GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN
IN PUBLICA COMMODA SEIT 1737

Flow Matching

Paul Wollenhaupt

Optimal Transport Seminar

May 7th, 2024

Generative Models

Lipman et al. 2023; Brown et al. 2020; Bubeck et al. 2023

Generative Models

- Given a dataset $\{x_i\}_{i=1}^{N} \overset{\text{i.i.d.}}{\sim} p_x$

Lipman et al. 2023; Brown et al. 2020; Bubeck et al. 2023

Generative Models

- Given a dataset $\{x_i\}_{i=1}^{N} \overset{\text{i.i.d.}}{\sim} p_x$
- Generate new datapoints $x_{n+1} \sim p_x$

Lipman et al. 2023; Brown et al. 2020; Bubeck et al. 2023

Generative Models

- Given a dataset $\{x_i\}_{i=1}^{N} \overset{\text{i.i.d.}}{\sim} p_x$
- Generate new datapoints $x_{n+1} \sim p_x$
- Better understanding of data

Lipman et al. 2023; Brown et al. 2020; Bubeck et al. 2023

Generative Models

- Given a dataset $\{x_i\}_{i=1}^{N} \overset{\text{i.i.d.}}{\sim} p_x$
- Generate new datapoints $x_{n+1} \sim p_x$
- Better understanding of data
- Can be very versatile

Lipman et al. 2023; Brown et al. 2020; Bubeck et al. 2023

Generative Models

- Given a dataset $\{x_i\}_{i=1}^N \overset{\text{i.i.d.}}{\sim} p_x$
- Generate new datapoints $x_{n+1} \sim p_x$
- Better understanding of data
- Can be very versatile
  - ChatGPT

Lipman et al. 2023; Brown et al. 2020; Bubeck et al. 2023

Generative Models

- Given a dataset $\{x_i\}_{i=1}^{N} \overset{\text{i.i.d.}}{\sim} p_x$
- Generate new datapoints $x_{n+1} \sim p_x$
- Better understanding of data
- Can be very versatile
  - ChatGPT
  - Text to Image

Lipman et al. 2023; Brown et al. 2020; Bubeck et al. 2023

Generative Models

- Given a dataset $\{x_i\}_{i=1}^{N} \overset{\text{i.i.d.}}{\sim} p_x$
- Generate new datapoints $x_{n+1} \sim p_x$
- Better understanding of data
- Can be very versatile
  - ChatGPT
  - Text to Image

Lipman et al. 2023; Brown et al. 2020; Bubeck et al. 2023

Generative Models

- Given a dataset $\{x_i\}_{i=1}^{N} \overset{\text{i.i.d.}}{\sim} p_x$
- Generate new datapoints $x_{n+1} \sim p_x$
- Better understanding of data
- Can be very versatile
  - ChatGPT
  - Text to Image $\rightarrow$ **Flow Matching**

Lipman et al. 2023; Brown et al. 2020; Bubeck et al. 2023

Normalising Flows

Rezende and Mohamed 2016

Normalising Flows

- Start with known distribution $z \sim p_z$

Rezende and Mohamed 2016

Normalising Flows

- Start with known distribution $z \sim p_z$
- Apply diffeomorphism f_θ to z

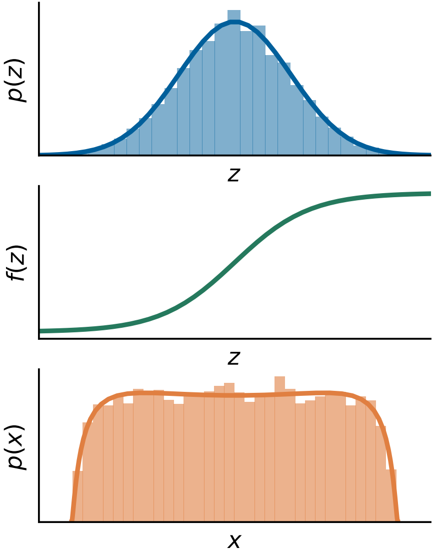
$$p_\theta(x) = p_z(f_\theta^{-1}(x)) \cdot \left| \det \frac{\partial f_\theta^{-1}(x)}{\partial x} \right|$$

Rezende and Mohamed 2016

Normalising Flows

- Start with known distribution $z \sim p_z$
- Apply diffeomorphism f_θ to z

$$p_\theta(x) = p_z(f_\theta^{-1}(x)) \cdot \left| \det \frac{\partial f_\theta^{-1}(x)}{\partial x} \right|$$



Rezende and Mohamed 2016

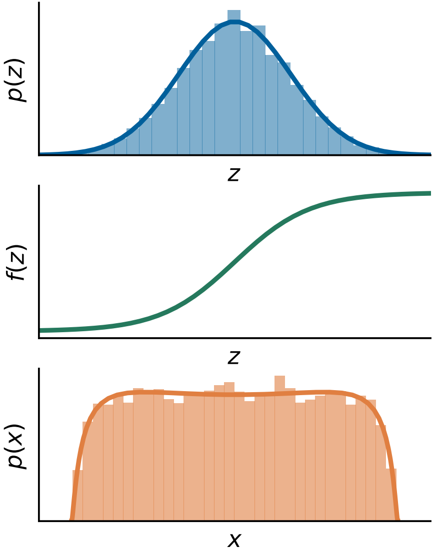
Normalising Flows

- Start with known distribution $z \sim p_z$
- Apply diffeomorphism f_θ to z

$$p_\theta(x) = p_z(f_\theta^{-1}(x)) \cdot \left| \det \frac{\partial f_\theta^{-1}(x)}{\partial x} \right|$$

- Maximize likelihood of data

$$\theta^* = \arg\max_\theta \sum_{i=1}^{N} \log p_\theta(x_i)$$



Rezende and Mohamed 2016

Continuous Normalizing Flows

Chen et al. 2019; Grathwohl et al. 2018; Hutchinson 1990

Continuous Normalizing Flows

- Define the transformation as an ODE

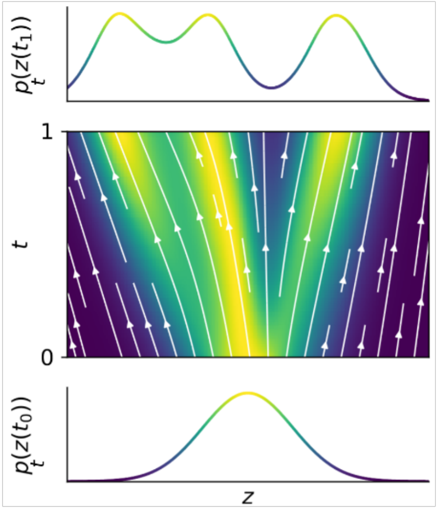
$$x = z(t_1) = \int_{t_0}^{t_1} v_\theta(z(t), t)\, \mathrm{d}t$$

Chen et al. 2019; Grathwohl et al. 2018; Hutchinson 1990

Continuous Normalizing Flows

- Define the transformation as an ODE

$$x = z(t_1) = \int_{t_0}^{t_1} v_\theta(z(t), t)\, \mathrm{d}t$$



Chen et al. 2019; Grathwohl et al. 2018; Hutchinson 1990

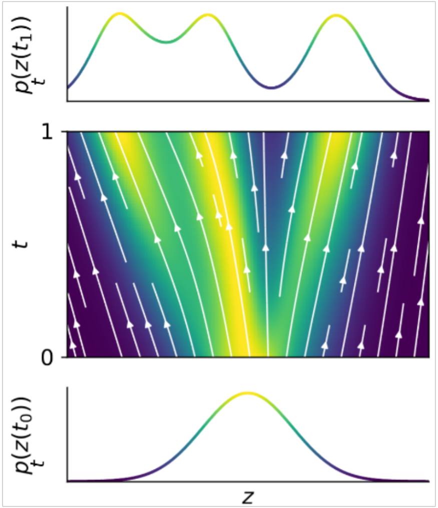
Continuous Normalizing Flows

- Define the transformation as an ODE

$$x = z(t_1) = \int_{t_0}^{t_1} v_\theta(z(t), t)\, \mathrm{d}t$$

- Instantaneous change of density

$$\frac{\partial \log p_t(z(t))}{\partial t} = -\nabla \cdot v_\theta(z(t), t)$$



Chen et al. 2019; Grathwohl et al. 2018; Hutchinson 1990

Continuous Normalizing Flows

- Define the transformation as an ODE

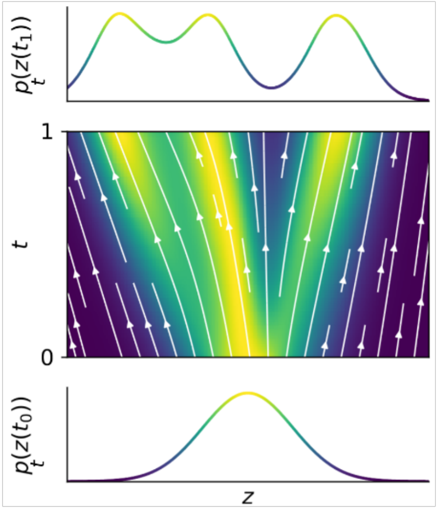
$$x = z(t_1) = \int_{t_0}^{t_1} v_\theta(z(t), t)\, \mathrm{d}t$$

- Instantaneous change of density

$$\frac{\partial \log p_t(z(t))}{\partial t} = -\nabla \cdot v_\theta(z(t), t)$$

- Solve the ODE for $\log p_t(z(t_1))$

$$\log p_t(z(t_0)) - \int_{t_0}^{t_1} \nabla \cdot v_\theta(z(t), t)\, \mathrm{d}t$$



Chen et al. 2019; Grathwohl et al. 2018; Hutchinson 1990

Diffusion Models - Overview

- Gradually add normal noise to data

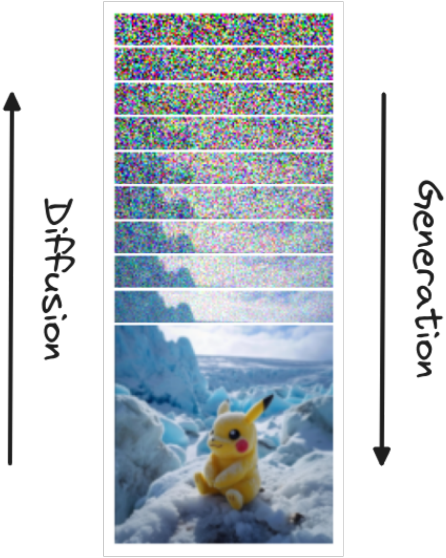
$$\mathrm{d}x = f(x, t)\,\mathrm{d}t + g(t)\,\mathrm{d}\omega$$

Sohl-Dickstein et al. 2015; Ho, Jain, and Abbeel 2020; Song et al. 2021; Anderson 1982

Diffusion Models - Overview

- Gradually add normal noise to data

$$dx = f(x, t)\,dt + g(t)\,d\omega$$



Sohl-Dickstein et al. 2015; Ho, Jain, and Abbeel 2020; Song et al. 2021; Anderson 1982

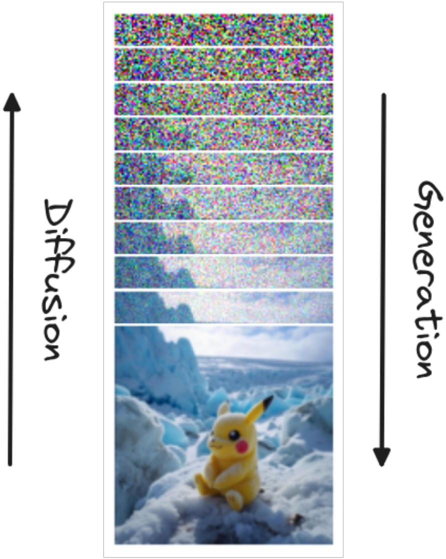
Diffusion Models - Overview

- Gradually add normal noise to data

$$dx = f(x,t)\,dt + g(t)\,d\omega$$

- Reverse the diffusion process

$$dx = \big(f(x,t) - g^2(t)\nabla_x \log p_t(x)\big)\,dt + g(t)\,d\bar{\omega}$$



Sohl-Dickstein et al. 2015; Ho, Jain, and Abbeel 2020; Song et al. 2021; Anderson 1982

Diffusion Models - Overview

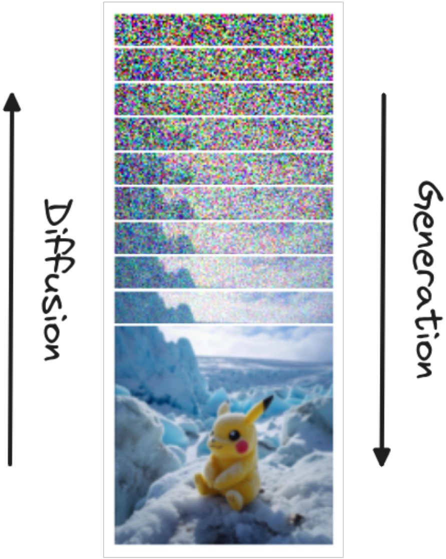
- Gradually add normal noise to data

$$\mathrm{d}x = f(x,t)\,\mathrm{d}t + g(t)\,\mathrm{d}\omega$$

- Reverse the diffusion process

$$\mathrm{d}x = \left(f(x,t) - g^2(t)\nabla_x \log p_t(x)\right)\mathrm{d}t \\ + g(t)\,\mathrm{d}\bar{\omega}$$

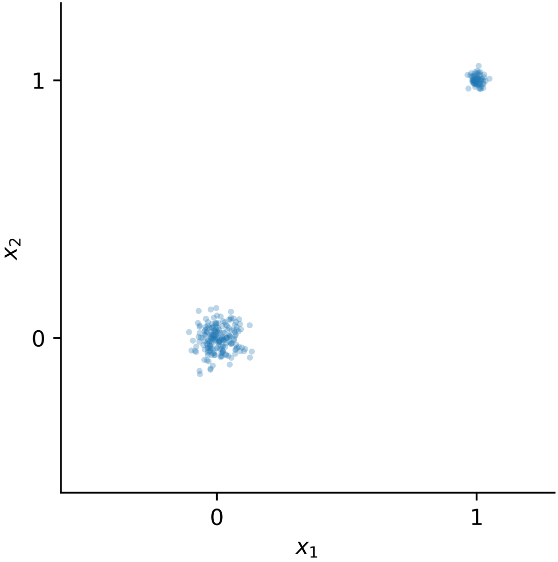
- Learn the score function $\nabla_x \log p_t(x)$



Sohl-Dickstein et al. 2015; Ho, Jain, and Abbeel 2020; Song et al. 2021; Anderson 1982

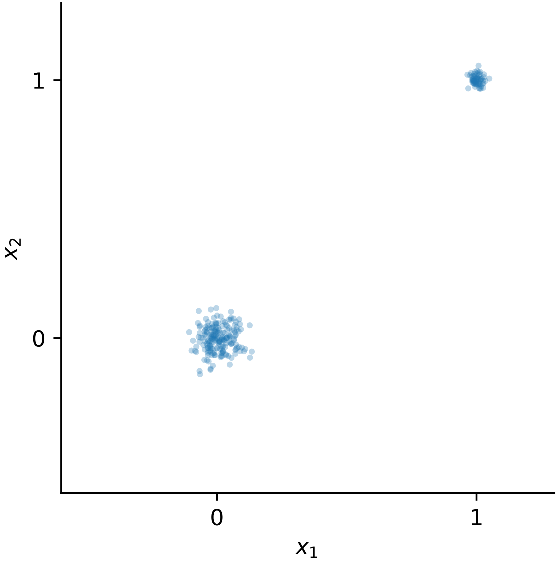
Toy Dataset Intuition

Toy Dataset Intuition



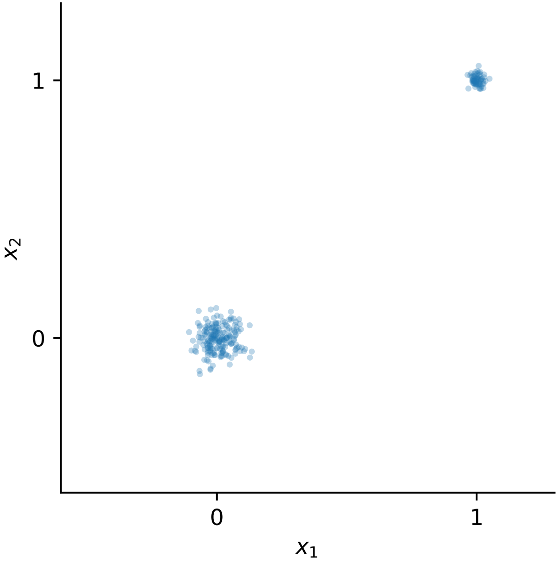
Toy Dataset Intuition

- Two modes



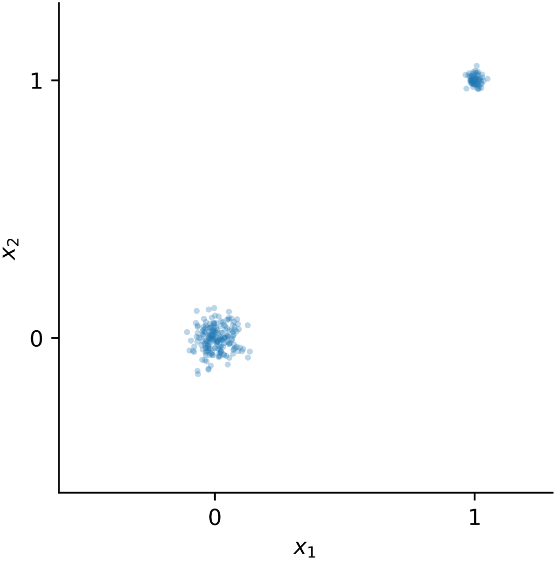
Toy Dataset Intuition

- Two modes
- Imbalanced



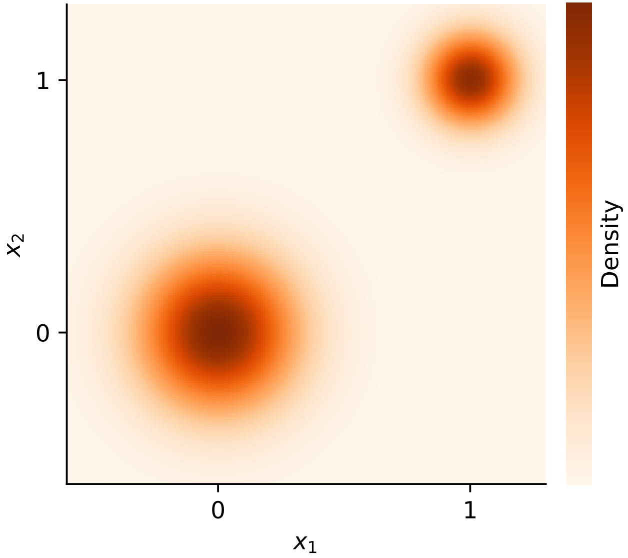
Toy Dataset Intuition

- Two modes
- Imbalanced
- Far apart



Density

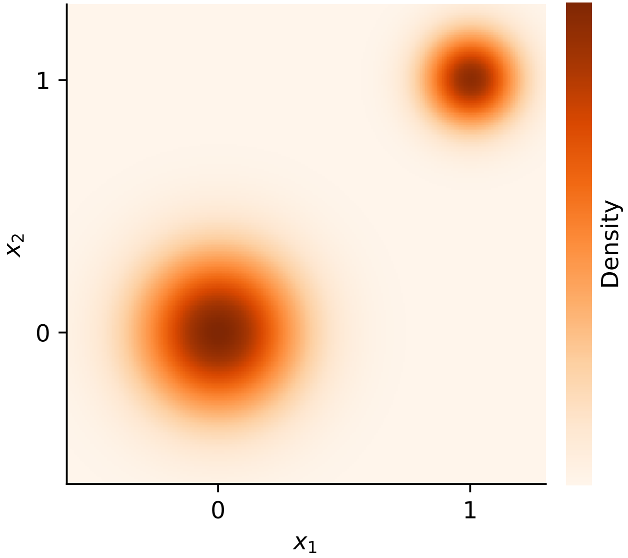
Density



Density

- #Datapoints / Area

$$p(x)$$

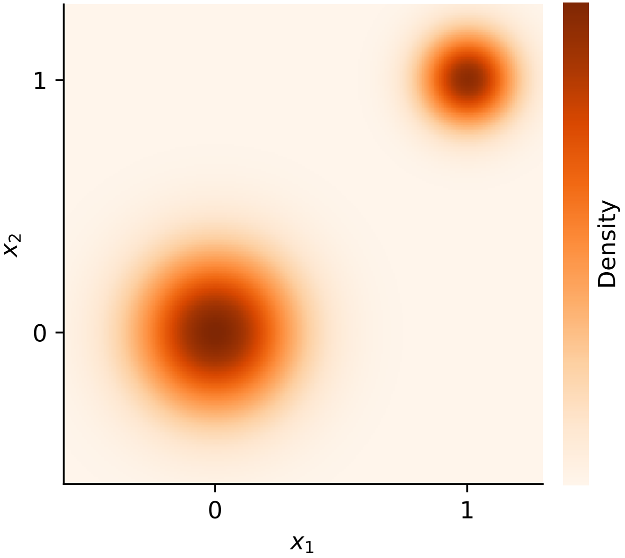


Density

- #Datapoints / Area

$$p(x)$$

- New datapoints?



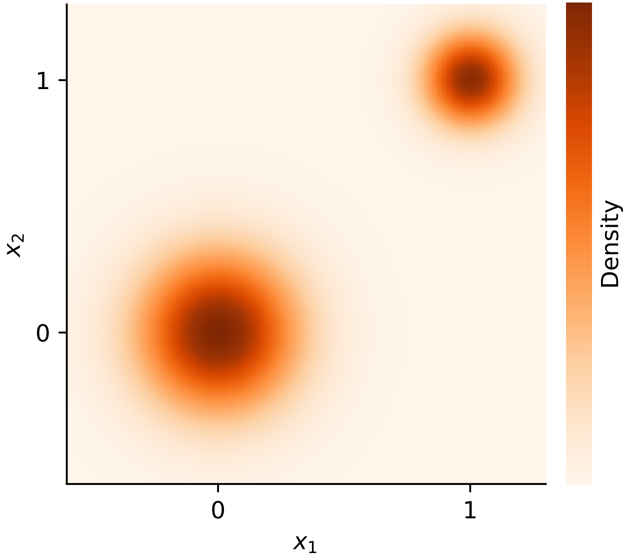
Density

- #Datapoints / Area

$$p(x)$$

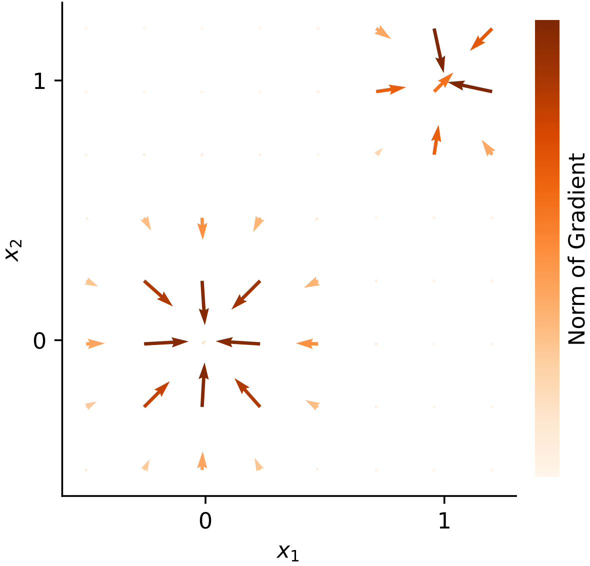
- New datapoints?

▷ Optimize density



Solution: Gradient Ascent

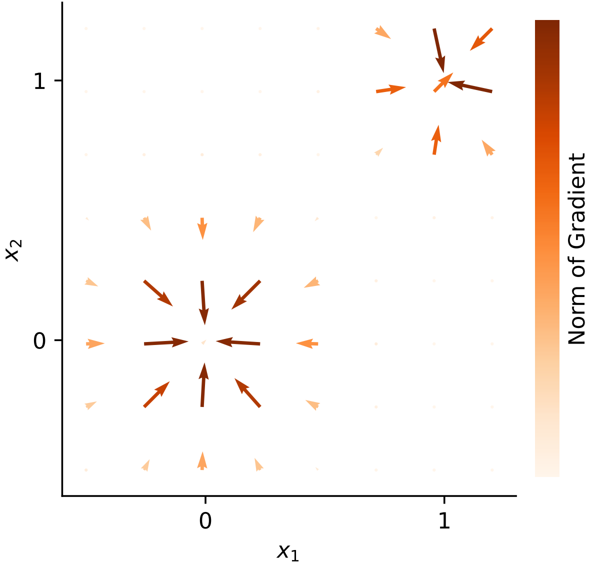
Solution: Gradient Ascent



Solution: Gradient Ascent

- Steepest ascent direction

$$\nabla_x p(x)$$

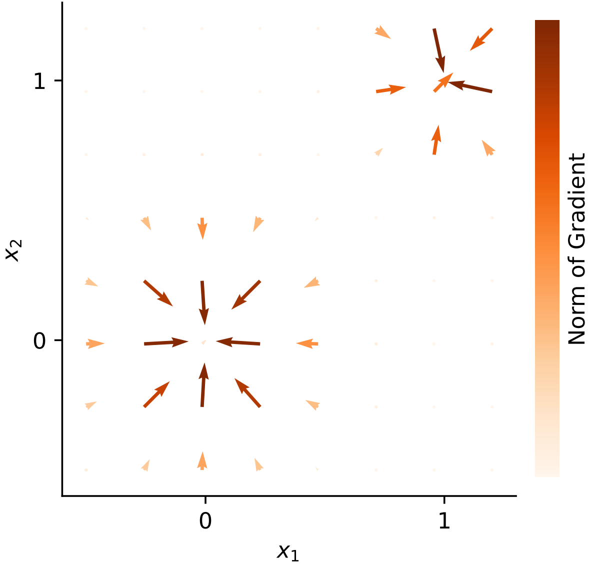


Solution: Gradient Ascent

- Steepest ascent direction

$$\nabla_x p(x)$$

- Take a small Step



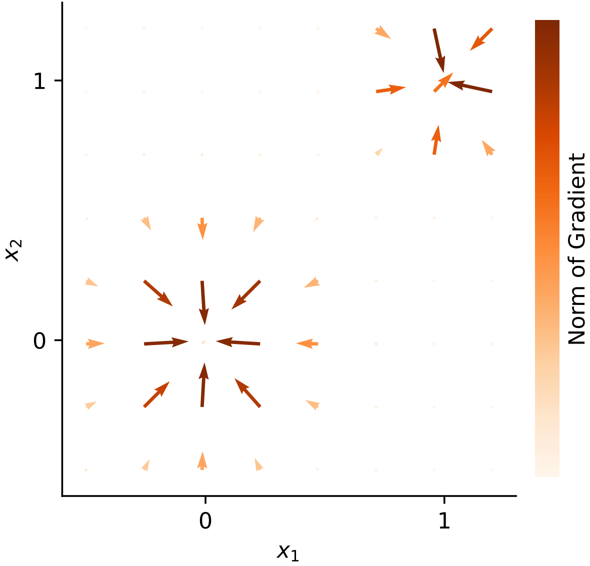
Solution: Gradient Ascent

- Steepest ascent direction

$$\nabla_x p(x)$$

- Take a small Step

- Repeat until converged

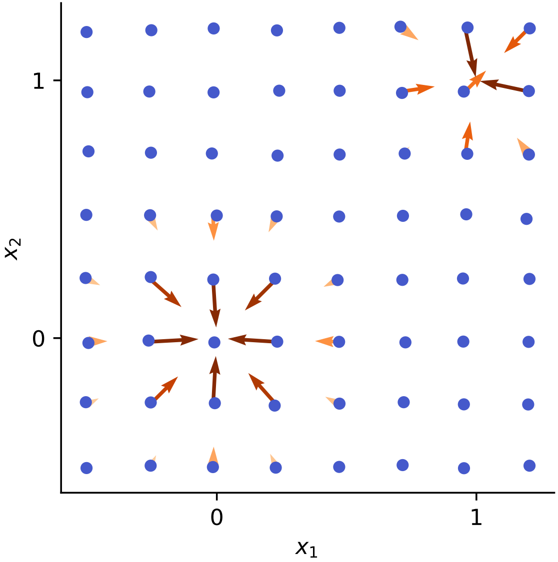


Solution: Gradient Ascent

- Steepest ascent direction

$$\nabla_x p(x)$$

- Take a small Step
- Repeat until converged

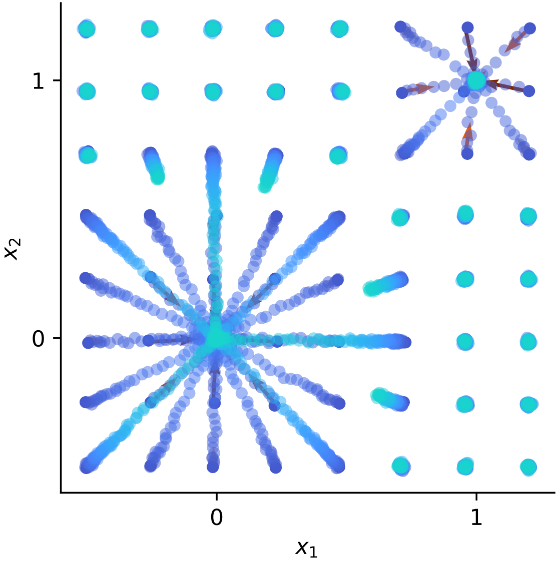


Solution: Gradient Ascent

- Steepest ascent direction

$$\nabla_x p(x)$$

- Take a small Step
- Repeat until converged

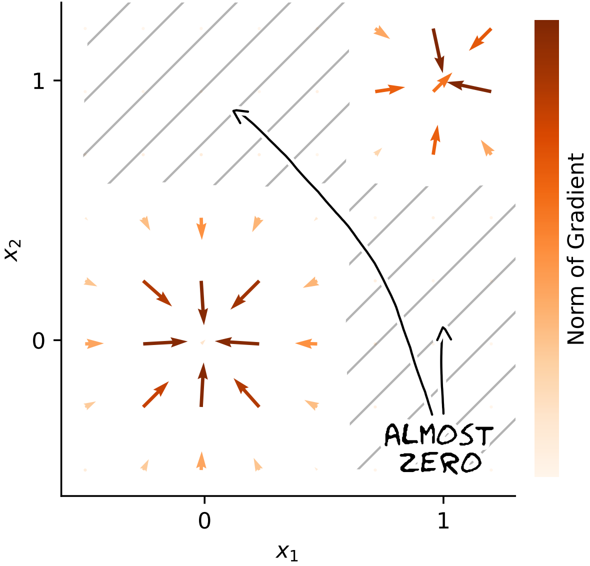


Solution: Gradient Ascent

- Steepest ascent direction

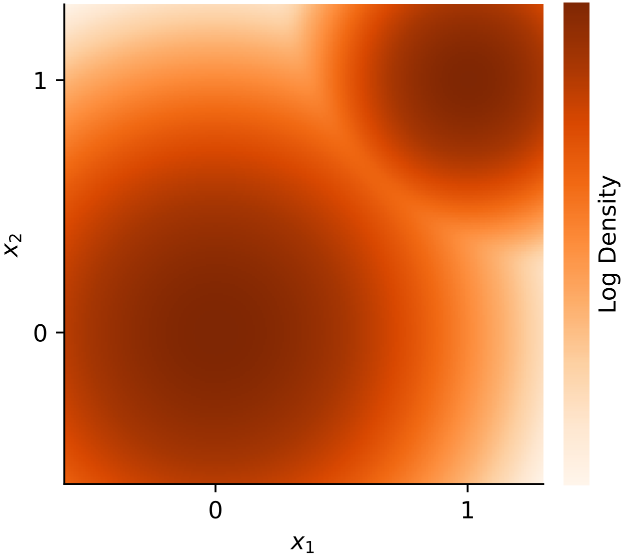
$$\nabla_x p(x)$$

- Take a small Step
- Repeat until converged



Solution: Logarithm

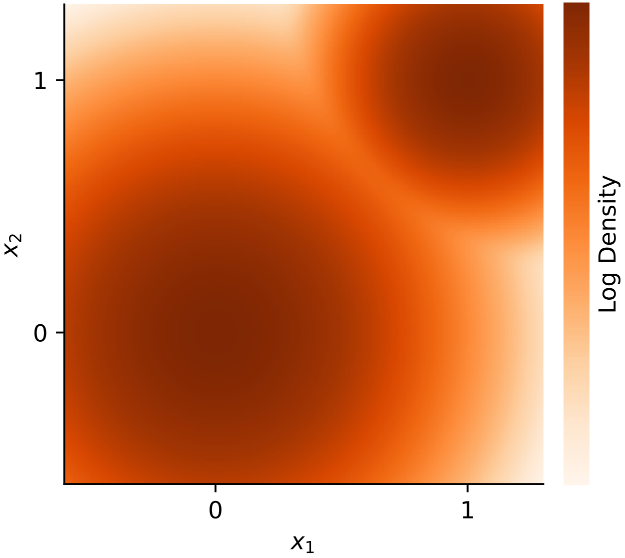
Solution: Logarithm



Solution: Logarithm

- Better for small numbers

$$\log p(x)$$



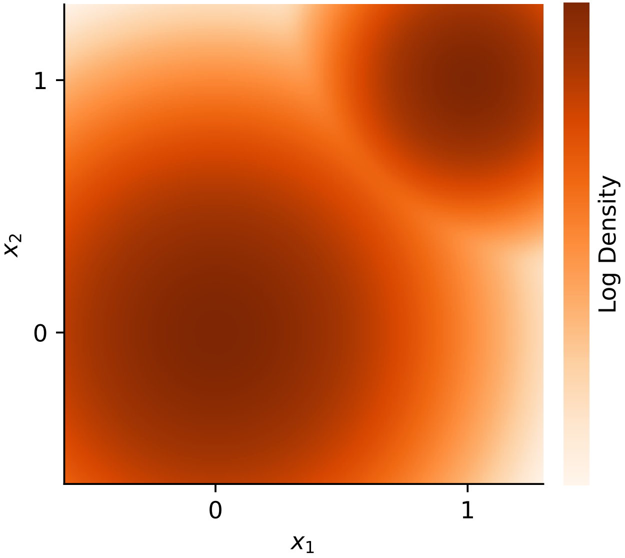
Solution: Logarithm

- Better for small numbers

$$\log p(x)$$

- Better behaved gradient

$$\nabla_x \log p(x)$$



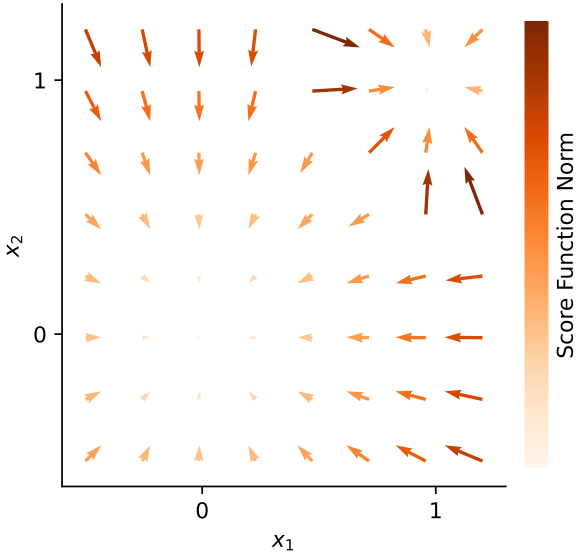
Solution: Logarithm

- Better for small numbers

$$\log p(x)$$

- Better behaved gradient

$$\nabla_x \log p(x)$$



Solution: Logarithm

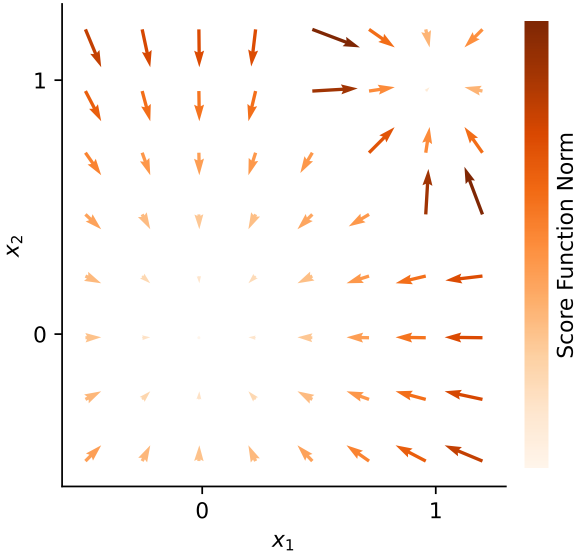
- Better for small numbers

$$\log p(x)$$

- Better behaved gradient

$$\nabla_x \log p(x)$$

▷ Score Function



Solution: Logarithm

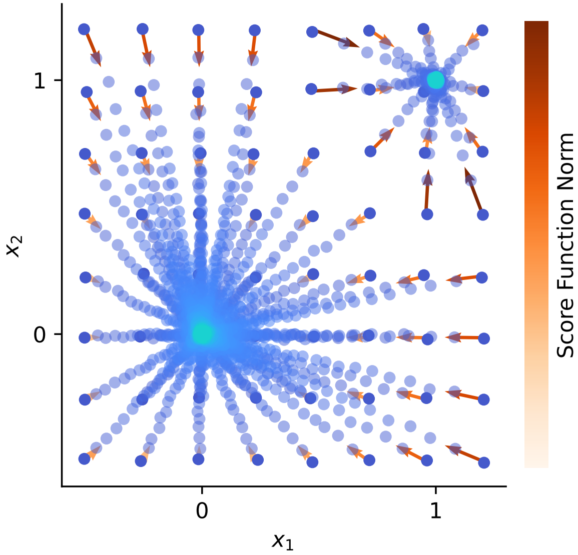
- Better for small numbers

$$\log p(x)$$

- Better behaved gradient

$$\nabla_x \log p(x)$$

▷ Score Function



Solution: Logarithm

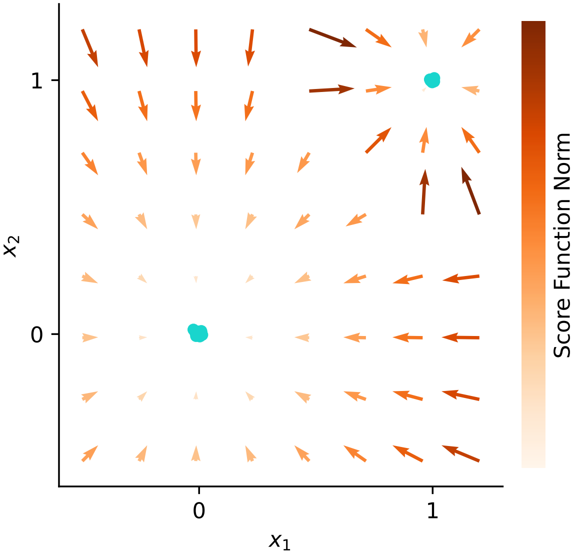
- Better for small numbers

$$\log p(x)$$

- Better behaved gradient

$$\nabla_x \log p(x)$$

▷ Score Function



Solution: Add Noise

Solution: Add Noise

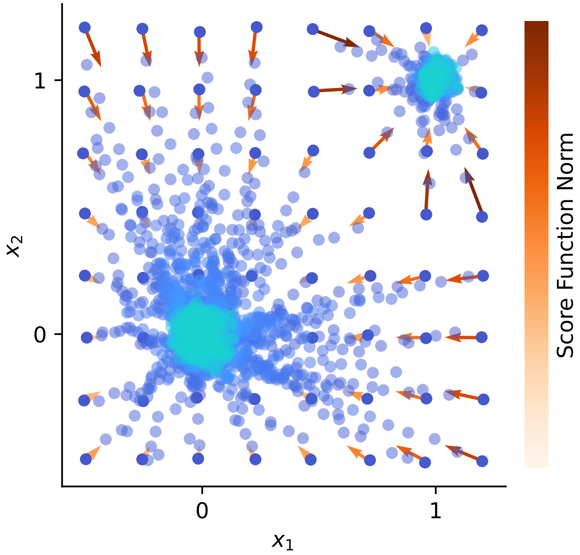
- Add normal noise

Solution: Add Noise

- Add normal noise
- In each step

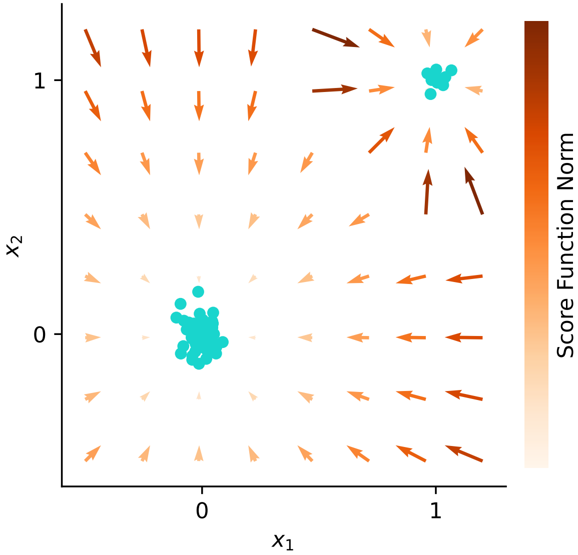
Solution: Add Noise

- Add normal noise
- In each step



Solution: Add Noise

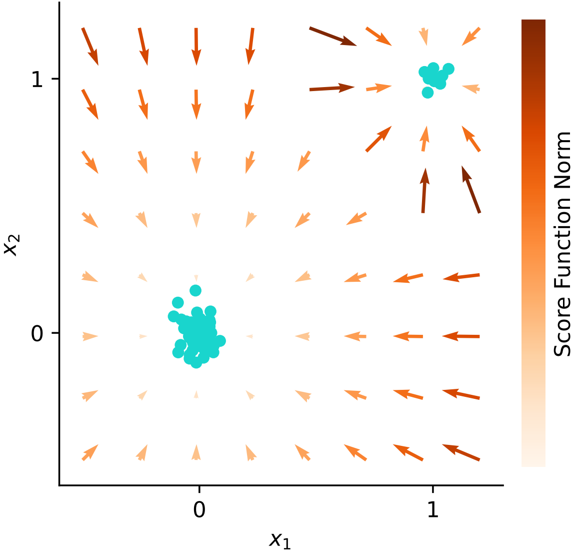
- Add normal noise
- In each step



Solution: Add Noise

- Add normal noise
- In each step

▷ Langevin Dynamics



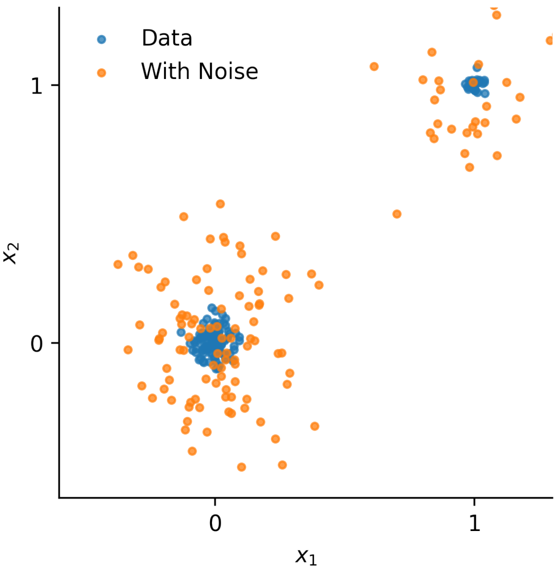
Score Matching

Score Matching

- Noisy Dataset

Score Matching

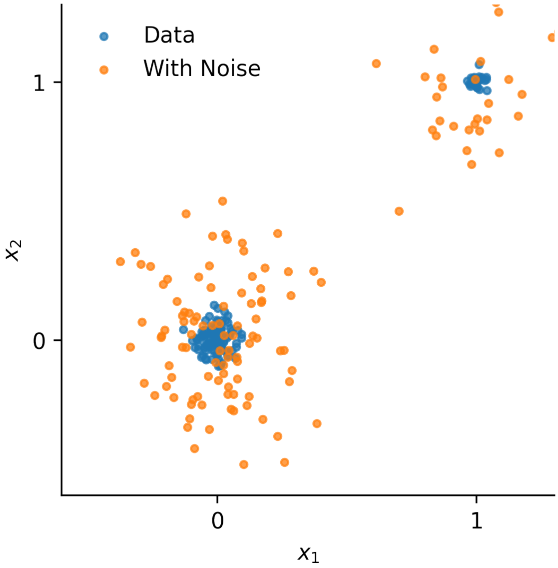
- Noisy Dataset



Score Matching

- Noisy Dataset
- Denoising Model

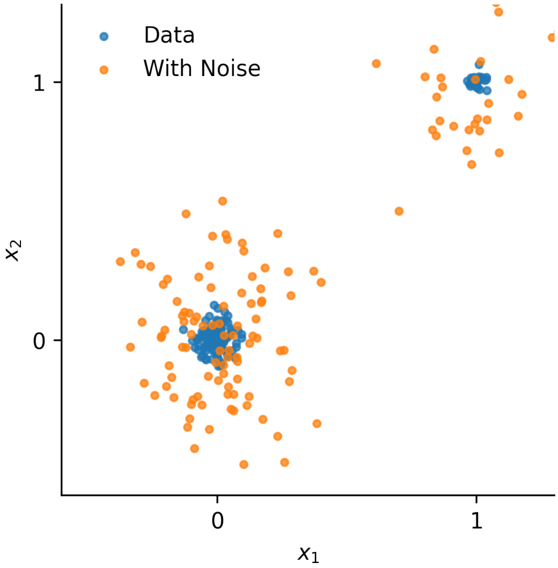
$$\mathrm{E}\left[\|m_\theta(\tilde{x}) - x\|_2^2\right]$$



Score Matching

- Noisy Dataset
- Denoising Model

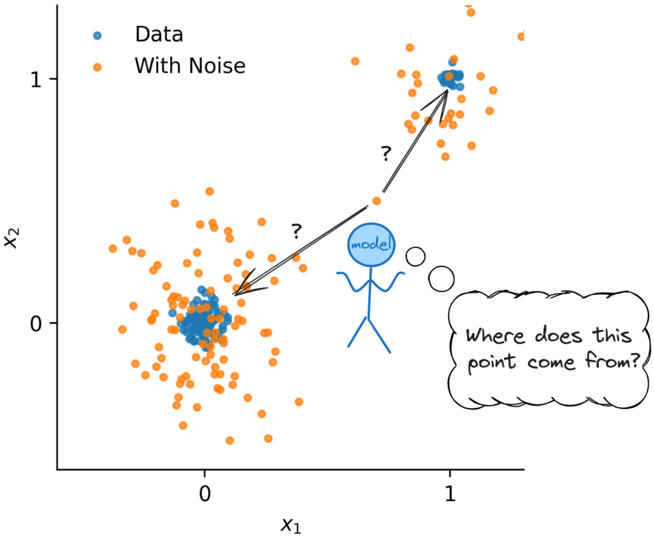
$$E\left[\|m_\theta(\tilde{x}) - x\|_2^2\right]$$



Score Matching

- Noisy Dataset
- Denoising Model

$$E\left[\|m_\theta(\tilde{x}) - x\|_2^2\right]$$



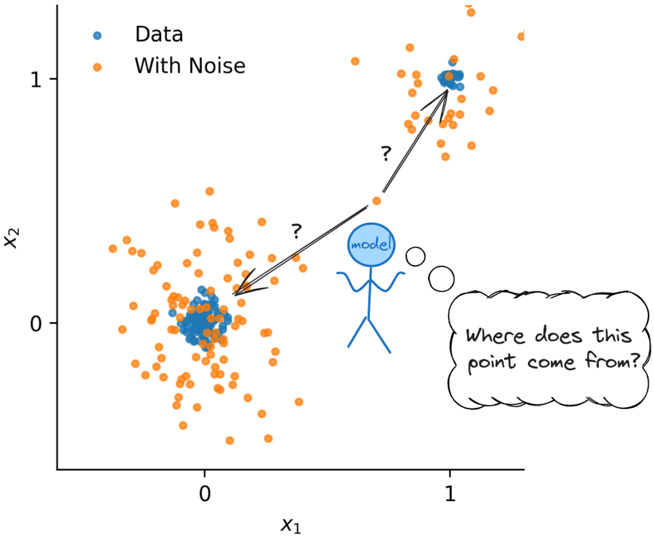
Score Matching

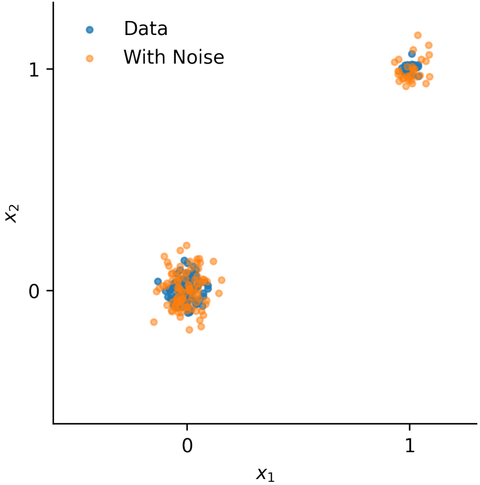
- Noisy Dataset
- Denoising Model

$$E\left[\|m_\theta(\tilde{x}) - x\|_2^2\right]$$

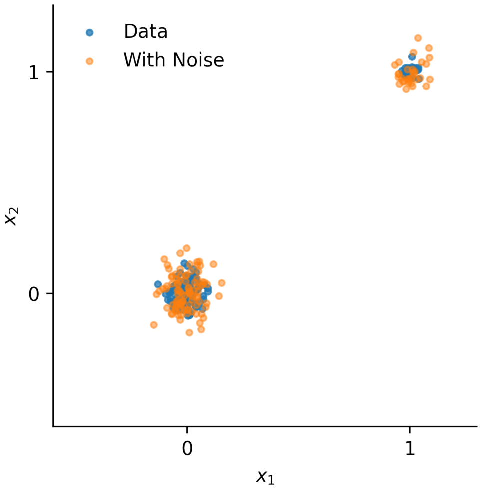
▷ Approximates Score

$$\frac{\tilde{x} - m_\theta(\tilde{x})}{\sigma^2}$$

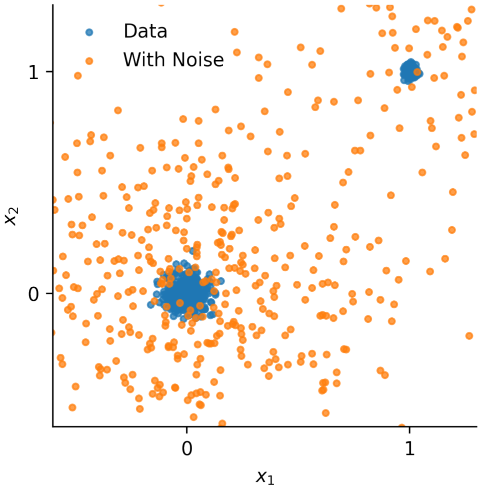




Accurate score, bad coverage



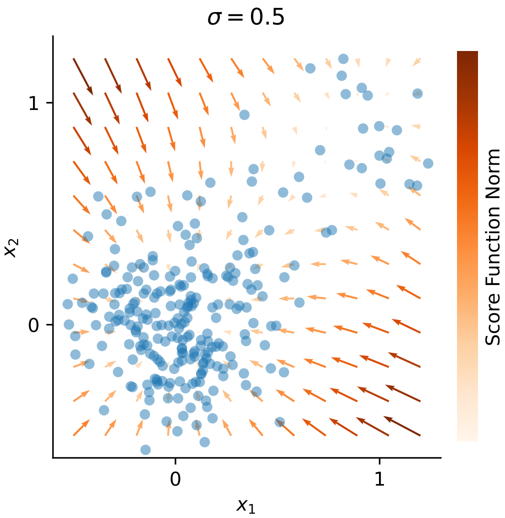
Accurate score, bad coverage



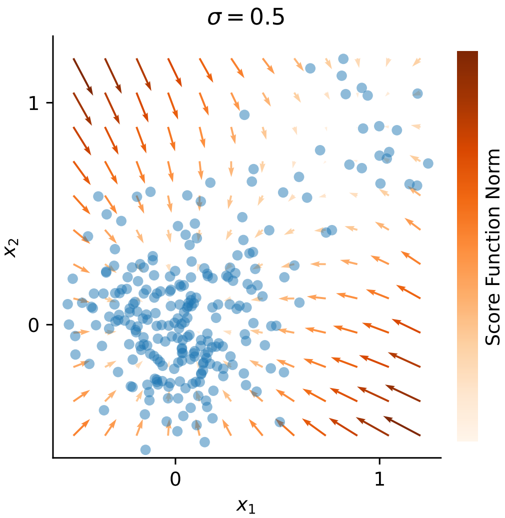
Good coverage, biased score

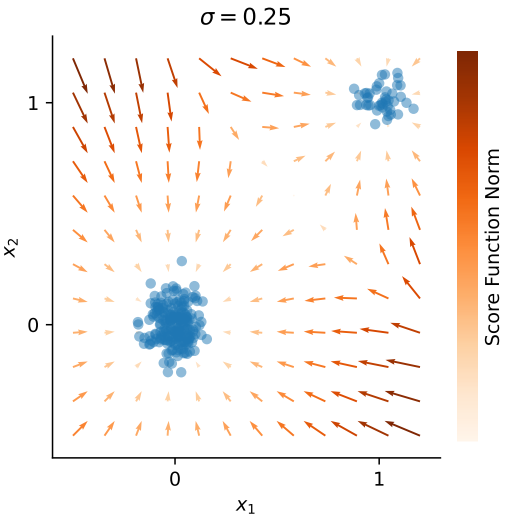
Annealing I

Annealing I



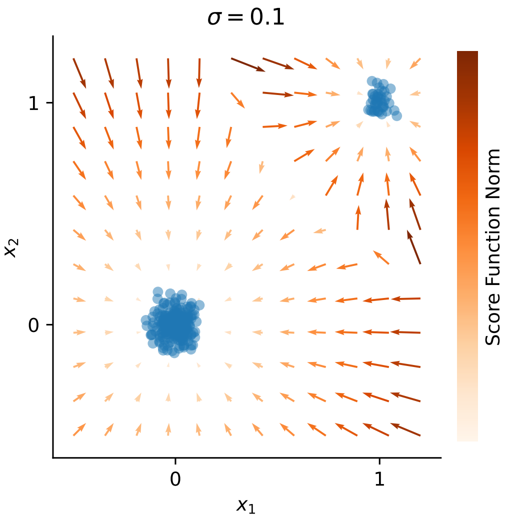
Annealing I


$\sigma = 0.5$
x_2
1
0
0
1
x_1
Score Function Norm

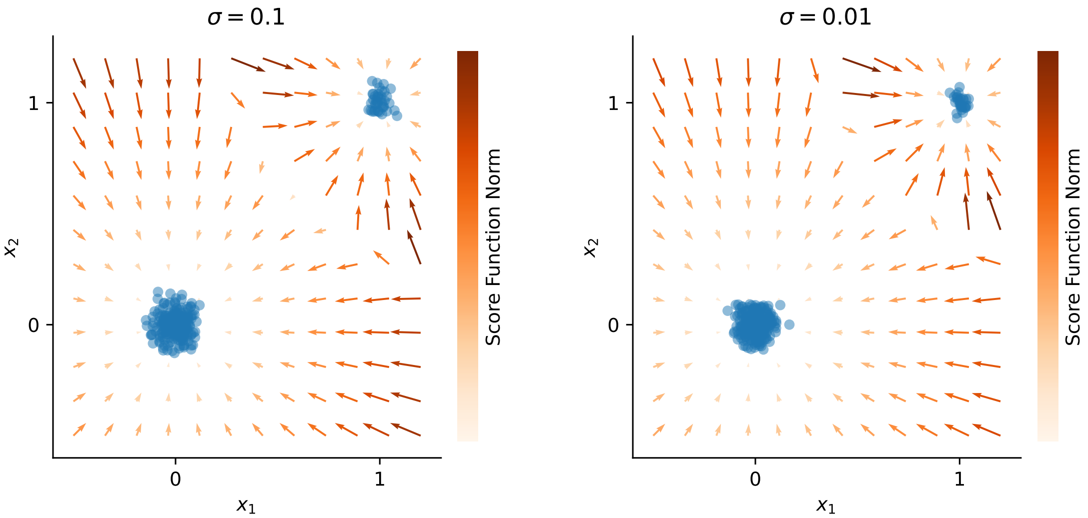

$\sigma = 0.25$
x_2
1
0
0
1
x_1
Score Function Norm

Annealing II

Annealing II



Annealing II



Probability Flow ODE

Song et al. 2021

Probability Flow ODE

- Infinite number of noise levels

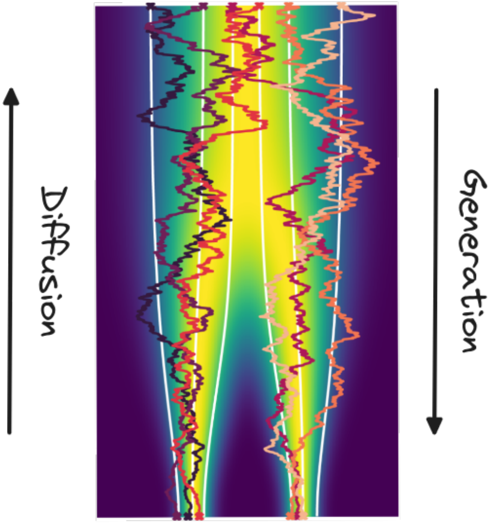
$$\mathrm{d}x = -\sigma(t)^2 \nabla_x \log p_t(x)\, \mathrm{d}t + \sigma(t)\, \mathrm{d}\bar{\omega}$$

Song et al. 2021

Probability Flow ODE

- Infinite number of noise levels

$$dx = -\sigma(t)^2 \nabla_x \log p_t(x)\, dt + \sigma(t)\, d\bar{\omega}$$



Song et al. 2021

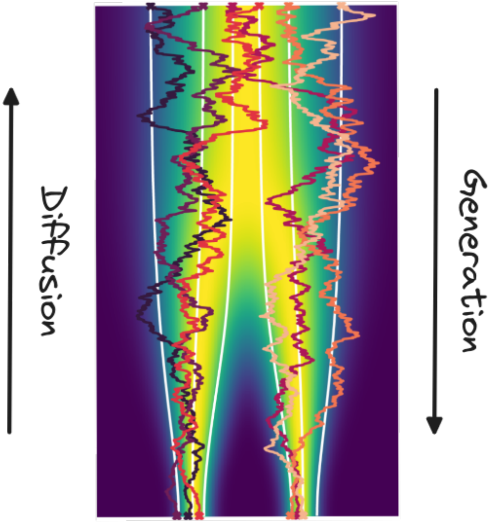
Probability Flow ODE

- Infinite number of noise levels

$$dx = -\sigma(t)^2 \nabla_x \log p_t(x)\, dt + \sigma(t)\, d\bar{\omega}$$

- ODE with same marginal distributions

$$dx = -\frac{\sigma(t)^2}{2} \nabla_x \log p_t(x)\, dt$$



Song et al. 2021

Probability Flow ODE

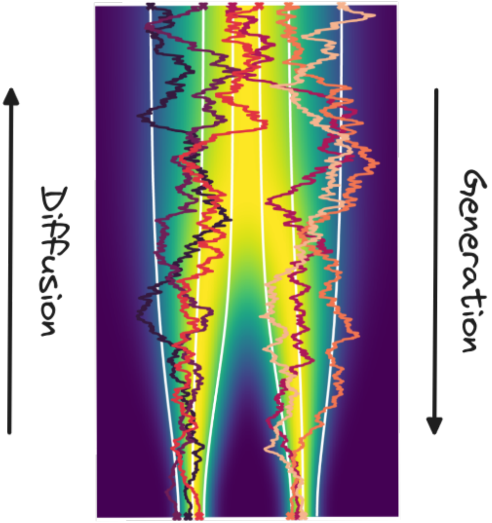
- Infinite number of noise levels

$$dx = -\sigma(t)^2 \nabla_x \log p_t(x)\, dt + \sigma(t)\, d\bar{\omega}$$

- ODE with same marginal distributions

$$dx = -\frac{\sigma(t)^2}{2} \nabla_x \log p_t(x)\, dt$$

▷ Defines a continuous normalising flow



Song et al. 2021

Flow Matching

Lipman et al. 2023

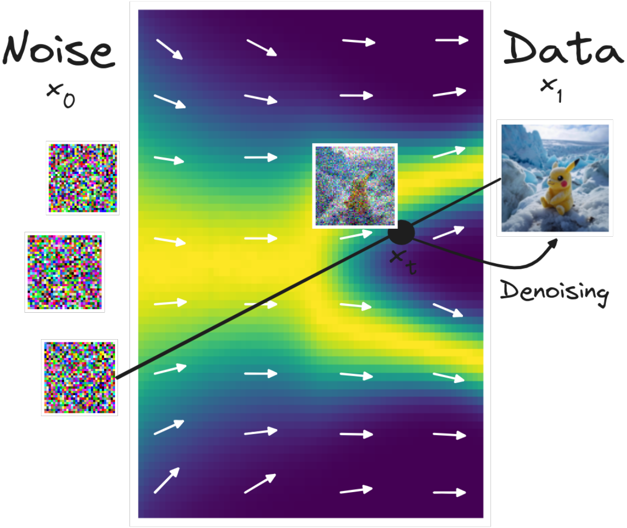
Flow Matching

- Sample noise x_0, data x_1

Lipman et al. 2023

Flow Matching

- Sample noise x_0, data x_1

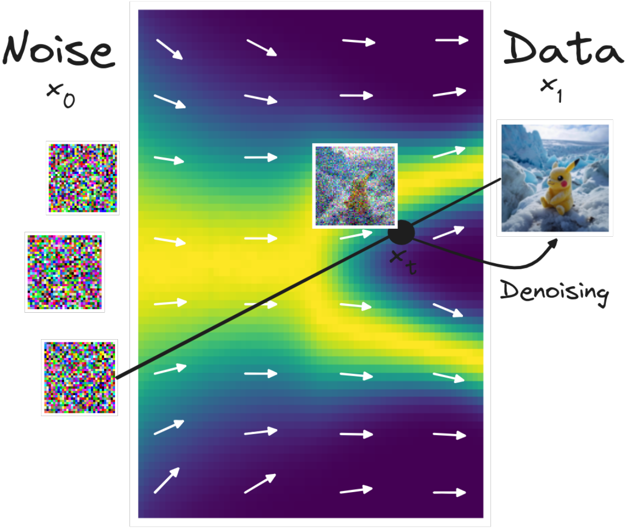


Lipman et al. 2023

Flow Matching

- Sample noise x_0, data x_1
- Interpolate with $t \in [0, 1]$

$$x_t = tx_1 + (1 - t)x_0$$



Lipman et al. 2023

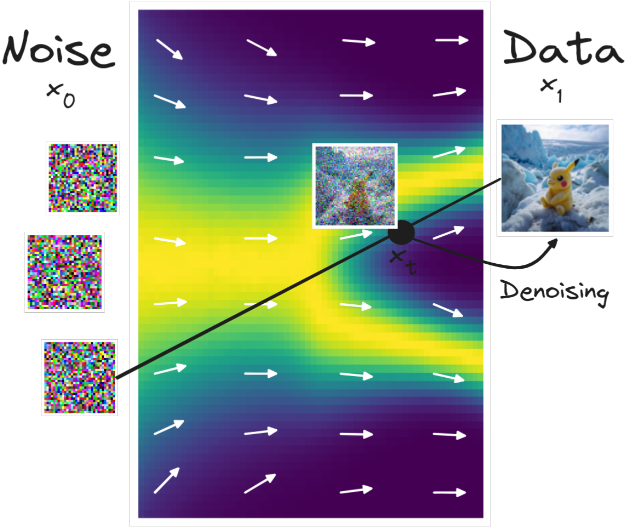
Flow Matching

- Sample noise x_0, data x_1
- Interpolate with $t \in [0, 1]$

$$x_t = tx_1 + (1 - t)x_0$$

- Model the denoising direction

$$\mathrm{E}_{x_t,t}\left[x_1 \mid x_t, t\right]$$



Lipman et al. 2023

Flow Matching

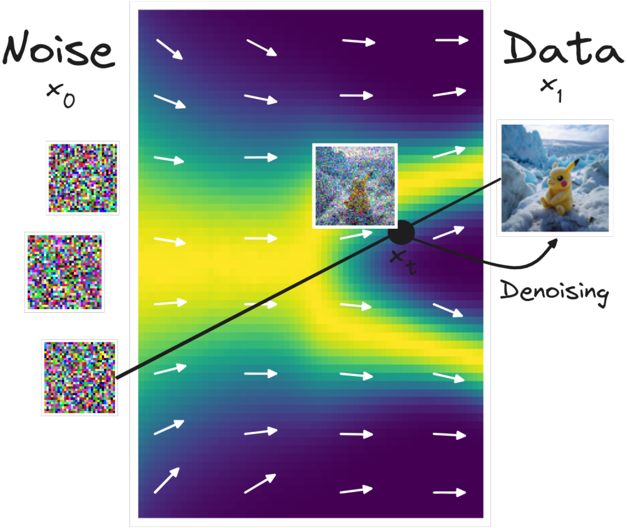
- Sample noise x_0, data x_1
- Interpolate with $t \in [0, 1]$

$$x_t = tx_1 + (1 - t)x_0$$

- Model the denoising direction

$$\mathrm{E}_{x_t,t}\left[x_1 \mid x_t, t\right]$$

- Flow v_θ points in that direction



Lipman et al. 2023

Mini Batch OT Flow Matching

Tong et al. 2024

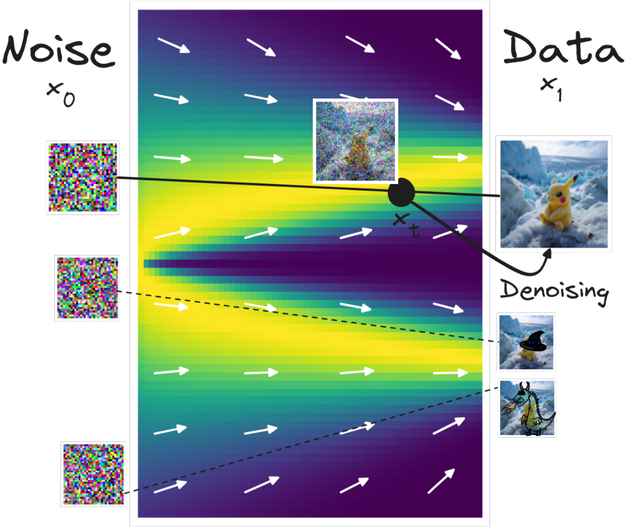
Mini Batch OT Flow Matching

- Batch sample $\left\{x_0^{(i)}, x_1^{(i)}\right\}_{i=1}^{n}$

Tong et al. 2024

Mini Batch OT Flow Matching

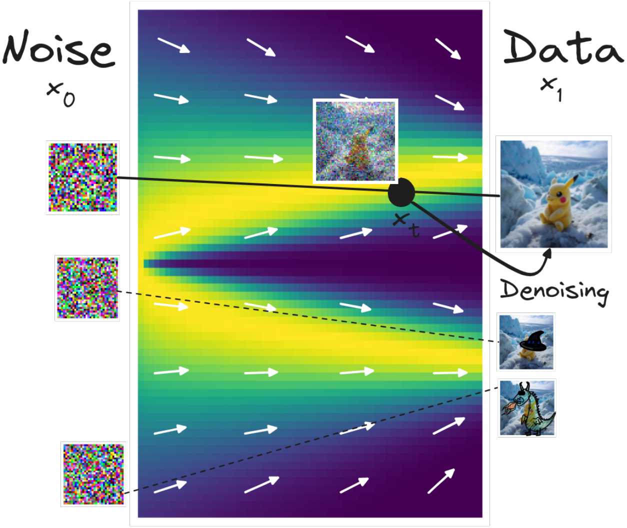
- Batch sample $\left\{x_0^{(i)}, x_1^{(i)}\right\}_{i=1}^{n}$



Tong et al. 2024

Mini Batch OT Flow Matching

- Batch sample $\left\{x_0^{(i)}, x_1^{(i)}\right\}_{i=1}^{n}$
- Compute OT assignments Π

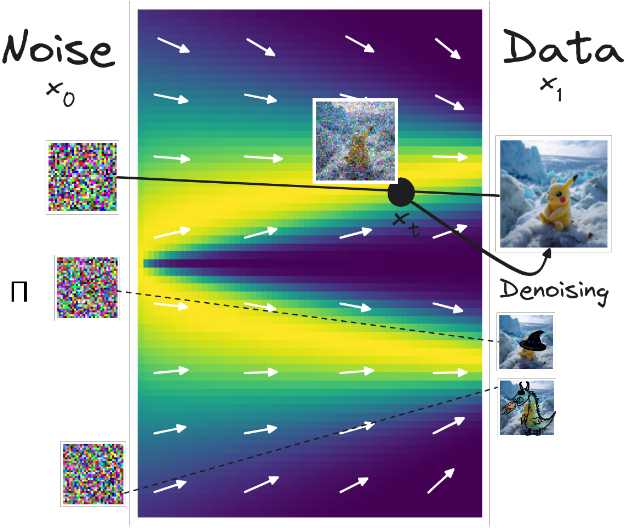


Tong et al. 2024

Mini Batch OT Flow Matching

- Batch sample $\left\{x_0^{(i)}, x_1^{(i)}\right\}_{i=1}^{n}$
- Compute OT assignments Π
- Construct geodesic points $x_t^{(i)}$

$$x_t = tx_1^{(j)}+(1-t)x_0^{(i)}, \ (x_0^{(i)}, x_1^{(j)}) \in \Pi$$

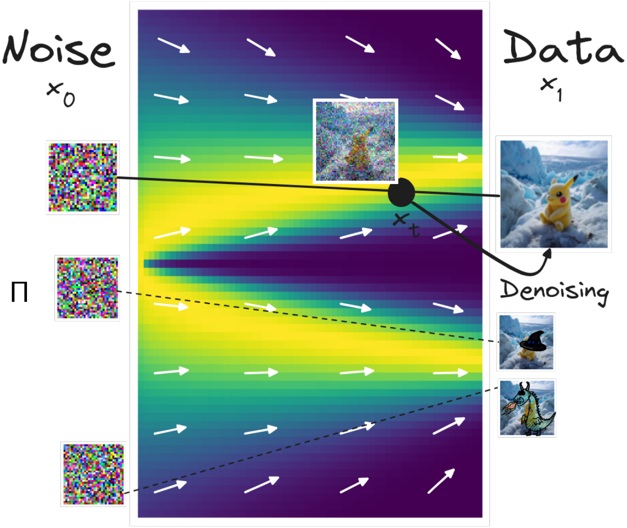


Tong et al. 2024

Mini Batch OT Flow Matching

- Batch sample $\left\{x_0^{(i)}, x_1^{(i)}\right\}_{i=1}^{n}$
- Compute OT assignments Π
- Construct geodesic points $x_t^{(i)}$

  $x_t = tx_1^{(j)} + (1-t)x_0^{(i)}, \ (x_0^{(i)}, x_1^{(j)}) \in \Pi$
- Learn denoising direction

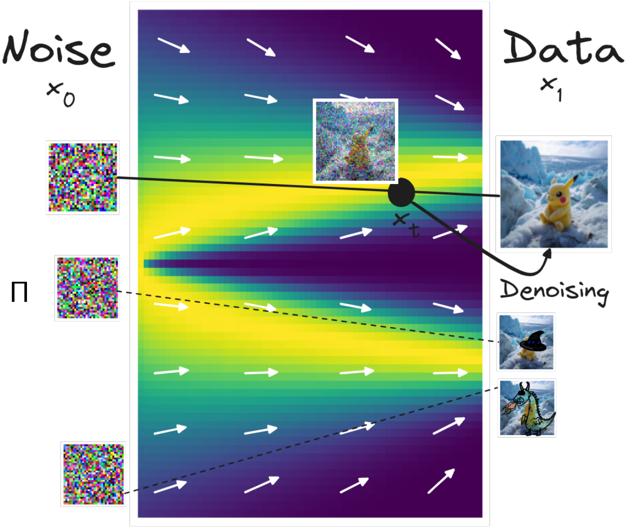


Tong et al. 2024

Mini Batch OT Flow Matching

- Batch sample $\left\{x_0^{(i)}, x_1^{(i)}\right\}_{i=1}^{n}$
- Compute OT assignments Π
- Construct geodesic points $x_t^{(i)}$

  $$x_t = tx_1^{(j)} + (1-t)x_0^{(i)},\ (x_0^{(i)}, x_1^{(j)}) \in \Pi$$
- Learn denoising direction
- ODE paths become straight lines, as $n \to \infty$



Tong et al. 2024

Summary

- Generative Models can take the form of velocity fields
- Simulation-based training is computationally expensive
- Therefore we fix intermediate distributions
- Diffusion models add noise to the source distribution
- Flow matching interpolates between source and target points
- Both learn denoising models for sampling
- Improve flow matching using mini batch OT

Thanks! Questions?

References

Anderson, **Brian D.O.** (1982). “Reverse-time diffusion equation models”. In: *Stochastic Processes and their Applications* 12.3, pp. 313–326. ISSN: 0304-4149.

Brown, **Tom B.** et al. (2020). *Language Models are Few-Shot Learners*. arXiv: 2005.14165 [cs.CL].

Bubeck, **Sébastien** et al. (2023). *Sparks of Artificial General Intelligence: Early experiments with GPT-4*. arXiv: 2303.12712 [cs.CL].

Chen, **Ricky T. Q.** et al. (2019). *Neural Ordinary Differential Equations*. arXiv: 1806.07366 [cs.LG].

Grathwohl, **Will** et al. (2018). *FFJORD: Free-form Continuous Dynamics for Scalable Reversible Generative Models*. arXiv: 1810.01367 [cs.LG].

Ho, **Jonathan**, **Ajay** Jain, and **Pieter** Abbeel (2020). *Denoising Diffusion Probabilistic Models*. arXiv: 2006.11239 [cs.LG].

Hutchinson, **M.F.** (1990). “A stochastic estimator of the trace of the influence matrix for laplacian smoothing splines”. In: *Communications in Statistics - Simulation and Computation* 19.2, pp. 433–450.

Lipman, **Yaron** et al. (2023). *Flow Matching for Generative Modeling*. arXiv: 2210.02747 [cs.LG].

Rezende, **Danilo Jimenez** and **Shakir** Mohamed (2016). *Variational Inference with Normalizing Flows*. arXiv: 1505.05770 [stat.ML].

Sohl-Dickstein, **Jascha** et al. (2015). *Deep Unsupervised Learning using Nonequilibrium Thermodynamics*. arXiv: 1503.03585 [cs.LG].

Song, **Yang** et al. (2021). *Score-Based Generative Modeling through Stochastic Differential Equations*. arXiv: 2011.13456 [cs.LG].

Tong, **Alexander** et al. (2024). *Improving and generalizing flow-based generative models with minibatch optimal transport*. arXiv: 2302.00482 [cs.LG].