DISCRETE DIFFUSION WITH SAMPLE-EFFICIENT ESTIMATORS FOR CONDITIONALS

Anonymous authors
Paper under double-blind review

ABSTRACT

We study a discrete denoising diffusion framework that integrates a sample-efficient estimator of single-site conditionals with round-robin noising and denoising dynamics for generative modeling over discrete state spaces. Rather than approximating a discrete analog of a score function, our formulation treats single-site conditional probabilities as the fundamental objects that parameterize the reverse diffusion process. We employ a sample-efficient method known as Neural Interaction Screening Estimator (NeurISE) to estimate these conditionals in the diffusion dynamics. Controlled experiments on synthetic Ising models, MNIST, and scientific data sets produced by a D-Wave quantum annealer, synthetic Potts model and one-dimensional quantum systems demonstrate the proposed approach. On the binary data sets, these experiments demonstrate that the proposed approach outperforms popular existing methods including ratio-based approaches, achieving improved performance in total variation, cross-correlations, and kernel density estimation metrics.

1 INTRODUCTION

Generative modeling over discrete spaces is fundamental to a wide range of applications, including molecular design, language modeling, and policy learning in reinforcement learning Ho & Ermon (2016); Bengio et al. (2003); Jin et al. (2018). In these settings, data consist of categorical or binary variables with complex statistical dependencies, and accurately capturing their joint structure requires models that can scale to high-dimensional combinatorial configuration spaces.

While diffusion models have revolutionized generative modeling in continuous domains Ho et al. (2020), their direct application to discrete data has received increased attention. Continuous-time formulations rely on Gaussian noise and score estimation through gradients of log-densities, quantities that are not well defined in discrete spaces. Naive relaxations, such as adding continuous noise to one-hot encodings, break the discrete structure and often yield poor sample quality or unstable training.

These limitations motivate the need for a principled framework for *discrete diffusion processes* that preserves the combinatorial structure of the data, allows tractable inference, and retains the interpretability and scalability that made diffusion models successful in continuous domains.

A number of works have considered extending diffusion models to discrete spaces. Austin et al. (2021) considers denoising diffusion for discrete data and discrete time (both absorbing and uniform diffusion), optimizing the variational lower bound (VLB) of the log-likelihood. A continuous-time framework for discrete diffusion models is introduced in Campbell et al. (2022), which also optimizes the VLB of the log-likelihood. The work Sun et al. (2022) performs score matching for continuous-time diffusion by learning conditionals using cross-entropy. In Lou et al. (2024), the authors propose an approach to learn the discrete version of the score using a score-entropy function to ensure non-negativity of the score along the training iterations. For a broader survey on discrete diffusion models see Li et al. (2025).

This paper's contribution begins with making explicit that for forward transitions the canonical time-reversed kernel can be parameterized entirely through ratios of probabilities between configurations that differ at one coordinate, and that these ratios reduce exactly to ratios of single-site conditional distributions given the remaining coordinates. So reverse diffusion can be implemented by learning

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.

local conditionals instead of a global density or discrete *score.* The main methodological contribution following this observation is to plug in state-of-the-art estimator for learning discrete conditionals. Motivated by theoretical results on the sample-efficiency of interaction-screening estimators, including RISE Vuffray et al. (2016) and GRISE Jayakumar et al. (2020) for learning discrete conditional distributions, we use the Neural Interaction Screening Estimator (NeurISE) Jayakumar et al. (2020) that has shown strong empirical advantage for learning discrete conditionals with unknown structure. We apply this estimator to the round-robin noising diffusion scheme previously also explored in Varma et al. (2024).

We evaluate the proposed estimation approach for ratios of conditionals using NeuRISE on a range of discrete generative modeling benchmarks, including synthetic Ising on a 25-variable system, binarized MNIST, and quantum annealing (D-Wave) datasets. Across these settings, we compare against the classifier based estimator of Varma et al. (2024) and three other methods: ELBO-based method D3PM Austin et al. (2021), the score based approach of SEDD method Lou et al. (2024), and the flow matching approach of discrete flow matching Gat et al. (2024). We demonstrate consistent improvements in distributional accuracy with comparable run times with existing models. A limitation of our study is that we didn't train on very large architectures or data sets. Instead, our experiments are intentionally designed around controlled smaller-scale benchmarks. This choice allows us to study the statistical behavior of the proposed estimator and compare denoising algorithms under repeated trials, varying sample sizes, and measurable error decay. In particular, the 25-variable Ising model serves as a key controlled setting in which exact samples can be produced and total variation distance can be computed exactly, enabling a detailed comparison of how different methods improve with additional training data.

An important observation we make is that, under the round-robin noising scheme, autoregressive generation arises as the hard-noise limit of the reverse-time sampler. In this limit, each reverse step resamples one coordinate from its single-site conditional distribution, and applying these updates in a fixed order recovers an autoregressive sampling procedure. This provides a unified view of the diffusion sampler and the autoregressive sampler associated with Varma et al. (2024), without introducing a separate autoregressive model. This connection is distinct from that of Ou et al. (2024), who relate absorbing diffusion to any-order autoregressive models at the level of the training objective. In our setting, the connection occurs at the sampler level. That is, under round-robin noising, autoregressive generation emerges directly from the finite-step hard-noise reverse process.

2 PROBLEM

Let Σ denote a discrete set of alphabets with cardinality $|\Sigma| = p$. We define the configuration space Σ^q for q discrete variables or coordinates, with elements denoted as $\sigma := (\sigma_1, ...\sigma_q)$. Given training samples from a probability distribution $\mu : \Sigma^q \to \mathbb{R}$, *the goal is to construct a diffusion-based generative model from which new samples can be tractably generated.*

We will specifically focus on a general class of probabilistic models over Σ^q defined using a Hamiltonian $H : \Sigma^q \to \mathbb{R} : \mu(\sigma) = \frac{1}{Z} \exp(H(\sigma)), \quad \sigma \in \Sigma^q$, where the partition function $Z = \sum_{\sigma \in \Sigma^q} \exp(H(\sigma))$ ensures normalization. This formulation defines an energy-based probability distribution from the exponential family, where the energy function $H(\sigma)$ typically encodes interactions between components of σ, such as pairwise terms or external fields.

An important example of such a model is the *Ising model*, where $\Sigma := \{-1, 1\}$which is a pairwise Markov random field over Σ^q with interactions defined by an undirected graph $G = (V, E)$, where $V = [q]$ and $E \subseteq V \times V$. The Hamiltonian takes the form: $H(\sigma) = \sum_{(i,j)\in E} J_{ij}\sigma_i\sigma_j + \sum_{i\in[q]} h_i\sigma_i$. Here $J_{ij} \in \mathbb{R}$ represents the strength of the interaction between nodes i and j, $h_i \in \mathbb{R}$ represents an external bias or field at node i, $\sigma_i \in \Sigma$ for all $i \in [q]$. This model defines a probability distribution:

$$\mu(\sigma) = \frac{1}{Z} \exp\left(\sum_{(i,j)\in E} J_{ij}\sigma_i\sigma_j + \sum_{i\in[q]} h_i\sigma_i \right), \quad \sigma \in \Sigma^q. \tag{1}$$

3 DISCRETE DIFFUSION THROUGH CONDITIONALS

We consider a denoising diffusion framework over the configuration space Σ^q, which consists of a known forward Markov process and a learned reverse process. The goal is to construct a forward Markov process $\{X_n\}_{n=0}^T$ such that $X_0 \sim \mu_0$ is the data distribution that we are interested in learning. The forward process then makes sure that the distribution μ_n of X_n converges to a tractable noise distribution, from which it is easy to sample: $\lim_{n\to\infty} \mu_n = \mu_{\text{noise}}$. The Markov chain over Σ^q evolves according to a known transition kernel $k_n : \Sigma^q \times \Sigma^q \to \mathbb{R}_{\geq 0}$ which defines the conditionals $k_n(\sigma, \tilde{\sigma}) := \mathbb{P}(X_{n+1} = \sigma | X_n = \tilde{\sigma})$. The forward evolution of the distribution μ_n of the process X_n is then given by:

$$\mu_{n+1}(\sigma) = \sum_{\tilde{\sigma} \in \Sigma^q} k_n(\sigma, \tilde{\sigma}) \mu_n(\tilde{\sigma}). \tag{2}$$

REVERSE PROCESS

To sample from the target distribution μ_0, we define a reverse process $\{Y_n\}_{n=0}^T$ such that $Y_n \sim \mu_{T-n}$. This reverse process is governed by a family of time-inhomogeneous transition kernels $k_n^{\text{rev}} : \Sigma^q \times \Sigma^q \to \mathbb{R}_{\geq 0}$, which satisfy the backward recurrence:

$$\mu_n(\sigma) = \sum_{\tilde{\sigma} \in \Sigma^q} k_n^{\text{rev}}(\sigma, \tilde{\sigma}) \mu_{n+1}(\tilde{\sigma}). \tag{3}$$

We obtain a natural candidate for the reverse-time transition kernel via Bayes' rule is the following:

$$k_n^{\text{rev}}(\sigma, \tilde{\sigma}) = k_n(\tilde{\sigma}, \sigma) \cdot \frac{\mu_n(\sigma)}{\sum_{\hat{\sigma} \in \Sigma^q} k_n(\tilde{\sigma}, \hat{\sigma}) \mu_n(\hat{\sigma})} = \frac{k_n(\tilde{\sigma}, \sigma)}{\sum_{\hat{\sigma} \in \Sigma^q} k_n(\tilde{\sigma}, \hat{\sigma}) \cdot \frac{\mu_n(\hat{\sigma})}{\mu_n(\sigma)}}. \tag{4}$$

This expression shows that the reverse kernel depends only on the forward transition probabilities and on ratios of the form $\mu_n(\hat{\sigma})/\mu_n(\sigma)$. Consequently, if these ratios can be accurately estimated we can construct accurate approximations to the reverse kernels, k_n^{rev}. We can then sample from the noise distribution μ_{noise} and iteratively apply the reverse transitions to generate new samples from an approximation of the data distribution.

This captures the core idea of denoising diffusion probabilistic models (DDPMs). The forward process gradually drives the data distribution toward a simple noise distribution, while the reverse process reconstructs samples by inverting this dynamics using learned conditional structure. Theorem C.1 in the Appendix formalizes this intuition as discrete-time discrete-state analogue of existing results in literature for continuous-time continuous space De Bortoli et al. (2021); Chen et al. (2022); Chen & Ying (2024), and continuous-time discrete state Zhang et al. (2024); Ren et al. (2024), added for the sake of completeness.

ROUND-ROBIN FORWARD NOISING

We now describe a choice of forward dynamics, according to a noising scheme introduced in Varma et al. (2024), that will be used in this paper. In this choice of the forward process, we gradually introduce noise into the configuration by modifying one coordinate (e.g., a pixel or spin) at a time. An advantage of this form of noising is that the number of ratios that are required to be learned for each time step is much smaller than with other schemes Austin et al. (2021); Sun et al. (2022); Lou et al. (2024), where all the variables are noised simultaneously. The precise scheme for noising is the following:

1. A randomization parameter $\varepsilon \in [0, 1]$ is fixed.
2. At each time step $n \in \{1, \ldots, T\}$, a specific coordinate is selected in round-robin order: the n-th coordinate of σ is selected as $u = ((n-1) \mod q) + 1$.
3. With probability ε the coordinate σ_u is left unchanged. With probability $1 - \varepsilon$ the coordinate value σ_u is uniformly randomly sampled from Σ.

Since there are p number of elements in Σ, the conditional probabilities if this forward noising process are given by:

$$k_n(\sigma, \tilde{\sigma}) = \begin{cases} \frac{1-\varepsilon}{p}, & \text{if } \sigma_{-u} = \tilde{\sigma}_{-u},\ \sigma_u \neq \tilde{\sigma}_u, \\ \frac{1-\varepsilon}{p} + \varepsilon, & \text{if } \sigma = \tilde{\sigma}, \\ 0, & \text{otherwise}, \end{cases}$$

where $\sigma_{-u} \in \Sigma^{q-1}$ denotes the configuration excluding the u-th coordinate. For notational convinience in the forthcoming expressions, we define the parameters, $a = \frac{1-\varepsilon}{p},\ \ b = \frac{1-\varepsilon}{p} + \varepsilon$, the probability of noising the chosen coordinate to an alphabet different from its current value, and the probability of it picking the current alphabet again, respectively.

Since the only admissible transitions are one coordinate transition away at the noised coordinate $u = ((n-1) \mod q) + 1$, we can substitute the expression for $k_n(\sigma, \tilde{\sigma})$ in equation 4 to express the reverse kernel as,

$$k_n^{\text{rev}}(\sigma, \tilde{\sigma}) = \frac{a\,\mu_n(\sigma)}{a\,\mu_n(\sigma) + b\,\mu_n(\tilde{\sigma}) + a \sum\limits_{\hat{\sigma} \in \mathcal{N}_u(\tilde{\sigma}) \setminus \{\sigma, \tilde{\sigma}\}} \mu_n(\hat{\sigma})}$$

with $\mathcal{N}_u(\tilde{\sigma}) := \{\hat{\sigma} \in \Sigma^q : \hat{\sigma}_{-u} = \tilde{\sigma}_{-u}\}$. Therefore, this gives rise to three possibilities,

$$k_n^{\text{rev}}(\sigma, \tilde{\sigma}) = \frac{a\,\mu_n(\sigma)}{b\,\mu_n(\tilde{\sigma}) + a \sum\limits_{\mathcal{N}_u(\tilde{\sigma}) \setminus \{\tilde{\sigma}\}} \mu_n(\hat{\sigma})}, \tag{5}$$

when $\sigma_{-u} = \tilde{\sigma}_{-u},$ but $\sigma \neq \tilde{\sigma}$. Similarly,

$$k_n^{\text{rev}}(\sigma, \sigma) = \frac{b\,\mu_n(\sigma)}{b\,\mu_n(\sigma) + a \sum\limits_{\mathcal{N}_u(\sigma) \setminus \{\sigma\}} \mu_n(\hat{\sigma})}, \tag{6}$$

and $k_n(\sigma, \tilde{\sigma}) = 0$ otherwise. Of special interest is the case when the discrete set is binary: $\Sigma := \{-1, 1\}$ is binary. Then, we get,

$$k_n(\sigma, \tilde{\sigma}) = \begin{cases} \frac{1-\varepsilon}{2}, & \text{if } \sigma_{-u} = \tilde{\sigma}_{-u},\ \sigma_u \neq \tilde{\sigma}_u, \\ \frac{1+\varepsilon}{2}, & \text{if } \sigma = \tilde{\sigma}, \\ 0, & \text{otherwise}, \end{cases} . \tag{7}$$

The corresponding reverse transition probabilities are:

$$k_n^{\text{rev}}(\sigma, \tilde{\sigma}) = (1-\varepsilon) \cdot \frac{\mu_n(\sigma)}{(1-\varepsilon)\mu_n(\sigma) + (1+\varepsilon)\mu_n(\tilde{\sigma})}, \quad \text{if } \sigma_{-u} = \tilde{\sigma}_{-u} \text{ and } \sigma_u \neq \tilde{\sigma}_u,$$

and

$$k_n^{\text{rev}}(\sigma, \sigma) = (1+\varepsilon) \cdot \frac{\mu_n(\sigma)}{(1+\varepsilon)\mu_n(\sigma) + (1-\varepsilon)\mu_n(\tilde{\sigma})}.$$

The Hard Noise Autoregressive Limit

We now consider a special case of the forward process in which noise is *harsh*: $\varepsilon = 0$. This corresponds to a full randomization of the selected coordinate at each step, and hence all the information is lost in the variable after the corresponding noising step. Our goal in this section is to show that we recover auto-regressive generation in this limit.

Let $T = q$, and define a time-inhomogeneous Markov kernel that updates only the n-th coordinate at time step $n \in \{1, \ldots, T\}$. The forward transition kernel simplifies to,

$$k_n(\sigma, \tilde{\sigma}) = \begin{cases} \frac{1}{p}, & \text{if } \sigma_{-n} = \tilde{\sigma}_{-n}, \\ 0, & \text{otherwise}, \end{cases} \tag{8}$$

Over $T = p$ steps, this procedure fully randomizes each coordinate once, resulting in convergence to the uniform distribution over Σ^q. By the reverse kernel construction, we obtain

$$\mathbb{P}(X_n = \sigma \mid X_{n+1} = \tilde{\sigma}) = k_n^{\text{rev}}(\sigma, \tilde{\sigma}) = \mathbf{1}_{\sigma_{-n} = \tilde{\sigma}_{-n}} \cdot \frac{\mu_n(\sigma)}{\sum_{\hat{\sigma} \in \mathcal{N}_n(\sigma)} \mu_n(\hat{\sigma})}.$$

This corresponds to sampling the n-th coordinate conditioned on the others:

$$\mathbb{P}(X_n^n = \sigma_n \mid X_n^{-n} = \tilde{\sigma}_{-n}) = \frac{\mu_n(\sigma)}{\sum_{\hat{\sigma} \in \mathcal{N}_n(\sigma)} \mu_n(\hat{\sigma})},$$

where the sum is taken over all configurations $\hat{\sigma} \in \Sigma^q$ that agree with σ outside coordinate n.

We now express this reverse process recursively. Let $X_T = \tilde{\sigma}$ be a configuration sampled from the noise distribution. At each reverse step n, the process samples a configuration $X_n = \sigma$ such that $\sigma_{-n} = \tilde{\sigma}_{-n}$, while drawing σ_n from the corresponding conditional distribution. Formally,

$$\mathbb{P}(X_n = \sigma \mid X_{n+1} = \tilde{\sigma}) = \mathbf{1}_{\sigma_{-n} = \tilde{\sigma}_{-n}} \cdot \mathbb{P}(X_n^n = \sigma_n \mid X_n^{-n} = \tilde{\sigma}_{-n}).$$

Unrolling the reverse chain from T to 0 yields

$$\mathbb{P}(X_0 = \sigma \mid X_T = \tilde{\sigma}) = \prod_{n=1}^{T} \mathbb{P}(X_{T-n} = \sigma_{T-n} \mid X_{T-n+1} = \tilde{\sigma}_{T-n+1}). \tag{9}$$

Since the two configurations differ only at coordinate n,

$$\mathbb{P}(X_{T-n} = \sigma \mid X_{T-n+1} = \tilde{\sigma}) = \mathbf{1}_{\sigma_{-n} = \tilde{\sigma}_{-n}} \, \mathbb{P}(\sigma_n \mid \tilde{\sigma}_{-n}).$$

Substituting into equation 9, we obtain

$$\mathbb{P}(X_0 = \sigma \mid X_T = \tilde{\sigma}) = \prod_{n=1}^{T} \mathbf{1}_{\sigma_{-n} = \tilde{\sigma}_{-n}} \, \mathbb{P}(\sigma_n \mid \tilde{\sigma}_{-n}).$$

If the reverse update keeps non-updated coordinates fixed, i.e. $X_n^{-n} = X_{n+1}^{-n}$, then

$$\mathbb{P}(X_0 = \sigma \mid X_T = \tilde{\sigma}) = \prod_{n=1}^{T} \mathbb{P}(\sigma_n \mid \tilde{\sigma}_{>n}), \tag{10}$$

which recovers an autoregressive factorization over the discrete alphabet Σ.

4 Learning Conditionals using Neural Interaction Screening

Let μ_n denote the probability distribution of the random variable X_n. In order to implement the reverse dynamics, we need to estimate the ratio $\frac{\mu_n(\tilde{\sigma})}{\mu_n(\sigma)}$ from samples of the forward process. Suppose $\sigma, \tilde{\sigma} \in \Sigma^q$ differ at only one coordinate n, i.e., $\tilde{\sigma}_i = \sigma_i \quad \text{for all } i \neq n, \ \tilde{\sigma}_n \neq \sigma_n$. Then, for any distribution μ_n over Σ^q, we have:

$$\frac{\mu_n(\tilde{\sigma})}{\mu_n(\sigma)} = \frac{\mu_n(\tilde{\sigma}_n \mid \sigma_{-n})}{\mu_n(\sigma_n \mid \sigma_{-n})},$$

where $\sigma_{-n} \in \Sigma^{q-1}$ denotes the shared values of the configuration outside the n-th coordinate.

This identity follows directly from the definition of conditional probability $\mu_n(\sigma) = \mu_n(\sigma_n \mid \sigma_{-n}) \cdot \mu_n(\sigma_{-n})$. Taking the ratio, we obtain:

$$\frac{\mu_n(\tilde{\sigma})}{\mu_n(\sigma)} = \frac{\mu_n(\tilde{\sigma}_n \mid \sigma_{-n}) \cdot \mu_n(\sigma_{-n})}{\mu_n(\sigma_n \mid \sigma_{-n}) \cdot \mu_n(\sigma_{-n})} = \frac{\mu_n(\tilde{\sigma}_n \mid \sigma_{-n})}{\mu_n(\sigma_n \mid \sigma_{-n})}.$$

This expression provides a tractable way to compute (or approximate) the required ratio using only local conditionals for the revese dynamics.

To compute the required single-site conditional distributions, we use the *Neural Interaction Screening Estimator* (NeurISE) Jayakumar et al. (2020), which learns local conditionals in discrete graphical models by neural parameterization of partial energy functions. This local conditional modeling is well matched to the reverse diffusion kernel, which depends only on ratios of single-site conditionals between configurations differing at one coordinate, enabling efficient and scalable implementation of the reverse-time dynamics without explicitly modeling the global distribution. Following NeurISE Jayakumar et al. (2020), we introduce the centered indicator embedding

$$\Phi_s(r) := \begin{cases} 1 - \dfrac{1}{q}, & r = s, \\ -\dfrac{1}{q}, & r \neq s, \end{cases} \qquad s, r \in \Sigma.$$

We then define the vector-valued embedding, $\Phi(r) := \big(\Phi_1(r), \ldots, \Phi_q(r)\big) \in \mathbb{R}^q$,

Suppose that $\mu_n(\sigma) \propto \exp(H(\sigma))$ is a Gibbs distribution for some Hamiltonian function $H : \Sigma^q \to \mathbb{R}$. For any coordinate $u \in [1, , , q]$, there always exists a decomposition $H(\sigma) = H_{-u}(\sigma_{-u}) + H_u(\sigma)$, where H_{-u} does not depend on σ_u.w Substituting it into the Gibbs distribution $\mu_n(\sigma) \propto \exp(H(\sigma))$ yields

$$\mu_n(\sigma_u \mid \sigma_{-u}) = \frac{\exp(H_{-u}(\sigma_{-u}) + H_u(\sigma))}{\sum_{r\in\Sigma} \exp(H_{-u}(\sigma_{-u}) + H_u(\sigma_u = r, \sigma_{-u}))}.$$

Since $H_{-u}(\sigma_{-u})$ does not depend on σ_u, we get

$$\mu_n(\sigma_u \mid \sigma_{-u}) = \frac{\exp(H_u(\sigma))}{\sum_{r\in\Sigma} \exp(H_u(\sigma_u = r, \sigma_{-u}))}.$$

Therefore, the partial energy H_u determines the single-site conditional distribution up to an additive function of σ_{-u}, and can be written as $H_u(\sigma) = \log \mu_n(\sigma_u \mid \sigma_{-u}) + \mathrm{const}(\sigma_{-u})$. For each σ_{-u}, $H_u(\cdot, \sigma_{-u})$ can chosen be to satisfy $\sum_{r\in\Sigma} H_u(r, \sigma_{-u}) = 0$. The functions Φ_r form a basis for functions, $f : \Sigma \to \mathbb{R}$ that average to 0 or are *centered*. Therefore, for each coordinate $u \in [q]$, we approximate this partial energy using a neural network $\mathrm{NN}_\theta : \Sigma^{q-1} \to \mathbb{R}^q$ Specifically, we use this parameterization at the Hamiltonian level as follows,

$$\widetilde{H}_u(\sigma; w) = \big\langle \Phi(\sigma_u), \mathrm{NN}_\theta(\sigma_{-u}) \big\rangle = \sum_{s=1}^{q} \Phi_s(\sigma_u)\, \mathrm{NN}_\theta(\sigma_{-u})_s.$$

This representation is fully general without loss of expressivity. Given samples $\{\sigma^{(n)}\}_{n=1}^N$ from the forward process at time n, the *NeurISE loss* as presented in Jayakumar et al. (2020), for site u is

$$\mathcal{L}_u(\theta) = \frac{1}{N} \sum_{n=1}^{N} \exp\Big(-\big\langle \Phi(\sigma_u^{(n)}), \mathrm{NN}_\theta(\sigma_{-u}^{(n)}) \big\rangle \Big).$$

Here θ represents the trainable parameters of the neural net. Since we have conditionals that need to be learned be for each time step, we introduce a neural network $NN_\theta : \mathbb{R} \times \mathbb{R}^q \times \mathbb{R}^{q-1} \to \mathbb{R}^p$ that accepts arguments (t, u, σ_{-u}) where the coordinate u is encoded as a one-hot vector. This gives us the composite loss

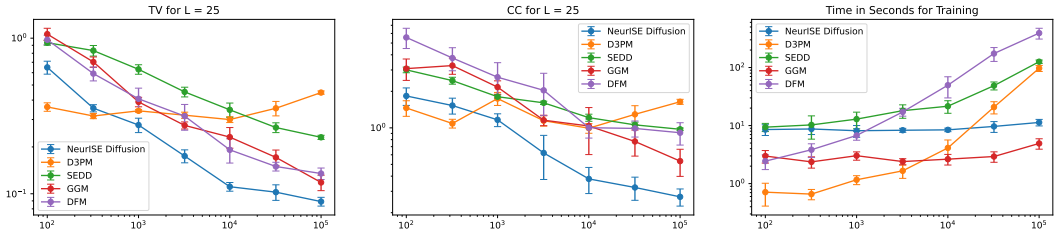
$$\mathcal{L}_u(\theta) = \frac{1}{TN} \sum_{s=1}^{T} \sum_{n=1}^{N} \exp\big(-\big\langle \Phi((X_s^n)_u), \mathrm{NN}_\theta(t, u, (X_s^n)_{-u} \big\rangle \big). \tag{11}$$

Learned Conditional Distribution. Once trained, the approximate conditional distribution is recovered by setting

$$\widehat{\mu}_n(\sigma_u \mid \sigma_{-u}) = \frac{\exp\big(\big\langle \Phi(\sigma_u), \mathrm{NN}_u(\sigma_{-u}) \big\rangle\big)}{\sum_{r\in\Sigma} \exp\big(\big\langle \Phi(r), \mathrm{NN}_u(\sigma_{-u}) \big\rangle\big)}. \tag{12}$$

Therefore, for any pair $(\sigma, \tilde{\sigma})$ differing at coordinate u, the ratio required for the reverse diffusion kernel is given by

$$\frac{\mu_n(\tilde{\sigma})}{\mu_n(\sigma)} \approx \frac{\exp\big(\big\langle \Phi(\tilde{\sigma}_u), \mathrm{NN}_u(\sigma_{-u}) \big\rangle\big)}{\exp\big(\big\langle \Phi(\sigma_u), \mathrm{NN}_u(\sigma_{-u}) \big\rangle\big)}.$$



(a) TV (b) Cross-correlation Error (c) Training Time

Figure 1: Trend of TV and Cross-correlation error as a function of training set size for Ising models, averaged over 5 Ising models and 10 trials per data set.

5 Numerical Experiments

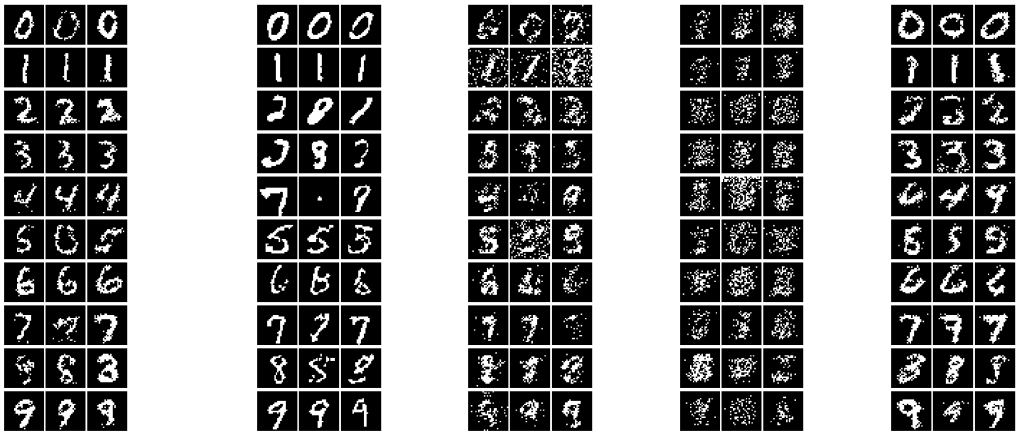
In this section, we compare our denoising method that we refer to as *NeurISE diffusion*, with the algorithm proposed in Varma et al. (2024) (GGM), ELBO-based method of D3PM Austin et al. (2021), the score matching approach (SEDD) proposed in Lou et al. (2024) and discrete flow matching (DFM) Gat et al. (2024). The relevant code used to run the experiments can be found here. We implemented our own version of GGM and SEDD, the D3PM implementation was adapted from an unofficial publicly available implementation Ryu (2024), and DFM was adapted from the publicly provide code in the guide Lipman et al. (2024). The core parametric model we will use in our method and each of these methods will be multilayer perceptrons with different depth, depending on the test cases, with batch normalization layers. We make this choice to study the performance of these methods on an equal footing, decoupled from representational differences coming from the model.

5.1 Test Case 1: Binary Data

Ising Model We first compare different methods on small-scale synthetic data. The benchmark is based on the *Edwards-Anderson (EA) model* Edwards & Anderson (1975); Bhatt & Young (1988), which is a specific instance of the binary Ising model defined by the distribution in Equation equation 1. The EA model Hamiltonian is defined over $p = L^2$ binary variables $\sigma_i \in \{-1, +1\}$, arranged on a two-dimensional square lattice of size $L \times L$. The graph $E \subseteq [q] \times [q]$ corresponds to the set of nearest-neighbor pairs on a 2D periodic grid. Each spin σ_i interacts with its right and bottom neighbors, with periodic boundary conditions applied in both directions. The pairwise couplings J_{ij} are symmetric random variables sampled independently for each edge $(i, j) \in E$ as: $J_{ij} = J_{ji} \in \{-1.2, +1.2\}$. The local fields $h_i \in \{-0.05, +0.05\}$ are also sampled independently for each node $i \in [q]$.

In our experiments, we use a lattice size of $L = 5$, resulting in a model with $q = 25$ binary variables. The results of using a two layer MLP can be seen for different diffusion methods in Figure 1. We test the models for different values of training data across averaged over 5 different choices of EA models, with 10 runs for differing values of training set size: $[100, 320, 1000...10^5]$. The training samples were generated using an exact sampler. The number of test samples were fixed to be 10^5 across all three experiments. We find that the NeuRISE-based denoising estimator shows the sharpest decay in total variation (TV) distance with increase in sample size, and performs better than the GGM based estimator of Varma et al. (2024), the SEDD approach proposed in Lou et al. (2024) and the discrete flow matching method of Gat et al. (2024). For the model presented in Lou et al. (2024), the configurations had to be one-hot coded to make the algorithm work. The model D3PM Austin et al. (2021) performs well for low number of samples but its performance deteriorates as the size of the training set is increased. Interestingly, D3PM does not show monotonic decay of TV as the number of training samples decrease. In each case, we also compute the difference between the cross-correlation matrices of the generated samples and the test data, where the correlations are defined as $C_{ij} = \frac{1}{N} \sum_{k=1}^{N} \sigma_i^{(k)} \sigma_j^{(k)}$. The decay of cross-correlation errors show a similar trend as that for the TV. Cross-correlation metric has the advantage of tractability for larger models where TV can't be efficiently computed.

Another important study that was performed is the comparison of the NeuRISE-based diffusion for different choice of noise parameter. From our experiments, it doesn't seem like the denoising scheme



(a) NeurISE Diffusion (b) D3PM (c) SEDD (d) GGM (e) DFM

Figure 2: Class-conditional MNIST samples. Each subfigure shows generated samples arranged with one row per digit, from 0 to 9.

with soft noise significantly outperform harsh noise setting, which corresponds to autoregressive generation. In fact, for small training sets, the harsh noise case uniformly performs better than the other schemes: see Appendix B for details.

MNIST We evaluate the proposed discrete NeurISE diffusion model on the binarized MNIST dataset, which consists of grayscale images of handwritten digits. Images are discretized into a binarized alphabet by thresholding pixel intensities at a fixed midpoint value, assigning pixels to one of two categories depending on whether their intensity lies above or below the threshold. This results in binary-valued vectors in Σ^q, where $q = 784$ denotes the number of pixels in each image. For this benchmark, we allowed a hyperoptimization schedule to search over MLPs of upto 5 layers. We used the MMD metric Gretton et al. (2012) and average cross-correlation of the samples, to compare the performance of the models for conditional sampling task. As can be seen in Table 1a, the NeuRISE based learning of conditionals achieve the lowest MMD and cross-correlation error. Samples of generated images can be seen in Figure 2. Compared to the EA model, we see that D3PM and DFM achieves a much better comparable performance in this setting. Note that the metrics used here do not compute a worst case error between distributions as TV does. This indicates that D3PM and DFM are good at reproducing a lower-order projection of the dataset that aligns with such metrics, but maybe weaker at true distribution learning.

D-Wave Dataset To demonstrate our method on a scientific application with real data, we use the diffusion model to learn a binary dataset produced by D-Wave's Advantage quantum annealer McGeoch & Farré (2020). This dataset is generated by performing repeated quantum annealer runs on the D-Wave machine, with a randomized set of input Hamiltonians. For our experiments we choose $q = 2000$ qubits, which forms a subsection of the annealer and train a diffusion model on the data produced by this portion of the chip. In this example, 8×10^4 samples were used for training, and 2×10^4 samples were reserved for testing. Results in Table 1b again show the advantage of NeurISE Diffusion in all metrics.

Comment on performance of various methods We note that D3PM shows saturation in its error curves, rather than the expected decay with increasing training sample size, for the 5×5 Ising benchmark. On the other hand, it performs well in single-instance MNIST and the D-Wave data sets. This suggests less stable finite-sample scaling under these metrics, possibly because D3PM optimizes a hybrid objective consisting of a variational-bound term and an auxiliary denoising cross-entropy term. While the cross-entropy component encourages accurate denoising, the combined variational nature of the objective might lead to error saturation. Conversely, GGM shows a more regular sample-size trend on the 5×5 Ising benchmark, but its absolute performance is comparably poorer on the higher-dimensional MNIST and D-Wave datasets. We emphasize that these observed features are not a consequence of an unfortunate hyperparameter choice as hyperparameters for all methods have been optimized using *hyperopt* package, see Appendix D for details.

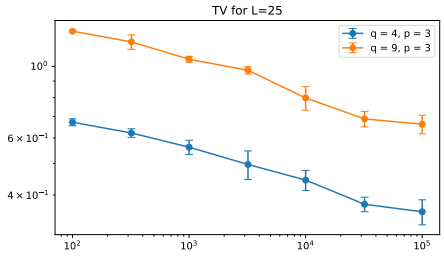
Model	Avg. MMD	Avg. Correlation
Neurise Diff.	1.0e−2	1.5e−6
D3PM	1.5e−2	2.8e−6
SEDD	4.1e−1	7.1e−6
GGM	9.1e−1	8.2e−5
DFM	3.0e−2	3.4e−6

(a) MNIST dataset comparison.

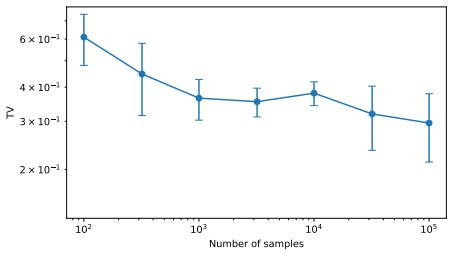
Model	MMD	Avg. Correlation
Neurise Diff.	0.16	1.18e−5
D3PM	0.28	1.81e−5
SEDD	65.03	5.81e−5
GGM	96.4	6.56e−5
DFM	0.22	2.3e−5

(b) D-Wave dataset comparison.

Table 1: Comparison of generative models on MNIST and D-Wave datasets.



(a) Non-binary Potts model.



(b) GHZ state.

Figure 3: Trends for NeuRISE Diffusion as a function of training set size, averaged over 10 trials for each size. The test sample size was 10^5 for each experiment. Error bars represent one standard deviation over trials. Left: TV for a non-binary Potts model. Right: cross-correlation error of NeuRISE Diffusion trained to learn the GHZ state.

5.2 Test case 2: Multi-Alphabet Potts Models

To demonstrate the consistency of our method in the multi-alphabet case, we consider the Potts version of the EA model Subsection 5.1. Let $\Sigma = \{0, 1, \ldots, p-1\}$ and let $\sigma = (\sigma_1, \ldots, \sigma_q) \in \Sigma^q$ with $p = L^2$. We consider a q-state Potts model on an $L \times L$ periodic lattice with Hamiltonian, $H(\sigma) = -\sum_{(i,j)\in E} J_{ij}\, \mathbf{1}\{\sigma_i = \sigma_j\} - \sum_{i=1}^{p} h_{i,\sigma_i},$, where E denotes the set of nearest-neighbor pairs on the lattice. Here $J_{ij} = J_{ji} \in \{-J, +J\}$ are random couplings, $h_{i,s} \in \{-h, +h\}$ are state-dependent local fields, and $\mathbf{1}\{\cdot\}$ is the indicator function. We test the model for two lattices, $L = 2$ and $L = 3$, which corresponds to $q = 4$ and $q = 9$ states, respectively. As can be seen in Figure 3a, the TV error decreases as the number of training samples are increased.

5.3 Test case 3: Quantum Tomography of GHZ state

To test our methods for the multi-alphabet case for a scientifically relevant applications, we use quantum tomography data obtained from the simulation of a four-outcome measurement $(p = 4)$ on the Greenberger–Horne–Zeilinger (GHZ) state. This dataset is commonly used in the study of neural net based approaches to the representation of quantum states Torlai et al. (2018); Jayakumar et al. (2024). We study the efficacy of NeuRISE Diffusion on this model with 20 qubits ($q = 20$) in Figure 3b. The cross-correlation is generalized to the multi-alphabet case as $C_{ij} = \frac{1}{N}\sum_{k=1}^{N}\sum_{a\in\Sigma}\delta_{\sigma_i^{(k)}=a}\delta_{\sigma_j^{(k)}=a}$. We see that the cross-correlation error goes down significantly after 10^4 samples, indicating that the model is able to learn a faithful generative model for this quantum state.

Conclusion

We introduced a discrete diffusion framework that combines round-robin single-site noising with Neural Interaction Screening (NeurISE) to model high-dimensional categorical data. By learning single-site conditional distributions at intermediate diffusion steps, the proposed approach enables an sample efficient reverse denoising process without requiring full joint likelihood estimation. Empirical results on a variety of synthetic and scientific datasets demonstrate that the method effectively captures complex dependency structures in both image-based and physically motivated discrete systems. Our code is provided in the Anonymous Github repository.

AI USE STATEMENT

In this work, we used generative AI tools for help with coding and general editing. We have not used generative AI tools for generating ideas, writing the majority of script, or for the proofs. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

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

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155, 2003.

Daniel Berend and Aryeh Kontorovich. On the convergence of the empirical distribution. *arXiv preprint arXiv:1205.6711*, 2012.

RN Bhatt and AP Young. Numerical studies of ising spin glasses in two, three, and four dimensions. *Physical Review B*, 37(10):5606, 1988.

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

Hongrui Chen and Lexing Ying. Convergence analysis of discrete diffusion model: Exact implementation through uniformization. *arXiv preprint arXiv:2402.08095*, 2024.

Sitan Chen, Sinho Chewi, Jerry Li, Yuanzhi Li, Adil Salim, and Anru R Zhang. Sampling is as easy as learning the score: theory for diffusion models with minimal data assumptions. *arXiv preprint arXiv:2209.11215*, 2022.

Valentin De Bortoli, James Thornton, Jeremy Heng, and Arnaud Doucet. Diffusion schrödinger bridge with applications to score-based generative modeling. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

Samuel Frederick Edwards and Phil W Anderson. Theory of spin glasses. *Journal of Physics F: Metal Physics*, 5(5):965, 1975.

Itai Gat, Tal Remez, Neta Shaul, Felix Kreuk, Ricky TQ Chen, Gabriel Synnaeve, Yossi Adi, and Yaron Lipman. Discrete flow matching. *Advances in Neural Information Processing Systems*, 37: 133345–133385, 2024.

Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The journal of machine learning research*, 13(1):723–773, 2012.

Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. *Advances in neural information processing systems*, 29, 2016.

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

Abhijith Jayakumar, Andrey Lokhov, Sidhant Misra, and Marc Vuffray. Learning of discrete graphical models with neural networks. *Advances in Neural Information Processing Systems*, 33:5610–5620, 2020.

Abhijith Jayakumar, Marc Vuffray, and Andrey Y Lokhov. Learning energy-based representations of quantum many-body states. *Physical Review Research*, 6(3):033201, 2024.

Wengong Jin, Regina Barzilay, and Tommi Jaakkola. Junction tree variational autoencoder for molecular graph generation. In *International conference on machine learning*, pp. 2323–2332. PMLR, 2018.

Tianyi Li, Mingda Chen, Bowei Guo, and Zhiqiang Shen. A survey on diffusion language models. *arXiv preprint arXiv:2508.10875*, 2025.

Yaron Lipman, Marton Havasi, Peter Holderrieth, Neta Shaul, Matt Le, Brian Karrer, Ricky T. Q. Chen, David Lopez-Paz, Heli Ben-Hamu, and Itai Gat. Flow matching guide and code, 2024. URL https://arxiv.org/abs/2412.06264.

Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution. In *International Conference on Machine Learning*, pp. 32819–32848. PMLR, 2024.

Catherine McGeoch and Pau Farré. The D-wave advantage system: An overview. Technical Report 14-1049A-A, D-Wave Systems Inc., Burnaby, BC V5G 4M9, Canada, September 2020. URL https://www.dwavequantum.com/media/s3qbjp3s/14-1049a-a_the_d-wave_advantage_system_an_overview.pdf.

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

Yinuo Ren, Haoxuan Chen, Grant M Rotskoff, and Lexing Ying. How discrete and continuous diffusion meet: Comprehensive analysis of discrete diffusion models via a stochastic integral framework. *arXiv preprint arXiv:2410.03601*, 2024.

Daniel Rudolf, Aaron Smith, and Matias Quiroz. Perturbations of markov chains. *arXiv preprint arXiv:2404.10251*, 2024.

Simo Ryu. Minimal implementation of a d3pm (structured denoising diffusion models in discrete state-spaces), in pytorch. https://github.com/cloneofsimo/d3pm, 2024.

Javier E Santos, Zachary R Fox, Nicholas Lubbers, and Yen Ting Lin. Blackout diffusion: generative diffusion models in discrete-state spaces. In *International Conference on Machine Learning*, pp. 9034–9059. PMLR, 2023.

Haoran Sun, Lijun Yu, Bo Dai, Dale Schuurmans, and Hanjun Dai. Score-based continuous-time discrete diffusion models. *arXiv preprint arXiv:2211.16750*, 2022.

Giacomo Torlai, Guglielmo Mazzola, Juan Carrasquilla, Matthias Troyer, Roger Melko, and Giuseppe Carleo. Neural-network quantum state tomography. *Nature physics*, 14(5):447–450, 2018.

Harshit Varma, Dheeraj Nagaraj, and Karthikeyan Shanmugam. Glauber generative model: Discrete diffusion models via binary classification. *arXiv preprint arXiv:2405.17035*, 2024.

Marc Vuffray, Sidhant Misra, Andrey Lokhov, and Michael Chertkov. Interaction screening: Efficient and sample-optimal learning of ising models. *Advances in neural information processing systems*, 29, 2016.

Zikun Zhang, Zixiang Chen, and Quanquan Gu. Convergence of score-based discrete diffusion models: A discrete-time analysis. *arXiv preprint arXiv:2410.02321*, 2024.

A Pseudcode for NeurISE Diffusion

In this section, we present the Neurise based denoising diffusion algorithm introduced in the paper.

B Robustness Study for NeurISE Diffusion

B.1 Soft Noise vs Harsh Noise

In this section, we show our comparison of the harsh noise setting with soft noise ones, to facilitate comparison of autoregressive vs diffusion model. From our observation, the harsh noise setting performs better at low number of training samples and for large number of training samples, the models show very similar performance. See Figure 4. Here, N denotes the total number of time-steps used in the noising and denosing phase, and e denotes the noise parameter ε.

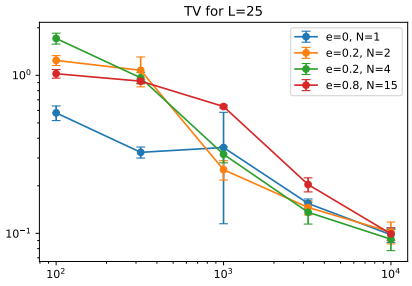
Algorithm 1 Discrete Diffusion with NeurISE

Input: alphabet Σ with $|\Sigma| = p$, dimension q, steps T , noise $\varepsilon \in [0, 1]$, data distribution μ_0 (samples , $\sigma_0 \sim \mu_0$), Number of samples N.

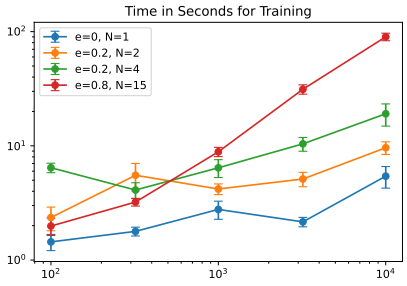
Forward diffusion

Sample time index $t \sim \mathrm{Unif}(\{1, 2, \ldots, T\})$
Initialize $\sigma \leftarrow \sigma_0$
for $n = 1, 2, \ldots, N$ **do**
  $u \leftarrow ((t-1) \bmod q) + 1$
  Set $\sigma_{-u} \leftarrow \sigma_{-u}$
  With probability $1 - \varepsilon$ change coordinate u according $\sigma_u \sim \mathrm{Unif}(\Sigma)$
end for
Output forward tuple (t, σ_0, σ_t) where $\sigma_t \leftarrow \sigma$
Learn conditionals with NeurISE.
Goal: estimate single-site conditionals $\hat{\mu}_s(\cdot \mid \sigma_{s,-u})$ for $s = 0, \ldots, T-1$
for $s = 0, 1, \ldots, T-1$ **do**
  Generate noised samples at time s by running the forward kernel on data to obtain a batch $\{\sigma_s^{(n)}\}_{n=1}^N$
  for $u = 1, 2, \ldots, q$ **do**
    Train NeurISE network NN_θ by minimizing the NeurISE objective equation 11
    Obtain conditional estimator $\hat{\mu}_s(\cdot \mid \sigma_{s,-u})$ via equation 12
  end for
end for

Reverse sampling (denoising).
Initialize $\tilde{\sigma}_T \sim \mathrm{Unif}(\Sigma^q)$
for $r = T-1, T-2, \ldots, 0$ **do**
  $u \leftarrow (r \bmod q) + 1$
  Set $\hat{k}_n^{rev}$ according to equation 5-equation 6
  Sample $\tilde{\sigma}_{T-1} \sim \hat{k}_n^{rev}(\sigma_T, \cdot)$
end for
return $\tilde{\sigma}_0$



(a) TV



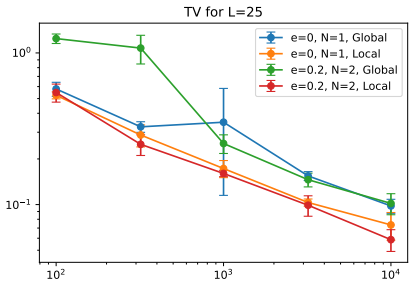
(b) Training Time

Figure 4: Trend of TV and Training time for Ising models, averaged 10 trials per data set. The test sample size was taken to be 10^4 for each experiment. The harsh noise version of the problem performs competitively with situations where noise is soft.

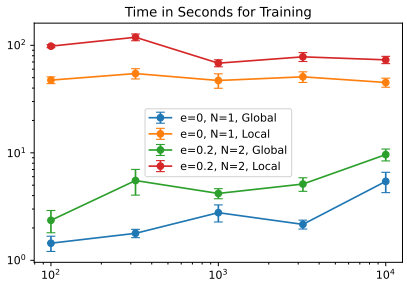
B.2 Local vs Global Neural Network

In this section, we compare the performance of diffusion models trained using two architectures: (i) a collection of local neural networks, with one network per time step, and (ii) a single global neural network shared across the entire time horizon. As shown in Fig. 5, the local architecture consistently outperforms the global one, despite both approaches having approximately the same total number of trainable parameters. Specifically, in the local setting, each time step is modeled by a single-hidden-layer MLP with 5 hidden units, whereas the global model uses a fixed network of width 125.

We tested NeuRISE Diffusion for two different lattices: 2×2 and 3×3, each with alphabet size 3. As can be seen Figure 3a, the TV decreases in a statistically expected way as the number of training samples is increased from 10^2 to 10^5.



(a) TV



(b) Training Time

Figure 5: Trend of TV and Training time for Ising models, averaged 10 trials per data set. The test sample size was taken to be 10^4 for each experiment. Learning individual conditionals using separate neural networks achieves lower TV, than using one global neural network across all conditionals.

C Theory

For the presentation in this section, we introduce some additional notation. Given a probability distribution μ on Σ^q, the action of the reverse kernel k_n^{rev} on μ is defined by

$$(\mu k_n^{\text{rev}})(\tilde{\sigma}) = \sum_{\sigma \in \Sigma^q} \mu(\sigma)\, k_n^{\text{rev}}(\sigma, \tilde{\sigma}), \qquad \tilde{\sigma} \in \Sigma^q.$$

For two kernels k_n^{rev} and k_{n-1}^{rev}, their composition is the kernel

$$(k_n^{\mathrm{rev}} k_{n-1}^{\mathrm{rev}})(\sigma, \sigma') = \sum_{\tilde{\sigma} \in \Sigma^q} k_n^{\mathrm{rev}}(\sigma, \tilde{\sigma})\, k_{n-1}^{\mathrm{rev}}(\tilde{\sigma}, \sigma'),$$

which corresponds to applying k_n^{rev} first and k_{n-1}^{rev} second.

More generally, the composed reverse kernel

$$\mathcal{K}_{0:T-1}^{\mathrm{rev}} = k_{T-1}^{\mathrm{rev}} \cdots k_0^{\mathrm{rev}}$$

satisfies

$$(\mathcal{K}_{0:T-1}^{\mathrm{rev}})(\sigma_T, \sigma_0) = \sum_{\sigma_1, \ldots, \sigma_{T-1} \in \Sigma^q} \prod_{t=0}^{T-1} k_t^{\mathrm{rev}}(\sigma_{t+1}, \sigma_t),$$

where σ_T denotes the state at time T and σ_0 the state at time 0.

Finally, the distribution obtained by initializing the reverse chain from μ_{noise} is

$$\begin{aligned} \nu_0(\sigma_0) &= (\mu_{\mathrm{noise}} \mathcal{K}_{0:T-1}^{\mathrm{rev}})(\sigma_0) \\ &= \sum_{\sigma_T \in \Sigma^q} \mu_{\mathrm{noise}}(\sigma_T)(\mathcal{K}_{0:T-1}^{\mathrm{rev}})(\sigma_T, \sigma_0). \end{aligned}$$

The following result provides analogue of the convergence guarantees established for score-based diffusion models in the continuous time De Bortoli et al. (2021); Chen et al. (2022); Chen & Ying (2024); Zhang et al. (2024); Ren et al. (2024). Notably, the analysis is much simpler than in De Bortoli et al. (2021); Chen et al. (2022); Chen & Ying (2024); Zhang et al. (2024); Ren et al. (2024) due to the discrete time setting. The key insight is that the discrepancy between the output distribution of the approximate reverse chain and the true data distribution decomposes cleanly into two contributions: (i) the extent to which the forward process has mixed toward the noise distribution, and (ii) the cumulative error incurred when approximating the reverse kernels. This decomposition makes precise the tradeoff underlying DDPM-style generative modeling. Accurate sampling requires both sufficiently fast diffusion of the forward process to the noise distribution and sufficiently accurate estimation of the reverse-time dynamics. Here, the total variation (TV) distance $\|\cdot\|_{\mathrm{TV}}$ between two distributions is defined by $\|\hat{\mu} - \mu\|_{\mathrm{TV}} = \frac{1}{2} \sum_{\sigma \in \Sigma^q} |\hat{\mu}(\sigma) - \mu(\sigma)|$.

Theorem C.1. *Let $\{X_n\}_{n=0}^T$ be the Markov chain on Σ^q with forward transition kernels $k_n : \Sigma^q \times \Sigma^q \to \mathbb{R}_{\geq 0}$. Fix a noise reference distribution μ_{noise} on Σ^q and assume that for some $\delta_T \in [0, 1]$,*

$$\|\mu_T - \mu_{\mathrm{noise}}\|_{\mathrm{TV}} \leq \delta_T. \tag{13}$$

Let $\{k_n^{\mathrm{rev}}\}_{n=0}^{T-1}$ be a well-defined family of reverse kernels that satisfy equation 3. Consider approximate reverse kernels $\{\widehat{k}_n^{\mathrm{rev}}\}_{n=0}^{T-1}$ such that for all $n = 0, \ldots, T-1$,

$$\sup_{\sigma \in \Sigma^q} \left\| \widehat{k}_n^{\mathrm{rev}}(\cdot, \sigma) - k_n^{\mathrm{rev}}(\cdot, \sigma) \right\|_{\mathrm{TV}} \leq \eta. \tag{14}$$

Initialize the approximate reverse chain with the noise reference, i.e. $Y_T \sim \mu_{\mathrm{noise}}$, and let $\widehat{\mu}_0$ denote the law of the output Y_0 obtained by applying $\widehat{k}_{T-1}^{\mathrm{rev}}, \ldots, \widehat{k}_0^{\mathrm{rev}}$.

Then the output distribution satisfies

$$\|\hat{\mu}_0 - \mu_0\|_{\mathrm{TV}} \;\leq\; \underbrace{\delta_T}_{\textit{Mixing error}} \;+\; \underbrace{T\,\eta.}_{\textit{Reverse kernel estimation error}} \tag{15}$$

Proof. Let $\mathcal{K}_{0:T-1}^{\mathrm{rev}}$ denote the composition of the exact reverse kernels $k_{T-1}^{\mathrm{rev}}, \ldots, k_0^{\mathrm{rev}}$ (applied in this order), and let $\widehat{\mathcal{K}}_{0:T-1}^{\mathrm{rev}}$ denote the composition of the approximate reverse kernels $\widehat{k}_{T-1}^{\mathrm{rev}}, \ldots, \widehat{k}_0^{\mathrm{rev}}$.

Let ν_0 be the law of the output obtained by running the *exact* reverse chain initialized at time T from μ_{noise}, i.e.

$$\nu_0 := \mu_{\mathrm{noise}}\, \mathcal{K}_{0:T-1}^{\mathrm{rev}}.$$

Since the kernels $\{k_n^{\mathrm{rev}}\}$ satisfy equation 3, initializing the exact reverse chain from μ_T yields μ_0. Hence,

$$\mu_0 = \mu_T \, \mathcal{K}_{0:T-1}^{\mathrm{rev}}.$$

By the data processing inequality

$$\|\nu_0 - \mu_0\|_{\mathrm{TV}} \leq \|\mu_{\mathrm{noise}} - \mu_T\|_{\mathrm{TV}} \leq \delta_T. \tag{IE}$$

Let $\widehat{\mu}_0$ be the law of the output of the *approximate* reverse chain initialized from μ_{noise}, i.e.

$$\widehat{\mu}_0 := \mu_{\mathrm{noise}} \, \widehat{\mathcal{K}}_{0:T-1}^{\mathrm{rev}}.$$

We bound $\|\widehat{\mu}_0 - \nu_0\|_{\mathrm{TV}}$ by a telescoping argument as is used in pertubation theory of Markov chains Rudolf et al. (2024). Define intermediate distributions for $m = 0, 1, \ldots, T$:

$$\rho^{(m)} := \mu_{\mathrm{noise}} \, \widehat{k}_{T-1}^{\mathrm{rev}} \cdots \widehat{k}_{T-m}^{\mathrm{rev}} \, k_{T-m-1}^{\mathrm{rev}} \cdots k_0^{\mathrm{rev}},$$

with the convention that $\rho^{(0)} = \nu_0$ and $\rho^{(T)} = \widehat{\mu}_0$. Then by the triangle inequality,

$$\|\widehat{\mu}_0 - \nu_0\|_{\mathrm{TV}} = \|\rho^{(T)} - \rho^{(0)}\|_{\mathrm{TV}} \leq \sum_{m=1}^{T} \|\rho^{(m)} - \rho^{(m-1)}\|_{\mathrm{TV}}.$$

Fix $m \in \{1, \ldots, T\}$ and set

$$\alpha^{(m)} := \mu_{\mathrm{noise}} \, \widehat{k}_{T-1}^{\mathrm{rev}} \cdots \widehat{k}_{T-m+1}^{\mathrm{rev}},$$

so that $\rho^{(m)} = \alpha^{(m)} \widehat{k}_{T-m}^{\mathrm{rev}} k_{T-m-1}^{\mathrm{rev}} \cdots k_0^{\mathrm{rev}}$ and $\rho^{(m-1)} = \alpha^{(m)} k_{T-m}^{\mathrm{rev}} k_{T-m-1}^{\mathrm{rev}} \cdots k_0^{\mathrm{rev}}$. Using contraction of TV under a common kernel,

$$\|\rho^{(m)} - \rho^{(m-1)}\|_{\mathrm{TV}} \leq \left\| \alpha^{(m)} \widehat{k}_{T-m}^{\mathrm{rev}} - \alpha^{(m)} k_{T-m}^{\mathrm{rev}} \right\|_{\mathrm{TV}}.$$

For any distribution α and kernels P, Q on Σ^q,

$$\|\alpha P - \alpha Q\|_{\mathrm{TV}} \leq \sup_{\sigma \in \Sigma^q} \|P(\cdot, \sigma) - Q(\cdot, \sigma)\|_{\mathrm{TV}}.$$

Applying this with $\alpha = \alpha^{(m)}$, $P = \widehat{k}_{T-m}^{\mathrm{rev}}$, $Q = k_{T-m}^{\mathrm{rev}}$ and the estimation error, we get,

$$\|\rho^{(m)} - \rho^{(m-1)}\|_{\mathrm{TV}} \leq \eta. \tag{KE}$$

Combining gives

$$\|\widehat{\mu}_0 - \nu_0\|_{\mathrm{TV}} \leq T\eta.$$

Finally, by the triangle inequality,

$$\begin{aligned} \|\widehat{\mu}_0 - \mu_0\|_{\mathrm{TV}} &\leq \|\widehat{\mu}_0 - \nu_0\|_{\mathrm{TV}} + \|\nu_0 - \mu_0\|_{\mathrm{TV}} \\ &\leq T\eta + \delta_T, \end{aligned}$$

which concludes the proof. □

In the following Theorem we bound the error of the approximate reverse chain when initialized from the true noise distribution μ_{noise}. In practice, however, the reverse process is initialized from an empirical approximation $\frac{1}{N} \sum_{i=1}^{N} \delta_{X_i^{\mathrm{data}}}$ of μ_{noise}. The following corollary shows that this additional source of error contributes additively to the final bound and captures the effect of sampling error from the noise distribution. It partially explains why masked diffusion models have been observed to perform better Austin et al. (2021); Santos et al. (2023); Lou et al. (2024) in practice, than when the noise distribution is uniform.

Corollary C.2. *(Initialization error) In the setting of Theorem C.1, let $\widehat{\mu}_{\mathrm{noise}}$ be any distribution on Σ^q such that for some $\gamma \in [0, 1]$,*

$$\left\| \widehat{\mu}_{\mathrm{noise}} - \mu_{\mathrm{noise}} \right\|_{\mathrm{TV}} \leq \gamma. \tag{16}$$

Initialize the approximate reverse chain with $Y_T \sim \widehat{\mu}_{\mathrm{noise}}$, and let $\widetilde{\mu}_0$ denote the law of the output Y_0 obtained by applying $\widehat{k}_{T-1}^{\mathrm{rev}}, \ldots, \widehat{k}_0^{\mathrm{rev}}$.

Then the output distribution satisfies

$$\left\|\widetilde{\mu}_0 - \mu_0\right\|_{\mathrm{TV}} \leq \underbrace{\delta_T}_{\textit{Mixing Error}} + \underbrace{T\eta}_{\textit{Reverse kernel estimation error}} + \underbrace{\gamma\cdot}_{\textit{Noise sampling error}} \tag{17}$$

Proof. Let $\widehat{\mathcal{K}}^{\mathrm{rev}}_{0:T-1}$ denote the composition of the approximate reverse kernels. Define

$$\widehat{\mu}_0 := \mu_{\mathrm{noise}}\,\widehat{\mathcal{K}}^{\mathrm{rev}}_{0:T-1}, \qquad \widetilde{\mu}_0 := \widehat{\mu}_{\mathrm{noise}}\,\widehat{\mathcal{K}}^{\mathrm{rev}}_{0:T-1}.$$

The TV norm under the action of a Markov kernel Q remains preserved (this follows trivially from $\sum_{\sigma\in\Sigma^q} Q(\sigma,\tilde{\sigma}) = 1$) and hence,

$$\begin{aligned}\left\|\widetilde{\mu}_0 - \widehat{\mu}_0\right\|_{\mathrm{TV}} &= \left\|(\widehat{\mu}_{\mathrm{noise}} - \mu_{\mathrm{noise}})\widehat{\mathcal{K}}^{\mathrm{rev}}_{0:T-1}\right\|_{\mathrm{TV}} \\ &\leq \left\|\widehat{\mu}_{\mathrm{noise}} - \mu_{\mathrm{noise}}\right\|_{\mathrm{TV}} \leq \gamma.\end{aligned}$$

The claim follows by the triangle inequality together with Theorem C.1, which gives $\|\widehat{\mu}_0 - \mu_0\|_{\mathrm{TV}} \leq \delta_T + T\eta$. □

One can use this corollary to see the effect of error due to sampling from the noise distribution. For instance, let $\widehat{\mu}_{\mathrm{noise}} = \frac{1}{N}\sum_{i=1}^{N}\delta_{X_i^{\mathrm{noise}}}$ be the approximating empirical distribution based on N i.i.d. samples $\{X_1^{\mathrm{noise}}, ...X_N^{\mathrm{noise}}\}$ from μ_{noise}. Then from results of (Berend & Kontorovich, 2012) one can quantify the effect of sampling error from the noise distribution, on the distance of the sampled distribution from the data distribution. In the special case, when $\mu^{\mathrm{noise}} = \delta_{\sigma_{\mathrm{mask}}}$ for some $\sigma_{\mathrm{mask}} \in \Sigma^q$, then it is to see that one can in fact, take a stronger bound by setting $\gamma = 0$. While this might explain partially why diffusion models with absorbing states perform better as observed in literature Austin et al. (2021); Santos et al. (2023); Lou et al. (2024), it can be that the estimation error of the reversal kernel, as captured by η is high in such situations, as the distribution becomes much more concentrated. On the other hand, in experiments, we observed uniform distribution performed better. We conjecture this is due to this noising process increasing the temperature of the distribution and due to fact that higher-temperature distributions being easier to learn via NeurISE Jayakumar et al. (2020).

C.1 Non-uniqueness of Reverse Processes

In this section, we highlight that, in general, the reverse process associated with a forward Markov chain is not unique. Even when the marginal distributions μ_t at each time $t \in \{0,\ldots,T\}$ are fixed, there may exist multiple valid reverse dynamics that recover the same marginals.

Let $\{X_t\}_{t=0}^T$ be a forward Markov process over Σ^q, with $X_t \sim \mu_t$ for each t. The canonical construction of the reverse process introduced in the main text uses Bayes' rule:

$$\begin{aligned}\mathbb{P}(X_t = \sigma \mid X_{t+1} = \tilde{\sigma}) &= \frac{\mathbb{P}(X_t = \sigma, X_{t+1} = \tilde{\sigma})}{\mathbb{P}(X_{t+1} = \tilde{\sigma})} \\ &= \frac{\mathbb{P}(X_{t+1} = \tilde{\sigma} \mid X_t = \sigma)\cdot\mathbb{P}(X_t = \sigma)}{\mathbb{P}(X_{t+1} = \tilde{\sigma})}.\end{aligned}$$

This defines a valid reverse kernel based on the forward transition probabilities and the marginal distributions μ_t. However, other reverse processes may exist that yield the same marginals.

Let $\{Y_t\}_{t=0}^T$ be another sequence of random variables over Σ^q such that:

$$\mathbb{P}(Y_t = \sigma) = \mathbb{P}(X_t = \sigma) = \mu_t(\sigma) \quad \text{for all } t \in \{0,\ldots,T\}. \tag{18}$$

Then Y_t is a valid alternative reverse process if it satisfies the marginal constraints above.

As an illustrative example, consider the extreme case where the reverse kernel is marginally independent of the conditioning variable:

$$\mathbb{P}(Y_t = \sigma \mid Y_{t+1} = \tilde{\sigma}) = \mu_t(\sigma). \tag{19}$$

In other words, the reverse step simply resamples from the marginal μ_t, ignoring the previous state Y_{t+1}.

Now fix $Y_T = X_T \sim \mu_T$, and define Y_t recursively using (2). Then we show by induction that:

$$\begin{aligned}\mathbb{P}(Y_t = \sigma) &= \sum_{\hat{\sigma} \in \Sigma^q} \mathbb{P}(Y_t = \sigma \mid Y_{t+1} = \hat{\sigma}) \cdot \mathbb{P}(Y_{t+1} = \hat{\sigma}) \\ &= \sum_{\hat{\sigma} \in \Sigma^q} \mu_t(\sigma) \cdot \mu_{t+1}(\hat{\sigma}) \\ &= \mu_t(\sigma) \cdot \sum_{\hat{\sigma} \in \Sigma^q} \mu_{t+1}(\hat{\sigma}) \\ &= \mu_t(\sigma),\end{aligned}$$

since μ_{t+1} is a probability distribution and thus sums to 1.

In this degenerate case, the reverse kernel $k_t^{\text{rev}} : \Sigma^q \times \Sigma^q \to \mathbb{R}_{\geq 0}$ is defined by:

$$k_t^{\text{rev}}(\sigma, \tilde{\sigma}) := \mathbb{P}(Y_t = \sigma \mid Y_{t+1} = \tilde{\sigma}) = \mu_t(\sigma) \quad \forall \tilde{\sigma} \in \Sigma^q. \tag{20}$$

That is, $k_t^{\text{rev}}(\cdot, \tilde{\sigma})$ is simply the marginal distribution μ_t, regardless of the value of $\tilde{\sigma}$. This reverse kernel completely ignores the conditioning state and independently resamples $\sigma \sim \mu_t$ at each step.

While this kernel does not capture the time-reversal of the actual forward dynamics, it still guarantees the correct marginal distributions at every time step:

$$\begin{aligned}\mu_t(\sigma) &= \sum_{\tilde{\sigma} \in \Sigma^q} k_t^{\text{rev}}(\sigma, \tilde{\sigma}) \cdot \mu_{t+1}(\tilde{\sigma}) \\ &= \mu_t(\sigma) \cdot \sum_{\tilde{\sigma} \in \Sigma^q} \mu_{t+1}(\tilde{\sigma}) = \mu_t(\sigma).\end{aligned}$$

This construction shows that the reverse process is not uniquely determined by the marginal sequence $\{\mu_t\}_{t=0}^T$, and highlights a family of reverse dynamics that can be arbitrarily different from the canonical reverse Markov process.

In fact, the set of all admissible reverse kernels that satisfy the marginal condition is convex; any convex combination of two valid reverse kernels $k_{t,1}^{\text{rev}}$ and $k_{t,2}^{\text{rev}}$ also yields a valid reverse kernel,

$$k_t^{\text{rev}} = \lambda k_{t,1}^{\text{rev}} + (1 - \lambda) k_{t,2}^{\text{rev}}, \quad \text{for any } \lambda \in [0, 1].$$

This further underscores the flexibility and ambiguity inherent in defining reverse-time dynamics.

It is important to emphasize that the transitions defined by general reverse kernels are *not local*. Unlike the canonical reverse process where transitions are typically constrained to move between configurations that differ by a single spin (i.e., Hamming distance one) this degenerate reverse kernel allows transitions between any two configurations in Σ^q, regardless of their Hamming distance:

$$k_t^{\text{rev}}(\sigma, \tilde{\sigma}) = \mu_t(\sigma) \quad \text{for all } \sigma, \tilde{\sigma} \in \Sigma^q.$$

In other words, starting from any configuration $\tilde{\sigma}$, the reverse process can jump to any other configuration $\sigma \in \Sigma^q$ in a single step, with probability determined solely by the marginal $\mu_t(\sigma)$. There is no notion of continuity or neighborhood preserved by the dynamics. This contrasts sharply with reverse processes, where transitions are typically limited to configurations that differ by only one coordinate.

Thus, while the degenerate reverse process is mathematically valid and correctly reproduces the marginal distributions μ_t, it does not preserve the locality structure of the forward process. Its ability to transition freely between any two configurations in Σ^q , without regard for neighborhood structure, leads to a reverse kernel that is inherently non-local. In high-dimensional spaces, such non-local kernels operate over the entire $\Sigma^q \times \Sigma^q$ transition space, making them exponentially more complex to represent, learn, or approximate. On the other hand, kernels for local update rules scale linearly and generalize more easily.

D Numerical Implementation Details

D.1 Model Architecture

We use the following architecture for each of the cases :

- Input block: $\mathrm{Linear}(d_{\text{in}} \to h) \to \mathrm{LayerNorm}(h) \to \mathrm{SiLU}$, where d_{in} is dependent on the denoising algorithm.
- Hidden blocks (up to $D-1$ blocks, depending on depth $D \in \{1, \ldots, 5\}$): $\mathrm{Linear}(h \to h) \to \mathrm{LayerNorm}(h) \to \mathrm{SiLU}$.
- Output layer: $\mathrm{Linear}(h \to 2)$.

D.2 Shared hyperparameter sweep

For every dataset–denoising algorithm combination, we run a small hyperparameter optimization loop over the following parameter using the *hyperopt package* in Python:

- **Depth:** $D \in \{1, .., 5\}$
- **Width:** $h \in \{64, 128, 256, 512\}$
- **Noise parameter** (Only for NeurISE Diffusion): $\varepsilon \in (0, 1)$
- **Noising time horizon (Only for NeurISE Diffusion):** $T \in [0, 2, ...10]$
- **Learning rate:** $\mathrm{lr} \in (10^{-4}, 5 \times 10^{-2})$ with log uniform distribution.
- **Weight decay:** $w \in (10^{-8}, 10^{-3}$ with log uniform distribution.
- **Batch Size:** $[64, 128, 256, 512]$

D.3 SEDD Implementation details

In the implementation of SEDD Lou et al. (2024), we introduced a final layer that enforced positivity in the output of the layer for the score approximation. While it is claimed that the loss function introduced in Lou et al. (2024) naturally forces the output of the network towards non-negativity, we did not observe this in our implementation, and in fact found that the training algorithm returned NaNs if the final layer was not appropriately augmented. Additionally, the inputs were required to be one-hot coded entirely in order for algorithm to show any significant learning. In contrast, for the Neurise Diffusion, GGM, D3PM and DFM only conditioning parameters were required to be one-hot coded.