ITERATIVE VALUE-GUIDED TILTING FOR FINE-TUNING DISCRETE DIFFUSION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Discrete diffusion models have demonstrated considerable success across generative tasks ranging from natural language to biological sequences. However, fine-tuning these models on fine-grained black-box rewards, such as binding affinity or molecular property scores, remains challenging. We propose a novel algorithm for discrete diffusion models that requires only black-box reward evaluations and iteratively alternates between learning a soft-value function from samples of the current model and updating the generative model using the resulting value estimates. This iterative procedure progressively steers the model toward higher-reward distributions without requiring reward gradients or a pre-collected reward-labeled dataset. We validate our method on DNA sequence design and protein inverse folding tasks, outperforming comparable methods that rely only on black-box reward evaluations while remaining competitive with approaches that additionally exploit reward gradients or labeled data. Using a simple two-dimensional example, we further illustrate the benefit of successive rounds of value estimation and model optimization over a single optimization round.

1 INTRODUCTION

Discrete diffusion models have emerged as powerful generative models for a wide range of discrete data, including natural language and biological sequences (Austin et al., 2021; Campbell et al., 2022; Lou et al., 2024; Sahoo et al., 2024; Shi et al., 2024). In this work, we consider the problem of adapting such models to a specific downstream objective. More precisely, starting from a pretrained base model with associated data distribution p_0^{base}, our goal is to generate samples attaining high values of a black-box reward function r, having only access to evaluations of r.

A standard formalization of this problem is KL-regularized reward maximization. For $A > 0$, its solution is

$$p_0^A = \arg\max_p \left\{ \mathbb{E}_{X_0 \sim p}[r(X_0)] - \frac{1}{A}\mathrm{KL}\big(p \,\|\, p_0^{\text{base}}\big) \right\} \propto p_0^{\text{base}} \mathrm{e}^{Ar} \,, \tag{1}$$

where A controls the strength of the reward tilt, with $A = 0$ corresponding to the base distribution. Most existing approaches seek to adapt the pretrained model so as to approximately sample from p_0^A, relying, for instance, on inference-time reward guidance (Li et al., 2024), differentiable reward optimization, or policy-gradient methods (Wang et al., 2024; Su et al., 2025; Han et al., 2025).

A fundamental difficulty arises when high-reward configurations have very small probability under the pretrained model. In this regime, the tilted target p_0^A may be substantially different from p_0^{base}, so that samples from the base model provide little information about the regions on which the target distribution concentrates. Directly adapting the model to p_0^A can therefore become statistically and computationally challenging. This phenomenon is closely related to rare-event simulation, where one similarly seeks to characterize or sample from regions that have very small probability under a reference distribution (Bucklew, 2004; Rubino & Tuffin, 2009; Cérou & Guyader, 2007). In the simplest rare-event setting, the reference distribution plays the role of p_0^{base}, while the exponential weight e^{Ar} is replaced by an indicator of a rare event.

To mitigate this distributional mismatch, we propose an annealing strategy. Building on recent work on iterative reward tilting (Potaptchik et al., 2025; Pachebat et al., 2025; Chen et al., 2026), we connect p_0^{base} to p_0^A through a sequence of intermediate distributions with progressively increasing

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

reward tilt. Rather than attempting to reach the final target in a single adaptation step, the model is successively fine-tuned along this sequence, so that each update only needs to bridge a moderate distributional shift.

Our contributions are fourfold.

- We introduce Iterative Value-Guided Tilting (IVT), a reward-based fine-tuning framework for discrete diffusion models that requires only black-box reward evaluations and does not backpropagate through r. IVT connects the pretrained distribution p_0^{base} to the target p_0^A through intermediate distributions with progressively increasing reward tilt. At each stage, samples from the current model are evaluated by the reward oracle and used to learn a soft-value function that specifies how the denoiser should change. The resulting denoiser serves as the target for updating the generative model.
- Building on the coordinate-wise characterization of the denoising posterior established in Wan et al. (2026), we express each local reward tilt through a function whose output mirrors the structure of the denoiser's output and specifies how it should change from one phase to the next. For a sequence of length L and vocabulary size K, a single network predicts the $L \times K$ table of corrections in one forward pass, rather than through separate evaluations for each position and candidate token.
- For the exact-update version of IVT, we establish a convergence guarantee for masked discrete diffusion models, characterizing the propagation of initialization, critic and model-distillation errors across tilting steps. In particular, for any fixed number of tilting phases, the KL divergence between the exact tilted and generated distributions vanishes as the initialization and approximation errors tend to zero.
- We evaluate IVT on DNA sequence design and protein inverse folding, obtaining competitive results using only black-box reward evaluations for fine-tuning. In controlled two-dimensional experiments where high-reward regions are rarely sampled under the base distribution, we show that multiple successive tilt rounds yield a better approximation of the reward-tilted target than a single large tilt under matched compute budgets.

Notation. We consider a discrete state space of the form $\mathsf{X} = \mathsf{V}^L$, where $\mathsf{V} = \{\mathbf{e}_1, \dots, \mathbf{e}_K\}$, $\mathbf{e}_k$ is the k-th standard basis vector of $\mathbb{R}^K$, and $L \geq 1$. For text generation, K corresponds to the vocabulary size and L to the sequence length. For any positive integer n, we write $[n] = \{1, \dots, n\}$ and denote by $\boldsymbol{\Delta}_n$ the probability simplex in $\mathbb{R}^n$. Simplex over elements of V is also denoted by $\boldsymbol{\Delta}_{\mathsf{V}}$. For masked diffusion, the mask token is defined once and for all as $\mathsf{m} \coloneqq \mathbf{e}_K$, and $\bar{\mathsf{V}} \coloneqq \mathsf{V} \setminus \{\mathsf{m}\}$ denotes the clean vocabulary. In that case, for $x \in \mathsf{X}$, we write $\mathcal{M}(x) \coloneqq \{\ell \in [L] : x^\ell = \mathsf{m}\}$ for the set of its masked positions. We write $\langle \cdot, \cdot \rangle$ for the Euclidean scalar product on $\mathbb{R}^K$ and δ_x for the Dirac mass at x. Random sequences are denoted by upper-case letters and their realizations by lower-case letters. For $x \in \mathsf{X}$, we denote by x^ℓ its ℓ-th component and $x^{-\ell}$ the sequence obtained from x by removing its ℓ-th component. Finally, for $y \in \mathsf{V}$, we write $x^{\ell \to y}$ for the sequence obtained from x by replacing its ℓ-th component x^ℓ with y.

2 Preliminaries on discrete diffusion models

Discrete diffusion models generate samples by reversing a Markov noising process that progressively transforms the data distribution into a simple reference distribution (Austin et al., 2021; Campbell et al., 2022; Sun et al., 2023; Lou et al., 2024). We consider a data distribution p_0^{data} on X and review how to construct a generative distribution p_0^{base} that approximates p_0^{data} through a discrete diffusion model.

We normalize diffusion time to $[0, 1]$ and consider a continuous-time Markov chain initialized at $X_0 \sim p_0^{\mathrm{data}}$, whose coordinates evolve independently conditional on the initial state. For $0 \leq s < t \leq 1$, its transition probabilities are

$$q_{t|s}(x_t \mid x_s) = \prod_{\ell=1}^L q_{t|s}^\ell(x_t^\ell \mid x_s^\ell) \,, \quad q_{t|s}^\ell(v \mid u) = \alpha_{t|s}\, \delta_u(v) + (1 - \alpha_{t|s})\, \boldsymbol{\pi}(v) \,, \qquad u, v \in \mathsf{V} \,, \tag{2}$$

where $\boldsymbol{\pi} \in \boldsymbol{\Delta}_{\mathsf{V}}$ is a reference distribution and $\alpha_{t|s} = \alpha_t / \alpha_s$. The noise schedule α is supposed to be non-increasing on $[0, 1]$, and satisfies $\alpha_0 = 1$, $\alpha_1 = 0$. Typical choices are $\alpha_t = 1 - t$ and either the uniform reference distribution or the absorbing-mask reference distribution $\boldsymbol{\pi} = \delta_{\mathsf{m}}$. In the latter case, data sequences take values in $\bar{\mathsf{V}}^L$. The time-t marginal is $p_t^{\mathrm{data}}(x_t) = \sum_{x_0 \in \mathsf{X}} p_0^{\mathrm{data}}(x_0) \prod_{\ell=1}^L q_{t|0}^\ell(x_t^\ell \mid x_0^\ell)$, $\quad p_1^{\mathrm{data}} = \boldsymbol{\pi}^{\otimes L}$. With this notation, it can be easily shown

using Bayes' rule that the reversed process is also Markov, with transition probabilities $p^{\mathrm{data}}_{s|t}(x_s \mid x_t) = p^{\mathrm{data}}_s(x_s) q_{t|s}(x_t \mid x_s) / p^{\mathrm{data}}_t(x_t)$. Thus, exact reverse sampling transports $\boldsymbol{\pi}^{\otimes L}$ back to p^{data}_0. These transitions are generally intractable, motivating learned approximations.

To approximate the reverse dynamics, two important objects are the denoising posterior and the leave-one-out (LOO) posterior (Gourevitch et al., 2026), defined token-wise, for $t \in [0,1]$, $x_t \in \mathsf{X}$, $y \in \mathsf{V}$ and $\ell \in [L]$, respectively, by

$$\big\langle y, \mathbf{d}^{\ell}_t(x_t) \big\rangle := \mathbb{P}(X^{\ell}_0 = y \mid X_t = x_t)\,, \quad \Big\langle y, \mathbf{d}^{\mathrm{loo},\ell}_t(x_t) \Big\rangle := \mathbb{P}(X^{\ell}_0 = y \mid X^{-\ell}_t = x^{-\ell}_t)\,, \tag{3}$$

where all conditional distributions below are defined on conditioning events of positive probability. Collecting these vectors over ℓ gives maps $\mathbf{d}_t, \mathbf{d}^{\mathrm{loo}}_t : \mathsf{X} \to \boldsymbol{\Delta}^L_{\mathsf{V}}$. The argument x_t is retained in the LOO notation for uniformity, although $\mathbf{d}^{\mathrm{loo},\ell}_t(x_t)$ depends only on $x^{-\ell}_t$. These are exact quantities determined by the data distribution and the forward process, rather than learned functions. Equivalently, they are conditional expectations of the one-hot encoding of each clean token. These functions are related by simple conversion formulae; see Section A.3.

Given a learned approximation $\mathbf{d}^{\mathrm{base}}_t$ of the denoising posterior, the generative distribution is defined by reversing the forward process with transitions that average over the conditional bridges $q_{s|0,t}(x_s \mid x_0, x_t)$, for $0 \le s < t \le 1$:

$$p^{\mathrm{base}}_{s|t}(x_s \mid x_t) = \prod_{\ell=1}^{L} \sum_{y \in \mathsf{V}} \Big\langle y, \mathbf{d}^{\mathrm{base},\ell}_t(x_t) \Big\rangle\, q^{\ell}_{s|0,t}(x^{\ell}_s \mid \delta_y, x^{\ell}_t)\,. \tag{4}$$

Here for $\mu = (\mu^1, \dots, \mu^L) \in \boldsymbol{\Delta}^L_{\mathsf{V}}$, $s,t \in [0,1]$ with $s < t$, $x_s, x_t \in \mathsf{X}$, we define the extended bridge as $q_{s|0,t}(x_s \mid \mu, x_t) = \prod_{\ell=1}^{L} q^{\ell}_{s|0,t}(x^{\ell}_s \mid \mu^{\ell}, x^{\ell}_t)$, with

$$q^{\ell}_{s|0,t}(z \mid \mu^{\ell}, v) = \frac{q^{\ell}_{t|s}(v \mid z)[\alpha_s \mu^{\ell}(z) + (1-\alpha_s)\boldsymbol{\pi}(z)]}{\alpha_t \mu^{\ell}(v) + (1-\alpha_t)\boldsymbol{\pi}(v)}\,, \quad z, v \in \mathsf{V}\,. \tag{5}$$

Note that for $\mu^{\ell} = \delta_y$, for $y \in \mathsf{V}$, it is the conditional law of X^{ℓ}_s given $X^{\ell}_0 = y$ and $X^{\ell}_t = v$. In particular, using the exact posteriors yields the exact token-wise reverse marginals, although their product generally differs from the joint reverse transition $p_{s|t}$.

Given a time grid $0 = t_0 < t_1 < \cdots < t_N = 1$, generation starts from $\boldsymbol{\pi}^{\otimes L}$ and applies (4) in decreasing time order. Denoting by x_n the state at grid time t_n, the resulting output distribution is $p^{\mathrm{base}}_0(x_0) = \sum_{x_1,\dots,x_N \in \mathsf{X}} \boldsymbol{\pi}^{\otimes L}(x_N) \prod_{n=0}^{N-1} p^{\mathrm{base}}_{t_n|t_{n+1}}(x_n \mid x_{n+1})$, We refer to the intermediate marginals of this generative chain as $p^{\mathrm{base}}_{t_n}$. Note, however, that the forward marginals p^0_t need not coincide with the intermediate marginals p^{base}_t of the discretized generative chain.

In the next section, we consider a reward function $r : \mathsf{X} \to \mathbb{R}$ and the corresponding reward-tilted distribution p^A_0 defined in (1). We present then our method to sample from this distribution by iteratively tilting the base generative model p^{base}_0.

3 Iterative Value-Guided Tilting (IVT)

The reward-tilted path. Recall that our objective is to adapt the base distribution p^{base}_0 toward the exponentially reward-tilted target $p^A_0(x_0) \propto p^{\mathrm{base}}_0(x_0)\, \mathrm{e}^{Ar(x_0)}$. Rather than constructing a model targeting p^A_0 in a single step, we consider the entire family of reward-tilted distributions

$$p^a_0(x_0) \propto p^{\mathrm{base}}_0(x_0)\, \mathrm{e}^{ar(x_0)}, \quad a \in [0, A]\,. \tag{6}$$

The parameter a continuously controls the strength of the reward tilt: $p^0_0 = p^{\mathrm{base}}_0$, while p^A_0 is the desired final distribution. This type of progressive reward tilting has recently been considered for fine-tuning both continuous and discrete diffusion models (Potaptchik et al., 2025; Pachebat et al., 2025; Chen et al., 2026).

We discretize this path using a sequence of increasing tilt levels $0 = a_0 < a_1 < \cdots < a_J = A$, and write $\delta_j := a_{j+1} - a_j$ for the tilt increment at stage j. To describe a generic transition along this path,

we write a for the current tilt level and δ for the additional tilt. Consecutive distributions then satisfy $p_0^{a+\delta}(x_0) \propto p_0^a(x_0)\, \mathrm{e}^{\delta r(x_0)}$. Hence, rather than tackling the potentially difficult problem of moving directly from p_0^{base} to p_0^A, we decompose it into a sequence of smaller problems: at each step, given a diffusion model targeting p_0^a, we construct one targeting the slightly more tilted distribution $p_0^{a+\delta}$. We explain later why composing several such local updates is preferable to applying the full tilt A in a single step.

We first study this generic local transition $a \to a + \delta$, independently of the choice of tilt schedule. The associated transitions and marginals are denoted by $p^a_{s|t}$ and p^a_t, respectively. More generally, we use a superscript a to denote quantities associated with the current model.

A local reward-tilting step. Suppose that we have access to a discrete diffusion model targeting p_0^a, with an associated forward process whose transitions are $(q_{t|s})_{s<t,\, s,t\in[0,1]}$, *i.e.*, we have an approximation of the corresponding denoising posterior or LOO posterior. Our goal is to obtain the corresponding approximation for a model targeting $p_0^{a+\delta}$. To simplify the presentation, we relate here the denoising posterior associated with $p_0^{a+\delta}$ to that associated with p_0^a. The corresponding construction for consecutive LOO posteriors is given in Section B.

We begin by relating the reverse transitions of the model targeting p_0^a to those of the model targeting $p_0^{a+\delta}$. By the definition of the forward marginals at time t, for any x_t such that $p_t^a(x_t) > 0$, we have

$$p_t^{a+\delta}(x_t) \propto \mathrm{e}^{H_t^a(x_t)}\, p_t^a(x_t)\ , \ \text{where } H_t^a(x_t) = \log \textstyle\sum_{x_0 \in \mathsf{V}^L} p^a_{0|t}(x_0 \mid x_t) \mathrm{e}^{\delta r(x_0)}\ . \tag{7}$$

A direct proof is given in Section D.1. We refer to H_t^a as the *soft-value function* at time t and tilt a, and extend this definition by setting $H_0^a = \delta r$. Bayes' rule then gives

$$p^{a+\delta}_{s|t}(x_s \mid x_t) = \mathrm{e}^{H_s^a(x_s) - H_t^a(x_t)} p^a_{s|t}(x_s \mid x_t)\ . \tag{8}$$

Thus, given a diffusion model targeting p_0^a that approximates the reverse transitions $p^a_{s|t}$, transforming it into one targeting the next tilted distribution $p_0^{a+\delta}$ amounts to estimating the corresponding soft-value function. In particular, since our current model approximates $p^a_{0|t}(x_0 \mid x_t)$, knowledge of this soft-value function allows us to construct the corresponding target $p^{a+\delta}_{0|t}(x_0 \mid x_t)$ directly. However, estimating the soft-value function is challenging in high-dimensional spaces, as it requires averaging over all possible configurations of the clean sequence x_0. An alternative is to relate the denoiser associated with p_0^a directly to the one associated with $p_0^{a+\delta}$. The next proposition establishes this relationship by reformulating a result of Wan et al. (2026). Define the *conditional soft value* at time t and tilt a, denoted by h_t^a, for $\ell \in [L]$, $x_0^\ell \in \mathsf{V}$, and $x_t \in \mathsf{X}$, as

$$h_t^{a,\ell}(x_0^\ell; x_t) \coloneqq \log \frac{\displaystyle\sum_{x_0^{-\ell} \in \mathsf{V}^{L-1}} \mathrm{e}^{\delta r(x_0)} p^a_{0|t}(x_0 \mid x_t)}{\left\langle x_0^\ell, \mathbf{d}_t^{a,\ell}(x_t) \right\rangle}\ , \tag{9}$$

whenever $\left\langle x_0^\ell, \mathbf{d}_t^{a,\ell}(x_t) \right\rangle > 0$. Indeed, if $(X_t^a)_{t\in[0,1]}$ is the forward process starting from p_0^a, then $h_t^{a,\ell}(x_0^\ell; x_t) = \log \mathbb{E}\left[\mathrm{e}^{\delta r(X_0^a)} \mid X_t^a = x_t,\, X_0^{a,\ell} = x_0^\ell\right]$. When $\left\langle x_0^\ell, \mathbf{d}_t^{a,\ell}(x_t) \right\rangle = 0$, we assign $h_t^{a,\ell}(x_0^\ell; x_t)$ any finite value, e.g. zero, since the corresponding term vanishes in all identities below. Equivalently, vocabulary sums may be restricted to the support of $\mathbf{d}_t^{a,\ell}(x_t)$. For example, for masked diffusion, this occurs whenever x_t^ℓ is unmasked and $x_0^\ell \neq x_t^\ell$.

Proposition 3.1. *Let $\ell \in \{1, \dots, L\}$. Then the denoiser at tilt level $a + \delta$ satisfies the following identity for $\ell \in [L]$ and $x_t \in \mathsf{X}$:*

$$\mathbf{d}_t^{a+\delta,\ell}(x_t) = \sum_{j\in[K]} \frac{\left\langle \mathbf{e}_j, \mathbf{d}_t^{a,\ell}(x_t) \right\rangle \exp(h_t^{a,\ell}(\mathbf{e}_j; x_t))\, \mathbf{e}_j}{\displaystyle\sum_{j'\in[K]} \left\langle \mathbf{e}_{j'}, \mathbf{d}_t^{a,\ell}(x_t) \right\rangle \exp(h_t^{a,\ell}(\mathbf{e}_{j'}; x_t))}\ . \tag{10}$$

A proof is given in Section D.2.

Given an approximate conditional soft value $h_t^{\varphi_a,\ell}$ and denoiser $\mathbf{d}_t^{\theta_a,\ell}$, we define $\widetilde{\mathbf{d}}_t^{a+\delta,\ell}$ by the plug-in version of (10), replacing $h_t^{a,\ell}$ and $\mathbf{d}_t^{a,\ell}$ with their approximations $h_t^{\varphi_a,\ell}$ and $\mathbf{d}_t^{\theta_a,\ell}$. This provides a teacher target that we distill into a new model $\mathbf{d}_t^{\theta_{a+\delta}}$. We next describe this distillation step and the overall iterative tilting procedure, before discussing how the conditional soft value is learned.

Completing an iterative tilting step: distilling the teacher into the student. Recall that we assume access to an approximate model $h_t^{\varphi_a}$ for the conditional soft value h_t^a and an approximate denoiser $\mathbf{d}_t^{\theta_a}$ for $\mathbf{d}_t^a$, with respective parameters φ_a, θ_a. These approximations define the teacher target $\widetilde{\mathbf{d}}_t^{a+\delta}$ through the plug-in construction in (10). We distill this target into a new approximate denoiser $\mathbf{d}_t^{\theta_{a+\delta}}$ by minimizing the loss function $\mathrm{L}_{\mathrm{model}}(\theta_{a+\delta}) = \int_0^1 \mathbb{E}[\mathrm{KL}(\widetilde{\mathbf{d}}_t^{a+\delta}(X_t^{\theta_a}) \,||\, \mathbf{d}_t^{\theta_{a+\delta}}(X_t^{\theta_a}))]\,\mathrm{d}t$, where $(X_t^{\theta_a})_{t\in[0,1]}$ is the forward process starting from $p_0^{\theta_a}$, and the expectation is taken over the randomness of this process. In practice, we approximate this expectation by sampling a batch of rollouts from the current model θ_a and then sampling a time t uniformly at random for each rollout. The KL divergence is computed token-wise and averaged across tokens. Indeed, since both the model approximations of our denoisers are assumed to be parameterized in factorized form, the KL divergence decomposes across positions, giving

$$\boxed{\mathrm{L}_{\mathrm{model}}(\theta_{a+\delta}) = \int_0^1 \mathbb{E}_{\substack{X_0\sim p_0^{\theta_a},\\ X_t\sim q_{t|0}(\cdot|X_0)}} \left[\sum_{\ell=1}^{L} \mathrm{KL}\Big(\widetilde{\mathbf{d}}_t^{a+\delta,\ell}(X_t) \,\Big\|\, \mathbf{d}_t^{\theta_{a+\delta},\ell}(X_t)\Big)\right] \mathrm{d}t.}$$

Assuming that, at each stage j, an approximation $h_t^{\varphi_{a_j},\ell}$ of the conditional soft value is available, we can iterate this procedure to obtain a sequence of models $\mathbf{d}_t^{\theta_{a_j}}$ approximating the denoisers $\mathbf{d}_t^{a_j}$ at tilt levels a_j, for $j = 0, \ldots, J$. We call this method Iterative Value-Guided Tilting (IVT) and summarize it in Algorithm 1, as well as give a practical and more detailed version of it in Algorithm 2. The next section describes how to learn the conditional soft value $h_t^{a,\ell}$ used to construct the distillation target.

Algorithm 1 Iterative value-guided tilting

Require: Tilt levels $0 = a_0 < a_1 < \cdots < a_J = A$, initial model θ_{a_0}

1: **for** $j = 0, \ldots, J-1$ **do**
2: Set $\delta_j = a_{j+1} - a_j$
3: Sample a bank of rollouts $X_0^{(i)} \sim p_0^{\theta_{a_j}}$ and evaluate $r(X_0^{(i)})$
4: Learn $h_t^{\varphi_{a_j},\ell}$ under the current model θ_{a_j}
5: Construct $\widetilde{\mathbf{d}}_t^{a_{j+1},\ell}(X_t)$ from corrupted versions X_t of the rollout samples
6: Distill $\widetilde{\mathbf{d}}_t^{a_{j+1},\ell}$ into $\theta_{a_{j+1}}$
7: **end for**

4 Learning the conditional soft value

We now describe how to learn the conditional soft value used in the plug-in version of (10). Throughout this section, we fix a tilt level $a \in [0, A)$ and an increment $\delta > 0$ such that $a + \delta \le A$, and use the factorized forward process in (2).

Exact regression for the exponentiated conditional soft value. By definition, the exponential of the conditional soft value is a conditional mean: $\mathrm{e}^{h_t^{a,\ell}(y;x_t)} = \mathbb{E}\left[\mathrm{e}^{\delta r(X_0^a)} \mid X_t^a = x_t,\, X_0^{a,\ell} = y\right]$. Thus, squared-error regression of $\mathrm{e}^{h_t^{\varphi_a,\ell}}$ against $\mathrm{e}^{\delta r(X_0^a)}$ may be used to recover this conditional mean. Following Wan et al. (2026), this can also be achieved using the generalized KL divergence, which has the same conditional-mean minimizer. For a positive target u and prediction v, we define $\Psi(u, v) = u\log(u/v) - u + v$ and obtain the objective

$$\varphi_a \mapsto \int_0^1 \mathbb{E}\left[\sum_{\ell=1}^{L} \Psi\Big(\mathrm{e}^{\delta r(X_0^{\theta_a})}, \mathrm{e}^{h_t^{\varphi_a,\ell}(X_0^{\theta_a,\ell};X_t^{\theta_a})}\Big)\right] \mathrm{d}t\,.$$

We use the vector parameterization $h_t^{\varphi_a}(x_0; x_t) = \big(h_t^{\varphi_a,1}(x_0^1; x_t), \ldots, h_t^{\varphi_a,L}(x_0^L; x_t)\big)$. Here $(X_t^{\theta_a})_{t\in[0,1]}$ is the forward process starting from $p_0^{\theta_a}$, and the expectation is taken over the randomness of this process. This chosen parameterization learns, similarly to a denoising posterior, a

function from X to $\mathbb{R}^{L\times K}$. Omitting the terms independent of φ_a gives the practical objective

$$\mathrm{L}_{\mathrm{gKL}}(\varphi_a) = \int_0^1 \mathbb{E}\left[\sum_{\ell=1}^{L}\left\{\mathrm{e}^{h_t^{\varphi_a,\ell}(X_0^{\theta_a,\ell};X_t^{\theta_a})} - \mathrm{e}^{\delta r(X_0^{\theta_a})} h_t^{\varphi_a,\ell}(X_0^{\theta_a,\ell};X_t^{\theta_a})\right\}\right]\mathrm{d}t. \tag{11}$$

First-order regression. Recent work instead proposes learning the conditional expected reward $\mathbb{E}[r(X_0^a) \mid X_t^a]$ as the first-order counterpart of the exponential soft value (Li et al., 2024; Dou et al., 2026). We also explore this approach: in our setting, the corresponding token-wise quantity is $v_t^{a,\ell}(y;x_t) = \mathbb{E}\Big[r(X_0^a) \mid X_t^a = x_t,\, X_0^{a,\ell} = y\Big]$. where $(X_t^a)_{t\in[0,1]}$ is the forward process initialized from p_0^a. This quantity gives the first-order approximation of the conditional soft value.

Lemma 4.1. *Let $(X_t^a)_{t\in[0,1]}$ be the forward process initialized from p_0^a in (6), with transitions independent of a. For $a \in (0,A)$, $t\in[0,1]$, $\ell\in[L]$, $x_t\in\mathsf{X}$, and $y\in\mathsf{V}$, whenever $\mathbb{P}(X_t^a = x_t, X_0^{a,\ell} = y) > 0$, we have*

$$\partial_a \log\left\langle y, \mathbf{d}_t^{a,\ell}(x_t)\right\rangle = v_t^{a,\ell}(y;x_t) - \mathbb{E}[r(X_0^a) \mid X_t^a = x_t]\ . \tag{12}$$

The identity also holds at $a=0$ and $a=A$ with one-sided derivatives. Moreover, for $a<A$, as $\delta\downarrow 0$ with $a+\delta\le A$, $h_t^{a,\ell}(y;x_t) = \delta\, v_t^{a,\ell}(y;x_t) + \mathcal{O}(\delta^2)$.

A proof is given in Section D.3. The second term on the right-hand side of (12) is independent of y and cancels when the denoiser is normalized over the vocabulary. Thus, a first-order tilt step may use $\delta v_t^{a,\ell}$ in place of $h_t^{a,\ell}$. We learn this conditional reward using a regression loss:

$$\mathrm{L}_{\mathrm{FO}}(\varphi_a) = \int_0^1 \mathbb{E}\left[\sum_{\ell=1}^{L}\left(v_t^{\varphi_a,\ell}(X_0^{\theta_a,\ell};X_t^{\theta_a}) - r(X_0^{\theta_a})\right)^2\right]\mathrm{d}t$$

and then set $h_t^{\varphi_a,\ell} = \delta v_t^{\varphi_a,\ell}$.

5 Convergence along the tilt path for masked diffusion

We provide an error-propagation guarantee across the J phases of the IVT algorithm, specialized to the masked corruption process. The analysis allows an arbitrary increasing tilt grid $0 = a_0 < a_1 < \cdots < a_J = A$, $\delta_j := a_{j+1} - a_j$. For the sake of brevity, we write $\theta_{a_j} = \theta_j$ and $\varphi_{a_j} = \varphi_j$ throughout this section and the corresponding proof. The numerical discretization of the reverse-time sampler is a separate source of error and is not considered here.

Approximation assumptions

H1 (Bounded reward)**.** There exists $B_r < \infty$ such that $|r(x_0)| \le B_r$ for every $x_0 \in \bar{\mathsf{V}}^L$.

H2 (Initial accuracy)**.** Define $\eta_t := -\dot\alpha_t/(1-\alpha_t)$. There exists $\varepsilon_0 \ge 0$ such that

$$\int_0^1 \eta_t \mathbb{E}\left[\sum_{\ell\in\mathcal{M}(X_t^0)}\sum_{y\in\bar{\mathsf{V}}}\left\langle y, \mathbf{d}_t^{a_0,\ell}(X_t^{a_0})\right\rangle\left|\log\left(\frac{\left\langle y, \mathbf{d}_t^{a_0,\ell}(X_t^{a_0})\right\rangle}{\left\langle y, \mathbf{d}_t^{\theta_0,\ell}(X_t^{a_0})\right\rangle}\right)\right|\right]\mathrm{d}t \le \varepsilon_0\ .$$

H3 (Per-phase critic fit)**.** For every phase $j\in\{0,\dots,J-1\}$, there exists $\varepsilon_{C,j}\ge 0$ such that

$$\int_0^1 \eta_t \mathbb{E}\left[\sum_{\ell\in\mathcal{M}(X_t^{\theta_j})}\Psi\left(\mathrm{e}^{h_t^{\theta_j,\ell}(X_0^{\theta_j,\ell};X_t^{\theta_j})},\, \mathrm{e}^{h_t^{\varphi_j,\ell}(X_0^{\theta_j,\ell};X_t^{\theta_j})}\right)\right]\mathrm{d}t \le \varepsilon_{C,j}\ , \tag{13}$$

$$\int_0^1 \eta_t \mathbb{E}\left[\sum_{\ell\in\mathcal{M}(X_t^{\theta_j})}\left(\mathrm{e}^{h_t^{\theta_j,\ell}(X_0^{\theta_j,\ell};X_t^{\theta_j})} - \mathrm{e}^{h_t^{\varphi_j,\ell}(X_0^{\theta_j,\ell};X_t^{\theta_j})}\right)^2\right]\mathrm{d}t \le \varepsilon_{C,j}^2\ , \tag{14}$$

where $\Psi(u,v) = u\log(u/v) - u + v$ and $h_t^{\theta_j,\ell}(y;x_t) = \log\mathbb{E}\left[\mathrm{e}^{\delta_j r(X_0^{\theta_j})} \mid X_t^{\theta_j} = x_t,\, X_0^{\theta_j,\ell} = y\right]$. Both assumptions concern critic fitting at phase j. Assumption (13) corresponds to the weighted version of the entropy-based loss $\mathrm{L}_{\mathrm{gKL}}$, whereas (14) corresponds to the weighted version of the quadratic loss.

H4 (Per-phase model fit)**.** For every phase $j \in \{0, \dots, J-1\}$, there exists $\varepsilon_{M,j} \geq 0$ such that

$$\int_0^1 \eta_t \mathbb{E}\left[\sum_{\ell \in \mathcal{M}(X_t^{\theta_j})} \mathrm{KL}\left(\widetilde{\mathbf{d}}_t^{a_{j+1},\ell}(X_t^{\theta_j}) \,\middle\|\, \mathbf{d}_t^{\theta_{j+1},\ell}(X_t^{\theta_j})\right)\right] \mathrm{d}t \leq \varepsilon_{M,j} \,, \tag{15}$$

$$\int_0^1 \eta_t \mathbb{E}\left[\sum_{\ell \in \mathcal{M}(X_t^{\theta_j})} \sum_{y \in \bar{\mathsf{V}}} \left(\left\langle y, \widetilde{\mathbf{d}}_t^{a_{j+1},\ell}(X_t^{\theta_j})\right\rangle - \left\langle y, \mathbf{d}_t^{\theta_{j+1},\ell}(X_t^{\theta_j})\right\rangle\right)^2\right] \mathrm{d}t \leq \varepsilon_{M,j}^2 \,. \tag{16}$$

Assumptions (15) and (16) correspond, respectively, to the weighted versions of the KL loss $\mathrm{L}_{\mathrm{model}}$ and the quadratic loss used to distill the teacher into the student at each phase.

Phase-error accumulation Define the denoising error after phase j by

$$E_j := \int_0^1 \eta_t \mathbb{E}\left[\sum_{\ell \in \mathcal{M}(X_t^{a_j})} \sum_{y \in \bar{\mathsf{V}}} \left\langle y, \mathbf{d}_t^{a_j,\ell}(X_t^{a_j})\right\rangle \left|\log\left(\frac{\left\langle y, \mathbf{d}_t^{a_j,\ell}(X_t^{a_j})\right\rangle}{\left\langle y, \mathbf{d}_t^{\theta_j,\ell}(X_t^{a_j})\right\rangle}\right)\right|\right] \mathrm{d}t \,.$$

The convergence of the IVT algorithm for the masked corruption process is established as follows.

Theorem 5.1. *Assume **H**1 to 4 and suppose there exists constants $C_1, C_2, C_3, C_4, C_4', C_5 \in (0, \infty)$ such that*

$$\mathbb{P}(X_0^{a_j} = x_0) \leq C_1 \mathbb{P}(X_0^{\theta_j} = x_0), \qquad C_2 \mathrm{e}^{h_t^{\varphi_j,\ell}(y;x_t)} \leq \mathrm{e}^{h_t^{\theta_j,\ell}(y;x_t)} \leq C_3 \mathrm{e}^{h_t^{\varphi_j,\ell}(y;x_t)} \,,$$
$$\left\langle y, \mathbf{d}_t^{\theta_j,\ell}(x_t)\right\rangle \leq C_5 \left\langle y, \mathbf{d}_t^{a_j,\ell}(x_t)\right\rangle \,,$$
$$C_4' \left\langle y, \mathbf{d}_t^{\theta_j,\ell}(x_t)\right\rangle \leq \mathbb{P}(X_0^{\theta_j,\ell} = y | X_t^{\theta_j} = x_t) \leq C_4 \left\langle y, \mathbf{d}_t^{\theta_j,\ell}(x_t)\right\rangle$$

for every j, t, x_0, x_t, y. Then the denoising error satisfies the recursion

$$E_{j+1} \leq (\mathrm{e}^{4\delta_j B_r} + C_3' C_5' \mathrm{e}^{6\delta_j B_r}) E_j + 4\sqrt{2}(1 + C_3')\, \delta_j B_r L \mathrm{e}^{6\delta_j B_r} \sqrt{E_j} + c_j \varepsilon_{C,j} + d_j \varepsilon_{M,j} \,,$$

where $C_3' = 1/\min\{1, 1/C_3\}$, $C_5' = \max\{1, C_5\}$, $c_j := C_1 \mathrm{e}^{5\delta_j B_r}(\sqrt{L} + 1 + C_3'\sqrt{L}/C_4')$ and $d_j := \mathbf{C}_1 C_3 C_4 / C_2 \mathrm{e}^{4\delta_j B_r}(1 + \sqrt{L|\bar{\mathsf{V}}|})$. Consequently, for any fixed $J \in \mathbb{N}^*$, define

$$\varepsilon = \max\left\{\varepsilon_0 \,,\, \max_j \varepsilon_{C,j} \,,\, \max_j \varepsilon_{M,j}\right\} \,,$$

and assume $\varepsilon \in [0, 1]$. Then there exists a finite constant $\widetilde{C}_J$ depending on $C_1, C_2, C_3, C_4, C_5, C_4', |\mathsf{V}|, L, B_r, \{\delta_j\}_{j=0}^{J-1}$ such that the generated error satisfies

$$\mathrm{KL}\left(p_0^{a_J} \middle\| p_0^{\theta_J}\right) \leq \widetilde{C}_J \varepsilon^{2^{-J}} \,.$$

In particular, $\mathrm{KL}(p_0^{a_J} \| p_0^{\theta_J}) \to 0$ as $\varepsilon \to 0$.

Proof. The proof is postponed to Section D.4 □

6 Experiments

We evaluate IVT on two synthetic problems with exact targets and on the regulatory DNA and protein inverse-folding benchmarks, similarly to Wang et al. (2024).

6.1 Two-dimensional synthetic experiments

Following So et al. (2026), we use $\mathsf{X} = \{\mathsf{m}, 1, \dots, 91\}^2$, with a uniform base distribution over the 91^2 fully unmasked states. We consider rewards for which the high-reward region occupies only a small fraction of the state space, and is therefore rarely sampled under the base distribution, with either one or two separated modes; see Figure 1. The finite state space allows exact normalization of the tilted target and exact computation of the terminal total variation distance.

We compare J phases along the linear schedule $a_j = jA/J$ at fixed total compute. Across values of J, we hold fixed the numbers of terminal rollouts D, conditional-soft-value updates Q, and distillation updates U, allocating approximately D/J, Q/J, and U/J to each phase; see Section E.1.2. We test the first-order regression and both exponential tilt variants from Section 4.

Across both targets, all three estimators of the conditional soft value, and several compute budgets, Figure 2 shows that increasing the number of intermediate tilts can substantially improve over a single large update when high-reward states are rare under the base distribution, thus justifying using multiple tilt steps and not limiting ourselves to one. Beyond a certain number of phases, however, performance can deteriorate again, plausibly because the fixed total compute is spread over too many phases, leaving fewer optimization updates within each one. Additional experimental details in Appendix E.1.

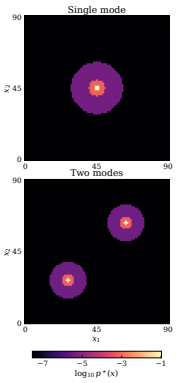


Figure 1: Tilted targets.

6.2 Regulatory DNA sequence design

We next consider regulatory DNA sequence design, following the setup considered in DRAKES (Wang et al., 2024). Fine-tuning and evaluation use separate HepG2 activity predictors: one provides the reward optimized by IVT, while the other is held out and used to report *Pred-Activity*. We use the exponential tilt variant of IVT with $J = 80$ phases of size $h = 0.25$, for a total tilt $A = 20$.

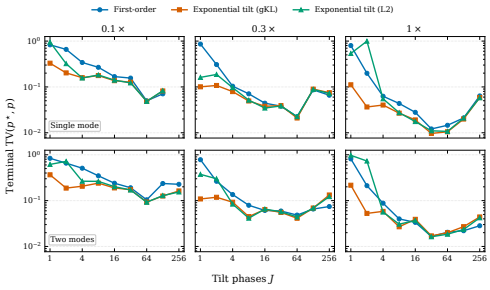


Figure 2: Terminal total variation across tilt phases under fixed compute budgets.

We group baselines by the reward information they require: *offline labels* use pre-collected reward labels or preferences, *reward gradients* differentiate through the reward, and *black-box reward* uses reward evaluations only. IVT belongs to the last regime, requiring neither offline reward labels nor reward gradients. All baseline results are reported from DRAKES, SDPO, and GILC.

Besides *Pred-Activity*, we report *ATAC-Acc.*, 3-mer Corr., JASPAR-Corr., and App-Log-Lik. to assess whether reward optimization preserves realistic properties of regulatory DNA. None of these evaluation metrics is used during fine-tuning. Full experimental and evaluation details are given in Section E.2.

	Method	HepG2 ↑	ATAC (%) ↑	3-mer ↑	JASPAR ↑	Log-lik. ↑
	Pretrained	0.17 (0.04)	1.5 (0.2)	−0.061 (0.034)	0.249 (0.015)	−261 (0.6)
L	SDPO (Han et al., 2025)	6.30 (0.003)	94.8 (0.01)	0.900 (0.003)	0.936 (0.003)	−246 (0.5)
L	D2-DPO (Borso et al., 2025)	2.97 (0.03)	35.6 (0.9)	**0.944 (0.002)**	0.883 (0.005)	−252 (0.4)
L	VRPO (Zhu et al., 2025)	4.60 (0.01)	15.8 (0.2)	0.838 (0.002)	0.865 (0.005)	−255 (0.8)
▽	DRAKES (Wang et al., 2024)	5.61 (0.07)	92.5 (0.6)	0.887 (0.002)	0.911 (0.002)	−264 (0.6)
▽	GILC-DB (Dou et al., 2026)	**7.04 (0.26)**	95.2 (2.1)	0.900 (0.044)	0.935 (0.039)	−267 (1.8)
■	SVDD (Li et al., 2024)	4.84 (0.28)	51.9 (6.3)	0.870 (0.061)	0.826 (0.057)	−269 (3.1)
■	diffu-GRPO (Zhao et al., 2025)	5.86 (0.04)	33.0 (0.8)	0.783 (0.001)	0.903 (0.004)	−245 (0.4)
■	DDPP-IS (Rector-Brooks et al., 2025)	4.07 (0.02)	50.0 (0.3)	0.711 (0.001)	0.723 (0.004)	−253 (0.9)
■	GILC-PG (Dou et al., 2026)	5.21 (0.18)	84.0 (1.5)	0.910 (0.027)	**0.937 (0.031)**	−270 (2.4)
■	**IVT** ($h = 0.25$, $A = 20$)	6.97 (0.01)	**97.0 (0.6)**	0.825 (0.002)	0.9354 (0.003)	**−239.4 (0.1)**

Table 1: Model performance on DNA sequence design. We report the mean across 3 seeds, with standard errors in parentheses. Groups denote offline labels (L), reward gradients (▽), and black-box rewards (■). Bold and underlined values denote the best and second-best results.

Results. Table 1 reports the quantitative results. Among methods using only black-box reward evaluations, IVT achieves the strongest overall performance, with the highest held-out Pred-Activity, ATAC-Acc, and Log-Lik. It is also competitive with methods that additionally rely on offline reward-labeled data or reward gradients, while requiring only online reward evaluations during fine-tuning. At fixed compute and final tilt, Figure 3 also shows that replacing a single large update by several

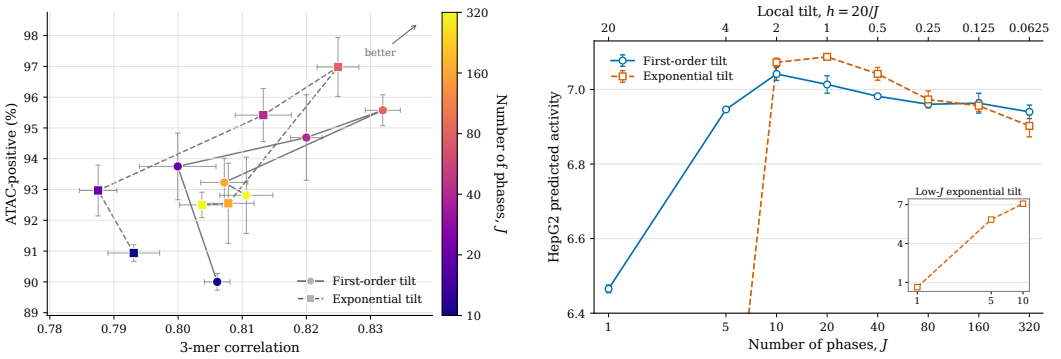


Figure 3: Effect of the number of tilting phases at fixed total tilt $A = 20$ and fixed compute budget. Left: ATAC-positive rate against 3-mer correlation, with color indicating the number of phases J. Right: held-out HepG2 predicted activity against J, with local tilt $h = 20/J$; the zoomed view shows the exponential-tilt results for small J. Error bars show standard errors across three seeds.

smaller tilts improves held-out Pred-Activity for both estimators. As in Section 6.1, increasing the number of phases eventually reveals a trade-off rather than yielding uniform gains. For $J > 10$, held-out Pred-Activity, JASPAR-Corr, and App-Log-Lik vary only mildly across configurations, so the main trade-off is between ATAC-Acc and 3-mer Corr. Among the values tested, $J = 80$ provides the best balance between these two metrics.

6.3 Protein Inverse Folding

We next consider protein inverse folding, where the goal is to generate stable amino-acid sequences compatible with a prescribed three-dimensional backbone. We follow the setup considered in DRAKES (Wang et al., 2024), using separate reward models for fine-tuning and evaluation. Starting from the pretrained conditional generator, we apply first-order IVT with $J = 64$ phases of size $h = 2$, for a total tilt $A = 128$. Implementation details are given in Section E.3.

For each of three sampling seeds, we generate 128 sequences for each of twelve held-out backbones. *Pred-ddG* is computed using the held-out evaluation oracle, while ESMFold self-consistency RMSD (*scRMSD*) checks whether the generated sequence remains compatible with the prescribed backbone (Lin et al., 2023). Our primary metric is the fraction of sequences satisfying both Pred-ddG > 0 and scRMSD < 2 Å. We additionally report the corresponding individual success rates and median values; full evaluation details are given in Section E.3.

	Method	ddG (med.) ↑	ddG > 0 (%) ↑	RMSD (med.) ↓	RMSD < 2 (%) ↑	Joint (%) ↑
	Pretrained	−0.544 (0.037)	36.6 (1.0)	0.849 (0.013)	90.9 (0.6)	34.4 (0.5)
L	SDPO (Han et al., 2025)	1.400 (0.014)	87.1 (0.01)	0.938 (0.005)	88.9 (0.3)	75.5 (0.3)
L	D2-DPO (Borso et al., 2025)	0.500 (0.051)	66.4 (0.3)	0.909 (0.005)	**93.6 (0.8)**	61.0 (0.5)
L	VRPO (Zhu et al., 2025)	0.548 (0.032)	61.1 (0.1)	0.883 (0.004)	93.5 (0.7)	56.6 (0.3)
▽	DRAKES (Wang et al., 2024)	1.095 (0.026)	86.4 (0.2)	0.918 (0.006)	91.8 (0.5)	78.6 (0.7)
▽	GILC-DB (Dou et al., 2026)	1.430 (0.073)	**97.9 (1.9)**	0.968 (0.015)	84.3 (2.0)	**82.4 (2.5)**
■	SVDD (Li et al., 2024)	0.694 (0.076)	69.3 (2.1)	0.850 (0.030)	89.7 (0.9)	65.0 (3.3)
■	diffu-GRPO (Zhao et al., 2025)	1.286 (0.021)	76.8 (0.3)	1.192 (0.005)	57.1 (0.8)	37.2 (1.4)
■	DDPP-IS (Rector-Brooks et al., 2025)	−0.130 (0.047)	46.7 (0.8)	**0.829 (0.008)**	89.3 (0.6)	43.3 (0.5)
■	GILC-PG (Dou et al., 2026)	0.719 (0.091)	75.6 (3.5)	0.914 (0.012)	92.4 (2.3)	69.8 (3.2)
■	**IVT**	**1.514 (0.026)**	87.3 (1.1)	0.952 (0.012)	87.2 (1.0)	75.5 (1.7)

Table 2: Model performance on protein inverse folding. We report the mean across 3 seeds, with standard errors in parentheses. Groups denote offline labels (L), reward gradients (▽), and black-box rewards (■). Bold and underlined values denote the best and second-best results.

Table 2 reports the quantitative results. Among methods using only black-box reward evaluations, IVT achieves the highest median Pred-ddG, fraction with positive Pred-ddG, and joint success rate. It is also competitive with methods that additionally rely on offline reward-labeled data or reward gradients, while requiring only online reward evaluations during fine-tuning.

AI USE STATEMENT

In this work, we used generative AI tools to formulate mathematical claims, assist in the writing and presentation of proofs, help develop theoretical models or conceptual frameworks, provide feedback on research methodology and experimental design, and assist with the implementation of methods. We have not used generative AI tools to provide critical ingredients for proving mathematical claims, propose or refine hypotheses, or interpret experimental results. The generation of synthetic datasets, assistance with translation, cleaning or reformatting of datasets, and qualitative or thematic data analysis are not applicable to this work.

We have reviewed all AI-assisted work. We independently double-checked mathematical claims and AI-assisted proofs, and AI-assisted code was inspected and tested. We also checked AI-assisted contributions to conceptual development and research methodology against the relevant literature to ensure appropriate attribution and identify prior work. We take responsibility for the final content of this work, including all text, claims, code, and other artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work does not involve human subjects or personally identifiable information. Our biological sequence-design experiments use existing benchmark datasets and predictive models.

REPRODUCIBILITY STATEMENT

The main text describes the method and experimental setup. The appendix provides a comprehensive description of the experimental details, including implementation choices, model architectures, hyperparameters, evaluation protocols, and computational budgets. Anonymous code for reproducing the reported results is included in the supplementary material.

REFERENCES

Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces. In *Advances in Neural Information Processing Systems*, volume 34, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/958c530554f78bcd8e97125b70e6973d-Abstract.html.

Umberto Borso, Davide Paglieri, Jude Wells, and Tim Rocktäschel. Preference-based alignment of discrete diffusion models, 2025. URL https://arxiv.org/abs/2503.08295.

James Antonio Bucklew. *Introduction to Rare Event Simulation*. Springer, New York, 2004. doi: 10.1007/978-1-4757-4078-3.

Andrew Campbell, Joe Benton, Valentin De Bortoli, Tom Rainforth, George Deligiannidis, and Arnaud Doucet. A continuous time framework for discrete denoising models. In *Advances in Neural Information Processing Systems*, volume 35, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/b5b528767aa35f5b1a60fe0aaeca0563-Abstract.html.

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

Hanqun Cao, Haosen Shi, Chenyu Wang, Sinno Jialin Pan, and Pheng-Ann Heng. GLID^{2}E: A gradient-free lightweight fine-tune approach for discrete biological sequence design. In *Advances in Neural Information Processing Systems*, 2025. URL https://papers.nips.cc/paper_files/paper/2025/hash/cc84bfabe6389d8883fc2071c848f62a-Abstract-Conference.html.

Jaime A. Castro-Mondragon, Rafael Riudavets-Puig, Ieva Rauluseviciute, Roza Berhanu Lemma, Laura Turchi, Romain Blanc-Mathieu, Jérémy Lucas, Paul Boddie, Aziz Khan, Nicolás Manosalva Pérez, Oriol Fornes, Tiffany Y. Leung, Aitor Aguirre, Fatiha Hammal, Daniel Schmelter, Danilo Baranasic, Bénice Ballester, Anneli Sandelin, Björn Lenhard, Kristoffer Vandepoele, Wyeth W. Wasserman, François Parcy, and Alexandre Mathelier. Jaspar 2022: the 9th release of the open-access database of transcription factor binding profiles. *Nucleic Acids Research*, 50 (D1):D165–D173, 2022. doi: 10.1093/nar/gkab1113. URL https://academic.oup.com/nar/article/50/D1/D165/6446529.

Frédéric Cérou and Arnaud Guyader. Adaptive multilevel splitting for rare event analysis. *Stochastic Analysis and Applications*, 25(2):417–443, 2007. doi: 10.1080/07362990601139628.

Yuyuan Chen, Shiyi Wang, Peter Potaptchik, Jaeyeon Kim, and Michael S. Albergo. Discrete tilt matching, 2026. URL https://arxiv.org/abs/2604.18739.

Wenda Chu, Zihui Wu, Yifan Chen, Yang Song, and Yisong Yue. Split gibbs discrete diffusion posterior sampling, 2025. URL https://arxiv.org/abs/2503.01161.

Giovanni Conforti, Alain Durmus, Le-Tuyet-Nhi Pham, and Gael Raoul. Non-asymptotic convergence of discrete diffusion models: Masked and random walk dynamics, 2025. URL https://arxiv.org/abs/2512.00580.

Justas Dauparas, Ivan Anishchenko, Nathaniel Bennett, Hua Bai, Robert J. Ragotte, Lukas F. Milles, Basile I. M. Wicky, Alexis Courbet, Rob J. de Haas, Neville Bethel, et al. Robust deep learning–based protein sequence design using ProteinMPNN. *Science*, 378(6615):49–56, 2022. doi: 10.1126/science.add2187.

Hongkun Dou, Zike Chen, Fengji Li, Hongjue Li, and Yue Deng. Plug-and-play guidance for discrete diffusion models via gradient-informed logit correction, 2026. URL https://arxiv.org/abs/2606.06303.

Samson Gourevitch, Yazid Janati, Dario Shariatian, Umut Simsekli, Eric Moulines, Eric P. Xing, and Alain Durmus. Uniform diffusion models revisited: Leave-one-out denoiser and absorbing state reformulation, 2026. URL https://arxiv.org/abs/2605.22765.

Jiaqi Han, Austin Wang, Minkai Xu, Wenda Chu, Meihua Dang, Haotian Ye, Huayu Chen, Yisong Yue, and Stefano Ermon. Discrete diffusion trajectory alignment via stepwise decomposition, 2025. URL https://arxiv.org/abs/2507.04832.

Jaewoo Lee, Minsu Kim, Sanghyeok Choi, Inhyuck Song, Sujin Yun, Hyeongyu Kang, Woocheol Shin, Taeyoung Yun, Kiyoung Om, and Jinkyoo Park. Diffusion alignment as variational expectation-maximization, 2025. URL https://arxiv.org/abs/2510.00502.

Xiner Li, Yulai Zhao, Chenyu Wang, Gabriele Scalia, Gokcen Eraslan, Surag Nair, Tommaso Biancalani, Shuiwang Ji, Aviv Regev, Sergey Levine, and Masatoshi Uehara. Derivative-free guidance in continuous and discrete diffusion models with soft value-based decoding, 2024. URL https://arxiv.org/abs/2408.08252.

Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, et al. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023. doi: 10.1126/science.ade2574.

Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 32819–32848. PMLR, 2024. URL https://proceedings.mlr.press/v235/lou24a.html.

Zijing Ou, Chinmay Pani, and Yingzhen Li. Inference-time scaling of discrete diffusion models via importance weighting and optimal proposal design, 2025. URL https://arxiv.org/abs/2505.22524.

Jean Pachebat, Giovanni Conforti, Alain Durmus, and Yazid Janati. Iterative tilting for diffusion fine-tuning, 2025. URL https://arxiv.org/abs/2512.03234.

Peter Potaptchik, Cheuk-Kit Lee, and Michael S. Albergo. Tilt matching for scalable sampling and fine-tuning, 2025. URL https://arxiv.org/abs/2512.21829.

Jarrid Rector-Brooks, Mohsin Hasan, Zhangzhi Peng, Zachary Quinn, Cheng-Hao Liu, Sarthak Mittal, Nouha Dziri, Michael Bronstein, Yoshua Bengio, Pranam Chatterjee, Alexander Tong, and Avishek Joey Bose. Steering masked discrete diffusion models via discrete denoising posterior prediction, 2025. URL https://arxiv.org/abs/2410.08134.

Gerardo Rubino and Bruno Tuffin (eds.). *Rare Event Simulation using Monte Carlo Methods*. John Wiley & Sons, 2009. ISBN 9780470772690. doi: 10.1002/9780470745403.

Subham Sekhar Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin T. Chiu, Alexander Rush, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/eb0b13cc515724ab8015bc978fdde0ad-Abstract-Conference.html.

Jiaxin Shi, Kehang Han, Zhe Wang, Arnaud Doucet, and Michalis K. Titsias. Simplified and generalized masked diffusion for discrete data. In *Advances in Neural Information Processing Systems*, volume 37, pp. 103131–103167, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/bad233b9849f019aead5e5cc60cef70f-Abstract-Conference.html.

Oswin So, Brian Karrer, Chuchu Fan, Ricky T. Q. Chen, and Guan-Horng Liu. Discrete adjoint matching, 2026. URL https://arxiv.org/abs/2602.07132.

Xingyu Su, Xiner Li, Masatoshi Uehara, Sunwoo Kim, Yulai Zhao, Gabriele Scalia, Ehsan Hajiramezanali, Tommaso Biancalani, Degui Zhi, and Shuiwang Ji. Iterative distillation for reward-guided fine-tuning of diffusion models in biomolecular design, 2025. URL https://arxiv.org/abs/2507.00445.

Haoran Sun, Lijun Yu, Bo Dai, Dale Schuurmans, and Hanjun Dai. Score-based continuous-time discrete diffusion models. In *International Conference on Learning Representations*, 2023.

Kotaro Tsuboyama, Justas Dauparas, Jonathan Chen, Elodie Laine, Yasser Mohseni Behbahani, Jonathan J. Weinstein, Niall M. Mangan, Sergey Ovchinnikov, and Gabriel J. Rocklin. Mega-scale experimental analysis of protein folding stability in biology and design. *Nature*, 620(7973): 434–444, 2023. doi: 10.1038/s41586-023-06328-6.

Zhengyan Wan, Yidong Ouyang, Liyan Xie, Fang Fang, Hongyuan Zha, and Guang Cheng. Discrete guidance matching: Exact guidance for discrete flow matching. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 148461–148493, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/f0470962fc538827f3130757fb707b71-Paper-Conference.pdf.

Chenyu Wang, Masatoshi Uehara, Yichun He, Amy Wang, Tommaso Biancalani, Avantika Lal, Tommi Jaakkola, Sergey Levine, Hanchen Wang, and Aviv Regev. Fine-tuning discrete diffusion models via reward optimization with applications to DNA and protein design, 2024. URL https://arxiv.org/abs/2410.13643.

Siyan Zhao, Devaansh Gupta, Qinqing Zheng, and Aditya Grover. d1: Scaling reasoning in diffusion large language models via reinforcement learning, 2025. URL https://arxiv.org/abs/2504.12216.

Fengqi Zhu, Rongzhen Wang, Shen Nie, Xiaolu Zhang, Chunwei Wu, Jun Hu, Jun Zhou, Jianfei Chen, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. LLaDA 1.5: Variance-reduced preference optimization for large language diffusion models, 2025. URL https://arxiv.org/abs/2505.19223.

APPENDIX CONTENTS

A Background and related work **14**

A.1 Comparison with the closest methods 14

A.2 Broader related work 15

A.3 Relation between the denoising posterior and the LOO 15

B Leave-one-out formulation of iterative tilting **17**

C Practical implementation of IVT for Masked Diffusion Models **20**

D Proofs **22**

D.1 Proof of Equation 7 22

D.2 Proof of Proposition 3.1 22

D.3 Proof of Lemma 4.1 23

D.4 Proof of Theorem 5.1 24

E Experimental details **34**

E.1 Two-dimensional experiments 34

E.2 Regulatory DNA experiment 38

E.3 Protein inverse-folding experiment 39

A Background and related work

A.1 Comparison with the closest methods

IVT is most closely related to methods that use conditional value estimates to steer the denoising distribution toward high-reward samples, either for guidance at inference time or as a target for distillation. These methods mainly differ in what value quantity they estimate, how this quantity is used to construct the target denoising distribution, and which distribution provides the samples used to learn each update. We compare the four closest approaches below.

We first compare IVT with Discrete Guidance Matching (DGM; Wan et al., 2026). DGM studies essentially the same coordinate-wise quantity as our conditional soft value: its guidance network predicts the $L \times K$ table of corrections needed to transform the denoising distribution toward the reward-tilted target. The main difference lies in how this correction is learned. DGM estimates it directly from the base model toward p_0^A, corresponding to a single tilt step. In contrast, IVT decomposes the full tilt into a sequence of intermediate distributions and learns a local correction from samples of the current model at each phase. The two methods also differ in how the learned correction is used. DGM keeps the guidance network separate from the generative model and applies it during sampling, whereas IVT distills the corrected denoising distribution into the model after each phase. These two constructions are therefore different both in the distribution from which the correction is learned and in whether the resulting correction is used only at inference time or incorporated through model updates. As shown in Section 6, the iterative construction can be particularly useful when high-reward samples have low probability under the pretrained distribution, where a sequence of smaller tilts provides a more gradual path toward the final target.

Discrete Tilt Matching (DTM; Chen et al., 2026) is also closely related to IVT. Like IVT, it follows a sequence of intermediate reward tilts from p_0^{base} to p_0^A rather than performing a single global update. At each stage, DTM constructs target distributions for the relevant noisy states and distills them into the generative model through a cross-entropy objective, closely paralleling the KL-based distillation step we use. The main difference lies in how these targets are constructed. IVT first learns the conditional soft value and uses it to form a normalized categorical teacher, whereas DTM directly builds stochastic reward-dependent targets whose expectation matches the desired population target, without fitting a separate value model. This difference suggests a bias–variance tradeoff between the two approaches. DTM relies on unbiased stochastic targets, whose individual realizations may have substantial variance, while IVT replaces them with predictions from a learned value model, potentially yielding smoother distillation targets at the cost of bias from value-estimation error.

Soft Value-Based Decoding (SVDD; Li et al., 2024) is also closely related to IVT through its use of soft values. SVDD-MC learns the first-order approximation of the soft value by regressing the terminal reward on noisy states, while SVDD-PM approximates it by evaluating the reward on the clean prediction of the pretrained model. The main differences are how the tilt is applied and how the resulting value is used. SVDD targets the final reward tilt directly from the pretrained model and uses the value only at inference time to select among candidate reverse transitions. In contrast, IVT decomposes the full tilt into intermediate distributions, learns a local token-wise correction under the current model at each phase, and distills it into the generator. As with DGM, SVDD therefore operates from a fixed pretrained model toward the final tilt, whereas IVT progressively incorporates successive local tilts into the model.

Finally, Value-guided Iterative Distillation for Diffusion models (VIDD; Su et al., 2025) is also closely related to IVT in that it repeatedly distills reward-tilted teacher policies into the generative model. VIDD samples one-step transitions from a periodically refreshed roll-out policy, approximates their soft values by evaluating the reward on the predicted clean sample, and updates the model through value-weighted maximum likelihood. IVT instead learns the full table of token-wise corrections from current-model samples and their forward corruptions, explicitly normalizes the resulting categorical teacher, and distills it. The iterative constructions also differ: VIDD mixes pretrained and current-model roll-ins and iteratively refreshes its roll-out policy, whereas IVT follows prescribed increments of the reward tilt and resamples from the current model at each phase. Thus, VIDD uses sampled value-weighted transitions and off-policy exploration, while IVT uses a learned dense correction along an explicit annealed tilt path.

Taken together, these comparisons show that IVT combines three existing ideas in a specific way: token-wise value learning, successive local reward tilts, and denoiser distillation.

A.2 Broader related work

More broadly, reward-guided adaptation methods for discrete diffusion models differ in the information they assume about the downstream objective. DRAKES (Wang et al., 2024) formulates adaptation as KL-regularized reward maximization and differentiates through the reward using a Gumbel–Softmax relaxation. Other methods consider gradient-free adaptation for biological sequences (Cao et al., 2025), optimization from a fixed offline preference dataset (Han et al., 2025), or variational formulations of diffusion alignment (Lee et al., 2025). These approaches therefore rely on different forms of supervision, even when evaluated on similar downstream tasks. IVT assumes only access to a black-box reward oracle: it does not require reward gradients or a pre-collected reward-labeled dataset, but instead evaluates samples from the current model at each phase and uses the resulting data to learn the next update.

Inference-time guidance provides an alternative to fine-tuning for drawing samples from a desired reward-tilted distribution, while leaving the pretrained generator unchanged. Beyond the value-based approach discussed above, SGDD (Chu et al., 2025) performs posterior sampling through a split-Gibbs procedure that alternates reward-guided and diffusion-prior updates, while Ou et al. (2025) formulate inference-time control through Sequential Monte Carlo with importance weighting and approximate optimal proposals. More recently, GILC (Dou et al., 2026) uses the pretrained denoising network as a variational proxy and directly applies reward-informed corrections to its clean-prediction logits. IVT instead uses reward information to fine-tune the generative model itself. Once fine-tuning is complete, samples can be generated directly from the adapted model without an additional guidance procedure at inference time, making subsequent sampling cheaper at the cost of an upfront adaptation stage.

The iterative tilting principle has also been developed for continuous generative models. Iterative Tilting (Pachebat et al., 2025) and Tilt Matching (Potaptchik et al., 2025) progressively transform a base distribution toward a reward-tilted target through a sequence of local updates, formulated in terms of the score or velocity field of the underlying continuous generative process. IVT follows the same general principle in discrete space, but constructs each local update by learning conditional soft values and using them to form a categorical target for distillation.

A.3 Relation between the denoising posterior and the LOO

Fix t, x_t, and ℓ. Conditional independence and Bayes' rule give

$$\left\langle y, \mathbf{d}_t^{\ell}(x_t) \right\rangle = \frac{q_{t|0}^{\ell}(x_t^{\ell} \mid y) \left\langle y, \mathbf{d}_t^{\mathrm{loo},\ell}(x_t) \right\rangle}{\alpha_t \left\langle x_t^{\ell}, \mathbf{d}_t^{\mathrm{loo},\ell}(x_t) \right\rangle + (1-\alpha_t)\boldsymbol{\pi}(x_t^{\ell})}. \tag{17}$$

If $q_{t|0}^{\ell}(x_t^{\ell} \mid y) > 0$ for every possible clean token y, this relation can be inverted:

$$\left\langle y, \mathbf{d}_t^{\mathrm{loo},\ell}(x_t) \right\rangle = \frac{\left\langle y, \mathbf{d}_t^{\ell}(x_t) \right\rangle / q_{t|0}^{\ell}(x_t^{\ell} \mid y)}{\sum_z \left\langle z, \mathbf{d}_t^{\ell}(x_t) \right\rangle / q_{t|0}^{\ell}(x_t^{\ell} \mid z)}, \tag{18}$$

where the sum is over possible clean tokens. This positivity condition holds for uniform corruption when $t > 0$ and for masked corruption at masked positions. In the latter case, $\mathbf{d}_t^{\ell}(x_t) = \mathbf{d}_t^{\mathrm{loo},\ell}(x_t)$; at an unmasked position, $\mathbf{d}_t^{\ell}(x_t) = \delta_{x_t^{\ell}}$ and the token is retained during reverse sampling. The distinction between these posteriors matters when parameterizing reverse transitions; see Gourevitch et al. (2026).

For $p_t^{\mathrm{data}}(x_t) > 0$, define the discrete (or concrete) score by $s_t^{\ell}(y; x_t) := p_t^{\mathrm{data}}(x_t^{\ell \to y})/p_t^{\mathrm{data}}(x_t)$, including $s_t^{\ell}(x_t^{\ell}; x_t) = 1$. Conditioning on $X_t^{-\ell}$ yields the LOO representation

$$s_t^{\ell}(y; x_t) = \frac{\alpha_t \left\langle y, \mathbf{d}_t^{\mathrm{loo},\ell}(x_t) \right\rangle + (1-\alpha_t)\boldsymbol{\pi}(y)}{\alpha_t \left\langle x_t^{\ell}, \mathbf{d}_t^{\mathrm{loo},\ell}(x_t) \right\rangle + (1-\alpha_t)\boldsymbol{\pi}(x_t^{\ell})}. \tag{19}$$

Under the positivity condition used in (18), substitution gives the score directly from the denoiser:

$$s_t^\ell(y; x_t) = \sum_z \left\langle z, \mathbf{d}_t^\ell(x_t) \right\rangle \frac{q_{t|0}^\ell(y \mid z)}{q_{t|0}^\ell(x_t^\ell \mid z)}. \tag{20}$$

Conversely, let $S_t^\ell(x_t) = \sum_{z \in \mathsf{V}} s_t^\ell(z; x_t)$. Since the numerator in (19) sums to one, for $0 < t < 1$ we recover

$$\begin{aligned} \left\langle y, \mathbf{d}_t^{\mathrm{loo},\ell}(x_t) \right\rangle &= \frac{1}{\alpha_t} \left[\frac{s_t^\ell(y; x_t)}{S_t^\ell(x_t)} - (1 - \alpha_t)\boldsymbol{\pi}(y) \right], \\ \left\langle y, \mathbf{d}_t^\ell(x_t) \right\rangle &= q_{t|0}^\ell(x_t^\ell \mid y) S_t^\ell(x_t) \left\langle y, \mathbf{d}_t^{\mathrm{loo},\ell}(x_t) \right\rangle. \end{aligned} \tag{21}$$

These inversion formulas apply to exact scores; arbitrary positive score predictions need not yield valid posterior probabilities. For absorbing-mask corruption, at a masked position and for $y \neq \mathsf{m}$, (20) simplifies to

$$s_t^\ell(y; x_t) = \frac{\alpha_t}{1 - \alpha_t} \left\langle y, \mathbf{d}_t^\ell(x_t) \right\rangle, \qquad x_t^\ell = \mathsf{m}. \tag{22}$$

To connect these scores to continuous-time sampling, set $\beta_t = -\alpha_t'/\alpha_t$ for $t < 1$. The forward rate from x to $x^{\ell \to y}$, for $y \neq x^\ell$, is $Q_t(x^{\ell \to y} \mid x) = \beta_t \boldsymbol{\pi}(y)$; all other off-diagonal rates vanish. The time-reversal identity gives

$$\bar{Q}_t(x_t^{\ell \to y} \mid x_t) = s_t^\ell(y; x_t) Q_t(x_t \mid x_t^{\ell \to y}) = \beta_t \boldsymbol{\pi}(x_t^\ell) s_t^\ell(y; x_t). \tag{23}$$

The diagonal entries are the negatives of the total outgoing rates, and reverse dynamics are simulated in decreasing diffusion time. Replacing the exact score by a nonnegative approximation therefore defines approximate reverse rates.

B LEAVE-ONE-OUT FORMULATION OF ITERATIVE TILTING

We now give the same local tilt construction when the reverse model is parameterized by the LOO posterior rather than by the denoising posterior. We retain the setting of Section 4: $a \in [0, A)$, $\delta > 0$, $a + \delta \le A$, and the factorized forward process in (2). At tilt level a, define the exact LOO posterior by

$$\left\langle y, \mathbf{d}_t^{a,\mathrm{loo},\ell}(x_t) \right\rangle := \mathbb{P}\Big(X_0^{a,\ell} = y \mid X_t^{a,-\ell} = x_t^{-\ell} \Big) .$$

Its argument is written as x_t for consistency with the denoiser notation, although its row at position ℓ depends only on $x_t^{-\ell}$. We define the *leave-one-out conditional soft value* by

$$h_t^{a,\mathrm{loo},\ell}(y; x_t^{-\ell}) := \log \mathbb{E}\Big[\mathrm{e}^{\delta r(X_0^a)} \mid X_t^{a,-\ell} = x_t^{-\ell}, X_0^{a,\ell} = y \Big] \ . \tag{24}$$

This definition applies whenever $\left\langle y, \mathbf{d}_t^{a,\mathrm{loo},\ell}(x_t) \right\rangle > 0$. When this probability is zero, we use any finite extension of $h_t^{a,\mathrm{loo},\ell}(y; x_t^{-\ell})$, because the corresponding term in every normalized update below has zero weight. The following lemma shows that the LOO conditional soft value is not an additional quantity to learn: under the factorized corruption used throughout the paper, it is exactly the conditional soft value introduced in (9).

Lemma B.1. *Under factorized forward corruption as in (2), including the standard masked and uniform corruption processes, the conditional soft value satisfies*

$$h_t^{a,\ell}(x_0^\ell; x_t) = h_t^{a,\mathrm{loo},\ell}(x_0^\ell; x_t^{-\ell}).$$

Proof. Under factorized forward corruption, the joint distribution of the clean and corrupted sequences is

$$\mathbb{P}(X_0^a = x_0, X_t^a = x_t) = p_0^a(x_0) \prod_{k=1}^{L} q_{t|0}^k(x_t^k \mid x_0^k).$$

Fix $X_0^{a,\ell} = y$, and let x_0 be any clean sequence such that $x_0^\ell = y$. Splitting the product at position ℓ, the joint probability appearing in Bayes' formula becomes

$$p_0^a(x_0)\, q_{t|0}^\ell(x_t^\ell \mid y) \prod_{k \neq \ell} q_{t|0}^k(x_t^k \mid x_0^k).$$

Bayes' rule therefore gives

$$\begin{aligned}
&\mathbb{P}\Big(X_0^{a,-\ell} = x_0^{-\ell} \mid X_t^a = x_t,\, X_0^{a,\ell} = y \Big) \\
&\quad = \frac{p_0^a(x_0) q_{t|0}^\ell(x_t^\ell \mid y) \prod_{k \neq \ell} q_{t|0}^k(x_t^k \mid x_0^k)}{\sum_{\tilde{x}_0^{-\ell} \in \mathsf{V}^{L-1}} p_0^a\big((y, \tilde{x}_0^{-\ell})\big) q_{t|0}^\ell(x_t^\ell \mid y) \prod_{k \neq \ell} q_{t|0}^k(x_t^k \mid \tilde{x}_0^k)} \\
&\quad = \frac{p_0^a(x_0) \prod_{k \neq \ell} q_{t|0}^k(x_t^k \mid x_0^k)}{\sum_{\tilde{x}_0^{-\ell} \in \mathsf{V}^{L-1}} p_0^a\big((y, \tilde{x}_0^{-\ell})\big) \prod_{k \neq \ell} q_{t|0}^k(x_t^k \mid \tilde{x}_0^k)} \\
&\quad = \mathbb{P}\Big(X_0^{a,-\ell} = x_0^{-\ell} \mid X_t^{a,-\ell} = x_t^{-\ell},\, X_0^{a,\ell} = y \Big) \ .
\end{aligned}$$

The second equality cancels the common factor $q_{t|0}^\ell(x_t^\ell \mid y)$, while the last equality is Bayes' formula when conditioning only on $X_t^{a,-\ell} = x_t^{-\ell}$ and $X_0^{a,\ell} = y$. The two conditional expectations of $\mathrm{e}^{\delta r(X_0^a)}$ are therefore equal. Taking logarithms proves the claim. □

The LOO posteriors at two consecutive tilt levels are consequently related by the same multiplicative correction as the denoisers.

Proposition B.2. *For $\ell \in [L]$ and every $x_t \in \mathsf{X}$ such that $\mathbb{P}(X_t^{a,-\ell} = x_t^{-\ell}) > 0$, the LOO posterior at tilt level $a+\delta$ satisfies*

$$\mathbf{d}_t^{a+\delta,\mathrm{loo},\ell}(x_t) = \sum_{k\in[K]} \frac{\left\langle \mathbf{e}_k, \mathbf{d}_t^{a,\mathrm{loo},\ell}(x_t)\right\rangle \exp\left(h_t^{a,\mathrm{loo},\ell}(\mathbf{e}_k; x_t^{-\ell})\right) \mathbf{e}_k}{\sum_{k'\in[K]} \left\langle \mathbf{e}_{k'}, \mathbf{d}_t^{a,\mathrm{loo},\ell}(x_t)\right\rangle \exp\left(h_t^{a,\mathrm{loo},\ell}(\mathbf{e}_{k'}; x_t^{-\ell})\right)} .$$

Proof. Fix t, ℓ, and $x_t^{-\ell}$ satisfying the stated positivity condition. The relation $p_0^{a+\delta}(x_0) \propto p_0^a(x_0)\mathrm{e}^{\delta r(x_0)}$ gives, for every $y \in \mathsf{V}$,

$$\begin{aligned}
&\mathbb{P}\left(X_0^{a+\delta,\ell} = y \mid X_t^{a+\delta,-\ell} = x_t^{-\ell}\right) \\
&\quad = \frac{\mathbb{P}\left(X_0^{a,\ell} = y \mid X_t^{a,-\ell} = x_t^{-\ell}\right) \mathbb{E}\left[\mathrm{e}^{\delta r(X_0^a)} \mid X_t^{a,-\ell} = x_t^{-\ell}, X_0^{a,\ell} = y\right]}{\mathbb{E}\left[\mathrm{e}^{\delta r(X_0^a)} \mid X_t^{a,-\ell} = x_t^{-\ell}\right]}.
\end{aligned}$$

The two factors in the numerator are respectively $\left\langle y, \mathbf{d}_t^{a,\mathrm{loo},\ell}(x_t)\right\rangle$ and $\exp(h_t^{a,\mathrm{loo},\ell}(y; x_t^{-\ell}))$. Applying the conditional law of total expectation to the denominator and partitioning over $X_0^{a,\ell}$ gives the normalizing sum in Proposition B.2. This proves the component-wise identity, and collecting its components gives the stated vector identity. □

Suppose now that $\mathbf{d}_t^{\theta_a,\mathrm{loo}}$ approximates the LOO posterior at the current tilt and that $h_t^{\varphi_a}$ approximates the conditional soft value. In view of Lemma B.1, the same value model defines the LOO teacher

$$\widetilde{\mathbf{d}}_t^{a+\delta,\mathrm{loo},\ell}(x_t) := \sum_{k\in[K]} \frac{\left\langle \mathbf{e}_k, \mathbf{d}_t^{\theta_a,\mathrm{loo},\ell}(x_t)\right\rangle \exp\left(h_t^{\varphi_a,\ell}(\mathbf{e}_k; x_t)\right) \mathbf{e}_k}{\sum_{k'\in[K]} \left\langle \mathbf{e}_{k'}, \mathbf{d}_t^{\theta_a,\mathrm{loo},\ell}(x_t)\right\rangle \exp\left(h_t^{\varphi_a,\ell}(\mathbf{e}_{k'}; x_t)\right)} . \tag{25}$$

Thus no conversion from an approximate LOO posterior to an approximate denoiser is needed before taking a tilt step. The target in (25) can be distilled directly into a new LOO model by minimizing

$$\mathrm{L}_{\mathrm{model}}^{\mathrm{loo}}(\theta_{a+\delta}) = \int_0^1 \mathbb{E}\left[\sum_{\ell=1}^{L} \mathrm{KL}\left(\widetilde{\mathbf{d}}_t^{a+\delta,\mathrm{loo},\ell}(X_t^{\theta_a}) \,\middle\|\, \mathbf{d}_t^{\theta_{a+\delta},\mathrm{loo},\ell}(X_t^{\theta_a})\right)\right] \mathrm{d}t ,$$

where $X_0^{\theta_a} \sim p_0^{\theta_a}$ and $X_t^{\theta_a} \sim q_{t|0}(\cdot \mid X_0^{\theta_a})$. As for the denoiser formulation, both the current model and the value model are frozen while this categorical target is distilled into $\theta_{a+\delta}$.

The regression stage is also unchanged. Indeed, Lemma B.1 gives

$$\mathrm{e}^{h_t^{a,\mathrm{loo},\ell}(y;x_t^{-\ell})} = \mathbb{E}\left[\mathrm{e}^{\delta r(X_0^a)} \mid X_t^a = x_t, X_0^{a,\ell} = y\right] .$$

Consequently, the squared and generalized-KL regression objectives in Section 4 learn the LOO conditional soft value without modification. A sampled pair $(X_0^{\theta_a}, X_t^{\theta_a})$ again supervises, at each position ℓ, the entry indexed by $X_0^{\theta_a,\ell}$, and the $L \times K$ value table can be produced in one forward evaluation.

For the first-order variant, define

$$v_t^{a,\mathrm{loo},\ell}(y; x_t^{-\ell}) := \mathbb{E}\left[r(X_0^a) \mid X_t^{a,-\ell} = x_t^{-\ell}, X_0^{a,\ell} = y\right] .$$

The same cancellation used in Lemma B.1 shows that $v_t^{a,\mathrm{loo},\ell}(y; x_t^{-\ell}) = v_t^{a,\ell}(y; x_t)$. Moreover, on the support of the current LOO row,

$$\begin{aligned}
\partial_a \log \left\langle y, \mathbf{d}_t^{a,\mathrm{loo},\ell}(x_t)\right\rangle &= v_t^{a,\mathrm{loo},\ell}(y; x_t^{-\ell}) \\
&\quad - \mathbb{E}\left[r(X_0^a) \mid X_t^{a,-\ell} = x_t^{-\ell}\right] , \\
h_t^{a,\mathrm{loo},\ell}(y; x_t^{-\ell}) &= \delta v_t^{a,\mathrm{loo},\ell}(y; x_t^{-\ell}) + \mathcal{O}(\delta^2) .
\end{aligned}$$

The second term in the logarithmic derivative is independent of y and disappears upon normalization. Hence the first-order LOO teacher is obtained from (25) by replacing $h_t^{\varphi_a,\ell}$ with $\delta v_t^{\varphi_a,\ell}$, exactly as in the denoiser formulation.

Finally, at stage j of IVT, taking $a = a_j$ and $\delta = \delta_j = a_{j+1} - a_j$ in (25) links the LOO model at tilt a_j directly to its teacher at tilt a_{j+1}. The complete cycle is therefore identical to the one in Algorithm 1: samples from the current model are rewarded, the common conditional soft value is learned, the locally tilted LOO teacher is constructed, and that teacher is distilled into the next LOO model.

C Practical implementation of IVT for Masked Diffusion Models

We give a direct implementation of IVT for absorbing-mask diffusion using the notation of Section 4. At each tilt level a_j, a bank of clean sequences is sampled from the current model $p_0^{\theta_{a_j}}$ and evaluated once by the reward oracle. The same bank is then used to learn the conditional soft value $h_t^{\varphi_{a_j},\ell}$ and to distill the resulting target $\widetilde{\mathbf{d}}_t^{a_{j+1}}$ into the next model $\mathbf{d}_t^{\theta_{a_{j+1}}}$.

Let $0 = a_0 < a_1 < \cdots < a_J = A$ be the tilt schedule and let $\delta_j := a_{j+1} - a_j$. At stage j, freeze the current model parameters θ_{a_j} and construct the phase-local rewarded bank

$$\mathcal{B}_j = \left\{ \left(X_0^{(i)}, r(X_0^{(i)}) \right) : X_0^{(i)} \sim p_0^{\theta_{a_j}} \right\}_{i=1}^{M}.$$

The oracle is evaluated only on these clean sequences and is never differentiated.

To train the conditional soft value model, sample $X_0^{\theta_{a_j}}$ and its stored reward $r(X_0^{\theta_{a_j}})$ from $\mathcal{B}_j$, draw $t \sim \mathrm{Unif}([0,1])$, and sample $X_t^{\theta_{a_j}} \sim q_{t|0}(\cdot \mid X_0^{\theta_{a_j}})$.

Only positions $\ell \in \mathcal{M}(X_t^{\theta_{a_j}})$ are used in the two objectives below. At each such position, the target is normalized over supported clean tokens; in particular, the mask token is excluded. Equivalently, the predicted update may be defined arbitrarily on a zero-probability token because that coordinate retains zero mass after reweighting.

For the exact variant, the samplewise form of $\mathrm{L}_{\mathrm{gKL}}$ from Section 4 is

$$\ell_{\mathrm{gKL}}(\varphi_{a_j}) = \frac{1}{|\mathcal{M}(X_t^{\theta_{a_j}})|} \sum_{\ell \in \mathcal{M}(X_t^{\theta_{a_j}})} \Big\{ \mathrm{e}^{h_t^{\varphi_{a_j},\ell}(X_0^{\theta_{a_j},\ell}; X_t^{\theta_{a_j},-\ell})} - \mathrm{e}^{\delta_j r(X_0^{\theta_{a_j}})} h_t^{\varphi_{a_j},\ell}(X_0^{\theta_{a_j},\ell}; X_t^{\theta_{a_j},-\ell}) \Big\}. \tag{26}$$

For the first-order variant, the samplewise form of L_{FO} is

$$\ell_{\mathrm{FO}}(\varphi_{a_j}) = \frac{1}{|\mathcal{M}(X_t^{\theta_{a_j}})|} \sum_{\ell \in \mathcal{M}(X_t^{\theta_{a_j}})} \left(v_t^{\varphi_{a_j},\ell}(X_0^{\theta_{a_j},\ell}; X_t^{\theta_{a_j},-\ell}) - r(X_0^{\theta_{a_j}}) \right)^2. \tag{27}$$

The normalization by $|\mathcal{M}(X_t^{\theta_{a_j}})|$ first averages the position-wise losses within each corrupted sequence. Consequently, every sequence has the same total weight irrespective of how many of its positions are masked; without this normalization, more strongly corrupted sequences would contribute proportionally more to the optimization. In both variants, the reward supervises only the coordinate indexed by the observed clean token $X_0^{\theta_{a_j},\ell}$. It is not used as the label of a counterfactual coordinate $y \neq X_0^{\theta_{a_j},\ell}$.

After fitting and freezing the conditional soft value model, we construct the target denoiser by plugging the approximate denoiser and conditional soft value into (10), which gives

$$\widetilde{\mathbf{d}}_t^{a_{j+1},\ell}(x_t) = \mathrm{softmax}\Big(\log \mathbf{d}_t^{\theta_{a_j},\ell}(x_t) + \big(h_t^{\varphi_{a_j},\ell}(y; x_t)\big)_{y \in \mathsf{V}} \Big), \tag{28}$$

where the logarithm and softmax are applied coordinate-wise over the vocabulary. For the first-order variant, the vector $\big(h_t^{\varphi_{a_j},\ell}(y; x_t)\big)_{y \in \mathsf{V}}$ in (28) is replaced by $\delta_j \big(v_t^{\varphi_{a_j},\ell}(y; x_t)\big)_{y \in \mathsf{V}}$. Since the softmax is invariant to a common shift of all its logits, the predicted update may be centered at each position before it is added to the current denoiser logits, for example by subtracting its average over the vocabulary. This operation leaves the target denoiser exactly unchanged.

The parameters of the next model are initialized from the current model, $\theta_{a_{j+1}} \leftarrow \theta_{a_j}$. For a newly sampled time and forward corruption of a sequence from $\mathcal{B}_j$, the concrete token-wise contribution

to $\mathrm{L}_{\mathrm{model}}$ is

$$\ell_{\mathrm{model}}(\theta_{a_{j+1}}) = \frac{1}{|\mathcal{M}(X_t^{\theta_{a_j}})|} \sum_{\ell \in \mathcal{M}(X_t^{\theta_{a_j}})} \sum_{k \in [K]} \left\langle \mathbf{e}_k, \widetilde{\mathbf{d}}_t^{a_{j+1},\ell}(X_t^{\theta_{a_j}}) \right\rangle \times \log \frac{\left\langle \mathbf{e}_k, \widetilde{\mathbf{d}}_t^{a_{j+1},\ell}(X_t^{\theta_{a_j}}) \right\rangle}{\left\langle \mathbf{e}_k, \mathbf{d}_t^{\theta_{a_{j+1}},\ell}(X_t^{\theta_{a_j}}) \right\rangle}. \tag{29}$$

Thus, at each masked position, the KL divergence is computed as the categorical KL over the K vocabulary probabilities and then averaged across masked positions. Since the target denoiser is frozen, the target-entropy term is constant with respect to $\theta_{a_{j+1}}$, and the same update may be implemented as categorical cross-entropy. The current denoiser and the conditional soft value model remain frozen during this optimization, and gradients are propagated only through $\theta_{a_{j+1}}$.

Algorithm 2 Practical Iterative Value-Guided Tilting

Input: tilt levels $0 = a_0 < a_1 < \cdots < a_J = A$; initial denoiser parameters θ_{a_0}; initial conditional soft value parameters φ_{a_0}; reward oracle r; forward kernel $q_{t|0}$; bank size M; conditional soft value and denoiser optimization budgets.

for $j = 0, \ldots, J-1$ **do**
  Set $\delta_j \leftarrow a_{j+1} - a_j$ and freeze θ_{a_j}.
  Sample $X_0^{(i)} \sim p_0^{\theta_{a_j}}$ and store $r(X_0^{(i)})$ for $i \in \{1, \ldots, M\}$.
  Learning the conditional soft value.
  At $j = 0$, initialize φ_{a_0} with a token-independent output. For $j > 0$, **warm-start** $\varphi_{a_j} \leftarrow \varphi_{a_{j-1}}$ and retain the conditional soft value optimizer state.
  for each conditional soft value update **do**
    Sample $X_0^{\theta_{a_j}}$ and its stored reward from the current bank, sample $t \sim \mathrm{Unif}([0,1])$, and draw $X_t^{\theta_{a_j}} \sim q_{t|0}(\cdot \mid X_0^{\theta_{a_j}})$.
    Update φ_{a_j} using (26) or (27).
  end for
  Distilling the target into our model.
  Freeze φ_{a_j} and initialize $\theta_{a_{j+1}} \leftarrow \theta_{a_j}$ while retaining the denoiser optimizer state.
  for each denoiser update **do**
    Sample a sequence from the current bank, sample $t \sim \mathrm{Unif}([0,1])$, and draw a fresh forward corruption $X_t^{\theta_{a_j}}$.
    Construct $\widetilde{\mathbf{d}}_t^{a_{j+1},\ell}(X_t^{\theta_{a_j}})$ using (28).
    Update only $\theta_{a_{j+1}}$ using (29).
  end for
  Retain $\theta_{a_{j+1}}$, φ_{a_j}, and both optimizer states, and discard the current bank.
end for

Output: reward-tilted masked denoiser with parameters θ_{a_J}.

The rewarded bank is phase-local and is discarded after the transition $a_j \to a_{j+1}$. For every $j > 0$, the conditional soft value parameters and their optimizer state are warm-started from the preceding stage because consecutive stages correspond to nearby tilted distributions. The token-independent initialization at the first stage leaves the current denoiser unchanged after the normalization in (28).

D PROOFS

D.1 PROOF OF EQUATION 7

Equation (7).

For every x_t such that $p_t^a(x_t) > 0$,

$$p_t^{a+\delta}(x_t) \propto \mathrm{e}^{H_t^a(x_t)} p_t^a(x_t) \,,$$
$$H_t^a(x_t) = \log \sum_{x_0 \in \mathsf{V}^L} p_{0|t}^a(x_0 \mid x_t) \mathrm{e}^{\delta r(x_0)} \,. \tag{7}$$

Back to Equation (7).

Proof of (7). Define the normalization constant

$$Z_{a,\delta} \coloneqq \sum_{x_0 \in \mathsf{V}^L} p_0^a(x_0) \mathrm{e}^{\delta r(x_0)} \,.$$

Then $p_0^{a+\delta}(x_0) = Z_{a,\delta}^{-1} p_0^a(x_0) \mathrm{e}^{\delta r(x_0)}$. For every x_t such that $p_t^a(x_t) > 0$, expanding the time-t marginal and using Bayes' rule gives

$$\begin{aligned} p_t^{a+\delta}(x_t) &= \sum_{x_0 \in \mathsf{V}^L} q_{t|0}(x_t \mid x_0) p_0^{a+\delta}(x_0) \\ &= Z_{a,\delta}^{-1} \sum_{x_0 \in \mathsf{V}^L} q_{t|0}(x_t \mid x_0) p_0^a(x_0) \mathrm{e}^{\delta r(x_0)} \\ &= Z_{a,\delta}^{-1} p_t^a(x_t) \sum_{x_0 \in \mathsf{V}^L} p_{0|t}^a(x_0 \mid x_t) \mathrm{e}^{\delta r(x_0)} \\ &= Z_{a,\delta}^{-1} \mathrm{e}^{H_t^a(x_t)} p_t^a(x_t) \,. \end{aligned}$$

Since $Z_{a,\delta}$ does not depend on x_t, this proves the claimed proportionality. □

D.2 PROOF OF PROPOSITION 3.1

Proposition 3.1.

Let $\ell \in \{1, \dots, L\}$. Then the denoiser at tilt level $a + \delta$ satisfies the following identity for $\ell \in [L]$ and $x_t \in \mathsf{X}$:

$$\mathbf{d}_t^{a+\delta,\ell}(x_t) = \sum_{j \in [K]} \frac{\left\langle \mathbf{e}_j, \mathbf{d}_t^{a,\ell}(x_t) \right\rangle \exp(h_t^{a,\ell}(\mathbf{e}_j; x_t)) \mathbf{e}_j}{\sum_{j' \in [K]} \left\langle \mathbf{e}_{j'}, \mathbf{d}_t^{a,\ell}(x_t) \right\rangle \exp(h_t^{a,\ell}(\mathbf{e}_{j'}; x_t))} \,.$$

For reference, the token-wise function used above is defined, for $x_0^\ell \in \mathsf{V}$, by

$$h_t^{a,\ell}(x_0^\ell; x_t) = \log \frac{\sum_{x_0^{-\ell} \in \mathsf{V}^{L-1}} \mathrm{e}^{\delta r(x_0)} p_{0|t}^a(x_0 \mid x_t)}{\left\langle x_0^\ell, \mathbf{d}_t^{a,\ell}(x_t) \right\rangle} \,. \tag{9}$$

Back to Proposition 3.1.

Proof. Fix t, ℓ, and x_t with $p_t^a(x_t) > 0$. Bayes' rule and the relation $p_0^{a+\delta}(x_0) = Z_{a,\delta}^{-1} p_0^a(x_0) \mathrm{e}^{\delta r(x_0)}$ give the clean posterior directly as

$$p_{0|t}^{a+\delta}(x_0 \mid x_t) = \frac{q_{t|0}(x_t \mid x_0) p_0^a(x_0) \mathrm{e}^{\delta r(x_0)}}{\sum_{\tilde{x}_0 \in \mathsf{V}^L} q_{t|0}(x_t \mid \tilde{x}_0) p_0^a(\tilde{x}_0) \mathrm{e}^{\delta r(\tilde{x}_0)}} = \frac{p_{0|t}^a(x_0 \mid x_t) \mathrm{e}^{\delta r(x_0)}}{\sum_{\tilde{x}_0 \in \mathsf{V}^L} p_{0|t}^a(\tilde{x}_0 \mid x_t) \mathrm{e}^{\delta r(\tilde{x}_0)}} .$$

The first denominator is the sum of the numerator over every possible clean sequence, and the common factor $Z_{a,\delta}^{-1}$ has canceled. In the second equality, the common factor $p_t^a(x_t)$ arising from Bayes' rule has likewise canceled.

The ℓ-t row of our denoiser is the marginal of this denoising posterior at position ℓ. Thus, for a token $y \in \mathsf{V}$, writing $x_0 = (y, x_0^{-\ell})$ for any clean sequence satisfying $x_0^\ell = y$, we obtain

$$\left\langle y, \mathbf{d}_t^{a+\delta,\ell}(x_t) \right\rangle = \sum_{x_0^{-\ell} \in \mathsf{V}^{L-1}} p_{0|t}^{a+\delta}\big((y, x_0^{-\ell}) \mid x_t\big) = \frac{\sum_{x_0^{-\ell} \in \mathsf{V}^{L-1}} p_{0|t}^a\big((y, x_0^{-\ell}) \mid x_t\big) \mathrm{e}^{\delta r((y, x_0^{-\ell}))}}{\sum_{\tilde{x}_0 \in \mathsf{V}^L} p_{0|t}^a(\tilde{x}_0 \mid x_t) \mathrm{e}^{\delta r(\tilde{x}_0)}} .$$

By the definition of $h_t^{a,\ell}$, the numerator of this ratio is $\left\langle y, \mathbf{d}_t^{a,\ell}(x_t) \right\rangle \mathrm{e}^{h_t^{a,\ell}(y; x_t)}$. Partitioning the denominator according to the value z of the clean token at position ℓ gives

$$\begin{aligned} &\sum_{\tilde{x}_0 \in \mathsf{V}^L} p_{0|t}^a(\tilde{x}_0 \mid x_t) \mathrm{e}^{\delta r(\tilde{x}_0)} \\ &\quad = \sum_{z \in \mathsf{V}} \sum_{\tilde{x}_0^{-\ell} \in \mathsf{V}^{L-1}} p_{0|t}^a\big((z, \tilde{x}_0^{-\ell}) \mid x_t\big) \mathrm{e}^{\delta r((z, \tilde{x}_0^{-\ell}))} \\ &\quad = \sum_{z \in \mathsf{V}} \left\langle z, \mathbf{d}_t^{a,\ell}(x_t) \right\rangle \mathrm{e}^{h_t^{a,\ell}(z; x_t)} . \end{aligned}$$

Substitution yields the claimed normalized update for the component indexed by y. Multiplying this component identity by the corresponding one-hot vector and summing over $y \in \mathsf{V}$ gives the vector identity in Proposition 3.1. If the current denoiser assigns zero probability to y, the corresponding marginal sum is zero at both tilt levels, and the same formula holds under the stated finite extension of $h_t^{a,\ell}$. □

D.3 Proof of Lemma 4.1

Lemma 4.1.

Let $(X_t^a)_{t \in [0,1]}$ be the forward process initialized from p_0^a in (6), with transitions independent of a. For $a \in (0, A)$, $t \in [0, 1]$, $\ell \in [L]$, $x_t \in \mathsf{X}$, and $y \in \mathsf{V}$, whenever $\mathbb{P}(X_t^a = x_t, X_0^{a,\ell} = y) > 0$, we have

$$\partial_a \log \left\langle y, \mathbf{d}_t^{a,\ell}(x_t) \right\rangle = v_t^{a,\ell}(y; x_t) - \mathbb{E}[r(X_0^a) \mid X_t^a = x_t] . \tag{30}$$

The identity also holds at $a = 0$ and $a = A$ with one-sided derivatives. Moreover, for $a < A$, as $\delta \downarrow 0$ with $a + \delta \le A$,

$$h_t^{a,\ell}(y; x_t) = \delta \, v_t^{a,\ell}(y; x_t) + \mathcal{O}(\delta^2) .$$

Back to Lemma 4.1.

Proof. Fix t, ℓ, x_t, and y satisfying the positivity condition. Set $w_t^a(x_0; x_t) := p_0^{\text{base}}(x_0)\mathrm{e}^{ar(x_0)}q_{t|0}(x_t \mid x_0)$. By Bayes' rule and (6),

$$\left\langle y, \mathbf{d}_t^{a,\ell}(x_t)\right\rangle = \frac{\sum_{x_0\in\mathsf{X}:\, x_0^\ell=y} w_t^a(x_0; x_t)}{\sum_{x_0\in\mathsf{X}} w_t^a(x_0; x_t)}.$$

Both sums are positive on the stated support, which is independent of a. Since X is finite and the forward kernel does not depend on a, we can differentiate term by term, using $\partial_a w_t^a(x_0; x_t) = r(x_0)w_t^a(x_0; x_t)$. Taking the logarithmic derivative yields

$$\begin{aligned}\partial_a \log\left\langle y, \mathbf{d}_t^{a,\ell}(x_t)\right\rangle &= \frac{\sum_{x_0\in\mathsf{X}:\, x_0^\ell=y} r(x_0)w_t^a(x_0; x_t)}{\sum_{x_0\in\mathsf{X}:\, x_0^\ell=y} w_t^a(x_0; x_t)} - \frac{\sum_{x_0\in\mathsf{X}} r(x_0)w_t^a(x_0; x_t)}{\sum_{x_0\in\mathsf{X}} w_t^a(x_0; x_t)} \\ &= \mathbb{E}\Big[r(X_0^a) \mid X_t^a = x_t, X_0^{a,\ell} = y\Big] - \mathbb{E}[r(X_0^a) \mid X_t^a = x_t]\,.\end{aligned}$$

The first expectation is $v_t^{a,\ell}(y; x_t)$ by definition, proving (12). The same calculation gives the one-sided derivatives at the endpoints.

Finally, r is bounded because X is finite. Expanding the conditional exponential moment at $\delta = 0$ gives

$$\mathbb{E}\Big[\mathrm{e}^{\delta r(X_0^a)} \mid X_t^a = x_t, X_0^{a,\ell} = y\Big] = 1 + \delta\, v_t^{a,\ell}(y; x_t) + \mathcal{O}(\delta^2).$$

Taking logarithms and using $\log(1 + u) = u + \mathcal{O}(u^2)$ proves the expansion of $h_t^{a,\ell}$. □

D.4 Proof of Theorem 5.1

Back to Theorem 5.1.

Proof of Theorem 5.1. Data processing inequality and Girsanov's theorem (Conforti et al., 2025, Theorem G.0.1) give

$$\begin{aligned}\mathrm{KL}\left(p_0^{a_J} \,\middle\|\, p_0^{\theta_J}\right) &= \mathrm{KL}\left(\mathrm{Law}(\bar{X}_1^{a_J}) \,\middle\|\, \mathrm{Law}(\bar{X}_1^{\theta_J})\right) \le \mathrm{KL}\left(\bar{\mathbb{P}}_{[0,1]}^{a_J} \,\middle\|\, \bar{\mathbb{P}}_{[0,1]}^{\theta_J}\right) \\ &= \mathrm{KL}\left(\bar{p}_0^{a_J} \,\middle\|\, \bar{p}_0^{\theta_J}\right) + \int_0^1 \mathbb{E}\Big[\sum_{y\neq \bar{X}_t^{a_J}} \Psi(\bar{Q}_t^{a_J}(y|\bar{X}_t^{a_J}), \bar{Q}_t^{\theta_J}(y|\bar{X}_t^{a_J}))\Big]\mathrm{d}t\,, \qquad (31)\end{aligned}$$

where $\Psi(u, v) = u\log(u/v) - u + v$. Here $(\bar{X}_t)_{t\in[0,1]}$, $\bar{\mathbb{P}}_{[0,1]}$, $(\bar{Q}_t)_{t\in[0,1]}$ denote the time-reversed process, its law and its generator, respectively. We define $X_{[0,1]}^{\theta_j}$ as the forward corruption initialized from $p_0^{\theta_j}$ whereas $\bar{X}_{[0,1]}^{\theta_j}$ describes the learned reversed process driven by $\mathbf{d}^{\theta_j,\ell}$. The initial error $\mathrm{KL}(\bar{p}_0^{a_J} \,\|\, \bar{p}_0^{\theta_J})$ vanishes as $\bar{p}_0^{a_j} = \bar{p}_0^{\theta_j} = \boldsymbol{\pi}^{\otimes L}$ for any phase j. It remains to control the propagation error to finish the proof. To this end, denote

$$E_j = \int_0^1 \mathbb{E}\left[\sum_{y\neq \bar{X}_t^{a_j}} \bar{Q}_t^{a_j}(y|\bar{X}_t^{a_j}) \left|\log\left(\frac{\bar{Q}_t^{a_j}(y|\bar{X}_t^{a_j})}{\bar{Q}_t^{\theta_j}(y|\bar{X}_t^{a_j}))}\right)\right|\right]\mathrm{d}t\,,$$

The definition of the masking process and the relation

$$\bar{Q}_t^a(x_t^{\ell\to y}|x_t) = \eta_{1-t}\left\langle y, \mathbf{d}_{1-t}^{a,\ell}(x_t)\right\rangle\,, \qquad \bar{Q}_t^\theta(x_t^{\ell\to y}|x_t) = \eta_{1-t}\left\langle y, \mathbf{d}_{1-t}^{\theta,\ell}(x_t)\right\rangle$$

for masked position $\ell \in \mathcal{M}(x_t)$ of x_t give

$$E_{j+1} = \int_0^1 \mathbb{E}\left[\sum_{\ell\in\mathcal{M}(\bar{X}_t^{a_{j+1}})}\sum_{y\in\bar{\mathsf{V}}} \bar{Q}_t^{a_{j+1}}\left|\log\left(\frac{\bar{Q}_t^{a_{j+1}}}{\bar{Q}_t^{\theta_{j+1}}}\right)\right|((\bar{X}_t^{a_{j+1}})^{\ell\to y}|\bar{X}_t^{a_{j+1}})\right]\mathrm{d}t$$
$$= \int_0^1 \eta_{1-t}\mathbb{E}\left[\sum_{\ell\in\mathcal{M}(X_{1-t}^{a_{j+1}})}\sum_{y\in\bar{\mathsf{V}}}\left\langle y, \mathbf{d}_{1-t}^{a_{j+1},\ell}(X_{1-t}^{a_{j+1}})\right\rangle\left|\log\left(\frac{\left\langle y, \mathbf{d}_{1-t}^{a_{j+1},\ell}(X_{1-t}^{a_{j+1}})\right\rangle}{\left\langle y, \mathbf{d}_{1-t}^{\theta_{j+1},\ell}(X_{1-t}^{a_{j+1}})\right\rangle}\right)\right|\right]\mathrm{d}t\ ,$$

where $\eta_t := -\dot{\alpha}_t/(1-\alpha_t)$. Plugging the relation between the backward generator and the denoiser into $\mathrm{KL}(\bar{\mathbb{P}}^{a_j}_{[0,1]}\|\bar{\mathbb{P}}^{\theta_j}_{[0,1]})$ gives

$$\mathrm{KL}(\bar{\mathbb{P}}^{a_j}_{[0,1]}\|\bar{\mathbb{P}}^{\theta_j}_{[0,1]})$$
$$\overset{(31)}{=} \int_0^1 \eta_{1-t}\mathbb{E}\Big[\sum_{\ell\in\mathcal{M}(X_{1-t}^{a_j})}\sum_{y\in\bar{\mathsf{V}}}\Psi\left(\left\langle y, \mathbf{d}_{1-t}^{a_j,\ell}(X_{1-t}^{a_j})\right\rangle, \left\langle y, \mathbf{d}_{1-t}^{\theta_j,\ell}(X_{1-t}^{a_j})\right\rangle\right)\Big]\mathrm{d}t \le E_j\ ,$$

since

$$\sum_{y\in\bar{\mathsf{V}}}\left\langle y, \mathbf{d}_{1-t}^{a_j,\ell}(X_{1-t}^{a_j})\right\rangle = 1 = \sum_{y\in\bar{\mathsf{V}}}\left\langle y, \mathbf{d}_{1-t}^{\theta_j,\ell}(X_{1-t}^{a_j})\right\rangle\ .$$

Recall that p^a satisfies the tilt-update

$$p_t^{a_{j+1}}(x) = \frac{\mathrm{e}^{H_t^{a_j}(x)}p_t^{a_j}(x)}{\sum_{y\in\mathsf{X}}\mathrm{e}^{H_t^{a_j}(y)}p_t^{a_j}(y)} \overset{(\mathbf{H}1)}{\le} \mathrm{e}^{2\delta_j B_r}p_t^{a_j}(x)\ , \qquad x\in\mathsf{X}.$$

where H is defined in (7) and $\delta_j = a_{j+1} - a_j$. This relation allows us to write E_{j+1} in terms of the previous tilt a_j:

$$E_{j+1} \le \mathrm{e}^{2\delta_j B_r}\int_0^1 \eta_{1-t}\mathbb{E}\left[\sum_{\ell\in\mathcal{M}(X_{1-t}^{a_j})}\sum_{y\in\bar{\mathsf{V}}}\left\langle y, \mathbf{d}_{1-t}^{a_{j+1},\ell}(X_{1-t}^{a_j})\right\rangle\left|\log\left(\frac{\left\langle y, \mathbf{d}_{1-t}^{a_{j+1},\ell}(X_{1-t}^{a_j})\right\rangle}{\left\langle y, \mathbf{d}_{1-t}^{\theta_{j+1},\ell}(X_{1-t}^{a_j})\right\rangle}\right)\right|\right]\mathrm{d}t$$
$$= \mathrm{e}^{2\delta_j B_r}\int_0^1 \eta_t\mathbb{E}\left[\sum_{\ell\in\mathcal{M}(X_t^{a_j})}\sum_{y\in\bar{\mathsf{V}}}\left\langle y, \mathbf{d}_t^{a_{j+1},\ell}(X_t^{a_j})\right\rangle\left|\log\left(\frac{\left\langle y, \mathbf{d}_t^{a_{j+1},\ell}(X_t^{a_j})\right\rangle}{\left\langle y, \mathbf{d}_t^{\theta_{j+1},\ell}(X_t^{a_j})\right\rangle}\right)\right|\right]\mathrm{d}t\ ,$$

where the last line follows the change of variables. Substitute the token-wise tilt update in (10)

$$\left\langle y, \mathbf{d}_t^{a_{j+1},\ell}(x_t)\right\rangle = \left\langle y, \mathbf{d}_t^{a_j,\ell}(x_t)\right\rangle \mathrm{e}^{h_t^{a_j,\ell}(y;x_t)}/Z_t^{a_j,\ell}(x_t)\ ,$$
$$Z_t^{a_j,\ell}(x_t) = \sum_{z\in\bar{\mathsf{V}}}\left\langle z, \mathbf{d}_t^{a_j,\ell}(x_t)\right\rangle \mathrm{e}^{h_t^{a_j,\ell}(z;x_t)}\ ,$$

and decompose

$$\left|\log\left(\frac{\left\langle y, \mathbf{d}_t^{a_{j+1},\ell}(x_t)\right\rangle}{\left\langle y, \mathbf{d}_t^{\theta_{j+1},\ell}(x_t)\right\rangle}\right)\right| = \left|\log\left(\frac{\left\langle y, \mathbf{d}_t^{a_j,\ell}(x_t)\right\rangle}{\left\langle y, \mathbf{d}_t^{\theta_j,\ell}(x_t)\right\rangle}\right) + \log\left(\frac{\mathrm{e}^{h_t^{\theta_j,\ell}(y;x_t)}}{\mathrm{e}^{h_t^{\varphi_j,\ell}(y;x_t)}}\right)\right.$$
$$\left. + \log\left(\frac{\left\langle y, \widetilde{\mathbf{d}}_t^{a_{j+1},\ell}(x_t)\right\rangle}{\left\langle y, \mathbf{d}_t^{\theta_{j+1},\ell}(x_t)\right\rangle}\right) + \log\left(\frac{\mathrm{e}^{h_t^{a_j,\ell}(y;x_t)}}{\mathrm{e}^{h_t^{\theta_j,\ell}(y;x_t)}}\right) + \log\left(\frac{Z_t^{\theta_j,\varphi_j,\ell}(x_t)}{Z_t^{a_j,\ell}(x_t)}\right)\right|,$$

where $\widetilde{\mathbf{d}}_t^{a_{j+1},\ell}(x_t)$ denotes the intermediate denoiser update after learning h^{φ_j,ℓ_t}:

$$\left\langle y, \widetilde{\mathbf{d}}_t^{a_{j+1},\ell}(x_t)\right\rangle = \left\langle y, \mathbf{d}_t^{\theta_j,\ell}(x_t)\right\rangle \mathrm{e}^{h_t^{\varphi_j,\ell}}(y;x_t)/Z_t^{\theta_j,\varphi_j,\ell}(x_t)\ ,$$
$$Z_t^{\theta_j,\varphi_j,\ell}(x_t) = \sum_{z\in\bar{\mathsf{V}}}\left\langle z, \mathbf{d}_t^{\theta_j,\ell}(x_t)\right\rangle \mathrm{e}^{h_t^{\varphi_j,\ell}}(z;x_t)\ .$$

By definition and **H**1,

$$\mathrm{e}^{-\delta_j B_r} \le \mathrm{e}^{h_t^{a_j,\ell}(y;x_t)} = \mathbb{E}\left[\mathrm{e}^{\delta_j r(X_0^{a_j})} \middle| X_0^{a_j,\ell} = y\,,\, X_t^{a_j} = x_t\right] \le \mathrm{e}^{\delta_j B_r}\,.$$

Similarly, $Z_t^{a_j,\ell} \in [\mathrm{e}^{-\delta_j B_r}, \mathrm{e}^{\delta_j B_r}]$. Combining the above estimates and applying the triangle inequality to E_{j+1}, we obtain

$$E_{j+1} \le \widetilde{E}_1 + \widetilde{E}_2 + \widetilde{E}_3 + \widetilde{E}_4 + \widetilde{E}_5\,,$$

with

$$\widetilde{E}_1 = \mathrm{e}^{4\delta_j B_r} \int_0^1 \eta_t \mathbb{E}\left[\sum_{\ell\in\mathcal{M}(X_t^{a_j})}\sum_{y\in\bar{\mathsf{V}}} \left\langle y, \mathbf{d}_t^{a_j,\ell}(X_t^{a_j})\right\rangle \left|\log\left(\frac{\left\langle y, \mathbf{d}_t^{a_j,\ell}(X_t^{a_j})\right\rangle}{\left\langle y, \mathbf{d}_t^{\theta_j,\ell}(X_t^{a_j})\right\rangle}\right)\right|\right] \mathrm{d}t\,,$$

$$\widetilde{E}_2 = \mathrm{e}^{3\delta_j B_r} \int_0^1 \eta_t \mathbb{E}\left[\sum_{\ell\in\mathcal{M}(X_t^{a_j})}\sum_{y\in\bar{\mathsf{V}}} \left\langle y, \mathbf{d}_t^{a_j,\ell}(X_t^{a_j})\right\rangle \mathrm{e}^{h_t^{a_j}(y;X_t^{a_j})} \left|\log\left(\frac{\mathrm{e}^{h_t^{\theta_j,\ell}(y;X_t^{a_j})}}{\mathrm{e}^{h_t^{\varphi_j,\ell}(y;X_t^{a_j})}}\right)\right|\right] \mathrm{d}t\,,$$

$$\widetilde{E}_3 = \mathrm{e}^{3\delta_j B_r} \int_0^1 \eta_t \mathbb{E}\left[\sum_{\ell\in\mathcal{M}(X_t^{a_j})}\sum_{y\in\bar{\mathsf{V}}} \mathrm{e}^{h_t^{a_j}(y;X_t^{a_j})} \left\langle y, \mathbf{d}_t^{a_j,\ell}(X_t^{a_j})\right\rangle \left|\log\left(\frac{\left\langle y, \widetilde{\mathbf{d}}_t^{a_{j+1},\ell}(X_t^{a_j})\right\rangle}{\left\langle y, \mathbf{d}_t^{\theta_{j+1},\ell}(X_t^{a_j})\right\rangle}\right)\right|\right] \mathrm{d}t\,,$$

$$\widetilde{E}_4 = \mathrm{e}^{3\delta_j B_r} \int_0^1 \eta_t \mathbb{E}\left[\sum_{\ell\in\mathcal{M}(X_t^{a_j})}\sum_{y\in\bar{\mathsf{V}}} \left\langle y, \mathbf{d}_t^{a_j,\ell}(X_t^{a_j})\right\rangle \mathrm{e}^{h_t^{a_j}(y;X_t^{a_j})} \left|\log\left(\frac{\mathrm{e}^{h_t^{a_j,\ell}(y;X_t^{a_j})}}{\mathrm{e}^{h_t^{\theta_j,\ell}(y;X_t^{a_j})}}\right)\right|\right] \mathrm{d}t\,,$$

$$\widetilde{E}_5 = \mathrm{e}^{3\delta_j B_r} \int_0^1 \eta_t \mathbb{E}\left[\sum_{\ell\in\mathcal{M}(X_t^{a_j})}\sum_{y\in\bar{\mathsf{V}}} \left\langle y, \mathbf{d}_t^{a_j,\ell}(X_t^{a_j})\right\rangle \mathrm{e}^{h_t^{a_j}(y;X_t^{a_j})} \left|\log\left(\frac{Z_t^{\theta_j,\varphi_j,\ell}(X_t^{a_j})}{Z_t^{a_j,\ell}(X_t^{a_j})}\right)\right|\right] \mathrm{d}t\,.$$

By definition,

$$\widetilde{E}_1 = \mathrm{e}^{4\delta_j B_r} E_j\,. \tag{32}$$

We rewrite $\widetilde{E}_2$ using the definition of $h_t^{a_j}$ and tower property:

$$\widetilde{E}_2 = \mathrm{e}^{3\delta_j B_r} \int_0^1 \eta_t \mathbb{E}\left[\sum_{\ell\in\mathcal{M}(X_t^{a_j})} \mathrm{e}^{\delta_j r(X_0^{a_j})} \left|\log\left(\frac{\mathrm{e}^{h_t^{\theta_j,\ell}(X_0^{a_j,\ell};X_t^{a_j})}}{\mathrm{e}^{h_t^{\varphi_j,\ell}(X_0^{a_j,\ell};X_t^{a_j})}}\right)\right|\right] \mathrm{d}t\,.$$

If there exists $C_1 \in (0,\infty)$ such that

$$\mathbb{P}(X_0^{a_j} = x_0) \le C_1 \mathbb{P}(X_0^{\theta_j} = x_0) \tag{33}$$

for any phase j, $x_0\mathsf{X}$, we have

$$\mathbb{P}(X_0^{a_j} = x_0, X_t^{a_j} = x_t) \le C_1 \mathbb{P}(X_0^{\theta_j} = x_0, X_t^{\theta_j} = x_t)\,,$$

for any t, x_t, since $X_{[0,1]}^{a_j}$ and $X_{[0,1]}^{\theta_j}$ have the same forward kernel. Thus, we can change the measure from $\mathbb{P}^{a_j}$ to $\mathbb{P}^{\theta_j}$ as follows:

$$\widetilde{E}_2 \le C_1 \mathrm{e}^{3\delta_j B_r} \int_0^1 \eta_t \mathbb{E}\left[\sum_{\ell\in\mathcal{M}(X_t^{\theta_j})} \mathrm{e}^{\delta_j r(X_0^{\theta_j})} \left|\log\left(\frac{\mathrm{e}^{h_t^{\theta_j,\ell}(X_0^{\theta_j,\ell};X_t^{\theta_j})}}{\mathrm{e}^{h_t^{\varphi_j,\ell}(X_0^{\theta_j,\ell};X_t^{\theta_j})}}\right)\right|\right] \mathrm{d}t\,.$$

Leveraging the tower property gives

$$\widetilde{E}_2 \le C_1 \mathrm{e}^{3\delta_j B_r} \int_0^1 \eta_t \mathbb{E}\left[\sum_{\ell\in\mathcal{M}(X_t^{\theta_j})} \mathrm{e}^{h_t^{\theta_j,\ell}(X_0^{\theta_j,\ell};X_t^{\theta_j})} \left|\log\left(\frac{\mathrm{e}^{h_t^{\theta_j,\ell}(X_0^{\theta_j,\ell};X_t^{\theta_j})}}{\mathrm{e}^{h_t^{\varphi_j,\ell}(X_0^{\theta_j,\ell};X_t^{\theta_j})}}\right)\right|\right] \mathrm{d}t$$

$$\le C_1 \mathrm{e}^{3\delta_j B_r} \int_0^1 \eta_t \mathbb{E}\left[\sum_{\ell\in\mathcal{M}(X_t^{\theta_j})} \Psi\left(\mathrm{e}^{h_t^{\theta_j,\ell}(X_0^{\theta_j,\ell};X_t^{\theta_j})}, \mathrm{e}^{h_t^{\varphi_j,\ell}(X_0^{\theta_j,\ell};X_t^{\theta_j})}\right)\right] \mathrm{d}t$$

$$+ C_1 \mathrm{e}^{3\delta_j B_r} \int_0^1 \eta_t \mathbb{E}\left[\sum_{\ell\in\mathcal{M}(X_t^{\theta_j})} \left|\mathrm{e}^{h_t^{\theta_j,\ell}(X_0^{\theta_j,\ell};X_t^{\theta_j})} - \mathrm{e}^{h_t^{\varphi_j,\ell}(X_0^{\theta_j,\ell};X_t^{\theta_j})}\right|\right] \mathrm{d}t\ .$$

The first term corresponds to the entropy-based critic loss, while the second can be related to the quadratic critic loss by applying Cauchy-Schwarz inequality:

$$\int_0^1 \eta_t \mathbb{E}\left[\sum_{\ell=1}^L \mathbb{1}_{\{X_t^{\theta_j,\ell}=\mathsf{m}\}} \left|\mathrm{e}^{h_t^{\theta_j,\ell}(X_0^{\theta_j,\ell};X_t^{\theta_j})} - \mathrm{e}^{h_t^{\varphi_j,\ell}(X_0^{\theta_j,\ell};X_t^{\theta_j})}\right|\right] \mathrm{d}t$$

$$\le \int_0^1 \eta_t \mathbb{E}\left[\sqrt{\sum_{\ell=1}^L \mathbb{1}_{\{X_t^{\theta_j,\ell}=\mathsf{m}\}}} \sqrt{\sum_{\ell=1}^L \mathbb{1}_{\{X_t^{\theta_j,\ell}=\mathsf{m}\}} \left(\mathrm{e}^{h_t^{\theta_j,\ell}(X_0^{\theta_j,\ell};X_t^{\theta_j})} - \mathrm{e}^{h_t^{\varphi_j,\ell}(X_0^{\theta_j,\ell};X_t^{\theta_j})}\right)^2}\right] \mathrm{d}t$$

$$\le \int_0^1 \eta_t \sqrt{\mathbb{E}\left[\sum_{\ell=1}^L \mathbb{1}_{\{X_t^{\theta_j,\ell}=\mathsf{m}\}}\right] \mathbb{E}\left[\sum_{\ell\in\mathcal{M}(X_t^{\theta_j})} \left(\mathrm{e}^{h_t^{\theta_j,\ell}(X_0^{\theta_j,\ell};X_t^{\theta_j})} - \mathrm{e}^{h_t^{\varphi_j,\ell}(X_0^{\theta_j,\ell};X_t^{\theta_j})}\right)^2\right]} \mathrm{d}t$$

$$\le \sqrt{\int_0^1 L\eta_t(1-\alpha_t)\mathrm{d}t} \sqrt{\int_0^1 \eta_t \mathbb{E}\left[\sum_{\ell\in\mathcal{M}(X_t^{\theta_j})} \left(\mathrm{e}^{h_t^{\theta_j,\ell}(X_0^{\theta_j,\ell};X_t^{\theta_j})} - \mathrm{e}^{h_t^{\varphi_j,\ell}(X_0^{\theta_j,\ell};X_t^{\theta_j})}\right)^2\right] \mathrm{d}t}$$

$$\overset{\mathbf{H}3}{\le} \sqrt{L}\varepsilon_{C,j}\ .$$

Here the second and third inequalities follow from the fact that

$$\mathbb{E}\left[\sum_{\ell=1}^L \mathbb{1}_{\{X_t^{\theta_j,\ell}=\mathsf{m}\}}\right] = L\mathbb{P}(X_t^{\theta_j,\ell}=\mathsf{m}) = L(1-\alpha_t)\ ,$$

and

$$\int_0^1 \eta_t(1-\alpha_t)\mathrm{d}t = \int_0^1 (-\dot{\alpha}_t)\mathrm{d}t = \alpha_0 - \alpha_1 = 1\ .$$

Therefore, **H**3 implies the upper bound on $\widetilde{E}_2$:

$$\widetilde{E}_2 \le C_1 \mathrm{e}^{3\delta_j B_r}(\sqrt{L}+1)\varepsilon_{C,j}\ . \tag{34}$$

Changing the measure from $\mathbb{P}^{a_j}$ to $\mathbb{P}^{\theta_j}$ analogously for $\widetilde{E}_3$, we obtain

$$\widetilde{E}_3 \le C_1 \mathrm{e}^{3\delta_j B_r} \int_0^1 \eta_t \mathbb{E}\left[\sum_{\ell\in\mathcal{M}(X_t^{\theta_j})} \mathrm{e}^{h_t^{\theta_j,\ell}(X_0^{\theta_j,\ell};X_t^{\theta_j})} \left|\log\left(\frac{\left\langle X_0^{\theta_j,\ell}, \widetilde{\mathbf{d}}_t^{a_{j+1},\ell}(X_t^{\theta_j})\right\rangle}{\left\langle X_0^{\theta_j,\ell}, \mathbf{d}_t^{\theta_{j+1},\ell}(X_t^{\theta_j})\right\rangle}\right)\right|\right] \mathrm{d}t\ .$$

If there exists $C_2, C_3, C_4, C_4' \in (0,\infty)$ such that

$$C_2 \mathrm{e}^{h_t^{\varphi_j,\ell}(y;x_t)} \le \mathrm{e}^{h_t^{\theta_j,\ell}(y;x_t)} \le C_3 \mathrm{e}^{h_t^{\varphi_j,\ell}(y;x_t)} \tag{35}$$

and

$$C_4'\Big\langle y, \mathbf{d}_t^{\theta_j,\ell}(x_t)\Big\rangle \le \mathbb{P}(X_0^{\theta_j,\ell} = y | X_t^{\theta_j} = x_t) \le C_4\Big\langle y, \mathbf{d}_t^{\theta_j,\ell}(x_t)\Big\rangle \tag{36}$$

for every phase j, $y \in \bar{\mathsf{V}}$, $x_t \in \mathsf{X}$, $t \in [0,1]$ and $\ell \in [L]$, $\widetilde{E}_3$ satisfies

$$\begin{aligned}
\widetilde{E}_3 &\le C_1C_3C_4\mathrm{e}^{3\delta_j B_r}\int_0^1 \eta_t \mathbb{E}\Bigg[\sum_{\ell\in\mathcal{M}(X_t^{\theta_j})}\sum_{y\in\bar{\mathsf{V}}} \mathrm{e}^{h_t^{\varphi_j,\ell}(y;X_t^{\theta_j})}\Big\langle y, \mathbf{d}_t^{\theta_j,\ell}(X_t^{\theta_j})\Big\rangle \\
&\qquad\qquad \left|\log\left(\frac{\Big\langle y, \widetilde{\mathbf{d}}_t^{a_{j+1},\ell}(X_t^{\theta_j})\Big\rangle}{\Big\langle y, \mathbf{d}_t^{\theta_{j+1},\ell}(X_t^{\theta_j})\Big\rangle}\right)\right|\Bigg]\mathrm{d}t \\
&= C_1C_3C_4\mathrm{e}^{3\delta_j B_r}\int_0^1 \eta_t \mathbb{E}\Bigg[\sum_{\ell\in\mathcal{M}(X_t^{\theta_j})}\sum_{y\in\bar{\mathsf{V}}} Z_t^{\theta_j,\varphi_j,\ell}(X_t^{\theta_j})\Big\langle y, \widetilde{\mathbf{d}}_t^{a_{j+1},\ell}(X_t^{\theta_j})\Big\rangle \\
&\qquad\qquad \left|\log\left(\frac{\Big\langle y, \widetilde{\mathbf{d}}_t^{a_{j+1},\ell}(X_t^{\theta_j})\Big\rangle}{\Big\langle y, \mathbf{d}_t^{\theta_{j+1},\ell}(X_t^{\theta_j})\Big\rangle}\right)\right|\Bigg]\mathrm{d}t\,.
\end{aligned}$$

The normalization constant $Z_t^{\theta_j,\varphi_j,\ell}$ obeys

$$\begin{aligned}
Z_t^{\theta_j,\varphi_j,\ell}(x_t) &= \sum_{z\in\bar{\mathsf{V}}} \mathrm{e}^{h_t^{\varphi_j,\ell}(z;x_t)}\Big\langle z, \mathbf{d}_t^{\theta_j,\ell}(x_t)\Big\rangle \\
&\overset{(35)}{\le} \sum_{z\in\bar{\mathsf{V}}} \mathrm{e}^{h_t^{\theta_j,\ell}(z;x_t)}\Big\langle z, \mathbf{d}_t^{\theta_j,\ell}(x_t)\Big\rangle / C_2 \le \mathrm{e}^{\delta_j B_r}/C_2\,,
\end{aligned}$$

where the last inequality follows from **H**1. Consequently,

$$\widetilde{E}_3 \le \frac{C_1C_3C_4}{C_2}\mathrm{e}^{4\delta_j B_r}\int_0^1 \eta_t\mathbb{E}\left[\sum_{\ell\in\mathcal{M}(X_t^{\theta_j})}\sum_{y\in\bar{\mathsf{V}}}\Big\langle y, \widetilde{\mathbf{d}}_t^{a_{j+1},\ell}(X_t^{\theta_j})\Big\rangle\left|\log\left(\frac{\Big\langle y, \widetilde{\mathbf{d}}_t^{a_{j+1},\ell}(X_t^{\theta_j})\Big\rangle}{\Big\langle y, \mathbf{d}_t^{\theta_{j+1},\ell}(X_t^{\theta_j})\Big\rangle}\right)\right|\right]\mathrm{d}t\,.$$

Note that $\Psi(u,v) = u\log(u/v) - u + v \ge u(1-v/u) - u + v = 0$ for any $u, v > 0$. Therefore the triangle inequality yields $|u\log(u/v)| \le \Psi(u,v) + |u-v|$, hence

$$\begin{aligned}
&\int_0^1 \eta_t\mathbb{E}\left[\sum_{\ell\in\mathcal{M}(X_t^{\theta_j})}\sum_{y\in\bar{\mathsf{V}}}\Big\langle y, \widetilde{\mathbf{d}}_t^{a_{j+1},\ell}(X_t^{\theta_j})\Big\rangle\left|\log\left(\frac{\Big\langle y, \widetilde{\mathbf{d}}_t^{a_{j+1},\ell}(X_t^{\theta_j})\Big\rangle}{\Big\langle y, \mathbf{d}_t^{\theta_{j+1},\ell}(X_t^{\theta_j})\Big\rangle}\right)\right|\right]\mathrm{d}t \\
&\le \int_0^1 \eta_t\mathbb{E}\left[\sum_{\ell\in\mathcal{M}(X_t^{\theta_j})}\sum_{y\in\bar{\mathsf{V}}}\Psi\left(\Big\langle y, \widetilde{\mathbf{d}}_t^{a_{j+1},\ell}(X_t^{\theta_j})\Big\rangle, \Big\langle y, \mathbf{d}_t^{\theta_{j+1},\ell}(X_t^{\theta_j})\Big\rangle\right)\right]\mathrm{d}t \\
&\qquad + \int_0^1 \eta_t\mathbb{E}\left[\sum_{\ell\in\mathcal{M}(X_t^{\theta_j})}\sum_{y\in\bar{\mathsf{V}}}\left|\Big\langle y, \widetilde{\mathbf{d}}_t^{a_{j+1},\ell}(X_t^{\theta_j})\Big\rangle - \Big\langle y, \mathbf{d}_t^{\theta_{j+1},\ell}(X_t^{\theta_j})\Big\rangle\right|\right]\mathrm{d}t\,.
\end{aligned}$$

The KL model loss controls the first term

$$\begin{aligned}
&\int_0^1 \eta_t\mathbb{E}\left[\sum_{\ell\in\mathcal{M}(X_t^{\theta_j})}\sum_{y\in\bar{\mathsf{V}}}\Psi\left(\Big\langle y, \widetilde{\mathbf{d}}_t^{a_{j+1},\ell}(X_t^{\theta_j})\Big\rangle, \Big\langle y, \mathbf{d}_t^{\theta_{j+1},\ell}(X_t^{\theta_j})\Big\rangle\right)\right]\mathrm{d}t \\
&= \int_0^1 \eta_t\mathbb{E}\left[\sum_{\ell\in\mathcal{M}(X_t^{\theta_j})}\mathrm{KL}\left(\widetilde{\mathbf{d}}_t^{a_{j+1},\ell}(X_t^{\theta_j})\,\middle\|\,\mathbf{d}_t^{\theta_{j+1},\ell}(X_t^{\theta_j})\right)\right]\mathrm{d}t \overset{\mathbf{H}4}{\le} \varepsilon_{M,j}\,,
\end{aligned}$$

since the last two terms cancel out thanks to

$$\sum_{y\in\bar{\mathsf{V}}}\left\langle y,\widetilde{\mathbf{d}}_t^{a_{j+1},\ell}(X_t^{\theta_j})\right\rangle = 1 = \sum_{y\in\bar{\mathsf{V}}}\left\langle y,\mathbf{d}_t^{\theta_{j+1},\ell}(X_t^{\theta_j})\right\rangle \ .$$

The second term is addressed by the Cauchy-Schwarz inequality and **H**4 similarly as before:

$$\begin{aligned}
&\int_0^1 \eta_t\mathbb{E}\left[\sum_{\ell\in\mathcal{M}(X_t^{\theta_j})}\sum_{y\in\bar{\mathsf{V}}}\left|\left\langle y,\widetilde{\mathbf{d}}_t^{a_{j+1},\ell}(X_t^{\theta_j})\right\rangle - \left\langle y,\mathbf{d}_t^{\theta_{j+1},\ell}(X_t^{\theta_j})\right\rangle\right|\right]\mathrm{d}t \\
&\leq \sqrt{L|\bar{\mathsf{V}}|}\sqrt{\int_0^1 \eta_t\mathbb{E}\left[\sum_{\ell\in\mathcal{M}(X_t^{\theta_j})}\sum_{y\in\bar{\mathsf{V}}}\left(\left\langle y,\widetilde{\mathbf{d}}_t^{a_{j+1},\ell}(X_t^{\theta_j})\right\rangle - \left\langle y,\mathbf{d}_t^{\theta_{j+1},\ell}(X_t^{\theta_j})\right\rangle\right)^2\right]\mathrm{d}t} \\
&\leq \sqrt{L|\bar{\mathsf{V}}|}\varepsilon_{M,j} \ .
\end{aligned}$$

Therefore,

$$\widetilde{E}_3 \leq \mathbf{C}\mathrm{e}^{4\delta_j B_r}(1+\sqrt{L|\bar{\mathsf{V}}|})\varepsilon_{M,j} \ , \tag{37}$$

where $\mathbf{C} = C_1C_3C_4/C_2$. Regarding $\widetilde{E}_4$, using the inequality

$$\left|\log\frac{u}{v}\right| \leq \frac{|u-v|}{\min(u,v)}$$

and the fact that both $\mathrm{e}^{h_t^a(x_0^\ell;x_t)}$ and $\mathrm{e}^{h_t^\theta(x_0^\ell;x_t)}$ admit the uniform lower bound $\mathrm{e}^{-\delta_j B_r}$, we obtain that

$$\begin{aligned}
\widetilde{E}_4 &= \mathrm{e}^{3\delta_j B_r}\int_0^1 \eta_t\mathbb{E}\left[\sum_{\ell\in\mathcal{M}(X_t^{a_j})}\mathrm{e}^{h_t^{a_j}(X_0^{a_j,\ell};X_t^{a_j})}\left|\log\left(\frac{\mathrm{e}^{h_t^{a_j,\ell}(X_0^{a_j,\ell};X_t^{a_j})}}{\mathrm{e}^{h_t^{\theta_j,\ell}(X_0^{a_j,\ell};X_t^{a_j})}}\right)\right|\right]\mathrm{d}t \\
&\leq \mathrm{e}^{5\delta_j B_r}\int_0^1 \eta_t\mathbb{E}\left[\sum_{\ell\in\mathcal{M}(X_t^{a_j})}\left|\mathrm{e}^{h_t^{a_j,\ell}(X_0^{a_j,\ell};X_t^{a_j})} - \mathrm{e}^{h_t^{\theta_j,\ell}(X_0^{a_j,\ell};X_t^{a_j})}\right|\right]\mathrm{d}t \ .
\end{aligned}$$

Plugging the definition of the conditional soft value gives

$$\begin{aligned}
\widetilde{E}_4 &\leq \mathrm{e}^{5\delta_j B_r}\int_0^1 \eta_t\mathbb{E}_{(x_0^\ell,x_t)\sim\mathbb{P}^{a_j}} \\
&\left[\sum_{\ell\in\mathcal{M}(x_t)}\left|\mathbb{E}\left[\mathrm{e}^{\delta_j r(X_0^{a_j})}\middle|X_0^{a_j,\ell}=x_0^\ell\,,X_t^{a_j}=x_t\right] - \mathbb{E}\left[\mathrm{e}^{\delta_j r(X_0^{\theta_j})}\middle|X_0^{\theta_j,\ell}=x_0^\ell\,,X_t^{\theta_j}=x_t\right]\right|\right]\mathrm{d}t \ .
\end{aligned}$$

By **H**1, $\mathrm{e}^{\delta_j r(x_0)} \in [\mathrm{e}^{-\delta_j B_r}\,,\,\mathrm{e}^{\delta_j B_r}]$ for any $x_0\in\bar{\mathsf{V}}^L$, thus

$$\begin{aligned}
&\left|\mathbb{E}\left[\mathrm{e}^{\delta_j r(X_0^{a_j})}\middle|X_0^{a_j,\ell}=x_0^\ell\,,X_t^{a_j}=x_t\right] - \mathbb{E}\left[\mathrm{e}^{\delta_j r(X_0^{\theta_j})}\middle|X_0^{\theta_j,\ell}=x_0^\ell\,,X_t^{\theta_j}=x_t\right]\right| \\
&= \left|\sum_{x_0^{-\ell}\in\bar{\mathsf{V}}^{L-1}}(\mathrm{e}^{\delta_j r(x_0)} - \mathrm{e}^{-\delta_j B_r})\left(\mathbb{P}_{0|t}^{a_j,\ell}(x_0|x_0^\ell\,,x_t) - \mathbb{P}_{0|t}^{\theta_j,\ell}(x_0|x_0^\ell\,,x_t)\right)\right| \\
&\leq 2(\mathrm{e}^{\delta_j B_r} - \mathrm{e}^{-\delta_j B_r})\left\|\mathbb{P}_{0|t}^{a_j,\ell}(\cdot|x_0^\ell\,,x_t) - \mathbb{P}_{0|t}^{\theta_j,\ell}(\cdot|x_0^\ell\,,x_t)\right\|_{\mathrm{TV}} \ ,
\end{aligned}$$

where we denote

$$\begin{aligned}
\mathbb{P}_{0|t}^{a_j,\ell}(x_0|x_0^\ell,x_t) &= \mathbb{P}\left(X_0^{a_j}=x_0\middle|X_0^{a_j,\ell}=x_0^\ell\,,\,X_t^{a_j}=x_t\right) \\
\mathbb{P}_{0|t}^{\theta_j,\ell}(x_0|x_0^\ell,x_t) &= \mathbb{P}\left(X_0^{\theta_j}=x_0\middle|X_0^{\theta_j,\ell}=x_0^\ell\,,\,X_t^{\theta_j}=x_t\right) \ .
\end{aligned}$$

The preceding upper bound together with the inequality $\mathrm{e}^x \geq x+1$ yield

$$\tilde{E}_4 \leq 4\delta_j B_r \mathrm{e}^{6\delta_j B_r} \int_0^1 \eta_t \mathbb{E}_{(x_0^\ell, x_t)\sim \mathbb{P}^{a_j}} \left[\sum_{\ell \in \mathcal{M}(x_t)} \left\| \mathbb{P}_{0|t}^{a_j,\ell}(\cdot|x_0^\ell, x_t) - \mathbb{P}_{0|t}^{\theta_j,\ell}(\cdot|x_0^\ell, x_t) \right\|_{\mathrm{TV}} \right] \mathrm{d}t \ .$$

Applying Pinsker's and Cauchy–Schwarz inequalities and the chain rule for KL together, we obtain $\widetilde{E}_4$

$$\begin{aligned}
&\leq 4\delta_j B_r \mathrm{e}^{6\delta_j B_r} \int_0^1 \eta_t \sum_{\ell=1}^L \mathbb{E}_{(x_0^\ell, x_t)\sim \mathbb{P}^{a_j}} \left[\mathbb{1}_{\{x_t^\ell = \mathsf{m}\}} \sqrt{1/2 \mathrm{KL}\left(\mathbb{P}_{0|t}^{a_j,\ell}(\cdot|x_0^\ell, x_t) \middle\| \mathbb{P}_{0|t}^{\theta_j,\ell}(\cdot|x_0^\ell, x_t) \right)} \right] \mathrm{d}t \\
&\leq 2\sqrt{2}\delta_j B_r \mathrm{e}^{6\delta_j B_r} \int_0^1 \eta_t \sum_{\ell=1}^L \sqrt{(1-\alpha_t) \mathbb{E}_{(x_0^\ell, x_t)\sim \mathbb{P}^{a_j}} \left[\mathrm{KL}\left(\mathbb{P}_{0|t}^{a_j,\ell}(\cdot|x_0^\ell, x_t) \middle\| \mathbb{P}_{0|t}^{\theta_j,\ell}(\cdot|x_0^\ell, x_t) \right) \right]} \mathrm{d}t \\
&\leq 2\sqrt{2}\delta_j B_r \mathrm{e}^{6\delta_j B_r} \int_0^1 \eta_t \sum_{\ell=1}^L \sqrt{(1-\alpha_t) \mathrm{KL}\left(\mathbb{P}_{0,t}^{a_j} \middle\| \mathbb{P}_{0,t}^{\theta_j} \right)} \mathrm{d}t \ .
\end{aligned}$$

Note that $\mathrm{KL}(\mathbb{P}_{0,t}^{a_j} \| \mathbb{P}_{0,t}^{\theta_j}) = \mathrm{KL}(p_0^{a_j} \| p_0^{\theta_j})$ as the two processes have the same forward kernel. Thus, the data processing inequality implies

$$\mathrm{KL}\left(\mathbb{P}_{0,t}^{a_j} \middle\| \mathbb{P}_{0,t}^{\theta_j} \right) = \mathrm{KL}(p_0^{a_j} \| p_0^{\theta_j}) \leq \mathrm{KL}\left(\bar{\mathbb{P}}_{[0,1]}^{a_j} \middle\| \bar{\mathbb{P}}_{[0,1]}^{\theta_j} \right) \leq E_j \ ,$$

for any $t \in [0,1]$. Therefore

$$\begin{aligned}
\widetilde{E}_4 &\leq 2\sqrt{2}\, \delta_j B_r L \mathrm{e}^{6\delta_j B_r} \sqrt{E_j} \int_0^1 (-\dot{\alpha}_t / \sqrt{1-\alpha_t}) \mathrm{d}t \\
&\leq 2\sqrt{2}\, \delta_j B_r L \mathrm{e}^{6\delta_j B_r} \sqrt{E_j}\, (2\sqrt{1-\alpha_1} - 2\sqrt{1-\alpha_0}) \\
&= 4\sqrt{2}\, \delta_j B_r L \mathrm{e}^{6\delta_j B_r} \sqrt{E_j} \ . \qquad (38)
\end{aligned}$$

It remains to control $\widetilde{E}_5$ to close the recursion. **H**1 implies

$$\begin{aligned}
\widetilde{E}_5 &\leq \mathrm{e}^{4\delta_j B_r} \int_0^1 \eta_t \mathbb{E}\left[\sum_{\ell \in \mathcal{M}(X_t^{a_j})} \underbrace{\sum_{y \in \bar{\mathsf{V}}} \left\langle y, \mathbf{d}_t^{a_j,\ell}(X_t^{a_j}) \right\rangle}_{=1} \left| \log\left(\frac{Z_t^{\theta_j,\varphi_j,\ell}(X_t^{a_j})}{Z_t^{a_j,\ell}(X_t^{a_j})} \right) \right| \right] \mathrm{d}t \\
&\leq \mathrm{e}^{4\delta_j B_r} \int_0^1 \eta_t \mathbb{E}\left[\sum_{\ell \in \mathcal{M}(X_t^{a_j})} \left| \log\left(\frac{Z_t^{\theta_j,\varphi_j,\ell}(X_t^{a_j})}{Z_t^{a_j,\ell}(X_t^{a_j})} \right) \right| \right] \mathrm{d}t \ .
\end{aligned}$$

Using the inequality $|\log(u/v)| \leq |u-v| / \min(u,v)$ together with the triangle inequality, we get

$$\begin{aligned}
\left| \log\left(\frac{Z_t^{\theta_j,\varphi_j,\ell}(x_t)}{Z_t^{a_j,\ell}(x_t)} \right) \right| &\leq \frac{\left| Z_t^{a_j,\ell}(x_t) - Z_t^{\theta_j,\varphi_j,\ell}(x_t) \right|}{\mathrm{e}^{-\delta_j B_r} \min\{1, 1/C_3\}} \\
&\leq C_3' \mathrm{e}^{\delta_j B_r} \left(\left| Z_t^{a_j,\ell}(x_t) - Z_t^{\theta_j,\ell}(x_t) \right| + \left| Z_t^{\theta_j,\ell}(x_t) - Z_t^{\theta_j,\varphi_j,\ell}(x_t) \right| \right) \ ,
\end{aligned}$$

where $C_3' \coloneqq 1/\min\{1, 1/C_3\}$ and $Z_t^{\theta_j,\ell}(x_t) \coloneqq \sum_{z \in \bar{\mathsf{V}}} \left\langle z, \mathbf{d}_t^{\theta_j,\ell}(x_t) \right\rangle \mathrm{e}^{h_t^{\theta_j,\ell}(z;x_t)}$. Therefore, the quantity $\widetilde{E}_5$ is controlled by

$$\widetilde{E}_5 \leq \widetilde{E}_{5.1} + \widetilde{E}_{5.2} \ ,$$

where

$$\begin{aligned}
\widetilde{E}_{5.1} &= C_3' \mathrm{e}^{5\delta_j B_r} \int_0^1 \eta_t \mathbb{E}\left[\sum_{\ell \in \mathcal{M}(X_t^{a_j})} \left| Z_t^{a_j,\ell}(X_t^{a_j}) - Z_t^{\theta_j,\ell}(X_t^{a_j}) \right| \right] \mathrm{d}t \ , \\
\widetilde{E}_{5.2} &= C_3' \mathrm{e}^{5\delta_j B_r} \int_0^1 \eta_t \mathbb{E}\left[\sum_{\ell \in \mathcal{M}(X_t^{a_j})} \left| Z_t^{\theta_j,\ell}(X_t^{a_j}) - Z_t^{\theta_j,\varphi_j,\ell}(X_t^{a_j}) \right| \right] \mathrm{d}t \ .
\end{aligned}$$

By definition of the normalization constant and the tower property, we deduce that

$$
\begin{aligned}
\widetilde{E}_{5.1} &= C_3' \, \mathrm{e}^{5\delta_j B_r} \\
&\quad \int_0^1 \eta_t \mathbb{E}_{x_t \sim p_t^{a_j}} \Bigg[\sum_{\ell \in \mathcal{M}(x_t)} \Bigg| \sum_{y \in \bar{\mathsf{V}}} \Big\langle y, \mathbf{d}_t^{a_j,\ell}(x_t) \Big\rangle \, \mathbb{E}\left[\mathrm{e}^{\delta_j r(X_0^{a_j})} \middle| X_t^{a_j} = x_t, X_0^{a_j,\ell} = y \right] \\
&\qquad\qquad - \sum_{y \in \bar{\mathsf{V}}} \Big\langle y, \mathbf{d}_t^{\theta_j,\ell}(x_t) \Big\rangle \, \mathbb{E}\left[\mathrm{e}^{\delta_j r(X_0^{\theta_j})} \middle| X_t^{\theta_j} = x_t, X_0^{\theta_j,\ell} = y \right] \Bigg| \Bigg] \mathrm{d}t \\
&\leq C_3' \, \mathrm{e}^{5\delta_j B_r} \\
&\quad \int_0^1 \eta_t \mathbb{E}_{x_t \sim p_t^{a_j}} \Bigg[\sum_{\ell \in \mathcal{M}(x_t)} \Bigg| \sum_{y \in \bar{\mathsf{V}}} \Big\langle y, \mathbf{d}_t^{a_j,\ell}(x_t) \Big\rangle \, \mathbb{E}\left[\mathrm{e}^{\delta_j r(X_0^{a_j})} \middle| X_t^{a_j} = x_t, X_0^{a_j,\ell} = y \right] \\
&\qquad\qquad - \sum_{y \in \bar{\mathsf{V}}} \Big\langle y, \mathbf{d}_t^{a_j,\ell}(x_t) \Big\rangle \, \mathbb{E}\left[\mathrm{e}^{\delta_j r(X_0^{\theta_j})} \middle| X_t^{\theta_j} = x_t, X_0^{\theta_j,\ell} = y \right] \Bigg| \Bigg] \mathrm{d}t \\
&\quad + C_3' \, \mathrm{e}^{5\delta_j B_r} \\
&\quad \int_0^1 \eta_t \mathbb{E}_{x_t \sim p_t^{a_j}} \Bigg[\sum_{\ell \in \mathcal{M}(x_t)} \Bigg| \sum_{y \in \bar{\mathsf{V}}} \Big(\Big\langle y, \mathbf{d}_t^{a_j,\ell}(x_t) \Big\rangle - \Big\langle y, \mathbf{d}_t^{\theta_j,\ell}(x_t) \Big\rangle \Big) \\
&\qquad\qquad \mathbb{E}\left[\mathrm{e}^{\delta_j r(X_0^{\theta_j})} \middle| X_t^{\theta_j} = x_t, X_0^{\theta_j,\ell} = y \right] \Bigg| \Bigg] \mathrm{d}t \\
&\leq C_3' \, \mathrm{e}^{5\delta_j B_r} \int_0^1 \eta_t \mathbb{E}_{(x_t, x_0^\ell) \sim \mathbb{P}^{a_j}} \Bigg[\sum_{\ell \in \mathcal{M}(x_t)} \Bigg| \mathbb{E}\left[\mathrm{e}^{\delta_j r(X_0^{a_j})} \middle| X_t^{a_j} = x_t, X_0^{a_j,\ell} = x_0^\ell \right] \\
&\qquad\qquad - \mathbb{E}\left[\mathrm{e}^{\delta_j r(X_0^{\theta_j})} \middle| X_t^{\theta_j} = x_t, X_0^{\theta_j,\ell} = x_0^\ell \right] \Bigg| \Bigg] \mathrm{d}t \\
&\quad + C_3' \, \mathrm{e}^{6\delta_j B_r} \int_0^1 \eta_t \mathbb{E}_{x_t \sim p_t^{a_j}} \Bigg[\sum_{\ell \in \mathcal{M}(x_t)} \sum_{y \in \bar{\mathsf{V}}} \Big| \Big\langle y, \mathbf{d}_t^{a_j,\ell}(x_t) \Big\rangle - \Big\langle y, \mathbf{d}_t^{\theta_j,\ell}(x_t) \Big\rangle \Big| \Bigg] \mathrm{d}t \, .
\end{aligned}
$$

Arguing analogously to the treatment of $\widetilde{E}_4$ for the first term and using the inequality $|u - v| \leq \max\{u, v\} |\log(u/v)|$ we obtain

$$
\begin{aligned}
&\widetilde{E}_{5.1} \leq 4\sqrt{2} \, C_3' \, \delta_j B_r L \mathrm{e}^{6\delta_j B_r} \sqrt{E_j} + C_3' \, \mathrm{e}^{6\delta_j B_r} \\
&\int_0^1 \eta_t \mathbb{E}_{x_t \sim p_t^{a_j}} \Bigg[\sum_{\ell \in \mathcal{M}(x_t)} \sum_{y \in \bar{\mathsf{V}}} \max\left\{ \Big\langle y, \mathbf{d}_t^{a_j,\ell}(x_t) \Big\rangle \, , \Big\langle y, \mathbf{d}_t^{\theta_j,\ell}(x_t) \Big\rangle \right\} \left| \log\left(\frac{\left\langle y, \mathbf{d}_t^{a_j,\ell}(x_t) \right\rangle}{\left\langle y, \mathbf{d}_t^{\theta_j,\ell}(x_t) \right\rangle} \right) \right| \Bigg] \mathrm{d}t \, .
\end{aligned}
$$

If there exists $C_5 \in (0, \infty)$ such that

$$
\Big\langle y, \mathbf{d}_t^{\theta_j,\ell}(x_t) \Big\rangle \leq C_5 \Big\langle y, \mathbf{d}_t^{a_j,\ell}(x_t) \Big\rangle
$$

, for $C_5' = \max\{1, C_5\}$, we get

$$
\begin{aligned}
&\widetilde{E}_{5.1} \leq 4\sqrt{2} \, C_3' \, \delta_j B_r L \mathrm{e}^{6\delta_j B_r} \sqrt{E_j} \\
&\qquad + C_3' C_5' \, \mathrm{e}^{6\delta_j B_r} \int_0^1 \eta_t \mathbb{E}\Bigg[\sum_{\ell \in \mathcal{M}(X_t^{a_j})} \sum_{y \in \bar{\mathsf{V}}} \Big\langle y, \mathbf{d}_t^{a_j,\ell}(X_t^{a_j}) \Big\rangle \left| \log\left(\frac{\left\langle y, \mathbf{d}_t^{a_j,\ell}(X_t^{a_j}) \right\rangle}{\left\langle y, \mathbf{d}_t^{\theta_j,\ell}(X_t^{a_j}) \right\rangle} \right) \right| \Bigg] \mathrm{d}t \\
&\quad = 4\sqrt{2} \, C_3' \, \delta_j B_r L \mathrm{e}^{6\delta_j B_r} \sqrt{E_j} + C_3' C_5' \, \mathrm{e}^{6\delta_j B_r} E_j \, .
\end{aligned}
$$

The last term $\widetilde{E}_{5.2}$ is handled by (33), (36), the tower property and Cauchy-Schwarz inequality:

$$\widetilde{E}_{5.2} = C_3' \, \mathrm{e}^{5\delta_j B_r} \int_0^1 \eta_t \mathbb{E}_{x_t \sim p_t^{a_j}} \left[\sum_{\ell \in \mathcal{M}(x_t)} \left| \sum_{y \in \bar{\mathsf{V}}} \left\langle y, \mathbf{d}_t^{\theta_j,\ell}(x_t) \right\rangle \left(\mathrm{e}^{h_t^{\theta_j,\ell}(y;x_t)} - \mathrm{e}^{h_t^{\varphi_j,\ell}(y;x_t)} \right) \right| \right] \mathrm{d}t$$

$$\leq \frac{C_1 C_3'}{C_4'} \, \mathrm{e}^{5\delta_j B_r} \int_0^1 \eta_t \mathbb{E} \left[\sum_{\ell \in \mathcal{M}(X_t^{\theta_j})} \mathbb{E} \left[\left| \mathrm{e}^{h_t^{\theta_j,\ell}(X_0^{\theta_j,\ell};X_t^{\theta_j})} - \mathrm{e}^{h_t^{\varphi_j,\ell}(X_0^{\theta_j,\ell};X_t^{\theta_j})} \right| \middle| X_t^{\theta_j} = x_t \right] \right] \mathrm{d}t$$

$$\leq \frac{C_1 C_3'}{C_4'} \, \mathrm{e}^{5\delta_j B_r} \int_0^1 \eta_t \sum_{\ell=1}^{L} \mathbb{E} \left[\mathbb{1}_{\{X_t^{\theta_j,\ell} = \mathsf{m}\}} \left| \mathrm{e}^{h_t^{\theta_j,\ell}(X_0^{\theta_j,\ell};X_t^{\theta_j})} - \mathrm{e}^{h_t^{\varphi_j,\ell}(X_0^{\theta_j,\ell};X_t^{\theta_j})} \right| \right] \mathrm{d}t$$

$$\leq \frac{C_1 C_3'}{C_4'} \, \mathrm{e}^{5\delta_j B_r}$$

$$\int_0^1 \eta_t \sum_{\ell=1}^{L} \sqrt{\mathbb{E} \left[\mathbb{1}_{\{X_t^{\theta_j,\ell} = \mathsf{m}\}} \right] \mathbb{E} \left[\mathbb{1}_{\{X_t^{\theta_j,\ell} = \mathsf{m}\}} \left(\mathrm{e}^{h_t^{\theta_j,\ell}(X_0^{\theta_j,\ell};X_t^{\theta_j})} - \mathrm{e}^{h_t^{\varphi_j,\ell}(X_0^{\theta_j,\ell};X_t^{\theta_j})} \right)^2 \right]} \mathrm{d}t \, .$$

Moreover, we know that

$$\mathbb{E} \left[\mathbb{1}_{\{X_t^{\theta_j,\ell} = \mathsf{m}\}} \right] = \mathbb{P}(X_t^{\theta_j,\ell} = \mathsf{m}) = (1 - \alpha_t) \, .$$

This, together with the Cauchy-Schwarz inequality and **H**3, implies

$$\widetilde{E}_{5.2} \leq \frac{C_1 C_3'}{C_4'} \, \mathrm{e}^{5\delta_j B_r} \int_0^1 \eta_t \sqrt{1-\alpha_t} \sum_{\ell=1}^{L} \sqrt{\mathbb{E} \left[\mathbb{1}_{\{X_t^{\theta_j,\ell} = \mathsf{m}\}} \left(\mathrm{e}^{h_t^{\theta_j,\ell}(X_0^{\theta_j,\ell};X_t^{\theta_j})} - \mathrm{e}^{h_t^{\varphi_j,\ell}(X_0^{\theta_j,\ell};X_t^{\theta_j})} \right)^2 \right]} \mathrm{d}t$$

$$\leq \frac{C_1 C_3'}{C_4'} \, \mathrm{e}^{5\delta_j B_r} \int_0^1 \eta_t \sqrt{1-\alpha_t} \sqrt{L \sum_{\ell=1}^{L} \mathbb{E} \left[\mathbb{1}_{\{X_t^{\theta_j,\ell} = \mathsf{m}\}} \left(\mathrm{e}^{h_t^{\theta_j,\ell}(X_0^{\theta_j,\ell};X_t^{\theta_j})} - \mathrm{e}^{h_t^{\varphi_j,\ell}(X_0^{\theta_j,\ell};X_t^{\theta_j})} \right)^2 \right]} \mathrm{d}t$$

$$\leq \frac{C_1 C_3'}{C_4'} \, \sqrt{L} \mathrm{e}^{5\delta_j B_r} \sqrt{\int_0^1 \eta_t (1-\alpha_t) \mathrm{d}t} \sqrt{\int_0^1 \eta_t \, \mathbb{E} \left[\sum_{\ell \in \mathcal{M}(X_t^{\theta_j})} \left(\mathrm{e}^{h_t^{\theta_j,\ell}(X_0^{\theta_j,\ell};X_t^{\theta_j})} - \mathrm{e}^{h_t^{\varphi_j,\ell}(X_0^{\theta_j,\ell};X_t^{\theta_j})} \right)^2 \right] \mathrm{d}t}$$

$$\leq \frac{C_1 C_3'}{C_4'} \, \sqrt{L} \mathrm{e}^{5\delta_j B_r} \varepsilon_{C,j} \, .$$

Putting $\widetilde{E}_{5.1}$ and $\widetilde{E}_{5.2}$ together yields

$$\widetilde{E}_5 \leq 4\sqrt{2} \, C_3' \, \delta_j B_r L \mathrm{e}^{6\delta_j B_r} \sqrt{E_j} + C_3' C_5' \, \mathrm{e}^{6\delta_j B_r} E_j + \frac{C_1 C_3'}{C_4'} \, \sqrt{L} \mathrm{e}^{5\delta_j B_r} \varepsilon_{C,j} \, . \tag{39}$$

Combining (32), (34), (37), (38) and (39), we obtain the recursion

$$\begin{aligned} E_{j+1} \leq \, & (\mathrm{e}^{4\delta_j B_r} + C_3' C_5' \mathrm{e}^{6\delta_j B_r}) E_j + 4\sqrt{2}(1 + C_3') \, \delta_j B_r L \mathrm{e}^{6\delta_j B_r} \sqrt{E_j} \\ & + C_1 \mathrm{e}^{5\delta_j B_r} (\sqrt{L} + 1 + C_3' \sqrt{L}/C_4') \varepsilon_{C,j} + \mathbf{C} \mathrm{e}^{4\delta_j B_r} (1 + \sqrt{L|\bar{\mathsf{V}}|}) \varepsilon_{M,j} \, . \end{aligned} \tag{40}$$

Define

$$\varepsilon = \max \left\{ \varepsilon_0 \, , \, \max_j \varepsilon_{C,j} \, , \, \max_j \varepsilon_{M,j} \right\} \, ,$$

and assume $0 \leq \varepsilon \leq 1$. Fix $J \in \mathbb{N}^*$ and let us prove by induction that for every phase $0 \leq j \leq J$, there exists $\widetilde{C}_j$ depending on $B_r, L, |\bar{\mathsf{V}}|, \{\delta_k\}_{k=0}^{j-1}, C_1, C_2, C_3, C_4, C_5', C_3', C_4'$ such that

$$E_j \leq \widetilde{C}_j \varepsilon^{2^{-j}} \, . \tag{41}$$

For $j = 0$, by **H**2, (41) holds as

$$E_0 \leq \varepsilon_0 \leq \varepsilon \, .$$

Suppose (41) holds for phase $j \in \{0, \dots, J-1\}$, *i.e.*, there exists $\widetilde{C}_j$ independent of ε such that $E_j \leq \widetilde{C}_j \varepsilon^{2^{-j}}$. Plugging that into (40) yields

$$E_{j+1} \leq (\mathrm{e}^{4\delta_j B_r} + C_3' C_5' \mathrm{e}^{6\delta_j B_r}) \widetilde{C}_j \varepsilon^{2^{-j}} + 4\sqrt{2\widetilde{C}_j}(1 + C_3')\, \delta_j B_r L \mathrm{e}^{6\delta_j B_r} \varepsilon^{2^{-(j+1)}} + b_j \varepsilon \;,$$

where

$$b_j = C_1 \mathrm{e}^{5\delta_j B_r} (\sqrt{L} + 1 + C_3' \sqrt{L}/C_4') + \mathbf{C} \mathrm{e}^{4\delta_j B_r} (1 + \sqrt{L|\bar{\mathsf{V}}|}) \;.$$

The fact that $\varepsilon \in [0, 1]$ leads to

$$\varepsilon^{2^{-j}} \leq \varepsilon^{2^{-(j+1)}} \qquad \text{and} \qquad \varepsilon \leq \varepsilon^{2^{-(j+1)}} \;.$$

Therefore

$$E_{j+1} \leq \underbrace{\left((\mathrm{e}^{4\delta_j B_r} + C_3' C_5' \mathrm{e}^{6\delta_j B_r}) \widetilde{C}_j + 4\sqrt{2\widetilde{C}_j}(1 + C_3') \delta_j B_r L \mathrm{e}^{6\delta_j B_r} + b_j \right)}_{=: \widetilde{C}_{j+1}} \varepsilon^{2^{-(j+1)}} \;,$$

which concludes the proof by induction. We deduce that there exists $\widetilde{C}_J$ such that

$$E_J \leq \widetilde{C}_J \varepsilon^{2^{-J}} \;.$$

Recall that $\mathrm{KL}(\mathbb{P}^{a_J}_{[0,1]} \| \mathbb{P}^{\theta_J}_{[0,1]}) \leq E_J$, thus (31) implies

$$\mathrm{KL}(p_0^{a_J} \| p_0^{\theta_J}) \leq \widetilde{C}_J \varepsilon^{2^{-J}} \;,$$

which yields the convergence of our IVT algorithm as $\varepsilon \to 0$. The proof then concludes.

□

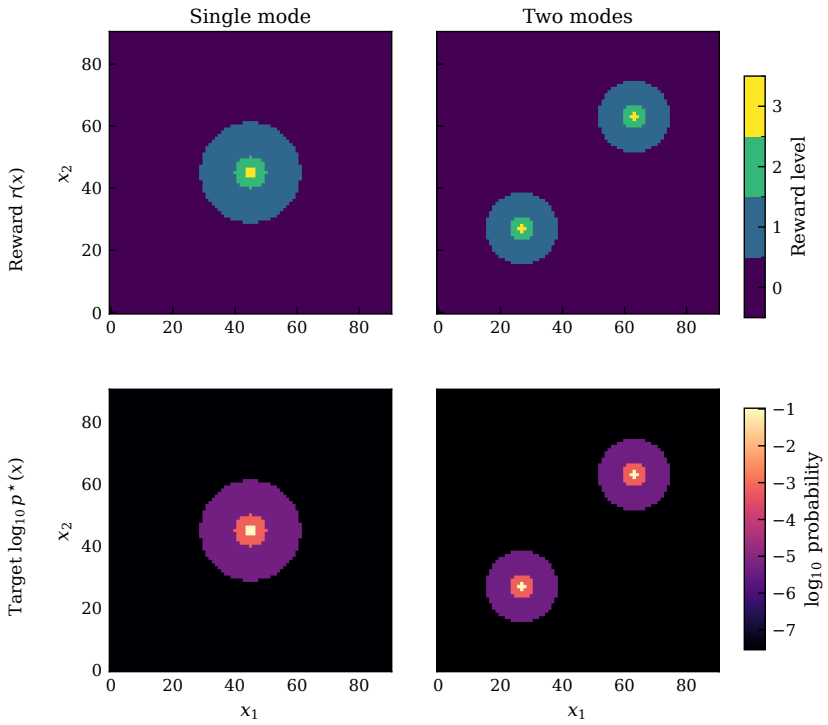


Figure 4: Reward landscapes (top) and corresponding exact targets under the final tilt $A = 5$ (bottom) for the single-mode and two-mode experiments. Target probabilities are shown on a base-10 logarithmic scale.

E Experimental details

E.1 Two-dimensional experiments

E.1.1 Architecture and implementation details

Denoiser parameterization. Both experiments use length-two sequences with tokens in $\{1, \dots, 91\}$ and the mask token m. The denoiser is a table of logits $\Theta \in \mathbb{R}^{92\times92\times91\times2}$. For each partially masked state, it assigns a probability to every valid choice of a masked position and its token. When both positions are masked, the model chooses which position to reveal first and which token to place there; when one position remains masked, it only chooses its token. The probability of a completed sequence is the sum of the probabilities of generating it through the two possible reveal orders.

Initialization. The denoiser is initialized to the uniform base distribution: the 2×91 transitions from (m, m) have probability $1/(2 \cdot 91)$, and the 91 transitions from any state with one mask have probability $1/91$.

Reward geometry. The 91×91 grid has coordinates in $\{0, \dots, 90\}^2$, corresponding to tokens 1 through 91. For the single-mode target, let d^2 be the squared distance to $(45, 45)$. The reward is 3 for $d^2 \le 2$, 2 for $2 < d^2 \le 25$, 1 for $25 < d^2 \le 261$, and 0 otherwise. For the two-mode target, d^2 is the minimum squared distance to $(27, 27)$ and $(63, 63)$, with cutoffs 1, 13, and 130. The rewards and exact targets appear in Figure 4. With final tilt $A = 5$, phase j targets $p_{a_j}(x) \propto p_0(x)\mathrm{e}^{a_j r(x)}, a_j = \frac{5j}{J}$.

Conditional soft value model. Given a partially masked two-token state, the value model predicts the value obtained by assigning each of the 91 possible tokens to either position. It therefore produces 2×91 outputs in a single forward pass. The input encodes the two current tokens with

learned 32-dimensional embeddings, together with the normalized coordinates and mask status of both positions, and is processed by two width-256 SiLU layers. Initially, output weights are zero and output biases equal the logarithm of the empirical mean of $e^{(A/J)r}$ on the first terminal bank. Parameters are warm-started across phases, but Adam is reset each phase. The batch size is 256, the learning rate is 10^{-3}, and weight decay is zero.

Phase-local training. Each phase freezes the denoiser as the teacher and samples a new terminal bank. Value-model minibatches are drawn with replacement; distillation uses every endpoint once in random order. For re-noising, $t \sim \mathrm{Unif}([0, 1])$ and each position is retained independently with probability t; fully unmasked states are rejected, leaving one or two masks with equal probability. Distillation enumerates all tokens at every masked position and minimizes categorical forward KL to the locally tilted teacher. With one mask, candidate rewards are evaluated exactly. Denoiser Adam uses learning rate 10^{-2} and zero weight decay, and its parameters and optimizer state persist across phases. Neither optimizer uses clipping or a learning-rate schedule.

Evaluation. We enumerate all 91^2 completed states to compute the model distribution, target normalizer, and total variation distance exactly. All runs are executed on CPU.

E.1.2 Resource accounting for the fixed-total-compute experiments

Table 3 reports the full-run budgets used in Figure 2, where D, Q, and U denote the numbers of terminal rollouts, value-model updates, and denoiser updates, respectively. For J phases, the total budgets D, Q, and U are divided approximately evenly across phases. Thus, changing J changes only their per-phase allocation, not the total compute budget.

Compute level	**Rollouts** D	**Value updates** Q	**Denoiser updates** U
$0.1\times$	32,000	1,600	1,000
$0.3\times$	96,000	4,800	3,000
$1\times$	320,000	16,000	10,000

Table 3: Full-run compute budgets.

E.1.3 Additional visualizations

The following heat maps show the best number of phases at each compute level. Columns compare the pretrained distribution, exact target, and alignment variants; rows correspond to compute levels.

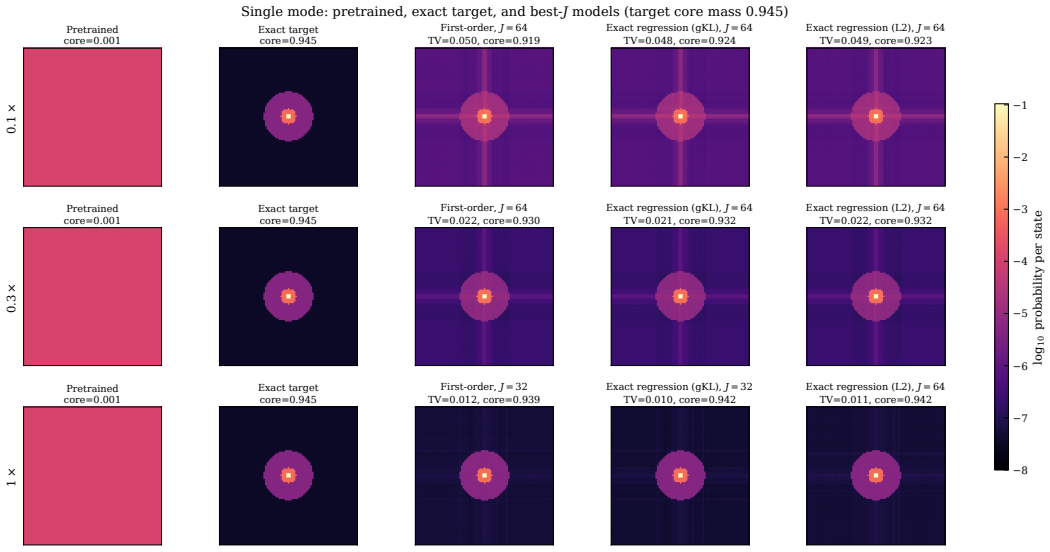


Figure 5: Best single-mode distributions at each compute level, with total variation and target-core mass.

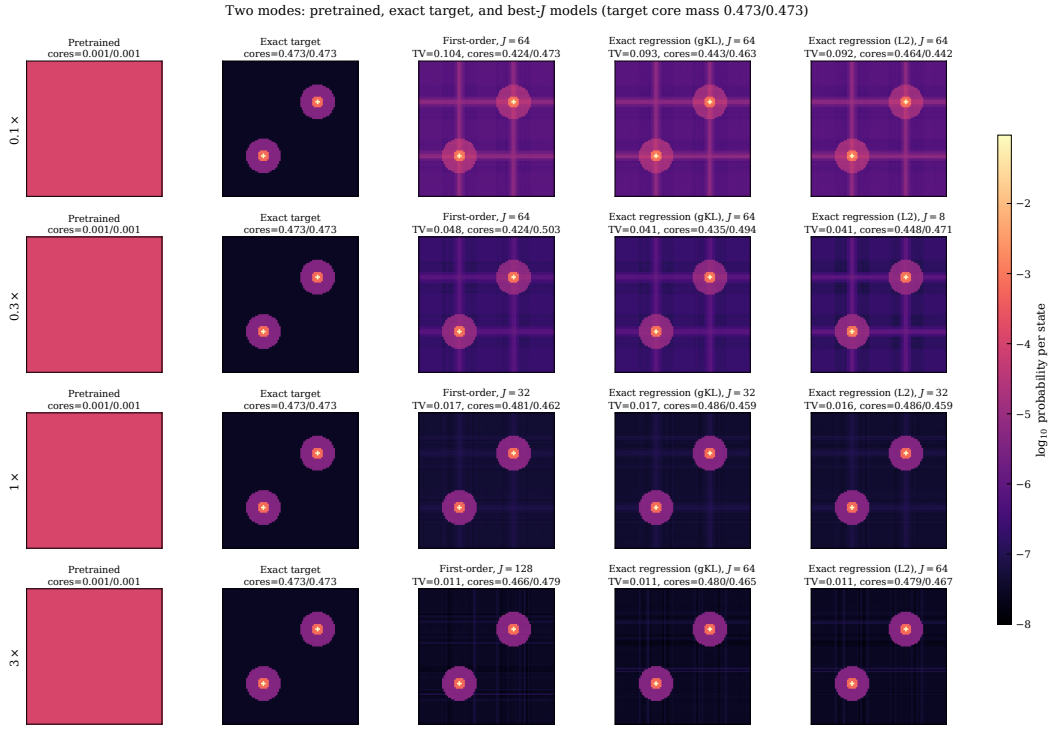


Figure 6: Best two-mode distributions at each compute level, with total variation and separate core masses for the two modes.

E.2 REGULATORY DNA EXPERIMENT

We use the Gosai regulatory-DNA benchmark and the pretrained masked diffusion model released with DRAKES (Wang et al., 2024), together with the data representation and evaluation protocol used by DRAKES and SDPO (Han et al., 2025). The goal is to generate enhancer sequences with high activity in the HepG2 cell line while preserving the other properties of active regulatory DNA. The benchmark contains approximately 700,000 regulatory sequences of length 200 base pairs with experimentally measured activity, and the masked diffusion model is pretrained on the full sequence set (Sahoo et al., 2024). We report the choices that allow to reproduce our experiments.

Tilt path	
Regression variant	Exact exponential-reward regression
Number of phases J	80
Tilt increment h	0.25
Final tilt A	20
Reported checkpoint	Phase 80
On-policy samples	
Training-bank size	65,536 per phase
Validation-bank size	4,096 per phase
Training-oracle queries	5,776,200 in total
Conditional soft value model	
Optimizer and learning rate	AdamW, 10^{-4}
Optimization budget	8,192 updates per phase
Batch size	128
Warm-started quantities	Parameters and optimizer state
Denoiser distillation	
Optimizer and learning rate	AdamW, 10^{-5}
Optimization budget	1,024 updates per phase
Batch size	128
Carried across phases	Parameters and optimizer state

Table 4: Principal parameters and hyperparameters of the regulatory DNA experiment.

Reward models and evaluation. Following Wang et al. (2024), the dataset is divided into disjoint subsets on which two Enformer-based predictors of HepG2 activity are trained independently. The first predictor serves as the fine-tuning oracle, and its predictions provide the only reward signal used by IVT. The second predictor is never exposed during optimization and is used only to report *Pred-Activity*. This separation assesses whether the improvements obtained during fine-tuning transfer to an independently trained activity predictor rather than merely exploiting the oracle used for optimization.

The complementary metrics assess whether reward optimization preserves the reference sequence distribution. We compare the generated sequences with the top 0.1% of measured sequences ranked by activity and report the correlations of their 3-mer frequencies (*3-mer Corr.*) and JASPAR motif frequencies (*JASPAR-Corr.*) (Castro-Mondragon et al., 2022). We also report the fraction classified as accessible chromatin (*ATAC-Acc.*) and the median approximate log-likelihood under the pretrained diffusion model (*App-Log-Lik.*), estimated using the discrete-diffusion ELBO (Sahoo et al., 2024). None of these evaluation metrics is used during fine-tuning.

On-policy sequence banks. At the beginning of a phase, the current denoiser is frozen as the teacher and produces a fresh reward-labeled training bank together with a separate validation bank. Both banks are replaced at the next tilt phase. They are generated from the fully masked state using the standard ancestral sampler for the masked diffusion reverse process, with 32 reverse steps and batch size 128. During optimization, sequences are sampled with replacement from the current bank, and corruption times are resampled for every batch using antithetic sampling on $[10^{-3}, 1]$.

Conditional soft value model. The conditional soft value model uses the same convolutional backbone as the pretrained denoiser. It processes the sequence with 20 one-dimensional residual convolutional blocks, each using 128 hidden channels, and replaces the denoiser output projection by a four-token output head. Given a corrupted sequence, one forward pass produces an $L \times 4$ table whose entry for position ℓ and token y estimates the log conditional exponential reward associated with assigning y at that position.

Initialization across phases. At the first phase, the conditional soft value backbone is initialized from the pretrained denoiser. Its output-head weights are initialized to zero, and every output bias is initialized to the log empirical mean of e^{hr} on the first bank. At every later phase, both the conditional soft value parameters and their optimizer state are warm-started from the preceding phase. Once the conditional soft value model has been fitted and frozen, the next denoiser is distilled from the locally tilted target. The student is initialized from the current denoiser, and its optimizer state is likewise retained across phases. Both optimizers use zero weight decay, a gradient-norm threshold of 1, and no learning-rate schedule. Distillation uses neither a fixed-reference KL penalty nor a fine-tuning exponential moving average.

Exact final-position values. The conditional soft value model normally estimates an expectation over the possible clean completions of a corrupted sequence. However, when a sampled corrupted sequence contains only one masked position, choosing a candidate token uniquely determines the completed sequence, so this expectation can be evaluated exactly. We therefore fill the remaining position successively with A, C, G, and T, evaluate the reward of each completed sequence, and use the four resulting corrections directly in the categorical distillation target instead of using the conditional soft value prediction. These reward evaluations are included in the oracle-query budget reported in Table 4.

Training time. The complete training run uses one NVIDIA L40S GPU and takes approximately 11 hours.

E.3 Protein inverse-folding experiment

We use the DRAKES protein inverse-folding benchmark (Wang et al., 2024), its released ProteinMPNN-based backbone-conditional discrete flow model (Dauparas et al., 2022; Campbell et al., 2024), and the Megascale stability data (Tsuboyama et al., 2023). We retain the data representation, backbone splits, and fine-tuning and evaluation oracles used by DRAKES and SDPO (Han et al., 2025). The choices needed to reproduce our experiment are reported below.

Data splits and reward models. The DRAKES split contains 339 training backbones, together with ten validation and twelve test backbones that are disjoint from the training set. The validation and test backbones are not used to fit either the conditional soft value model or the denoiser. As in DRAKES, we use two separately trained ProteinMPNN-based stability predictors. The fine-tuning oracle, trained only on the Megascale training split, supplies the only reward labels used by IVT, whereas the evaluation oracle is never exposed during optimization and is used only to score the generated sequences.

Evaluation protocol. For each of three sampling seeds, we generate 128 sequences for each of the twelve held-out test backbones. *Pred-ddG*, computed by the evaluation oracle, measures the predicted stability improvement relative to the corresponding wild-type sequence, with larger values indicating higher predicted stability. To assess whether a generated sequence remains compatible with the conditioning structure, we fold it with ESMFold and compare the predicted structure with the target backbone using self-consistency RMSD (*scRMSD*) (Lin et al., 2023). Our primary metric is the fraction of sequences satisfying both Pred-ddG > 0 and scRMSD $< 2\,\text{Å}$, which measures successful predicted stability improvement while preserving the target structure. We additionally report the individual success rates and the median Pred-ddG and scRMSD.

On-policy sequence banks. At the beginning of each phase, the current denoiser is frozen as the teacher and generates a new reward-labeled bank containing 12 sequences for each of the 339

Tilt path	
Regression variant	First-order reward regression
Number of phases J	64
Tilt increment h	2
Final tilt A	128
Reported checkpoint	Phase 64
On-policy samples	
Training backbones	339
Samples per backbone	12 per phase
Training-bank size	4,068 per phase
Fine-tuning-oracle queries for training banks	260,352 in total
Conditional soft value model	
Backbone architecture	ProteinMPNN/FMIF
Hidden dimension and graph layers	128; 3 encoder and 3 decoder layers
Optimizer and learning rate	AdamW, 10^{-4}
Optimization budget	512 updates per phase
Batch size	64
Warm-started quantities	Parameters and optimizer state
Denoiser distillation	
Optimizer and learning rate	AdamW, 10^{-5}
Optimization budget	256 updates per phase
Batch size	64
Carried across phases	Parameters and optimizer state

Table 5: Principal parameters and hyperparameters of the protein inverse-folding experiment.

training backbones. Backbones therefore receive uniform weight, and the resulting training bank contains 4,068 sequences. The bank is discarded and regenerated at the following tilt phase.

Sequences are generated from the fully masked state using the standard stochastic continuous-time sampler of the released FMIF model. We use 50 reverse steps, sampling temperature 0.1, and sampling batch size 32. During optimization, examples are sampled with replacement from the current training bank. A new corruption time is drawn independently for every example and update, uniformly on $[10^{-2}, 1]$.

Conditional soft value model. The conditional soft value model uses the same backbone-conditioned ProteinMPNN architecture as the pretrained FMIF denoiser, with hidden dimension 128, three encoder layers, three decoder layers, and 30 nearest neighbors. The denoiser output projection is replaced by a 21-token value head. Given a target backbone and a corrupted sequence, one forward pass produces an $L \times 21$ table. Its entry at residue position ℓ and token y estimates the conditional expected terminal ddG reward associated with assigning y at that position.

Initialization across phases. At the first phase, the conditional soft value backbone is initialized from the pretrained denoiser. Its output-head weights are initialized to zero, and every output bias is initialized to the empirical mean reward of the first training bank. At every subsequent phase, the conditional soft value parameters and their optimizer state are warm-started from the preceding phase.

Once the conditional soft value model has been fitted and frozen, the current denoiser is updated by distillation from the locally tilted target. The student starts from the current denoiser, while its optimizer state is retained across phases. Both optimizers use zero weight decay, a gradient-norm threshold of 1, and no learning-rate schedule. Distillation uses neither a fixed-reference KL penalty nor an exponential moving average.

Training time. The complete 64-phase training run uses one NVIDIA L40S GPU and takes approximately 5 hours, excluding the subsequent ESMFold and scRMSD evaluation.