Formulating Discrete Probability Flow Through Optimal Transport

Pengze Zhang *
Sun Yat-sen University
zhangpz3@mail2.edu.cn

Hubery Yin*
WeChat, Tencent Inc.
hubery@tencent.com

Chen Li
WeChat, Tencent Inc.
chaselli@tencent.com

Xiaohua Xie †
Sun Yat-sen University
xiexiaoh6@mail.edu.cn

Abstract

Continuous diffusion models are commonly acknowledged to display a deterministic probability flow, whereas discrete diffusion models do not. In this paper, we aim to establish the fundamental theory for the probability flow of discrete diffusion models. Specifically, we first prove that the continuous probability flow is the Monge optimal transport map under certain conditions, and also present an equivalent evidence for discrete cases. In view of these findings, we are then able to define the discrete probability flow in line with the principles of optimal transport. Finally, drawing upon our newly established definitions, we propose a novel sampling method that surpasses previous discrete diffusion models in its ability to generate more certain outcomes. Extensive experiments on the synthetic toy dataset and the CIFAR-10 dataset have validated the effectiveness of our proposed discrete probability flow. Code is released at: https://github.com/PangzeCheung/Discrete-Probability-Flow.

1 Introduction

The emerging diffusion-based models [47, 22, 49, 50] have been proven to be an effective technique for modeling data distribution, and generating high-quality texts [34, 16], images [37, 13, 44, 41, 42, 23] and videos [24, 21, 43, 55, 19]. Considering their generative capabilities have surpassed the previous state-of-the-art results achieved by generative adversarial networks [13], there has been a growing interest in exploring the potential of diffusion models in various advanced applications [45, 36, 52, 59, 11, 35, 53, 56, 20, 57].

Diffusion models are widely recognized for generating samples in a stochastic manner [50], which complicates the task of defining an encoder that translates a sample to a certain latent space. For instance, by following the configuration proposed by [22], it has been observed that generated samples from any given initial point have the potential to span the entire support of the data distribution. To achieve a deterministic sampling process while preserving the generative capability, Song *et al.*[50] proposed the probability flow, which provides a deterministic map between the data space and the latent space for continuous diffusion models. Unfortunately, the situation differs when it comes to discrete models. For instance, considering two binary distributions $(P_0 = \frac{1}{2}, P_1 = \frac{1}{2})$ and $(P_0 = \frac{1}{3}, P_1 = \frac{2}{3})$, there is no deterministic map that can transform the former distribution to the latter one, as it would simply be a permutation. Although some previous research has been conducted

*Equal contribution. This work was done when Pengze Zhang was an intern at WeChat.
†Corresponding author.

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

on discrete diffusion models with discrete [26, 25, 4, 14, 10, 28, 18] and continuous [7, 51] time configurations, these works primarily focus on improving the sampling quality and efficiency, while sampling certainty has received less attention. More specifically, there is a conspicuous absence of existing literature addressing the probability flow in discrete diffusion models.

The aim of this study is to establish the fundamental theory of the probability flow for discrete diffusion models. Our paper contributes in the following ways. Firstly, we provide proof that under some conditions the probability flow of continuous diffusion coincides with the Monge optimal transport map during any finite time interval within the range of $(0, \infty)$. Secondly, we propose a discrete analogue of the probability flow under the framework of optimal transport, which we have defined as the *discrete probability flow*. Additionally, we identify several properties that are shared by both the continuous and discrete probability flow. Lastly, we propose a novel sampling method based on the aforementioned observations, and we demonstrate its effectiveness in significantly improving the certainty of the sampling outcomes on both synthetic toy dataset and CIFAR-10 dataset.

Proofs for all Propositions are given in the Appendix. For consistency, the probability flow and infinitesimal transport of a process X_t is signified by $\hat{X}_t$ and $\tilde{X}_t$ respectively.

2 Background on Diffusion Models and Optimal Transport

First of all, we review some important concepts from the theory of diffusion models, optimal transport and gradient flow.

2.1 Continuous state diffusion models

Diffusion models are generative models that consist of a forward process and a backward process. The forward process transforms the data distribution $p_{data}(x_0)$ into a tractable reference distribution $p_T(x_T)$. The backward process then generates samples from the initial points drawn from $p_T(x_T)$. According to [30], the forward process is modeled as the (time-dependent) Ornstein-Uhlenbeck (OU) process:

$$dX_t = -\theta_t X_t dt + \sigma_t dB_t, \tag{1}$$

where $\theta_t \geq 0, \sigma_t > 0, \forall t \geq 0$ and B_t is the Brownian Motion (BM). The backward process is the reverse-time process of the forward process [2]:

$$dX_t = [-\theta_t X_t - \sigma_t^2 \nabla_{X_t} \log p(X_t, t)]dt + \sigma_t d\tilde{B}_t, \tag{2}$$

where $\tilde{B}_t$ is the reverse-time Brownian motion and $p(X_t, t)$ is the single-time marginal distribution of the forward process, which also serves as the solution to the Fokker-Planck equation [39]:

$$\frac{\partial}{\partial t} p(x, t) = \theta_t \nabla_x (x p(x, t)) + \frac{1}{2} \sigma_t^2 \Delta_x p(x, t). \tag{3}$$

In order to train a diffusion model, the primary objective is to minimize the discrepancy between the model output $s_\theta(x_t, t)$ and the Stein score function $s(x_t, t) = \nabla_{x_t} \log p(x_t, t)$ [27]. Song *et al.* [49] demonstrate that, it is equivalent to match $s_\theta(x_t, t)$ with the conditional score function:

$$\theta^* = \arg\min_\theta \mathbb{E}_t \left\{ \lambda_t \mathbb{E}_{x_0, x_t} \left[\| s_\theta(x_t, t) - \nabla_{x_t} \log p(x_t, t | x_0, 0) \|^2 \right] \right\}, \tag{4}$$

where λ_t is a weighting function, t is uniformly sampled over $[0, T]$ and $p(x_t, t|x_0, 0)$ is the forward conditional distribution.

It is noted that every Ornstein-Uhlenbeck process has an associated probability flow, which is a deterministic process that shares the same single-time marginal distribution [50]. The probability flow is governed by the following Ordinary Differential Equation (ODE):

$$d\hat{X}_t = [-\theta_t \hat{X}_t - \frac{1}{2} \sigma_t^2 s(\hat{X}_t, t)]dt. \tag{5}$$

In accordance with the global version of Picard-Lindelöf theorem [1] and the adjoint method[40, 8], the map

$$\begin{aligned} T_{s,t} : \mathbb{R}^n &\longrightarrow \mathbb{R}^n, \\ \hat{X}_s &\longmapsto \hat{X}_t. \end{aligned} \tag{6}$$

is a diffeomorphism $\forall t \geq s > 0$. The diffeomorphism naturally gives a transport map.

2.2 Discrete state diffusion models

In the realm of discrete state diffusion models, there are two primary classifications: the Discrete Time Discrete State (DTDS) models and the Continuous Time Discrete State (CTDS) models, which are founded on Discrete Time Markov Chains (DTMC) and Continuous Time Markov Chains (CTMC), correspondingly. Campbell *et al.*[7] conducted a comparative analysis of these models and determined that CTDS outperforms DTDS. The DTDS models construct the forward process through the utilization of the conditional distribution $q_{t+1|t}(x_{t+1}|x_t)$ and employ a neural network to approximate the reverse conditional distribution $q_{t|t+1}(x_t|x_{t+1}) = \frac{q_{t+1|t}(x_{t+1}|x_t)q_t(x_t)}{q_{t+1}(x_{t+1})}$. In practical applications, it is preferable to parameterize this model using $p^\theta_{0|t+1}$ [26, 4] and obtain $p^\theta_{k|k+1}$ through

$$\begin{aligned} p^\theta_{k|k+1}(x_k|x_{k+1}) &= \sum_{x_0} q_{k|k+1,0}(x_k|x_{k+1}, x_0) p^\theta_{0|k+1}(x_0|x_{k+1}) \\ &= \sum_{x_0} q_{k+1|k}(x_{k+1}|x_k) \frac{q_{k|0}(x_k|x_0)}{q_{k+1|0}(x_{k+1}|x_0)} p^\theta_{0|k+1}(x_0|x_{k+1}). \end{aligned} \tag{7}$$

In contrast to DTDS models, a CTDS model is characterized by the (infinitesimal) generator [3], or transition rate, $Q_t(x, y)$. The Kolmogorov forward equation [15] is:

$$\frac{\partial}{\partial t} q_{t|s}(x_t|x_s) = \sum_y q_{t|s}(y|x_s) Q_t(y, x_t). \tag{8}$$

The reverse process is:

$$\frac{\partial}{\partial s} q_{s|t}(x_s|x_t) = \sum_y q_{s|t}(y|x_t) R_t(y, x_s). \tag{9}$$

The generator of the reverse process can be written by [7, 51]:

$$R_t(y, x) = \frac{q_t(x)}{q_t(y)} Q_t(x, y) = \sum_{y_0} \frac{q_{t|0}(x|y_0)}{q_{t|0}(y|y_0)} q_{0|t}(y_0|y) Q_t(x, y). \tag{10}$$

There are various approaches to train the model, such as the Evidence Lower Bound (ELBO) technique [7], and the score-based approach [51]. It has been observed that the reverse generator can be factorized over dimensions, allowing parallel sampling for each dimension during the reverse process. However, it is important to note that this independence is only possible when the time interval for each step is small.

2.3 Optimal transport

The *optimal transport problem* can be formulated in two primary ways, namely the Monge formulation and the Kantorovich formulation [46]. Suppose there are two probability measures μ and ν on $(\mathbb{R}^n, \mathcal{B})$, and a cost function $c : \mathbb{R}^n \times \mathbb{R}^n \to [0, +\infty]$. The *Monge problem* is

$$\text{(MP)} \ \inf_{\mathrm{T}} \left\{ \int c(x, \mathrm{T}(x)) \, \mathrm{d}\mu(x) : \mathrm{T}_\# \mu = \nu \right\}. \tag{11}$$

The measure $\mathrm{T}_\#\mu$ is defined through $\mathrm{T}_\#\mu(A) = \mu(\mathrm{T}^{-1}(A))$ for every $A \in \mathcal{B}$ and is called the *pushforward* of μ through T.

It is evident that the Monge Problem (MP) transports the entire mass from a particular point, denoted as x, to a single point $\mathrm{T}(x)$. In contrast, Kantorovich provided a more general formulation, referred to as the *Kantorovich problem*:

$$\text{(KP)} \ \inf_{\gamma} \left\{ \int_{\mathbb{R}^n \times \mathbb{R}^n} c \, \mathrm{d}\gamma : \gamma \in \mathit{\Pi}(\mu, \nu) \right\}, \tag{12}$$

where $\mathit{\Pi}(\mu, \nu)$ is the set of *transport plans*, i.e.,

$$\mathit{\Pi}(\mu, \nu) = \left\{ \gamma \in \mathcal{P}(\mathbb{R}^n \times \mathbb{R}^n) : (\pi_x)_\# \gamma = \mu, (\pi_y)_\# \gamma = \nu \right\}, \tag{13}$$

where π_x and π_y are the two projections of $\mathbb{R}^n \times \mathbb{R}^n$ onto $\mathbb{R}^n$. For measures absolutely continuous with respect to the Lebesgue measure, these two problems are equivalent [54]. However, when the measures are discrete, they are entirely distinct as the constraint of the Monge Problem may never be fulfilled.

2.4 Fokker-Planck equation by gradient flow

According to [29], the Fokker-Planck equation represents the gradient flow of a functional in a metric space. In particular, for Brownian motion, its Fokker-Planck equation, which is also known as the heat diffusion equation, can be expressed as:

$$\frac{\partial}{\partial t}p(x,t) = \frac{1}{2}\Delta p(x,t), \tag{14}$$

and it represents the gradient flow of the Gibbs-Boltzmann entropy multiplied by $-\frac{1}{2}$:

$$-\frac{1}{2}S(p) = \frac{1}{2}\int_{\mathbb{R}^n} p(x)\log p(x)\,\mathrm{d}x. \tag{15}$$

It is worth noting that Eq. 15 is the gradient flow of Eq. 14 under the *2-wasserstein metric* (W_2).

Chow *et al.* [9] have developed an analogue in the discrete setting by introducing the discrete Gibbs-Boltzmann entropy:

$$S(p) = \sum_i p_i \; log \; p_i, \tag{16}$$

and deriving the gradient flow using a newly defined metric (Definition 1 in [9]). Since the discrete model is defined on graph $G(V, E)$, where $V = \{a_1, ..., a_N\}$ is the set of vertices, and E is the set of edges, the discrete Fokker-Planck equation with a constant potential can be written as:

$$\frac{d}{dt}p_i = \sum_{j\in N(i)} p_j - p_i, \tag{17}$$

where $N(i) = \{j \in \{1, 2, ..., N\} | \{a_i, a_j\} \in E\}$ represents the one-ring neighborhood.

3 Continuous probability flow

3.1 The equivalence of Ornstein-Uhlenbeck processes and Brownian motion

The diffusion models that are commonly utilized in machine learning are founded on Ornstein-Uhlenbeck processes. First of all, we demonstrate that it is feasible to deterministically convert a time-dependent Ornstein-Uhlenbeck process into a standard Brownian motion.

Proposition 1. *Let X_t and Y_t be a time-dependent Ornstein-Uhlenbeck process and a Brownian motion respectively: $dX_t = -\theta_t X_t dt + \sigma_t dB_t^{(1)}$, $dY_t = dB_t^{(2)}$, where $B_t^{(1)}$ and $B_t^{(2)}$ are two independent Brownian motions and $\theta_t \geq 0, \sigma_t > 0, \forall t \geq 0$. Let $\phi_t = \exp(\int_0^t \theta_\tau \,\mathrm{d}\tau)$, $\beta_t = \int_0^t (\sigma_\tau \phi_\tau)^2 \,\mathrm{d}\tau$. Then X_t coincides in law with $\phi_t^{-1} Y_{\beta_t}$.*

Building upon the aforementioned proposition, the primary focus of this paper is centered around the standard Brownian motion $dY_t = dB_t$.

3.2 Probability flow is a Monge map

Khrulkov *et al.* [31] have proposed a conjecture that the probability flow of Ornstein-Uhlenbeck process is a Monge map. However, they only provided a proof for a simplified case. We demonstrate that under some conditions, the conjecture is correct.

It is important to highlight that the continuous optimal transports presented in this paper are defined exclusively with the cost function: $c(x, y) = \frac{1}{2}|x - y|^2$.

Within the context of generative models, a collection of training samples denoted as $\{x_i\}_{i=1}^N$ is typically provided, and these samples are intrinsically defined by a distribution:

$$p(x, 0) = \frac{1}{N}\sum_{i=1}^{N} \delta(x - x_i), \tag{18}$$

where $\delta(x)$ represents the Dirac delta function. Given a Brownian motion with an initial distribution in the form of Equation (18), the single-time marginal distribution is [39]

$$p_B(x,t) = \frac{1}{N}\sum_{i=1}^{N}(2\pi t)^{-\frac{n}{2}}\exp(-\frac{|x-x_i|^2}{2t}). \tag{19}$$

The probability flow is defined as [50]:

$$d\hat{Y}_t = -\frac{1}{2}\nabla_{\hat{Y}_t}\log p_B(\hat{Y}_t,t)dt. \tag{20}$$

According to [1, 40, 8], the solution exists for all $t > 0$ and the map $\hat{Y}_{t+s}(\hat{Y}_t)$ is a diffeomorphism for all $t > 0, s \geq 0$. We have discovered that $\hat{Y}_{t+s}(\hat{Y}_t)$ is the Monge map under some conditions and the time does not reach 0 or $+\infty$.

Proposition 2. *Given that Y_0 follows the initial condition (18), and all x_is lie on the same line, the diffeomorphism $\hat{Y}_{t+s}(\hat{Y}_t)$ is the Monge optimal transport map between $p_B(x,t)$ and $p_B(x,t+s)$, $\forall\, t > 0, s \geq 0$.*

There is a counterexample [33] to demonstrate that the probability flow map does not necessarily provide optimal transport. It is important to note that their case differs from our assumptions in two ways. Firstly, they consider the limit case of $\hat{Y}_{+\infty}(\hat{Y}_0)$. Secondly, the initial distribution of the counterexample does not conform to the form specified in Equation (18). Therefore, their counterexample is not applicable to our situation.

It has been shown that the heat diffusion equation can be regarded as the *gradient flow* of the Gibbs-Boltzmann entropy concerning the W_2 *metric* [29]. As W_2 is associated with optimal transport, it is reasonable to anticipate that the "infinitesimal transport" $\hat{Y}_{t+dt}(\hat{Y}_t)$ is optimal [31].

In order to interpret the concept of "infinitesimal transport", we utilize the generator of the process Y_t. Let $C_c^2(\mathbb{R}^n)$ denote the set of twice continuously differentiable functions on $\mathbb{R}^n$ with compact support. The generator A_t is defined as follows [39]:

$$\hat{A}_t f = \lim_{\Delta t \to 0^+}\frac{f(\hat{Y}_{t+\Delta t}) - f(\hat{Y}_t)}{\Delta t}, \forall f \in C_c^2(\mathbb{R}^n). \tag{21}$$

It is straightforward to verify that

$$\hat{A}_t = -\frac{1}{2}\nabla_x \log p_B(x,t)^T \nabla_x. \tag{22}$$

We define the "infinitesimal transport" to be the diffeomorphism $\tilde{Y}_{t+s}(\tilde{Y}_t)$ where $\tilde{Y}_{t+s}$ evolves according to the following equation

$$d\tilde{Y}_{t+s} = -\frac{1}{2}\nabla_{\tilde{Y}_t}\log p_B(\tilde{Y}_t(\tilde{Y}_{t+s}),t)ds, \tag{23}$$

with the initial condition $\tilde{Y}_t = \hat{Y}_t$. The generator of $\tilde{Y}_{t+s}$ is

$$\tilde{A}_{t+s} = -\frac{1}{2}\nabla_{\tilde{Y}_t}\log p_B(\tilde{Y}_t(\tilde{Y}_{t+s}),t)\nabla_x. \tag{24}$$

Proposition 3. *Given any $t > 0$, there exists a $\delta_t > 0$ s.t. $\forall\, 0 < s < \delta_t$, the diffeomorphism $\tilde{Y}_{t+s}(\tilde{Y}_t)$ with the initial condition $\tilde{Y}_t = \hat{Y}_t$ is the Monge optimal transport map.*

Let us return to the original Ornstein-Uhlenbeck process X_t. As it is merely a deterministic transformation of the Brownian motion Y_t, we can anticipate that the probability flow of X_t, denoted by $\hat{X}_t$, will be a Monge map. In fact, this expectation holds true:

Proposition 4. *Given that X_0 follows the initial condition (18), and all x_is lie on the same line, the diffeomorphism $\hat{X}_{t+s}(\hat{X}_t)$ is the Monge optimal transport map for all $t > 0, s \geq 0$.*

4 Discrete probability flow

The continuous probability flow is deterministic, which means the "mass" at $\hat{Y}_t$ is entirely transported to $\hat{Y}_{t+s}$ during the time interval $[t, t+s]$. However, it is widely acknowledged that for discrete distributions μ and ν, there may not exist a T such that $\mathrm{T}_{\#}\mu = \nu$. As a result, discrete diffusions cannot possess a deterministic probability flow. To establish the concept of the *discrete probability flow*, we employ the methodology of optimal transport. First of all, a discrete diffusion model is proposed as an analogue of Brownian motion. Secondly, we modified the forward process to create an optimal transport map, which is used to define the discrete probability flow. Finally, a novel sampling technique is introduced, which significantly improves the certainty of the sampling outcomes.

4.1 Constructing discrete probability flow

It is demonstrated that the process described by Equation (17) is a discrete equivalent of the heat diffusion process (14) [9]. We adopt this process as our discrete diffusion model and represent it in a more comprehensive notation.

The discrete diffusion model has K dimensions and S states. The states are denoted by $i = (i_1, i_2, \ldots, i_K)$, where $i_j \in \{1, 2, \ldots, S\}$. The Kolmogorov forward equation for this process is

$$\frac{d}{dt} P_j^i(t|s) = \sum_{j'} P_{j'}^i(t|s) {Q_D}_j^{j'}(t), \tag{25}$$

where $P_j^i(t|s)$ means $P(x_t = j | x_s = i)$ and Q_D is defined as:

$${Q_D}_j^i = \begin{cases} 1, & d_D(i,j) = 1, \\ -\sum_{j' \in \{k : d_D(i,k)=1\}} {Q_D}_{j'}^i, & d_D(i,j) = 0, \\ 0, & otherwise, \end{cases} \tag{26}$$

where $d_D(i,j) = \sum_{l=1}^{K} |i_l - j_l|$. If we let the solution of the Equation (25) be denoted by $P_D(t|s)$ and assume an initial condition P_0, the single-time marginal distribution can be computed as follows:

$$P_{Di}(t) = \sum_j P_{0j} {Q_D}_i^j(t|0). \tag{27}$$

It is noteworthy that the process defined by Q_D is not an optimal transport map, as there exist *mutual flows* between the states (i.e., there exists two states i, j with $Q_j^i > 0$ and $Q_i^j > 0$). Therefore, we propose a modified version that will be proved to be a solution to the Kantorovich problem, namely, an optimal transport plan. The modified version is defined by the following Q:

$$Q_j^i(t) = \begin{cases} \frac{ReLU(P_{Di}(t) - P_{Dj}(t))}{P_{Di}(t)}, & d_D(i,j) = 1, \\ -\sum_{j' \in \{k : d_D(i,k)=1\}} Q_{j'}^i(t), & d_D(i,j) = 0, \\ 0, & otherwise. \end{cases} \tag{28}$$

where

$$ReLU(x) = \begin{cases} x, & x > 0, \\ 0, & x \leq 0. \end{cases} \tag{29}$$

In order to avoid singular cases, We define $Q_j^i(t)$ to be 0 when $P_{Di}(t) = 0$. In fact, it is easy to verify that $P_{Di}(t) > 0$ for all $t > 0$, $i \in \{1, 2, \ldots, K\}$. We will show that the process defined by Q is equivalent in distribution to the one generated by Q_D.

Proposition 5. *The processes generated by Q_D and Q have the same single-time marginal distribution $\forall t > 0$.*

Proposition 6. *Given any $t > 0$, there exists a $\delta_t > 0$ s.t. $\forall\, 0 < s < \delta_t$, the process generated by Q provides an optimal transport map from $P_D(t)$ to $P_D(t+s)$ under the cost d_D.*

Proposition 6 demonstrates that Q_D generates a Kantorovich plan between $P_D(t)$ and $P_D(t+s)$ under a certain cost function. On the other hand, the continuous probability flow is the Monge map between $p_B(x,t)$ and $p_B(x,t+s)$. Therefore, it is reasonable to define the process defined by Q_D as the *discrete probability flow* of the original process defined by Q.

Furthermore, the "infinitesimal transport" of the discrete process, which is defined by $\frac{d}{ds}\hat{P}(t+s) = \hat{P}(t+s)Q(t)$, also provides an optimal transport map.

Proposition 7. *Given any $t > 0$, there exists a $\delta_t > 0$ s.t. $\forall\, 0 < s < \delta_t$, the process above provides an optimal transport map from $\hat{P}(t)$ to $\hat{P}(t+s)$ under the cost d_D.*

4.2 Sampling by discrete probability flow

In order to train the modified model, we employ a score-based method described in the Score-based Continuous-time Discrete Diffusion Model (SDDM) [51]. Specifically, we directly learn the conditional probability $P^\theta(i_l(t)|\{i_1,\ldots,i_{l-1},i_{l+1},\ldots,i_K\}(t))$. According to proposition 5, it follows that $P^\theta = {P^\theta}_D$, and consequently, the training process is identical to that of [51]. For the sake of brevity, we will employ the notation $P^\theta_{i_l|i\setminus i_l}(t)$ to replace $P^\theta(i_l(t)|\{i_1,\ldots,i_{l-1},i_{l+1},\ldots,i_K\}(t))$.

The generator of the reverse process is

$$R^i_j(t) = \begin{cases} ReLU(\frac{P^\theta_{D_{j_l|i\setminus i_l}}(t)}{P^\theta_{D_{i_l|i\setminus i_l}}(t)} - 1), & d_D(i,j) = 1 \text{ and } i_l \neq j_l, \\ -\sum_{j'\in\{k:d_D(i,k)=1\}} R^i_{j'}(t), & d_D(i,j) = 0, \\ 0, & otherwise. \end{cases} \tag{30}$$

We use the Euler's method to generate samples. Given the time step length ϵ, the transition probabilities for dimension l is:

$$P^\theta(i_l(t-\epsilon)|i(t)) = \begin{cases} \epsilon R^{i(t)}_{i_1(t),\ldots,i_l(t-\epsilon),\ldots,i_k(t)}(t), & i_l(t-\epsilon) \neq i_l(t), \\ 1 + \epsilon R^{i(t)}_{i(t)}(t), & i_l(t-\epsilon) = i_l(t). \end{cases} \tag{31}$$

When ϵ is small, the reverse conditional distribution has the factorized probability:

$$P^\theta(i(t-\epsilon)|i(t)) = \Pi_{l=1}^K P^\theta(i_l(t-\epsilon)|i(t)) \tag{32}$$

In this way, it becomes possible to generate samples by sequentially sampling from the reverse conditional distribution 32.

Transition to higher probability states The reverse process of the continuous probability flow, as described in Equation (20), causes particles to move towards areas with higher logarithmic probability densities. As the logarithm function is monotonically increasing, this reverse flow pushes particles to higher probability density states. This phenomenon is also observed in the discrete probability flow. By examining the reverse generator, as shown in Equation (30), it can be determined that the transition rate $R^i_j(t) > 0$ only when the destination state j has a higher probability than the source state i. This implies that transitions only occur in higher probability states. In contrast, the original continuous reverse process (2) and the discrete reverse process from (10) allow any transitions.

Reduction of Standard Deviation We measure the certainty of the sampling method by the expectation of the Conditional Standard Deviation (CSD):

$$CSD_{s,t}(X) = \mathbb{E}_{X_t}[\text{Std}(X_s|X_t)], \tag{33}$$

where $\text{Std}(X_s|X_t) = \text{Var}^{\frac{1}{2}}(X_s|X_t) = \mathbb{E}^{\frac{1}{2}}_{X_s}[X_s - \mathbb{E}_{X_s}[X_s|X_t]|X_t]$. $CSD_{s,t}(X)$ is 0 when the process is deterministic, such as the continuous probability flow. In the discrete situation, there does not exist any deterministic map. However, our discrete probability flow significantly reduces $CSD_{s,t}(X)$. Table 2 presents numerical evidence of this phenomenon. Therefore, we posit that the discrete probability flow enhances the certainty of the sampling outcomes.

Table 1: Comparison of generation quality for SDDM and DPF, in terms of MMD with Laplace kernel using bandwith=0.1. Lower values indicate superior quality.

	2spirals	8gaussians	checkerboard	circles	moons	pinwheel	swissroll
discrete dimension = 32, state size = 2							
SDDM	2.18e-06	4.28e-06	1.33e-06	6.22e-06	5.62e-06	2.10e-06	4.27e-06
DPF (ours)	1.89e-05	1.09e-05	2.22e-05	3.27e-05	2.42e-05	1.60e-05	2.18e-05
discrete dimension = 16, state size = 5							
SDDM	2.06e-4	1.01e-4	2.43e-4	1.74e-4	2.20e-4	3.37e-4	1.43e-4
DPF (ours)	3.87e-4	5.87e-4	4.93e-4	3.83e-4	3.43e-4	6.64e-4	3.20e-4
discrete dimension = 12, state size = 10							
SDDM	5.52e-4	3.01e-4	4.39e-4	4.22e-4	2.71e-4	2.90e-4	3.39e-4
DPF (ours)	7.19e-4	3.49e-4	5.99e-4	6.65e-4	4.34e-4	4.14e-4	5.17e-4

Table 2: Comparison of certainty for SDDM and DPF, in terms of CSD on 4,000 initial points, each of which has 10 generated samples. Lower values indicate superior certainty.

	2spirals	8gaussians	checkerboard	circles	moons	pinwheel	swissroll
discrete dimension = 32, state size = 2							
SDDM	14.3053	14.1882	14.7433	14.4327	14.1739	14.0450	14.0548
DPF (ours)	2.1719	1.7945	2.0693	1.7210	2.0573	2.1834	1.8892
discrete dimension = 16, state size = 5							
SDDM	14.4645	14.6143	14.6963	14.4807	14.2397	14.2466	14.2659
DPF (ours)	1.9711	1.9367	1.4172	1.7185	1.7668	1.9633	1.6665
discrete dimension = 12, state size = 10							
SDDM	12.8463	12.7933	13.0158	12.9232	12.6665	12.7634	12.7880
DPF (ours)	1.8123	1.3178	1.1348	1.4625	1.4859	1.8435	1.5227

5 Related Work

The concept of probability flow was initially introduced in [50] as a deterministic alternative to the Itô diffusion. In the work [48], they presented the Denoising Diffusion Implicit Model (DDIM) and demonstrated its equivalence to the probability flow. Subsequently, [31] investigated the relationship between the probability flow and optimal transport. They hypothesized that the probability flow could be considered a Monge optimal transition map and provided a proof for a specific case. Additionally, they conducted numerical experiments that supported their conjecture, showing negligible errors. However, [33] has discovered an initial distribution that renders probability flow not optimal.

The discrete diffusion models were first introduced by [47], who considered a binary model. Following the success of continuous diffusion models, discrete models have garnered more attention. The bulk of research on discrete models has focused primarily on the design of the forward process [26, 25, 4, 6, 28, 18, 10]. Continuous time discrete state models were introduced by [7] and subsequently developed by [51].

6 Experiments

We conduct numerical experiments using our novel sampling method by Discrete Probability Flow (DPF) on synthetic data. The primary goal is to demonstrate that our method can generate samples of comparable quality with higher certainty.

Experiments are conducted on synthetic data using the same setup as SDDM [51], with the exception that we replaced the generator Q with Equation (26). In addition to the binary situation ($S = 2$) studied in [51], we also perform experiments on synthetic data with the state size S set to 5 and 10. To evaluate the quality of the generated samples, we generated 40,000 / 4,000 samples for binary data / other type of data using SDDM and DPF, and measured the Maximum Mean Discrepancy (MMD) with the Laplace kernel [17]. The results are shown in Table 1. It can be seen that the MMD value obtained using DPF is slightly higher than that of SDDM, which may be attributed to the structure of the reverse generator 10. Specifically, DPF approximates an additional term, $Q_t(y, x)$, with the neural network, which potentially introduces additional errors to the sampling process, leading to a higher MMD value compared to SDDM. However, such difference is minimal and does not significantly impact the quality of the generated samples. As evident from the visualization of the

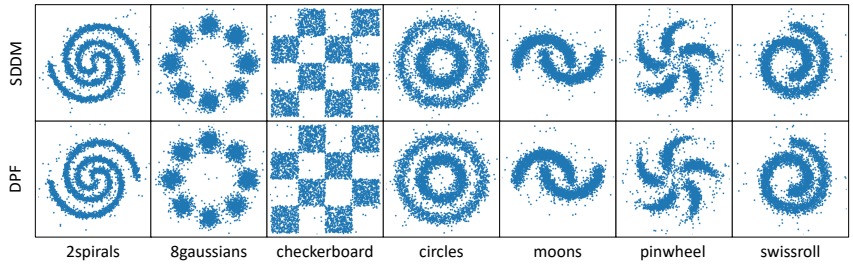


Figure 1: Visualization of the generation quality on generated binary samples for SDDM and DPF.

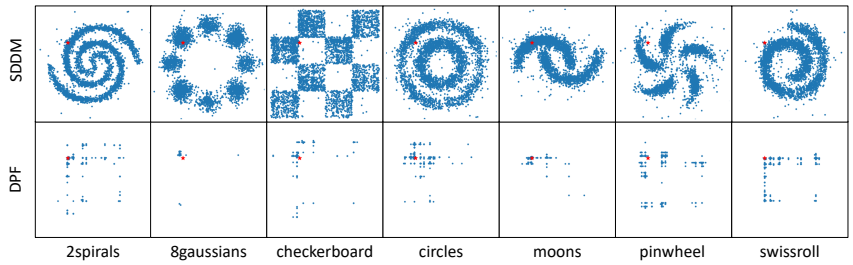


Figure 2: Visualization of the generating certainty on generated binary samples for SDDM and DPF. All the samples (in blue) are randomly generated from the single initial point (in red).

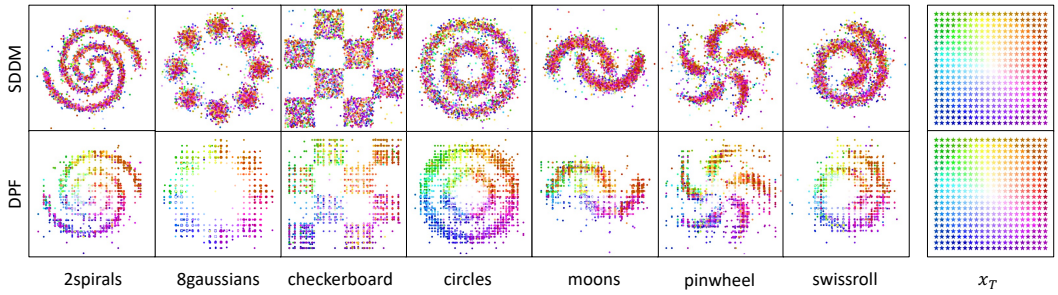


Figure 3: Visualization of the generated binary samples from the given initial points $\boldsymbol{x}_T$. Different colors distinguish the generated samples from different initial points $\boldsymbol{x}_T$.

distributions obtained from SDDM and DPF in Figure 1, it is clear that DPF can generate samples that are comparable to those generated by SDDM.

In addition, we also compare the sampling certainty of DPF and SDDM by computing $CSD_{s,t}$ using a Monte-Carlo based method. Specifically, we set $s = 0$ and $t = T$, and sample 4,000 x_ts with 10 x_ss for each x_t. We then estimate $\mathbb{E}(x_s|x_t)$ and $\text{Std}(x_s|x_t)$ using the sample mean and sample standard deviation, respectively. The results of certainty are presented in Table 2. Our findings indicate that DPF significantly reduces the CSD, which suggests a higher certainty. Additionally, we visualize the results of 4,000 generated samples (in blue) from a single initial point (in red) in the binary case in Figure 2. It is apparent that the sampling of SDDM exhibits high uncertainty, as it can sample the entire pattern from a single initial point. In contrast, our method reduces such uncertainty and is only able to sample a limited number of states.

To provide a more intuitive representation of the generated samples originating from various initial points, we select 20×20 initial points arranged in the grid, and distinguish them using different colors. Subsequently, we visualize the results by sampling 10 outcomes from each initial point, as shown in Figure 3. We observe that the visualization of SDDM samples appears disorganized, indicating significant uncertainty. In contrast, the visualization of DPF samples exhibits clear regularity, manifesting in two aspects: (1) the generated samples from the same initial point using DPF are clustered by color, demonstrating the better sampling certainty of our DPF. (2) Both of the generated samples and initial points are colored similarly at each position. For example, in the lower

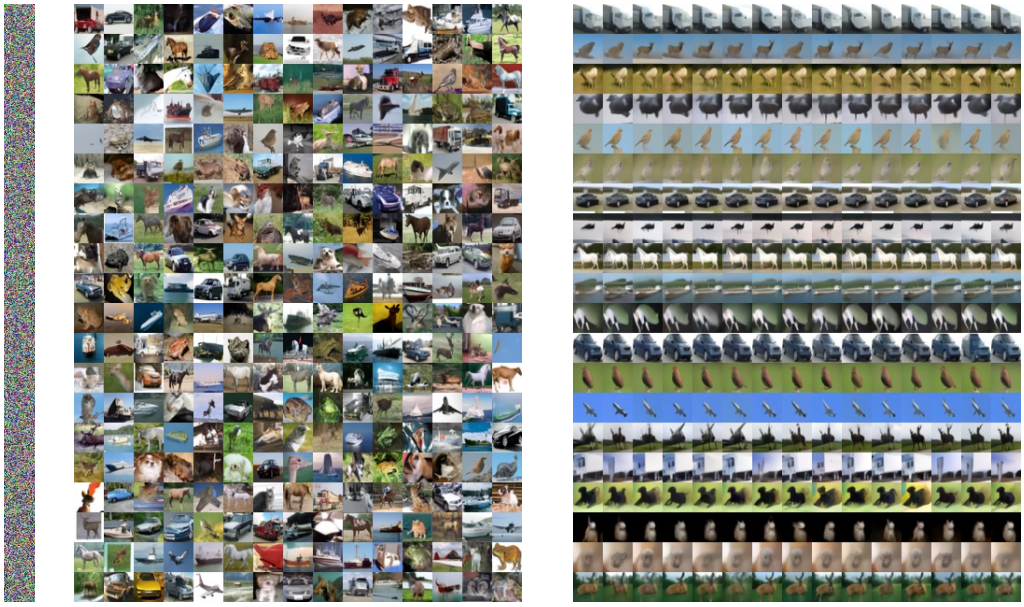


Figure 4: Image modeling on CIFAR-10 dataset. The figure is divided into three groups: initial points x_T, sampling results of τLDR-0, and sampling results of our DPF. For each row, the sampled images are obtained from the same initial point.

right area, a majority of the generated samples are colored purple, which corresponds to the color assigned to the initial points x_T in that area. This observation demonstrates that most of the sampling results obtained through DPF are closer to their respective initial points, aligning with our design intention of optimal transport. It is worth noting that similar phenomena are observed across different state sizes, and we have provided these results in the Appendix.

Finally, we extended our DPF to the CIFAR-10 dataset, and compare it with the τLDR-0 method proposed in [7]. The visualization results are shown in Figure 4. It can be seen that our method greatly reduces the uncertainty of generating images by sampling from the same initial x_T. Detailed experimental settings and more experimental results are presented in the Appendix.

7 Discussion

In this study, we introduce a discrete counterpart of the probability flow and established its connections with the continuous formulations. We began by demonstrating that the continuous probability flow corresponds to a Monge optimal transport map. Subsequently, we proposed a method to modify a discrete diffusion model to achieve a Kantorovich plan, which naturally defines the discrete probability flow. We also discovered shared properties between continuous and discrete probability flows. Finally, we propose a novel sampling method that significantly reduces sampling uncertainty. However, there are still remaining aspects to be explored in the context of the discrete probability flow. For instance, to obtain more general conclusions under a general initial condition, the semi-group method [58] could be employed. Additionally, while we have proven the existence of a Kantorovich plan in a small time interval, it is possible to extend this to a global solution. Moreover, the definition of the probability flow has been limited to a specific type of discrete diffusion model, which also could be extended to a broader range of models. These topics remain open for future studies.

8 Acknowledgments and Disclosure of Funding

We would like to thank all the reviewers for their constructive comments. Our work was supported in National Natural Science Foundation of China (NSFC) under Grant No.U22A2095 and No.62072482.

References

[1] Herbert Amann. *Ordinary differential equations: an introduction to nonlinear analysis*, volume 13. Walter de gruyter, 2011.

[2] Brian DO Anderson. Reverse-time diffusion equation models. *Stochastic Processes and their Applications*, 12(3):313–326, 1982.

[3] William J Anderson. *Continuous-time Markov chains: An applications-oriented approach*. Springer Science & Business Media, 2012.

[4] 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.

[5] Sergio Blanes, Fernando Casas, Jose-Angel Oteo, and José Ros. The magnus expansion and some of its applications. *Physics reports*, 470(5-6):151–238, 2009.

[6] Sam Bond-Taylor, Peter Hessey, Hiroshi Sasaki, Toby P Breckon, and Chris G Willcocks. Unleashing transformers: parallel token prediction with discrete absorbing diffusion for fast high-resolution image generation from vector-quantized codes. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXIII*, pages 170–188. Springer, 2022.

[7] 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.

[8] Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.

[9] Shui-Nee Chow, Wen Huang, Yao Li, and Haomin Zhou. Fokker–planck equations for a free energy functional or markov process on a graph. *Archive for Rational Mechanics and Analysis*, 203:969–1008, 2012.

[10] Max Cohen, Guillaume Quispe, Sylvain Le Corff, Charles Ollion, and Eric Moulines. Diffusion bridges vector quantized variational autoencoders. *arXiv preprint arXiv:2202.04895*, 2022.

[11] Guillaume Couairon, Jakob Verbeek, Holger Schwenk, and Matthieu Cord. Diffedit: Diffusion-based semantic image editing with mask guidance. In *International Conference on Learning Representations*, 2023.

[12] Hanjun Dai, Rishabh Singh, Bo Dai, Charles Sutton, and Dale Schuurmans. Learning discrete energy-based models via auxiliary-variable local exploration. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

[13] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. In *Advances in Neural Information Processing Systems*, volume 34, pages 8780–8794, 2021.

[14] Patrick Esser, Robin Rombach, Andreas Blattmann, and Bjorn Ommer. Imagebart: Bidirectional context with multinomial diffusion for autoregressive image synthesis. *Advances in Neural Information Processing Systems*, 34:3518–3532, 2021.

[15] William Feller. On the theory of stochastic processes, with particular reference to applications. In *Selected Papers I*, pages 769–798. Springer, 2015.

[16] Shansan Gong, Mukai Li, Jiangtao Feng, Zhiyong Wu, and LingPeng Kong. Diffuseq: Sequence to sequence text generation with diffusion models. *arXiv preprint arXiv:2210.08933*, 2022.

[17] 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.

[18] Shuyang Gu, Dong Chen, Jianmin Bao, Fang Wen, Bo Zhang, Dongdong Chen, Lu Yuan, and Baining Guo. Vector quantized diffusion model for text-to-image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10696–10706, 2022.

[19] William Harvey, Saeid Naderiparizi, Vaden Masrani, Christian Weilbach, and Frank Wood. Flexible diffusion modeling of long videos. In *Advances in Neural Information Processing Systems*, volume 35, pages 27953–27965, 2022.

[20] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-or. Prompt-to-prompt image editing with cross-attention control. In *International Conference on Learning Representations*, 2023.

[21] Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022.

[22] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

[23] Jonathan Ho, Chitwan Saharia, William Chan, David J. Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded diffusion models for high fidelity image generation. *Journal of Machine Learning Research*, 23(47):1–33, 2022.

[24] Jonathan Ho, Tim Salimans, Alexey A. Gritsenko, William Chan, Mohammad Norouzi, and David J. Fleet. Video diffusion models. In *Advances in Neural Information Processing Systems*, 2022.

[25] Emiel Hoogeboom, Alexey A Gritsenko, Jasmijn Bastings, Ben Poole, Rianne van den Berg, and Tim Salimans. Autoregressive diffusion models. *arXiv preprint arXiv:2110.02037*, 2021.

[26] Emiel Hoogeboom, Didrik Nielsen, Priyank Jaini, Patrick Forré, and Max Welling. Argmax flows and multinomial diffusion: Learning categorical distributions. *Advances in Neural Information Processing Systems*, 34:12454–12465, 2021.

[27] Aapo Hyvärinen and Peter Dayan. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(4), 2005.

[28] Daniel D Johnson, Jacob Austin, Rianne van den Berg, and Daniel Tarlow. Beyond in-place corruption: Insertion and deletion in denoising probabilistic models. *arXiv preprint arXiv:2107.07675*, 2021.

[29] Richard Jordan, David Kinderlehrer, and Felix Otto. The variational formulation of the fokker–planck equation. *SIAM journal on mathematical analysis*, 29(1):1–17, 1998.

[30] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *arXiv preprint arXiv:2206.00364*, 2022.

[31] Valentin Khrulkov, Gleb Ryzhakov, Andrei Chertkov, and Ivan Oseledets. Understanding DDPM latent codes through optimal transport. In *International Conference on Learning Representations*, 2023.

[32] Serge Lang. *Fundamentals of differential geometry*, volume 191. Springer Science & Business Media, 2012.

[33] Hugo Lavenant and Filippo Santambrogio. The flow map of the fokker–planck equation does not provide optimal transport. *Applied Mathematics Letters*, 133:108225, 2022.

[34] Xiang Li, John Thickstun, Ishaan Gulrajani, Percy S Liang, and Tatsunori B Hashimoto. Diffusion-lm improves controllable text generation. In *Advances in Neural Information Processing Systems*, volume 35, pages 4328–4343. Curran Associates, Inc., 2022.

[35] Zhen Liu, Yao Feng, Michael J. Black, Derek Nowrouzezahrai, Liam Paull, and Weiyang Liu. Meshdiffusion: Score-based generative 3d mesh modeling. In *International Conference on Learning Representations*, 2023.

[36] Chenlin Meng, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Image synthesis and editing with stochastic differential equations. *arXiv preprint arXiv:2108.01073*, 2021.

[37] Alexander Quinn Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob Mcgrew, Ilya Sutskever, and Mark Chen. GLIDE: Towards photorealistic image generation and editing with text-guided diffusion models. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162, pages 16784–16804, 2022.

[38] Jorge Nocedal and Stephen J Wright. *Numerical optimization*. Springer, 1999.

[39] Bernt Oksendal. *Stochastic differential equations: an introduction with applications*. Springer Science & Business Media, 2013.

[40] LS PONTRJAGIN. The mathematical theory of optimal processes. *Interscience*, 1962.

[41] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022.

[42] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022.

[43] Ludan Ruan, Yiyang Ma, Huan Yang, Huiguo He, Bei Liu, Jianlong Fu, Nicholas Jing Yuan, Qin Jin, and Baining Guo. Mm-diffusion: Learning multi-modal diffusion models for joint audio and video generation. *arXiv preprint arXiv:2212.09478*, 2022.

[44] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, Jonathan Ho, David J Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. In *Advances in Neural Information Processing Systems*, volume 35, pages 36479–36494, 2022.

[45] Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J. Fleet, and Mohammad Norouzi. Image super-resolution via iterative refinement. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(4):4713–4726, 2023.

[46] Filippo Santambrogio. Optimal transport for applied mathematicians. *Birkäuser, NY*, 55(58-63):94, 2015.

[47] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pages 2256–2265. PMLR, 2015.

[48] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021.

[49] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019.

[50] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

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

[52] Haoru Tan, Sitong Wu, and Jimin Pi. Semantic diffusion network for semantic segmentation. In *Advances in Neural Information Processing Systems*, volume 35, pages 8702–8716, 2022.

[53] Guy Tevet, Sigal Raab, Brian Gordon, Yoni Shafir, Daniel Cohen-or, and Amit Haim Bermano. Human motion diffusion model. In *International Conference on Learning Representations*, 2023.

[54] Cédric Villani et al. *Optimal transport: old and new*, volume 338. Springer, 2009.

[55] Vikram Voleti, Alexia Jolicoeur-Martineau, and Chris Pal. Mcvd - masked conditional video diffusion for prediction, generation, and interpolation. In *Advances in Neural Information Processing Systems*, volume 35, pages 23371–23385, 2022.

[56] Qiang Wang, Haoge Deng, Yonggang Qi, Da Li, and Yi-Zhe Song. Sketchknitter: Vectorized sketch generation with diffusion models. In *International Conference on Learning Representations*, 2023.

[57] Sirui Xu, Yu-Xiong Wang, and Liangyan Gui. Stochastic multi-person 3d motion forecasting. In *International Conference on Learning Representations*, 2023.

[58] Kôsaku Yosida and JG Taylor. Functional analysis. *Physics Today*, 20(1):127–129, 1967.

[59] Xiaohui Zeng, Arash Vahdat, Francis Williams, Zan Gojcic, Or Litany, Sanja Fidler, and Karsten Kreis. Lion: Latent point diffusion models for 3d shape generation. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

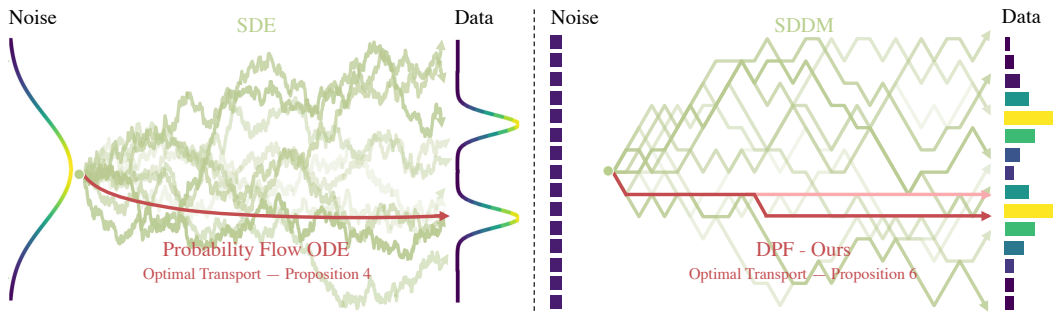
[60] Dinghuai Zhang, Nikolay Malkin, Zhen Liu, Alexandra Volokhova, Aaron Courville, and Yoshua Bengio. Generative flow networks for discrete probabilistic modeling. In *International Conference on Machine Learning*, volume 162, pages 26412–26428, 2022.

Appendix

Contents

A Overview of our DPF

To elucidate our methodology more intuitively, we include schematic diagrams in Figure 5, illustrating the sampling procedure from various diffusion models. Broadly speaking, diffusion models can be classified into two categories based on the nature of the underlying data space: continuous diffusion models and discrete diffusion models. Figure 5 (a) provides an illustration of a continuous diffusion model using a Stochastic Differential Equation (SDE) that transforms a prior noise distribution into



(a) Continuous Diffusion Model (b) Discrete Diffusion Model

Figure 5: Schematic representation of different diffusion models.

the data distribution. The stochastic nature of the sampling process in continuous diffusion models allows samples generated from a single initial point to span the entire space (green line), but this feature limits its practical applicability. To overcome this limitation, probability flow is introduced, which ensures that the generated sample from an initial point follows a deterministic path (red line). This enhancement enables the continuous diffusion model to be more manageable and applicable in a broader range of scenarios.

In this paper, our concentration is primarily on discrete diffusion models. An example of such a model, based on SDDM with 15 states, is depicted in Figure 5 (b). Similar to SDE, it is observed that the sampling process is also susceptible to uncertainty (green line). One potential solution could involve incorporating probability flow into discrete diffusion in a similar manner as in the continuous models. Nonetheless, as previously mentioned in the introduction, this is not a viable option in discrete models due to the lack of a deterministic mapping between the latent space and the data space. Thus, there is a necessity for a redefined probability flow that is tailored to discrete diffusion models, and this forms the core of this paper. This study examines the probability flow of discrete diffusion models through the concept of optimal transport. Initially, we demonstrate that the continuous probability flow coincides with the Monge optimal transport map (Proposition 4). We then leverage this result to develop a similar probability flow for discrete diffusion models using optimal transport (Proposition 6). Finally, we propose a novel sampling methodology for discrete models that significantly reduces the uncertainty (red line) in the sampling process.

B Definitions and Theorems Employed in this Appendix

For the sake of reader convenience, we hereby provide a comprehensive list of the definitions and theorems utilized in this paper. Additionally, we limit our representation to the case within $\mathbb{R}^n$.

Theorem 1. (Theorem 1.48 in [46]) *Suppose that μ is a probability measure on $(\mathbb{R}^n, \mathcal{B})$ such that $\int |x|^2 \, \mathrm{d}\mu(x) < \infty$ and that $u : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ is convex and differentiable μ-a.e. Set $\mathrm{T} = \nabla u$ and suppose $\int |\mathrm{T}(x)|^2 \, \mathrm{d}\mu(x) < \infty$. Then T is optimal for the transport cost $c(x, y) = \frac{1}{2}|x - y|^2$ between the measures μ and $\nu = \mathrm{T}_{\#}\mu$.*

Definition 2. *The optimization problem under constraint is formally defined as follows:*

$$\min_{x \in \mathbb{R}^n} f(x) \quad \text{subject to} \begin{cases} c_i(x) = 0, i \in \mathcal{E} \\ c_i(x) \geq 0, i \in \mathcal{I}. \end{cases} \tag{34}$$

The Lagrangian for this constrained optimization problem is defined as:

$$\mathcal{L}(x, \lambda) = f(x) - \sum_{i \in \mathcal{E} \cup \mathcal{I}} \lambda_i c_i(x). \tag{35}$$

Here, λ_i represents the Lagrange multiplier associated with the i^{th} constraint. The active set at any feasible x is defined as the union of the set $\mathcal{E}$ with the indices of the active inequality constraints, that is:

$$\mathcal{A}(x) = \mathcal{E} \cup \{i \in \mathcal{I} : c_i(x) = 0\}. \tag{36}$$

Definition 3. (Definition 12.1 in [38]) *Given the point x^*, we say that the Linear Independence Constraint Qualification (LICQ) holds if the set of active constraint gradients $\{\nabla c_i(x^*), i \in \mathcal{A}(x^*)\}$ is linearly independent.*

Theorem 4. (Theorem 12.1 in [38], the Karush-Kuhn-Tucker (KKT) conditions) *Suppose that x^* is a local solution of the problem (34) and that the LICQ holds at x^*. Then there is a Lagrange multiplier vector λ^* , with components $\lambda_i^*, i \in \mathcal{E} \cup \mathcal{I}$, such that the following conditions are satisfied at (x^*, λ^*)*

$$\nabla_x \mathcal{L}(x^*, \lambda^*) = 0, \tag{37a}$$
$$c_i(x^*) = 0, \quad \forall i \in \mathcal{E}, \tag{37b}$$
$$c_i(x^*) \geq 0, \quad \forall i \in \mathcal{I}, \tag{37c}$$
$$\lambda^* \geq 0, \quad \forall i \in \mathcal{I}, \tag{37d}$$
$$\lambda_i^* c_i(x^*) = 0, \quad \forall i \in \mathcal{E} \cup \mathcal{I}. \tag{37e}$$

Remark 5. According to Theorem 4, the Karush-Kuhn-Tucker (KKT) conditions serve as necessary conditions. In the case of linear programming, these conditions are not only necessary but also sufficient. To demonstrate this, let us consider the standard form of a linear programming problem:

$$\min c^T x, \quad \text{subject to } Ax = b, x \geq 0. \tag{38}$$

We can write the Lagrangian function for 38 as

$$\mathcal{L}(x, \pi, s) = c^T x - \pi^T (Ax - b) - s^T x. \tag{39}$$

The KKT conditions are

$$A^T \pi + s = c, \tag{40a}$$
$$Ax = b, \tag{40b}$$
$$x \geq 0, \tag{40c}$$
$$s \geq 0, \tag{40d}$$
$$x^T s = 0. \tag{40e}$$

Suppose we have a vector triple (x^*, π^*, s^*) that satisfies Equation (40). In such a scenario, we can deduce that:

$$c^T x^* = (A^T \pi^* + s)^T x^* = (\pi^*)^T A x^* = b^T \pi^*. \tag{41}$$

Let us consider another feasible point denoted by $\bar{x}$, which satisfies the conditions $A\bar{x} = b$ and $\bar{x} \geq 0$. we can conclude that:

$$c^T \bar{x} = (A^T \pi^* + s^*)^T \bar{x} = b^T \pi^* + \bar{x}^T s^* \geq b^T \pi^* = c^T x^*. \tag{42}$$

The inequality (42) demonstrates that the KKT conditions serve as sufficient conditions.

Theorem 6. (Theorem 8.5.1 in [39]) *Let X_t be an Itô diffusion given by*

$$\mathrm{d}X_t = b(X_t)\,\mathrm{d}t + \sigma(X_t)\,\mathrm{d}B_t, \;\; b \in \mathbb{R}^n, \sigma \in \mathbb{R}^{n \times m}, X_0 = x, \tag{43}$$

and let Y_t be an Itô process given by

$$\mathrm{d}Y_t = u(t, \omega)\,\mathrm{d}t + v(t, \omega)\,\mathrm{d}B_t, \;\; u \in \mathbb{R}^n, v \in \mathbb{R}^{n \times m}, Y_0 = x. \tag{44}$$

Assume that

$$u(t, \omega) = c(t, \omega) b(Y_t) \;\; \textit{and} \;\; vv^T(t, \omega) = c(t, \omega) \sigma \sigma^T(Y_t), \tag{45}$$

for a.a. t, ω. Define β_t and α_t as:

$$\beta_t = \beta(t, \omega) = \int_0^t c(s, \omega)\,\mathrm{d}s \;\; \textit{and} \;\; \alpha_t = \inf\{s : \beta_s > t\}. \tag{46}$$

Then Y_{α_t} coincides in law with X_t, denoted by $Y_{\alpha_t} \simeq X_t$.

Theorem 7. (Theorem 4.1 of Chapter V, §4 in [32], Poincaré's lemma). *Let U be an open ball in $\mathbb{R}^n$ and let ω be a differential form of degree ≥ 1 on U such that $\mathrm{d}\omega = 0$. Then there exists a differential form ϕ on U such that $\mathrm{d}\phi = \omega$.*

Remarks. The conclusion remains valid when the open ball U is substituted with the entirety of $\mathbb{R}^n$.

Theorem 8. (Theorem 4 in [5]) *The solution of the differential equation $Y' = A(t)Y$ with initial condition $Y(0) = Y_0$ can be written as $Y(t) = \exp(\Omega(t))Y_0$ with $\Omega(t)$ defined by*

$$\Omega' = \mathrm{d}\exp_\Omega^{-1}(A(t)), \;\; \Omega(0) = O. \tag{47}$$

where

$$\mathrm{d}\exp_\Omega^{-1}(A(t)) = \sum_0^\infty \frac{B_k}{k!}\mathrm{ad}_\Omega^k(A), \tag{48}$$

and B_k is the Bernoulli numbers. $\mathrm{ad}_\Omega^k(A)$ is defined through

$$\mathrm{ad}_\Omega(A) = [\Omega, A], \;\; \mathrm{ad}_\Omega^j(A) = [\Omega, \mathrm{ad}_\Omega^{j-1}(A)], \;\; \mathrm{ad}_\Omega^0(A) = A, \quad j \in \mathbb{N}, \tag{49}$$

where $[A, B] = AB - BA$ is the Lie-bracket.

Remarks. If $A(s)A(t) = A(t)A(s), \forall s, t \geq 0$, $\Omega(t)$ has the simple form $\Omega(t) = \int_0^t A(s)\,\mathrm{d}s$.

C Proofs

C.1 Proof of Proposition 1

Proof. By Itô formula:

$$\begin{aligned}
\mathrm{d}(\phi_t X_t) =& \phi_t\theta_t X_t\,\mathrm{d}t + \phi_t\,\mathrm{d}X_t \\
=& \phi_t\theta_t X_t\,\mathrm{d}t - \phi_t\theta_t X_t\,\mathrm{d}t + \phi_t\sigma_t\,\mathrm{d}B_t \\
=& \phi_t\sigma_t\,\mathrm{d}B_t.
\end{aligned} \tag{50}$$

By Theorem 6, $\phi_{\alpha_t}X_{\alpha_t} \simeq Y_t$, which means X_t coincides in law with $\phi_t^{-1}Y_{\beta_t}$ □

Remarks. Proposition 1 posits that the Ornstein-Uhlenbeck (OU) process is essentially a scaling of Brownian motion with a change in time. Consequently, the VE SDEs, VP SDEs, sub-VP SDEs in [50] , as well as the models presented in [30], can be regarded as equivalent.

C.2 Proof of Proposition 2

Lemma B.2.1 *Let H_t be the Hessian matrices $\nabla_{x_t}^2 \log p_B(x_t, t)$, then $H_s H_t = H_t H_s, \forall s, t \geq 0$.*

Proof.

$$\begin{aligned}
H_t =& \nabla_{x_t}^2 \log p_B(x_t, t) \\
=& \nabla_{x_t} \frac{\sum_i \exp(-\frac{|x_t - x_i|^2}{2t})(-\frac{x_t - x_i}{t})}{\sum_j \exp(-\frac{|x_t - x_j|^2}{2t})} \\
=& \underbrace{\sum_i \nabla_{x_t}(\frac{\exp(-\frac{|x_t - x_i|^2}{2t})}{\sum_j \exp(-\frac{|x_t - x_j|^2}{2t})})(-\frac{x_t - x_i}{t})}_{A} + \underbrace{\frac{\sum_i \exp(-\frac{|x_t - x_i|^2}{2t})}{\sum_j \exp(-\frac{|x_t - x_j|^2}{2t})}(-\frac{1}{t})I}_{B}.
\end{aligned} \tag{51}$$

B is a scalar matrix, then it commutes with any matrix.

$$
\begin{aligned}
A =& \underbrace{(\sum_j \exp(-\frac{|x_t - x_j|^2}{2t}))^{-2}}_{C} \sum_i [\exp(-\frac{|x_t - x_i|^2}{2t})(-\frac{x_t - x_i}{t}) \sum_j \exp(-\frac{|x_t - x_j|^2}{2t}) \\
& - \exp(-\frac{|x_t - x_i|^2}{2t}) \sum_j \exp(-\frac{|x_t - x_j|^2}{2t})(-\frac{x_t - x_j}{t})](-\frac{x_t - x_i}{t})^T \\
=& C \sum_{i,j} \exp(-\frac{|x_t - x_i|^2}{2t} - \frac{|x_t - x_j|^2}{2t})(\frac{x_t - x_i}{t})(\frac{x_t - x_i}{t})^T \\
& - C \sum_{i,j} \exp(-\frac{|x_t - x_i|^2}{2t} - \frac{|x_t - x_j|^2}{2t})(\frac{x_t - x_j}{t})(\frac{x_t - x_i}{t})^T \\
=& C \sum_{i<j} \exp(-\frac{|x_t - x_i|^2}{2t} - \frac{|x_t - x_j|^2}{2t}) \frac{1}{t^2} [(x_t - x_i)(x_t - x_i) + (x_t - x_j)(x_t - x_j) \\
& - (x_t - x_j)(x_t - x_i)^T - (x_t - x_i)(x_t - x_j)^T] \\
=& C \sum_{i<j} \exp(-\frac{|x_t - x_i|^2}{2t} - \frac{|x_t - x_j|^2}{2t}) \frac{1}{t^2} (x_j - x_i)(x_j - x_i)^T.
\end{aligned}
\tag{52}
$$

As x_is lie on the same line, $x_j - x_i$ can be denoted by $x_j - x_i = C_{i,j} v, \forall i, j$, where v is a fixed vector. It has $(x_j - x_i)(x_j - x_i)^T = C_{i,j}^2 v v^T$. It is clear that $C_{i,j}^2 v v^T$ and $C_{k,l}^2 v v^T$ commutes $\forall i, j, k, l$. Furthermore, as t and x_t only appear in the coefficients, H_t and H_s commute with one another. □

Proof of Proposition2. If Y_0 follows the initial condition (18), and x_is lie on the same line, Y_t will governed by the equation (20). Employing the trick in [8], For a fixed T, we define

$$
a(t) = \nabla_{\hat{Y}_t} \hat{Y}_T. \tag{53}
$$

Then we can derive

$$
\begin{aligned}
\frac{\mathrm{d}a(t)}{\mathrm{d}t} &= \lim_{\epsilon \to 0^+} \frac{a(t+\epsilon) - a(t)}{\epsilon} \\
&= \lim_{\epsilon \to 0^+} \frac{a(t+\epsilon) - a(t+\epsilon) \nabla_{\hat{Y}_t} (\hat{Y}_t - \epsilon \frac{1}{2} \nabla_{\hat{Y}_t} \log p_B(\hat{Y}_t, t) + \mathcal{O}(\epsilon^2))}{\epsilon} \\
&= \lim_{\epsilon \to 0^+} \frac{\epsilon a(t+\epsilon) \nabla_{\hat{Y}_t}^2 \log p_B(\hat{Y}_t, t) + \mathcal{O}(\epsilon^2)}{2\epsilon} \\
&= \frac{1}{2} a(t) \nabla_{\hat{Y}_t}^2 \log p_B(\hat{Y}_t, t),
\end{aligned}
\tag{54}
$$

where ∇^2 is the *Hessian operator*. Based on Lemma B.2.1, theorem 8 and the fact that $a(T) = \nabla_{\hat{Y}_T} \hat{Y}_T = I$, $a(t) = \nabla_{\hat{Y}_t} \hat{Y}_T$ is symmetric. Then theorem 7 shows that the equation $\nabla_{\hat{Y}_t} u(\hat{Y}_t) = \hat{Y}_T(\hat{Y}_t)$ has a solution. Furthermore, since $a(t)$ is a matrix exponential of a symmetric matrix, it must be positive semi-definite. Consequently, the solution u is convex. According to theorem 1, the map $\hat{Y}_T(\hat{Y}_t)$ is optimal for the quadratic transport cost. □

C.3 Proof of Proposition 3

Proof. The definition of $\tilde{Y}_{t+s}$ is given by Equation (23). It can be observed that the term $\tilde{Y}_t(\tilde{Y}_{t+s})$ on the right-hand side indicates that the evolution speed of $\tilde{Y}_{t+s}$ is constant, which implies that all particles travel at a uniform rate. Consequently, for a given initial condition $\tilde{Y}_t$,

$$
\tilde{Y}_{t+s} = \tilde{Y}_t - \frac{1}{2} \nabla_{\tilde{Y}_t} \log p_B(\tilde{Y}_t) s. \tag{55}
$$

Further, we have:

$$
\nabla_{\tilde{Y}_t} \tilde{Y}_{t+s} = I - \frac{1}{2} \nabla_{\tilde{Y}_t}^2 \log p_B(\tilde{Y}_t) s. \tag{56}
$$

It is evident that $\nabla_{\tilde{Y}_t}\tilde{Y}_{t+s}$ is symmetric and for small values of s, it is also positive semi-definite. Based on the same reasoning as the proof of Proposition 2, we can conclude that the map $\tilde{Y}_{t+s}(\tilde{Y}_t)$ is optimal for the quadratic transport cost. □

C.4 Proof of Proposition 4

proof. As Proposition 1 establishes that $X_t = \phi_t^{-1} Y_{\beta_t}$, the single-time marginal distribution $p_{OU}(x_t, t)$ for the Ornstein-Uhlenbeck process can be expressed as follows:

$$p_{OU}(xt, t) = \frac{1}{N}\sum_i^N (2\pi\beta_t\phi_t^{-2})^{-\frac{n}{2}} \exp(-\frac{|x_t - \phi_t^{-1}x_i|^2}{2\beta_t\phi_t^{-2}}). \tag{57}$$

The probability flow ODE for Ornstein-Uhlenbeck process is:

$$\mathrm{d}\hat{X}_t = [-\theta_t\hat{X}_t - \frac{1}{2}\sigma_t^2\nabla_{\hat{X}_t}\log p_{OU}(\hat{X}_t, t)]dt. \tag{58}$$

We start from $\hat{Y}_t$ with the change of variable $Z_t = \phi_t^{-1}\hat{Y}_{\beta_t}$:

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t}Z_t =& \frac{\mathrm{d}\phi_t^{-1}}{\mathrm{d}t}\hat{Y}_{\beta_t} + \phi_t^{-1}\frac{\mathrm{d}\hat{Y}_{\beta_t}}{\mathrm{d}\beta_t}\frac{\mathrm{d}\beta_t}{\mathrm{d}t} \\
=& -\phi_t^{-1}\theta_t\hat{Y}_{\beta_t} + \phi_t^{-1}(\sigma_t\phi_t)^2[-\frac{1}{2}\nabla_{\hat{Y}_{\beta_t}}p_B(\hat{Y}_t, \beta_t)] \\
=& -\theta_t Z_t - \frac{1}{2}\phi_t\sigma_t^2\frac{\sum_i \exp(-\frac{|\hat{Y}_{\beta_t}-x_i|^2}{2\beta_t})\frac{\hat{Y}_{\beta_t}-x_i}{\beta_t}}{\sum_j \exp(-\frac{|\hat{Y}_{\beta_t}-x_j|^2}{2\beta_t})} \\
=& -\theta_t Z_t - \frac{1}{2}\sigma_t^2\frac{\sum_i \exp(-\frac{|\phi_t^{-1}\hat{Y}_{\beta_t}-\phi_t^{-1}x_i|^2}{2\beta_t\phi_t^{-2}})\frac{\phi_t^{-1}\hat{Y}_t-\phi_t^{-1}x_i}{\beta_t\phi_t^{-2}}}{\sum_j \exp(-\frac{|\phi_t^{-1}\hat{Y}_{\beta_t}-\phi_t^{-1}x_j|^2}{2\beta_t\phi_t^{-2}})} \\
=& -\theta_t Z_t - \frac{1}{2}\sigma_t^2\frac{\sum_i \exp(-\frac{Z_t-\phi_t^{-1}x_i}{2\beta_t\phi_t^{-2}})\frac{Z_t-\phi_t^{-1}x_i}{\beta_t\phi_t^{-2}}}{\sum_j \exp(-\frac{|Z_t-\phi_t^{-1}x_j|^2}{2\beta_t\phi_t^{-2}})} \\
=& -\theta_t Z_t - \frac{1}{2}\sigma_t^2\nabla_{Z_t}\log p_{OU}(Z_t, t).
\end{aligned} \tag{59}$$

As $\phi_0 = 1$ and $\beta_0 = 0$, the initial distribution of X_0 and Z_0 is the same. Consequently, $\hat{X}_t$ and Z_t follow the same ODE with identical initial conditions. Thus, we have $\hat{X}_t = Z_t = \phi_t^{-1}\hat{Y}_{\beta_t}$ and $\nabla_{\hat{X}_t}\hat{X}_{t+s} = \frac{\phi_t}{\phi_{t+s}}\nabla_{\hat{Y}_{\beta_t}}\hat{Y}_{\beta_{t+s}}$, which is symmetric and positive semi-definite by Proposition 2. Therefore, we can conclude that the map $\hat{X}_{t+s}(\hat{X}_t)$ is optimal for the quadratic transport cost. □

C.5 Proof of Proposition 5

Proof. For the original process (discrete analogue of Brownian motion), the transition rate is:

$$Q_D{}_j^i = \begin{cases} 1, & d_D(i,j) = 1, \\ \sum\limits_{j\in N(i)} -Q_D{}_j^i, & i = j, \\ 0, & others, \end{cases} \tag{60}$$

where $N(i) = \{k : d_D(i,k) = 1\}$. The Kolmogorov forward equation of this process is written as:

$$\begin{aligned}\frac{dP_{Di}(t)}{dt} =& \sum_{i'} P_{Di'}(t) Q_{D}{}_{i}^{i'} \\ =& P_{Di}(t) \times \sum_{i' \in N(i)} -Q_{D}{}_{i'}^{i} + \sum_{i' \in N(i)} P_{Di'}(t) \times 1 \\ &+ \sum_{i' \in \{k : d_D(i,k) > 1\}} P_{Di'}(t) \times 0 \\ =& \sum_{i' \in N(i)} (P_{Di'}(t) - P_{Di}(t)).\end{aligned} \tag{61}$$

In contrast, the transition rate of our new process is:

$$Q_j^i = \begin{cases} \frac{ReLU(P_{Di}(t) - P_{Dj}(t))}{P_{Di}(t)}, & d_D(i,j) = 1 \\ \sum_{d_D(i,j)=1} -Q_j^i, & i = j \\ 0, & others. \end{cases} \tag{62}$$

Our new process can be written as:

$$\begin{aligned}\frac{dP_i(t)}{dt} =& \sum_{i'} P_{i'}(t) Q_i^{i'} \\ =& P_i(t) \times \sum_{i' \in N(i)} -Q_{i'}^{i} \\ &+ \sum_{i' \in N(i)} P_{i'}(t) \times \frac{ReLU(P_{Di'}(t) - P_{Di}(t))}{P_{Di'}(t)} \\ &+ \sum_{i' \in \{k : d_D(i,k) > 1\}} P_{i'}(t) \times 0 \\ =& - \sum_{i' \in N(i)} P_i(t) \frac{ReLU(P_{Di}(t) - P_{Di'}(t))}{P_{Di}(t)} \\ &+ \sum_{i' \in N(i)} P_{i'}(t) \frac{ReLU(P_{Di'}(t) - P_{Di}(t))}{P_{Di}{}'(t)}.\end{aligned} \tag{63}$$

Substitute $P = P_D$ in Equation (63), we get the same form in 61, which means P_D also solves the Equation (63). Thus, $P_i(t) = P_{Di}(t)$, $\forall t \geq 0, i \in \{1, 2, \cdots, S\}^K$, according to Picard-Lindelöf theorem. □

C.6 Proof of Proposition 6

Let $a = P(t)$, $b = P(t + \varepsilon)$. As our generator only allows flux between adjacent states, we define the transport map $\Pi^* \in \mathbb{R}^{k \times k}$ as:

$$\Pi^{*}{}_{j}^{i} = \int_t^{t+\epsilon} P_i(t) Q_j^i(t) \, \mathrm{d}t, \tag{64}$$

which is the probability transported from state i to state j in the time interval $[t, t + \epsilon]$. As the probability $P(t)$ is continuous with respect to time t, we choose the ϵ such that the sign of all the quantities $P_i(t) - P_j(t)$ for $\{i, j \in \{1, 2, \cdots, S\}^K : d_D(i,j) = 1\}$ do not change. Under this assumption, the flux directions do not change at every state.

We claim that Π^* solves the optimal transport problem:

$$\min_{\Pi} \sum_{i,j} \Pi^i_j d_D(i,j),$$
$$s.t. \begin{cases} \sum_i \Pi^i = b, \\ \sum_j \Pi_j = a, \\ \Pi \geq 0. \end{cases} \tag{65}$$

Proof. The Lagrangian function for this optimization problem is:

$$L(\Pi, \psi, \phi, \lambda) = \sum_{i,j} \Pi^i_j d_D(i,j) + \psi_i(\Pi^i_j - a_i) + \phi_j(\Pi^i_j - b_j) - \lambda^i_j \Pi^i_j, \tag{66}$$

where ψ_i, ϕ_j and λ^i_j are Lagrange multipliers. According to Remark 5 and the fact that this is a linear programming, Π^* is optimal if and only if there exists a set of ϕ_i, ψ_j and λ^i_j that satisfy the following equations for $\forall\, i, j$:

$$d_D(i,j) + \psi_i + \phi_j - \lambda^i_j = 0 \tag{67a}$$
$$\lambda^i_j \geq 0 \tag{67b}$$
$$\lambda^i_j \Pi^i_j = 0 \tag{67c}$$
$$\sum_i \Pi^i_j = b_j \tag{67d}$$
$$\sum_j \Pi^i_j = a_i \tag{67e}$$
$$\Pi^i_j \geq 0. \tag{67f}$$

Firstly, we consider the i, j pairs where $d_D(i,j) \leq 1$. In this case ${\Pi^*}^i_j$ may > 0 (thus $\lambda^i_j = 0$). Besides, the Equation (64) indicates that ${\Pi^*}^i_i > 0$, thus we have $\lambda^i_i = 0$. Then the Equation (67a) comes to:

$$\psi_i + \phi_i = 0. \tag{68}$$

According to the construction of our generator Q, there is no mutual flux, thus we obtain:

$$\Pi^{i_1,\ldots,i_l,\ldots,i_K}_{i_1,\ldots,i_l+1,\ldots,i_K} \neq 0 \text{ or } \Pi^{i_1,\ldots,i_l+1,\ldots,i_K}_{i_1,\ldots,i_l,\ldots,i_K} \neq 0, \ \ \forall\, i. \tag{69}$$

By substituting this result into complementary slackness condition 67c, we have:

$$\lambda^{i_1,\ldots,i_l,\ldots,i_K,}_{i_1,\ldots,i_l+1,\ldots,i_K} = 0 \text{ or } \lambda^{i_1,\ldots,i_l+1,\ldots,i_K}_{i_1,\ldots,i_l,\ldots,i_K} = 0. \tag{70}$$

Since $d_D([i_1,\ldots,i_l,\ldots,i_K],[i_1,\ldots,i_l+1,\ldots,i_K]) = 1$, from Equation (67a), we can obtain:

$$1 + \psi_{i_1,\ldots,i_l,\ldots,i_K} + \phi_{i_1,\ldots,i_l+1,\ldots,i_K} = 0 \text{ or } 1 + \psi_{i_1,\ldots,i_l+1,\ldots,i_K} + \phi_{i_1,\ldots,i_l,\ldots,i_K} = 0. \tag{71}$$

Solving Equations (68) and (71) simultaneously, we get:

$$\begin{cases} \psi_{i_1,\ldots,i_l+1,\ldots,i_K} = 1 + \psi_{i_1,\ldots,i_l,\ldots,i_K} \\ \phi_{i_1,\ldots,i_l+1,\ldots,i_K} = -1 + \phi_{i_1,\ldots,i_l,\ldots,i_K} \end{cases} \text{ or } \begin{cases} \psi_{i_1,\ldots,i_l+1,\ldots,i_K} = -1 + \psi_{i_1,\ldots,i_l,\ldots,i_K} \\ \phi_{i_1,\ldots,i_l+1,\ldots,i_K} = 1 + \phi_{i_1,\ldots,i_l,\ldots,i_K} \end{cases}. \tag{72}$$

Therefore, given $\psi_{0,\ldots,0}$, $\psi_{i_1,\ldots,i_K}$ and $\phi_{i_1,\ldots,i_K}$ can be calculated by:

$$\psi_{i_1,\ldots,i_K} = \psi_{0,\ldots,0} + m^{i_1,\ldots,i_K}_{0,\ldots,0} - n^{i_1,\ldots,i_K}_{0,\ldots,0}, \tag{73}$$
$$\phi_{i_1,\ldots,i_K} = -\psi_{i_1,\ldots,i_K}, \tag{74}$$

where $m^{i_1,\ldots,i_K}_{0,\ldots,0} + n^{i_1,\ldots,i_K}_{0,\ldots,0} = d_D(0,i)$, $m^{i_1,\ldots,i_K}_{0,\ldots,0} \in \mathbb{N}_0$, and $n^{i_1,\ldots,i_K}_{0,\ldots,0} \in \mathbb{N}_0$. $\mathbb{N}_0$ represents the set of all non-negative integers. The quantity $m^{i_1,\ldots,i_K}_{0,\ldots,0}$ is the number where $\Pi^{i_1,\ldots,i_l,\ldots,i_K}_{i_1,\ldots,i_l+1,\ldots,i_K} \neq 0$

Algorithm 1: Generative Reverse Process with Discrete Probability Flow (DPF)

$t \leftarrow T$

$i_t^{1:K} \sim P_T(i_T^{1:K})$

while $t > 0$ **do**

Compute matrix $P_D^\theta = [P_D^\theta[l,j]]_{K\times S}$, where $P_D^\theta[l,j] = P^\theta_{D\, i^l_{t-\tau}=j|i_t\setminus i^l_t}(t)$, $l = 1,\dots,K$, $j = 1,\dots,S$ with softmax operation on the results of $K \times S$ forward pass of the model

Encoded the three candidate states (i.e., i_t^l, $i_t^l - 1$ and $i_t^l + 1$) for $i^l_{t-\tau}$ with one-hot code: $O_{stay} \leftarrow I_{K\times K}[i_t]$; $O_{sub} \leftarrow I_{K\times K}[i_t - 1]$; $O_{add} \leftarrow I_{K\times K}[i_t + 1]$

Fetch the probability P_D^θ for the above candidate state: $P_{stay} \leftarrow \sum_j (O_{stay} \circ P_D^\theta)$; $P_{sub} \leftarrow \sum_j (O_{sub} \circ P_D^\theta)$; $P_{add} \leftarrow \sum_j (O_{add} \circ P_D^\theta)$

$R^{i_t}_{i_{t-\tau}}(t) \leftarrow O_{sub} \circ ReLU(P_{sub}/P_{stay} - 1) + O_{add} \circ ReLU(P_{add}/P_{stay} - 1) - O_{stay} \circ (ReLU(P_{sub}/P_{stay} - 1) + ReLU(P_{add}/P_{stay} - 1))$ with Equation (30)

$P^\theta(i^l_{t-\tau}|i_t) \leftarrow \tau R^{i_t}_{i_{t-\tau}}(t) + O_{stay}$ with Equation (31)

$i_{t-\tau} \leftarrow \text{Categorical}\Big(P^\theta(i^l_{t-\tau}|i_t)\Big)$

$t \leftarrow t - \tau$

end

and $n^{i_1,\dots,i_K}_{0,\dots,0}$ is the number where $\Pi^{i_1,\dots,i_l+1,\dots,i_K}_{i_1,\dots,i_l,\dots,i_K} \neq 0$. Consequently, we find all the Lagrange multipliers for $d_D(i,j) \leq 1$

Then, we consider i, j pairs when $d_D(i,j) > 1$ which indicates $\Pi^{*i}_{\ j} = 0$. We use ψ_i and ϕ_j in Equation (73) and (74). To satisfy the KKT condition, we only need to verify that there is λ^i_j satisfies Equation (67b) and Equation (67a). From Equation (67a), λ^i_j can be written as:

$$\lambda^i_j = d_D(i,j) + \psi_i + \phi_j \tag{75}$$

Let $r_l = \min(i_l, j_l)$, it has:

$$\begin{aligned}
d_D(i,j) &= d_D(i,r) + d_D(j,r) && (76)\\
\psi_i &= \psi_r + m^r_i - n^r_i && (77)\\
\phi_j &= -\psi_j = -\psi_r - m^r_j + n^r_j && (78)\\
d_D(i,r) &= m^r_i + n^r_i && (79)\\
d_D(j,r) &= m^r_j + n^r_j && (80)\\
m^r_i, n^r_i, m^r_j, n^r_j &\geq 0. && (81)
\end{aligned}$$

Substitute the above results to Equation (75), we have:

$$\lambda^i_j = 2(m^r_i + n^r_j) \geq 0 \tag{82}$$

As a result, we find all the Lagrange multipliers. Since Equations (67d) and (67e) are naturally satisfied by the construction of Π^*, we conclude that the KKT conditions are met at Π^*:

① Primal Feasibility: (67d), (67e), (67f)

② Dual Feasibility: (67a), (67b)

③ Complementary slackness: (67c)

According to Remark 5, the KKT conditions indicate Π^* is a solution to the optimal transport problem (65). □

C.7 Proof of Proposition 7

This is a special case of Proposition 6, where the generator Q remains constant throughout time.

Table 3: Average MMD between different distributions of data.

State size	2	5	10
Average MMD	5.336e-3	2.201e-2	6.531e-3

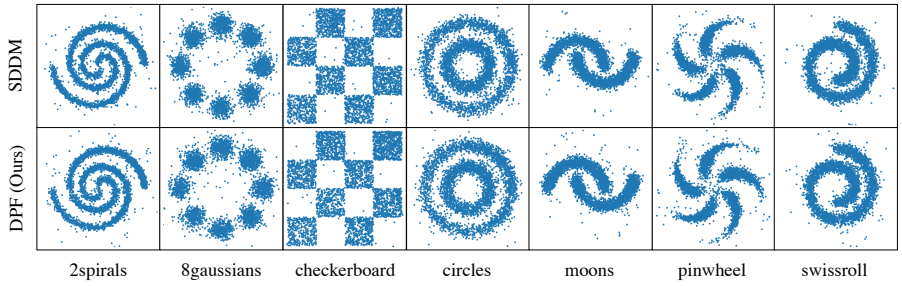


Figure 6: Visualization of the generation quality on generated samples with state size = 5 for SDDM and DPF.

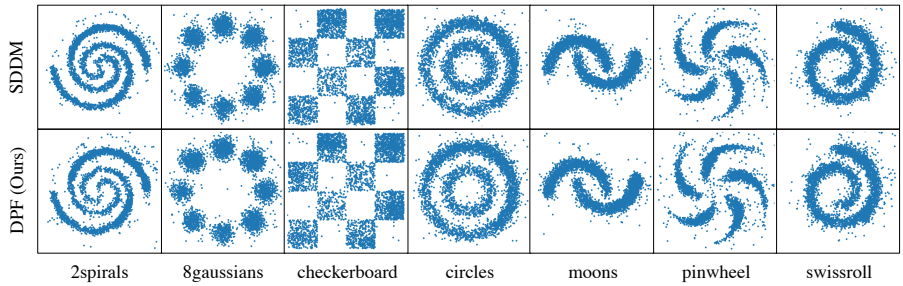


Figure 7: Visualization of the generation quality on generated samples with state size = 10 for SDDM and DPF.

D Experiment

D.1 Algorithm

Our training process follows the same procedure as SDDM, with the distinction that our forward process incorporates the rate we formulated in Equation (28) to align with optimal transport theory. The loss function employed during the training process is as follows:

$$\theta^* = \arg\min_{\theta} \int_0^T \sum_{i_t \in \{1,2,\cdots,S\}^K} q_t(i_t) \left[\sum_{l=1}^{K} -\log P_t(i_t^l | i_t \backslash i_t^l) \right] dt. \tag{83}$$

The sampling process with the proposed discrete probability flow is shown in Algorithm 1. In our algorithm, as R is non-zero only when $d_D(i_t, i_{t-\tau}) \leq 1$, the calculation of the reverse transition rate R (as defined in Equation 30) is divided into three cases: staying in the current state ($i_{t-\tau}^l = i_t^l$, i.e., "stay"), jumping to the next state ($i_{t-\tau}^l = i_t^l + 1$, i.e., "add"), and jumping to the previous state ($i_{t-\tau}^l = i_t^l - 1$, i.e., "sub"). By combining the rate in these situations, we can derive $P_\theta(i_{t-\tau}^l | i_t)$ from (31) and (32), which allows us to sample the next state accordingly. This process continues iteratively until $t = 0$.

D.2 Synthetic Dataset

Following [60, 12, 51], we utilize synthetic data for model validation. Initially, we generate 2D floating-point data from seven distinct distributions using an infinite data oracle. By employing the same settings as [51], we convert each dimension of the data into 16-bit Gray code, resulting in a dataset with discrete dimension = 32 and state size = 2. However, it is not sufficient to validate our method solely on the dataset with state size = 2, since Q in Equation (28) does not cover cases where

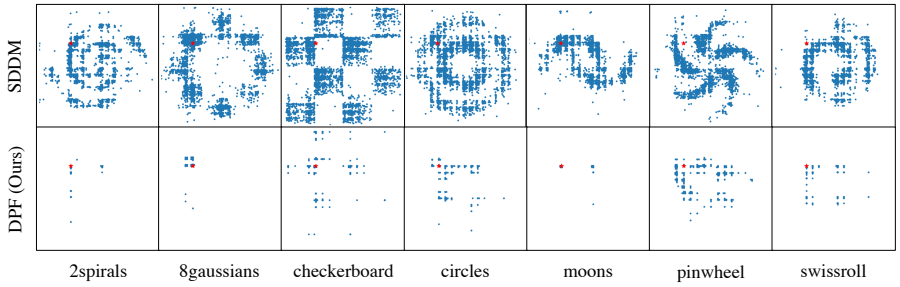


Figure 8: Visualization of the generating certainty on generated samples with state size = 5 for SDDM and DPF. All the samples (in blue) are randomly generated from the single initial point (in red).

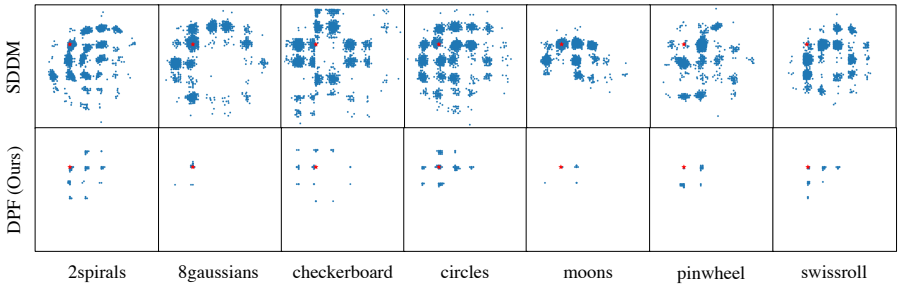


Figure 9: Visualization of the generating certainty on generated samples with state size = 10 for SDDM and DPF. All the samples (in blue) are randomly generated from the single initial point (in red).

$d_D(i,j) > 1$. Therefore, we further transform the same data into 8-bit 5-base code and 6-bit decimal code respectively, thereby creating two additional datasets: one with dimension = 16 and state size = 5, and another with dimension = 12 and state size = 10.

D.3 Experiment Details

In the experiments, our neural network consists of a 3-layer MLP with 256 channels [60, 51]. We employ the Adam optimizer with a learning rate of 1e-4. The model is trained on a single NVIDIA Quadro RTX 8000, utilizing a batch size of 128 for 300,000 iterations. During training, the parameter t is uniformly sampled from the range of 0 to 1. For the sampling process, the data is generated through 1,000 steps (i.e.,τ is set to 0.001).

D.4 Quality of Generated Samples

To evaluate the sampling quality, we generate 40,000 samples for binary data, and 4,000 samples for other type of synthetic data by SDDM and our method. Then we compare these generated samples to true data using Laplace MMD. This evaluation is repeated 10 times, and the average results are presented in Table 1. It is worth noting that the unbiased estimation of MMD [17] is an approximation by Monte Carlo method, which may cause negative results. It is observed that MMD score of our method is slightly higher than that of SDDM. This is mainly caused by the approximation of Q_t. In the sampling process, two terms are present on the right-hand side of Eq. 10: $\frac{q_t(y)}{q_t(x)}$ and $Q_t(y,x)$. In SDDM, only one term, i.e., $\frac{q_t(y)}{q_t(x)}$ is estimated using a neural network, as $Q_{Dt}(y,x)$ is known. Different from SDDM, both terms in our method are evaluated using quantities approximated by the neural network, since our $Q_t(y,x)$ is dependent on $\frac{q_t(y)}{q_t(x)}$ (Eq. 27). This approximation may lead to slightly inferior quality than the SDDM using precise $Q_{Dt}(y,x)$. Due to this being a neural network fitting error, we currently have no feasible alternative approximations to achieve a superior outcome.

To assess the significance of these differences, we presented the MMD between different distributions of real data in Table 3. Taking this result as a reference, we can find that the gap of the MMD score

Table 4: Comparison of the average L_1 distance between the generated samples and initial point. Lower values indicate that the generated sample is closer to initial point.

	2spirals	8gaussians	checkerboard	circles	moons	pinwheel	swissroll
discrete dimension = 32, state size = 2							
SDDM	13.5595	13.3025	13.4710	13.5848	13.6485	13.4875	13.6962
DPF (ours)	1.5965	1.3855	0.7525	1.1875	1.8693	1.8135	1.6955
discrete dimension = 16, state size = 5							
SDDM	12.7220	12.4698	12.4833	12.5390	12.6745	12.6238	12.7510
DPF (ours)	1.5265	1.6155	0.7090	1.1668	1.7038	1.8088	1.5888
discrete dimension = 12, state size = 10							
SDDM	11.3433	11.0083	11.0243	11.2205	11.6850	11.3895	11.6333
DPF (ours)	1.7655	1.1940	0.7143	1.1588	2.0493	1.9283	1.7695

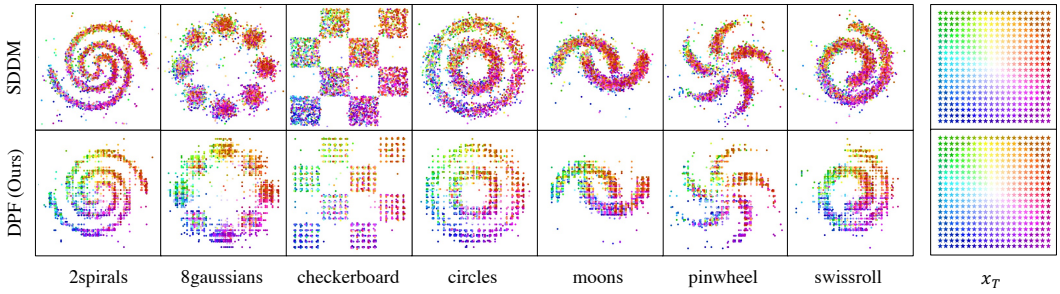


Figure 10: Visualization of the generated samples with state size = 5 from the given initial points $\boldsymbol{x}_T$. Different colors distinguish the generated samples from different initial points $\boldsymbol{x}_T$.

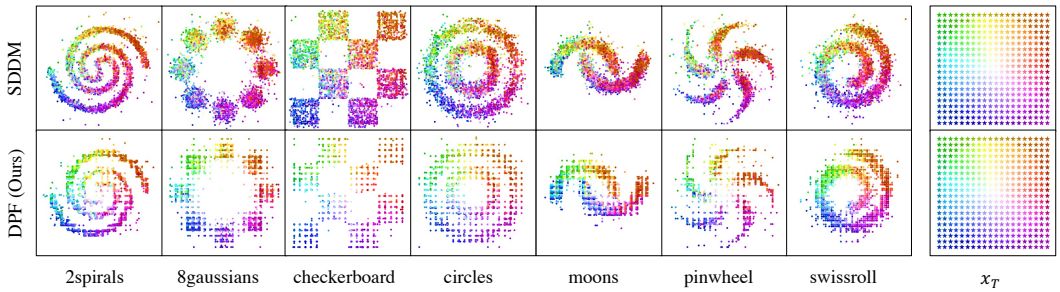


Figure 11: Visualization of the generated samples with state size = 10 from the given initial points $\boldsymbol{x}_T$. Different colors distinguish the generated samples from different initial points $\boldsymbol{x}_T$.

between DPF and SDDM is very small, which is not enough to affect the quality of the generation. This conclusion is further supported by the visualization of the generated samples in Figure 1, Figure 6 and Figure 7 also confirm this point, which demonstrates that DPF produces samples of comparable quality to SDDM.

D.5 Standard Deviation of Generated Samples

To evaluate the certainty of generated samples, we randomly select a 2D float-point and fix it as the initial point. In this experiments, we use the same initial point (-1.91, 1.57). In the binary case, the point is converted into Gray code, whereas in the 5-base and decimal cases, the original code is utilized. For each dataset, we generate 4,000 points and compute the Expectation of the Conditional Standard Deviation (33). Our DPF method results in a significant reduction in the CSD score, as presented in Table 2. For example, on the checkerboard dataset with $S = 5$, our DPF achieves the best score of 1.4103, which is 89% lower than the score achieved by SDDM. We also visualize these results in Figure 2, Figure 8, and Figure 9, where the red star represents the initial point, and the blue points denote the generated samples. Furthermore, it is evident that SDDM generates samples from a single initial point across the entire space, especially for datasets with $S = 2$. In contrast,

Table 5: Comparison of average trajectory length. Lower value indicates a better transport plan.

	2spirals	8gaussians	checkerboard	circles	moons	pinwheel	swissroll
discrete dimension = 32, state size = 2							
SDDM	32.0075	31.7275	31.8010	32.0258	32.0240	31.8650	31.9988
DPF (ours)	1.6135	1.3980	0.7640	1.1995	1.8883	1.8265	1.7065
discrete dimension = 16, state size = 5							
SDDM	26.0680	25.3258	25.9708	25.7565	25.8815	25.9793	26.0810
DPF (ours)	1.5425	1.6390	0.7275	1.1758	1.7102	1.8178	1.5973
discrete dimension = 12, state size = 10							
SDDM	21.8558	21.7828	21.7778	21.9100	22.3180	21.9985	22.2688
DPF (ours)	1.7835	1.1995	0.7213	1.1793	2.0623	1.9433	1.7870

our method can only reach a limited number of states from the initial point, indicating the superior sampling certainty of our approach.

We can also observe that our sampling results tend to form rectangles in Figure 2, Figure 8, and Figure 9. This phenomenon arises from the construction of our synthetic dataset. Specifically, we construct the synthetic dataset states and dimensions by encoding the x-axis and y-axis coordinates of the toy dataset (normalized to [0, 1]) into $K/2$-bit S-ary. This is equivalent to dividing the data space into rectangle regions, where the first few dimensions determine the approximate location of the data. Since our proposed method significantly reduces the uncertainty, each dimension (including the first few dimensions) has only a limited number of possible values. As a result, the points in Figure 2, Figure 8, and Figure 9 appear to form rectangles.

D.6 Generated Samples from Different Initial Points

To display the generated samples from various initial points, we select a 20×20 grid of initial points and mark them with distinct colors. Subsequently, we generate 10 samples for each initial point and presented the results in Figure 3, Figure 10, and Figure 11. It is apparent that the samples obtained through SDDM sampling are mixed together. In contrast, the results obtained by our method exhibit strong regularity, with the generated samples clustering together based on their respective colors. This observation suggests that our method offers improved certainty in the sampling process.

D.7 Distance Between the Generated Samples and Initial Points

Our DPF is designed based on the theory of optimal transport, as demonstrated in Proposition 6. Here, we aim to reflect this finding through experimentation as well. To accomplish this, we utilize the generated samples from Figure 3, Figure 10 and Figure 11, and calculate the average L_1 distance from the generated samples to the corresponding initial point:

$$d_D(i(0), i(T)) = \sum_{l=1}^{S} |i^l(0) - i^l(T)|. \tag{84}$$

The results are presented in Table 4. It is evidence that DPF greatly reduces the distance between the generated samples and the initial point. Moreover, combined with the visualization results in Figure 3, Figure 10 and Figure 11, we observe that our method's sampling outcomes tend to concentrate around the high probability states near the initial point. This outcome aligns with our optimal transport design, further verifying the efficacy of our approach.

However, there is an illusion that the difference between SDDM and PDF decreases as the state size increases. This is mainly because that the Figure 3, Figure 10 and Figure 11 are visualized in the 'float space' instead of the 'encoding space'. Specifically, our synthetic data with a state size of and a dimension size of is established by encoding the x and y coordinates of the toy dataset (normalized to [0, 1]) to $K/2$-bit S-ary respectively. In this encoding, the first dimension of the encoding has the greatest impact on the data position. For example, in binary encoding (state size = 2), the first bit divides the data space into two parts, and determines the part in which it resides. However, as the number of states increases, the space is divided into more parts, and the small change of the first bit can not significantly change the position of the number it represents. This will lead to a narrowing of the gap between our DPF and SDDM in the visualization. Therefore, in such situations,

Table 6: Comparison of transport efficiency. Larger values indicate better transport efficiency.

	2spirals	8gaussians	checkerboard	circles	moons	pinwheel	swissroll
discrete dimension = 32, state size = 2							
SDDM	42.36%	41.93%	42.36%	42.42%	42.62%	42.33%	42.80%
DPF (ours)	98.95%	99.11%	98.49%	99.00%	98.99%	99.29%	99.36%
discrete dimension = 16, state size = 5							
SDDM	48.80%	49.24%	48.07%	48.68%	48.97%	48.59%	48.89%
DPF (ours)	98.96%	98.56%	97.46%	99.23%	99.63%	99.50%	99.47%
discrete dimension = 12, state size = 10							
SDDM	51.90%	50.54%	50.62%	51.21%	52.36%	51.77%	52.24%
DPF (ours)	98.99%	99.54%	99.02%	98.26%	99.36%	99.22%	99.02%

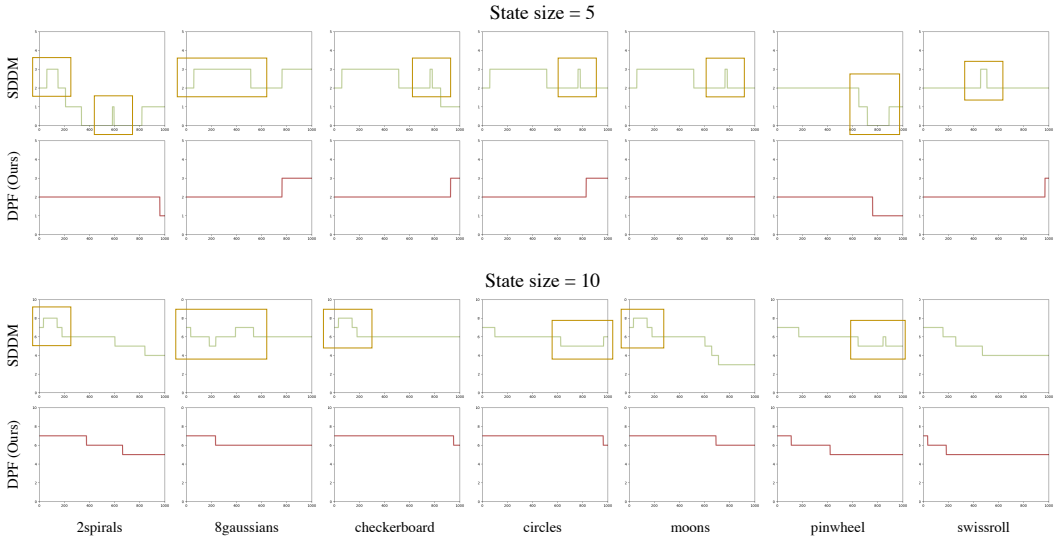


Figure 12: Visualization of the sampling trajectory. The yellow box highlights the duplicated trajectories encountered during the sampling process.

the quantitative results in Table 2 are more appropriate for verifying the reduction of uncertainty in the encoding space.

D.8 Sampling Trajectory Length

Merely examining the distance between the initial point and the generated samples is insufficient to verify that our DPF aligns with optimal transport principles, as the sampling process may follow different trajectories. Therefore, we calculate the cumulative consumption of the sampling trajectory in Figure 3, Figure 10 and Figure 11 according to the following formula:

$$d_{tra}(i(0),\dots,i(T)) = \sum_{t\in\{\tau,2\tau\dots,T\}} d_D(i(t), i(t-\tau)). \tag{85}$$

The results are shown in Table 5. It is evidence that there is a significant decrease in the trajectory length of DPF compared to the SDDM. For instance, our DPF achieves the best score on the checkerboard dataset with S = 5 with a score of 0.7640, which is 97% lower than the score of SDDM. This suggests that the consumption during our sampling process is lower, which is in line with the optimal transport design.

D.9 Transport Efficiency

We also examine the transport efficiency during the sampling process, which can be calculated as the ratio of L_1 distance between the initial point and generated sample to the sampling trajectory length:

$$E_{(i(0),\dots,i(T))} = \frac{d_D(i(0), i(T))}{d_{tra}(i(0),\dots,i(T))}. \tag{86}$$

Table 7: Application on higher dimension or state scenarios. Lower CSD indicate superior certainty.

	2spirals	8gaussians	checkerboard	circles	moons	pinwheel	swissroll
discrete dimension = 20, state size = 50							
SDDM	25.8777	26.5288	25.5106	25.7398	25.6984	25.6984	6.4767
DPF (ours)	2.7113	4.4274	2.6217	3.7554	3.0774	3.3054	3.8183
discrete dimension = 50, state size = 5							
SDDM	47.2706	47.5810	47.4964	47.2733	47.0047	46.9103	46.9819
DPF (ours)	2.0335	1.8134	0.7418	1.7143	1.2840	1.4245	1.5720

A higher value indicates a more optimal sampling trajectory selected by the model from the initial point to the generated sample, i.e., the higher transport efficiency. The results are presented in Table 6. Notably, we observed that only approximately 50% of the trajectory length of SDDM contributes to the actual distance between the initial point and generated samples. In contrast, the transport efficiency of our DPF is close to 100%, which means most jumps in our trajectory efficiently contribute to the final transition. This finding demonstrates that the transport plan selected by our DPF is more effective, aligning with our theoretical derivation.

D.10 Visualization of Sampling Trajectory

The visualization of the sampling trajectory for the 0-th dimension of the dataset is shown in Figure 12. It is evident that the sampling trajectory of SDDM often exhibits duplicate trajectories, which is also the reason for the low transport efficiency of SDDM in Table 6. In contrast, our method, which adheres to the principles of optimal transport theory, ensures that the sampling process only moves toward high probability states, thereby avoiding the occurrence of duplicate trajectories.

D.11 Higher dimension or state scenarios

To further verify our method is still applicable in higher dimension or state scenarios, we increased the number of states and dimension to 50 for experiments. Specifically, we set $S = 50, K = 20$ and $S = 5, K = 50$ to avoid dimension redundancy in the $K/2$-bit S-ary encoding for the toy dataset (float64) coordinates (i.e., $50^{20/2} < 2^{64}$ and $5^{50/2} < 2^{64}$). The results of this experiment are shown in Table 7, which clearly demonstrate that our method can significantly reduce sampling uncertainty even with larger state and dimension sizes.

D.12 Image Modeling

In addition to the transition rate designed in Eq. 26, our method can also be extended to a broad range of transition rates. For example, we can extend our discrete probability flow to the method in [7]. For general Q_{D_t} with ${Q_D}_j^i(t) = {Q_D}_i^j(t)$, define:

$$Q_j^i(t) = \begin{cases} {Q_D}_j^i(t) \frac{ReLU(P_{D_i}(t) - P_{D_j}(t))}{P_{D_i}(t)}, & i \neq j, \\ -\sum_{j \neq i} Q_j^i, & i = j. \end{cases} \tag{87}$$

Q_{D_t} and Q_t have the same single-time marginal distribution. Let $q_t = P_D(t)$ and $x \neq y$, the reverse transition rate can be written as:

$$R_t(x, y) = \frac{q_t(y)}{q_t(x)} Q_t(y, x) \tag{88a}$$

$$= \frac{q_t(y)}{q_t(x)} Q_{D_t}(y, x) \frac{ReLU(q_t(y) - q_t(x))}{q_t(y)} \tag{88b}$$

$$= Q_{D_t}(y, x) RELU(\frac{q_t(y)}{q_t(x)} - 1) \tag{88c}$$

In the same way, the reverse rate in the paper [7] can be written into the following form:

$$\hat{R}_t^{1:D}(\boldsymbol{x}^{1:D}, \tilde{\boldsymbol{x}}^{1:D}) = \sum_{d=1}^{D} R_t^d(\tilde{x}^d, x^d) \delta_{\boldsymbol{x}^{1:D \setminus d}, \tilde{\boldsymbol{x}}^{1:D \setminus d}} RELU(\sum_{x_0^d} q_{0|t}(x_0^d | \boldsymbol{x}^{1:D}) \frac{q_{t|0}(\tilde{x}^d | x_0^d)}{q_{t|0}(x^d | x_0^d)} - 1) \tag{89}$$

Table 8: Comparison of certainty for τLDR-0 and DPF on the Cifar-10 dataset. Here, CSD , class-std, and class-entropy are calculated on 1,000 initial points, each of which has 10 generated images. Lower values indicate superior certainty.

	CSD	class-std	class-entropy
τLDR-0 [7]	57.6898	2.6628	1.7703
DPF (ours)	9.4420	1.1819	0.5291

In this way, the mutual flow between states is eliminated, greatly reducing the sampling uncertainty. To validate this, we validated our DPF on the CIFAR-10 dataset, using the pre-trained discrete diffusion model provided by the paper [7]. Firstly, we selected 1,000 initial points, and sampled 10 images from each initial point. To measure the sampling certainty on the image data, we used a pre-trained CIFAR-10 classifier to classify the image, and introduce two new metrics, i.e., class-std and class-entropy. The class-std calculates the standard deviation of the categories of the images sampled from the same initial point. While the class-entropy calculates the entropy of the category distribution of the images sampled from the same initial point. Lower class-std and class-entropy indicate better sampling certainty. The experimental results, shown in Table 8, demonstrate that our method can significantly reduce the sample uncertainty compared to the τLDR-0 method. Additionally, we visualized the sampled images in Fig. 4. It was clear that from an initial point, our method samples almost the same images, while the original sampling method obtains totally different images.

E Discussion

E.1 Narrow time interval limited in Proposition 6.

We limit the time frame to a narrow interval, as the validity of the proof hinges on the constancy of the sign of $P_i(t) - P_j(t)$. Alternatively, if both equations in Eq. (70) are established concurrently, a contradiction arises whereby 2 equals 0. Consequently, the KKT condition cannot be satisfied by any suitable Lagrange multipliers, thereby rendering the plan sub-optimal.

From an intuitive standpoint, DPF only avoids instantaneous mutual flow, which does not ensure the elimination of mutual flow during finite interval. For example, if we assume $P_i(t) > P_j(t)$ in the interval $[t, t + \epsilon/2)$ and $P_i(t) < P_j(t)$ in $(t + \epsilon/2, t + \epsilon]$, it follows that $\Pi^i_j > 0$ and $\Pi^j_i > 0$. Assuming $\Pi^i_j > \Pi^j_i$, we can demonstrate that the given plan is sub-optimal. If we define a new plan as ${\Pi^*}^i_j = \Pi^i_j - \Pi^j_i$ and ${\Pi^*}^j_i = 0$, we can verify that the resultant plan Π^* incurs a lower transportation cost than Π. The preceding derivation establishes the tightness of our announcement, indicating that the optimal transport plan cannot be extended across the entire time interval.

In order to confirm the existence of such a scenario, we explicitly construct it in the case where $K = 1$. Since $P(t) = P(0)e^{Q_D t}$, we can obtain the analytical solution through eigen decomposition with difference equations, yielding the following outcomes: $\lambda_i = 2cos(i\pi/S) - 2$ and $v_i = (1, cos(\theta_i/2), ..., cos((2S-1)\theta_i/2))$, where $\theta_i = \arccos((\lambda_i + 2)/2)$. Subsequently, we can assess a basic scenario wherein $S = 3$ and $P(t = 0) = (0.1, 0, 0.9)$. It can be observed that $P_0(t = 0) > P_1(t = 0)$ and $P_0(t = 0.1) < P_1(t = 0.1)$. However, the discussion presented above does not deny the existence of a long term optimal transport process. And from an application perspective, it is worth finding out a process with minimal uncertainty.

E.2 Definition of probability flow on universal discrete process.

In contrast to continuous processes, which necessitate the stochastic term to be a Brownian motion, there are few assumptions regarding the discrete stochastic term. As a result, the consideration of the drift term becomes unnecessary as it can be assimilated into the stochastic term. However, it is worth exploring the potential distinctive properties that may arise from treating these two terms separately.

E.3 Practical applications.

Analogously to the effect of continuous probability flow on the continuous diffusion model, we believe that reducing uncertainty can also bring many benefits to discrete diffusion models. For

instance, by selecting appropriate initial data, we can generate results that are pertinent to the initial data to attain controllable generation. Additionally, due to the excellent property of sampling certainty reduction, we can perform operations such as interpolation in latent code to complete data editing.

E.4 Infinite horizon case in Proposition 2.

The infinite horizon case is not addressed in our study due to the presence of singularities that pose significant challenges. For instance, in the case of Brownian motion, the distribution at $t = \infty$ assumes a uniform distribution over the entire $\mathbb{R}^n$, which is not well-defined.

Additionally, the probability flow of Brownian motion at $t = 0$ also experiences a singularity. By taking the limit of the right-hand side of Eq. 20 and let $d_i = x_t - x_i$, we obtain:

$$\begin{aligned}\lim_{t\to 0^+} -\frac{1}{2}\nabla_{x_t} p_B(x_t, t) &= \lim_{t\to 0^+} \frac{\sum_i exp(-\frac{d_i^2}{2t})\frac{d_i}{2t}}{\sum_j exp(-\frac{d_j^2}{2t})} \\ &= \lim_{z\to+\infty} \sum_i \frac{d_i}{\sum_j exp((d_i^2 - d_j^2)z)/z}.\end{aligned} \tag{90}$$

Since

$$\lim_{z\to+\infty} exp((d_i^2 - d_j^2)z)/z = \begin{cases} +\infty & \text{if } d_i^2 > d_j^2, \\ 0 & \text{if } d_i^2 \le d_j^2, \end{cases} \tag{91}$$

we have

$$\lim_{t\to 0} -\frac{1}{2}\nabla_{x_t} p_B(x_t, t) = \lim_{z\to+\infty} \frac{d_{i_{min}}}{\sum_j exp((d_{i_{min}}^2 - d_j^2)z)/z}, \tag{92}$$

where $i_{min} = \arg\min_i d_i^2$. (noting that i_{min} may not be unique, but we exclude this scenario as it does not significantly affect our analysis). Consequently, we obtain:

$$\lim_{t\to 0^+} -\frac{1}{2}\nabla_{x_t} p_B(x_t, t) = \begin{cases} 0, & \text{if } x_{t=0} = x_{i_{min}}, \\ d_{i_{min}} * \infty, & \text{else.} \end{cases} \tag{93}$$

where $d_{i_{min}} * \infty$ indicates that the vector is oriented in the direction of $d_{i_{min}}$ and has an infinite norm. Consequently, the right-hand side of Eq. 20 lacks Lipschitz continuity, leading to non-unique solutions. Actually, if Eq. 20 has a unique solution near $t = 0$, the distribution $p_B(x, t)$ will always be a summation of Dirac deltas, which contradicts Eq. 18. Due to our reliance on the solution of ODE, we are unable to analyze the behavior in the vicinity of $t = 0$. Consequently, we have limited our study to finite intervals.