Action Matching: A Variational Method for Learning Stochastic Dynamics from Samples

Anonymous Author(s)
Affiliation
Address
email

Abstract

Stochastic dynamics are ubiquitous in many fields of science, from the evolution of quantum systems in physics to diffusion-based models in machine learning. Existing methods such as score matching can be used to simulate these physical processes by assuming that the dynamics is a diffusion, which is not always the case. In this work, we propose a method called "Action Matching" that enables us to learn a much broader family of stochastic dynamics. Our method requires access only to samples from different time-steps, makes no explicit assumptions about the underlying dynamics, and can be applied even when samples are uncorrelated (i.e., are not part of a trajectory). Action Matching directly learns the underlying mechanism that moves samples in time without modeling the distributions at each time-step. In this work, we showcase how Action Matching can be used for generative modeling for computer vision tasks and discuss potential applications in other areas of science.

1 Introduction

The problem of learning stochastic dynamics is one of the most fundamental problems in many different fields of science. In physics, Porous medium equations in physics (Vázquez, 2007) describes many natural phenomena from this perspective, such as Fokker Planck equation in statistical mechanics, Vlasov equation for plasma, and Nonlinear heat equation. Another prominent example is from Quantum Mechanics where the state of physical systems is a distribution, whose evolution is described by the Schrödinger equation. Recently, stochastic dynamics have achieved very promising results in machine learning applications. The most promising examples of this approach are the diffusion-based generative models (Song et al., 2020; Ho et al., 2020).

In this paper we approach the problem of *Learning Stochastic Dynamics* from their samples. Suppose we observe the time evolution of some random variable X_t with the density q_t, from t_0 to t_1. Having access to samples from the density q_t at different points in time $t \in [t_0, t_1]$, we want to build a model of the dynamics, i.e., we want to learn how to move samples in time such that they respect the marginals q_t.

2 Why is the problem of learning stochastic dynamics well-defined?

Continuity Equation Suppose we have a set of particles, initially distributed as q_{t_0}. Let each particle follow a time-dependent ODE (continuous flow) with the velocity field $v_t(x)$ as follows

$$\frac{\partial}{\partial t} x(t) = v_t(x(t)) \,, \quad x(t_0) = x \,, \tag{1}$$

Under review at the NeurIPS 2022 Workshop on Score-Based Methods. Do not distribute.

From fluid mechanics, we know that the equation governing the evolution of the density q_t of the particles is the *continuity equation*, which is described as

$$\frac{\partial}{\partial t} q_t = -\nabla \cdot (q_t v_t)\,, \tag{2}$$

where $\nabla\cdot$ denotes the divergence operator. Note that even though we arrived at the continuity equation using ODEs, the continuity equation is rich enough to model density evolutions of a wide range of stochastic processes including those of SDEs (see Equation 37 of Song et al. (2020)). In fact, as the following theorem shows, under mild conditions, any density evolution can be modeled by the continuity equation:

Theorem 1 (Adapted from Theorem 8.3.1 of Ambrosio et al. (2008))**.** *Consider a continuous dynamic with the density evolution of q_t, which satisfies mild conditions (absolute continuity in the 2-Wasserstein space of distributions). Then, there exists a unique vector field of the form $v_t^*(x) = \nabla s_t^*(x)$ such that q_t and ∇s_t^* satisfy the continuity equation*

$$\frac{\partial}{\partial t} q_t = -\nabla \cdot (q_t \nabla s_t^*(x))\,. \tag{3}$$

In other words, the ODE $\frac{\partial}{\partial t} x(t) = \nabla s_t^*(x)$ can be used to move samples in time such that the marginals are q_t.

See the Appendix A for more technical details of this theorem.

Using Theorem 1, the problem of learning the dynamics can be boiled down to learning the unique vector field ∇s_t^*, only using samples from q_t. Once we have learned the vector field ∇s_t^*, we can move samples in time by simulating the ODE in Eq. (1) with the velocity ∇s_t^*. The continuity ensures that samples at any given time time $t \in [t_0, t_1]$ are distributed as q_t.

3 Action Matching

The main development of this paper is the "action matching" method, which allows us to recover the *true action* ∇s_t^* of a continuous dynamic and thereby simulate it, while having access only to samples from q_t. In order to do so, we define the *variational action* $s_t(x)$, parameterized by a neural network, that approximates $s_t^*(x)$ when the following objective is minimized

$$\text{ACTION-GAP}(s, s^*) = \frac{1}{2} \int \mathbb{E}_{q_t(x)} \|\nabla s_t(x) - \nabla s_t^*(x)\|^2 dt\,. \tag{4}$$

Note that this objective is intractable, as we do not have access to ∇s^*. We now propose action matching as a variational framework for optimizing this objective.

We first show that the problem of minimizing the intractable Eq. (4), is tightly related to estimating another intractable quantity: the kinetic energy of a continuous dynamics. As discussed in Theorem 2 in Appendix A, we can attribute a kinetic energy $\mathcal{K}$ quantity to any absolutely continuous dynamics q_t, using the true action s^*, as follows

$$\mathcal{K} \coloneqq \frac{1}{2} \int \mathbb{E}_{q_t(x)} \|\nabla s_t^*(x)\|^2 dt\,,$$

where q_t and s^* satisfy the continuity equation. In order to estimate the intractable $\mathcal{K}$, we define a tractable variational *kinetic energy lower bound* (KILBO) as a functional of an arbitrary variational action s as follows:

$$\text{KILBO}(s) = \underbrace{\mathbb{E}_{s_{t_1}(x)}[q_{t_1}(x)] - \mathbb{E}_{s_{t_0}(x)}[q_{t_0}(x)]}_{\text{action-increment}} \underbrace{- \int \mathbb{E}_{q_t(x)} \left[\frac{1}{2} \|\nabla s_t(x)\|^2 + \frac{\partial s_t}{\partial t}(x) \right] dt}_{\text{smoothness (regularization)}} \tag{5}$$

The following proposition establishes that Eq. (4) is the gap between KILBO and the true kinetic energy.

Proposition 1. *For an arbitrary variational action s, $\text{KILBO}(s)$ is a lower bound on the true kinetic $\mathcal{K}(s^*)$, and the gap can be characterized with*

$$\mathcal{K} = \text{KILBO}(s) + \text{ACTION-GAP}(s, s^*)\,. \tag{6}$$

Thus, since $\mathcal{K}$ is not a function of s, the following optimization problems are equivalent

$$\arg\max_s \{\text{KILBO}(s)\} = \arg\min_s \{\text{ACTION-GAP}(s, s^*)\}\,, \tag{7}$$

where the equality is up to an additive constant. The KILBO *gap is tight iff* $\nabla s_t(x) = \nabla s_t^*(x)$.

See Appendix B for the proof. Proposition 1 indicates that maximizing the KILBO results in estimating the true kinetic energy, as well as matching the variational action to the true action. Note that unlike the intractable $\mathcal{K}(s^*)$, maximizing KILBO is tractable, as we can use the samples of q_t to obtain an unbiased low variance estimate of KILBO.

KILBO can be decomposed into an *action-increment* and a smoothness term. If we only optimize the action-increment term, we learn large values for $s_t(x)$ at t_1 and small values for $s_t(x)$ at t_0. In this case, $s_t(x)$ tends to learn a degenerate function with sharp transitions in both x and t directions. The smoothness term acts as a regularization term by penalizing large gradients with respect to both x direction $\left(\frac{1}{2}\|\nabla s_t(x)\|^2\right)$, and the t direction $\left(\frac{\partial s_t(x)}{\partial t}\right)$.

Analogy with Variational Inference and Generative Modeling: In order to better understand the variational objective of KILBO in action matching, we make an analogy with variational inference (VI) in generative modeling. In VI, we have access to samples from the data distribution $q(x)$ and would like to learn a model distribution to approximate it. In action matching, we have access to samples from a continuous dynamic $q_t(x)$ and would like to learn a model distribution for all t, *implicitly*, by learning how to move particles. In VI, we optimize a variational lower bound of the log-likelihood, where the bound is a functional of the variational posterior that approximates the true posterior. In action matching, we optimize a variational lower bound on the true kinetic energy, where the bound is a functional of the variational action, whose gradient approximates the gradient of the true action. In the VI objective, we have a reconstruction term, along with a regularization term which prevents learning degenerate variational posteriors. In the action matching objective, we have an action-increment term, along with a regularization term which prevents learning degenerate variational actions.

4 Generative Modeling using Action Matching

Action Matching can be used for generative modeling. First, we define a dynamics (i.e., noising process) that transforms samples from the target distribution $q_0 = \pi$ to samples of a prior distribution $q_1(x)$ (e.g., standard Gaussian). Action Matching is then used to learn the vector field $\nabla s^\star$ of the chosen dynamics. Finally, we take samples from the prior and transform them to samples of the target by moving them in time with $\nabla s^\star$.

To learn the vector field $\nabla s^\star$, Action Matching only requires samples from the intermediate distributions $q_t, t \in [0, 1]$, that define the noising process. In this paper, we focus on a broad family of noising processes that can be expressed as

$$x_t = f_t(x_0) + \sigma_t \varepsilon, \;\; x_0 \sim \pi(x), \;\; \varepsilon \sim p(\varepsilon)\,, \tag{8}$$

where $f_t(x_0)$ is a deterministic transformation of the data, which is the identity at $t = 0$, and $\sigma_t \varepsilon$ is the scaled noise sample, which is zero at $t = 0$.

Action Matching vs. Score Matching Variants of the score matching method such as Denoising Score Matching (Vincent, 2011) have been proposed for learning diffusion models. For example the dynamics of both VP-SDE and VE-SDE (Song et al., 2020) can be described by the following process

$$x_t = \alpha_t x_0 + \sigma_t \varepsilon, \;\; \varepsilon \sim \mathcal{N}(0, 1)\,, \tag{9}$$

where α_t monotonically contracts the data to zero, and σ_t is the noise schedule changing from $\sigma_0 = 0$ to $\sigma_1 = 1$. Equations (8) and (9) showcase that Action Matching generalizes diffusion, by allowing f_t to be any non-linear function and $\varepsilon \sim p$ any noise model. Note that Denoising Score Matching applies only for the diffusion with tractable conditionals, which restricts the family of sampling processes to Eq. (9). For Action Matching, we only need access to samples of the dynamics, and no knowledge of the conditional distributions is needed.

Algorithm 1 Generative Modeling using Action Matching

Require: dataset $\{x^i\}_{i=1}^N,\ x^i \sim \pi(x) = q_0(x)$
Require: parametric model $s_t(x, \theta)$
  for learning iterations **do**
    sample a batch of data $\{x_0^i\}_i^n \sim q_0(x_0) = \pi(x)$
    sample a batch of noise $\{\varepsilon^i\}_i^n \sim q_1(x_1) = p(\varepsilon)$
    sample times $\{t^i\}_i^n \sim \text{Uniform}[0, 1]$
    sample two batches $\{x_1^i\}_i^n, \{x_{t^i}^i\}_i^n$ using $x_{t^i}^i = f_{t^i}(x_0^i) + \sigma_{t^i}\varepsilon^i$
    $\mathrm{L} = \frac{1}{n}\sum_i^n \left[s_0(x_0^i) - s_1(x_1^i) + \frac{1}{2}\left\|\nabla s_t(x_{t^i}^i)\right\|^2 + \frac{\partial s_t(x_{t^i}^i)}{\partial t} \right]$
    update the model $\theta \leftarrow \text{Optimizer}(\theta, \nabla_\theta \mathrm{L}_\theta)$
  end for
  return trained model $s_t(x, \theta^*)$

Torus Another example we consider is the dynamics of diffusion on a **torus**. We consider every coordinate of the data vector to be in $[0, 1]$ with periodic boundary conditions. Then, the dynamics

$$x_t = (x_0 + \sigma_t \varepsilon) \mod 1, \quad \varepsilon \sim \mathcal{N}(0, 1) \tag{10}$$

converges to the uniform distribution on the torus when $\sigma_t \to \infty$. We demonstrate Action Matching learning these dynamics in experiments.

Once we define the sampling process for $q_t,\ \forall t \in [0, 1]$, we apply Action Matching as described in Algorithm 1. It samples points with different time-steps and then minimizes the objective (4) w.r.t. the parameters θ of $s_t(x, \theta)$. In practice, we found that the performance of Algorithm 1 might be hindered by high variance of the estimate. To reduce the variance of the optimized objective (4) we propose to weight it over time and also adaptively select the distribution of sampled time-steps. We discuss these details in Appendix C.

Sampling We sample from the target distribution via the trained function $s_t(x(t), \theta^*)$ by solving the following ODE backward in time:

$$\frac{\partial}{\partial t} x = -\nabla_x s_t(x(t), \theta^*), \quad x(t = 1) = \varepsilon\,, \quad \varepsilon \sim p(\varepsilon). \tag{11}$$

Evaluating the log-likelihood we can evaluate the unnormalized density of the trained model at any point x by integrating the same ODE forward, i.e.,

$$\log q_0(x) = \log q_1(x(1)) + \int_0^1 dt\ \nabla^2 s_t(x(t)), \quad \frac{\partial}{\partial t} x = \nabla_x s_t(x(t), \theta^*), \quad x(t = 0) = x, \tag{12}$$

where we assume $q_1(x)$ to be known prior distribution.

5 Experiments

To demonstrate that Action Matching is a practical approach we consider image generation task. Although we do not think that this is the main application of Action Matching, we consider this task to be a good illustration of Action Matching practical potential since image generation is a challenging high-dimensional stochastic dynamics.

We learn Action Matching for MNIST and CelebA (Liu et al., 2015) datasets. For the architecture of the neural network parameterizing s_t, we follow (Salimans & Ho, 2021). In more details, we parameterize $s_t(x)$ as $\|\text{unet}(t, x) - x\|^2$, where $\text{unet}(t, x)$ is the U-net architecture (Ronneberger et al., 2015). For diffusion, we take VP-SDE from (Song et al., 2020), which corresponds to $\alpha_t = \exp(-\frac{1}{2}\int \beta(s)ds)$ and $\sigma_t = \sqrt{1 - \exp(-\int \beta(s)ds)}$, where $\beta(s) = 0.1 + 19.9t$. For torus, we first normalize the data such that every pixel is in $[0.25, 0.75]$ and take $\sigma_t = t$. We train all models for $300k$ iterations and demonstrate sampled images in Appendix E.

References

Luigi Ambrosio, Nicola Gigli, and Giuseppe Savaré. *Gradient flows: in metric spaces and in the space of probability measures*. Springer Science & Business Media, 2008.

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

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*, pp. 3730–3738, 2015.

Felix Otto. The geometry of dissipative evolution equations: the porous medium equation. 2001.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241. Springer, 2015.

Tim Salimans and Jonathan Ho. Should ebms model the energy or the score? In *Energy Based Models Workshop-ICLR 2021*, 2021.

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

Juan Luis Vázquez. *The porous medium equation: mathematical theory*. Oxford University Press on Demand, 2007.

Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural computation*, 23(7):1661–1674, 2011.

A Absolutely Continuous Curves and Continuity Equation

Theorem 2 (Absolutely Continuous Curves in $\mathcal{P}_2(X)$ and Continuity Equation, adopted from Theorem 8.3.1 of Ambrosio et al. (2008))**.** *Consider a continuous dynamic $q_t : [t_0, t_1] \to \mathcal{P}_2(X)$ which is defined as an absolutely continuous curve in the 2-Wasserstein space of distributions $\mathcal{P}_2(X)$. There exists unique vector field in the form $\nabla s_t^*(x)$ such that $(q_t, \nabla s_t^*)$ satisfies the continuity equation*

$$\frac{\partial}{\partial t} q_t = -\nabla \cdot (q_t \nabla s_t^*) \,. \tag{13}$$

In other words, the dynamics $\frac{\partial}{\partial t} x(t) = \nabla s_t^*(x)$ generates q_t. Moreover, among all the vector fields $v_t(x)$ generating the dynamics, $\nabla s_t^*(x)$ achieves the minimum kinetic energy functional $\mathcal{K}$ defined as

$$\mathcal{K}(v_t) \coloneqq \frac{1}{2} \int \|v_t\|^2 q_t(x) dx dt \,, \qquad \nabla s_t^* = \arg\min_{v_t} \mathcal{K}(v_t) \,, \tag{14}$$

Furthermore, this unique $\nabla s_t^*(x)$ can be used to find the arc length of the curve

$$\mathbf{L}(s) = \int \|\nabla s_t^*\| q_t(x) dx dt \,. \tag{15}$$

Theorem 2 illustrates that the vector field $\nabla s_t^*(x)$ plays a crucial role in defining any continuous dynamics. In fact, in Riemannian geometry interpretation of the Wasserstein space (Otto, 2001), $\nabla s_t^*(x)$ is called the *tangent vector field* of the continuous dynamics.

B Action Matching

Proposition. *For an arbitrary variational action s, $\text{KILBO}(s)$ is a lower bound on the true kinetic $\mathcal{K}(s^*)$, and the gap can be characterize with*

$$\mathcal{K} = \text{KILBO}(s) + \text{ACTION-GAP}(s, s^*) \,. \tag{16}$$

Thus, since $\mathcal{K}$ is not a function of s, the following optimization problems are equivalent

$$\arg\max_s \{\text{KILBO}(s)\} = \arg\min_s \{\text{ACTION-GAP}(s, s^*)\}\,, \tag{17}$$

where the equality is up to an additive constant. The KILBO *gap is tight iff* $\nabla s_t(x) = \nabla s_t^*(x)$.

Proof.

$$\int dx\ dt\ q_t\omega_t\|\nabla s - \nabla s^*\|^2 = \int dx\ dt\ q_t\omega_t\frac{1}{2}\|\nabla s\|^2 - \int dx\ dt\ q_t\omega_t\langle\nabla s, \nabla s^*\rangle = \tag{18}$$

$$= \int dx\ dt\ q_t\omega_t\frac{1}{2}\|\nabla s\|^2 + \int dx\ dt\ s\omega_t\langle\nabla, q_t\nabla s^*\rangle = \tag{19}$$

$$= \int dx\ dt\ q_t\omega_t\frac{1}{2}\|\nabla s\|^2 - \int dx\ dt\ s\omega_t\dot{q}_t = \tag{20}$$

$$= \int dx\ dt\ q_t\omega_t\frac{1}{2}\|\nabla s\|^2 + \int dx\ dt\ q_t\left(\frac{\partial s}{\partial t}\omega_t + s\frac{\partial\omega_t}{\partial t}\right) - \tag{21}$$

$$- \int dx\ s_{t_1}\omega_{t_1}q_{t_1} + \int dx\ s_{t_0}\omega_{t_0}q_{t_0} \tag{22}$$

□

C Generative Modeling in Practice

Algorithm 2 Generative Modeling using Action Matching (In Practice)

Require: dataset $\{x^i\}_{i=1}^N,\ x^i \sim \pi(x) = q_0(x)$
Require: parameteric model $s_t(x, \theta)$, weight schedule $\omega(t)$
  for learning iterations **do**
    sample a batch of data $\{x_0^i\}_i^n \sim q_0(x_0)$
    sample a batch of noise $\{\varepsilon^i\}_i^n \sim q_1(x_1)$
    sample times $\{t^i\}_i^n \sim p(t)$
    sample two batches $\{x_1^i\}_i^n, \{x_{t^i}^i\}_i^n$ using $x_{t^i}^i = f_{t^i}(x_0^i) + \sigma_{t^i}\varepsilon^i$
    $\mathrm{L} = \sum_i^n \frac{1}{p(t^i)}\left[s_0(x_0^i)\omega(0) - s_1(x_1^i)\omega(1) + \frac{1}{2}\left\|\nabla s_t(x_{t^i}^i)\right\|^2\omega(t^i) + \frac{\partial s_t(x_{t^i}^i)}{\partial t}\omega(t^i) + s_t(x_t^i)\frac{\partial\omega(t^i)}{\partial t^i}\right]$
    update the model $\theta \leftarrow \text{Optimizer}(\theta, \nabla_\theta \mathrm{L}_\theta)$
  end for
  return trained model $s_t(x, \theta^*)$

In practice, we found that the naive application of Action Matching (Algorithm 1) for complicated dynamics such as image generation might exhibit poor convergence due to the large variance of objective estimate. Moreover, the optimization problem

$$\min_{s_t} \frac{1}{2}\int q_t^*(x)\|\nabla s_t(x) - \nabla s_t^*(x)\|^2 dxdt \tag{23}$$

might be ill posed due to the singularity of the ground truth vector field ∇s_t^*. Indeed, consider the sampling process

$$x_t = f_t(x_0) + \sigma_t\varepsilon,\ \ x_0 \sim \pi(x),\ \ \varepsilon \sim \mathcal{N}(x\,|\,0, 1)\,, \tag{24}$$

where the target distribution is a mixture of delta-functions

$$\pi(x) = \frac{1}{N}\sum_i^N \delta(x - x^i). \tag{25}$$

Denoting the distribution of x_t as $q_t^*(x)$, we can solve the continuity equation

$$\frac{\partial q_t^*}{\partial t} = -\nabla\cdot(q_t^*\nabla s_t^*) \tag{26}$$

analytically (see Appendix D). The ground truth vector field is

$$\nabla s_t^* = \frac{1}{\sum_i q_t^i} \sum_i q_t^i \left[(x - f_t(x^i)) \frac{\partial}{\partial t} \log \sigma_t + \frac{\partial f_t(x^i)}{\partial t} \right], \quad q_t^i(x) = \mathcal{N}(x \,|\, f_t(x^i), \sigma_t^2). \tag{27}$$

For generative modeling, it's essential that $q_0 = \pi(x)$; hence, $\lim_{t\to 0} \sigma_t = 0$ and $\lim_{t\to 0} f_t(x) = x$. Assuming that σ_t^2 is continuous and differentiable at 0, in the limit, we have

$$\lim_{t\to 0} \|\nabla s_t^*(x)\|^2 \propto \lim_{t\to 0} \frac{1}{\sigma_t^2}, \quad \text{and} \quad \lim_{t\to 0} \frac{1}{2} \int q_t^*(x) \|\nabla s_t^*(x)\|^2 dx dt \propto \lim_{t\to 0} \frac{1}{\sigma_t^3}. \tag{28}$$

Thus, the loss can be properly defined only on the interval $t \in (\delta, 1]$, where $\delta > 0$. In practice, we want to set δ as small as possible; hence, ideally, we want to learn s_t on the whole interval $t \in [0, 1]$. We can get rid of the singularity just by reweighting the objective in time, i.e.,

$$\frac{1}{2} \int q_t^*(x) \|\nabla s_t(x) - \nabla s_t^*(x)\|^2 dx dt \implies \frac{1}{2} \int \omega(t) q_t^*(x) \|\nabla s_t(x) - \nabla s_t^*(x)\|^2 dx dt. \tag{29}$$

To give an example, we can take $\sigma_t = \sqrt{t}$ and $f_t(x) = x\sqrt{1-t}$, then $\omega(t) = (1-t)t^{3/2}$ cancels out the singularities at $t = 0$ and $t = 1$.

The second modification of the original Algorithm 1 is the sampling of time-steps for the estimation of the time integral. Namely, the optimization of (29) is equivalent to the minimization of the following objective

$$\mathrm{L}(s) = \underbrace{\int \omega(t_1) s_{t_1}(x) q_{t_1}^*(x) dx - \int \omega(t_0) s_{t_0}(x) q_{t_0}^*(x) dx}_{\text{boundary part}} + \tag{30}$$

$$\underbrace{+ \int_{t_0}^{t_1} \int q_t^*(x) \left[\frac{1}{2} \omega(t) \|\nabla s_t(x)\|^2 + \omega(t) \frac{\partial s_t(x)}{\partial t} + s_t(x) \frac{\partial \omega(t)}{\partial t} \right] dx dt}_{\text{middle part}}, \tag{31}$$

which consists of two parts. Estimation of the boundary part involves only sampling from $q_{t_0}^*$ and $q_{t_1}^*$, while the middle part estimate depends on the distribution of time samples, i.e.,

$$\int_{t_0}^{t_1} \underbrace{\frac{p(t)}{p(t)}}_{=1} \int q_t^*(x) \left[\frac{1}{2} \omega(t) \|\nabla s_t(x)\|^2 + \omega(t) \frac{\partial s_t(x)}{\partial t} + s_t(x) \frac{\partial \omega(t)}{\partial t} \right] dx dt \simeq \tag{32}$$

$$\simeq \mathbb{E}_{t\sim p(t)} \mathbb{E}_{x\sim q_t^*(x)} \frac{1}{p(t)} \left[\frac{1}{2} \omega(t) \|\nabla s_t(x)\|^2 + \omega(t) \frac{\partial s_t(x)}{\partial t} + s_t(x) \frac{\partial \omega(t)}{\partial t} \right]. \tag{33}$$

Note that for every choice of $p(t)$ we get an unbiased estimate of the original objective function. Thus, we can design $p(t)$ to reduce the variance of the middle part of the objective. In our experiments, we observed that simply taking $p(t)$ proportionally to the standard deviation of the corresponding integrand significantly reduces the variance, i.e.,

$$p(t) \propto \sqrt{\mathbb{E}_{x\sim q_t^*} (\zeta_t - \mathbb{E}_{x\sim q_t^*} \zeta_t)^2}, \quad \zeta_t = \frac{1}{2} \omega(t) \|\nabla s_t(x)\|^2 + \omega(t) \frac{\partial s_t(x)}{\partial t} + s_t(x) \frac{\partial \omega(t)}{\partial t}. \tag{34}$$

We implement sampling from this distribution by aggregating the estimated variances throughout the training with exponential moving average and then follow by linear interpolation between the estimates. We refer the reader to the supplemented code for further details of implementation.

D "Sparse Data" Regime

We start with the case where the dataset consists only of a single point $x_0 \in \mathbb{R}^d$

$$q_0(x) = \delta(x - x_0), \quad k_t(x_t \,|\, x) = \mathcal{N}(x_t \,|\, f_t(x), \sigma_t^2). \tag{35}$$

Then the distribution at time t is

$$q_t(x) = \int dx' \; q_0(x') k_t(x \,|\, x') = \mathcal{N}(x \,|\, f_t(x_0), \sigma_t^2). \tag{36}$$

The ground truth vector field v comes from the continuity equation

$$\frac{\partial q_t}{\partial t} = -\langle \nabla, q_t v\rangle \quad \Longrightarrow \quad \frac{\partial}{\partial t}\log q_t = -\langle \nabla \log q_t, v\rangle - \langle \nabla, v\rangle. \tag{37}$$

For our dynamics, we have

$$\begin{aligned}
\frac{\partial}{\partial t}\log q_t = & \frac{\partial}{\partial t}\Big[-\frac{d}{2}\log(2\pi\sigma_t^2) - \frac{1}{2\sigma_t^2}\|x - f_t(x_0)\|^2\Big] && (38)\\
= & -d\frac{\partial}{\partial t}\log\sigma_t + \frac{1}{\sigma_t^2}\|x - f_t(x_0)\|^2\frac{\partial}{\partial t}\log\sigma_t + \frac{1}{\sigma_t^2}\Big\langle x - f_t(x_0), \frac{\partial f_t(x_0)}{\partial t}\Big\rangle && (39)\\
= & -d\frac{\partial}{\partial t}\log\sigma_t + \frac{1}{\sigma_t^2}\Big\langle x - f_t(x_0), (x - f_t(x_0))\frac{\partial}{\partial t}\log\sigma_t + \frac{\partial f_t(x_0)}{\partial t}\Big\rangle; && (40)\\
\nabla\log q_t = & -\frac{1}{\sigma_t^2}(x - f_t(x_0)); && (41)\\
\frac{\partial}{\partial t}\log q_t = & -d\frac{\partial}{\partial t}\log\sigma_t - \Big\langle \nabla\log q_t, (x - f_t(x_0))\frac{\partial}{\partial t}\log\sigma_t + \frac{\partial f_t(x_0)}{\partial t}\Big\rangle. && (42)
\end{aligned}$$

Matching the corresponding terms in the continuity equation, we get

$$v = (x - f_t(x_0))\frac{\partial}{\partial t}\log\sigma_t + \frac{\partial f_t(x_0)}{\partial t}. \tag{43}$$

For the set of delta-functions, we denote

$$q_0(x) = \sum_i \delta(x - x^i), \quad q_t(x) = \sum_i q_t^i(x), \quad q_t^i(x) = \mathcal{N}(x \mid f_t(x^i), \sigma_t^2). \tag{44}$$

Due to the linearity of the continuity equation w.r.t. q, we have

$$\sum_i \frac{\partial q_t^i}{\partial t} = \sum_i \langle \nabla, q_t^i v\rangle \quad \Longrightarrow \quad \sum_i q_t^i\Big(\frac{\partial}{\partial t}\log q_t^i + \langle \nabla\log q_t^i, v\rangle + \langle \nabla, v\rangle\Big) = 0. \tag{45}$$

We first solve the equation for $\frac{\partial f_t}{\partial t} = 0$, then for $\frac{\partial}{\partial t}\log\sigma_t = 0$ and join the solutions. For $\frac{\partial f_t}{\partial t} = 0$, we look for the solution in the following form

$$v_\sigma = \frac{A}{\sum_i q_t^i}\sum_i \nabla q_t^i, \quad q_t^i(x) = \mathcal{N}(x \mid f_t^i(x^i), \sigma_t^2). \tag{46}$$

Then we have

$$\begin{aligned}
\langle \nabla, v_\sigma\rangle = & \Big\langle \nabla\frac{A}{\sum_i q_t^i}, \sum_i \nabla q_t^i\Big\rangle + \frac{A}{\sum_i q_t^i}\sum_i \nabla^2 q_t^i && (47)\\
= & -\frac{A}{(\sum_i q_t^i)^2}\Big\|\sum_i \nabla q_t^i\Big\|^2 + \frac{A}{\sum_i q_t^i}\sum_i q_t^i\Big[\big\|\nabla\log q_t^i\big\|^2 - \frac{d}{\sigma^2}\Big], && (48)\\
\Big(\sum_i q_t^i\Big)\langle \nabla, v_\sigma\rangle = & -\frac{A}{\sum_i q_t^i}\Big\|\sum_i \nabla q_t^i\Big\|^2 + A\sum_i q_t^i\Big[\big\|\nabla\log q_t^i\big\|^2 - \frac{d}{\sigma^2}\Big], && (49)
\end{aligned}$$

and from (45) we have

$$\sum_i q_t^i\Big(-d\frac{\partial}{\partial t}\log\sigma_t + \Big\langle \nabla\log q_t^i, v_\sigma + \sigma_t^2\frac{\partial}{\partial t}\log\sigma_t\nabla\log q_t^i\Big\rangle + \langle \nabla, v_\sigma\rangle\Big) = 0. \tag{50}$$

From these two equations we have

$$\begin{aligned}
\sum_i q_t^i\langle \nabla, v_\sigma\rangle = & -\frac{A}{\sum_i q_t^i}\Big\|\sum_i \nabla q_t^i\Big\|^2 + A\sum_i q_t^i\Big[\big\|\nabla\log q_t^i\big\|^2 - \frac{d}{\sigma^2}\Big] = && (51)\\
= & \sum_i q_t^i\Big(d\frac{\partial}{\partial t}\log\sigma_t\Big) - \frac{A}{\sum_i q_t^i}\Big\|\sum_i \nabla q_t^i\Big\|^2 - \sigma_t^2\frac{\partial}{\partial t}\log\sigma_t\sum_i q_t^i\big\|\nabla\log q_t^i\big\|^2. && (52)
\end{aligned}$$

Thus, we have

$$A = -\sigma_t^2 \frac{\partial}{\partial t} \log \sigma_t. \tag{53}$$

For $\frac{\partial}{\partial t} \log \sigma_t = 0$, we simply check that the solution is

$$v_f = \frac{1}{\sum_i q_t^i} \sum_i q_t^i \frac{\partial f_t(x^i)}{\partial t}. \tag{54}$$

Indeed, the continuity equation turns into

$$\sum_i q_t^i \left(\left\langle \nabla \log q_t^i, v_f - \frac{\partial f_t(x^i)}{\partial t} \right\rangle + \langle \nabla, v_f \rangle \right) = 0. \tag{55}$$

From the solution and the continuity equation we write $\sum_i q_i \langle \nabla, v_f \rangle$ in two different ways.

$$\begin{aligned} \sum_i q_t^i \langle \nabla, v_f \rangle = & -\frac{1}{\sum_i q_t^i} \left\langle \sum_i \nabla q_t^i, \sum_i q_t^i \frac{\partial f_t(x^i)}{\partial t} \right\rangle + \sum_i \left\langle \nabla q_t^i, \frac{\partial f_t(x^i)}{\partial t} \right\rangle & (56) \\ = & -\left\langle \sum_i \nabla q_t^i, v_f \right\rangle + \sum_i \left\langle \nabla q_t^i, \frac{\partial f_t(x^i)}{\partial t} \right\rangle & (57) \end{aligned}$$

Thus, we see that (54) is indeed a solution.

Finally, unifying v_σ and v_f, we have the full solution

$$v = -\left(\frac{\partial}{\partial t} \log \sigma_t \right) \frac{\sigma_t^2}{\sum_i q_t^i} \sum_i \nabla q_t^i + \frac{1}{\sum_i q_t^i} \sum_i q_t^i \frac{\partial f_t(x^i)}{\partial t}, \quad q_t^i(x) = \mathcal{N}(x \mid f_t(x^i), \sigma_t^2), \tag{58}$$

$$v = \frac{1}{\sum_i q_t^i} \sum_i q_t^i \left[(x - f_t(x^i)) \frac{\partial}{\partial t} \log \sigma_t + \frac{\partial f_t(x^i)}{\partial t} \right] \tag{59}$$

E Image Examples

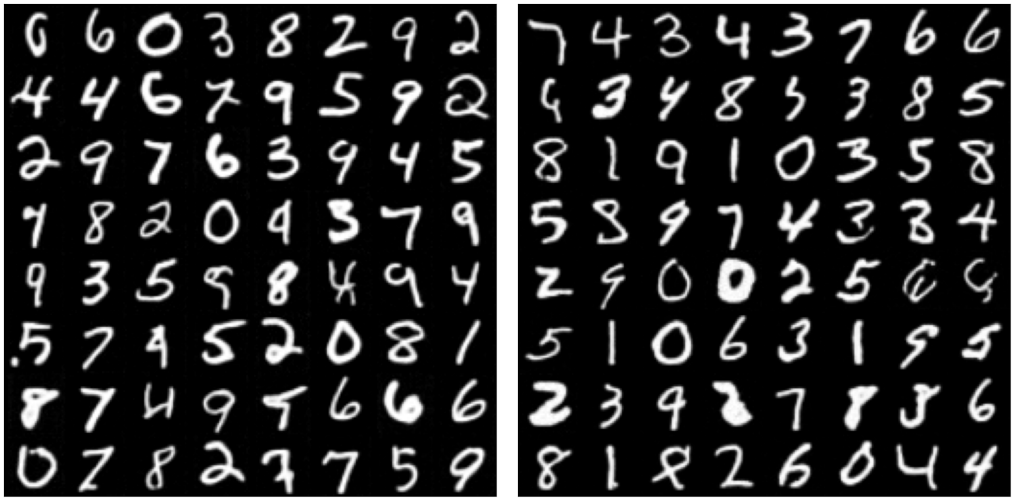

Figure 1: Action Matching on MNIST diffusion (on the left), torus (on the right).

Figure 2: Action Matching on CelebA diffusion (on the left), torus (on the right).