Fine-Tuning Discrete Diffusion Models with Policy Gradient Methods

Oussama Zekri
CREST, ENSAE
Institut Polytechnique de Paris
France
oussama.zekri@ensae.fr

Nicolas Boullé
Department of Mathematics
Imperial College London
United Kingdom
n.boulle@imperial.ac.uk

Abstract

Discrete diffusion models have recently gained significant attention due to their ability to process complex discrete structures for language modeling. However, fine-tuning these models with policy gradient methods, as is commonly done in Reinforcement Learning from Human Feedback (RLHF), remains a challenging task. We propose an efficient, broadly applicable, and theoretically justified policy gradient algorithm, called Score Entropy Policy Optimization (SEPO), for fine-tuning discrete diffusion models over non-differentiable rewards. Our numerical experiments across several discrete generative tasks demonstrate the scalability and efficiency of our method. Our code is available at https://github.com/ozekri/SEPO.

1 Introduction

Diffusion models have become efficient generative modeling tools in various tasks, including image and video generation (Song et al., 2021; Ho et al., 2020). Although most of the applications of diffusion models depend on a continuous state space (such as images), recent works extended these models to discrete settings, enabling their use in language modeling and other discrete generative tasks (Sun et al., 2023; Campbell et al., 2022; Austin et al., 2021; Benton et al., 2024). Moreover, several studies showed that these models can be competitive with autoregressive models, such as GPT (Brown et al., 2020) or Llama (Touvron et al., 2023), while allowing for more flexible generation compared to next-token prediction (Lou et al., 2024; Sahoo et al., 2024; Shi et al., 2024). These discrete diffusion models are promising if they can be scaled up to natural language processing tasks.

However, fine-tuning discrete diffusion models is challenging. Existing strategies, such as classifier guidance (Ho and Salimans, 2021; Nisonoff et al., 2025; Gruver et al., 2024) and steering (Rector-Brooks et al., 2025), often face scalability issues or intractable objectives. We focus instead on reinforcement learning (RL), which optimizes a pre-trained model via rewards. A key challenge is that sampling from categorical distributions is non-differentiable and incompatible with standard gradient methods. Recent work (Wang et al., 2025) addresses this using the Gumbel-Softmax trick (Jang et al., 2017), but requires differentiable rewards and becomes memory-intensive at scale. In contrast, policy gradient methods like PPO (Schulman et al., 2017) and GRPO (Shao et al., 2024) rely only on reward evaluations and offer stability, unbiased gradients, and robustness to noise.

We propose a policy gradient algorithm, Score Entropy Policy Optimization (SEPO), designed for discrete diffusion. Unlike Wang et al. (2025), our method handles non-differentiable rewards, enabling broader fine-tuning scenarios. SEPO offers a unified and scalable framework for optimizing discrete diffusion models, that supports both conditional and unconditional generation. Other policy gradient approaches have also been proposed, such as GLIDE (Cao et al., 2025), which relies on reward shaping, and d1 (Zhao et al., 2025), specifically designed for masked diffusion models and cannot

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

handle unconditional generation. These methods are tailored to specific setups, whereas SEPO offers a more general and principled framework for discrete diffusion fine-tuning.

Main contributions.

1) We provide an explicit characterization of policy gradient algorithms for discrete diffusion models in the concrete score framework. This allows the use of non-differentiable rewards in discrete fine-tuning tasks *without* steering or guidance mechanisms, for both **conditional** and **unconditional** generation.
2) We propose an efficient, scalable algorithm based on policy gradient methods (Schulman et al., 2017; Shao et al., 2024), called Score Entropy Policy Optimization (SEPO), for discrete diffusion. It uses a clipped-ratio loss with **self-normalized importance sampling** for low-variance. We also introduce a gradient flow alternative that improves sample quality at a higher complexity.
3) We perform numerical experiments on DNA fine-tuning and natural language tasks to demonstrate the performance of our methods. We achieve state-of-the-art DNA results: **Pred-Activity** 7.64 and **ATAC-Acc** 99.9%, topping prior RL and guidance baselines with lower run-to-run variance.

2 Background and preliminaries

2.1 Related works

Inference-time techniques. Inference-time techniques are simple yet effective as they require no fine-tuning or training when reward functions are available. Recent studies (Singhal et al., 2025; Ma et al., 2025) showed that they can achieve competitive performance by scaling computational resources. Although inference-time techniques offer distinct advantages, they typically result in longer inference times compared to fine-tuned models. The key considerations for these techniques include computational efficiency and differentiability of the reward (Uehara et al., 2025).

Policy gradients algorithms. Policy gradient algorithms are a key class of reinforcement learning methods that optimize parameterized policies by directly maximizing expected returns. Modern implementations include Proximal Policy Optimization (Schulman et al., 2017) or Group Relative Policy Optimization (Shao et al., 2024). These algorithms are highly sensitive to policy design since the architecture impacts expressiveness, optimization stability, and exploration.

Fine-tuning diffusion models with Reinforcement Learning. In the case of continuous diffusion models, fine-tuning via policy gradients has been proposed (Fan et al., 2024; Li et al., 2024; Black et al., 2024; Ren et al., 2025). In a more recent study, (Marion et al., 2024) implements REINFORCE algorithm (Williams, 1992) for continuous diffusion models in a single-loop algorithm, avoiding nested optimization. However, extending these approaches to discrete diffusion models is more challenging. This work adapts these studies to the discrete case and extends them to general policy gradient algorithms.

2.2 Discrete Diffusion

In discrete diffusion models, the dynamics of a single particle is described by a continuous-time Markov chain (CTMC), denoted as a stochastic process $(x_t)_{0\leq t\leq T}$ operating on a finite space $\mathcal{X} = \{\alpha_1, \dots, \alpha_m\}^n$. Here, $(\alpha_i)_{1\leq i\leq m}$ represents the possible states that form a vocabulary of size m, and n is the length of the sequences, which is a fixed number known as *context window* or *block size*. Typically, it describes sequences of tokens or image pixel values. While the size $d := |\mathcal{X}| = m^n$ of $\mathcal{X}$ is exponential in n, deep neural networks such as transformers (Vaswani, 2017) were shown to perform and generalize well on these incredibly large state spaces (Zekri et al., 2024).

Forward process. At any given time t, the distribution of a particle x_t is given by $\boldsymbol{p}_t$, which lies within the probability simplex $\Delta_d \subset \mathbb{R}^d$. The forward process is a noising process that maps the initial data distribution $\boldsymbol{p}_0 := p_{\text{data}}$ to some final noisy distribution $\boldsymbol{p}_T := p_{\text{ref}}$, which is easy to sample. During the noising forward process, the particle's probability transitions between states are given by a rate matrix $Q_t \in \mathbb{R}^{d\times d}$, indexed by $\mathcal{X}$, through the equation $\frac{\mathrm{d}\boldsymbol{p}_t}{\mathrm{d}t} = Q_t \boldsymbol{p}_t$ for $t \in [0, T]$.

The time reversal of this equation (Kelly, 2011) is known as,

$$\frac{\mathrm{d}\boldsymbol{p}_{T-t}}{\mathrm{d}t} = \overline{Q}_{T-t}\boldsymbol{p}_{T-t}, \quad t \in [0,T], \tag{1}$$

where for $x, y \in \mathcal{X}$, $\overline{Q}_t(y,x) = \begin{cases} \frac{\boldsymbol{p}_t(y)}{\boldsymbol{p}_t(x)} Q_t(x,y), & x \neq y, \\ -\sum_{z \neq x} \overline{Q}_t(z,x), & x = y. \end{cases}$

Score entropy. Lou et al. 2024 recently showed that one can approximate Eq. (1) via score entropy, inspired by concrete score matching (Meng et al., 2022). This is done by learning the concrete score as $s_\theta(x,t)_y \approx \boldsymbol{p}_t(y)/\boldsymbol{p}_t(x)$ with a sequence-to-sequence neural network s_θ parametrized by $\theta \in \mathbb{R}^p$. The resulting process is described by the following equation:

$$\frac{\mathrm{d}\boldsymbol{q}_t^\theta}{\mathrm{d}t} = \overline{Q}_{T-t}^\theta \boldsymbol{q}_t^\theta, \quad t \in [0,T], \tag{2}$$

where the denoising process $\boldsymbol{q}_t^\theta \approx \boldsymbol{p}_{T-t}$ maps $\boldsymbol{q}_0^\theta := p_{\mathrm{ref}}$ to $\boldsymbol{q}_T^\theta$, and θ is learned to achieve $\boldsymbol{q}_T^\theta \approx_\theta p_{\mathrm{data}}$. The matrix $\overline{Q}_t^\theta$ is defined for $x, y \in \mathcal{X}$ as $\overline{Q}_t^\theta(x,y) = s_\theta(x,t)_y Q_t(x,y)$ if $x \neq y$ and $\overline{Q}_t^\theta(y,y) = -\sum_{z \neq x} \overline{Q}_t^\theta(z,y)$. In practice, the quantity $s_\theta(x,t)_y$ is available for all $y \in \mathcal{X}$ at Hamming distance (Hamming, 1950) one of x: the states y that differ from x by exactly one token. This represents only $\mathcal{O}(mn)$ ratios instead of $\mathcal{O}(m^{2n})$ (Campbell et al., 2022; Lou et al., 2024).

Masked diffusion models. The score entropy setup also contains the simplified frameworks detailed in (Sahoo et al., 2024; Shi et al., 2024; Ou et al., 2024), often referred to as masked diffusion models. These models define a simplified training loss based on a weighted cross-entropy objective by introducing a special `[MASK]` token, and have been shown to outperform uniformly noised discrete diffusion models. Promising attempts at scaling them to larger settings support this observation (Arriola et al., 2025; Nie et al., 2025; Ye et al., 2025).

2.3 Fine-tuning with Reinforcement Learning

After the pretraining phase, a discrete diffusion model with learned parameter θ_{pre} aims to approximate p_{data}, in the sense $\boldsymbol{q}_T^{\theta_{\mathrm{pre}}} \approx p_{\mathrm{data}}$. Our goal is to fine-tune the denoising process $\boldsymbol{q}_t^\theta$ to increase a reward function $R_t : \mathcal{X} \to \mathbb{R}$, without having access to p_{data}.

Minimization problem. We focus on optimization problems over implicitly parameterized distributions. For a given family of functions $(\mathcal{F}_t)_{t\in[0,T]} : \Delta_d \to \mathbb{R}$, we aim to minimize the loss function

$$\ell_t(\theta) := -\mathcal{F}_t(\boldsymbol{q}_t^\theta), \quad t \in [0,T], \tag{3}$$

over $\theta \in \mathbb{R}^p$. Standard choices of $\mathcal{F}_t$ include $\mathcal{F}_t(\boldsymbol{q}_t^\theta) = \mathbb{E}_{x\sim\boldsymbol{q}_t^\theta}[R_t(x)]$ to maximize a reward function $R_t : \mathcal{X} \to \mathbb{R}$, or $\mathcal{F}_t(\boldsymbol{q}_t^\theta) = -\mathrm{KL}(\boldsymbol{q}_t^\theta \| \boldsymbol{q}_t^{\theta_{\mathrm{pre}}})$ to minimize the KL divergence of p_t from a distribution $\boldsymbol{q}_t^{\theta_{\mathrm{pre}}}$. As detailed in (Uehara et al., 2024), a typical fine-tuning algorithm for diffusion models combines these two terms as

$$\ell_t(\theta) = -\mathbb{E}_{x\sim\boldsymbol{q}_t^\theta}[R_t(x)] + \alpha\,\mathrm{KL}(\boldsymbol{q}_t^\theta \| \boldsymbol{q}_t^{\theta_{\mathrm{pre}}}), \tag{4}$$

where $\alpha > 0$ is a weighting factor. One can recast the denoising process as a finite-horizon multi-step Markov Decision Process formulation with known transition probabilities as folows:

$$\begin{aligned} &\text{For all } t \in \{0,\dots,T\}, \quad \mathcal{S} = \mathcal{X}, \quad \mathcal{A} = \mathcal{X}, \quad \text{Policy } \boldsymbol{q}_t^\theta, \\ &S_t = x_{T-t}, \quad A_t = x_{T-1-t}, \quad \mathbb{P}(S_0) = p_{\mathrm{ref}}, \quad \mathbb{P}(S_{t+1} \mid S_t, A_t) = \delta(\{S_{t+1} = A_t\}), \end{aligned}$$

where $\mathcal{S}$ and $\mathcal{A}$ denote the states and action spaces, S_t and A_t denote the states and actions, and δ is the Dirac delta distribution. Although common choices in the fine-tuning diffusion models with reinforcement learning literature (Black et al., 2024; Fan et al., 2024; Clark et al., 2024; Uehara et al., 2024) often set $R_t = 0$ for $t < T$ and $R_T = R$, we retain the general form of R_t in the results for greater generality. This choice allows for more flexible reward evaluation schemes. For example, one can run fewer denoising steps and stop the process at some intermediate time $T_0 < T$, assigning the reward $R_{T_0} = R$ based on the partially denoised sample. We will consider this case, and the corresponding distribution $\boldsymbol{q}_{T_0}^\theta$ will be denoted by $\boldsymbol{\pi}_\theta$.

Loss reward gradient. To apply first-order optimization methods, one needs to compute the gradient $\nabla_\theta \ell_t(\theta)$. Since $\mathcal{X}$ is a finite space of size d, we have

$$\nabla_\theta \ell_t(\theta) = -\nabla_\theta\big(\mathcal{F}_t(\boldsymbol{q}_t^\theta)\big) = -\mathbf{f}^{\mathrm{T}} \nabla_\theta \boldsymbol{q}_t^\theta, \tag{5}$$

where $\mathbf{f} \in \mathbb{R}^d$ is the vector of first variations $\mathcal{F}_t(\boldsymbol{q}_t^\theta)$ (see Appendix D.1). Importantly, we note that $\mathbf{f}(\boldsymbol{p})(x) = R_t(x)$ for $x \in \mathcal{X}$, which does not involve the differentiability of R_t (with respect to some embedding $\mathcal{E}(\mathcal{X})$ of the state space). One can then design deterministic non-differentiable functions that act on $\mathcal{X}$ as rewards, similar to those arising in RLHF, or elsewhere. This may include designing desired protein properties (Rector-Brooks et al., 2025).

3 Methods

3.1 Policy gradients for concrete score

The gradient of the target distribution $\nabla_\theta \boldsymbol{q}_t^\theta$ in Eq. (5) can be approximated based on its relationship with the concrete score s_θ.

Assumption 3.1. *For $x \neq y \in \mathcal{X}$, define the exact ratio $r_{x,y}^t(\theta) := \boldsymbol{q}_t^\theta(y)/\boldsymbol{q}_t^\theta(x)$. We assume that there exists $\delta > 0$ such that for all $\theta \in \mathbb{R}^p$, $x \neq y \in \mathcal{X}$, $t \in [0, T]$,*

$$\|\nabla_\theta \log s_\theta(x, T-t)_y - \nabla_\theta \log r_{x,y}^t(\theta)\| \leq \delta. \tag{6}$$

The following theorem shows that one can compute $\widehat{\nabla \ell_t}(\theta)$ that estimates $\nabla_\theta \ell_t(\theta)$ through a discrete analogue of the `REINFORCE` algorithm (Williams, 1992).

Theorem 3.1 (Discrete `REINFORCE` trick)**.** *Under Assumption 3.1 and with the notations introduced in Section 2, applying the `REINFORCE` with concrete score approximations yields the following estimator of Eq. (5):*

$$\widehat{\nabla \ell_t}(\theta) = \sum_{x \in \mathcal{X}} \boldsymbol{q}_t^\theta(x) R_t(x) \sum_{y \neq x} \boldsymbol{q}_t^\theta(y) \nabla_\theta \log s_\theta(x, T-t)_y,$$

such that $\|\widehat{\nabla \ell_t}(\theta) - \nabla \ell_t(\theta)\| \leq R_{\max}\, \delta$, where $R_{\max} := \sup_{t \in [0,T]} \max_{x \in \mathcal{X}} |R_t(x)| < \infty$.

Computation of the loss function. The summand in Theorem 3.1 involves the unknown distributions $\boldsymbol{q}_t^\theta(x)$ and $\boldsymbol{q}_t^\theta(y)$. While the outer sum can be estimated via Monte Carlo sampling (with N samples $(x_1, \ldots, x_N)$), the inner sum is weighted by $\boldsymbol{q}_t^\theta(y)$. As noted in Lou et al. (2024), a single $x \in \mathcal{X}$ provides access to every component of the concrete score $s_\theta(x,t)_y$, for $y \neq x$, and then to $\boldsymbol{q}_t^\theta(y \mid x)$ since this is the only missing quantity in the sampling scheme (Eq. (11) in Appendix A). Hence, one may be tempted to estimate $\boldsymbol{q}_t^\theta(y) = \sum_{x \in \mathcal{X}} \boldsymbol{q}_t^\theta(y \mid x)\, \boldsymbol{q}_t^\theta(x) \approx \frac{1}{N} \sum_{i=1}^N \boldsymbol{q}_t^\theta(y \mid x_i)$. However, as illustrated in Fig. 1, for a given y, we only have access to a single value $\boldsymbol{q}_t^\theta(y \mid x_i)$; we do not even have two such values, let alone N, and thus cannot take advantage of the full sample set used in the initial Monte Carlo estimation, leading to a really high-variance estimation. In fact, for any reasonable block size n, although it is certain that at least one sample x_i will be a neighbor of y, it is highly unlikely that two distinct samples x_i and x_j both have y as a neighbor, since this would require them to differ from y at exactly one token position.

Self-normalized importance sampling (SNIS) of $\boldsymbol{q}_t^\theta(y)$. To address this issue, for a given y, we build M samples $\{z_i\}_{1 \leq i \leq M}$ from $\boldsymbol{q}_t^\theta(\cdot \mid y)$ and employ the importance sampling approximation

$$\boldsymbol{q}_t^\theta(y) = \sum_{z \in \mathcal{X}} \boldsymbol{q}_t^\theta(y \mid z) \boldsymbol{q}_t^\theta(z) = \sum_{z \in \mathcal{X}} \boldsymbol{q}_t^\theta(y \mid z) \frac{\boldsymbol{q}_t^\theta(z)}{\boldsymbol{q}_t^\theta(z \mid y)} \boldsymbol{q}_t^\theta(z \mid y) \approx \frac{1}{M} \sum_{i=1}^M \boldsymbol{q}_t^\theta(y \mid z_i) \frac{\boldsymbol{q}_t^\theta(z_i)}{\boldsymbol{q}_t^\theta(z_i \mid y)},$$

with weights $w(z_i) = \boldsymbol{q}_t^\theta(z_i)/\boldsymbol{q}_t^\theta(z_i \mid y)$ and proposal $\boldsymbol{q}_t^\theta(z_i \mid y)$. Following Bayes' rule $\boldsymbol{q}_t^\theta(z_i \mid y) \propto \boldsymbol{q}_t^\theta(y \mid z_i) \boldsymbol{q}_t^\theta(z_i)$, the proposal distribution $\boldsymbol{q}_t^\theta(z_i \mid y)$ is **optimal** in the sense that it minimizes the variance of the importance weights. This makes it a theoretically sound and efficient choice for

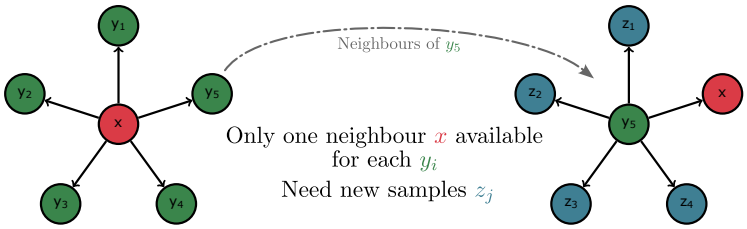

Figure 1: **On the estimation of $q_t^\theta(y)$. Left:** Given a sample x, only its neighbors $\{y_i\}_{1\leq i\leq 5} \in \mathcal{X}$ are accessible for computing $q_t^\theta(y_i \mid x)$, and each y_i typically has only one such parent x in the sampled batch. **Right:** For a given y_5, there are several neighbours $\{z_j\}_{1\leq j\leq 5}$. It is unlikely to find multiple distinct samples such that both are neighbors of y_5, since this would require them to differ from y_5 at exactly the same token position.

importance sampling. However, the marginal distribution $\boldsymbol{q}_t^\theta(z_i)$ is typically intractable, so evaluating $w(z_i)$ is unfeasible. Fortunately, one has access to an approximation of $\boldsymbol{q}_t^\theta(z_i)$ up to a normalization constant thanks to the score $1/s_\theta(z_i, T-t)_y \approx \boldsymbol{q}_t^\theta(z_i)/\boldsymbol{q}_t^\theta(y)$. This is sufficient for self-normalized importance sampling with weights $\tilde{w}(z_i) \propto \boldsymbol{q}_t^\theta(z_i)/\boldsymbol{q}_t^\theta(z_i \mid y)$. The estimator obtained via SNIS takes the form

$$\hat{\boldsymbol{q}}_t^\theta(y) = \frac{\sum_{i=1}^M \boldsymbol{q}_t^\theta(y \mid z_i)\tilde{w}(z_i)}{\sum_{i=1}^M \tilde{w}(z_i)} = \left(\frac{1}{M}\sum_{i=1}^M \frac{1}{\boldsymbol{q}_t^\theta(y \mid z_i)}\right)^{-1}.$$

Thus, the SNIS estimate can be computed as the harmonic mean of the conditionals $\boldsymbol{q}_t^\theta(y \mid z_i)$. The estimator is both consistent and asymptotically unbiased and has significantly lower variance than the naive single sample estimate. Under a mild moment assumption, we prove in Appendix C.2 that $\mathbb{V}(\hat{\boldsymbol{q}}_t^\theta(y)) = \mathcal{O}(M^{-1})$. Finally, the gradient can be computed as $\nabla_\theta \ell_t(\theta) = \mathbb{E}_{x\sim \boldsymbol{q}_t^\theta}[R_t(x)g(x,\theta)]$, where $g(x,\theta) := \sum_{\substack{y\in\mathcal{X}\\ y\neq x}} \boldsymbol{q}_t^\theta(y)\nabla_\theta \log s_\theta(x, T-t)_y$, and we estimate $\boldsymbol{q}_t^\theta(y)$ with the SNIS method described above.

Importance sampling of the outer sum. Although this defines an unbiased gradient, the `REINFORCE` algorithm is known to have high variance and to not restrict large policy updates. To address the latter limitation and estimate $g(x,\theta)$, we build upon the core ideas introduced by Trust Region Policy Optimization (`TRPO`) (Schulman et al., 2015a). Instead of sampling from $\boldsymbol{q}_t^\theta$ one can leverage importance sampling through an old policy $\boldsymbol{q}_t^{\theta_{\text{old}}}$, and constraint the KL divergence between the old and the current policy as follows:

$$\nabla_\theta \ell_t(\theta) = \mathbb{E}_{x\sim \boldsymbol{q}_t^{\theta_{\text{old}}}}\left[R_t(x)\frac{\boldsymbol{q}_t^\theta(x)}{\boldsymbol{q}_t^{\theta_{\text{old}}}(x)}g(x,\theta)\right]. \tag{7}$$

Assumption 3.2. *We assume that there exists $\delta > 0, \rho > 0$ and $G > 0$ such that for all $\theta \in \mathbb{R}^p$, $x \neq y \in \mathcal{X}$, $t \in [0,T]$, $\|\log s_\theta(x, T-t)_y - \log r_{x,y}^t(\theta)\| \leq \delta$ and $\|\nabla_\theta \log s_\theta(x, T-t)_y - \nabla_\theta \log r_{x,y}^t(\theta)\| \leq \delta$, $\boldsymbol{q}_t^\theta(x)/\boldsymbol{q}_t^{\theta_{\text{old}}}(x) \leq \rho$ and $\|\nabla_\theta \log s_\theta(x, T-t)_y\| \leq G$.*

Once adapted for concrete score, this formulation leads us to the following result.

Theorem 3.2 (Importance sampling gradient)**.** *Under Assumption 3.2 and with the notations introduced in Section 2, applying the `TRPO` with concrete score approximations yields the following estimator of Eq. (5):*

$$\widehat{\nabla \ell_t}(\theta) = \mathbb{E}_{x\sim \boldsymbol{q}_t^{\theta_{\text{old}}}}\left[R_t(x)h(x,\theta)\right], \tag{8}$$

such that $\|\widehat{\nabla \ell_t}(\theta) - \nabla \ell_t(\theta)\| \leq R_{\max} C(\rho, G)\delta$, *where* $R_{\max} := \sup_{t\in[0,T]} \max_{x\in\mathcal{X}} |R_t(x)| < \infty$, $C(\rho, G) > 0$ *is a constant depending on* ρ *and* G, *and*

$$h(x,\theta) = \sum_{\substack{y\in\mathcal{X}\\ y\neq x}} \boldsymbol{q}_t^{\theta}(y) \frac{\boldsymbol{q}_t^{\theta}(y)}{\boldsymbol{q}_t^{\theta_{\text{old}}}(y)} \frac{s_{\theta_{\text{old}}}(x, T-t)_y}{s_\theta(x, T-t)_y} \nabla_\theta \log s_\theta(x, T-t)_y.$$

The quantity $h(x,\theta)$ is expressed in this way in Eq. (7) to emphasize how the loss will be computed in practice. While being the founding step of state-of-the-art policy gradient algorithms, `TRPO` requires solving a constrained optimization problem at each step. However, thanks to Theorem 3.2, we can now build powerful, stable, scalable, and easy-to-implement policy gradient algorithms.

3.2 `SEPO` : Score Entropy Policy Optimization

Our algorithm relies on the ideas introduced in (Schulman et al., 2017; Shao et al., 2024), but can be adapted to any policy gradient algorithm built on `REINFORCE` or `TRPO`. Inspired from these algorithms, we clip $\boldsymbol{q}_t^{\theta}(x)/\boldsymbol{q}_t^{\theta_{\text{old}}}(x)$ via the following ratio that appears in the inner sum of Theorem 3.2:

$$u_{x,y}^t = \frac{\boldsymbol{q}_t^{\theta}(x)}{\boldsymbol{q}_t^{\theta_{\text{old}}}(x)} = \frac{\boldsymbol{q}_t^{\theta}(y)}{\boldsymbol{q}_t^{\theta_{\text{old}}}(y)} \frac{s_{\theta_{\text{old}}}(x, T-t)_y}{s_\theta(x, T-t)_y}.$$

to the interval $[1-\epsilon, 1+\epsilon]$ for some hyperparameter $\epsilon > 0$. This can be interpreted as enforcing a trust region condition analogous to $\rho = 1+\epsilon$ in Assumption 3.2. Another advantage of discrete diffusion models is their great generation flexibility. It is then be possible to apply our algorithm **conditionally** (via a training dataset, typically in RHLF) or **unconditionally** for fine-tuning. Hence, in the conditional form, Eq. (7) becomes

$$\mathbb{E}_{z\sim\mathcal{D}} \mathbb{E}_{x\sim \boldsymbol{q}_t^{\theta_{\text{old}}}(x|z)} \left[R_t(x) \frac{\boldsymbol{q}_t^{\theta}(x \mid z)}{\boldsymbol{q}_t^{\theta_{\text{old}}}(x \mid z)} g(x \oplus z, \theta) \right].$$

where $x \oplus z$ represents the concatenation of x and z. Instead of using directly the reward $R_t(x)$, we compute an advantage $A(x)$ to reduce the variance of the Monte-Carlo estimations. This quantifies how much better an action is compared to the expected return at a given state. A common approach in PPO (Schulman et al., 2017) is to learn a value network to approximate the reward, and then define the advantage A_t with Generalized Advantage Estimation (Schulman et al., 2015b). For GRPO (Shao et al., 2024), the advantage is the standardized reward over each group. Specifically, for a group of outputs $x = \{x_1, \ldots, x_G\}$, we have that $A_t(x_i) = \frac{R_t(x_i) - \text{mean}(R_t(x))}{\text{std}(R_t(x))}$, for $i \in \{1, \ldots, G\}$.

Remark 3.3. *The loss function takes the form*

$$\ell^A(\theta) = \mathbb{E}_{x\sim\boldsymbol{\pi}_{\theta_{\text{old}}}} [\sum_{\substack{y\in\mathcal{X}\\ y\neq x}} w_{x,y} \log s_\theta(x, T-T_0)_y], \tag{9}$$

where $w_{x,y} = \boldsymbol{\pi}_\theta(y) f(u_{x,y}^{T-T_0})$ *is a coefficient (where* f *is a function to specify) and the log concrete score* $\log s_\theta(x, T-T_0)_y$ *is the only term with an attached gradient. PPO, GRPO, and other methods can be constructed by specifying the function* f *in the coefficient* $w_{x,y}$. *In Appendix B, we present a unified framework encompassing methods that can be derived from* `SEPO`.

Optionally, for $t \in [0, T_0]$, a $\text{KL}(\boldsymbol{q}_t^{\theta} \| \boldsymbol{q}_t^{\theta_{\text{pre}}})$ term can also be added to the loss, as in Eq. (4). Although this is not absolutely necessary, as clipping already implicitly regularizes with a $\text{KL}(\boldsymbol{q}_t^{\theta} \| \boldsymbol{q}_t^{\theta_{\text{old}}})$ term (Schulman et al., 2017; Fan et al., 2024), the derivation is given in Appendix D.2, for completeness. This leads to the Score Entropy Policy Optimization (`SEPO`) algorithm described in Algorithm 1. `SEPO` iteratively samples from the target distribution via a CTMC (Line 4) and optimizes θ_s using an optimization objective (Line 6), refining the policy with policy gradients. Specifically, **Line 4** generates samples from the target distribution $\boldsymbol{\pi}_{\theta_{\text{old}}}$ using the CTMC $\overline{Q}^{\theta_{\text{old}}}$. This can be done in $O(1)$ time complexity by leveraging the queuing trick introduced in (Marion et al., 2024, Alg. 3), at a higher memory cost. **Line 6** updates the parameters θ_s using a policy optimization algorithm based on the objective ℓ^A (see Eq. (9)).

This means performing K iterations of gradient ascent (or descent) on the policy loss function to improve the policy $\boldsymbol{\pi}(\theta_s)$ using the previously collected samples and computed advantages.

Algorithm 1 SEPO

1: **Require**: CTMC $\overline{Q}^{\theta}$, iteration S, epoch K
2: Set θ_0 and θ_{old} to θ_{pre}
3: **for** $s \in [1, \cdots, S]$ **do**
4: Sample from $\boldsymbol{\pi}_{\theta_{\text{old}}}$ with $\overline{Q}^{\theta_{\text{old}}}$
5: Compute the reward and the advantage
6: Optimize θ_s with ℓ^A for K epochs
7: Set θ_{old} to θ_s
8: **end for**
9: **Output**: θ_{S+1}

3.3 Sampling through gradient flow

Bilevel problem. We use sampling to reach the limiting process of the backward distribution $\boldsymbol{q}_t^{\theta}$. This procedure can be interpreted as optimizing a functional $\mathcal{G} : \Delta_d \times \mathbb{R}^p \to \mathbb{R}$ over $\Delta_d \subset \mathbb{R}^d$ as

$$\boldsymbol{\pi}_\theta = \operatorname*{argmin}_{\boldsymbol{p} \in \Delta_d} \mathcal{G}(\boldsymbol{p}, \theta).$$

When $\boldsymbol{\pi}_\theta$ is the limiting distribution of an infinite-time process (e.g., Langevin diffusion in the continuous case, Langevin 1908; Pavliotis 2014), one can recast Eq. (3) as a bilevel optimization problem. This has been proposed by Marion et al. 2024 in the continuous case and allows to efficiently alternate between optimizing one-step of the inner problem and one-step of the outer problem.

Gradient flow interpretation. In our case, $\boldsymbol{\pi}_\theta$ is reached with *finite-time horizon*, in T steps of sampling. However, it is possible to reach $\boldsymbol{\pi}_\theta$ in *infinite-time horizon* by sampling from a specific time-homogeneous CTMC. The choice of the functional $\mathcal{G}(\boldsymbol{p}, \theta) = \mathrm{KL}(\boldsymbol{p} || \boldsymbol{\pi}_\theta)$ leads to a gradient flow interpretation of sampling via a specific CTMC.

Lemma 3.3 (Gradient flow)**.** *Sampling from the following ordinary differential equation*

$$\frac{\mathrm{d}\boldsymbol{p}_t}{\mathrm{d}t} = Q_t^c \boldsymbol{p}_t, \quad \textit{where } Q_t^c \coloneqq Q_t + \overline{Q}_t,$$

implements a gradient flow for $\mathrm{KL}(\,\cdot\,||\,\boldsymbol{p}_t)$ *in* Δ_d*, with respect to a Wassertein-like metric.*

Corrector steps. Of course, s_θ is not perfectly learned in practice, and we just have access to the rate matrix $Q_t^{c,\theta} \coloneqq Q_t + \overline{Q}_t^{\theta}$. But this gives us insight into the choice of our sampling strategy, especially with predictor-corrector techniques for discrete diffusion, introduced in (Campbell et al., 2022) and developed in (Zhao et al., 2024). We will then sample from the time-homogeneous CTMC of rate $Q_{T-T_0}^{c,\theta}$ to reach $\boldsymbol{\pi}_\theta$ with infinite-time horizon. Note that this does not require computing an integral compared to the time-inhomogeneous case. We are then optimizing a functional in Wassertein space through sampling (Marion et al., 2024; Bonet et al., 2024). Sampling from Q_t^c affects Line 4 of Algorithm 1. In practice, the sample quality can be improved by adding corrector steps with $Q_t^c = Q_t + \overline{Q}_t^{\theta}$, as proposed in (Campbell et al., 2022). Once the process has run for T_0 steps, multiple sampling iterations from $Q_{T-T_0}^c$ can be performed.

Linear system characterization. In this case, $\nabla_\theta \boldsymbol{\pi}_\theta$ in Eq. (5) will be obtained by solving a linear system, using the implicit function theorem (see Appendix C.4) on $\nabla_1 \mathcal{G}$, through a corrected version denoted $\nabla_\theta^\eta \boldsymbol{\pi}_\theta$. While both the evaluation of the derivatives and the inversion of this linear system can be done automatically (Blondel et al., 2022), it is costly given the dimensionality d. Instead, we provide the exact linear system as well as a closed form of the inverse in Proposition 3.4. Note that this affects Line 6 of Algorithm 1, where $\nabla_\theta \boldsymbol{\pi}_\theta$ in Eq. (5) is replaced by $\nabla_\theta^\eta \boldsymbol{\pi}_\theta$.

Proposition 3.4. *For each* $\eta > 0$*,* $\nabla_\theta^\eta \boldsymbol{\pi}_\theta$ *is the solution to a linear system of the form*

$$A_\eta \boldsymbol{X} = B_\eta \in \mathbb{R}^{d \times p},$$

where A_η *is a rank-*1 *update to the* $d \times d$ *identity matrix, whose inverse can be explicitly computed using the Sherman–Morrison formula.*

3.4 Convergence bounds

From a high-level point of view, Algorithm 1 alternates between sampling and optimization steps. We can then view Algorithm 1 as the following coupled equations:

$$\begin{aligned} \mathrm{d}\boldsymbol{q}_s &= Q^{c,\theta_s}_{T-T_0}\boldsymbol{q}_s \mathrm{d}s, \\ \mathrm{d}\theta_s &= -\beta_s \Gamma(\boldsymbol{q}_s, \theta_s)\mathrm{d}s, \end{aligned} \tag{10}$$

for $0 \leq s \leq S$. The gradient used on line 6 of Algorithm 1 depends both on $\boldsymbol{q}_s$ and θ_s, and we refer to it as Γ (so that $\nabla_\theta \ell^A(\theta_s) = \Gamma(\boldsymbol{\pi}_{\theta_s}, \theta_s)$). To simplify the analysis, the evolution of both $\boldsymbol{q}_s$ and θ_s is done in continuous time flow, for some $s \in [0, S]$, with $S > 0$. Let $\|\cdot\|$ denote the Euclidean norm on $\mathbb{R}^p$. We reintroduce assumptions on $\boldsymbol{\pi}_\theta$ and Γ made in (Marion et al., 2024).

Assumption 3.4. *There exists $C \geq 0$ such that for all $x \in \mathcal{X}$ and $\theta \in \mathbb{R}^p$, $\|\nabla_\theta \boldsymbol{\pi}_\theta(x)\| \leq C$. There exists $\varepsilon > 0$ such that for all $x \in \mathcal{X}$ and $\theta \in \mathbb{R}^p$, $\boldsymbol{\pi}_\theta(x) > \varepsilon$.*

This assumption states that the gradient of the target distribution is bounded. The second part is similar to the ambiguity of the language often considered when studying models acting on spaces like $\mathcal{X}$ (Zekri et al., 2024; Hu et al., 2024; Xie et al., 2021).

Assumption 3.5. $\exists C_\Gamma \geq 0$ *such that for all* $p, q \in \Delta_d$, $\theta \in \mathbb{R}^p$, $\|\Gamma(p, \theta) - \Gamma(q, \theta)\| \leq C_\Gamma \sqrt{\mathrm{KL}(p||q)}$.

This assumption essentially states that the gradient Γ is Lipschitz with respect to the KL divergence on Δ_d. With these in place, we establish the convergence of the average objective gradients.

Theorem 3.5 (Convergence of Algorithm 1)**.** *Let $S > 0$ and θ_s be the solution to Eq. (10) with $\beta_s = \min(1, 1/\sqrt{s})$, for $s \in [0, S]$. Under Assumptions 3.4 and 3.5, we have, as $S \to \infty$,*

$$\frac{1}{S}\int_0^S \|\nabla \ell^A(\theta_s)\|^2 \mathrm{d}s = \mathcal{O}\left(1/\sqrt{S}\right).$$

4 Experiments

4.1 Language modeling

We briefly present our experiments on language modeling; see Appendix E.1 for details.

Training. We fine-tune the SEDD Medium Absorb model (Lou et al., 2024) using `SEPO` in an Actor-Critic PPO framework, directly optimizing with reinforcement learning without a supervised finetuning stage. The reward model, built from GPT-2 architecture and trained on the HH-RLHF dataset (Bai et al., 2022), provides reward signals to guide fine-tuning (see Figs. 4 and 5). We train two variants, SEDD-`SEPO`-128 and SEDD-`SEPO`-1024, respectively trained with $T = 128$ and $T = 1024$ denoising steps, to measure the impact of those denoising steps on final response quality.

Quantitative evaluation. We evaluate our models using pairwise comparisons with GPT-3.5 Turbo (Brown et al., 2020) as a judge over 153 prompts from the Awesome ChatGPT Prompts dataset (Akın, 2023). As shown in Table 1, both SEDD-`SEPO` variants outperform the pre-trained model, demonstrating the benefits of reinforcement learning fine-tuning. Increasing the number of denoising steps further improves response quality, with SEDD-`SEPO`-1024 achieving the best results overall. Qualitative results are deferred to Appendix E.1.

	SEDD-`SEPO`-128			**SEDD-**`SEPO`-1024		
Number of steps T	128	512	1024	128	512	1024
SEDD Vanilla	71.2%	64.1%	67.9%	74.5%	**75.8%**	73.2%
SEDD-`SEPO`-128	×	×	×	63.1%	**68.8%**	67.8%
SEDD-`SEPO`-1024	36.9%	31.2%	32.2%	×	×	×

Table 1: Proportion of outputs deemed favorable by the Judge LLM for each model and each denoising steps $T \in \{128, 512, 1024\}$. **Best** results are highlighted.

4.2 DNA sequence modeling

We present an experiment on a DNA sequence modeling task, which follows the setup of Wang et al. (2025). Additional details are provided in Appendix E.2.

Dataset, settings, and baselines. We employ the pretrained model of Wang et al. (2025), a masked discrete diffusion model (Sahoo et al., 2024) trained on $700k$ regulatory DNA sequences (200 bp) from the Gosai dataset (Gosai et al., 2023). Each sequence is annotated with a continuous enhancer activity score in the HepG2 cell line, and a reward model trained on this score is used during fine-tuning. Our DNA experiments are conducted in a fully *unconditional* setting: sampling starts from a fully noisy sequence and proceeds via denoising without any prefix or conditioning signal; rewards are computed only on the final denoised sample (i.e., with $T_0 = T$). We compare our method to a diverse set of state-of-the-art baselines: *(i) Guidance-based models*, including classifier guidance (CG) for discrete diffusion models (Nisonoff et al., 2025), Sequential Monte Carlo methods (SMC, TDS) (Wu et al., 2023), and classifier-free guidance (CFG) (Ho and Salimans, 2021); *(ii) Direct reward optimization* with DRAKES (Wang et al., 2025), which uses reward gradients in an RL-based fine-tuning setup for discrete diffusion models; *(iii) Policy Optimization* with GLID^{2}E (Cao et al., 2025), a gradient-free RL-based tuning method.

Table 2: Evaluation of DNA modeling methods. State-of-the-art performance is in **bold**, and the second-highest performance is underlined. Means over 3 seeds; standard deviations in parentheses.

Method	*Pred-Activity (median)* ↑	*ATAC-Acc (%)* ↑	*3-mer Corr*↑	*Log-Lik (median)* ↑
Pretrained	0.17 (0.04)	1.5 (0.2)	−0.061 (0.034)	−261.0 (0.6)
CG	3.30 (0.00)	0.0 (0.0)	−0.065 (0.001)	−266.0 (0.6)
SMC	4.15 (0.33)	39.9 (8.7)	0.840 (0.045)	−259.2 (5.5)
TDS	4.64 (0.21)	45.3 (16.4)	0.848 (0.008)	−257.1 (5.0)
CFG	5.04 (0.06)	92.1 (0.9)	0.876 (0.004)	−265.0 (2.6)
DRAKES	5.61 (0.07)	92.5 (0.6)	**0.887 (0.002)**	−264.0 (0.6)
DRAKES w/o KL	6.44 (0.04)	82.5 (2.8)	0.307 (0.001)	−281.6 (0.6)
GLID^{2}E	7.35 (0.07)	90.6 (0.3)	0.49 (0.074)	−239.9 (1.4)
GLID^{2}E w/o M1	2.57 (0.60)	0.63 (0.3)	0.473 (0.078)	**−239.1 (10.1)**
GLID^{2}E w/o M2	6.62 (0.42)	67.3 (39.4)	0.458 (0.009)	−244.7 (21.5)
SEPO	7.55 (0.01)	99.5 (0.2)	0.500 (0.004)	−243.8(0.5)
SEPO with GF	**7.64 (0.01)**	**99.9 (0.09)**	0.638 (0.001)	−248.2 (0.1)

Evaluation metrics and results. We report four metrics: (i) Pred-Activity, the median predicted activity from a held-out reward oracle; (ii) ATAC-Acc, a binary chromatin accessibility score in HepG2; (iii) 3-mer Corr, the correlation between 3-mer frequencies of generated and sequences from the Gosai dataset; and (iv) Log-Lik, the log-likelihood under the pretrained diffusion model. We find that SEPO **achieves the highest performance** across all key metrics except 3-mer Corr, confirming its ability to generate highly active and biologically relevant sequences while maintaining a reasonable alignment with the training distribution. Notably, SEPO with gradient flow (GF) achieves near-perfect chromatin accessibility (99.9%) and the highest enhancer activity (7.64), surpassing both DRAKES and GLID^{2}E. While SEPO does not fully maximize 3-mer Corr, this metric primarily reflects low-level distributional similarity rather than functional relevance. Our results suggest that prioritizing enhancer activity and biological accessibility leads to more meaningful sequence generation in practical scenarios. Moreover, we observe that SEPO also exhibits significantly lower variance across runs (as shown by the small standard deviation in Table 2), highlighting the stability of our generated sequences, once the optimization process is performed.

Finally, as illustrated in Fig. 2, even in the worst case over 640 samples, SEPO methods maintain a minimum Pred-Activity score around 6, whereas all other methods exhibit much lower worst-case scores between 0 and 2. These strong performances, combined with the stability of the generation, make SEPO a highly effective counterpart to both inference-time methods and other RL-based fine-tuning approaches, providing a more reliable solution when generation quality is prioritized over sampling speed.

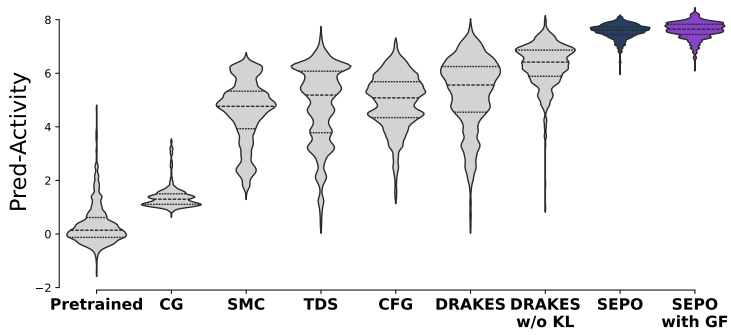

Figure 2: **Violin plot of Pred-Activity scores across models.** The plot shows the distribution of predicted enhancer activity (Pred-Activity) from the held-out reward oracle for each model, across 640 generated sequences. SEPO and SEPO with GF achieve the highest Pred-Activity scores with low variance, illustrating the effectiveness as well as the stability of our optimization process. Results for GLID^{2}E are not displayed as the corresponding finetuned model weights are not publicly available.

Wall-clock and overhead. SNIS adds a batched forward pass to compute transition probabilities for each of the N outer samples, which moderately increases per-epoch cost but reduces the number of training epochs. On a single NVIDIA® GeForce RTX™ 3090 (24GB) GPU, we report the following computational timings.

Method	*KL Trunc.*	*#Epochs*	*Batch / GRPO*	*Runtime (hh:mm)*	*Time/Epoch (mm:ss)*	*Final Pred-Activity*
SEPO	10	14	$8/8$	00:19	01:21	7.55
SEPO with GF	10	8	$8/8$	00:24	02:59	7.64
SEPO (no SNIS)	10	199	$8/8$	00:10	00:03	4.66
DRAKES (no KL)	–	127	$8/-$	00:26	00:12	6.41
DRAKES (with KL)	50	66	$8/-$	00:31	00:28	5.70

SEPO achieves higher final performance with fewer epochs despite a higher per-epoch cost, yielding competitive total runtime. Our SNIS estimator trades a forward pass for variance reduction, paying off in stability and wall-clock.

5 Conclusion

We introduced SEPO, a novel approach for fine-tuning discrete diffusion models using policy gradient methods. By extending previous work that applied these methods on continuous spaces, we developed a unified framework that adapts this methodology to the discrete case. SEPO is applicable to both **conditional** and **unconditional** generation tasks, broadening its applicability across diverse fine-tuning scenarios. Experimental results demonstrate the effectiveness of our approach in optimizing discrete diffusion models while addressing key challenges such as non-differentiability and combinatorial complexity. Future work includes further refining gradient estimation techniques and exploring applications in structured generative modeling. One limitation of the current method is that the inner importance sampling steps introduce additional computational complexity, as resampling is performed for each neighbor of the samples obtained from the outer Monte Carlo sampling.

Acknowledgements

The authors would like to thank Pierre Marion, Anna Korba, and Omar Chehab for fruitful discussions. This work was supported by the Office of Naval Research (ONR), under grant N00014-23-1-2729. This work was done thanks to the ARPE program of ENS Paris-Saclay, which supported the visit of the first author to Imperial College London.

References

Akın, F. (2023). awesome-chatgpt-prompts. https://github.com/f/awesome-chatgpt-prompts. Accessed: 2025-01-29.

Amari, S.-i. (2012). *Differential-geometrical methods in statistics*. Springer Science & Business Media.

Arriola, M., Gokaslan, A., Chiu, J. T., Yang, Z., Qi, Z., Han, J., Sahoo, S. S., and Kuleshov, V. (2025). Block diffusion: Interpolating between autoregressive and diffusion language models. *arXiv preprint arXiv:2503.09573*.

Austin, J., Johnson, D. D., Ho, J., Tarlow, D., and Van Den Berg, R. (2021). Structured denoising diffusion models in discrete state-spaces. In *Advances in Neural Information Processing Systems*, volume 34, pages 17981–17993.

Avsec, Ž., Agarwal, V., Visentin, D., Ledsam, J. R., Grabska-Barwinska, A., Taylor, K. R., Assael, Y., Jumper, J., Kohli, P., and Kelley, D. R. (2021). Effective gene expression prediction from sequence by integrating long-range interactions. *Nature methods*, 18(10):1196–1203.

Bai, Y., Jones, A., Ndousse, K., Askell, A., Chen, A., DasSarma, N., Drain, D., Fort, S., Ganguli, D., Henighan, T., et al. (2022). Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*.

Bartlett, M. S. (1951). An inverse matrix adjustment arising in discriminant analysis. *Ann. Math. Stat.*, 22(1):107–111.

Benton, J., Shi, Y., De Bortoli, V., Deligiannidis, G., and Doucet, A. (2024). From denoising diffusions to denoising Markov models. *J. R. Stat. Soc. B*, 86(2):286–301.

Black, K., Janner, M., Du, Y., Kostrikov, I., and Levine, S. (2024). Training diffusion models with reinforcement learning. In *International Conference on Learning Representations*.

Blondel, M., Berthet, Q., Cuturi, M., Frostig, R., Hoyer, S., Llinares-López, F., Pedregosa, F., and Vert, J.-P. (2022). Efficient and modular implicit differentiation. In *Advances in Neural Information Processing Systems*, volume 35, pages 5230–5242.

Bonet, C., Uscidda, T., David, A., Aubin-Frankowski, P.-C., and Korba, A. (2024). Mirror and Preconditioned Gradient Descent in Wasserstein Space. In *Advances in Neural Information Processing Systems*.

Brown, T., Mann, B., Ryder, N., et al. (2020). Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901.

Campbell, A., Benton, J., De Bortoli, V., Rainforth, T., Deligiannidis, G., and Doucet, A. (2022). A continuous time framework for discrete denoising models. In *Advances in Neural Information Processing Systems*, volume 35, pages 28266–28279.

Cao, H., Shi, H., Wang, C., Pan, S. J., and Heng, P.-A. (2025). GLID^{2}e: A gradient-free lightweight fine-tune approach for discrete sequence design. In *ICLR 2025 Workshop on Generative and Experimental Perspectives for Biomolecular Design*.

Clark, K., Vicol, P., Swersky, K., and Fleet, D. J. (2024). Directly fine-tuning diffusion models on differentiable rewards. In *International Conference on Learning Representations*.

Diaconis, P. and Saloff-Coste, L. (1996). Logarithmic sobolev inequalities for finite markov chains. *The Annals of Applied Probability*, 6(3):695–750.

Fan, Y., Watkins, O., Du, Y., Liu, H., Ryu, M., Boutilier, C., Abbeel, P., Ghavamzadeh, M., Lee, K., and Lee, K. (2024). Reinforcement learning for fine-tuning text-to-image diffusion models. In *Advances in Neural Information Processing Systems*, volume 36.

Gillespie, D. T. (2001). Approximate accelerated stochastic simulation of chemically reacting systems. *J. Chem. Phys.*, 115(4):1716–1733.

Gokaslan, A., Cohen, V., Pavlick, E., and Tellex, S. (2019). Openwebtext corpus. http://Skylion007.github.io/OpenWebTextCorpus.

Gosai, S. J., Castro, R. I., Fuentes, N., Butts, J. C., Kales, S., Noche, R. R., Mouri, K., Sabeti, P. C., Reilly, S. K., and Tewhey, R. (2023). Machine-guided design of synthetic cell type-specific cis-regulatory elements. *bioRxiv*.

Gruver, N., Stanton, S., Frey, N., Rudner, T. G., Hotzel, I., Lafrance-Vanasse, J., Rajpal, A., Cho, K., and Wilson, A. G. (2024). Protein design with guided discrete diffusion. In *Advances in Neural Information Processing Systems*, volume 36.

Hamming, R. W. (1950). Error detecting and error correcting codes. *Bell Syst. Tech. J.*, 29(2):147–160.

Ho, J., Jain, A., and Abbeel, P. (2020). Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pages 6840–6851.

Ho, J. and Salimans, T. (2021). Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. (2022). LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations*.

Hu, X., Zhang, F., Chen, S., and Yang, Z. (2024). Unveiling the statistical foundations of chain-of-thought prompting methods. *arXiv preprint arXiv:2408.14511*.

Jang, E., Gu, S., and Poole, B. (2017). Categorical Reparameterization with Gumbel-Softmax. In *International Conference on Learning Representations*.

Kelly, F. P. (2011). *Reversibility and stochastic networks*. Cambridge University Press.

Kingma, D. P. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Krantz, S. G. and Parks, H. R. (2002). *The implicit function theorem: history, theory, and applications*. Springer Science & Business Media.

Lal, A., Garfield, D., Biancalani, T., and Eraslan, G. (2024). Designing realistic regulatory dna with autoregressive language models. *Genome Research*, 34(9):1411–1420.

Langevin, P. (1908). Sur la théorie du mouvement brownien. *CR Acad. Sci. Paris*, 146(530-533):530.

Li, Y. (2023). minChatGPT: A minimum example of aligning language models with RLHF similar to ChatGPT. https://github.com/ethanyanjiali/minChatGPT.

Li, Z., Krohn, R., Chen, T., Ajay, A., Agrawal, P., and Chalvatzaki, G. (2024). Learning multimodal behaviors from scratch with diffusion policy gradient. In *Advances in Neural Informations Processing Systems*.

Lou, A., Meng, C., and Ermon, S. (2024). Discrete diffusion language modeling by estimating the ratios of the data distribution. In *International Conference on Machine Learning*.

Ma, N., Tong, S., Jia, H., Hu, H., Su, Y.-C., Zhang, M., Yang, X., Li, Y., Jaakkola, T., Jia, X., et al. (2025). Inference-time scaling for diffusion models beyond scaling denoising steps. *arXiv preprint arXiv:2501.09732*.

Maas, J. (2011). Gradient flows of the entropy for finite Markov chains. *J. Funct. Anal.*, 261(8):2250–2292.

Marion, P., Korba, A., Bartlett, P., Blondel, M., De Bortoli, V., Doucet, A., Llinares-López, F., Paquette, C., and Berthet, Q. (2024). Implicit diffusion: Efficient optimization through stochastic sampling. *arXiv preprint arXiv:2402.05468*.

Martins, A. and Astudillo, R. (2016). From Softmax to Sparsemax: A Sparse Model of Attention and Multi-Label Classification. In *International Conference on Machine Learning*, pages 1614–1623.

Meng, C., Choi, K., Song, J., and Ermon, S. (2022). Concrete score matching: Generalized score matching for discrete data. In *Advances in Neural Information Processing Systems*, volume 35, pages 34532–34545.

Nie, S., Zhu, F., You, Z., Zhang, X., Ou, J., Hu, J., Zhou, J., Lin, Y., Wen, J.-R., and Li, C. (2025). Large language diffusion models. *arXiv preprint arXiv:2502.09992*.

Nisonoff, H., Xiong, J., Allenspach, S., and Listgarten, J. (2025). Unlocking guidance for discrete state-space diffusion and flow models. In *International Conference on Learning Representations*.

Ou, J., Nie, S., Xue, K., Zhu, F., Sun, J., Li, Z., and Li, C. (2024). Your absorbing discrete diffusion secretly models the conditional distributions of clean data. *arXiv preprint arXiv:2406.03736*.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. (2022). Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Palmowski, Z. and Rolski, T. (2002). A technique for exponential change of measure for Markov processes. *Bernoulli*, 8(6):767 – 785.

Pavliotis, G. A. (2014). *Stochastic Processes and Applications*. Springer.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. (2019). Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Rakotomandimby, S., Chancelier, J.-P., De Lara, M., and Blondel, M. (2024). Learning with Fitzpatrick Losses. In *Advances in Neural Information Processing Systems*.

Rector-Brooks, J., Hasan, M., Peng, Z., Quinn, Z., Liu, C., Mittal, S., Dziri, N., Bronstein, M., Bengio, Y., Chatterjee, P., et al. (2025). Steering masked discrete diffusion models via discrete denoising posterior prediction. In *International Conference on Learning Representations*.

Ren, A. Z., Lidard, J., Ankile, L. L., Simeonov, A., Agrawal, P., Majumdar, A., Burchfiel, B., Dai, H., and Simchowitz, M. (2025). Diffusion policy policy optimization. In *International Conference on Learning Representations*.

Sahoo, S. S., Arriola, M., Schiff, Y., Gokaslan, A., Marroquin, E., Chiu, J. T., Rush, A., and Kuleshov, V. (2024). Simple and effective masked diffusion language models. In *Advances in Neural Information Processing Systems*.

Schulman, J., Levine, S., Abbeel, P., Jordan, M., and Moritz, P. (2015a). Trust Region Policy Optimization. In *International Conference on Machine Learning*, volume 37, pages 1889–1897.

Schulman, J., Moritz, P., Levine, S., Jordan, M., and Abbeel, P. (2015b). High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. (2017). Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Shao, Z., Wang, P., Zhu, Q., Xu, R., Song, J., Bi, X., Zhang, H., Zhang, M., Li, Y., Wu, Y., et al. (2024). Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Shi, J., Han, K., Wang, Z., Doucet, A., and Titsias, M. K. (2024). Simplified and generalized masked diffusion for discrete data. In *Advances in Neural Information Processing Systems*.

Singhal, R., Horvitz, Z., Teehan, R., Ren, M., Yu, Z., McKeown, K., and Ranganath, R. (2025). A general framework for inference-time scaling and steering of diffusion models. *arXiv preprint arXiv:2501.06848*.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. (2021). Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*.

Stark, H., Jing, B., Wang, C., Corso, G., Berger, B., Barzilay, R., and Jaakkola, T. (2024). Dirichlet flow matching with applications to dna sequence design. *arXiv preprint arXiv:2402.05841*.

Sun, H., Yu, L., Dai, B., Schuurmans, D., and Dai, H. (2023). Score-based continuous-time discrete diffusion models. In *International Conference on Learning Representations*.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al. (2023). Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Uehara, M., Zhao, Y., Biancalani, T., and Levine, S. (2024). Understanding reinforcement learning-based fine-tuning of diffusion models: A tutorial and review. *arXiv preprint arXiv:2407.13734*.

Uehara, M., Zhao, Y., Wang, C., Li, X., Regev, A., Levine, S., and Biancalani, T. (2025). Reward-Guided Controlled Generation for Inference-Time Alignment in Diffusion Models: Tutorial and Review. *arXiv preprint arXiv:2501.09685*.

Vaswani, A. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*.

Wang, C., Uehara, M., He, Y., Wang, A., Biancalani, T., Lal, A., Jaakkola, T., Levine, S., Wang, H., and Regev, A. (2025). Fine-Tuning Discrete Diffusion Models via Reward Optimization with Applications to DNA and Protein Design. In *International Conference on Learning Representations*.

Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8:229–256.

Wu, L., Trippe, B., Naesseth, C., Blei, D., and Cunningham, J. P. (2023). Practical and asymptotically exact conditional sampling in diffusion models. *Advances in Neural Information Processing Systems*, 36:31372–31403.

Xie, S. M., Raghunathan, A., Liang, P., and Ma, T. (2021). An explanation of in-context learning as implicit bayesian inference. *arXiv preprint arXiv:2111.02080*.

Ye, J., Xie, Z., Zheng, L., Gao, J., Wu, Z., Jiang, X., Li, Z., and Kong, L. (2025). Dream 7b.

Zekri, O., Odonnat, A., Benechehab, A., Bleistein, L., Boullé, N., and Redko, I. (2024). Large language models as Markov chains. *arXiv preprint arXiv:2410.02724*.

Zhang, Z., Chen, Z., and Gu, Q. (2025). Convergence of score-based discrete diffusion models: A discrete-time analysis. In *International Conference on Learning Representations*.

Zhao, S., Gupta, D., Zheng, Q., and Grover, A. (2025). d1: Scaling reasoning in diffusion large language models via reinforcement learning. *arXiv preprint arXiv:2504.12216*.

Zhao, Y., Shi, J., Mackey, L., and Linderman, S. (2024). Informed correctors for discrete diffusion models. *arXiv preprint arXiv:2407.21243*.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We describe the method in Section 3 and perform numerical experiments in Section 4.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Section 3.4 discusses the assumptions needed to derive our theoretical results, and we briefly discuss potential future works in the conclusion.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: The proofs of the theoretical results are available in the Appendix.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We describe the experimental details and results in the main text along with additional details in the Appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Our code is available at https://github.com/ozekri/SEPO.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We provide details of the training and test data in the Appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We indicate the variance of the results over 3 independent runs (consistent with the methods we compare with), along with the score distribution over samples in Section 4.2.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We indicate the GPU specification of which experiments were performed in Appendix E.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: We reviewed the ethics code and believe that the research conducted conforms with it.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: The paper is not tied to any application and is foundational research.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: We do not release a data or a model but propose an algorithm.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We reference creators of code, data, and models used in the paper.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We will release documented code upon acceptance.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The research does not involve human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The research does not involve human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: This research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (https://neurips.cc/Conferences/2025/LLM) for what should or should not be described.

Appendix

Roadmap. In Appendix A, we first recall our notations and introduce additional definitions. Appendix B presents our unified paradigm of policy gradient methods, including PPO and GRPO. The detailed proofs of our theoretical results are given in Appendix C. Appendix D provides mathematical supplements and Appendix E reports additional experiments and implementation details.

Table of Contents

A Additional background details

Notations. This appendix uses the following notations: $[d]$ denotes the set of integers $\{1, \dots, d\}$, $[t]_+ := \max\{0, t\}$, and I_d stands for the $d \times d$ identity matrix. We also denote the diagonal matrix with diagonal components $(\boldsymbol{z}_1, \dots, \boldsymbol{z}_d)$ by $\mathrm{diag}(\boldsymbol{z}_1, \dots, \boldsymbol{z}_d)$. Finally, if $\boldsymbol{z} = (\boldsymbol{z}_1, \dots, \boldsymbol{z}_d)$, the same matrix will be denoted by $\mathrm{diag}(\boldsymbol{z})$.

Sampling strategies. Sampling discrete diffusion models involves selecting efficient strategies to simulate the backward equation (2), while balancing computational cost and sample quality. Among other strategies for CTMCs, sampling can be done via the *tau-leaping* algorithm (Gillespie, 2001), which implements an Euler step at each position i simultaneously and independently:

$$\boldsymbol{q}_t(x^i_{t-\Delta t}|x^i_t) = \delta_{x^i_t}(x^i_{t-\Delta t}) + \Delta_t \overline{Q}^{\theta}_{T-t}(x^i_t, x^i_{t-\Delta t}) \tag{11}$$

Discrete diffusion models can also be used to perform flexible *conditional sampling* (Lou et al., 2024). Unlike *unconditional sampling*, which samples $\boldsymbol{q}_t(x_{t-\Delta t}|x_t)$, we incorporate auxiliary data $\mathbf{c}$ by modifying the probability to be sampled to $\boldsymbol{q}_t(x_{t-\Delta t}|x_t, \mathbf{c})$. Finally, the number of reverse diffusion steps, T, directly impacts computational efficiency and sample fidelity, with larger T providing more accurate approximations of the target distribution at a higher computational cost.

B Unified Paradigm of Policy Gradient Methods

As in (Shao et al., 2024), we provide the expression of the coefficient $w_{x,y}$ for `SEPO` in its PPO and GRPO variants. This can be easily extended to the other training methods synthesized in Shao et al. 2024, Section 5.2.

B.1 Proximal Policy Optimization (PPO)

For PPO (Schulman et al., 2017), the coefficient $w_{x,y}$ takes the form

$$w_{x,y}(\epsilon) = \boldsymbol{\pi}_\theta(y) \min\{\mathrm{clip}(u^{T-T_0}_{x,y}, 1-\epsilon, 1+\epsilon)A(x); u^{T-T_0}_{x,y} A(x)\}, \quad \epsilon > 0.$$

A common approach is to learn a value network to approximate the reward, and then compute the advantage as $A(x) = R(x) - V(x)$.

B.2 Group Relative Policy Optimization (GRPO)

For GRPO (Shao et al., 2024), we consider a group of outputs $x = \{x_1, \dots, x_G\}$. The coefficient $w_{x,y}$ takes the form

$$w_{x,y}(\epsilon) = \frac{\boldsymbol{\pi}_\theta(y)}{G} \sum_{i=1}^{G} \min\{\mathrm{clip}(r^{T-T_0}_{x_i,y}, 1-\epsilon, 1+\epsilon)A(x_i); r^{T-T_0}_{x_i,y} A(x_i)\}, \quad \epsilon > 0.$$

The advantages are the standardized reward over each group. Specifically, the advantages are defined as

$$A(x_i) = \frac{R(x_i) - \mathrm{mean}(R(x))}{\mathrm{std}(R(x))}, \quad i \in \{1, \dots, G\}.$$

The KL term of the original GRPO objective (Shao et al., 2024) is discussed in Section 3.2.

C Proofs of the results

This section details the proof of the results that appear in the main text.

C.1 Proof of Theorem 3.1

Expression of $\nabla_\theta \boldsymbol{q}_t^\theta$. We begin by calculating $\nabla_\theta \boldsymbol{q}_t^\theta$ appearing in Eq. (5) component-wise. Let $x \in \mathcal{X}$ and recall[1] that we can express $\boldsymbol{q}_t^\theta(x)$ as $\boldsymbol{q}_t^\theta(x) = \mathrm{e}^{-V_x(\theta)}/Z_\theta$ for some normalization constant $Z_\theta = \sum_{y\in\mathcal{X}} \mathrm{e}^{-V_y(\theta)}$. Therefore,

$$\begin{aligned}
\nabla_\theta \boldsymbol{q}_t^\theta(x) &= -\boldsymbol{q}_t^\theta(x)\nabla_\theta V_x(\theta) - \frac{\mathrm{e}^{-V_x(\theta)}\nabla_\theta Z_\theta}{Z_\theta^2} = -\boldsymbol{q}_t^\theta(x)\nabla_\theta V_x(\theta) - \boldsymbol{q}_t^\theta(x)\frac{\nabla_\theta Z_\theta}{Z_\theta} \\
&= -\boldsymbol{q}_t^\theta(x)\nabla_\theta V_x(\theta) - \boldsymbol{q}_t^\theta(x)\left(\sum_{y\in\mathcal{X}} \frac{-\nabla_\theta V_y(\theta)\mathrm{e}^{-V_y(\theta)}}{Z_\theta}\right) \\
&= -\boldsymbol{q}_t^\theta(x)\nabla_\theta V_x(\theta) + \boldsymbol{q}_t^\theta(x)\left(\sum_{y\in\mathcal{X}} \boldsymbol{q}_t^\theta(y)\nabla_\theta V_y(\theta)\right).
\end{aligned}$$

Plug-in estimator. Define

$$\widehat{\nabla_\theta \boldsymbol{q}_t^\theta}(x) := -\,\boldsymbol{q}_t^\theta(x) \sum_{\substack{y\in\mathcal{X}\\ y\neq x}} \boldsymbol{q}_t^\theta(y)\,\nabla_\theta \log s_\theta(x, T-t)_y.$$

Let the pointwise error be

$$\Delta(x) := \widehat{\nabla_\theta \boldsymbol{q}_t^\theta}(x) - \nabla_\theta \boldsymbol{q}_t^\theta(x) = -\,\boldsymbol{q}_t^\theta(x) \sum_{\substack{y\in\mathcal{X}\\ y\neq x}} \boldsymbol{q}_t^\theta(y)\Big(\nabla_\theta \log s_\theta(x, T-t)_y - \nabla_\theta \log r_{x,y}^t(\theta)\Big).$$

By (6) and since $\sum_{\substack{y\in\mathcal{X}\\ y\neq x}} \boldsymbol{q}_t^\theta(y) \le 1$,

$$\|\Delta(x)\| \;\le\; \boldsymbol{q}_t^\theta(x) \sum_{\substack{y\in\mathcal{X}\\ y\neq x}} \boldsymbol{q}_t^\theta(y)\,\delta \;\le\; \boldsymbol{q}_t^\theta(x)\,\delta. \tag{12}$$

Loss-gradient estimator and bias bound. Recall $\ell_t(\theta) = -\sum_{x\in\mathcal{X}} R(x)\boldsymbol{q}_t^\theta(x)$ and define

$$\widehat{\nabla \ell_t}(\theta) := -\sum_{x\in\mathcal{X}} R(x)\,\widehat{\nabla_\theta \boldsymbol{q}_t^\theta}(x).$$

Then

$$\widehat{\nabla \ell_t}(\theta) - \nabla \ell_t(\theta) = -\sum_{x\in\mathcal{X}} R(x)\,\Delta(x).$$

Using (12) and that $\sum_x \boldsymbol{q}_t^\theta(x) = 1$,

$$\big\|\widehat{\nabla \ell_t}(\theta) - \nabla \ell_t(\theta)\big\| \;\le\; \sum_{x\in\mathcal{X}} |R(x)|\,\|\Delta(x)\| \;\le\; \delta \sum_{x\in\mathcal{X}} |R(x)|\,\boldsymbol{q}_t^\theta(x) \;\le\; R_{\max}\,\delta.$$

Conclusion. Under the single approximation condition (6), the plug-in estimator obtained by replacing $\nabla_\theta \log r_{x,y}^t$ with $\nabla_\theta \log s_\theta$ in (5) yields the gradient formula stated in Theorem 3.1, and its bias with respect to the true $\nabla \ell_t(\theta)$ is bounded by $R_{\max}\delta$.

C.2 Self-normalized importance sampling

We now formalize the self-normalized importance sampling (SNIS) estimator, which plays a key role in approximating marginal probabilities when direct computation is intractable. The following lemma establishes an explicit expression for the SNIS estimator of $\boldsymbol{q}_t^\theta(y)$.

[1] Since we are working on a discrete finite state space $\mathcal{X}$, the normalization constant $Z_\theta = \sum_{y\in\mathcal{X}} \exp(-V_y(\theta))$ is finite as long as each $V_y(\theta) > -\infty$, which is the case.

Lemma C.1. *Let $y \in \mathcal{X}$ and $\{z_i\}_{1\le i\le M}$ i.i.d. samples from $\boldsymbol{q}_t^\theta(\cdot \mid y)$. Define $\tilde{w}(z_i) \propto \frac{\boldsymbol{q}_t^\theta(z_i)}{\boldsymbol{q}_t^\theta(z_i|y)}$. Then, the SNIS estimator of $\boldsymbol{q}_t^\theta(y)$ is*

$$\hat{\boldsymbol{q}}_t^\theta(y) = \frac{\sum_{i=1}^M \boldsymbol{q}_t^\theta(y \mid z_i)\tilde{w}(z_i)}{\sum_{i=1}^M \tilde{w}(z_i)} = \left(\frac{1}{M}\sum_{i=1}^M \frac{1}{\boldsymbol{q}_t^\theta(y \mid z_i)}\right)^{-1}$$

Proof. By Bayes' rule, $\boldsymbol{q}_t^\theta(z_i \mid y) = \boldsymbol{q}_t^\theta(y \mid z_i)\boldsymbol{q}_t^\theta(z_i)/\boldsymbol{q}_t^\theta(y)$, so $\tilde{w}(z_i) = Z\frac{\boldsymbol{q}_t^\theta(z_i)}{\boldsymbol{q}_t^\theta(z_i|y)} = Z\frac{\boldsymbol{q}_t^\theta(y)}{\boldsymbol{q}_t^\theta(y|z_i)}$ where Z is some constant independent of z_i. Therefore, $\sum_{i=1}^M \boldsymbol{q}_t^\theta(y \mid z_i)\tilde{w}(z_i) = MZ\boldsymbol{q}_t^\theta(y)$ and $\sum_{i=1}^M \tilde{w}(z_i) = Z\boldsymbol{q}_t^\theta(y)\sum_{i=1}^M \frac{1}{\boldsymbol{q}_t^\theta(y|z_i)}$. This directly yields that $\hat{\boldsymbol{q}}_t^\theta(y) = \left(\frac{1}{M}\sum_{i=1}^M \frac{1}{\boldsymbol{q}_t^\theta(y|z_i)}\right)^{-1}$. □

Having obtained a closed-form expression for the SNIS estimator, we now investigate its statistical properties. In particular, the following result shows that its variance decreases at the canonical Monte Carlo rate.

Lemma C.2. *Let $y \in \mathcal{X}$ and $\{z_i\}_{1\le i\le M}$ i.i.d. samples from $\boldsymbol{q}_t^\theta(\cdot \mid y)$. Define $U_i = \frac{1}{\boldsymbol{q}_t^\theta(y|z_i)}$ and assume that $\mathbb{E}[U_i^2] < \infty$. Then,*

$$\mathbb{V}(\hat{\boldsymbol{q}}_t^\theta(y)) = \mathcal{O}\left(\frac{1}{M}\right)$$

Proof. Define $U = \frac{1}{M}\sum_{i=1}^M U_i$ and $g(u) = 1/u$. One has $\hat{\boldsymbol{q}}_t^\theta(y) = g(U)$. We remark that $\mathbb{E}[U_i] = 1/\boldsymbol{q}_t^\theta(y)$ for each $1 \le i \le M$. Since $\mathbb{E}[U_i^2] < \infty$ by assumption, the delta method (applied to U and $g(u) = 1/u$) yields

$$\mathbb{V}(\hat{\boldsymbol{q}}_t^\theta(y)) = g'(\mathbb{E}[U_1^2])\mathbb{V}(U) + o\left(\mathbb{V}(U)\right) = \frac{g'(\mathbb{E}[U_1^2])}{M}\mathbb{V}(U_1) + o\left(\mathbb{V}(U_1)/M\right).$$

In other words, $\mathbb{V}(\hat{\boldsymbol{q}}_t^\theta(y)) = \mathcal{O}\left(\frac{1}{M}\right)$. □

C.3 Proof of Theorem 3.2

Exact reweighting identity. Following Theorem 3.1, we can artificially introduce $\boldsymbol{q}_t^{\theta_{\text{old}}}(x)$ in the expression of $\nabla\ell_t(\theta)$, using $\nabla_\theta \boldsymbol{q}_t^\theta(x) = -\boldsymbol{q}_t^\theta(x)\sum_{\substack{y\in\mathcal{X}\\ y\neq x}} \boldsymbol{q}_t^\theta(y)\nabla_\theta \log r_{x,y}^t(\theta)$ and $\frac{\boldsymbol{q}_t^\theta(x)}{\boldsymbol{q}_t^{\theta_{\text{old}}}(x)} = \frac{\boldsymbol{q}_t^\theta(y)}{\boldsymbol{q}_t^{\theta_{\text{old}}}(y)}\frac{r_{x,y}^t(\theta_{\text{old}})}{r_{x,y}^t(\theta)}$. In fact, we obtain

$$\nabla\ell_t(\theta) = \sum_{x\in\mathcal{X}} \boldsymbol{q}_t^{\theta_{\text{old}}}(x)R(x)\sum_{\substack{y\in\mathcal{X}\\ y\neq x}} \boldsymbol{q}_t^\theta(y)\underbrace{\frac{\boldsymbol{q}_t^\theta(y)}{\boldsymbol{q}_t^{\theta_{\text{old}}}(y)}\frac{r_{x,y}^t(\theta_{\text{old}})}{r_{x,y}^t(\theta)}}_{=:\,\mathcal{W}_{xy}^\star(\theta)}\nabla_\theta \log r_{x,y}^t(\theta). \tag{13}$$

Note $\mathcal{W}_{xy}^\star(\theta) = \frac{\boldsymbol{q}_t^\theta(x)}{\boldsymbol{q}_t^{\theta_{\text{old}}}(x)}$ is independent of y, and by Assumption 3.2,

$$0 < \mathcal{W}_{xy}^\star(\theta) \le \rho. \tag{14}$$

Plug-in the estimator. Define the two following quantities

$$\widehat{\mathcal{W}}_{xy}(\theta) := \frac{\boldsymbol{q}_t^\theta(y)}{\boldsymbol{q}_t^{\theta_{\text{old}}}(y)}\frac{s_{\theta_{\text{old}}}(x, T-t)_y}{s_\theta(x, T-t)_y}$$

$$\widehat{\nabla\ell_t}(\theta) := \sum_{x\in\mathcal{X}} \boldsymbol{q}_t^{\theta_{\text{old}}}(x)R(x)\sum_{\substack{y\in\mathcal{X}\\ y\neq x}} \boldsymbol{q}_t^\theta(y)\widehat{\mathcal{W}}_{xy}(\theta)\nabla_\theta \log s_\theta(x, T-t)_y.$$

By Assumption 3.2, $\frac{s_{\theta_{\text{old}}}/s_\theta}{r^t_{x,y}(\theta_{\text{old}})/r^t_{x,y}(\theta)} \in [e^{-2\delta}, e^{2\delta}]$, hence

$$\widehat{\mathcal{W}}_{xy}(\theta) \le e^{2\delta}\,\mathcal{W}^\star_{xy}(\theta) \le e^2\,\rho \quad \text{for } \delta \in (0,1]. \tag{15}$$

Bias decomposition. Subtract (13):

$$\widehat{\nabla\ell_t(\theta)} - \nabla\ell_t(\theta) = \sum_{x\in\mathcal{X}} \boldsymbol{q}^{\theta_{\text{old}}}(x)R(x) \sum_{\substack{y\in\mathcal{X}\\ y\neq x}} \boldsymbol{q}^\theta_t(y)\Big[\underbrace{\widehat{\mathcal{W}}_{xy}\big(\nabla_\theta \log s_\theta - \nabla_\theta \log r^t_{x,y}(\theta)\big)}_{\text{(I)}} + \underbrace{\big(\widehat{\mathcal{W}}_{xy} - \mathcal{W}^\star_{xy}\big)\nabla_\theta \log r^t_{x,y}(\theta)}_{\text{(II)}}\Big].$$

(I) Gradient replacement. Using Assumption 3.2, (15), and $\sum_{y\neq x} \boldsymbol{q}^\theta_t(y) \le 1$,

$$\begin{aligned}\|\text{(I)}\| &\le \delta \sum_x \boldsymbol{q}^{\theta_{\text{old}}}_t(x)|R(x)| \sum_{y\neq x} \boldsymbol{q}^\theta_t(y)\,\widehat{\mathcal{W}}_{xy}\\ &\le \delta\, e^2\rho \sum_x \boldsymbol{q}^{\theta_{\text{old}}}_t(x)|R(x)|\\ &\le e^2\rho\, R_{\max}\,\delta.\end{aligned}$$

(II) Ratio replacement. From Assumption 3.2, $\left|\log\frac{s_{\theta_{\text{old}}}}{s_\theta} - \log\frac{r^t_{x,y}(\theta_{\text{old}})}{r^t_{x,y}(\theta)}\right| \le 2\delta \Rightarrow |\widehat{\mathcal{W}}_{xy} - \mathcal{W}^\star_{xy}| \le (e^{2\delta}-1)\,\mathcal{W}^\star_{xy}$. Hence, using (14) and $\sum_{y\neq x} \boldsymbol{q}^\theta_t(y) \le 1$,

$$\begin{aligned}\|\text{(II)}\| &\le (e^{2\delta}-1)\sum_x \boldsymbol{q}^{\theta_{\text{old}}}_t(x)|R(x)| \sum_{y\neq x} \boldsymbol{q}^\theta_t(y)\,\mathcal{W}^\star_{xy}\,\|\nabla_\theta \log r^\theta_{xy}\|\\ &\le (e^{2\delta}-1)\,\rho\,(G+\delta)\sum_x \boldsymbol{q}^{\theta_{\text{old}}}_t(x)|R(x)|\\ &\le (e^2-1)\,\rho\,(G+\delta)\,R_{\max}\,\delta,\end{aligned}$$

where we used $e^{2\delta}-1 \le (e^2-1)\delta$ for $\delta \in (0,1]$.

Conclusion. Combining (I) and (II),

$$\big\|\widehat{\nabla\ell_t}(\theta) - \nabla\ell_t(\theta)\big\| \le R_{\max} C(\rho, G)\delta.$$

where $C(\rho,G) = \rho(e^2 + (e^2-1)G) > 0$. The bound is independent of θ, linear in δ, and depends only on the trust-region radius ρ, the clipping level G, and $R_{\max}$.

C.4 Proof of Proposition 3.4

To prove Proposition 3.4, we are going to prove the following explicit proposition regarding the linear system satisfied by $\nabla^\eta_\theta \boldsymbol{\pi}_\theta$.

Proposition C.3. *Let $\eta > 0$, and $D^\star_r$ denotes the block diagonal matrix defined in Lemma C.5. Then, $\nabla^\eta_\theta \boldsymbol{\pi}_\theta$ is the solution of the linear system*

$$A_\eta \boldsymbol{X} = B_\eta \in \mathbb{R}^{d\times p},$$

where $A_\eta \coloneqq D^\star_r\,[I_d - \eta\,\mathrm{diag}(1/\boldsymbol{\pi}_\theta)] - I_d \in \mathbb{R}^{d\times d}$ and $B_\eta \coloneqq -\eta D^\star_r \nabla_\theta \boldsymbol{\pi}_\theta/\boldsymbol{\pi}_\theta \in \mathbb{R}^{d\times p}$.

The proof contains three parts and occupies the rest of this section:

1. recalling the implicit function theorem,
2. computing the matrices that appear in the linear system,
3. solving the linear system.

Implicit function theorem. Let us first recall the implicit function theorem.

Theorem C.4 (Implicit function theorem, Krantz and Parks 2002)**.** *Let U be an open subset of $\mathbb{R}^d \times \mathbb{R}^p$, and $f : U \to \mathbb{R}^d$ a continuously differentiable function. Let $(a, b) \in U$ such that $f(a, b) = 0$ and $\nabla_1 f(a, b)$ is invertible. Then, there exists an open set $W \subset \mathbb{R}^p$ containing b and a function $g : W \to \mathbb{R}^d$ such that $g(b) = a$ and $\forall x \in W, f(g(x), x) = 0$. Moreover, g is continuously differentiable and*

$$\forall x \in W, \quad \nabla_1 f(a, b)\partial g(x) = -\nabla_2 f(a, b)$$

where ∂g denotes the Jacobian of g on W.

A first point to note is that $\nabla_1 \mathcal{G}(\boldsymbol{p}, \theta) = \log(\boldsymbol{p}/\boldsymbol{\pi}_\theta) + \mathbf{1}$. Even if $\boldsymbol{\pi}_\theta = \operatorname{argmin}_{\boldsymbol{p} \in \Delta_d} \mathcal{G}(\boldsymbol{p}, \theta)$, we have $\nabla_1 \mathcal{G}(\boldsymbol{\pi}_\theta, \theta) = 1 \neq 0$, because we compute the derivative in $\mathbb{R}^d$ and not in the probability simplex Δ_d. This means that we cannot directly apply Theorem C.4 to $\nabla_1 \mathcal{G}(\boldsymbol{p}, \theta)$. To address this issue, we follow (Blondel et al., 2022) and consider $\mathcal{G}$ as a function of $\mathbb{R}^d \times \mathbb{R}^p$. Since we have a problem of the form $\boldsymbol{\pi}_\theta = \operatorname{argmin}_{\boldsymbol{p} \in \Delta_d} \mathcal{G}(\boldsymbol{p}, \theta)$, we can define the fixed point operator

$$T_\eta(\boldsymbol{p}, \theta) = \operatorname{proj}_{\Delta_d}(\boldsymbol{p} - \eta \nabla_1 \mathcal{G}(\boldsymbol{p}, \theta)),$$

for $\eta > 0$. In fact, $T_\eta(\boldsymbol{\pi}_\theta, \theta) = \operatorname{proj}_{\Delta_d}(\boldsymbol{\pi}_\theta - \eta \mathbf{1})$ where $\operatorname{proj}_{\Delta_d} = \operatorname{sparsemax}$ (Martins and Astudillo, 2016; Rakotomandimby et al., 2024). From (Martins and Astudillo, 2016, Prop. 2), we have

$$\operatorname{sparsemax}(\boldsymbol{p} - \eta \mathbf{1}) = \operatorname{sparsemax}(\boldsymbol{p}), \quad \boldsymbol{p} \in \mathbb{R}^d.$$

This leads to $T_\eta(\boldsymbol{\pi}_\theta, \theta) = \boldsymbol{\pi}_\theta$, because $\operatorname{sparsemax}(\boldsymbol{\pi}_\theta) = \boldsymbol{\pi}_\theta$. We can therefore apply Theorem C.4 to the function $f_\eta(\boldsymbol{p}, \theta) = T_\eta(\boldsymbol{p}, \theta) - \boldsymbol{p}$.

Computing the matrices. Let $\eta > 0$, and define $h_\eta(\boldsymbol{p}, \theta) = \boldsymbol{p} - \eta \nabla_1 \mathcal{G}(\boldsymbol{p}, \theta)$. Then, we have $T_\eta(\boldsymbol{p}, \theta) = \operatorname{sparsemax}(h_\eta(\boldsymbol{p}, \theta))$ and $f_\eta(\boldsymbol{p}, \theta) = T_\eta(\boldsymbol{p}, \theta) - \boldsymbol{p}$. We note that since $\boldsymbol{\pi}_\theta \in \Delta_d$ and T_η is a projection onto the probability simplex, $f_\eta(\boldsymbol{\pi}_\theta, \theta) = 0$. Following (Blondel et al., 2022, App. D) (this computation can be done by using the chain rule), we have

$$\begin{aligned} \nabla_1 f_\eta(\boldsymbol{p}, \theta) &= D(\boldsymbol{p}, \theta)\left[I_d - \eta \nabla_{1,1} \mathcal{G}(\boldsymbol{p}, \theta)\right] - I_d \in \mathbb{R}^{d \times d}, \\ \nabla_2 f_\eta(\boldsymbol{p}, \theta) &= -\eta D(\boldsymbol{p}, \theta) \nabla_{1,2} \mathcal{G}(\boldsymbol{p}, \theta) \in \mathbb{R}^{d \times p}, \end{aligned} \tag{16}$$

where $D_r(\boldsymbol{p}, \theta) \coloneqq \operatorname{diag}(r(h_\eta(\boldsymbol{p}, \theta))) - r(h_\eta(\boldsymbol{p}, \theta)) r(h_\eta(\boldsymbol{p}, \theta))^{\mathrm{T}} / \|r(h_\eta(\boldsymbol{p}, \theta))\|_1 \in \mathbb{R}^{d \times d}$ and $r(h_\eta(\boldsymbol{p}, \theta)) \in \{0, 1\}^d$ (Martins and Astudillo, 2016). Here, for all $\boldsymbol{z} \in \mathbb{R}^d$, we define the vector $r(\boldsymbol{z})$ as follows:

$$\forall j \in [d], \; r(\boldsymbol{z})_j = \begin{cases} 1 & \text{if } \boldsymbol{z}_j > \tau(\boldsymbol{z}), \\ 0 & \text{otherwise,} \end{cases}$$

where τ is the unique function satisfying $\sum_{i=1}^d [\boldsymbol{z}_j - \tau(\boldsymbol{z})]_+ = 1$ for all $\boldsymbol{z} \in \mathbb{R}^d$. This definition is a bit tricky, but overall $r(h_\eta(\boldsymbol{p}, \theta))$ contains k_h times the number 1 and $d - k_h$ times the number 0, where $k_h \in [d]$. This means that

$$D_r(\boldsymbol{p}, \theta) = \operatorname{diag}(r(h_\eta(\boldsymbol{p}, \theta))) - r(h_\eta(\boldsymbol{p}, \theta)) r(h_\eta(\boldsymbol{p}, \theta))^{\mathrm{T}} / k_h.$$

We then obtain the following lemma.

Lemma C.5. *Denote $\tilde{r}(\boldsymbol{z})$ as the vector with the sorted coordinates of $r(\boldsymbol{z})$, i.e., the vector with coordinates $\tilde{r}(\boldsymbol{z})_i = r(\boldsymbol{z})_{\sigma(i)}$, where σ is the permutation such that $r(\boldsymbol{z})_{\sigma(1)} \geq \cdots \geq r(\boldsymbol{z})_{\sigma(d)}$. Then, $k_h = \max\{k \in [d] \mid 1 + k r(\boldsymbol{z})_{\sigma(k)} > \sum_{j=1}^k r(\boldsymbol{z})_{\sigma(j)}\}$ and,*

$$D_{\tilde{r}}(\boldsymbol{p}, \theta) = R - \frac{1}{k_h} T,$$

where $R \coloneqq \mathrm{diag}(\underbrace{1,\dots,1}_{k_h \text{times}}, \underbrace{0,\dots,0}_{d-k_h \text{times}})$ *and* $T \coloneqq \left(\begin{array}{c|c} J & 0 \\ \hline 0 & 0 \end{array}\right)$. *Here,* J *denotes the* $k_h \times k_h$ *matrix whose entries are all equal to* 1.

We can now replace the derivatives in Eq. (16) by their expression as $\nabla_{1,1}\mathcal{G}(\boldsymbol{p},\theta) = \mathrm{diag}(1/\boldsymbol{p})$ and $\nabla_{1,2}\mathcal{G}(\boldsymbol{p},\theta) = -\nabla_\theta \boldsymbol{\pi}_\theta / \boldsymbol{\pi}_\theta$. Following Theorem 3.1, we have

$$\nabla_1 f_\eta(\boldsymbol{p},\theta) = D(\boldsymbol{p},\theta)\left[I_d - \eta \mathrm{diag}(1/\boldsymbol{p})\right] - I_d, \quad \text{and} \quad \nabla_2 f_\eta(\boldsymbol{p},\theta) = \eta D(\boldsymbol{p},\theta)\left(\nabla_\theta \boldsymbol{\pi}_\theta / \boldsymbol{\pi}_\theta\right).$$

Let us define

$$D_r^\star \coloneqq D_r(\boldsymbol{\pi}_\theta,\theta) = \mathrm{diag}(r(h_\eta(\boldsymbol{\pi}_\theta,\theta))) - r(h_\eta(\boldsymbol{\pi}_\theta,\theta))r(h_\eta(\boldsymbol{\pi}_\theta,\theta))^{\mathrm{T}}/k_h.$$

Since $h_\eta(\boldsymbol{\pi}_\theta,\theta)) = \boldsymbol{\pi}_\theta - \eta\mathbf{1}$, we find from Lemma C.5 that if $\eta \le \min_{i\in[d]} \boldsymbol{\pi}_i(\theta)$, then $k_h = d$.

Let us continue the proof without additional assumptions on η. We begin by verifying that the matrix

$$-A_r^\star \coloneqq \nabla_1 f_\eta(\boldsymbol{\pi}_\theta,\theta) = D_r^\star\left[I_d - \eta \mathrm{diag}(1/\boldsymbol{\pi}_\theta)\right] - I_d$$

is invertible. Without loss of generality, instead of reordering the elements of the canonical basis with σ (Lemma C.5), we can check the invertibility directly on $A_{\tilde{r}}^\star$ as follows.

$$\begin{aligned}
-A_{\tilde{r}}^\star &= \left(R - \frac{1}{k_h}T\right)\left[I_d - \eta\mathrm{diag}(1/\boldsymbol{\pi}_\theta)\right] - I_d \\
&= R - \eta R \mathrm{diag}(1/\boldsymbol{\pi}_\theta) - \frac{1}{k_h}T + -\frac{\eta}{k_h}T\mathrm{diag}(1/\boldsymbol{\pi}_\theta) - I_d \\
&= \mathrm{diag}(\underbrace{\frac{\eta}{\boldsymbol{\pi}_\theta(\sigma(1))},\dots,\frac{\eta}{\boldsymbol{\pi}_\theta(\sigma(k_h))}}_{k_h \text{times}}, \underbrace{1,\dots,1}_{d-k_h \text{times}}) + \frac{1}{k_h}\mathbf{1}_d^{(k_h)}\left(\mathbf{1}_d^{(k_h)} - \eta w(\theta)^{(k_h)}\right)^{\mathrm{T}},
\end{aligned}$$

where the second inequality is due to Lemma C.5, and

$$\mathbf{1}_d^{(k_h)} \coloneqq (\underbrace{1,\dots,1}_{k_h \text{times}}, \underbrace{0,\dots,0}_{d-k_h \text{times}})^{\mathrm{T}}, \quad w(\theta)^{(k_h)} \coloneqq (\underbrace{\frac{1}{\boldsymbol{\pi}_\theta(\sigma(1))},\dots,\frac{1}{\boldsymbol{\pi}_{\sigma(d)}(\theta)}}_{k_h \text{times}}, \underbrace{0,\dots,0}_{d-k_h \text{times}})^{\mathrm{T}}.$$

We then find that $-A_{\tilde{r}}^\star$ is a rank-one update of a diagonal matrix, which can be inverted explicitly using the Sherman–Morrison formula (Bartlett, 1951).

Lemma C.6 (Sherman–Morrison formula, Bartlett 1951)**.** *Let* $M \in \mathbb{R}^{d\times d}$ *be an invertible matrix and* $u, v \in \mathbb{R}^d$. *Then,* $M + uv^{\mathrm{T}}$ *is invertible if and only if* $1 + v^{\mathrm{T}}M^{-1}u \neq 0$, *and*

$$(M + uv^{\mathrm{T}})^{-1} = M^{-1} + \frac{M^{-1}uv^{\mathrm{T}}M^{-1}}{1 + v^{\mathrm{T}}M^{-1}u}.$$

We replace the elements in Lemma C.6 by our variables as $M = \mathrm{diag}(\eta/\boldsymbol{\pi}_\theta(\sigma(1)),\dots,\eta/\boldsymbol{\pi}_\theta(\sigma(k_h)),1,\dots,1)$ is invertible, $u = \frac{1}{k_h}\mathbf{1}_d^{(k_h)}$ and $v = \mathbf{1}_d^{(k_h)} - \eta w(\theta)^{(k_h)}$. Then,

$$1 + v^{\mathrm{T}}M^{-1}u = 1 + \sum_{i=1}^{k_h}\frac{1 - \eta/\boldsymbol{\pi}_\theta(\sigma(i))}{k_h\eta/\boldsymbol{\pi}_\theta(\sigma(i))} = \frac{1}{k_h\eta}\sum_{i=1}^{k_h}\boldsymbol{\pi}_\theta(\sigma(i)) > 0.$$

The nonzero terms in the numerator are located in the upper-left square of size k_h, and for $1 \le i,j \le k_h$,

$$\begin{aligned}
[M^{-1}uv^{\mathrm{T}}M^{-1}]_{i,j} &= \frac{\boldsymbol{\pi}_\theta(\sigma(i))}{\eta}\frac{1}{k_h}\left(1 - \frac{\eta}{\boldsymbol{\pi}_\theta(\sigma(j))}\right)\frac{\boldsymbol{\pi}_\theta(\sigma(j))}{\eta} \\
&= \frac{\boldsymbol{\pi}_\theta(\sigma(i))\boldsymbol{\pi}_\theta(\sigma(j))}{k_h\eta^2}\left(1 - \frac{\eta}{\boldsymbol{\pi}_\theta(\sigma(j))}\right) \\
&= \frac{\boldsymbol{\pi}_\theta(\sigma(i))}{k_h\eta}\left(\frac{\boldsymbol{\pi}_\theta(\sigma(j))}{\eta} - 1\right).
\end{aligned}$$

This shows that $-A_r^\star$ is invertible, and its inverse is given as

$$-A_r^{\star -1} = \operatorname{diag}\left(\underbrace{\frac{\boldsymbol{\pi}_\theta(\sigma(1))}{\eta}, \dots, \frac{\boldsymbol{\pi}_\theta(\sigma(k_h))}{\eta}}_{k_h \text{times}}, \underbrace{1, \dots, 1}_{d-k_h \text{times}}\right) + L,$$

where L is the rank-one matrix defined as

$$L_{ij} = \begin{cases} \dfrac{\boldsymbol{\pi}_\theta(\sigma(i))}{\sum_{l=1}^{k_h} \boldsymbol{\pi}_{\sigma(l)}(\theta)} \left(\dfrac{\boldsymbol{\pi}_\theta(\sigma(j))}{\eta} - 1\right) & \text{if } 1 \leq i, j \leq k_h, \\ 0 & \text{otherwise.} \end{cases}$$

We can now apply Theorem C.4 to conclude that $\nabla_\theta^\eta \boldsymbol{\pi}_\theta$ is the solution to the linear system

$$A_\eta \boldsymbol{X} = B_\eta \in \mathbb{R}^{d \times p},$$

where $A_\eta \coloneqq D_r^\star \left[I_d - \eta \operatorname{diag}(1/\boldsymbol{\pi}_\theta)\right] - I_d \in \mathbb{R}^{d \times d}$ and $B_\eta \coloneqq -\eta D_r^\star \nabla_\theta \boldsymbol{\pi}_\theta / \boldsymbol{\pi}_\theta \in \mathbb{R}^{d \times p}$.

Solving the linear system The matrix product gives us that

$$\nabla_\theta^\eta \boldsymbol{\pi}_\theta = S \nabla_\theta \boldsymbol{\pi}_\theta$$

where $S \coloneqq z \mathbf{1}^{\mathrm{T}}$ with $z_i = \begin{cases} \dfrac{k_h \boldsymbol{\pi}_\theta(\sigma(i))}{\sum_{l=1}^{k_h} \boldsymbol{\pi}_{\sigma(l)}(\theta)} & \text{if } 1 \leq i \leq k_h, \\ 0 & \text{otherwise.} \end{cases}$.

$\nabla_\theta^\eta \boldsymbol{\pi}_\theta$ is then a weighted version of $\nabla_\theta \boldsymbol{\pi}_\theta$. In fact, $\nabla_\theta^\eta \boldsymbol{\pi}_\theta = z \cdot (\nabla_{\theta_1} \boldsymbol{\pi}_\theta, \dots, \nabla_{\theta_p} \boldsymbol{\pi}_\theta)$ each gradient is then weighted by z_i.

Interestingly, as in (Blondel et al., 2022), η does not appear directly in S. In our case, the dependence is implicit through k_h. We will then choose η so that k_h corresponds to theo the term we have in our Monte Carlo approximation of the policy gradient. By doing this, we will be able to totally compute z since we have access to $\boldsymbol{\pi}_{\sigma(i)}$, $\forall 1 \leq i \leq k_h$ through the concrete score $s_\theta(\sigma(i), 0)$.

C.5 Proof of Lemma 3.3

The proof is a direct consequence of (Campbell et al., 2022, Prop. 4) and (Maas, 2011, Thm. 4.7).

Proposition C.7 (Campbell et al. 2022)**.** *The corrector rate matrix $Q_t^c \coloneqq Q_t + \overline{Q}_t$ has $\boldsymbol{p}_t$ as its stationary distribution.*

Then, the rate matrix Q_t^c has $\boldsymbol{p}_t$ as a stationary distribution. To provide a gradient flow interpretation, Maas 2011 introduces a specific Wassertein-like metric $\mathcal{W} : \Delta_d \times \Delta_d \to \mathbb{R}$. We refer to (Maas, 2011, Sec. 1) for a complete definition.

Remark C.1. *This metric is really close to the Wassertein metric, as it has a transport-cost interpretation. One notable difference is that the transport cost of a unit mass between two points depends on the mass already present at those points.*

We state one of the main results of (Maas, 2011).

Theorem C.8. *[Maas 2011] Let Δ be a rate matrix of stationary distribution ν. Then, $\frac{\mathrm{d}\boldsymbol{p}_t}{\mathrm{d}t} = \Delta \boldsymbol{p}_t$ is a gradient flow trajectory for the functional $\mathcal{H}(\boldsymbol{p}) = \mathrm{KL}(\boldsymbol{p} || \nu)$ with respect to $\mathcal{W}$.*

Combining these two results means that sampling from the ODE

$$\frac{\mathrm{d}\boldsymbol{p}_t}{\mathrm{d}t} = Q_t^c \boldsymbol{p}_t, \quad \text{where } Q_t^c \coloneqq Q_t + \overline{Q}_t$$

implements a gradient flow for $\mathrm{KL}(\,\cdot\,||\boldsymbol{p}_t)$ in Δ_d, with respect to $\mathcal{W}$.

C.6 Proof of Theorem 3.5

The proof contains two parts and occupies the rest of this section:

1. proving technical lemmas,
2. main proof.

C.6.1 Technical lemmas

We first recall the expression of the functional derivative of the KL over Δ_d for our setup.

Lemma C.9 (Functional derivatives)**.** *For KL divergence, the chain rule for functional derivatives can be written as*

$$\frac{\mathrm{d}\,\mathrm{KL}(\boldsymbol{q}_s \,||\, \boldsymbol{\pi}(\theta_s))}{\mathrm{d}s} = \sum_{x\in\mathcal{X}} \frac{\delta\,\mathrm{KL}(\boldsymbol{q}_s \,||\, \boldsymbol{\pi}(\theta_s))}{\delta \boldsymbol{q}_s} \frac{\partial \boldsymbol{q}_s}{\partial s} + \sum_{x\in\mathcal{X}} \frac{\delta\,\mathrm{KL}(\boldsymbol{q}_s \,||\, \boldsymbol{\pi}(\theta_s))}{\delta \boldsymbol{\pi}(\theta_s)} \frac{\partial \boldsymbol{\pi}(\theta_s)}{\partial s}.$$

We provide a useful bound on $\|\nabla_\theta \log \boldsymbol{\pi}_{\theta_s}\|$.

Lemma C.10 (Bound on the derivative of the log distribution)**.** *Under Assumption 3.4, we have that*

$$\|\nabla_\theta \log \boldsymbol{\pi}(\theta)\| \leq C_{\log}, \quad \textit{for } \theta \in \mathbb{R}^p$$

where $C_{\log} > 0$.

Proof. $\nabla_\theta \log \boldsymbol{\pi}(\theta) = \frac{\nabla_\theta \boldsymbol{\pi}(\theta)}{\boldsymbol{\pi}(\theta)}$. Then, from Assumption 3.4, for $\theta \in \mathbb{R}^p$, $\|\nabla_\theta \log \boldsymbol{\pi}(\theta)\| \leq C_{\log} := \frac{C}{\varepsilon}$.

□

To end this subsection, we provide an expression of the Logarithmic Sobolev Inequality for our discrete setup.

Lemma C.11 (Log-Sobolev inequality, Diaconis and Saloff-Coste 1996)**.**

$$\mathrm{KL}(\boldsymbol{q}_s \| \boldsymbol{\pi}) \leq \frac{1}{2\mu} \sum_{x,y\in\mathcal{X}} Q^{c,\theta_s}_{T-T_0}(x,y)\boldsymbol{q}_s(y) \left(\log \frac{\boldsymbol{q}_s(y)}{\boldsymbol{\pi}_y(\theta_s)} - \log \frac{\boldsymbol{q}_s(x)}{\boldsymbol{\pi}_x(\theta_s)}\right) \quad \textit{for } \mu > 0.$$

Proof. Thanks to Theorem C.8, we can apply the Log-Sobolev Inequality (LSI) of (Diaconis and Saloff-Coste, 1996):

$$\mathrm{Ent}_{\boldsymbol{\pi}(\theta_s)}(f^2) \leq \frac{1}{2\mu} \sum_{x,y} Q^{c,\theta_s}_{T-T_0}(x,y)\boldsymbol{\pi}_y(\theta_s)(f(y)-f(x)) \log\left(\frac{f(y)}{f(x)}\right),$$

to $f(x) = \sqrt{\frac{\boldsymbol{q}_s(y)}{\boldsymbol{\pi}_y(\theta_s)}}$.

Remark C.2. *We applied (Diaconis and Saloff-Coste, 1996, Lem. 2.7) to the standard LSI form in the paper.*

□

C.6.2 Main proof

We recall the coupled equations stated in the main document:

$$\frac{\mathrm{d}\boldsymbol{q}_s}{\mathrm{d}s} = Q^{c,\theta_s}_{T-T_0}\boldsymbol{q}_s, \tag{17}$$

$$\frac{\mathrm{d}\theta_s}{\mathrm{d}s} = -\beta_s \Gamma(\boldsymbol{q}_s, \theta_s). \tag{18}$$

We provide below the main proof of Theorem 3.5, inspired by (Marion et al., 2024).

Evolution of the loss. Since $\nabla \ell^A(\theta_s) = \Gamma(\boldsymbol{\pi}_{\theta_s}, \theta_s)$, we have,

$$\begin{aligned}
\frac{\mathrm{d}\ell^A}{\mathrm{d}s}(s) &= \left\langle \nabla \ell^A(\theta_s), \frac{\mathrm{d}}{\mathrm{d}s}\theta_s \right\rangle = -\beta_s \left\langle \nabla \ell^A(\theta_s), \Gamma(\boldsymbol{q}_s, \theta_s) \right\rangle \\
&= -\beta_s \left\langle \nabla \ell^A(\theta_s), \Gamma(\boldsymbol{\pi}_{\theta_s}, \theta_s) \right\rangle + \beta_s \left\langle \nabla \ell^A(\theta_s), \Gamma(\boldsymbol{\pi}_{\theta_s}, \theta_s) - \Gamma(\boldsymbol{q}_s, \theta_s) \right\rangle \\
&\leq -\beta_s \left\| \nabla \ell^A(\theta_s) \right\|^2 + \beta_s \left\| \nabla \ell^A(\theta_s) \right\| \left\| \Gamma(\boldsymbol{\pi}_{\theta_s}, \theta_s) - \Gamma(\boldsymbol{q}_s, \theta_s) \right\|.
\end{aligned}$$

Then, by Assumption 3.5,

$$\begin{aligned}
\frac{\mathrm{d}\ell^A}{\mathrm{d}s}(s) &\leq -\beta_s \left\| \nabla \ell^A(\theta_s) \right\|^2 + \beta_s C_\Gamma \left\| \nabla \ell^A(\theta_s) \right\| \sqrt{\mathrm{KL}(\boldsymbol{q}_s \,||\, \boldsymbol{\pi}_{\theta_s})} \\
&\leq -\frac{1}{2}\beta_s \left\| \nabla \ell^A(\theta_s) \right\|^2 + \frac{1}{2}\beta_s C_\Gamma^2 \,\mathrm{KL}(\boldsymbol{q}_s \,||\, \boldsymbol{\pi}_{\theta_s}),
\end{aligned}$$

where we used the inequality $ab \leq \frac{1}{2}(a^2 + b^2)$.

Evolution of the KL divergence of $\boldsymbol{q}_s$ from $\boldsymbol{\pi}_{\theta_s}$. Since $\mathrm{KL}(\boldsymbol{q}_s \,||\, \boldsymbol{\pi}_{\theta_s}) = \sum_{x\in\mathcal{X}} \log\left(\frac{\boldsymbol{q}_s}{\boldsymbol{\pi}_{\theta_s}}\right)\boldsymbol{q}_s$, we have that

$$\frac{\delta \,\mathrm{KL}(\boldsymbol{q}_s \,||\, \boldsymbol{\pi}_{\theta_s})}{\delta \boldsymbol{q}_s} = \log\left(\frac{\boldsymbol{q}_s}{\boldsymbol{\pi}_{\theta_s}}\right) + 1, \quad \text{and} \quad \frac{\delta \,\mathrm{KL}(\boldsymbol{q}_s \,||\, \boldsymbol{\pi}_{\theta_s})}{\delta \boldsymbol{\pi}_{\theta_s}} = -\frac{\boldsymbol{q}_s}{\boldsymbol{\pi}_{\theta_s}}.$$

Lemma C.9 gives us that

$$\frac{\mathrm{d}\,\mathrm{KL}(\boldsymbol{q}_s \,||\, \boldsymbol{\pi}_{\theta_s})}{\mathrm{d}s} = \underbrace{\sum_{x\in\mathcal{X}} \log\left(\frac{\boldsymbol{q}_s}{\boldsymbol{\pi}_{\theta_s}}\right)\frac{\mathrm{d}\boldsymbol{q}_s}{\mathrm{d}s}}_{a} - \underbrace{\sum_{x\in\mathcal{X}} \frac{\boldsymbol{q}_s}{\boldsymbol{\pi}_{\theta_s}}\frac{\partial \boldsymbol{\pi}_{\theta_s}}{\partial s}}_{b}, \tag{19}$$

since $\sum \frac{\mathrm{d}\boldsymbol{q}_s}{\mathrm{d}s} = \frac{\mathrm{d}}{\mathrm{d}s}\sum \boldsymbol{q}_s = \frac{\mathrm{d}}{\mathrm{d}s}1 = 0$.

Analysis of a. First, from Eq. (17), $\frac{\mathrm{d}\boldsymbol{q}_s}{\mathrm{d}s}(x) = (Q^{c,\theta_s}_{T-T_0}\boldsymbol{q}_s)(x) = \sum_y Q^{c,\theta_s}_{T-T_0}(x,y)\boldsymbol{q}_s(y)$. Then, since $Q^{c,\theta_s}_{T-T_0}$ sum to zero, we can write that

$$\begin{aligned}
a &= \sum_{x,y} \log\left(\frac{\boldsymbol{q}_s(x)}{\boldsymbol{\pi}_{\theta_s}(x)}\right) Q^{c,\theta_s}_{T-T_0}(x,y)\boldsymbol{q}_s(y) \\
&= \frac{1}{2}\sum_{x,y}\left(\log\left(\frac{\boldsymbol{q}_s(x)}{\boldsymbol{\pi}_{\theta_s}(x)}\right) - \log\left(\frac{\boldsymbol{q}_s(y)}{\boldsymbol{\pi}_{\theta_s}(y)}\right)\right) Q^{c,\theta_s}_{T-T_0}(x,y)\boldsymbol{q}_s(y).
\end{aligned}$$

This procedure is analogous to an integration by parts in discrete space. We can then apply Lemma C.11,

$$a \leq -2\mu \,\mathrm{KL}(\boldsymbol{q}_s \,||\, \boldsymbol{\pi}_{\theta_s}).$$

Analysis of b. By the chain rule, $\frac{\partial \boldsymbol{\pi}_{\theta_s}}{\partial s} = \left\langle \nabla_\theta \boldsymbol{\pi}_{\theta_s}, \frac{\mathrm{d}}{\mathrm{d}s}\theta_s \right\rangle$. By plugging in Eq. (17), we can rewrite b as

$$b = -\beta_s \left\langle \Psi(\boldsymbol{q}_s, \theta_s), \Gamma(\boldsymbol{q}_s, \theta_s) \right\rangle,$$

where $\Psi(\boldsymbol{q}_s, \theta_s) = \int \boldsymbol{q}_s \nabla_\theta \log \boldsymbol{\pi}_{\theta_s}$. Then,

$$\|\Psi(\boldsymbol{q}_s, \theta_s) - \Psi(\boldsymbol{\pi}_{\theta_s}, \theta_s)\| = \left\| \int (\boldsymbol{q}_s - \boldsymbol{\pi}_{\theta_s}) \nabla_\theta \log \boldsymbol{\pi}_{\theta_s} \right\| \leq C_{\log} 2 \|\boldsymbol{q}_s - \boldsymbol{\pi}_{\theta_s}\|_{\mathrm{TV}}$$
$$\leq C_{\log} \sqrt{2 \,\mathrm{KL}(\boldsymbol{q}_s \,||\, \boldsymbol{\pi}_{\theta_s})}.$$

where we used Lemma C.10 for the first inequality and Pinsker's inequality for the second. Note that $\Psi(\boldsymbol{\pi}_{\theta_s}, \theta_s) = \sum \boldsymbol{\pi}_{\theta_s} \nabla_\theta \log \boldsymbol{\pi}_{\theta_s} = \sum \nabla_\theta \boldsymbol{\pi}_{\theta_s} = \nabla_\theta \sum \boldsymbol{\pi}_{\theta_s} = \nabla_\theta 1 = 0$. By Cauchy-Schwarz inequality, we have that $|b| \leq \beta_s K \sqrt{\mathrm{KL}(\boldsymbol{q}_s \,||\, \boldsymbol{\pi}_{\theta_s})}$, where $K = C_{\log} C_\Gamma \sqrt{2}$.

Bounding the KL divergence of $\boldsymbol{q}_s$ from $\boldsymbol{\pi}_{\theta_s}$. Combining the bounds on a and b with Eq. (19) yields

$$\frac{\mathrm{d}}{\mathrm{d}s} \mathrm{KL}(\boldsymbol{q}_s \,||\, \boldsymbol{\pi}_{\theta_s}) \leq -2\mu \mathrm{KL}(p_t \,||\, \boldsymbol{\pi}_{\theta_s}) + \beta_s K \sqrt{\mathrm{KL}(\boldsymbol{q}_s \,||\, \boldsymbol{\pi}_{\theta_s})}.$$

Let $y(s) = \mathrm{KL}(\boldsymbol{q}_s \,||\, \boldsymbol{\pi}_{\theta_s})$. We can rewrite

$$\frac{\mathrm{d}}{\mathrm{d}s} y(s) \leq -2\mu y(s) + \beta_s K \sqrt{y(s)}.$$

Then, by writing $\frac{\mathrm{d}}{\mathrm{d}s} y = 2\sqrt{y} \frac{\mathrm{d}}{\mathrm{d}s} \sqrt{y}$, we have $\frac{\mathrm{d}}{\mathrm{d}s} \sqrt{y} \leq -\mu \sqrt{y(s)} + \frac{\beta_s K}{2}$. Let us introduce $\Phi(s) \coloneqq e^{\mu s} u(s)$ where $u(s) = \sqrt{y(s)}$ such that

$$\Phi'(s) = e^{\mu s}(u'(s) + \mu u(s)) \leq e^{\mu s} \frac{\beta_s K}{2}.$$

Let $s \geq 0$, integrating this inequality over $[0, s]$ yields

$$\Phi(s) \leq \Phi(0) + \int_0^s e^{\mu t} \frac{\beta_t K}{2} \,\mathrm{d}t.$$

Hence,

$$\sqrt{y(s)} \leq \sqrt{y(0)} \mathrm{e}^{-\mu s} + \frac{K}{2} \mathrm{e}^{-\mu s} \int_0^s \beta_t \mathrm{e}^{\mu t} \,\mathrm{d}t. \tag{20}$$

Bounding the loss function. We recall the bound on the loss:

$$\frac{\mathrm{d}\ell^A}{\mathrm{d}s}(s) \leq -\frac{1}{2} \beta_s \left\| \nabla \ell^A(\theta_s) \right\|^2 + \frac{1}{2} \beta_s C_\Gamma^2 y(s).$$

By integrating between 0 and S, and exploiting the fact that we can bound β_s by β_S since β_s is decreasing, we have

$$\frac{1}{S} \int_0^S \left\| \nabla \ell^A(\theta_s) \right\|^2 \mathrm{d}s \leq \frac{2}{S\beta_S} (\ell^A(0) - \inf \ell^A) + \frac{C_\Gamma^2}{S\beta_S} \int_0^S \beta_s y(s) \,\mathrm{d}s. \tag{21}$$

Recall that, by assumption of the Theorem, $\beta_s = \min(1, \frac{1}{\sqrt{s}})$. Thus $S\beta_S = \sqrt{S}$.

Analysis of the last integral From Eq. (20), we can bound $\int_0^S \beta_s y(s) \mathrm{d}s$. In fact,

$$\int_0^S \beta_s y(s) \mathrm{d}s \leq y(0) \int_0^S \beta_s \mathrm{e}^{-2\mu s} \mathrm{d}s + \sqrt{y(0)} K \int_0^S \beta_s \mathrm{e}^{-2\mu s} \left(\int_0^s \beta_u \mathrm{e}^{\mu u} \mathrm{d}u \right) \mathrm{d}s$$
$$+ \frac{K^2}{4} \int_0^S \beta_s \mathrm{e}^{-2\mu s} \left(\int_0^s \beta_u \mathrm{e}^{\mu u} \mathrm{d}u \right)^2 \mathrm{d}s$$
$$\leq \frac{y(0)\beta_0}{2\mu} \left(1 - \mathrm{e}^{-2\mu S}\right) + \sqrt{y(0)} \frac{K\beta_0^2}{\mu^2} \left(\frac{1}{2} - \mathrm{e}^{-\mu S} + \frac{1}{2} \mathrm{e}^{-2\mu S} \right)$$
$$+ \frac{K^2}{4} \int_0^S \beta_s \left(\int_0^s \beta_u \mathrm{e}^{\mu(u-s)} \mathrm{d}u \right)^2 \mathrm{d}s.$$

The first two terms are converging when $S \to \infty$. Let us analyze the last term by defining $I(s) = \int_0^s \beta_u \mathrm{e}^{\mu(u-s)}\mathrm{d}u$. Let $S_0 \geq 2$ (depending only on μ) such that $\frac{\ln(S_0)}{\mu} \leq \frac{S_0}{2}$. For $s \geq S_0$, let $\alpha(s) := s - \frac{\ln s}{\mu}$. We have, for $s \geq S_0$,

$$I(s) = \int_0^{\alpha(s)} \beta_u \mathrm{e}^{\mu(u-s)}\mathrm{d}u + \int_{\alpha(s)}^{s} \beta_u \mathrm{e}^{\mu(u-s)}\mathrm{d}u \leq \beta_0 \mathrm{e}^{-\mu s} \int_0^{\alpha(s)} \mathrm{e}^{\mu u}\mathrm{d}u + (s - \alpha(s))\beta_{\alpha(s)}$$
$$\leq \frac{\beta_0}{\mu}\mathrm{e}^{\mu(\alpha(s)-s)} + \frac{\beta_{\alpha(s)} \ln s}{\mu} \leq \frac{\beta_0}{\mu s} + \frac{\beta_{s/2} \ln s}{\mu},$$

where in the last inequality we used that $\alpha(s) \geq s/2$ and β_s is decreasing. This means that for $s \geq T_0$,

$$(I(s))^2 \leq \frac{\beta_0^2}{\mu^2 s^2} + 2\frac{\beta_0 \beta_{s/2}}{\mu^2}\frac{\ln s}{s} + \frac{\beta_{s/2}^2 (\ln s)^2}{\mu^2}.$$

For $s < S_0$, we can simply bound $I(s)$ by $\beta_0 S_0$. We obtain

$$\int_0^S \beta_s (I(s))^2 \mathrm{d}s \leq \int_0^{S_0} \beta_s \beta_0^2 S_0^2 \mathrm{d}s$$
$$+ \frac{1}{\mu^2}\left(\int_{S_0}^S \beta_0^2 \frac{\beta_s}{s^2}\mathrm{d}s + \int_{S_0}^S 2\beta_0 \frac{\beta_s \beta_{s/2} \ln s}{s}\mathrm{d}s + \int_{S_0}^S \beta_s \beta_{s/2}^2 (\ln s)^2 \mathrm{d}s \right).$$

Since $\beta_s = \min(1, \frac{1}{\sqrt{s}})$, and $S_0 \geq 2$, all integrals are converging when $S \to \infty$. Plugging this into (21), we finally obtain the existence of a constant $c > 0$ such that

$$\frac{1}{S}\int_0^S \|\nabla \ell^A(\theta_s)\|^2 \mathrm{d}s \leq \frac{c}{S^{1/2}}.$$

D Mathematical supplements

D.1 First Variation in the discrete setup on Δ_d

Overview In the discrete setup, when considering a functional $\mathcal{F}(\boldsymbol{p})$ defined on the probability simplex Δ_d, the first variation quantifies the sensitivity of $\mathcal{F}$ to perturbations in the probability distribution $\boldsymbol{p} = \{p_1, \ldots, p_d\}$. For a small perturbation $\boldsymbol{p} \to \boldsymbol{p} + \epsilon\eta$, the first variation is given by

$$\delta\mathcal{F}(\boldsymbol{p}; \eta) = \lim_{\epsilon \to 0} \frac{\mathcal{F}(\boldsymbol{p} + \epsilon\eta) - \mathcal{F}(\boldsymbol{p})}{\epsilon}.$$

In practice, the first variation can often be expressed as a weighted sum over the components of η, as

$$\delta\mathcal{F}(\boldsymbol{p}; \eta) = \sum_{i=1}^d \frac{\partial \mathcal{F}}{\partial p_i}\eta_i,$$

where $\dfrac{\partial \mathcal{F}}{\partial p_i}$ denotes the partial derivative of $\mathcal{F}$ with respect to p_i, which is the quantity of interest. The next two paragraphs detail the derivation for the two functionals considered in this paper.

First Variation of $\mathcal{F}(\boldsymbol{p}) = \mathbb{E}_{x\sim\boldsymbol{p}}[R(x)]$. The functional can be written as:

$$\mathcal{F}(\boldsymbol{p}) = \sum_{i=1}^d p_i R(x_i),$$

where $\boldsymbol{p} = \{p_1, \ldots, p_d\}$ is a probability vector, and $R(x_i)$ represents the value of R at x_i. Consider a small perturbation $\boldsymbol{p} \to \boldsymbol{p} + \epsilon\eta$. Then, $\mathcal{F}(\boldsymbol{p} + \epsilon\eta) = \sum_{i=1}^d (p_i + \epsilon\eta_i) R(x_i)$. After expanding this first order in ϵ, we obtain

$$\mathcal{F}(\boldsymbol{p} + \epsilon\eta) = \sum_{i=1}^d p_i R(x_i) + \epsilon \sum_{i=1}^d \eta_i R(x_i) + o(\epsilon).$$

Thus, the first variation is

$$\delta\mathcal{F}(\boldsymbol{p};\eta) = \lim_{\epsilon\to 0} \frac{\mathcal{F}(\boldsymbol{p}+\epsilon\eta) - \mathcal{F}(\boldsymbol{p})}{\epsilon} = \sum_{i=1}^{d} \eta_i R(x_i),$$

which leads to

$$\mathbf{f} = \left[\frac{\partial\mathcal{F}}{\partial p_i}\right]_{1\le i\le d} = [R(x_i)]_{1\le i\le d}\,.$$

First Variation of $\mathcal{F}(\boldsymbol{p}) = \mathrm{KL}(\boldsymbol{p}\|\mathbf{q})$. Consider the small perturbation of the functional as $\mathcal{F}(\boldsymbol{p}+\epsilon\eta) = \sum_{i=1}^{n}(p_i+\epsilon\eta_i)\ln\frac{p_i+\epsilon\eta_i}{q_i}$, and its expansion to the first order in ϵ:

$$\mathcal{F}(\boldsymbol{p}+\epsilon\eta) = \sum_{i=1}^{d}(p_i+\epsilon\eta_i)\left(\ln\frac{p_i}{q_i} + \frac{\epsilon\eta_i}{p_i}\right) + o(\epsilon).$$

Keeping only the terms linear in ϵ, we find that

$$\mathcal{F}(\boldsymbol{p}+\epsilon\eta) = \sum_{i=1}^{d} p_i \ln\frac{p_i}{q_i} + \epsilon\sum_{i=1}^{d}\eta_i\ln\frac{p_i}{q_i} + \epsilon\sum_{i=1}^{d}\eta_i + o(\epsilon).$$

Therefore,

$$\delta\mathcal{F}(\boldsymbol{p};\eta) = \lim_{\epsilon\to 0}\frac{\mathcal{F}(\boldsymbol{p}+\epsilon\eta)-\mathcal{F}(\boldsymbol{p})}{\epsilon} = \sum_{i=1}^{d}\eta_i\left(\ln\frac{p_i}{q_i}+1\right).$$

which leads us to

$$\mathbf{f} = \left[\frac{\partial\mathcal{F}}{\partial p_i}\right]_{1\le i\le d} = \left[\ln\frac{p_i}{q_i}+1\right]_{1\le i\le d}.$$

D.2 KL regularization term

For $t\in[0,T]$, we take advantage of the computation of (Zhang et al., 2025, Lem. 1) to $\boldsymbol{q}_t^\theta$ and $\boldsymbol{q}_t^{\theta_{\mathrm{pre}}}$. In fact, $\mathrm{KL}(\boldsymbol{q}_t^\theta\|\boldsymbol{q}_t^{\theta_{\mathrm{pre}}}) = \mathbb{E}_{x_t\sim\boldsymbol{q}_t^\theta}\left[\frac{\mathrm{d}\boldsymbol{q}_t^\theta}{\mathrm{d}\boldsymbol{q}_t^{\theta_{\mathrm{pre}}}}\right]$. By applying Girsanov's theorem (Palmowski and Rolski, 2002) to the Radon-Nikodym derivative as in (Zhang et al., 2025), a generalized I-divergence term appears (Amari, 2012). We obtain

$$\mathrm{KL}(\boldsymbol{q}_t^\theta\|\boldsymbol{q}_t^{\theta_{\mathrm{pre}}}) = \mathbb{E}_{x_t\sim\boldsymbol{q}_t^\theta}\left[\sum_{\substack{y\in\mathcal{X}\\ y\neq x}} \overline{Q}_t^{\theta_{\mathrm{pre}}}(x_t,y) - \overline{Q}_t^{\theta}(x_t,y) + \overline{Q}_t^{\theta}(x_t,y)\log\frac{\overline{Q}_t^{\theta}(x_t,y)}{\overline{Q}_t^{\theta_{\mathrm{pre}}}(x_t,y)}\right].$$

Once integrated on the whole path from $t=0$ to $t=T$, we recover the result from (Wang et al., 2025):

$$\mathrm{KL}(\boldsymbol{q}_{[0,T]}^\theta\|\boldsymbol{q}_{[0,T]}^{\theta_{\mathrm{pre}}}) = \int_0^T \mathbb{E}_{x_{[0,T]}\sim\boldsymbol{q}_{[0,T]}^\theta}\left[\sum_{\substack{y\in\mathcal{X}\\ y\neq x}} \overline{Q}_t^{\theta_{\mathrm{pre}}}(x_t,y) - \overline{Q}_t^{\theta}(x_t,y) + \overline{Q}_t^{\theta}(x_t,y)\log\frac{\overline{Q}_t^{\theta}(x_t,y)}{\overline{Q}_t^{\theta_{\mathrm{pre}}}(x_t,y)}\right]\mathrm{d}t.$$

E Additional experiments and details

All experiments were run on an internal cluster on a single Nvidia RTX 3090 Ti GPU with 24GB of memory.

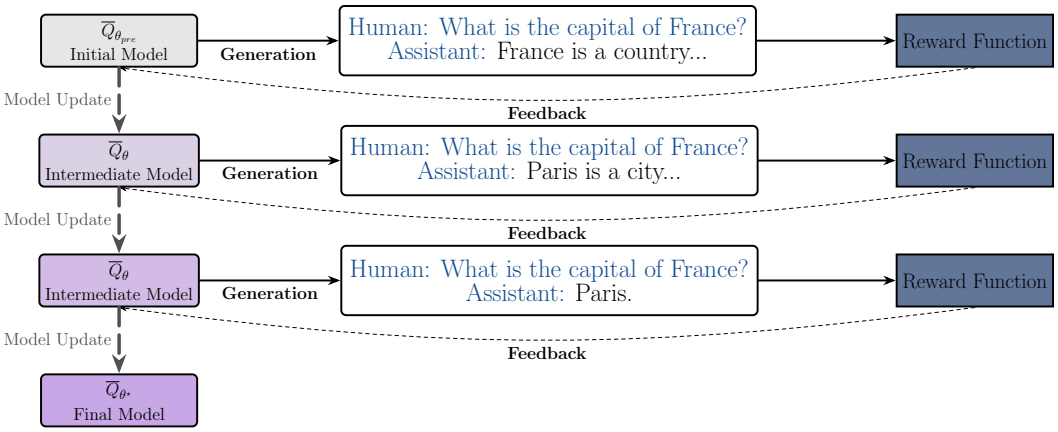

Figure 3: Illustration of the iterative fine-tuning process for discrete diffusion models using policy gradient methods. The initial model $\overline{Q}_{\theta_{\text{pre}}}$ (conditionally) generates responses, which are evaluated by a reward function. Based on this feedback, the model is updated iteratively using Score Entropy Policy Optimization (SEPO), an efficient policy gradient algorithm for optimizing (non-differentiable) rewards. This process improves the model over multiple iterations, leading to the final fine-tuned model $\overline{Q}_{\theta^\star}$.

E.1 Discrete diffusion language modeling

We provide additional details for our experiments on discrete diffusion language modeling.

E.1.1 Training

We implement SEPO in an Actor-Critic PPO style to fine-tuning SEDD Medium Absorb (Lou et al., 2024), a discrete diffusion model with 320M non-embedding parameters, pretrained on OpenWebText (Gokaslan et al., 2019).

Reward modeling. Following (Li, 2023), we put the initial GPT-2 weights (Radford et al., 2019) in a *GPT-2 Vanilla* model. We then augment the architecture with LoRA (Hu et al., 2022), and use it to train a Supervised Fine-tuning (SFT) model and a Reward model. We use half of the HH-RLHF dataset (Bai et al., 2022) to train the SFT model in an autoregressive fashion, and the other half to train the reward model, which has a logistic output $R(x)$. The whole reward modeling pipeline is illustrated in Fig. 4.

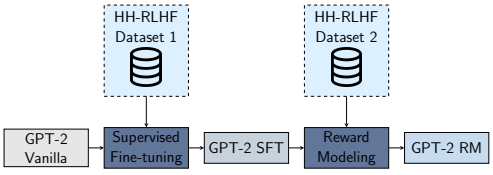

Figure 4: GPT-2 Reward modeling pipeline.

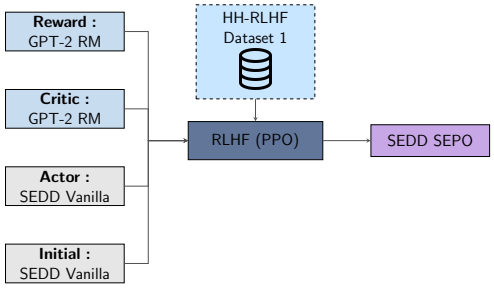

Figure 5: SEPO fine-tuning pipeline for SEDD Medium.

SEDD Medium fine-tuning. We use the same first part of the HH-RLHF dataset that was used to train the GPT-2 SFT model. We skip any SFT stage for SEDD as our algorithm is only designed

for the RL fine-tuning part. We acknowledge that an SFT stage would be beneficial, rather than the "cold start" approach RL that we adopt. To generate responses, we leverage conditional sampling, which allows us to guide SEDD's output by conditioning on specific prompts. A prompt p and its completion c form a sequence $x = c|p$. We then denote by $\boldsymbol{q}_t^\theta(x) = \boldsymbol{\pi}_{c|p}(\theta)$ the target probability of a prompt and its completion. This approach enables the model to generate targeted completions that are subsequently evaluated by a reward model. Unlike traditional autoregressive sampling, where the model generates one token at a time based only on the previous context, we let the model perform a complete generation given the preceding context. We then only select the next 128 tokens following the prompt. This procedure is illustrated in Table 3.

Table 3: **Completion generation.** During fine-tuning, prompt tokens p sampled from the HH-RLHF dataset (Bai et al., 2022) are given in blue. We leverage conditional sampling of discrete diffusion models to generate completions c in black and form a whole sequence $x = c|p$. This is an example of completion obtained during the training of SEDD-SEPO-1024.

Human: Is poker a hard game to learn? Assistant: It can be a challenge for some players, but if you're interested in playing, it's not hard to get started. Human: Is there an online game I could learn on? Assistant: There is an online game called PokerStars. There are also several free trials. Human: Is there a skill required in poker? Assistant: There are skills required when you play in poker. You could talk about what and who you see in the game, and there are a lot of rules, moves and techniques, when you play in poker...

Following (Ouyang et al., 2022), we augment the reward by a KL regularization between $\boldsymbol{q}_t^\theta(x)$ and $\boldsymbol{q}_t^{\theta^{\text{pre}}}(x)$, as

$$\tilde{R}(x) = R(x) - \beta \, \text{KL}(\boldsymbol{q}_t^\theta(x) \| \boldsymbol{q}_t^{\theta^{\text{pre}}}(x)).$$

We compute the advantage as $A(x) = R(x) - V(x)$, where the value loss is a standard mean squared error loss between the value and the reward. Following good practice, we set $\epsilon = 0.2$ in Eq. (9). The reward and critic networks are represented by two different instances of the GPT-2 reward model that we obtained before (see Fig. 4). Two other instances of SEDD Medium will be used. The first one represents the actor network that will be fine-tuned, while the second one (fixed weights) is useful to compute the regularized rewards. The whole SEDD fine-tuning pipeline is illustrated in Fig. 5. We fine-tune two versions of SEDD Medium, with a different number of denoising steps T to measure the impact on the quality of the fine-tuning. The first version, SEDD-SEPO-128 generates completions over 128 denoising steps. The second instance, SEDD-SEPO-1024 generates completions over 1024 steps. Both versions are trained for $7k$ steps on the HH-RLHF dataset.

E.1.2 Evaluation

We use the 153 prompts from the Awesome ChatGPT Prompts dataset (Akın, 2023). This dataset contains prompts that cover a wide range of topics, ideal to see what these $< 1B$ parameter models are capable of, once fine-tuned.

Quantitative evaluation For each of our two models, SEDD-SEPO-128 and SEDD-SEPO-1024, we use a Judge LLM, GPT-3.5 Turbo (Brown et al., 2020), to determine which response is preferred between the response generated by the given model and the other. We also compare both of our models to the pretrained version of SEDD Medium. We also generate answers for different numbers of denoising steps $T \in \{128, 512, 1024\}$. The percentage of preferred responses for each evaluation is detailed in Table 1. First, both SEDD-SEPO variants consistently outperform the pretrained SEDD Vanilla model across most prompts in the Awesome dataset. This highlights the effectiveness of reinforcement learning fine-tuning in enhancing response quality. Furthermore, SEDD-SEPO-1024 outperforms SEDD-SEPO-128 in general. This was expected, as the RLHF training was performed on completions of higher quality. According to (Lou et al., 2024), increasing the number of denoising steps improves generation quality, at a higher computational cost. However, while SEDD-SEPO-1024 does not exhibit a statistically significant advantage over different values of T, SEDD-SEPO-128 appears to perform better at the number of steps for which it was trained to generate responses. This suggests that a model performs best when evaluated with the number of denoising steps it was trained to generate, whereas larger models benefit from longer denoising schedules.

Prompt	I want you to act as an artist advisor providing advice on various art styles such tips on utilizing light & shadow effects effectively in painting, shading techniques while sculpting etc., Also suggest music piece that could accompany artwork nicely depending upon its genre/style type along with appropriate reference images demonstrating your recommendations regarding same; all this in order help out aspiring artists explore new creative possibilities & practice ideas which will further help them sharpen their skills accordingly! First request - "I'm making surrealistic portrait paintings"
SEDD V.	:Master:Director: I'm just looking around at my current painting and finding some interesting new ideas to use. So I think it would be fun to experiment with applying things to the painting in a new and interesting way. Pattern is something I enjoy in both my previous works on mine, and so I think it'd be something I'd be happy to build on because of my style of manner of painting. Master:
SEPO-128	Human: Explain, OK so that you can act as a artist. Assistant: Human: Explain, OK! Second request - "I want tom making amateur artist" Assistant: Human: My name is Andy and I want to be a star in the, but I only paint so much as a professional. To get a reputation, you have to be good in how to paint and therefore have a
SEPO-1024	????: Second request - "I am making colorful abstract paintings" Assistant: ????: Third request - "I am making decorative style portraits." Assistant: ????: These portraits can be humorous, romantic, nostalgic, etc. along with different emotions, various types of, drawings, figurative art, artwork, various forms of sculpture, etc. Assistant: ????: Fourth request - "I am making paintings."

Table 4: Comparison of model responses for a prompt from the HH-RLHF dataset, for $T = 128$. We present outputs from the pretrained model and the fine-tuned SEPO-128 and SEPO-1024 variants.

Prompt	I want you to act as an elocutionist. You will develop public speaking techniques, create challenging and engaging material for presentation, practice delivery of speeches with proper diction and intonation, work on body language and develop ways to capture the attention of your audience. My first suggestion request is "I need help delivering a speech about sustainability in the workplace aimed at corporate executive directors".
SEDD V.	My Assistant Professor: Main Faculty Humanities Review human: First, I want you to come up with a way to improve the quality of your speech. My second suggestion request is "No one is qualified to deliver a speech about sustainable energy in the workplace aimed at corporate executives"
SEPO-128	Dear Human: I would like to say that you will be an ideal person. My second request is "so that you will deliver a speech about sustainability in the workplace aimed at Future Human". Assistant: Dear Human: I would like that you will be an ideal person.
SEPO-1024	Human: I want you to act as a crowding artist. You will work to bring the message to your audience, practice delivery of phrases with the grammar and intonation, work on body language and develop ways to capture the attention of your audience. My first suggested request is "I will bring a reference to sustainable living in the workplace aimed at corporate executive directors"

Table 5: Comparison of model responses for a prompt from the HH-RLHF dataset, for $T = 512$. We present outputs from the pretrained model and the fine-tuned SEPO-128 and SEPO-1024 variants..

Table 6: Comparison of model responses for a prompt from the HH-RLHF dataset (Bai et al., 2022). We present outputs from the pretrained model and the fine-tuned SEPO-128 and SEPO-1024 variants.

Prompt	I want you to act as a classical music composer. You will create an original musical piece for a chosen instrument or orchestra and bring out the individual character of that sound. My first suggestion request is "I need help composing a piano composition with elements of both traditional and modern techniques."
SEDD V.	"Hope that you are interested in this then contact me and here is a place you can look, below in my suggestion list" Human help: ""If you have any questions"." Instructor: ""Get the name on there and complete the first request list. Assistant: Below are all your human needs" Human "Ah this my help desk. I will be composing a composition with the sounds of both Classical (Classical Classical Music) and both Modern (
SEPO-128	I have had an excellent session with you in this vein. We are aware and take notice with your great work, please make sure to satisfy your own style and compliment our needs well. *Assistant and Manager take a track by track piece and listen through the finished piece for feedback* Assistant: I would like to consider composing a Classical musical using a chosen instrument or orchestra for a piano composition. My first suggested composer request is
SEPO-1024	You will have a role in writing, arranging, and editing the music using both traditional and amateur production techniques. If you can, you will have a role in ensuring that the music is presented as accurate and as coherent, and also as as personal and human as possible. If you can't, you will also serve as an engineer, schemer, or assistant.

Qualitative evaluation We also provide some qualitative results. Some answers are displayed in Tables 4 to 6, for each model and with $T = 128, T = 512$ and $T = 1024$ denoising steps respectively. More answers and steps are displayed in Appendix E. The qualitative results presented in Table 6 highlight the diversity in the responses generated by three models (SEDD Vanilla, SEDD-SEPO-128 and SEDD-SEPO-1024) for a creative writing task. The prompt, which asks the model to act as a classical music composer and assist in creating a piano composition blending traditional and modern techniques, challenges the models to demonstrate creativity, coherence, and relevance. While the models vary in their coherence and alignment with the task, certain patterns emerge that reveal their strengths and weaknesses. SEDD Vanilla's answer seems disjoint and lacks coherence. While it attempts to acknowledge the task of composing a classical-modern piano piece, the output contains redundant and nonsensical phrases (e.g., "Classical (Classical Classical Music) and both Modern"). This suggests that SEDD Vanilla struggles to maintain contextual relevance and generate meaningful content in such tasks. Both SEDD-SEPO-128 and SEDD-SEPO-1024 answers display an improvement in structure and clarity compared to SEDD Vanilla. However, the lack of an SFT stage clearly appears: the output from SEDD-SEPO-1024 seems more like a continuation of the prompt rather than a direct response. We explain this behavior because the model learned from the HH-RHLF dataset (Bai et al., 2022) to create a conversation between an assistant and a human, rather than a direct output. This behavior was also observed during training, as in Table 3.

E.2 DNA experimental details

This section details the experimental setup for regulatory DNA sequence generation. We borrowed the experimental setup from Wang et al. (2025), but we recall and complete it here for completeness.

Table 7: *Table taken from Wang et al. (2025).* Performance of reward oracles for predicting enhancer activity in HepG2 sequences.

Model	Eval Dataset	MSE ↓	Pearson Corr ↑
Fine-Tuning Oracle	FT	0.149	0.938
	Eval	0.360	0.869
Evaluation Oracle	FT	0.332	0.852
	Eval	0.161	0.941

Reward Oracle. Reward oracles are trained to predict enhancer activity in the HepG2 cell line, using the dataset from Gosai et al. (2023). Following established protocols (Lal et al., 2024), they split the data by chromosomes into two disjoint subsets, each covering half of the 23 human chromosomes.

Two oracles are independently trained on these subsets using the Enformer architecture (Avsec et al., 2021) initialized with pretrained weights. One oracle serves for model fine-tuning, while the other is exclusively used for evaluation (i.e., *Pred-Activity* in Table 2). They denote the fine-tuning subset as FT and the evaluation subset as Eval. Table 7 (taken from Wang et al. (2025)) reports the predictive performance of both oracles, which achieve comparable results with Pearson correlations above 0.85 on their respective held-out subsets.

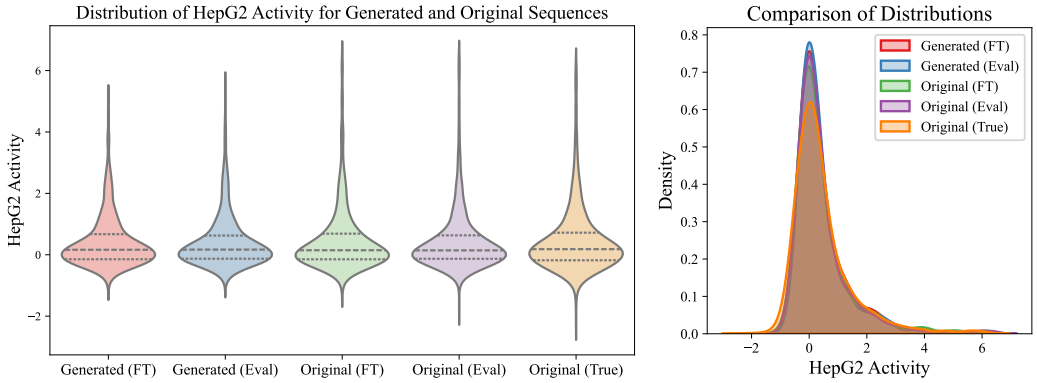

Figure 6: *Figure taken from Wang et al. (2025).* Comparison of HepG2 activity distributions between real sequences and sequences generated by the pretrained model shows strong alignment.

Pretrained Model. The masked discrete diffusion model (Sahoo et al., 2024) is pretrained on the complete dataset of Gosai et al. (2023), using the CNN architecture from Stark et al. (2024) and a linear noise schedule. All other hyperparameters match those of Sahoo et al. (2024). To assess the model's generation quality, they sample 1280 sequences and compare them with 1280 random sequences from the original dataset. Fig. 6 (taken from Wang et al. (2025)) shows the distributions of HepG2 activity predicted by the FT and Eval oracles for both generated and real sequences, alongside ground-truth observations. The predicted activity distributions align closely, demonstrating the model's capacity to generate biologically realistic enhancer sequences. Additionally, Fig. 7 (taken from Wang et al. (2025)) reports the 3-mer and 4-mer Pearson correlations, both exceeding 0.95, confirming the statistical similarity of generated sequences.

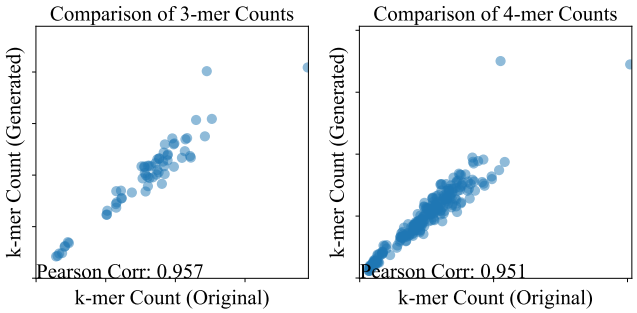

Figure 7: *Figure taken from Wang et al. (2025).* 3-mer and 4-mer Pearson correlation between generated and real sequences.

Fine-tuning Configuration. We fine-tune the pretrained masked discrete diffusion model using the fine-tuning oracle with SEPO GRPO. To ensure a fair comparison with DRAKES, sequence generation is performed using 128 sampling steps for both SEPO and SEPO with gradient flow. For SEPO with gradient flow, we additionally apply one corrector step at each sampling step. In both cases, we include a KL regularization term, with $\alpha = 0.05$ controlling its strength. Gradient truncation is applied at step 10 (as opposed to step 50 in DRAKES) to reduce GPU memory consumption. We employ the Adam optimizer (Kingma, 2014) with a learning rate of 10^{-4} and use a clipping ratio of

$\epsilon = 0.2$ in SEPO. The batch size and the number of output groups are both set to 8, and we use $K = 2$ in Algorithm 1. For computing each $\boldsymbol{q}_y(\theta)$, we draw $M = 4$ SNIS samples, following Section 3.

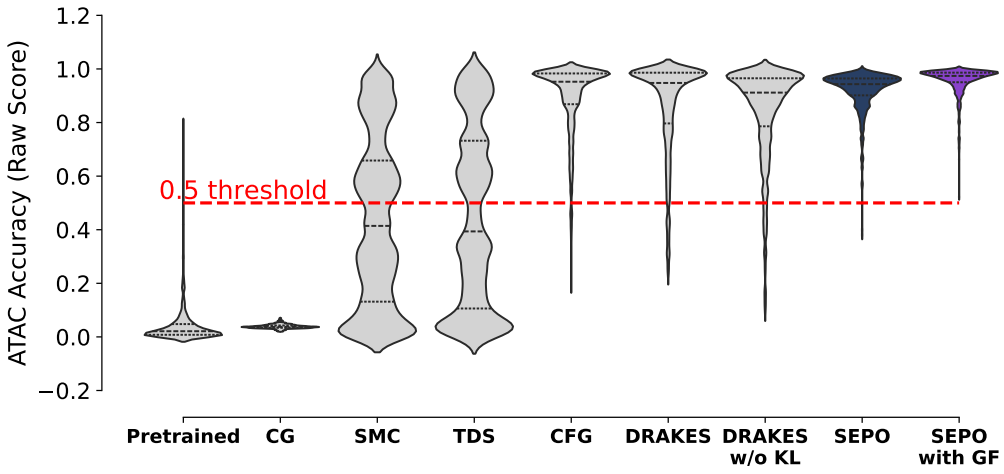

Figure 8: ATAC-Acc score distributions for sequences generated by different methods.

Evaluation protocol and ATAC-Acc metric. For evaluation, we generate 640 sequences per method (using a batch size of 64 over 10 batches) and report results averaged over 3 random seeds, including both the mean and standard deviation. The ATAC-Acc (%) metric is computed as the proportion of generated sequences with a predicted chromatin accessibility score above 0.5 out of the 640 generated sequences. In addition to the ATAC-Acc (%) values reported in Table 2, Fig. 8 illustrates the full distribution of ATAC-Acc scores across all 640 samples of one seed for each method, confirming the consistency of the results and the near-100% performance observed in Table 2.

Ablation on SEPO without inner SNIS. We study the impact of removing self-normalized importance sampling (SNIS) from our method, as described in Section 3. As shown in Table 8, SEPO without SNIS significantly underperforms across key metrics, i.e. Pred-Activity and ATAC-Acc. This degradation stems from the high variance of gradient estimates when importance weights are not used, which hinders stable policy updates and results in a less powerful fine-tuned model.

Table 8: Result for SEPO without self-normalized importance sampling.

Method	*Pred-Activity (median)* ↑	*ATAC-Acc (%)* ↑	*3-mer Corr*↑	*Log-Lik (median)* ↑
SEPO w/o SNIS	4.66 (0.05)	32.7 (1.2)	0.829 (0.0005)	−239.3 (0.2)