Learning generative models from observations using expectation maximisation

Rozet++ 2024

Jed Homer - Bayes & AI Seminar - 17/07/2025

Learning generative models from observations using expectation maximisation

Rozet++ 2024

Jed Homer - Bayes & AI Seminar - 17/07/2025

- Result
- Problem
- Fitting $p(x)$ given only y
- Diffusion
- Algorithm

- We only have access to y, a **corrupted** realisation of a **latent** x

- It is possible to fit a model for the latents $p(x)$

Bayesian inverse problems

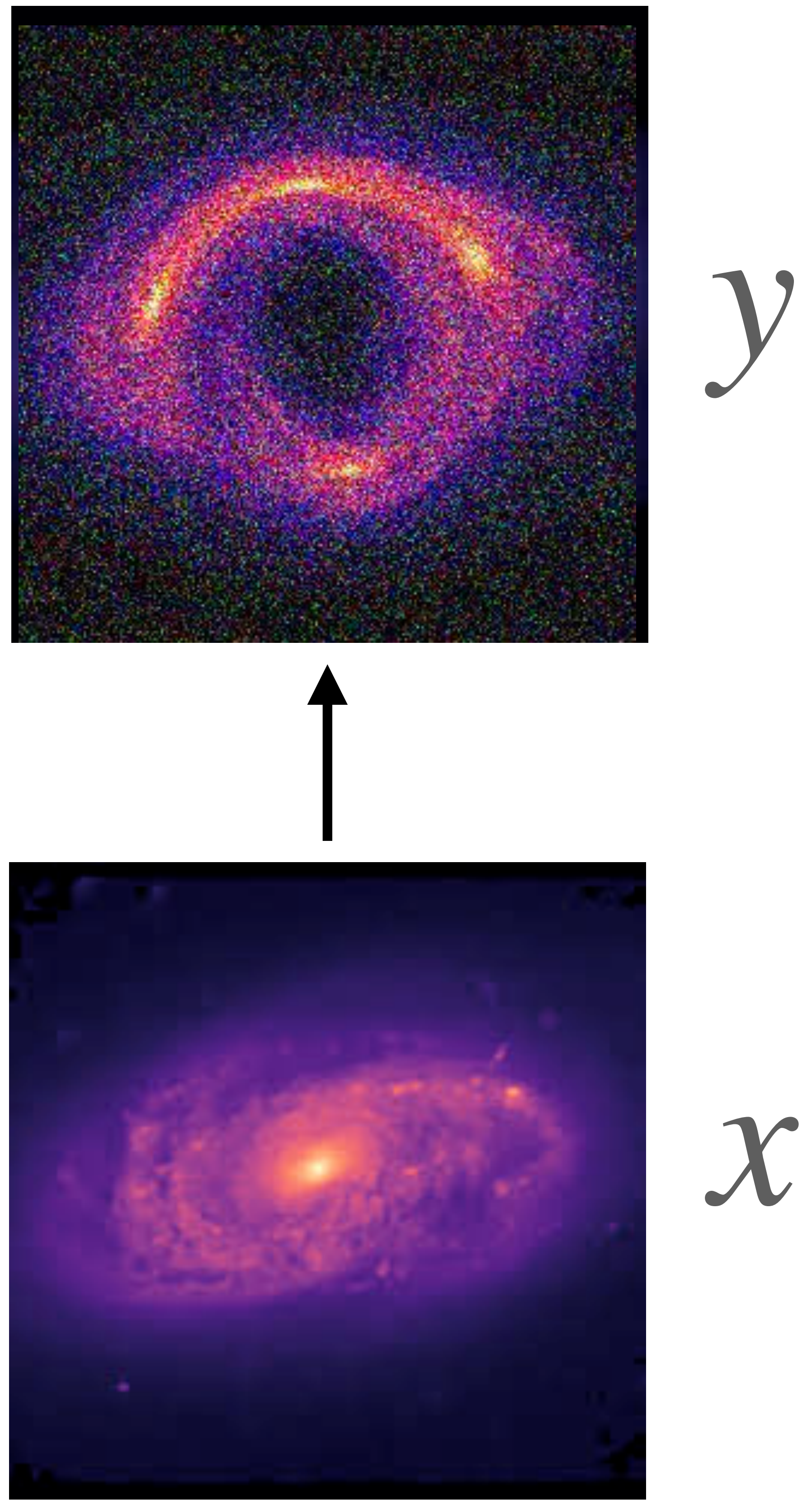
- y is a **noisy and corrupted** realisation of x

$$p(x|y) \propto p(y|x) \cdot p(x)$$

- **Assumed likelihood** of y in terms of x

$$p(y|x) = \mathcal{G}[y|\mathrm{A}x, \Sigma_y]$$

...which may be different for each x



What quantifies a good prior $p(x)$?

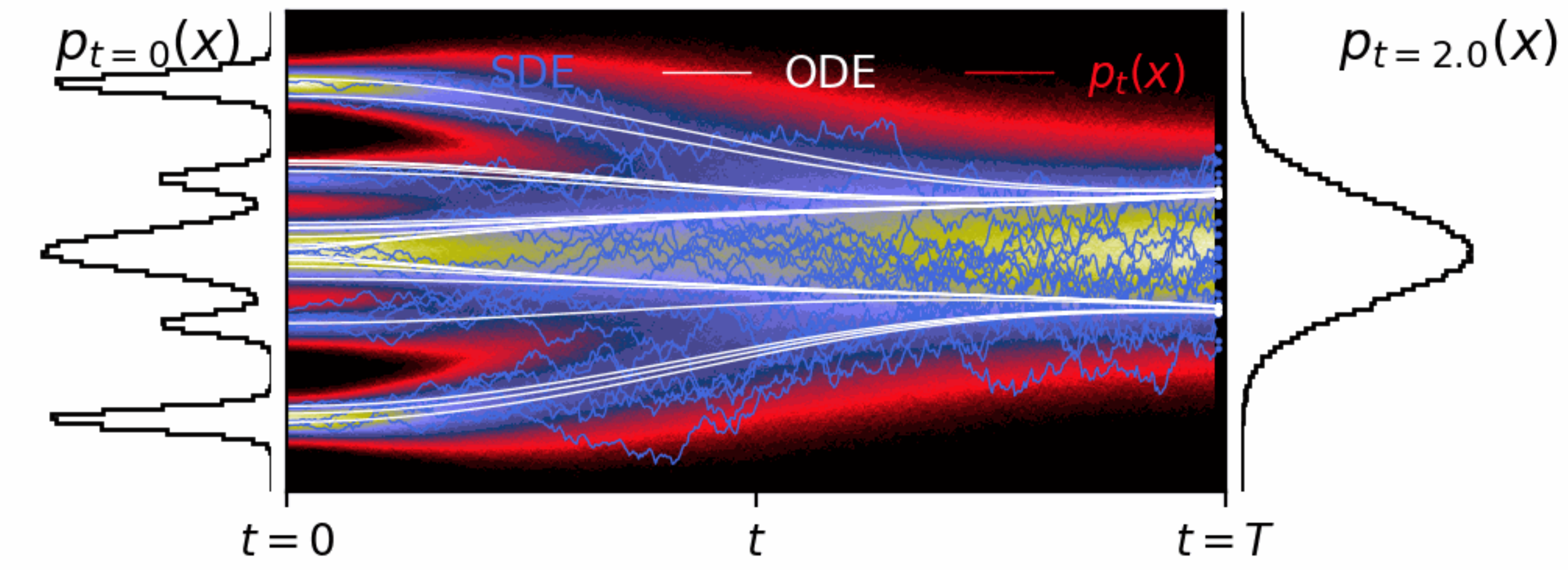
- The **evidence of the data** given a **prior** $p(x)$ and a **likelihood** $p(y|x)$

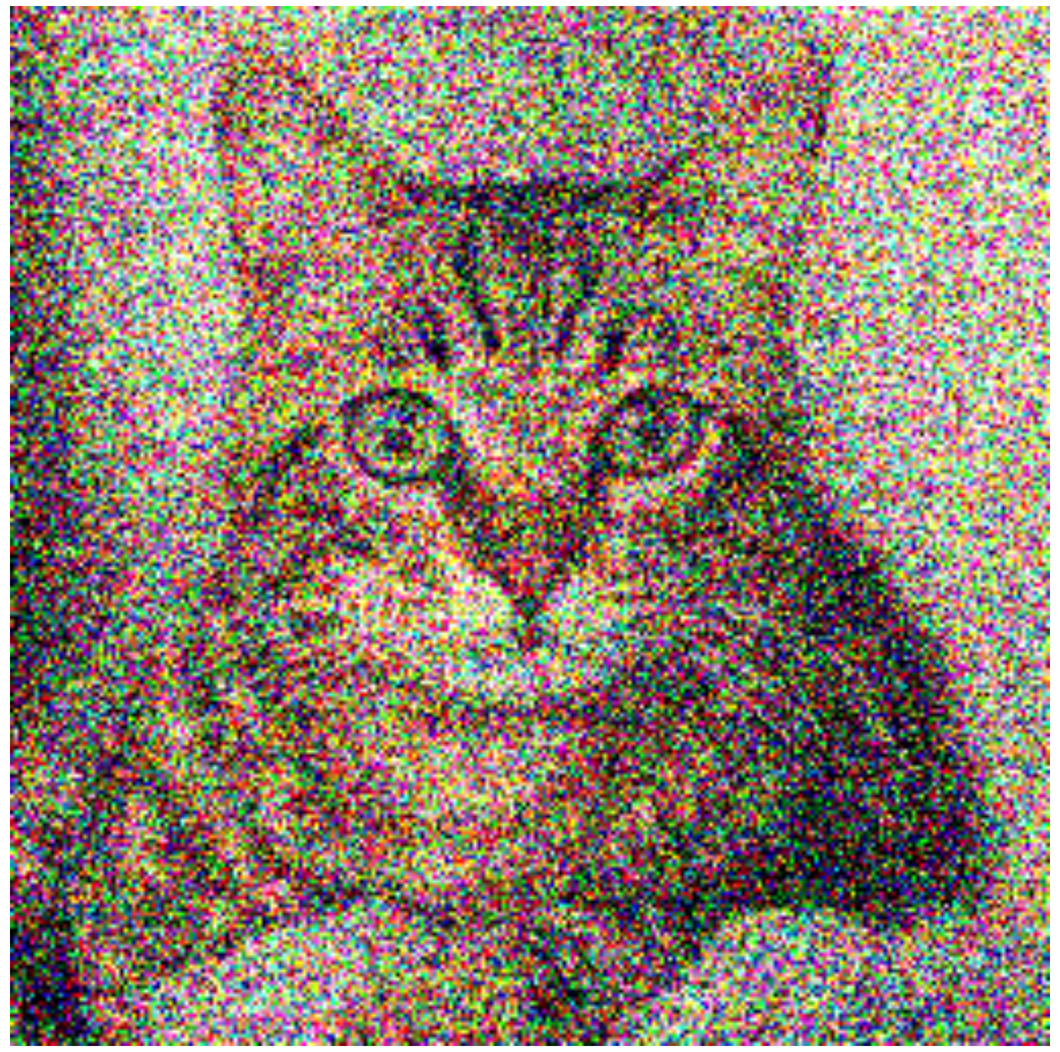
$$p(y) = \int \mathrm{d}x \, p(y|x)p(x)$$

- What is a good parameterisation for $p(x)$ given high-dimensional x?

Diffusion models: a form for $p(x)$

$p(x_t \mid x)$ ⟶





⟵ $p_\theta(x \mid x_t)$

- At each time t minimise

$$\theta^* = \min_\theta \mathbb{E}_x\left[\mathbb{E}_{x_t|x}\left[\| \nabla_{x_t} \log p(x \mid x_t) - \nabla_{x_t} \log p_\theta(x_t; t) \|_2^2\right]\right]$$

homerjed/sbgm

How to fit to a model for $p(x)$ given only y?

- This is a famously difficult task known as **density deconvolution**

  - *Extreme density deconvolution* [Bovy++2009],
  - *AmbientDiffusion* [Darras++2023, ++2024],
  - *Flow Density Deconvolution* [Dockhorn++2020],
  - *Noise2NoiseFlow* [Maleky++2022].

How to fit to a model for $p(x)$ given only y?

- Maximise the **model evidence of the data** given the model prior

$$\theta^* = \min_{\theta} \mathcal{D}_{KL}[\,p(y) \,\|\, p_\theta(y)]$$

- Expectation maximisation: guaranteed to monotonically increase over iterations

Expectation Maximisation

- Using samples $y \sim p(y)$, generate a training set for x to fit $p_\theta(x)$

$$\pi_k(x) = \int \mathrm{d}y \; p_{\theta_k}(x \,|\, y) p(y)$$

- Maximise the **model evidence of the data** given the model prior

$$\theta_{k+1} = \min_\theta \mathscr{D}_{KL}[\pi_k(x) \,\|\, p_\theta(x)]$$

- Guaranteed to monotonically increase over iterations (local minimum)

Diffusion posterior sampling for x

- What is $p_\theta(x \mid y)$? Why do we need it?

- Given that our model $p(y \mid x)$ is analytic, we can sample $p_\theta(x \mid y)$ using Bayes

$$\nabla_{x_t} \log p(x_t \mid y) = \nabla_{x_t} \log p(y \mid x_t) + \nabla_{x_t} \log p(x_t)$$

homerjed/sbgm

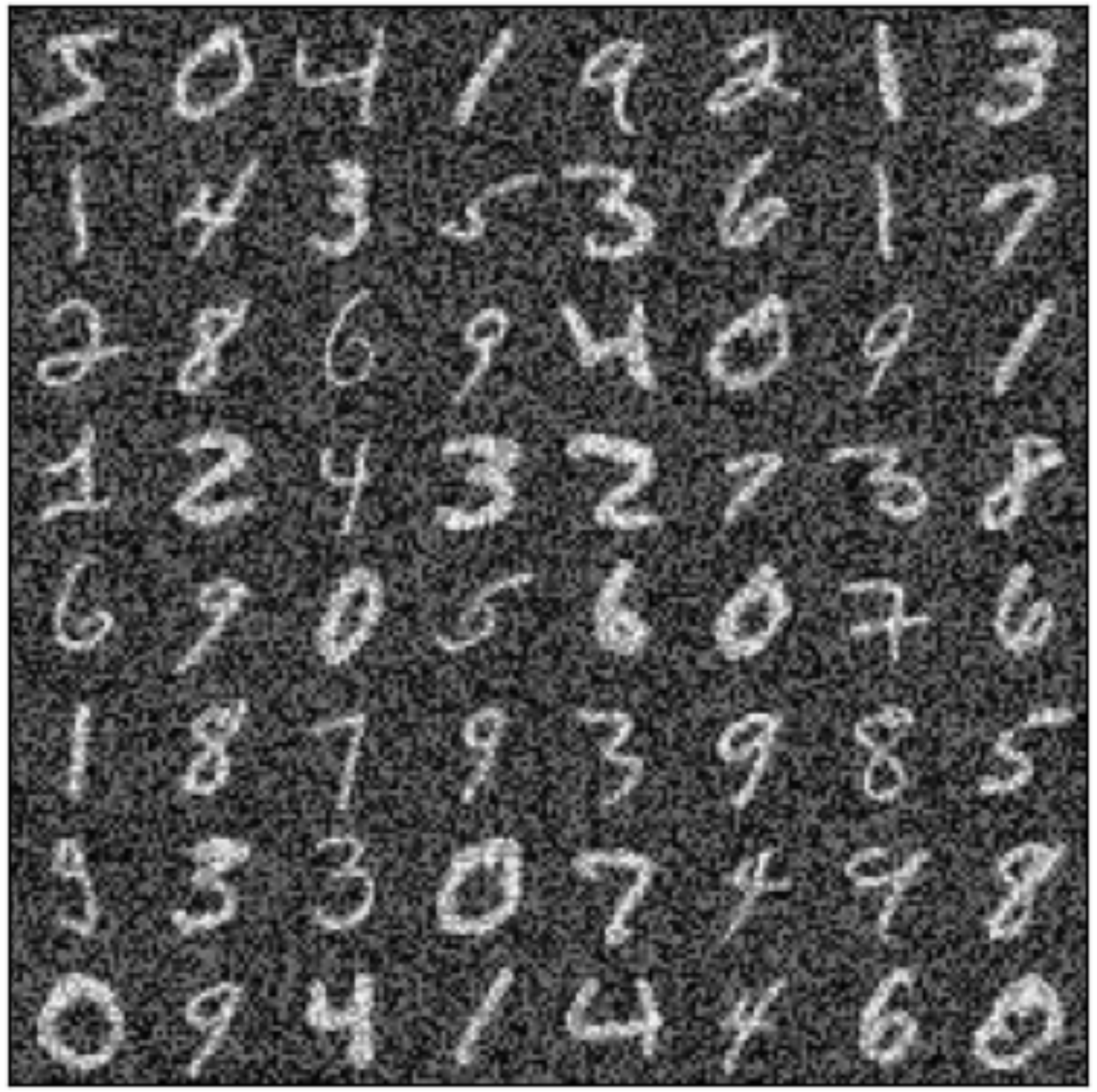
Algorithm

- dataset $\{y\}$ # dataset is fixed
- initialise θ_0
- for k in $1:K$:
  - get batch y
  - sample $x|y,\theta_k$
  - minimise diffusion loss for x
  - update model parameters θ_k
- return θ_K

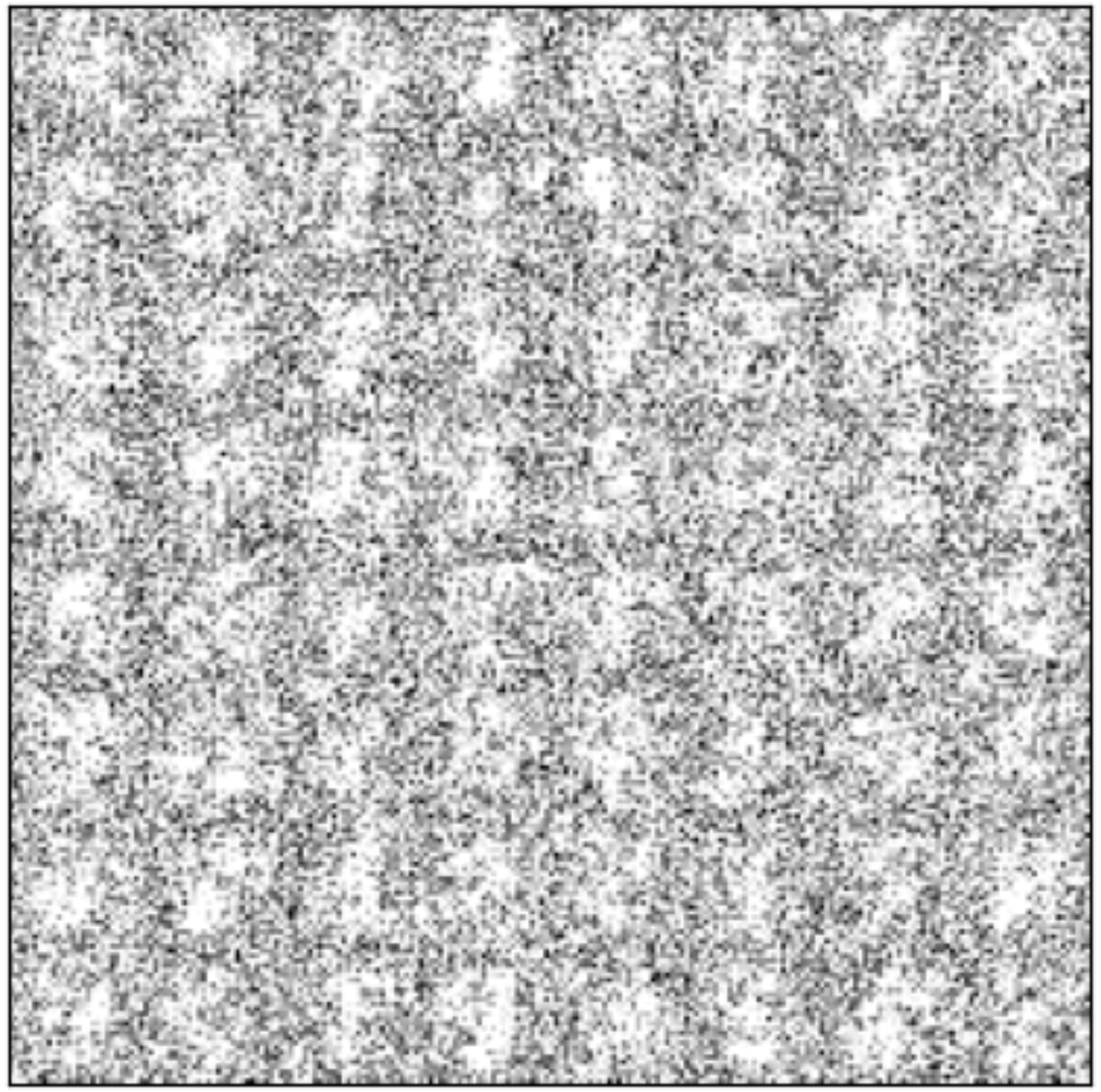
homerjed/sbgm

Example: MNIST

- 40 minutes of training, no GPU, transformer-based diffusion



$$y \sim \mathcal{G}[y \,|\, x, \Sigma_y]$$



$$x \sim p_\theta(x \,|\, y)$$

homerjed/emflows

Details I did not cover

- **Accurate diffusion posterior sampling**
  - Bottleneck of this method

$$\nabla_{x_t} \log p(x_t \mid y) = \nabla_{x_t} \log p(y \mid x_t) + \nabla_{x_t} \log p(x_t)$$

  - Only an **approximation** to this term exists...

$$q(y \mid x_t) = \int \mathrm{d}x \, p(y \mid x) p(x \mid x_t) = \mathcal{G}[y \mid \mathbb{E}[x \mid x_t], \Sigma_y + \mathbb{V}[x \mid x_t]]$$

$$\Longrightarrow \quad \nabla_{x_t} \log q(y \mid x_t) = \nabla_{x_t} \mathbb{E}[x \mid x_t]^\top \left(\Sigma + \mathbb{V}[x \mid x_t]\right)^{-1} \left(y - \mathbb{E}[x \mid x_t]\right)$$

  - Express as $Ax = b$, **CG solve** to avoid calculating $\mathbb{V}[x \mid x_t] = \Sigma_t \nabla_{x_t}^T d_\theta(x_t, t)$