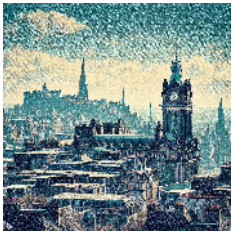
Advanced Topics in Machine Learning (deep generative modelling)

Lecture 10: Diffusion models III

Nikolay Malkin

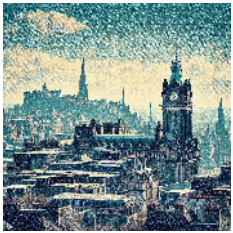
24 March 2026

Dynamics-based generative models in practice



'Edinburgh from Calton Hill, pointillist style'

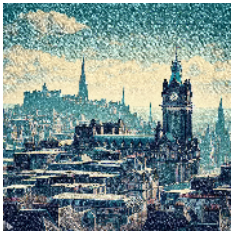
Dynamics-based generative models in practice



'Edinburgh from Calton Hill, pointillist style'

[Janner et al., ICML'22]

Dynamics-based generative models in practice

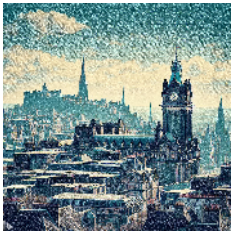


'Edinburgh from Calton Hill, pointillist style'

[Graikos et al., NeurIPS'22]

[Janner et al., ICML'22]

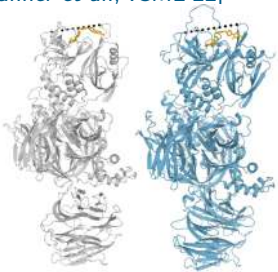
Dynamics-based generative models in practice



'Edinburgh from Calton Hill, pointillist style'

[Graikos et al., NeurIPS'22]

[Janner et al., ICML'22]



AlphaFold 3

Dynamics-based generative models in practice

[Zhang and Gienger, 2024]

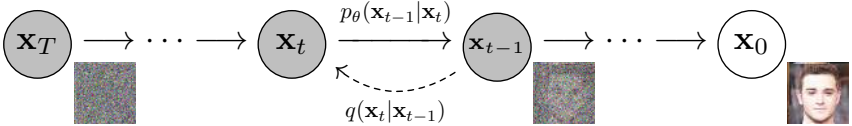
Outline of weeks 8-10

What is a diffusion model, how to understand it from different perspectives, and where to use it in practice:

Outline of weeks 8-10

What is a diffusion model, how to understand it from different perspectives, and where to use it in practice:

- **Lecture 8:** Diffusion models are hierarchical latent variable models / deep VAEs



[Ho et al., 'Denoising diffusion probabilistic models']

Outline of weeks 8-10

What is a diffusion model, how to understand it from different perspectives, and where to use it in practice:

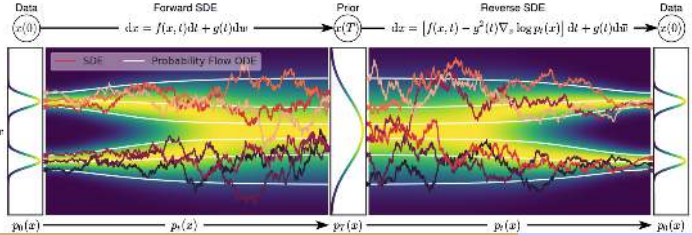
- **Lecture 8:** Diffusion models are hierarchical latent variable models / deep VAEs
- **Lecture 9:** Diffusion models are score-based models

[Song et al., 'Generative modeling by estimating...' blog post]

Outline of weeks 8-10

What is a diffusion model, how to understand it from different perspectives, and where to use it in practice:

- **Lecture 8:** Diffusion models are hierarchical latent variable models / deep VAEs
- **Lecture 9:** Diffusion models are score-based models
- **Lecture 10:** Diffusion models are continuous-time processes



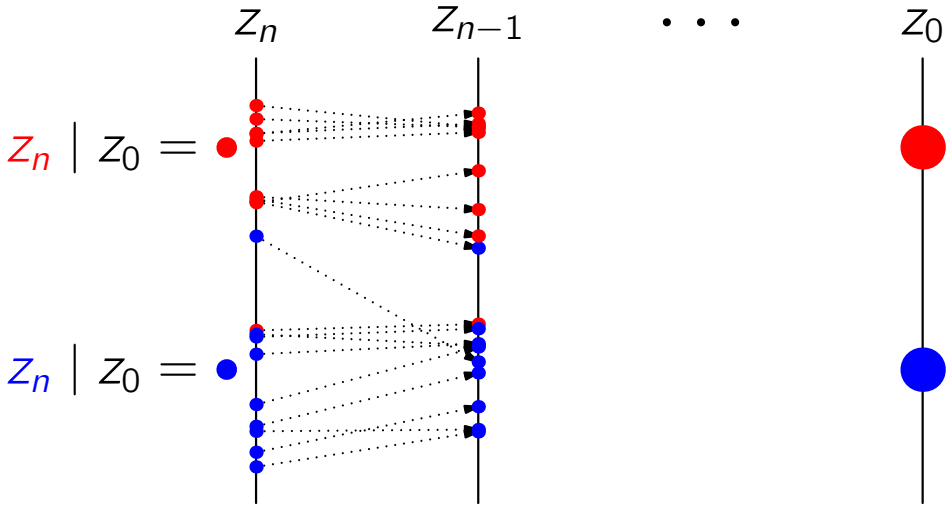
[Song et al., 'Score-based generative modeling. . . ']

Outline of Lecture 10

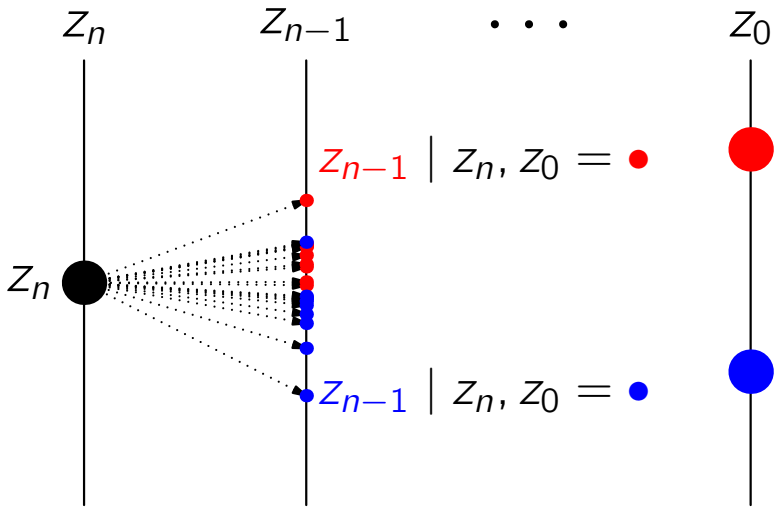
- Review of Lecture 9
- Ordinary and stochastic differential equations
  - Density evolution and reversal
- Score matching and SDEs
- SDEs vs. ODEs & generalisations

- Review of Lecture 9
- Ordinary and stochastic differential equations
  - Density evolution and reversal
- Score matching and SDEs
- SDEs vs. ODEs & generalisations

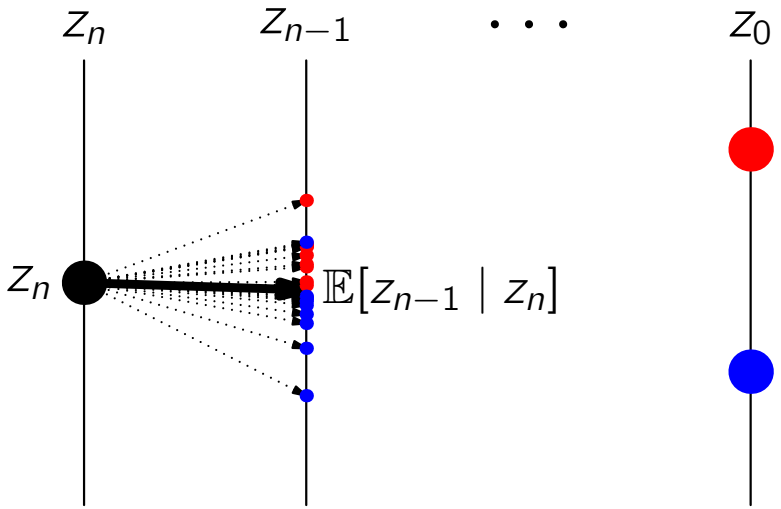
From last time: Simplifying the denoising mean



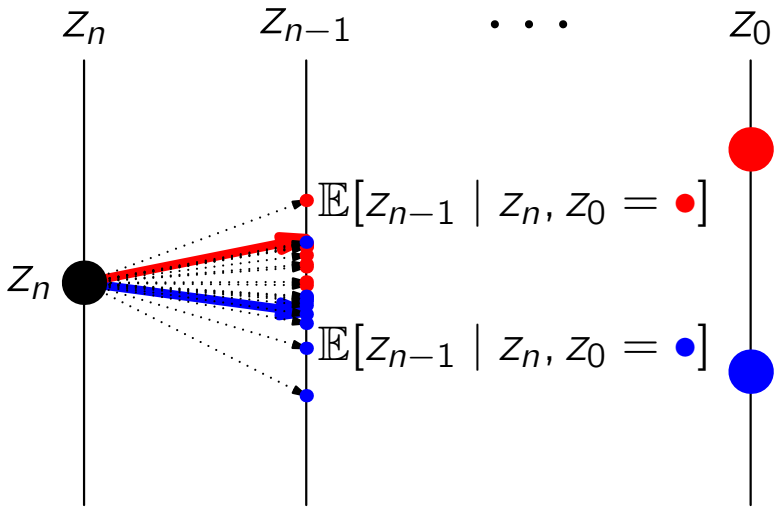
From last time: Simplifying the denoising mean



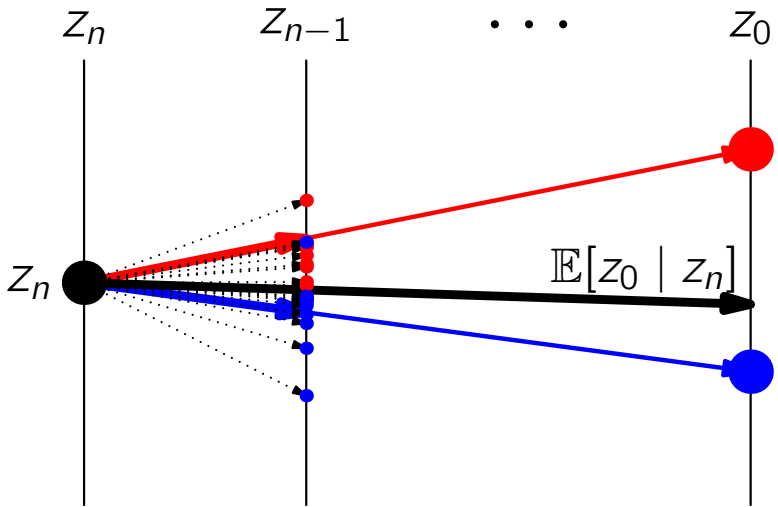
From last time: Simplifying the denoising mean



From last time: Simplifying the denoising mean



From last time: Simplifying the denoising mean



From last time: Denoising is score matching

$$\mathbb{E}[z_{n-1} \mid z_n] = \frac{1}{V_n}\left(V_{n-1} z_n + \sigma_n^2 \mathbb{E}[z_0 \mid z_n]\right)$$

$$\nabla_z \log p(z_n) = \frac{1}{V_n}\left(\mathbb{E}[z_0 \mid z_n] - z_n\right)$$

From last time: Denoising is score matching

$$\mathbb{E}[z_{n-1} \mid z_n] = \frac{1}{V_n}\left(V_{n-1} z_n + \sigma_n^2 \mathbb{E}[z_0 \mid z_n]\right)$$
$$\nabla_z \log p(z_n) = \frac{1}{V_n}\left(\mathbb{E}[z_0 \mid z_n] - z_n\right)$$

- Learning to denoise one step $\longleftrightarrow$ learning to denoise all the way $\longleftrightarrow$ learning the score of the noised data

From last time: Denoising is score matching

$$\mathbb{E}[z_{n-1} \mid z_n] = \frac{1}{V_n}\left(V_{n-1} z_n + \sigma_n^2 \mathbb{E}[z_0 \mid z_n]\right)$$

$$\nabla_z \log p(z_n) = \frac{1}{V_n}\left(\mathbb{E}[z_0 \mid z_n] - z_n\right)$$

- Learning to denoise one step $\longleftrightarrow$ learning to denoise all the way $\longleftrightarrow$ learning the score of the noised data
- Diffusion models are denoising autoencoders

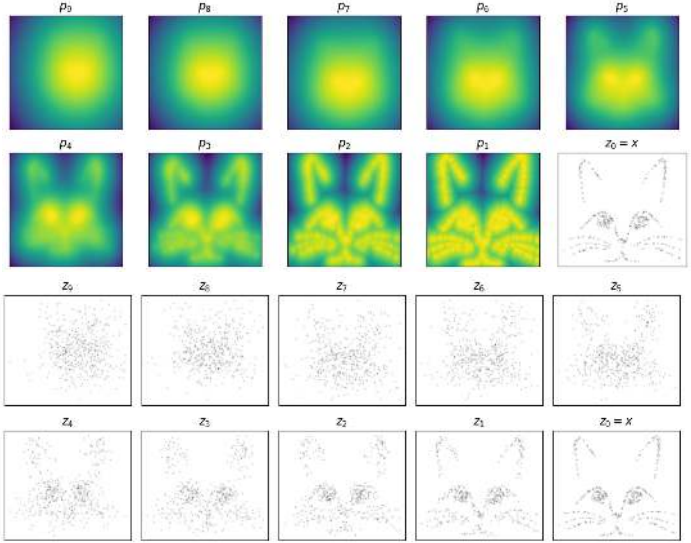
From last time: Denoising is score matching

$$\mathbb{E}[z_{n-1} \mid z_n] = \frac{1}{V_n}\left(V_{n-1} z_n + \sigma_n^2 \mathbb{E}[z_0 \mid z_n]\right)$$

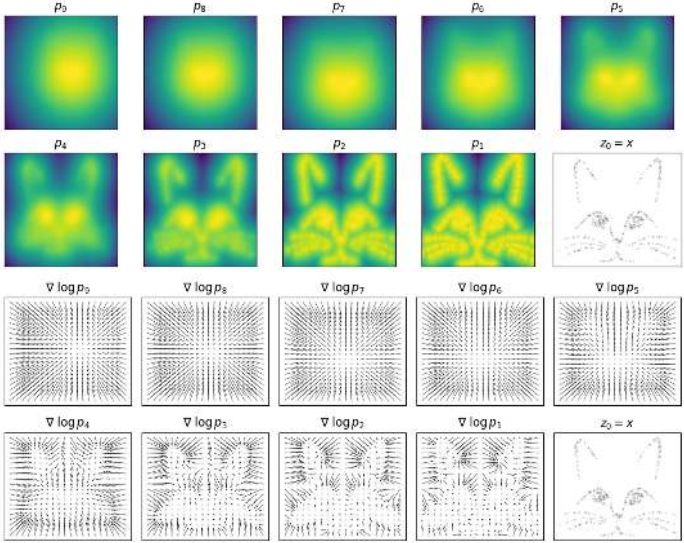
$$\nabla_z \log p(z_n) = \frac{1}{V_n}\left(\mathbb{E}[z_0 \mid z_n] - z_n\right)$$

- Learning to denoise one step $\longleftrightarrow$ learning to denoise all the way $\longleftrightarrow$ learning the score of the noised data
- Diffusion models are denoising autoencoders
- This is important in the continuous-time case (today)...

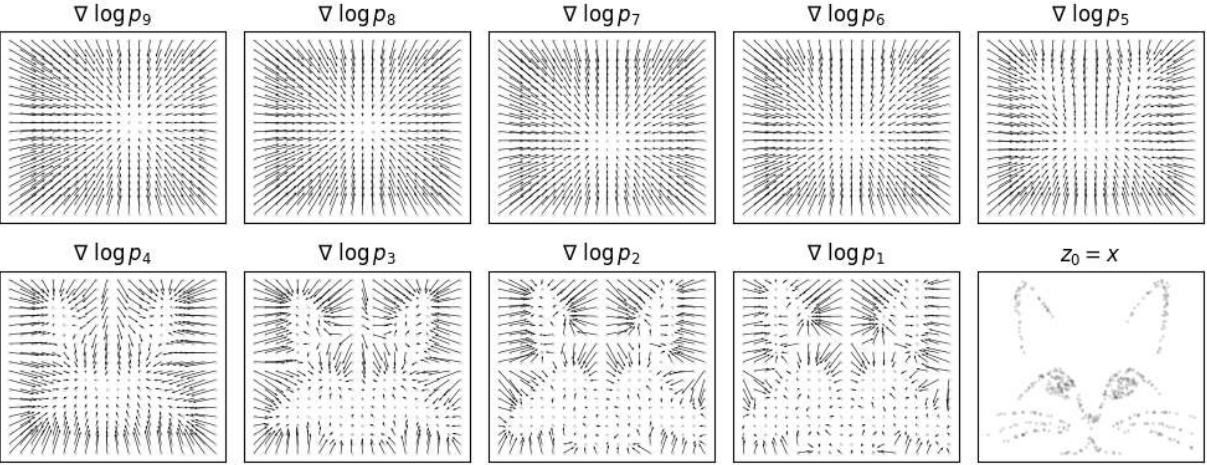
From last time: Scoring a (2D) cat



From last time: Scoring a (2D) cat

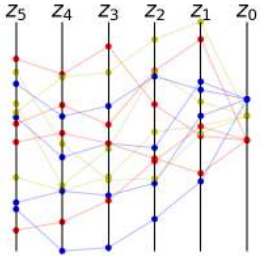


From last time: Scoring a (2D) cat

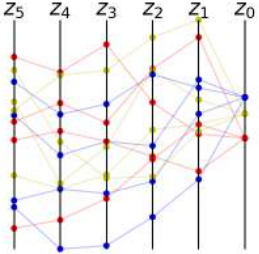


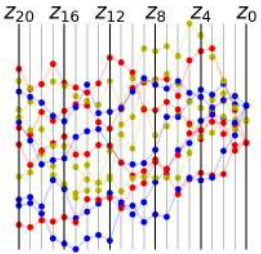
- Review of Lecture 9
- Ordinary and stochastic differential equations
  - Density evolution and reversal
- Score matching and SDEs
- SDEs vs. ODEs & generalisations

Overview of continuous-time models

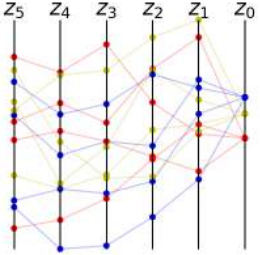


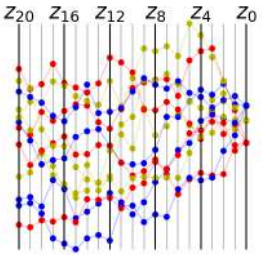
Overview of continuous-time models

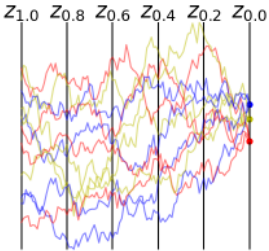




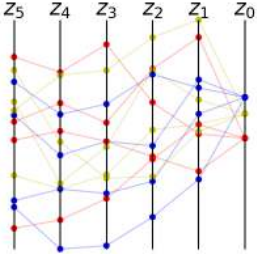
Overview of continuous-time models

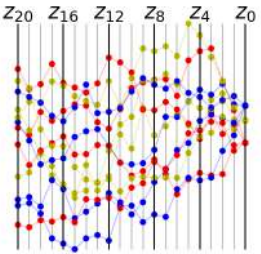


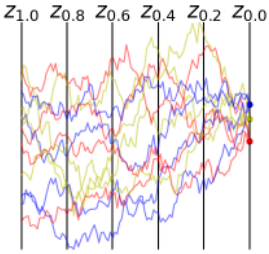


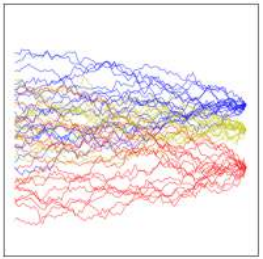


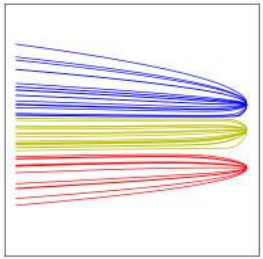
Overview of continuous-time models











Stochastic

Deterministic

Review of ODEs

$$dx_t = f(x_t, t)\, dt$$

Review of ODEs

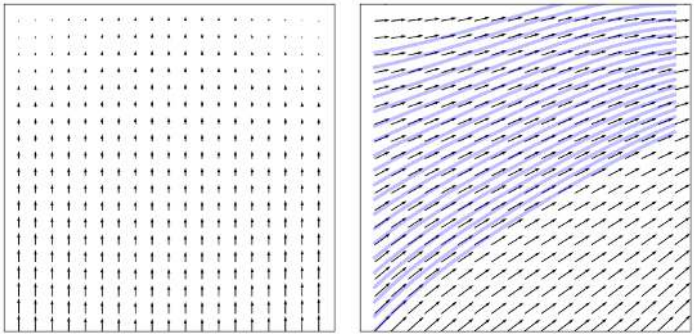
$$dx_t = f(x_t, t)\, dt$$

- $f(x_t, t)$ is the rate of change of x_t
  - Special case: $f(x_t, t) = f(t) \rightsquigarrow x_t = x_0 + \int_0^t f(s)\, ds$

Review of ODEs

$$dx_t = f(x_t, t)\, dt$$

- $f(x_t, t)$ is the rate of change of x_t
  - Special case: $f(x_t, t) = f(t) \rightsquigarrow x_t = x_0 + \int_0^t f(s)\, ds$
- Initial condition $x_0 \rightsquigarrow$ solution x_t as function of t



Review of ODEs

$$dx_t = f(x_t, t)\, dt$$

- $f(x_t, t)$ is the rate of change of x_t
  - Special case: $f(x_t, t) = f(t) \rightsquigarrow x_t = x_0 + \int_0^t f(s)\, ds$
- Initial condition $x_0 \rightsquigarrow$ solution x_t as function of t
- Numerical solution (Euler integration):

$$x_{t+\Delta t} = x_t + f(x_t, t)\Delta t$$

  - More accurate methods: Runge-Kutta, adaptive step size, etc.

Review of ODEs

$$dx_t = f(x_t, t)\, dt$$

- $f(x_t, t)$ is the rate of change of x_t
  - Special case: $f(x_t, t) = f(t) \rightsquigarrow x_t = x_0 + \int_0^t f(s)\, ds$
- Initial condition $x_0 \rightsquigarrow$ solution x_t as function of t
- Numerical solution (Euler integration):

$$x_{t+\Delta t} = x_t + f(x_t, t)\Delta t$$

  - More accurate methods: Runge-Kutta, adaptive step size, etc.
  - Assumptions required for **existence/uniqueness of solutions** and for **convergence of integration schemes**

Review of SDEs

$$dx_t = \underbrace{f(x_t, t)}_{\text{drift}}\, dt + \underbrace{g(t)}_{\text{diffusion}}\, dw_t$$

Review of SDEs

$$dx_t = \underbrace{f(x_t, t)}_{\text{drift}}\, dt + \underbrace{g(t)}_{\text{diffusion}}\, dw_t$$

- Numerical solution (Euler-Maruyama integration):

$$x_{t+\Delta t} = x_t + f(x_t, t)\Delta t + g(t)\sqrt{\Delta t}\,\xi_t, \quad \xi_t \sim \mathcal{N}(0, I_d)$$

($\sqrt{\Delta t}\xi_t = \Delta w_t = w_{t+\Delta t} - w_t$, where w_t is a Brownian motion.)

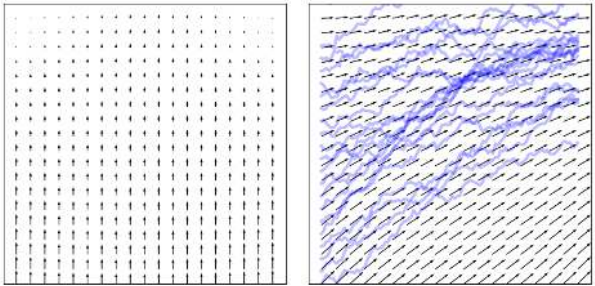
Review of SDEs

$$dx_t = \underbrace{f(x_t, t)}_{\text{drift}}\, dt + \underbrace{g(t)}_{\text{diffusion}}\, dw_t$$

- Numerical solution (Euler-Maruyama integration):

$$x_{t+\Delta t} = x_t + f(x_t, t)\Delta t + g(t)\sqrt{\Delta t}\,\xi_t, \quad \xi_t \sim \mathcal{N}(0, I_d)$$

($\sqrt{\Delta t}\xi_t = \Delta w_t = w_{t+\Delta t} - w_t$, where w_t is a Brownian motion.)



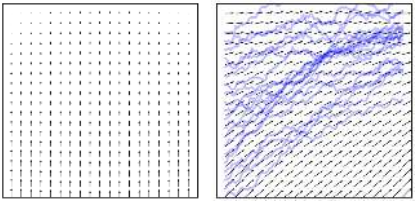
Review of SDEs

$$dx_t = \underbrace{f(x_t, t)}_{\text{drift}}\, dt + \underbrace{g(t)}_{\text{diffusion}}\, dw_t$$

- Numerical solution (Euler-Maruyama integration):

$$x_{t+\Delta t} = x_t + f(x_t, t)\Delta t + g(t)\sqrt{\Delta t}\,\xi_t, \quad \xi_t \sim \mathcal{N}(0, I_d)$$

($\sqrt{\Delta t}\xi_t = \Delta w_t = w_{t+\Delta t} - w_t$, where w_t is a Brownian motion.)



- Why the $\sqrt{\Delta t}$?

Review of SDEs

$$dx_t = \underbrace{f(x_t, t)}_{\text{drift}}\, dt + \underbrace{g(t)}_{\text{diffusion}}\, dw_t$$

- Numerical solution (Euler-Maruyama integration):

$$x_{t+\Delta t} = x_t + f(x_t, t)\Delta t + g(t)\sqrt{\Delta t}\,\xi_t, \quad \xi_t \sim \mathcal{N}(0, I_d)$$

  ($\sqrt{\Delta t}\xi_t = \Delta w_t = w_{t+\Delta t} - w_t$, where w_t is a Brownian motion.)
- Why the $\sqrt{\Delta t}$? Properties of Gaussians, again...
  - Sum of N Gaussians with std $\frac{1}{\sqrt{N}}$ is Gaussian with std 1

Review of SDEs

$$dx_t = \underbrace{f(x_t, t)}_{\text{drift}}\, dt + \underbrace{g(t)}_{\text{diffusion}}\, dw_t$$

- Numerical solution (Euler-Maruyama integration):

$$x_{t+\Delta t} = x_t + f(x_t, t)\Delta t + g(t)\sqrt{\Delta t}\,\xi_t, \quad \xi_t \sim \mathcal{N}(0, I_d)$$

($\sqrt{\Delta t}\xi_t = \Delta w_t = w_{t+\Delta t} - w_t$, where w_t is a Brownian motion.)

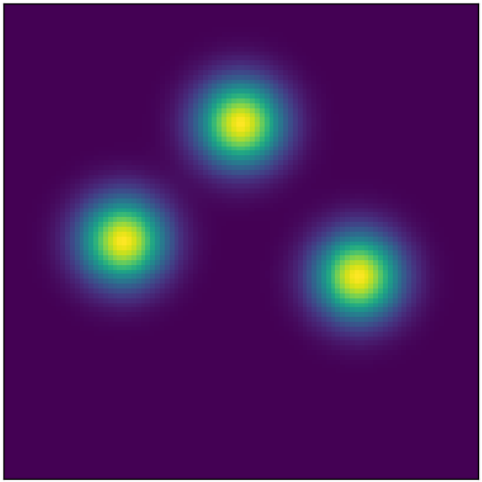
- Why the $\sqrt{\Delta t}$? Properties of Gaussians, again...
  - Sum of N Gaussians with std $\frac{1}{\sqrt{N}}$ is Gaussian with std 1

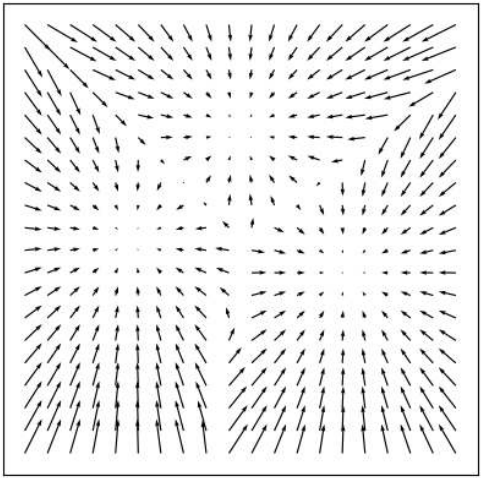
Note that $g(t)$ is a function only of time, not of x_t. Beware! Intuitions may fail you if you try to generalise to the case where g depends on x_t...

- Read about Itô and Stratonovich SDEs if you want to know more

The Langevin SDE

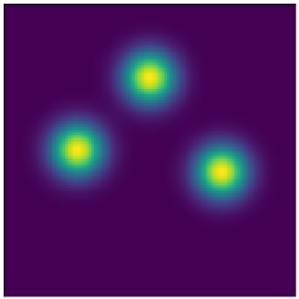
- Recall: the score of a distribution with density p is $\nabla \log p$

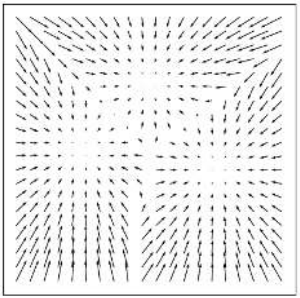




The Langevin SDE

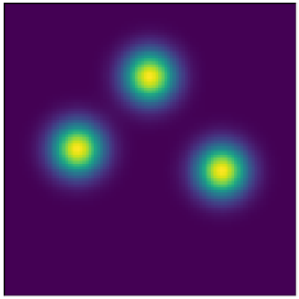
- Recall: the score of a distribution with density p is $\nabla \log p$
- The Langevin SDE: $dx_t = \nabla \log p(x_t)\, dt + \sqrt{2}\, dw_t$

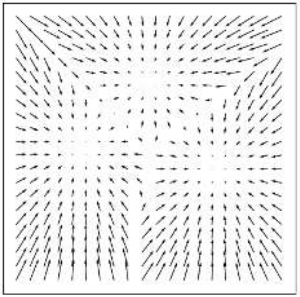




The Langevin SDE

- Recall: the score of a distribution with density p is $\nabla \log p$
- The Langevin SDE: $dx_t = \nabla \log p(x_t)\, dt + \sqrt{2}\, dw_t$





- Under basic assumptions the stationary distribution has density p
  - Proof using Fokker-Planck equation (this next)
  - Many extensions and generalisations used for sampling complex distributions (ULA, MALA, HMC, ...)

Evolution of density for ODEs

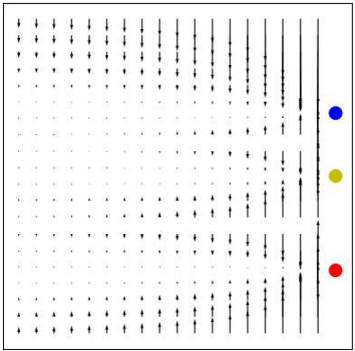
$$dx_t = f(x_t, t)\, dt$$

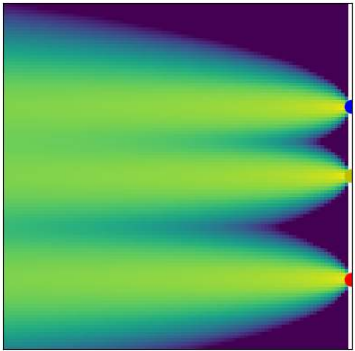
- Initial condition $x_0 \rightsquigarrow$ solution x_t

Evolution of density for ODEs

$$dx_t = f(x_t, t)\, dt$$

- Initial condition $x_0 \rightsquigarrow$ solution x_t
- Density $p_0(x_0) \rightsquigarrow$ marginal densities $p_t(x_t)$





Evolution of density for ODEs

$$dx_t = f(x_t, t)\, dt$$

- Initial condition $x_0 \rightsquigarrow$ solution x_t
- Density $p_0(x_0) \rightsquigarrow$ marginal densities $p_t(x_t)$
- The **continuity equation**:

$$\frac{\partial}{\partial t} p_t(x_t) = -\underbrace{\nabla \cdot (p_t(x_t) f(x_t, t))}$$

  - (Recall: $\nabla \cdot h(x, t) = \sum_i \frac{\partial}{\partial x_i} h_i(x, t)$

Evolution of density for ODEs

$$dx_t = f(x_t, t)\, dt$$

- Initial condition $x_0 \rightsquigarrow$ solution x_t
- Density $p_0(x_0) \rightsquigarrow$ marginal densities $p_t(x_t)$
- The **continuity equation**:

$$\frac{\partial}{\partial t} p_t(x_t) = -\underbrace{\nabla \cdot (p_t(x_t) f(x_t, t))}_{\text{divergence = rate of expansion at } x_t}$$

  - (Recall: $\nabla \cdot h(x,t) = \sum_i \frac{\partial}{\partial x_i} h_i(x,t)$ and the divergence theorem: $\int_V \nabla \cdot h \, dx = \int_{\partial V} h \cdot \vec{n} \, dS$.)

Evolution of density for ODEs

Evolution of density for SDEs

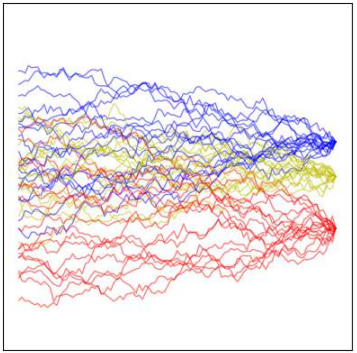
$$dx_t = f(x_t, t)\, dt + g(t)\, dw_t$$

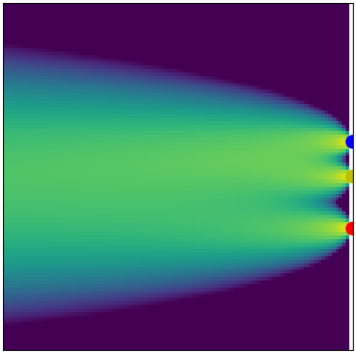
- Initial condition $x_0 \rightsquigarrow$ distribution over solutions x_t

Evolution of density for SDEs

$$dx_t = f(x_t, t)\, dt + g(t)\, dw_t$$

- Initial condition $x_0 \rightsquigarrow$ distribution over solutions x_t
- Density $p_0(x_0) \rightsquigarrow$ marginal densities $p_t(x_t)$





Evolution of density for SDEs

$$dx_t = f(x_t, t)\, dt + g(t)\, dw_t$$

- Initial condition $x_0 \rightsquigarrow$ distribution over solutions x_t
- Density $p_0(x_0) \rightsquigarrow$ marginal densities $p_t(x_t)$
- The **Fokker-Planck(-Kolmogorov) equation**:

$$\frac{\partial}{\partial t} p_t(x_t) = -\nabla \cdot (p_t(x_t) f(x_t, t)) + \frac{g(t)^2}{2} \Delta p_t(x_t)$$

where $\Delta p(x) = \nabla \cdot \nabla p(x) = \sum_i \frac{\partial^2}{\partial x_i^2} p(x)$ is the Laplacian

Time reversal

$$dx_t = f(x_t, t)\, dt$$

Simulating an ODE backwards in time is easy:

- Forward: $x_{t+\Delta t} = x_t + f(x_t, t)\Delta t$
- Backward: $x_{t-\Delta t} = x_t - f(x_t, t)\Delta t$

Time reversal

$$dx_t = f(x_t, t)\, dt$$

Simulating an ODE backwards in time is easy:

- Forward: $x_{t+\Delta t} = x_t + f(x_t, t)\Delta t$
- Backward: $x_{t-\Delta t} = x_t - f(x_t, t)\Delta t$

$$dx_t = f(x_t, t)\, dt + g(t)\, dw_t$$

Simulating an SDE backwards in time is hard:

- Forward: $x_{t+\Delta t} = x_t + f(x_t, t)\Delta t + g(t)\sqrt{\Delta t}\xi_t$
- Backward: ?? $x_{t-\Delta t} = x_t - f(x_t, t)\Delta t + g(t)\sqrt{\Delta t}\xi_t$??

Time reversal

$$dx_t = f(x_t, t)\, dt$$

Simulating an ODE backwards in time is easy:

- Forward: $x_{t+\Delta t} = x_t + f(x_t, t)\Delta t$
- Backward: $x_{t-\Delta t} = x_t - f(x_t, t)\Delta t$

$$dx_t = f(x_t, t)\, dt + g(t)\, dw_t$$

Simulating an SDE backwards in time is hard:

- Forward: $x_{t+\Delta t} = x_t + f(x_t, t)\Delta t + g(t)\sqrt{\Delta t}\xi_t$
- Backward: ?? $x_{t-\Delta t} = x_t - f(x_t, t)\Delta t + g(t)\sqrt{\Delta t}\xi_t$?? **No!**

Time reversal

$$dx_t = f(x_t, t)\, dt$$

Simulating an ODE backwards in time is easy:

- Forward: $x_{t+\Delta t} = x_t + f(x_t, t)\Delta t$
- Backward: $x_{t-\Delta t} = x_t - f(x_t, t)\Delta t$

$$dx_t = f(x_t, t)\, dt + g(t)\, dw_t$$

Simulating an SDE backwards in time is hard:

- Forward: $x_{t+\Delta t} = x_t + f(x_t, t)\Delta t + g(t)\sqrt{\Delta t}\xi_t$
- Backward: ?? $x_{t-\Delta t} = x_t - f(x_t, t)\Delta t + g(t)\sqrt{\Delta t}\xi_t$?? **No!**
- Nelson's identity (1967) (Anderson's identity (1982)): the reverse SDE is

$$dx_t = (f(x_t, t) - g(t)^2 \nabla \log p_t(x_t))\, dt + g(t)\, \underbrace{\overline{dw}_t}_{\text{noise added in reverse sampling}}$$

- Review of Lecture 9
- Ordinary and stochastic differential equations
  - Density evolution and reversal
- Score matching and SDEs
- SDEs vs. ODEs & generalisations

Time reversal for Brownian motion

A simple noising SDE without drift:

$$dx_t = dw_t$$

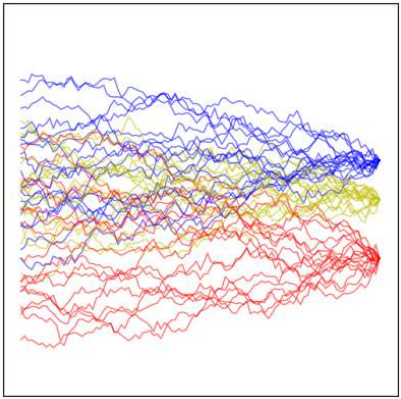
Time reversal for Brownian motion

A simple noising SDE without drift:

$$dx_t = dw_t$$

Initial conditions $p_0 \rightsquigarrow$ marginal densities p_t

- $x_t = x_0 + \mathcal{N}(0, tI_d)$ $([x_t \mid x_0] \sim \mathcal{N}(x_0, tI_d))$
- $p_t(x) = \frac{1}{\|\mathcal{D}\|} \sum_{x_0 \in \mathcal{D}} \mathcal{N}(x; x_0, tI_d)$



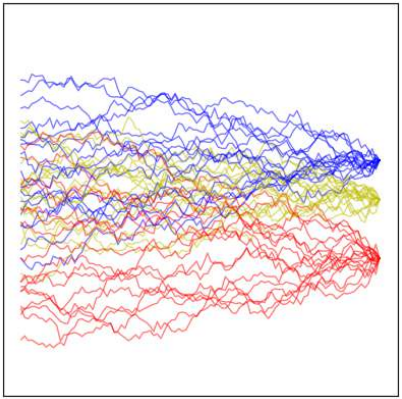
Time reversal for Brownian motion

A simple noising SDE without drift:

$$dx_t = dw_t$$

Initial conditions $p_0 \rightsquigarrow$ marginal densities p_t

- $x_t = x_0 + \mathcal{N}(0, tI_d)$ $([x_t \mid x_0] \sim \mathcal{N}(x_0, tI_d))$
- $p_t(x) = \frac{1}{\|\mathcal{D}\|} \sum_{x_0 \in \mathcal{D}} \mathcal{N}(x; x_0, tI_d)$



Nelson's identity: reverse SDE is

$$dx_t = -\nabla \log p_t(x_t)\, dt + \overline{dw}_t$$

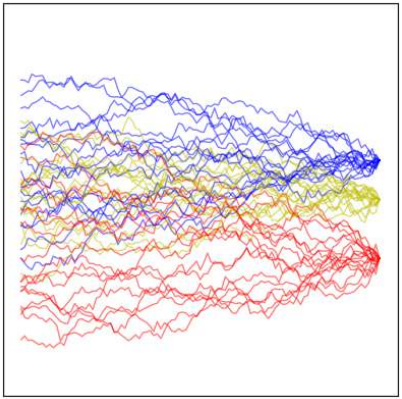
Time reversal for Brownian motion

A simple noising SDE without drift:

$$dx_t = dw_t$$

Initial conditions $p_0 \rightsquigarrow$ marginal densities p_t

- $x_t = x_0 + \mathcal{N}(0, tI_d)$ $([x_t \mid x_0] \sim \mathcal{N}(x_0, tI_d))$
- $p_t(x) = \frac{1}{\|\mathcal{D}\|} \sum_{x_0 \in \mathcal{D}} \mathcal{N}(x; x_0, tI_d)$



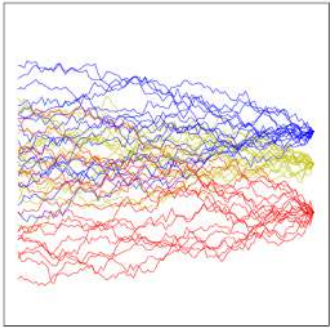
Nelson's identity: reverse SDE is

$$dx_t = -\nabla \log p_t(x_t)\, dt + \overline{dw}_t$$

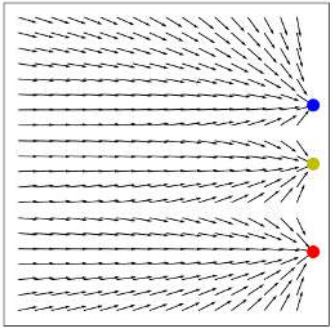
- Learning to reverse noising SDE $\longleftrightarrow$ learning the score of the noised data

Time reversal for Brownian motion

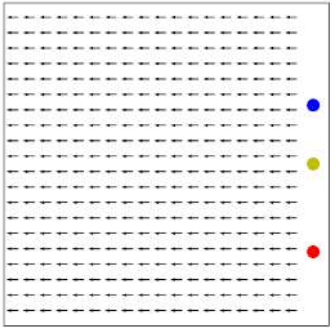
SDE trajectories



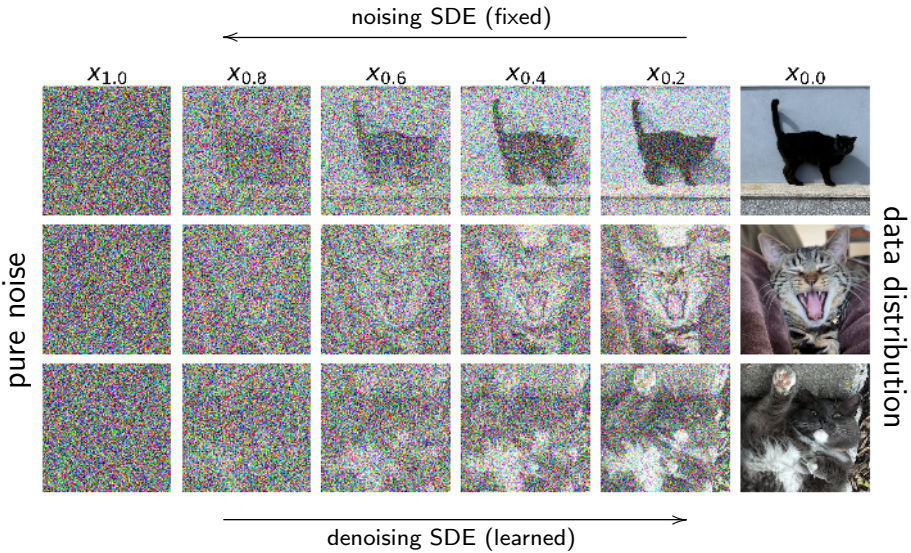
Reverse-time drift



Forward-time drift



SDEs as generative models

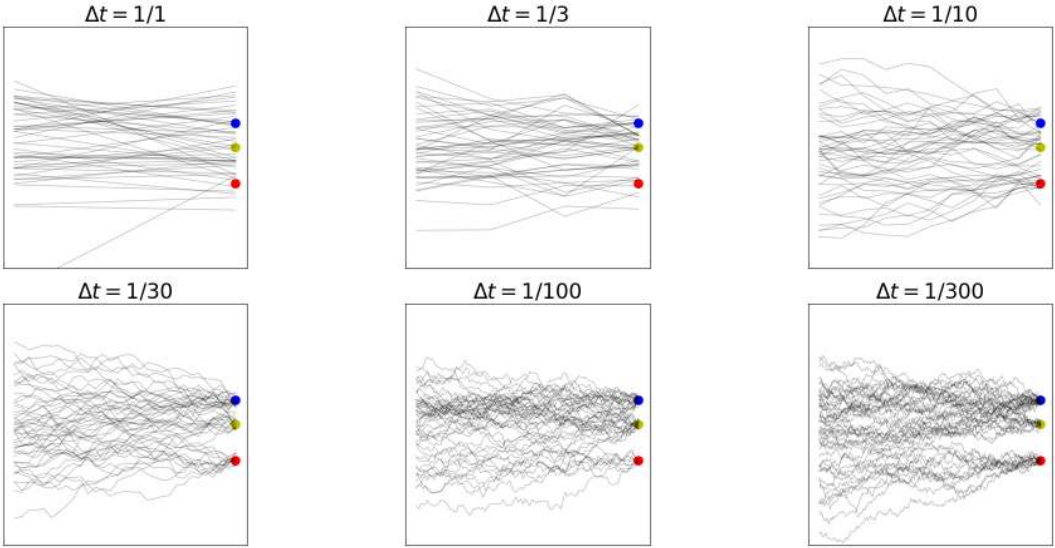


SDEs as generative models

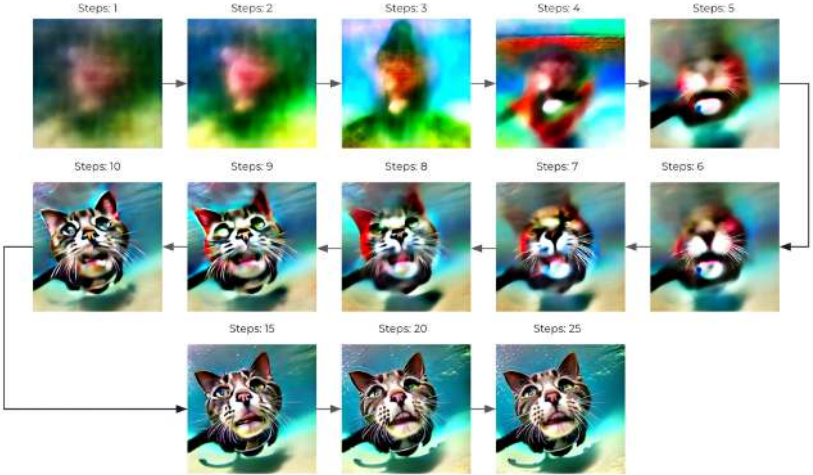
- Learning score of noised data $\longleftrightarrow$ learning reverse SDE drift
- We can fit an approximation of the score for **all** values of t, then sample the reverse SDE with different numbers of steps
  - Instead of conditioning a denoiser / score model on $n \in \{1, \dots, N\}$, we condition it on $t \in [0, T]$
  - Each step is one approximate Langevin step (on a different target density)

[Song et al., 'Generative modeling by estimating...' blog post]

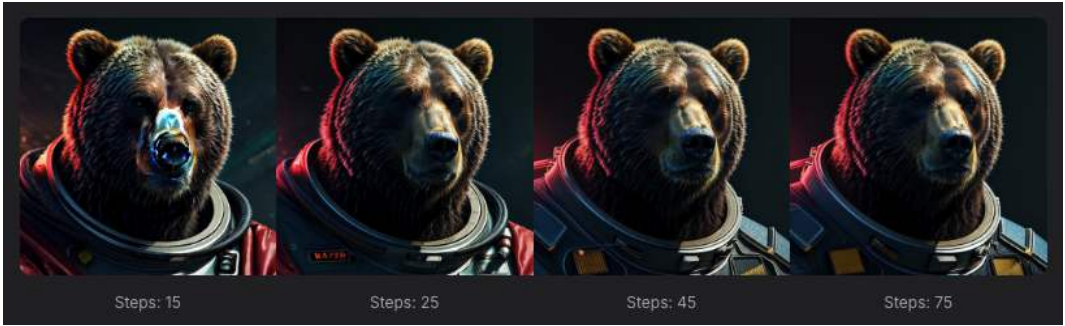
SDEs as generative models



SDEs as generative models



SDEs as generative models



The probability flow ODE

- Forward SDE: $dx_t = f(x_t, t)\, dt + g(t)\, dw_t$
- Reverse SDE: $dx_t = (f(x_t, t) - g(t)^2 \nabla \log p_t(x_t))\, dt + g(t)\, \overline{dw}_t$

The probability flow ODE

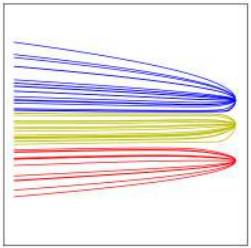
- Forward SDE: $dx_t = f(x_t, t)\, dt + g(t)\, dw_t$
- Reverse SDE: $dx_t = (f(x_t, t) - g(t)^2 \nabla \log p_t(x_t))\, dt + g(t)\, \overline{dw}_t$
- Magic fact: this **probability flow ODE** has the same marginal densities p_t as the SDEs given the same initial conditions:

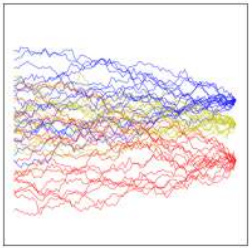
$$dx_t = \left(f(x_t, t) - \frac{1}{2} g(t)^2 \nabla \log p_t(x_t) \right) dt$$

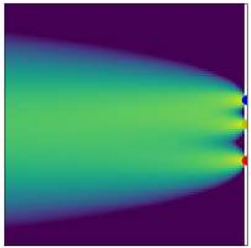
The probability flow ODE

- Forward SDE: $dx_t = f(x_t, t)\, dt + g(t)\, dw_t$
- Reverse SDE: $dx_t = (f(x_t, t) - g(t)^2 \nabla \log p_t(x_t))\, dt + g(t)\, \overline{dw}_t$
- Magic fact: this **probability flow ODE** has the same marginal densities p_t as the SDEs given the same initial conditions:

$$dx_t = \left(f(x_t, t) - \frac{1}{2} g(t)^2 \nabla \log p_t(x_t) \right) dt$$



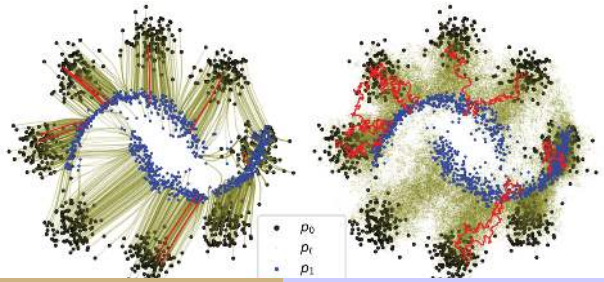




The probability flow ODE

- Forward SDE: $dx_t = f(x_t, t)\,dt + g(t)\,dw_t$
- Reverse SDE: $dx_t = (f(x_t, t) - g(t)^2 \nabla \log p_t(x_t))\,dt + g(t)\,\overline{dw}_t$
- Magic fact: this **probability flow ODE** has the same marginal densities p_t as the SDEs given the same initial conditions:

$$dx_t = \left(f(x_t, t) - \frac{1}{2} g(t)^2 \nabla \log p_t(x_t) \right) dt$$



The probability flow ODE

- Forward SDE: $dx_t = f(x_t, t)\,dt + g(t)\,dw_t$
- Reverse SDE: $dx_t = (f(x_t, t) - g(t)^2 \nabla \log p_t(x_t))\,dt + g(t)\,\overline{dw}_t$
- Magic fact: this **probability flow ODE** has the same marginal densities p_t as the SDEs given the same initial conditions:

$$dx_t = \left(f(x_t, t) - \frac{1}{2} g(t)^2 \nabla \log p_t(x_t) \right) dt$$

  - Proof: F-P equation of SDE = continuity equation of ODE (tutorial exercise)
  - Allows to generate by sampling ODE instead of reverse SDE
  - Also allows density estimation (via Hutchinson trace estimator (tutorial exercise))

- Review of Lecture 9
- Ordinary and stochastic differential equations
  - Density evolution and reversal
- Score matching and SDEs
- SDEs vs. ODEs & generalisations

To noise or not to noise?

Advantages of stochastic modelling:

- Less sensitive to initial integration error, often better samples
- More expressive: non-Gaussian/non-Markovian noise, discrete spaces, ...
- Allows inference of trajectories, not just terminals
- Easier to adapt / fine-tune in certain ways (*e.g.*, using RL)

To noise or not to noise?

Advantages of stochastic modelling:

- Less sensitive to initial integration error, often better samples
- More expressive: non-Gaussian/non-Markovian noise, discrete spaces, ...
- Allows inference of trajectories, not just terminals
- Easier to adapt / fine-tune in certain ways (*e.g.*, using RL)

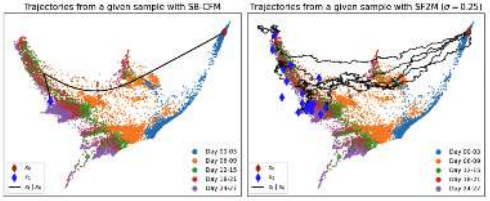


Figure 3: Simulation of trajectories from a given cell on 2D EB data. **Left:** Probability flow ODE trajectory, approximated by SB-CFM (Tong et al., 2024). **Right:** Five SDE trajectories from $[SF]^2M$; more target samples (20) in blue.

[Tong et al., 'Simulation-free Schrödinger bridges', AISTATS'24]

To noise or not to noise?

Advantages of stochastic modelling:

- Less sensitive to initial integration error, often better samples
- More expressive: non-Gaussian/non-Markovian noise, discrete spaces, ...
- Allows inference of trajectories, not just terminals
- Easier to adapt / fine-tune in certain ways (*e.g.*, using RL)

Advantages of deterministic (ODE) modelling:

- Integration in fewer steps (smooth paths, faster sampling, amortised integration)
- Deterministic mapping from x_0 to x_T (latent encoding!)
- Sometimes easier to generalise (flow matching – next slide)
- Easier to adapt in other ways (by modifying the noise distribution)

Flow matching

Algorithms for training the drift of an ODE

$$dx_t = f(x_t, t; \theta)\, dt$$

without assuming a underlying SDE...


Published as a conference paper at ICLR 2023

FLOW MATCHING FOR GENERATIVE MODELING

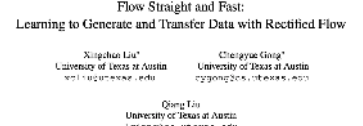
Yaron Lipman[1,2] Ricky T. Q. Chen[1] Heli Ben-Hamu[2] Maximilian Nickel[1] Matt Le[1]
[1]Meta AI (FAIR) [2]Weizmann Institute of Science

Stochastic Interpolants:
A Unifying Framework for Flows and Diffusions

Michael S. Albergo*[1], Nicholas M. Boffi*[2], and Eric Vanden-Eijnden[2]

[1]Center for Cosmology and Particle Physics, New York University
[2]Courant Institute of Mathematical Sciences, New York University

November 7, 2023


Flow Straight and Fast:
Learning to Generate and Transfer Data with Rectified Flow

Xingchao Liu* University of Texas at Austin
Chengyue Gong* University of Texas at Austin
Qiang Liu University of Texas at Austin

Improving and Generalizing Flow-Based Generative Models with Minibatch Optimal Transport

https://pypi.org/project/torchcfm/

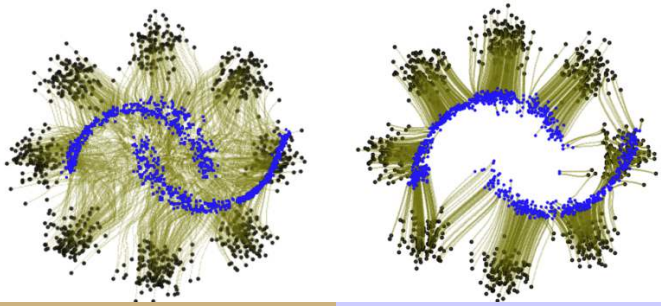
Flow matching

Algorithms for training the drift of an ODE

$$dx_t = f(x_t, t; \theta)\, dt$$

without assuming a underlying SDE...

- Suitable for data-to-data modelling (bridges)
- Connections to optimal transport (straighter paths)



Conclusion and looking forward

- Reversing one noising step
  $\longleftrightarrow$ predicting the mean of clean data given noised data
  $\longleftrightarrow$ learning the score of the noisy data distribution
  $\longleftrightarrow$ learning the reverse SDE drift
- The continuous-time perspective is useful:
  - Varying the number of steps at inference time
  - Sampling by either SDE or probability flow ODE
  - Allows generalisations (*e.g.*, flow matching)

Conclusion and looking forward

- Reversing one noising step
  $\longleftrightarrow$ predicting the mean of clean data given noised data
  $\longleftrightarrow$ learning the score of the noisy data distribution
  $\longleftrightarrow$ learning the reverse SDE drift
- The continuous-time perspective is useful:
  - Varying the number of steps at inference time
  - Sampling by either SDE or probability flow ODE
  - Allows generalisations (*e.g.*, flow matching)

Nothing to look forward to – this was the last content lecture!

Conclusion and looking forward

- Reversing one noising step
  $\longleftrightarrow$ predicting the mean of clean data given noised data
  $\longleftrightarrow$ learning the score of the noisy data distribution
  $\longleftrightarrow$ learning the reverse SDE drift
- The continuous-time perspective is useful:
  - Varying the number of steps at inference time
  - Sampling by either SDE or probability flow ODE
  - Allows generalisations (*e.g.*, flow matching)

Nothing to look forward to – this was the last content lecture!
Next time, semi-structured review and questions.