# Strong Stochastic Flow Maps

**Sam McCallum**[1*] **Zander W. Blasingame**[2*] **Timothy Herschell**[1] **Niklas Rindtorff**[2]
**Alexander Tong**[2†] **James Foster**[1†]
[1]University of Bath [2]AITHYRA
[*]Equal contribution [†]Equal supervision
Correspondence to sm2942@bath.ac.uk and zblasingame@aithyra.at

## Abstract

Flow and diffusion models generate high-quality samples in many modalities; however, many network evaluations are required during inference due to numerical integration of an underlying differential equation. Flow maps alleviate this problem by learning the solution map of the differential equation directly, enabling few-step sampling. Yet, current methods are restricted to approximating the solution map of ODEs. These methods can be used to learn the transition kernel of an SDE, thereby obtaining a solution map that recovers the marginal distributions of the process (weak convergence) rather than the solution path (strong convergence). We propose STRONG STOCHASTIC FLOW MAPS (SSFMs) as a novel framework for learning the *strong* solution map of additive-noise SDEs, directly generalizing deterministic flow maps to the stochastic setting. Further, a polynomial approximation to Brownian motion is introduced and shown to converge pathwise. These results enable a simulation-free training objective for the solution map of diffusion models. We demonstrate that SSFMs outperform previous stochastic flow map methods on image generation and enable few-step sampling of molecular systems. Code available at https://github.com/sammccallum/ssfm.

## 1 Introduction

Deep generative models based on *neural differential equations* (Chen et al. 2018; Kidger 2022) have become one of the most successful model classes for solving a variety of problems such as generative modeling (Song, Sohl-Dickstein, et al. 2021; Liu, Gong, et al. 2023) and time-series data (Oh et al. 2024; Walker et al. 2024). The application of neural differential equations to generative modeling—diffusion models (Sohl-Dickstein et al. 2015; Ho et al. 2020; Song, Sohl-Dickstein, et al. 2021) and flow matching models (Peluchetti 2021; Lipman, Chen, et al. 2023; Liu, Gong, et al. 2023)—have become state-of-the-art for many different tasks including image (Rombach et al. 2022) and video (Blattmann et al. 2023) generation, protein design (Watson et al. 2023; Rector-Brooks et al. 2026), and Boltzmann sampling (Rehman et al. 2026).

While quite expressive generative models, neural differential equations often require a large *number of function evaluations* (NFEs) of the learned vector fields to integrate the underlying differential equation. As such there has been great interest in learning how to improve the computational efficiency of these models, *e.g.*, proposing better numerical schemes to reduce the number of NFEs whilst maintaining similar performance (Lu et al. 2022; Gonzalez et al. 2023; Zhang and Chen 2023). Recently, another direction has looked at how to learn the solution, or flow, map associated with a neural *ordinary differential equation* (ODE) (Liu, Gong, et al. 2023; Song, Dhariwal, et al. 2023; Boffi et al. 2024; Heek et al. 2024; Kim et al. 2024; Geng, Deng, et al. 2025; Sabour et al. 2025). These methods, which learn a *neural flow map*, have obtained state-of-the-art performance with low NFEs in image generation (Geng, Deng, et al. 2025; Geng, Lu, et al. 2025) and Boltzmann sampling (Rehman et al. 2026) compared to the 100s of steps required with diffusion models.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

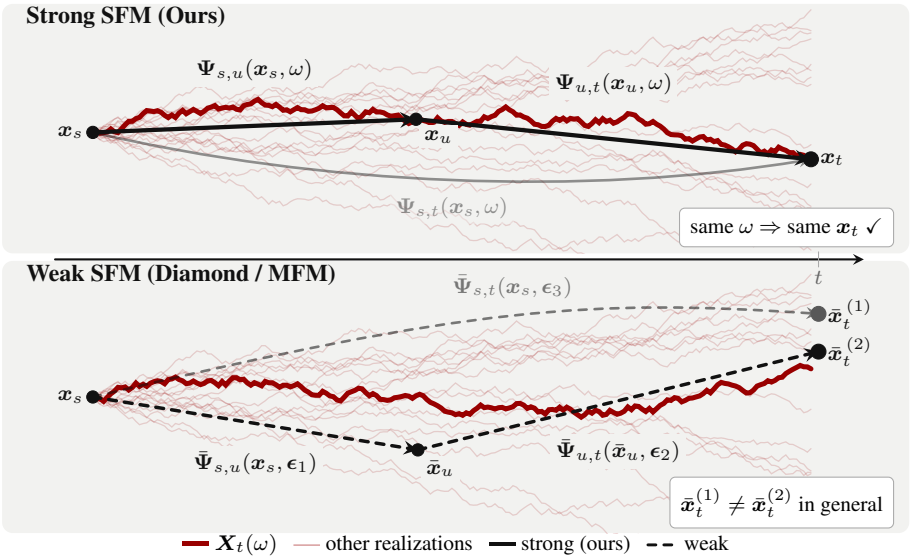


Figure 1: **Strong *vs*. weak stochastic flow maps.** *Top*: The strong stochastic flow map solution is consistent pathwise for a given realization of the Brownian motion $\boldsymbol{W}_t(\omega)$. *Bottom*: The weak stochastic flow map samples independently from the marginal distribution at each time.

Previous work has focused on learning a *deterministic* map from the source noise to the target distribution. Recent work has extended these deterministic maps to the stochastic setting by approximating the transition distributions of a stochastic process (Holderrieth, Chen, et al. 2026; Passaro et al. 2026; Potaptchik et al. 2026). However, such approaches do not allow for the estimation of pathwise observables as they are fundamentally decoupled from the underlying stochastic differential equation (SDE) and are only capable of computing *weak* (convergence in distribution) solutions of the stochastic process (Øksendal 2003).

We introduce STRONG STOCHASTIC FLOW MAPS (SSFMs), a novel framework which obtains the *strong* (convergence in path) solution map to additive-noise SDEs. This naturally extends ODE-based flow maps, which are pathwise solution maps, to the stochastic setting. Specifically, given a realization of the Brownian path and an initial condition, we learn the pathwise solution map of the additive-noise SDE. To efficiently implement such a model we require a novel set of tools which we construct in this work. Our main contributions are summarized as follows:

- We develop a novel framework for learning the strong solution to additive-noise SDEs in contrast with prior works which consider weak approximations.
- We formalize a training objective for learning this map which admits a simulation-free algorithm for obtaining the strong solution map of diffusion models.
- We introduce a polynomial approximation of the Brownian motion and prove that it converges in $\alpha$-Hölder distance, has tractable coefficients, and admits closed form Chen relations.
- We demonstrate this formulation achieves state-of-the-art performance on few-step stochastic image generation and enables sampling of equilibrium molecular conformations with as few as 1-2 NFEs.

## 2 Background and related work

**Neural SDEs.** Neural SDEs are the stochastic extension of neural ODEs (Chen et al. 2018), independently proposed by Li et al. (2020) and Kidger et al. (2021) which aim to learn the drift and diffusion coefficients for an arbitrary SDE. In this work we will focus specifically on the case of additive-noise SDEs with known diffusion coefficients, given by

$$\mathrm{d}\boldsymbol{X}_t = \boldsymbol{f}^\theta(t, \boldsymbol{X}_t)\,\mathrm{d}t + g(t)\,\mathrm{d}\boldsymbol{W}_t, \tag{1}$$

where $\boldsymbol{f}^\theta$ is the drift coefficient we aim to learn. Previous work has focused on learning these coefficients by integrating the SDE numerically and then performing backpropagation through the

numerical solution. However, for many SDEs studied in generative modelling we can learn the drift coefficient via *simulation-free* training which has underpinned the popularity of different *matching* objectives such as score matching and flow matching. In this work, we focus on diffusion SDEs; however, our results hold for any additive-noise SDE.

**Flow matching and diffusion.** In this section we introduce the necessary background on flow/diffusion models (Peluchetti 2021; Lipman, Chen, et al. 2023; Liu, Gong, et al. 2023; Tong et al. 2024; Albergo et al. 2025). We denote data samples as $\boldsymbol{X}_1 \in \mathbb{R}^d$ drawn from the data distribution with density $p_1 \equiv q(\boldsymbol{x})$. We take the source distribution to be a unit Gaussian with density $p_0 \equiv p(\boldsymbol{x})$ with $\boldsymbol{X}_0 \sim p_0$. Following Lipman, Havasi, et al. (2024) we consider the scenario of *affine Gaussian probability paths* where we define a random variable $\boldsymbol{X}_t \coloneqq \alpha_t \boldsymbol{X}_1 + \sigma_t \boldsymbol{X}_0$ with noise schedule $(\alpha_t, \sigma_t)$, where $\alpha_t, \sigma_t \geq 0$ with $\alpha_0 = \sigma_1 = 0$, $\alpha_1 = \sigma_0 = 1$, and $\alpha_t$ ($\sigma_t$ resp.) is strictly monotonically increasing (decreasing resp.) and continuously differentiable.

In the flow matching framework we learn the marginal vector field as

$$\boldsymbol{u}_t(\boldsymbol{X}) = \int_{\mathbb{R}^d} \boldsymbol{u}_{t|1}(\boldsymbol{X}|\boldsymbol{X}_1) p_{1|t}(\boldsymbol{X}_1|\boldsymbol{X}) \, \mathrm{d}\boldsymbol{X}_1, \qquad p_{1|t}(\boldsymbol{X}_1|\boldsymbol{X}) = \frac{p_{t|1}(\boldsymbol{X}|\boldsymbol{X}_1) p_1(\boldsymbol{X}_1)}{p_t(\boldsymbol{X})}, \tag{2}$$

where $\boldsymbol{u}_{t|1}(\cdot|\boldsymbol{X}_1)$ is the vector field conditioned on a data sample $\boldsymbol{X}_1$. This vector field can be shown to satisfy,

$$\boldsymbol{X}_0 \sim p_0, \quad \boldsymbol{X}_0 + \int_0^t \boldsymbol{u}_s(\boldsymbol{X}_s) \, \mathrm{d}s = \boldsymbol{X}_t \sim p_t, \tag{3}$$

such that the solution to (3) at time $t = 1$ yields samples from the data distribution $q \equiv p_1$.

The ODE in Equation (3) is referred to as the *probability flow ODE* (Song, Sohl-Dickstein, et al. 2021) and can be written as an SDE with the same marginal distributions, given by

$$\mathrm{d}\boldsymbol{X}_t = \left[ 2\boldsymbol{u}_t(\boldsymbol{X}_t) - \frac{\dot{\alpha}_t}{\alpha_t} \boldsymbol{X}_t \right] \mathrm{d}t + \nu_t \, \mathrm{d}\boldsymbol{W}_t, \quad \nu_t^2 = 2\frac{\dot{\alpha}_t}{\alpha_t}\sigma_t^2 - 2\sigma_t \dot{\sigma}_t, \tag{4}$$

where $\boldsymbol{W}_t$ is the standard Brownian motion. This derivation follows straightforwardly from Anderson (1982) and has been discussed more recently in the context of diffusion models (Maoutsa et al. 2020; Song, Sohl-Dickstein, et al. 2021; Holderrieth, Singer, et al. 2026). The implications for model performance when sampling from either the SDE or ODE formulation with equivalent marginals is discussed in (Nie et al. 2024; Albergo et al. 2025).

**Flow maps.** Since integrating the generative models in (3) or (4) requires many function evaluations, recent work has proposed to instead learn the integral, or solution map, directly at training time. These works include neural flows (Biloš et al. 2021), consistency models (Song, Dhariwal, et al. 2023; Heek et al. 2024; Kim et al. 2024), shortcut models (Frans et al. 2025), mean flows (Geng, Deng, et al. 2025), and flow maps (Boffi et al. 2024, 2025; Sabour et al. 2025). There exists a variety of techniques for training such models. However, training can largely be broken down into two loss components: one term learns the instantaneous behavior at $s = t$ and another learns the flow map $t > s$. Training via such an objective is referred to as self-distillation (Boffi et al. 2025).

Table 1: Comparison of flow-based models.

| Method | Few-step | Pathwise | Stochastic |
|---|---|---|---|
| Flow models | ✗ | ✓ | ✗ |
| Diffusion models | ✗ | ✓ | ✓ |
| GLASS flows | ✗ | ✗ | ✓ |
| Flow maps | ✓ | ✓ | ✗ |
| Weak SFMs | ✓ | ✗ | ✓ |
| **SSFM (ours)** | ✓ | ✓ | ✓ |

**Weak stochastic flow maps.** Recently, several works have explored a stochastic extension of deterministic flow maps (Kiyohara et al. 2025; Holderrieth, Chen, et al. 2026; Passaro et al. 2026; Potaptchik et al. 2026), which can be characterized as learning the transition kernel $p_{t|s}(\boldsymbol{X}_t|\boldsymbol{X}_s)$ of some underlying SDE. Specifically, most of these methods (Holderrieth, Chen, et al. 2026; Passaro et al. 2026; Potaptchik et al. 2026) proceed by defining a deterministic ODE flow map where each step of the map is defined by an inner flow model,[1] given by

$$\bar{\boldsymbol{X}}_0 \sim p_0, \quad \bar{\boldsymbol{X}}_1 = \bar{\boldsymbol{X}}_0 + \int_0^1 \bar{\boldsymbol{u}}_\tau(\bar{\boldsymbol{X}}_\tau|\boldsymbol{X}_s, s) \, \mathrm{d}\tau, \quad \bar{\boldsymbol{X}}_1 \sim p_{t|s}(\cdot|\boldsymbol{X}_s). \tag{5}$$

[1] In contrast, Kiyohara et al. (2025) propose a variational objective.

Since these methods learn the transition kernel $p_{t|s}$ they are described as exhibiting *weak* (in distribution) convergence (Øksendal 2003). This is in contrast to the pathwise solution map of the underlying additive-noise SDE which generates $\boldsymbol{X}_t$ given an initial condition $\boldsymbol{X}_s$ and a realization of the Brownian motion $\{\boldsymbol{W}_u\}_{s\leq u\leq t}$. We summarize the relationship between SSFMs and prior methods in Table 1.

## 3 Strong Stochastic Flow Maps

In this section we describe the transition from deterministic flow maps to strong stochastic flow maps. We present a natural extension of deterministic flow maps by learning the solution map from both an initial condition and a realization of the Brownian path. We then show how this solution map can be obtained by minimization of an appropriate self-distillation objective.

### 3.1 The Itô map

Consider the following additive-noise Itô SDE,

$$\mathrm{d}\boldsymbol{X}_t = \boldsymbol{f}(t, \boldsymbol{X}_t)\,\mathrm{d}t + \boldsymbol{g}(t)\,\mathrm{d}\boldsymbol{W}_t, \tag{6}$$

where $\boldsymbol{X}_t \in \mathbb{R}^d$ is a continuous-valued stochastic process with initial condition $\boldsymbol{X}_0$, $\boldsymbol{f} : \mathbb{R}\times\mathbb{R}^d \to \mathbb{R}^d$ is the drift function, $\boldsymbol{g} : \mathbb{R} \to \mathbb{R}^{d\times w}$ is the diffusion coefficient and $\boldsymbol{W}_t \in \mathbb{R}^w$ is Brownian motion. It is assumed that $\boldsymbol{f}$ and $\boldsymbol{g}$ are suitably regular such that a unique strong solution $\boldsymbol{X}_t$ exists (see Assumption B.1).

It is natural to consider the solution map of this SDE, $\boldsymbol{\Psi}_{s,t} : (\boldsymbol{X}_s, \boldsymbol{W}_{[s,t]}) \mapsto \boldsymbol{X}_t$, where the Brownian path is written as $\boldsymbol{W}_{[s,t]} = \{\boldsymbol{W}_u\}_{s\leq u\leq t}$. This map $\boldsymbol{\Psi}_{s,t}$ is known as the Itô map.[2] For an SDE with additive noise and suitably regular coefficients, the Itô map is well-posed and continuous (Friz and Victoir 2010). It is this map that we aim to approximate with a neural network, $\boldsymbol{\Psi}^{\theta}_{s,t}$.

Since the Itô map is defined with respect to both an initial condition, $\boldsymbol{X}_0$, and a realization of $\boldsymbol{W}_{[s,t]}$, the convergence to the solution $\boldsymbol{X}_t$ is pathwise. In stochastic analysis, a process approximating such a pathwise solution is called *strongly* convergent (Øksendal 2003). We therefore refer to our method as *Strong Stochastic Flow Maps*.

#### 3.1.1 Constructing the Itô map

Analogously to the deterministic flow map, we can derive a tangent condition that the Itô map satisfies. This result will allow us to associate the vector fields of an SDE with the Itô map.

**Lemma 3.1** (Tangent Condition)**.** *Let $\boldsymbol{\Psi}_{s,t}(\boldsymbol{X}_s, \boldsymbol{W}_{[s,t]})$ denote the Itô map for (6). Then,*

$$\lim_{s\to t} \mathrm{d}\boldsymbol{\Psi}_{s,t}(\boldsymbol{X}_s, \boldsymbol{W}_{[s,t]}) = \boldsymbol{f}(t, \boldsymbol{X}_t)\,\mathrm{d}t + \boldsymbol{g}(t)\,\mathrm{d}\boldsymbol{W}_t. \tag{7}$$

To construct the Itô map we therefore propose an Euler-Maruyama step-like object that satisfies the tangent condition. We will show that such a parameterization attains the Itô map when trained according to the objective in (14).

**Proposition 3.2** (Strong Stochastic Flow Map)**.** *Consider an Euler-Maruyama parameterization of the Itô map,*

$$\boldsymbol{\Psi}_{s,t}(\boldsymbol{X}_s, \boldsymbol{W}_{[s,t]}) = \boldsymbol{X}_s + \boldsymbol{f}_{s,t}(\boldsymbol{X}_s, \boldsymbol{W}_{[s,t]})(t-s) + \boldsymbol{g}_{s,t}(\boldsymbol{W}_{[s,t]})(\boldsymbol{W}_t - \boldsymbol{W}_s), \tag{8}$$

*where $\boldsymbol{f}, \boldsymbol{g}$ are twice continuously differentiable, Lipschitz in both time arguments, satisfy the conditions that $\boldsymbol{f}_{t,t}(\boldsymbol{X}_t, \boldsymbol{W}_{[s,t]}) = \boldsymbol{f}_{t,t}(\boldsymbol{X}_t)$, and $\boldsymbol{g}_{t,t}(\boldsymbol{W}_{[s,t]}) = \boldsymbol{g}_{t,t}$ (i.e. the coefficients are independent of $\boldsymbol{W}_{[s,t]}$ for $s = t$). Then, $\boldsymbol{\Psi}_{s,t}$ satisfies the tangent condition (7) if and only if*

$$\boldsymbol{f}_{t,t}(x_t) = \boldsymbol{f}(t, x_t), \quad \boldsymbol{g}_{t,t} = \boldsymbol{g}(t).$$

[2] Note that for a general SDE with state-dependent diffusion, $\boldsymbol{g}(t, \boldsymbol{X}_t)$, the Itô map is not a well-defined continuous function. This is solved by rough path theory and the Itô-Lyons map (Lyons 1998).

The proposed stochastic flow map in Proposition 3.2 introduces a drift $\boldsymbol{f}_{s,t}$ and a diffusion $\boldsymbol{g}_{s,t}$. The intuition behind these functions is that they act as the normalized drift and diffusion integrals, respectively. Therefore, they must depend on the underlying driving path $\boldsymbol{W}_{[s,t]}$.

We see from Proposition 3.2 that for the stochastic flow map to satisfy the tangent condition, the drift integral must collapse to the drift function, $\boldsymbol{f}_{t,t}(x_t) = \boldsymbol{f}(t, x_t)$, and the diffusion integral must collapse to the diffusion coefficient, $\boldsymbol{g}_{t,t} = \boldsymbol{g}(t)$, as $s \to t$. This indicates that $\boldsymbol{f}_{t,t}(x_t)$ and $\boldsymbol{g}_{t,t}$ can be estimated by the matching objective in (14).

Next, we must establish an objective that constrains the finite-time, $t > s$, behaviour of the stochastic flow map. Consider the semigroup condition,

$$\boldsymbol{\Psi}_{s,t}(\boldsymbol{X}_s, \boldsymbol{W}_{[s,t]}) = \boldsymbol{\Psi}_{u,t}(\boldsymbol{\Psi}_{s,u}(\boldsymbol{X}_s, \boldsymbol{W}_{[s,u]}), \boldsymbol{W}_{[u,t]}), \tag{9}$$

for $s < u < t$. Note this condition is satisfied by the Itô map (see the proof of Proposition 3.3). It is possible to show that if the strong stochastic flow map construction from Proposition 3.2 satisfies the tangent condition (7) and the semigroup property, then this map *is* the Itô map. This is given by Proposition 3.3.

**Proposition 3.3** (Semigroup condition)**.** *Let $\boldsymbol{\Psi}_{s,t}(\boldsymbol{X}_s, \boldsymbol{W}_{[s,t]})$ denote the strong stochastic flow map satisfying (7) and (8). Then $\boldsymbol{\Psi}_{s,t}(\boldsymbol{X}_s, \boldsymbol{W}_{[s,t]})$ is the Itô map if and only if the semigroup property (9) holds.*

### 3.1.2 Constructing the Brownian motion

Since the Itô map requires a realization of the Brownian path $\boldsymbol{W}_{[s,t]}$, so too do the drift and diffusion integral functions $\boldsymbol{f}_{s,t}, \boldsymbol{g}_{s,t}$. These functions will ultimately be parameterized as neural networks. We therefore require an efficient characterization of $\boldsymbol{W}_{[s,t]}$ that we can pass as input.

A naïve approach is to simply pass a piecewise linear approximation to $\boldsymbol{W}_{[s,t]}$ over $N$ intervals. However, to obtain any reasonable approximation to the path, too many intervals would be required. We instead look to pass the coefficients of a polynomial expansion of $\boldsymbol{W}_{[s,t]}$. Specifically, we consider a piecewise polynomial expansion to Brownian motion in terms of shifted Legendre polynomials (Foster et al. 2020; Habermann 2021).

This polynomial expansion has a number of desirable properties: (1) the coefficients of this expansion appear in the stochastic Taylor expansion of the SDE in (6) enabling larger time-step approximation; (2) the coefficients are independently and normally distributed allowing for tractable and exact sampling; (3) the coefficients admit closed-form Chen relations (Chen 1954, 1957) such that two coefficients over the sub-intervals $[s, u]$, $[u, t]$ combine into a single coefficient over $[s, t]$. This property enables the semigroup objective to be implemented. These results are shown in Theorem 3.4.

The following results are theoretically dense, relying on elements of rough path theory (Lyons 1998). However, the polynomial expansion of Brownian motion is visually intuitive; see Figure 3.

**Definition 3.1** (Polynomial approximation for Brownian motion)**.** Let $\widetilde{P}_n : [0, 1] \to \mathbb{R}$ denote the $n$-th shifted Legendre polynomial on $[0, 1]$. We define the $n$-th coefficient of the polynomial expansion as,

$$\boldsymbol{I}_{s,t}^{(n)} = \int_s^t \widetilde{P}_n\left(\frac{u-s}{t-s}\right) \mathrm{d}\boldsymbol{W}_u. \tag{10}$$

As introduced in (Foster et al. 2020; Habermann 2021), the degree-$N$ polynomial approximation of the Brownian motion on $[s, t]$ takes the form,

$$\boldsymbol{W}_{u,v}^{(N)} = \sum_{n=0}^{N-1} \frac{2n+1}{t-s} \boldsymbol{I}_{s,t}^{(n)} \int_u^v \widetilde{P}_n\left(\frac{r-s}{t-s}\right) \mathrm{d}r, \tag{11}$$

for each increment $[u, v] \subseteq [s, t]$.

**Theorem 3.4** (Properties of $\boldsymbol{W}_{u,v}^{(N)}$)**.** *The polynomial approximation of the Brownian motion in Definition 3.1 has the following properties:*

1. *Converges to Brownian motion in the $\alpha$-Hölder distance, with $\alpha \in [0, \frac{1}{2})$,*
2. *The coefficients $\boldsymbol{I}_{s,t}^{(n)}$ are independently and normally distributed with*

$$\boldsymbol{I}_{s,t}^{(n)} \sim \mathcal{N}\left(\boldsymbol{0}, \frac{(t-s)}{2n+1}\boldsymbol{I}\right), \tag{12}$$

3. *The coefficients $\boldsymbol{I}_{s,t}^{(n)}$ admit closed form Chen relations.*

### 3.2 Training

Given the polynomial approximation in Definition 3.1, we can write the Strong Stochastic Flow Map as

$$\boldsymbol{\Psi}_{s,t}^{\theta}(\boldsymbol{X}_s, \boldsymbol{I}_{s,t}^{(N)}) = \boldsymbol{X}_s + \boldsymbol{f}_{s,t}^{\theta}(\boldsymbol{X}_s, \boldsymbol{I}_{s,t}^{(N)})(t-s) + \boldsymbol{g}_{s,t}^{\theta}(\boldsymbol{I}_{s,t}^{(N)})(\boldsymbol{W}_t - \boldsymbol{W}_s), \tag{13}$$

where the coefficients up to degree $N$, given by $\boldsymbol{I}_{s,t}^{(N)} = \{\boldsymbol{I}_{s,t}^{(n)}\}_{n \leq N}$, are passed to the neural network terms $\boldsymbol{f}_{s,t}^{\theta}, \boldsymbol{g}_{s,t}^{\theta}$. Note that $\boldsymbol{W}_t - \boldsymbol{W}_s = \boldsymbol{I}_{s,t}^{(0)}$.

#### 3.2.1 Self-distillation objective

We have seen that the stochastic flow map in (8) *is* the Itô map if and only if the diagonal integrals are equal to the SDE coefficients, $\boldsymbol{f}_{t,t}(x_t) = \boldsymbol{f}(t, x_t), \boldsymbol{g}_{t,t} = \boldsymbol{g}(t)$, and the semigroup condition is satisfied.

These two properties allow us to write down an objective for the Itô map. This is shown by Theorem 3.5.

**Theorem 3.5** (Self-distillation Objective)**.** *Let $\boldsymbol{\Psi}_{s,t}(\boldsymbol{X}_s, \boldsymbol{W}_{[s,t]})$ denote the Itô map for (6). Then, this map is given by the strong stochastic flow map in (8) where $\boldsymbol{\Psi}$ is the unique global minimizer over $\hat{\boldsymbol{\Psi}}$ of*

$$\mathcal{L}_{SD}(\hat{\boldsymbol{\Psi}}) = \mathcal{L}_{\boldsymbol{f},\boldsymbol{g}}(\hat{\boldsymbol{\Psi}}) + \mathcal{L}_D(\hat{\boldsymbol{\Psi}}), \tag{14}$$

*where*

$$\mathcal{L}_{\boldsymbol{f},\boldsymbol{g}}(\hat{\boldsymbol{\Psi}}) = \mathbb{E}_{t,\boldsymbol{X}_t}\left[\|\boldsymbol{f}(t, \boldsymbol{X}_t) - \hat{\boldsymbol{f}}_{t,t}(\boldsymbol{X}_t)\|_2^2 + \|\boldsymbol{g}(t) - \hat{\boldsymbol{g}}_{t,t}\|_2^2\right], \tag{15}$$

$$\mathcal{L}_D(\hat{\boldsymbol{\Psi}}) = \mathbb{E}_{s,u,t,\boldsymbol{X}_s,\boldsymbol{W}_{[s,t]}}\left[\|\hat{\boldsymbol{\Psi}}_{s,t}(\boldsymbol{X}_s, \boldsymbol{W}_{[s,t]}) - \hat{\boldsymbol{\Psi}}_{u,t}(\hat{\boldsymbol{\Psi}}_{s,u}(\boldsymbol{X}_s, \boldsymbol{W}_{[s,u]}), \boldsymbol{W}_{[u,t]})\|_2^2\right]. \tag{16}$$

The objective in (14) is written in terms of the drift and diffusion coefficients of the SDE for which we wish to learn the Itô map. In the case of flow models (flow matching, diffusion, stochastic interpolants) the drift coefficient is known conditioned on an end-point sample, $\boldsymbol{X}_1$ (and/or $\boldsymbol{X}_0$), obtained from the marginal at $t = 1$ (and/or $t = 0$).

The results presented in this section are proved in Appendix B.

#### 3.2.2 General training method

By Theorems 3.5 and 3.4, this parameterization trained according to (14) converges to the Itô map. This objective characterizes the Itô map under exact expectations with global minimum realizable over the class of functions $\hat{\boldsymbol{\Psi}}$ satisfying the regularity assumptions. In practice, we parameterize $\boldsymbol{\Psi}$ as a neural network with a finite number of Brownian input coefficients and take expectations over mini-batches. The training algorithm therefore learns an approximation to the Itô map.

The general training algorithm is given in Algorithm 1. This requires a ground truth for $\boldsymbol{f}$ and $\boldsymbol{g}$ which can be constructed based on the task; for diffusion this is obtained by the reverse SDE with $\boldsymbol{f}(t, \boldsymbol{X}_t)$ derived from the conditional score – resulting in a simulation-free objective (see Appendix C.2).

**Algorithm 1** Strong Stochastic Flow Map Training

**Require:** Batch size $M$, split $\eta \in (0,1)$, polynomial degree $N$, threshold $\Delta t$, EMA decay $\beta \in (0,1)$

**repeat**
  Sample $s \sim \mathcal{U}[0,1]$, $\boldsymbol{X}_s \sim p_s$
  ▷ Matching objective (batch size $\lfloor \eta M \rfloor$)
  Sample $t \sim \mathcal{U}[s,\ s{+}\Delta t]$ and $\boldsymbol{I}_{s,t}^{(N)} \sim$ (12)
  $\hat{\boldsymbol{X}}_t \leftarrow \boldsymbol{X}_s + \boldsymbol{f}(s,\boldsymbol{X}_s)(t{-}s) + \boldsymbol{g}(s)(\boldsymbol{W}_t{-}\boldsymbol{W}_s)$
  $\mathcal{L}_{\boldsymbol{f},\boldsymbol{g}} \leftarrow (t{-}s)^{-1} \left\| \hat{\boldsymbol{X}}_t - \boldsymbol{\Psi}_{s,t}^{\theta}(\boldsymbol{X}_s, \boldsymbol{I}_{s,t}^{(N)}) \right\|^2$
  ▷ Distillation objective (batch size $M - \lfloor \eta M \rfloor$)
  Sample $t \sim \mathcal{U}[s{+}\Delta t,\ 1]$ and set $u \leftarrow \frac{1}{2}(s{+}t)$
  Sample $\boldsymbol{I}_{s,u}^{(N)}, \boldsymbol{I}_{u,t}^{(N)} \sim$ (12) and compute $\boldsymbol{I}_{s,t}^{(N)}$ via (22)
  $\boldsymbol{X}_{\text{pred}} \leftarrow \boldsymbol{\Psi}_{s,t}^{\theta}(\boldsymbol{X}_s, \boldsymbol{I}_{s,t}^{(N)})$
  $\boldsymbol{X}_{\text{tgt}} \leftarrow \texttt{stopgrad}\big(\boldsymbol{\Psi}_{u,t}^{\theta}\big(\boldsymbol{\Psi}_{s,u}^{\theta}(\boldsymbol{X}_s, \boldsymbol{I}_{s,u}^{(N)}),\ \boldsymbol{I}_{u,t}^{(N)}\big)\big)$
  $\mathcal{L}_{\text{D}} \leftarrow (t{-}s)^{-1} \left\| \boldsymbol{X}_{\text{tgt}} - \boldsymbol{X}_{\text{pred}} \right\|^2$
  ▷ Update
  $\theta \leftarrow \theta - \lambda \nabla_\theta (\mathcal{L}_{\boldsymbol{f},\boldsymbol{g}} + \mathcal{L}_{\text{D}})$
  $\hat{\theta} \leftarrow \beta\hat{\theta} + (1{-}\beta)\,\theta$
**until** $\theta$ converges

Note that $\mathcal{L}_{\boldsymbol{f},\boldsymbol{g}}$ provides a training signal for $s = t$ (*i.e.* $\boldsymbol{f}_{t,t}^{\theta}, \boldsymbol{g}_{t,t}^{\theta}$), but $\mathcal{L}_{\text{D}}$ evaluates the model at $t > s$. This forces the network to generalize from $s = t$ to $t > s$ via continuity. To alleviate this, we instead match a small Euler-Maruyama step (with $t > s$) of the ground truth SDE rather than matching coefficients. This can be shown to result in a weighted coefficient matching objective (see Lemma C.1).

## 4 Experiments

We consider three experiments that demonstrate the properties and performance of the SSFM model. Firstly, we ablate the algorithmic properties on a non-linear SDE and verify the effectiveness of the polynomial approximation to Brownian motion. Second, we apply the model to CIFAR-10 and CelebA-64 image generation and show that SSFMs outperform previous deterministic and stochastic flow maps. Third, we consider generation of equilibrium molecular conformations on Alanine-Dipeptide and Chignolin datasets, where the SSFM model is capable of generating accurate samples in as few as two network evaluations.

### 4.1 Non-linear SDE

We first investigate the performance of the SSFM model on a toy system: a non-linear drift, additive-noise SDE, given by

$$\mathrm{d}\boldsymbol{X}_t = [\boldsymbol{X}_t - \boldsymbol{X}_t^3]\,\mathrm{d}t + \sqrt{\beta_t}\,\mathrm{d}\boldsymbol{W}_t, \tag{17}$$

where $\beta_t$ is a linear interpolation between $\beta_{\text{min}} = 0.1$ and $\beta_{\text{max}} = 20$. This system is intended to mimic the variance preserving reverse diffusion SDE. The SSFM is learned via Algorithm 1 with the ground truth constructed from (17).

In Figure 2, we ablate the SSFM accuracy as a function of the polynomial degree. As the number of coefficients passed to the model increases, the strong convergence error decreases; this is most pronounced at larger step sizes. It can also been seen that the accuracy gained by each additional coefficient added diminishes, but is far from saturated at $N = 4$.

In Figure 3, we show the underlying 4-th degree polynomial expansion of Brownian motion and the learned flow map evaluated at 16-steps. We see that the SSFM is capable of accurately coarsening the SDE dynamics. We include additional plots for this system in Appendix E.1.

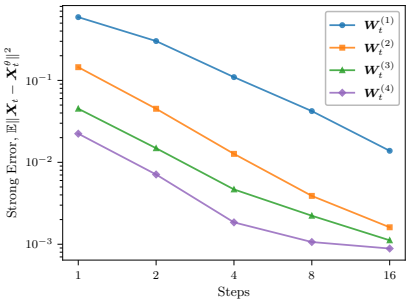


Figure 2: Strong error of the SSFM as a function of the polynomial degree.

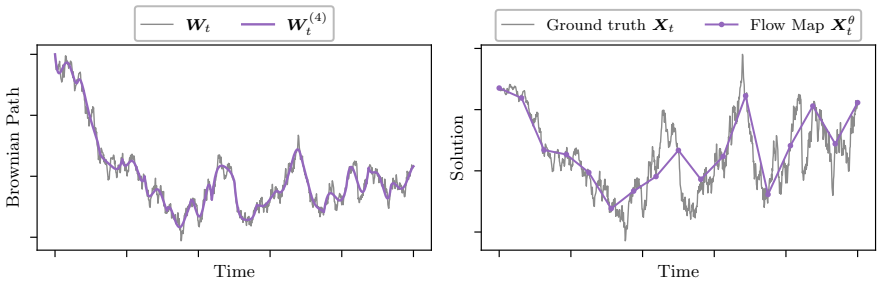


Figure 3: *Left*: Ground-truth Brownian path $\boldsymbol{W}_t$ and 4-th degree polynomial approximation $\boldsymbol{W}_t^{(4)}$ over 16 intervals. *Right*: Ground-truth SDE solution driven by $\boldsymbol{W}_t$ and learned 16-step flow map driven by $\boldsymbol{W}_t^{(4)}$.

Table 2: FID (↓) on CIFAR-10 and CelebA-64 across different NFE values. We provide deterministic flow maps as a reference, but compare to other stochastic methods.

| Dataset | Method | NFE 2 | NFE 4 | NFE 8 | NFE 16 |
|---|---|---|---|---|---|
| CIFAR-10 | *Deterministic flows* | | | | |
| | Consistency training (Song, Dhariwal, et al. 2023) | 5.83 | — | — | — |
| | Flow map (LSD) (Boffi et al. 2025) | 4.37 | 3.34 | 3.33 | 3.57 |
| | Flow map (PSD-M) (Boffi et al. 2025) | 8.43 | 5.96 | 5.07 | 4.64 |
| | Flow map (PSD-U) (Boffi et al. 2025) | 7.95 | 6.03 | 5.32 | 5.16 |
| | Flow map (Holderrieth, Chen, et al. 2026) | 4.60 | 4.18 | 4.88 | — |
| | *Weak stochastic flows* | | | | |
| | GLASS (Holderrieth, Singer, et al. 2026) | 157.55 | 39.47 | 11.60 | — |
| | Diamond Map (Holderrieth, Chen, et al. 2026) | 5.80 | 5.80 | 6.73 | — |
| | *Strong stochastic flows* | | | | |
| | SSFM (Ours) | **4.93**$_{\pm 0.43}$ | **3.49**$_{\pm 0.15}$ | **3.29**$_{\pm 0.16}$ | **3.35**$_{\pm 0.24}$ |
| CelebA-64 | *Deterministic flows* | | | | |
| | Flow map (LSD) (Boffi et al. 2025) | 5.74 | 3.18 | 2.18 | 1.96 |
| | Flow map (PSD-M) (Boffi et al. 2025) | 11.75 | 7.89 | 6.06 | 5.09 |
| | Flow map (PSD-U) (Boffi et al. 2025) | 11.02 | 7.47 | 6.00 | 5.63 |
| | Flow map (Holderrieth, Chen, et al. 2026) | 4.05 | 3.08 | 3.15 | — |
| | *Weak stochastic flows* | | | | |
| | GLASS (Holderrieth, Singer, et al. 2026) | 95.25 | 51.80 | 26.33 | — |
| | Diamond Map (Holderrieth, Chen, et al. 2026) | 9.16 | 6.74 | 5.78 | — |
| | *Strong stochastic flows* | | | | |
| | SSFM (Ours) | **5.65** | **3.89** | **3.60** | **3.79** |

### 4.2 Image generation

We demonstrate the strong stochastic flow map construction for image generation on the CIFAR-10 and CelebA-64 datasets (Krizhevsky, Hinton, et al. 2009; Liu, Luo, et al. 2015). The training procedure follows Algorithm 1, where the ground truth SDE is obtained via the reverse time variance preserving diffusion SDE (Song, Sohl-Dickstein, et al. 2021). See Appendix C.2 for a description of this SDE. The $\boldsymbol{f}^\theta, \boldsymbol{g}^\theta$ networks are independently parameterized with the EDM2 architecture (Karras et al. 2024). The number of polynomial coefficients was chosen to be $N = 3$.

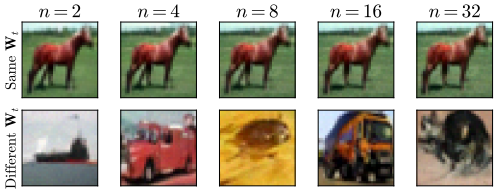


Figure 4: *Top*: SSFM with fixed $\boldsymbol{X}_0$ and same $\boldsymbol{W}_t$ across step counts. *Bottom*: SSFM with fixed $\boldsymbol{X}_0$, different $\boldsymbol{W}_t$ across step counts.

Table 3: PMF squared error (↓) and JS divergence (↓) on Alanine-Dipeptide across step counts.

| | | NFE | | | | |
|---|---|---|---|---|---|---|
| | | 2 | 4 | 10 | 20 | 100 |
| PMF Error | Two-for-One | 16.68 | 15.71 | 11.41 | 3.56 | 0.087 |
| | Diffusion | 16.73 | 15.67 | 11.35 | 3.40 | 0.084 |
| | Mixture | 22.07 | 17.27 | 11.56 | 3.39 | 0.078 |
| | Fokker-Planck | 15.48 | 15.71 | 11.35 | 3.50 | 0.092 |
| | Both | 21.83 | 17.09 | 11.57 | 3.39 | 0.087 |
| | **SSFM** | $\mathbf{0.25}_{\pm 0.01}$ | $\mathbf{0.17}_{\pm 0.01}$ | $\mathbf{0.11}_{\pm 0.00}$ | $\mathbf{0.09}_{\pm 0.01}$ | $\mathbf{0.072}_{\pm 0.005}$ |
| JS ($\times 10^{-2}$) | Two-for-One | 48.19 | 48.57 | 38.48 | 16.78 | 0.813 |
| | Diffusion | 48.74 | 48.38 | 38.31 | 16.29 | 0.787 |
| | Mixture | 52.43 | 49.50 | 38.71 | 16.28 | 0.770 |
| | Fokker-Planck | 47.16 | 48.25 | 38.31 | 16.57 | 0.836 |
| | Both | 52.34 | 49.27 | 38.66 | 16.37 | 0.830 |
| | **SSFM** | $\mathbf{1.92}_{\pm 0.02}$ | $\mathbf{1.32}_{\pm 0.03}$ | $\mathbf{0.89}_{\pm 0.06}$ | $\mathbf{0.76}_{\pm 0.07}$ | $\mathbf{0.610}_{\pm 0.030}$ |

The results can be seen in Table 2. The strong stochastic flow map models outperform the weak formulations across all step sizes considered. Additionally, under the strong construction we find that stochastic flow maps become competitive with deterministic flow maps and often outperform. We verify the strong convergence property of the SSFM in Figure 4. Here, we see that for a fixed initial condition $\boldsymbol{X}_0$ and the same Brownian path $\boldsymbol{W}_t$, the resulting sample is the same for all step counts. Further, the stochasticity of the flow map is retained; for the same $\boldsymbol{X}_0$ with different $\boldsymbol{W}_t$, we obtain different generated samples.

### 4.3 Molecular systems

We demonstrate the strong stochastic flow map on the molecular system Alanine Dipeptide (ALDP) (Plainer et al. 2025) and all-atom Chignolin. We compare SSFM to regular diffusion baselines established in (Plainer et al. 2025) for ALDP. For all-atom Chignolin, we compare SSFM to a *de novo* diffusion model. The results can be seen in Tables 3 and 4. See Appendix E for full experimental details and additional results including pooled Ramachandran and tICA plots for Chignolin.

For ALDP, as expected, the SSFM enables markedly more accurate sampling than the diffusion baselines at small step counts. As steps $\to$ 100, the methods converge to achieving comparable performance. Notably, the SSFM at 20 steps is competitive with the diffusion baselines at 100 steps, a $5\times$ reduction in NFEs.

For Chignolin, the SSFM achieves lower error relative to the diffusion baseline across small step counts for the backbone torus and tICA projection. As expected, the methods converge to achieve comparable performance in the many-step limit. Notably, the SSFM is able to sample accurate conformations in the 1-4 NFE regime.

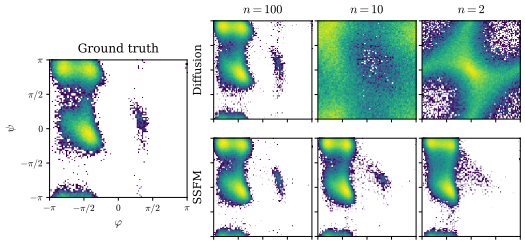


Figure 5: Ramachandran plots for Alanine-Dipeptide showing ground truth data, diffusion mixture baseline and SSFM prediction across step counts.

Table 4: Chignolin Wasserstein metrics (↓) across step counts.

| | | NFE | | | | | |
|---|---|---|---|---|---|---|---|
| Metric | Method | 1 | 2 | 4 | 8 | 16 | 32 |
| $\mathbb{T}$-$W_2$ | Diffusion | $4.69_{\pm 0.02}$ | $5.88_{\pm 0.01}$ | $5.58_{\pm 0.01}$ | $2.71_{\pm 0.01}$ | $1.78_{\pm 0.01}$ | $\mathbf{1.68}_{\pm 0.03}$ |
| | **SSFM** | $\mathbf{2.41}_{\pm 0.03}$ | $\mathbf{2.03}_{\pm 0.03}$ | $\mathbf{1.79}_{\pm 0.02}$ | $\mathbf{1.74}_{\pm 0.03}$ | $\mathbf{1.71}_{\pm 0.02}$ | $1.71_{\pm 0.01}$ |
| tICA-$W_2$ | Diffusion | $1.99_{\pm 0.03}$ | $19.29_{\pm 0.14}$ | $3.22_{\pm 0.03}$ | $0.75_{\pm 0.01}$ | $\mathbf{0.31}_{\pm 0.01}$ | $\mathbf{0.26}_{\pm 0.02}$ |
| | **SSFM** | $\mathbf{0.80}_{\pm 0.01}$ | $\mathbf{0.51}_{\pm 0.01}$ | $\mathbf{0.41}_{\pm 0.03}$ | $\mathbf{0.38}_{\pm 0.03}$ | $0.36_{\pm 0.03}$ | $0.33_{\pm 0.03}$ |

Table 5: Committer times for Alanine Dipeptide backbone states.

| Method | $\tau(G)$ | $\tau(A)$ | $\tau(B)$ | evals/path | s/path |
|---|---|---|---|---|---|
| Weak SFM (rollout) | 0.159 | 0.217 | 0.234 | 297 | 31.45 |
| SSFM (binary search) | 0.104 | 0.162 | 0.230 | **19** | **2.18** |

**Path-dependent committer times.** It is possible to efficiently calculate the committer times (along diffusion time) for the molecular systems using the SSFM model. The committer time is defined as the minimum time $\tau$ when the probability of generating a specific state $b_i$ exceeds a threshold, denoted $p(\boldsymbol{X}_1 \in b_i | \boldsymbol{X}_t) > \gamma$. We investigate the committer times for the Alanine Dipeptide backbone states $b_i \in \{A, B, G\}$ where each state is defined by backbone angles. Computing this time $\tau$ with the SSFM can be achieved efficiently via a binary search algorithm which we detail in Appendix E.5.

See Table 5 for the resulting $\tau_i$ found when each backbone mode $b_i$ is determined with probability $\gamma = 0.9$. We see that the SSFM requires $15.6\times$ fewer function evaluations, and therefore achieves a $14.4\times$ faster runtime, than the weak stochastic flow map.

Note that the differences in $\tau_i$ result from the committer time not being strictly monotone over $[0, 1]$. We verify that these are valid committer times in Table 6. We see that the correct $> 0.9$ fraction of states within mode $b_i$ are obtained. Note that in the original dataset the fractions are $G = 0.007$, $A = 0.393$, $B = 0.60$. This algorithm therefore provides an efficient procedure for upsampling any desired mode that is undersampled in the dataset.

Table 6: Alanine Dipeptide state sampling.

| Committer / Final mode | $G$ | $A$ | $B$ |
|---|---|---|---|
| $G$ | **0.9475** | 0.0273 | 0.0252 |
| $A$ | 0.0048 | **0.9117** | 0.0836 |
| $B$ | 0.0033 | 0.0773 | **0.9194** |

## 5 Conclusion

In this work, we have introduced a novel theoretical framework for learning the Itô map to any additive-noise SDE. This framework was used to construct a class of flow maps, termed STRONG STOCHASTIC FLOW MAPS (SSFMs), that approximate the Itô map to compute a strong solution to additive-noise SDEs. On image generation experiments, this parameterization was shown to outperform previous stochastic flow map models. When applied to sampling molecular conformations, SSFMs obtained accurate results in as few as 1-2 network evaluations and matched the performance of current diffusion based generative models in the many-step regime. We expect this work to open up further improvements in reward alignment of generative models via pathwise estimators, and to accelerate molecular simulation and generative modeling tasks.

### Limitations & future work

The current framework was demonstrated on additive-noise SDEs; extending this framework to state-dependent SDEs is possible with the theory developed here and could be explored in future work. The diffusion SDEs integrate over a fixed time interval $[0, 1]$; future work may look to apply the method to SDEs in physical time, such as stochastic timeseries models and molecular dynamics simulations. The flow map integration error depends on the polynomial approximation of the Brownian path through the degree which introduces a hyperparameter to be tuned. The diffusion network introduces an overhead in comparison to ODE flow map models; however we believe it is possible to construct a drift-only SSFM model via exact additive-noise diffusion integral sampling.

### Acknowledgments

SM is supported by the EPSRC grant EP/S022945/1. AITHYRA is supported by the Austrian Academy of Sciences and the not-for-profit Boehringer Ingelheim Stiftung. The research was enabled in part by computational resources provided by AITHYRA. JF is supported by the Department of Mathematical Sciences at the University of Bath.

## References


Albergo, Michael, Boffi, Nicholas M, and Vanden-Eijnden, Eric (2025). "Stochastic interpolants: A unifying framework for flows and diffusions". In: *Journal of Machine Learning Research* 26.209, pp. 1–80 (cit. on p. 3).

Anderson, Brian DO (1982). "Reverse-time diffusion equation models". In: *Stochastic Processes and their Applications* 12.3, pp. 313–326 (cit. on p. 3).

Biloš, Marin, Sommer, Johanna, Rangapuram, Syama Sundar, Januschowski, Tim, and Günnemann, Stephan (2021). "Neural Flows: Efficient Alternative to Neural ODEs". In: *Advances in Neural Information Processing Systems*. Ed. by A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (cit. on p. 3).

Blasingame, Zander W. and Liu, Chen (2026). "Rex: A Family of Reversible Exponential (Stochastic) Runge-Kutta Solvers". In: *Forty-third International Conference on Machine Learning* (cit. on p. 24).

Blattmann, Andreas, Rombach, Robin, Ling, Huan, Dockhorn, Tim, Kim, Seung Wook, Fidler, Sanja, and Kreis, Karsten (2023). "Align your latents: High-resolution video synthesis with latent diffusion models". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 22563–22575 (cit. on p. 1).

Boffi, Nicholas M, Albergo, Michael S, and Vanden-Eijnden, Eric (2024). "Flow map matching". In: *arXiv preprint arXiv:2406.07507* (cit. on pp. 1, 3).

Boffi, Nicholas M, Albergo, Michael S, and Vanden-Eijnden, Eric (2025). "How to build a consistency model: Learning flow maps via self-distillation". In: *arXiv preprint arXiv:2505.18825* (cit. on pp. 3, 8, 24, 25).

Chen, Kuo-Tsai (1954). "Iterated integrals and exponential homomorphisms". In: *Proceedings of the London Mathematical Society* 3.1, pp. 502–512 (cit. on p. 5).

Chen, Kuo-Tsai (1957). "Integration of paths, geometric invariants and a generalized Baker-Hausdorff formula". In: *Annals of Mathematics* 65.1, pp. 163–178 (cit. on p. 5).

Chen, Ricky T. Q., Rubanova, Yulia, Bettencourt, Jesse, and Duvenaud, David K (2018). "Neural Ordinary Differential Equations". In: *Advances in Neural Information Processing Systems*. Ed. by S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett. Vol. 31. Curran Associates, Inc. (cit. on pp. 1, 2).

Foster, James, Lyons, Terry, and Oberhauser, Harald (2020). "An optimal polynomial approximation of Brownian motion". In: *SIAM Journal on Numerical Analysis* 58.3, pp. 1393–1421 (cit. on pp. 5, 21).

Frans, Kevin, Hafner, Danijar, Levine, Sergey, and Abbeel, Pieter (2025). "One Step Diffusion via Shortcut Models". In: *The Thirteenth International Conference on Learning Representations* (cit. on pp. 3, 24).

Friz, Peter K and Victoir, Nicolas B (2010). *Multidimensional stochastic processes as rough paths: theory and applications*. Vol. 120. Cambridge University Press (cit. on pp. 4, 20).

Garsia, A. M., Rodemich, E., Rumsey, H., and Rosenblatt, M. (1970). "A Real Variable Lemma and the Continuity of Paths of Some Gaussian Processes". In: *Indiana University Mathematics Journal* 20.6, pp. 565–578. ISSN: 00222518, 19435258 (cit. on p. 20).

Geng, Zhengyang, Deng, Mingyang, Bai, Xingjian, Kolter, J Zico, and He, Kaiming (2025). "Mean Flows for One-step Generative Modeling". In: *The Thirty-ninth Annual Conference on Neural Information Processing Systems* (cit. on pp. 1, 3, 24).

Geng, Zhengyang, Lu, Yiyang, Wu, Zongze, Shechtman, Eli, Kolter, J Zico, and He, Kaiming (2025). "Improved mean flows: On the challenges of fastforward generative models". In: *arXiv preprint arXiv:2512.02012* (cit. on pp. 1, 24, 25).

Gonzalez, Martin, Fernandez, Nelson, Tran, Thuy Vinh Dinh, Gherbi, Elies, Hajri, Hatem, and Masmoudi, Nader (2023). "SEEDS: Exponential SDE Solvers for Fast High-Quality Sampling

from Diffusion Models". In: *Thirty-seventh Conference on Neural Information Processing Systems* (cit. on pp. 1, 24).

Habermann, Karen (2021). "A semicircle law and decorrelation phenomena for iterated Kolmogorov loops". In: *Journal of the London Mathematical Society* 103.2, pp. 558–586. ISSN: 1469-7750. DOI: 10.1112/jlms.12384 (cit. on p. 5).

Heek, Jonathan, Hoogeboom, Emiel, and Salimans, Tim (2024). *Multistep Consistency Models* (cit. on pp. 1, 3).

Ho, Jonathan, Jain, Ajay, and Abbeel, Pieter (2020). "Denoising Diffusion Probabilistic Models". In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin. Vol. 33. Curran Associates, Inc., pp. 6840–6851 (cit. on p. 1).

Holderrieth, Peter, Chen, Douglas, Eyring, Luca, Shah, Ishin, Anantharaman, Giri, He, Yutong, Akata, Zeynep, Jaakkola, Tommi, Boffi, Nicholas Matthew, and Simchowitz, Max (2026). "Diamond Maps: Efficient Reward Alignment via Stochastic Flow Maps". In: *arXiv preprint arXiv:2602.05993* (cit. on pp. 2, 3, 8, 24).

Holderrieth, Peter, Havasi, Marton, Yim, Jason, Shaul, Neta, Gat, Itai, Jaakkola, Tommi, Karrer, Brian, Chen, Ricky T. Q., and Lipman, Yaron (2025). "Generator Matching: Generative modeling with arbitrary Markov processes". In: *The Thirteenth International Conference on Learning Representations* (cit. on p. 24).

Holderrieth, Peter, Singer, Uriel, Jaakkola, Tommi, Chen, Ricky T. Q., Lipman, Yaron, and Karrer, Brian (2026). "GLASS Flows: Efficient Inference for Reward Alignment of Flow and Diffusion Models". In: *The Fourteenth International Conference on Learning Representations* (cit. on pp. 3, 8).

Jelinčič, Andraž, Foster, James, and Kidger, Patrick (2024). "Single-seed generation of Brownian paths and integrals for adaptive and high order SDE solvers". In: *arXiv preprint arXiv:2405.06464* (cit. on p. 25).

Jiang, Lei, Ge, Wen, Cariou-Kotlarek, Niels, Yi, Mingxuan, Chen, Po-Yu, Yang, Lingyi, Buet-Golfouse, Francois, Mittal, Gaurav, and Ni, Hao (2025). "Sig-DEG for Distillation: Making Diffusion Models Faster and Lighter". In: *arXiv preprint arXiv:2508.16939* (cit. on p. 24).

Karhunen, Kari (1950). "Über die Struktur stationärer zufälliger Funktionen". In: *Arkiv för Matematik* 1.2, pp. 141–160 (cit. on p. 25).

Karras, Tero, Aittala, Miika, Lehtinen, Jaakko, Hellsten, Janne, Aila, Timo, and Laine, Samuli (2024). "Analyzing and improving the training dynamics of diffusion models". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 24174–24184 (cit. on pp. 8, 27).

Kidger, Patrick (2022). "On Neural Differential Equations". Available at https://arxiv.org/abs/2202.02435. Ph.D. thesis. Oxford University (cit. on pp. 1, 25).

Kidger, Patrick, Foster, James, Li, Xuechen, and Lyons, Terry (2021). "Efficient and Accurate Gradients for Neural SDEs". In: *Advances in Neural Information Processing Systems*. Ed. by A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (cit. on pp. 2, 25).

Kim, Dongjun, Lai, Chieh-Hsin, Liao, Wei-Hsiang, Murata, Naoki, Takida, Yuhta, Uesaka, Toshimitsu, He, Yutong, Mitsufuji, Yuki, and Ermon, Stefano (2024). "Consistency Trajectory Models: Learning Probability Flow ODE Trajectory of Diffusion". In: *The Twelfth International Conference on Learning Representations* (cit. on pp. 1, 3).

Kiyohara, Naoki, Johns, Edward, and Li, Yingzhen (2025). "Neural Stochastic Flows: Solver-Free Modelling and Inference for SDE Solutions". In: *The Thirty-ninth Annual Conference on Neural Information Processing Systems* (cit. on pp. 3, 24).

Köhler, Jonas, Krämer, Andreas, and Noé, Frank (2021). "Smooth normalizing flows". In: *Advances in Neural Information Processing Systems* 34, pp. 2796–2809 (cit. on p. 28).

Krizhevsky, Alex, Hinton, Geoffrey, et al. (2009). "Learning multiple layers of features from tiny images". In: (cit. on p. 8).

Kullback, Solomon and Leibler, Richard A (1951). "On information and sufficiency". In: *The annals of mathematical statistics* 22.1, pp. 79–86 (cit. on p. 25).

Li, Xuechen, Wong, Ting-Kam Leonard, Chen, Ricky T. Q., and Duvenaud, David K. (Aug. 2020). "Scalable Gradients and Variational Inference for Stochastic Differential Equations". In: *Proceedings of The 2nd Symposium on Advances in Approximate Bayesian Inference*. Ed. by Cheng Zhang, Francisco Ruiz, Thang Bui, Adji Bousso Dieng, and Dawen Liang. Vol. 118. Proceedings of Machine Learning Research. PMLR, pp. 1–28 (cit. on pp. 2, 25).

Lipman, Yaron, Chen, Ricky T. Q., Ben-Hamu, Heli, Nickel, Maximilian, and Le, Matthew (2023). "Flow Matching for Generative Modeling". In: *The Eleventh International Conference on Learning Representations* (cit. on pp. 1, 3).

Lipman, Yaron, Havasi, Marton, Holderrieth, Peter, Shaul, Neta, Le, Matt, Karrer, Brian, Chen, Ricky TQ, Lopez-Paz, David, Ben-Hamu, Heli, and Gat, Itai (2024). *Flow Matching Guide and Code* (cit. on p. 3).

Liu, Xingchao, Gong, Chengyue, and Liu, Qiang (2023). "Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow". In: *The Eleventh International Conference on Learning Representations* (cit. on pp. 1, 3).

Liu, Ziwei, Luo, Ping, Wang, Xiaogang, and Tang, Xiaoou (Dec. 2015). "Deep Learning Face Attributes in the Wild". In: *Proceedings of International Conference on Computer Vision (ICCV)* (cit. on p. 8).

Loève, Michel (1945). "Sur les fonctions aléatoires stationnaires du second ordre". In: *Revue Scientifique* 83, pp. 297–303 (cit. on p. 25).

Lu, Cheng, Zhou, Yuhao, Bao, Fan, Chen, Jianfei, Li, Chongxuan, and Zhu, Jun (2022). "DPM-Solver: A Fast ODE Solver for Diffusion Probabilistic Model Sampling in Around 10 Steps". In: *Advances in Neural Information Processing Systems*. Ed. by Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (cit. on p. 1).

Lyons, Terry J (1998). "Differential equations driven by rough signals". In: *Revista Matemática Iberoamericana* 14.2, pp. 215–310 (cit. on pp. 4, 5).

Maoutsa, Dimitra, Reich, Sebastian, and Opper, Manfred (2020). "Interacting particle solutions of fokker–planck equations through gradient–log–density estimation". In: *Entropy* 22.8, p. 802 (cit. on p. 3).

Nie, Shen, Guo, Hanzhong Allan, Lu, Cheng, Zhou, Yuhao, Zheng, Chenyu, and Li, Chongxuan (2024). "The Blessing of Randomness: SDE Beats ODE in General Diffusion-based Image Editing". In: *The Twelfth International Conference on Learning Representations* (cit. on p. 3).

Oh, YongKyung, Lim, Dongyoung, and Kim, Sungil (2024). "Stable Neural Stochastic Differential Equations in Analyzing Irregular Time Series Data". In: *The Twelfth International Conference on Learning Representations* (cit. on p. 1).

Øksendal, Bernt (July 2003). *Stochastic Differential Equations: An Introduction with Applications*. Universitext. Berlin, Germany: Springer Berlin Heidelberg. ISBN: 9783662036204. DOI: 10.1007/978-3-642-14394-6 (cit. on pp. 2, 4, 15, 18).

Pan, Zhengkai, Potaptchik, Peter, Yao, Wenxi, Albergo, Michael S, and Pidstrigach, Jakiw (2026). "Itô maps for any-step SDEs". In: *arXiv preprint arXiv:2606.11156* (cit. on p. 25).

Passaro, Romeo, Blasingame, Zander W., Bronstein, Michael M., and Tong, Alexander (2026). "Stochastic Few-step Models". In: *ICLR 2026 2nd Workshop on Deep Generative Model in Machine Learning: Theory, Principle and Efficacy* (cit. on pp. 2, 3, 24).

Peluchetti, Stefano (2021). *Non-Denoising Forward-Time Diffusions* (cit. on pp. 1, 3).

Plainer, Michael, Wu, Hao, Klein, Leon, Günnemann, Stephan, and Noe, Frank (2025). "Consistent Sampling and Simulation: Molecular Dynamics with Energy-Based Diffusion Models". In: *The Thirty-ninth Annual Conference on Neural Information Processing Systems* (cit. on pp. 9, 28).

Poli, Michael, Massaroli, Stefano, Yamashita, Atsushi, Asama, Hajime, and Park, Jinkyoo (2020). "Hypersolvers: Toward Fast Continuous-Depth Models". In: *Advances in Neural Information*

*Processing Systems*. Ed. by H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin. Vol. 33. Curran Associates, Inc., pp. 21105–21117 (cit. on p. 24).

Potaptchik, Peter, Saravanan, Adhi, Mammadov, Abbas, Prat, Alvaro, Albergo, Michael S, and Teh, Yee Whye (2026). "Meta Flow Maps enable scalable reward alignment". In: *arXiv preprint arXiv:2601.14430* (cit. on pp. 2, 3, 24).

Rector-Brooks, Jarrid, Lambert, Théophile, Skreta, Marta, Roth, Daniel, Long, Yueming, Li, Zi-Qi, Zhang, Xi, Cretu, Miruna, Li, Francesca-Zhoufan, Ganapathy, Tanvi, Jin, Emily, Bose, Avishek Joey, Yang, Jason, Neklyudov, Kirill, Bengio, Yoshua, Tong, Alexander, Arnold, Frances H., and Liu, Cheng-Hao (2026). "General Multimodal Protein Design Enables DNA-Encoding of Chemistry". In: arXiv: 2604.05181 [cs.LG] (cit. on p. 1).

Rehman, Danyal, Akhound-Sadegh, Tara, Gazizov, Artem, Bengio, Yoshua, and Tong, Alexander (2026). "FALCON: Few-step Accurate Likelihoods for Continuous Flows". In: *The Fourteenth International Conference on Learning Representations* (cit. on p. 1).

Rezende, Danilo and Mohamed, Shakir (July 2015). "Variational Inference with Normalizing Flows". In: *Proceedings of the 32nd International Conference on Machine Learning*. Ed. by Francis Bach and David Blei. Vol. 37. Proceedings of Machine Learning Research. Lille, France: PMLR, pp. 1530–1538 (cit. on p. 25).

Rombach, Robin, Blattmann, Andreas, Lorenz, Dominik, Esser, Patrick, and Ommer, Björn (2022). "High-resolution image synthesis with latent diffusion models". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695 (cit. on p. 1).

Sabour, Amirmojtaba, Fidler, Sanja, and Kreis, Karsten (2025). "Align Your Flow: Scaling Continuous-Time Flow Map Distillation". In: *The Thirty-ninth Annual Conference on Neural Information Processing Systems* (cit. on pp. 1, 3).

Sohl-Dickstein, Jascha, Weiss, Eric, Maheswaranathan, Niru, and Ganguli, Surya (2015). "Deep unsupervised learning using nonequilibrium thermodynamics". In: *International conference on machine learning*. pmlr, pp. 2256–2265 (cit. on p. 1).

Song, Yang and Dhariwal, Prafulla (2024). "Improved Techniques for Training Consistency Models". In: *The Twelfth International Conference on Learning Representations* (cit. on p. 24).

Song, Yang, Dhariwal, Prafulla, Chen, Mark, and Sutskever, Ilya (23–29 Jul 2023). "Consistency Models". In: *Proceedings of the 40th International Conference on Machine Learning*. Ed. by Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett. Vol. 202. Proceedings of Machine Learning Research. PMLR, pp. 32211–32252 (cit. on pp. 1, 3, 8, 24).

Song, Yang, Sohl-Dickstein, Jascha, Kingma, Diederik P, Kumar, Abhishek, Ermon, Stefano, and Poole, Ben (2021). "Score-Based Generative Modeling through Stochastic Differential Equations". In: *International Conference on Learning Representations* (cit. on pp. 1, 3, 8).

Tong, Alexander, Fatras, Kilian, Malkin, Nikolay, Huguet, Guillaume, Zhang, Yanlei, Rector-Brooks, Jarrid, Wolf, Guy, and Bengio, Yoshua (2024). "Improving and generalizing flow-based generative models with minibatch optimal transport". In: *Transactions on Machine Learning Research*. Expert Certification. ISSN: 2835-8856 (cit. on p. 3).

Walker, Benjamin, McLeod, Andrew Donald, Qin, Tiexin, Cheng, Yichuan, Li, Haoliang, and Lyons, Terry (2024). "Log Neural Controlled Differential Equations: The Lie Brackets Make A Difference". In: *Forty-first International Conference on Machine Learning* (cit. on p. 1).

Watson, Joseph L, Juergens, David, Bennett, Nathaniel R, Trippe, Brian L, Yim, Jason, Eisenach, Helen E, Ahern, Woody, Borst, Andrew J, Ragotte, Robert J, Milles, Lukas F, et al. (2023). "De novo design of protein structure and function with RFdiffusion". In: *Nature* 620.7976, pp. 1089–1100 (cit. on p. 1).

Zhang, Qinsheng and Chen, Yongxin (2023). "Fast Sampling of Diffusion Models with Exponential Integrator". In: *The Eleventh International Conference on Learning Representations* (cit. on pp. 1, 24).

# Appendices


A Contribution statement . . . 15

B Proofs . . . 15

B.1 Constructing the Itô map . . . 15

Proof of Lemma 3.1 • Proof of Proposition 3.2 • Proof of Proposition 3.3 • Proof of Theorem 3.5

B.2 Constructing the Brownian motion . . . 19

Primer on rough path theory • Proof of Theorem 3.4

C Training . . . 23

C.1 Euler-Maruyama step objective . . . 23

C.2 Diffusion SDEs . . . 24

D Extended related work . . . 24

E Experimental details and further experiments . . . 25

E.1 Non-linear SDE . . . 25

E.2 Image generation . . . 27

E.3 Molecular systems: ALDP . . . 28

E.4 Molecular systems: Chignolin . . . 30

E.5 Path-dependent committer times . . . 30

E.6 Hardware . . . 32

E.7 Repositories . . . 32

F Broader impacts . . . 32


## A Contribution statement

SM and ZB initially conceived the idea and developed it over correspondence with NR and JF. The development of the theory was led by SM. The rough path convergence of the Brownian polynomial was shown by TH. The image experiments and related open-source code were led by SM and ZB. The molecular experiments and related open-source code were led by SM (ALDP) and NR (Chignolin). SM and ZB wrote the paper, with contributions from all other authors. AT and JF guided the project.

## B Proofs

In this section we prove the results provided in the main text. Throughout, we assume that the stochastic differential equations studied satisfy the following assumption.

**Assumption B.1** (Unique strong solution)**.** *We assume that the additive-noise stochastic differential equation given by*

$$d\boldsymbol{X}_t = \boldsymbol{f}(t, \boldsymbol{X}_t)\,\mathrm{d}t + \boldsymbol{g}(t)\,\mathrm{d}\boldsymbol{W}_t,$$

*where $\boldsymbol{f} : \mathbb{R} \times \mathbb{R}^d \to \mathbb{R}^d, \boldsymbol{g} : \mathbb{R} \to \mathbb{R}^{d\times w}$ with $\boldsymbol{W}_t \in \mathbb{R}^w$ and initial condition $\boldsymbol{X}_0$ with finite second moment, has the following properties such that a unique continuous solution $\boldsymbol{X}_t$ exists. Let $\boldsymbol{f}, \boldsymbol{g}$ be measurable functions satisfying*

$$|\boldsymbol{f}(t, \boldsymbol{x})| + |\boldsymbol{g}(t)| \leq C(1 + |\boldsymbol{x}|)$$
$$|\boldsymbol{f}(t, \boldsymbol{x}) - \boldsymbol{f}(t, \boldsymbol{y})| \leq D|\boldsymbol{x} - \boldsymbol{y}|$$

*for constants $C, D$. See Øksendal (2003) for a full discussion on the existence and unique of strong solutions to SDEs.*

### B.1 Constructing the Itô map

#### B.1.1 Proof of Lemma 3.1

We first prove a simple lemma that relates the Itô map to the SDE.

**Lemma B.1** (Stochastic Lagrangian equation)**.** *Let $\boldsymbol{\Psi}_{s,t}(\boldsymbol{X}_s, \boldsymbol{W}_{[s,t]})$ denote the Itô map for (6). Then,*

$$\mathrm{d}\boldsymbol{\Psi}_{s,t}(\boldsymbol{X}_s, \boldsymbol{W}_{[s,t]}) = \boldsymbol{f}(t, \boldsymbol{\Psi}_{s,t}(\boldsymbol{X}_s, \boldsymbol{W}_{[s,t]}))\mathrm{d}t + \boldsymbol{g}(t)\mathrm{d}\boldsymbol{W}_t. \tag{18}$$

*Proof.* Consider the derivative of the stochastic flow map,

$$\begin{aligned}\mathrm{d}\boldsymbol{\Psi}_{s,t}(\boldsymbol{X}_s, \boldsymbol{W}_{[s,t]}) &= \mathrm{d}\boldsymbol{X}_t, \\ &= \boldsymbol{f}(t, \boldsymbol{X}_t)\mathrm{d}t + \boldsymbol{g}(t)\mathrm{d}\boldsymbol{W}_t, \\ &= \boldsymbol{f}(t, \boldsymbol{\Psi}_{s,t}(\boldsymbol{X}_s, \boldsymbol{W}_{[s,t]}))\mathrm{d}t + \boldsymbol{g}(t)\mathrm{d}\boldsymbol{W}_t.\end{aligned}$$

□

Now, we are able to prove Lemma 3.1 which we restate here.

**Lemma 3.1** (Tangent Condition)**.** *Let $\boldsymbol{\Psi}_{s,t}(\boldsymbol{X}_s, \boldsymbol{W}_{[s,t]})$ denote the Itô map for (6). Then,*

$$\lim_{s \to t} \mathrm{d}\boldsymbol{\Psi}_{s,t}(\boldsymbol{X}_s, \boldsymbol{W}_{[s,t]}) = \boldsymbol{f}(t, \boldsymbol{X}_t)\ \mathrm{d}t + \boldsymbol{g}(t)\ \mathrm{d}\boldsymbol{W}_t. \tag{7}$$

*Proof.* By Lemma B.1 we have that the Itô map satisfies the stochastic Lagrangian equation. Taking the limit as $s \to t$, then given the continuity of the Itô map we have

$$\begin{aligned}\lim_{s \to t} \mathrm{d}\boldsymbol{\Psi}_{s,t}(\boldsymbol{X}_s, \boldsymbol{W}_{[s,t]}) &= \lim_{s \to t} \boldsymbol{f}(t, \boldsymbol{\Psi}_{s,t}(\boldsymbol{X}_s, \boldsymbol{W}_{[s,t]}))\mathrm{d}t + \boldsymbol{g}(t)\mathrm{d}\boldsymbol{W}_t \\ &= \boldsymbol{f}(t, \boldsymbol{\Psi}_{t,t}(\boldsymbol{X}_t, \boldsymbol{W}_{[t,t]}))\mathrm{d}t + \boldsymbol{g}(t)\mathrm{d}\boldsymbol{W}_t \\ &= \boldsymbol{f}(t, \boldsymbol{X}_t)\mathrm{d}t + \boldsymbol{g}(t)\mathrm{d}\boldsymbol{W}_t\end{aligned}$$

□

### B.1.2 Proof of Proposition 3.2

Next, we prove Proposition 3.2.

**Proposition 3.2** (Strong Stochastic Flow Map)**.** *Consider an Euler-Maruyama parameterization of the Itô map,*

$$\boldsymbol{\Psi}_{s,t}(\boldsymbol{X}_s, \boldsymbol{W}_{[s,t]}) = \boldsymbol{X}_s + \boldsymbol{f}_{s,t}(\boldsymbol{X}_s, \boldsymbol{W}_{[s,t]})(t - s) + \boldsymbol{g}_{s,t}(\boldsymbol{W}_{[s,t]})(\boldsymbol{W}_t - \boldsymbol{W}_s), \tag{8}$$

*where $\boldsymbol{f}, \boldsymbol{g}$ are twice continuously differentiable, Lipschitz in both time arguments, satisfy the conditions that $\boldsymbol{f}_{t,t}(\boldsymbol{X}_t, \boldsymbol{W}_{[s,t]}) = \boldsymbol{f}_{t,t}(\boldsymbol{X}_t)$, and $\boldsymbol{g}_{t,t}(\boldsymbol{W}_{[s,t]}) = \boldsymbol{g}_{t,t}$ (i.e. the coefficients are independent of $\boldsymbol{W}_{[s,t]}$ for $s = t$). Then, $\boldsymbol{\Psi}_{s,t}$ satisfies the tangent condition (7) if and only if*

$$\boldsymbol{f}_{t,t}(x_t) = \boldsymbol{f}(t, x_t), \quad \boldsymbol{g}_{t,t} = \boldsymbol{g}(t).$$

*Proof.* Consider the application of Itô's lemma to $\boldsymbol{\Psi}_{s,t}(\boldsymbol{X}_s, \boldsymbol{W}_{[s,t]})$, given by

$$\begin{aligned}
\mathrm{d}\boldsymbol{\Psi}_{s,t}(\boldsymbol{X}_s, \boldsymbol{W}_{[s,t]}) &= \left(\partial_t \boldsymbol{\Psi}_{s,t}(\boldsymbol{X}_s, \boldsymbol{W}_{[s,t]}) + \frac{1}{2}\partial^2_{\boldsymbol{W}_t} \boldsymbol{\Psi}_{s,t}(\boldsymbol{X}_s, \boldsymbol{W}_{[s,t]})\right) \mathrm{d}t \\
&\quad + \partial_{\boldsymbol{W}_t} \boldsymbol{\Psi}_{s,t}(\boldsymbol{X}_s, \boldsymbol{W}_{[s,t]}) \mathrm{d}\boldsymbol{W}_t, \\
&= \Big(\boldsymbol{f}_{s,t}(\boldsymbol{X}_s, \boldsymbol{W}_{[s,t]}) + (t-s)\partial_t \boldsymbol{f}_{s,t}(\boldsymbol{X}_s, \boldsymbol{W}_{[s,t]}) + (\boldsymbol{W}_t - \boldsymbol{W}_s)\partial_t \boldsymbol{g}_{s,t}(\boldsymbol{W}_{[s,t]}) \\
&\quad + \frac{1}{2}(t-s)\partial^2_{\boldsymbol{W}_t} \boldsymbol{f}_{s,t}(\boldsymbol{X}_s, \boldsymbol{W}_{[s,t]}) \\
&\quad + \frac{1}{2}(\boldsymbol{W}_t - \boldsymbol{W}_s)\partial^2_{\boldsymbol{W}_t} \boldsymbol{g}_{s,t}(\boldsymbol{W}_{[s,t]}) + \partial_{\boldsymbol{W}_t} \boldsymbol{g}_{s,t}(\boldsymbol{W}_{[s,t]})\Big) \mathrm{d}t \\
&\quad + \Big((t-s)\partial_{\boldsymbol{W}_t} \boldsymbol{f}_{s,t}(\boldsymbol{X}_s, \boldsymbol{W}_{[s,t]}) \\
&\quad + \boldsymbol{g}_{s,t}(\boldsymbol{W}_{[s,t]}) + (\boldsymbol{W}_t - \boldsymbol{W}_s)\partial_{\boldsymbol{W}_t} \boldsymbol{g}_{s,t}(\boldsymbol{W}_{[s,t]})\Big) \mathrm{d}\boldsymbol{W}_t.
\end{aligned}$$

Now, taking the limit $s \to t$, we obtain

$$\lim_{s\to t} \mathrm{d}\boldsymbol{\Psi}_{s,t}(\boldsymbol{X}_s, \boldsymbol{W}_{[s,t]}) = \Big(\boldsymbol{f}_{t,t}(\boldsymbol{X}_t, \boldsymbol{W}_{[t,t]}) + \partial_{\boldsymbol{W}_t} \boldsymbol{g}_{t,t}(\boldsymbol{W}_{[t,t]})\Big)\mathrm{d}t + \boldsymbol{g}_{t,t}\mathrm{d}\boldsymbol{W}_t.$$

Finally, using the conditions that $\boldsymbol{f}_{t,t}(\boldsymbol{X}_t, \boldsymbol{W}_{[t,t]}) = \boldsymbol{f}_{t,t}(\boldsymbol{X}_t)$ and $\boldsymbol{g}_{t,t}(\boldsymbol{W}_{[t,t]}) = \boldsymbol{g}_{t,t}$, this reduces to

$$\lim_{s\to t} \mathrm{d}\boldsymbol{\Psi}_{s,t}(\boldsymbol{X}_s, \boldsymbol{W}_{[s,t]}) = \boldsymbol{f}_{t,t}(\boldsymbol{X}_t)\mathrm{d}t + \boldsymbol{g}_{t,t}\mathrm{d}\boldsymbol{W}_t.$$

To satisfy the tangent condition (7) we must have $\boldsymbol{f}_{t,t}(\boldsymbol{X}_t) = \boldsymbol{f}(t, \boldsymbol{X}_t)$ and $\boldsymbol{g}_{t,t} = \boldsymbol{g}(t)$. Conversely, substituting $\boldsymbol{f}_{t,t}(\boldsymbol{X}_t) = \boldsymbol{f}(t, \boldsymbol{X}_t), \boldsymbol{g}_{t,t} = \boldsymbol{g}(t)$ recovers the tangent condition. □

### B.1.3 Proof of Proposition 3.3

Now onto the proof that the strong stochastic flow map attains the Itô map when satisfying the tangent and semigroup conditions.

**Proposition 3.3** (Semigroup condition)**.** *Let $\boldsymbol{\Psi}_{s,t}(\boldsymbol{X}_s, \boldsymbol{W}_{[s,t]})$ denote the strong stochastic flow map satisfying (7) and (8). Then $\boldsymbol{\Psi}_{s,t}(\boldsymbol{X}_s, \boldsymbol{W}_{[s,t]})$ is the Itô map if and only if the semigroup property (9) holds.*

*Proof.* First observe that the Itô map satisfies the semigroup condition,

$$\begin{aligned}
\boldsymbol{\Psi}_{s,t}(\boldsymbol{X}_s, \boldsymbol{W}_{[s,t]}) &= \boldsymbol{\Psi}_{u,t}(\boldsymbol{\Psi}_{s,u}(\boldsymbol{X}_s, \boldsymbol{W}_{[s,u]}), \boldsymbol{W}_{[u,t]}) \\
&= \boldsymbol{\Psi}_{u,t}(\boldsymbol{X}_u, \boldsymbol{W}_{[u,t]}) \\
&= \boldsymbol{X}_t
\end{aligned}$$

Next, we want to show the inverse implication. Let $\boldsymbol{\Psi}_{s,t}(\boldsymbol{X}_s, \boldsymbol{W}_{[s,t]})$ denote the strong stochastic flow map satisfying (7), (8) and (9). We show that this map is the Itô map for (6). This follows by considering the semigroup condition,

$$\begin{aligned}
\boldsymbol{\Psi}_{s,t+h}(\boldsymbol{X}_s, \boldsymbol{W}_{[s,t+h]}) &= \boldsymbol{\Psi}_{t,t+h}(\boldsymbol{\Psi}_{s,t}(\boldsymbol{X}_s, \boldsymbol{W}_{[s,t]}), \boldsymbol{W}_{[t,t+h]}) \\
&= \boldsymbol{\Psi}_{s,t}(\boldsymbol{X}_s, \boldsymbol{W}_{[s,t]}) + h\boldsymbol{f}_{t,t+h}(\boldsymbol{\Psi}_{s,t}(\boldsymbol{X}_s, \boldsymbol{W}_{[s,t]}), \boldsymbol{W}_{[t,t+h]}) \\
&\quad + (\boldsymbol{W}_{t+h} - \boldsymbol{W}_t)\boldsymbol{g}_{t,t+h}(\boldsymbol{W}_{[t,t+h]})
\end{aligned}$$

Define $\hat{\boldsymbol{X}}_t = \boldsymbol{\Psi}_{s,t}(\boldsymbol{X}_s, \boldsymbol{W}_{[s,t]})$ and re-arrange to get

$$\hat{\boldsymbol{X}}_{t+h} - \hat{\boldsymbol{X}}_t = h\boldsymbol{f}_{t,t+h}(\hat{\boldsymbol{X}}_t, \boldsymbol{W}_{[t,t+h]}) + (\boldsymbol{W}_{t+h} - \boldsymbol{W}_t)\boldsymbol{g}_{t,t+h}(\boldsymbol{W}_{[t,t+h]}).$$

Next consider taking the following sum over a partition $t = u_0 < u_1 < \cdots < u_n = t + h$,

$$\begin{aligned}
\sum_i \hat{\boldsymbol{X}}_{u_{i+1}} - \hat{\boldsymbol{X}}_{u_i} &= \sum_i (u_{i+1} - u_i)\boldsymbol{f}_{u_i,u_{i+1}}(\hat{\boldsymbol{X}}_{u_i}, \boldsymbol{W}_{[u_i,u_{i+1}]}) \\
&\quad + \sum_i (\boldsymbol{W}_{u_{i+1}} - \boldsymbol{W}_{u_i})\boldsymbol{g}_{u_i,u_{i+1}}(\boldsymbol{W}_{[u_i,u_{i+1}]})
\end{aligned}$$

The LHS of this sum telescopes to $\hat{\boldsymbol{X}}_{t+h} - \hat{\boldsymbol{X}}_t$. We now take the limit of this partition with $\max_i(u_{i+1} - u_i) \to 0$.

Since we have $\boldsymbol{f}_{u,u}(\hat{\boldsymbol{X}}_u, \boldsymbol{W}_{[u,u]}) = \boldsymbol{f}_{u,u}(\hat{\boldsymbol{X}}_u)$ and $\boldsymbol{g}_{u,u}(\boldsymbol{W}_{[u,u]}) = \boldsymbol{g}_{u,u}$ by the independence assumption on (8), consider the following equivalent expression obtained by adding and subtracting the diagonal coefficients,

$$\begin{aligned}\hat{\boldsymbol{X}}_{t+h} - \hat{\boldsymbol{X}}_t = &\sum_i (u_{i+1} - u_i)\boldsymbol{f}_{u_i,u_i}(\hat{\boldsymbol{X}}_{u_i}) \\ &+ \sum_i (u_{i+1} - u_i)\Big(\boldsymbol{f}_{u_i,u_{i+1}}(\hat{\boldsymbol{X}}_{u_i}, \boldsymbol{W}_{[u_i,u_{i+1}]}) - \boldsymbol{f}_{u_i,u_i}(\hat{\boldsymbol{X}}_{u_i})\Big) \\ &+ \sum_i (\boldsymbol{W}_{u_{i+1}} - \boldsymbol{W}_{u_i})\boldsymbol{g}_{u_i,u_i} + \sum_i (\boldsymbol{W}_{u_{i+1}} - \boldsymbol{W}_{u_i})\Big(\boldsymbol{g}_{u_i,u_{i+1}}(\boldsymbol{W}_{[u_i,u_{i+1}]}) - \boldsymbol{g}_{u_i,u_i}\Big)\end{aligned}$$

The two difference terms (in both $\boldsymbol{f}$ and $\boldsymbol{g}$) decay to zero as $\max_i(u_{i+1} - u_i) \to 0$. This follows from the Lipschitz-in-time assumption on $\boldsymbol{f}, \boldsymbol{g}$. Specifically, for the $\boldsymbol{f}$ residual we have

$$\begin{aligned}\Big|\sum_i (u_{i+1} - u_i)\Big(\boldsymbol{f}_{u_i,u_{i+1}}(\hat{\boldsymbol{X}}_{u_i}, \boldsymbol{W}_{[u_i,u_{i+1}]}) - \boldsymbol{f}_{u_i,u_i}(\hat{\boldsymbol{X}}_{u_i})\Big)\Big| &\le L_f \sum_i |u_{i+1} - u_i||u_{i+1} - u_i|, \\ &\le L_f \max_i(|u_{i+1} - u_i|) \sum_i |u_{i+1} - u_i|, \\ &\le L_f h \max_i(|u_{i+1} - u_i|),\end{aligned}$$

which tends to zero as $\max_i(u_{i+1} - u_i) \to 0$. For the $\boldsymbol{g}$ residual, we have

$$\begin{aligned}\Big|\sum_i (\boldsymbol{W}_{u_{i+1}} - \boldsymbol{W}_{u_i})\Big(\boldsymbol{g}_{u_i,u_{i+1}}(\boldsymbol{W}_{[u_i,u_{i+1}]}) - \boldsymbol{g}_{u_i,u_i}\Big)\Big| &\le L_g \sum_i |(\boldsymbol{W}_{u_{i+1}} - \boldsymbol{W}_{u_i})||u_{i+1} - u_i| \\ &\le L_g \max_i(|\boldsymbol{W}_{u_{i+1}} - \boldsymbol{W}_{u_i}|) \sum_i |u_{i+1} - u_i|, \\ &= h L_g \max_i(|\boldsymbol{W}_{u_{i+1}} - \boldsymbol{W}_{u_i}|),\end{aligned}$$

which tends to zero as $\max_i(u_{i+1} - u_i) \to 0$ by the uniform continuity of $\boldsymbol{W}_t$. We are left with

$$\hat{\boldsymbol{X}}_{t+h} - \hat{\boldsymbol{X}}_t = \lim_{\max_i(u_{i+1}-u_i)\to 0} \sum_i (u_{i+1} - u_i)\boldsymbol{f}_{u_i,u_i}(\hat{\boldsymbol{X}}_{u_i}) + \sum_i (\boldsymbol{W}_{u_{i+1}} - \boldsymbol{W}_{u_i})\boldsymbol{g}_{u_i,u_i}.$$

Therefore, both Riemann and Itô (Stratonovich, equivalently) converge (Øksendal 2003) to give

$$\begin{aligned}\hat{\boldsymbol{X}}_{t+h} &= \hat{\boldsymbol{X}}_t + \int_t^{t+h} \boldsymbol{f}_{u,u}(\hat{\boldsymbol{X}}_u)\mathrm{d}u + \int_t^{t+h} \boldsymbol{g}_{u,u}\mathrm{d}\boldsymbol{W}_u \\ &= \hat{\boldsymbol{X}}_t + \int_t^{t+h} \boldsymbol{f}(u, \hat{\boldsymbol{X}}_u)\mathrm{d}u + \int_t^{t+h} \boldsymbol{g}(u)\mathrm{d}\boldsymbol{W}_u,\end{aligned}$$

where $\boldsymbol{f}_{t,t}(\boldsymbol{X}_t) = \boldsymbol{f}(t, \boldsymbol{X}_t)$ and $\boldsymbol{g}_{t,t} = \boldsymbol{g}(t)$ follows from Proposition 3.2. Since this holds for any $t \ge s$ and $h \ge 0$, $\hat{\boldsymbol{X}}_t$ is a strong solution to the SDE with coefficients $\boldsymbol{f}(t, \boldsymbol{X}_t), \boldsymbol{g}(t)$. By uniqueness, this implies that $\boldsymbol{\Psi}_{s,t}(\boldsymbol{X}_s, \boldsymbol{W}_{[s,t]})$ is the Itô map for (6). □

#### B.1.4 Proof of Theorem 3.5

Given the results above, we are able to prove Theorem 3.5.

**Theorem 3.5** (Self-distillation Objective)**.** *Let $\boldsymbol{\Psi}_{s,t}(\boldsymbol{X}_s, \boldsymbol{W}_{[s,t]})$ denote the Itô map for (6). Then, this map is given by the strong stochastic flow map in (8) where $\boldsymbol{\Psi}$ is the unique global minimizer over $\hat{\boldsymbol{\Psi}}$ of*

$$\mathcal{L}_{SD}(\hat{\boldsymbol{\Psi}}) = \mathcal{L}_{\boldsymbol{f},\boldsymbol{g}}(\hat{\boldsymbol{\Psi}}) + \mathcal{L}_D(\hat{\boldsymbol{\Psi}}), \tag{14}$$

*where*

$$\mathcal{L}_{\boldsymbol{f},\boldsymbol{g}}(\hat{\boldsymbol{\Psi}}) = \mathbb{E}_{t,\boldsymbol{X}_t}\left[\|\boldsymbol{f}(t,\boldsymbol{X}_t) - \hat{\boldsymbol{f}}_{t,t}(\boldsymbol{X}_t)\|_2^2 + \|\boldsymbol{g}(t) - \hat{\boldsymbol{g}}_{t,t}\|_2^2\right], \tag{15}$$

$$\mathcal{L}_D(\hat{\boldsymbol{\Psi}}) = \mathbb{E}_{s,u,t,\boldsymbol{X}_s,\boldsymbol{W}_{[s,t]}}\left[\|\hat{\boldsymbol{\Psi}}_{s,t}(\boldsymbol{X}_s,\boldsymbol{W}_{[s,t]}) - \hat{\boldsymbol{\Psi}}_{u,t}(\hat{\boldsymbol{\Psi}}_{s,u}(\boldsymbol{X}_s,\boldsymbol{W}_{[s,u]}),\boldsymbol{W}_{[u,t]})\|_2^2\right]. \tag{16}$$

*Proof.* We have that for any $\hat{\boldsymbol{\Psi}}$,

$$\mathcal{L}_{\boldsymbol{f},\boldsymbol{g}}(\hat{\boldsymbol{\Psi}}) \geq \mathcal{L}_{\boldsymbol{f},\boldsymbol{g}}(\boldsymbol{\Psi}) = 0,$$
$$\mathcal{L}_{\mathrm{D}}(\hat{\boldsymbol{\Psi}}) \geq 0.$$

This follows since $\mathcal{L}_{\boldsymbol{f},\boldsymbol{g}}$ is convex in $\hat{\boldsymbol{f}}_{t,t}, \hat{\boldsymbol{g}}_{t,t}$ with unique global minimizer $\boldsymbol{f}_{t,t}, \boldsymbol{g}_{t,t}$. Note that the Itô map satisfies $\mathcal{L}_{\mathrm{D}}(\boldsymbol{\Psi}) = 0$. This implies that for the Itô map we obtain the minimum of the objective, given by

$$\mathcal{L}_{\mathrm{SD}}(\boldsymbol{\Psi}) = 0.$$

To show that the minimizer is unique, consider any $\hat{\boldsymbol{\Psi}}$ that obtains the minimum,

$$\mathcal{L}_{SD}(\hat{\boldsymbol{\Psi}}) = 0$$

Then we must have

$$\hat{\boldsymbol{f}}_{t,t} = \boldsymbol{f}(t,\boldsymbol{X}_t), \quad \hat{\boldsymbol{g}}_{t,t} = \boldsymbol{g}(t),$$
$$\mathcal{L}_{\mathrm{D}}(\hat{\boldsymbol{\Psi}}) = 0.$$

By Propositions 3.2 (since $\hat{\boldsymbol{f}}_{t,t} = \boldsymbol{f}_{t,t}$, $\hat{\boldsymbol{g}}_{t,t} = \boldsymbol{g}_{t,t}$,, the tangent condition holds) and 3.3 (since $\mathcal{L}_{\mathrm{D}} = 0$, the semigroup condition holds), this implies $\hat{\boldsymbol{\Psi}}$ is the Itô map for (6). □

### B.2 Constructing the Brownian motion

#### B.2.1 Primer on rough path theory

A (step-2) rough path is an element of $G^2(\mathbb{R}^d) = \mathbb{R} \oplus \mathbb{R}^d \oplus \mathbb{R}^{d\times d}$. We define a metric on $\mathbb{X} = (1, \mathbb{X}^{(1)}, \mathbb{X}^{(2)}) \in G^2(\mathbb{R}^d)$ by,

$$d(\mathbb{X}_s, \mathbb{X}_t) \coloneqq |\mathbb{X}_{s,t}^{(1)}| + |\mathbb{X}_{s,t}^{(2)}|^{\frac{1}{2}},$$

where we have used $|\cdot|$ to indicate the *Euclidean norm on* $\mathbb{R}^{\otimes k}$ for the appropriate choice of $k \in \{1,2\}$. To measure the "difference" between two rough paths $\mathbb{X}, \mathbb{Y} \in G^2(\mathbb{R}^2)$ we define a pathwise metric, for each $\alpha \in [0, \frac{1}{2})$, the $\alpha$-Hölder metric by,

$$d_{\alpha\text{-Höl}}(\mathbb{X},\mathbb{Y}) \coloneqq \sup_{0\leq s<t\leq 1}\Big(\frac{|\mathbb{X}_{s,t}^{(1)} - \mathbb{Y}_{s,t}^{(1)}| + |\mathbb{X}_{s,t}^{(2)} - \mathbb{Y}_{s,t}^{(2)}|^{\frac{1}{2}}}{|t-s|^{\alpha}}\Big).$$

To avoid ambiguity of indices being used to represent both the *level* of a rough path, and in Definition 3.1 the order of our approximation, we will sometimes use the projection operator, $\pi_k : G^2(\mathbb{R}^d) \to (\mathbb{R}^d)^{\otimes k}$ for $k = 1, 2$, to explicitly distinguish the first and second levels of the rough path respectively, i.e. $\pi_k(\mathbb{X}) = \mathbb{X}^{(k)}$.

For both Brownian motion $\boldsymbol{W}_t$ and the polynomial approximation $\boldsymbol{W}_t^{(N)}$ from Definition 3.1, we lift the paths to rough paths $\mathbb{W}$ and $\mathbb{W}^{(N)}$ respectively by defining the canonical Stratonovich lifts,

$$\pi_2(\mathbb{W}_{s,t}) \coloneqq \int_s^t \boldsymbol{W}_{s,u} \otimes \mathrm{d}\boldsymbol{W}_u,$$
$$\pi_2(\mathbb{W}_{s,t}^{(N)}) \coloneqq \int_s^t \boldsymbol{W}_{s,u}^{(N)} \otimes \mathrm{d}\boldsymbol{W}_u^{(N)}.$$

#### B.2.2 Proof of Theorem 3.4

**Theorem 3.4** (Properties of $\boldsymbol{W}_{u,v}^{(N)}$)**.** *The polynomial approximation of the Brownian motion in Definition 3.1 has the following properties:*

1. *Converges to Brownian motion in the $\alpha$-Hölder distance, with $\alpha \in [0, \frac{1}{2})$,*
2. *The coefficients $\boldsymbol{I}_{s,t}^{(n)}$ are independently and normally distributed with*

$$\boldsymbol{I}_{s,t}^{(n)} \sim \mathcal{N}\left(\mathbf{0}, \frac{(t-s)}{2n+1}\boldsymbol{I}\right), \tag{12}$$

3. *The coefficients $\boldsymbol{I}_{s,t}^{(n)}$ admit closed form Chen relations.*

*Proof.* Property 2 is simply an application of Itô's isometry, using the fact that the $n^{\text{th}}$ (shifted) Legendre polynomial has a normalisation constant of $\frac{1}{2n+1}$,

$$\boldsymbol{I}_{s,t}^{(n)} = \int_s^t \widetilde{P}_n\Big(\frac{r-s}{t-s}\Big)\,\mathrm{d}\boldsymbol{W}_r \sim \mathcal{N}\big(0, K(s,t)\big),$$

where,

$$\begin{aligned} K(s,t) &= \mathbb{E}\Big[\int_s^t \widetilde{P}_n\Big(\frac{r-s}{t-s}\Big)\,\mathrm{d}\boldsymbol{W}_r \otimes \int_s^t \widetilde{P}_n\Big(\frac{r-s}{t-s}\Big)\,\mathrm{d}\boldsymbol{W}_r\Big] \\ &= \boldsymbol{I}\int_s^t \widetilde{P}_n\Big(\frac{r-s}{t-s}\Big)^2\,\mathrm{d}r \\ &= \frac{t-s}{2n+1}\boldsymbol{I}. \end{aligned}$$

Property 1 and 3 require slightly more work, so are proven individually as Proposition B.3 and Proposition B.5 respectively. □

The following lemma will be required to establish rough-path convergence of the polynomial approximation $\boldsymbol{W}_t$.

**Lemma B.2** (Uniform bounds on $\mathbb{W}$)**.** *For $\alpha \in [0, \frac{1}{2})$, there exists $C_\alpha \in L^2(\mathbb{P})$ such that for all $[u,v] \subseteq [s,t]$,*

$$\|\mathbb{W}_{u,v}\| \le C_\alpha |v-u|^\alpha \tag{19}$$

*where $\|\cdot\| : G^2(\mathbb{R}^d) \to \mathbb{R}$ denotes the norm,*

$$\|\mathbb{X}_{u,v}\| = \max\Big(\big\|\mathbb{X}_{u,v}^{(1)}\big\|_{L^2}, \big\|\mathbb{X}_{u,v}^{(2)}\big\|_{L^2}^{\frac{1}{2}}\Big).$$

*Proof.* This is a standard application of the Garsia–Rodemich–Rumsey lemma (Garsia et al. 1970, Lemma 1.1), which can be generalised to any metric as seen in (Friz and Victoir 2010, Proposition A.8). Choosing $p(u) = |u|^{1/2}$, $\Psi(u) = u^q$ for some $q \ge \frac{4}{1-2\alpha}$ and $B = \int_0^1\int_0^1 \Psi\big(\frac{\|\mathbb{W}_{u,v}\|}{p(v-u)}\big)\,\mathrm{d}u\,\mathrm{d}v$, we obtain,

$$\|\mathbb{W}_{u,v}\| \le C_\alpha |v-u|^{1/2-2/q} \le C_\alpha |v-u|^\alpha,$$

where $C_\alpha := \frac{8q(4B)^{1/q}}{q-4}$. To verify $C_\alpha \in L^2(\mathbb{P})$ we use $\|\mathbb{W}_{u,v}\|_{L^2(\mathbb{P})} = |v-u|^{1/2}$

$$\begin{aligned} \mathbb{E}[C_\alpha^2] &\lesssim \mathbb{E}[B^{2/q}] \\ &\le \int_0^1\int_0^1 \frac{\|\mathbb{W}_{u,v}\|_{L^2(\mathbb{P})}^2}{|v-u|^{2\alpha}}\,\mathrm{d}u\,\mathrm{d}v \\ &= \int_0^1\int_0^1 |v-u|^{1-2\alpha}\,\mathrm{d}u\,\mathrm{d}v \\ &< \infty. \end{aligned}$$

□

**Proposition B.3** (Rough path convergence of the polynomial approximation)**.** *The polynomial expansion of Brownian motion given in Definition 3.1 converges in the rough-path sense, i.e. for any* $\alpha \in [0, \frac{1}{2})$*,*

$$d_{\alpha-H\ddot{o}l}(\mathbb{W}, \mathbb{W}^N) \xrightarrow{a.s.} 0,$$

*as* $N \to \infty$*.*

*Proof.* As shown in (Foster et al. 2020, Theorem 2.4), we can define the following filtration,

$$\left\{\mathscr{F}_N := \sigma(\{\boldsymbol{I}_{s,t}^{(n)} \ : \ n = 0, 1, \cdots, N\})\right\}_{N \geq 0},$$

so that the polynomial approximation admits the representation,

$$\mathbb{W}_{u,v}^{(N)} = \mathbb{E}[\mathbb{W}_{u,v} \,|\, \mathscr{F}_N],$$

for any $[u, v] \subseteq [s, t]$. Thus, by taking expectations with respect to $\mathscr{F}_N$ in Lemma B.2,

$$\|\mathbb{W}_{u,v}^{(N)}\| \leq \widetilde{C}_\alpha |v - u|^\alpha, \tag{20}$$

where $\widetilde{C}_\alpha := \sup_{n \geq 0} \mathbb{E}[C_\alpha \,|\, \mathscr{F}_n]$ and where we continue to use $\|\cdot\|$ as defined in Lemma B.2. By Doob's maximal inequality $\widetilde{C}_\alpha$ is finite a.s. since $\mathbb{E}[C_\alpha \,|\, \mathscr{F}_n]$ is a (discrete) martingale.

Combining Equation (19) and Equation (20), the sequence $\{\mathbb{W}_{u,v} - \mathbb{W}_{u,v}^{(N)}\}_{N \geq 0}$ is uniformly bounded and uniformly equicontinuous. Thus, invoking the Arzelà-Ascoli theorem, we conclude that there exists a uniformly convergent subsequence and since pointwise convergence is proven in (Foster et al. 2020), we know that this limit is zero, so,

$$\mathbb{W}_{u,v}^{(N)} \xrightarrow{a.s.} \mathbb{W}_{u,v} \qquad \text{uniformly as } N \to \infty.$$

Finally, defining $C^* = \max(C_\beta, \widetilde{C}_\beta)^{\frac{\alpha}{\beta}}$, use the following inequality,

$$\begin{aligned}
\frac{\|\pi_k(\mathbb{W}_{u,v} - \mathbb{W}_{u,v}^{(N)})\|_{L_2}}{|v-u|^{k\alpha}} &\leq \left(\frac{\|\pi_k(\mathbb{W}_{u,v} - \mathbb{W}_{u,v}^{(N)})\|_{L_2}}{|v-u|^{\beta}}\right)^{\frac{\alpha}{\beta}} \left(\sup_{0 \leq s < t \leq 1} \|\pi_k(\mathbb{W}_{u,v} - \mathbb{W}_{u,v}^{(N)})\|_{L_2}\right)^{1-\frac{\alpha}{\beta}} \\
&\leq C^* \left(\sup_{0 \leq u < v \leq 1} \|\pi_k(\mathbb{W}_{u,v} - \mathbb{W}_{u,v}^{(N)})\|_{L_2}\right)^{1-\frac{\alpha}{\beta}} \\
&\xrightarrow{a.s.} 0 \qquad \text{uniformly as } N \to \infty.
\end{aligned}$$

Thus, $d_{\alpha-\text{Höl}}(\mathbb{W}, \mathbb{W}^{(N)}) \xrightarrow{a.s.} 0$ as $N \to \infty$. □

Finally, we move to deriving the explicit Chen relations for the integrals $\boldsymbol{I}_{s,t}^{(n)}$ To achieve this, we first establish the following related dilation rules for the (shifted) Legendre polynomials.

**Lemma B.4** (Dilation rule for shifted Legendre polynomials)**.** *Let* $\widetilde{P}_n(x)$ *denote the* $n^{th}$ *shifted Legendre polynomial on* $[0, 1]$*. Then,*

$$\widetilde{P}_n(x) = \sum_{m=0}^{n} c_{n,m} \widetilde{P}_m(2x) = \sum_{m=0}^{n} (-1)^{n+m} c_{n,m} \widetilde{P}_m(2x - 1), \tag{21}$$

*where,*

$$c_{n,m} := (-1)^n (2m+1) \sum_{k=m}^{n} \left(-\frac{1}{2}\right)^k \frac{(n+k)!}{(n-k)!(k-m)!(k+m+1)!}.$$

*Proof.* Using the orthogonality relations for Legendre polynomials, we derive the following integral expression for $c_{n,m}$,

$$(2m+1) \int_0^1 \widetilde{P}_n\left(\frac{u}{2}\right) \widetilde{P}_m(u) \, \mathrm{d}u = (2m+1) \sum_{m=0}^{n} c_{n,m} \int_0^1 \widetilde{P}_n(u) \widetilde{P}_k(u) \, \mathrm{d}u = c_{n,m}.$$

We can evaluate the integral explicitly. By substituting Rodrigues' formula for $\widetilde{P}_m(u)$, before applying integration by parts $m$-times (noting that the boundary terms always vanish),

$$\begin{aligned}\int_0^1 \widetilde{P}_n\Big(\frac{u}{2}\Big)P_m(u)\,\mathrm{d}u &= \frac{1}{m!}\int_0^1 \widetilde{P}_n\Big(\frac{u}{2}\Big)\frac{\mathrm{d}^m}{\mathrm{d}u^m}[x^m(x-1)^m]\,\mathrm{d}u \\ &= \frac{1}{m!}\int_0^1 x^m(1-x)^m\frac{\mathrm{d}^m}{\mathrm{d}u^m}\widetilde{P}_n\Big(\frac{u}{2}\Big)\,\mathrm{d}u.\end{aligned}$$

Next, substituting the explicit form for the shifted Legendre polynomial,

$$\widetilde{P}_n\Big(\frac{u}{2}\Big) = (-1)^n\sum_{k=0}^n\binom{n}{k}\binom{n+k}{k}\Big(-\frac{x}{2}\Big)^k,$$

we obtain,

$$\begin{aligned}\int_0^1 \widetilde{P}_n\Big(\frac{u}{2}\Big)P_m(u)\,\mathrm{d}u &= \frac{1}{m!}\int_0^1 x^m(1-x)^m\frac{\mathrm{d}^m}{\mathrm{d}u^m}\Big[(-1)^n\sum_{k=0}^n\binom{n}{k}\binom{n+k}{k}\Big(-\frac{x}{2}\Big)^k\Big]\,\mathrm{d}u \\ &= \frac{(-1)^n}{m!}\sum_{k=m}^n\Big(-\frac{1}{2}\Big)^k\binom{n}{k}\binom{n+k}{k}\int_0^1 x^m(1-x)^m\frac{\mathrm{d}^m}{\mathrm{d}u^m}[x^k]\,\mathrm{d}u \\ &= \frac{(-1)^n}{m!}\sum_{k=m}^n\Big(-\frac{1}{2}\Big)^k\frac{k!}{(k-m)!}\binom{n}{k}\binom{n+k}{k}\int_0^1 x^k(1-x)^m\,\mathrm{d}u \\ &= \frac{(-1)^n}{m!}\sum_{k=m}^n\Big(-\frac{1}{2}\Big)^k\frac{k!}{(k-m)!}\binom{n}{k}\binom{n+k}{k}\frac{k!\,m!}{(k+m+1)!} \\ &= (-1)^n\sum_{k=m}^n\Big(-\frac{1}{2}\Big)^k\frac{(n+k)!}{(n-k)!(k-m)!(k+m+1)!}.\end{aligned}$$

For the second equation, use the symmetry of the Legendre polynomial,

$$\begin{aligned}\widetilde{P}_n(x) &= (-1)^n\widetilde{P}_n(1-x) \\ &= (-1)^n\sum_{m=0}^n c_{n,m}\widetilde{P}_m\big(2(1-x)\big) \\ &= (-1)^n\sum_{m=0}^n c_{n,m}\widetilde{P}_m\big(1-(2x-1)\big) \\ &= \sum_{m=0}^n(-1)^{n+m}c_{n,m}\widetilde{P}_m(2x-1).\end{aligned}$$

□

The proof of Proposition B.5 now follows naturally, by splitting the domain of the integral $\boldsymbol{I}_{s,t}^{(n)}$ into two and applying the relations we have just derived.

**Proposition B.5** (Chen relations for $\boldsymbol{I}^{(n)}$)**.** *Let $u = \dfrac{s+t}{2}$ be the midpoint of times $s \le t$. Then,*

$$\boldsymbol{I}_{s,t}^{(n)} = \sum_{m=0}^n c_{n,m}\Big(\boldsymbol{I}_{s,u}^{(m)} + (-1)^{n+m}\boldsymbol{I}_{u,t}^{(m)}\Big), \tag{22}$$

*where $c_{n,m}$ are defined as in Lemma B.4.*

*Proof.* By splitting $[s,t]$ into the two half-domains $[s,u]$ and $[u,t]$, we can apply the dilation rules for the Legendre polynomials given in Lemma B.4. We use the identities,

$$2\Big(\frac{r-s}{t-s}\Big) = \frac{r-s}{u-s},$$
$$2\Big(\frac{r-s}{t-s}\Big) - 1 = \frac{r-u}{t-u},$$

so that,

$$\begin{aligned}
\boldsymbol{I}_{s,t}^{(n)} &= \int_s^t \widetilde{P}_n\Big(\frac{r-s}{t-s}\Big)\,\mathrm{d}\boldsymbol{W}_r \\
&= \sum_{m=0}^{n} c_{n,m}\Big(\int_s^u \widetilde{P}_m\Big(\frac{r-s}{u-s}\Big)\,\mathrm{d}\boldsymbol{W}_r + (-1)^{n+m}\int_u^t \widetilde{P}_m\Big(\frac{r-u}{t-u}\Big)\,\mathrm{d}\boldsymbol{W}_r\Big) \\
&= \sum_{m=0}^{n} c_{n,m}\Big(\boldsymbol{I}_{s,u}^{(m)} + (-1)^{n+m}\boldsymbol{I}_{u,t}^{(m)}\Big),
\end{aligned}$$

as required. □

# C Training

## C.1 Euler-Maruyama step objective

Here we show that matching an Euler-Maruyama step with the strong stochastic flow map model is equivalent to a weighted coefficient loss.

**Lemma C.1** (Euler-Maruyama objective)**.** *Consider the following objective*

$$\mathcal{L}_{EM} = \big\|\boldsymbol{X}_t - \hat{\boldsymbol{X}}_t\big\|^2,$$

*where* $\hat{\boldsymbol{X}}_t = \boldsymbol{\Psi}_{s,t}(\boldsymbol{X}_s, \boldsymbol{W}_{[s,t]})$ *via* (8) *and*

$$\boldsymbol{X}_t = \boldsymbol{X}_s + \boldsymbol{f}(s,\boldsymbol{X}_s)(t-s) + \boldsymbol{g}(s)(\boldsymbol{W}_t - \boldsymbol{W}_s).$$

*Since the true coefficients* $\boldsymbol{f}(s,\boldsymbol{X}_s), \boldsymbol{g}(s)$ *depend only on* $s$*, it suffices to restrict the model to coefficients* $\boldsymbol{f}_{s,t}(\boldsymbol{X}_s)$ *and* $\boldsymbol{g}_{s,t}$ *that are independent of* $\boldsymbol{W}_{[s,t]}$*. Then*

$$\mathbb{E}[\mathcal{L}_{EM} \mid \boldsymbol{X}_s] = (t-s)^2\mathcal{L}_f + (t-s)\mathcal{L}_g,$$

*where* $\mathcal{L}_f = \big(\boldsymbol{f}(s,\boldsymbol{X}_s) - \boldsymbol{f}_{s,t}(\boldsymbol{X}_s)\big)^2$ *and* $\mathcal{L}_g = \big(\boldsymbol{g}(s) - \boldsymbol{g}_{s,t}\big)^2$.

*Proof.* Under the independence assumption, $\boldsymbol{f}_{s,t}(\boldsymbol{X}_s)$ and $\boldsymbol{g}_{s,t}$ are independent of $\boldsymbol{W}_{[s,t]}$, so

$$\boldsymbol{X}_t - \hat{\boldsymbol{X}}_t = \big(\boldsymbol{f}(s,\boldsymbol{X}_s) - \boldsymbol{f}_{s,t}(\boldsymbol{X}_s)\big)(t-s) + \big(\boldsymbol{g}(s) - \boldsymbol{g}_{s,t}\big)(\boldsymbol{W}_t - \boldsymbol{W}_s).$$

Expanding the square,

$$\begin{aligned}
\mathcal{L}_{\text{EM}} = (t-s)^2 &\big(\boldsymbol{f}(s,\boldsymbol{X}_s) - \boldsymbol{f}_{s,t}(\boldsymbol{X}_s)\big)^2 \\
&+ 2(t-s)\big(\boldsymbol{f}(s,\boldsymbol{X}_s) - \boldsymbol{f}_{s,t}(\boldsymbol{X}_s)\big)\big(\boldsymbol{g}(s) - \boldsymbol{g}_{s,t}\big)(\boldsymbol{W}_t - \boldsymbol{W}_s) \\
&+ \big(\boldsymbol{g}(s) - \boldsymbol{g}_{s,t}\big)^2(\boldsymbol{W}_t - \boldsymbol{W}_s)^2.
\end{aligned}$$

Taking $\mathbb{E}[\cdot \mid \boldsymbol{X}_s]$ and using $\mathbb{E}[\boldsymbol{W}_t - \boldsymbol{W}_s \mid \boldsymbol{X}_s] = 0$ and $\mathbb{E}[(\boldsymbol{W}_t - \boldsymbol{W}_s)^2 \mid \boldsymbol{X}_s] = t-s$,

$$\mathbb{E}[\mathcal{L}_{\text{EM}} \mid \boldsymbol{X}_s] = (t-s)^2\big(\boldsymbol{f}(s,\boldsymbol{X}_s) - \boldsymbol{f}_{s,t}(\boldsymbol{X}_s)\big)^2 + (t-s)\big(\boldsymbol{g}(s) - \boldsymbol{g}_{s,t}\big)^2.$$

Substituting for $\mathcal{L}_f, \mathcal{L}_g$ gives the result. □

### C.2 Diffusion SDEs

To apply Algorithm 1 to diffusion SDEs, we must derive the ground truth reverse diffusion SDE. Consider the variance preserving formulation,

$$\mathrm{d}\boldsymbol{X}_t = -\frac{1}{2}\beta_t \boldsymbol{X}_t \,\mathrm{d}t + \sqrt{\beta_t}\,\mathrm{d}\boldsymbol{W}_t.$$

The reverse diffusion SDE is given by

$$\mathrm{d}\boldsymbol{X}_t = [-\frac{1}{2}\beta_t \boldsymbol{X}_t - \beta_t \nabla_{\boldsymbol{X}_t} \log p(t, \boldsymbol{X}_t)]\,\mathrm{d}t + \sqrt{\beta_t}\,\mathrm{d}\boldsymbol{W}_t,$$

where $\nabla_{\boldsymbol{X}_t} \log p(t, \boldsymbol{X}_t)$ is the score.

Given a data sample $\boldsymbol{X}_1$, an expression for the score can be obtained. Specifically, we sample the forward process via $\boldsymbol{X}_t = \alpha_t \boldsymbol{X}_1 + \sigma_t \epsilon$, $\epsilon \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$, where

$$\alpha_t = \exp\left(-\tfrac{1}{2}\int_0^t \beta_s \, ds\right), \quad \sigma_t^2 = 1 - \alpha_t^2.$$

Then, the score is given by

$$\nabla_{\boldsymbol{X}_t} \log p(t, \boldsymbol{X}_t) = -\frac{\epsilon}{\sigma_t}.$$

We are therefore able to build the ground truth SDE required for training the stochastic flow map by

$$\mathrm{d}\boldsymbol{X}_t = [-\frac{1}{2}\beta_t \boldsymbol{X}_t + \beta_t \frac{\epsilon}{\sigma_t}]\,\mathrm{d}t + \sqrt{\beta_t}\,\mathrm{d}\boldsymbol{W}_t.$$

## D Extended related work

**Few-step models.** The training objectives for SSFMs with small step sizes are related to generator matching, *i.e.*, matching the drift and diffusion coefficients (Holderrieth, Havasi, et al. 2025). The semigroup loss is the stochastic analogue to the one used in deterministic flow maps studied by (Boffi et al. 2025; Frans et al. 2025). Consistency models (Song and Dhariwal 2024; Song, Dhariwal, et al. 2023) use a small Euler step which is related to our small stochastic Euler-Maruyama step; however, the actual loss itself is quite different as we predict the small jump rather than using consistency. Other methods (Boffi et al. 2025; Geng, Deng, et al. 2025; Geng, Lu, et al. 2025) train with losses which require time derivatives of the flow map, a requirement that does not extend naturally to the stochastic setting, as the Brownian motion is nowhere differentiable w.r.t. time.

**Few-step sampling of SDEs.** An active area of research has been accelerating inference with diffusion SDEs via more efficient numerical schemes (Blasingame and Liu 2026; Gonzalez et al. 2023; Zhang and Chen 2023), thereby decreasing the NFE to use these models; however, all these techniques are at inference time and use the same models. Recent work by Jiang et al. (2025) looks at using *partial signatures* of the Brownian motion to distill pre-trained diffusion SDEs into taking larger step sizes by learning a hyper-solver (Poli et al. 2020). This is significantly different from our work as it requires numerically integrating the entire SDE, *i.e.*, it does not have a simulation-free manner of training even for flow/diffusion models and relies on a pre-trained diffusion model teacher. SSFMs on the other hand enable scalable training of stochastic flow maps and can be trained without a pre-trained teacher model.

**Stochastic few-step models.** Recent work by Holderrieth, Chen, et al. (2026), Kiyohara et al. (2025), Passaro et al. (2026), and Potaptchik et al. (2026) have looked at weak approximations to the diffusion SDE. The later three works Holderrieth, Chen, et al. (2026), Passaro et al. (2026), and Potaptchik et al. (2026) learn an "inner" flow map to learn the transition kernel as we detail in the main paper.[3] Beyond providing a strong solution, SSFMs also work for arbitrary additive-noise SDEs, whereas existing weak approaches are typically formulated for specific diffusion model SDEs. The work of Kiyohara et al. (2025) also adopts an Euler-Maruyama type parametrization for each layer

[3] Passaro et al. (2026) discusses the nuances between these works in more detail.

within the context discrete normalizing flow (Rezende and Mohamed 2015) framework, *i.e.*, for a flow map $\boldsymbol{\Upsilon}^{\theta}_{s,t}(\boldsymbol{x}) = (\boldsymbol{f}^{\theta}_{L} \circ \boldsymbol{f}^{\theta}_{L-1} \circ \cdots \circ \boldsymbol{f}^{\theta}_{1})(\boldsymbol{x})$, each layer $\boldsymbol{f}_\ell$ is given as

$$\boldsymbol{f}_\ell(\boldsymbol{x}; s, t) = \boldsymbol{x} + (t-s)\boldsymbol{\mu}^{\theta}_{\ell}(\boldsymbol{x}; s, t) + \sqrt{t-s}\boldsymbol{\sigma}^{\theta}_{\ell}(\boldsymbol{x}; s, t) \odot \boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}). \tag{23}$$

This map based on the normalizing flow above is then trained to learn the transition kernel of an SDE $p_{t|s}(\boldsymbol{x}_t|\boldsymbol{x}_s)$ by minimizing both the forward and reverse Kullback-Leibler divergence (Kullback and Leibler 1951). This work is different from the other stream of weak stochastic flow maps as they learn via a parameterized normalizing flow instead of the standard flow map losses (Boffi et al. 2025; Geng, Lu, et al. 2025); the paper focuses on learning the solution maps for latent neural SDEs (Kidger et al. 2021; Li et al. 2020). Concurrent work to ours in Pan et al. (2026) also looks at learning the Itô map but uses a sinusoidal-basis Karhunen–Loéve expansion of the Brownian path (Karhunen 1950; Loève 1945). This expansion does not admit closed-form Chen relations, which forces the network to always consume a global expansion – significantly weakening the approximation.

# E Experimental details and further experiments

In the image and molecule generation experiments we take the number of polynomial coefficients $N = 3$. We use the Virtual Brownian Tree (VBT) implementation in Diffrax to generate the coefficients $\boldsymbol{I}^{(3)}_{s,t}$ (Jelinčič et al. 2024; Kidger 2022). It should be noted that the VBT implementation introduces a constant scaling factor on each coefficient.

## E.1 Non-linear SDE

We include additional informative plots for the non-linear SDE experiment that would not fit in the main text. In Figure 6 we show the polynomial approximation of Brownian motion for varying degree polynomials. In Figure 7 we show the SSFM predictions for many step sizes when trained with the $N = 4$ degree polynomial.

In Figure 8 we show the strong error obtained by the SSFM as a function of the polynomial degree. We compare the single increment model presented, $\boldsymbol{\Psi}_{s,t}(\boldsymbol{X}_s, \boldsymbol{I}^{N}_{s,t})$, to a double increment model where the SSFM is implemented as $\boldsymbol{\Psi}_{s,t}(\boldsymbol{X}_s, \boldsymbol{I}^{N}_{s,u}, \boldsymbol{I}^{N}_{u,t})$. This allows us to decouple the error introduced by integration and the error introduced by the Brownian polynomial approximation. Specifically, if the double increment model taking $n$ steps matches the error of the single increment model taking $2n$ steps, then the error is entirely explained by the polynomial approximation and not integration error. Vice versa, the error is integration limited.

We see that in the few-step regime the strong error is limited by the Brownian polynomial approximation, while in the many step regime the strong error is limited by integration. Interestingly, we see that the strong error is reduced most effectively by including the $(n+1)$-th degree coefficient rather than passing a second $n$-th degree coefficient on half the time interval. Note that this may not hold in general on problems where integration is more challenging.

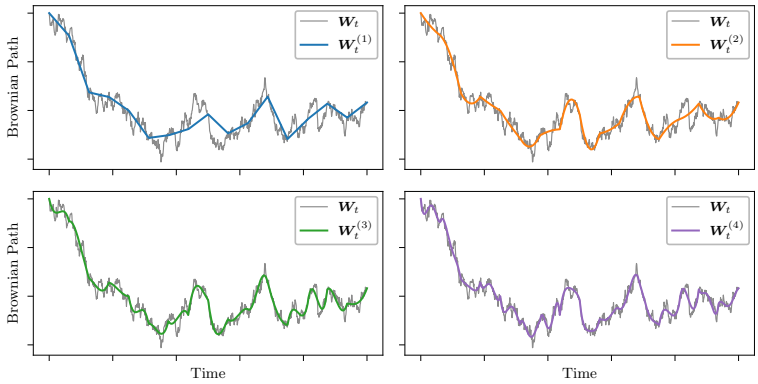


Figure 6: Brownian path $\boldsymbol{W}_t$ and associated polynomial approximations $\boldsymbol{W}^{(N)}_t$ over 16-steps.

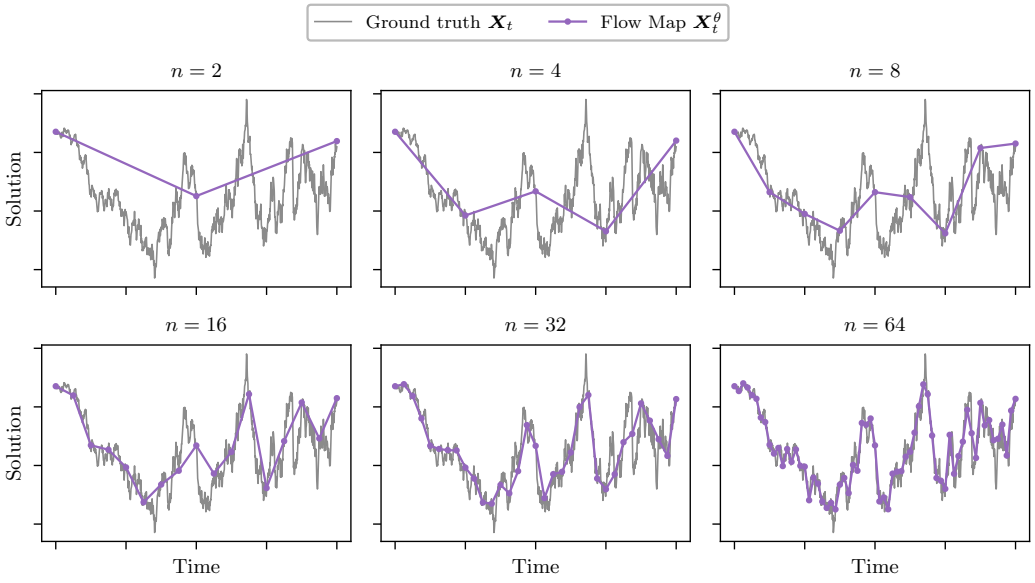


Figure 7: Ground truth SDE and SSFM prediction for many step counts.

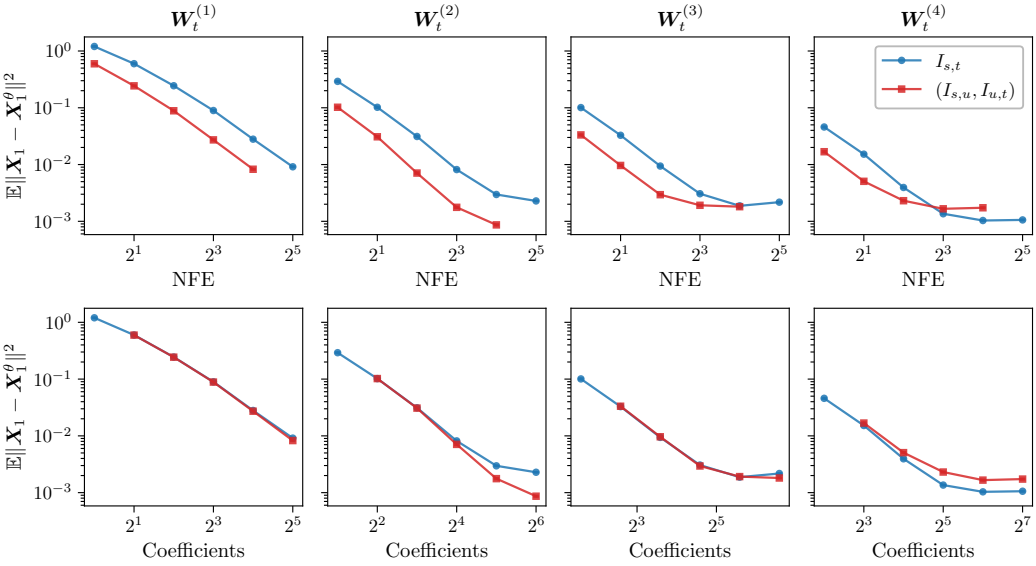


Figure 8: Strong error as a function of the polynomial degree for single $I_{s,t}$ and double $(I_{s,u}, I_{u,t})$ input coefficient models.

### E.2 Image generation

For the image generation experiments we setup the variance preserving SDE (see Appendix C.2) as the ground truth and follow Algorithm 1 to train the SSFM. The drift and diffusion networks are parameterized with the EDM2 architecture (Config C) (Karras et al. 2024) with hyperparameters given in Table 10.

For evaluation, we generate 50k samples with the SSFM and compute FID against the CIFAR-10 / CelebA datasets. Uniform step placement is used so that for $n$ steps, the step size is given by $(1 - t_\epsilon)/N$, where $t_\epsilon = 10^{-5}$. The CIFAR-10 inference speed on $1\times$ NVIDIA H100 is shown in Table 7.

We include additional experimental results for the image generation experiments below. Specifically, we measure the strong error and FID metrics as a function of the number of flow map steps and polynomial degree coefficients passed to the model.

**Strong Error.** The strong error obtained by the SSFM on CIFAR-10 dataset is reported in Table 8, We sample 8192 different initial conditions $\boldsymbol{X}_0$ and Brownian paths $\boldsymbol{W}_{[0,1]}$. The reference solution is obtained by the SDE solved via Euler-Maruyama (EM) with 4096 discretization steps over $[0, 1]$. For each $(\boldsymbol{X}_0, \boldsymbol{W}_{[0,1]})$, we compute the SSFM solution and compare to the EM reference (shared Brownian path). To verify that the strong solution is obtained, we also compare to SSFM solution conditioned on a different realization of the Brownian path (denoted Weak SFM).

**Polynomial Degree.** We measure the effect of the Brownian polynomial degree on the CIFAR-10 FID in Table 9. For degrees 1, 2, 3 and 4, the SSFM model was trained under otherwise identical settings and the 50k-sample FID computed. We see that at the effect is most pronounced in the few step regime, where at 2 flow map steps the degree-3/4 models obtain at least a $15\times$ improvement over the degree-1 model. In the many step regime, each model converges to a similar FID.

Table 7: CIFAR-10 inference speed and throughput across step counts for generating 50k images.

| NFE | Wall-clock (ms) | Per sample ($\mu$s) | Samples/s |
|---|---|---|---|
| 2 | 27,920 | 558 | 1,790 |
| 4 | 51,564 | 1,031 | 969 |
| 8 | 104,150 | 2,083 | 480 |
| 16 | 211,275 | 4,225 | 236 |

Table 8: Strong error on the CIFAR-10 dataset.

| Method | 2 | 4 | 8 | 16 | 32 | 64 |
|---|---|---|---|---|---|---|
| Euler-Maruyama | 6.341 | 2.385 | 0.958 | 0.335 | 0.105 | 0.031 |
| Weak SFM | 1.9122 | 1.9447 | 1.9415 | 1.9341 | 1.9264 | 1.9177 |
| **SSFM** | **0.1034** | **0.0704** | **0.0599** | **0.0509** | **0.0412** | **0.0269** |

Table 9: Polynomial degree ablation on the CIFAR-10 dataset measured in terms of FID.

| NFE | $\boldsymbol{W}^{(1)}$ | $\boldsymbol{W}^{(2)}$ | $\boldsymbol{W}^{(3)}$ | $\boldsymbol{W}^{(4)}$ |
|---|---|---|---|---|
| 2 | 93.87 | 9.07 | **4.93** | 6.02 |
| 4 | 12.28 | 4.50 | **3.49** | 5.67 |
| 8 | 5.71 | 3.81 | **3.29** | 5.76 |
| 16 | 3.63 | 3.62 | **3.35** | 4.83 |

Table 10: Image generation hyperparameters.

| Hyperparameter | CIFAR-10 | CelebA-64 |
|---|---|---|
| *Drift network (EDM2 U-Net)* | | |
| Base channels | 128 | 128 |
| Channel multipliers | $(2,\ 2,\ 2)$ | $(1, 2, 3, 4)$ |
| Attention resolutions | $16 \times 16$ | $16 \times 16, 8 \times 8$ |
| Attention head dimension | 64 | 64 |
| GroupNorm groups | 8 | 8 |
| Dropout | 0.13 | 0.13 |
| *Diffusion network (EDM2 U-Net)* | | |
| Base channels | 64 | 64 |
| Channel multipliers | $(2,\ 2,\ 2)$ | $(1, 2, 3, 4)$ |
| Attention resolutions | $16 \times 16$ | $16 \times 16, 8 \times 8$ |
| Attention head dimension | 64 | 64 |
| GroupNorm groups | 8 | 8 |
| Dropout | 0 | 0 |
| *Loss* | | |
| Step size split $\Delta t$ | 0.01 | 0.01 |
| Max distillation step $h_{\max}$ | 0.52 | 0.52 |
| Batch split $\eta$ | 0.75 | 0.75 |
| *Optimization* | | |
| Optimizer | Adam ($\beta_2 = 0.99$) | Adam ($\beta_2 = 0.99$) |
| Peak learning rate | $10^{-3}$ | $10^{-3}$ |
| Min learning rate | $10^{-5}$ | $10^{-5}$ |
| LR schedule | warmup + cosine decay | warmup + cosine decay |
| Warmup steps | 5,000 | 5,000 |
| Gradient clip (global norm) | 1.0 | 1.0 |
| Batch size | 512 | 256 |
| Training steps | 400,000 | 800,000 |
| EMA decay | 0.999 | 0.999 |

### E.3 Molecular systems: ALDP

The alanine dipeptide dataset follows (Plainer et al. 2025) with 50k samples from a molecular dynamics simulation in implicit solvent (Köhler et al. 2021), coarse grained to five atoms [C, N, CA, C, N]. It is available from the *ScoreMD* repository accompanying (Plainer et al. 2025).

The evaluation metrics calculated are the potential of mean force (PMF) squared error and the Jensen-Shannon (JS) divergence. These metrics both compare the difference in the equilibrium free energy surfaces of the ground truth system and the model prediction.

A graph transformer architecture is used with hyperparameters listed in Table 11. The variance preserving SDE is used as the ground truth to construct the SSFM following Algorithm 1.

The diffusion baseline models were obtained from the *ScoreMD* repository and results collected via the evaluation code provided. The SSFM model evaluation was also computed from the *ScoreMD* code to ensure a fair comparison. The inference metrics for generating the 120k samples on $1\times$ NVIDIA A6000 is shown in Table 13.

**Strong error and polynomial degree.** We present additional ablation results for the polynomial degree's effect on the strong error in Table 12. The strong error measurement setup is described in Appendix E.2. We see that for ALDP the strong error obtained by the SSFM is significantly decreased in comparison to the Euler-Maruyama solver and the weak SFM solution. The effect of the polynomial degree is most pronounced in the few-step regime, where the degree-3 model achieves a $14\times$ lower strong error than the degree-1 model. In the many-step regime, the SSFM obtains a comparable error for all degrees and to direct Euler-Maruyama integration.

Table 11: ALDP generation hyperparameters.

| Hyperparameter | ALDP |
|---|---|
| *Drift/Diffusion network (graph-transformer)* | |
| Hidden dim | 96 |
| Transformer blocks | 3/2 |
| Attention heads | 8 |
| Head dim | 64 |
| Feed-forward multiplier | 4 |
| Time-embedding dim | 64 |
| Uncertainty MLP Fourier dim | 64 |
| *Loss* | |
| Step size split $\Delta t$ | $10^{-3}$ |
| Max distillation step $h_{\max}$ | 0.52 |
| Batch split $\eta$ | 0.75 |
| *Optimization* | |
| Optimizer | AdamW |
| Peak learning rate | $10^{-3}$ |
| Min learning rate | $10^{-5}$ |
| LR schedule | warmup + cosine decay |
| Warmup steps | 1,000 |
| Gradient clip (global norm) | 10 |
| Batch size | 1,024 |
| Training steps | 400,000 |
| EMA decay | 0.999 |

Table 12: Strong error on the ALDP dataset as a function of the number of Brownian polynomial coefficients.

| | NFE | | | | | |
|---|---|---|---|---|---|---|
| Method | 2 | 4 | 8 | 16 | 32 | 64 |
| Euler-Maruyama | 791.5 | 60.40 | 8.976 | 2.110 | 0.4518 | 0.09549 |
| Weak SFM | 62.30 | 62.50 | 62.79 | 62.85 | 62.86 | 62.87 |
| SSFM, $\mathbf{W}^{(1)}$ | 1.272 | 0.5101 | 0.1890 | 0.08873 | 0.03807 | 0.01841 |
| SSFM, $\mathbf{W}^{(2)}$ | 0.2223 | 0.07388 | 0.04235 | 0.02120 | **0.01207** | **0.007882** |
| SSFM, $\mathbf{W}^{(3)}$ | **0.09068** | **0.05849** | **0.02456** | **0.01772** | 0.01242 | 0.008751 |

Table 13: Alanine-Dipeptide inference speed and throughput across step counts for generating 120k samples.

| NFE | Wall-clock (ms) | Per sample ($\mu$s) | Samples/s |
|---|---|---|---|
| 2 | 27.7 | 6.9 | 144,584 |
| 4 | 54.8 | 13.7 | 72,926 |
| 10 | 135.1 | 33.8 | 29,616 |
| 20 | 268.9 | 67.2 | 14,875 |
| 100 | 1,346.3 | 336.6 | 2,971 |

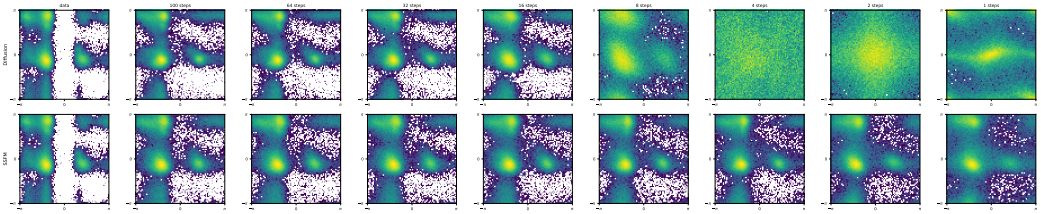

Figure 9: Pooled Chignolin Ramachandran plots. All backbone $\phi/\psi$ torsions are superimposed for ground-truth data (*left-most column*) and generated samples from the diffusion baseline (*top*) and SSFM (*bottom*) across sampling step counts: [100, 64, 32, 16, 8, 4, 2, 1].

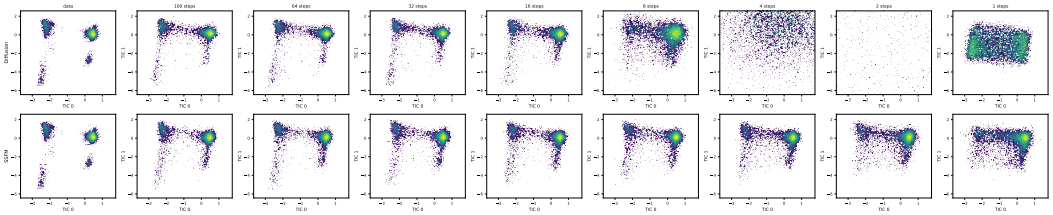

Figure 10: Chignolin tICA plots. Ground-truth data is shown in the *left-most column*. The two primary tICA components and projected samples are shown from the diffusion baseline (*top*) and SSFM (*bottom*) across sampling step counts: [100, 64, 32, 16, 8, 4, 2, 1].

### E.4 Molecular systems: Chignolin

Chignolin, a 10-residue mini-protein with sequence `GYDPETGTWG`, was modelled in full atomic detail (138 atoms, all hydrogens included). The reference ensemble is drawn from a still-internal addendum of the *many-peptides* molecular-dynamics dataset: long all-atom MD trajectories that have been subsampled to $N = 10{,}000$ approximately decorrelated conformers, which constitute the entire reference set used below. Coordinates are stored in nanometres and are used without re-centering.

We report two complementary Wasserstein distances, both estimated against the reference set with 10,000 samples. The first is the backbone torsion torus-Wasserstein distance $\mathbb{T}$-$\mathrm{W}_2$. For each conformer we extract the 9 backbone $(\phi, \psi)$ torsion pairs.

The second metric is a tICA-$\mathrm{W}_2$. We project conformers into the dataset's precomputed 2-dimensional time-lagged independent component (tICA) coordinates – built from $\mathrm{C}/\mathrm{N}/\mathrm{S}$ pairwise distances together with backbone $\phi/\psi/\omega$ sine/cosine features (1765 features reduced to the two slowest components) – and compute the exact 2D $\mathrm{W}_2$ between generated and reference tICA densities.

We compare a strong stochastic flow map (SSFM) against a diffusion baseline, both trained on identical data with matched optimization. The SSFM instantiates Algorithm 1 over a variance-preserving (VP) SDE. The drift and diffusion networks are parameterised as Diffusion Transformer (DiT) networks (see Table 14 for hyperparameters). The Brownian polynomial degree is $N = 3$.

The baseline is an x-prediction diffusion model: a single DiT trained to denoise under the same VP SDE and sampled with ancestral VP reverse steps. All transformer backbones are DiTs with adaLN-zero modulation, sinusoidal time embeddings, and per-atom learned positional embeddings.

Additional plotted results showing the pooled Ramachandran and tICA projection are given in Figures 9 and 10 along with the full step count Wasserstein metrics in Table 15. The strong error obtained by the SSFM is also shown in Table 16 as a function of the number of steps and polynomial degree coefficients; see Appendix E.2 for a description of the strong error measurement.

### E.5 Path-dependent committer times

It is possible to efficiently calculate the committer times (along diffusion time) for the molecular systems using the SSFM model. The committer time is defined as the minimum time $\tau$ when the probability of generating a specific state $b_i$ exceeds a threshold, denoted $p(\boldsymbol{X}_1 \in b_i | \boldsymbol{X}_t) > \gamma$. We

Table 14: Chignolin generation hyperparameters.

| | SSFM | Diffusion |
|---|---|---|
| *Data* | | |
| Sequence | GYDPETGTWG | |
| Atoms | 138 | |
| Reference frames | 10,000 | |
| *Drift/Diffusion network (DiT)* | | |
| Hidden size | 192 | 192 |
| Transformer blocks | $6/2$ | 6 |
| Attention heads | 6 | 6 |
| MLP ratio | 4 | 4 |
| Time Fourier dim | 256 | 256 |
| Input channels | 12 | 3 |
| Dropout | 0.0 | 0.0 |
| *Loss* | | |
| Step size split$\Delta t$ | $10^{-3}$ | — |
| Max distillation step $h_{\max}$ | 1.0 | — |
| Batch split $\eta$ | 0.75 | — |
| *Optimization* | | |
| Optimizer | AdamW | AdamW |
| Peak learning rate | $2 \times 10^{-4}$ | $2 \times 10^{-4}$ |
| Min learning rate | $10^{-5}$ | $10^{-5}$ |
| LR schedule | warmup + cosine decay | |
| Warmup steps | 5000 | 5000 |
| Weight decay | $10^{-4}$ | $10^{-4}$ |
| Gradient clip (global norm) | 1.0 | 1.0 |
| Batch size | 64 | 64 |
| Training steps | $5 \times 10^5$ | $5 \times 10^5$ |
| EMA decay | 0.999 | — |
| Parameters | 5.71M | 4.19M |

Table 15: Chignolin Wasserstein metrics (↓) across all step counts.

| | | NFE | | | |
|---|---|---|---|---|---|
| Metric | Method | 1 | 2 | 4 | 8 |
| T-$W_2$ | Diffusion | $4.69_{\pm 0.02}$ | $5.88_{\pm 0.01}$ | $5.58_{\pm 0.01}$ | $2.71_{\pm 0.01}$ |
| | **SSFM** | $\mathbf{2.41_{\pm 0.03}}$ | $\mathbf{2.03_{\pm 0.03}}$ | $\mathbf{1.79_{\pm 0.02}}$ | $\mathbf{1.74_{\pm 0.03}}$ |
| tICA-$W_2$ | Diffusion | $1.99_{\pm 0.03}$ | $19.29_{\pm 0.14}$ | $3.22_{\pm 0.03}$ | $0.75_{\pm 0.01}$ |
| | **SSFM** | $\mathbf{0.80_{\pm 0.01}}$ | $\mathbf{0.51_{\pm 0.01}}$ | $\mathbf{0.41_{\pm 0.03}}$ | $\mathbf{0.38_{\pm 0.03}}$ |
| | | NFE | | | |
| Metric | Method | 16 | 32 | 64 | 100 |
| T-$W_2$ | Diffusion | $1.78_{\pm 0.01}$ | $\mathbf{1.68_{\pm 0.03}}$ | $\mathbf{1.65_{\pm 0.02}}$ | $\mathbf{1.64_{\pm 0.02}}$ |
| | **SSFM** | $\mathbf{1.71_{\pm 0.02}}$ | $1.71_{\pm 0.01}$ | $1.67_{\pm 0.02}$ | $1.66_{\pm 0.02}$ |
| tICA-$W_2$ | Diffusion | $\mathbf{0.31_{\pm 0.01}}$ | $\mathbf{0.26_{\pm 0.02}}$ | $\mathbf{0.27_{\pm 0.04}}$ | $\mathbf{0.27_{\pm 0.03}}$ |
| | **SSFM** | $0.36_{\pm 0.03}$ | $0.33_{\pm 0.03}$ | $0.32_{\pm 0.03}$ | $0.32_{\pm 0.04}$ |

Table 16: Strong error on the Chignolin dataset as a function of the number of Brownian polynomial coefficients.

| | NFE | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Method | 1 | 2 | 4 | 8 | 16 | 32 | 64 | 128 |
| Euler-Maruyama | 10.08 | 4.724 | 2.055 | 1.010 | 0.5316 | 0.2959 | 0.1671 | 0.09950 |
| Weak SFM | 1.302 | 1.406 | 1.417 | 1.418 | 1.418 | 1.418 | 1.418 | 1.418 |
| SSFM, $\mathbf{W}^{(1)}$ | 0.7529 | 0.6915 | 0.5410 | 0.4356 | 0.3132 | 0.2184 | 0.1482 | **0.09829** |
| SSFM, $\mathbf{W}^{(2)}$ | 0.7033 | 0.5443 | 0.3662 | 0.2348 | **0.1724** | **0.1444** | **0.1264** | 0.1067 |
| SSFM, $\mathbf{W}^{(3)}$ | **0.6077** | **0.4086** | **0.2752** | **0.2190** | 0.1737 | 0.1478 | 0.1325 | 0.1108 |

investigate the committer times for the Alanine Dipeptide backbone states $b_i \in \{A, B, G\}$ where each state is defined by backbone angles

$$A = \{(\phi, \psi) : \phi < 0,\ -2.1 < \psi < 0.7\},$$
$$G = \{(\phi, \psi) : \phi > 0\},$$
$$B = [-\pi, \pi)^2 \setminus (A \cup G).$$

Computing this time $\tau$ with the SSFM can be achieved efficiently via a binary search algorithm. For a fixed Brownian path, we follow the algorithm below starting from $s = 0, t = \Delta t$ for some initial step size $\Delta t$:

1. Step with $\boldsymbol{X}_t = \boldsymbol{\Psi}_{s,t}(\boldsymbol{X}_s, \boldsymbol{W}_{[s,t]})$
2. Compute $p(\boldsymbol{X}_1 \in b_i | \boldsymbol{X}_t)$ via $M$ lookaheads $\boldsymbol{X}_1 = \boldsymbol{\Psi}_{t,1}(\boldsymbol{X}_t, \boldsymbol{W}_{[t,1]})$
3. If $p > \gamma$, set $t = t - \Delta t/2$
4. Else, set $s = t$ and $t = t + \Delta t$
5. Repeat from 1 until convergence to finest time partition

This algorithm has runtime complexity $O(M \log N)$, where $N$ is the number points in the finest time partition and $M$ is the number of lookaheads. A weak stochastic flow map displays $O(MN)$ complexity, since every point in the finest partition must be traversed due to the absence of a pathwise-consistent map.

Once these committer times/states $(t, \boldsymbol{X}_t)$ have been selected by the search algorithm, we can use $(t, \boldsymbol{X}_t)$ as initial conditions for generation. This allows us to efficiently select a mode that we wish to sample from. For each mode $b_i$ we use its committer state $(t, \boldsymbol{X}_t)$ and sample 4096 Brownian paths on $[t, 1]$. Then, we generate samples by stepping with the SSFM over $[t, 1]$.

### E.6 Hardware

All experiments were run on either one/two NVIDIA RTX A6000 GPUs or one/two NVIDIA H100 GPUs.

### E.7 Repositories

We made use of the following repositories and resources:

1. `patrick-kidger/diffrax` (for VBT)
2. `patrick-kidger/equinox` (for neural networks in JAX)
3. `Lightning-AI/torchmetrics` (for FID)
4. `noegroup/ScoreMD` (for Alanine-Dipeptide dataset and diffusion baselines)

## F Broader impacts

We propose a framework for few-step integration of additive-noise SDEs, with demonstrated applications in image generation and molecular equilibrium sampling. The primary positive impact

of this work is the acceleration of molecular sampling relevant to ai4science applications which employ diffusion models. Improvements to image generation efficiency reduce inference-time energy consumption. There is a risk that improved image generation could be misused to produce harmful synthetic media; however, as SSFMs represent an efficiency improvement, we do not believe this work introduces risks beyond those already present in deployed diffusion models.

## NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The theoretical claims introduced in the abstract and introduction are verified in Section 3 and the experimental claims are empirically shown in Section 4 with additional details for the proofs and experimental details provided in the appendices.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We address the limitations in the "Limitations & future work" section (page 10).

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: The theoretical results from the main paper are proved with full assumptions stated in Appendices B and C.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide all the experimental details in Appendix E. Additionally, the random seeds we used are fixed in the configuration files of our code.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We provide open access to our code and data with sufficient instructions to reproduce the main experimental results in the readme. The link to our code is supplied in the abstract of this paper.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We provide all the experimental details in Appendix E.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The main experimental results contain error bars obtained via three training repeats with different seeds. The CelebA experiment was run only once due to the heavy computational demands.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: See Appendix E.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We conducted research conforming in every aspect with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss the broader impacts of this work in Appendix F.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.

- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The models we trained do not have a high-risk for misuse.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We provide details on the datasets and code used in Section 4 and Appendix E.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: Our code contains documentation in the main README file of our code.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: We did not perform crowdsourcing.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: Our research only used publicly available datasets and as such IRB approvals were not required.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: We do not use LLMs in the core method development or as non-standard components.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.