PREFERENCE-BASED ALIGNMENT OF DISCRETE DIFFUSION MODELS

Umberto Borso[1,2]*, **Davide Paglieri** [2], **Jude Wells**[2], **Tim Rocktäschel**[2]
[1]ETH Zurich, [2]Centre for Artificial Intelligence, University College London

ABSTRACT

Diffusion models (Ho et al., 2020; Song et al., 2020) have achieved state-of-the-art performance across multiple domains (Austin et al., 2021; Watson et al., 2023; Anand & Achim, 2022), with recent advancements extending their applicability to discrete data (Lou et al., 2023; Shi et al., 2024; Campbell et al., 2022; 2024). However, aligning discrete diffusion models with task-specific preferences remains challenging, particularly in scenarios where explicit reward functions are unavailable. In this work, we introduce **D**iscrete **D**iffusion **DPO** (**`D2-DPO`**), the first adaptation of Direct Preference Optimization (DPO) (Rafailov et al., 2024) to discrete diffusion models formulated as continuous-time Markov chains. Our approach derives a novel loss function that directly fine-tunes the generative process using preference data while preserving fidelity to a reference distribution. We validate **`D2-DPO`** on a structured binary sequence generation task, demonstrating that the method effectively aligns model outputs with preferences while maintaining structural validity. Our results highlight that **`D2-DPO`** enables controlled fine-tuning without requiring explicit reward models, making it a practical alternative to reinforcement learning-based approaches. Future research will explore extending **`D2-DPO`** to more complex generative tasks, including language modeling and protein sequence generation, as well as investigating alternative noise schedules, such as uniform noising, to enhance flexibility across different applications.

1 INTRODUCTION

Diffusion models have emerged as powerful generative models, achieving state-of-the-art results in a variety of domains, including image generation (Ho et al., 2020; Song et al., 2020) and molecular design (Watson et al., 2023; Anand & Achim, 2022). While originally formulated in continuous spaces, recent advancements have extended diffusion models to discrete domains (Austin et al., 2021; Campbell et al., 2022), including language modelling (Lou et al., 2023; Shi et al., 2024; Sahoo et al., 2024; Ou et al., 2024), symbolic music composition (Campbell et al., 2022) and biological sequence generation (Campbell et al., 2024). Discrete diffusion models have demonstrated remarkable effectiveness in tasks where autoregressive approaches struggle, particularly in capturing long-range dependencies and modelling global consistency. However, in many applications, generating plausible sequences alone is insufficient. One often seeks to optimize generation with respect to specific task objectives, such as increasing factual accuracy in text generation, generating more harmonious music compositions, or designing protein sequences with improved stability.

To address this challenge, recent works have explored fine-tuning pre-trained discrete diffusion models to optimize task-specific reward functions (Wang et al., 2024). However, explicitly defining a reward function is often infeasible when generation quality depends on subjective or hard-to-quantify criteria. In such cases, experts' feedback can provide valuable guidance: they can qualitatively compare generated candidates and express preferences based on fundamental knowledge of the domain.

Direct Preference Optimization (DPO) has recently emerged as a powerful method for fine-tuning generative models based on preference data, eliminating the need for explicit reward modelling. It has been successfully applied in natural language processing to align model responses with human

[*] Work done at Centre for Artificial Intelligence, University College London. Correspondence to `uborso@student.ethz.ch`

feedback (Rafailov et al., 2024), in text-to-image generation to improve adherence to human aesthetic preferences (Wallace et al., 2024), and in protein design to enhance the stability of generated sequences (Widatalla et al., 2024). Despite its success in autoregressive and continuous generative models, DPO has not been explored for discrete diffusion models, which differ fundamentally in their formulation and training dynamics.

In this work, we introduce **D**iscrete **D**iffusion **DPO** (**D2-DPO**), the first adaptation of DPO to discrete diffusion models. Unlike continuous diffusion models, which leverage score-matching, discrete diffusion models are formulated as Continuous-Time Markov Chains (CTMCs), requiring a different optimization framework. We derive a novel loss function that directly fine-tunes discrete diffusion models using pairwise preference data while preserving fidelity to a reference distribution.

Our key contributions are as follows. Firstly, we introduce **D2-DPO**, a DPO-based optimization framework tailored for CTMCs, enabling preference alignment in discrete diffusion models without requiring an explicit reward function. Secondly, we show that under a masking-state noising process, our preference-based objective simplifies to an intuitive closed-form expression, providing theoretical insights into its effectiveness. Thirdly, we empirically validate **D2-DPO** on a structured sequence generation task, demonstrating that it successfully aligns discrete diffusion models with preferences while maintaining distributional coherence.

2 Background and Notation

2.1 Discrete Diffusion Models

Continuous-Time Markov Chain (CTMC). A CTMC describes a sequence of discrete states $\{x_t\}$ evolving over continuous time $t \in [0, 1]$. The process begins at $t = 0$ with an initial state $x_0 \sim p_0$, and transitions between states occur stochastically governed by a rate matrix $R_t \in \mathbb{R}^{\mathcal{X}\times\mathcal{X}}$. The probability of transitioning from state x_t to x_{t+dt} over an infinitesimal time interval dt is given by:

$$p_{t+dt|t}\,(x_{t+dt}|x_t) = \delta(x_t, x_{t+dt}) + R_t(x_t, x_{t+dt})dt, \tag{1}$$

where δ is the Kronecker delta function, which equals 1 when $x_{t+dt} = x_t$ and 0 otherwise. The off-diagonal elements of the rate matrix, $R_t(j, k) \geq 0$ for $j \neq k$, specify the rate at which probability mass transitions from state j to state k at time t. The diagonal elements $R_t(j, j) = -\sum_{k\neq j} R_t(j, k)$ represent the total rate at which probability mass moves out of state j and are thus negative.

Noising Process. The noising process $q_{t|1}(x_t|x_1)$ progressively perturbs the data distribution $p_1(x) = p_{\text{data}}(x)$ gradually transforming it into the noise prior $p_0(x) = p_{\text{noise}}(x)$ as $t \to 0$. A widely used approach is the masking-state noise process (Shi et al., 2024; Sahoo et al., 2024; Ou et al., 2024) which gradually maps all states $x \in \mathcal{X}$ to a masked state M as $t \to 0$. Under this scheme, the noise prior is $p^{\text{mask}}_{\text{noise}}(x) = \delta\{M, x\}$, and the state space is augmented to $\mathcal{X} \cup \{M\}$. The corresponding transition kernel for this process is given by:

$$q^{\text{mask}}_{t|1}\,(x_t|x_1) = t\delta(x_1, x_t) + (1 - t)\delta(M, x_t). \tag{2}$$

Generative Modelling. To generate samples from $p_{\text{data}}(x)$, we begin by drawing the initial noisy state from the noise prior, $x_0 \sim p_{\text{noise}}(x)$, and then simulate the trajectory $\{x_t\}_{t=0}^{t=1}$ by iteratively applying the transition kernel $p_{t+dt|t}(x_{t+dt}|x_t)$. This process allows the system to evolve towards the target distribution, ensuring that the final state at $t = 1$ is effectively a sample from the clean data distribution, i.e., $x_1 \sim p_{\text{data}}(x)$.

Reconstructing the transition kernel in equation 1 requires knowledge of the rate matrix $R_t(x_t, x_{t+dt})$. Campbell et al. (2024) demonstrate that this matrix can be expressed as an expectation over a simpler conditional rate matrix. Specifically, we can write:

$$R_t(x_t, x_{t+dt}) = \mathbb{E}_{p_{1|t}(x_1|x_t)}\left[R^q_t(x_t, x_{t+dt}|x_1)\right], \tag{3}$$

where $p_{1|t}(x_1|x_t)$ represents the denoising distribution, which we approximate using a neural network $p^\theta_{1|t}(x_1|x_t)$. We define the rate matrix $R^\theta_t(x_t, x_{t+dt})$ by substituting $p^\theta_{1|t}(x_1|x_t)$ into the expectation. The conditional rate matrix $R^q_t(x_t, x_{t+dt}|x_1)$ depends on the chosen noise schedule and is defined as:

$$R_t^q(x_t, x_{t+dt}|x_1) = \frac{\mathrm{ReLU}\left(\partial_t q_{t|1}(x_{t+dt}|x_1) - \partial_t q_{t|1}(x_t|x_1)\right)}{S \cdot q_{t|1}(x_t|x_1)}. \tag{4}$$

2.2 Direct Preference Optimization

Bradley-Terry (BT) Model. We assume access to a dataset of pairwise preferences $\mathcal{P}$ over clean data samples x_1. Each preference is represented as a tuple (x_1^w, x_1^l, c), where $c \in \mathcal{C}$ represents a conditioning variable, x_1^w is the preferred sample, and x_1^l is the less preferred sample. The ranking between samples is assumed to follow an unknown latent reward function $r(c, x_1)$, such that $x_1^w \succ x_1^l \iff r(c, x_1^w) > r(c, x_1^l)$. To model the probability of preferring x_1^w over x_1^l, we adopt the Bradley-Terry (BT) model:

$$p_{\mathrm{BT}}(x_1^w \succ x_1^l|c) = \sigma(r(c, x_1^w) - r(c, x_1^l)), \tag{5}$$

where $\sigma(\cdot)$ is the sigmoid function. Given a dataset of preferences, a parametric reward function can be learned by maximum likelihood estimation:

$$L_{\mathrm{BT}}(\phi) = -\mathbb{E}_{c, x_1^w, x_1^l}\left[\log\sigma\left(r_\phi\left(c, x_1^w\right) - r_\phi\left(c, x_1^l\right)\right)\right]. \tag{6}$$

RLHF. Given a learned reward function $r_\phi(c, x_1)$, RLHF seeks to optimize a conditional generative model $p_\theta(x_1|c)$ such that the expected reward is maximized while maintaining distributional regularization. The objective function takes the form:

$$\max_{p_\theta} \mathbb{E}_{c \sim \mathcal{C}, x_1 \sim p_\theta(x_1|c)}\left[r\left(c, x_1\right)\right] - \beta \mathbb{D}_{\mathrm{KL}}\left[p_\theta\left(x_1|c\right) \,||\, p_{\mathrm{ref}}\left(x_1|c\right)\right]. \tag{7}$$

Here, $p_{\mathrm{ref}}(x_1|c)$ is a reference model, and β controls regularization.

DPO. The optimizer of the RLHF objective in equation 7 can be written as:

$$p_\theta(x_1|c) = p_{\mathrm{ref}}(x_1|c)\exp(r(c, x_1)/\beta)/Z(c), \tag{8}$$

where $Z(c) = \sum_{x_1} p_{\mathrm{ref}}\left(x_1|c\right)\exp\left(r\left(c, x_1\right)/\beta\right)$ is a normalizing factor. Solving for $r(c, x_1)$ and substituting this into Equation equation 6, we obtain the DPO loss function:

$$L_{\mathrm{DPO}}(\theta) = -\mathbb{E}_{c, x_1^w, x_1^l}\left[\log\sigma\left(\beta\log\frac{p_\theta\left(x_1^w|c\right)}{p_{\mathrm{ref}}\left(x_1^w|c\right)} - \beta\log\frac{p_\theta\left(x_1^l|c\right)}{p_{\mathrm{ref}}\left(x_1^l|c\right)}\right)\right]. \tag{9}$$

This formulation eliminates the need for explicit reward modeling, allowing direct optimization of the generative model parameters θ without requiring an RL-based policy update.

3 DPO for Discrete Diffusion Models

To facilitate computations, we approximate the CTMC with a discrete-time representation. We partition the continuous time interval $[0, 1]$ into equally spaced steps t_n with $n \in \{0, ..., N\}$, such that the process is described by a discrete-time Markov chain. Denoting the discrete-time states as $x_n = x_{t_n}$ we express the transition probabilities as

$$p_\theta(x_{n+1}|x_n) = \delta(x_{n+1}, x_n) + R_n^\theta(x_n, x_{n+1})\Delta t. \tag{10}$$

Here, $R_n^\theta(x_n, x_{n+1})$ denotes the time-discretized rate matrix that governs state transitions. Building on the approach of Wallace et al. (2024) we can express the DPO objective in discrete time $L_{\mathrm{DT}}(\theta) =$

$$-\log\sigma\left(\beta N \mathbb{E}_{\substack{n \sim \mathcal{U}\{0,N\} \\ x_n^{w,l} \sim q(x_n|x_N^{w,l}) \\ x_{n+1}^{w,l} \sim q(x_{n+1}|x_n^{w,l}, x_N^{w,l})}}\left[\log\frac{p_\theta(x_{n+1}^w|x_n^w)}{p_{\mathrm{ref}}(x_{n+1}^w|x_n^w)} - \log\frac{p_\theta(x_{n+1}^l|x_n^l)}{p_{\mathrm{ref}}(x_{n+1}^l|x_n^l)}\right]\right) \tag{11}$$

where $q(x_n|x_N)$ is the discrete time equivalent of $q_{t|1}(x_t|x_1)$, and $q(x_{n+1}|x_n, x_N)$ is the discrete time equivalent of equation 14. We omit c for compactness. By substituting the transition probability expansion for rate matrices, and taking the continuous-time limit ($N \to \infty$, $\Delta t \to 0$), the final **D2-DPO** loss for CTMCs is obtained:

$$L_{\texttt{D2-DPO}}(\theta) = -\mathbb{E}_{\substack{(x_1^w, x_1^l)\sim\mathcal{P}, t\sim\mathcal{U}[0,1] \\ x^w\sim q(x_t|x_1^w), x^l\sim q(x_t|x_1^l)}} \log\sigma\left[\beta\, \mathcal{D}^{\theta}_{\text{ref}}(x_t^w|x_1^w) - \beta\, \mathcal{D}^{\theta}_{\text{ref}}(x_t^l|x_1^l)\right] \tag{12}$$

with

$$\mathcal{D}^{\theta}_{\text{ref}}(x_t|x_1) = \sum_{j\neq x_t} R_t^q(x_t, j|x_1) \log \frac{R_t^{\theta}(x_t, j)}{R_t^{\text{ref}}(x_t, j)} + R_t^{\text{ref}}(x_t, j) - R_t^{\theta}(x_t, j)\,, \tag{13}$$

where $R_t^q(x_t, x_{t+dt}|x_1)$ depends on the chosen noise schedule and is defined as per equation 4, while $R_t^{\theta}(x_t, x_{t+dt})$ and $R_t^{\text{ref}}(x_t, x_{t+dt})$ are estimated as per equation 3. We defer the full derivation to Appendix C and the multi-dimensional case to Appendix D. In Appendix E, we show how this objective can be efficiently optimized for the masking-state noise process.

4 Preliminary Experiments

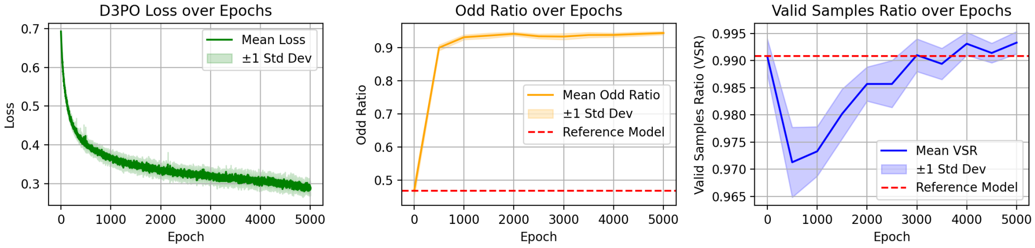

Figure 1: Results for preference-based alignment using the **D2-DPO** loss. (Left) Training loss monotonically decreases over epochs. (Center) Ratio of generated sequences corresponding to odd integers increases w.r.t. reference model. (Right) Fraction of generated sequences with valid structure remains close to 1.

To validate the effectiveness of **D2-DPO**, we conduct a small-scale experiment demonstrating how the proposed loss in Equation equation 12 enables preference alignment in a discrete diffusion model. Building on the framework of Campbell et al. (2024), we first pre-train a masking-state discrete diffusion model to generate structured binary representations of integers. Specifically, each integer $i \in \{0, \ldots, N\}$ is represented as a binary sequence of length N, denoted as $b_i \in 0, 1^N$. The first i bits are set to 1, while the remaining bits are set to 0. The pre-trained model learns to generate valid sequences that adhere to this structured encoding rather than producing arbitrary binary strings.

We then fine-tune the model using our preference-based objective in equation 12 to bias the generative distribution toward binary sequences that represent odd integers. To achieve this, we construct a dataset of pairwise preferences, where the preferred sample x^w corresponds to an odd integer and the less preferred sample x^l corresponds to an even integer. Figure 1 summarizes the fine-tuning process. On the left, the training loss steadily decreases, indicating stable optimization. In the centre, the odd-integer ratio,proportion of generated sequences corresponding to odd integers, rapidly rises above 0.9, confirming model successfully shifts its generative distribution toward odd numbers. On the right, the Valid Samples Ratio (VSR) measures the fraction of generated sequences that correctly follow the structured binary encoding of integers. After an initial dip, the VSR steadily recovers and surpasses the reference baseline, confirming that fine-tuning does not compromise structural validity.

5 Conclusion and Future Work

We introduce Discrete Diffusion DPO (**D2-DPO**), a novel extension of the DPO framework to diffusion models formulate as continuous-time Markov chains. Our derivation yields a computationally efficient loss function that aligns the generative sampling process with preference data while preserving fidelity to the reference distribution. Experiments on a structured binary sequence generation task confirmed that **D2-DPO** successfully biases discrete diffusion models towards preferred outputs while preserving structural validity.

Future work will explore scalability to larger models and more complex sequence generation tasks, such as language modelling and protein design. Additionally, we aim to investigate alternative noise schedules, including the uniform noise schedule, where the prior is a uniform distribution over states, potentially enhancing flexibility in different applications.

References

Namrata Anand and Tudor Achim. Protein structure and sequence generation with equivariant denoising diffusion probabilistic models. *arXiv preprint arXiv:2205.15019*, 2022.

Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne Van Den Berg. Structured denoising diffusion models in discrete state-spaces. *Advances in Neural Information Processing Systems*, 34:17981–17993, 2021.

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

Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. *arXiv preprint arXiv:2305.13301*, 2023.

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

Andrew Campbell, Jason Yim, Regina Barzilay, Tom Rainforth, and Tommi Jaakkola. Generative flows on discrete state-spaces: Enabling multimodal flows with applications to protein co-design, 2024. URL `https://arxiv.org/abs/2402.04997`.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

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

Ying Fan, Olivia Watkins, Yuqing Du, Hao Liu, Moonkyung Ryu, Craig Boutilier, Pieter Abbeel, Mohammad Ghavamzadeh, Kangwook Lee, and Kimin Lee. Reinforcement learning for fine-tuning text-to-image diffusion models. *Advances in Neural Information Processing Systems*, 36, 2024.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

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

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

Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion language modeling by estimating the ratios of the data distribution. 2023.

Hunter Nisonoff, Junhao Xiong, Stephan Allenspach, and Jennifer Listgarten. Unlocking guidance for discrete state-space diffusion and flow models. *arXiv preprint arXiv:2406.01572*, 2024.

Jingyang Ou, Shen Nie, Kaiwen Xue, Fengqi Zhu, Jiacheng Sun, Zhenguo Li, and Chongxuan Li. Your absorbing discrete diffusion secretly models the conditional distributions of clean data. *arXiv preprint arXiv:2406.03736*, 2024.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36, 2024.

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

Anirban Sarkar, Ziqi Tang, Chris Zhao, and Peter Koo. Designing dna with tunable regulatory activity using discrete diffusion. *bioRxiv*, pp. 2024–05, 2024.

Jiaxin Shi, Kehang Han, Zhe Wang, Arnaud Doucet, and Michalis K Titsias. Simplified and generalized masked diffusion for discrete data. *arXiv preprint arXiv:2406.04329*, 2024.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

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

Chenyu Wang, Masatoshi Uehara, Yichun He, Amy Wang, Tommaso Biancalani, Avantika Lal, Tommi Jaakkola, Sergey Levine, Hanchen Wang, and Aviv Regev. Fine-tuning discrete diffusion models via reward optimization with applications to dna and protein design. *arXiv preprint arXiv:2410.13643*, 2024.

Joseph L Watson, David Juergens, Nathaniel R Bennett, Brian L Trippe, Jason Yim, Helen E Eisenach, Woody Ahern, Andrew J Borst, Robert J Ragotte, Lukas F Milles, et al. De novo design of protein structure and function with rfdiffusion. *Nature*, 620(7976):1089–1100, 2023.

Talal Widatalla, Rafael Rafailov, and Brian Hie. Aligning protein generative models with experimental fitness via direct preference optimization. *bioRxiv*, pp. 2024–05, 2024.

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

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3836–3847, 2023.

Huaisheng Zhu, Teng Xiao, and Vasant G Honavar. DSPO: Direct score preference optimization for diffusion model alignment. In *The Thirteenth International Conference on Learning Representations*, 2025. URL `https://openreview.net/forum?id=xyfb9HHvMe`.

Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*, 2019.

A Appendix Structure

The appendix is structured as follows. Appendix B discusses related work, covering advancements in discrete diffusion models, fine-tuning techniques, and preference-based optimization in diffusion models. Appendix C provides a detailed derivation of the **D2-DPO** loss for discrete diffusion models, starting from a discrete-time approximation of the CTMC formulation and extending it to the continuous-time limit. Appendix D generalizes the **D2-DPO** loss to multi-dimensional data, presenting a factorized transition model that enables tractable optimization in structured sequence generation tasks. Appendix E derives the **D2-DPO** loss for the masking noise process, adapting the framework for discrete diffusion models that use an absorbing-state corruption scheme. Appendix E.1 extends the masking noise derivation to cases with additional re-masking noise, allowing for bidirectional transitions between masked and unmasked states. Appendix E.2 provides a complexity analysis of the derived loss functions for the masking state noise process, showing that preference-based fine-tuning with **D2-DPO** is computationally efficient.

B Related Work

Discrete Diffusion Models. Diffusion models have achieved strong generative performance in continuous spaces (Ho et al., 2020; Song et al., 2020), with recent extensions to discrete spaces enabling applications in language modelling and biological sequence design (Austin et al., 2021; Campbell et al., 2022; Lou et al., 2023; Shi et al., 2024; Sahoo et al., 2024; Ou et al., 2024). Compared to autoregressive models, discrete diffusion models better capture long-range dependencies and generate structured sequences such as DNA and protein sequences (Sarkar et al., 2024; Campbell et al., 2024).

Fine-Tuning and Alignment of Discrete Diffusion Models. Fine-tuning diffusion models for controlled generation typically involves guidance techniques, RL-based optimization, or classifier-free methods. Guidance methods such as classifier-based guidance (Dhariwal & Nichol, 2021; Song et al., 2020) have been extended to discrete spaces (Nisonoff et al., 2024), but require costly iterative inference. RL-based fine-tuning has been explored for optimizing reward functions in continuous diffusion models (Fan et al., 2024; Black et al., 2023) and discrete diffusion models (Wang et al., 2024). Classifier-free fine-tuning (Ho & Salimans, 2022; Zhang et al., 2023) conditions on high-reward samples, but is limited by reward sparsity in structured sequence generation. Our work departs from these approaches by proposing preference-based fine-tuning for discrete diffusion models, enabling optimization without an explicit reward model.

Preference-Based Alignment of Diffusion Models. Preference-based optimization methods such as Reinforcement Learning from Human Feedback (RLHF) Ziegler et al. (2019) and Direct Preference Optimization (DPO) (Rafailov et al., 2024) have been highly effective for fine-tuning LLMs and continuous diffusion models. Unlike RL-based methods, DPO directly fine-tunes a model using pairwise preference comparisons, bypassing the need for a reward model (Ethayarajh et al., 2024; Azar et al., 2024). Recent adaptations of DPO to text-to-image diffusion models (Zhu et al., 2025; Wallace et al., 2024; Yang et al., 2024; Li et al., 2024) have shown promising results but are not applicable to discrete diffusion models.

Our work extends DPO to discrete diffusion models, deriving a loss function that respects their underlying CMTC formulation. This enables preference-based fine-tuning without the need of a reward model.

C Full Derivation of 1-Dimensional **D2-DPO** Loss

C.1 Conditional Denoising Kernel.

Here we provide an expression for the infinitesimal transition probability $q_{t+dt|t,1}(x_{t+dt}|x_t, x_1)$ in terms of the conditional rate matrix $R_t^q(x_t, x_{t+dt}|x_1)$ which will be useful later in the derivation of the **D2-DPO** loss.

Given a noise process $q_{t|1}(x_t|x_1)$ we can define the joint probability over two successive states x_t and x_{t+dt} as $q_{t,t+dt|1}(x_t, x_{t+dt}|x_1)$. Using the chain rule of probability:

$$q_{t,t+dt|1}(x_t, x_{t+dt}|x_1) = q_{t|1}(x_t|x_1)q_{t+dt|t,1}(x_{t+dt}|x_t, x_1)$$

where $q_{t+dt|t,1}(x_{t+dt}|x_t, x_1)$ can be interpreted as an infinitesimal denoising probability, conditioned on clean data x_1. Similarly to equation 1, we can write this infinitesimal transition probability in terms of a rate matrix:

$$q_{t+dt|t,1}(x_{t+dt}|x_t, x_1) = \delta(x_t, x_{t+dt}) + R_t^q(x_t, x_{t+dt}|x_1)dt \tag{14}$$

where the conditional rate matrix $R_t^q(x_t, x_{t+dt}|x_1)$ is given as per equation 4.

C.2 Discrete-Time Approximation

We consider a time-discretization of the CTMC to simplify calculations. In practice, we approximate the time evolution of the sequence trajectory $\{x_t\}$ using discrete steps of size Δt, and successively take the limit as $\Delta t \to 0$ to recover the continuous time case. We partition the the time interval $[0, 1]$ with discrete time steps t_n, $n \in \{0, ..., N\}$ where $t_0 = 0$ and $t_N = 1$. We define $\Delta t = t_n - t_{n-1} = 1/(N+1)$ hence recovering the continuous time case when $N \to \infty$. With a slight abuse of notation we write $x_n = x_{t_n}$.

Considering a CTMC with this time partitioning converts the problem into a discrete time Markov Chain with transition kernel $p_\theta(x_{n+1}|x_n)$ which is the time-discrete equivalent of $p^\theta_{t+dt|t}(x_{t+dt} \mid x_t)$ that naturally emerges from equation 1 by identifying $dt = \Delta t$ and evaluating at $t = t_n$. Hence we have:

$$p_\theta(x_{n+1}|x_n) := p^\theta_{t_n+\Delta t|t_n}(x_{t_n+\Delta t}|x_{t_n}) \tag{15}$$

$$= \delta(x_{n+1}, x_n) + R^\theta_n(x_n, x_{n+1})\Delta t \tag{16}$$

Following the Markov assumption we can factorize the joint probability over paths in discrete time

$$p_\theta(x_{0:N}) = p_\theta(x_0) \prod_{n=1}^{N} p_\theta(x_{n+1}|x_n). \tag{17}$$

We define $\mathcal{R}_{\text{DT}}(c, x_{0:N})$ as the reward on the whole trajectory in discrete time, such that we can define $r_{\text{DT}}(c, x_1)$ as:

$$r_{\text{DT}}(c, x_N) = \mathbb{E}_{x_{0:N-1} \sim p_\theta(x_{0:N-1}|x_N, c)} \mathcal{R}_{\text{DT}}\left(c, x_{0:N}\right) \tag{18}$$

C.3 RLHF loss for discrete diffusion models

Now our derivation proceeds along the lines of Wallace et al. (2024), who derive a DPO loss function for classical diffusion models in discrete time. The RLHF objective in Eq. equation 7 can be adapted to the diffusion framework as:

$$\begin{aligned}
&\max_{p_\theta} \mathbb{E}_{x_N \sim p_\theta(x_N|c)}[r_{\text{DT}}\left(c, x_N\right)] - \beta \mathbb{D}_{\text{KL}}\left[p_\theta\left(x_N \mid c\right) \| p_{\text{ref}}\left(x_N \mid c\right)\right] \\
=& \min_{p_\theta} -\mathbb{E}_{x_N \sim p_\theta(x_N|c)}[r_{\text{DT}}\left(c, x_N\right)] + \beta \mathbb{D}_{\text{KL}}\left[p_\theta\left(x_N \mid c\right) \| p_{\text{ref}}\left(x_N \mid c\right)\right] \\
\leq& \min_{p_\theta} -\mathbb{E}_{x_N \sim p_\theta(x_N|c)}[r_{\text{DT}}\left(c, x_N\right)] + \beta \mathbb{D}_{\text{KL}}\left[p_\theta\left(x_{0:N} \mid c\right) \| p_{\text{ref}}\left(x_{0:N} \mid c\right)\right] \\
=& \min_{p_\theta} -\mathbb{E}_{x_{0:N} \sim p_\theta(x_{0:N}|c)}[\mathcal{R}_{\text{DT}}\left(c, x_{0:N}\right)] + \beta \mathbb{D}_{\text{KL}}\left[p_\theta\left(x_{0:N} \mid c\right) \| p_{\text{ref}}\left(x_{0:N} \mid c\right)\right] \\
=& \min_{p_\theta} -\mathbb{E}_{x_{0:N} \sim p_\theta(x_{0:N}|c)}[\mathcal{R}_{\text{DT}}\left(c, x_{0:N}\right)] + \beta \mathbb{E}_{x_{0:N} \sim p_\theta(x_{0:N}|c)}\left[\log \frac{p_\theta\left(x_{0:N} \mid c\right)}{p_{\text{ref}}\left(x_{0:N} \mid c\right)}\right] \\
=& \min_{p_\theta} \mathbb{E}_{x_{0:N} \sim p_\theta(x_{0:N}|c)}\left[\log \frac{p_\theta\left(x_{0:N} \mid c\right)}{p_{\text{ref}}\left(x_{0:N} \mid c\right) \exp(\mathcal{R}_{\text{DT}}\left(c, x_{0:N}\right)/\beta)}\right] \\
=& \min_{p_\theta} \mathbb{E}_{x_{0:N} \sim p_\theta(x_{0:N}|c)}\left[\log \frac{p_\theta\left(x_{0:N} \mid c\right)}{p_{\text{ref}}\left(x_{0:N} \mid c\right) \exp(\mathcal{R}_{\text{DT}}\left(c, x_{0:N}\right)/\beta)/Z(c)} + \log Z(c)\right] \\
=& \min_{p_\theta} \mathbb{D}_{\text{KL}}\left[p_\theta\left(x_{0:N} \mid c\right) || p_{\text{ref}}\left(x_{0:N} \mid c\right) \exp(\mathcal{R}_{\text{DT}}\left(c, x_{0:N}\right)/\beta)/Z(c)\right]
\end{aligned}$$

where $c \sim \mathcal{C}$, and on the third line we used the joint KL-divergence $\mathbb{D}_{\text{KL}}[p_\theta(x_{0:N} \mid c) \| p_{\text{ref}}(x_{0:N} \mid c)]$ as upper bound of the marginal $\mathbb{D}_{\text{KL}}[p_\theta(x_N \mid c) \| p_{\text{ref}}(x_N \mid c)]$. The unique global solution to this optimisation problem is given by:

$$p^*_\theta\left(x_{0:N} \mid c\right) = p_{\text{ref}}\left(x_{0:N} \mid c\right) \exp(\mathcal{R}_{\text{DT}}\left(c, x_{0:N}\right)/\beta)/Z(c) \quad ,$$

Hence we can re-parametrize the reward function as:

$$\mathcal{R}_{\text{DT}}\left(c, x_{0:N}\right) = \beta \log \frac{p^*_\theta\left(x_{0:N} \mid c\right)}{p_{\text{ref}}\left(x_{0:N} \mid c\right)} + \beta \log Z(c)$$

which leads to:

$$\begin{aligned} r_{\text{DT}}(c, x_N) &= \mathbb{E}_{x_{0:N-1} \sim p_\theta(x_{0:N-1} | x_N, c)} \mathcal{R}_{\text{DT}}\left(c, x_{0:N}\right) && (19) \\ &= \beta \mathbb{E}_{x_{0:N-1} \sim p_\theta(x_{0:N-1} | x_N, c)} \left[\log \frac{p_\theta^*\left(x_{0:N} \mid c\right)}{p_{\text{ref}}\left(x_{0:N} \mid c\right)} \right] + \beta \log Z(c) && (20) \end{aligned}$$

C.4 `D2-DPO` Loss

We can substitute equation 20 into the BT model loss in equation 6 to get the *per-example* DPO loss in the discrete time approximation:

$$\begin{aligned} L_{\text{DT}}(\theta) &= -\log \sigma \left(\beta \mathbb{E}_{\substack{x^w_{0:N-1} \sim p_\theta(x^w_{0:N-1} | x^w_N) \\ x^l_{0:N-1} \sim p_\theta(x^l_{0:N-1} | x^l_N)}} \left[\log \frac{p_\theta(x^w_{0:N})}{p_{\text{ref}}(x^w_{0:N})} - \log \frac{p_\theta(x^l_{0:N})}{p_{\text{ref}}(x^l_{0:N})} \right] \right) \\ &= -\log \sigma \left(\beta \mathbb{E}_{\substack{x^w_{0:N-1} \sim p_\theta(x^w_{0:N-1} | x^w_N) \\ x^l_{0:N-1} \sim p_\theta(x^l_{0:N-1} | x^l_N)}} \left[\sum_{n=0}^{N-1} \log \frac{p_\theta(x^w_{n+1} | x^w_n)}{p_{\text{ref}}(x^w_{n+1} | x^w_n)} - \log \frac{p_\theta(x^l_{n+1} | x^l_n)}{p_{\text{ref}}(x^l_{n+1} | x^l_n)} \right] \right) \\ &= -\log \sigma \left(\beta \mathbb{E}_{\substack{x^w_{0:N-1} \sim p_\theta(x^w_{0:N-1} | x^w_N) \\ x^l_{0:N-1} \sim p_\theta(x^l_{0:N-1} | x^l_N)}} N \mathbb{E}_n \left[\log \frac{p_\theta(x^w_{n+1} | x^w_n)}{p_{\text{ref}}(x^w_{n+1} | x^w_n)} - \log \frac{p_\theta(x^l_{n+1} | x^l_n)}{p_{\text{ref}}(x^l_{n+1} | x^l_n)} \right] \right) \end{aligned}$$

where we omit c for simplicity. Since sampling from the reverse process $p_\theta(x_{0:N-1} \mid x_N)$ is intractable, we approximate it with the forward process $q(x_{0:N-1} \mid x_N)$:

$$\begin{aligned} L_{\text{DT}}(\theta) &= -\log \sigma \left(\beta \mathbb{E}_{\substack{x^w_{0:N-1} \sim q(x^w_{0:N-1} | x^w_N) \\ x^l_{0:N-1} \sim q(x^l_{0:N-1} | x^l_N)}} N \mathbb{E}_n \left[\log \frac{p_\theta(x^w_{n+1} | x^w_n)}{p_{\text{ref}}(x^w_{n+1} | x^w_n)} - \log \frac{p_\theta(x^l_{n+1} | x^l_n)}{p_{\text{ref}}(x^l_{n+1} | x^l_n)} \right] \right) \\ &= -\log \sigma \left(\beta N \mathbb{E}_n \mathbb{E}_{\substack{x^w_{n+1,n} \sim q(x_{n+1,n} | x^w_N) \\ x^l_{n+1,n} \sim q(x_{n+1,n} | x^l_N)}} \left[\log \frac{p_\theta(x^w_{n+1} | x^w_n)}{p_{\text{ref}}(x^w_{n+1} | x^w_n)} - \log \frac{p_\theta(x^l_{n+1} | x^l_n)}{p_{\text{ref}}(x^l_{n+1} | x^l_n)} \right] \right) \end{aligned}$$

Using the chan rule we write $q(x_{n+1,n} | x_N) = q(x_n | x_N) q(x_{n+1} | x_n, x_N)$, where $q(x_n | x_N)$ is the discrete time equivalent of $q_{t|1}(x_t | x_1)$, and $q(x_{n+1} | x_n, x_N)$ is the discrete time equivalent of equation 14. Hence we get:

$$\begin{aligned} L_{\text{DT}}(\theta) = -\log \sigma \Bigg(& \beta N \mathbb{E}_n \mathbb{E}_{\substack{x^w_n \sim q(x_n | x^w_N) \\ x^w_{n+1} \sim q(x_{n+1} | x^w_n, x^w_N)}} \left[\log \frac{p_\theta(x^w_{n+1} | x^w_n)}{p_{\text{ref}}(x^w_{n+1} | x^w_n)} \right] \\ & - \mathbb{E}_{\substack{x^l_n \sim q(x_n | x^l_N) \\ x^l_{n+1} \sim q(x_{n+1} | x^l_n, x^l_N)}} \left[\log \frac{p_\theta(x^l_{n+1} | x^l_n)}{p_{\text{ref}}(x^l_{n+1} | x^l_n)} \right] \Bigg) \qquad (21) \end{aligned}$$

Following Campbell et al. (2022) we will expand the expression for $\mathbb{E}_{x_{n+1} \sim q(x_{n+1} | x_n, x_N)} \left[\log \frac{p_\theta(x_{n+1} | x_n)}{p_{\text{ref}}(x_{n+1} | x_n)} \right]$ starting from $\log p_\theta(x_{n+1} | x_n)$:

$$\begin{aligned} \log p_\theta(x_{n+1} | x_n) &= \log(\delta_{x_n, x_{n+1}} + R^\theta_n(x_n, x_{n+1}) \Delta t) \\ &= \delta_{x_n, x_{n+1}} \log(1 + R^\theta_n(x_n, x_n) \Delta t) + (1 - \delta_{x_n, x_{n+1}}) \log(R^\theta_n(x_n, x_{n+1}) \Delta t) \\ &= \delta_{x_n, x_{n+1}} R^\theta_n(x_n, x_n) \Delta t + (1 - \delta_{x_n, x_{n+1}}) \log(R^\theta_n(x_n, x_{n+1}) \Delta t) \end{aligned}$$

where on the last line we used $\log(1+z) = z - \frac{z^2}{2} + o\left(z^2\right)$ which is valid for $|z| \leq 1, z \neq -1$. For any finite $R^\theta_n\left(x_n, x_n\right)$, Δt can be taken small enough such that the series expansion holds. Next we look at the expectation of this expression with respect to the distribution $q(x_{n+1}|x_n, x_N)$:

$$\begin{aligned}
&\mathbb{E}_{x_{n+1}\sim q(x_{n+1}|x_n,x_N)}[\log p_\theta(x_{n+1}|x_n)] = \\
&= \sum_{x_{n+1}} (\delta_{x_n,x_{n+1}} + R^q_n(x_n, x_{n+1}|x_N)\Delta t)\Big[\delta_{x_n,x_{n+1}} R^\theta_n(x_n, x_n)\Delta t + \\
&\qquad\qquad\qquad\qquad (1 - \delta_{x_n,x_{n+1}}) \log(R^\theta_n(x_n, x_{n+1})\Delta t)\Big] \\
&= \delta_{x_n,x_{n+1}}(1 + R^q_n(x_n, x_{n+1}|x_N)\Delta t)R^\theta_n(x_n, x_n)\Delta t + \\
&\qquad\qquad \sum_{x_{n+1}\neq x_n} R^q_n(x_n, x_{n+1}|x_N)\Delta t \; \log R^\theta_n(x_n, x_{n+1})\Delta t \\
&= R^\theta_n(x_n, x_n)\Delta t + R^q_n(x_n, x_{n+1}|x_N)R^\theta_n(x_n, x_n)(\Delta t)^2 + \\
&\qquad\qquad \sum_{x_{n+1}\neq x_n} R^q_n(x_n, x_{n+1}|x_N)\Delta t \; \log R^\theta_n(x_n, x_{n+1})\Delta t \\
&= R^\theta_n(x_n, x_n)\Delta t + \sum_{x_{n+1}\neq x_n} R^q_n(x_n, x_{n+1}|x_N)\Delta t \; \log R^\theta_n(x_n, x_{n+1})\Delta t + o(\Delta t) \\
&= o(\Delta t) + \Delta t \sum_{x_{n+1}\neq x_n} R^q_n(x_n, x_{n+1}|x_N) \; \log R^\theta_n(x_n, x_{n+1})\Delta t - R^\theta_n(x_n, x_{n+1})
\end{aligned}$$

where $R^q_n(x_n, x_{n+1}|x_N)$ is the rate matrix associated with the transition kernel $q(x_{n+1}|x_n, x_N)$. When considering a discrete approximation of continuous time, i.e. $\Delta t \to 0$, $o(\Delta t)$ represents higher-order corrections (terms that vanish faster than Δt). Hence when considering the limit $\Delta t \to 0$ these terms can be ignored, leading to

$$\begin{aligned}
&\mathbb{E}_{x_{n+1}\sim q(x_{n+1}|x_n,x_N)}[\log p_\theta(x_{n+1}|x_n)] = \\
&\Delta t \sum_{x_{n+1}\neq x_n} R^q_n(x_n, x_{n+1}|x_N) \; \log R^\theta_n(x_n, x_{n+1})\Delta t - R^\theta_n(x_n, x_{n+1})
\end{aligned}$$

Now we use this expression to write:

$$\begin{aligned}
&\mathbb{E}_{x_{n+1}\sim q(x_{n+1}|x_n,x_N)} \left[\log \frac{p_\theta(x_{n+1}|x_n)}{p_{\text{ref}}(x_{n+1}|x_n)}\right] \\
&= \mathbb{E}_{x_{n+1}\sim q(x_{n+1}|x_n,x_N)}[\log p_\theta(x_{n+1}|x_n)] - \mathbb{E}_{x_{n+1}\sim q(x_{n+1}|x_n,x_N)}[p_{\text{ref}}(x_{n+1}|x_n)] \\
&= \Delta t \sum_{x_{n+1}\neq x_n} R^q_n(x_n, x_{n+1}|x_N) \; \log \frac{R^\theta_n(x_n, x_{n+1})}{R^{\text{ref}}_n(x_n, x_{n+1})} + R^{\text{ref}}_n(x_n, x_{n+1}) - R^\theta_n(x_n, x_{n+1})
\end{aligned}$$

Plugging this expression into the DPO loss $L_{\text{DT}}(\theta)$ we get:

$$\begin{aligned}
&L_{\text{DT}}(\theta) = -\log\sigma\Bigg[\beta \sum_{n=0}^{N} \mathbb{E}_{\substack{x^w_n\sim q(x_n|x^w_N)\\ x^l_n\sim q(x_n|x^l_N)}} \\
&\Delta t \left(\sum_{x_{n+1}\neq x^w_n} R^\theta_n(x^l_n, x_{n+1}|x_N) \; \log \frac{R^\theta_n(x^w_n, x_{n+1})}{R^{\text{ref}}_n(x^w_n, x_{n+1})} + R^{\text{ref}}_n(x^w_n, x_{n+1}) - R^\theta_n(x^w_n, x_{n+1})\right) \\
&-\Delta t \left(\sum_{x_{n+1}\neq x^l_n} R^\theta_n(x^l_n, x_{n+1}|x_N) \; \log \frac{R^\theta_n(x^l_n, x_{n+1})}{R^{\text{ref}}_n(x^l_n, x_{n+1})} + R^{\text{ref}}_n(x^l_n, x_{n+1}) - R^\theta_n(x^l_n, x_{n+1})\right)\Bigg]
\end{aligned}$$

Taking the limit of the discrete time loss $L_{\text{DT}}(\theta)$ as $N \to \infty$ (and hence $\Delta t = 1/N \to 0$) we get back to the continuous time case:

$$
\begin{aligned}
L_{\mathrm{CT}}(\theta) = \lim_{\substack{N\to\infty\\ \Delta t\to 0}} L_{\mathrm{DT}}(\theta) = -\log\sigma\Bigg[\beta\mathbb{E}_{\substack{x_n^w\sim q(x_n|x_N^w)\\ x_n^l\sim q(x_n|x_N^l)}}\int_0^1 dt \\
\Bigg(\sum_{x_{n+1}\neq x_n^w} R_n^\theta(x_n^l, x_{n+1}|x_N)\ \log\frac{R_n^\theta(x_n^w, x_{n+1})}{R_n^{\mathrm{ref}}(x_n^w, x_{n+1})} + R_n^{\mathrm{ref}}(x_n^w, x_{n+1}) - R_n^\theta(x_n^w, x_{n+1}) \\
- \sum_{x_{n+1}\neq x_n^l} R_n^\theta(x_n^l, x_{n+1}|x_N)\ \log\frac{R_n^\theta(x_n^l, x_{n+1})}{R_n^{\mathrm{ref}}(x_n^l, x_{n+1})} + R_n^{\mathrm{ref}}(x_n^l, x_{n+1}) - R_n^\theta(x_n^l, x_{n+1})\Bigg)\Bigg]
\end{aligned}
$$

We can estimate the integral with Monte Carlo if we consider it to be an expectation with respect to a uniform distribution over times $t \in [0, 1]$.

$$
\begin{aligned}
L_{\mathrm{CT}}(\theta) = -\log\sigma\Bigg[\beta\mathbb{E}_{\substack{t\sim\mathcal{U}[0,1]\\ x_t^w\sim q_{t|1}(x_t|x_1^w)\\ x_t^l\sim q_{t|1}(x_t|x_1^l)}} \\
\Bigg(\sum_{j\neq x_t^w} R_t^q(x_t^w, j|x_1^w)\log\frac{R_t^\theta(x_t^w, j)}{R_t^{\mathrm{ref}}(x_t^w, j)} + R_t^{\mathrm{ref}}(x_t^w, j) - R_t^\theta(x_t^w, j) \\
- \sum_{j\neq x_t^l} R_t^q(x_t^l, j|x_1^l)\log\frac{R_t^\theta(x_t^l, j)}{R_t^{\mathrm{ref}}(x_t^l, j)} + R_t^{\mathrm{ref}}(x_t^l, j) - R_t^\theta(x_t^l, j)\Bigg)\Bigg]
\end{aligned}
$$

Note that $-\log\sigma$ is a convex function and we can apply Jensen's inequality to yield:

$$
L_{\mathrm{CT}}(\theta) \leq -\mathbb{E}_{\substack{t\sim\mathcal{U}[0,1]\\ x_t^w\sim q_{t|1}(x_t|x_1^w)\\ x_t^l\sim q_{t|1}(x_t|x_1^l)}} \log\sigma\Bigg[\beta\,\mathcal{D}_{\mathrm{ref}}^\theta(x_t^w|x_1^w) - \beta\,\mathcal{D}_{\mathrm{ref}}^\theta(x_t^l|x_1^l)\Bigg]
$$

where

$$
\mathcal{D}_{\mathrm{ref}}^\theta(x_t|x_1) = \sum_{j\neq x_t} R_t^q(x_t, j|x_1)\log\frac{R_t^\theta(x_t, j)}{R_t^{\mathrm{ref}}(x_t, j)} + R_t^{\mathrm{ref}}(x_t, j) - R_t^\theta(x_t, j) \quad .
$$

where $R_t^q(x_t, x_{t+dt}|x_1)$ depends on the chosen noise schedule and is defined as per equation 4, while $R_t^\theta(x_t, x_{t+dt})$ and $R_t^{\mathrm{ref}}(x_t, x_{t+dt})$ are estimated as per equation 3.

D MULTI-DIMENSIONAL **D2-DPO**

In this section we adapt the **D2-DPO** loss to account for D-dimensional data. Consider $\boldsymbol{x} \in \{1, \cdots, S\}^D$ is a D-dimensional vector with components x^d where $d = 1, \ldots, D$. We derive the DPO loss for this general case. The derivation proceeds in the same way as for the 1-dimensional case above, up to equation 21. For the D-dimensional case have:

$$
\begin{aligned}
L_{\mathrm{DT}}(\theta) = -\log\sigma\Bigg(\beta N\mathbb{E}_n\mathbb{E}_{\substack{\boldsymbol{x}_n^w\sim q(\boldsymbol{x}_n|\boldsymbol{x}_N^w)\\ \boldsymbol{x}_{n+1}^w\sim q(\boldsymbol{x}_{n+1}|\boldsymbol{x}_n^w,\boldsymbol{x}_N^w)}}\left[\log\frac{p_\theta(\boldsymbol{x}_{n+1}^w|\boldsymbol{x}_n^w)}{p_{\mathrm{ref}}(\boldsymbol{x}_{n+1}^w|\boldsymbol{x}_n^w)}\right] \\
-\mathbb{E}_{\substack{\boldsymbol{x}_n^l\sim q(\boldsymbol{x}_n|\boldsymbol{x}_N^l)\\ \boldsymbol{x}_{n+1}^l\sim q(\boldsymbol{x}_{n+1}|\boldsymbol{x}_n^l,\boldsymbol{x}_N^l)}}\left[\log\frac{p_\theta(\boldsymbol{x}_{n+1}^l|\boldsymbol{x}_n^l)}{p_{\mathrm{ref}}(\boldsymbol{x}_{n+1}^l|\boldsymbol{x}_n^l)}\right]\Bigg)
\end{aligned}
$$

In order to model transitions across multiple dimensions in a single time-step, we consider the following factorization of the transition probability:

$$
p_\theta\left(\boldsymbol{x}_{n+1} \mid \boldsymbol{x}_n\right) = \prod_{d=1}^{D} p_\theta^d\left(x_{n+1}^d \mid \boldsymbol{x}_n\right) \ .
$$

By considering each dimension x_{n+1}^d to be conditionally independent given the current vector $\boldsymbol{x}_n$, we can tractably account for multi-dimensional transitions in a single timestep. Similarly, we factorize

$$q\left(\boldsymbol{x}_{n+1} \mid \boldsymbol{x}_n, \boldsymbol{x}_N\right) = \prod_{d=1}^{D} q^d\left(x_{n+1}^d \mid x_n^d, x_N^d\right) \,,$$

which aligns with the structure of the forward diffusion process, where noise is added independently across dimensions. Using this factorization we can rewrite the expectation terms as:

$$\mathop{\mathbb{E}}_{\substack{\boldsymbol{x}_n \sim q(\boldsymbol{x}_n|\boldsymbol{x}_N) \\ \boldsymbol{x}_{n+1} \sim q(\boldsymbol{x}_{n+1}|\boldsymbol{x}_n,\boldsymbol{x}_N)}} \left[\log \frac{p_\theta(\boldsymbol{x}_{n+1}|\boldsymbol{x}_n)}{p_{\text{ref}}(\boldsymbol{x}_{n+1}|\boldsymbol{x}_n)}\right] = \sum_{d=1}^{D} \mathop{\mathbb{E}}_{\substack{\boldsymbol{x}_n \sim q(\boldsymbol{x}_n|\boldsymbol{x}_N) \\ x_{n+1}^d \sim q^d\left(x_{n+1}^d|x_n^d,x_N^d\right)}} \left[\log \frac{p_\theta^d(x_{n+1}^d|\boldsymbol{x}_n)}{p_{\text{ref}}^d(x_{n+1}^d|\boldsymbol{x}_n)}\right]$$

Proof

$$\begin{aligned}
\mathop{\mathbb{E}}_{\substack{\boldsymbol{x}_n \sim q(\boldsymbol{x}_n|\boldsymbol{x}_N) \\ \boldsymbol{x}_{n+1} \sim q(\boldsymbol{x}_{n+1}|\boldsymbol{x}_n,\boldsymbol{x}_N)}} \left[\log \frac{p_\theta(\boldsymbol{x}_{n+1}|\boldsymbol{x}_n)}{p_{\text{ref}}(\boldsymbol{x}_{n+1}|\boldsymbol{x}_n)}\right] &= \mathop{\mathbb{E}}_{\substack{\boldsymbol{x}_n \sim q(\boldsymbol{x}_n|\boldsymbol{x}_N) \\ \boldsymbol{x}_{n+1} \sim q(\boldsymbol{x}_{n+1}|\boldsymbol{x}_n,\boldsymbol{x}_N)}} \left[\log \frac{\prod_{d=1}^{D} p_\theta^d(x_{n+1}^d|\boldsymbol{x}_n)}{\prod_{d=1}^{D} p_{\text{ref}}^d(x_{n+1}^d|\boldsymbol{x}_n)}\right] \\
&= \sum_{d=1}^{D} \mathop{\mathbb{E}}_{\substack{\boldsymbol{x}_n \sim q(\boldsymbol{x}_n|\boldsymbol{x}_N) \\ \boldsymbol{x}_{n+1} \sim q(\boldsymbol{x}_{n+1}|\boldsymbol{x}_n,\boldsymbol{x}_N)}} \left[\log \frac{p_\theta^d(x_{n+1}^d|\boldsymbol{x}_n)}{p_{\text{ref}}^d(x_{n+1}^d|\boldsymbol{x}_n)}\right] \\
&= \sum_{d=1}^{D} \mathop{\mathbb{E}}_{\substack{\boldsymbol{x}_n \sim q(\boldsymbol{x}_n|\boldsymbol{x}_N) \\ x_{n+1}^d \sim q^d\left(x_{n+1}^d|x_n^d,x_N^d\right)}} \left[\log \frac{p_\theta^d(x_{n+1}^d|\boldsymbol{x}_n)}{p_{\text{ref}}^d(x_{n+1}^d|\boldsymbol{x}_n)}\right]
\end{aligned}$$

Where on the last line we use the fact that the term inside the expectation depends on $\boldsymbol{x}_{n+1}$ only via its d-dimensional component x_{n+1}^d.

Substituting this expression into the DPO loss we get:

$$\begin{aligned}
L_{\text{DT}}(\theta) = -\log\sigma\Bigg(\beta N \mathbb{E}_n \sum_{d=1}^{D} & \mathbb{E}_{\boldsymbol{x}_n^w \sim q(\boldsymbol{x}_n|\boldsymbol{x}_N^w)} \mathbb{E}_{x_{n+1}^d \sim q^d\left(x_{n+1}^d|x_n^d,x_N^{d,w}\right)} \left[\log \frac{p_\theta^d(x_{n+1}^d|\boldsymbol{x}_n^w)}{p_{\text{ref}}^d(x_{n+1}^d|\boldsymbol{x}_n^w)}\right] \\
& - \mathbb{E}_{\boldsymbol{x}_n^l \sim q(\boldsymbol{x}_n|\boldsymbol{x}_N^l)} \mathbb{E}_{x_{n+1}^d \sim q^d\left(x_{n+1}^d|x_n^d,x_N^{d,l}\right)} \left[\log \frac{p_\theta^d(x_{n+1}^d|\boldsymbol{x}_n^l)}{p_{\text{ref}}^d(x_{n+1}^d|\boldsymbol{x}_n^l)}\right]\Bigg)
\end{aligned}$$

We can now follow the same derivation steps as for the 1-dimensional case, leading to:

$$L_{\text{CT}}(\theta) = -\mathbb{E}_{t\sim\mathcal{U}(0,1),\boldsymbol{x}_t^w\sim q(\boldsymbol{x}_t|\boldsymbol{x}_1^w),\boldsymbol{x}_t^l\sim q(\boldsymbol{x}_t|\boldsymbol{x}_1^l)} \log\sigma\left[\beta\sum_{d=1}^{D}\left(\mathcal{D}_{\text{ref}}^{\theta,d}(\boldsymbol{x}_t^w|\boldsymbol{x}_1^w) - \mathcal{D}_{\text{ref}}^{\theta,d}(\boldsymbol{x}_t^l|\boldsymbol{x}_1^l)\right)\right]$$

where

$$\mathcal{D}_{\text{ref}}^{\theta,d}(\boldsymbol{x}_t|\boldsymbol{x}_1) = \sum_{j^d \neq x_t^d} R_t^{d,q}(x_t^d, j^d|x_1^d)\log\frac{R_t^{d,\theta}(\boldsymbol{x}_t, j^d)}{R_t^{d,\text{ref}}(\boldsymbol{x}_t, j^d)} + R_t^{d,\text{ref}}(\boldsymbol{x}_t, j^d) - R_t^{d,\theta}(\boldsymbol{x}_t, j^d) \qquad (22)$$

where $R_t^{d,\theta}(\boldsymbol{x}, j^d) = \mathbb{E}_{p_{1|t}^{d,\theta}(x_1^d|\boldsymbol{x})}[R_t^{d,\theta}(x^d, j^d|x_1^d)]$, and x^d denotes the d-dimensional component of vector $\boldsymbol{x}$.

E D2-DPO Loss for Masking State Models

In this section we adapt the D2-DPO loss for the specific case of masking noise process. In D dimensions we consider independent corruption processes in each dimension, similar to the factorization assumptions made in continuous diffusion models where the forward noising processes proceed independently in each dimension.

$$\begin{aligned} q^{\text{mask}}_{t|1}\left(\boldsymbol{x}_t \mid \boldsymbol{x}_1\right) &= \prod_{d=1}^{D} q^{\text{mask},d}_{t|1}\left(x^d_t \mid x^d_1\right) \\ &= \prod_{d=1}^{D} \left(t\delta\left\{x^d_t, x^d_1\right\} + (1-t)\delta\left\{x^d_t, M\right\}\right) \end{aligned}$$

In this case, the conditional rate matrix for the masking process can be derived in closed form as:

$$\begin{aligned} R^{q,d}_t\left(x^d_t, x^d_{t+dt} \mid x^d_1\right) &= \frac{\operatorname{ReLU}\left(\partial_t q^{\text{mask},d}_{t|1}\left(x^d_{t+dt} \mid x^d_1\right) - \partial_t q^{\text{mask},d}_{t|1}\left(x^d_t \mid x^d_1\right)\right)}{S \cdot q^{\text{mask},d}_{t|1}\left(x^d_t \mid x^d_1\right)} \\ &= \frac{1}{1-t}\delta\left\{x^d_t, M\right\}\delta\left\{x^d_{t+dt}, x^d_1\right\} \end{aligned} \tag{23}$$

We can then express the unconditional rate matrix as:

$$\begin{aligned} R^{d,\theta}_t(\boldsymbol{x}_t, x^d_{t+dt}) &= \mathbb{E}_{p^{\theta}_{1|t}(x^d_1|\boldsymbol{x}_t)}\left[R^{\text{mask},d}_t\left(x^d_t, x^d_{t+dt} \mid x^d_1\right)\right] \\ &= \mathbb{E}_{p^{\theta}_{1|t}(x^d_1|\boldsymbol{x}_t)}\left[\frac{1}{1-t}\delta\left\{x^d_t, M\right\}\delta\left\{x^d_{t+dt}, x^d_1\right\}\right] \\ &= \frac{1}{1-t}\delta\left\{x^d_t, M\right\} p^{\theta}_{1|t}\left(x^d_1 = x^d_{t+dt} \mid \boldsymbol{x}_t\right) \end{aligned} \tag{24}$$

which vanishes for $x^d_t \neq M$ and for $x^d_{t+dt} = M$ as $p^{\theta}_{1|t}\left(x^d_1 = M \mid \boldsymbol{x}_t\right) = 0$, meaning $\boldsymbol{x}_1$ cannot have any masked dimensions. Substituting equation 23 and equation 24 into equation 22:

$$\begin{aligned} \mathcal{D}^{\theta,d}(\boldsymbol{x}_t|\boldsymbol{x}_1) &= \sum_{j^d \neq x^d_t} R^{d,q}_t(x^d_t, j^d|x^d_1)\log\frac{R^{d,\theta}_t(\boldsymbol{x}_t, j^d)}{R^{d,\text{ref}}_t(\boldsymbol{x}_t, j^d)} + R^{d,\text{ref}}_t(\boldsymbol{x}_t, j^d) - R^{d,\theta}_t(\boldsymbol{x}_t, j^d) \\ &= \frac{\delta\{x^d_t, M\}}{1-t}\sum_{j^d \neq M}\delta\{x^d_1, j^d\}\log\frac{p^{\theta}_{1|t}\left(j^d \mid \boldsymbol{x}_t\right)}{p^{\text{ref}}_{1|t}\left(j^d \mid \boldsymbol{x}_t\right)} + p^{\text{ref}}_{1|t}\left(j^d \mid \boldsymbol{x}_t\right) - p^{\theta}_{1|t}\left(j^d \mid \boldsymbol{x}_t\right) \\ &= \frac{\delta\{x^d_t, M\}}{1-t}\log\frac{p^{\theta}_{1|t}\left(x^d_1 \mid \boldsymbol{x}_t\right)}{p^{\text{ref}}_{1|t}\left(x^d_1 \mid \boldsymbol{x}_t\right)} \end{aligned} \tag{25}$$

Where on the last line we use the fact that the neural network $p^{\theta}_{1|t}\left(\cdot \mid \boldsymbol{x}_t\right)$ outputs a probability distribution over all unmasked tokens to write $\sum_{j^d \neq M} p^{\text{ref}}_{1|t}\left(x^d_1 = j^d \mid \boldsymbol{x}_t\right) = \sum_{j^d \neq M} p^{\theta}_{1|t}\left(x^d_1 = j^d \mid \boldsymbol{x}_t\right) = 1$. Hence the final loss is:

$$\begin{aligned} L^{\text{mask}}_{\text{CT}}(\theta) = -\mathbb{E}_{\substack{t\sim\mathcal{U}[0,1] \\ \boldsymbol{x}^w_t \sim q(\boldsymbol{x}_t|\boldsymbol{x}^w_1) \\ \boldsymbol{x}^l_t \sim q(\boldsymbol{x}_t|\boldsymbol{x}^l_1)}} \log\sigma\Bigg[\frac{\beta}{1-t}\sum_{d=1}^{D} \\ \left(\delta\{x^{d,w}_t, M\}\log\frac{p^{\theta}_{1|t}\left(x^{d,w}_1 \mid \boldsymbol{x}^w_t\right)}{p^{\text{ref}}_{1|t}\left(x^{d,w}_1 \mid \boldsymbol{x}^w_t\right)} - \delta\{x^{d,l}_t, M\}\log\frac{p^{\theta}_{1|t}\left(x^{d,l}_1 \mid \boldsymbol{x}^l_t\right)}{p^{\text{ref}}_{1|t}\left(x^{d,l}_1 \mid \boldsymbol{x}^l_t\right)}\right)\Bigg] \end{aligned}$$

Similar to the classical DPO loss in equation 9, this loss is based on the difference in log probabilities assigned to recovering the original samples under the learned model $p^{\theta}_{1|t}$ compared to a reference model $p^{\text{ref}}_{1|t}$. However, this difference is weighted by a masking indicator, ensuring that only masked dimensions contribute to the loss. Intuitively, the effect of optimizing this objective is to increase the model's likelihood of reconstructing the preferred sample x^w while reducing the likelihood of reconstructing the dis-preferred sample x^l, making x^w more likely to be recovered during the unmasking process.

E.1 Masking with Additional Uniform Noise

We now consider the case in which we introduce a non-zero probability to transition from an unmasked state back to a masked state during the denoising process. Intuitively this allows more flexibility at inference time as the model could potentially recover from errors by re-masking certain tokes. Campbell et al. (2024) show that such an additional noise process is in detailed balance with the noise-free process and hence does not affect the final data distribution at time $t = 1$. They also show that the resulting rate matrix for a noise process with coefficient η is given by:

$$\begin{aligned}
R_t^{d,\theta}(\boldsymbol{x}_t, j^d) &= \frac{1+\eta t}{1-t} p^{\theta}_{1|t}\left(x_1^d = j^d \mid \boldsymbol{x}_t\right) \delta\left\{x_t^d, M\right\} + \eta\left(1 - \delta\left\{x_t^d, M\right\}\right) \delta\left\{j^d, M\right\} \\
&= \begin{cases} \frac{1+\eta t}{1-t} p^{\theta}_{1|t}\left(x_1^d = j^d \mid \boldsymbol{x}_t\right) & \text{for } x_t^d = M, j^d \neq M \\ \eta & \text{for } x_t^d \neq M, j^d = M \\ 0 & \text{otherwise} \end{cases}
\end{aligned}$$

While $R_t^{d,q}(x_t^d, x_{t+dt}^d | x_1^d)$ remains unaffected. Substituting this into equation 22:

$$\begin{aligned}
\mathcal{D}_t^{\theta,d}(\boldsymbol{x}) &= \sum_{j^d \neq x^d} R_t^{d,q}(x_t^d, j^d | x_1^d) \log \frac{R_t^{d,\theta}(\boldsymbol{x}, j^d)}{R_t^{d,\text{ref}}(\boldsymbol{x}, j^d)} + R_t^{d,\text{ref}}(\boldsymbol{x}, j^d) - R_t^{d,\theta}(\boldsymbol{x}, j^d) \\
&= \frac{1+\eta t}{1-t} \delta\{x_t^d, M\} \log \frac{p^{\theta}_{1|t}\left(x_1^d \mid \boldsymbol{x}_t\right)}{p^{\text{ref}}_{1|t}\left(x_1^d \mid \boldsymbol{x}_t\right)} + \left(1 - \delta\left\{x_t^d, M\right\}\right) \delta\left\{j^d, M\right\} \left(\eta \log \frac{\eta}{\eta} + \eta - \eta\right) \\
&= \frac{1+\eta t}{1-t} \delta\{x_t^d, M\} \log \frac{p^{\theta}_{1|t}\left(x_1^d \mid \boldsymbol{x}_t\right)}{p^{\text{ref}}_{1|t}\left(x_1^d \mid \boldsymbol{x}_t\right)} \qquad (26)
\end{aligned}$$

which is the same as for the noiseless reverse process, up to a multiplicative constant $1 + \eta t$. Hence the final loss is:

$$\begin{aligned}
L_{\text{CT}}^{\text{mask}}(\theta) = -\mathbb{E}_{\substack{t\sim\mathcal{U}[0,1] \\ \boldsymbol{x}_t^w \sim q(\boldsymbol{x}_t | \boldsymbol{x}_1^w) \\ \boldsymbol{x}_t^l \sim q(\boldsymbol{x}_t | \boldsymbol{x}_1^l)}} \log \sigma \Bigg[\frac{\beta(1+\eta t)}{1-t} \sum_{d=1}^{D} \\
\left(\delta\{x_t^{d,w}, M\} \log \frac{p^{\theta}_{1|t}\left(x_1^{d,w} \mid \boldsymbol{x}_t^w\right)}{p^{\text{ref}}_{1|t}\left(x_1^{d,w} \mid \boldsymbol{x}_t^w\right)} - \delta\{x_t^{d,l}, M\} \log \frac{p^{\theta}_{1|t}\left(x_1^{d,l} \mid \boldsymbol{x}_t^l\right)}{p^{\text{ref}}_{1|t}\left(x_1^{d,l} \mid \boldsymbol{x}_t^l\right)} \right) \Bigg]
\end{aligned}$$

E.2 Complexity Analysis for Masking Noise Process

For the masking noise process, the derived expressions for $\mathcal{D}_t^{\theta,d}(\boldsymbol{x})$ in Equations equation 25 and equation 26 provide a computationally efficient way to estimate the **D2-DPO** loss function. In practice, the denoising models $p^{\theta}_{1|t}$ and $p^{\text{ref}}_{1|t}$ take as input a noisy vector $\boldsymbol{x}_t \in \{1, \dots, S, M\}^D$ and output probability vectors $p_{1|t}(\boldsymbol{x}_1 \mid \boldsymbol{x}_t) \in [0,1]^D$. Since the loss function requires evaluating the probability of reconstructing each dimension x_1^d, this can be directly accessed as the d^{th} component of the model's output.

Due to the structure of the masking noise process, computing the sum $\sum_{d=1}^{D} \mathcal{D}_t^{\theta,d}(\boldsymbol{x})$ is particularly efficient. The required probability vectors $p^{\theta}_{1|t}(\boldsymbol{x}_1 \mid \boldsymbol{x}_t)$ and $p^{\text{ref}}_{1|t}(\boldsymbol{x}_1 \mid \boldsymbol{x}_t)$ can be obtained with a single forward pass for each model. As a result, evaluating $\sum_{d=1}^{D} \mathcal{D}_t^{\theta,d}(\boldsymbol{x})$ requires exactly two model queries: one for the learned model $p^{\theta}_{1|t}$ and one for the reference model $p^{\text{ref}}_{1|t}$.

When estimating the *per-example* **D2-DPO** loss using a batch of size T to approximate the expectation over $t \sim \mathcal{U}[0,1]$, the total number of model queries scales to $2T = O(T)$. For a dataset containing P preference pairs, the overall computational complexity becomes $O(PT)$, reflecting a linear dependence on both the number of preferences and batch size. This scaling ensures that preference optimization in discrete diffusion models remains computationally efficient, making it practical for large-scale generative modeling tasks.