Generative Modeling by Estimating Gradients of the Data Distribution

Based on

https://yang-song.net/blog/2021/score/

Song et al., 2019 (NeurIPS)

Overview

- Blog-post inspired (quite analogous to the paper)
  - Includes some ‘generativre modeling’ recap
- I’ll go fast through the overall idea to get a full picture
  - We can the clarify details
- Triple-check me

Objective of generative modeling

dataset $\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}$, underlying data distribution $p(\mathbf{x})$.

Objecive: Learn $p_\theta(\mathbf{x})$ (with option to sample)

Energy-based:

$$p_\theta(\mathbf{x}) = \frac{e^{-f_\theta(\mathbf{x})}}{Z_\theta},$$

Objective: maximum log likelihood

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

$f_\theta(\mathbf{x})$ is often called an unnormalized probabilistic model, or energy-based model [7].

Neural net predicts energy of a given state (sample).

Good: We have a formulation to train a model of our data distribution based on maximum log likelihood

Complications:

1. Z_θ **intractable**
2. **How to sample from** f_θ **?**

One solution

$$p_\theta(\mathbf{x}) = \frac{e^{-f_\theta(\mathbf{x})}}{Z_\theta},$$

intractable

- Invertible models (last week)
  - z = f_theta(x)
  - p(z) = N(0, 1; z)
  - p_theta(x) = p(z) * |det df(x)/dx|

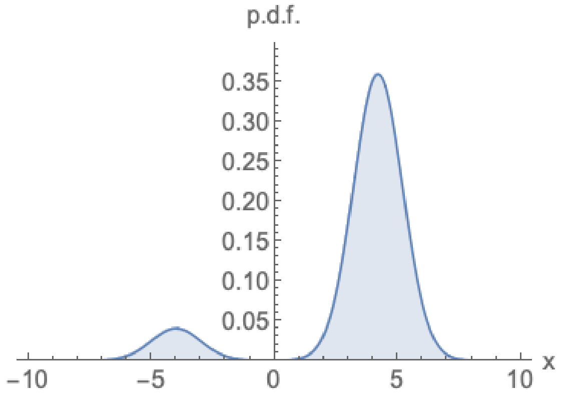
New way (today)

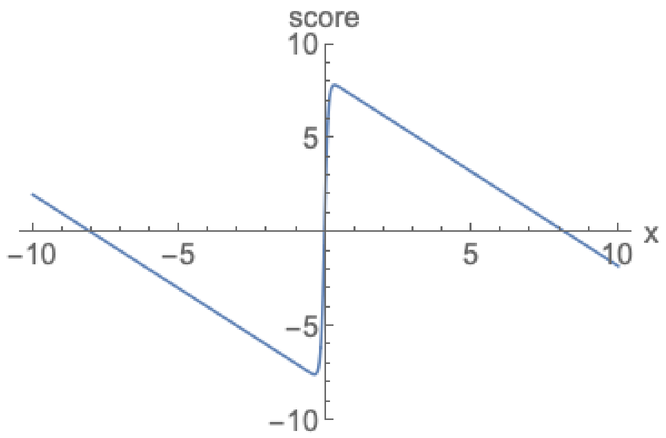
$$p_\theta(\mathbf{x}) = \frac{e^{-f_\theta(\mathbf{x})}}{Z_\theta},$$

intractable

$\nabla_{\mathbf{x}} \log p(\mathbf{x})$ is called: (Stein) score function

$$\nabla_{\mathbf{x}} \log p_\theta(\mathbf{x}) = -\nabla_{\mathbf{x}} f_\theta(\mathbf{x}) - \underbrace{\nabla_{\mathbf{x}} \log Z_\theta}_{=0} = -\nabla_{\mathbf{x}} f_\theta(\mathbf{x})$$





Remaining challenges:
- How to train this (no more log likelihood maximization)
- How to sample from it

How to learn the score function?

Minimize Fisher divergence:

$$\mathbb{E}_{p(\mathbf{x})}[||\nabla_{\mathbf{x}} \log p(\mathbf{x}) - \mathbf{s}_\theta(\mathbf{x})||_2^2]$$

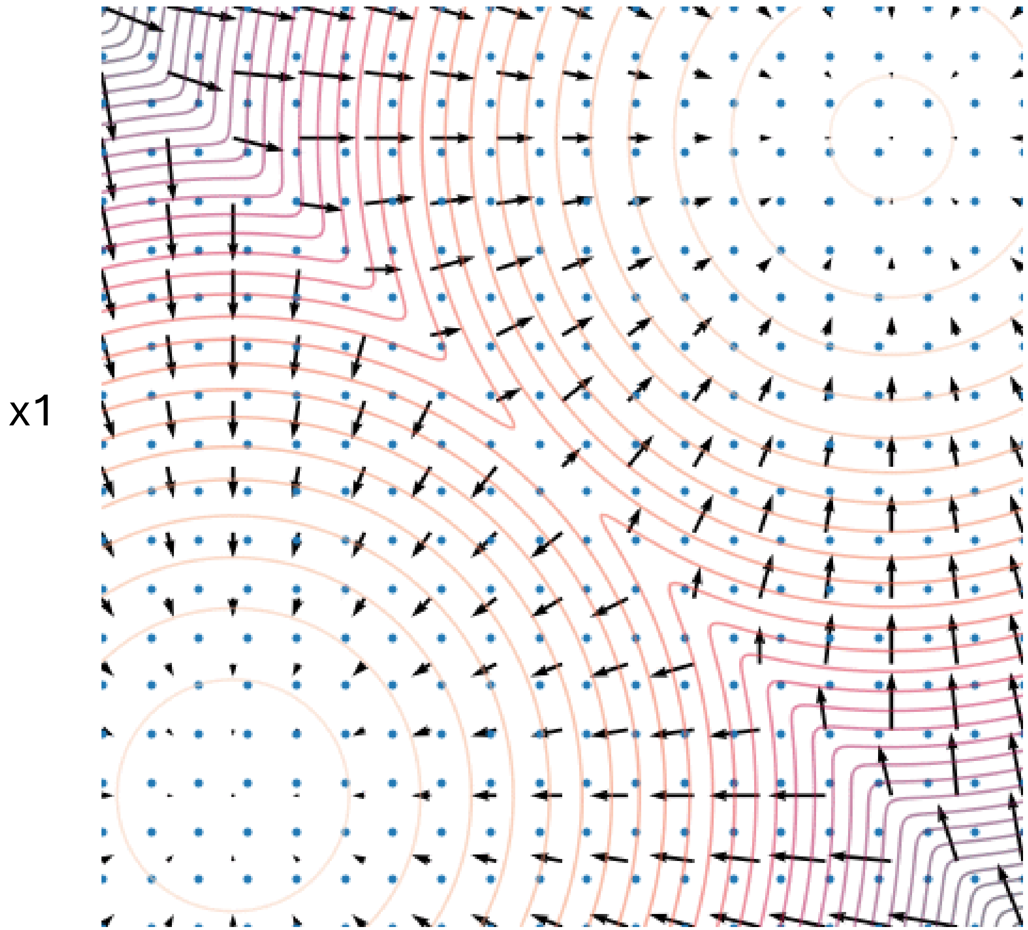
Unknown ☹

there exists a family of methods called **score matching** [3] [16, 17, 31] that minimize the Fisher divergence without knowledge of the ground-truth data score. Score matching objectives can

“Score matching objectives”: Objective that leads to a
match between scores of two distributions (p(x), p_theta(x))

Postponed: Appendix 1

How to sample?



x2

We just trained this

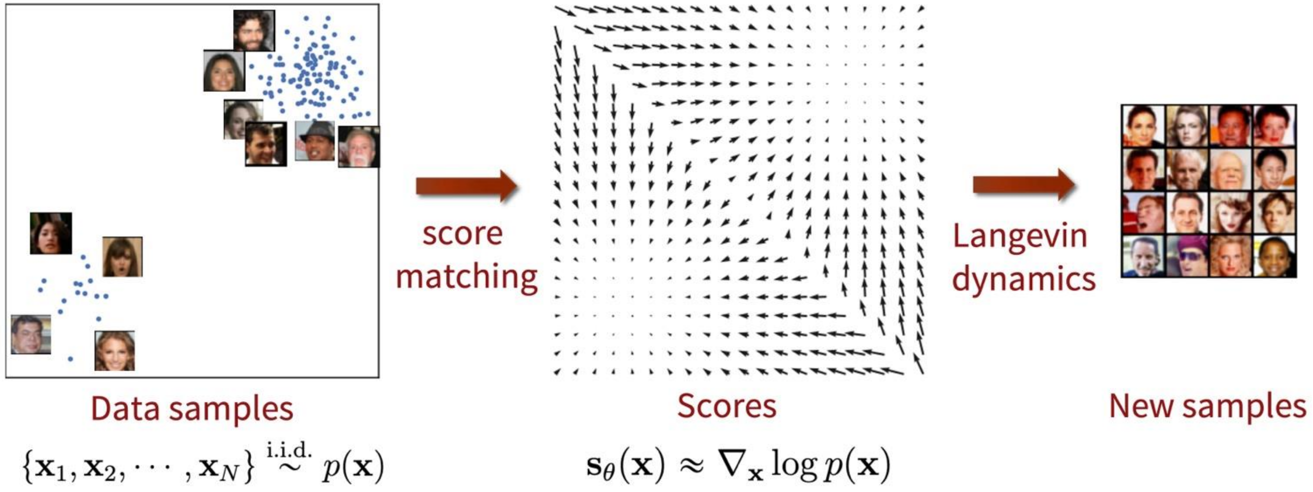
$$\mathbf{x}_0 \sim \pi(\mathbf{x}),$$

$$\mathbf{x}_{i+1} \leftarrow \mathbf{x}_i + \epsilon \nabla_{\mathbf{x}_i} \log p(\mathbf{x}) + \sqrt{2\epsilon}\, \mathbf{z}_i, \quad i = 0, 1, \cdots, K,$$

Provably converges to a sample from p(x) (if eps -> 0 and K -> inf)

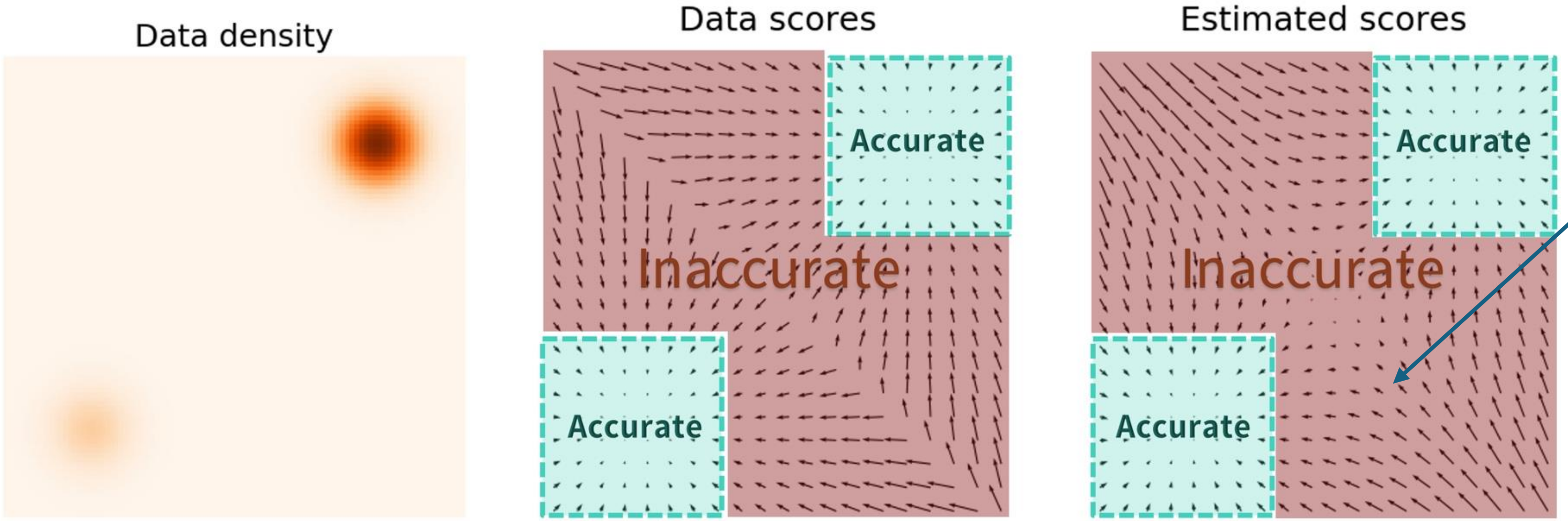
Langevin (Monte Carlo) sampling

Challenge: Low density regions -> poor score estimates



$$\mathbb{E}_{p(\mathbf{x})}[\|\nabla_\mathbf{x} \log p(\mathbf{x}) - \mathbf{s}_\theta(\mathbf{x})\|_2^2]$$

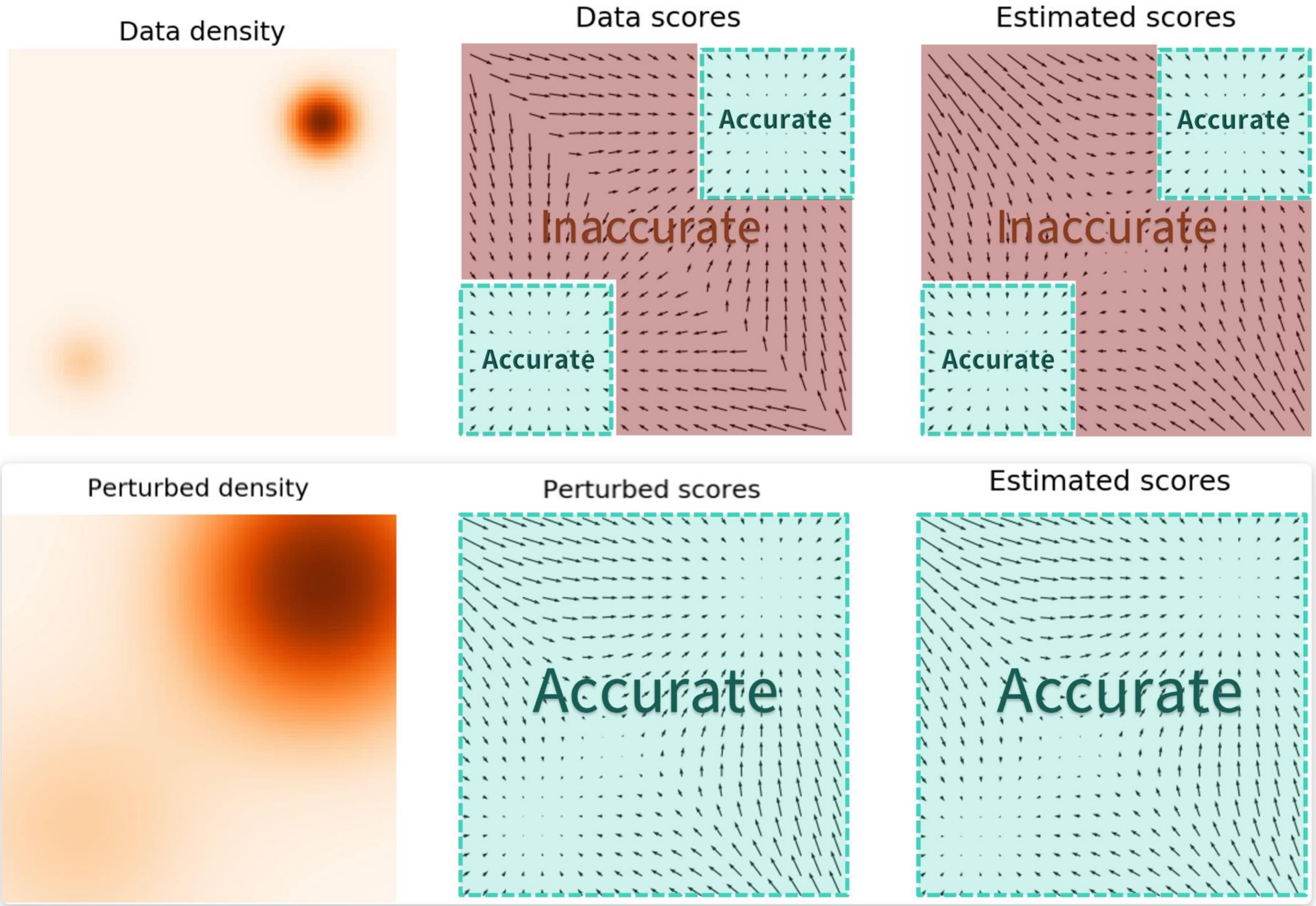
$$= \int p(\mathbf{x})\|\nabla_\mathbf{x} \log p(\mathbf{x}) - \mathbf{s}_\theta(\mathbf{x})\|_2^2 \mathrm{d}\mathbf{x}.$$



$$\mathbf{x}_0 \sim \pi(\mathbf{x}),$$

Alternative intuition: manifold hypothesis

Noise to the rescue: Intuition

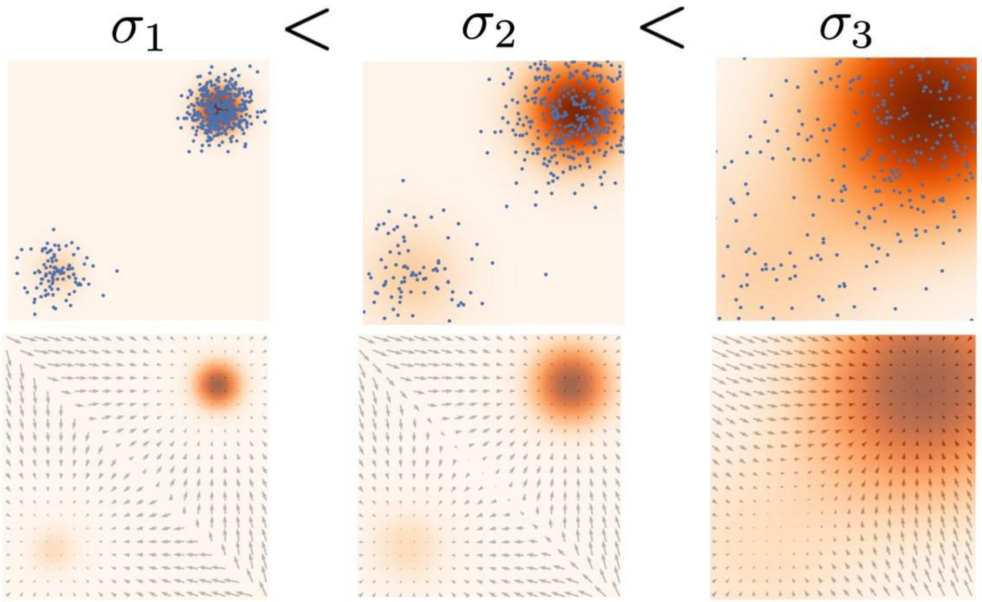


Simply add noise to samples and model

How to add noise?
- Too much: no signal
- Too little: poor space coverage

Note the 'i' in s_theta(x,i)

Noise Conditional Score-Based Model



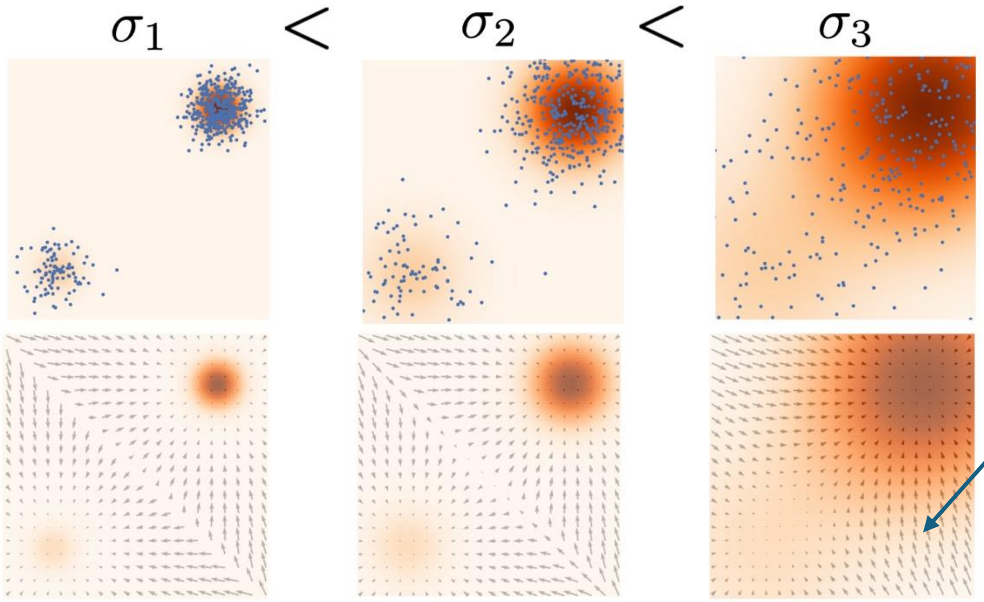
$$p_{\sigma_i}(\mathbf{x}) = \int p(\mathbf{y})\mathcal{N}(\mathbf{x}; \mathbf{y}, \sigma_i^2 I)\mathrm{d}\mathbf{y}.$$

Note that we can easily draw samples from $p_{\sigma_i}(\mathbf{x})$ by sampling $\mathbf{x} \sim p(\mathbf{x})$ and computing $\mathbf{x} + \sigma_i \mathbf{z}$, with $\mathbf{z} \sim \mathcal{N}(0, I)$.

train $\quad \mathbf{s}_\theta(\mathbf{x}, i) \approx \nabla_\mathbf{x} \log p_{\sigma_i}(\mathbf{x})$ for all $i = 1, 2, \cdots, L$.

min $\quad \sum_{i=1}^{L} \lambda(i) \mathbb{E}_{p_{\sigma_i}(\mathbf{x})}[\|\nabla_\mathbf{x} \log p_{\sigma_i}(\mathbf{x}) - \mathbf{s}_\theta(\mathbf{x}, i)\|_2^2],$ (sum of weighted Fishers) => compatible with score matching

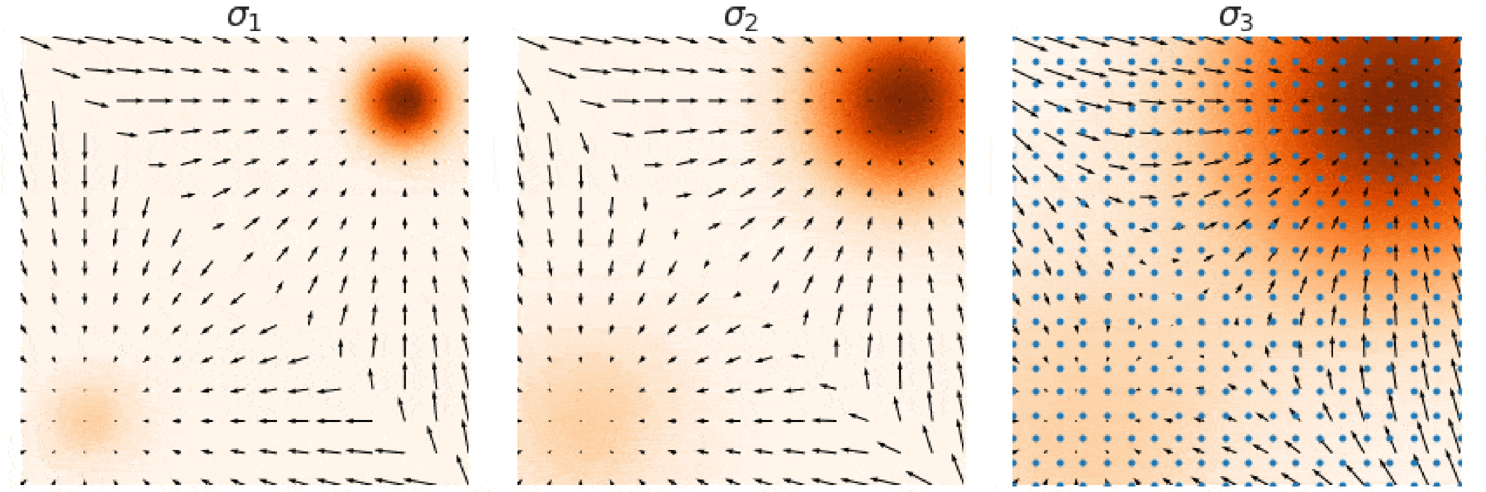
Sampling from Noise Conditional Score-Based Model



- σ_3 model is good at the beginning (but poor final result)

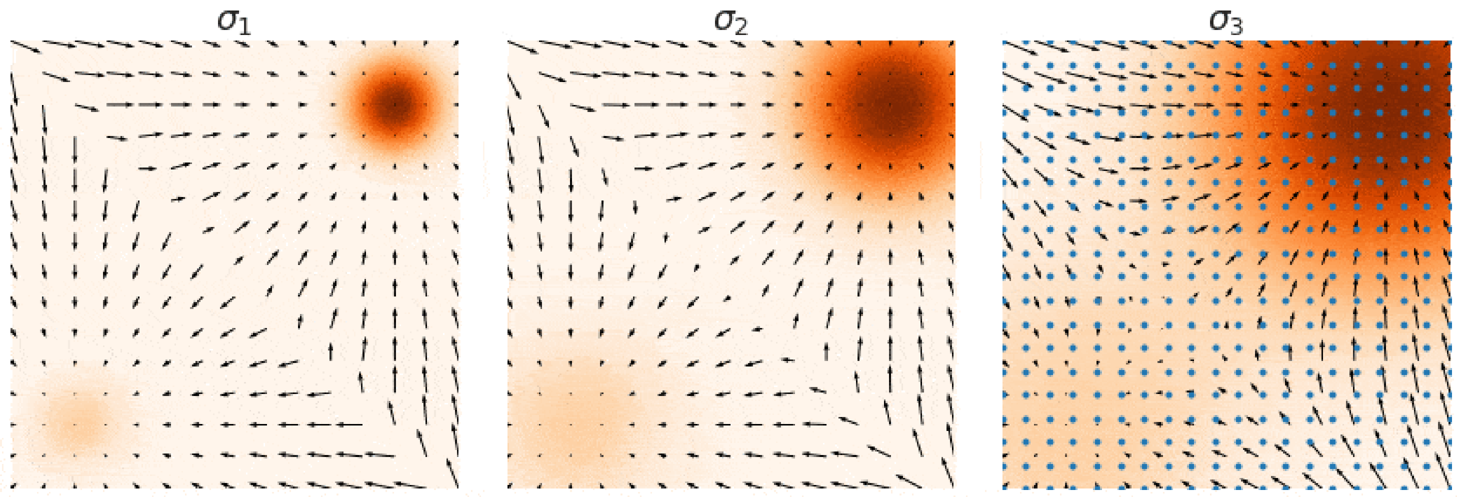
$$\mathbf{x}_0 \sim \pi(\mathbf{x}),$$

- σ_1 model is good to refine



Recap

- Model score instead of p(x)
- Training: "score matching objective" $\mathbb{E}_{p(\mathbf{x})}[\|\nabla_{\mathbf{x}} \log p(\mathbf{x}) - \mathbf{s}_\theta(\mathbf{x})\|_2^2]$
- Sampling: Langevin $\mathbf{x}_{i+1} \leftarrow \mathbf{x}_i + \epsilon \nabla_{\mathbf{x}} \log p(\mathbf{x}) + \sqrt{2\epsilon}\, \mathbf{z}_i, \quad i = 0, 1, \cdots, K,$
- Circumvent low density: Add noise



Appendix: Score matching objectives

Minimize Fisher divergence:

$$\mathbb{E}_{p(\mathbf{x})}[||\nabla_{\mathbf{x}} \log p(\mathbf{x}) - \mathbf{s}_\theta(\mathbf{x})||_2^2]$$

Unknown :(☹

Equivalent (up to a constant):

$$\mathbb{E}_{p_{\text{data}}(\mathbf{x})}\left[\operatorname{tr}(\nabla_{\mathbf{x}} \mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x})) + \frac{1}{2}\left\|\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x})\right\|_2^2\right]$$

Equivalent to "Denoising score matching":

$$\frac{1}{2}\mathbb{E}_{q_\sigma(\tilde{\mathbf{x}}|\mathbf{x})p_{\text{data}}(\mathbf{x})}[||\mathbf{s}_{\boldsymbol{\theta}}(\tilde{\mathbf{x}}) - \nabla_{\tilde{\mathbf{x}}} \log q_\sigma(\tilde{\mathbf{x}} \mid \mathbf{x})||_2^2]$$

Denoising

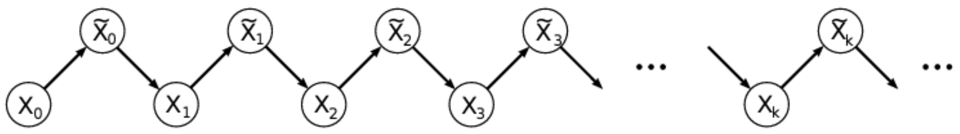
Connection to other types of models

Diffusion models

Same model family. Unifiying framework: Song 2021 ICLR, Ho 2020

	Score-based model	Diffusion probabilistic model
Perturbation	Multiple scales of noise	Multiple scales of noise
Training objective	Score matching	ELBO
Sampling	Langevin (MCMC)	Learned decoder
Unique ability	Calculate log-likelihoods exactly	Can be made faster?

Generative stochastic networks



MCMC sampling, Denoising model