NEURAL INFORMATION PROCESSING SYSTEMS

IMPERIAL

Canon

Neural Stochastic Flows: Solver-Free Modelling and Inference for SDE Solutions

Naoki Kiyohara[1,2], Edward Johns[1], Yingzhen Li[1]

[1]Imperial College London [2]Canon Inc.

Project page

Stochastic Differential Equations (SDEs) are Everywhere

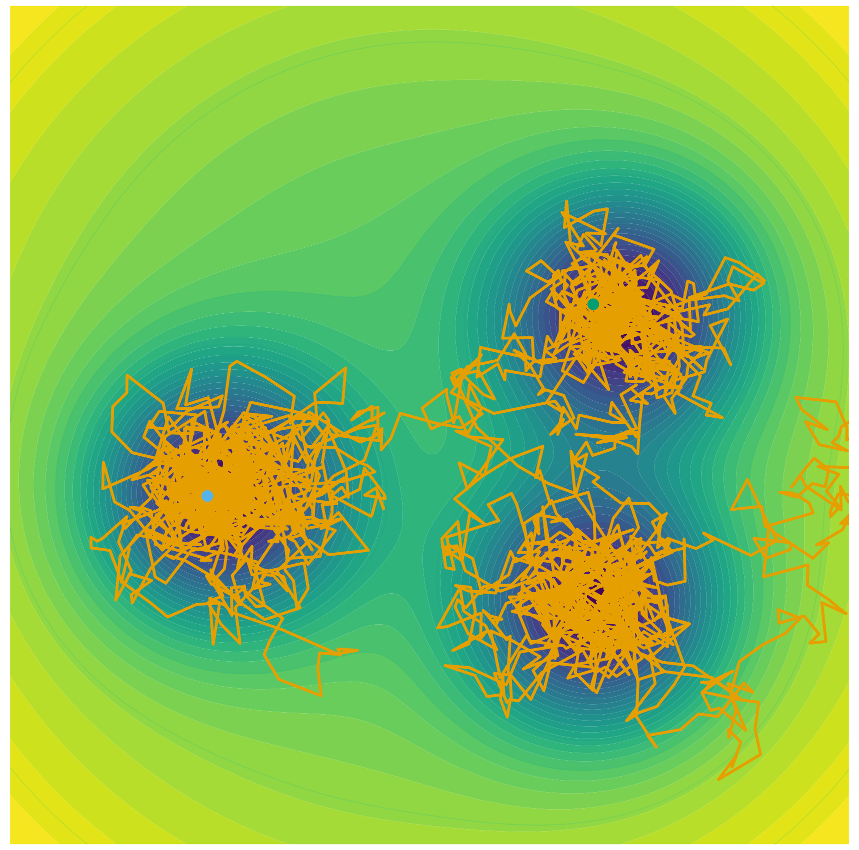
$$\text{Itô SDE:} \qquad dx_t = \boxed{\mu\left(x_t, t\right) dt} + \boxed{\sigma\left(x_t, t\right) dW_t,} \qquad x_0 \sim p_0$$

Drift term (deterministic)

Diffusion term (stochastic)

Finance

Physics



Generative Modelling

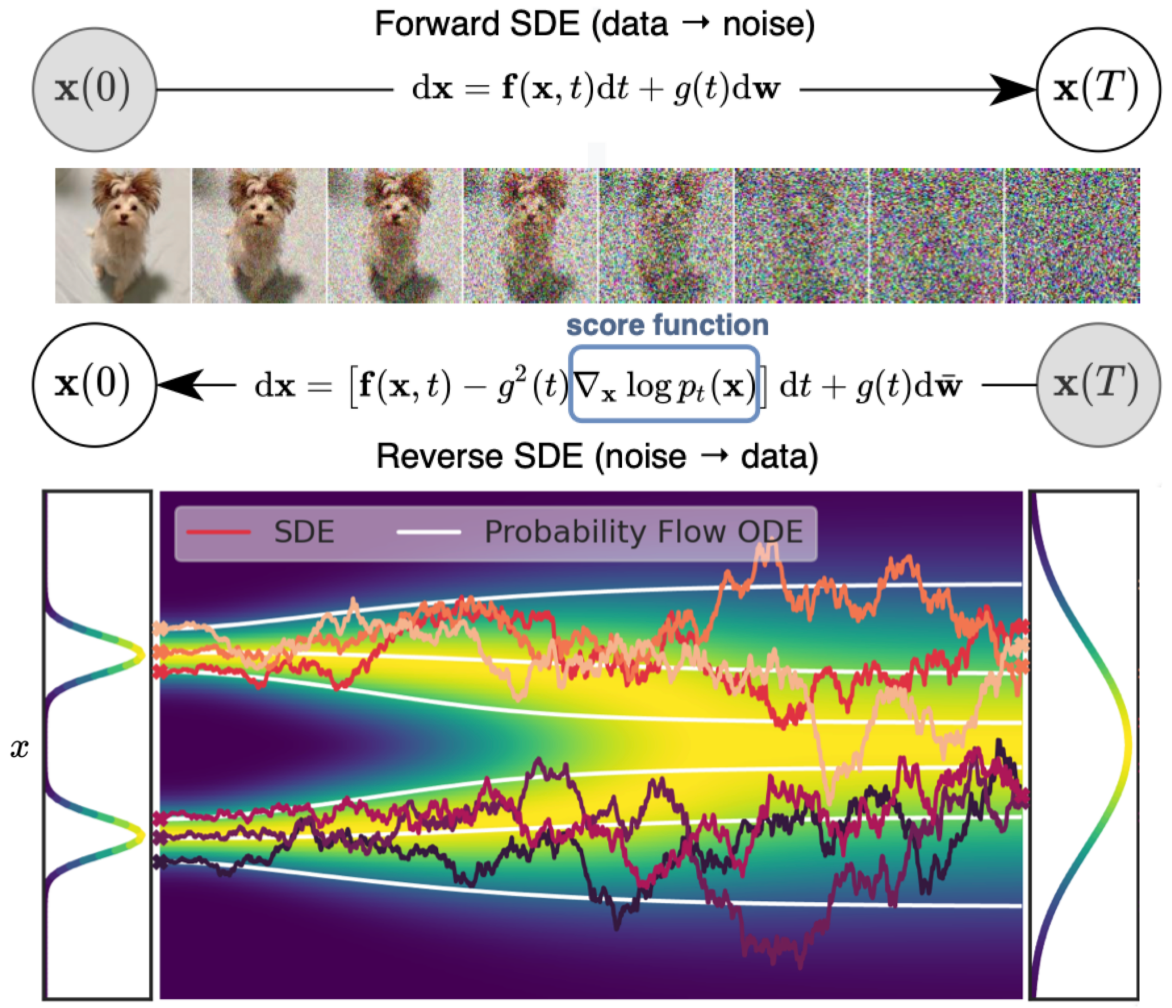


Figure adapted from [47]

Neural SDEs

Modelling SDE Terms using Neural Networks

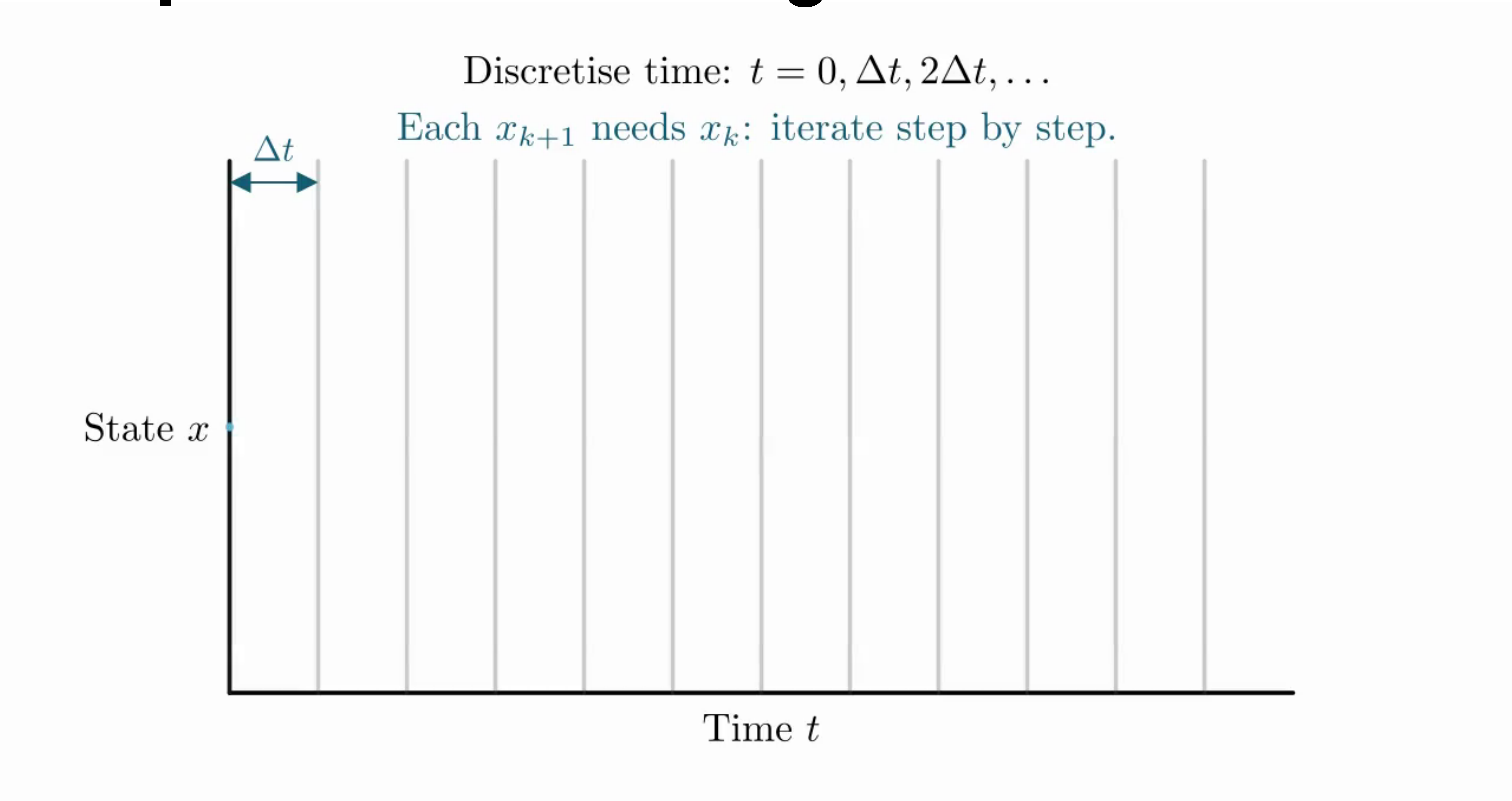
$$\text{Itô SDE:} \qquad dx_t = \boxed{\mu\left(x_t, t\right)} dt + \boxed{\sigma\left(x_t, t\right)} dW_t, \qquad x_0 \sim p_0$$

Modelled by neural networks

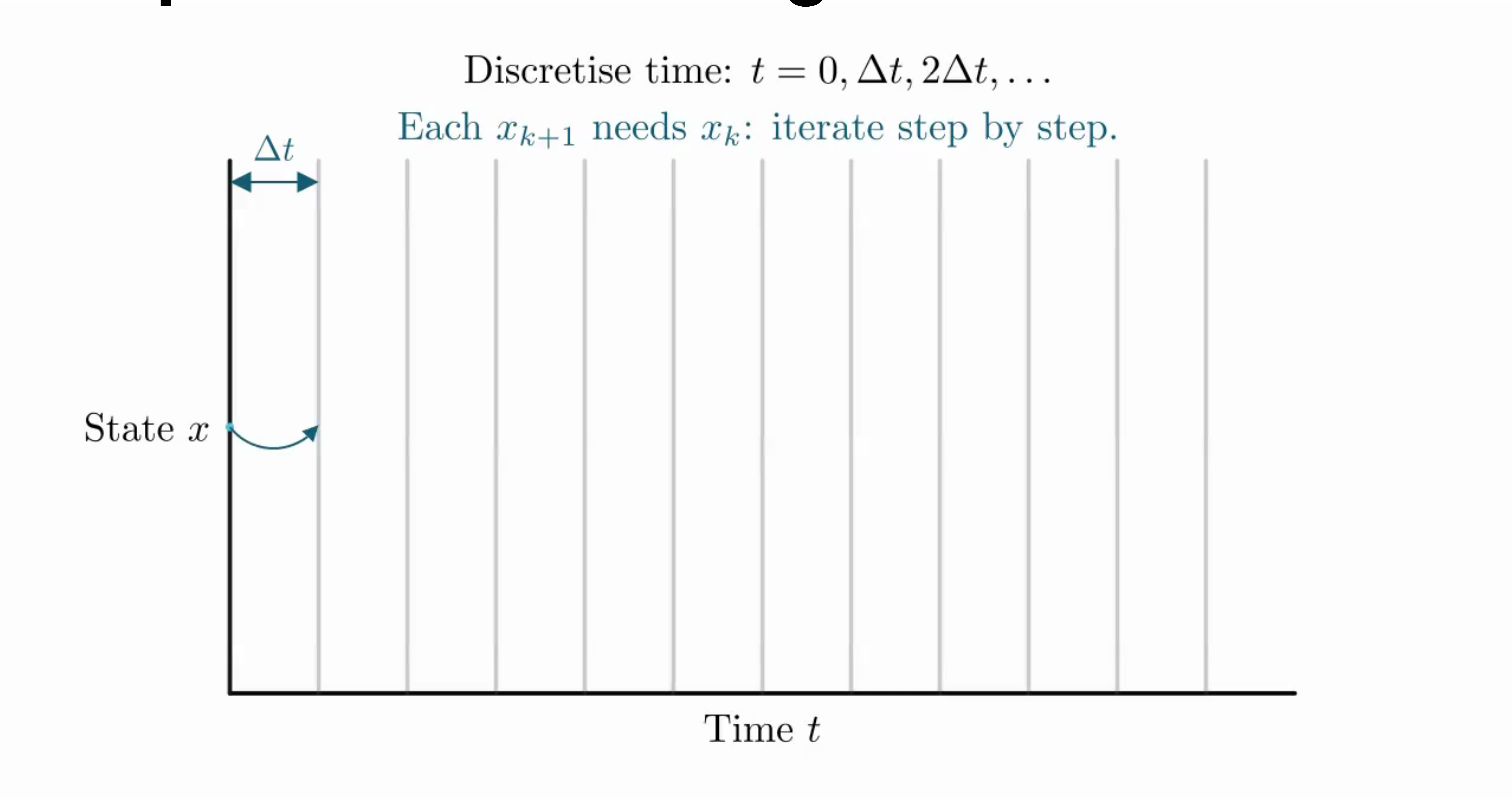
$$x_{t_{k+1}} = x_{t_k} + \mu\left(x_{t_k}, t_k\right) \Delta t + \sigma\left(x_{t_k}, t_k\right) \Delta W_k$$

$$t_k = k\Delta t, \qquad \Delta W_k = W_{t_{k+1}} - W_{t_k} \sim \mathcal{N}(0, \Delta t)$$

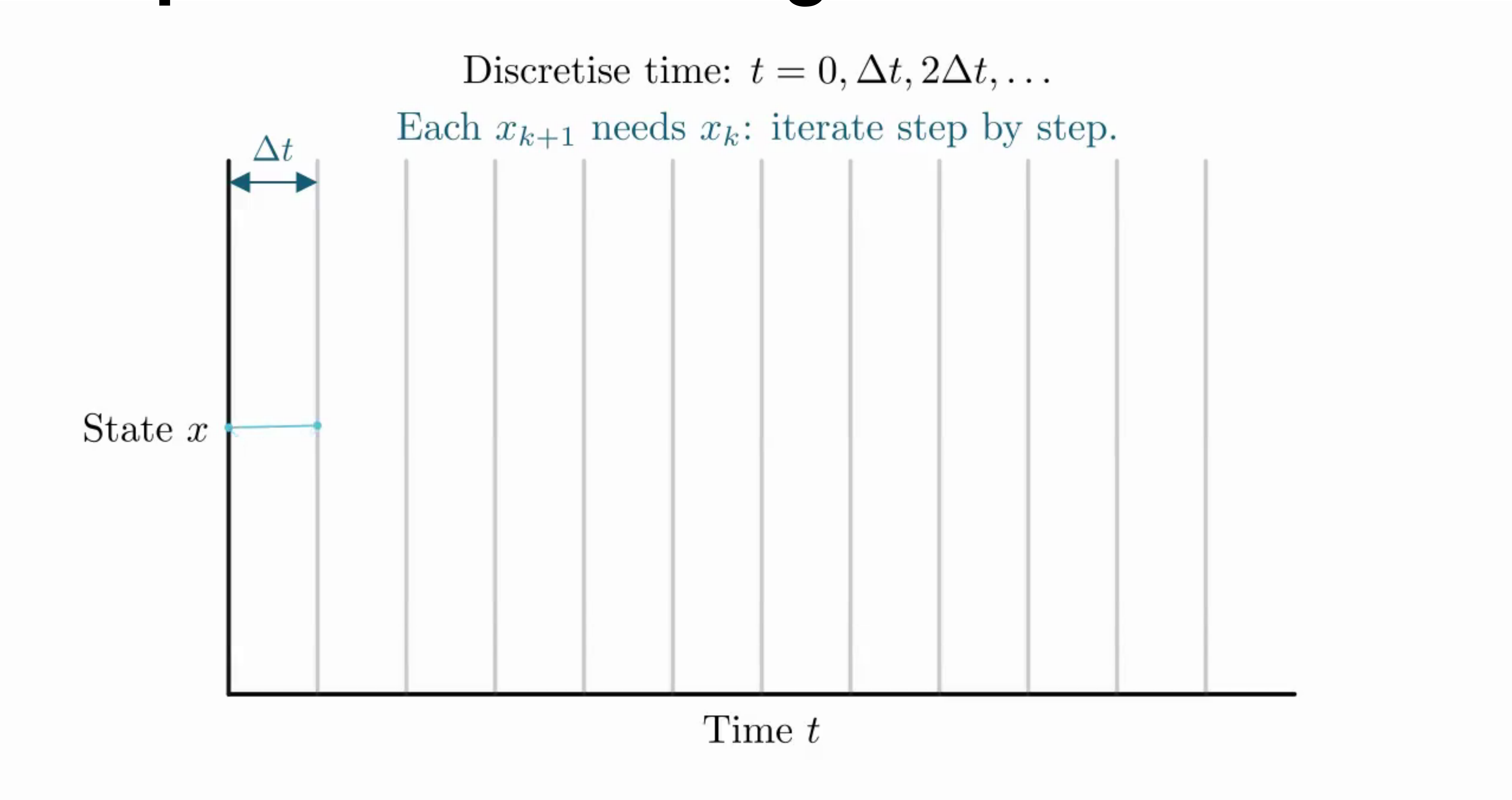
Computational Challenges of Neural SDEs



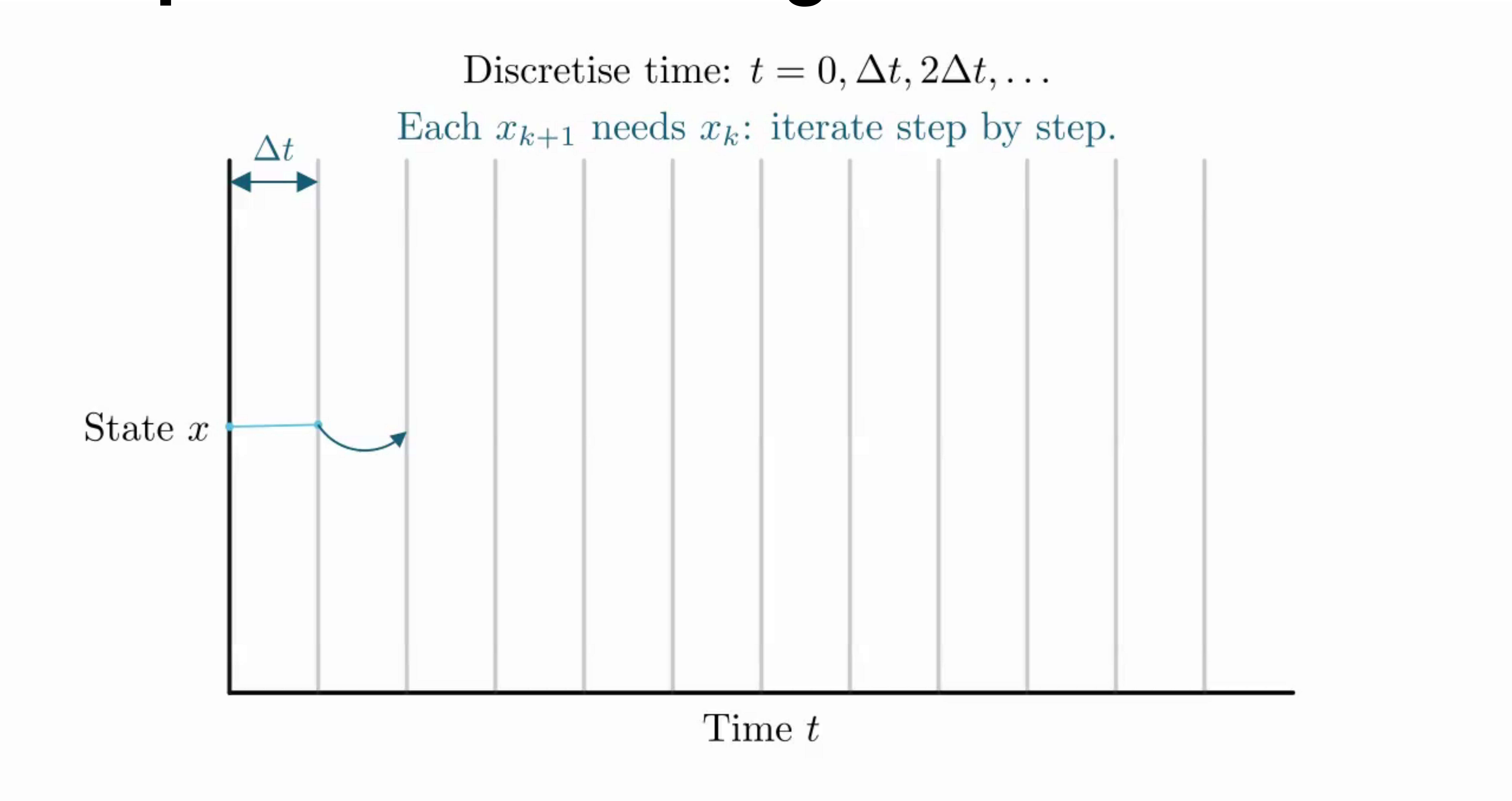
Computational Challenges of Neural SDEs



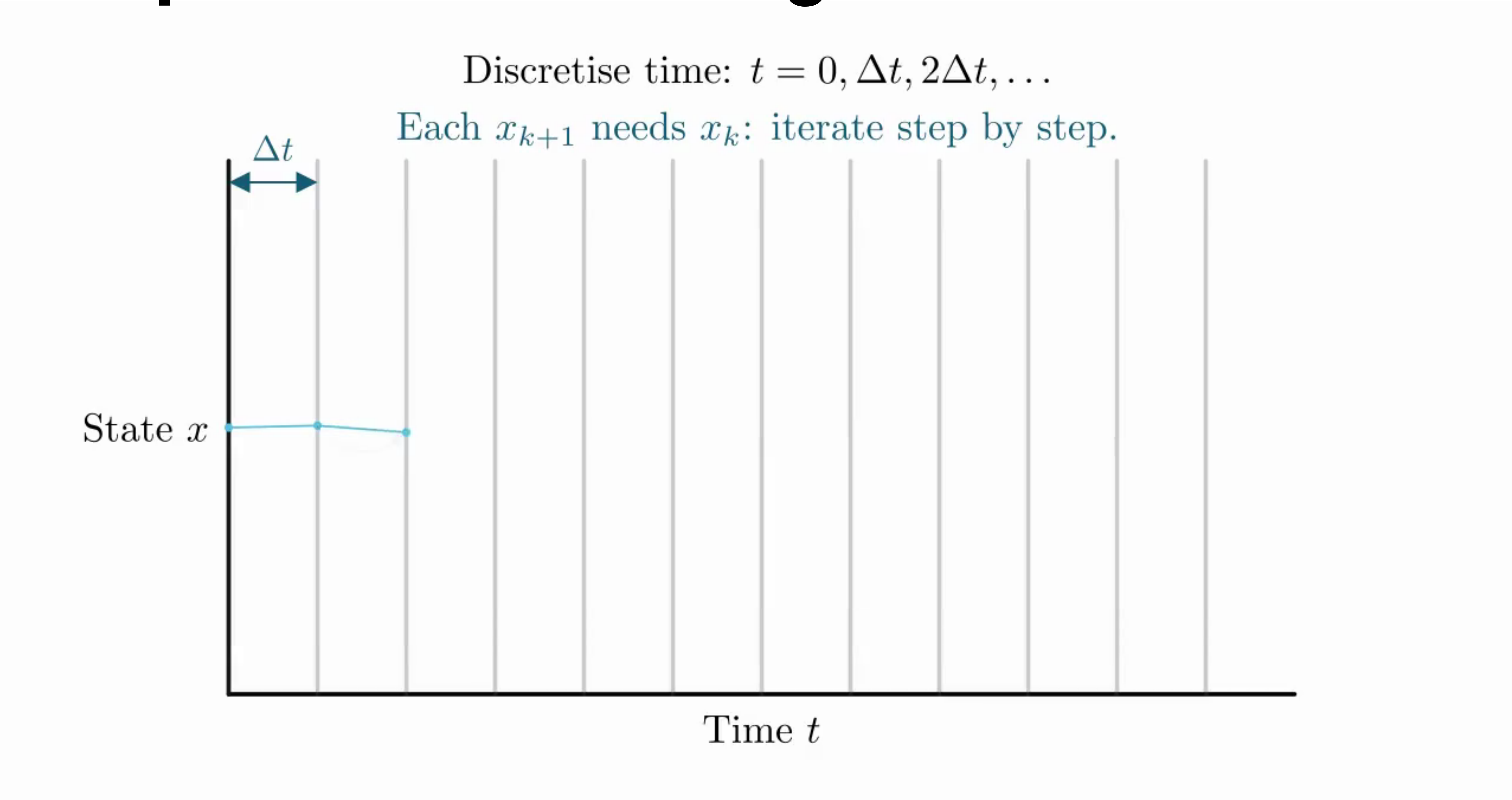
Computational Challenges of Neural SDEs



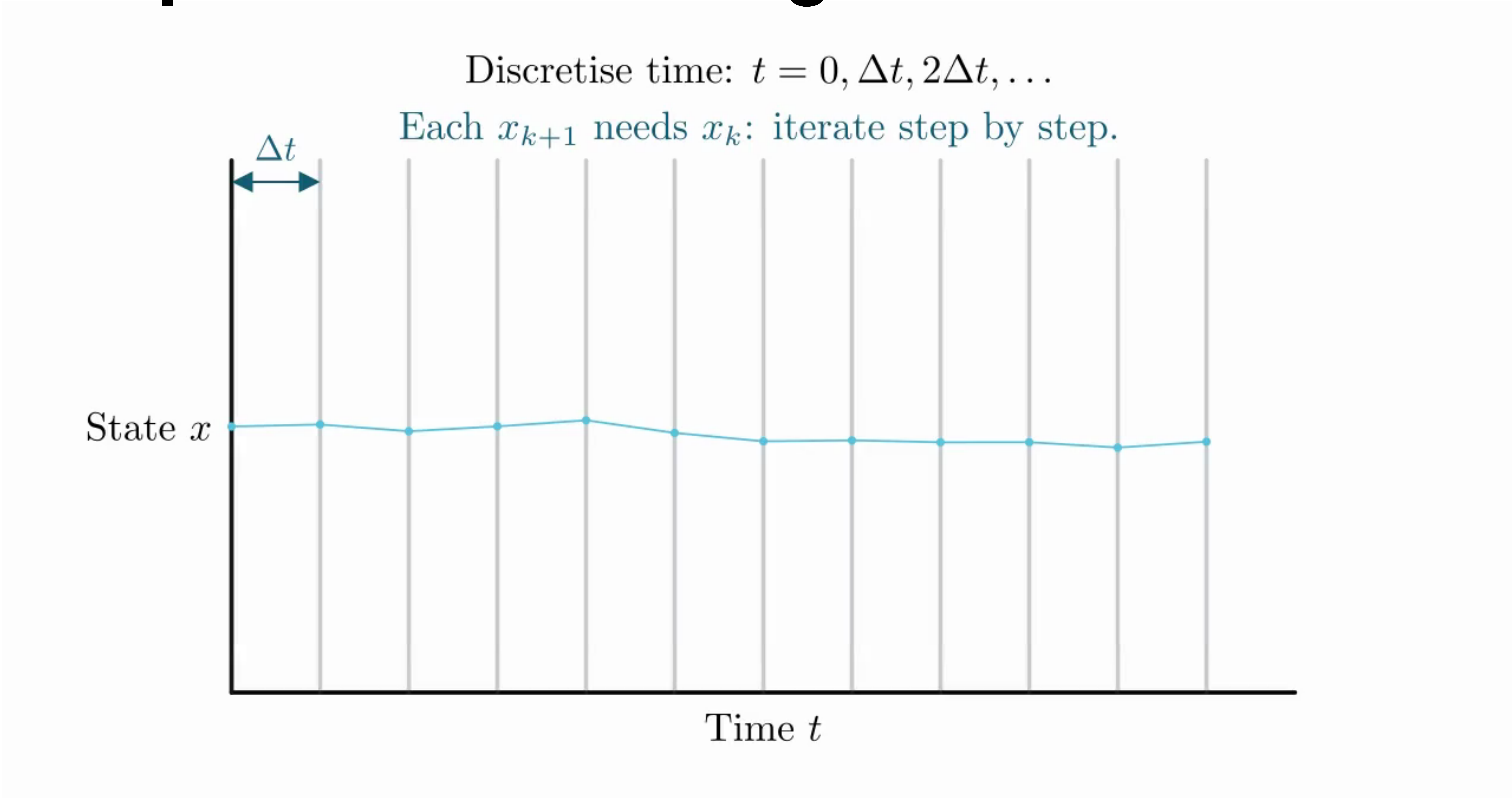
Computational Challenges of Neural SDEs



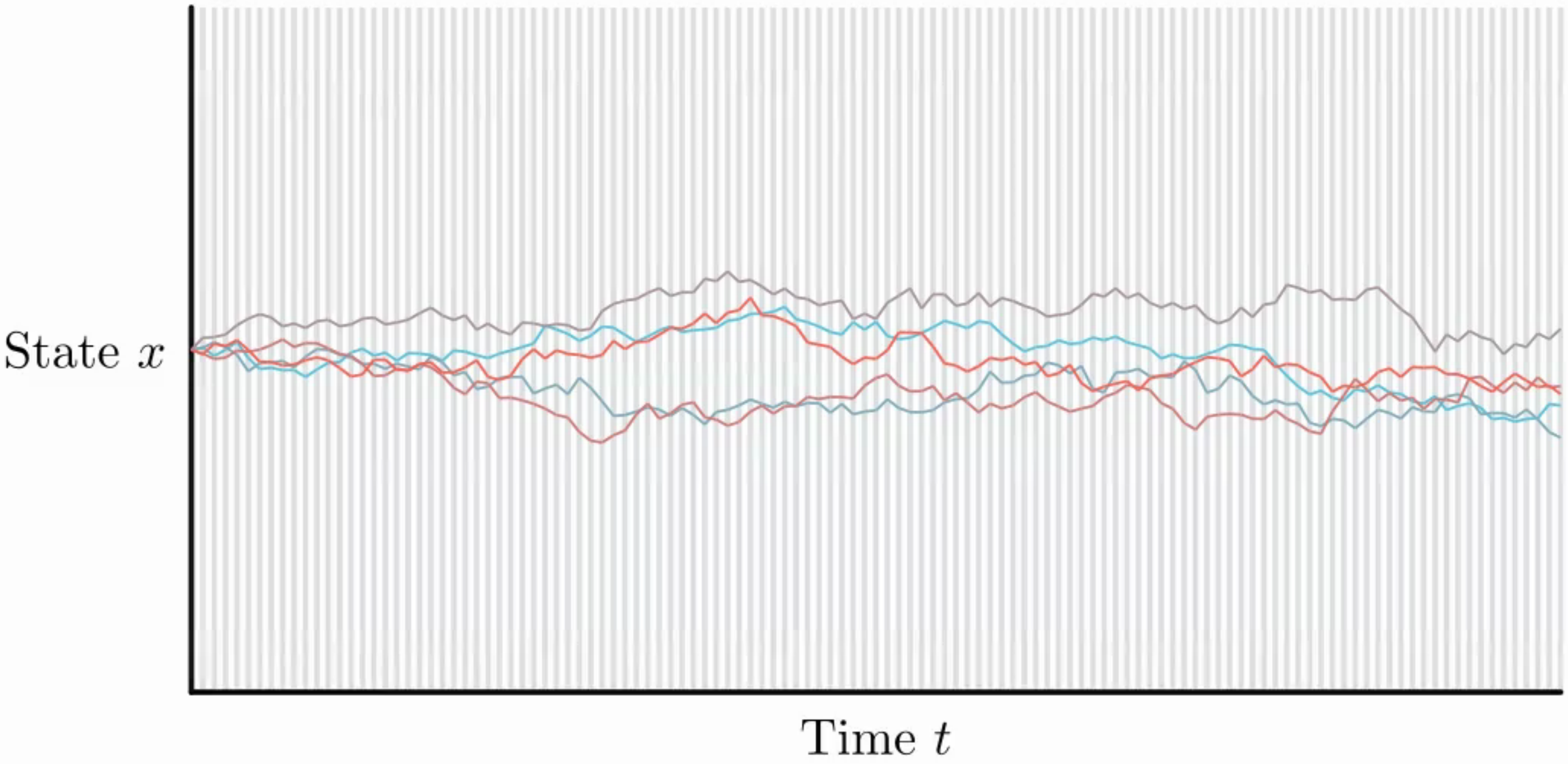
Computational Challenges of Neural SDEs



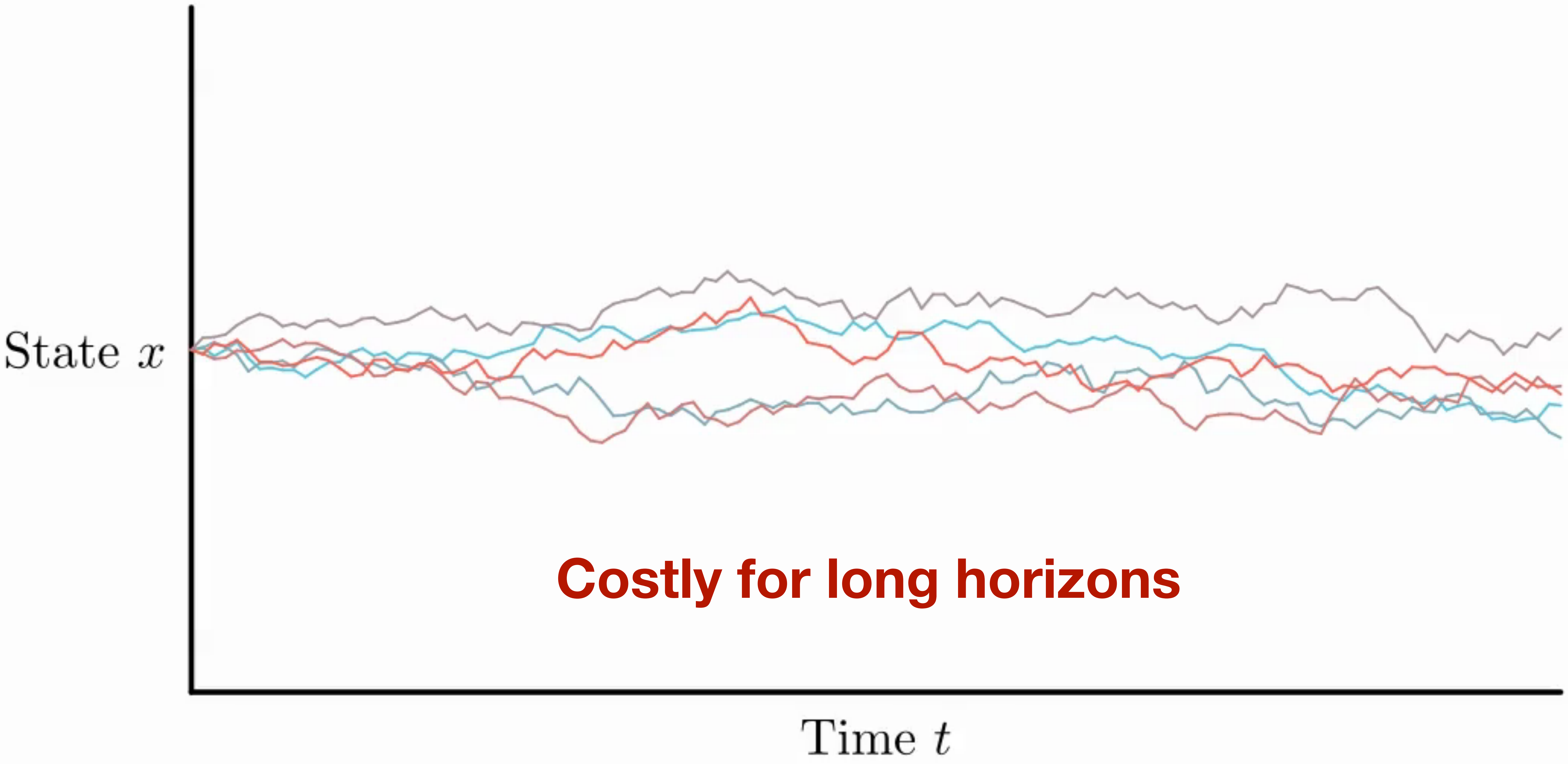
Computational Challenges of Neural SDEs



Computational Challenges of Neural SDEs



Computational Challenges of Neural SDEs



Strong Solution and Weak Solution of the SDEs

$$\text{Itô SDE:} \quad dx_t = \mu(x_t, t)\, dt + \sigma(x_t, t)\, dW_t, \quad x_0 \sim p_0$$

Strong solution

A solution x_t defined on a given probability space with a given Brownian motion. You do not change the noise; x_t must adapt to the fixed W_t.

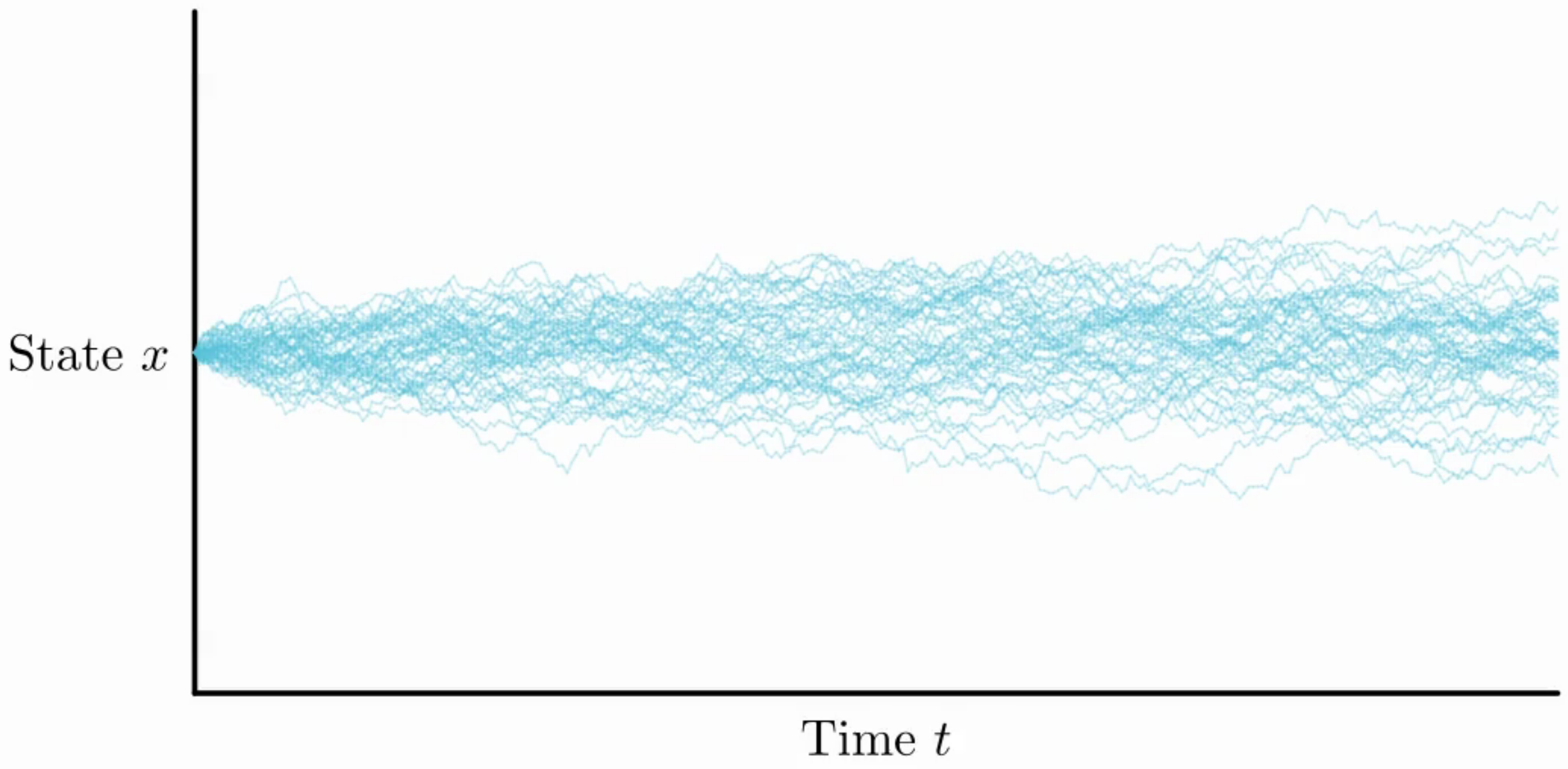
- Neural SDE: model sample paths w.r.t. a fixed Brownian motion.

Weak solution

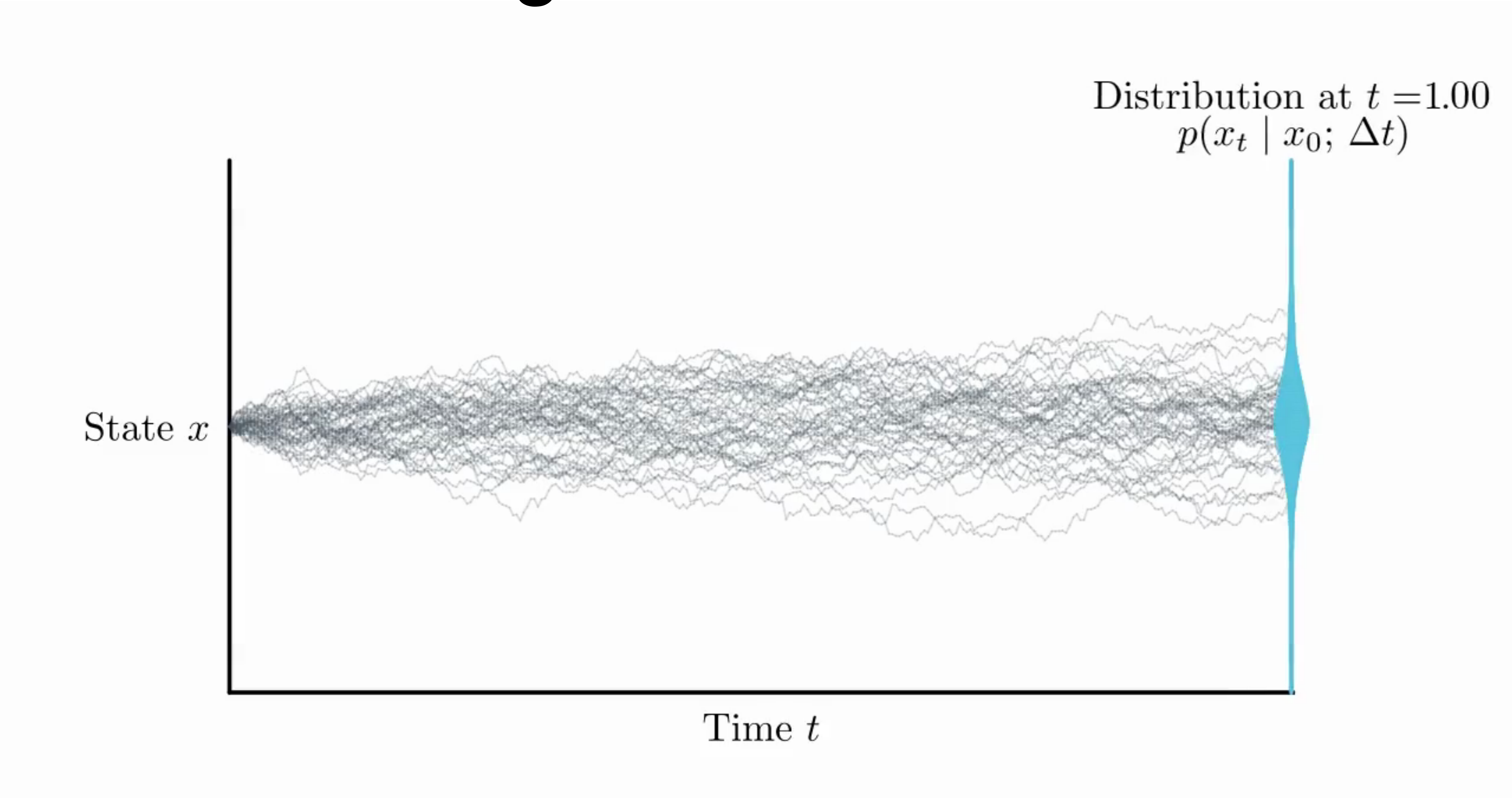
A solution x_t where the probability space and the Brownian motion are part of what you can choose. You only require that x_t and some Brownian motion satisfy the SDE and have the correct distribution.

- **Our idea: SDE as a Markov model—characterised by the transition law $p(x_t \mid x_s)$.**

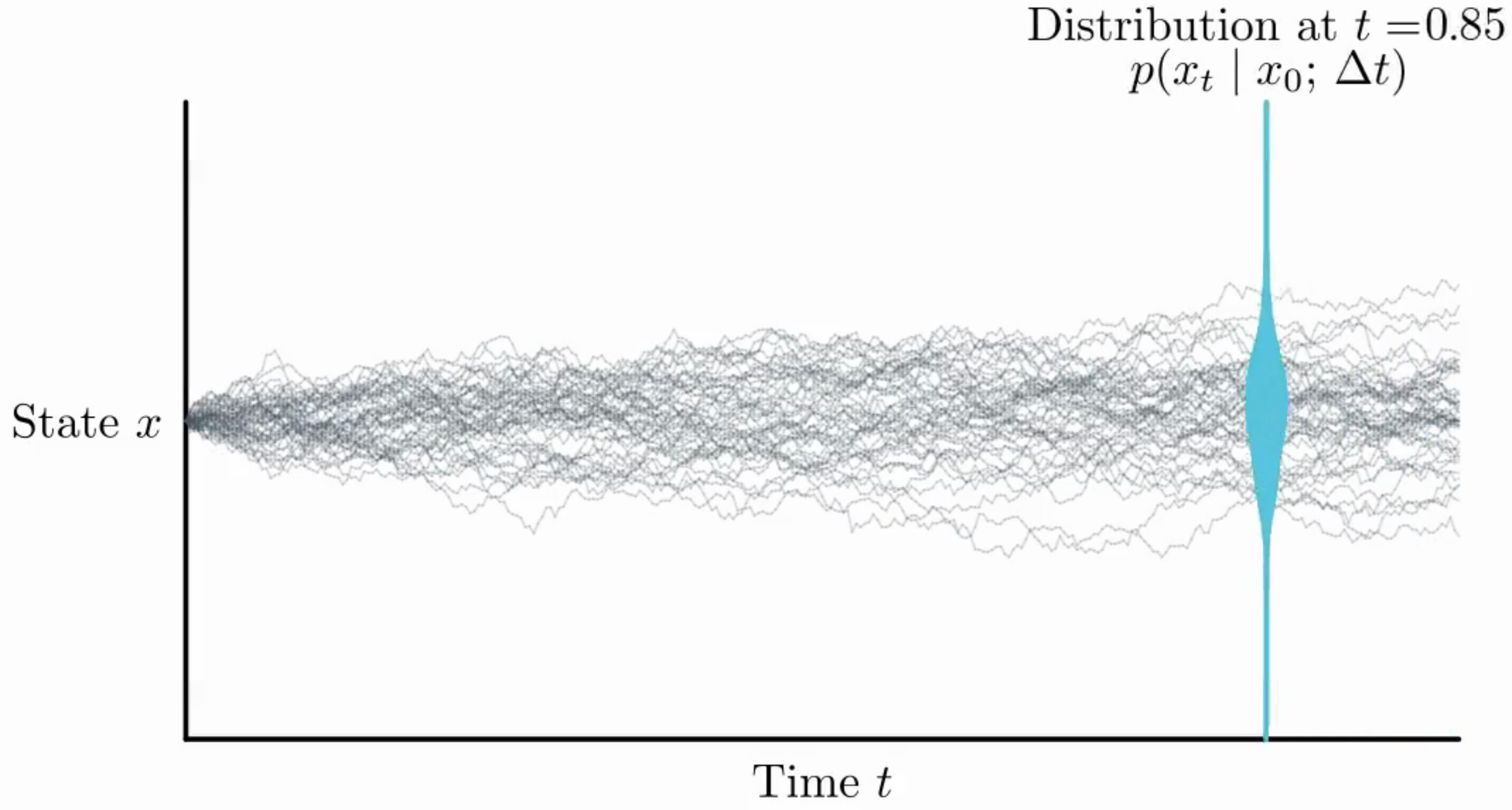
Direct Modelling of Transition Distributions



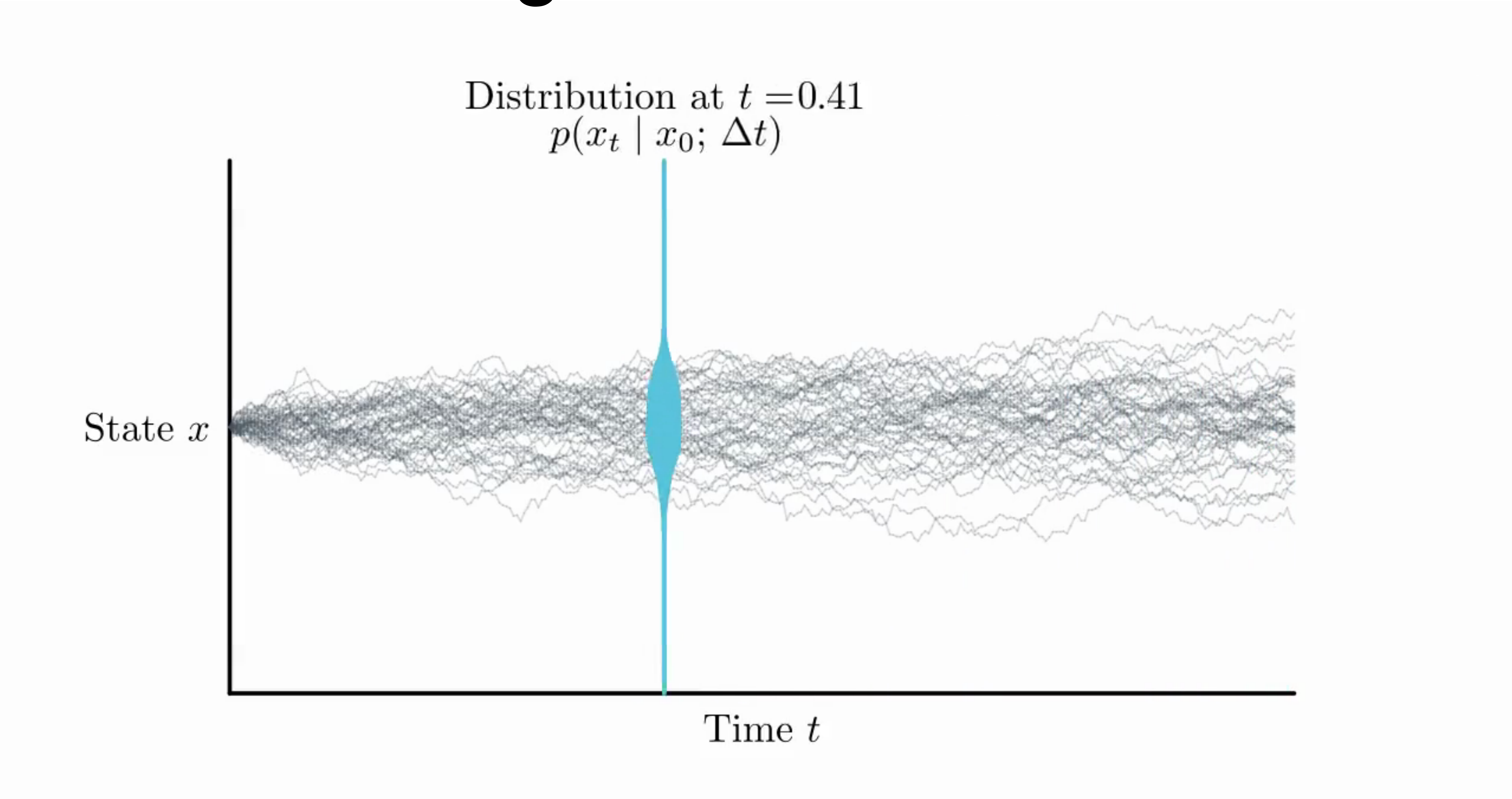
Direct Modelling of Transition Distributions



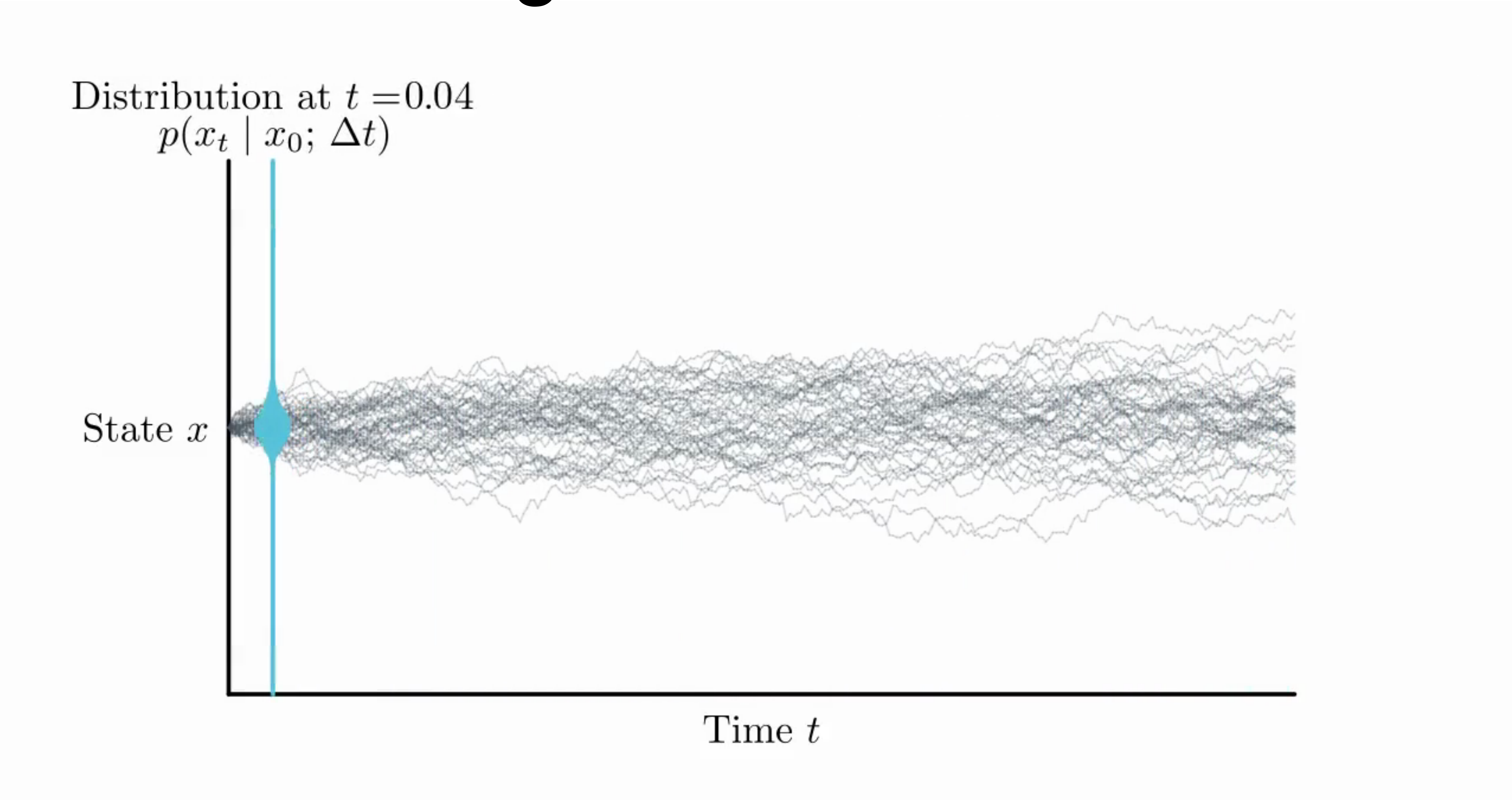
Direct Modelling of Transition Distributions



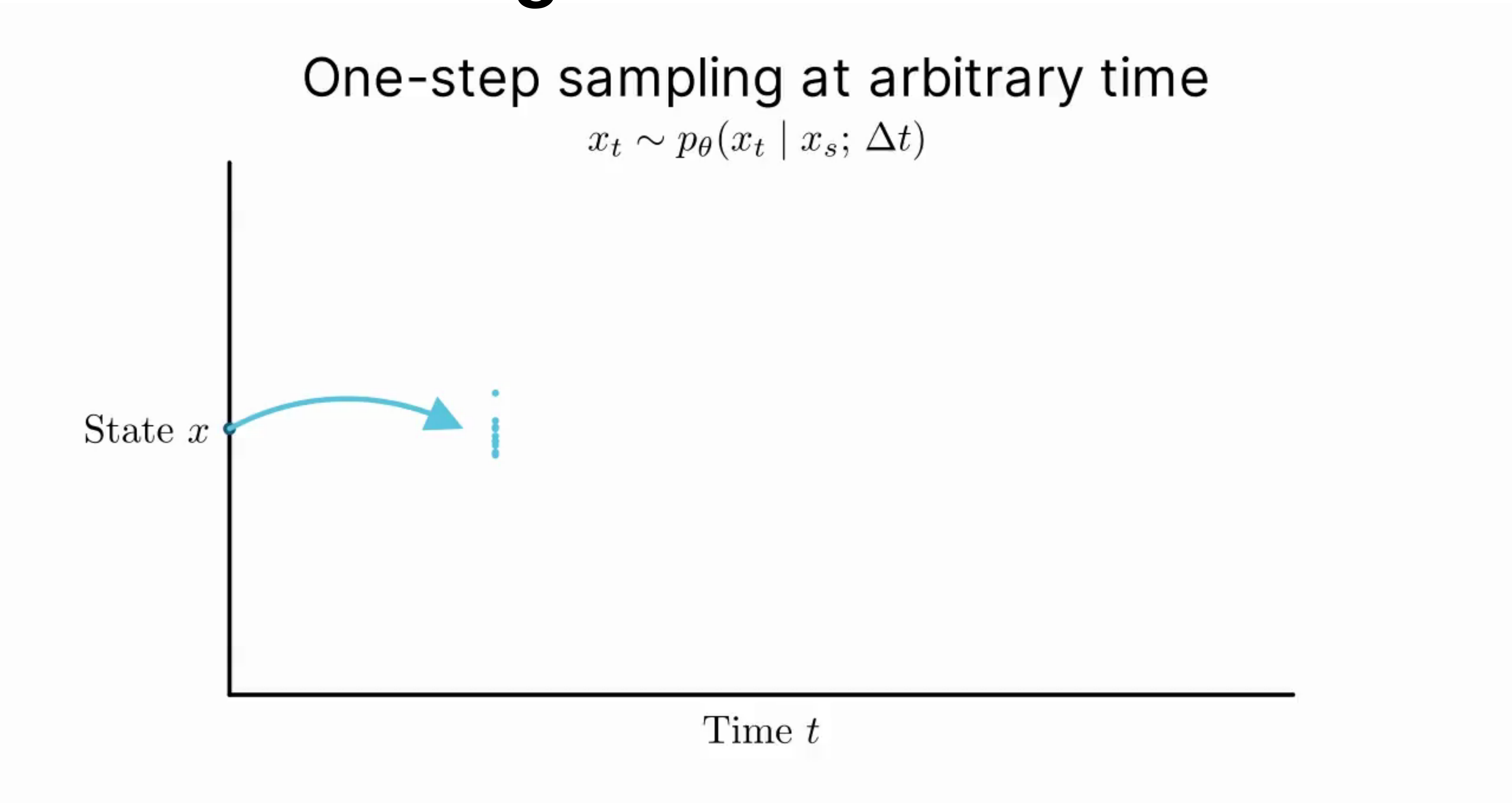
Direct Modelling of Transition Distributions



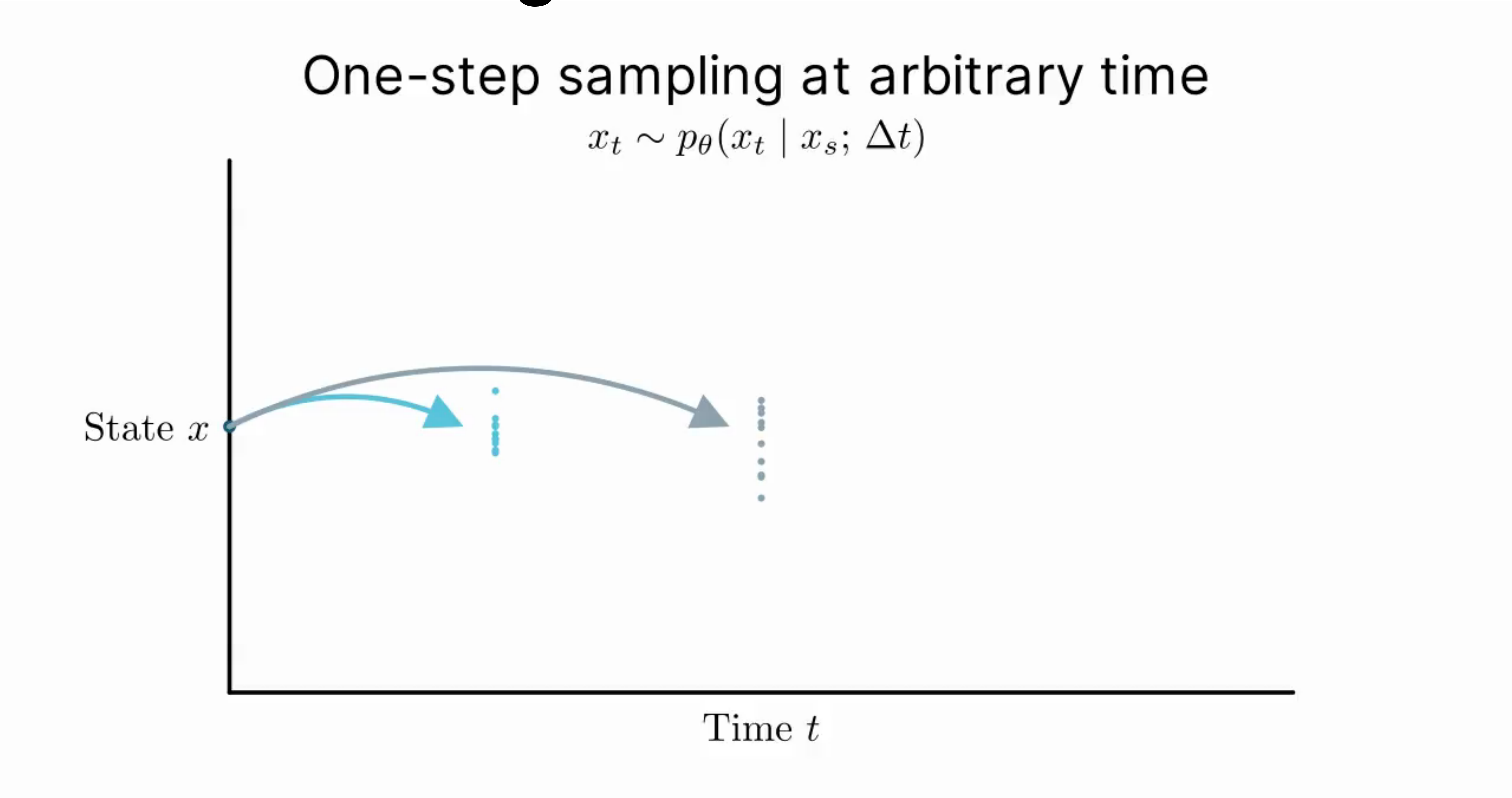
Direct Modelling of Transition Distributions



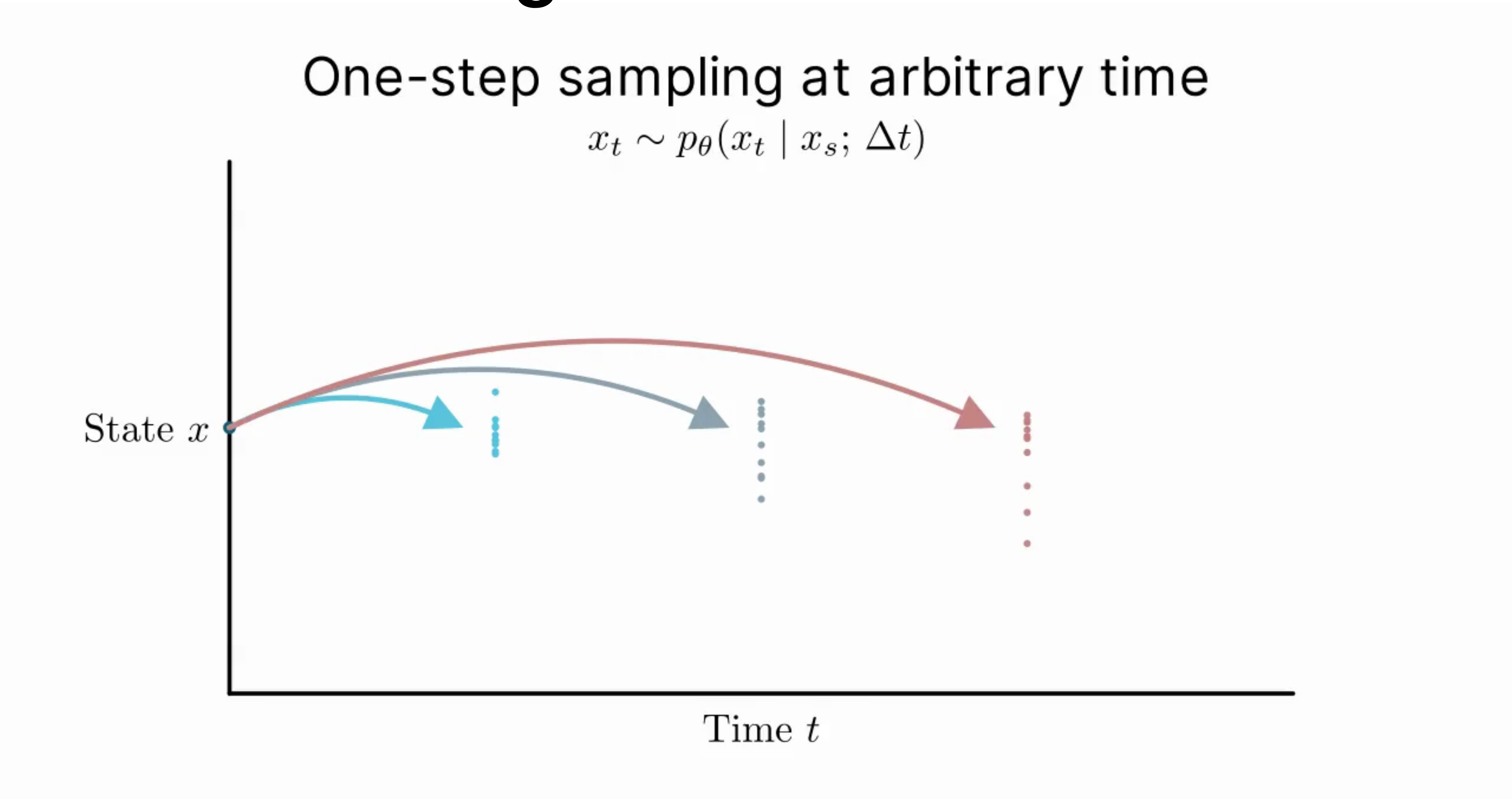
Direct Modelling of Transition Distributions



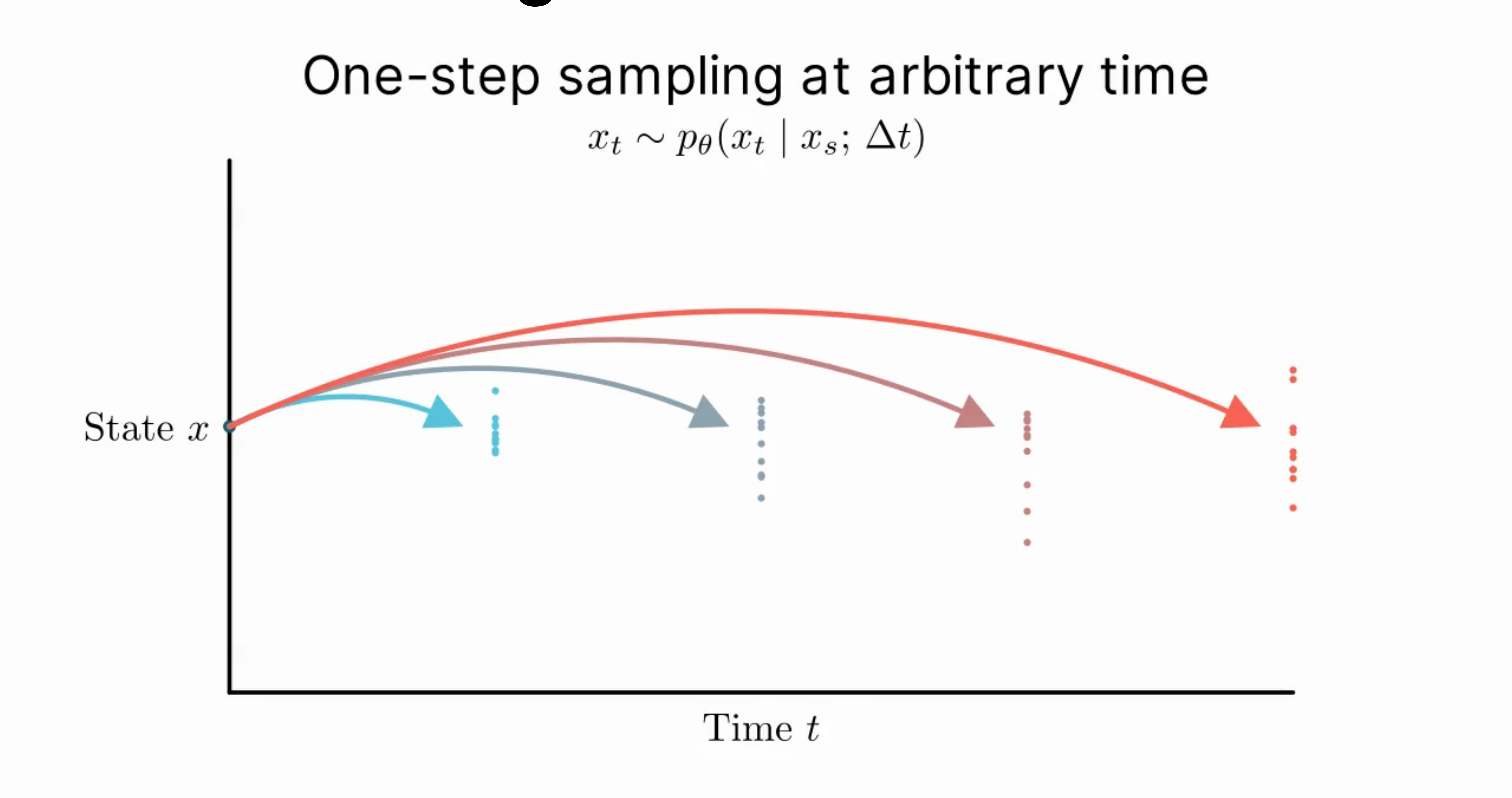
Direct Modelling of Transition Distributions



Direct Modelling of Transition Distributions



Direct Modelling of Transition Distributions



Mathematical Requirements for the Transition Kernel

Properties of $p_{\boldsymbol{\theta}}(\boldsymbol{x}_t \mid \boldsymbol{x}_s; s, \Delta t = t - s)$

1. **Independence.**

   For any $0 \leq t_1 \leq \cdots \leq t_n$, $p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t_{k+1}} \mid \boldsymbol{x}_{t_k}; t_k, t_{k+1} - t_k)$ are independent.

2. **Identity.**

   When $t = s$, $p_{\boldsymbol{\theta}}(\boldsymbol{x}_t \mid \boldsymbol{x}_s; s, 0) = \delta(\boldsymbol{x}_t - \boldsymbol{x}_s)$

3. **Flow property.**

   For any $0 \leq t_i \leq t_j \leq t_k$, $p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t_k} \mid \boldsymbol{x}_{t_i}; t_i, t_k - t_i) = \int p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t_k} \mid \boldsymbol{x}_{t_j}; t_j, t_k - t_j)\, p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t_j} \mid \boldsymbol{x}_{t_i}; t_i, t_j - t_i)\, \mathrm{d}\boldsymbol{x}_{t_j}$

4. **Stationarity (for autonomous SDEs).**

   $p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t_j} \mid \boldsymbol{x}_s; t_i, t_j - t_i) = p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t_j + r} \mid \boldsymbol{x}_s; t_i + r, t_j - t_i)$

Mathematical Requirements for the Transition Kernel

Properties of $p_{\boldsymbol{\theta}}(\boldsymbol{x}_t \mid \boldsymbol{x}_s; s, \Delta t = t - s)$

1. **Independence.** **✓ Realised by non-overlapping sampling.**

   For any $0 \leq t_1 \leq \cdots \leq t_n$, $p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t_{k+1}} \mid \boldsymbol{x}_{t_k}; t_k, t_{k+1} - t_k)$ are independent.

2. **Identity.**

   When $t = s$, $p_{\boldsymbol{\theta}}(\boldsymbol{x}_t \mid \boldsymbol{x}_s; s, 0) = \delta(\boldsymbol{x}_t - \boldsymbol{x}_s)$

3. **Flow property.**

   For any $0 \leq t_i \leq t_j \leq t_k$, $p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t_k} \mid \boldsymbol{x}_{t_i}; t_i, t_k - t_i) = \int p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t_k} \mid \boldsymbol{x}_{t_j}; t_j, t_k - t_j)\, p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t_j} \mid \boldsymbol{x}_{t_i}; t_i, t_j - t_i)\, \mathrm{d}\boldsymbol{x}_{t_j}$

4. **Stationarity (for autonomous SDEs).**

   $p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t_j} \mid \boldsymbol{x}_s; t_i, t_j - t_i) = p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t_j+r} \mid \boldsymbol{x}_s; t_i + r, t_j - t_i)$

Architectural Design

Conditional normalising flow for $p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t_j} \mid \boldsymbol{x}_{t_i}; t_i, \Delta t)$

Base distribution

Using conditioning $\boldsymbol{c} := (\boldsymbol{x}_{t_i}, \Delta t, t_i)$,

$$\boldsymbol{z} = \boldsymbol{x}_{t_i} + \Delta t \cdot \mathrm{MLP}_{\mu}(\boldsymbol{c}; \boldsymbol{\theta}_{\mu}) + \sqrt{\Delta t} \cdot \mathrm{MLP}_{\sigma}(\boldsymbol{c}; \boldsymbol{\theta}_{\sigma}) \odot \boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$$

Conditional affine coupling layers [4, 13, 28]

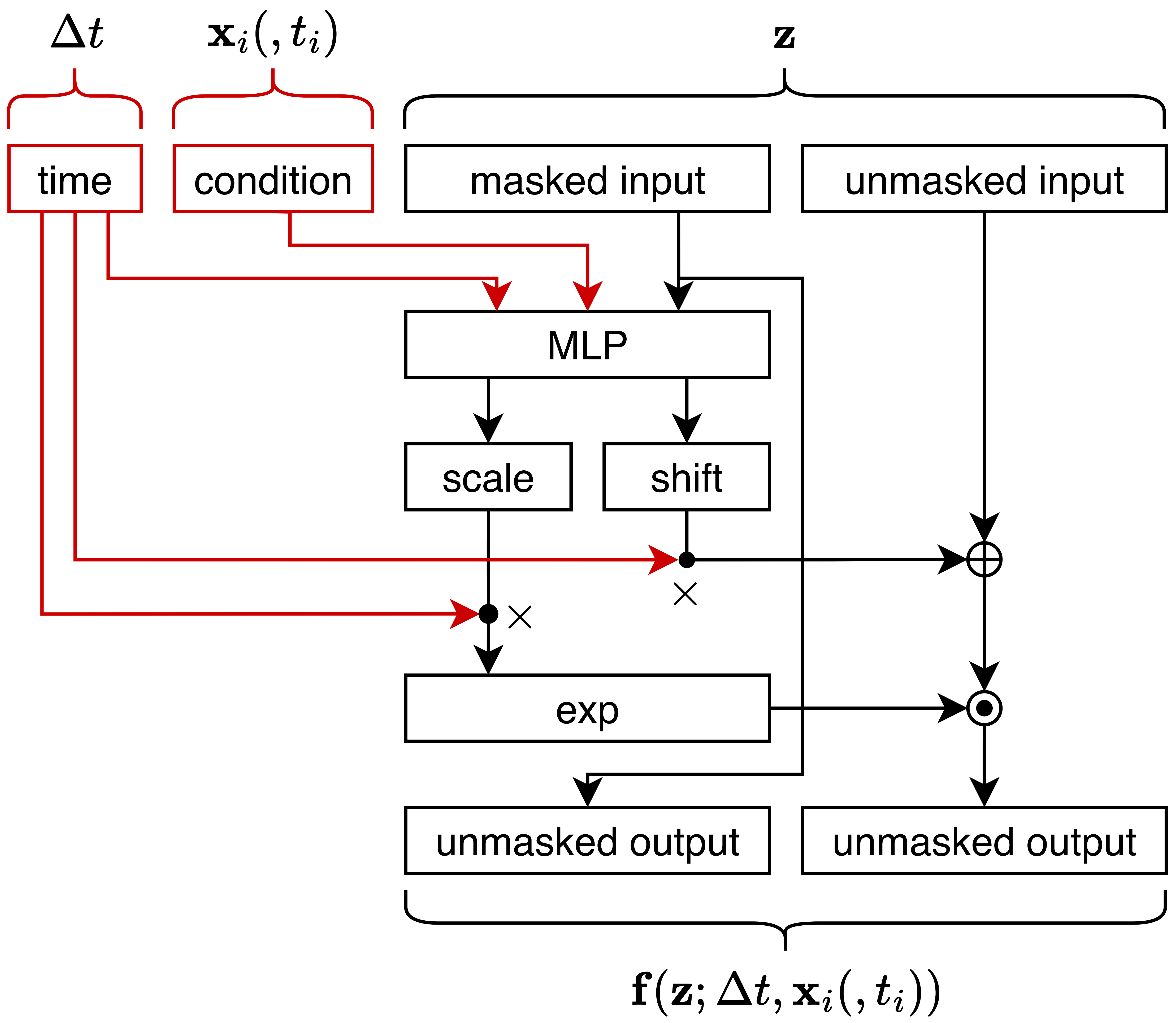
$$\boldsymbol{x}_{t_j} = \boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{c}) = \boldsymbol{f}_L(\cdot; \boldsymbol{c}, \boldsymbol{\theta}_L) \circ \boldsymbol{f}_{L-1}(\cdot; \boldsymbol{c}, \boldsymbol{\theta}_{L-1}) \circ \cdots \circ \boldsymbol{f}_1(\boldsymbol{z}; \boldsymbol{c}, \boldsymbol{\theta}_1)$$

With $\boldsymbol{z}_{\mathrm{A}}, \boldsymbol{z}_{\mathrm{B}} = \mathrm{Split}(\boldsymbol{z})$, each layer $\boldsymbol{f}$ is designed as:

$$\mathrm{Concat}\Big(\boldsymbol{z}_{\mathrm{A}}, \boldsymbol{z}_{\mathrm{B}} \odot \exp\big(\Delta t\, \mathrm{MLP}^{(i)}_{\mathrm{scale}}(\boldsymbol{z}_{\mathrm{A}}, \boldsymbol{c}; \boldsymbol{\theta}^{(i)}_{\mathrm{scale}})\big) + \Delta t\, \mathrm{MLP}^{(i)}_{\mathrm{shift}}(\boldsymbol{z}_{\mathrm{A}}, \boldsymbol{c}; \boldsymbol{\theta}^{(i)}_{\mathrm{shift}})\Big)$$

Architectural Design

Conditional affine coupling part for $p_\theta(\boldsymbol{x}_{t_j} \mid \boldsymbol{x}_{t_i}; t_i, \Delta t)$



Architectural Design

Conditional normalising flow for $p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t_j} \mid \boldsymbol{x}_{t_i}; t_i, \Delta t)$

Base distribution

Using conditioning $\boldsymbol{c} := (\boldsymbol{x}_{t_i}, \Delta t, t_i)$,

$$\boldsymbol{z} = \boldsymbol{x}_{t_i} + \Delta t \cdot \mathrm{MLP}_\mu(\boldsymbol{c}; \boldsymbol{\theta}_\mu) + \sqrt{\Delta t} \cdot \mathrm{MLP}_\sigma(\boldsymbol{c}; \boldsymbol{\theta}_\sigma) \odot \boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$$

Conditional affine coupling layers [4, 13, 28]

$$\boldsymbol{x}_{t_j} = \boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{c}) = \boldsymbol{f}_L(\cdot; \boldsymbol{c}, \boldsymbol{\theta}_L) \circ \boldsymbol{f}_{L-1}(\cdot; \boldsymbol{c}, \boldsymbol{\theta}_{L-1}) \circ \cdots \circ \boldsymbol{f}_1(\boldsymbol{z}; \boldsymbol{c}, \boldsymbol{\theta}_1)$$

With $\boldsymbol{z}_\mathrm{A}, \boldsymbol{z}_\mathrm{B} = \mathrm{Split}(\boldsymbol{z})$, each layer $\boldsymbol{f}$ is designed as:

$$\mathrm{Concat}\Big(\boldsymbol{z}_\mathrm{A}, \boldsymbol{z}_\mathrm{B} \odot \exp\big(\Delta t\, \mathrm{MLP}^{(i)}_{\mathrm{scale}}(\boldsymbol{z}_\mathrm{A}, \boldsymbol{c}; \boldsymbol{\theta}^{(i)}_{\mathrm{scale}})\big) + \Delta t\, \mathrm{MLP}^{(i)}_{\mathrm{shift}}(\boldsymbol{z}_\mathrm{A}, \boldsymbol{c}; \boldsymbol{\theta}^{(i)}_{\mathrm{shift}})\Big)$$

✓ **Identity when $\Delta t = 0$.** ✓ **Stationary when we drop t_i from $\boldsymbol{c}$.**

Mathematical Requirements for the Transition Kernel

Properties of $p_{\boldsymbol{\theta}}(\boldsymbol{x}_t \mid \boldsymbol{x}_s; s, \Delta t = t - s)$

1. **Independence.** **✓ Realised by non-overlapping sampling.**

   For any $0 \leq t_1 \leq \cdots \leq t_n$, $p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t_{k+1}} \mid \boldsymbol{x}_{t_k}; t_k, t_{k+1} - t_k)$ are independent.

2. **Identity.** **✓ Realised by architecture.**

   When $t = s$, $p_{\boldsymbol{\theta}}(\boldsymbol{x}_t \mid \boldsymbol{x}_s; s, 0) = \delta(\boldsymbol{x}_t - \boldsymbol{x}_s)$

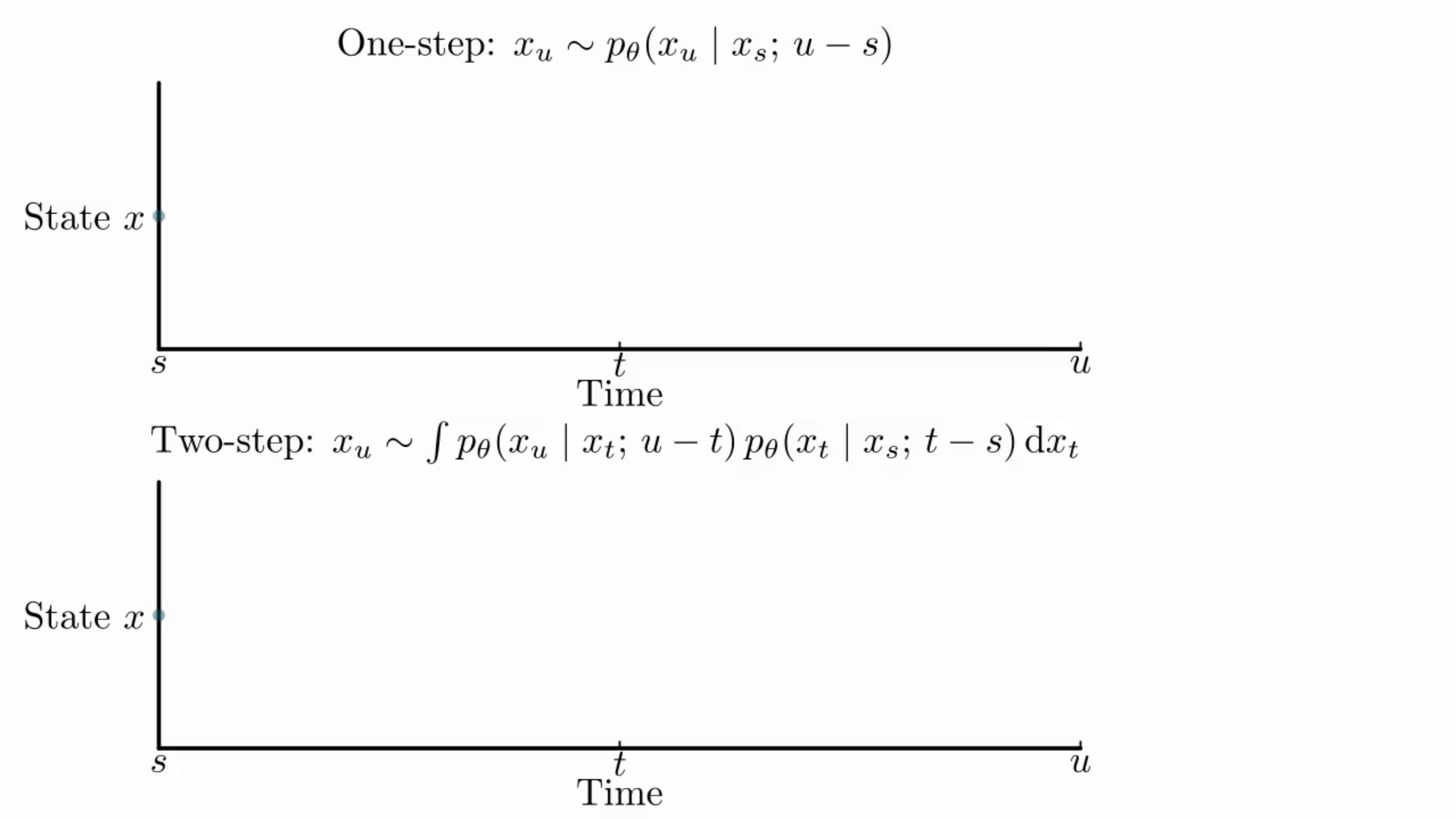
3. **Flow property.**

   For any $0 \leq t_i \leq t_j \leq t_k$, $p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t_k} \mid \boldsymbol{x}_{t_i}; t_i, t_k - t_i) = \int p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t_k} \mid \boldsymbol{x}_{t_j}; t_j, t_k - t_j)\, p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t_j} \mid \boldsymbol{x}_{t_i}; t_i, t_j - t_i)\, \mathrm{d}\boldsymbol{x}_{t_j}$

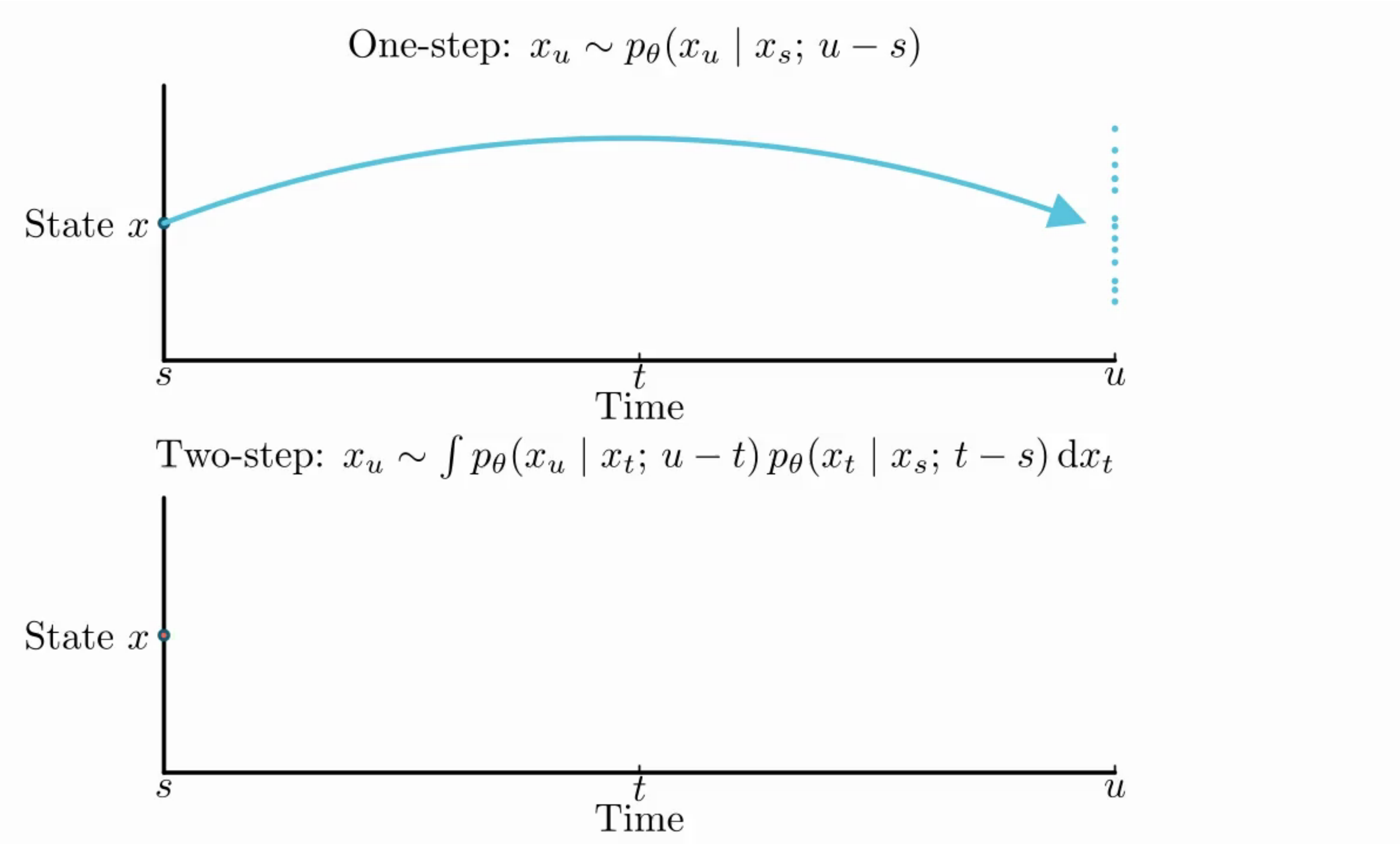
4. **Stationarity (for autonomous SDEs).** **✓ Realised by architecture optionally.**

   $p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t_j} \mid \boldsymbol{x}_s; t_i, t_j - t_i) = p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t_j+r} \mid \boldsymbol{x}_s; t_i + r, t_j - t_i)$

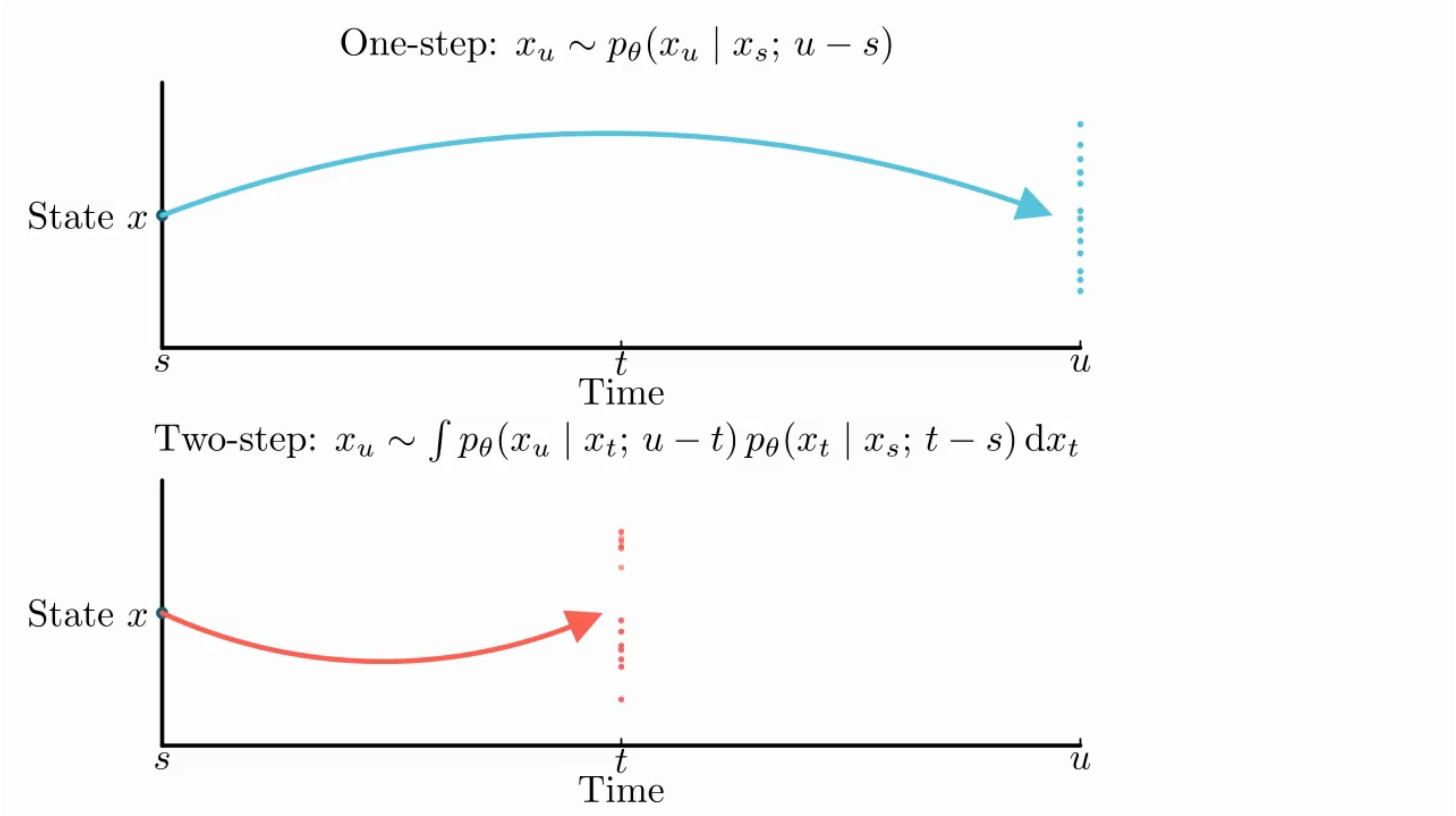
Regularisation for Flow Consistency



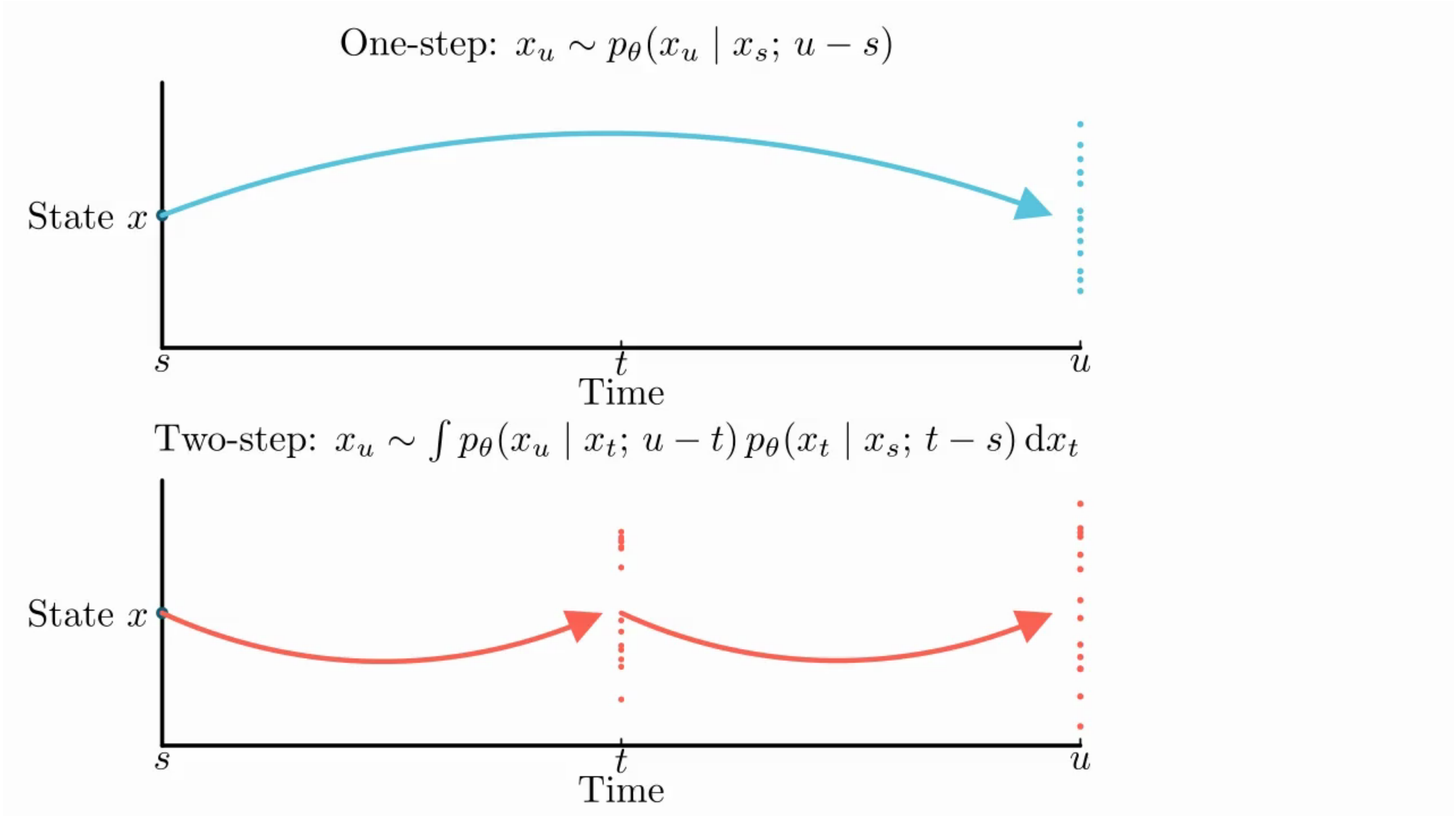
Regularisation for Flow Consistency



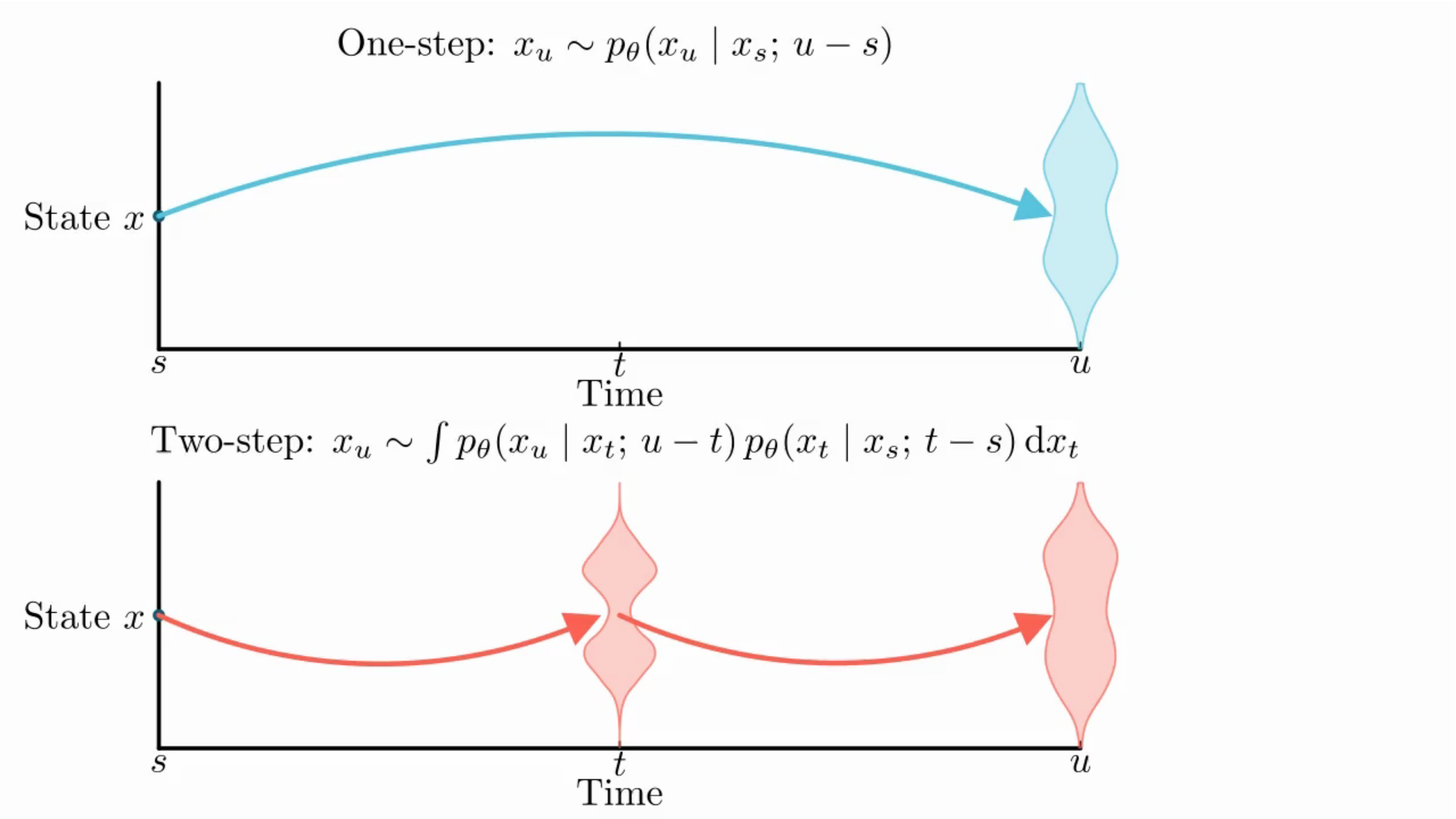
Regularisation for Flow Consistency



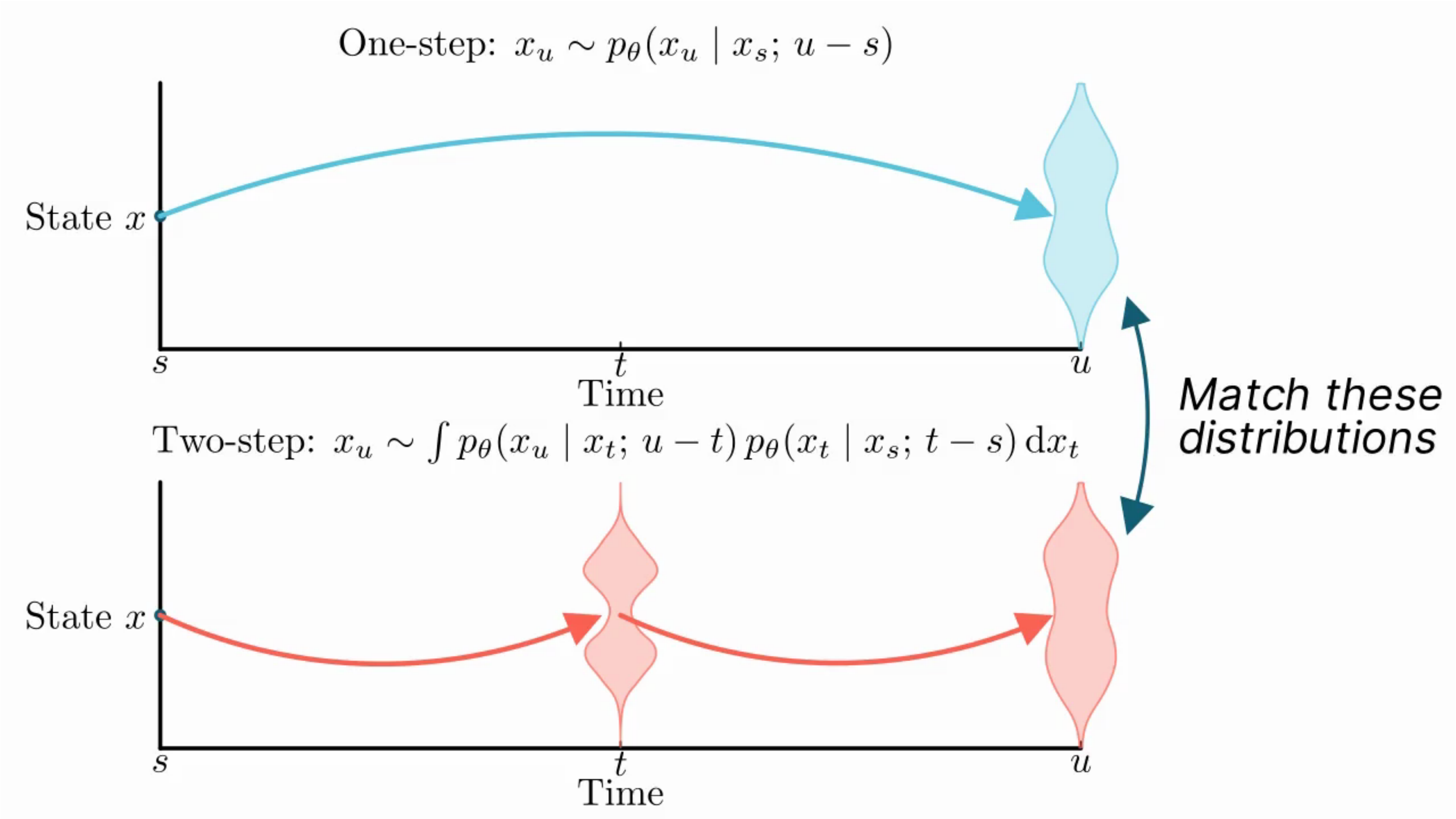
Regularisation for Flow Consistency



Regularisation for Flow Consistency



Regularisation for Flow Consistency



Regularisation for Flow Consistency

$$\mathcal{L}_{\text{flow}} \; = D\left[p_{\boldsymbol{\theta}}^{1}\left(\mathbf{x}_{t_k} \mid \mathbf{x}_{t_i}\right), \int p_{\boldsymbol{\theta}}^{2}\left(\mathbf{x}_{t_k} \mid \mathbf{x}_{t_j}\right) p_{\boldsymbol{\theta}}^{2}\left(\mathbf{x}_{t_j} \mid \mathbf{x}_{t_i}\right) d\mathbf{x}_{t_j}\right]$$

Challenges in computing the flow loss

- The marginalisation is generally intractable
  → **Direct access to the marginalised density is not available**
- Naive Monte Carlo estimation would require many samples

We derive the upper bounds for KL divergences

Regularisation for Flow Consistency

Upper bounds for KL divergences

1-step → 2-step KL divergence

$$D_{\mathrm{KL}}\left(p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t_k} \mid \boldsymbol{x}_{t_i}) \,\middle\|\, \int p_{\boldsymbol{\theta}}\left(\boldsymbol{x}_{t_k} \mid \boldsymbol{x}_{t_j}\right) p_{\boldsymbol{\theta}}\left(\boldsymbol{x}_{t_j} \mid \boldsymbol{x}_{t_i}\right) \mathrm{d}\boldsymbol{x}_{t_j}\right)$$

$$\leq \mathbb{E}_{\boldsymbol{x}_{t_k} \sim p_{\boldsymbol{\theta}}(\cdot \mid \boldsymbol{x}_{t_i})}\left[\mathbb{E}_{\boldsymbol{x}_{t_j} \sim b_{\boldsymbol{\xi}}(\cdot \mid \boldsymbol{x}_{t_i}, \boldsymbol{x}_{t_k})}\left[\log\left(\frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t_k} \mid \boldsymbol{x}_{t_i})\, b_{\boldsymbol{\xi}}\left(\boldsymbol{x}_{t_j} \mid \boldsymbol{x}_{t_i}, \boldsymbol{x}_{t_k}\right)}{p_{\boldsymbol{\theta}}\left(\boldsymbol{x}_{t_k} \mid \boldsymbol{x}_{t_j}\right) p_{\boldsymbol{\theta}}\left(\boldsymbol{x}_{t_j} \mid \boldsymbol{x}_{t_i}\right)}\right)\right]\right] =: \mathcal{L}_{\text{flow, 1-to-2}}(\boldsymbol{\theta}, \boldsymbol{\xi}; t_i, t_j, t_k)\,.$$

2-step → 1-step KL divergence

$$D_{\mathrm{KL}}\left(\int p_{\boldsymbol{\theta}}\left(\boldsymbol{x}_{t_k} \mid \boldsymbol{x}_{t_j}\right) p_{\boldsymbol{\theta}}\left(\boldsymbol{x}_{t_j} \mid \boldsymbol{x}_{t_i}\right) \mathrm{d}\boldsymbol{x}_{t_j} \,\middle\|\, p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t_k} \mid \boldsymbol{x}_{t_i})\right)$$

$$\leq \mathbb{E}_{\boldsymbol{x}_{t_j} \sim p_{\boldsymbol{\theta}}(\cdot \mid \boldsymbol{x}_{t_i})}\left[\mathbb{E}_{\boldsymbol{x}_{t_k} \sim p_{\boldsymbol{\theta}}(\cdot \mid \boldsymbol{x}_{t_j})}\left[\log\left(\frac{p_{\boldsymbol{\theta}}\left(\boldsymbol{x}_{t_j} \mid \boldsymbol{x}_{t_i}\right) p_{\boldsymbol{\theta}}\left(\boldsymbol{x}_{t_k} \mid \boldsymbol{x}_{t_j}\right)}{b_{\boldsymbol{\xi}}\left(\boldsymbol{x}_{t_j} \mid \boldsymbol{x}_{t_i}, \boldsymbol{x}_{t_k}\right) p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t_k} \mid \boldsymbol{x}_{t_i})}\right)\right]\right] =: \mathcal{L}_{\text{flow, 2-to-1}}(\boldsymbol{\theta}, \boldsymbol{\xi}; t_i, t_j, t_k)\,.$$

$$\mathcal{L}_{\text{flow}}(\boldsymbol{\theta}, \boldsymbol{\xi}) := \mathop{\mathbb{E}}_{p(t_i, t_j, t_k)}\left[\mathcal{L}_{\text{flow, 1-to-2}}(\boldsymbol{\theta}, \boldsymbol{\xi}; t_i, t_j, t_k) + \mathcal{L}_{\text{flow, 2-to-1}}(\boldsymbol{\theta}, \boldsymbol{\xi}; t_i, t_j, t_k)\right]$$

Mathematical Requirements for the Transition Kernel

Properties of $p_{\boldsymbol{\theta}}(\boldsymbol{x}_t \mid \boldsymbol{x}_s; s, \Delta t = t - s)$

1. **Independence.** **✓ Realised by non-overlapping sampling.**

   For any $0 \leq t_1 \leq \cdots \leq t_n$, $p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t_{k+1}} \mid \boldsymbol{x}_{t_k}; t_k, t_{k+1} - t_k)$ are independent.

2. **Identity.** **✓ Realised by architecture.**

   When $t = s$, $p_{\boldsymbol{\theta}}(\boldsymbol{x}_t \mid \boldsymbol{x}_s; s, 0) = \delta(\boldsymbol{x}_t - \boldsymbol{x}_s)$

3. **Flow property.** **✓ Approximately realised by flow loss.**

   For any $0 \leq t_i \leq t_j \leq t_k$, $p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t_k} \mid \boldsymbol{x}_{t_i}; t_i, t_k - t_i) = \int p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t_k} \mid \boldsymbol{x}_{t_j}; t_j, t_k - t_j)\, p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t_j} \mid \boldsymbol{x}_{t_i}; t_i, t_j - t_i)\, \mathrm{d}\boldsymbol{x}_{t_j}$

4. **Stationarity (for autonomous SDEs).** **✓ Realised by architecture optionally.**

   $p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t_j} \mid \boldsymbol{x}_s; t_i, t_j - t_i) = p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t_j + r} \mid \boldsymbol{x}_s; t_i + r, t_j - t_i)$

Training Objective

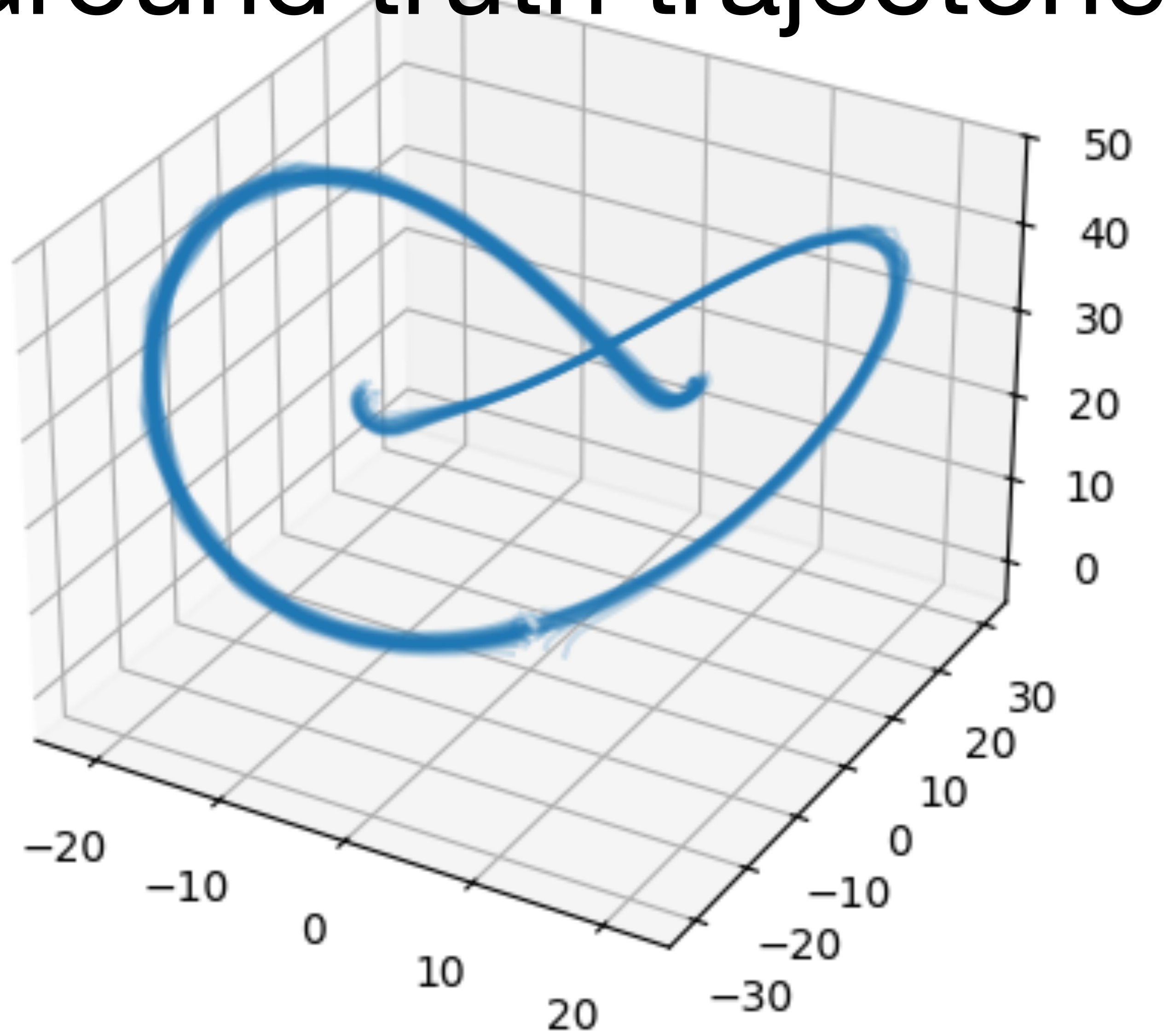
$$\mathcal{L} = \mathop{\mathbb{E}}_{\boldsymbol{x}_{t_j}, \boldsymbol{x}_{t_i} \sim \mathcal{D}} \left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t_j} \mid \boldsymbol{x}_{t_i}) \right] + \lambda \mathcal{L}_{\text{flow}}(\boldsymbol{\theta}, \boldsymbol{\xi})$$

Negative log likelihood **Flow loss**

Experimental Configurations

Chaotic dynamics—Stochastic Lorenz Attractor

Ground truth trajectories



$$\mathrm{d}x = \sigma(y - x)\mathrm{d}t + \alpha_x \, \mathrm{d}W_1$$

$$\mathrm{d}y = (x(\rho - z) - y)\mathrm{d}t + \alpha_y \, \mathrm{d}W_2$$

$$\mathrm{d}z = (xy - \beta z)\mathrm{d}t + \alpha_z \, \mathrm{d}W_3$$

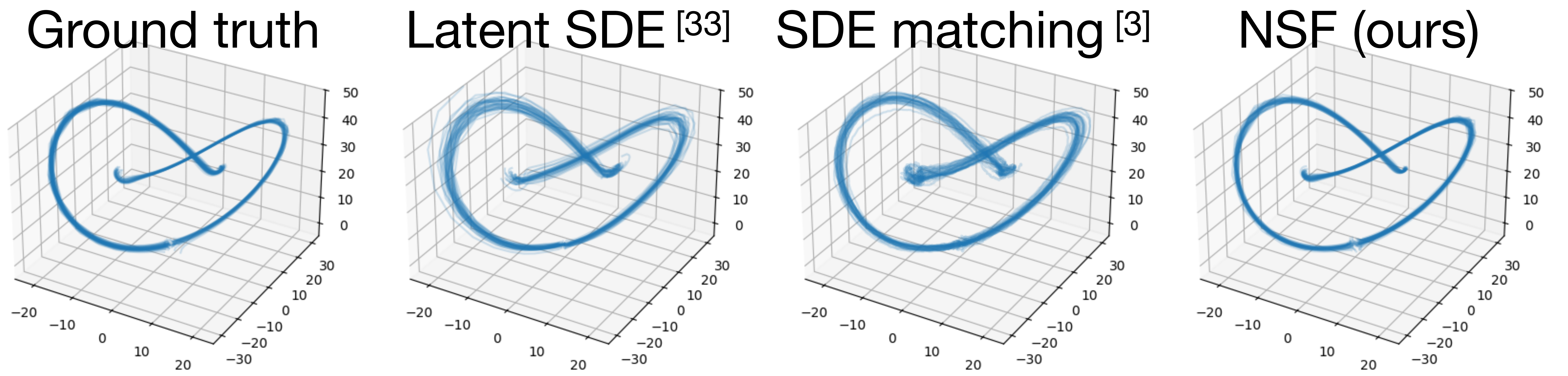
$$\sigma = 10, \rho = 28, \beta = 8/3, \alpha = (0.15, 0.15, 0.15)$$

$$\text{Initial state: } \mathcal{N}(0, I); \text{ Duration: } t \in [0, 1]$$

Trained on 1,024 trajectories, tested on 1,024 trajectories.

Experimental Results

With a comparable network size



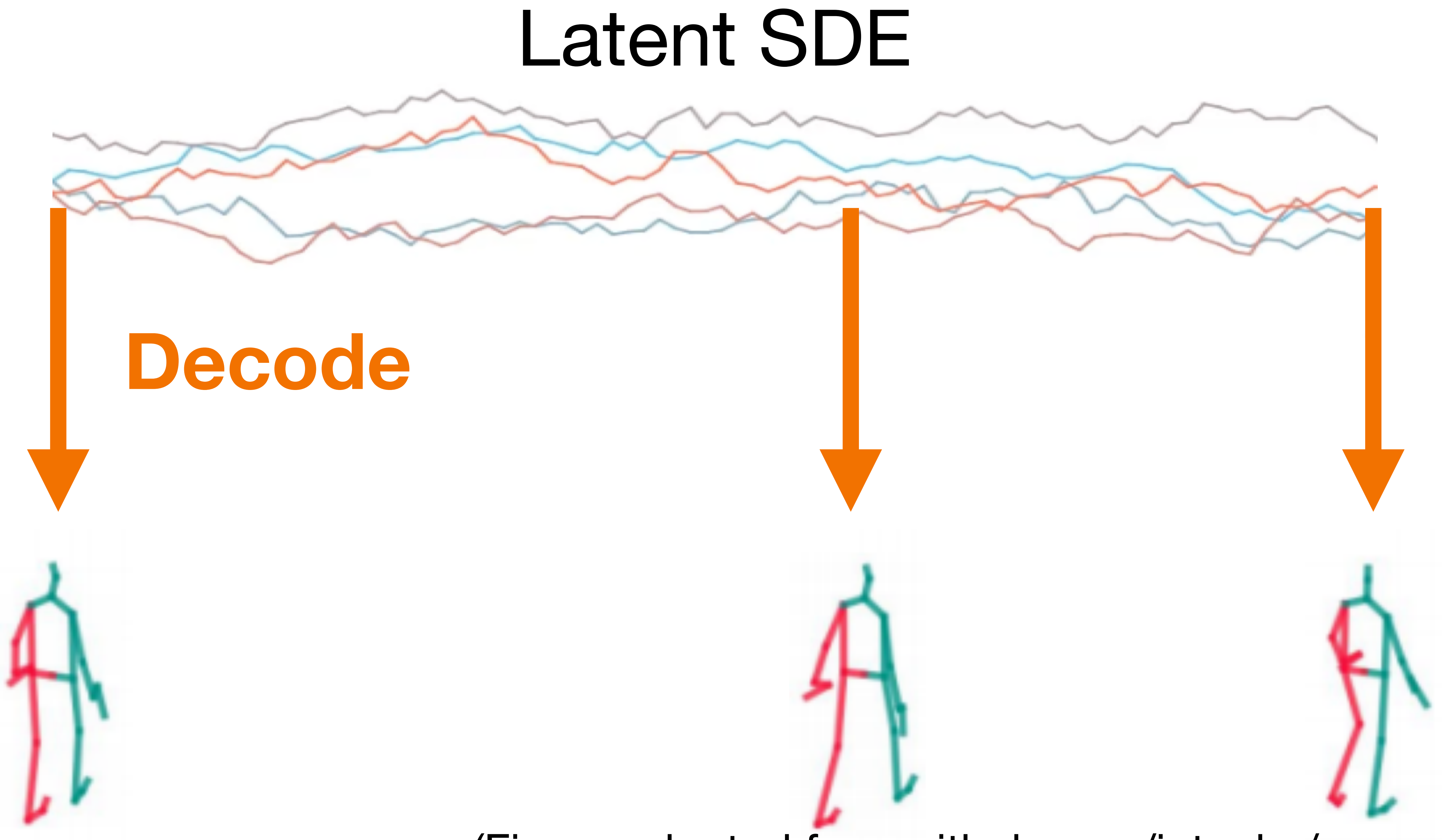
Method	t = 0.25		t = 0.5		t = 0.75		t = 1.0	
	KL ↓	kFLOPs ↓	KL ↓	kFLOPs ↓	KL ↓	kFLOPs ↓	KL ↓	kFLOPs ↓
Latent SDE [33]	2.1 ± 0.9	959	1.8 ± 0.1	1,917	0.9 ± 0.3	2,839	1.5 ± 0.5	3,760
SDE matching [3]	6.3 ± 0.4	1,917	11.7 ± 0.5	3,834	7.9 ± 0.3	5,677	6.0 ± 0.3	7,520
NSF (ours)	**0.8 ± 0.7**	**53**	**1.3 ± 0.1**	**53**	**0.6 ± 0.3**	**53**	**0.2 ± 0.6**	**53**

Latent SDEs and Latent Neural Stochastic Flows

Extension for partially-observed/high-dimensional data

Real-world time-series data is often:

- Partially-observed
- Irregularly-sampled
- High-dimensional



(Figure adapted from github.com/jutanke/mocap)

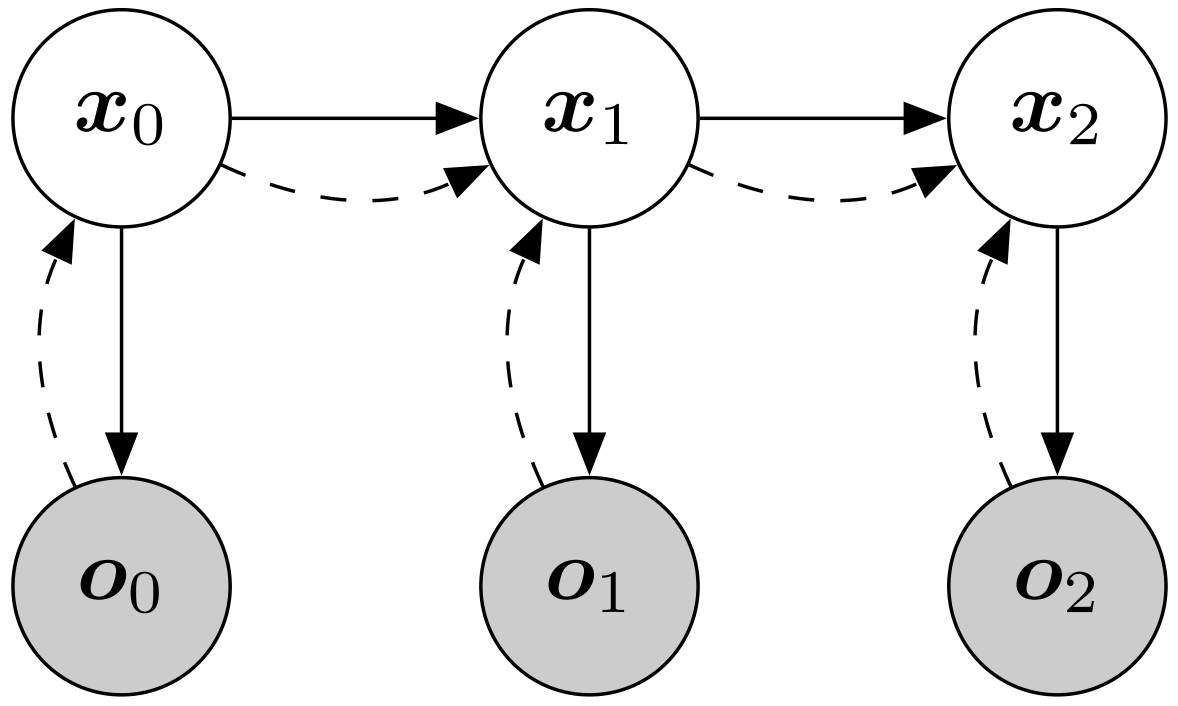
Latent SDEs are powerful tools, but they require high computational costs.

We replace SDEs with Neural Stochastic Flows.

Latent Neural Stochastic Flows

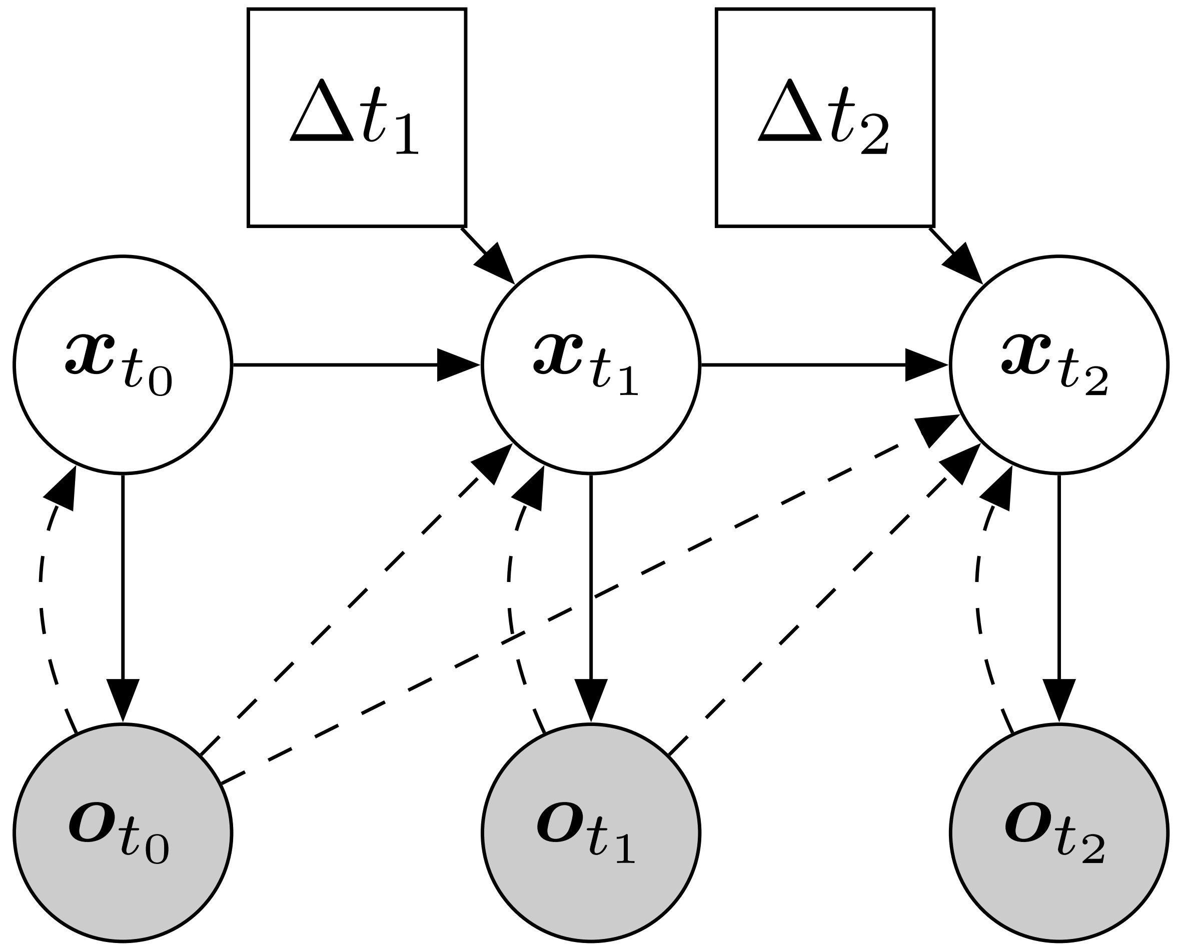
Extension for partially-observed/high-dimensional data

Variational State Space Model [8, 21, 22, 31]



A proven template for sequential modelling, including RL (PlaNet[16], Dreamer, etc.)

Latent NSF (ours)



✓ **VSSM structure remains the same by NSF's closed-form log density**

✓ **One-step transition for arbitrary Δt**

Experimental Results with Latent NSF

Real-world/high-dimensional sequential prediction

CMU Motion Capture (50-dims.)

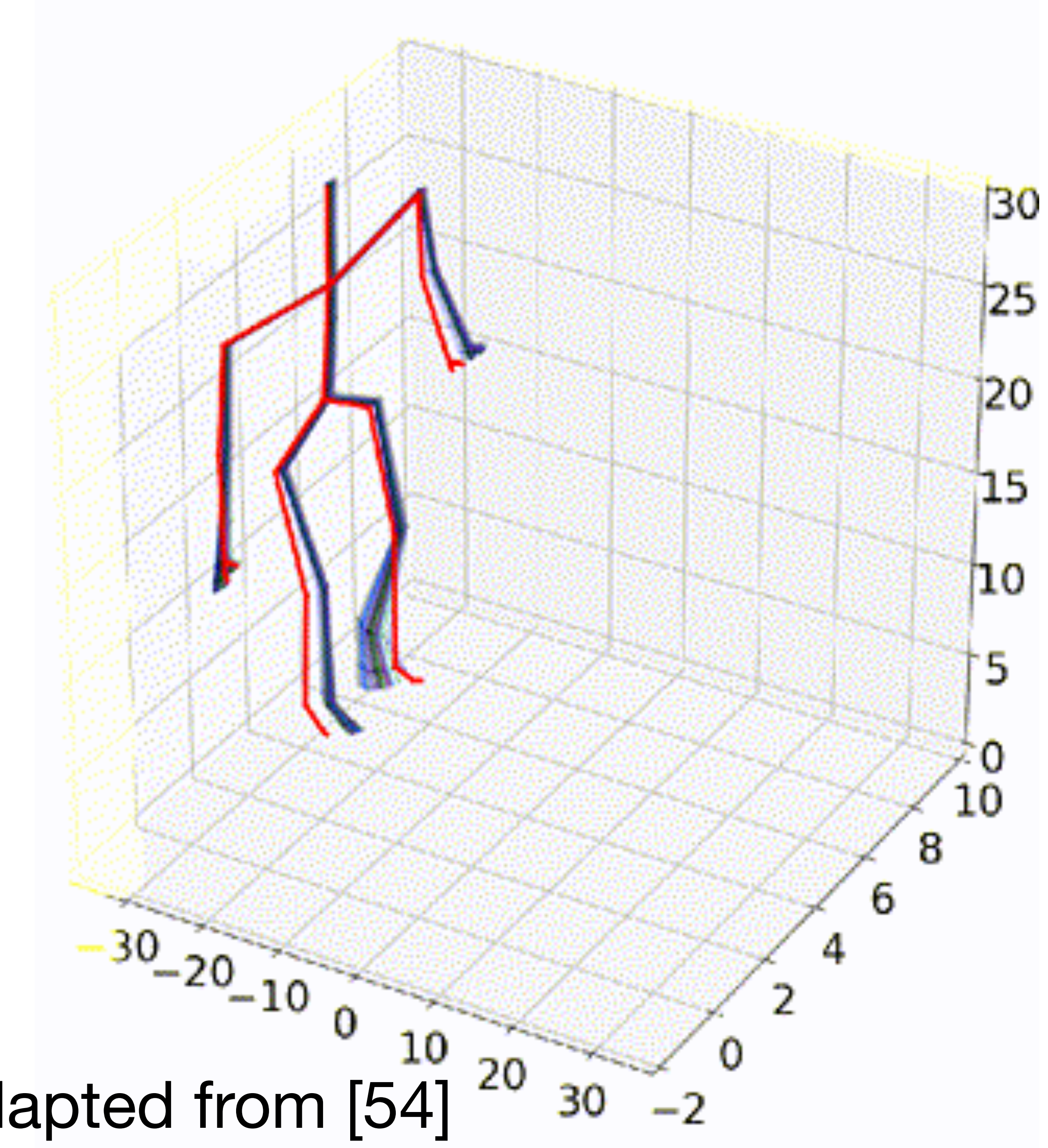


Figure adapted from [54]

Setup 1: The same length for train/test

Setup 2: Training length < Test length

Stochastic Moving MNIST (64x64 px)

Experimental Results with Latent NSF

Real-world/high-dimensional sequential prediction

CMU Motion Capture (50-dims.)

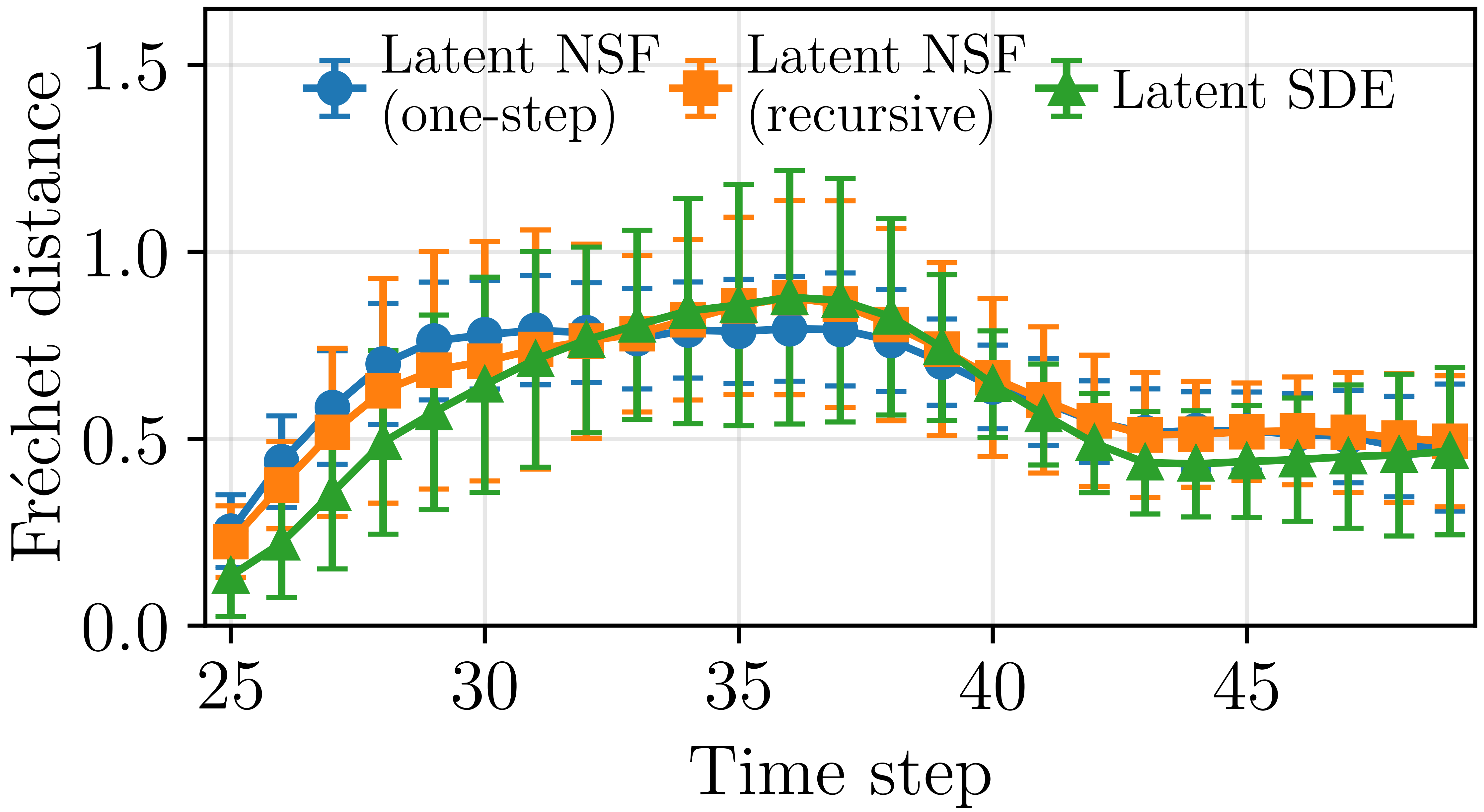
Compared by mean square error

Methods	Setup 1	Setup 2
npODE [17]	$22.96^{\dagger}$	–
Neural ODE [6]	$22.49 \pm 0.88^{\dagger}$	–
ODE2VAE-KL [54]	$8.09 \pm 1.95^{\dagger}$	–
Latent ODE [43]	$5.98 \pm 0.28^{*}$	$31.62 \pm 0.05^{\S}$
Latent SDE [33]	12.91 ± 2.90^{1}	$9.52 \pm 0.21^{\S}$
Latent Approx SDE [46]	$7.55 \pm 0.05^{\S}$	10.50 ± 0.86
ARCTA [9]	$7.62 \pm 0.93^{\ddagger}$	9.92 ± 1.82
NCDSSM [2]	$5.69 \pm 0.01^{\S}$	$4.74 \pm 0.01^{\S}$
SDE matching [3]	$\mathbf{5.20} \pm \mathbf{0.43}^{2}$	4.26 ± 0.35
Latent NSF (ours)	8.62 ± 0.32	$\mathbf{3.41} \pm \mathbf{0.27}$

✔ Best/tied-best performance

✔ Two orders of magnitude faster

Stochastic Moving MNIST (64x64 px)



✔ Comparable performance

✔ 2x faster

Position of Our Work

Method(s)	**Target dynamics**	**Solver-free training**	**Solver-free inference**
Neural ODEs [6]	General ODEs	✗	✗
Neural flows [4]	General ODEs	✓	✓
Score-based diffusion via reverse SDEs/PF-ODEs [47]; flow matching [34]	Pre-defined diffusion SDEs/ODEs	✓	✗
Progressive distillation [44]	Pre-defined diffusion SDEs/ODEs	✗	✓
Consistency models [26, 48]; rectified flows [35]	Pre-defined diffusion SDEs/ODEs	✓	✓
Neural (latent) SDEs [25, 33, 38, 46, 50]	**General Itô SDEs**	✗	✗
ARCTA [9]; SDE matching [3]	**General Itô SDEs**	✓	✗
Neural Stochastic Flows (ours)	**General Itô SDEs**	✓	✓

Summary

- **We proposed Neural Stochastic Flows, which enable solver-free in both training and inference, with closed-form log-densities.**
- **Extended to Latent NSF: a variational state-space model using NSF as the latent transition for partially-observed/high-dimensional data.**

Key Benefits:

- **One-step prediction** for arbitrary time gaps.
- **Speed**: up to two orders of magnitude faster.
- **Stable training** with tractable densities.

Limitations:

- **Chapman–Kolmogorov relation is enforced approximately.**
- Affine-coupling design constrains **expressivity**.

Project page

Paper & Refs.