Convergence Analysis of Discrete Diffusion Model: Exact Implementation through Uniformization

Hongrui Chen [*][1] and Lexing Ying [†][1]

[1]Department of Mathematics, Stanford University, Stanford, CA 94305, USA.

Abstract. Diffusion models have achieved huge empirical success in data generation tasks. Recently, some efforts have been made to adapt the framework of diffusion models to discrete state space, providing a more natural approach for modeling intrinsically discrete data, such as language and graphs. This is achieved by formulating both the forward noising process and the corresponding reversed process as continuous time Markov chains. In this paper, we investigate the theoretical properties of the discrete diffusion model. Specifically, we introduce an algorithm leveraging the uniformization of continuous Markov chains, implementing transitions on random time points. Under reasonable assumptions on the learning of the discrete score function, we derive total variation distance and Kullback–Leibler divergence guarantees for sampling from any distribution on a hypercube. Our results align with state-of-the-art achievements for diffusion models in $\mathbb{R}^d$ and further underscore the advantages of discrete diffusion models in comparison to the $\mathbb{R}^d$ setting.

1 Introduction

Generative modeling is one of the central tasks in machine learning, which aims to learn a probability distribution from data and generate data from the learned distribution. The diffusion model has emerged as a powerful and versatile framework in generative modeling, achieving state-of-the-art performance in a variety of data generation tasks, including image generation [3, 28], audio generation [33], video generation [17, 43], text-to-image synthesis [30, 31], and computational biology [15]. The general framework of the score-based generative model involves:

1) defining a forward noising process to gradually diffuse the data distribution to some simple distribution (like standard Gaussian);

2) learning a reversed process to denoising the simple distribution to the data distribution by estimating the score functions of the forward diffusion process.

Works on the diffusion model focus on the forward processes defined in the Euclidean state space $\mathbb{R}^d$. In such scenarios, an ideal choice of the forward process is the Ornstein-Uhlenbeck (OU) process, which is driven by a stochastic differential equation (SDE) on $\mathbb{R}^d$,

and the corresponding reversed process is also given by an SDE. Nevertheless, certain data generation tasks present an intrinsic characteristic of discrete data. For example, natural language processing operates within a discrete token space, computer vision involves discrete representations of images, and molecular graph modeling engages with graph data in a discrete structure [18,19,44]. Thus, it is more natural to use diffusion processes on the discrete state space to model these discrete data distributions.

To this end, some recent works [5,7,25,27,32] have introduced a framework for diffusion models in discrete state spaces. This framework notably utilizes a continuous-time Markov chain (CTMC) in the discrete state space for the forward process, and the corresponding reverse process is also a CTMC. Moreover, mirroring the concept of score estimation in diffusion models on $\mathbb{R}^d$, they proposed a discrete score function given by the ratios of probability mass on different states, and the score entropy loss as a new score matching objective that is derived from Kullback–Leibler (KL) divergence divergence between the path measures of the forward and the reversed process. Combining the learning of the discrete score function through minimizing the score entropy and the sampling from the learned reversed process, a completed procedure for the diffusion model on discrete state space has been established.

However, despite the potential advantage of the discrete diffusion model, unlike the extensively studied SDE framework, the theoretical understanding of the CTMC framework has not been built. A line of works [4,8,10,21–23] concentrated on the theory of diffusion model on $\mathbb{R}^d$. Generally speaking, the established theoretical results can be summarized as follows:

- Sampling is as easy as learning the score: for arbitrary data distribution, suppose one can estimate the score function at multiple noise levels, then one can approximately sample from the data distribution.

- Quantitatively, under an L^2 accurate score estimator on the forward process, $\mathcal{O}(d\log(1/\delta)/\epsilon^2)$ iterations suffices to output a distribution that is ϵ^2-close in KL divergence to a distribution p_δ, where p_δ is a variance-δ Gaussian perturbation of the data distribution.

- There are three sources of error in the diffusion model: 1) the error from the inexact score estimator, 2) the error from insufficient mixing of the forward process, and 3) the discretization error. The discretization error causes the key challenges in the analysis due to the error propagation in the numerical simulation of a non-contractive dynamic.

In this paper, we take a step toward the theory of diffusion model in the CTMC framework and aim to understand how the theoretical property of discrete diffusion compares to the established theory for diffusion model on $\mathbb{R}^d$. Our results suggest that:

- One can implement the reversed CTMC in an exact way, i.e. without discretization error, through an algorithm based on the uniformization technique [12,14,40]. This presents a surprising advantage of the CTMC framework compared to the SDE framework, where discrete errors are significant in the analysis.

- The proposed algorithm is guaranteed to be efficient by our theoretical analysis. Quantitatively, assuming an ϵ-error bound on the score entropy loss, our purposed algorithm requires $\mathcal{O}(d \log(d/\epsilon\delta))$ steps to reach a distribution ϵ-close in KL divergence to a distribution p_δ, where p_δ is a δ-perturbation of the data distribution in TV distance. Combining the two errors, we also get a TV distance guarantee for sampling from the data distribution. Thus, we obtain a logarithm dependence on ϵ for the discrete diffusion model, in contrast to the ϵ^{-2} dependence for the SDE framework.

Organization of this paper. In Section 2, we introduce some preliminary background on CTMC and the uniformization technique for CTMC. In Section 3, we introduce the framework of the diffusion model using CTMC on discrete state space. In Section 4, we propose our new algorithm for numerical simulation of the reversed CTMC process and analyze the complexity and error of the proposed algorithm.

1.1   Related works

Diffusion models on discrete state space.   In the initial stages of development, the diffusion model was formulated as a discrete-time Markov process. It was first introduced in [35], marking the pioneering work in applying the diffusion process for data generation tasks. Although the focus has primarily been on continuous-space formulations, the discrete-time and discrete-space framework of the diffusion model was also described in the initial work [35] and then further explored in [2, 18, 34]. Some works, such as [29, 36], tackle the discrete data by embedding it to $\mathbb{R}^d$ and employing the continuous-space diffusion model framework with dequantization techniques. This approach has also been popular in other generative models like variational autoencoders (VAE) and auto-regressive models.

A significant breakthrough in diffusion models was achieved when Song *et al.* [36, 38] introduced a continuous framework through stochastic differential equations. As a discrete analogy of the SDE framework, [7] firstly established the CTMC framework for diffusion models on discrete state space. This paper introduces an evidence lower bound objective (ELBO) derived from the KL divergence between path measures and proposes multiple numerical methods for simulating the reversed process, including τ-leaping and prediction-correction. [27] proposed the concrete score matching method to learn the probability ratio term in the reversed CTMC. However, this objective does not align with the path-KL and might lack stability in practice. [39] alternatively derive the score-matching objective from the conditional marginal distributions. [32] proposed a different, forward process called blackout diffusion, which transforms data to an empty set instead of uniform distribution. [25, 44] applies the continuous-time discrete diffusion model to generate language data and, in particular, [25] scale the method to GPT2 data, demonstrating its potential effectiveness in handling large datasets.

Theory of diffusion models.   This paper is closely related to a series of works [4, 8, 10, 13, 21–23] focused on the theoretical analysis of diffusion models in $\mathbb{R}^d$. Specifically, these studies seek to answer the following question: given an L^2-accurate score estimator, how

closely does the distribution generated by the reverse SDE with the score estimator, in place of the actual score function, and with appropriate discretization, approximate the data distribution? This question was first addressed for smooth and isoperimetric distributions in [21], followed by a reduction of the isoperimetry assumption in [10] and the smoothness assumption in [8,22]. The state-of-art result, which is applicable to any distribution and shows a nearly linear dependence on the dimension d, is provided in [4]. In this paper, we answer this question for the discrete diffusion model. Our results match the state-of-the-art theory in [4] for the $\mathbb{R}^d$ setting, applying to any distribution on the hypercube and exhibiting a nearly linear dependence on d.

[7] also provides an error bound for the discrete diffusion model. However, this analysis relies on some strong assumptions like the L^∞ accuracy of the score estimator and the bounded probability ratio of the data distribution. In addition, their result also has a suboptimal quadratic dependence on the dimension d. In this paper, we will reduce all these strong assumptions and provide a nearly optimal bound that exhibits a linear dependence on d up to a logarithmic factor. We are aware that uniformization appeared in the proof of [7]. However, this work is the first to deploy uniformization as a working algorithm for discrete diffusion models and prove its efficiency.

In addition, some works focus on other aspects of the theoretical understanding of diffusion models. For example, [9,23] analyze the (approximately) deterministic algorithm for the reversed sampling of diffusion model, [16,24] studies the sample complexity of learning the score function.

2   Preliminaries on continuous-time Markov chain

Let $\mathcal{X}$ be a finite state space with $|\mathcal{X}| = N$. A CTMC $(X_t)_{t\geq0}$ on $\mathcal{X}$ is a continuous stochastic process that satisfies the Markov property. In this process, state changes occur based on an exponential clock associated with each state, with transitions to different states determined by the corresponding transition rates.

Mathematically, a continuous-time Markov chain $(X_t)_{t\geq0}$ on $\mathcal{X}$ is characterized by its infinitesimal generator $Q(t) \in \mathbb{R}^{N\times N}$ with

$$Q_{x,x}(t) = -\sum_{y\neq x} Q_{x,y}(t).$$

Let $P_{x,y}(s,t)$ denote the transition kernel of (X_t) from time s to time t, i.e.

$$\mathbb{P}(X_t = y | X_s = x) = P_{x,y}(s,t).$$

The infinitesimal transitions of the process are determined by the generator through

$$P(t,t+h) = I + hQ(t) + o(h).$$

This leads to the Kolmogorov forward equation

$$\frac{\mathrm{d}P(s,t)}{\mathrm{d}t} = P(s,t)Q(t).$$

In particular, for time-homogeneous cases $Q(t) \equiv Q$, we have $P(s,t) = e^{(t-s)Q}$.

To understand how the process (X_t) can be constructed, let us start with the simple case that Q is a time-homogeneous generator $Q(t) \equiv Q$. For each $x, y \in \mathcal{X}$, $Q_{x,y} \geq 0$ can be viewed as the transition rate from the state x to the state y and $-Q_{x,x} = \sum_y Q_{x,y}$ is the intensity of the exponential clock for the state x. Starting from a state $x \in \mathcal{X}$, the process transitions after a holding time determined by an $\mathrm{Exp}(Q_{x,x})$ random variable, with the transition probabilities defined as

$$\mathbb{P}(X_t = y | X_{t-} = x) = -\frac{Q_{x,y}}{Q_{x,x}}, \quad \forall y \neq x.$$

The time-inhomogeneous case is slightly more complicated. Intuitively, it can be understood as a time-homogeneous CTMC with a generator $Q(t)$ at each infinitesimal interval $[t, t + dt]$. So, a natural approach for numerical simulation involves discretizing the process and simulating the time-homogeneous CTMC within each distinct time interval. However, more sophisticated methods exist for simulating a time-inhomogeneous CTMC without the need for discretization. This method is known as uniformization. It decouples the clocks and transition mechanisms to a single Poisson point process, and a set of Markov transition kernels $(\tilde{P}(t))_{t \geq 0}$. The intensity of the Poisson point process λ uniformly bounds all clock intensities Q_{xx}, and the transition kernels are defined by

$$\tilde{P}_{x,y}(t) = \begin{cases} \dfrac{Q_{x,y}(t)}{\lambda}, & y \neq x, \\ 1 - \displaystyle\sum_{y \neq x} \dfrac{Q_{x,y}(t)}{\lambda}, & y = x \end{cases} \tag{2.1}$$

or in a matrix representation $\tilde{P}(t) = I + Q(t)/\lambda$. Simulating the CTMC involves changing the state according to $\tilde{P}(t)$ whenever a transition occurs in the Poisson point process. Formally, we have the following proposition.

Proposition 2.1 (Uniformization of CTMC). *Consider a CTMC (X_t) on a finite state space $\mathcal{X}$ with the generator $Q(t)$. Let $p(t)$ be the distribution of the CTMC at time t. Suppose $Q_{x,x}(t) \leq \lambda$ for any $x \in \mathcal{X}$ and $0 \leq t \leq T$. Let $(\tilde{P}(t))_{t \geq 0}$ be the transition kernels given by (2.1). Let $\tau_1 < \tau_2 < \cdots < \tau_n$ be the transition times within $[0, T]$ of a Poisson point process with intensity λ, or equivalently, n is drawn from $\mathrm{Poisson}(\lambda)$ and $\tau_1, \cdots, \tau_n$ is sorted i.i.d. samples from $\mathrm{Unif}([0, T])$. The distribution of X_T conditioning on the number of transition n by time T and the transition times $\tau_1, \tau_2, \cdots, \tau_n$ is given by $X_T | (n, \tau_1, \cdots, \tau_n) \sim p(0)\tilde{P}(\tau_1)\tilde{P}(\tau_2) \cdots \tilde{P}(\tau_n)$.*

In the time-homogeneous setting $Q(t) \equiv Q$, Proposition 2.1 can be simply deduced through a Taylor expansion

$$p(t) = p(0)e^{Qt} = p(0) \sum_{n=0}^{\infty} \tilde{P}^n \frac{(\lambda t)^n}{n!} e^{-\lambda t},$$

implying the transition of the CTMC e^{Qt} can be executed by applying the transition $\tilde{P}$ for $\mathrm{Poisson}(\lambda t)$ times. For a general setting, an intuitive way to understand the uniformization involves approximating the generator $Q(t)$ by a piece-wisely constant function. The

results of uniformization in the homogeneous setting can be easily adapted to cases with piece-wise constant $Q(t)$ by the Markov property. Since any general function $Q(t)$ can be approximated by piece-wise constant function to arbitrary precision, uniformization thus provides an exact simulation of the time-inhomogeneous CTMC process. A rigorous proof of this can be found in [41].

In practice, however, simulating CTMC through uniformization may entail substantial computational costs, as the number of necessary transitions is contingent upon the uniform bound imposed on all the clocks. This renders the computation resource-intensive, especially in scenarios involving stiff problems where $Q_{x,x}(t)$ dramatically changes across different states x and different times t. Nevertheless, we will demonstrate that, when applying this method to the discrete diffusion model, we can obtain a provable efficient algorithm by adaptively selecting the upper bound λ across different time intervals.

3 Framework of discrete diffusion model

3.1 General procedure

The forward and reversed CTMC. Let $p(0)$ be a data distribution on a $\mathcal{X}$. Consider a forward process $(X_t)_{0 \le t \le T}$ defined by a CTMC with a generator $Q(t)$ starting from $X_0 \sim p(0)$. The distribution of the forward process at time t is denoted by $p(t)$.

As an analogy of the reversed SDE used in diffusion model on $\mathbb{R}^d$ [1, 37], we construct a CTMC $(X_t^{\leftarrow})_{0 \le t \le T}$ as a time reversal of (X_t), meaning $(X_t^{\leftarrow})_{0 \le t \le T}$ is equivalent to $(X_{T-t})_{0 \le t \le T}$ in distribution. As discussed in [5, 7, 25], this time reversal can be achieved by a CTMC starting from $X_0^{\leftarrow} \sim p(T)$ and governed by the generator

$$Q_{x,y}^{\leftarrow}(t) := Q_{y,x}(t)\frac{p_y(t)}{p_x(t)}. \tag{3.1}$$

One can sample from the data distribution $p(0)$ if it is possible to simulate the reversed process $(X_t^{\leftarrow})$. However, the ratio term $p_y(t)/p_x(t)$ in the generator (3.1) is not available. This term is referred to as the discrete score function. The idea is to parameterize the discrete score function within a function class, such as neural networks, and learn it from training data.

Training objectives for learning the discrete score. We denote the discrete score function as $c_{x,y}(t) := p_y(t)/p_x(t)$ and $s_{x,y}(t)$ represents an estimation of $c_{x,y}(t)$ used for sampling. Consider the sampling dynamic $(\hat{X}_t^{\leftarrow})_{0 \le t \le T}$ that is a CTMC with the generator $\hat{Q}_{x,y}^{\leftarrow}(t) := Q_{y,x}(t)s_{x,y}(t)$, initiating from a simple distribution $\gamma \approx p(T)$. The following proposition gives an expression of the KL divergence between the true reversed process $(X_t^{\leftarrow})$ and the sampling dynamic $(\hat{X}_t^{\leftarrow})$.

Proposition 3.1. *Let $\mathbb{P}^{\leftarrow}$ and $\hat{\mathbb{P}}^{\leftarrow}$ be the path measure of $(X_t^{\leftarrow})$ and $(\hat{X}_t^{\leftarrow})$, respectively. We have*

$$\mathrm{KL}(\mathbb{P}^{\leftarrow}\|\hat{\mathbb{P}}^{\leftarrow}) = \mathrm{KL}(p(T)\|\gamma) + \int_0^T \mathbb{E}_{X_t \sim p(t)}\ell(c_{X_t}, s_{X_t})\mathrm{d}t, \tag{3.2}$$

where

$$\ell(c_x, s_x) = \sum_{y \neq x} Q_{y,x} \left(-c_{x,y}(t) + s_{x,y}(t) + c_{x,y}(t) \log \frac{c_{x,y}(t)}{s_{x,y}(t)} \right).$$

This is an analogy of the Girsanov theorem for the SDE framework. We defer an intuitive derivation to Appendix A.1 and refer to [5, Section A] for rigorous proof. Note that the first term in the right-hand side of (3.2) is fully characterized by the convergence of the forward process; thus, our focus should primarily be on the second term for learning the score function. Omitting the terms that are independent of the score estimator s, we obtain the implicit score entropy [5, 25]

$$\mathcal{L}_{\text{ISE}} = \int_0^T \mathbb{E}_{X_t \sim p(t)} \left[\sum_{Y \neq X_t} \left(Q_{Y,X_t} s_{X_t,Y}(t) - Q_{X_t,Y} \log s_{Y,X_t}(t) \right) \right] dt. \tag{3.3}$$

This objective (3.3) can be optimized by substituting the expectation with an empirical mean over samples.

For enhanced computational efficiency, a variant of implicit score entropy called the denoising score entropy is proposed [5, 25]

$$\mathcal{L}_{\text{DSE}} = \int_0^T \mathbb{E}_{X_0 \sim p(0), X_t \sim p(t)} \left[\sum_{Y \neq X_t} Q_{Y,X_t} \left(s_{X_t,Y}(t) - \frac{P_{X_0,Y}(0,t)}{P_{X_0,X_t}(0,t)} \log s_{X_t,Y}(t) \right) \right] dt, \tag{3.4}$$

where $P_{x,y}(s,t)$ is the transition kernel of the forward process (X_t). Note that the denoising score entropy includes a term from the transition kernel of the forward process, implying that the design of the forward process should facilitate explicit computation of this kernel. One example is the independent flips on hypercube, which is the main focus of this paper and will be introduced in Section 3.2.

Note that the implicit score entropy and the denoising score entropy are analogous to the implicit score matching objective [20] and the denoising score matching objective [42], respectively, in the setting of a diffusion model on $\mathbb{R}^d$.

3.2 Independent flips on hypercube as forward process

In this section, and throughout the remainder of this paper, we concentrate on a scenario where the state space is the d-dimensional hypercube $\mathcal{X} = \{0, 1\}^d$ and the forward process is given by independent flips on each component. This process is formally described as a CTMC with the generator

$$Q_{x,y}(t) = \begin{cases} 1, & d(x,y) = 1, \\ -d, & y = x, \\ 0, & \text{otherwise,} \end{cases} \tag{3.5}$$

where $d(x,y)$ denotes the Hamming distance between x and y. This process has several nice properties that are critical for the design and analysis of our algorithm in Section 4.

Explicit formula for the transition kernel. The special structure of the independent flipping process allows us to write the transition kernel explicitly by the following proposition.

Proposition 3.2. *Let $P_{x,y}(s,t)$ be the transition kernel of the CTMC with the generator Q given in (3.5). Let $g_w(t)$ be the discrete heat kernel*

$$g_w(t) = \frac{1}{2^d} \prod_{i=1}^{d} \left(1 + (-1)^{w_i} e^{-2t}\right), \quad \forall w \in \{0,1\}^d.$$

Then we have $P_{x,y}(s,t) = g_{y-x}(t-s)$, where the $y-x$ should be understood in modulo 2 sense.

Proof. By utilizing the tensor-product structure of Q, we can write the generator Q as $Q = \sum_{i=1}^{d} Q_i$, where Q_i represents a tensor product of d matrices of size 2×2. In this tensor product, the matrix

$$A := \begin{pmatrix} -1 & 1 \\ 1 & -1 \end{pmatrix}$$

occupies the i-th position, while the remaining positions are all identity. By the Kolmogorov forward equation, we have

$$P(s,t) = e^{(t-s)Q} = \left(e^{(t-s)A}\right)^{\otimes d} = \begin{pmatrix} \dfrac{1 + e^{-2(t-s)}}{2} & \dfrac{1 - e^{-2(t-s)}}{2} \\ \dfrac{1 - e^{-2(t-s)}}{2} & \dfrac{1 + e^{-2(t-s)}}{2} \end{pmatrix}^{\otimes d},$$

giving the expression $P_{x,y}(s,t) = g_{y-x}(t-s)$. □

This explicit formula of the transition kernel makes the denoising score entropy (3.4) tractable and will play an important role in our algorithm proposed in Section 4 for simulating the reversed process.

Convergence of the forward process. In the context of the diffusion model on $\mathbb{R}^d$, the OU forward process is known to converge exponentially to standard Gaussian, regardless of the complexity of the data distribution [8, 10]. Similarly, in our discrete setting, the independent flipping process exhibits an exponential convergence to the uniform distribution over $\{0,1\}^d$. We write this formally in the following proposition.

Proposition 3.3. *Let $p(t)$ be the distribution of the forward process with the generator Q given in (3.5) at time t. Let γ denote the uniform distribution over $\{0,1\}^d$. We have*

$$\mathrm{KL}(p(T)\|\gamma) \leq e^{-T}\mathrm{KL}(p(0)\|\gamma) \lesssim de^{-T}.$$

The proof is deferred to Appendix A.2. Proposition 3.3 suggests that to make the error from the insufficient mixing of the forward process less than ϵ, i.e. $\mathrm{KL}(p_T\|\gamma) \leq \epsilon$, we only need to simulate the forward process for time $T = \log(d/\epsilon)$, which depends on d and $1/\epsilon$ logarithmly.

The sparse structure. Since transitions only occur between neighbors on the hypercube each time, we can use a more concise way to represent the score function

$$c_x(t) = \left[\frac{p_{x+e_1}}{p_x}, \frac{p_{x+e_2}}{p_x}, \cdots, \frac{p_{x+e_d}}{p_x} \right]^\top \in \mathbb{R}^d.$$

Here, e_i denotes the vector with a 1 in the i-th coordinate and 0's elsewhere, and the addition operator is defined modulo 2. Similarly, we use $s_x(t)$ to denote the score estimator.

Note that for a state space of size 2^d, the discrete score function, defined by the ratio of probabilities, is generally a function $\{0,1\}^d \times \{0,1\}^d \to \mathbb{R}$. However, by leveraging the sparse structure of the transitions, we can simplify this to a $\{0,1\}^d \times d \to \mathbb{R}$ function. This simplification enables more efficient computation when learning the discrete score function. Furthermore, as we will discuss in Section 4, this sparse structure greatly facilitates the algorithmic implementation of the reversed process.

4   Main results

In this section, we present our algorithm and the analysis for implementing the sampling dynamic, which is a reversed process corresponding to the independent flipping forward process, with the discrete score function replaced by the score estimator. Because of (3.5), the sampling dynamic is given by a CTMC with the generator

$$\hat{Q}^{\leftarrow}_{x,y}(t) = \begin{cases} s_x(t)_i, & y = x + e_i, \\ -\sum_{i=1}^{d} s_x(t)_i, & y = x, \\ 0, & \text{otherwise.} \end{cases} \tag{4.1}$$

4.1   Assumptions

To begin with, we introduce some assumptions related to the learning of the discrete score function. Firstly, we assume the score estimator is ϵ-accurate in the score entropy loss.

Assumption 4.1. For some $\epsilon, \delta > 0$, the score estimator s satisfies

$$\mathcal{L}(s) = \frac{1}{T-\delta} \int_\delta^T \mathbb{E}_{X_t \sim p(t)} \ell\big(c_{X_t}(t), s_{X_t}(t)\big) \mathrm{d}t \leq \epsilon,$$

where $\ell : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is the Bregman distance with respect to the entropy function $h(x) = \sum_{i=1}^{d} x_i \log x_i$ given by

$$\ell(c,s) = c - s - \langle \nabla h(s), s - c \rangle = \sum_{i=1}^{d} \left(-c_i + s_i + c_i \log \frac{c_i}{s_i} \right).$$

Assumption 4.1 is an analogy of the L^2-accuracy assumption that is widely used in the theoretical works for the diffusion models in the SDE framework [4, 8, 10, 21–23]. There are, however, two primary distinctions between Assumption 4.1 and the corresponding

assumption for diffusion models on $\mathbb{R}^d$. First, in our discrete setting, the L^2 distance is replaced by the Bregman distance. This choice aligns with the structure of the CTMC framework, where the KL divergence between path measures is characterized by the Bregman distance (see Proposition 3.1) rather than the L^2 distance for the SDE framework. Second, we assume the score estimation error to be small on average over the time interval $[\delta, T]$, in contrast to the assumption in continuous diffusion models, which assume a small average error over specific discretization points. This difference arises because our algorithm employs the score estimator at randomly sampled times rather than at a fixed set of discretization points (see Algorithm 1 below). Both assumptions are reasonable because one can sample times either uniformly or from a discrete set during the stochastic gradient descent in the training process.

It is important to notice that the term $\mathcal{L}(s)$ in Assumption 4.1 is equivalent to (up to a term independent of s) the objective functions used in the learning procedure, including the implicit score entropy and the denoising score entropy discussed in Section 3.1. Thus, Assumption 4.1 is satisfied if the objective function is optimized to the extent that its function value is ϵ close to the minimum value.

The second assumption involves the uniform boundedness of the score estimator s. This leads to a bounded transition rate in the sampling dynamics, thereby enabling the algorithmic application of the uniformization technique. This assumption relies on the observation that the probability ratio of the forward process (or the true discrete score function) is uniformly bounded.

Proposition 4.1. *Let $p(t)$ be the distribution of the forward CTMC with generator Q given in (3.5). For $t > 0$, the probability ratio between two neighbor states is bounded by*

$$\frac{p_{x+e_i}(t)}{p_x(t)} \lesssim \max(1/t, 1), \quad \forall x \in \{0,1\}^d, \quad i \in \{1,\ldots,d\}.$$

Proof. Let g be the heat kernel defined in Proposition 3.2. Consider the conditional distribution $\tilde{p}(a|x) \propto p_a(0)g_{x-a}(t)$ that is the distribution of X_t conditional on X_0. Then we write the probability ratio as (in the following computation, all the addition is defined modulo 2)

$$
\begin{aligned}
\frac{p_{x+e_i}(t)}{p_x(t)} &= \frac{\sum_{a+w=x+e_i} p_a(0)g_w(t)}{\sum_{a+w=x} p_a(0)g_w(t)} \\
&= \frac{\sum_{a+w=x} p_a(0)g_{w+e_i}(t)}{\sum_{a+w=x} p_a(0)g_w(t)} \\
&= \frac{\sum_{a+w=x} p_a(0)\big(g_{w+e_i}(t)/g_w(t)\big)g_w(t)}{\sum_{a+w=x} p_a(0)g_w(t)}.
\end{aligned}
\tag{4.2}
$$

By the definition of g, we can compute the ratio

$$\frac{g_{w+e_i}(t)}{g_w(t)} = \frac{1-(-1)^{w_i}e^{-2t}}{1+(-1)^{w_i}e^{-2t}}.$$

Substitute this to (4.2), we obtain

$$
\frac{p_{x+e_i}(t)}{p_x(t)} = \frac{\sum_{a+w=x} p_a(0)\big((1-(-1)^{w_i}e^{-2t})/(1+(-1)^{w_i}e^{-2t})\big)g_w(t)}{\sum_{a+w=x} p_a(0)g_w(t)}
$$

$$
= \frac{\sum_a p_a(0)\big((1-(-1)^{x_i-a_i}e^{-2t})/(1+(-1)^{x_i-a_i}e^{-2t})\big)g_{x-a}(t)}{\sum_a p_a(0)g_{x-a}(t)}
$$

$$
= \mathbb{E}_{\tilde{p}(a|x)}\left[\frac{1-(-1)^{x_i-a_i}e^{-2t}}{1+(-1)^{x_i-a_i}e^{-2t}}\right] \lesssim \frac{1+e^{-2t}}{1-e^{-2t}}.
$$

Since $(1+e^{-2t})/(1-e^{-2t}) \lesssim \max(1,1/t)$, we complete the proof. $\square$

Note the bound in Proposition 4.1 is independent of the data distribution; thus, we do not need any assumption on the structure of data distribution. Based on Proposition 4.1, it is natural to assume the score estimator is also uniformly bounded.

Assumption 4.2. We assume there is a universal constant C such that

$$
\sum_{i=1}^d s_x(t)_i \leq Cd\max(1,1/t).
$$

In practice, the constraint in Assumption 4.2 can be imposed by slightly modifying the learned score function after training. For example, one can add a sigmoid layer before the output for the score estimator to satisfy the desired bound. This modification will not affect the accuracy of the learned score function according to Proposition 4.1.

4.2 Algorithm

We provide an algorithm, detailed in Algorithm 1 that exactly simulates the sampling dynamic. Our algorithm is based on the uniformization technique stated in Proposition 2.1. Note Proposition 2.1 requires the uniform boundness condition for the transition rates, which is not satisfied by the generator $\hat{Q}^{\leftarrow}$ in general. To address this, our algorithm implements early stopping with a terminal time of $T-\delta$. Then Assumption 4.2 ensures the bound. Moreover, since the bound in Assumption 4.2 varies over time, we apply the procedure from Proposition 2.1 with adaptive λ. Specifically, we introduce a partition $0 = t_0 < t_1 < \cdots < t_N = T-\delta$ and set different λ_k values in different time intervals. Combining all these ingredients leads to the formulation of Algorithm 1. The algorithm outputs a sample from the distribution $\hat{p}^{\leftarrow}(T-\delta)$, where $\hat{p}^{\leftarrow}(t)$ represents the distribution of the CTMC with the generator defined in (4.1) at time t.

4.3 Analysis of the algorithm

In this section, we provide a theoretical analysis of Algorithm 1 on its complexity and error. In particular, we are interested in the following two aspects of Algorithm 1:

- The algorithm complexity. In other words, how many steps are required to implement Algorithm 1?

- The error of the resulting distribution, i.e., the distance between the data distribution $p(0)$ and the distribution $\hat{p}^{\leftarrow}(T - \delta)$ obtained by Algorithm 1.

Algorithm 1 Exact Aalgorithm for Implementing the Sampling Dynamic.

Input: A learned score function s that satisfies Assumption 4.2 with a constant C, total time T, a time partition $0 = t_0 < t_1 < \cdots < t_N = T - \delta$, parameters $\lambda_1, \lambda_2, \cdots, \lambda_N$.
Output: A sample from $\hat{p}^{\leftarrow}(T - \delta)$.

Draw $Y_0 \sim \text{Unif}(\{0, 1\}^d)$.
for $k = 0$ **to** $N - 1$ **do**
　Draw $M \sim \text{Poisson}(\lambda_{k+1}(t_{k+1} - t_k))$.
　Sample M points i.i.d. from $\text{Unif}([t_k, t_{k+1}])$ and sort them as $\tau_1 < \tau_2 \cdots < \tau_M$.
　Set $Z_0 = Y_k$.
　for $j = 0$ **to** $M - 1$ **do**

$$\text{Set } Z_{j+1} = \begin{cases} Z_j + e_i, & \text{w.p.} \quad \dfrac{s_{Z_j}(\tau_j)_i}{\lambda_{k+1}}, \quad 1 \le i \le d, \\[2ex] Z_j, & \text{w.p.} \quad 1 - \displaystyle\sum_{i=1}^{d} \dfrac{s_{Z_j}(\tau_j)_i}{\lambda_{k+1}}. \end{cases}$$

　end for
　Set $Y_{k+1} = Z_M$.
end for
return Y_N.

Results for general data distribution. Firstly, we consider the setting that $p(0)$ is a general data distribution. We summarize the result in the following theorem.

Theorem 4.1. *Suppose Assumption* 4.2 *holds. By choosing the time partition such that*

$$t_{k+1} - t_k \le c(T - t_{k+1}), \quad \forall 0 \le k \le N - 1 \tag{4.3}$$

for some absolute constant c and choosing λ_k's by

$$\lambda_k = \frac{Cd}{\min(1, T - t_k)},$$

the implementation of Algorithm 1 *requires $M \sim \text{Poisson}(\lambda)$ steps with*

$$\lambda = \mathcal{O}\big(d\big(\log(1/\delta) + T\big)\big).$$

Moreover, if we further assume Assumption 4.1, *Algorithm* 1 *outputs a distribution $\hat{p}^{\leftarrow}(T - \delta)$ such that*

1. *The KL divergence between $\hat{p}^{\leftarrow}(T - \delta)$ and $p(\delta)$ is bounded by*

$$\mathrm{KL}\big(p(\delta)\|\hat{p}^{\leftarrow}(T - \delta)\big) \lesssim de^{-T} + T\epsilon.$$

   Choosing $T \asymp \log(d/\epsilon)$ makes this $\tilde{\mathcal{O}}(\epsilon)$ and $\lambda = \mathcal{O}(d\log(d/\epsilon\delta))$.

2. *The TV distance between $\hat{p}^{\leftarrow}(T - \delta)$ and the data distribution is bounded by*

$$\mathrm{TV}\big(p(0), \hat{p}^{\leftarrow}(T - \delta)\big) \lesssim \sqrt{de^{-T} + T\epsilon} + (1 - e^{-d\delta}).$$

   Choosing $T \asymp \log(d/\epsilon)$, $\delta \asymp \sqrt{\epsilon}/d$ makes the error $\tilde{\mathcal{O}}(\sqrt{\epsilon})$, and $\lambda = \mathcal{O}(d\log(d/\epsilon^{3/4}))$.

Remark 4.1. Note that Poisson distribution satisfies super-exponential decaying tails, so $\mathcal{O}(d\log(1/\lambda))$ steps are enough with high probability and $\lambda = \mathcal{O}(d\log(d/\epsilon\delta))$ or $\lambda = \mathcal{O}(d\log(d/\epsilon^{3/4}))$ describes the complexity of the algorithm.

Remark 4.2. The choice of the time partition (4.3) is quite flexible. This is because we only care about the total number of transitions rather than the number of intervals N.

The proof is deferred to Appendix A.3. Our nearly linear dependence on d and logarithm dependence on δ match the best result for the continuous diffusion model [4].

Results for data distribution with uniformly bounded score. In addition, if we further assume the discrete score function of the data distribution (i.e. the probability ratio between neighbor states) is uniformly bounded, we can simulate the sampling dynamic without early stopping (i.e. $\delta = 0$). In this case, we can improve the TV distance bound between the data distribution and the sampled distribution in Theorem 4.1(2) to a KL divergence bound.

Assumption 4.3. Suppose the data distribution $p(0)$ satisfies

$$\frac{p_{x+e_i}(0)}{p_x(0)} \leq L, \quad \forall\, x \in \{0,1\}^d, \quad i \in \{1,\ldots,d\}.$$

Theorem 4.2. *Under Assumptions 4.1-4.3, let $\delta = 0$, $T \asymp \log(d/\epsilon)$, by choosing the time partition $0 = t_1 < \cdots < t_N = T$ and parameter λ_k's appropriately in Algorithm 1, one can obtain a distribution $\hat{p}^{\leftarrow}(T)$ that is $\tilde{\mathcal{O}}(\epsilon)$ close to the data distribution within $\mathrm{Poisson}(\mathcal{O}(d\log(dL/\epsilon)))$ steps.*

The proof is deferred to Appendix A.4. Notably, the algorithm complexity only has a logarithm dependence on L, so we can get a reasonable guarantee even if L has an exponential dependence on d.

Discussion on the near optimality. Note that for the true reversed process, the expectation value of the intensity of each clock is given by

$$\mathbb{E}\sum_{i=1}^{d}\frac{p_{x+e_i}(t)}{p_x(t)} = \sum_{x\in\{0,1\}^d}\sum_{i=1}^{d}p_{x+e_i}(t) = \sum_{i=1}^{d}\sum_{x\in\{0,1\}^d}p_{x+e_i}(t) = d.$$

Consequently, the expected number of transitions within a time interval of length 1 is $\Theta(d)$. As a result, a linear dependence on d is unavoidable for simulating the reversed process.

Comparison with $\{0, 1, \ldots, n\}^d$ setting. [7, 25] consider the forward following process on $\{1, \ldots, n\}^d$ with the generator Q: For two distinct states $x, y \in \{0, 1, \ldots, n\}^d$, define $Q_{x,y} = 1$ if and only if x and y differ in exactly one entry. In this case, the typical number of transitions that occur within a unit time interval is $\Theta(n^2 d)$ and the convergence of the forward process requires at least $\Omega(1)$ time. So $\Omega(n^2 d)$ steps are required to simulate the reversed process. However, if we transform the data to the $\{0, 1\}^{d \log(n)}$ structure and utilize our hypercube framework, the n, d dependence of the algorithm complexity is reduced to $\tilde{\mathcal{O}}(d \log(n))$. Therefore, our hypercube framework provides a more efficient implementation of the discrete diffusion model.

Discussion on the generalization to general forward process. In practice, forward processes beyond independent flipping are often employed, such as the absorbing process (see, for example, [25, Section 3.3]). Our analysis has the potential to generalize to discrete diffusion models with general forward processes. Recall that the key ingredients in our analysis include: (1) a log-Sobolev inequality bound for the forward process, which ensures the rapid convergence of the forward process in KL divergence (Proposition 3.3); and (2) a uniform bound on the true score function (Proposition 4.1), which guarantees the uniform boundedness of the approximately accurate score estimator in the sampling dynamics (Assumption 4.2) and makes the uniformization-based algorithm practical. For a CTMC forward process with a general Q-matrix, one may establish the convergence results by proving these two ingredients for the given process.

5 Conclusion

In this paper, we consider an algorithm for discrete diffusion models based on uniformization and present the first theoretical analysis of the discrete diffusion model. Although our nearly linear dependence result aligns with the state-of-art result for diffusion models on $\mathbb{R}^d$ and is nearly optimal in the current framework, there are several interesting further directions to explore.

Faster algorithm with theoretical guarantee. Our algorithm provides an exact simulation of the reversed process, where the number of transitions corresponds to the worst-case bound of the clock intensities. Although we believe that the $\Omega(d)$ complexity is not improvable in general, there may be potential in investigating an approach that simulates transitions adaptive to the clock of the current state. This might require some discretization and further analysis to quantify the discretization error.

Improve the graph structure of the forward process. We consider the independent flipping process on the hypercube as the forward process. This process converges to the uniform distribution in $O(\log d)$ time and results in a reversed process that transitions $\Omega(d)$ times. A natural question is if one can employ a better structure for the design of the forward process so that the forward process still converges exponentially but the number of

transitions times is reduced. One idea is to apply the forward process on the Ramanujan graph [26], but the numerical simulation will become hard.

Appendix A   Omitted proof

A.1   An intuitive proof of Proposition 3.1

Proof. Let $\tilde{X}_t^{\leftarrow}$ be a CTMC with the estimated generator $\hat{Q}^{\leftarrow}$ starting from $\tilde{X}_0^{\leftarrow} \sim p(T)$. Let $\tilde{\mathbb{P}}^{\leftarrow}$ be the path measure of $(\tilde{X}_t^{\leftarrow})$. By the chain rule of KL divergence

$$\mathrm{KL}(\mathbb{P}^{\leftarrow}\|\hat{\mathbb{P}}^{\leftarrow}) = \mathrm{KL}\big(p(T)\|\gamma\big) + \mathrm{KL}(\mathbb{P}^{\leftarrow}\|\tilde{\mathbb{P}}^{\leftarrow}).$$

Now we give an intuitive computation of $\mathrm{KL}(\mathbb{P}^{\leftarrow}\|\tilde{\mathbb{P}}^{\leftarrow})$. Consider a path γ from time $t = 0$ to T (corresponding to the time of the reversed process $(X_t^{\leftarrow})_{0 \le t \le T}$). Let us first compute the probability ratio $\mathbb{P}^{\leftarrow}(\gamma)/\tilde{\mathbb{P}}^{\leftarrow}(\gamma)$. We discretize $\gamma(t)$ with step size ϵ as $x_0, x_1, \dots, x_L$, where $x_j = \gamma(\epsilon j)$ and $L = T/\epsilon$. $\mathbb{P}^{\leftarrow}(\gamma)/\tilde{\mathbb{P}}^{\leftarrow}(\gamma)$ is a product of multiple terms $\prod_i (\mathbb{P}^{\leftarrow}(x_{i+1}|x_i)/\tilde{\mathbb{P}}^{\leftarrow}(x_{i+1}|x_i))$:

- When $x_i = x_{i+1}$, the ratio is $(1 - \sum_{y \ne x_i} Q_{y,x_i} c_{x_i,y}\epsilon)/(1 - \sum_{y \ne x_i} Q_{y,x_i} s_{x_i,y}\epsilon)$.

- When $x_i \ne x_{i+1}$, the ratio is $(Q_{x_{i+1},x_i} c_{x_i,x_{i+1}}\epsilon)/(Q_{x_{i+1},x_i} s_{x_i,x_{i+1}}\epsilon) = c_{x_i,x_{i+1}}/s_{x_i,x_{i+1}}$.

Taking the product and log gives

$$\log \frac{\mathrm{d}\mathbb{P}^{\leftarrow}(\gamma)}{\mathrm{d}\tilde{\mathbb{P}}^{\leftarrow}(\gamma)} \approx \sum_{i:\text{no jump}} \left(\sum_{y \ne x_i} Q_{y,x_i}(-c_{x_i,y} + s_{x_i,y}) \right) \epsilon + \sum_{i:\text{jump}} \sum_{y \ne x_i} \log \frac{c_{x_i,y}}{s_{x_i,y}} \delta_{x_i \to y},$$

where $\delta_{x_i \to y}$ is equal to 1 if x_i jumps to y otherwise 0.

The KL divergence is then the expectation of this quantity with respect to $\mathbb{P}^{\leftarrow}$. To simplify, we use the fact that $\mathbb{E}\delta_{x_i \to y} = Q_{y,x_i} c_{x_i,y}\epsilon$ for $y \ne x_i$ and take the limit $\epsilon \to 0$ to obtain

$$\mathbb{E}_{\mathbb{P}^{\leftarrow}} \log \frac{\mathrm{d}\mathbb{P}^{\leftarrow}(\gamma)}{\mathrm{d}\tilde{\mathbb{P}}^{\leftarrow}(\gamma)}$$

$$= \mathbb{E}_{\mathbb{P}^{\leftarrow}} \left[\int_0^T \sum_{y \ne X_t^{\leftarrow}} Q_{y,X_t^{\leftarrow}}(T - t) \right.$$

$$\left. \times \left(-c_{X_t^{\leftarrow},y}(T - t) + s_{X_t^{\leftarrow},y}(T - t) + c_{X_t^{\leftarrow},y} \log \frac{c_{X_t^{\leftarrow},y}(T - t)}{s_{X_t^{\leftarrow},y}(T - t)} \right) \mathrm{d}t \right].$$

Finally, by reversing the time, we have

$$\mathbb{E}_{\mathbb{P}^{\leftarrow}} \log \frac{\mathrm{d}\mathbb{P}^{\leftarrow}(\gamma)}{\mathrm{d}\tilde{\mathbb{P}}^{\leftarrow}(\gamma)} = \mathbb{E}_{\mathbb{P}} \left[\int_0^T \sum_{y \ne X_t} Q_{y,X_t}(t) \left(-c_{X_t,y}(t) + s_{X_t,y}(t) + c_{X_t,y} \log \frac{c_{X_t,y}(t)}{s_{X_t,y}(t)} \right) \mathrm{d}t \right],$$

which completes the proof. $\square$

A.2 Proof of Proposition 3.3

Proof. For the first inequality, it is shown in [6, Theorem 5.1] that the uniform distribution over the hypercube satisfies the log-Sobolev inequality with constant 2 (with respect to the Markov semigroup associated with the generator Q). This implies the exponential mixing of the forward process in KL divergence (see, for example, [11])

$$\mathrm{KL}\big(p(T)\|\gamma\big) \le e^{-T}\mathrm{KL}\big(p(0)\|\gamma\big).$$

The second inequality is because the KL divergence between p_0 and the uniform distribution can be decomposed to

$$\begin{aligned}
\mathrm{KL}(p_0\|\gamma) &= \sum_{x\in\{0,1\}^d} p_x(0)\log p_x(0) - \sum_{x\in\{0,1\}^d} p_x(0)\log(1/2^d)\\
&\lesssim \sum_{x\in\{0,1\}^d} p_x(0)\log p_x(0) + d.
\end{aligned} \tag{A.1}$$

For distributions on a finite set, the maximum entropy is achieved by the uniform distribution, so the entropy term in (A.1) could be bounded by the entropy of the uniform distribution, which is d (up to a constant). $\qquad\square$

A.3 Proof of Theorem 4.1

Proof of the bound of λ.   In Algorithm 1, since the number of steps in each time interval $[t_k, t_{k+1}]$ is sampled from $M_k \sim \mathrm{Poisson}(\lambda_k(t_k - t_{k-1}))$, the total number of steps follows a Poisson distribution with parameter:

$$\lambda := \sum_{k=1}^{N} \lambda_k(t_k - t_{k-1}).$$

Now we evaluate μ. Recall that

$$\lambda_k = \frac{Cd}{\min(1, T - t_k)}.$$

Let $s_k = T - t_k$. Recall that

$$\lambda_k = \frac{Cd}{\min(1, s_k)}, \quad s_k - s_{k+1} \le cs_{k+1}.$$

For $\delta \le s_k \le 1$ we have

$$\begin{aligned}
\sum_{k:\delta \le s_k < 1} \lambda_k(s_{k-1} - s_k) &= \sum_{k:s_k < 1} \frac{Cd}{s_k}(s_{k-1} - s_k)\\
&\lesssim \sum_{k:s_k < 1} \frac{C(c+1)d}{s_{k-1}}(s_{k-1} - s_k)\\
&\lesssim \int_\delta^1 \frac{d}{s}\mathrm{d}s = d\log(1/\delta).
\end{aligned}$$

For $s_k \geq 1$, we have

$$\sum_{k:1\leq s_k\leq T} \lambda_k(s_{k-1} - s_k) = \sum_{k:1\leq s_k\leq T} Cd(s_k - s_{k-1}) = Cd(T-1) \lesssim dT.$$

Combining the two parts, we conclude that

$$\begin{aligned} \lambda &= \sum_{k=1}^{N} \lambda_k(t_k - t_{k-1}) = \sum_{k=1}^{N} \lambda_k(s_k - s_{k-1}) \\ &= \sum_{k:\delta\leq s_k<1} \lambda_k(s_{k-1} - s_k) + \sum_{k:1\leq s_k\leq T} \lambda_k(s_{k-1} - s_k) \\ &\lesssim d\big(T + \log(1/\delta)\big). \end{aligned}$$

Proof of the KL divergence bound. Since our algorithm exactly simulates the reversed process, from Proposition 3.1, the KL divergence between $p(\delta)$ and $\hat{p}^{\leftarrow}(T - \delta)$ is bounded by the KL divergence between the two path measures

$$\begin{aligned} &\mathrm{KL}\big(p(\delta)\|\hat{p}^{\leftarrow}(T-\delta)\big) \\ &\leq \mathrm{KL}\big(p(T)\|\mathrm{Unif}(\{0,1\}^d)\big) \\ &\quad + \int_{\delta}^{T} \mathbb{E}_{X_t\sim p(t)}\ell\big(c_{X_t}(t), s_{X_t}(t)\big), \end{aligned} \tag{A.2}$$

where $\ell : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is given by

$$\ell(c,s) = \sum_{i=1}^{d} \left(-c_i + s_i + c_i \log \frac{c_i}{s_i} \right).$$

In the right-hand side of (A.2), the first term is bounded by Proposition 3.3 that

$$\mathrm{KL}\big(p(T)\|\mathrm{Unif}(\{0,1\}^d)\big) \lesssim de^{-T},$$

the second term is bounded by $T\epsilon$ under Assumption 4.1. Thus, we obtain the desired bound.

Proof of the TV distance bound. We bound the TV distance between the data distribution p_0 and the perturbed distribution p_δ. Consider the forward process $(X_t)_{t\geq 0}$. Since (X_0, X_δ) gives a coupling of p_0 and p_δ, we have

$$\mathrm{TV}(p_0, p_\delta) \leq \mathbb{P}(X_0 \neq X_\delta).$$

$\mathbb{P}(X_0 \neq X_\delta)$ equals to the probability that a $\mathrm{Poisson}(d\delta)$ random variable is nonzero, which is $1 - e^{-d\delta}$. Thus by triangle inequality and Pinsker's inequality $\mathrm{TV} \lesssim \sqrt{\mathrm{KL}}$, we have

$$\mathrm{TV}\big(p_0, \hat{p}^{\leftarrow}(T-\delta)\big) \leq d\delta + \mathrm{TV}\big(p_\delta, \hat{p}^{\leftarrow}(T-\delta)\big) \lesssim (1 - e^{-d\delta}) + \sqrt{T\epsilon + de^{-T}}.$$

We complete the proof. $\qquad\square$

A.4 Proof of Theorem 4.2

Lemma A.1. *Let $p(t)$ be the distribution of the forward CTMC with generator Q given in (3.5). Suppose Assumption 4.3 holds. For any $t > 0$ we have*

$$\frac{p_{x+e_i}(t)}{p_x(t)} \leq L.$$

Proof. Let g be the heat kernel defined in Proposition 3.2. Consider the conditional distribution $\tilde{p}(a|x) \propto p_a(0)g_{x-a}(t)$ that is the distribution of X_t conditional on X_0. We write the probability ratio as

$$\begin{aligned}
\frac{p_{x+e_i}(t)}{p_x(t)} &= \frac{\sum_{a+w=x+e_i} p_a(0)g_w(t)}{\sum_{a+w=x} p_a(0)g_w(t)} \\
&= \frac{\sum_{a+w=x} p_{a+e_i}(0)g_w(t)}{\sum_{a+w=x} p_a(0)g_w(t)} \\
&= \frac{\sum_{a+w=x} p_a(0)\left(p_{a+e_i}(t)/p_a(t)\right)g_w(t)}{\sum_{a+w=x} p_a(0)g_w(t)} \\
&= \mathbb{E}_{\tilde{p}(a|x)}\frac{p_{a+e_i}(0)}{p_a(0)} \leq L,
\end{aligned}$$

where the last inequality comes from Assumption 4.3. We complete the proof. $\square$

Proof of Theorem 4.2. The KL divergence bound is similar to the proof of Theorem 4.1. We only need to consider the total number of transitions. We choose the time partition such that

$$\begin{aligned}
t_N &= T, \quad t_{N-1} = T - 1/L, \\
t_{k+1} - t_k &\leq c(T - t_{k+1}), \quad \forall\, t_{k+1} - t_k \leq c(T - t_{k+1}), \quad 1 \leq k \leq N-2
\end{aligned}$$

for some absolute constant c and choose λ_k's by

$$\lambda_N = \frac{d}{L}, \quad \lambda_k = \frac{Cd}{\min(1, T - t_k)}, \quad 1 \leq k \leq N-1.$$

Let $s_k = T - t_k$. The total number of steps follows a Poisson distribution with parameter $\lambda = \sum_{k=1}^{N} \lambda_k(t_k - t_{k-1})$. We have

$$\begin{aligned}
\lambda &= \lambda_N(t_N - t_{N-1}) + \sum_{k=1}^{N-1} \lambda_k(t_k - t_{k-1}) \\
&\leq 1 + \sum_{k:1/L \leq s_k < 1} \frac{Cd}{s_k}(s_{k-1} - s_k) + \sum_{k:1 \leq s_k \leq T} Cd(s_k - s_{k-1}) \\
&\lesssim d + \int_{1/L}^{1} \frac{d}{s}ds + \sum_{k:1 \leq s_k \leq T} d(s_k - s_{k-1}) \lesssim d\,(T + \log L).
\end{aligned}$$

We complete the proof. $\square$

Acknowledgments

References

[1] B. D. O. Anderson, Reverse-time diffusion equation models, *Stochastic Process. Appl.*, 12:313–326, 1982.

[2] J. Austin, D. D. Johnson, J. Ho, D. Tarlow, and R. van den Berg, Structured denoising diffusion models in discrete state-spaces, *arXiv:2107.03006*, 2023.

[3] O. Bar-Tal, L. Yariv, Y. Lipman, and T. Dekel, Multidiffusion: Fusing diffusion paths for controlled image generation, *arXiv:2302.08113*, 2023.

[4] J. Benton, V. D. Bortoli, A. Doucet, and G. Deligiannidis, Nearly d-linear convergence bounds for diffusion models via stochastic localization, *arXiv:2308.03686*, 2023.

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

[6] S. Boucheron, G. Lugosi, and P. Massart, *Concentration Inequalities: A Nonasymptotic Theory of Independence*, Oxford University Press, 2013.

[7] A. Campbell, J. Benton, V. De Bortoli, T. Rainforth, G. Deligiannidis, and A. Doucet, A continuous time framework for discrete denoising models, in: *Advances in Neural Information Processing Systems*, Vol. 35, Curran Associates, Inc., 28266–28279, 2022.

[8] H. Chen, H. Lee, and J. Lu, Improved analysis of score-based generative modeling: User-friendly bounds under minimal smoothness assumptions, *arXiv:2211.01916*, 2023.

[9] S. Chen, S. Chewi, H. Lee, Y. Li, J. Lu, and A. Salim, The probability flow ODE is provably fast, *arXiv:2305.11798*, 2023.

[10] S. Chen, S. Chewi, J. Li, Y. Li, A. Salim, and A. R. Zhang, Sampling is as easy as learning the score: Theory for diffusion models with minimal data assumptions, *arXiv:2209.11215*, 2022.

[11] S. Chewi, *Log-Concave Sampling*, 2023. `https://chewisinho.github.io`

[12] E. de Souza e Silva and H. R. Gail, Transient solutions for Markov chains, in: *Computational Probability. International Series in Operations Research & Management Science*, Vol. 24, Springer, 43–79, 2000.

[13] V. DeBortoli, Convergence of denoising diffusion models under the manifold hypothesis, *arXiv:2208.05314*, 2022.

[14] W. Grassmann, Transient solutions in Markovian queueing systems, *Comput. Oper. Res.*, 4(1):47–53, 1977.

[15] Z. Guo, J. Liu, Y. Wang, M. Chen, D. Wang, D. Xu, and J. Cheng, Diffusion models in bioinformatics: A new wave of deep learning revolution in action, *arXiv:2302.10907*, 2023.

[16] S. Gupta, A. Parulekar, E. Price, and Z. Xun, Sample-efficient training for diffusion, *arXiv:2311.13745*, 2023.

[17] J. Ho, T. Salimans, A. Gritsenko, W. Chan, M. Norouzi, and D. J. Fleet, Video diffusion models, *arXiv:2204.03458*, 2022.

[18] E. Hoogeboom, D. Nielsen, P. Jaini, P. Forré, and M. Welling, Argmax flows and multinomial diffusion: Learning categorical distributions, *arXiv:2102.05379*, 2021.

[19] H. Huang, L. Sun, B. Du, and W. Lv, Conditional diffusion based on discrete graph structures for molecular graph generation, in: *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 37(4), AAAI, 4302–4311, 2023.

[20] A. Hyvärinen, Estimation of non-normalized statistical models by score matching, *J. Mach. Learn. Res.*, 6:695–709, 2005.

[21] H. Lee, J. Lu, and Y. Tan, Convergence for score-based generative modeling with polynomial complexity, *arXiv:2206.06227*, 2022.

[22] H. Lee, J. Lu, and Y. Tan, Convergence of score-based generative modeling for general data distributions, *arXiv:2209.12381*, 2022.

[23] G. Li, Y. Wei, Y. Chen, and Y. Chi, Towards faster non-asymptotic convergence for diffusion-based generative models, *arXiv:2306.09251* , 2023.

[24] P. Li, Z. Li, H. Zhang, and J. Bian, On the generalization properties of diffusion models, *arXiv:2311.01797*, 2024.

[25] A. Lou, C. Meng, and S. Ermon, Discrete diffusion language modeling by estimating the ratios of the data distribution, *arXiv:2310.16834*, 2023.

[26] A. Lubotzky, R. Phillips, and P. Sarnak, Ramanujan graphs, *Combinatorica*, 8:261–277, 2017.

[27] C. Meng, K. Choi, J. Song, and S. Ermon, Concrete score matching: Generalized score matching for discrete data, *arXiv:2211.00802*, 2023.

[28] A. Nichol and P. Dhariwal, Improved denoising diffusion probabilistic models, *arXiv:2102.09672*, 2021.

[29] C. Niu, Y. Song, J. Song, S. Zhao, A. Grover, and S. Ermon, Permutation invariant graph generation via score-based generative modeling, in: *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, Vol. 108, PMLR, 4474–4484, 2020.

[30] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen, Hierarchical text-conditional image generation with CLIP latents, *arXiv:2204.06125*, 2022.

[31] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, and I. Sutskever, Zero-shot text-to-image generation, *arXiv:2102.12092*, 2021.

[32] J. E. Santos, Z. R. Fox, N. Lubbers, and Y. T. Lin, Blackout diffusion: Generative diffusion models in discrete-state spaces, *arXiv:2305.11089*, 2023.

[33] F. Schneider, ArchiSound: Audio generation with diffusion, *arXiv:2301.13267*, 2023.

[34] A. Seff, W. Zhou, F. Damani, A. Doyle, and R. P. Adams, Discrete object generation with reversible inductive construction, *arXiv:1907.08268*, 2019.

[35] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, Deep unsupervised learning using nonequilibrium thermodynamics, in: *Proceedings of Machine Learning Research*, Vol. 37, PMLR, 2256–2265, 2015.

[36] Y. Song, C. Durkan, I. Murray, and S. Ermon, Maximum likelihood training of score-based diffusion models, *arXiv:2101.09258*, 2021.

[37] Y. Song and S. Ermon, Generative modeling by estimating gradients of the data distribution, in: *Advances in Neural Information Processing Systems*, Vol. 32, Curran Associates, Inc., 11886–11889, 2019.

[38] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, and B. Poole, Score-based generative modeling through stochastic differential equations, in: *International Conference on Learning Representations*, Open-Review.net, 2020.

[39] H. Sun, L. Yu, B. Dai, D. Schuurmans, and H. Dai, Score-based continuous-time discrete diffusion models, *arXiv:2211.16750*, 2023.

[40] N. M. van Dijk, Approximate uniformization for continuous-time Markov chains with an application to performability analysis, *Stochastic Process. Appl.*, 40(2):339–357, 1992.

[41] N. M. Van Dijk, Uniformization for nonhomogeneous Markov chains, *Oper. Res. Lett.*, 12(5):283–291, 1992.

[42] P. Vincent, A connection between score matching and denoising autoencoders, *Neural Comput.*, 23:1661–1674, 2011.

[43] R. Yang, P. Srivastava, and S. Mandt, Diffusion probabilistic modeling for video generation, *arXiv: 2203.09481*, 2022.

[44] L. Zheng, J. Yuan, L. Yu, and L. Kong, A reparameterized discrete diffusion model for text generation, *arXiv:2302.05737*, 2023.