Score-based continuous-time discrete DM

Tianyu Xie
2022/11/27

1. Differentiating the Wasserstein Loss

Continuous time discrete diffusion models
Introduction

- Diffusion models are characterized by a forward Markov process that transforms an observation $x_0 \sim \pi_{data}(x_0)$ to a reference distribution $x_T \sim q_T(x_T)$.

- The forward process forms a joint distribution after T steps,

$$q_{0:T}(x_{0:T}) = \pi_{data}(x_0) \prod_{t=0}^{T-1} q_{t+1|t}(x_{t+1}|x_t).$$

  where the transition kernel q is usually modelled by Gaussian noise when x is continous random variable.

- Via the Bayes's rule, the backward process is

$$q_{0:T}(x_{0:T}) = q_T(x_T) \prod_{t=0}^{T-1} q_{t|t+1}(x_t|x_{t+1}), \quad q_{t|t+1}(x_t|x_{t+1}) = \frac{q_{t+1|t}(x_{t+1}|x_t) q_t(x_t)}{q_{t+1}(x_{t+1})},$$

Continuous time discrete diffusion models

Introduction

- In the backward process, the conditional distribution $q_{t|t+1}(x_t\,|\,x_{t+1})$ is untractable, often modelled by a neural network.

- Denote the score model at t-th step by r_t^θ. The training loss is

$$\ell_{vb} = \sum_{t=0}^{T-1} (1-\alpha_t)\, \mathbb{E}_{\pi_{data}} \mathbb{E}_{p_{\alpha_t}(x'|x)} \left[\left\| r_t^\theta (x') - \nabla_{x'} \log p_{\alpha_t} (x'|x) \right\|_2^2 \right], \tag{4}$$

  where $\alpha_t = \prod_{t=0}^{t-1} (1-\beta_t)$. in DDPM.

- If we extend the Markov process to continuous time, we obtain an SDE describing the diffusion process

$$\begin{aligned} \mathrm{d}x &= f(x,t)\,\mathrm{d}t + g(t)\mathrm{d}\boldsymbol{w}, && \text{forward SDE,} \\ \mathrm{d}x &= \left[f(x,t) - g^2(t)\nabla_x \log p_t(x)\mathrm{d}t \right] + g(t)\mathrm{d}\bar{\boldsymbol{w}}, && \text{reverse SDE,} \end{aligned}$$

Continuous time discrete diffusion models
Continuous time modeling

- For discrete variables, we can also define a similar Markovian process, but the score function is undefined.

- We consider the finite discrete state space $\mathscr{X} = \mathscr{C}^D$, where $\mathscr{C} = \{1,2,\ldots,C\}$ is a code book. To generalize score matching from a continuous space $\mathbb{R}^n$ to discrete space $\mathscr{X}$, we first model the forward process using a continuous time Markov chain $\{X_t\}_{t\in T}$, whose generator matrix is Q_t.

- In particular, if we let q denote the distribution for the forward process X_t, the transition probability will satisfy the Kolmogorov forward equation.

$$\frac{d}{dt}q_{t|s}(x_t|x_s) = \sum_{x\in\mathcal{X}} q_{t|s}(x|x_s)Q_t(x,x_t), \quad s < t$$

Continuous time discrete diffusion models
Continuous time modeling

- If the forward process starts at the target distribution $q_0 = \pi_{data}$, the marginal at time t takes the form

$$q_t(x_t) = \int_{\mathcal{X}} \pi_{\text{data}}(x_0) q_{t|0}(x_t|x_0) dx_0$$

- By properly choosing the rate matrix Q_t, we can achieve a final distribution close to a known target distribution q_t.

-

Proposition 3.1. *The reverse time process $\overline{X}_t$ of the continuous time Markov chain X_t is also a Markov process, whose transition probabilities $q_{s|t}(\cdot|\cdot)$ for $s < t$ satisfy:*

$$q_{s|t}(x_s|x_t) = \frac{q_s(x_s)}{q_t(x_t)} q_{t|s}(x_t|x_s), \quad s < t \tag{9}$$

Continuous time discrete diffusion models
Continuous time modeling

- The rate matrix satisfies

Proposition 3.2. *For a continuous time Markov chain* $\{X_t\}_{t\in[0,T]}$ *with distribution* q *and rate matrices* Q_t*, the rate matrices* R_t *for the reverse process satisfy:*

$$R_t(x,y) = \frac{q_t(y)}{q_t(x)} Q_t(y,x) \tag{10}$$

- **An important observation**: Therefore, once we know the ratio $q_t(y)/q_t(x)$, we can obtain the generative flow towards π_{data}.

Continuous time discrete diffusion models

Discrete scoring matching

- In general the reverse time transition probability is intractable since $q_t(y)/q_t(x)$ is intractable. However, this ratio behaves analogously to the score function $\nabla \log \pi(x)$.

- This comes from an intuitive conceptual relationship

$$\nabla_i \log \pi(x) \approx \frac{\nabla_i \pi(x)}{\pi(x)} = \frac{\pi(x+e_i) - \pi(x)}{\pi(x)} = \frac{\pi(x+e_i)}{\pi(x)} - 1;$$

- For this reason, Hyvarinen proposed the ratio

$$\frac{\pi(X^{\backslash d}, X^d = c)}{\pi(X^{\backslash d}, X^d = c) + \pi(X^{\backslash d}, X^d = 1 - c)}$$

as the score function for the score matching strategy over binary random variables.

Continuous time discrete diffusion models
Discrete score matching

- More generally, we extend the definition via the conditional distribution

$$\pi(X^d = c|x^{\backslash d}) = \frac{\pi(x^{\backslash d}, X^d = c)}{\sum_{c' \in \mathcal{C}} \pi(x^{\backslash d}, X^d = c')}$$

  yielding the discrete variable score function we seek to match.

- In fact, this score function is guaranteed by the property that the joint distribution is completely determined by its singleton conditional distributions.

Proposition 3.3. *Consider random variables* $X = (X_1, ..., X_D) \in \mathcal{X}$, *and two probability distributions* π_1, π_2. *We have* $\pi_1 = \pi_2$, *if and only if their conditional distributions are equal* $\pi_1(X^d = x^d|x^{\backslash d}) = \pi_2(X^d = x^d|x^{\backslash d})$, *for any* $x \in \mathcal{X}$ *and* $d = 1, ..., D$.

Continuous time discrete diffusion models
Discrete score matching

- We then model the conditional distribution with a time-dependent neural network, i.e.

$$q_t(X_t^d|x^{\backslash d}) \approx p_t(X_t^d|x^{\backslash d};\theta)$$

- Our target is then to minimize the expected cross entropy with respect to the conditional distributions along the forward process

$$\theta^* = \arg\min_\theta \int_0^T \sum_{x_t\in\mathcal{X}} q_t(x_t)\left[\sum_{d=1}^{D}\left(-\sum_{c\in\mathcal{C}} q_t(X_t^d = c|x_t^{\backslash d})\log p_t(X_t^d = c|x_t^{\backslash d};\theta)\right)\right]dt \quad (14)$$

Continuous time discrete diffusion models
Discrete score matching

- To deal with the untractable conditional distribution, the authors propose to the simplifier as pseudo likelihood

Proposition 3.4. *For the reverse process, the score matching loss function in Equation 14 can be simplified as pseudo-likelihood:*

$$\theta^* = \arg\min_\theta \int_0^T \sum_{x_t \in \mathcal{X}} q_t(x_t) \left[\sum_{d=1}^{D} -\log p_t(X^d = x_t^d | x_t^{\backslash d}; \theta) \right] dt \tag{15}$$

- The learned $p_t(X_t^d \,|\, x_t^{\backslash d}; \theta)$ determines a reverse process, and we use $p(\,\cdot\,; \theta)$ to denote its joint distribution in order to distinguish with the true reverse process. We will sometimes drop the θ if it does not create ambiguity.

Continuous time discrete diffusion models

Continuous time simulation for forward process

- The transition matrix from time s to time t can be written as the integral of the rate matrix

$$q_{t|s}(\cdot|\cdot) = \int_s^t \exp(Q_\tau) d\tau$$

- For general matrix $Q_t \in \mathbb{R}^{|X|\times|X|}$, this is integral intractable. So we let each dimension diffuse indepently with matrix $Q_t^d \in \mathbb{R}^{C\times C}$ with a fixes base rate Q and a time schedule $\beta(t)$.

$$Q_t^d = Q\beta(t)$$

- If we let $Q = P\Lambda P^{-1}$, then the sub-matrix function in each dimension can be easily computed as

$$q_{t|s}^d = P \exp\left(\Lambda \int_s^t \beta(\tau) d\tau\right) P^{-1}$$

Continuous time discrete diffusion models

Discrete time sampling for reverse process

- In this work, we assume a uniform stationary base rate

$$Q = \mathbf{1}\mathbf{1}^T - CI,$$

and a cosine style noise schedule

$$\int_0^t \beta(\tau)d\tau = -\left(\cos\frac{\pi}{2}t\right)^{\frac{1}{2}} + 1$$

- The rate for the reversed process to jump to y from x and time t is

$$R_t^d(x_t, y; \theta) = \frac{p_t(X_t^d = y^d | x_t^{\backslash d}; \theta)}{p_t(X_t^d = x_t^d | x_t^{\backslash d}; \theta)} Q_t(x_t, y)$$

- Such a jump rate depends on both the time t and the value in other dimensions $x_t^{\backslash d}$.

Continuous time discrete diffusion models

Discrete time sampling for reverse process

- However, we employ Euler's method to simulate all the dimensions of X_t in parallel. Specifically, given x_t at time t, we fix the rate matrix then determine the transition probabilities for dimension d at time $t - \epsilon$ according to

$$p^d_{t-\epsilon|t}(X^d_{t-\epsilon} = c | x_t^{\backslash d}; \theta) = \begin{cases} \epsilon R^d_t(x_t, X^d_{t-\epsilon} = c; \theta), & c \neq x^d_t \\ 1 - \epsilon \sum_{c' \neq x^d_t} R^d_t(x_t, X^d_{t-\epsilon} = c'; \theta), & c = x^d_t \end{cases}$$

  (we should clip the quantities to ensure all probabilities are non-negative.

- Then we collect a new value from each dimension to obtain a new state $y_{t-\epsilon}$, which has the factorized probability

$$p_{t-\epsilon|t}(X_{t-\epsilon} = y_{t-\epsilon} | x_t; \theta) = \prod_{d=1}^{D} p^d_{t-\epsilon|t}(X^d_{t-\epsilon} = y^d_{t-\epsilon} | x_t^{\backslash d}; \theta)$$

Continuous time discrete diffusion models
Analytical sampling for reverse process

- In this part, we try to design the implicit modeling of $p_t(X^d \mid x_t^{\backslash d}, \theta)$.

- Specifically, for distribution q, we have

$$q_t(X_t^d | x_t^{\backslash d}) = \sum_{x_0^d} q_{0|t}(x_0^d | x_t^{\backslash d}) q_{t|0}(X_t^d | x_0^d)$$

  where $q_{t|0}$ is tractable. Therefore, we only have to model the $q_{0|t}$ to predict x_0^d from $x_t^{\backslash d}$.

- Therefore, our model is

$$p_t(X_t^d | x_t^{\backslash d}; \theta) = \sum_{x_0^d} p_{0|t}(x_0^d | x_t^{\backslash d}; \theta) q_{t|0}(X_t^d | x_0^d) \quad (22)$$

  which provides a tractable transformation from $p_{0|t}(X_0^d \mid x_t^{\backslash d}; \theta)$ to $p_t(X_0^d \mid x_t^{\backslash d}; \theta)$

Continuous time discrete diffusion models

Analytical sampling for reverse process

- Hence, we can continue using the score matching loss to train $p_{0|t}(X_0^d \mid x_t^{\backslash d}; \theta)$.

- To conduct backward sampling via this new parameterization, we consider the true reverse process:

$$
\begin{aligned}
q_{t-\epsilon|t}(X_{t-\epsilon}^d|x_t) &= \sum_{x_0^d} q_{0|t}(x_0^d|x_t) q_{t-\epsilon|0,t}(X_{t-\epsilon}^d|x_0^d, x_t) \\
&= \sum_{x_0^d} \frac{q(x_t^d|x_0^d, x_t^{\backslash d}) q(x_0^d|x_t^{\backslash d})}{q(x_t^d|x_t^{\backslash d})} \frac{q(x_t|x_0^d, X_{t-\epsilon}^d) q(X_{t-\epsilon}^d|x_0^d)}{q(x_t|x_0^d)} \\
&\propto \sum_{x_0^d} q_{0|t}(x_0^d|x_t^{\backslash d}) q_{t|t-\epsilon}(x_t^d|x_{t-\epsilon}^d) q_{t-\epsilon|0}(X_{t-\epsilon}^d|x_0^d)
\end{aligned}
$$

- By substituting $p_t(X_0^d \mid x_t^{\backslash d}; \theta)$, we have

$$
p_{t-\epsilon|t}(X_{t-\epsilon}^d|x_t; \theta) \propto \sum_{x_0^d} p_{0|t}(x_0^d|x_t^{\backslash d}; \theta) q_{t|t-\epsilon}(x_t^d|X_{t-\epsilon}^d) q_{t-\epsilon|0}(X_{t-\epsilon}^d|x_0^d)
$$

Continuous time discrete diffusion models

Parameterization

- In this part, we consider the parameterization of $p_t(X_0^d \mid x_t^{\backslash d};\theta)$. WLOG the same designs can be directly applied to the parameterization of $p_{0|t}(X_0^d \mid x_t^{\backslash d};\theta)$.

- Energy based models

$$p_t(X_t^d = c | x_t^{\backslash d};\theta) = \frac{\exp\Big(-f_\theta([X_t^d = c, x_t^{\backslash d}], t)\Big)}{\sum_{c' \in \mathcal{C}} \exp\Big(-f_\theta([X_t^d = c', x_t^{\backslash d}], t)\Big)}$$

where f_θ is a deep neural network. However, this approach might be computationally prohibitive when modeling high dimensional data because this need $O(D \times C)$ rounds of evaluation.

Continuous time discrete diffusion models

Parameterization

- Masked Models: To alleviate the computation overhead of EBMs while preserving flexibility, a masked model is natural choice.
- Specifically, let a masking function

$$m_d(x) = [x^1, \ldots, x^{d-1}, \text{MASK}, x^{d+1}, \ldots, x^D]$$

replace the d-th dimension of a given x to a special mask token MASK. Then one can formulate the following conditional parameterizationThis

$$p_t(X_t^d | x_t^{\backslash d}; \theta) = \text{Softmax}\Big(f_\theta\big(m(d), t\big)\Big), \text{ where } f_\theta(x, t) : \{\mathcal{C} \cup \text{MASK}\}^D \times \mathbb{R} \mapsto \mathbb{R}^C$$

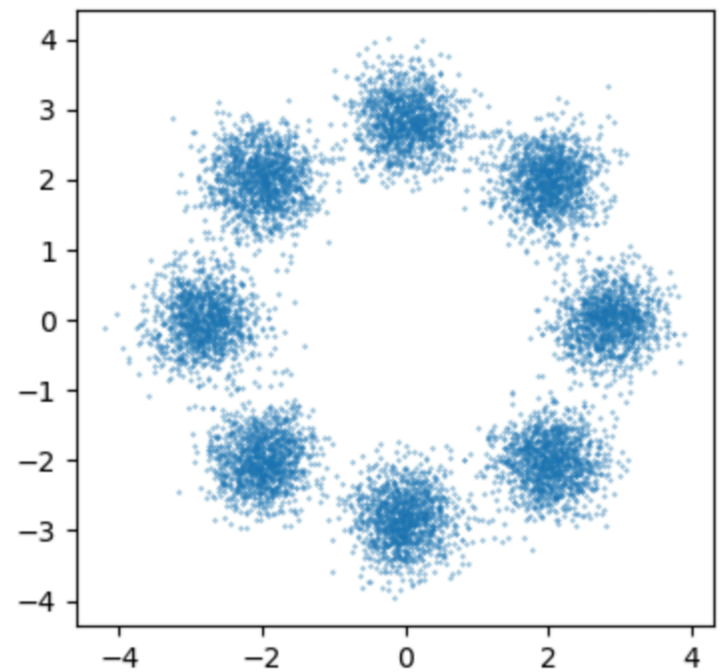
- This approach requires $O(D)$ rounds of evaluation of f_θ.

Continuous time discrete diffusion models
Parameterization

- Hollow transformers:

Continuous time discrete diffusion models
Experiments

- we verify this approach using seven different distributions using binary discrete data to evaluate different approaches.



- For a point (x, y), we convert it by quantizing both x and y with 16-bit representations using a Gray code. Therefore, we obtain the distribution of 32-dimensional binary discrete data.

Continuous time discrete diffusion models
Experiments

- We parameterize the energy function $f_\theta(x, t)$ using the same 3-layer MLP and a sinudoidal embedding of t ino each hidden layer before activation. The uniform rate constant is set to 1.0 and we use a time resolution of 1e-3 for simulation.

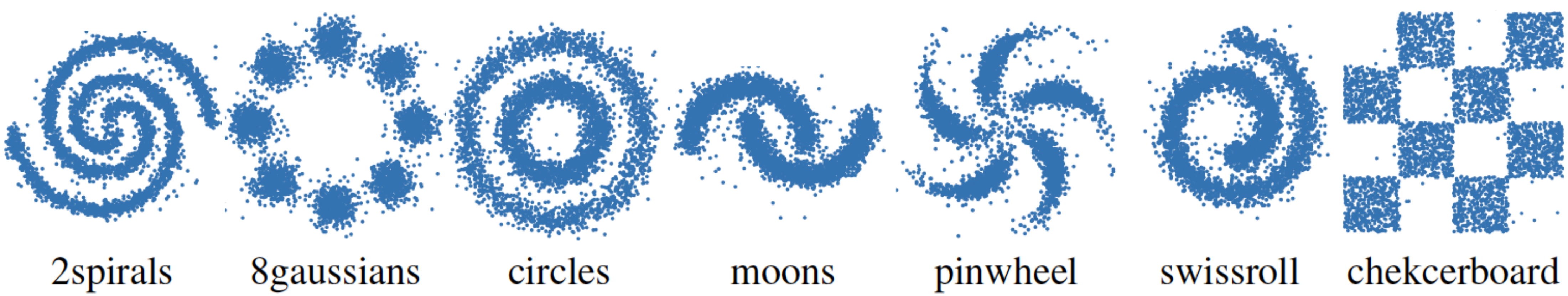


Figure 2: Visualization of sampled discrete binary data in 2D space via decoding of Gray codes.

Table 1: Quality of generated binary samples from the learned EBMs, in terms of MMD with exponential Hamming kernel using bandwidth=0.1 (in units of 1×10^{-4}, the lower the better).

	2spirals	8gaussians	circles	moons	pinwheel	swissroll	checkerboard
PCD (Tieleman, 2008)	2.160	0.954	0.188	0.962	0.505	1.382	2.831
ALOE+ (Dai et al., 2020)	0.149	0.078	0.636	0.516	1.746	0.718	12.138
EB-GFN (Zhang et al., 2022)	0.583	0.531	0.305	0.121	0.492	0.274	1.206
SDDM (this paper)	**0.120**	**0.020**	**0.132**	**0.088**	**0.191**	**0.129**	**0.335**

Continuous time discrete diffusion models
Experiments

- We also represent the raw CIFAR10 image to categorical space. The result is

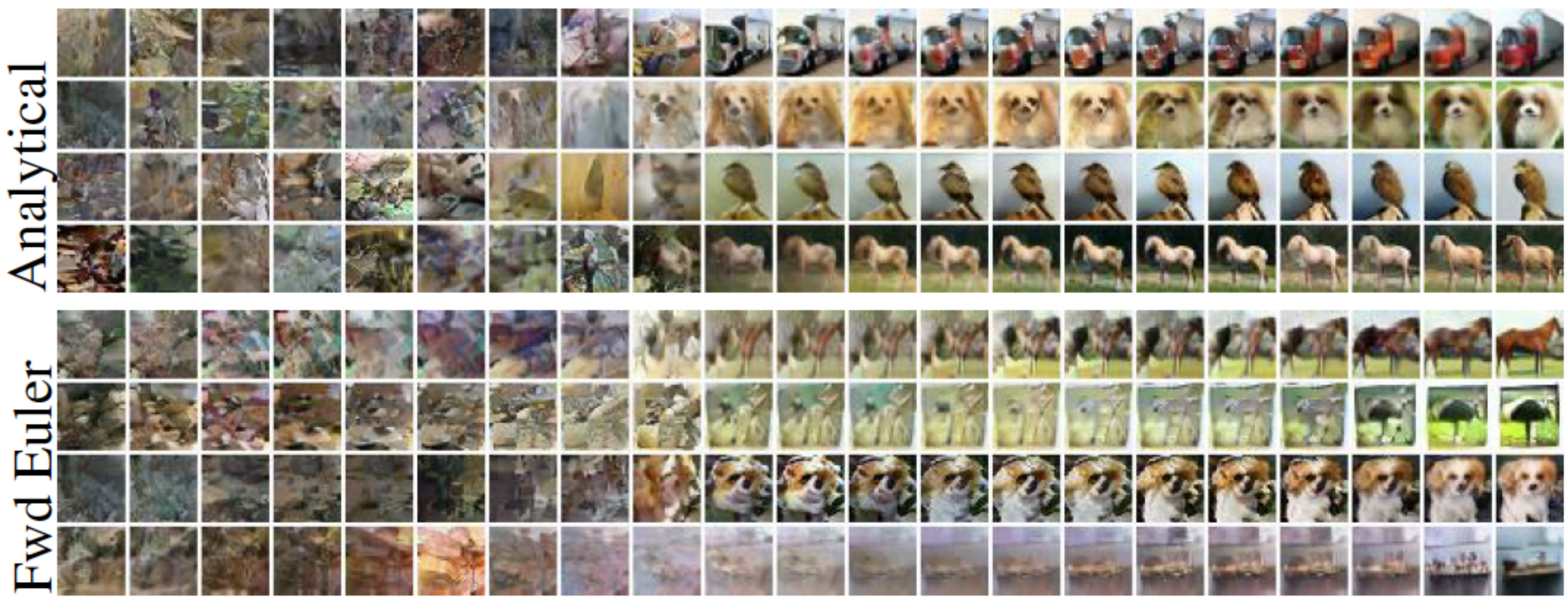


Figure 3: Visualization of reverse sampling with different samplers.

Continuous time discrete diffusion models
Experiments

- We find that the continuous-time counterparts generally improve the performance over the discrete-time counterparts. The reason is that it loses ordinal structure as a prior.

Table 2: Metrics on sample quality for different diffusion models on CIFAR10 dataset. Here Inception Score (IS) and Fréchet Inception Distance (FID) are compared. We follow the common practice to compare the 50,000 unconditionally sampled images from the model and the images from training dataset. Approaches with representations in different state spaces are listed in separate sections.

State space	Methods	IS ↑	FID ↓
Continuous state	DDPM (Ho et al., 2020)	9.46	3.17
	NCSN (Song et al., 2020)	9.89	2.20
Ordinal discrete state	D3PM Gauss (Austin et al., 2021)	8.56	7.34
	τLDR-0 (Campbell et al., 2022)	8.74	8.10
	τLDR-10 (Campbell et al., 2022)	9.49	3.74
Categorical discrete state	D3PM Uniform (Austin et al., 2021)	5.99	51.27
	D3PM Absorbing (Austin et al., 2021)	6.78	30.97
	SDDM-VQ (this paper)	8.91	11.98
	VQGAN (Esser et al., 2021) reconstruction	9.67	9.05