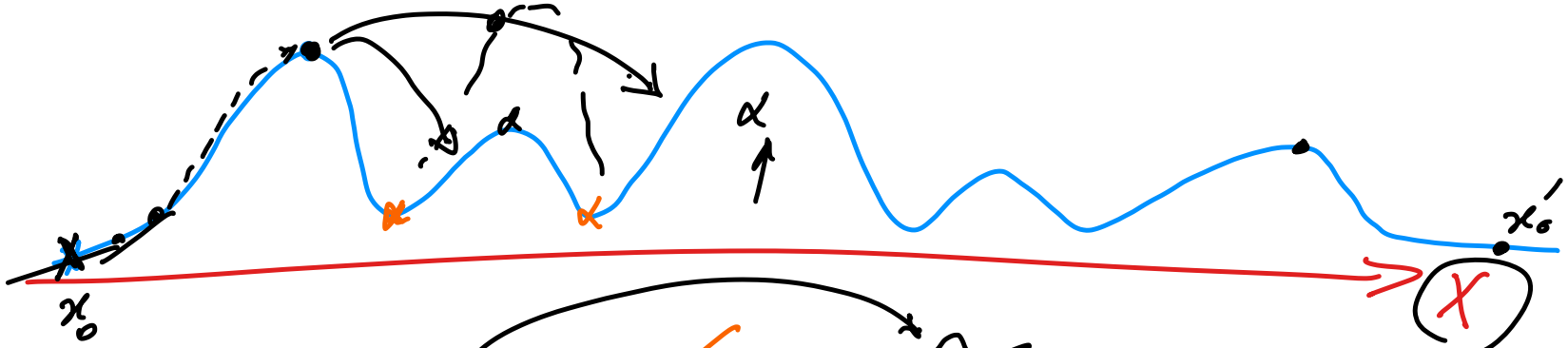


$$x_{t+1} = x_t + \alpha \nabla_x p(x) + (?)\varepsilon$$

$\nabla_x \log p(x)$

Score

← Langevin Dynamics

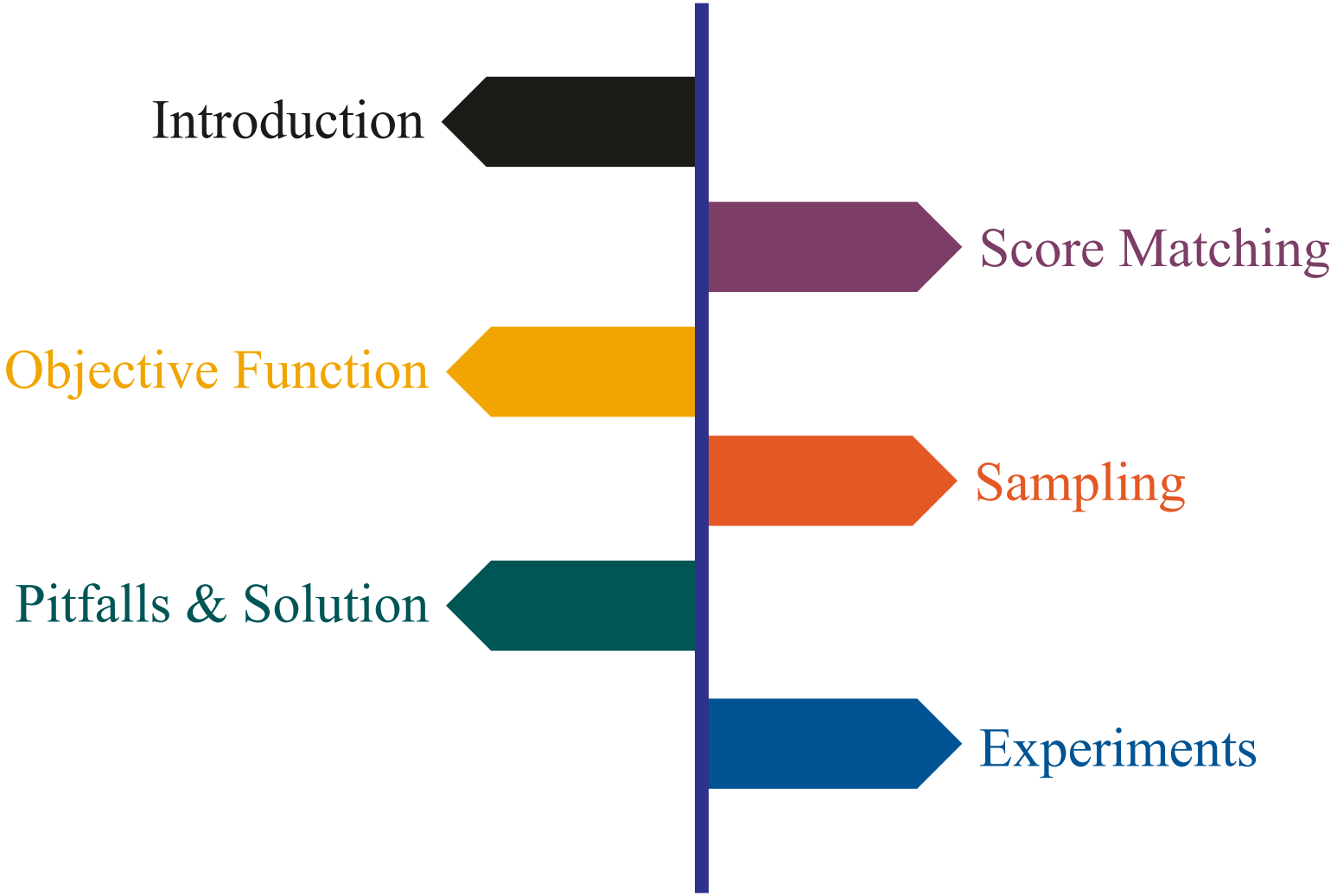
Score Matching

Deep Generative Models

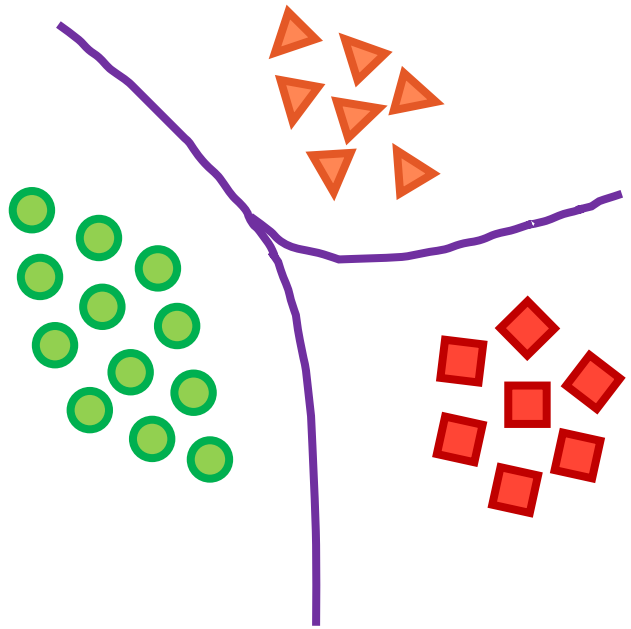
Presented by: **Ali Hedayatnia**

09-Dey-1402
Dec.-09-2023

Table of Contents

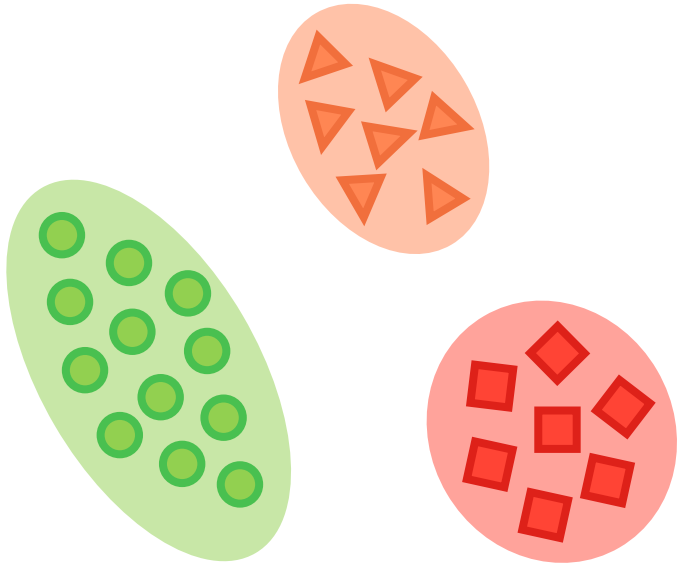


Models: Discriminative vs. Generative



Discriminative Models

Discriminate between objects



Generative Models

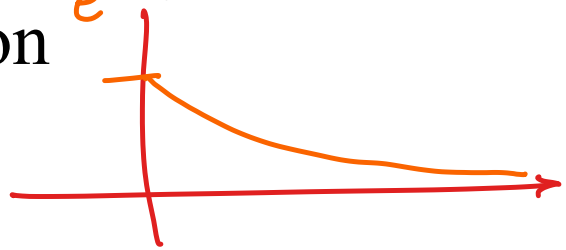
Learning distribution

Is it possible to learn a probability distribution?

- Probability Functions Property ($p(x)$):
  1. $\forall x \in X_{Data}$: $\quad 0 \leq p(x) \leq 1$
  2. $\int_{-\infty}^{\infty} p(x)\, dx = 1$
- How to learn $p(x)$?
  - Let's use neural network!!!
- Are these properties compatible with neural networks?
- Or, How can we make these properties compatible with neural networks?

Learning Distribution as a Normalized Energy Function

- A probability distribution can be represented as a normalized energy function



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

Gibbs Distribution

Partition

- Z_θ(Partition Function):

$$Z_\theta = \int_{-\infty}^{\infty} e^{-f_\theta(x)}\, dx = \int_{x_1} \int_{x_2} \cdots \int_{x_d} e^{-f_\theta(x)}\, dx_d \cdots dx_2\, dx_1$$

Learning Distribution as a Normalized Energy Function

- Is everything OK? NO!!
  - Z_θ is computationally infeasible.
- Is it possible to use MLE(Maximum Likelihood Estimation)?

$$\ln p_\theta(x) = -f_\theta(x) - \ln Z_\theta$$

$$\Rightarrow \nabla_\theta \ln p_\theta(x) = -\nabla_\theta f_\theta(x) - \nabla_\theta \ln Z_\theta$$

  - The logarithmic value of the distribution is a function of Z_θ!
- How shall we proceed with this delightful endeavor?
  - ***Score Matching*** should not be neglected

Score Matching

- What happen if we calculating $\nabla_x \ln p_\theta(x)$?

$$\nabla_x \ln p_\theta(x) = -\nabla_x f_\theta(x) - \nabla_x \ln Z_\theta = -\nabla_x f_\theta(x) \Longrightarrow$$

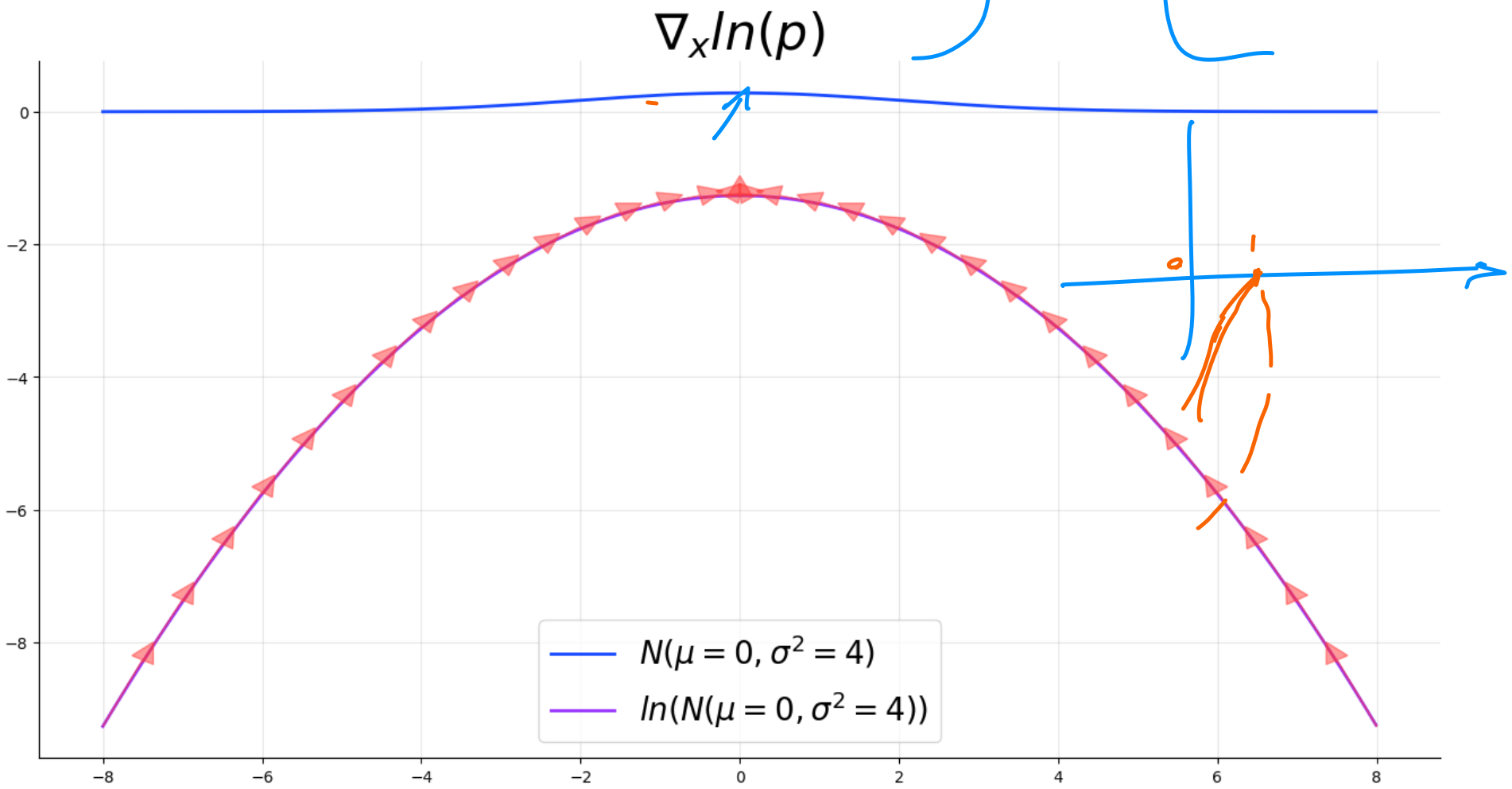
score → $\boxed{\nabla_x \ln p_\theta(x) = -\nabla_x f_\theta(x)}$

- $\nabla_x \ln p_\theta(x)$ is not a function of Z_θ ✓
- Is neural network able to learn $\nabla_x \ln p_\theta(x)$?
  - $\nabla_x \ln p_\theta(x)$ is called ***"score"***.

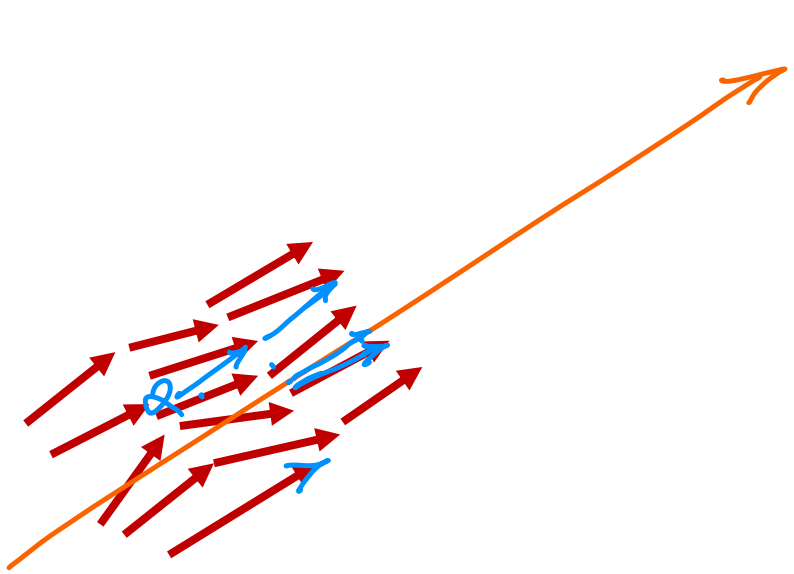
$$s_\theta(x): \mathbb{R}^D \to \mathbb{R}^D \qquad s_\theta(x) := \nabla_x \ln p_\theta(x)$$

- s_θ is a neural network that gives us $\nabla_x \ln p_\theta(x)$

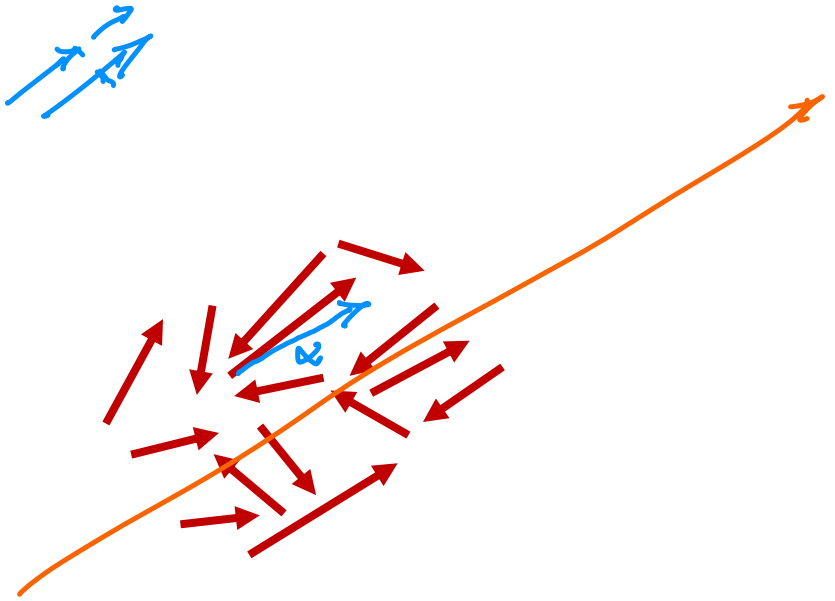
Score Matching



Objective Function



$\nabla_X \ln p_{data}(x)$



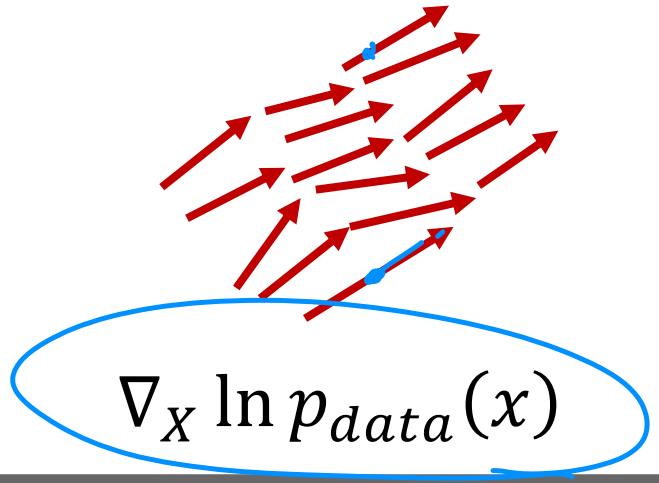
$\nabla_X \ln p_\theta(x)$

Objective Function

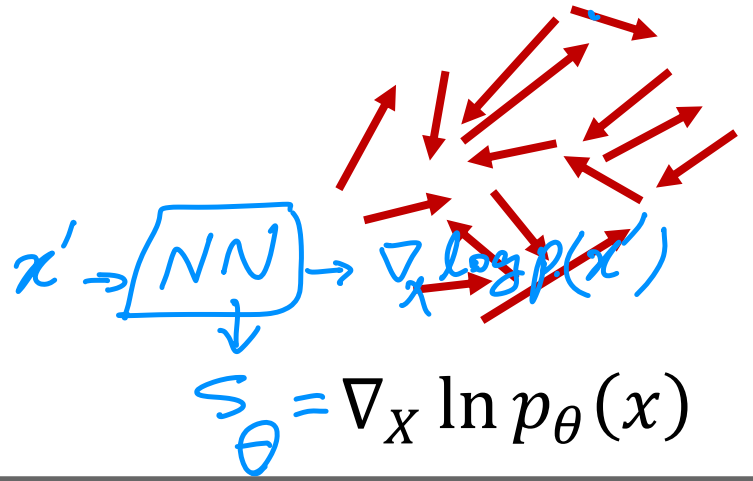
- How to align $\nabla_X \ln p_\theta(x)$ with $\nabla_X \ln p_{data}(x)$?
  - The L2-Loss function reveals the straightforward solution.

$$J(\theta) = \frac{1}{2}\mathbb{E}_{p_{data}(x)}[\|\nabla_X \ln p_{data}(x) - s_\theta(x)\|_2^2]$$

(Fisher-Divergence)



$\nabla_X \ln p_{data}(x)$



Objective Function

- Objective Function

$$J(\theta) = \frac{1}{2}\mathbb{E}_{p_{data}(x)}[\|\nabla_x \ln p_{data}(x) - s_\theta(x)\|_2^2]$$

(Fisher-Divergence)

- $J(\theta) = 0 \iff s_\theta(x) = \nabla_X \ln p_\theta(x) = \nabla_X \ln p_{data}(x)$

Local Consistency

(Score Matching ⇔ Maximum Likelihood)

- **Theorem:** Suppose that the probability density function of x satisfies $p_{data}(x) = p_\theta(x)$ for some θ^* and also that if $\theta \neq \theta^*$ then $p_\theta(x) \neq p_{data}(x)$. Suppose also that $p_\theta(x) > 0$. Then:

$$J(\theta) = 0 \iff \theta = \theta^*$$

- The consistency implies asymptotic unbiasedness.

Simplified Objective Function

- Let's analyze the objective function once more:

$$J(\theta) = \frac{1}{2}\mathbb{E}_{p_{data}(x)}[\|\nabla_x \ln p_{data}(x) - s_\theta(x)\|_2^2]$$

- Do you think there is no problem?
  - Nope!!!
  - We do not have access to the $p_{data}(x)$ and its gradient w.r.t. x.
- OMG!
  - Have we encountered an insurmountable obstacle?!!

Simplified Objective Function

- If the term in the norm is expanded, objective function can be simplified:

$$J(\theta) = \mathbb{E}_{p_{data}(x)}\left[\frac{1}{2}\|s_\theta(X)\|_2^2 + tr\left(\nabla_X s_\theta(X)\right)\right]$$

- The sample Version of J is obviously obtained from above equation as:

$$J(\theta) \approx \frac{1}{N}\sum_{x_i|_{i=1}^N \sim p_{data}(x)}\left[\frac{1}{2}\|s_\theta(x_i)\|_2^2 + tr\left(\nabla_X s_\theta(x_i)\right)\right]$$

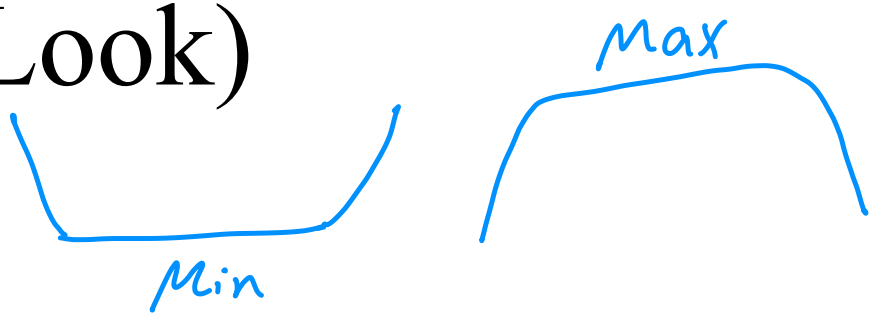
Simplified Objective Function (Closer Look)

- The objective function is formulated as a linear combination of two components:
  1. $\frac{1}{2}\|s_\theta(x_i)\|_2^2$
  2. $tr\left(\nabla_X s_\theta(x_i)\right)$

$$J(\theta) = \mathbb{E}_{p_{data}(x)}\left[\frac{1}{2}\|s_\theta(X)\|_2^2 + tr\left(\nabla_X s_\theta(X)\right)\right]$$

- Let's find out what each of these components are responsible for!

Simplified Objective Function (Closer Look)



Objective Function

1. $\frac{1}{2}\|s_\theta(x_i)\|_2^2$

$\nabla_x \log p(x)$

- Intuition:
  - Minimizing $s_\theta(x) = \nabla_X \ln p_\theta(x) \Longrightarrow$ Moving towards a stationary point
- Computation:
  - Easy
  - Fast

Simplified Objective Function (Closer Look)

Objective Function

2. $tr\left(\nabla_X s_\theta(x_i)\right)$

- Intuition:
  - Minimizing $tr\left(\nabla_X s_\theta(x_i)\right) = tr(\nabla_X^2 \ln p_\theta(x)) \Longrightarrow$ Finding Local Maxima
- Computation:
  - Hard
  - Slow (for d-dimentional data: O(d))

Simplified Objective Function (Weakness)

- Weakness:
  - $p_{data}(x)$ must be differentiable.
    - It can be reduced to Scale Score Matching.
  - Computing the loss function for data with high dimensionality is computationally infeasible.
- Solutions:
  - Reducing input data dimensionality
    - What?!!!!
  - Denoising Score Matching
  - Sliced Score Matching

Denoising Score Matching

Objective Function

- If the variable $\tilde{x}$ is derived by perturbing x through the application of a smoothing Gaussian kernel, the resulting distribution of $\tilde{x}$ is as follows:

$$\tilde{x} = x + \varepsilon_{(0,\sigma^2)}$$

$$p_\sigma(x, \tilde{x}) = p_\sigma(\tilde{x}|x)p_0(x)$$

$$\frac{\partial \log p_\sigma(\tilde{x}|x)}{\partial \tilde{x}} = \frac{1}{(2\pi)^{\frac{d}{2}}} \sigma^{-d} \exp\left(-\frac{\|\tilde{x} + x\|^2}{2\sigma^2}\right)$$

"*A Connection Between Score Matching and Denoising Autoencoders*", Pascal Vincent

Denoising Score Matching

Objective Function

- It can be shown that the simplified score matching objective is equivalent to the below objective:

$$J_{DSM_{p_\sigma}} = \mathbb{E}_{q_\sigma(x,\tilde{x})}\left[\frac{1}{2}\left\| s_\theta(\tilde{x}) - \frac{\partial \log p_\sigma(\tilde{x}|x)}{\partial \tilde{x}} \right\|^2\right]$$

$$\frac{\partial \log p_\sigma(\tilde{x}|x)}{\partial \tilde{x}} = \frac{(x-\tilde{x})}{\sigma^2}$$

$$\Rightarrow J_{DSM_{p_\sigma}} = \mathbb{E}_{q_\sigma(x,\tilde{x})}\left[\frac{1}{2}\left\| s_\theta(\tilde{x}) - \frac{(x-\tilde{x})}{\sigma^2} \right\|^2\right]$$

"*A Connection Between Score Matching and Denoising Autoencoders*", Pascal Vincent

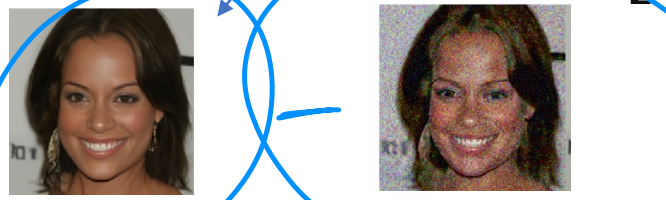
Denoising Score Matching

Objective Function

- $s_\theta(\tilde{x})$ tries to estimate the noise that was added to produced $\tilde{x}$

$$J_{DSM_{p_\sigma}} = \mathbb{E}_{q_\sigma(x,\tilde{x})}\left[\frac{1}{2}\left\| s_\theta(\tilde{x}) - \frac{(x-\tilde{x})}{\sigma^2}\right\|^2\right]$$

$$\Longrightarrow J_{DSM_{p_\sigma}} = \mathbb{E}_{p(x,\tilde{x})}\mathbb{E}_{q_\sigma(\tilde{x}|x)}\left[\frac{1}{2}\left\| s_\theta(\tilde{x}) - \frac{(x-\tilde{x})}{\sigma^2}\right\|^2\right]$$



"*A Connection Between Score Matching and Denoising Autoencoders*", Pascal Vincent

Sliced Score Matching

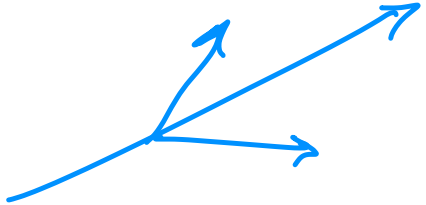
Objective Function

- The Fisher divergence score matching loss is:

$$J(\theta) = \frac{1}{2}\mathbb{E}_{p_{data}(x)}[\|\nabla_x \ln p_{data}(x) - s_\theta(x)\|_2^2]$$

- The Fisher divergence score matching loss can be approximated by projecting each gradient to a random vector:

$$J(\theta) \approx \frac{1}{2}\mathbb{E}_{p_v(x)}\mathbb{E}_{p_{data}(x)}[\|v^T\nabla_x \ln p_{data}(x) - v^T s_\theta(x)\|_2^2]$$



Sliced Score Matching

Objective Function

- It can be shown the projected approximation is equivalent to the below loss:

$$J(\theta) = \frac{1}{2}\mathbb{E}_{p_v(x)}\mathbb{E}_{p_{data}(x)}\left[v^T\nabla_x s_\theta(x)v + \frac{1}{2}(v^T s_\theta(x))^2\right]$$

Sampling

Sampling

- A neural network can be used to compute $\nabla_x \ln p_\theta(x)$, but the sampling method with it remains unclear.
  - For sampling, Just move in the direction of the gradient.
- How?
  - With an MCMC process called Langevin Dynamic

$$x_{t+1} := x_t + \epsilon \nabla_x \log p(x_t) + \sqrt{2\epsilon} z_{t+1} \quad where \quad z_i \sim \mathcal{N}(0, I)$$

$x_0, x_1, \ldots, x_T$

- ϵ: step size

Sampling

Algorithm 1- Langevin Dynamic Sampling with Score Function

Input: $\nabla_x \log p_\theta(\cdot)$, $\boldsymbol{x}$, $\boldsymbol{T}$

~~*Score Function*~~

1: $\tilde{\boldsymbol{x}}_0 \sim \pi(\boldsymbol{x})$

2: **for** $t \leftarrow 1$ **to** T

3: $\boldsymbol{z_t} \sim \mathcal{N}(0, I)$

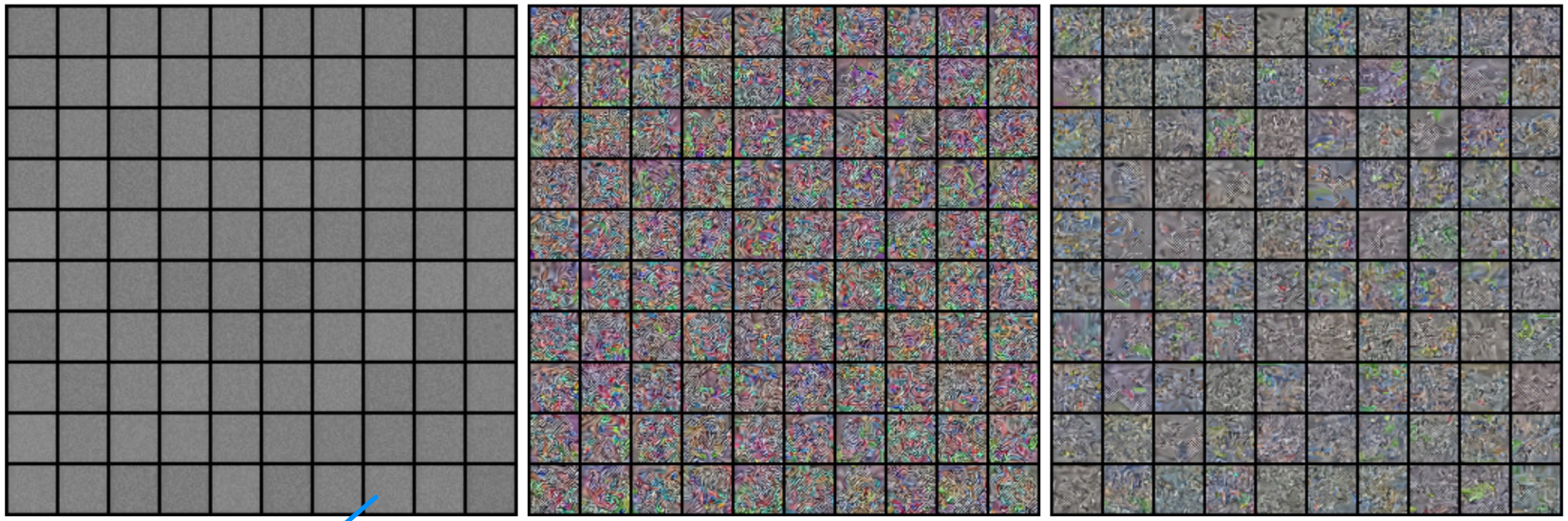
4: $\tilde{\boldsymbol{x}}_t := \tilde{\boldsymbol{x}}_{t-1} + \frac{\epsilon}{2} \nabla_x \log p_\theta(\tilde{\boldsymbol{x}}_{t-1}) + \sqrt{\epsilon} \boldsymbol{z_t}$

5: **end for**

6: **return** $\tilde{\boldsymbol{x}}_T$

Results

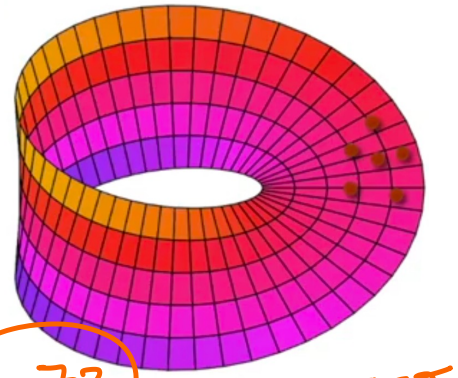
Sampling

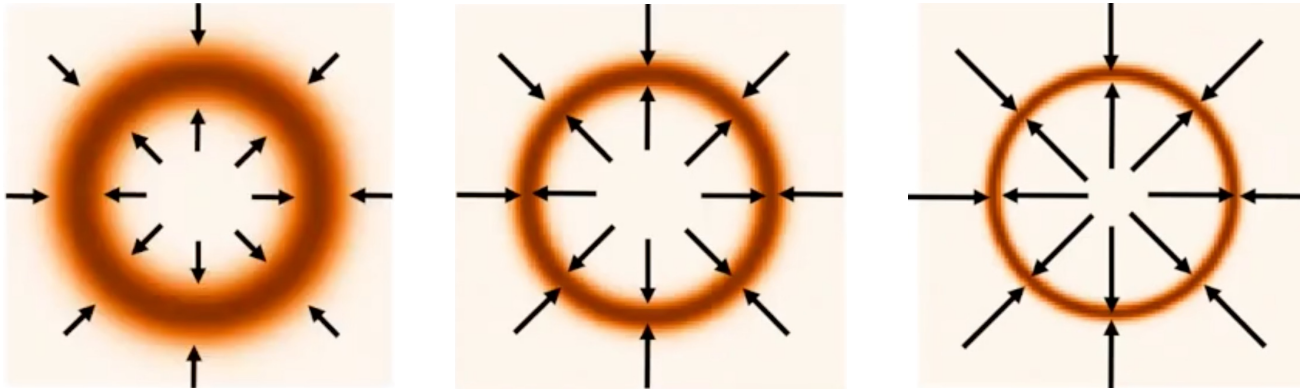


(a) MNIST (b) CelebA (c) CIFAR-10

Pitfall (I)

- Data lies on a lower dimensional manifold
  - Data score is undefined in the ambient dimension.

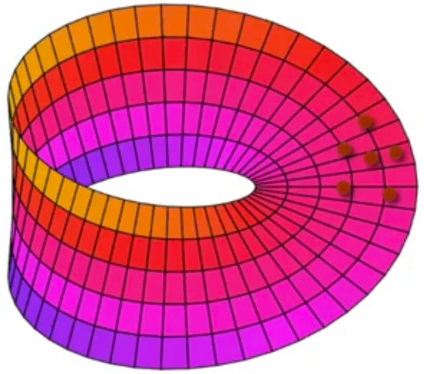




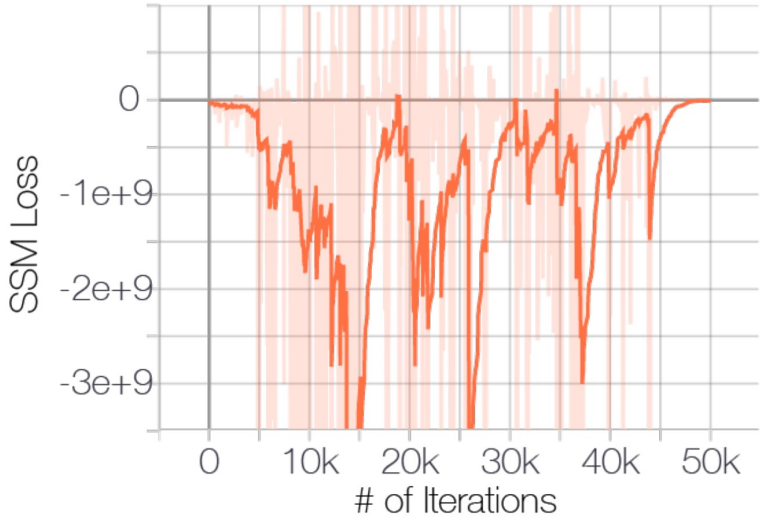
"*Generative Modeling by Estimating Gradients of the Data Distribution*", Song et al.

Pitfall (I)

- Data lies on a lower dimensional manifold
  - Data score is undefined in the ambient dimension.



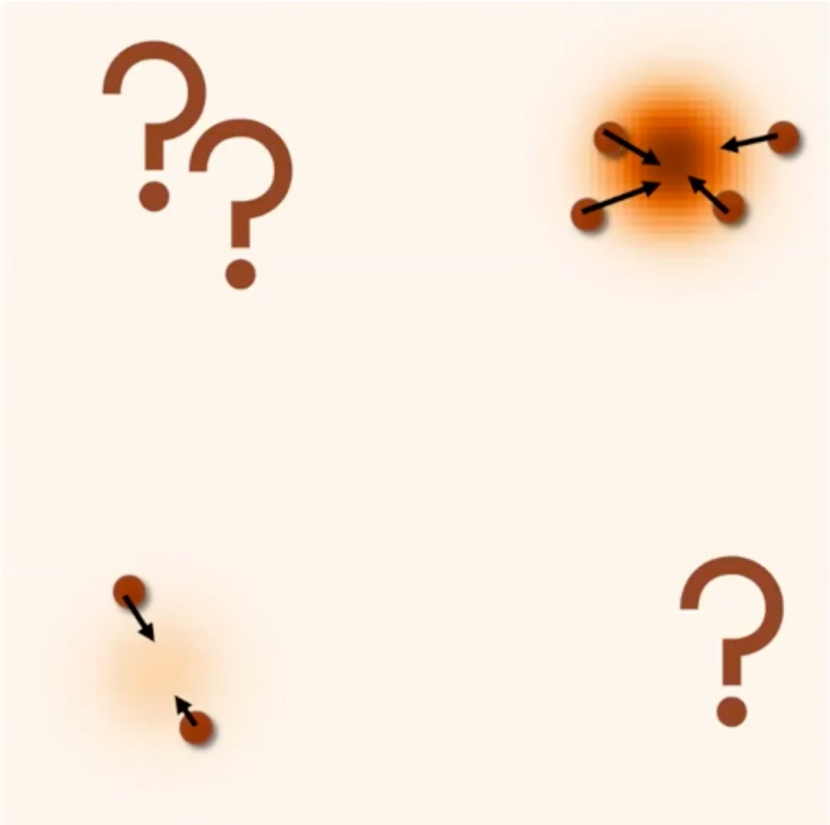
- Training loss of a ResNet model on CIFAR-10:



"*Generative Modeling by Estimating Gradients of the Data Distribution*", Song et al.

Pitfall (II)

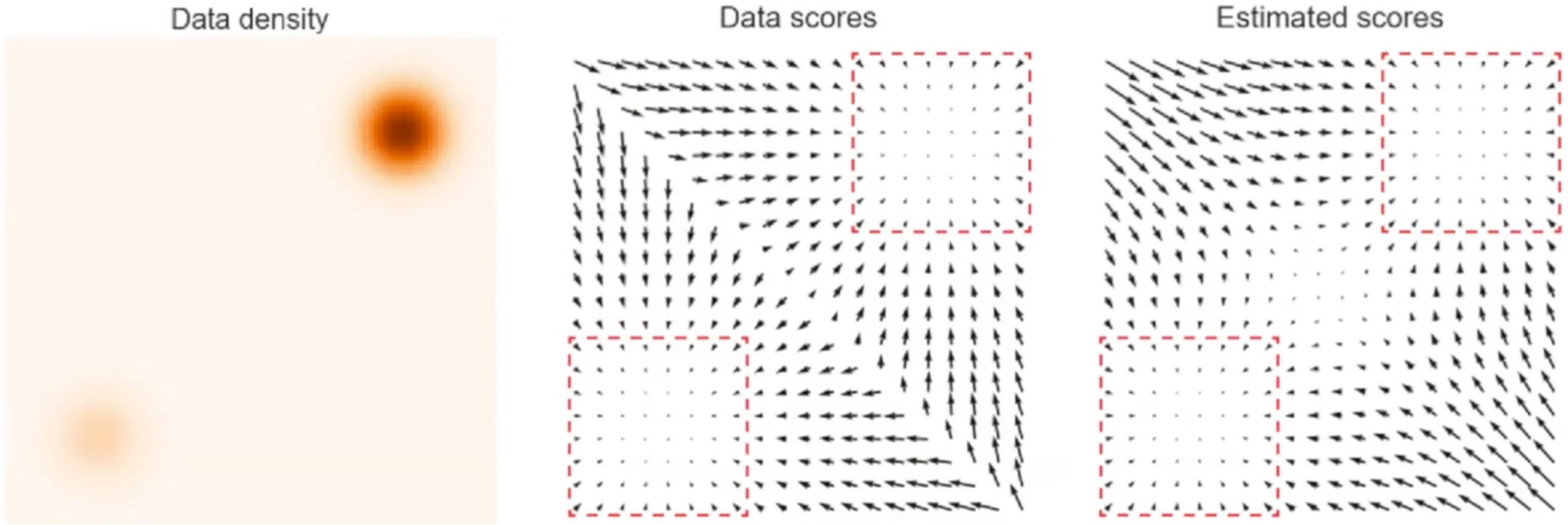
- Low-data density regions
  - **Data points tend to cluster around the modes of the distribution** (Smith, 2019)
  - The score estimation is going to be very inaccurate when you are far away from the high density regions of the data.



"*Generative Modeling by Estimating Gradients of the Data Distribution*", Song et al.

Pitfall (II)

- Low-data density regions



"*Generative Modeling by Estimating Gradients of the Data Distribution*", Song et al.

Pitfall (III)

- Slow mixing of Langevin Dynamics between data modes:

$$p_{data}(x) = \pi p_1(x) + (1-\pi)p_2(x)$$

$$\nabla_x \log p_{data}(x) = \nabla_x \log p_1(x) + \nabla_x \log p_2(x)$$

$\nabla_x \ln p(x) = \nabla_x \ln \pi + \nabla_x \ln p_1 + \nabla_x \ln(1-\pi) + \nabla_x \ln p_2$

0 0

"*Generative Modeling by Estimating Gradients of the Data Distribution*", Song et al.

Pitfall (III)

- Data close to the p_1:

$$\nabla_x \log p(x) \approx \nabla_x \log p_1(x)$$

- Data close to the p_2:

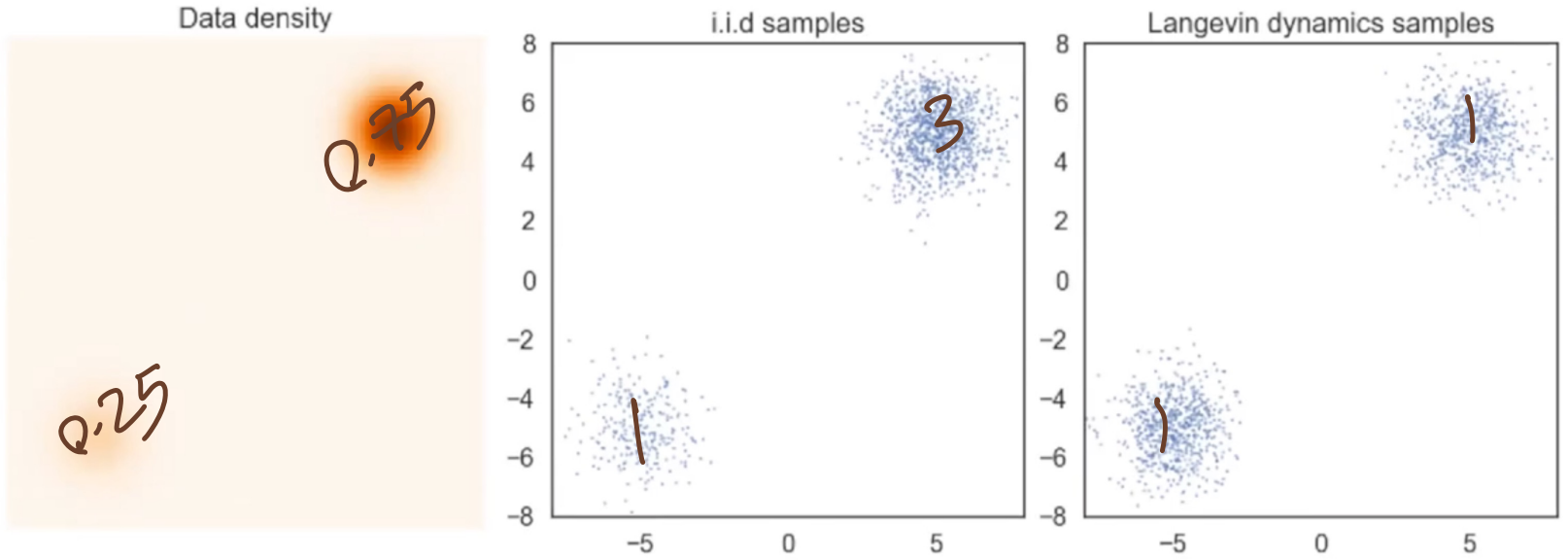
$$\nabla_x \log p(x) \approx \nabla_x \log p_2(x)$$

- The value of π has no effect on the gradient value.
  - The π coefficient should not influence the number of samples generated by the sampling process.

"*Generative Modeling by Estimating Gradients of the Data Distribution*", Song et al.

Pitfall (III)

- The value of π has no effect on the gradient value.
  - The π coefficient should not influence the number of samples generated by the sampling process.



"*Generative Modeling by Estimating Gradients of the Data Distribution*", Song et al.

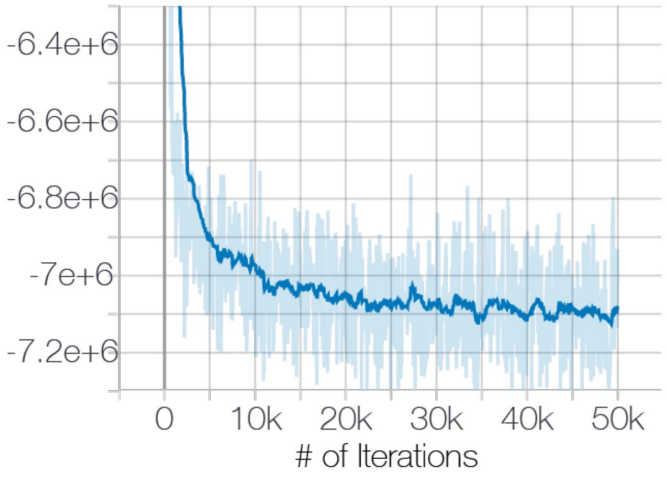
Solution?

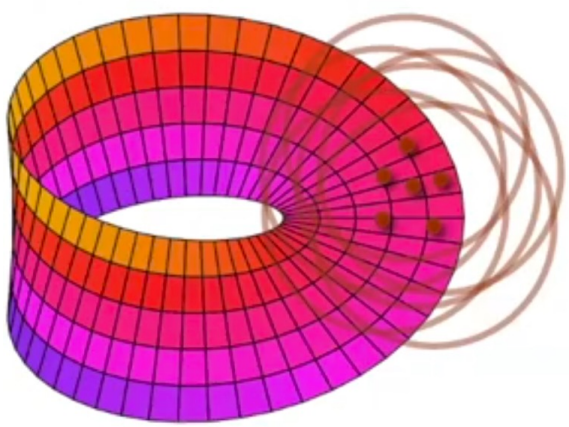
- What to do with these pitfalls?!
  - Solution: Gaussian Perturbation
    - Applying some Gaussian noise to the input data

"*Generative Modeling by Estimating Gradients of the Data Distribution*", Song et al.

Solution: Gaussian Perturbation

- Pitfall (I): Data lies on a lower dimensional manifold
  - Distribution becomes supported over the whole space.
    - Data score is well-defined in all dimensions.

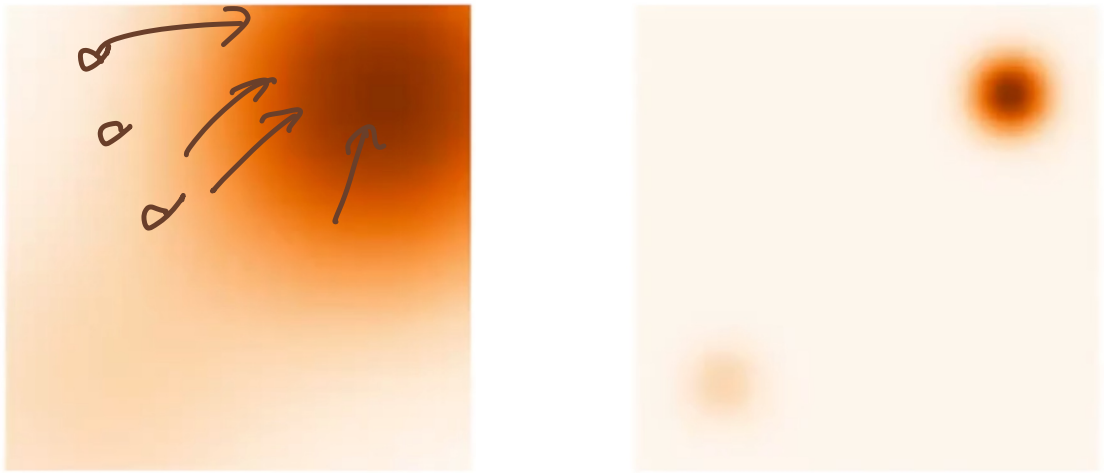




"*Generative Modeling by Estimating Gradients of the Data Distribution*", Song et al.

Solution: Gaussian Perturbation

- Pitfall (II): Low data-density regions
  - Signaling about what's the right thing to do even we are far away from the high density regions.

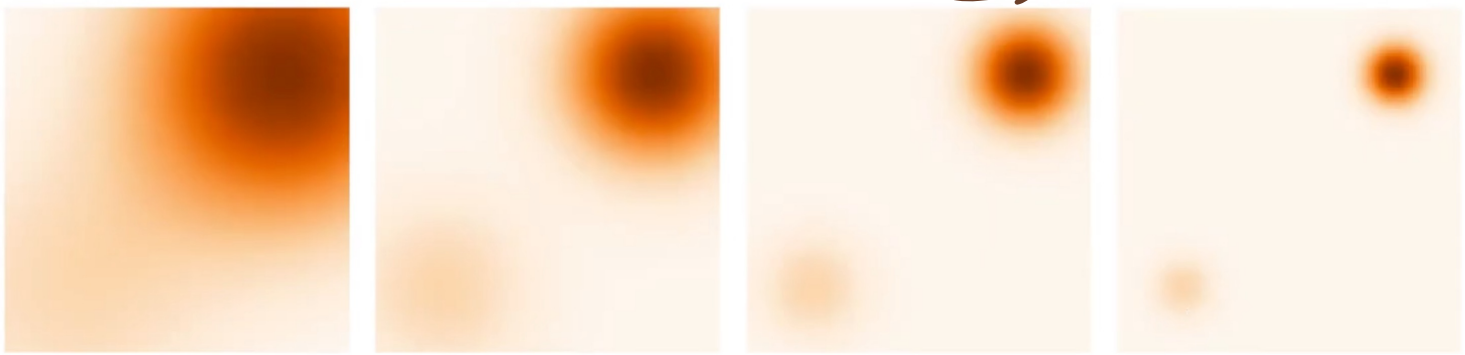


"*Generative Modeling by Estimating Gradients of the Data Distribution*", Song et al.

Solution: Gaussian Perturbation

- Pitfall (II): Low data-density regions
  - But, If the amount of added noise is too high, the structure of the distribution will be corrupted.
  - What to do?!!!
    - Consider multiple noise level

$$\sigma_1 > \sigma_2 > \cdots > \sigma_{L-1} > \sigma_L$$



"*Generative Modeling by Estimating Gradients of the Data Distribution*", Song et al.

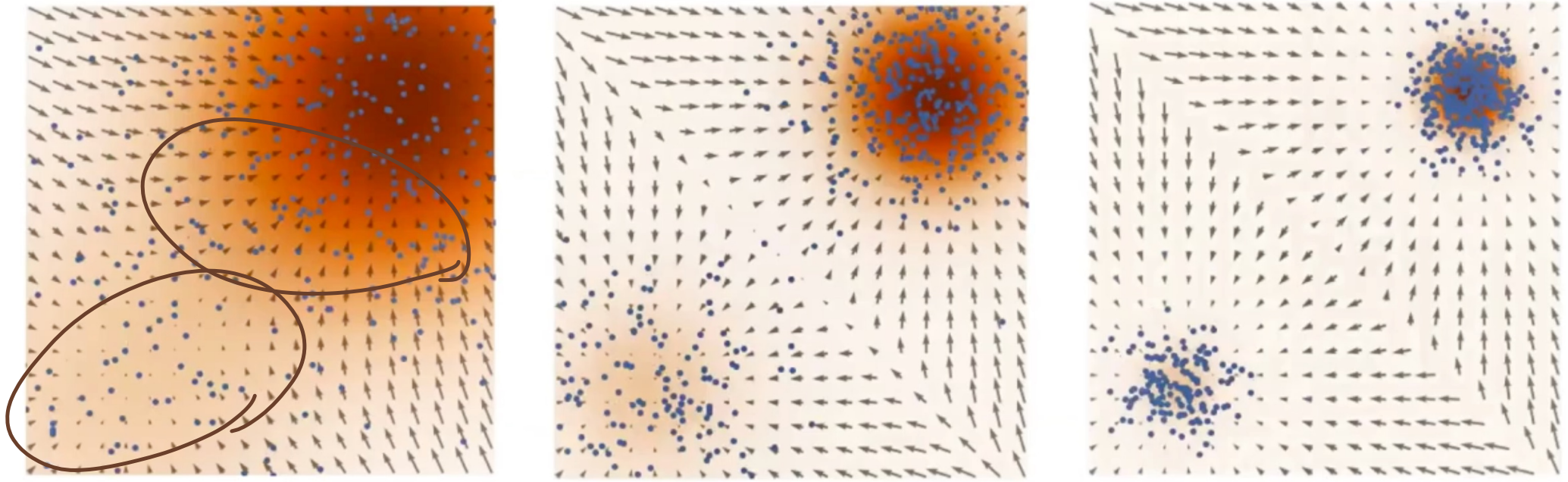
Solution: Gaussian Perturbation

- Pitfall (III): Slow mixing of Langevin Dynamics between data modes
  - The dominance effect of larger modes is considered

"*Generative Modeling by Estimating Gradients of the Data Distribution*", Song et al.

Annealed Langevin Dynamic Sampling

- Sample use σ_1, σ_2, …, and σ_L sequentially with Langevin dynamics.

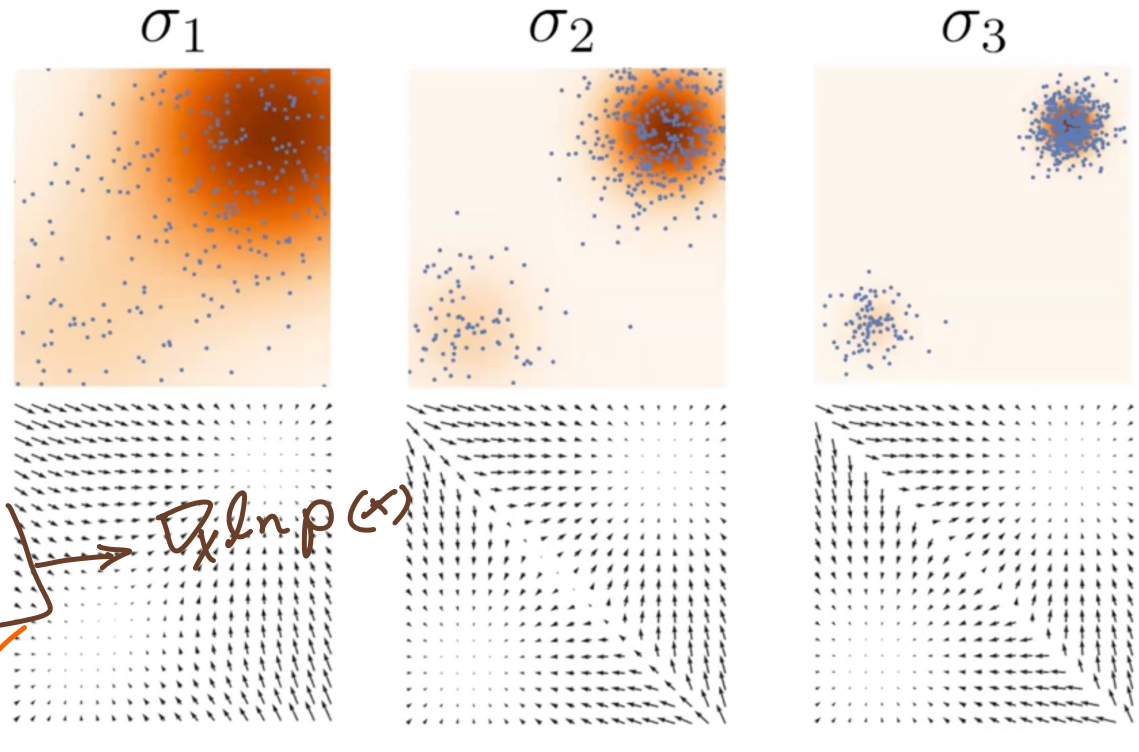


"*Generative Modeling by Estimating Gradients of the Data Distribution*", Song et al.

Annealed Langevin Dynamic Sampling

Pitfalls & Solution

- The score for any noise level should be estimated
  - Reason
    - Different underlying vector field
  - Drawback
    - Expensive
- How is it possible with only one neural network?
  - Just use an extra input which is sigma



"*Generative Modeling by Estimating Gradients of the Data Distribution*", Song et al.

Annealed Langevin Dynamic Sampling

$X = U + \varepsilon$
$\mathbb{E}[U|X] = X + \nabla_X \ln p$ Tweedie Formula

Algorithm 2- ~~Annealed Langevin~~ Dynamic Sampling with Score Function

Input: $s_\theta(\cdot,\cdot), \{\sigma_i\}_{i=1}^{L}, x, T$

1: $\tilde{x}_0 \sim \pi(x)$
2: **for** $i \leftarrow 1$ **to** L **do** → 10
3: $\alpha_i \leftarrow \epsilon \cdot \frac{\sigma_i^2}{\sigma_L^2}$ → 2e-4, 100
4: **for** $t \leftarrow 1$ **to** T **do** → O(TL), O(N)
5: $z_t \sim \mathcal{N}(0, I)$
6: $\tilde{x}_t := \tilde{x}_{t-1} + \frac{\alpha_i}{2} s_\theta(\tilde{x}_{t-1}, \sigma_i) + \sqrt{\alpha_i} z_t$ (Signal, noise)
7: **end for**
8: $\tilde{x}_0 \leftarrow \tilde{x}_T$
9: **end for**
10: **return** $\tilde{x}_T$

$\sigma_1 : x_0 \rightsquigarrow x_T$
$\sigma_2 : x_0 \rightsquigarrow x_T$
$\sigma_L : x_0 \rightsquigarrow x_T$

Sample ✓ → Return

Pitfalls & Solution

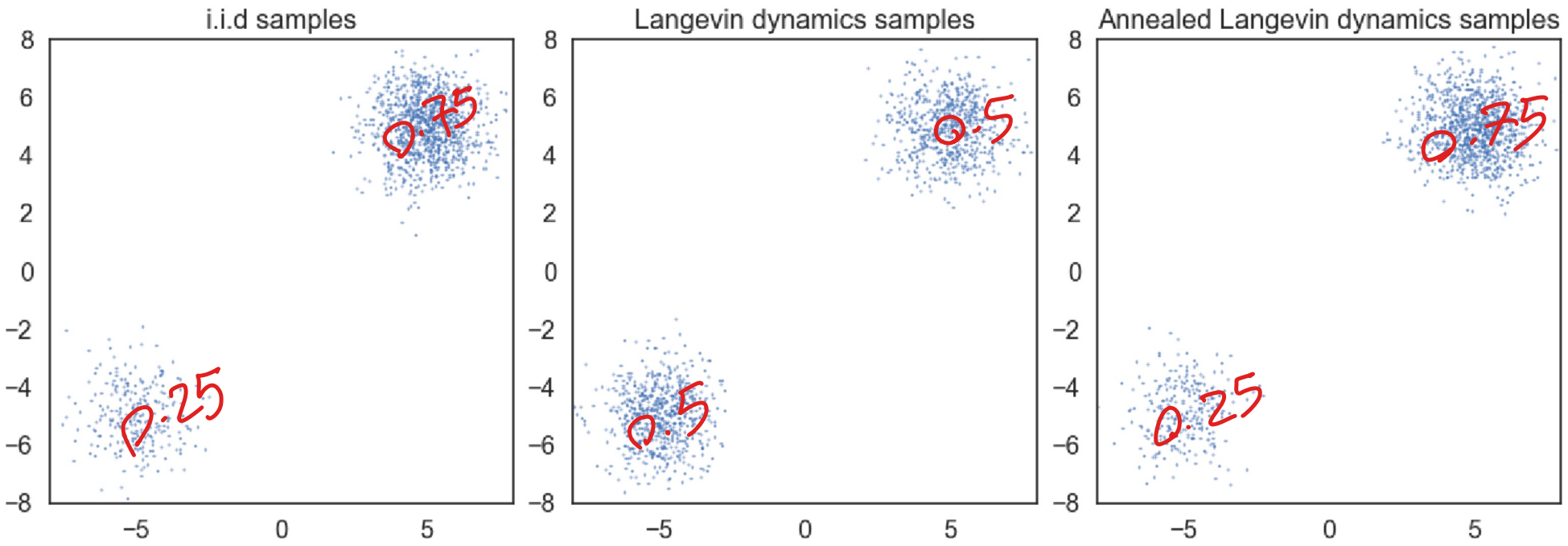
$\text{SNR}: \mathbb{E}\left[\left\| \frac{\alpha_i}{2} s_\theta \right\|^2\right] = \mathbb{E}\left[\frac{1}{4}\alpha_i \|s_\theta\|^2\right] \propto \frac{1}{4}\mathbb{E}\left[\sigma_i^2 \|s_\theta\|^2\right] \propto \frac{1}{4}$ (Constant)

$\because \alpha_i = \varepsilon \cdot \frac{\sigma_i^2}{\sigma_L^2}$

Imprically: $s_\theta(x, \sigma_i) \propto \frac{1}{\sigma_i} \Rightarrow \alpha_i \propto \sigma_i^2$

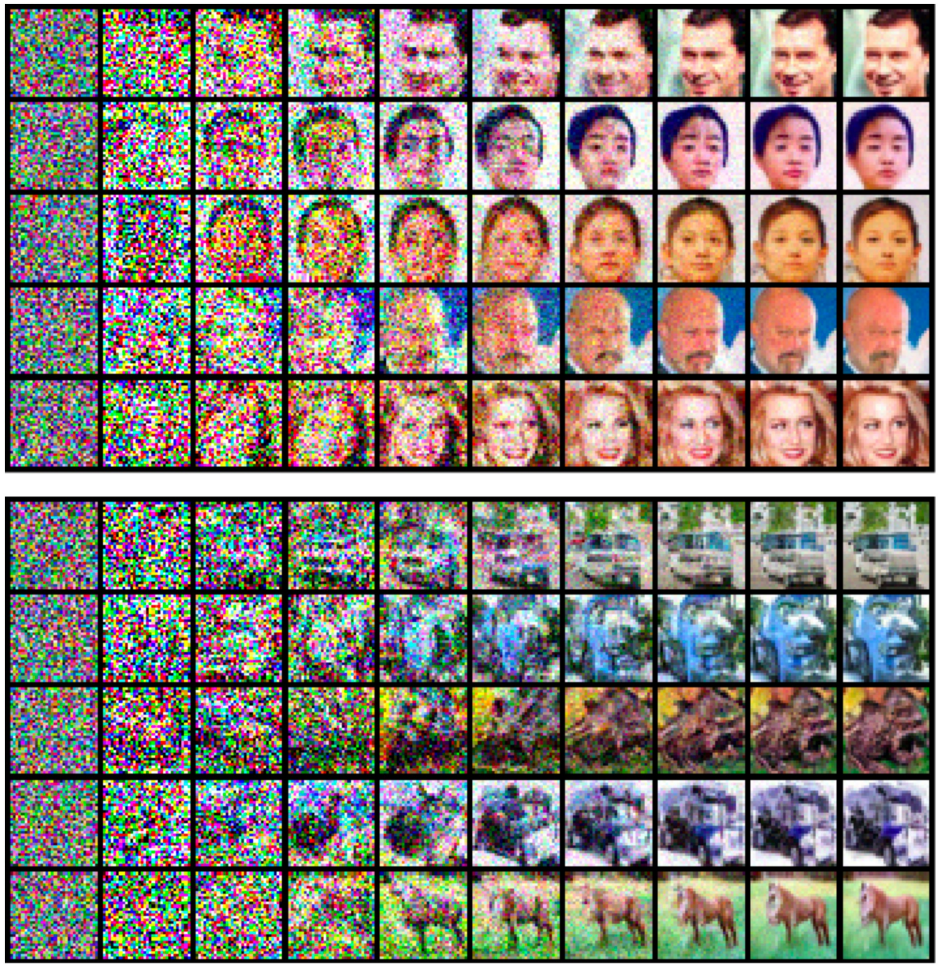
Experiments

Experiments



” Generative Modeling by Estimating Gradients of the Data Distribution”, Y. Song, S. Ermon

Experiments



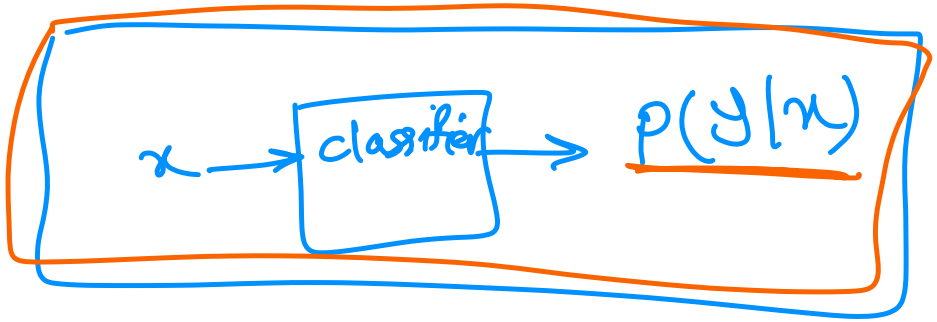
"*Generative Modeling by Estimating Gradients of the Data Distribution*", Song et al.

Experiments

Model	Inception	FID
CIFAR-10 Unconditional		
PixelCNN [59]	4.60	65.93
PixelIQN [42]	5.29	49.46
EBM [12]	6.02	40.58
WGAN-GP [18]	7.86 ± .07	36.4
MoLM [45]	7.90 ± .10	**18.9**
SNGAN [36]	8.22 ± .05	21.7
ProgressiveGAN [25]	8.80 ± .05	-
NCSN (Ours)	**8.87** ± .12	25.32
CIFAR-10 Conditional		
EBM [12]	8.30	37.9
SNGAN [36]	8.60 ± .08	25.5
BigGAN [6]	**9.22**	**14.73**

"*Generative Modeling by Estimating Gradients of the Data Distribution*", Song et al.

$x \sim p(x)$ ✓



✓

y: label

✓ $p(x|y)$

$$p(x|y) = \frac{p(y|x)\ p(x)}{p(y)}$$

$$\nabla_x \log p(x|y) = \nabla_x \left(\log p(y|x) + \log p(x) - \log p(y) \right)$$

$$= \nabla_x \log p(y|x) + \nabla_x \log p(x)$$

✓

THANKS FOR YOUR ATTENTION!

Appendices

Local Consistency

(Score Matching ⇔ Maximum Likelihood)

- **Theorem:** Suppose that the probability density function of x satisfies $p_{data}(x) = p_\theta(x)$ for some θ^* and also that if $\theta \neq \theta^*$ then $p_\theta(x) \neq p_{data}(x)$. Suppose also that $p_\theta(x) > 0$. Then:

$$J(\theta) = 0 \iff \theta = \theta^*$$

- The consistency implies asymptotic unbiasedness.

Appendix A

Local Consistency (Proof)

Appendix A

- **Is implied by:** We can see that $\theta = \theta^* \Longrightarrow J(\theta) = 0$ by substituting $p_{data}(x) = p_{\theta^*}(x)$ into $J(\theta^*)$:

$$J(\theta) = \frac{1}{2}\mathbb{E}_{p_{data}(x)}[\|\nabla_X \ln p_{data}(x) - s_\theta(x)\|_2^2]$$

$$= \frac{1}{2}\int p_{data}(x) \cdot \|\nabla_X \ln p_{data}(x) - s_\theta(x)\|_2^2 \, dx$$

$$= \frac{1}{2}\int p_{\theta^*}(x) \cdot \left\| \underbrace{\nabla_X \ln p_{\theta^*}(x)}_{s_\theta(x)} - s_\theta(x) \right\|_2^2 dx$$

$$= \frac{1}{2}\int p_{\theta^*}(x) \cdot \|s_\theta(x) - s_\theta(x)\|_2^2 \, dx = 0$$

Local Consistency (Proof)

- **Implies:** Going the other direction, we can show that $J(\theta) = 0 \Longrightarrow \theta = \theta^*$ by considering $p_{data}(x) = p_\theta(x)$ for some θ^*:

$$J(\theta) = \frac{1}{2}\mathbb{E}_{p_{data}(x)}[\|\nabla_X \ln p_{data}(x) - s_\theta(x)\|_2^2]$$

$$= \frac{1}{2}\int p_{data}(x) \cdot \|\nabla_X \ln p_{data}(x) - s_\theta(x)\|_2^2 \, dx$$

$$= \frac{1}{2}\int \underbrace{p_{\theta^*}(x)}_{\to 0} \cdot \|\nabla_X \ln p_{\theta^*}(x) - s_\theta(x)\|_2^2 \, dx = 0$$

Appendix A

Local Consistency (Proof)

So,

$$\|\nabla_X \ln p_{\theta^*}(x) - s_\theta(x)\|_2^2 = 0 \Longrightarrow \nabla_X \ln p_{\theta^*}(x) = s_\theta(x)$$
$$\Longrightarrow \nabla_X \ln p_{\theta^*}(x) = \nabla_X \ln p_\theta(x)$$

That means,

$$\ln p_\theta(x) = \ln p_{\theta^*}(x) + const \Longrightarrow p_\theta(x) \propto p_{\theta^*}(x)$$

and since $p_{\theta^*}(x)$ is a normalised probability distribution, we arrive at $p_\theta(x) = p_{\theta^*}(x)$. Now since the $p_{\theta^*}(x)$ is unique for the particular θ^*, we have that $\theta = \theta^*$.

Appendix A

Simplified Objective Function (Theorem for 1D Data)

- **Theorem:** Given a score function $s_\theta(x)$ which is differentiable w.r.t. X satisfies some weak regularity conditions ($\lim_{X \to |\infty|} p_{data}(X) = 0$). Then the score-matching function J can be written as:

$$J(\theta) = \mathbb{E}_{p_{data}(x)} \left[\frac{1}{2} \| s_\theta(X) \|_2^2 + tr\left(\nabla_X s_\theta(X) \right) \right]$$

Appendix B

Simplified Objective Function (Proof for 1D Data)

Appendix B

- **Proof:**

$$J(\theta) = \frac{1}{2}\mathbb{E}_{p_{data}(x)}\left[\left(\nabla_X \ln p_{data}(x) - s_\theta(x)\right)^2\right]$$

$$= \frac{1}{2}\int p_{data}(x) \cdot \left(\nabla_X \ln p_{data}(x) - s_\theta(x)\right)^2 dx$$

$$= \frac{1}{2}\int p_{data}(x) \cdot \left(\nabla_X \ln p_{data}(x)\right)^2 dx + \frac{1}{2}\int p_{data}(x) \cdot s_\theta(x)^2 \, dx$$

$$\underbrace{\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad}_{const\ w.r.t\ \theta}$$

$$- \int p_{data}(x) \cdot \nabla_X \ln p_{data}(x) \cdot s_\theta(x) \, dx$$

Simplified Objective Function (Proof for 1D Data)

- **Proof:**

$$\int p_{data}(x) \cdot s_\theta(x)^2 \, dx = \mathbb{E}_{p_{data}(x)}[(s_\theta(x))^2]$$

Appendix B

Simplified Objective Function (Proof for 1D Data)

- **Proof:**

$$-\int p_{data}(x)\cdot\nabla_X \ln p_{data}(x)\cdot s_\theta(x)\,dx$$

$$=-\int p_{data}(x)\cdot\frac{\nabla_X p_{data}(x)}{p_{data}(x)}\cdot s_\theta(x)\,dx=-\int \nabla_X p_{data}(x)\cdot s_\theta(x)\,dx$$

$$=\underbrace{-p_{data}(x)\cdot s_\theta(x)\Big|_{-\infty}^{+\infty}}_{0}+\int p_{data}(x)\cdot\nabla_X s_\theta(x)\,dx=\mathbb{E}_{p_{data}(x)}[\nabla_X s_\theta(x)]$$

Appendix B

Simplified Objective Function (Proof for 1D Data)

- **Proof:**

$$J(\theta) = \mathbb{E}_{p_{data}(x)}\left[\frac{1}{2}\left(s_\theta(x)\right)^2 + \nabla_x s_\theta(x)\right] + const$$

$$\underbrace{\qquad\qquad\qquad\qquad\qquad\qquad}_{J'(\theta)}$$

$$\Longrightarrow \boxed{J'(\theta) = \mathbb{E}_{p_{data}(x)}\left[\frac{1}{2}\left(s_\theta(x)\right)^2 + \nabla_x s_\theta(x)\right]}$$

Appendix B

Score Matching vs. Maximum Likelihood

- **Theorem:** Let $y = x + \sqrt{t}w$, for $t \geq 0$ and w a zero-mean white Gaussian vector. Denote $\tilde{p}_t(y)$ and $\tilde{q}_t(y)$ as the densities of y whe x has distribution $p(x)$ and $q(x)$, respectively. Assume that $\tilde{p}_t(y)$ and $\tilde{q}_t(y)$ are smooth and fast decaying, such that their logarithms has growth at most polynomial at infinity. We have:

$$\frac{d}{dt} D_{KL}\left(\tilde{p}_t(y) \parallel \tilde{q}_t(y)\right) = -\frac{1}{2} D_F\left(\tilde{p}_t(y) \parallel \tilde{q}_t(y)\right)$$

As $\tilde{p}_0(y) = p(x)$ and $\tilde{q}_0(y) = q(x)$, we further have:

$$\frac{d}{dt} D_{KL}\left(\tilde{p}_t(y) \parallel \tilde{q}_t(y)\right)\Big|_{t=0} = -\frac{1}{2} D_F\left(p(x) \parallel q(x)\right)$$

Appendix C

Interpretation and Generalization of Score Matching, Siwe Lyu.

Score Matching vs. Maximum Likelihood

- This theorem reveals some intriguing aspects of the relation between score matching and maximum likelihood:
  1. When searching for parameters in q_θ to match p within the scale space:
     - The focus can be directed towards prominent large-scale structures that endure the smoothing process.
     - Simultaneously, any artificial structures arising from sampling effects in the training data are disregarded.

$$\tilde{p}_t(y) = \int_x \frac{1}{(2\pi t)^{\frac{d}{2}}} \exp\left(-\frac{\|y-x\|}{2t}\right) p(x)dt$$

  2. The KL divergence between two densities always decreases as the scale factor increases.

Appendix C

Interpretation and Generalization of Score Matching, Siwe Lyu.

Score Matching vs. Maximum Likelihood

- This theorem reveals some intriguing aspects of the relation between score matching and maximum likelihood:
  3. Score matching seeks stability, aiming for an optimal parameter θ that minimizes changes in the KL divergence between two models when a small amount of noise is introduced in the training data. In contrast, maximum likelihood pursues extremity of the KL divergence.
     - Maximum likelihood estimation can be influenced by noisy training data, leading to the possibility of generating numerous incorrect extreme values.
     - Score matching could exhibit greater robustness to minor perturbations in training data.

Appendix C

Interpretation and Generalization of Score Matching, Siwe Lyu.

Simplified Objective Function (An Example)

- Consider estimation of parameters of the multivariate Gaussian density:

$$p(x;\mu,\Sigma) = \frac{1}{Z(\mu,\Sigma)} \exp\left(-\frac{1}{2}(x-\mu)^{\mathrm{T}}\Sigma^{-1}(x-\mu)\right)$$

$$\frac{1}{Z(\mu,\Sigma)} = \frac{1}{\sqrt{2\pi}^{\frac{d}{2}}} |\Sigma|^{-\frac{1}{2}}$$

- The unnormalized kernel is:

$$q(x) = \exp\left(-\frac{1}{2}(x-\mu)^{\mathrm{T}}\Sigma(x-\mu)\right)$$

App. D

Simplified Objective Function (An Example)

- We have:

$$s_\theta(x) = \nabla_x \ln q(x) = -\Sigma(x - \mu)$$

- And,

$$tr\left(\nabla_x s_\theta(x)\right) = -\sigma_{ii}$$

- The simplified objective is:

$$J(\theta) = \frac{1}{T}\sum_{t=1}^{T}\left[\sum_i -\sigma_{ii} + \frac{1}{2}(x^t - \mu)^T \Sigma\Sigma(x^t - \mu)\right]$$

Simplified Obj.

App. D

Simplified Objective Function (An Example)

- $\nabla_\mu J(\theta)$:

$$\nabla_\mu J(\theta) = \Sigma\Sigma\mu - \Sigma\Sigma\frac{1}{T}\sum_{t=1}^{T} x^t$$

- Σ is symmetric positive definite:

$$\mu = \frac{1}{T}\sum_{t=1}^{T} x^t$$

App. D

Simplified Objective Function (An Example)

- $\nabla_{\Sigma} J(\theta)$:

$$\nabla_{\mu} J(\theta) = -\mathrm{I} + \Sigma \frac{1}{2T} \sum_{t=1}^{T} (x^t - \mu)^T (x^t - \mu) + \frac{1}{2T} \left[\sum_{t=1}^{T} (x^t - \mu)(x^t - \mu)^T \right] \Sigma$$

- So we will have:

$$\Sigma = \frac{1}{T} \sum_{t=1}^{T} (x^t - \mu)(x^t - \mu)^T$$

App. D