Bridging discrete and continuous state spaces: Exploring the Ehrenfest process in time-continuous diffusion models

Ludwig Winkler [*1] **Lorenz Richter** [*2 3] **Manfred Opper** [1 4 5]

Abstract

Generative modeling via stochastic processes has led to remarkable empirical results as well as to recent advances in their theoretical understanding. In principle, both space and time of the processes can be discrete or continuous. In this work, we study time-continuous Markov jump processes on discrete state spaces and investigate their correspondence to state-continuous diffusion processes given by SDEs. In particular, we revisit the *Ehrenfest process*, which converges to an Ornstein-Uhlenbeck process in the infinite state space limit. Likewise, we can show that the time-reversal of the Ehrenfest process converges to the time-reversed Ornstein-Uhlenbeck process. This observation bridges discrete and continuous state spaces and allows to carry over methods from one to the respective other setting, such as for instance loss functions that lead to improved convergence. Additionally, we suggest an algorithm for training the time-reversal of Markov jump processes which relies on conditional expectations and can thus be directly related to denoising score matching. We demonstrate our methods in multiple convincing numerical experiments.

1. Introduction

Generative modeling based on stochastic processes has led to state-of-the-art performance in multiple tasks of interest, all aiming to sample artificial data from a distribution that is only specified by a finite set of training data (Nichol & Dhariwal, 2021). The general idea is based on the concept of time-reversal: we let the data points *diffuse* until they are close to the equilibrium distribution of the process, from which we assume to be able to sample readily, such that the time-reversal then brings us back to the desired target distribution (Sohl-Dickstein et al., 2015). In this general setup, one can make several choices and take different perspectives. While the original attempt considers discrete-time, continuous-space processes (Ho et al., 2020), one can show that in the small step-size limit the models converge to continuous-time, continuous-space processes given by stochastic differential equations (SDEs) (Song et al., 2021). This continuous-time framework then allows fruitful connections to mathematical tools such as partial differential equations, path space measures and optimal control (Berner et al., 2024). As an alternative, one can consider discrete state spaces in continuous time via Markov jump processes, which have been suggested for generative modeling in Campbell et al. (2022). Those are particularly promising for problems that naturally operate on discrete data, such as, e.g., text, images, graph structures or certain biological data, to name just a few. While discrete in space, an appealing property of those models is that time-discretization is not necessary – neither during training nor during inference[1].

While the connections between Markov jump processes and state-continuous diffusion processes have been studied extensively (see, e.g., Kurtz (1972)), a relationship between their time-reversals has only been looked at recently, where an exact correspondence is still elusive (Santos et al., 2023). In this work, we make this correspondence more precise, thus bridging the gap between discrete-state generative modeling with Markov jump processes and the celebrated continuous-state score-based generative modeling. A key ingredient will be the so-called *Ehrenfest process*, which can be seen as the discrete-state analog of the Ornstein-Uhlenbeck process, that is usually employed in the continuous setting, as well as a loss function that directly translates learning the rate functions of a time-reversed Markov jump process to score functions in the continuous-state analog.

[*] Equal contribution (the author order was determined by `numpy.random.rand(1)`) [1] Technical University of Berlin [2] Zuse Institute Berlin [3] dida Datenschmiede GmbH [4] University of Birmingham [5] University of Potsdam. Correspondence to: Ludwig Winkler <winkler@tu-berlin.de>, Lorenz Richter <richter@zib.de>.

Proceedings of the 41^{st} International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

[1] Note that this is not true for the time- and space-continuous SDE case, where training can be done simulation-free, however, inference relies on a discretization of the reverse stochastic process. However, see Section 4.2 for high-dimensional settings in Markov jump processes.

Our contributions can be summarized as follows:

- We propose a loss function via conditional expectations for training state-discrete diffusion models, which exhibits advantages compared to previous loss functions.
- We introduce the *Ehrenfest process* and derive the jump moments of its time-reversed version.
- Those jump moments allow an exact correspondence to score-based generative modeling, such that, for the first time, the two methods can now be directly linked to one another.
- In consequence, the bridge between discrete and continuous state spaces brings the potential that one setting can benefit from the respective other. E.g., we can motivate loss functions in the continuous setting and employ them in the discrete case, leading to improved performance.

This paper is organized as follows. After listing related work in Section 1.1 and defining notation in Section 1.2, we introduce the time-reversal of Markov jump processes in Section 2 and propose a loss function for learning this reversal in Section 2.1. We define the Ehrenfest process in Section 3 and study its convergence to an SDE in Section 3.1. In Section 3.2 we then establish the connection between the time-reversed Ehrenfest process and score-based generative modeling. Section 4 is devoted to computational aspects and Section 5 provides some numerical experiments that demonstrate our theory. Finally, we conclude in Section 6.

1.1. Related work

Starting with a paper by Sohl-Dickstein et al. (2015), a number of works have contributed to the success of diffusion-based generative modeling, all in the continuous-state setting, see, e.g., Ho et al. (2020); Song & Ermon (2020); Kingma et al. (2021); Nichol & Dhariwal (2021); Vahdat et al. (2021). We shall highlight the work by Song et al. (2021), which derives an SDE formulation of score-based generative modeling and thus builds the foundation for further theoretical developments, see, e.g., Berner et al. (2024) and Richter & Berner (2024). We note that the underlying idea of time-reversing a diffusion process dates back to work by Nelson (1967) and Anderson (1982).

Diffusion models on discrete state spaces have been considered by Hoogeboom et al. (2021) based on appropriate binning operations of continuous models. Song et al. (2020) proposed a method for discrete categorical data, however, did not perform any experiment. A purely discrete diffusion model, both in time and space, termed *Discrete Denoising Diffusion Probabilistic Models (D3PMs)* has been introduced in Austin et al. (2021). Continuous-time Markov jump processes on discrete spaces have first been applied to generative modeling in Campbell et al. (2022), where, however, different forward processes have been considered, for which the forward transition probability is approximated by solving the forward Kolmogorov equation. Sun et al. (2022) introduced the idea of categorical ratio matching for continuous-time Markov Chains by learning the conditional distribution occurring in the transition ratios of the marginals when computing the reverse rates. Recently, in a similar setting, Santos et al. (2023) introduced a pure death process as the forward process, for which one can derive an alternative loss function. Further, they formally investigate the correspondence between Markov jump processes and SDEs, however, in contrast to our work, without identifying a direct relationship between the corresponding learned models.

Finally, we refer to the monographs Gardiner et al. (1985); Van Kampen (1992); Brémaud (2013) for a general introduction to Markov jump processes.

1.2. Notation

For transition probabilities of a Markov jump process M we write $p_{t|s}(x|y) := \mathbb{P}\left(M(t) = x | M(s) = y\right)$ for $s, t \in [0, T]$ and $x, y \in \Omega$. With $p_t(x)$ we denote the (unconditional) probability of the process at time t. We use $p_{\text{data}} := p_0$. With $\delta_{x,y}$ we denote the Kronecker delta. For a function f, we say that $f(x) \in o(g(x))$ if $\lim_{x\to 0} \frac{f(x)}{g(x)} = 0$.

2. Time-reversed Markov jump processes

We consider Markov jump processes $M(t)$ that run on the time interval $[0, T] \subset \mathbb{R}$ and are allowed to take values in a discrete set $\Omega \overset{\sim}{\subset} \mathbb{Z}^d$. Usually, we consider $\Omega \cong \{0, \dots, S\}^d$ such that the cardinality of our space is $|\Omega| = (S + 1)^d$. Jumps between the discrete states appear randomly, where the rate of jumping from state y to x at time t is specified by the function $r_t(x|y)$. The jump rates determine the jump probability in a time increment Δt via the relation

$$p_{t+\Delta t|t}(x|y) = \delta_{x,y} + r_t(x|y)\Delta t + o(\Delta t), \qquad (1)$$

i.e. the higher the rate and the longer the time increment, the more likely is a transition between two corresponding states. For a more detailed introduction to Markov jump processes, we refer to Appendix B.1. In order to simulate the process backwards in time, we are interested in the rates of the time-reversed process $\overleftarrow{M}(t)$, denoted here with $\overleftarrow{r}_t(x|y)$, which determine the backward transition probability via

$$p_{t-\Delta t|t}(x|y) = \delta_{x,y} + \overleftarrow{r}_t(x|y)\Delta t + o(\Delta t). \qquad (2)$$

The following lemma provides a formula for the rates of the time-reversed process, cf. Campbell et al. (2022).

Lemma 2.1. *For two states $x, y \in \Omega$, the transition rates of the time-reversed process $\overleftarrow{M}(t)$ are given by*

$$\overleftarrow{r}_t(y|x) = \mathbb{E}_{x_0 \sim p_{0|t}(x_0|x)}\left[\frac{p_{t|0}(y|x_0)}{p_{t|0}(x|x_0)}\right] r_t(x|y), \tag{3}$$

where r_t is the rate function of the forward process $M(t)$.

Proof. See Appendix A. □

Remark 2.2 (Conditional expectation). We note that the expectation appearing in (3) is a conditional expectation, conditioned on the value $M(t) = x$. This can be compared to the SDE setting, where the score function that is needed for time-reversal can also be written as a conditional expectation, namely $\nabla_x \log p_t^{\mathrm{SDE}}(x) = \mathbb{E}_{x_0 \sim p_{0|t}^{\mathrm{SDE}}(x_0|x)}\left[\nabla_x \log p_{t|0}^{\mathrm{SDE}}(x|x_0)\right]$, see Lemma A.1 in the appendix for more details. We will elaborate on this correspondence in Section 3.2.

While the forward transition probability $p_{t|0}$ can usually be approximated (e.g. by solving the corresponding master equation, see Appendix B.1), the time-reversed transition function $p_{0|t}$ is typically not tractable, and we therefore must resort to a learning task. One idea is to approximate $p_{0|t} \approx p_{0|t}^{\theta}$ by a distribution parameterized in $\theta \in \mathbb{R}^p$ (e.g. via neural networks), see, e.g. Campbell et al. (2022) and Appendix C.2. We suggest an alternative method in the following.

2.1. Loss functions via conditional expectations

Recalling that any conditional expectation can be written as an L^2 projection (see Lemma A.2 in the appendix), we define the loss

$$\mathcal{L}_y(\varphi_y) = \mathbb{E}\left[\left(\varphi_y(x,t) - \frac{p_{t|0}(y|x_0)}{p_{t|0}(x|x_0)}\right)^2\right], \tag{4}$$

where the expectation is over $x_0 \sim p_{\mathrm{data}}, t \sim \mathcal{U}(0,T), x \sim p_{t|0}(x|x_0)$. Assuming a sufficiently rich function class $\mathcal{F}$, it then holds that the minimizer of the loss equals the conditional expectation in Lemma 2.1 for any $y \in \Omega$, i.e.

$$\operatorname*{arg\,min}_{\varphi_y \in \mathcal{F}} \mathcal{L}_y(\varphi_y) = \mathbb{E}_{x_0 \sim p_{0|t}(x_0|x)}\left[\frac{p_{t|0}(y|x_0)}{p_{t|0}(x|x_0)}\right]. \tag{5}$$

We can thus directly learn the conditional expectation. In contrast to approximating the reverse transition probability $p_{0|t}$, this has the advantage that we do not need to model a distribution, but a function, which is less challenging from a numerical perspective. Furthermore, we will see that the conditional expectation can be directly linked to the score function in the SDE setting, such that our approximating functions φ_y can be directly linked to the approximated score. We note that the loss has already been derived in a more general version in Meng et al. (2022) and applied to the setting of Markov jump processes in Lou et al. (2023), however, following a different derivation. A potential issue of the loss (4) is that we may need to approximate different functions φ_y for different $y \in \Omega$ or. This, however, can be coped with in multiple ways. On the one hand, we may learn functions $\varphi(\,\cdot\,; y) = \varphi_y$, i.e. we may consider the conditioning variable y as an additional input of the approximating function. Further, we may focus on *birth-death processes*, for which $r(y|x)$ is non-zero only for $y = x \pm 1$, such that we only need to learn 2 instead of $S - 1$ functions φ_y. In the next section we will argue that birth-death process are in fact favorable for multiple reasons. Lastly, we can do a Taylor expansion such that for certain processes it suffices to only consider one approximating function, as will be shown in Remark 3.3.

3. The Ehrenfest process

In principle, we are free to choose any forward process $M(t)$ for which we can compute the forward transition probabilities $p_{t|0}$ and which is close to its stationary distribution after a not too long run time T. In the sequel, we argue that the *Ehrenfest process* is particularly suitable – both from a theoretical and practical perspective. For notational convenience, we make the argument in dimension $d = 1$, noting, however, that a multidimensional extension is straightforward. For computational aspects in high-dimensional spaces we refer to Section 4.1.

We define the Ehrenfest process[2] as

$$E_S(t) := \sum_{i=1}^{S} Z_i(t), \tag{6}$$

where each Z_i is a process on the state space $\Omega = \{0, 1\}$ with transition rates $r(0|1) = r(1|0) = \frac{1}{2}$ (sometimes called *telegraph* or *Kac process*). We note that the *Ehrenfest process* is a birth-death process with values in $\{0, \dots, S\}$ and transition rates

$$r(x+1|x) = \frac{1}{2}(S - x), \qquad r(x-1|x) = \frac{x}{2}. \tag{7}$$

We observe that we can readily transform the time-independent rates in (7) to time-dependent rates

$$r_t(x \pm 1|x) := \lambda_t \, r(x \pm 1|x) \tag{8}$$

via a time transformation, where $\lambda : [0, T] \to \mathbb{R}$, see Appendix B.2. Without loss of generality, we will focus on the time-independent rates (7) in the sequel.

[2] The Ehrenfest process was introduced by the Russian-Dutch and German physicists Tatiana and Paul Ehrenfest to explain the second law of thermodynamics, see Ehrenfest & Ehrenfest-Afanassjewa (1907).

One compelling property of the Ehrenfest process is that we can sample without needing to simulate trajectories.

Lemma 3.1. *Assuming $E_S(0) = x_0$, the Ehrenfest process can be written as*

$$E_S(t) = E_{0,S}(t) + E_{1,S}(t), \tag{9}$$

where $E_{0,S}(t) \sim B(S - x_0, 1 - f(t))$ and $E_{1,S}(t) \sim B(x_0, f(t))$ are independent binomial random variables and $f(t) := \frac{1}{2}\left(1 + e^{-t}\right)$. Consequently, the forward transition probability is given by the discrete convolution

$$p_{t|0}(x|x_0) = \sum_{z \in \Omega} \mathbb{P}\left(E_{0,S}(t) = z\right) \mathbb{P}\left(E_{1,S}(t) = x - z\right). \tag{10}$$

Proof. See Appendix A. □

We note that the sum in (10) can usually be numerically evaluated without great effort.

3.1. Convergence properties in the infinite state space limit

It is known that certain (appropriately scaled) Markov jump processes converge to state-continuous diffusion processes when the state space size $S + 1$ tends to infinity (see, e.g., Kurtz (1972); Gardiner et al. (1985)). For the Ehrenfest process, this convergence can be studied quite rigorously. To this end, let us introduce the scaled Ehrenfest process

$$\widetilde{E}_S(t) := \frac{2}{\sqrt{S}}\left(E_S(t) - \frac{S}{2}\right) \tag{11}$$

with transition rates

$$r\left(x \pm \frac{2}{\sqrt{S}} \middle| x\right) = \frac{\sqrt{S}}{4}(\sqrt{S} \mp x), \tag{12}$$

now having values in $\Omega = \left\{-\sqrt{S}, -\sqrt{S} + \frac{2}{\sqrt{S}}, \dots, \sqrt{S}\right\}$. We are interested in the large state space limit $S \to \infty$, noting that this implies $\frac{2}{\sqrt{S}} \to 0$ for the transition steps, thus leading to a refinement of the state space. The following convergence result is shown in Sumita et al. (2004, Theorem 4.1).

Proposition 3.2 (State space limit of Ehrenfest process)**.** *In the limit $S \to \infty$, the scaled Ehrenfest process $\widetilde{E}_S(t)$ converges in law to the Ornstein-Uhlenbeck process X_t for any $t \in [0, T]$, where X_t is defined via the SDE*

$$\mathrm{d}X_t = -X_t \, \mathrm{d}t + \sqrt{2} \, \mathrm{d}W_t, \tag{13}$$

with W_t being standard Brownian motion.

For an illustration of the convergence we refer to Figure 1.

Note that the convergence of the scaled Ehrenfest process to the Ornstein-Uhlenbeck process implies

$$p_{t|0}(x|x_0) \approx p_{t|0}^{\mathrm{OU}}(x|x_0) := \mathcal{N}(x; \mu_t(x_0), \sigma_t^2), \tag{14}$$

where $p_{t|0}$ is the transition probability of the discrete Ehrenfest process, as before, and $p_{t|0}^{\mathrm{OU}}$ is the transition probability of the Ornstein-Uhlenbeck process, which operates in continuous space. In the above we have defined the shorthands $\mu_t(x_0) = x_0 e^{-t}$ and $\sigma_t^2 = (1 - e^{-2t})$. For the quantity in the conditional expectation (3) we can thus compute

$$\frac{p_{t|0}\left(x \pm \delta \middle| x_0\right)}{p_{t|0}(x|x_0)} \approx \exp\left(\frac{\mp 2(x - \mu_t(x_0))\delta - \delta^2}{2\sigma_t^2}\right) \tag{15a}$$

$$\approx \exp\left(-\frac{\delta^2}{2\sigma_t^2}\right)\left(1 \mp \frac{(x - \mu_t(x_0))\delta}{\sigma^2} + \frac{((x - \mu_t(x_0))\delta)^2}{2\sigma^4}\right), \tag{15b}$$

where we used approximation (14) in the first and a Taylor expansion in the second line, which is justified for small $\delta := \frac{2}{\sqrt{S}}$, i.e. large S.

Remark 3.3 (Learning of conditional expectation). Being able to relate the Ehrenfest process to the Ornstein-Uhlenbeck process allows us to suggest alternative loss functions for learning the conditional expectation necessary for the computation of the backward rates (cf. Lemma 2.1). In particular, note with (4) that approximation (15) allows us to define the loss

$$\mathcal{L}_{\text{Gauß}}^{\mathrm{b/d}}(\varphi) := \mathbb{E}\left[\left(\varphi(x, t) - \exp\left(\frac{\mp 2(x - \mu_t(x_0))\delta - \delta^2}{2\sigma_t^2}\right)\right)^2\right], \tag{16}$$

where $\mathrm{b/d}$ stand for *birth* and *death*, respectively. Further, we can write

$$\mathbb{E}_{x_0}\left[\frac{p_{t|0}\left(x \pm \delta \middle| x_0\right)}{p_{t|0}(x|x_0)}\right] \approx \exp\left(-\frac{\delta^2}{2\sigma_t^2}\right) \left(1 \mp \frac{(x - \mathbb{E}_{x_0}\left[\mu_t(x_0)\right])\delta}{\sigma^2} + \frac{\mathbb{E}_{x_0}\left[((x - \mu_t(x_0))\delta)^2\right]}{2\sigma^4}\right), \tag{17}$$

where $x_0 \sim p_{0|t}(x_0|x)$. In consequence, this allows us to consider the loss functions

$$\mathcal{L}_{\text{Taylor}}(\varphi_1) := \mathbb{E}\left[\left(\varphi_1(x, t) - \mu_t(x_0)\right)^2\right], \tag{18}$$

and

$$\mathcal{L}_{\mathrm{Taylor},2}(\varphi_2) := \mathbb{E}\left[\left(\varphi_2(x,t) - ((x-\mu_t(x_0))\delta)^2\right)^2\right] \tag{19}$$

for learning the two conditional expectations appearing in (17), where the expectations are again over $x_0 \sim p_{\mathrm{data}}, t \sim \mathcal{U}(0,T), x \sim p_{t|0}(x|x_0)$. We can also only consider the first order term in the Taylor expansion (15b) and thus only need to approximate one function via $\mathcal{L}_{\mathrm{Taylor}}$, which should be contrasted with (16), where two functions need to be approximated.

Since the scaled forward Ehrenfest process converges to the Ornstein-Uhlenbeck process, we can expect the time-reversed scaled Ehrenfest process to converge to the time-reversal of the Ornstein-Uhlenbeck process. We shall study this conjecture in more detail in the sequel.

3.2. Connections between time-reversal of Markov jump processes and score-based generative modeling

Inspecting Lemma 2.1, which specifies the rate function of a backward Markov jump process, we realize that the time-reversal essentially depends on two things, namely the forward rate function with switched arguments as well as the conditional expectation of the ratio between two forward transition probabilities. To gain some intuition, let us first assume that the state space size $S+1$ is large enough and that the transition density $p_{t|0}$ can be extended to $\mathbb{R}$ (which we call $\overline{p}_{t|0}$) such that it can be approximated via a Taylor expansion. We can then assume that

$$r\left(x \pm \frac{2}{\sqrt{S}}\middle| x\right) \approx r\left(x \middle| x \mp \frac{2}{\sqrt{S}}\right), \tag{20}$$

which intuitively means that the arguments in the rate functions can be switched if the state space is large enough[3]. Further, we assume

$$\frac{p_{t|0}\left(x \pm \frac{2}{\sqrt{S}}\middle| x_0\right)}{p_{t|0}(x|x_0)} \approx \frac{\overline{p}_{t|0}(x|x_0) \pm \frac{2}{\sqrt{S}}\nabla \overline{p}_{t|0}(x|x_0)}{\overline{p}_{t|0}(x|x_0)} \tag{21a}$$

$$= 1 \pm \frac{2}{\sqrt{S}} \nabla \log \overline{p}_{t|0}(x|x_0), \tag{21b}$$

where the conditional expectation of $\nabla \log \overline{p}_{t|0}(x|x_0)$ is reminiscent of the score function in SDE-based diffusion models (cf. Lemma A.1 in the appendix). This already hints at a close connection between the time-reversal of Markov jump processes and score-based generative modeling. Further, note that (21a) corresponds to (15b) for large enough S and $p_{t|0} \approx \overline{p}^{\mathrm{OU}}_{t|0}$.

We shall make the above observation more precise in the following. To this end, let us study the first and second jump moments of the Markov jump process, given as

$$b(x) := \sum_{y \in \Omega, y \neq x} (y-x) r(y|x), \tag{22}$$

$$D(x) := \sum_{y \in \Omega, y \neq x} (y-x)^2 r(y|x), \tag{23}$$

see Appendix B.3. For the scaled Ehrenfest process (11) we can readily compute

$$b(x) = -x, \qquad D(x) = 2, \tag{24}$$

which align with the drift and diffusion coefficient (which is the square root of D) of the Ornstein-Uhlebeck process in Proposition 3.2. In particular, we can show the following relation between the jump moments of the forward and the backward Ehrenfest processes, respectively.

Proposition 3.4. *Let b and D be the first and second jump moments of the scaled Ehrenfest process $\widetilde{E}_S$. The first and second jump moments of the time-reversed scaled Ehrenfest $\overleftarrow{\widetilde{E}}_S$ are then given by*

$$\overleftarrow{b}(x,t) = -b(x) + D(x)\, \mathbb{E}_{x_0 \sim p_{0|t}(x_0|x)}\left[\frac{\Delta_S p_{t|0}(x|x_0)}{p_{t|0}(x|x_0)}\right] + o(S^{-1/2}), \tag{25}$$

$$\overleftarrow{D}(x) = D(x) + o(S^{-1/2}), \tag{26}$$

where

$$\Delta_S p_{t|0}(x|x_0) := \frac{p_{t|0}(x + \frac{2}{\sqrt{S}}|x_0) - p_{t|0}(x|x_0)}{\frac{2}{\sqrt{S}}} \tag{27}$$

is a one step difference and $p_{t|0}$ and $p_{0|t}$ are the forward and reverse transition probabilities of the scaled Ehrenfest process.

Proof. See Appendix A. □

Remark 3.5 (Convergence of the time-reversed Ehrenfest process). We note that Proposition 3.4 implies that the time-reversed Ehrenfest process is expected to converge in law to the time-reversed Ornstein-Uhlenbeck process. This can be seen as follows. For $S \to \infty$, we know via Proposition 3.2 that the forward Ehrenfest process converges to the Ornstein-Uhlenbeck process, i.e. $p_{t|0}$ converges to $p^{\mathrm{OU}}_{t|0}$, where $p^{\mathrm{OU}}_{t|0}(x|x_0)$ is the transition density of the Ornstein-Uhlenbeck process (13) starting at $X_0 = x_0$. Together with the fact that the finite difference approximation operator Δ_S converges to the first derivative, this implies that $\mathbb{E}_{x_0 \sim p_{0|t}(x_0|x)}\left[\frac{\Delta_S p_{t|0}(x|x_0)}{p_{t|0}(x|x_0)}\right]$ is expected to converge

[3] To be more precise, we assume the statement in (29), which for the Ehrenfest process is shown in (58) in the appendix.

to $\mathbb{E}_{x_0\sim p^{\mathrm{OU}}_{0|t}(x_0|x)}\left[\nabla \log p^{\mathrm{OU}}_{t|0}(x|x_0)\right]$. Now, Lemma A.1 in the appendix shows that this conditional expectation is the score function of the Ornstein-Uhlenbeck process, i.e. $\nabla \log p_t^{\mathrm{OU}}(x) = E_{x_0\sim p^{\mathrm{OU}}_{0|t}(x_0|x)}\left[\nabla \log p^{\mathrm{OU}}_{t|0}(x|x_0)\right]$. Finally, we note that the first and second jump moments converge to the drift and the square of the diffusion coefficient of the limiting SDE, respectively (Gardiner et al., 1985).Therefore, the scaled time-reversed Ehrenfest process $\overleftarrow{E}_S(t)$ is expected to converge in law to the process Y_t given by

$$\mathrm{d}Y_t = \left(Y_t + 2\nabla \log p^{\mathrm{OU}}_{T-t}(Y_t)\right)\mathrm{d}t + \sqrt{2}\,\mathrm{d}W_t, \qquad (28)$$

which is the time-reversal of the Ornstein-Uhlenbeck process stated in (13). Note that we write (28) as a forward process from $t = 0$ to $t = T$, where W_t is a forward Brownian motion, which induces the time-transformation $t \mapsto T - t$ in the score function.

Remark 3.6 (Generalizations). Following the proof of Proposition 3.4, we note that the formulas for the first two jump moments of the time-reversed Markov jump process, stated in (25) and (26), are in fact valid for any (appropriately scaled) birth-death process whose transition rates fulfill[4]

$$\frac{1}{S}\left(r(x \pm \delta|x) - r(x|x \mp \delta)\right) = o(S^{-1}), \qquad (29)$$

where δ is a jump step size that decreases with the state space size $S + 1$.

Crucially, Remark 3.5 shows that we can directly link approximations in the (scaled) state-discrete setting to standard state-continuous score-based generative modeling via

$$\mathbb{E}_{x_0\sim p_{0|t}(x_0|x)}\left[\frac{p_{t|0}(x \pm \frac{2}{\sqrt{S}}|x_0)}{p_{t|0}(x|x_0)}\right] \approx 1 \pm \frac{2}{\sqrt{S}}\nabla \log p_t^{\mathrm{OU}}(x), \qquad (30)$$

see also the proof of Proposition 3.4 in Appendix A. In particular, this allows for transfer learning between the two cases. E.g., we can train a discrete model and use the approximation of the conditional expectation (up to scaling) as the score function in a continuous model. Likewise, we can train a continuous model and approximate the conditional expectation by the score. We have illustrated the latter approach in Figure 1, where we have used the (analytically available) score function that transports a standard Gaussian to a multimodal Gaussian mixture in a discrete-state Ehrenfest process that starts at a binomial distribution which is designed in such a way that it converges to the standard Gaussian for $S \to \infty$.

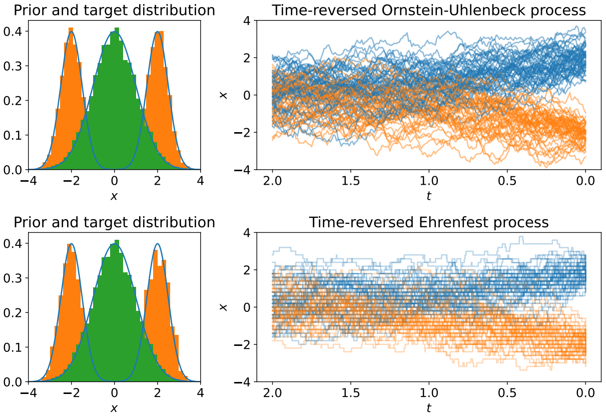

Figure 1. We display two time-reversed processes from $t = 2$ to $t = 0$ that transport a standard Gaussian (left panels, in green) to a multimodal Gaussian mixture model (left panels, in orange), or a binomial distribution to a binomial mixture, respectively, once using a diffusion process in continuous space (upper panel) and once a time-reversed (scaled) Ehrenfest process in discrete space with $S = 100$ (lower panel). Crucially, in both cases we use the (state-continuous) score function to employ the time-reversal, which for this problem is known analytically, see Appendix D.1. The plots demonstrate that the distributions of the processes seem indeed very close one another, implying that the approximation (30) is quite accurate even for a moderate state space size $S + 1$.

Similar to (4), the correspondence (30) motivates to train a state-discrete scaled Ehrenfest model with the loss defined by

$$\mathcal{L}_{\mathrm{OU}}(\widetilde{\varphi}) := \mathbb{E}\left[\left(\widetilde{\varphi}(x,t) - \nabla \log p^{\mathrm{OU}}_{t|0}(x|x_0)\right)^2\right] \qquad (31a)$$

$$= \mathbb{E}\left[\left(\widetilde{\varphi}(x,t) + \frac{x - \mu_t(x_0)}{\sigma_t^2}\right)^2\right], \qquad (31b)$$

where the expectation is over $x_0 \sim p_{\mathrm{data}}, t \sim \mathcal{U}(0,T), x \sim p_{t|0}(x|x_0)$ and where $\mu_t(x_0) = x_0 e^{-t}$ and $\sigma_t^2 = (1 - e^{-2t})$, as before. In fact, this loss is completely analog to the denoising score matching loss in the state-continuous setting. We later set $\varphi = 1 \pm \frac{2}{\sqrt{S}}\widetilde{\varphi}^*$, where $\widetilde{\varphi}^*$ is the minimizer of (31), to get the approximated conditional expectation. We refer to Appendix D.2 for further details.

Remark 3.7 (Ehrenfest process as discrete-state DDPM). To make the above considerations more precise, note that we can directly link the discrete-space Ehrenfest process to pretrained score models in continuous space, such as, e.g., the celebrated *denoising diffusion probabilistic models (DDPM)* (Ho et al., 2020). Those models usually transport a standard Gaussian to the target density that is supported

[4] Intuitively, the statement (29) means that if the arguments of the rate function can be switched up to constant order, this constant order will later be removed by sending S to infinity, since for the computation of the jump moments (as e.g. in (56) or (57) in the appendix) the rate function gets multiplied with the jump size, which scales inversely in S (for the Ehrenfest process with $1/\sqrt{S}$).

on $[-1,1]^d$. In order to cope with the fact that the scaled Ehrenfest process terminates (approximately) at a standard Gaussian irrespective of the size $S+1$, we typically choose $S = 255^2$ such that the interval $[-1,1]$ contains 256 states that correspond to the RGB color values of images, recalling that the increments between the states are $\frac{2}{\sqrt{S}}$. Further, noting the actual Ornstein-Uhlenbeck process that DDPM is trained on, we employ the time scaling $\lambda_t = \frac{1}{2}\beta(t)$, where β and further details are stated in Appendix D.2, and choose the (time-dependent) rates

$$r_t\left(x \pm \frac{2}{\sqrt{S}}\middle| x\right) = \beta(t)\frac{\sqrt{S}}{8}(\sqrt{S} \mp x), \tag{32}$$

according to (8) and (12).

4. Computational aspects

In this section, we comment on computational aspects that are necessary for the training and simulation of the time-reversal of our (scaled) Ehrenfest process. For convenience, we refer to Algorithms 1 and 2 in Appendix C.1 for the corresponding training and sampling algorithms, respectively.

4.1. Modeling of dimensions

In order to make computations feasible in high-dimensional spaces Ω^d, we typically factorize the forward process, such that each dimension propagates independently, cf. Campbell et al. (2022). Note that this is analog to the Ornstein-Uhlenbeck process in score-based generative modeling, in which the dimensions also do not interact, see, e.g., (13).

We thus consider

$$p_{t|0}(x|y) = \prod_{i=1}^{d} p_{t|0}^{(i)}(x^{(i)}|y^{(i)}), \tag{33}$$

where $p_{t|0}^{(i)}$ is the transition probability for dimension $i \in \{1,\dots,d\}$ and $x^{(i)}$ is the i-th component of $x \in \Omega^d$.

In Campbell et al. (2022) it is shown that the forward and backward rates then translate to

$$r_t(x|y) = \sum_{i=1}^{d} r_t^{(i)}(x^{(i)}|y^{(i)})\Gamma_{x^{\neg i}, y^{\neg i}}, \tag{34}$$

where $\Gamma_{x^{\neg i}, y^{\neg i}}$ is one if all dimensions except the i-th dimension agree, and

$$\overleftarrow{r}_t(x|y) = \sum_{i=1}^{d} \mathbb{E}\left[\frac{p_{t|0}(y^{(i)}|x_0^{(i)})}{p_{t|0}(x^{(i)}|x_0^{(i)})}\right] r_t^{(i)}(x^{(i)}|y^{(i)})\Gamma_{x^{\neg i}, y^{\neg i}}, \tag{35}$$

where the expectation is over $x_0^{(i)} \sim p_{0|t}(x_0^{(i)}|x)$. Equation (35) illustrates that the time-reversed process does not factorize in the dimensions even though the forward process does.

Note with (34) that for a birth-death process a jump appears only in one dimension at a time, which implies that

$$r_t(x \pm \delta_i | x) = r_t^{(i)}(x^{(i)} \pm \delta_i^{(i)} | x^{(i)}), \tag{36}$$

where now $\delta_i = (0,\dots,0,\delta_i^{(i)},0,\dots,0)^\top$ with $\delta_i^{(i)}$ being the jump step size in the i-th dimension. Likewise, (35) becomes

$$\overleftarrow{r}_t(x \pm \delta_i | x) = \mathbb{E}\left[\frac{p_{t|0}(y^{(i)}|x_0^{(i)})}{p_{t|0}(x^{(i)}|x_0^{(i)})}\right] r_t^{(i)}(x^{(i)}|x^{(i)} + \delta_i^{(i)}), \tag{37}$$

where the expectation is over $x_0^{(i)} \sim p_{0|t}(x_0^{(i)}|x)$, which still depends on all dimensions.

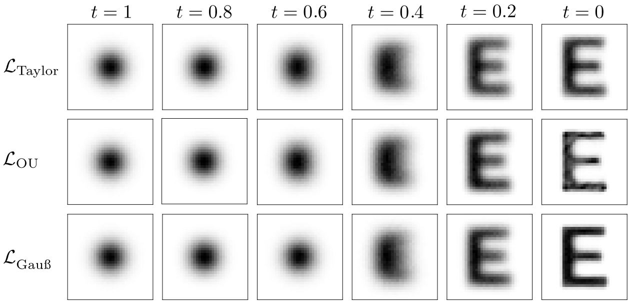

Figure 2. We plot histograms of 500.000 samples from the time-reversed scaled Ehrenfest process at different times. The processes have been trained with three different losses.

For each dimension $i \in \{1,\dots,d\}$ we can therefore approximate the conditional expectation appearing in (37) via the loss function (4) with two[5] functions $\varphi_{i,b} : \mathbb{R}^d \times [0,T] \to \mathbb{R}$ and $\varphi_{i,d} : \mathbb{R}^d \times [0,T] \to \mathbb{R}$. Alternatively, we can learn just two functions $\varphi_{b/d} : \mathbb{R}^d \times [0,T] \to \mathbb{R}^d$ for the entire space and identify $\varphi_{i,b/d} = \varphi_{b/d}^{(i)}$.

4.2. τ-leaping

The fact that jumps only happen in one dimension at a time implies that the naive implementation of changing component by component (e.g. by using the Gillespie's algorithm, see Gillespie (1976)) would require a very long sampling time. As suggested in Campbell et al. (2022), we can therefore rely on τ-leaping for an approximate simulation methods (Gillespie, 2001). The general idea is to not simulate jump by jump, but wait for a time interval of length τ and apply all jumps at once. One can show that the number of jumps is Poisson distributed with a mean of $\tau\, \overleftarrow{r}_t\,(x|y)$. For further details we refer to Algorithm 2.

[5] The same argument of course holds when the one dimensional loss function only depends on one function.

5. Numerical experiments

In this section, we demonstrate our theoretical insights in numerical experiments. If not stated otherwise, we always consider the scaled Ehrenfest process defined in (11). We will compare the different variants of the loss (4), namely $\mathcal{L}_{\text{Gauss}}$ defined in (16), $\mathcal{L}_{\text{Taylor}}$ defined in (18) and $\mathcal{L}_{\text{OU}}$ defined in (31). Our code can be found at `https://github.com/ludwigwinkler/EhrenfestDiffusion`.

5.1. Illustrative example

Let us first consider an illustrative example, for which the data distribution is tractable. We consider a process in $d = 2$ with $S = 32$, where the $(S + 1)^d = 33^2$ different state combinations in p_{data} are defined to be proportional to the pixels of an image of the letter "E". Since the dimensionality is $d = 2$, we can visually inspect the entire distribution at any time $t \in [0, T]$ by plotting 2-dimensional histograms of the simulated processes. With this experiment we can in particular check that modeling the dimensions of the forward process independently from one another (as explained in Section 4.1) is no restriction for the backward process. Indeed Figure 2 shows that the time-reversed process, which is learned with (versions of) the loss (4), can transport the prior distribution (which is approximately binomial, or, loosely speaking, a binned Gaussian) to the specified target. Again, note that this plot does not display single realizations, but entire distributions, which, in this case, are approximated with 500.000 samples. We realize that in this simple problem $\mathcal{L}_{\text{Gauß}}$ performs slightly better than $\mathcal{L}_{\text{OU}}$ and $\mathcal{L}_{\text{Taylor}}$. As expected, the approximations work sufficiently well even for a moderate state space size $S + 1$. As argued in Section 3.1, this should get even better with growing S. For further details, we refer to Appendix D.3.

5.2. MNIST

For a basic image modeling task, we consider the MNIST dataset, which consists of gray scale pixels and was resized to 32×32 to match the required input size of a U-Net neural network architecture[6], such that $d = 32 \times 32 = 1024$ and $S = 255$. As before, we train our time-reversed Ehrenfest model by using the variants of the loss introduced in Section 2.1. In Figure 3 we display generated samples from a model trained with $\mathcal{L}_{\text{OU}}$. The models with the other losses look equally good, so we omit them. For further details, we refer to Appendix D.4.

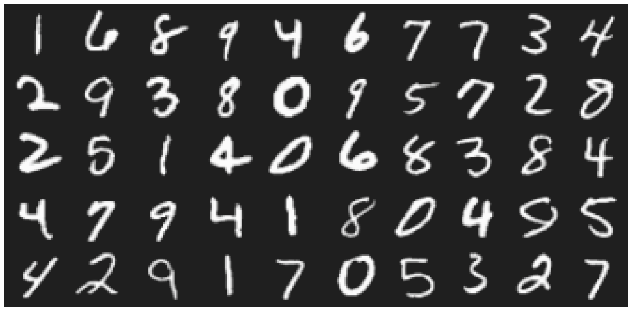

Figure 3. MNIST samples obtained with the time-reversed scaled Ehrenfest process which was trained with $\mathcal{L}_{\text{OU}}$.

5.3. Image modeling with CIFAR-10

As a more challenging task, we consider the CIFAR-10 data set, with dimension $d = 3 \times 32 \times 32 = 3072$, each taking 256 different values (Krizhevsky et al., 2009). In the experiments we again compare our three different losses, however, realize that $\mathcal{L}_{\text{Gauß}}$ did not produce satisfying results and had convergence issues, which might follow from numerical issues due to the exponential term appearing in (16). Further, we consider three different scenarios: we train a model from scratch, we take the U-Net model that was pretrained in the state-continuous setting, and we take the same model and further train it with our state-discrete training algorithm (recall Remark 3.7, which describes how to link the Ehrenfest process to DDPM).

We display the resulting metrics in Table 1. When using only transfer learning, the different losses indicate different ways of incorporating the pretrained model, see Appendix D.2. We realize that both losses produce comparable results, with small advantages for $\mathcal{L}_{\text{OU}}$. Even without having invested much time in finetuning hyperparameters and sampling strategies, we reach competitive performance with respect to the alternative methods LDR (Campbell et al., 2022) and D3PM (Austin et al., 2021). Remarkably, even the attempt with transfer learning returns good results, without having applied any further training. For further details, we refer to Appendix D.5, where we also display more samples in Figures 6-9.

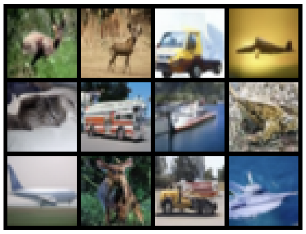

Figure 4. CIFAR-10 samples from the Ehrenfest process with a pretrained model, further finetuned with $\mathcal{L}_{\text{OU}}$.

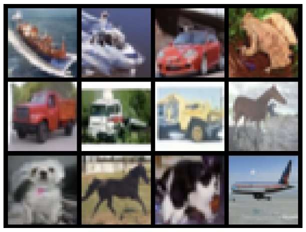

Figure 5. CIFAR-10 samples from the Ehrenfest process with a pretrained model, further finetuned with $\mathcal{L}_{\text{Taylor}}$.

[6] Taken from the repository `https://github.com/w86763777/pytorch-ddpm`.

		IS (↑)	FID (↓)
Ehrenfest (from scratch)	$\mathcal{L}_{\mathrm{OU}}$	9.50	5.08
	$\mathcal{L}_{\mathrm{Taylor}}$	9.66	5.12
	$\mathcal{L}_{\mathrm{Taylor2}}$	9.40	5.44
Ehrenfest (transfer learning)	$\mathcal{L}_{\mathrm{OU}}$	8.75	11.57
	$\mathcal{L}_{\mathrm{Taylor}}$	8.68	11.72
Ehrenfest (pretrained)	$\mathcal{L}_{\mathrm{OU}}$	9.14	6.63
	$\mathcal{L}_{\mathrm{Taylor}}$	9.06	6.91
	$\mathcal{L}_{\mathrm{Taylor2}}$	9.10	7.01
Alternative methods	τ-LDR (0)	8.74	8.10
	τ-LDR (10)	9.49	3.74
	D3PM Gauss	8.56	7.34
	D3PM Absorbing	6.78	30.97

Table 1. Performance in terms of Inception Score (IS) (Salimans et al., 2016) and Frechet Inception Distance (FID) (Heusel et al., 2017) on CIFAR-10 over 50.000 samples. We compare three losses and consider three different scenarios: we train a model in our discrete setting from scratch, we take the U-Net model that was pretrained in the state-continuous setting without any further training in the discrete setting (called "transfer learning") or we take the same model and further train it with our state-discrete algorithm, however only for a short time (called "pretraining").

6. Conclusion

In this work, we have related the time-reversal of discrete-space Markov jump processes to continuous-space score-based generative modeling, such that, for the first time, one can directly link models of the respective settings to one another. While we have focused on the theoretical connections, our numerical experiments demonstrate that we can already reach competitive performance with the new loss functions that we proposed. We suspect that further tuning and the now possible transfer learning between discrete and continuous state space will further enhance the performance. On the theoretical side, we anticipate that the convergence of the time-reversed jump processes to the reversed SDE can be generalized even further, which we leave to future work.

Acknowledgements

L.W. acknowledges support by the Federal Ministry of Education and Research (BMBF) for BIFOLD (01IS18037A). The research of L.R. has been partially funded by Deutsche Forschungsgemeinschaft (DFG) through the grant CRC 1114 "Scaling Cascades in Complex Systems" (project A05, project number 235221301). M.O. has been partially funded by Deutsche Forschungsgemeinschaft (DFG) through the grant CRC 1294 "Data Assimilation" (project number 318763901).

Impact statement

The goal of this work is to advance the theoretical understanding of generative modeling based on stochastic processes, eventually leading to improvements in applications as well. While there are potential societal consequences of our work in principle, we do not see any concrete issues and thus believe that we do not specifically need to highlight any.

References

Anderson, B. D. Reverse-time diffusion equation models. *Stochastic Processes and their Applications*, 12(3):313–326, 1982.

Austin, J., Johnson, D. D., Ho, J., Tarlow, D., and Van Den Berg, R. Structured denoising diffusion models in discrete state-spaces. *Advances in Neural Information Processing Systems*, 34:17981–17993, 2021.

Berner, J., Richter, L., and Ullrich, K. An optimal control perspective on diffusion-based generative modeling. *Transactions on Machine Learning Research*, 2024.

Brémaud, P. *Markov chains: Gibbs fields, Monte Carlo simulation, and queues*, volume 31. Springer Science & Business Media, 2013.

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

Ehrenfest, P. and Ehrenfest-Afanassjewa, T. *Über zwei bekannte Einwände gegen das Boltzmannsche H-Theorem.* Hirzel, 1907.

Gardiner, C. W. et al. *Handbook of stochastic methods*, volume 3. Springer Berlin, 1985.

Gillespie, D. T. A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *Journal of computational physics*, 22(4):403–434, 1976.

Gillespie, D. T. Approximate accelerated stochastic simulation of chemically reacting systems. *The Journal of chemical physics*, 115(4):1716–1733, 2001.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Hoogeboom, E., Nielsen, D., Jaini, P., Forré, P., and Welling, M. Argmax flows and multinomial diffusion: Learning categorical distributions. *Advances in Neural Information Processing Systems*, 34:12454–12465, 2021.

Kingma, D., Salimans, T., Poole, B., and Ho, J. Variational diffusion models. *Advances in Neural Information Processing Systems*, 34:21696–21707, 2021.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. 2009.

Kurtz, T. G. The relationship between stochastic and deterministic models for chemical reactions. *The Journal of Chemical Physics*, 57(7):2976–2978, 1972.

Kurtz, T. G. Approximation of discontinuous processes by continuous processes. In *Stochastic Nonlinear Systems in Physics, Chemistry, and Biology: Proceedings of the Workshop Bielefeld, Fed. Rep. of Germany, October 5–11, 1980*, pp. 22–35. Springer, 1981.

Loshchilov, I. and Hutter, F. SGDR: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

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

Meng, C., Choi, K., Song, J., and Ermon, S. Concrete score matching: Generalized score matching for discrete data. *Advances in Neural Information Processing Systems*, 35: 34532–34545, 2022.

Metzner, P. *Transition path theory for Markov processes*. PhD thesis, Freie Universität Berlin, 2008.

Nelson, E. Dynamical theories of Brownian motion. *Press, Princeton, NJ*, 1967.

Nichol, A. Q. and Dhariwal, P. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pp. 8162–8171. PMLR, 2021.

Richter, L. and Berner, J. Improved sampling via learned diffusions. In *International Conference on Learning Representations*, 2024.

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., and Chen, X. Improved techniques for training GANs. *Advances in neural information processing systems*, 29, 2016.

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

Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., and Ganguli, S. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pp. 2256–2265. PMLR, 2015.

Song, J., Meng, C., and Ermon, S. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

Song, Y. and Ermon, S. Improved techniques for training score-based generative models. *Advances in neural information processing systems*, 33:12438–12448, 2020.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021.

Sumita, U., Gotoh, J.-y., and Jin, H. Numerical exploration of dynamic behavior of the Ornstein-Uhlenbeck process via Ehrenfest process approximation. *Applied Probability Trust*, 2004:194–195, 2004.

Sun, H., Yu, L., Dai, B., Schuurmans, D., and Dai, H. Score-based continuous-time discrete diffusion models. *arXiv preprint arXiv:2211.16750*, 2022.

Vahdat, A., Kreis, K., and Kautz, J. Score-based generative modeling in latent space. *Advances in Neural Information Processing Systems*, 34:11287–11302, 2021.

Van Kampen, N. G. *Stochastic processes in physics and chemistry*, volume 1. Elsevier, 1992.

A. Proofs and additional statements

In this section, we provide the proofs of the statements in the main part and state some additional lemmas, which are helpful for our arguments.

Proof of Lemma 2.1. First, we note that the derivation of the backward rates is known, see, e.g., Campbell et al. (2022). For convenience, we repeat the essential part of the proof. Starting with the identity

$$\overleftarrow{r}_t(y|x)p_t(x) = r_t(x|y)p_t(y), \tag{38}$$

we can compute

$$\begin{aligned}
\overleftarrow{r}_t(y|x) &= \frac{p_t(y)}{p_t(x)} r_t(x|y) && (39a)\\
&= \frac{\sum_{x_0 \in \Omega} p_{t|0}(y|x_0)p_0(x_0)}{p_t(x)} r_t(x|y) && (39b)\\
&= \sum_{x_0 \in \Omega} \frac{p_{t|0}(y|x_0)}{p_t(x|x_0)} \frac{p_{t|0}(x|x_0)p_0(x_0)}{p_t(x)} r_t(x|y) && (39c)\\
&= \sum_{x_0 \in \Omega} \frac{p_{t|0}(y|x_0)}{p_{t|0}(x|x_0)}\, p_{0|t}(x_0|x) r_t(x|y) && (39d)\\
&= \mathbb{E}_{x_0 \sim p_{0|t}(x_0|x)} \left[\frac{p_{t|0}(x|x_0)}{p_{t|0}(y|x_0)} \right] r_t(x|y), && (39e)
\end{aligned}$$

which shows the identity. □

Lemma A.1 (Score function as conditional expectation)**.** *Consider the diffusion process X_t defined by the SDE*

$$\mathrm{d}X_t = b(X_t, s)\mathrm{d}t + \sigma(t)\mathrm{d}W_t, \qquad X_0 \sim p_{\mathrm{data}}, \tag{40}$$

with suitable drift function $b : \mathbb{R}^d \times [0, T] \to \mathbb{R}^d$ and diffusion coefficient $\sigma : [0, T] \to \mathbb{R}^{d \times d}$, let p_t^{SDE} be its marginal density and let $p_{t|s}^{\mathrm{SDE}}(x|y) := \mathbb{P}(X_t = x|X_s = y)$ for $t > s \geq 0$ be a transition probability. It then holds

$$\nabla_x \log p_t^{\mathrm{SDE}}(x) = \mathbb{E}_{x_0 \sim p_{0|t}^{\mathrm{SDE}}(x_0|x)} \left[\nabla_x \log p_{t|0}^{\mathrm{SDE}}(x|x_0) \right]. \tag{41}$$

Proof. Noting the identity $p_t^{\mathrm{SDE}}(x) = \int_{\mathbb{R}^d} p_{t|0}^{\mathrm{SDE}}(x|x_0)p_{\mathrm{data}}(x_0)\mathrm{d}x_0$, we can compute

$$\begin{aligned}
\nabla \log p_t^{\mathrm{SDE}}(x) &= \frac{\nabla_x p_t^{\mathrm{SDE}}(x)}{p_t^{\mathrm{SDE}}(x)} && (42a)\\
&= \frac{\int_{\mathbb{R}^d} \nabla_x \log p_{t|0}^{\mathrm{SDE}}(x|x_0)p_{t|0}^{\mathrm{SDE}}(x|x_0)p_{\mathrm{data}}(x_0)\mathrm{d}x_0}{\int_{\mathbb{R}^d} p_{t|0}^{\mathrm{SDE}}(x|x_0)p_{\mathrm{data}}(x_0)\mathrm{d}x_0} && (42b)\\
&= \mathbb{E}_{x_0 \sim p_{0|t}^{\mathrm{SDE}}(x_0|x)} \left[\nabla_x \log p_{t|0}^{\mathrm{SDE}}(x|x_0) \right], && (42c)
\end{aligned}$$

where we used $p_{0|t}^{\mathrm{SDE}}(x_0|x) = \frac{p_{t|0}^{\mathrm{SDE}}(x|x_0)p_{\mathrm{data}}(x_0)}{\int_{\mathbb{R}^d} p_{t|0}^{\mathrm{SDE}}(x|x_0)p_{\mathrm{data}}(x_0)\mathrm{d}x_0}$ by Bayes' formula. □

Lemma A.2 (Conditional expectation as L^2 projection)**.** *Let $A \in \mathbb{R}^d$ and $B \in \mathbb{R}$ be two random variables and let $\varphi \in C(\mathbb{R}^d, \mathbb{R})$. Then the solution to*

$$\varphi^* = \underset{\varphi \in C(\mathbb{R}^d, \mathbb{R})}{\arg\min}\ \mathbb{E}\left[(\varphi(A) - B)^2 \right] \tag{43}$$

is given by

$$\varphi^*(a) = \mathbb{E}[B|A = a]. \tag{44}$$

Proof. Let $\varphi^C(a) = \mathbb{E}[B|A = a]$. We compute

$$\mathbb{E}\left[(\varphi(A) - B)^2\right] = \mathbb{E}\left[(\varphi(A) - \varphi^C(A) + \varphi^C(A) - B)^2\right] \tag{45a}$$

$$= \mathbb{E}\left[(\varphi(A) - \varphi^C(A))^2\right] + \mathbb{E}\left[(\varphi^C(A) - B)^2\right] - 2\,\mathbb{E}\left[(\varphi(A) - \varphi^C(A))(\varphi^C(A) - B)\right], \tag{45b}$$

which is minimized by $\varphi = \varphi^C$ since the last term is equal to[7]

$$\mathbb{E}_A\left[\mathbb{E}_{B|A}\left[(\varphi(A) - \varphi^C(A))(\varphi^C(A) - B)\right]\right] = \mathbb{E}_A\left[(\varphi(A) - \varphi^C(A))\mathbb{E}_{B|A}\left[(\varphi^C(A) - B)\right]\right] = 0. \tag{46}$$

Therefore $\varphi^* = \varphi^C$. □

Proof of Lemma 3.1. We consider the Ehrenfest process as defined in (6), assuming that it starts at $E_S(0) = x_0$. We can write the process as

$$E_S(t) = \sum_{i=1}^{S} Z_i(t) = \sum_{Z_i(0)=0} Z_i(t) + \sum_{Z_i(0)=1} Z_i(t) =: E_{0,S}(t) + E_{1,S}(t), \tag{47}$$

where $E_{1,S}$ is a sum of x_0 independent Bernoulli random variables Z_i with

$$f(t) := \mathbb{P}(Z_i(t) = 1|Z_i(0) = 1) = \frac{1}{2}\left(1 + e^{-t}\right), \tag{48}$$

and where $E_{0,S}$ is the sum of $S - x_0$ random variables Z_i with $\mathbb{P}(Z_i(t) = 1|Z_i(0) = 0) = 1 - f(t) = \frac{1}{2}\left(1 - e^{-t}\right)$. Thus, both $E_{0,S}$ and $E_{1,S}$ are binomial random variables distributed as

$$E_{0,S}(t) \sim B(S - x_0, 1 - f(t)), \qquad E_{1,S}(t) \sim B(x_0, f(t)). \tag{49}$$

□

Proof of Proposition 3.4. We first recall the scaled Ehrenfest process from (11),

$$\widetilde{E}_S(t) := \frac{2}{\sqrt{S}}\left(E_S(t) - \frac{S}{2}\right), \tag{50}$$

and note that $\widetilde{E}_S \in \left\{-\sqrt{S}, -\sqrt{S} + \frac{2}{\sqrt{S}}, \dots, \sqrt{S}\right\}$, where the birth-death transitions transform from ±1 when using E_S to $\pm\frac{2}{\sqrt{S}}$ when using the scaled version $\widetilde{E}_S$. Accordingly, the reverse rates from Lemma 2.1 translate to

$$\overleftarrow{r}_t\left(x \pm \frac{2}{\sqrt{S}}\middle|x\right) = \mathbb{E}_{x_0 \sim p_{0|t}(x_0|x)}\left[\frac{p_{t|0}\left(x \pm \frac{2}{\sqrt{S}}\middle|x_0\right)}{p_{t|0}(x|x_0)}\right] r\left(x\middle|x \pm \frac{2}{\sqrt{S}}\right). \tag{51}$$

Let us introduce the notation (which slightly deviates from the notation in Proposition 3.4)

$$\Delta_\delta p_{t|0}(x|x_0) := \frac{p_{t|0}(x + \delta|x_0) - p_{t|0}(x|x_0)}{\delta} \tag{52}$$

and note the identity

$$p_{t|0}(x + \delta|x_0) = p(x|x_0) + \delta\,\Delta_\delta p_{t|0}(x|x_0), \tag{53}$$

[7] Here the notation $\mathbb{E}_A$ refers to the expectation over A, whereas $\mathbb{E}_{B|A}$ refers to the expectation over B conditional on A.

which is sometimes called *Newton's series for equidistant nodes* and can be seen as a discrete analog of a Taylor series, where, however, terms of order higher than one vanish. We can now compute the first jump moment

$$\begin{aligned}
\overleftarrow{b}(x) &:= \sum_{\delta\in\left\{-\frac{2}{\sqrt{S}},\frac{2}{\sqrt{S}}\right\}} \delta\, \overleftarrow{r}(x+\delta|x) = \sum_{\delta\in\left\{-\frac{2}{\sqrt{S}},\frac{2}{\sqrt{S}}\right\}} \delta\, \mathbb{E}_{x_0\sim p_{0|t}(x_0|x)}\left[\frac{p_{t|0}(x+\delta|x_0)}{p_{t|0}(x|x_0)}\right] r(x|x+\delta) && \text{(54a)}\\
&= \sum_{\delta\in\left\{-\frac{2}{\sqrt{S}},\frac{2}{\sqrt{S}}\right\}} \delta\, \mathbb{E}_{x_0\sim p_{0|t}(x_0|x)}\left[1+\frac{\delta\,\Delta_\delta p_{t|0}(x|x_0)}{p_{t|0}(x|x_0)}\right] r(x|x+\delta) && \text{(54b)}\\
&= \sum_{\delta\in\left\{-\frac{2}{\sqrt{S}},\frac{2}{\sqrt{S}}\right\}} \delta\, r(x|x+\delta) + \sum_{\delta\in\left\{-\frac{2}{\sqrt{S}},\frac{2}{\sqrt{S}}\right\}} \delta^2\, \mathbb{E}_{x_0\sim p_{0|t}(x_0|x)}\left[\frac{\Delta_\delta p_{t|0}(x|x_0)}{p_{t|0}(x|x_0)}\right] r(x|x+\delta) && \text{(54c)}\\
&= \sum_{\delta\in\left\{-\frac{2}{\sqrt{S}},\frac{2}{\sqrt{S}}\right\}} \delta\, r(x|x+\delta) + \mathbb{E}_{x_0\sim p_{0|t}(x_0|x)}\left[\frac{\Delta_{\bar\delta} p_{t|0}(x|x_0)}{p_{t|0}(x|x_0)}\right] \sum_{\delta\in\left\{-\frac{2}{\sqrt{S}},\frac{2}{\sqrt{S}}\right\}} \delta^2\, r(x|x+\delta) + o(\bar\delta) && \text{(54d)}\\
&= -b(x) + D(x)\mathbb{E}_{x_0\sim p_{0|t}(x_0|x)}\left[\frac{\Delta_{\bar\delta} p_{t|0}(x|x_0)}{p_{t|0}(x|x_0)}\right] + o(\bar\delta), && \text{(54e)}
\end{aligned}$$

where for $\bar\delta := \frac{2}{\sqrt{S}}$ we have used that $\Delta_{\bar\delta} p_{0|t}(x|x_0) = \Delta_{-\bar\delta} p_{0|t}(x|x_0) + o(\bar\delta)$ since

$$\begin{aligned}
\frac{p(x|x_0)-p(x-\bar\delta|x_0)}{\bar\delta} &= \frac{p(x+\bar\delta|x_0)-p(x|x_0)}{\bar\delta} + \frac{2\,p(x|x_0)-p(x+\bar\delta|x_0)-p(x-\bar\delta|x_0)}{\bar\delta} && \text{(55a)}\\
&= \frac{p(x+\bar\delta|x_0)-p(x|x_0)}{\bar\delta} + o(\bar\delta), && \text{(55b)}
\end{aligned}$$

as well as

$$\sum_{\delta\in\left\{-\frac{2}{\sqrt{S}},\frac{2}{\sqrt{S}}\right\}} \delta\, r(x|x+\delta) = -\sum_{\delta\in\left\{-\frac{2}{\sqrt{S}},\frac{2}{\sqrt{S}}\right\}} \delta\, r(x+\delta|x) + o(S^{-1/2}), \tag{56}$$

and

$$\sum_{\delta\in\left\{-\frac{2}{\sqrt{S}},\frac{2}{\sqrt{S}}\right\}} \delta^2 r(x|x+\delta) = \sum_{\delta\in\left\{-\frac{2}{\sqrt{S}},\frac{2}{\sqrt{S}}\right\}} \delta^2 r(x+\delta|x) + o(S^{-1}), \tag{57}$$

since

$$\begin{aligned}
r\left(x\middle|x\pm\frac{2}{\sqrt{S}}\right) &= \frac{\sqrt{S}}{4}\left(\sqrt{S}\pm\left(x\pm\frac{2}{\sqrt{S}}\right)\right) = \frac{\sqrt{S}}{4}\left(\sqrt{S}\pm x\right)\pm\frac{1}{2} && \text{(58a)}\\
&= r\left(x\mp\frac{2}{\sqrt{S}}\middle|x\right)\pm\frac{1}{2}. && \text{(58b)}
\end{aligned}$$

□

B. Background on time-continuous Markov jump processes

In this chapter we will provide some background on continuous-time, discrete-space Markov jump processes.

B.1. A brief introduction to Markov jump processes

In this section we will give a brief introduction to time-continuous Markov processes on a discrete state space, which is based on a summary in Metzner (2008, Section 2.2). We refer the interested reader to Gardiner et al. (1985); Van Kampen (1992); Brémaud (2013) for further details.

We denote with $M(t)$ an Ω-valued stochastic process on a discrete (countable) state space Ω with a continuous time parameter $0 \leq t < \infty$. The process is called a Markov process if for all times $t_{k+1} > t_k \geq \cdots \geq t_0 = 0$ and for any

$x_{k+1}, \dots, x_0 \in \Omega$ it holds

$$\mathbb{P}\left(M(t_{k+1}) = x_{k+1} \middle| M(t_k) = x_k, \dots, M(t_0) = x_0\right) = \mathbb{P}\left(M(t_{k+1}) = x_{k+1} \middle| M(t_k) = x_k\right). \tag{59}$$

The process is called *homogeneous* if the transition probability only depends on the time increment $t_{k+1} - t_k$. We denote with

$$p_{t|s}(x|y) = \mathbb{P}(M(t) = x | M(s) = y) \tag{60}$$

the transition probability for times $t > s > 0$ and define the matrix

$$P(t) := \left(p_{t|0}(x|y)\right)_{x,y\in\Omega}. \tag{61}$$

$P(t)$ is a stochastic matrix, i.e.

$$p_{t|0}(x|y) \geq 0, \qquad \sum_{x\in\Omega} p_{t|0}(x|y) = 1, \tag{62}$$

for each time $t \geq 0$ and each $x, y \in \Omega$. The family of transition matrices $\{P(t)\}_{t\geq 0}$ is called transition semi-group since it obeys the *Chapman-Kolmogorov equation*

$$P(t+s) = P(t)P(s) \tag{63}$$

for $s, t \geq 0$ with $P(0) = \mathrm{Id}$.

A local characterization of the transition semigroup of a Markov jump process can be obtained by considering the infinitesimal changes of the transition probabilities. One can show that the limit

$$R = \lim_{t\to 0^+} \frac{P(t) - \mathrm{Id}}{t} \tag{64}$$

exists (entrywise), which is sometimes written as

$$p_{t+\Delta t|t}(x|y) = \delta_{x,y} + r_t(x|y)\Delta t + o(\Delta t), \tag{65}$$

cf. equation (1) in Section 2. The matrix $R = (r(x|y))_{x,y\in\Omega}$ is called infinitesimal generator of the transition semigroup $\{P(t)\}_{t\geq 0}$ because it “generates” the transition semigroup via the relation

$$P(t) = e^{tR} = \sum_{n=0}^{\infty} \frac{t^n}{n!} R^n. \tag{66}$$

One can show that

$$0 \leq r(x|y) < \infty, \qquad \sum_{x\in\Omega} r(x|y) = 0, \tag{67}$$

for all $x, y \in \Omega$ with $x \neq y$, and we can interpret $r(x|y)$ as a transition rate from state y to x, measuring the average number of transitions per unit time. The diagonal elements of R are defined as

$$r(y|y) = -\sum_{x\neq y} r(x|y) \tag{68}$$

for each $y \in \Omega$. Analog to the state-continuous case, it holds the *backward Kolmogorov equation* for the conditional expectation $\psi(t) = \left(\mathbb{E}\left[f(X_t)|X_0 = x\right]\right)_{x\in\Omega}^\top$ for an observable $f : \Omega \to \mathbb{R}$, namely

$$\frac{\mathrm{d}}{\mathrm{d}t}\psi(t) = R\psi(t), \qquad \psi(0) = (f(x))_{x\in\Omega}^\top. \tag{69}$$

Further, for the vector of state probabilities $p(t) := (p_t(x))_{x\in\Omega}^\top$, recalling the notation $p_t(x) := \mathbb{P}(X_t = x)$, it holds the *forward Kolmogorov equation*, also known as *Master equation*, namely

$$\frac{\mathrm{d}}{\mathrm{d}t} p(t) = R\,p(t). \tag{70}$$

For the transition densities we have

$$\frac{\mathrm{d}}{\mathrm{d}t} p_{t|0}(x|y) = \sum_{z \in \Omega} R(x,z) p_{t|0}(z|y), \tag{71}$$

or in matrix notation

$$\frac{\mathrm{d}}{\mathrm{d}t} P(t) = RP(t). \tag{72}$$

It can be solved as

$$\left(p_{t|0}(x|y)\right)_{x,y\in\Omega} = \exp\left(\int_0^t \left(r_s(x|y)\right)_{x,y\in\Omega} \mathrm{d}s\right), \tag{73}$$

where $\exp$ is the matrix exponential.

B.2. Time-transformation of Markov jump processes

Note that we can always transform a Markov jump process with a time dependent rate R_t into one with a time independent rate R using a time transformation. This can be seen by looking at the master equation defined in (72), namely

$$\frac{\mathrm{d}}{\mathrm{d}t} P(t) = R_t P(t), \tag{74}$$

where the rate matrix R_t is time-dependent. For simplicity, let us assume $R_t = \lambda_t R$, where $\lambda : [0,T] \to \mathbb{R}$ and R is time-independent. We can now introduce the new time $\tau = g^{-1}(t)$, where $g : \mathbb{R} \to \mathbb{R}$ is assumed to be invertible. We can now compute

$$\frac{\mathrm{d}}{\mathrm{d}\tau} P(g(\tau)) = \frac{\mathrm{d}P(g(\tau))}{\mathrm{d}t}\frac{\mathrm{d}g(\tau)}{\mathrm{d}\tau} = RP(g(\tau))\lambda_{g(\tau)}\frac{\mathrm{d}g(\tau)}{\mathrm{d}\tau}. \tag{75}$$

Now, choosing $\frac{\mathrm{d}g(\tau)}{\mathrm{d}\tau} = \lambda_{g(\tau)}^{-1}$ and thus $t = g(\tau) = \int_0^\tau \lambda_{g(s)}^{-1} \mathrm{d}s$ (where we have assumed $g(0) = 0$), yields the equation

$$\frac{\mathrm{d}}{\mathrm{d}\tau} P(g(\tau)) = RP(g(\tau)), x \tag{76}$$

where now the rate matrix does not depend on time anymore.

B.3. Convergence of Markov jump processes

The convergence of Markov jump processes to SDEs in the limit of large state spaces (with appropriately scaled jump sizes) has formally been studied via the Kramers-Moyal expansion (Gardiner et al., 1985; Van Kampen, 1992). For more rigorous results we refer to, e.g., Kurtz (1972; 1981).

One can get some intuition by looking at the first two jump moments of the Markov jump process. The first jump moment is defined as

$$b(x) := \lim_{\Delta t \to 0} \frac{1}{\Delta t} \mathbb{E}\left[M(t+\Delta t) - M(t) | M(t) = x\right] \tag{77a}$$

$$= \lim_{\Delta t \to 0} \frac{1}{\Delta t}\left(\sum_{y\in\Omega} y\, p_{t+\Delta t|t}(y|x) - x\sum_{y\in\Omega} p_{t+\Delta t|t}(y|x)\right) \tag{77b}$$

$$= \lim_{\Delta t \to 0} \frac{1}{\Delta t}\left(\sum_{y\in\Omega; y\neq x} (y-x) p_{t+\Delta t|t}(y|x)\right) \tag{77c}$$

$$= \sum_{y\in\Omega; y\neq x} (y-x) r(y|x). \tag{77d}$$

Similarly, the second jump moment is defined as

$$D(x) := \lim_{\Delta t \to 0} \frac{1}{\Delta t} \mathbb{E}\left[\left(M(t+\Delta t) - M(t)\right)\left(M(t+\Delta t) - M(t)\right)^\top | M(t) = x\right] \tag{78a}$$

$$= \sum_{y\in\Omega; y\neq x} (y-x)(y-x)^\top r(y|x). \tag{78b}$$

Algorithm 1 Approximation of conditional expectation for the Ehrenfest process.

```
input  Batch size $K$, gradient steps $M$, two neural networks $\varphi_b$ and $\varphi_d$ with initial parameters $\theta_b^{(0)}$ and $\theta_d^{(0)}$, respectively,
    approximations $\widetilde{p}_{0|t} \approx p_{0|t}$ (typically either by (10) or (14)).
output  Approximations of the conditional expectations appearing in (3).
  for $m \leftarrow 0, \dots, M-1$ do
    Sample data points $x_0^{(1)}, \dots, x_0^{(K)} \sim p_{\mathrm{data}}$.
    Sample terminal times $t_1, \dots, t_K \sim \mathcal{U}(0, T)$.
    Simulate $x_{t_k}^{(k)}$ for each $k \in \{1, \dots, K\}$ according to the forward (scaled) Ehrenfest process (11). Note that every
    dimension can be sampled independently and simulation-free as binomial random variables, see Lemma 3.1.
    Compute two losses:
      $\widehat{\mathcal{L}}_b(\theta_b^{(m)}) = \frac{1}{Kd} \sum_{k=1}^{K} \sum_{i=1}^{d} \left( \varphi_b^{(i)}(x_{t_k}^{(k)}, t_k) - \frac{\widetilde{p}_{t_k|0}^{(i)}\left( x_{t_k}^{(i),(k)} + \frac{2}{\sqrt{S}} \middle| x_0^{(i),(k)} \right)}{\widetilde{p}_{t_k|0}^{(i)}\left( x_{t_k}^{(i),(k)} \middle| x_0^{(i),(k)} \right)} \right)^2 \qquad (79a)$
      $\widehat{\mathcal{L}}_d(\theta_d^{(m)}) = \frac{1}{Kd} \sum_{k=1}^{K} \sum_{i=1}^{d} \left( \varphi_d^{(i)}(x_{t_k}^{(k)}, t_k) - \frac{\widetilde{p}_{t_k|0}^{(i)}\left( x_{t_k}^{(i),(k)} - \frac{2}{\sqrt{S}} \middle| x_0^{(i),(k)} \right)}{\widetilde{p}_{t_k|0}^{(i)}\left( x_{t_k}^{(i),(k)} \middle| x_0^{(i),(k)} \right)} \right)^2 \qquad (79b)$
    Do gradient descent:
      $\theta_b^{(m+1)} \leftarrow \mathrm{step}\left( \theta^{(m)}, \nabla \widehat{\mathcal{L}}_b(\theta_b^{(m)}) \right) \qquad (80a)$
      $\theta_d^{(m+1)} \leftarrow \mathrm{step}\left( \theta^{(m)}, \nabla \widehat{\mathcal{L}}_d(\theta_d^{(m)}) \right) \qquad (80b)$
  end for
```

Note that the drift and diffusion coefficient for SDEs are defined analogously.

C. Computational aspects

In this section we comment on computational aspects of the time-reversal of Markov jump processes.

C.1. Approximation of the conditional expectation

For the approximation of the conditional expectation appearing in the backward rates from Lemma 2.1 we propose Algorithm 1 and for sampling from a time-reversed process with approximated backward rates we propose Algorithm 2. We note the slight abuse of notation, where, depending on the context, $x^{(i)}$ could either refer to the i-th component or i-th sample.

C.2. Learning the reversed transition probability

An alternative way to approximate the backward rates specified in Lemma 2.1 is to approximate the reversed transition probability $p_{0|t}(x_0|x)$ with a tractable distribution $p^\theta_{0|t}(x_0|x)$. To this end, one can consider the loss

$$\mathcal{L}(\theta) := -\mathbb{E}_{t \sim \mathcal{U}(0,T), x_0 \sim p_{\mathrm{data}}, x \sim p_{t|0}(x|x_0)} \left[\log p^\theta_{0|t}(x_0|x) \right]. \tag{87}$$

The following lemma motivates this loss (cf. Proposition 8 in Campbell et al. (2022)).

Lemma C.1. *It holds*

$$\mathcal{L}(\theta) = \mathbb{E}_{t \sim \mathcal{U}(0,T), x \sim p_t(x)} \left[D_{\mathrm{KL}} \left(p_{0|t}(x_0|x) | p^\theta_{0|t}(x_0|x) \right) \right] + C, \tag{88}$$

where C is a constant that does not depend on θ.

Algorithm 2 Sampling from data distribution p_{data}.

```
input Rate of the forward process $r_t$, approximation of the conditional expectations in (3) via $\varphi_b$ and $\varphi_b$ for the birth and
    death transitions, respectively, approximation of terminal distribution $p_{\text{ref}} \approx p_T$, leaping time $\tau > 0$.
output Data sample that is approximately distributed according to $p_{\text{data}}$.
  Sample $x_T \sim p_{\text{ref}}$.
  Set $t \leftarrow T$.
  while $t > 0$ do
    for $i = 1, \dots, d$ do
      Compute backward rates:
        $\overleftarrow{r}_t^{(i)}\left(x^{(i)} + \frac{2}{\sqrt{S}} \middle| x^{(i)}\right) = \varphi_b^{(i)}(x,t)\, r_t^{(i)}\left(x^{(i)} \middle| x^{(i)} + \frac{2}{\sqrt{S}}\right)$   (81)
        $\overleftarrow{r}_t^{(i)}\left(x^{(i)} - \frac{2}{\sqrt{S}} \middle| x^{(i)}\right) = \varphi_d^{(i)}(x,t)\, r_t^{(i)}\left(x^{(i)} \middle| x^{(i)} - \frac{2}{\sqrt{S}}\right)$   (82)
      Draw Poission random variable:
        $\rho_b^{(i)} \sim \text{Pois}\left(\tau\, \overleftarrow{r}_t^{(i)}\left(x^{(i)} + \frac{2}{\sqrt{S}} \middle| x^{(i)}\right)\right)$   (83)
        $\rho_d^{(i)} \sim \text{Pois}\left(\tau\, \overleftarrow{r}_t^{(i)}\left(x^{(i)} - \frac{2}{\sqrt{S}} \middle| x^{(i)}\right)\right)$   (84)
      Do leaping step:
        $x_{t-\tau}^{(i)} \leftarrow x_t^{(i)} + \rho_b^{(i)} \frac{2}{\sqrt{S}} - \rho_d^{(i)} \frac{2}{\sqrt{S}}$   (85)
        $x_{t-\tau}^{(i)} \leftarrow \text{Clamp}(x_{t-\tau}^{(i)}, -\sqrt{S}, \sqrt{S})$   (86)
    end for
    $t \leftarrow t - \tau$
  end while
  Return $x_0$.
```

Proof. Let C be a constant that does not depend on θ. We can compute

$$\mathbb{E}_{t\sim\mathcal{U}(0,T),x\sim p_t(x)}\left[D_{\text{KL}}\left(p_{0|t}(x_0|x)\middle|p^\theta_{0|t}(x_0|x)\right)\right] \tag{89a}$$

$$= \mathbb{E}_{t\sim\mathcal{U}(0,T),x\sim p_t(x)}\left[\mathbb{E}_{x_0\sim p_{0|t}(x_0|x)}\left[\log\frac{p_{0|t}(x_0|x)}{p^\theta_{0|t}(x_0|x)}\right]\right] \tag{89b}$$

$$= \mathbb{E}_{t\sim\mathcal{U}(0,T),x\sim p_t(x),x_0\sim p_{0|t}(x_0|x)}\left[\log\frac{p_{0|t}(x_0|x)}{p^\theta_{0|t}(x_0|x)}\right] \tag{89c}$$

$$= -\mathbb{E}_{t\sim\mathcal{U}(0,T),x_0\sim p_{\text{data}},x\sim p_{t|0}(x|x_0)}\left[\log p^\theta_{0|t}(x_0|x)\right] - C, \tag{89d}$$

where we used the tower property of conditional expectations and the identity $p_t(x)p_{0|t}(x_0|x) = p_{\text{data}}(x_0)p_{t|0}(x|x_0)$. Noting the definition of $\mathcal{L}$ in (87) concludes the proof. □

The above guarantees that $p^\theta_{0|t}(x_0|x) = p_{0|t}(x_0|x)$ if and only if $\mathcal{L}(\theta) = 0$.

We therefore can use Algorithm 3 for approximating the backward rates and Algorithm 4 for sampling the time-reversed process.

Note that all probabilities are probabilities on the discrete set $\Omega \subset \mathbb{Z}$, fulfilling e.g. $\sum_{x_0\in\Omega} p_{0|t}(x_0|x) = 1$ for all $x \in \Omega$ and $t \in [0,T]$. Specifically, $(p_{0|t}(x_0|x))_{x_0,x\in\Omega} \in [0,1]^{(S+1)\times(S+1)}$ is a (stochastic) matrix for all $t \in [0,T]$. In practice,

Algorithm 3 Approximation of reverse transition probability

```
input Batch size $K$, gradient steps $M$, parametrization $p_{0|t}^{\theta^{(0)}}$ with initial parameters $\theta^{(0)}$.
output $p_{0|t}^{\theta^{(M)}} \approx p_{0|t}$.
  for $m \leftarrow 0, \dots, M-1$ do
    Sample data point $x_0^{(1)}, \dots, x_0^{(K)} \sim p_{\mathrm{data}}$.
    Sample terminal times $t_1, \dots, t_K \sim \mathcal{U}(0, T)$.
    Simulate $x_{t_k}^{(k)}$ for each $k \in \{1, \dots, K\}$ according to the forward process.
    Compute $\widehat{\mathcal{L}}(\theta^{(m)}) = -\frac{1}{K} \sum_{k=1}^K \log p_{0|t}^{\theta^{(m)}}(x_0^{(k)}|x_{t_k}^{(k)})$
    $\theta^{(m+1)} \leftarrow \mathrm{step}\left(\theta^{(m)}, \nabla \widehat{\mathcal{L}}(\theta^{(m)})\right)$
  end for
```

Algorithm 4 Sampling from data distribution p_{data}.

```
input Rate of the forward process $r_t$, approximation of forward transition probabilities $\widetilde{p}_{t|0} \approx p_{t|0}$, approximation of
  reverse transition probabilities $p_{0|t}^{\theta} \approx p_{0|t}$, approximation of terminal distribution $p_{\mathrm{ref}} \approx p_T$.
output Data sample that is approximately distributed according to $p_{\mathrm{data}}$.
  Sample $x_T \sim p_{\mathrm{ref}}$.
  Simulate birth-death process from time $t = T$ to $t = 0$ with backward rates

    $$\overleftarrow{r}_t(x \pm 1|x) = \sum_{x_0 \in \Omega} p_{0|t}^{\theta}(x_0|x) \frac{\widetilde{p}_{t|0}(x \pm 1|x_0)}{\widetilde{p}_{t|0}(x|x_0)} r_t(x|x \pm 1). \qquad (95)$$

  Return $x_0$.
```

however, we often model $p_{0|t}(x_0|x)$ with a continuous distribution, parametrized by a neural network, e.g.

$$\widetilde{p}_{0|t}^{\theta}(x_0|x) := \mathcal{N}(x_0; \mu^{\theta}(x,t), \Sigma^{\theta}(x,t)), \tag{90}$$

where mean and covariance are learned with a neural network $\varphi : \mathbb{R} \times [0, T] \to \mathbb{R}^2$, i.e.

$$\varphi^{\theta}(x,t) = (\mu^{\theta}(x,t), \Sigma^{\theta}(x,t))^{\top}. \tag{91}$$

In order to recover probabilities on a discrete set, we can use the cumulative distribution function

$$\Phi(z; x, t) := \int_{-\infty}^{z} \widetilde{p}_{0|t}^{\theta}(x_0|x) \mathrm{d}x_0, \tag{92}$$

which is analytically available for a Gaussian. For a discrete state $x_0 \in \Omega$ we can then approximate the probability via

$$p_{0|t}(x_0|x) \approx p_{0|t}^{\theta}(x_0|x) := \Phi(x_0; x, t) - \Phi(x_0 - 1; x, t) \tag{93}$$

for $x_0 \in \Omega \setminus \{0, S\}$. For $x_0 = 0$ we consider $p_{0|t}^{\theta}(x_0|x) := \Phi(x_0; x, t)$ and for $x_0 = S$ we consider $p_{0|t}^{\theta}(x_0|x) := \Phi(-\infty; x, t) - \Phi(x_0 - 1; x, t)$.

For the forward probabilities $p_{t|0}$, we can either compute the matrix

$$\left(p_{t|0}(x|x_0)\right)_{x_0, x \in \Omega} = \exp\left(\int_0^t \left(r_s(x|x_0)\right)_{x_0, x \in \Omega} \mathrm{d}s\right), \tag{94}$$

where $\exp$ is the matrix exponential, or we can approximate $p_{t|0}(x|x_0)$ with Gaussians due to the convergence properties of the Ehrenfest process.

D. Numerical details

In this section we elaborate on numerical details regarding our experiments in Section 5.

D.1. A tractable Gaussian toy example

In order to illustrate the properties of the Ehrenfest process, we consider the following toy example. Let us start with the SDE setting and consider the data distribution

$$p_{\text{data}}(x) = \sum_{m=1}^{M} \gamma_m \mathcal{N}(x; \mu_m, \Sigma_m), \tag{96}$$

where $\sum_{m=1}^{M} \gamma_m = 1$ and $\mu_m \in \mathbb{R}^d, \Sigma_m \in \mathbb{R}^{d\times d}$. Further, for the inference SDE we consider the Ornstein-Uhlenbeck process

$$\mathrm{d}X_t = -AX_t \mathrm{d}t + B\,\mathrm{d}W_t, \qquad X_0 \sim p_{\text{data}}, \tag{97}$$

with matrices $A, B \in \mathbb{R}^{d\times d}$. For simplicity, let us consider $A = \alpha\mathbb{1}$ and $B = \beta\mathbb{1}$ with $\alpha, \beta \in \mathbb{R}$. Conditioned on an initial condition x_0, the marginal densities of X are then given by

$$p_t^{\text{SDE}}(x; x_0) = \mathcal{N}\left(x; x_0 e^{-\alpha t}, \frac{\beta^2}{2\alpha}\left(1 - e^{-2\alpha t}\right)\mathbb{1}\right). \tag{98}$$

We can therefore compute

$$p_t^{\text{SDE}}(x) = \int_{\mathbb{R}^d} p_t^{\text{SDE}}(x; x_0) p_{\text{data}}(x_0) \mathrm{d}x_0 \tag{99a}$$

$$= \sum_{m=1}^{M} \gamma_m \int_{\mathbb{R}^d} \mathcal{N}\left(x; y_0 e^{-\alpha t}, \frac{\beta^2}{2\alpha}\left(1 - e^{-2\alpha t}\right)\mathbb{1}\right) \mathcal{N}(x_0; \mu_m, \Sigma_m) \mathrm{d}x_0 \tag{99b}$$

$$= \sum_{m=1}^{M} \gamma_m \mathcal{N}\left(x; e^{-\alpha t}\mu_m, \frac{\beta^2}{2\alpha}\left(1 - e^{-2\alpha t}\right)\mathbb{1} + e^{-2\alpha t}\Sigma_m\right). \tag{99c}$$

We can now readily compute the score $\nabla \log p_t^{\text{SDE}}(x)$.

D.2. Connecting the Ehrenfest process to score-based generative modeling

As we have outline in Section 3.2, we can directly link the Ehrenfest process to score-based generative modeling in continuous time and space. In particular, we can use any model that has been trained in the typically used setting for our state-discrete Ehrenfest process. For instance, we can rely on DDPM models, which typically consider the forward SDE

$$\mathrm{d}X_t = -\frac{1}{2}\beta(t) X_t + \sqrt{\beta(t)} \mathrm{d}W_t \tag{100}$$

on the time interval $[0, 1]$, where $\beta : [0, 1] \to \mathbb{R}$ is a function that transforms time. This can be seen by looking at the Fokker-Planck equation. For the process (100) conditioned at the initial value $X_0 = x_0$ it holds that

$$X_t \sim \mathcal{N}\left(\exp\left(-\frac{1}{2}\int_0^t \beta(s) \mathrm{d}s\right) x_0, 1 - \exp\left(-\int_0^t \beta(s)\mathrm{d}s\right)\right). \tag{101}$$

In practice, we choose $\beta(t) := \beta_{\min} + t(\beta_{\max} - \beta_{\min})$ with $\beta_{\min} = 0.1, \beta_{\max} = 20$, as suggested in Song et al. (2021). Note that this typically guarantees that X_1 is approximately distributed according to $\mathcal{N}(0, 1)$, independent of x_0. For our experiments we use a model provided by the Diffuser package. Note that this model is actually not the score, but the scaled score $\widetilde{s}$ and one needs the transformation

$$\nabla \log p_t^{\text{SDE}}(x) = -\frac{\widetilde{s}(x, 1000\, t)}{\sqrt{1 - \exp\left(-\int_0^t \beta(s)\mathrm{d}s\right)}}. \tag{102}$$

In the DDPM framework, the samples X_t of the forward process $p_{t|0}(X_t|x_0)$ can be generated with the reparameterization of (101),

$$X_t = \underbrace{x_0 \exp\left(-\frac{1}{2}\int_0^t \beta(s)\mathrm{d}s\right)}_{\mu_t(x_0)} + \varepsilon \underbrace{\sqrt{1 - \exp\left(-\int_0^t \beta(s)\mathrm{d}s\right)}}_{\sigma_t} \tag{103}$$

where $\varepsilon \sim \mathcal{N}(0,1)$.

The DDPM framework trains a model $\widetilde{\varphi}(X_t,t)$ via the following loss function

$$\mathcal{L}_{\mathrm{OU}}(\widetilde{\varphi}) = \mathbb{E}\left[\left(\widetilde{\varphi}(X_t,t) - \nabla \log p_{t|0}^{\mathrm{OU}}(X_t|x_0)\right)^2\right] = \mathbb{E}\left[\left(\widetilde{\varphi}(X_t,t) + \frac{X_t - \mu_t(x_0)}{\sigma_t^2}\right)^2\right]. \tag{104}$$

Substituting X_t from (103) into the conditional score, the target in the criterion above simplifies to

$$\nabla \log p_{t|0}^{\mathrm{OU}}(X_t|x_0) = -\frac{X_t - \mu_t(x_0)}{\sigma_t^2} = -\frac{\varepsilon}{\sigma_t}. \tag{105}$$

We arrive at the simplified loss with the target

$$\mathcal{L}_{\mathrm{OU}}(\widetilde{\varphi}) = \mathbb{E}\left[\left(\widetilde{\varphi}(X_t,t) - \nabla \log p_{t|0}^{\mathrm{OU}}(X_t|x_0)\right)^2\right] = \mathbb{E}\left[\left(\widetilde{\varphi}(X_t,t) + \frac{\varepsilon}{\sigma_t}\right)^2\right] \tag{106}$$

which is the well known *denoising objective* of the DDPM framework.

As before it then holds

$$\mathbb{E}_{x_0 \sim p_{0|t}(x_0|x)}\left[\frac{p_{t|0}(x \pm \frac{2}{\sqrt{S}}|x_0)}{p_{t|0}(x|x_0)}\right] \approx 1 \pm \frac{2}{\sqrt{S}} \nabla \log p_t^{\mathrm{OU}}(x) = 1 \pm \frac{2}{\sqrt{S}} \widetilde{\varphi}^*(x,t), \tag{107}$$

where $\widetilde{\varphi}^*$ is the minimizer of (106). Further, we may also consider the loss

$$\widetilde{\mathcal{L}}_{\mathrm{OU}}(\widetilde{\varphi}) = \mathbb{E}\left[\left(\widetilde{\varphi}(X_t,t) + \varepsilon\right)^2\right], \tag{108}$$

which yields equivalent solutions when scaling the approximation as follows

$$\mathbb{E}_{x_0 \sim p_{0|t}(x_0|x)}\left[\frac{p_{t|0}(x \pm \frac{2}{\sqrt{S}}|x_0)}{p_{t|0}(x|x_0)}\right] \approx 1 \pm \frac{2}{\sqrt{S}} \nabla \log p_t^{\mathrm{OU}}(x) = 1 \pm \frac{2}{\sigma_t \sqrt{S}} \widetilde{\varphi}^*(x,t), \tag{109}$$

where now $\widetilde{\varphi}^*$ is the minimizer of $\widetilde{\mathcal{L}}_{\mathrm{OU}}$. Crucially, note that due to the equivalences showed in the main part of the paper, we may again replace the Ornstein-Uhlenbeck process X_t with the scaled Ehrenfest process in the loss functions if the state space is large enough.

D.3. Illustrative example

As an illustrative example we choose a distribution which is tractable and perceivable. We model a two dimensional distribution of pixels, which are distributed proportionally to the pixel value of an image of a capital “E”. The visualization of the data distribution is governed by its 33 × 33 pixels and a single sample from the distribution is a black pixel indexed by its location (x, y) on the 33 × 33 pixel grid. The diffusive forward process acts upon the coordinate and diffuses the black pixels into a two dimensional (approximately) binomial distribution at time $t = 1$.

We use the identical architecture as Campbell et al. (2022) used for their illustrative example. Subsequently, the architecture incorporates two residual blocks, each comprising a Multilayer Perceptron (MLP) with a single hidden layer characterized by a dimensionality of 32, a residual connection that links back to the MLP’s input, a layer normalization mechanism, and ultimately, a Feature-wise Linear Modulation (FiLM) layer, which is modulated in accordance to the time embedding. The architecture culminates in a terminal linear layer, delivering an output dimensionality of 2. The time embedding is accomplished utilizing the Transformer’s sinusoidal position embedding technique, resulting in an embedding of dimension 32. This embedding is subsequently refined through an MLP featuring a single hidden layer of dimension 32 and an output dimensionality of 128. In order to generate the FiLM parameters within each residual block, the time embedding undergoes processing via a linear layer, yielding an output dimension of 2.

We test our proposed reverse rate estimators by training them to reconstruct the data distribution at time $t = 0$. For evaluation, we draw 500.000 individual pixels proportionally to the approximated equilibrium distribution and plot their respective histograms at time $t = 0$ in Figure 2.

For training, we sample 1.000.000 pixel values proportional to the gray scale value of the "E" image serving as the true data distribution. We perform optimization with Adam with a learning rate of 0.001 and optimize for 100.000 time steps with a batch size of 2.000.

D.4. MNIST

The MNIST experiments were conducted with the scaled Ehrenfest process. The MNIST data set consists of 28×28 gray scale images which we resized to 32×32 in order to be processable by use our standard DDPM architecture. We used $S = 256^2$ states to ensure 256 states in the range of $[-1, 1]$ with a difference between states of $\frac{2}{\sqrt{S}}$. For optimization, we resorted to the default hyperparameters of Adam (Kingma & Ba, 2014) and used an EMA of 0.99 with a batch size of 128. For the rates we chose the continuous DDPM schedule proposed by Song and we stopped the reverse process at $t = 0.01$ due to vanishing diffusion and resulting high variance rates close to the data distribution.

D.5. Image modeling with CIFAR-10

We employ the standard DDPM architecture from Ho et al. (2020) and adapt the output layer to twice the size when required by the conditional expectation and the Gaussian predictor. The score and first order Taylor approximations did not need to be adapted. Whenever necessary, we adapted the architecture by doubling the final convolutional layer to six channels such that the first half (three channels) predicted the death rate and the second three channels predicted the birth rate. For the time dependent rate λ_t we tried the cosine schedule of (Nichol & Dhariwal, 2021) and the variance preserving SDE schedule of Song et al. (2021). The cosine schedule ensures the expected value of the scaled Ehrenfest process to converge to zero with $\mathbb{E}_{x_0}[x_t] = \cos\left(\frac{\pi}{2}\,t\right)^2 x_0$ and translates to a time dependent jump process rate of $\lambda_t = \frac{1}{4}\,\pi \tan\left(\frac{\pi}{2}\,t\right)$, which is unbounded close to the equilibrium distribution and therefore has to be clamped. We choose $\lambda_t \in [0, 500]$ in our case. Due to numerical considerations regarding the exploding rates, due to diminishing diffusion close to $t = 0$, we restricted the reverse process to times $t \in [0.01, 1]$. In general, we can transform any deliberately long sampling time T to $T = 1$ via the time transformation of the master equation, see Appendix B.2.

We use the standard procedure for training image generation diffusion models (Loshchilov & Hutter, 2016). In particular, we employ a linear learning rate warm up for 5.000 steps and a cosine annealing from 0.0002 to 0.00001 with the Adam optimizer. The batch size was chosen as 256 and an EMA with the factor 0.9999 was applied for the model used for sampling. For sampling we ran the reverse process for 1.000 steps and employed τ-leaping as showcased in Campbell et al. (2022) with a resulting $\tau = 0.001$. We also utilized the predictor-corrector sampling method starting at $t = 0.1$ to the minimum time of $t = 0.01$. Whereas Campbell et al. (2022) reported significant gains performing corrector sampling, we observe behavior close to other state-continuous diffusion models which only apply few or no corrector steps at all.

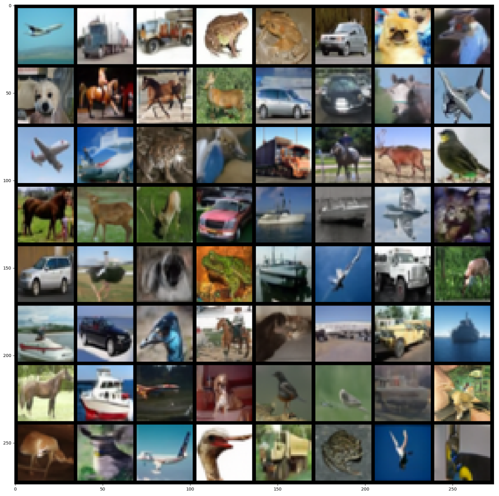

Figure 6. Samples from the reverse scaled Ehrenfest process obtained by finetuning the DDPM architecture with $\mathcal{L}_{\text{Taylor}}$ (18).

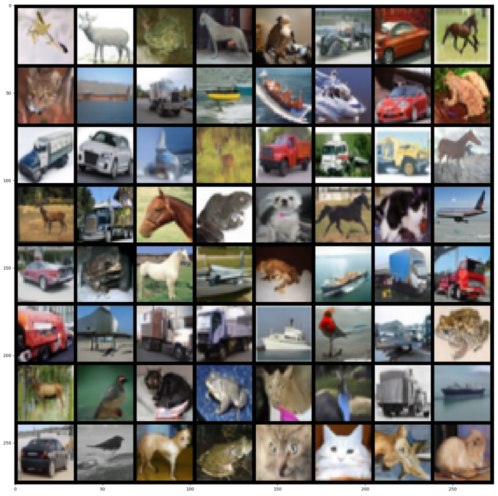

Figure 7. Samples from the reverse scaled Ehrenfest process obtained by finetuning the DDPM architecture with $\mathcal{L}_{\text{Taylor}}$ (18).

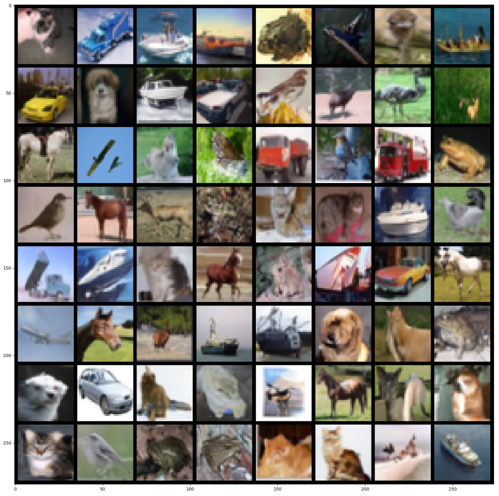

Figure 8. Samples from the reverse scaled Ehrenfest process obtained by finetuning the DDPM architecture with $\mathcal{L}_{\text{OU}}$ (31).

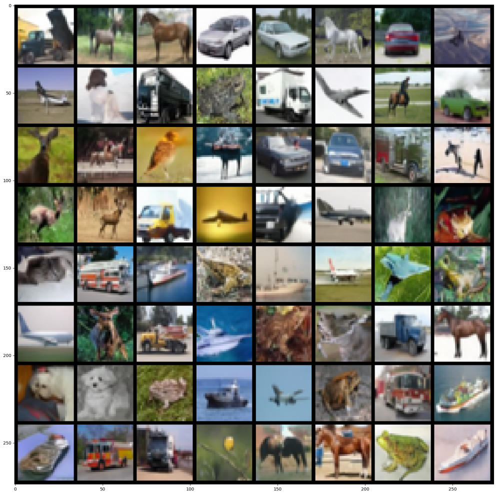

Figure 9. Samples from the reverse scaled Ehrenfest process obtained by finetuning the DDPM architecture with $\mathcal{L}_{\text{OU}}$ (31).