A talk at Ghent University on

Score-based generative modelling: intro and ongoing research

Vincent Dutordoir

UNIVERSITY OF CAMBRIDGE

Secondmind

Hi, I'm Vincent 👋

Hi, I'm Vincent 👋

- Finished my Bachelor's and Masters at Ghent University in 2017
- Moved to Cambridge (UK) straight after to work for a startup PROWLER.io
- Started my PhD at Cambridge University in 2020
- Senior ML researcher at Secondmind.ai

Outline

The talk will consist of two parts:

1. A tutorially overview of energy-based models, and how this has led to diffusion models.
2. Diffusion models for stochastic processes.

Credits to Yang Song, Michael Hutchinson, and Arnaud Doucet for some of the images and slide material.

Generative modelling

- Given: dataset $\{\boldsymbol{x}_i\}_{i=1}^n$
- Goal: fit a model $p_\theta(\boldsymbol{x})$ to the data distribution

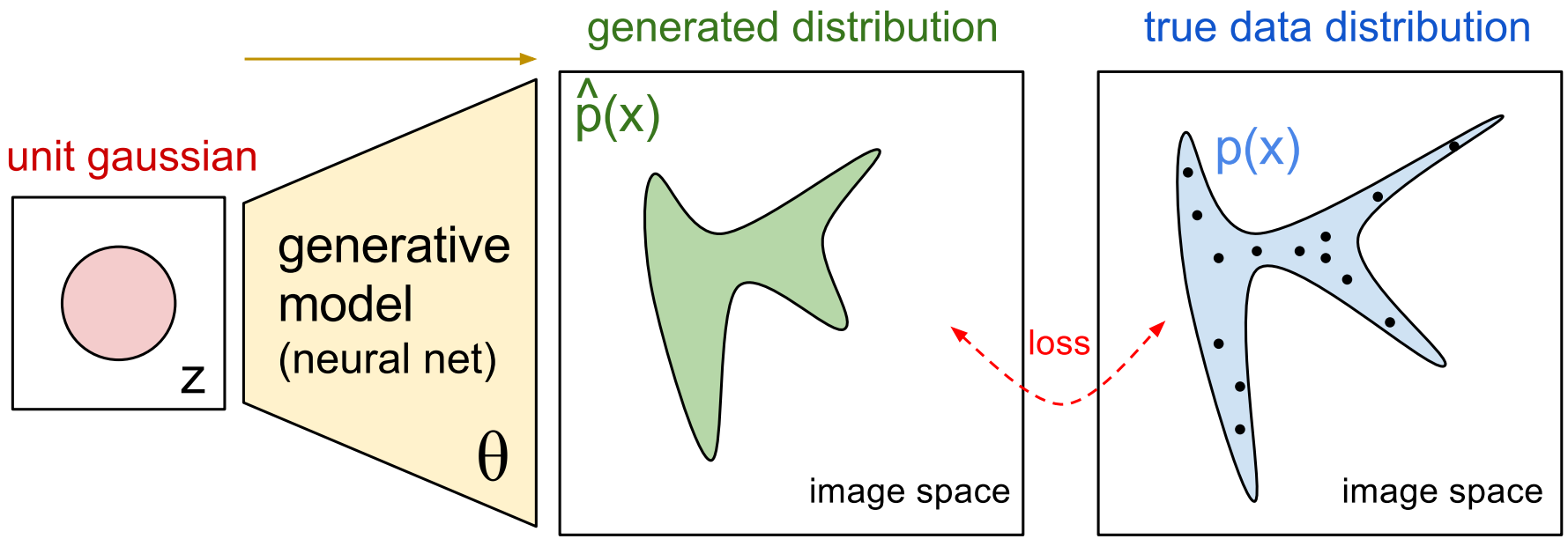


Illustration generative modelling (from: openai.com)

Energy-based models

A density defined through an *energy function* $U_\theta : \mathbb{R}^d \to \mathbb{R}$:

$$p_\theta(\boldsymbol{x}) = \frac{\mathrm{e}^{-U_\theta(x)}}{Z_\theta},$$

We can fit this energy function by maximising the log likelihood:

$$\theta^* = \max_\theta \sum_{i=1}^N \log p_\theta(\boldsymbol{x}_i)$$

⚠️ Intractable normalizing constant: $Z_\theta = \int_{\mathbb{R}^d} \mathrm{e}^{-U_\theta(\boldsymbol{x})} \mathrm{d}\boldsymbol{x}$.

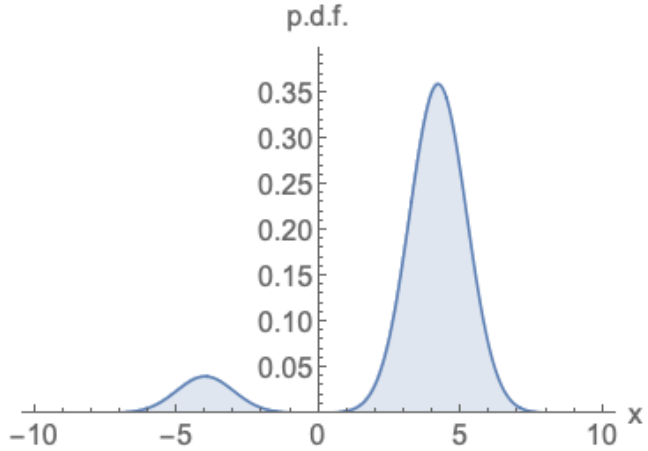
Score function

Modelling the density through the score function

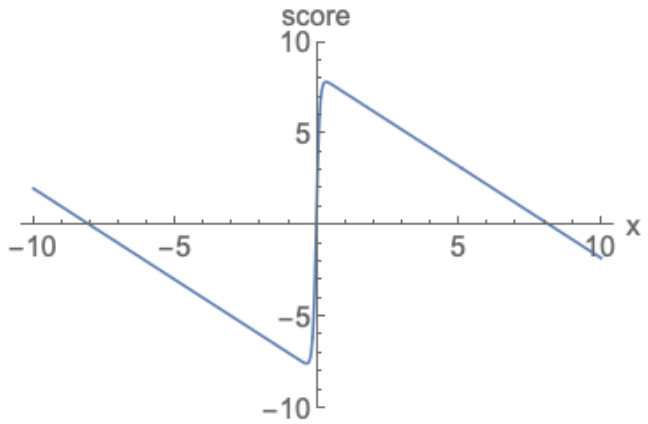
$$\nabla_{\boldsymbol{x}} \log p(\boldsymbol{x}) \approx \boldsymbol{s}_\theta(\boldsymbol{x}).$$

The score does *not* depend on the normalizing constant. Let $p(x) = \frac{q(x)}{Z}$

$$\nabla_x \log \frac{q(x)}{Z} = \nabla_x \log q(x) - \underbrace{\nabla_x \log Z}_{=0}$$



$p(x)$



$\nabla_x \log p(x)$

Langevin dynamics

Theorem 1 The density of $\boldsymbol{x}_t$ as $t \to \infty$ for the SDE

$$\mathrm{d}\boldsymbol{x}_t = \nabla \log p(\boldsymbol{x}_t)\mathrm{d}t + \sqrt{2}\mathrm{d}\boldsymbol{W}_t$$

is given by $p(\boldsymbol{x})$, where $\boldsymbol{W}_t$ is standard Brownian motion.

Euler-Marayuma

First-order discretization of continuous SDE. For a small stepsize γ we can

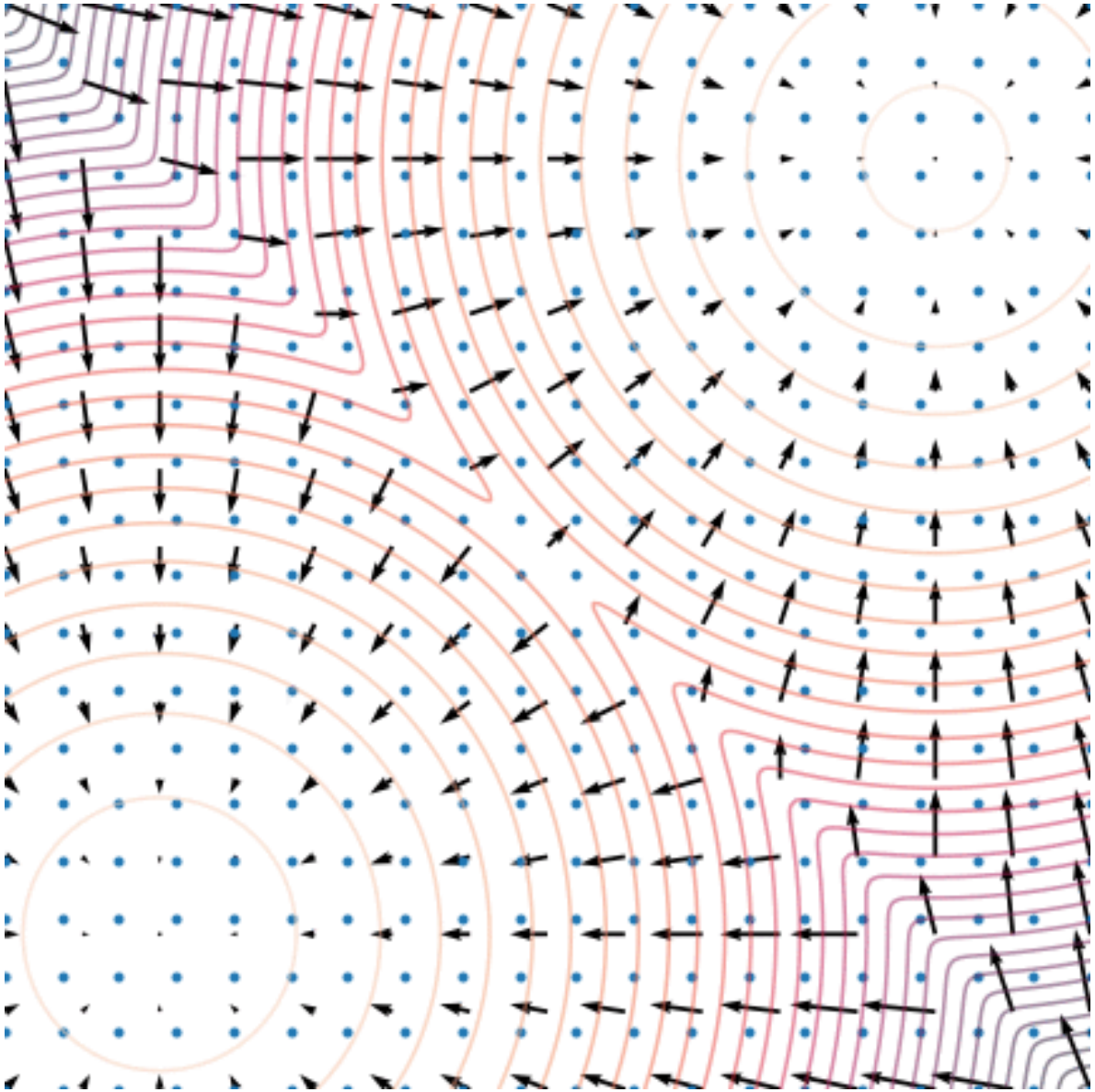
$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \gamma \nabla \log p(\boldsymbol{x}_k) + \sqrt{2}\boldsymbol{z}_k, \quad \boldsymbol{z}_k \sim \mathcal{N}(0, \gamma \mathbf{I})$$

Fisher divergence

We can train score-based models by minimizing the Fisher divergence between the model and the data distributions

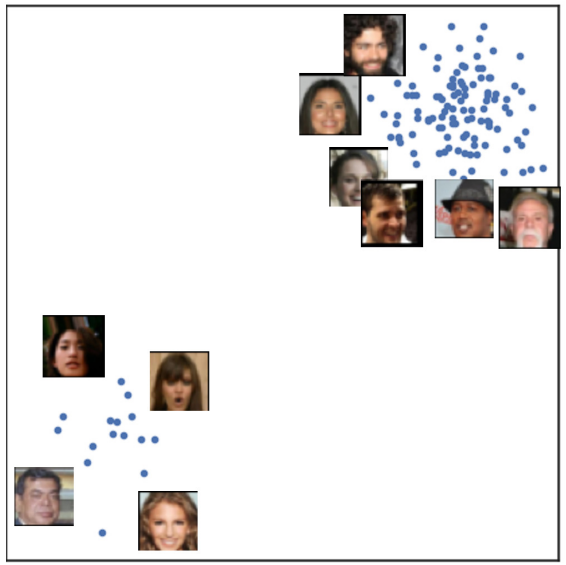
$$\mathbb{E}_{p(\boldsymbol{x})}\left[\|\nabla_{\boldsymbol{x}} \log p(\boldsymbol{x}) \ - \mathbf{s}_\theta(\boldsymbol{x})\|^2\right].$$

- Infeasible because it requires access to the unknown data score.
- Score-matching: Hyvärinen (2005), Vincent (2011), Song et al. (2019)



Using Langevin dynamics to sample from a mixture of two Gaussians. (from: Yang Song)

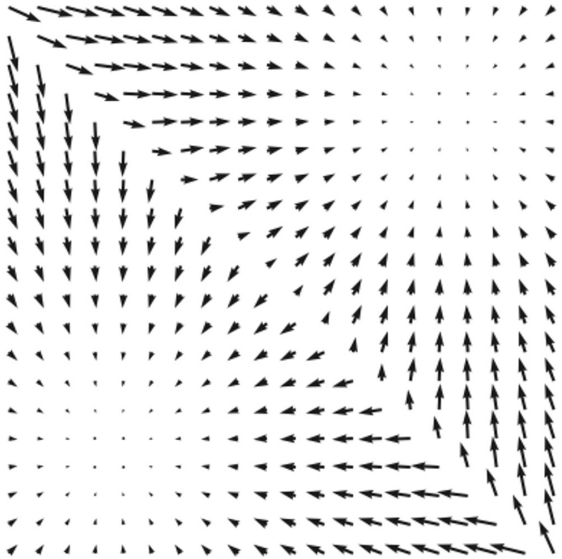
Naive score-based generative modeling



Data samples

$$\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\} \overset{\text{i.i.d.}}{\sim} p(\mathbf{x})$$

score matching



Scores

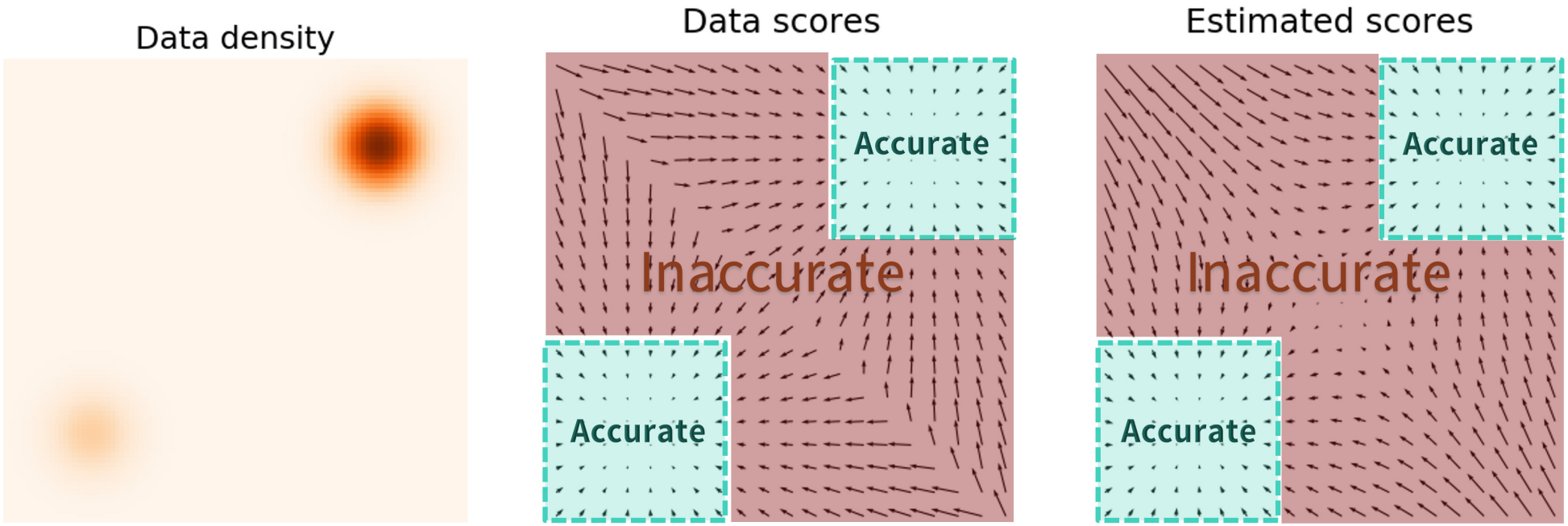
$$\mathbf{s}_\theta(\mathbf{x}) \approx \nabla_{\mathbf{x}} \log p(\mathbf{x})$$

Langevin dynamics

New samples

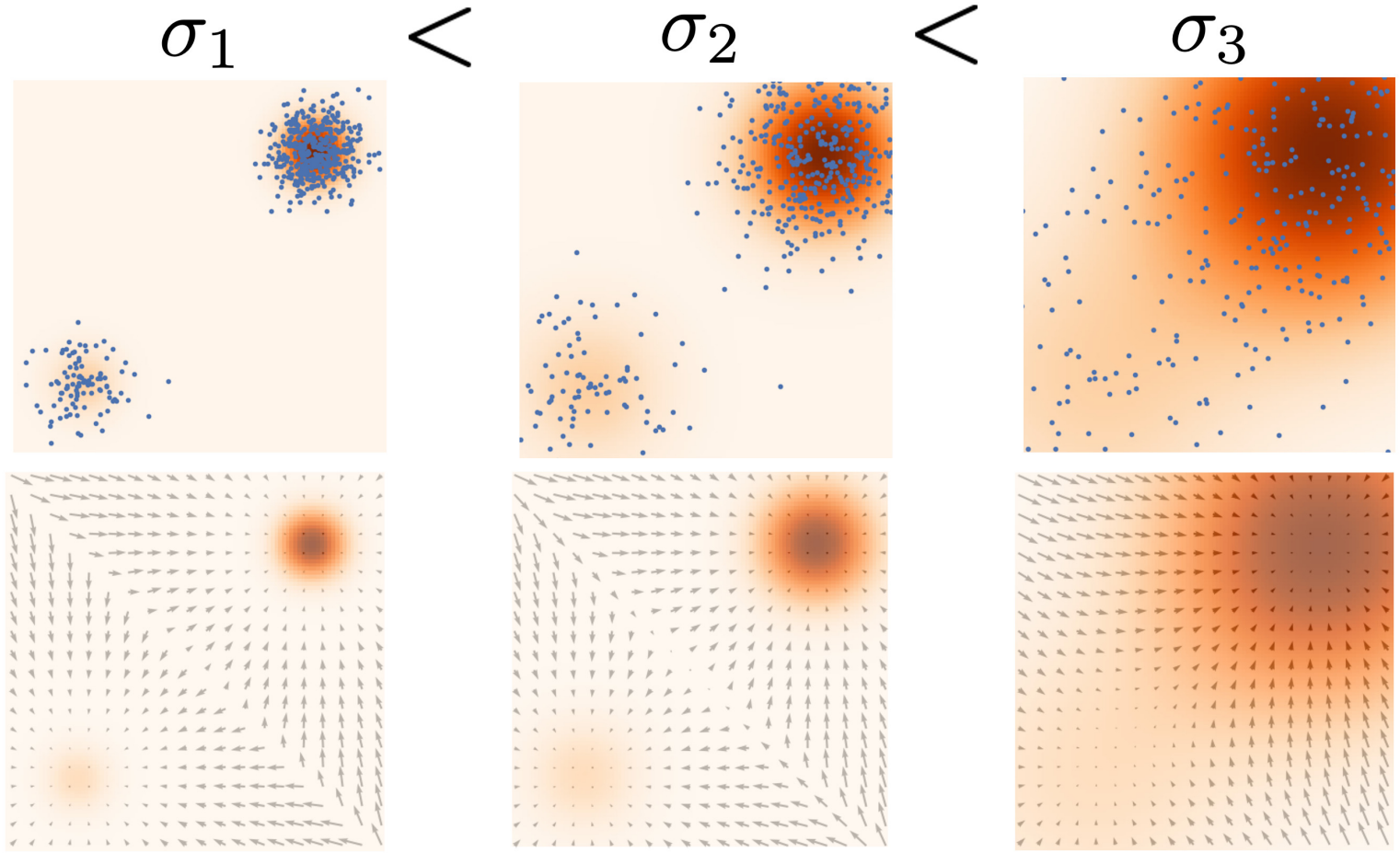
Score-based generative modeling with score matching + Langevin dynamics. (from: Yang Song)

Pitfalls



- Score is badly estimated in low-density areas
- Langevin dynamics has slow mixing rates

Multiple noise levels



Gaussian noise to perturb the data distribution

Song and Ermon (2019) suggest to perturb data points such that they populate low data density regimes.

Markov chain

Consider a Markov chain $\boldsymbol{x}_0 \sim p_0$ and $\boldsymbol{x}_{k+1} \sim p_{k+1|k}(\cdot|\boldsymbol{x}_k)$, which gives

Forward

$$p(\boldsymbol{x}_{0:K}) = p_0(\boldsymbol{x}_0) \prod_{k=0}^{K-1} p_{k+1|k}(\boldsymbol{x}_{k+1}|\boldsymbol{x}_k)$$

Backward

$$p(\boldsymbol{x}_{0:K}) = p_K(\boldsymbol{x}_K) \prod_{k=K-1}^{0} p_{k|k+1}(\boldsymbol{x}_k|\boldsymbol{x}_{k+1})$$

where $p_{k|k+1}(\boldsymbol{x}_k|\boldsymbol{x}_{k+1})$ is unknown but can be obtained with Bayes' rule.

Generative modelling with multiple noise levels

Let

$$p_0 = p_{data}$$

Choose

$$p_{k+1|k}(\boldsymbol{x}_{k+1}|\boldsymbol{x}_k) = \mathcal{N}(\boldsymbol{x}_{k+1}|\alpha\boldsymbol{x}_k, (1-\alpha^2)\mathbf{I})$$

such that for large enough K we have

$$p_K \approx p_{ref} = \mathcal{N}(0, \mathbf{I}).$$

Backward transition

1. For sampling we need the *reverse* kernel $p_{k|k+1}$, given through Bayes' rule

$$p_{k|k+1}(\boldsymbol{x}_k|\boldsymbol{x}_{k+1}) = \frac{p_{k+1|k}(\boldsymbol{x}_{k+1}|\boldsymbol{x}_k)p_k(\boldsymbol{x}_k)}{p_{k+1}(\boldsymbol{x}_{k+1})}$$

   which is unfortunately intractable!

2. Using a Taylor approximation one can show

$$p_{k|k+1}(\boldsymbol{x}_k|\boldsymbol{x}_{k+1}) \approx \mathcal{N}\left(\boldsymbol{x}_k|(2-\alpha)\boldsymbol{x}_{k+1} + (1-\alpha^2)\nabla \log p_{k+1}(\boldsymbol{x}_{k+1}), (1-\alpha^2)\mathbf{I}\right)$$

3. Approximate score with neural net $\boldsymbol{s}_\theta(\boldsymbol{x}_{k+1}, k+1) \approx \nabla \log p_{k+1}(\boldsymbol{x}_{k+1})$.

4. Sampling start with $\boldsymbol{x}_K \sim p_{ref}(\boldsymbol{x}_K)$ and then uses the reverse kernel

$$\boldsymbol{x}_k = (2-\alpha)\boldsymbol{x}_{k+1} + (1-\alpha^2)\boldsymbol{s}_\theta(\boldsymbol{x}_{k+1}, k+1) + (1-\alpha^2)\boldsymbol{\epsilon} \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}).$$

Score

The score $\nabla \log p_k(\boldsymbol{x}_k)$ is *required* but analytically *unavailable*. However, using

$$p_k(\boldsymbol{x}_k) = \int p(\boldsymbol{x}_0) p_{k|0}(\boldsymbol{x}_k|\boldsymbol{x}_0) \mathrm{d}\boldsymbol{x}_0$$

it follows

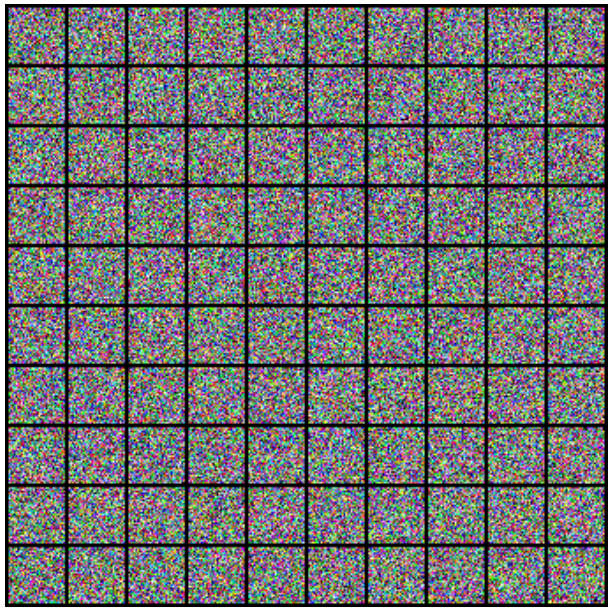
$$\nabla \log p_k(\boldsymbol{x}_k) = \mathbb{E}_{\boldsymbol{x}_0 \sim p(\cdot|\boldsymbol{x}_k)} \left[\nabla \log p_{k|0}(\boldsymbol{x}_k|\boldsymbol{x}_0) | \boldsymbol{x}_k \right]$$

A conditional expectation can be written as a regression problem (by definition), which gives

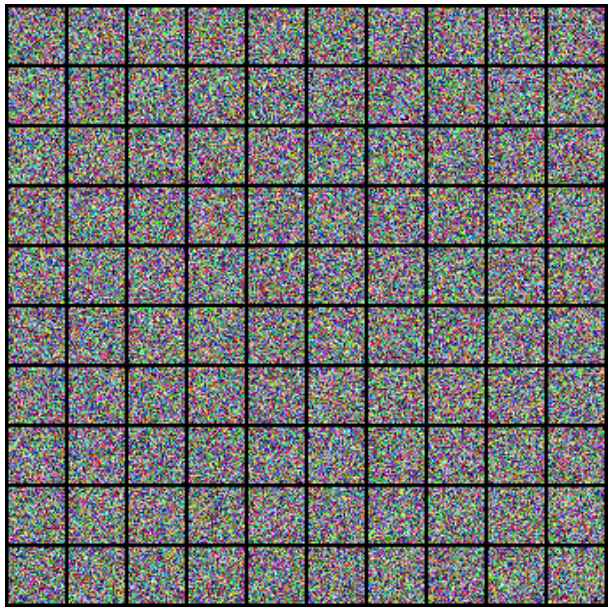
$$\nabla \log p_k(\boldsymbol{x}_k) = \operatorname{argmin}_\theta \mathbb{E}_{\boldsymbol{x}_0, \boldsymbol{x}_k} \left[\left\| \boldsymbol{s}_\theta(\boldsymbol{x}_k) - \nabla_{\boldsymbol{x}_k} \log p_{k|0}(\boldsymbol{x}_k|\boldsymbol{x}_0) \right\|^2 \right]$$

Annealed Langevin Dynamics

- $\approx$ Noise Conditional Score Network (NCSN) by Song and Ermon (2019)
- $\approx$ Denoising Diffusion Probabilistic Models (DDPM) by Ho, Jain, and Abbeel (2020)



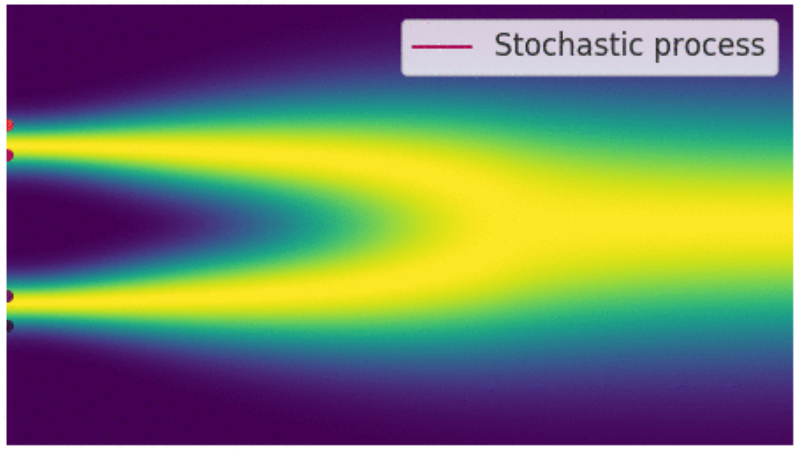
Celeb A



CIFAR-10

Perturbing data with an SDE in continuous time

From a (large) discrete set of noise scales $\rightarrow$ continuous number.



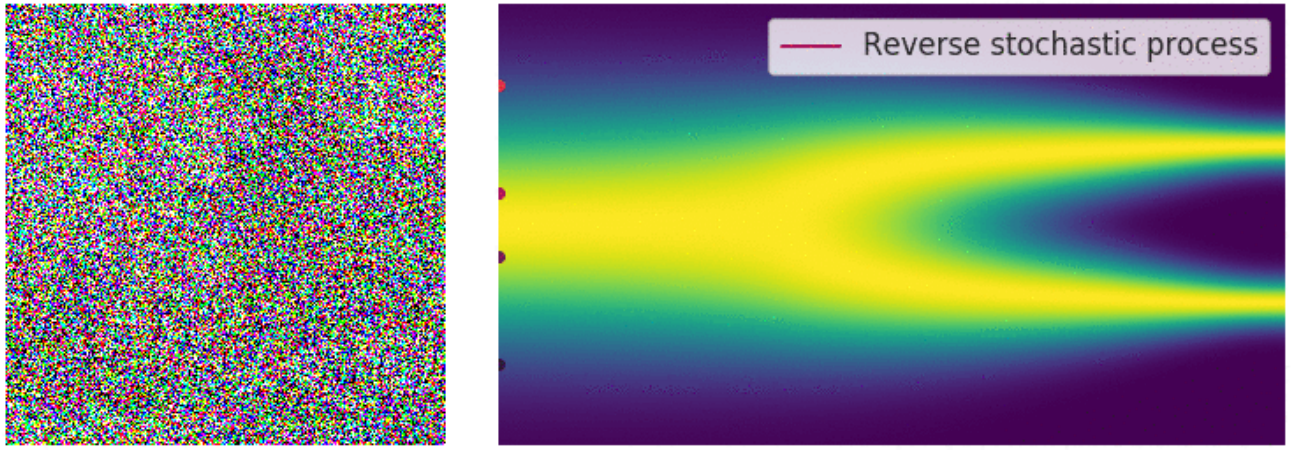
Forward SDE runs

The SDE can be written as

$$\mathrm{d}\boldsymbol{x}_t = f(\boldsymbol{x}_t, t)\mathrm{d}t + g(t)\mathrm{d}\mathbf{W}_t, \qquad \boldsymbol{x}_0 \sim p_{data}$$

where f and g are the drift and diffusion terms, and $\mathbf{W}_t$ is standard Brownian motion. Heuristically, you can think of it as "$\mathrm{d}\mathbf{W}/\mathrm{d}t \sim \mathcal{N}(0, \mathrm{d}t)$".

Reversing the SDE for sample generation



Generating data following the reverse SDE

Reverse process (Nelson's duality)

$$\mathrm{d}\boldsymbol{x}_t = \left[f(\boldsymbol{x}_t, t) - g^2(t)\nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x})\right]\mathrm{d}t + g(t)\mathrm{d}\bar{\mathbf{W}}_t, \quad \boldsymbol{x}_T \sim p_T$$

where $\mathrm{d}t$ represents a *negative* infinitesimal time step as $t = T \to 0$.

Generative modelling by approximating the reverse process

Exact reverse process

$$\mathrm{d}\boldsymbol{x}_t = \left[f(\boldsymbol{x}_t, t) - g^2(t)\nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x})\right]\mathrm{d}t + g(t)\mathrm{d}\bar{\mathbf{W}}_t, \quad \boldsymbol{x}_T \sim p_T$$

Generative model

$$\mathrm{d}\boldsymbol{x}_t = \left[f(\boldsymbol{x}_t, t) - g^2(t)\boldsymbol{s}_{\theta^*}(\boldsymbol{x}_t, t)\right]\mathrm{d}t + g(t)\mathrm{d}\bar{\mathbf{W}}_t, \quad \boldsymbol{x}_T \sim p_{ref}$$

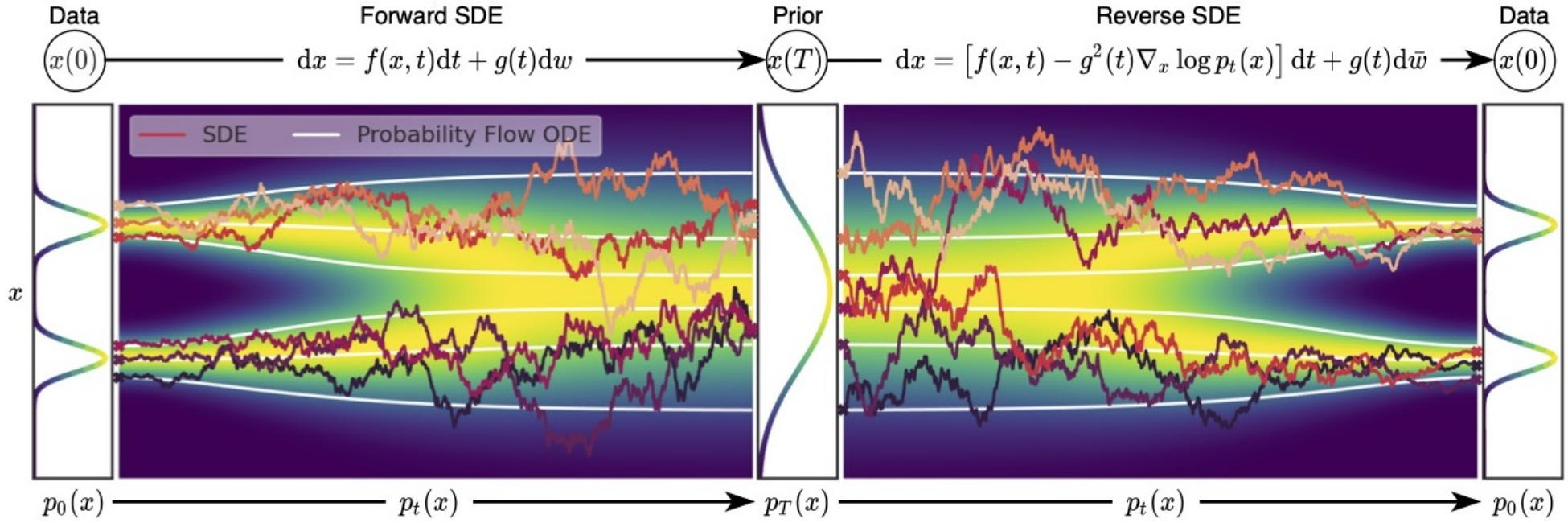
The score is learned using score-matching, similar to before

$$\theta^* = \mathrm{argmin}_\theta \left[\mathbb{E}_{\boldsymbol{x}_0, \boldsymbol{x}_t} \left\| \boldsymbol{s}_\theta(\boldsymbol{x}_t, t) - \nabla_{\boldsymbol{x}_t} \log p_{t|0}(\boldsymbol{x}_t|\boldsymbol{x}_0) \right\|^2\right]$$

For OU processes $p_{t|0}(\boldsymbol{x}_t|\boldsymbol{x}_0)$ can analytically be computed using the Fokker-Planck equations and leads to simple expression of the form

$$p_{t|0}(\boldsymbol{x}_t|\boldsymbol{x}_0) = \mathcal{N}\left(\boldsymbol{x}_t; \mathrm{e}^{-t}\boldsymbol{x}_0, (1 - \mathrm{e}^{-2t})\mathbf{I}\right)$$

Continuous-time denoising — Song et al. (2021)



Forward-Reverse

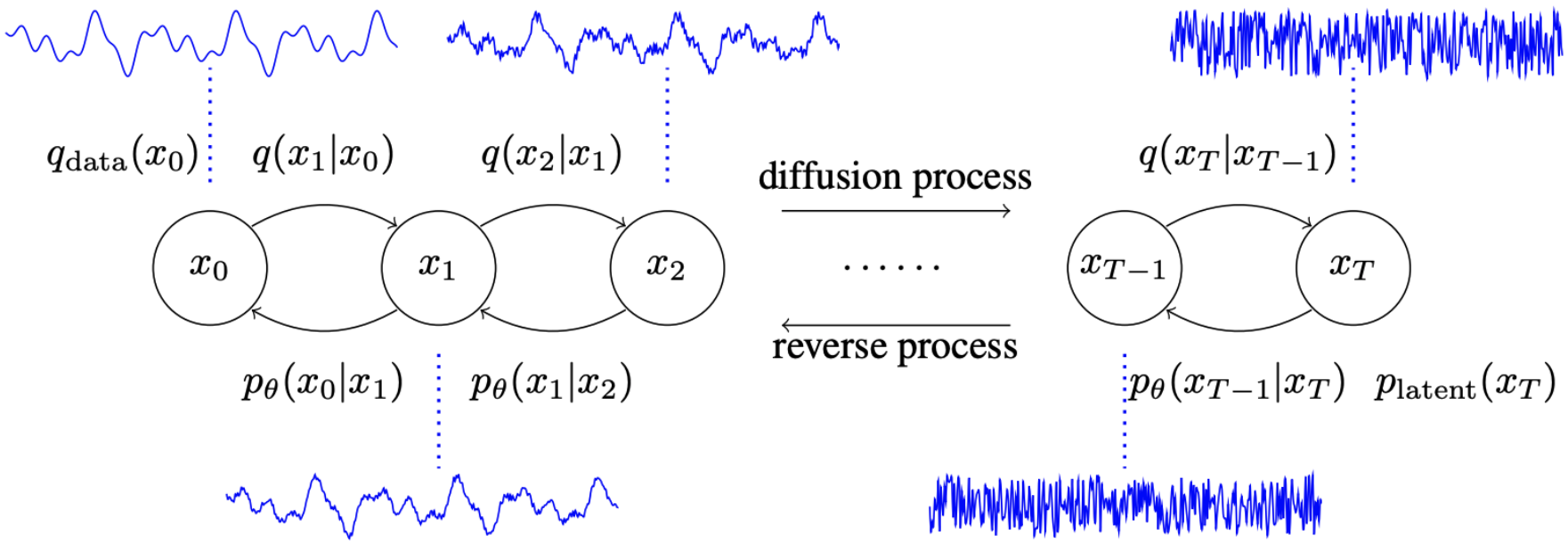
- Continuous-time formulation generalizes the discrete approaches.
- Log-likelihood computations $\log p_\theta(\boldsymbol{x}_0)$ (not shown here).

Neural Diffusion Processes

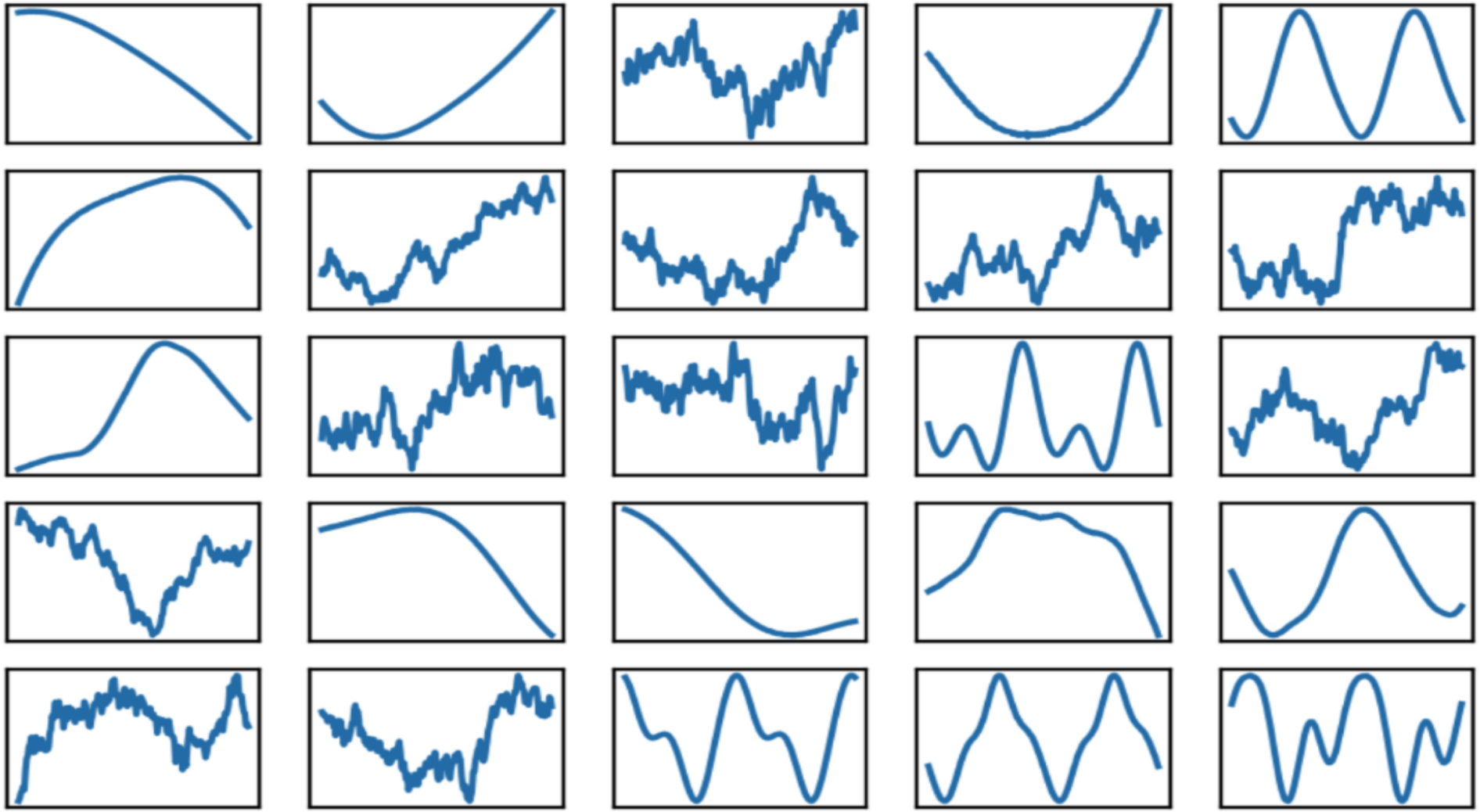
VD, Alan Saul, Zoubin Ghahramani and Fergus Simpson, Arxiv (2022)

Motivation

- Diffusion models have been used on different data modalities:



Diffusion models for ‘functions’



Distribution over functions

Perturbing function-values using OU process

Let $\mathbf{X} \in \mathbb{R}^{n \times d}$ and $\boldsymbol{y}(\boldsymbol{X}) \in \mathbb{R}^N$, we define the forward noising process as

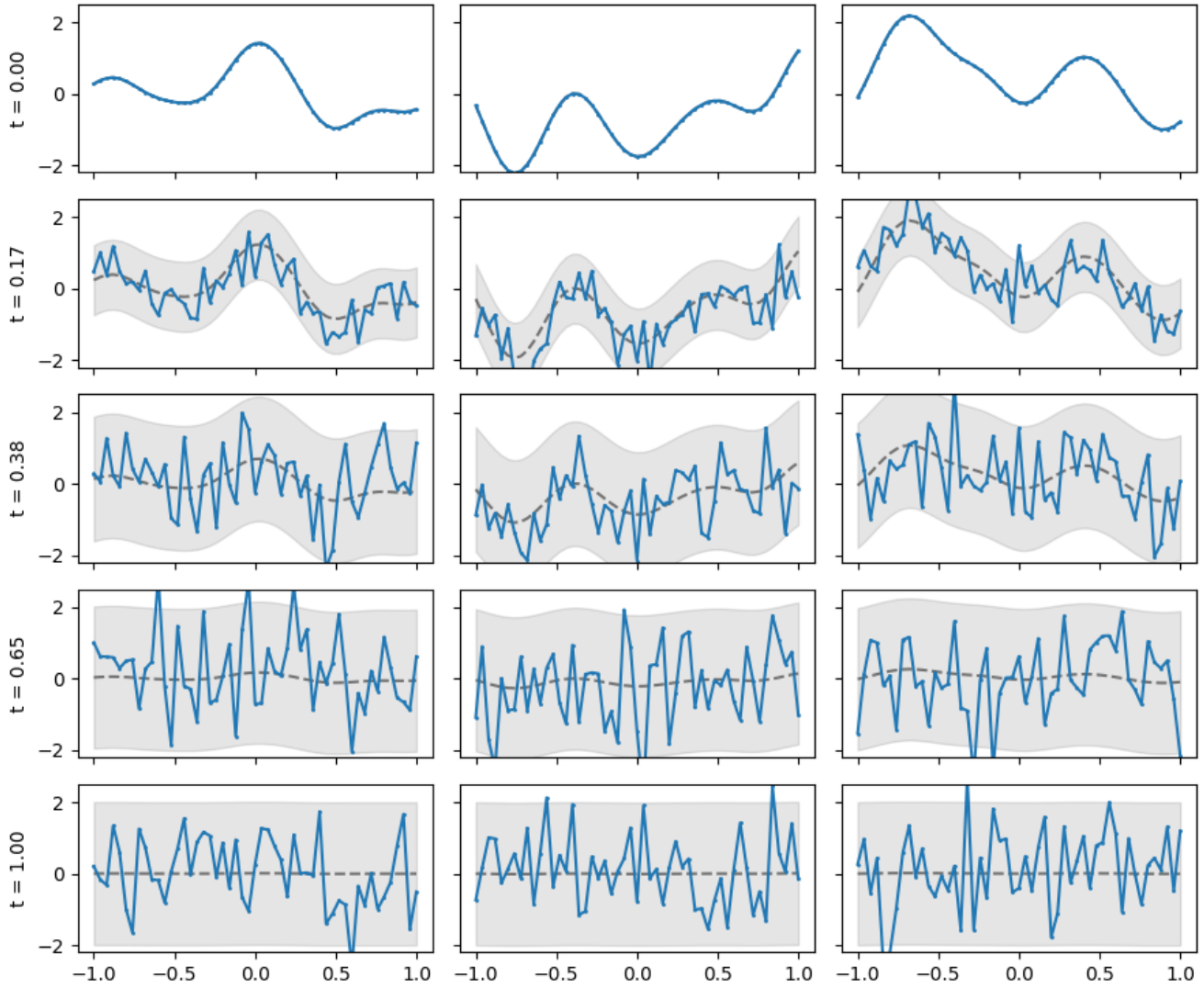
$$\mathrm{d}\boldsymbol{y}_t(\mathbf{X}) = -\frac{1}{2}\boldsymbol{y}_t(\mathbf{X})\mathrm{d}t + \mathrm{d}\mathbf{W}_t$$

Our random variable $\{\boldsymbol{y}_t\}_{t=0}^T$ is now a **function** which depends on inputs $\mathbf{X}$.

Using Fokker-Planck, we can compute the marginal density in closed-form for for this process

$$p_{t|0}(\boldsymbol{y}_t(\mathbf{X})|\boldsymbol{y}_0(\mathbf{X})) = \mathcal{N}\left(\mathrm{e}^{-\frac{1}{2}\boldsymbol{y}_0(\mathbf{X})}, (1-\mathrm{e}^{-t})\mathbf{I}\right)$$

Forward process

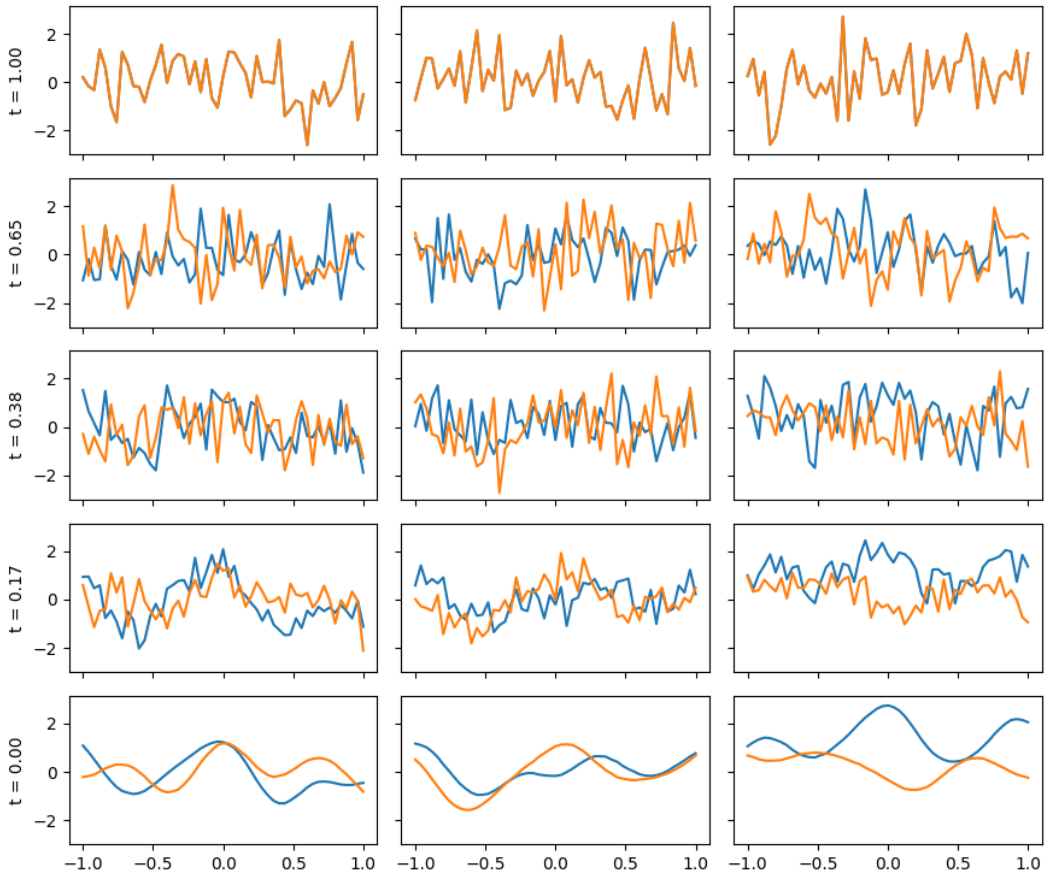


Forward NDP process

Reverse process

Generative model

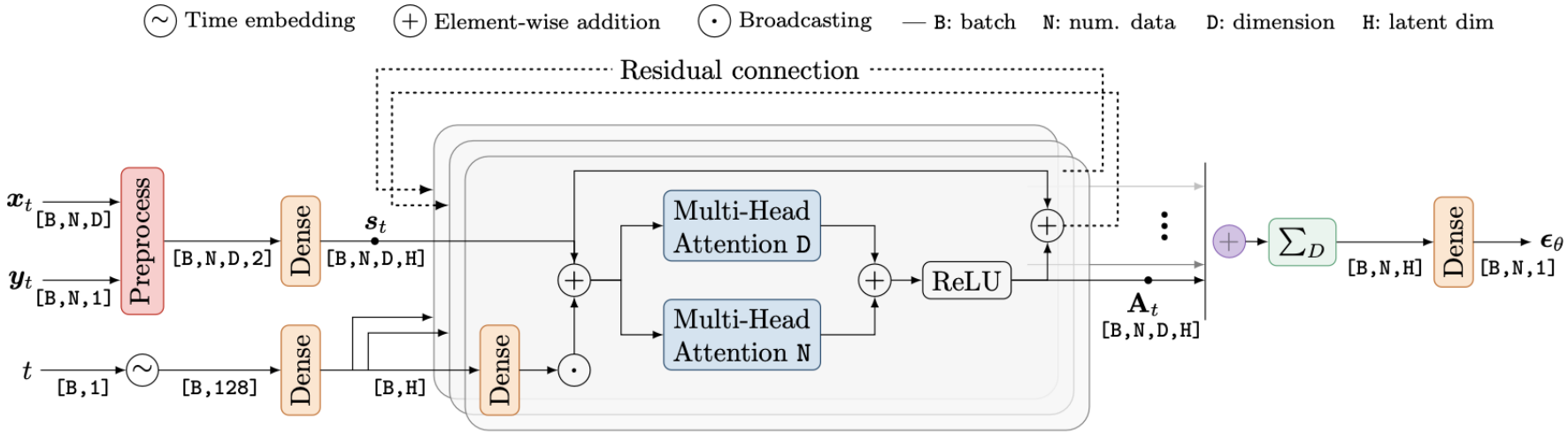
$$\mathrm{d}\boldsymbol{x}_t = \left[f(\boldsymbol{x}_t, t) - g^2(t)\, \boldsymbol{s}_{\theta^*}(\boldsymbol{x}_t, t)\right]\mathrm{d}t + g(t)\mathrm{d}\bar{\mathbf{W}}_t, \quad \boldsymbol{x}_T \sim p_{ref}$$



Learning the score

The minima of the Fisher divergence can be shown to be equivalent to

$$\theta^* = \text{argmin}_\theta \, \mathbb{E}_{\boldsymbol{y}_0, \boldsymbol{y}_t} \left[\left\| \boldsymbol{s}_\theta(\boldsymbol{y}_t, \mathbf{X}, t) - \nabla_{\boldsymbol{y}_t} \log p_{t|0}(\boldsymbol{y}_t | \boldsymbol{y}_0) \right\|^2 \right]$$



Score network architecture

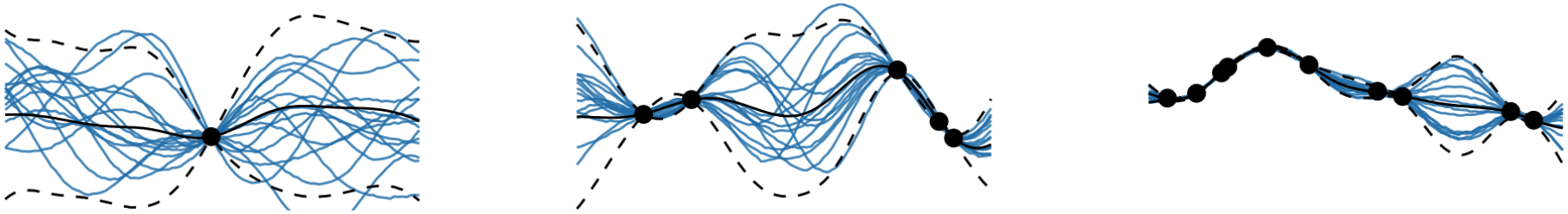
Encode properties of stochastic processes in score network $\boldsymbol{s}_\theta$:

- dimensionality invariance
- exchangeability

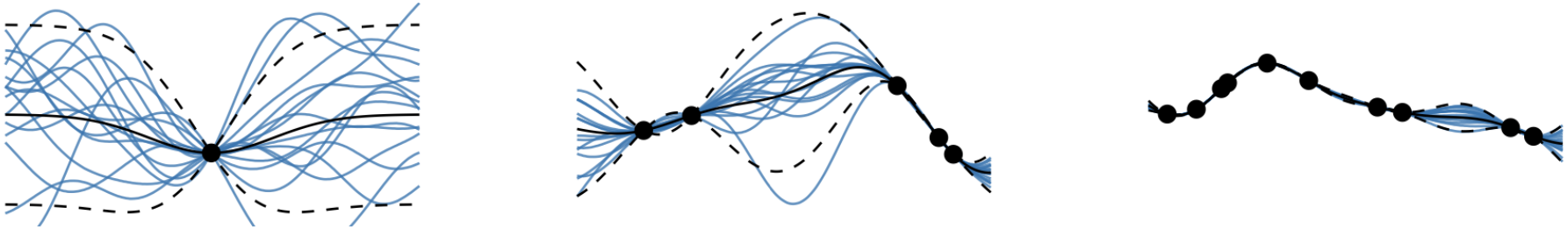
Predictions

We use an algorithm similar to the one used for image inpainting by Lugmayr et al. (2022).

This allows us to *condition* samples on observed data:



Neural Diffusion Process



Gaussian Processes

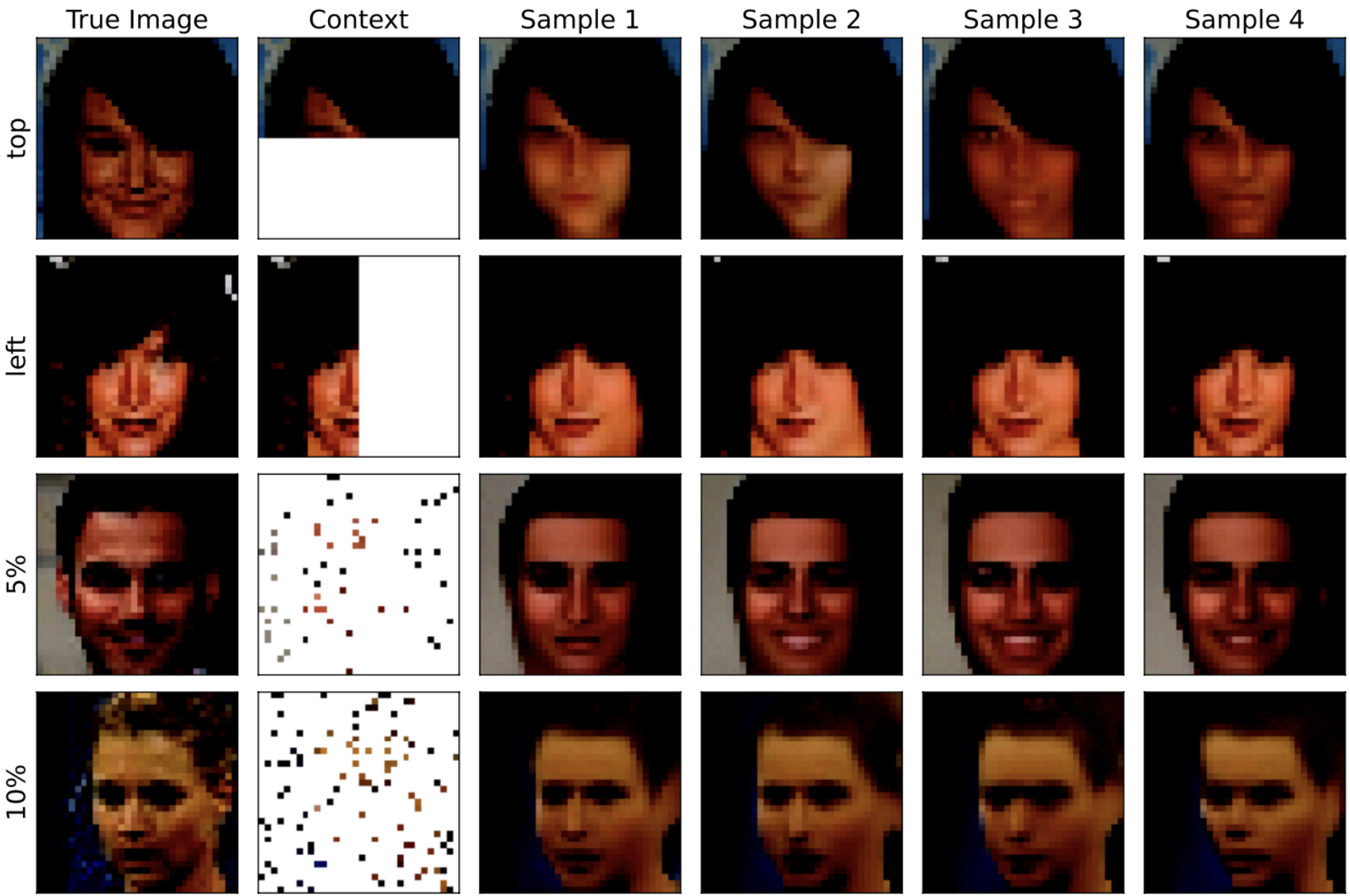
Experiment: Capturing *non-Gaussian* posteriors

Consider the following distribution over functions. Let $a \sim \mathcal{U}[-1, 1]$ then

$$f(x) = 0.0 \text{ if } x < a, \text{ else } 1.0$$

Experiment: Image Regression

Learning complex covariances from *data*.


True Image
Context
Sample 1
Sample 2
Sample 3
Sample 4
top
left
5%
10%

Thank you for your attention.

References

Dutordoir, Vincent, Alan Saul, Ghahramani Zoubin, and Fergus Simpson. 2022. "Neural Diffusion Processes." arXiv.

Ho, Jonathan, Ajay Jain, and Pieter Abbeel. 2020. "Denoising Diffusion Probabilistic Models." *Advances in Neural Information Processing Systems* 33: 6840–51.

Hyvärinen, Aapo. 2005. "Estimation of Non-Normalized Statistical Models by Score Matching." *Journal of Machine Learning Research* 6 (24): 695–709. http://jmlr.org/papers/v6/hyvarinen05a.html.

Lugmayr, Andreas, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. 2022. "RePaint: Inpainting Using Denoising Diffusion Probabilistic Models." arXiv. https://doi.org/10.48550/arXiv.2201.09865.

Song, Yang, and Stefano Ermon. 2019. "Generative Modeling by Estimating Gradients of the Data Distribution." *Advances in Neural Information Processing Systems* 32.

Song, Yang, Sahaj Garg, Jiaxin Shi, and Stefano Ermon. 2019. "Sliced Score Matching: A Scalable Approach to Density and Score Estimation," May.

Song, Yang, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. 2021. "Score-Based Generative Modeling Through Stochastic Differential Equations." In *International Conference on Learning Representations*.

Vincent, Pascal. 2011. "A Connection Between Score Matching and Denoising Autoencoders." *Neural Computation* 23 (7): 1661–74.