <https://openreview.net/forum?id=YCWjhGrJFD>. 3

570 Umberto Borso, Davide Paglieri, Jude Wells, and Tim Rocktäschel. Preference-based alignment of  
571 discrete diffusion models. *arXiv preprint arXiv:2503.08295*, 2025. 2, 3, 7, 8

572 Ralph Allan Bradley and Milton E Terry. Rank analysis of incomplete block designs: I. the method  
573 of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952. 2, 19

574 Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal,  
575 Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are  
576 few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020. 1, 2

577 Andrew Campbell, Joe Benton, Valentin De Bortoli, Thomas Rainforth, George Deligiannidis, and  
578 Arnaud Doucet. A continuous time framework for discrete denoising models. *Advances in Neural  
579 Information Processing Systems*, 35:28266–28279, 2022. 1, 2

580 Andrew Campbell, Jason Yim, Regina Barzilay, Tom Rainforth, and Tommi Jaakkola. Generative  
581 flows on discrete state-spaces: Enabling multimodal flows with applications to protein co-design.  
582 *arXiv preprint arXiv:2402.04997*, 2024. 1, 2, 7, 8

583 Jaime A Castro-Mondragon, Rafael Riudavets-Puig, Ieva Rauluseviciute, Roza Berhanu Lemma,  
584 Laura Turchi, Romain Blanc-Mathieu, Jeremy Lucas, Paul Boddie, Aziz Khan, Nicolás Manos-  
585 alva Pérez, et al. Jaspar 2022: the 9th release of the open-access database of transcription factor  
586 binding profiles. *Nucleic acids research*, 50(D1):D165–D173, 2022. 7

587 Huayu Chen, Guande He, Lifan Yuan, Ganqu Cui, Hang Su, and Jun Zhu. Noise contrastive alignment  
588 of language models with explicit rewards, 2024. URL <https://arxiv.org/abs/2402.05369>. 4

594 Paul F. Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei.  
 595 Deep reinforcement learning from human preferences. In Isabelle Guyon, Ulrike von Luxburg,  
 596 Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett  
 597 (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neu-*  
 598 *ral Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp.  
 599 4299–4307, 2017. URL <https://proceedings.neurips.cc/paper/2017/hash/d5e2c0adad503c91f91df240d0cd4e49-Abstract.html>. 1, 2  
 600

601 Kevin Clark, Paul Vicol, Kevin Swersky, and David J. Fleet. Directly fine-tuning diffusion models  
 602 on differentiable rewards. In *The Twelfth International Conference on Learning Representations*,  
 603 2024. URL <https://openreview.net/forum?id=1vmSEVL19f>. 3  
 604

605 Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser,  
 606 Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve  
 607 math word problems. *arXiv preprint arXiv:2110.14168*, 2021. 8, 23  
 608

609 Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Bingxiang He, Wei Zhu, Yuan Ni, Guotong Xie,  
 610 Ruobing Xie, Yankai Lin, et al. Ultrafeedback: Boosting language models with scaled ai feedback.  
 611 *arXiv preprint arXiv:2310.01377*, 2023. 8, 21  
 612

613 Ganqu Cui, Lifan Yuan, Zefan Wang, Hanbin Wang, Wendi Li, Bingxiang He, Yuchen Fan, Tianyu  
 614 Yu, Qixin Xu, Weize Chen, et al. Process reinforcement through implicit rewards. *arXiv preprint*  
 615 *arXiv:2502.01456*, 2025. 5  
 616

617 Justas Dauparas, Ivan Anishchenko, Nathaniel Bennett, Hua Bai, Robert J Ragotte, Lukas F Milles,  
 618 Basile IM Wicky, Alexis Courbet, Rob J de Haas, Neville Bethel, et al. Robust deep learning-based  
 619 protein sequence design using proteinmpnn. *Science*, 378(6615):49–56, 2022. 7  
 620

621 Zehao Dou and Yang Song. Diffusion posterior sampling for linear inverse problem solving: A  
 622 filtering perspective. In *The Twelfth International Conference on Learning Representations*, 2024.  
 623 3  
 624

625 Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B Hashimoto. Length-controlled  
 626 alpacaeval: A simple way to debias automatic evaluators. *arXiv preprint arXiv:2404.04475*, 2024.  
 627 8  
 628

629 Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. Kto: Model  
 630 alignment as prospect theoretic optimization. *arXiv preprint arXiv:2402.01306*, 2024. 2  
 631

632 Ying Fan, Olivia Watkins, Yuqing Du, Hao Liu, Moonkyung Ryu, Craig Boutilier, Pieter Abbeel,  
 633 Mohammad Ghavamzadeh, Kangwook Lee, and Kimin Lee. Dpok: Reinforcement learning for  
 634 fine-tuning text-to-image diffusion models. *Advances in Neural Information Processing Systems*,  
 635 36:79858–79885, 2023. 3  
 636

637 Shanshan Gong, Shivam Agarwal, Yizhe Zhang, Jiacheng Ye, Lin Zheng, Mukai Li, Chenxin An, Peilin  
 638 Zhao, Wei Bi, Jiawei Han, Hao Peng, and Lingpeng Kong. Scaling diffusion language models via  
 639 adaptation from autoregressive models. In *The Thirteenth International Conference on Learning*  
 640 *Representations*, 2025. URL <https://openreview.net/forum?id=j1tSLYKwg8>. 1  
 641

642 Sager J Gosai, Rodrigo I Castro, Natalia Fuentes, John C Butts, Susan Kales, Ramil R Noche,  
 643 Kousuke Mouri, Pardis C Sabeti, Steven K Reilly, and Ryan Tewhey. Machine-guided design of  
 644 synthetic cell type-specific cis-regulatory elements. *bioRxiv*, 2023. 1, 6  
 645

646 Yi Gu, Zhendong Wang, Yueqin Yin, Yujia Xie, and Mingyuan Zhou. Diffusion-rpo: Aligning  
 647 diffusion models through relative preference optimization. *arXiv preprint arXiv:2406.06382*, 2024.  
 648 3  
 649

650 Jiaqi Han, Mingjian Jiang, Yuxuan Song, Stefano Ermon, and Minkai Xu.  $f$ -po: Generalizing  
 651 preference optimization with  $f$ -divergence minimization. *arXiv preprint arXiv:2410.21662*, 2024a.  
 652 1, 2, 5  
 653

654 Jiaqi Han, Minkai Xu, Aaron Lou, Haotian Ye, and Stefano Ermon. Geometric trajectory diffusion  
 655 models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*,  
 656 2024b. URL <https://openreview.net/forum?id=OYmms5Mv9H>. 1  
 657

648 Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*,  
 649 2022. 3, 7, 8  
 650

651 Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in*  
 652 *neural information processing systems*, 33:6840–6851, 2020. 1

653 Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J  
 654 Fleet. Video diffusion models. *Advances in Neural Information Processing Systems*, 35:8633–8646,  
 655 2022. 1

656 Emiel Hoogeboom, Alexey A. Gritsenko, Jasmijn Bastings, Ben Poole, Rianne van den Berg,  
 657 and Tim Salimans. Autoregressive diffusion models. In *International Conference on Learning*  
 658 *Representations*, 2022a. URL <https://openreview.net/forum?id=Lm8T39vLDTE>.  
 659 1, 2  
 660

661 Emiel Hoogeboom, Victor Garcia Satorras, Clément Vignac, and Max Welling. Equivariant diffusion  
 662 for molecule generation in 3d. In *International conference on machine learning*, pp. 8867–8887.  
 663 PMLR, 2022b. 1

664 Chloe Hsu, Robert Verkuil, Jason Liu, Zeming Lin, Brian Hie, Tom Sercu, Adam Lerer, and Alexander  
 665 Rives. Learning inverse folding from millions of predicted structures. In *International conference*  
 666 *on machine learning*, pp. 8946–8970. PMLR, 2022. 1

667 Haozhe Ji, Cheng Lu, Yilin Niu, Pei Ke, Hongning Wang, Jun Zhu, Jie Tang, and Minlie Huang.  
 668 Towards efficient exact optimization of language model alignment. *The Forty-first International*  
 669 *Conference on Machine Learning*, 2024. URL <https://arxiv.org/abs/2402.00856>. 1,  
 670 2, 5  
 671

672 Xin Lai, Zhuotao Tian, Yukang Chen, Senqiao Yang, Xiangru Peng, and Jiaya Jia. Step-dpo: Step-  
 673 wise preference optimization for long-chain reasoning of llms. *arXiv preprint arXiv:2406.18629*,  
 674 2024. 2

675 Cassidy Laidlaw and Anca Dragan. The boltzmann policy distribution: Accounting for systematic  
 676 suboptimality in human models. In *International Conference on Learning Representations*, 2022.  
 677 URL [https://openreview.net/forum?id=\\_1\\_QjPGN5ye](https://openreview.net/forum?id=_1_QjPGN5ye). 5

678 Shufan Li, Konstantinos Kallidromitis, Akash Gokul, Yusuke Kato, and Kazuki Kozuka. Aligning  
 679 diffusion models by optimizing human utility. *arXiv preprint arXiv:2404.04465*, 2024. 3

680 Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy  
 681 Liang, and Tatsunori B. Hashimoto. Alpacaeval: An automatic evaluator of instruction-following  
 682 models. [https://github.com/tatsu-lab/alpaca\\_eval](https://github.com/tatsu-lab/alpaca_eval), 5 2023. 8

683 Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin,  
 684 Robert Verkuil, Ori Kabeli, Yaniv Shmueli, et al. Evolutionary-scale prediction of atomic-level  
 685 protein structure with a language model. *Science*, 379(6637):1123–1130, 2023. 8

686 Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the ratios  
 687 of the data distribution. *arXiv preprint arXiv:2310.16834*, 2023. 1, 2, 3

688 Cheng Lu, Huayu Chen, Jianfei Chen, Hang Su, Chongxuan Li, and Jun Zhu. Contrastive en-  
 689 ergy prediction for exact energy-guided diffusion sampling in offline reinforcement learning. In  
 690 *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Pro-  
 691 ceedings of Machine Learning Research*, pp. 22825–22855. PMLR, 23–29 Jul 2023. URL  
 692 <https://proceedings.mlr.press/v202/lu23d.html>. 4, 5

693 Yu Meng, Mengzhou Xia, and Danqi Chen. Simpo: Simple preference optimization with a reference-  
 694 free reward. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. 2, 8,  
 695 21

696 Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-  
 697 Rong Wen, and Chongxuan Li. Large language diffusion models. *arXiv preprint arXiv:2502.09992*,  
 698 2025. 1, 2, 8, 9, 21, 23, 24

702 Hunter Nisonoff, Junhao Xiong, Stephan Allenspach, and Jennifer Listgarten. Unlocking guidance  
 703 for discrete state-space diffusion and flow models. *arXiv preprint arXiv:2406.01572*, 2024. 3, 6, 7,  
 704 8

705 Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong  
 706 Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow  
 707 instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:  
 708 27730–27744, 2022. 1, 2, 3

709 Jan Peters and Stefan Schaal. Reinforcement learning by reward-weighted regression for operational  
 710 space control. In *Proceedings of the 24th international conference on Machine learning*, pp.  
 711 745–750, 2007. 3, 5

712 Angus Phillips, Hai-Dang Dau, Michael John Hutchinson, Valentin De Bortoli, George Deligiannidis,  
 713 and Arnaud Doucet. Particle denoising diffusion sampler. *arXiv preprint arXiv:2402.06320*, 2024.  
 714 3

715 Mihir Prabhudesai, Russell Mendonca, Zheyang Qin, Katerina Fragkiadaki, and Deepak Pathak.  
 716 Video diffusion alignment via reward gradients. *arXiv preprint arXiv:2407.08737*, 2024. 3

717 Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea  
 718 Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances  
 719 in Neural Information Processing Systems*, 36:53728–53741, 2023. 1, 2, 3, 4, 5, 6

720 Jarrid Rector-Brooks, Mohsin Hasan, Zhangzhi Peng, Cheng-Hao Liu, Sarthak Mittal, Nouha  
 721 Dziri, Michael M. Bronstein, Pranam Chatterjee, Alexander Tong, and Joey Bose. Steering  
 722 masked discrete diffusion models via discrete denoising posterior prediction. In *The Thirteenth  
 723 International Conference on Learning Representations*, 2025. URL <https://openreview.net/forum?id=Ombm8S40zN>. 3, 7

724 Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-  
 725 resolution image synthesis with latent diffusion models. 2022 ieee. In *CVF Conference on  
 726 Computer Vision and Pattern Recognition (CVPR)*, pp. 10674–10685, 2021. 1

727 Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar  
 728 Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic  
 729 text-to-image diffusion models with deep language understanding. *Advances in neural information  
 730 processing systems*, 35:36479–36494, 2022. 1

731 Subham Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin Chiu,  
 732 Alexander Rush, and Volodymyr Kuleshov. Simple and effective masked diffusion language  
 733 models. *Advances in Neural Information Processing Systems*, 37:130136–130184, 2024. 1, 2, 3, 6,  
 734 7

735 John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy  
 736 optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. 2

737 Jiaxin Shi, Kehang Han, Zhe Wang, Arnaud Doucet, and Michalis Titsias. Simplified and generalized  
 738 masked diffusion for discrete data. *Advances in neural information processing systems*, 37:  
 739 103131–103167, 2024. 1, 2, 3, 4, 6, 17

740 Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised  
 741 learning using nonequilibrium thermodynamics. In *International conference on machine learning*,  
 742 pp. 2256–2265. PMLR, 2015. 1

743 Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben  
 744 Poole. Score-based generative modeling through stochastic differential equations. In *International  
 745 Conference on Learning Representations*, 2021. URL <https://openreview.net/forum?id=PxtIG12RRHS>. 1

746 Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford,  
 747 Dario Amodei, and Paul F Christiano. Learning to summarize with human feedback. *Advances in  
 748 Neural Information Processing Systems*, 33:3008–3021, 2020. 1

756 Ibrahim I Taskiran, Katina I Spanier, Hannah Dickmänen, Niklas Kempynck, Alexandra Pančíková,  
 757 Eren Can Ekşi, Gert Hulselmans, Joy N Ismail, Koen Theunis, Roel Vandepoel, et al. Cell-type-  
 758 directed design of synthetic enhancers. *Nature*, 626(7997):212–220, 2024. 6

759

760 Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricuț,  
 761 Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: a family of highly  
 762 capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023. 1, 2

763 Kotaro Tsuboyama, Justas Dauparas, Jonathan Chen, Elodie Laine, Yasser Mohseni Behbahani,  
 764 Jonathan J Weinstein, Niall M Mangan, Sergey Ovchinnikov, and Gabriel J Rocklin. Mega-scale  
 765 experimental analysis of protein folding stability in biology and design. *Nature*, 620(7973):  
 766 434–444, 2023. 7

767 Bram Wallace, Meihua Dang, Rafael Rafailov, Linqi Zhou, Aaron Lou, Senthil Purushwalkam,  
 768 Stefano Ermon, Caiming Xiong, Shafiq Joty, and Nikhil Naik. Diffusion model alignment using  
 769 direct preference optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision  
 770 and Pattern Recognition*, pp. 8228–8238, 2024. 2, 3, 4, 5, 6, 19, 20

771

772 Chenyu Wang, Masatoshi Uehara, Yichun He, Amy Wang, Tommaso Biancalani, Avantika Lal,  
 773 Tommi Jaakkola, Sergey Levine, Hanchen Wang, and Aviv Regev. Fine-tuning discrete diffusion  
 774 models via reward optimization with applications to dna and protein design. *arXiv preprint  
 775 arXiv:2410.13643*, 2024. 1, 2, 3, 4, 5, 6, 7, 8, 20, 23

776

777 Luhuan Wu, Brian Trippe, Christian Naesseth, David Blei, and John P Cunningham. Practical and  
 778 asymptotically exact conditional sampling in diffusion models. *Advances in Neural Information  
 779 Processing Systems*, 36:31372–31403, 2023. 3, 7, 8

780

781 Minkai Xu, Lantao Yu, Yang Song, Chence Shi, Stefano Ermon, and Jian Tang. Geodiff: A geometric  
 782 diffusion model for molecular conformation generation. In *International Conference on Learning  
 783 Representations*, 2022. URL <https://openreview.net/forum?id=PzcvxEMzvQC>. 1

784

785 Minkai Xu, Tomas Geffner, Karsten Kreis, Weili Nie, Yilun Xu, Jure Leskovec, Stefano Ermon, and  
 786 Arash Vahdat. Energy-based diffusion language models for text generation. In *The Thirteenth  
 787 International Conference on Learning Representations*, 2025. URL <https://openreview.net/forum?id=sL2F9YCMxf>. 2

788

789 Kai Yang, Jian Tao, Jiafei Lyu, Chunjiang Ge, Jiaxin Chen, Weihan Shen, Xiaolong Zhu, and Xiu Li.  
 790 Using human feedback to fine-tune diffusion models without any reward model. In *Proceedings of  
 791 the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8941–8951, 2024. 3

792

793 Ling Yang, Ye Tian, Bowen Li, Xinchen Zhang, Ke Shen, Yunhai Tong, and Mengdi Wang. Mmada:  
 794 Multimodal large diffusion language models. *arXiv preprint arXiv:2505.15809*, 2025. 3

795

796 Jiacheng Ye, Zhihui Xie, Lin Zheng, Jiahui Gao, Zirui Wu, Xin Jiang, Zhenguo Li, and Lingpeng  
 797 Kong. Dream 7b, 2025. URL <https://hkunlp.github.io/blog/2025/dream>. 1, 2

798

799 Lingxiao Zhao, Xueying Ding, Lijun Yu, and Leman Akoglu. Improving and unifying  
 800 discrete&continuous-time discrete denoising diffusion. *arXiv e-prints*, pp. arXiv–2402, 2024. 2

801

802 Siyan Zhao, Devaansh Gupta, Qingqing Zheng, and Aditya Grover. d1: Scaling reasoning in diffusion  
 803 large language models via reinforcement learning. *arXiv preprint arXiv:2504.12216*, 2025. 2, 3, 7,  
 8

804

805 Kaiwen Zheng, Yongxin Chen, Hanzi Mao, Ming-Yu Liu, Jun Zhu, and Qinsheng Zhang. Masked  
 806 diffusion models are secretly time-agnostic masked models and exploit inaccurate categorical  
 807 sampling. In *The Thirteenth International Conference on Learning Representations*, 2025. URL  
 808 <https://openreview.net/forum?id=CTC7CmirNr>. 2

809

810 Lin Zheng, Jianbo Yuan, Lei Yu, and Lingpeng Kong. A reparameterized discrete diffusion model for  
 811 text generation. *arXiv preprint arXiv:2302.05737*, 2023. 1, 2

812

813 Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny  
 814 Zhou, and Le Hou. Instruction-following evaluation for large language models. *arXiv preprint  
 815 arXiv:2311.07911*, 2023. 8

810 Fengqi Zhu, Rongzhen Wang, Shen Nie, Xiaolu Zhang, Chunwei Wu, Jun Hu, Jun Zhou, Jianfei  
811 Chen, Yankai Lin, Ji-Rong Wen, et al. Llada 1.5: Variance-reduced preference optimization for  
812 large language diffusion models. *arXiv preprint arXiv:2505.19223*, 2025a. 3, 7, 8

813  
814 Huaisheng Zhu, Teng Xiao, and Vasant G Honavar. Dspo: Direct score preference optimization for dif-  
815 fusion model alignment. In *The Thirteenth International Conference on Learning Representations*,  
816 2025b. 2, 3

817  
818  
819  
820  
821  
822  
823  
824  
825  
826  
827  
828  
829  
830  
831  
832  
833  
834  
835  
836  
837  
838  
839  
840  
841  
842  
843  
844  
845  
846  
847  
848  
849  
850  
851  
852  
853  
854  
855  
856  
857  
858  
859  
860  
861  
862  
863

864 The appendix is structured as follows.  
 865

866

- 867 • In Appendix A, we provide detailed proofs of the theorems presented in the main paper and  
 868 additional theoretical derivations.
- 869 • In Appendix B, we provide more experiment details and hyperparameters for the experiments  
 870 in the paper.
- 871 • In Appendix C, we present more experiment results and ablations.
- 872 • In Appendix D, we offer discussions on the limitations and broader impact of the proposed  
 873 approach.

874

875 **A PROOFS**

876

877 **A.1 PROOF OF THEOREM 4.1**

878

879 **Theorem 4.1.** *The joint  $p^*(\mathbf{x}_{0:T}|\mathbf{c})$  induced by the optimal solutions  $\{\hat{p}^*(\mathbf{x}_0|\mathbf{x}_t, \mathbf{c})\}_{t=1}^T$  of Eq. 7  
 880 is also the optimal solution of the trajectory alignment objective in Eq. 5, with the chain reward  
 881  $\hat{r}(\mathbf{x}_{0:T}, \mathbf{c}) = \beta \sum_{t=1}^T r_t(\mathbf{x}_{t-1}, \mathbf{x}_t, \mathbf{c})$  where  $r_t(\mathbf{x}_{t-1}, \mathbf{x}_t, \mathbf{c}) = \log \frac{\mathbb{E}_{p'_{\text{ref}}(\mathbf{x}_0|\mathbf{x}_{t-1}, \mathbf{x}_t, \mathbf{c})}[\exp(\frac{1}{\beta_t} r(\mathbf{x}_0, \mathbf{c}))]}{\mathbb{E}_{\hat{p}_{\text{ref}}(\mathbf{x}_0|\mathbf{x}_t, \mathbf{c})}[\exp(\frac{1}{\beta_t} r(\mathbf{x}_0, \mathbf{c}))]}$ .*

882

883 *Proof.* Leveraging Eq. 4, the optimal solution for each subproblem in Eq. 7 is given by

884

$$885 \hat{p}^*(\mathbf{x}_0|\mathbf{x}_t, \mathbf{c}) = \frac{1}{Z_t(\mathbf{x}_t, \mathbf{c})} \hat{p}_{\text{ref}}(\mathbf{x}_0|\mathbf{x}_t, \mathbf{c}) \exp\left(\frac{1}{\beta_t} r(\mathbf{x}_0, \mathbf{c})\right), \quad \forall 1 \leq t \leq T, \quad (13)$$

886

887 where  $Z_t(\mathbf{x}_t, \mathbf{c}) = \sum_{\mathbf{x}_0} \hat{p}_{\text{ref}}(\mathbf{x}_0|\mathbf{x}_t, \mathbf{c}) \exp\left(\frac{1}{\beta_t} r(\mathbf{x}_0, \mathbf{c})\right) = \mathbb{E}_{\hat{p}_{\text{ref}}(\mathbf{x}_0|\mathbf{x}_t, \mathbf{c})} \left[ \exp\left(\frac{1}{\beta_t} r(\mathbf{x}_0, \mathbf{c})\right) \right]$ . The  
 888 transition kernels  $p^*(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{c})$  induced by the solutions can be derived as  
 889

$$890 p^*(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{c}) = \sum_{\mathbf{x}_0} \hat{p}^*(\mathbf{x}_0|\mathbf{x}_t, \mathbf{c}) q(\mathbf{x}_{t-1}|\mathbf{x}_0, \mathbf{x}_t), \quad (14)$$

901

$$902 = \sum_{\mathbf{x}_0} \frac{1}{Z_t(\mathbf{x}_t, \mathbf{c})} \hat{p}_{\text{ref}}(\mathbf{x}_0|\mathbf{x}_t, \mathbf{c}) \exp\left(\frac{1}{\beta_t} r(\mathbf{x}_0, \mathbf{c})\right) q(\mathbf{x}_{t-1}|\mathbf{x}_0, \mathbf{x}_t), \quad (15)$$

903

$$904 = p_{\text{ref}}(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{c}) \sum_{\mathbf{x}_0} \frac{\hat{p}_{\text{ref}}(\mathbf{x}_0|\mathbf{x}_t, \mathbf{c}) q(\mathbf{x}_{t-1}|\mathbf{x}_0, \mathbf{x}_t)}{Z_t(\mathbf{x}_t, \mathbf{c}) p_{\text{ref}}(\mathbf{x}_{t-1}|\mathbf{x}_t)} \exp\left(\frac{1}{\beta_t} r(\mathbf{x}_0, \mathbf{c})\right), \quad (16)$$

905

$$906 = \frac{1}{Z_t(\mathbf{x}_t, \mathbf{c})} p_{\text{ref}}(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{c}) \sum_{\mathbf{x}_0} \frac{\hat{p}_{\text{ref}}(\mathbf{x}_0|\mathbf{x}_t, \mathbf{c}) q(\mathbf{x}_{t-1}|\mathbf{x}_0, \mathbf{x}_t)}{p_{\text{ref}}(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{c})} \exp\left(\frac{1}{\beta_t} r(\mathbf{x}_0, \mathbf{c})\right), \quad (17)$$

907

$$908 = \frac{1}{Z_t(\mathbf{x}_t, \mathbf{c})} p_{\text{ref}}(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{c}) \mathbb{E}_{p'_{\text{ref}}(\mathbf{x}_0|\mathbf{x}_{t-1}, \mathbf{x}_t, \mathbf{c})} \left[ \exp\left(\frac{1}{\beta_t} r(\mathbf{x}_0, \mathbf{c})\right) \right], \quad (18)$$

909

910 where  $p'_{\text{ref}}(\mathbf{x}_0|\mathbf{x}_{t-1}, \mathbf{x}_t, \mathbf{c}) := \frac{\hat{p}_{\text{ref}}(\mathbf{x}_0|\mathbf{x}_t, \mathbf{c}) q(\mathbf{x}_{t-1}|\mathbf{x}_0, \mathbf{x}_t)}{p_{\text{ref}}(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{c})}$ . Specifically, Eq. 14 holds due to the  $\mathbf{x}_0$ -  
 911 parameterization of the transition kernel (see Austin et al. (2021); Shi et al. (2024)). Notably it  
 912 is straightforward to verify that  $p'_{\text{ref}}(\mathbf{x}_0|\mathbf{x}_{t-1}, \mathbf{x}_t, \mathbf{c})$  is a properly normalized distribution since  
 913  $\sum_{\mathbf{x}_0} p'_{\text{ref}}(\mathbf{x}_0|\mathbf{x}_{t-1}, \mathbf{x}_t, \mathbf{c}) = \frac{\sum_{\mathbf{x}_0} \hat{p}_{\text{ref}}(\mathbf{x}_0|\mathbf{x}_t, \mathbf{c}) q(\mathbf{x}_{t-1}|\mathbf{x}_0, \mathbf{x}_t)}{p_{\text{ref}}(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{c})} = \frac{p(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{c})}{p(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{c})} = 1$ .

914

918 Plugging it back into the Markovian factorization of the reverse process, we arrive at  
919

$$920 \quad p^*(\mathbf{x}_{0:T}|\mathbf{c}) = p(\mathbf{x}_T) \prod_{t=1}^{t=T} p^*(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{c}), \quad (19)$$

$$923 \quad = p(\mathbf{x}_T) \prod_{t=1}^{t=T} \left( \frac{1}{Z_t(\mathbf{x}_t, \mathbf{c})} p_{\text{ref}}(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{c}) \mathbb{E}_{p'_{\text{ref}}(\mathbf{x}_0|\mathbf{x}_{t-1}, \mathbf{x}_t, \mathbf{c})} [r(\mathbf{x}_0, \mathbf{c})] \right), \quad (20)$$

$$925 \quad = p(\mathbf{x}_T) \prod_{t=1}^{t=T} p_{\text{ref}}(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{c}) \prod_{t=1}^{t=T} \frac{\mathbb{E}_{p'_{\text{ref}}(\mathbf{x}_0|\mathbf{x}_{t-1}, \mathbf{x}_t, \mathbf{c})} [\exp(\frac{1}{\beta_t} r(\mathbf{x}_0, \mathbf{c}))]}{Z_t(\mathbf{x}_t, \mathbf{c})}, \quad (21)$$

$$929 \quad = p_{\text{ref}}(\mathbf{x}_{0:T}|\mathbf{c}) \exp \left( \sum_{t=1}^T \log \frac{\mathbb{E}_{p'_{\text{ref}}(\mathbf{x}_0|\mathbf{x}_{t-1}, \mathbf{x}_t, \mathbf{c})} [\exp(\frac{1}{\beta_t} r(\mathbf{x}_0, \mathbf{c}))]}{Z_t(\mathbf{x}_t, \mathbf{c})} \right), \quad (22)$$

$$932 \quad = p_{\text{ref}}(\mathbf{x}_{0:T}|\mathbf{c}) \exp \left( \sum_{t=1}^T \log \frac{\mathbb{E}_{p'_{\text{ref}}(\mathbf{x}_0|\mathbf{x}_{t-1}, \mathbf{x}_t, \mathbf{c})} [\exp(\frac{1}{\beta_t} r(\mathbf{x}_0, \mathbf{c}))]}{\mathbb{E}_{p_{\text{ref}}(\mathbf{x}_0|\mathbf{x}_t, \mathbf{c})} [\exp(\frac{1}{\beta_t} r(\mathbf{x}_0, \mathbf{c}))]} \right), \quad (23)$$

$$936 \quad = p_{\text{ref}}(\mathbf{x}_{0:T}|\mathbf{c}) \exp \left( \sum_{t=1}^T r_t(\mathbf{x}_{t-1}, \mathbf{x}_t, \mathbf{c}) \right), \quad (24)$$

$$938 \quad = p_{\text{ref}}(\mathbf{x}_{0:T}|\mathbf{c}) \exp \left( \frac{1}{\beta} \cdot \underbrace{\sum_{t=1}^T r_t(\mathbf{x}_{t-1}, \mathbf{x}_t, \mathbf{c})}_{\hat{r}(\mathbf{x}_{0:T}, \mathbf{c})} \right), \quad (25)$$

944 where  $r_t(\mathbf{x}_{t-1}, \mathbf{x}_t, \mathbf{c}) = \log \frac{\mathbb{E}_{p'_{\text{ref}}(\mathbf{x}_0|\mathbf{x}_{t-1}, \mathbf{x}_t, \mathbf{c})} [\exp(\frac{1}{\beta_t} r(\mathbf{x}_0, \mathbf{c}))]}{\mathbb{E}_{\hat{p}_{\text{ref}}(\mathbf{x}_0|\mathbf{x}_t, \mathbf{c})} [\exp(\frac{1}{\beta_t} r(\mathbf{x}_0, \mathbf{c}))]}$ . Eq. 24 directly implies that the induced  
945 distribution  $p^*(\mathbf{x}_{0:T}|\mathbf{c})$  is the optimal solution of the trajectory alignment objective in Eq. 5 with  
946  $\hat{r}(\mathbf{x}_{0:T}, \mathbf{c}) = \beta \sum_{t=1}^T r_t(\mathbf{x}_{t-1}, \mathbf{x}_t, \mathbf{c})$ , which concludes the proof.  $\square$

## A.2 PROOF OF PROPOSITION 4.2

950 **Proposition 4.2.** Let  $\theta^* = \arg \min \mathcal{L}_t(\theta)$  defined in Eq. 8. Then  $\hat{p}_{\theta^*}(\mathbf{x}_0|\mathbf{x}_t, \mathbf{c})$  is the optimal solution  
951 of the stepwise alignment objective in Eq. 7.

953 *Proof.* Recall the definition of  $\mathcal{L}_t(\theta)$  in Eq. 8:

$$955 \quad \mathcal{L}_t(\theta) := \mathbb{E}_{\mathbf{x}_t, \mathbf{c}} [D_{\text{KL}} [\hat{p}_r(\mathbf{x}_0|\mathbf{x}_t, \mathbf{c}) \| \tilde{p}_\theta(\mathbf{x}_0|\mathbf{x}_t, \mathbf{c})]]. \quad (26)$$

956 Since the KL-divergence is minimized when the two distributions are exactly matched, we have that  
957 the optimal  $\theta^*$  satisfies

$$958 \quad \hat{p}_r(\mathbf{x}_0|\mathbf{x}_t, \mathbf{c}) = \tilde{p}_{\theta^*}(\mathbf{x}_0|\mathbf{x}_t, \mathbf{c}). \quad (27)$$

959 By leveraging the definition of  $\hat{p}_r$  and  $\tilde{p}_\theta$ , we have

$$960 \quad \frac{1}{Z_r(\mathbf{x}_t, \mathbf{c})} \hat{p}_{\text{ref}}(\mathbf{x}_0|\mathbf{x}_t, \mathbf{c}) \exp(r(\mathbf{x}_0, \mathbf{c})) = \frac{1}{Z_{\theta^*}(\mathbf{x}_t, \mathbf{c})} \hat{p}_{\text{ref}}(\mathbf{x}_0|\mathbf{x}_t, \mathbf{c})^{(1-\beta_t)} \hat{p}_{\theta^*}(\mathbf{x}_0|\mathbf{x}_t, \mathbf{c})^{\beta_t}, \quad (28)$$

962 which simplifies to

$$963 \quad \left( \frac{\hat{p}_{\theta^*}(\mathbf{x}_0|\mathbf{x}_t, \mathbf{c})}{\hat{p}_{\text{ref}}(\mathbf{x}_0|\mathbf{x}_t, \mathbf{c})} \right)^{\beta_t} = \frac{Z_{\theta^*}(\mathbf{x}_t, \mathbf{c})}{Z_r(\mathbf{x}_t, \mathbf{c})} \exp(r(\mathbf{x}_0, \mathbf{c})), \quad (29)$$

966  $\square$

967 and finally gives us

$$969 \quad \hat{p}_{\theta^*}(\mathbf{x}_0|\mathbf{x}_t, \mathbf{c}) = \frac{1}{Z'(\mathbf{x}_t, \mathbf{c})} \hat{p}_{\text{ref}}(\mathbf{x}_0|\mathbf{x}_t, \mathbf{c}) \exp\left(\frac{1}{\beta_t} r(\mathbf{x}_0, \mathbf{c})\right), \quad (30)$$

971 where  $Z'(\mathbf{x}_t, \mathbf{c}) = \left(\frac{Z_r(\mathbf{x}_t, \mathbf{c})}{Z_{\theta^*}(\mathbf{x}_t, \mathbf{c})}\right)^{1/\beta_t}$ . The proof is therefore completed.

972 A.3 PROOF OF THE LOSS IN EQ. 8  
973974 Here we provide the detailed derivation on how to derive Eq. 8 from  $\mathcal{L}_t(\theta)$  step-by-step.  
975976 We start from the definition of  $\mathcal{L}_t(\theta)$  in Eq. 7:  
977

978 
$$\mathcal{L}_t(\theta) = \mathbb{E}_{\mathbf{x}_t, \mathbf{c}} [D_{\text{KL}} [\tilde{p}_r(\mathbf{x}_0 | \mathbf{x}_t, \mathbf{c}) \| \tilde{p}_\theta(\mathbf{x}_0 | \mathbf{x}_t, \mathbf{c})]], \quad (31)$$
  
979

980 
$$= \mathbb{E}_{\mathbf{x}_t, \mathbf{c}, \mathbf{x}_0 \sim \tilde{p}_r(\mathbf{x}_0 | \mathbf{x}_t, \mathbf{c})} \left[ \log \left( \frac{\tilde{p}_r(\mathbf{x}_0 | \mathbf{x}_t, \mathbf{c})}{\tilde{p}_\theta(\mathbf{x}_0 | \mathbf{x}_t, \mathbf{c})} \right) \right], \quad (32)$$
  
981

982 
$$= \mathbb{E}_{\mathbf{c}, (\mathbf{x}_0, \mathbf{x}_t) \sim \tilde{p}_r(\mathbf{x}_0, \mathbf{x}_t | \mathbf{c})} \left[ \log \left( \frac{\tilde{p}_r(\mathbf{x}_0 | \mathbf{x}_t, \mathbf{c})}{\tilde{p}_\theta(\mathbf{x}_0 | \mathbf{x}_t, \mathbf{c})} \right) \right], \quad (33)$$
  
983

984 
$$= \mathbb{E}_{\mathbf{c}, (\mathbf{x}_0, \mathbf{x}_t) \sim \hat{p}_{\text{ref}}(\mathbf{x}_0, \mathbf{x}_t | \mathbf{c})} \left[ \frac{\tilde{p}_r(\mathbf{x}_0, \mathbf{x}_t | \mathbf{c})}{\hat{p}_{\text{ref}}(\mathbf{x}_0, \mathbf{x}_t | \mathbf{c})} \log \left( \frac{\tilde{p}_r(\mathbf{x}_0 | \mathbf{x}_t, \mathbf{c}) / \hat{p}_{\text{ref}}(\mathbf{x}_0 | \mathbf{x}_t, \mathbf{c})}{\tilde{p}_\theta(\mathbf{x}_0 | \mathbf{x}_t, \mathbf{c}) / \hat{p}_{\text{ref}}(\mathbf{x}_0 | \mathbf{x}_t, \mathbf{c})} \right) \right], \quad (34)$$
  
985

986 
$$= \mathbb{E}_{\mathbf{c}, (\mathbf{x}_0, \mathbf{x}_t) \sim \hat{p}_{\text{ref}}(\mathbf{x}_0, \mathbf{x}_t | \mathbf{c})} \left[ \frac{\exp(r(\mathbf{x}_0, \mathbf{c}))}{\mathbb{E}_{\hat{p}_{\text{ref}}(\mathbf{x}_0, \mathbf{x}_t | \mathbf{c})} \exp(r(\mathbf{x}_0, \mathbf{c}))} \log \left( \frac{\tilde{p}_r(\mathbf{x}_0 | \mathbf{x}_t, \mathbf{c}) / \hat{p}_{\text{ref}}(\mathbf{x}_0 | \mathbf{x}_t, \mathbf{c})}{\tilde{p}_\theta(\mathbf{x}_0 | \mathbf{x}_t, \mathbf{c}) / \hat{p}_{\text{ref}}(\mathbf{x}_0 | \mathbf{x}_t, \mathbf{c})} \right) \right], \quad (35)$$
  
987

988 
$$= \mathbb{E}_{\mathbf{c}, (\mathbf{x}_0, \mathbf{x}_t) \sim \hat{p}_{\text{ref}}(\mathbf{x}_0, \mathbf{x}_t | \mathbf{c})} \left[ \frac{\exp(r(\mathbf{x}_0, \mathbf{c}))}{\mathbb{E}_{\hat{p}_{\text{ref}}(\mathbf{x}_0 | \mathbf{c})} \exp(r(\mathbf{x}_0, \mathbf{c}))} \log \left( \frac{\tilde{p}_r(\mathbf{x}_0 | \mathbf{x}_t, \mathbf{c}) / \hat{p}_{\text{ref}}(\mathbf{x}_0 | \mathbf{x}_t, \mathbf{c})}{\tilde{p}_\theta(\mathbf{x}_0 | \mathbf{x}_t, \mathbf{c}) / \hat{p}_{\text{ref}}(\mathbf{x}_0 | \mathbf{x}_t, \mathbf{c})} \right) \right], \quad (36)$$
  
989

990 
$$= -\mathbb{E}_{\mathbf{c}, (\mathbf{x}_0, \mathbf{x}_t) \sim \hat{p}_{\text{ref}}(\mathbf{x}_0, \mathbf{x}_t | \mathbf{c})} \left[ \frac{\exp(r(\mathbf{x}_0, \mathbf{c}))}{Z_r(\mathbf{c})} \log \left( \frac{1}{Z_\theta^t(\mathbf{x}_t, \mathbf{c}, \beta_t)} \left( \frac{p_\theta(\mathbf{x}_0 | \mathbf{x}_t, \mathbf{c})}{\hat{p}_{\text{ref}}(\mathbf{x}_0 | \mathbf{x}_t, \mathbf{c})} \right)^{\beta_t} \right) \right] + C, \quad (37)$$
  
991

992 where the last step extracts the constant  $C$  out of the numerator of  $\log$  since it is irrelevant to  $\theta$ .  
993994 Recalling the definition of the implicit reward, which is given by  $\tilde{r}_\theta(\mathbf{x}_0, \mathbf{x}_t, \mathbf{c}, \beta_t) = \beta_t (\log p_\theta(\mathbf{x}_0 | \mathbf{x}_t, \mathbf{c}) - \log \hat{p}_{\text{ref}}(\mathbf{x}_0 | \mathbf{x}_t, \mathbf{c}))$ , we have that  $\left( \frac{\hat{p}_\theta(\mathbf{x}_0 | \mathbf{x}_t, \mathbf{c})}{\hat{p}_{\text{ref}}(\mathbf{x}_0 | \mathbf{x}_t, \mathbf{c})} \right)^{\beta_t} = \exp(\tilde{r}_\theta(\mathbf{x}_0, \mathbf{x}_t, \mathbf{c}, \beta_t))$ .  
995 Therefore, we can further simplify  
996

997 
$$\mathcal{L}_t(\theta) = -\mathbb{E}_{\mathbf{c}, (\mathbf{x}_0, \mathbf{x}_t) \sim \hat{p}_{\text{ref}}(\mathbf{x}_0, \mathbf{x}_t | \mathbf{c})} \left[ \frac{\exp(r(\mathbf{x}_0, \mathbf{c}))}{Z_r(\mathbf{c})} \log \left( \frac{\exp(\tilde{r}_\theta(\mathbf{x}_0, \mathbf{x}_t, \mathbf{c}, \beta_t))}{Z_\theta^t(\mathbf{x}_t, \mathbf{c}, \beta_t)} \right) \right] + C, \quad (38)$$
  
998

999 where  $C$  is a constant irrelevant to  $\theta$  and  $Z_r(\mathbf{c})$  and  $Z_\theta^t(\mathbf{x}_t, \mathbf{c}, \beta_t)$  are the partition functions.  
10001001 A.4 DERIVATION OF SDPO LOSS IN THE DPO SETTING  
10021003 Recall our proposed loss function  $\mathcal{L}(\theta)$ :  
1004

1005 
$$\mathcal{L}(\theta) = -\mathbb{E}_{t, \mathbf{c}, \mathbf{x}_0, q(\mathbf{x}_t | \mathbf{x}_0)} \sum_{i=1}^N \left( \frac{\exp(r(\mathbf{x}_0^{(i)}, \mathbf{c}))}{\sum_{j=1}^N \exp(r(\mathbf{x}_0^{(j)}, \mathbf{c}))} \cdot \log \frac{\exp(\tilde{r}_\theta(\mathbf{x}_0^{(i)}, \mathbf{x}_t^{(i)}, \mathbf{c}, \beta_t))}{\sum_{j=1}^N \exp(\tilde{r}_\theta(\mathbf{x}_0^{(j)}, \mathbf{x}_t^{(j)}, \mathbf{c}, \beta_t))} \right), \quad (39)$$
  
1006

1007 with  
1008

1009 
$$\tilde{r}_\theta(\mathbf{x}_0, \mathbf{x}_t, \mathbf{c}, \beta_t) = \beta \left( \frac{\log(\mathbf{x}_0^\top \mathbf{f}_\theta(\mathbf{x}_t, t, \mathbf{c}))}{w(t)} - \frac{\log(\mathbf{x}_0^\top \mathbf{f}_{\text{ref}}(\mathbf{x}_t, t, \mathbf{c}))}{w(t)} \right). \quad (40)$$
  
1010

1011 Here we derive a specific instance of  $\mathcal{L}(\theta)$  in the DPO pairwise preference setting, and draw connection  
1012 of it to Wallace et al. (2024).  
10131014 In particular, in DPO preference pair setting for each context  $\mathbf{c}$  there are two completions, namely,  
1015  $N = 2$  in our case. Furthermore, one completion is labeled as the preferred (chosen) response  
1016  $\mathbf{x}_0^{(w)}$  and the other as rejected sample  $\mathbf{x}_0^{(l)}$ . Since no explicit real-valued reward on the chosen and  
1017 rejected sample is provided, the Bradley-Terry (BT) model (Bradley & Terry, 1952) is adopted, which  
1018

1026 corresponds to, in our case, setting  $r(\mathbf{x}_0^{(w)}, \mathbf{c}) = 0$  and  $r(\mathbf{x}_0^{(l)}, \mathbf{c}) = -\infty$ . Under this specification,  
 1027  $\mathcal{L}(\theta)$  is simplified as  
 1028

$$\begin{aligned} 1029 \quad \mathcal{L}(\theta) &= -\mathbb{E}_{t, \mathbf{c}, \mathbf{x}_0, q(\mathbf{x}_t | \mathbf{x}_0)} \left[ \right. \\ 1030 \quad &\frac{\exp(r(\mathbf{x}_0^{(w)}, \mathbf{c}))}{\exp(r(\mathbf{x}_0^{(w)}, \mathbf{c})) + \exp(r(\mathbf{x}_0^{(l)}, \mathbf{c}))} \cdot \log \frac{\exp(\tilde{r}_\theta(\mathbf{x}_0^{(w)}, \mathbf{x}_t^{(w)}))}{\exp(\tilde{r}_\theta(\mathbf{x}_0^{(w)}, \mathbf{x}_t^{(w)})) + \exp(\tilde{r}_\theta(\mathbf{x}_0^{(l)}, \mathbf{x}_t^{(l)}))} \\ 1031 \quad &+ \left. \frac{\exp(r(\mathbf{x}_0^{(l)}, \mathbf{c}))}{\exp(r(\mathbf{x}_0^{(w)}, \mathbf{c})) + \exp(r(\mathbf{x}_0^{(l)}, \mathbf{c}))} \cdot \log \frac{\exp(\tilde{r}_\theta(\mathbf{x}_0^{(l)}, \mathbf{x}_t^{(l)}))}{\exp(\tilde{r}_\theta(\mathbf{x}_0^{(w)}, \mathbf{x}_t^{(w)})) + \exp(\tilde{r}_\theta(\mathbf{x}_0^{(l)}, \mathbf{x}_t^{(l)}))} \right], \\ 1032 \quad &= -\mathbb{E}_{t, \mathbf{c}, \mathbf{x}_0, q(\mathbf{x}_t | \mathbf{x}_0)} \left[ \frac{1}{1+0} \log \frac{1}{1 + \exp(\tilde{r}_\theta(\mathbf{x}_0^{(l)}, \mathbf{x}_t^{(l)}) - \tilde{r}_\theta(\mathbf{x}_0^{(w)}, \mathbf{x}_t^{(w)}))} + 0 \right], \quad (41) \\ 1033 \quad &= -\mathbb{E}_{t, \mathbf{c}, \mathbf{x}_0, q(\mathbf{x}_t | \mathbf{x}_0)} \left[ \log \frac{1}{1 + \exp(\tilde{r}_\theta(\mathbf{x}_0^{(l)}, \mathbf{x}_t^{(l)}) - \tilde{r}_\theta(\mathbf{x}_0^{(w)}, \mathbf{x}_t^{(w)}))} \right], \quad (42) \\ 1034 \quad &= -\mathbb{E}_{t, \mathbf{c}, \mathbf{x}_0, q(\mathbf{x}_t | \mathbf{x}_0)} \log \sigma \left( \tilde{r}_\theta(\mathbf{x}_0^{(w)}, \mathbf{x}_t^{(w)}) - \tilde{r}_\theta(\mathbf{x}_0^{(l)}, \mathbf{x}_t^{(l)}) \right), \quad (43) \end{aligned}$$

1044 where  $\tilde{r}_\theta(\mathbf{x}_0^{(w)}, \mathbf{x}_t^{(w)})$  is shorthand for  $\tilde{r}_\theta(\mathbf{x}_0^{(w)}, \mathbf{x}_t^{(w)}, \mathbf{c}, \beta_t)$  and similarly for the losing sample. Eq. 43  
 1045 underscores an interesting connection of our loss to that of Wallace et al. (2024) specifically in the  
 1046 DPO setting, since both share the same form of negative logsigmoid over the margin between the  
 1047 implicit rewards of the winning and losing sample, with the difference being the definition of the  
 1048 implicit reward (Eq. 12), depending on whether using Gaussian diffusion (Wallace et al., 2024) or  
 1049 discrete diffusion as in this work. Notably, since we leverage a general formulation of stepwise  
 1050 decomposition that reduces the problem to a stepwise distribution matching objective, we are able to  
 1051 generalize to the setting with arbitrary number of samples and reward model, which is not revealed  
 1052 in Wallace et al. (2024).  
 1053

## 1054 B MORE EXPERIMENT DETAILS

### 1055 B.1 DNA SEQUENCE DESIGN

1058 We use the pre-trained model and fine-tuning reward oracle from Wang et al. (2024) for finetuning  
 1059 with SDPO. In the first stage of finetuning, we train on the original enhancer dataset (also used for  
 1060 pre-training), without using re-labeled samples. In the next two stages, we generate 10000 samples  
 1061 from the finetuned model, label the samples with the reward model, and continue finetuning on the  
 1062 relabeled data. In all three stages, we use the original pre-trained checkpoint as the reference model.  
 1063 We provide hyperparameter configurations in Table 5.

1064 1065 Table 5: Detailed hyperparameters for DNA design task.

1066 Stage	1067 # relabeled samples	1068 $\beta$	1069 $N$	1070 # Epochs	1071 Learning Rate
1068 Stage 1	1069 0	1070 0.92	1071 25	1072 10	1073 9e-5
1069 Stage 2	1070 10k	1071 0.4	1072 200	1073 2	1074 1e-5
1070 Stage 3	1071 10k	1072 0.064	1073 918	1074 2	1075 7.4e-6

### 1073 B.2 DETAILED HYPERPARAMETERS FOR PROTEIN INVERSE FOLDING TASK.

1075 We also use the pre-trained model and fine-tuning reward oracle from Wang et al. (2024) for finetuning  
 1076 with SDPO. Likewise, we finetune on the original pre-training dataset without re-labeling any new  
 1077 samples. In the second stage, we re-label 12800 generated samples with the reward oracle, then  
 1078 continue finetuning with SDPO. Differing in our setup from the DNA experiment, we find that  
 1079 using the previously finetuned checkpoint as a reference model during Stage 2 results in superior  
 1080 performance. We provide hyperparameter configurations in Table 6.

1080 Table 6: Detailed hyperparameters for protein inverse folding task.  
1081

1082 Stage	1083 # relabeled samples	1084 $\beta$	1085 N	1086 # Epochs	1087 Learning Rate
1082 Stage 1	1083 0	1084 0.047	1085 25	1086 9	1087 5.7e-6
1082 Stage 2	1083 12.8k	1084 0.063	1085 200	1086 5	1087 8.5e-5

1088 **Algorithm 1 SDPO Training Algorithm**1089 **Input:** Initial dataset  $\mathcal{D}^{(0)}$ , pretrained discrete diffusion model  $p_{\text{ref}}$  and trainable model  $p_{\theta}$ .

```

1090 1: for iter  $k = 0, \dots, K$  do
1091 2:   for step  $l = 0, \dots, L$  do
1092 3:     Sample  $\{(\mathbf{x}_0^{(i)}, \mathbf{c})\}_{i=1}^N \sim \mathcal{D}^{(k)}$  for each prompt  $\mathbf{c}$  in the batch
1093 4:     Sample  $\mathbf{x}_t^{(i)} \sim q(\mathbf{x}_t | \mathbf{x}_0)$  given  $\mathbf{x}_0^{(i)}$ 
1094 5:     Compute loss  $\mathcal{L}(\theta)$  via Eq. 11
1095 6:      $\theta = \theta - \lambda \nabla_{\theta} \mathcal{L}(\theta)$  ▷ SDPO loss
1096 7:      $\theta = \theta - \lambda \nabla_{\theta} \mathcal{L}(\theta)$  ▷ Gradient update
1097 8:     Generate  $\mathcal{D}^{(k+1)}$  via  $p_{\theta}(\mathbf{x}_{0:T} | \mathbf{c}) = p(\mathbf{c})p(\mathbf{x}_T) \prod_{t=1}^T p_{\theta}(\mathbf{x}_{t-1} | \mathbf{x}_t, \mathbf{c})$  ▷ Optional iterative labeling
1098 9:      $p_{\text{ref}} \leftarrow p_{\theta}$  ▷ Optional reference model update
1100
1101 Return: Optimized model  $p_{\theta}$ 

```

## 1100 B.3 LANGUAGE MODELING

1102 We leverage the open-source checkpoint<sup>1</sup> of LLaDA-8B-Instruct (Nie et al., 2025) as the base model  
1103 to perform our SDPO. We use UltraFeedback (Cui et al., 2023) dataset labeled by Meng et al. (2024)  
1104 as the finetuning dataset<sup>2</sup>. We operate in the pairwise setting with  $N = 2$  on the dataset, with labeled  
1105 pairs of winning and losing samples with rewards. We use 8 Nvidia 80G A100 GPUs with DeepSpeed  
1106 enabled during finetuning, due to the scale of the model. We use per device batch size 2 and gradient  
1107 accumulation of 16 steps, leading to an effective global batch size of 256. We set the learning rate  
1108 to 1e-6 and  $\beta$  to 1.0 and train the model for 2 epochs. At inference time, we reuse the inference  
1109 hyperparameters adopted in Nie et al. (2025) for GSM8K without any additional tuning, which  
1110 include total length 256, block size 8, and total number of steps 256. For IFEval and AlpacaEval 2.0,  
1111 we keep the same set of hyperparameters except setting block size to 32. We always adopt the low  
1112 confidence remasking strategy, following Nie et al. (2025).

## 1114 B.4 COMPLEXITY ANALYSIS

1116 Computational and memory complexity. As an offline preference optimization approach, SDPO  
1117 is not bottlenecked by online data generation during training, and the offline data generation can  
1118 be fully parallelized. In detail, it is of  $\mathcal{O}(NM(L^2D + LD^2))$  for computational complexity and  
1119  $\mathcal{O}(NM(L^2 + LD + D^2))$  for memory complexity, where  $N$  is the number of Monte-Carlo samples,  
1120  $M$  is the number of attention blocks,  $L$  is sequence length, and  $D$  is the latent dimension. The  
1121 complexity comes from standard Transformer-based architecture, on top of which the coefficient  
1122 of is multiplied for Monte-Carlo estimation, making it irrelevant of diffusion steps. The inference  
1123 complexity remains unaffected.

## 1125 B.5 ALGORITHM DETAILS

1127 For completeness, we include the entire training algorithm in Alg. 1. We also include the MC  
1128 estimator (Alg. 2) to compute the trajectory-level reward  $r_t$  used in our reward correlation analysis in  
1129 ablation study. Note that the function  $\text{Constrain}(\mathbf{x}_0^{(i)}, \mathbf{x}_{t-1})$  means setting every unmasked token in  
1130  $\mathbf{x}_{t-1}$  to the same place in  $\mathbf{x}_0^{(i)}$  with the same token value.

1<sup>1</sup><https://huggingface.co/GSAI-ML/LLaDA-8B-Instruct>

2<sup>2</sup><https://huggingface.co/datasets/princeton-nlp/llama3-ultrafeedback>

1134

**Algorithm 2 SDPO Trajectory-level Reward Evaluation**

1135

**Input:** model  $p_\theta$  and  $p_{\text{ref}}$ , number of MC samples  $K$ , reward model  $r(\cdot)$ 

1136

1: **for** step  $t = T, \dots, 0$  **do**

1137

2: Sample  $\mathbf{x}_{t-1} \sim p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_t)$ 

1138

3: Sample  $\{\mathbf{x}_0^{(i)}\}_{i=1}^K$  from  $\hat{p}_{\text{ref}}(\mathbf{x}_0 | \mathbf{x}_t)$ 

1139

4: Estimate the denominator  $d = \mathbb{E}_{p_{\text{ref}}(\mathbf{x}_0 | \mathbf{x}_t)} \left[ \exp \left( \frac{1}{\beta_t} r(\mathbf{x}_0, \mathbf{c}) \right) \right] \approx \frac{1}{K} \sum_{i=1}^K \exp \left( \frac{1}{\beta_t} r(\mathbf{x}_0^{(i)}, \mathbf{c}) \right)$ 

1140

5:  $\tilde{\mathbf{x}}_0^{(i)} \leftarrow \text{Constrain}(\mathbf{x}_0^{(i)}, \mathbf{x}_{t-1})$ 

1141

6: Estimate the numerator  $n = \mathbb{E}_{p'_{\text{ref}}(\mathbf{x}_0 | \mathbf{x}_{t-1}, \mathbf{x}_t)} \left[ \exp \left( \frac{1}{\beta_t} r(\mathbf{x}_0, \mathbf{c}) \right) \right] \approx \frac{1}{K} \sum_{i=1}^K \exp \left( \frac{1}{\beta_t} r(\tilde{\mathbf{x}}_0^{(i)}, \mathbf{c}) \right)$ 

1142

7: Compute the stepwise reward  $r_t = \log \frac{n}{d}$ 

1143

**Return:** chain reward  $\hat{r} = \beta \sum_t r_t$ 

1144

1145

Table 7: Dataset size ablation results. Even in highly limited data settings (&lt; 10% of original dataset), SDPO achieves strong results.

1146

1147

1148

	Pred-Activity	ATAC-Acc	3-mer Corr	App-Log-lik
25k $\rightarrow$ 20k relabeled	5.56	0.40	0.795	-237
50k $\rightarrow$ 20k relabeled	6.02	0.756	0.793	-248
700k $\rightarrow$ 20k relabeled	6.30	0.948	0.900	-246

1149

1150

1151

1152

1153

Table 8: Detailed results on different values of  $\beta$  on DNA design task.

1154

1155

$\beta$	Pred-Activity (median) $\uparrow$	Pred-Activity-std	ATAC-Acc $\uparrow$ (%)
0.10	6.33	0.68	0.84
0.15	6.29	0.68	0.85
0.20	6.26	0.70	0.86
0.25	6.22	0.74	0.89
0.30	6.17	0.74	0.91
0.35	6.15	0.75	0.92
0.40	6.08	0.76	0.92
0.45	6.01	0.85	0.92
0.50	6.00	0.83	0.93
0.55	5.97	0.85	0.94
0.60	5.90	0.84	0.94
0.65	5.78	0.90	0.96
0.70	5.79	0.90	0.95

1156

1157

1158

1159

1160

1161

1162

1163

1164

1165

1166

1167

1168

1169

## C MORE EXPERIMENT RESULTS

1170

1171

1172

### C.1 ABLATION STUDY ON DATASET SIZE

1173

1174

To study the effect of the data quantity on model performance, we perform an additional ablation on the DNA sequence task in Table 7. Our results demonstrate that SDPO can achieve strong results even in highly limited data settings, where the first stage of fine-tuning uses a small random subset of the original training dataset (700k samples). We follow this with two stages of iterative re-labeling and fine-tuning, according to our established setup.

1175

1176

1177

1178

1179

We provide detailed ablation results on different values of  $\beta$  in Table 8 and Table 9 for DNA sequence design and protein inverse folding tasks, respectively. We observe that when  $\beta$  becomes smaller, which indicates less regularized distribution w.r.t. the reference distribution, the model is granted more flexibility in optimization and generally achieves higher reward. Meanwhile, some other metrics such as ATAC-Acc that relates to the stability of the generated sample will tend to drop due to over-optimizing the model.

1180

1181

1182

### C.2 MORE RESULTS ON $\beta$

1183

1184

1185

1186

1187

1188

We provide detailed ablation results on different values of  $\beta$  in Table 8 and Table 9 for DNA sequence design and protein inverse folding tasks, respectively. We observe that when  $\beta$  becomes smaller, which indicates less regularized distribution w.r.t. the reference distribution, the model is granted more flexibility in optimization and generally achieves higher reward. Meanwhile, some other metrics such as ATAC-Acc that relates to the stability of the generated sample will tend to drop due to over-optimizing the model.

1188  
1189Table 9: Detailed results on different values of  $\beta$  on protein inverse folding task.1190  
1191  
1192  
1193  
1194  
1195  
1196  
1197  
1198

beta	Pred-ddG (median) $\uparrow$	Pred-ddG-std	%(ddG>0) (%) $\uparrow$
0.05	1.026	1.001	0.742
0.10	1.119	1.093	0.760
0.25	0.348	1.176	0.569
0.50	-0.058	1.010	0.461
0.75	-0.350	1.215	0.430
1.00	-0.410	1.192	0.418

1199  
1200

## C.3 MORE RESULTS ON PROTEIN INVERSE FOLDING TASK

1201  
1202  
1203  
1204  
1205  
1206  
1207  
1208  
1209

We provide additional experimental results on the protein inverse folding task, where we demonstrate strong performance in both the re-labeling setting (*a.k.a.* iterative labeling) and the re-labeling-free setting (only using the original dataset for finetuning). The results are depicted in Table 10. In this set of experiments, we obtain the runs that achieve better balance between the target reward (Pred-ddG) and other metrics, such as scRMSD and Success rate. In particular, the experiment with re-labeling achieves better Pred-ddG (1.179) compared with the baseline DRAKES Wang et al. (2024) (1.095) while obtaining better scRMSD and matching its success rate of 78.6%. Furthermore, we observe that without iterative relabeling, SDPO can already obtain strong performance, which additionally underscores the robustness of our design.

1210  
1211

Table 10: Additional evaluation on the protein inverse folding task.

1212  
1213  
1214  
1215

Stage	# relabeled samples	Pred-ddG (median)	%(ddG>0)	scRMSD (median)	%(scRMSD)< 2	Success rate
w/o Re-labeling	0	1.165	85.5	0.936	90.5	77.0
w/ Re-labeling	12.8k	1.179	84.8	0.914	92.0	78.6

1216

## C.4 RESPONSE SAMPLES ON LLADA-8B-INSTRUCT

1217  
1218  
1219  
1220  
1221  
1222

Here we present several text samples using the questions from GSM8K Cobbe et al. (2021). It is observed that by employing our SDPO to LLADA-8B-Instruct, the finetuned model exhibits clearer reasoning logic and better presentation of the response.

1223

## Response Samples on GSM8K

1224  
1225  
1226  
1227

**Question** [Few-shot Prompts] Janet’s ducks lay 16 eggs per day. She eats three for breakfast every morning and bakes muffins for her friends every day with four. She sells the remainder at the farmers’ market daily for \$2 per fresh duck egg. How much in dollars does she make every day at the farmers’ market?

1228

## Answer by LLADA-8B-Instruct Nie et al. (2025)

1229  
1230  
1231  
1232

Janet’s ducks lay 16 eggs per day. She eats 3 eggs for breakfast every morning, so she has  $16 - 3 = 13$  eggs left. She bakes 4 eggs for her friends every day, so she has  $13 - 4 = 9$  eggs left. She sells the remaining eggs at the farmers’ market for \$2 per egg, so she makes  $9 * \$2 = \$18$  every day.

1233

#### 18

1234

## Answer by LLADA-8B-Instruct-SDPO (Ours)

1235  
1236

To determine how much Janet makes every day at the farmers’ market, we need to follow these steps:

1237  
1238  
1239

1. Calculate the total number of eggs laid by Janet’s ducks per day.
2. Subtract the number of eggs Janet eats for breakfast and bakes for her friends.
3. Calculate the number of eggs left for sale.
4. Multiply the number of eggs left for sale by the price per egg to find her daily earnings.

1240

Let’s go through these steps:

1241

1. Janet’s ducks lay 16 eggs per day.
2. She eats 3 eggs for breakfast every morning.
3. She bakes 4 eggs for her friends every day.

1242 Now, let's calculate the number of eggs left for sale:  
 1243

$$1244 \quad 16 - 3 - 4 = 9$$

1245 So, Janet has 9 eggs left for sale each day.  
 1246

1247 Next, we multiply the number of eggs left for sale by the price per egg:  
 1248

$$1249 \quad 9 \times 2 = 18$$

1250 Therefore, Janet makes \$18 every day at the farmers' market.  
 1251

1252 The final answer is 1253 18  
 1254

### Response Samples on GSM8K

1255 **Question** *[Few-shot Prompts] Carlos is planting a lemon tree. The tree will cost \$90 to plant.  
 1256 Each year it will grow 7 lemons, which he can sell for \$1.5 each. It costs \$3 a year to water and  
 1257 feed the tree. How many years will it take before he starts earning money on the lemon tree?*

1258 **Answer by LLaDA-8B-Instruct Nie et al. (2025)**

1259 To determine how many years it will take for Carlos to start earning money on the lemon tree, we  
 1260 need to calculate the net profit per year and then see how many years it will take for the net profit  
 1261 to cover the initial cost of planting the tree.

1262 1. \*\*Calculate the net profit per year:\*\*

1263 - Revenue from selling lemons: 7 lemons \* \$1.5 per lemon = \$10.50

1264 - Cost to water and feed the tree: \$3

1265 - Net profit per year: \$10.50 - \$3 = \$7.50

1266 2. \*\*Determine the number of years needed to cover the initial cost:\*\*

1267 - Initial cost of planting the tree: \$90

1268 - Net profit per year: \$7.50

1269 - Number of years needed: \$90 / \$7.50 per year = 12 years

1270 Therefore, it will take Carlos 12 years to start earning money on the lemon tree.

1271 **Answer by LLaDA-8B-Instruct-SDPO (Ours)**

1272 First, let's calculate the annual revenue from selling the lemons:

1273 - The tree grows 7 lemons each year.

1274 - Each lemon sells for \$1.5.

1275 - Annual revenue = 7 lemons \* \$1.5/lemon = \$10.5.

1276 Next, let's consider the annual cost of watering and feeding the tree:

1277 - It costs \$3 per year to water and feed the tree.

1278 Now, let's calculate the net annual profit:

1279 - Net annual profit = Annual revenue - Annual cost

1280 - Net annual profit = \$10.5 - \$3 = \$7.5.

1281 Finally, let's determine how many years it will take for the net annual profit to cover the initial  
 1282 cost of planting the tree:

1283 - Number of years = Initial cost / Net annual profit

1284 - Number of years = \$90 / \$7.5 = 12 years.

1285 Therefore, it will take Carlos 12 years to start earning money on the lemon tree.

## D DISCUSSIONS

1286 **Limitation.** Our framework relies on the reward model while, in practice, such model may be noisy  
 1287 or even harmful, which will potentially lead to undesired consequence.

1288 **Broader impact.** As demonstrated in the paper, our approach can help finetune pretrained discrete  
 1289 diffusion models for better alignment towards certain reward, which can have significant practical  
 1290 impact in various domains, such as natural language modeling and biochemical sciences. Our  
 1291 approach can serve as a critical building block towards designing useful DNA and protein sequences,  
 1292 building helpful and harmless chatbots and even performant and effective large language model  
 1293 agentic systems.

1296 **E THE USE OF LARGE LANGUAGE MODELS**  
1297

1298 We did not use Large Language Models for research ideation and paper writing in this work.  
1299

1300

1301

1302

1303

1304

1305

1306

1307

1308

1309

1310

1311

1312

1313

1314

1315

1316

1317

1318

1319

1320

1321

1322

1323

1324

1325

1326

1327

1328

1329

1330

1331

1332

1333

1334

1335

1336

1337

1338

1339

1340

1341

1342

1343

1344

1345

1346

1347

1348

1349