Denoising Score Matching

Filippo Garagnani

Mentor: Prof. Iacopo Masi
Elicsir Foundation – 3° Orthogonal Weekend

March 22, 2026

A Quote

Can one say: ”Where there is no doubt, there is no knowledge either”?

— *Ludwig Wittgenstein*, On Certainty

Introduction

Discriminative Modeling

Task: Determining output y from sample x.

Discriminative Modeling

Task: Determining output y from sample x.

We want a function

$$f(x) \simeq p(y \mid x)$$

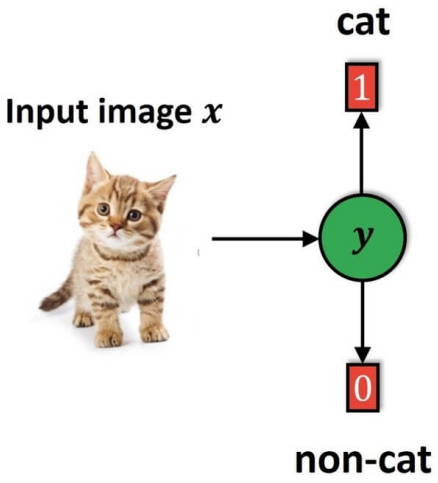
Discriminative Modeling

Task: Determining output y from sample x.

We want a function

$$f(x) \simeq p(y \mid x)$$

We do **not** learn anything about how data is structured in space.



Generative Modeling

Task: Model how data is distributed.

Generative Modeling

Task: Model how data is distributed.

We learn

$$p(x)$$

Generative Modeling

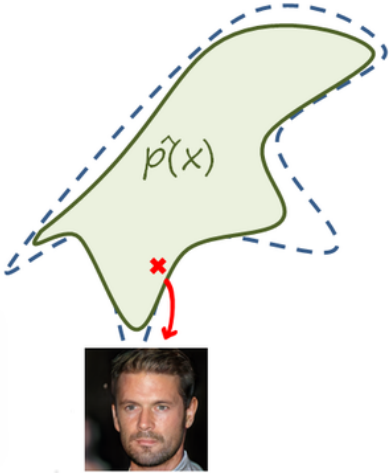
Task: Model how data is distributed.

We learn

$$p(x)$$

We could then generate new samples: $x \sim p(x)$

Evaluate the plausibility of a sample: $p(\mathbf{x})$



Learning $p(x)$

Problem Setting – Generative Modeling

We have a specific finite collection of samples:

- **Dataset**: $\mathcal{D} = \{x_i\}_{i=1}^N$
- **Samples**: $x_i \in \mathbb{R}^d$

Problem Setting – Generative Modeling

We have a specific finite collection of samples:

- **Dataset**: $\mathcal{D} = \{x_i\}_{i=1}^N$
- **Samples**: $x_i \in \mathbb{R}^d$

Assume data samples have been drawn from an unknown underlying distribution:

$$x_i \sim p_{\text{data}}(x)$$

Problem Setting – Generative Modeling

Problem. p_{data} **cannot be computed.**

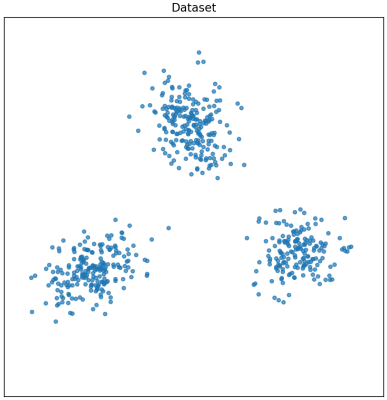
Problem Setting – Generative Modeling

Problem. p_{data} **cannot be computed**.
However,

Problem Setting – Generative Modeling

Problem. p_{data} **cannot be computed**.
However, we can parametrize a model distribution $p_\phi(x)$ such that:

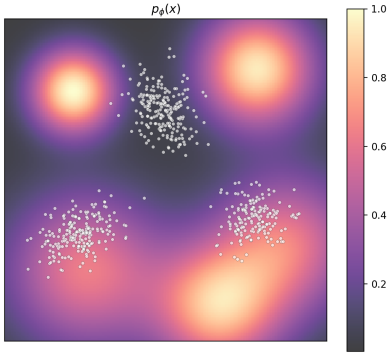
$$p_{\text{data}}(x) \simeq p_\phi(x)$$



Problem Setting – Generative Modeling

Problem. p_{data} **cannot be computed**.
However, we can parametrize a model distribution $p_\phi(x)$ such that:

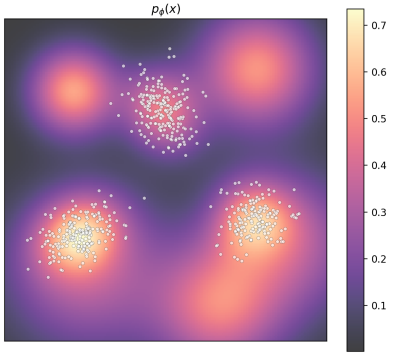
$$p_{\text{data}}(x) \simeq p_\phi(x)$$



Problem Setting – Generative Modeling

Problem. p_{data} **cannot be computed**.
However, we can parametrize a model distribution $p_\phi(x)$ such that:

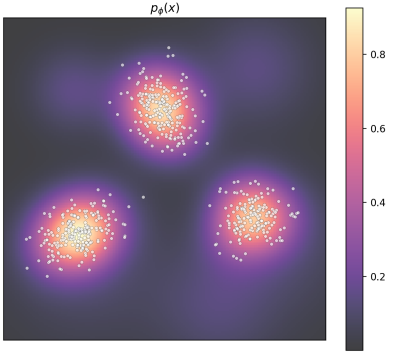
$$p_{\text{data}}(x) \simeq p_\phi(x)$$



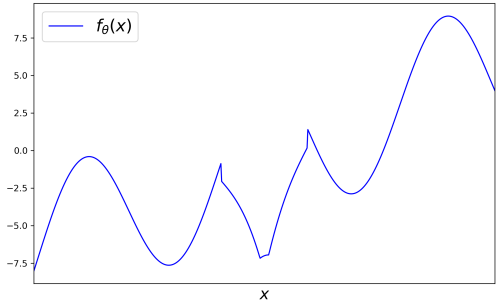
Problem Setting – Generative Modeling

Problem. p_{data} **cannot be computed**.
However, we can parametrize a model distribution $p_\phi(x)$ such that:

$$p_{\text{data}}(x) \simeq p_\phi(x)$$



Given a generic function f_ϕ (parametrized by ϕ), for it to be a probability density function it must satisfy:

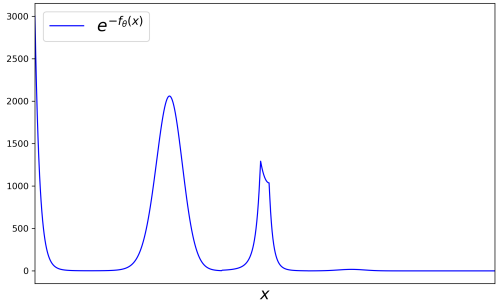


[1] [Hinton & Sejnowski, 1985]
[2] [LeCun, Chopra & Hadsell, 2006]

Energy-Based Models

Given a generic function f_ϕ (parametrized by ϕ), for it to be a probability density function it must satisfy:

1. **Non-Negativity**. $g_\phi(x) \equiv e^{-f_\phi(x)}$.

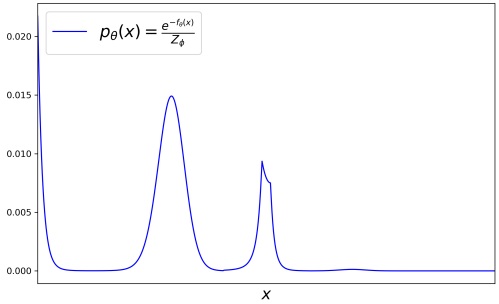


[1] [Hinton & Sejnowski, 1985]
[2] [LeCun, Chopra & Hadsell, 2006]

Energy-Based Models

Given a generic function f_ϕ (parametrized by ϕ), for it to be a probability density function it must satisfy:

1. **Non-Negativity**. $g_\phi(x) \equiv e^{-f_\phi(x)}$.
2. **Normalization to 1**. $p_\phi(x) \equiv \frac{1}{Z_\phi} g_\phi(x)$ $(Z_\phi \equiv \int_{\mathbf{x}} g_\phi(\mathbf{x}))$.



[1] [Hinton & Sejnowski, 1985]
[2] [LeCun, Chopra & Hadsell, 2006]

Problem. Usually, Z_ϕ is **intractable**.

Problem. Usually, Z_ϕ is **intractable**.
However,

Problem. Usually, Z_ϕ is **intractable**.
However, we can bypass using it by:

1. Transforming it into an **additive term**: $\log p_\phi(x) = \log g_\phi(x) - \log Z_\phi$.

Problem. Usually, Z_ϕ is **intractable**.
However, we can bypass using it by:

1. Transforming it into an **additive term**: $\log p_\phi(x) = \log g_\phi(x) - \log Z_\phi$.

2. **Derivating** w.r.t. to x:

$$\begin{aligned}\nabla_x \log p_\phi(x) &= \nabla_x \log g_\phi(x) - \cancelto{0}{\nabla_x \log Z_\phi} \\ &= -\nabla_x f_\phi(x)\end{aligned}$$

Score Matching

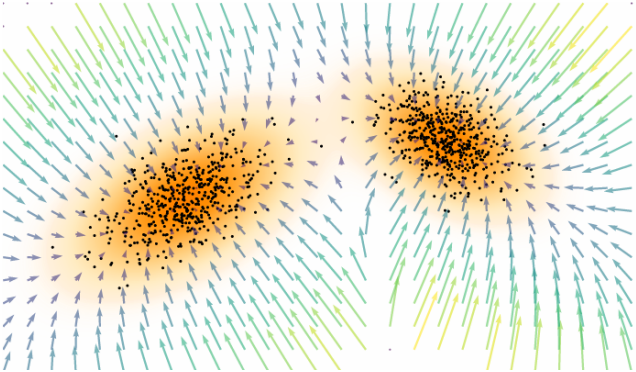
Score Function

Definition

Given a probability density function $p(x)$, its **score function** $s(x)$ is:

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

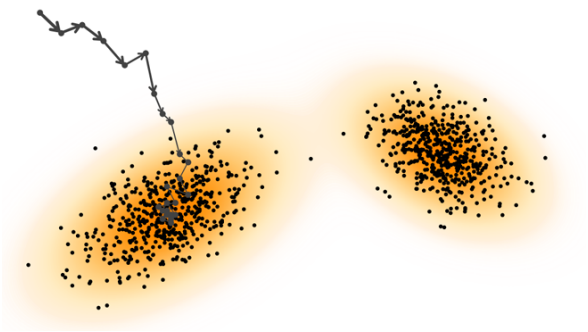
Score Function



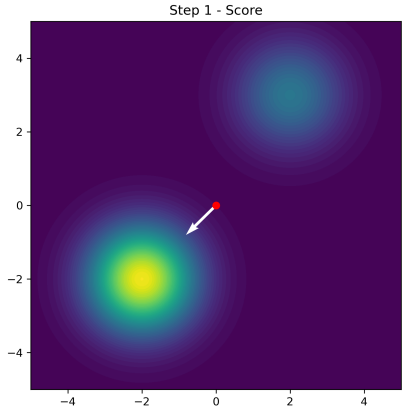
Generate with Score

Langevin Dynamics.

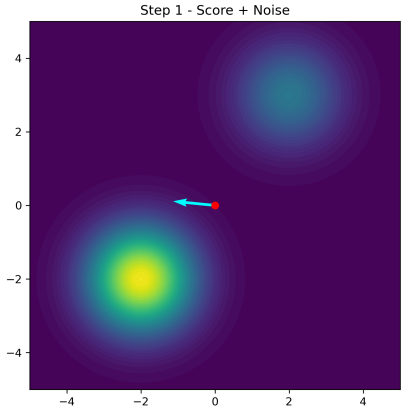
1: Initialize $x_0,\ \eta$
2: **for** $t \leftarrow 1$ to T **do**
3: $\quad \mathbf{z}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
4: $\quad x_t \leftarrow x_{t-1} + \eta\, \mathbf{s}(x_{t-1}) + \sqrt{2\eta}\, \mathbf{z}_t$
5: **end for**
6: **return** x_T



Langevin Dynamics

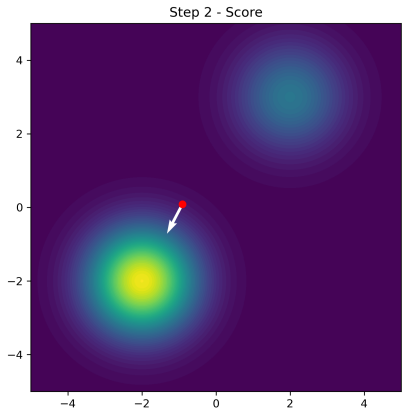


$\nabla_x \log p(x)$

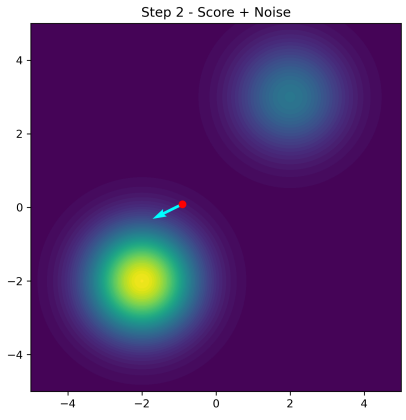


$\nabla_x \log p(x) + \mathbf{z}_t$

Langevin Dynamics

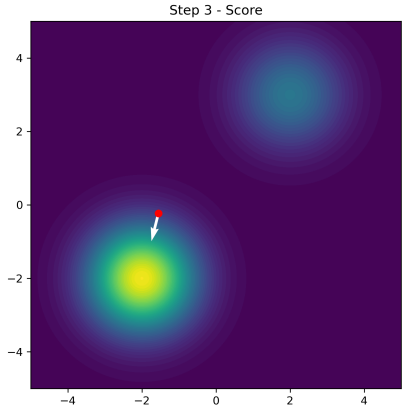


$\nabla_x \log p(x)$

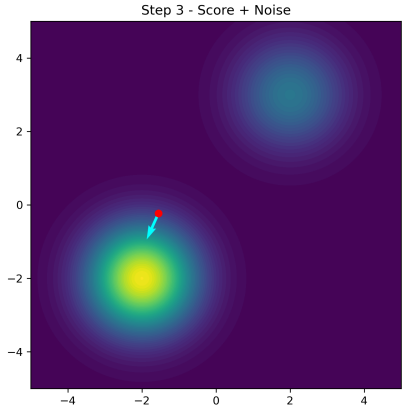


$\nabla_x \log p(x) + \mathbf{z}_t$

Langevin Dynamics

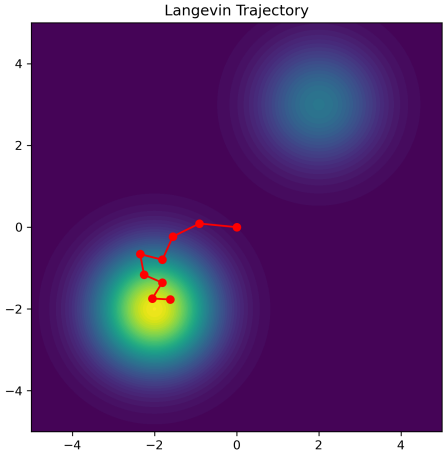


$$\nabla_x \log p(x)$$



$$\nabla_x \log p(x) + \mathbf{z}_t$$

Langevin Dynamics



Score Matching

To approximate $s_{\text{data}}(x)$, we can find:

$$\phi^\star = \min_\phi \mathbb{E}_{x\sim p_{\text{data}}} \left[\|s_\phi(x) - s_{\text{data}}(x)\|^2\right]$$

Score Matching

To approximate $s_{\text{data}}(x)$, we can find:

$$\phi^\star = \min_\phi \mathbb{E}_{x\sim p_{\text{data}}} \left[\|s_\phi(x) - s_{\text{data}}(x)\|^2\right]$$

And then employ $s_{\phi^\star}(x)$ to generate data.

Score Matching

Problem. To compute $s_{\text{data}}(x)$, we would need $p_{\text{data}}(x)$.

[3] [Hyvärinen, 2005]

Problem. To compute $s_{\text{data}}(x)$, we would need $p_{\text{data}}(x)$. However,

[3] [Hyvärinen, 2005]

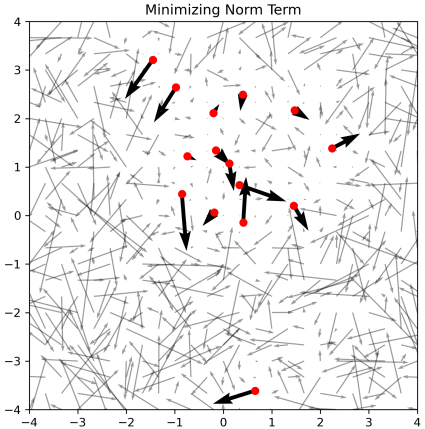
Problem. To compute $s_{\text{data}}(x)$, we would need $p_{\text{data}}(x)$.
However, it can be shown that $\phi^\star$ is:

$$\begin{aligned}\phi^\star &= \min_\phi \mathbb{E}_{x\sim p_{\text{data}}}\left[\|s_\phi(x) - s_{\text{data}}(x)\|^2\right] \\ &= \min_\phi \mathbb{E}_{x\sim p_{\text{data}}}\left[\,\text{Tr}(\nabla_x\, s_\phi(x)) + \|s_\phi(x)\|^2\,\right]\end{aligned}$$

[3] [Hyvärinen, 2005]

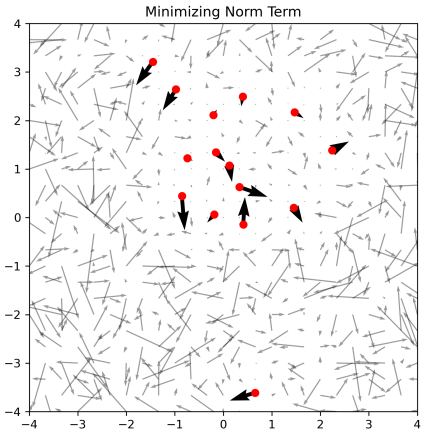
Score Matching – Norm Term

$$\min_\phi \mathbb{E}_{x\sim p_{\text{data}}} \left[\operatorname{Tr}(\nabla_x s_\phi(x)) + \|s_\phi(x)\|^2 \right]$$



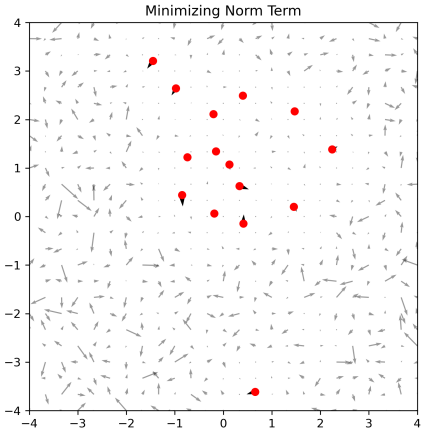
Score Matching – Norm Term

$$\min_\phi \mathbb{E}_{x\sim p_{\text{data}}} \left[\operatorname{Tr}(\nabla_x s_\phi(x)) + \|s_\phi(x)\|^2 \right]$$



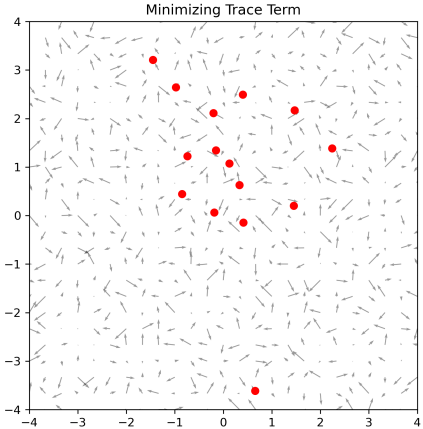
Score Matching – Norm Term

$$\min_\phi \mathbb{E}_{x\sim p_{\text{data}}} \left[\operatorname{Tr}(\nabla_x s_\phi(x)) + \|s_\phi(x)\|^2 \right]$$



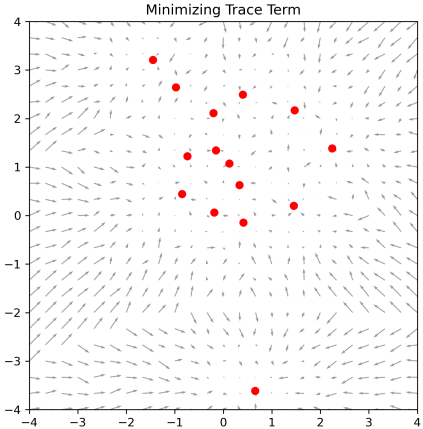
Score Matching – Trace Term

$$\min_\phi \mathbb{E}_{x\sim p_{\text{data}}} \left[\text{Tr}(\nabla_x s_\phi(x)) + \|s_\phi(x)\|^2 \right]$$



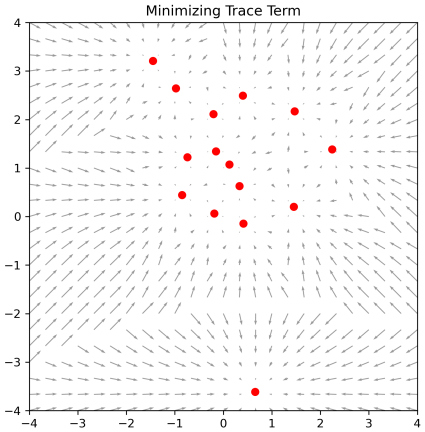
Score Matching – Trace Term

$$\min_\phi \mathbb{E}_{x\sim p_{\text{data}}} \left[\operatorname{Tr}(\nabla_x s_\phi(x)) + \|s_\phi(x)\|^2 \right]$$



Score Matching – Trace Term

$$\min_\phi \mathbb{E}_{x\sim p_{\text{data}}} \left[\operatorname{Tr}(\nabla_x s_\phi(x)) + \|s_\phi(x)\|^2 \right]$$



Score Matching – Key Problems

1. $\mathrm{Tr}(\nabla_x s_\phi(x))$ is intractable – we would need to compute $O(n^2)$ derivatives.

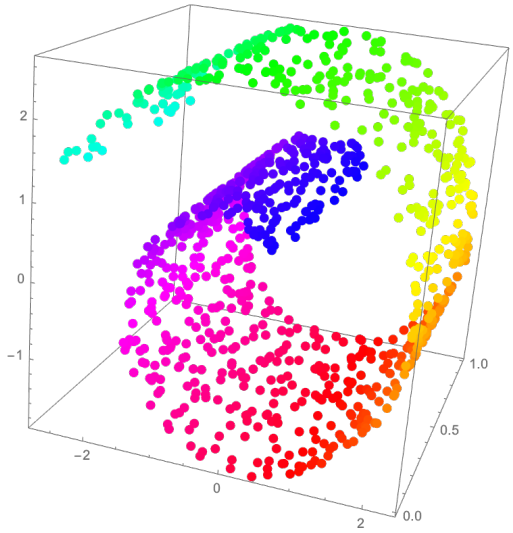
Score Matching – Key Problems

1. $\mathrm{Tr}(\nabla_x s_\phi(x))$ is intractable – we would need to compute $O(n^2)$ derivatives.
2. Score Matching usually gives accurate results in regions *very close* to the samples.

Score Matching – Key Problems

1. $\mathrm{Tr}(\nabla_x s_\phi(x))$ is intractable – we would need to compute $O(n^2)$ derivatives.
2. Score Matching usually gives accurate results in regions *very close* to the samples.
3. Manifold Hypothesis – no samples exist outside the manifold.

Manifold Hypothesis



Denoising Score Matching

Denoising Score Matching

Perturbing data points x randomly:

$$\hat{x} = x + \sigma\epsilon, \quad \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$$

[4] [Vincent, 2010]

Denoising Score Matching

Perturbing data points x randomly:

$$\hat{x} = x + \sigma\epsilon, \quad \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$$

Determines a **corruption process** $q_\sigma(\hat{x}|x) \sim \mathcal{N}(x, \sigma^2\mathbf{I})$:

$$q_\sigma(\hat{x}) = \int_x q_\sigma(\hat{x}|x)\, p_{\text{data}}(x)\, dx$$

[4] [Vincent, 2010]

Denoising Score Matching

Perturbing data points x randomly:

$$\hat{x} = x + \sigma\epsilon, \quad \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$$

Determines a **corruption process** $q_\sigma(\hat{x}|x) \sim \mathcal{N}(x, \sigma^2\mathbf{I})$:

$$q_\sigma(\hat{x}) = \int_x q_\sigma(\hat{x}|x)\, p_{\text{data}}(x)\, dx$$

It can be proven that:

$$\nabla_{\hat{x}} \log q_\sigma(\hat{x}) \simeq \nabla_x \log p_{\text{data}}(x)$$

[4] [Vincent, 2010]

Denoising Score Matching

Problem. Unfeasible:

$$q_\sigma(\hat{x}) = \int_x q_\sigma(\hat{x}|x)\, p_{\text{data}}(x)\; dx$$

Denoising Score Matching

Problem. Unfeasible:

$$q_\sigma(\hat{x}) = \int_x q_\sigma(\hat{x}|x)\, p_{\text{data}}(x)\; dx$$

Solution - approximation:

$$q_\sigma(\hat{x}) \approx \frac{1}{N} \sum_{i=1}^{N} q_\sigma(\hat{x}|x_i)$$

Denoising Score Matching

The objective is:

$$\phi^\star = \min_\phi \mathbb{E}_{x\sim p_{\text{data}},\, \hat{x}\sim\mathcal{N}(x,\,\sigma^2\mathbf{I})}\left[\left\| s_\phi(\hat{x}) - \nabla_{\hat{x}} \log q_\sigma(\hat{x}|x)\right\|^2\right]$$

The objective is:

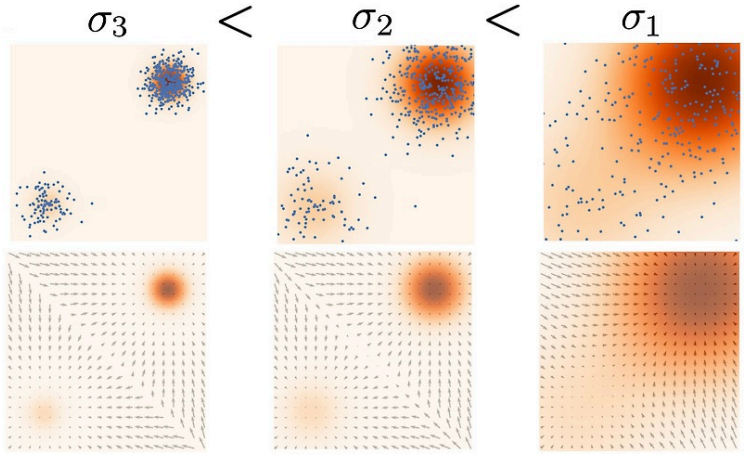
$$\phi^\star = \min_\phi \mathbb{E}_{x\sim p_{\text{data}},\, \hat{x}\sim\mathcal{N}(x,\,\sigma^2\mathbf{I})}\left[\left\| s_\phi(\hat{x}) - \nabla_{\hat{x}} \log q_\sigma(\hat{x}|x)\right\|^2\right]$$

$$= \min_\phi \mathbb{E}_{x\sim p_{\text{data}},\, \hat{x}\sim\mathcal{N}(x,\,\sigma^2\mathbf{I})}\left[\left\| s_\phi(\hat{x}) - \left(-\frac{\hat{x}-x}{\sigma^2}\right)\right\|^2\right]$$

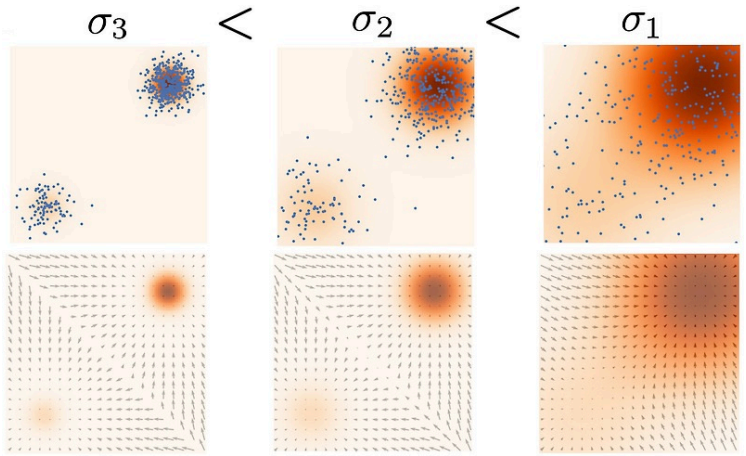
The objective is:

$$\begin{aligned}
\phi^\star &= \min_\phi \mathbb{E}_{x\sim p_{\text{data}},\, \hat{x}\sim\mathcal{N}(x,\,\sigma^2\mathbf{I})}\left[\left\| s_\phi(\hat{x}) - \nabla_{\hat{x}} \log q_\sigma(\hat{x}|x)\right\|^2\right] \\
&= \min_\phi \mathbb{E}_{x\sim p_{\text{data}},\, \hat{x}\sim\mathcal{N}(x,\,\sigma^2\mathbf{I})}\left[\left\| s_\phi(\hat{x}) - \left(-\frac{\hat{x}-x}{\sigma^2}\right)\right\|^2\right] \\
&= \min_\phi \mathbb{E}_{x\sim p_{\text{data}},\, \epsilon\sim\mathcal{N}(0,\,\mathbf{I})}\left[\left\| s_\phi(x+\sigma\epsilon) + \frac{\epsilon}{\sigma}\right\|^2\right]
\end{aligned}$$

Denoising Score Matching

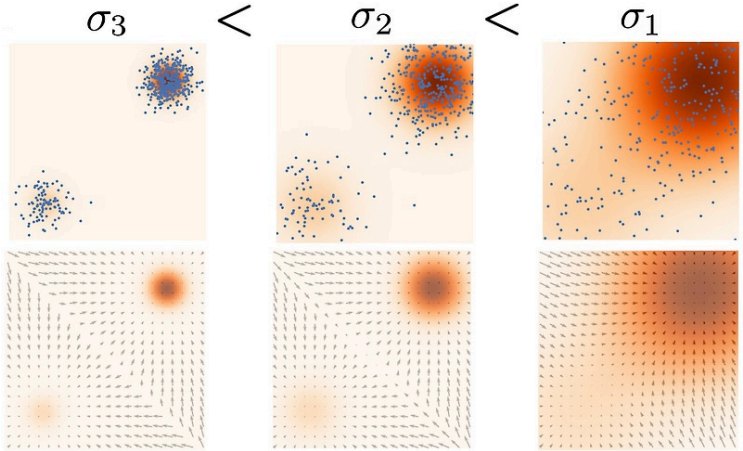


Denoising Score Matching



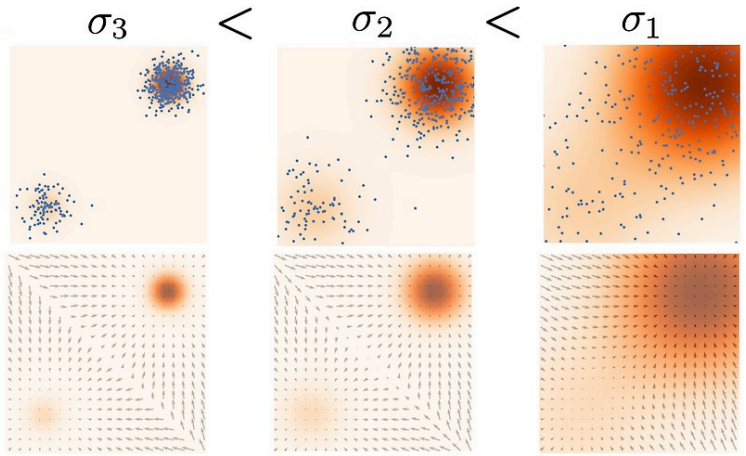
A fixed σ cannot actually yield to an usable model.

Denoising Score Matching



A fixed σ cannot actually yield to an usable model.
However,

Denoising Score Matching



A fixed σ cannot actually yield to an usable model.
However, many σ can.

Noise Conditional Score Networks

Noise Conditional Score Networks

Select a sequence of noise levels: $\{\sigma_i\}_{i=1}^{L}$, with $\sigma_1 > \sigma_2 > \cdots > \sigma_L$

[4] [Song & Ermon, 2019]

Noise Conditional Score Networks

Select a sequence of noise levels: $\{\sigma_i\}_{i=1}^{L}$, with $\sigma_1 > \sigma_2 > \cdots > \sigma_L$
Learn a **score function** $\mathbf{s}_\phi(\hat{x},\ \sigma)$ to approximate $\nabla_{\hat{x}} \log q_\sigma(\hat{x} \mid x)$, $\forall\sigma$ and $\forall x$.

[4] [Song & Ermon, 2019]

Noise Conditional Score Networks

Select a sequence of noise levels: $\{\sigma_i\}_{i=1}^{L}$, with $\sigma_1 > \sigma_2 > \cdots > \sigma_L$
Learn a **score function** $\mathbf{s}_\phi(\hat{x},\ \sigma)$ to approximate $\nabla_{\hat{x}} \log q_\sigma(\hat{x} \mid x)$, $\forall\sigma$ and $\forall x$.

[4] [Song & Ermon, 2019]

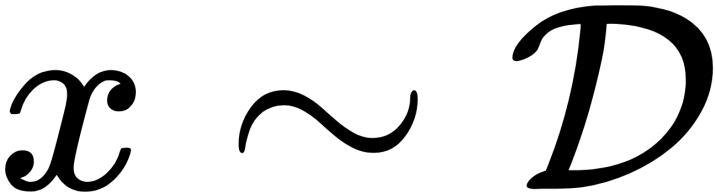
Noise Conditional Score Networks

For a minibatch of size N, the loss for a noise level σ_i is:

$$\mathcal{L}_i(\phi) = \frac{1}{N} \sum_{n=1}^{N} \left\| s_\phi(\hat{x}^{(n)}, \sigma_i) + \frac{\hat{x}^{(n)} - x^{(n)}}{\sigma_i^2} \right\|^2$$

$$\mathcal{L}_i(\phi) = \frac{1}{N} \sum_{n=1}^{N} \left\| s_\phi(x^{(n)} + \sigma_i \epsilon^{(n)}, \sigma_i) + \frac{\epsilon^{(n)}}{\sigma_i} \right\|^2$$

NCSN – Example



NCSN – Example

$$\sim \mathcal{D}$$

NCSN – Example

$$\sigma \sim \{\sigma_1, \ldots, \sigma_L\}$$

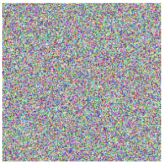
NCSN – Example

$$\sigma$$

$$\epsilon \sim \mathcal{N}(0, I)$$

NCSN – Example

σ



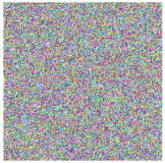
$\sim \mathcal{N}(0, I)$

NCSN – Example

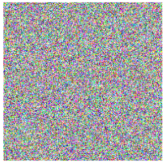
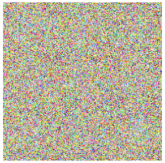
$$\sigma \times \epsilon$$

NCSN – Example

$\sigma \times$



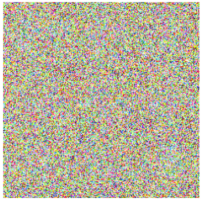
NCSN – Example

$\sigma \times$  $=$ 

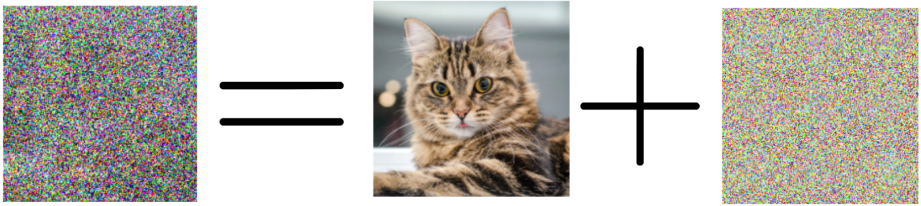
NCSN – Example

$$\hat{x} = x + \sigma\epsilon$$

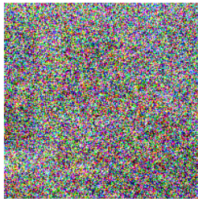
NCSN – Example

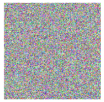
$$\hat{x} = \text{  } + \text{  }$$

NCSN – Example



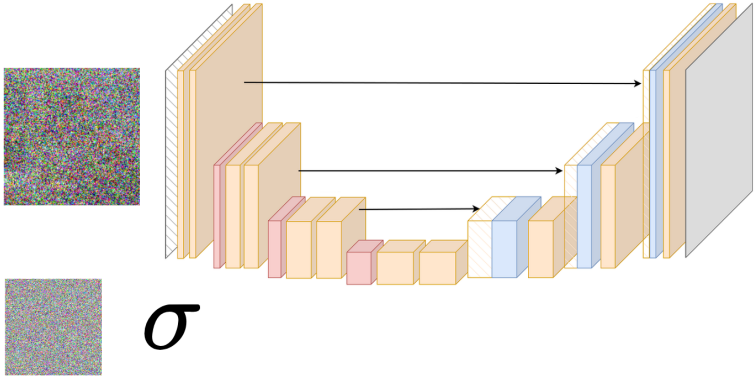
NCSN – Example



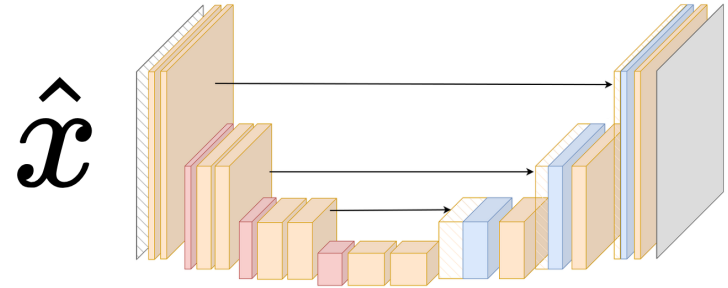


σ

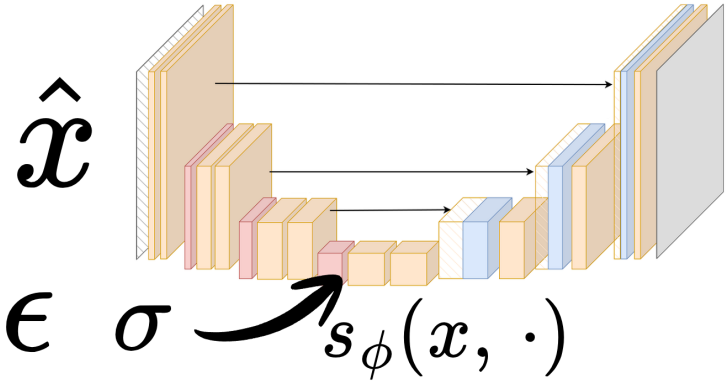
NCSN – Example



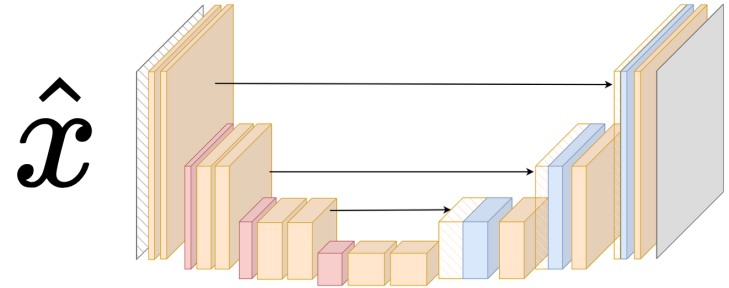
NCSN – Example



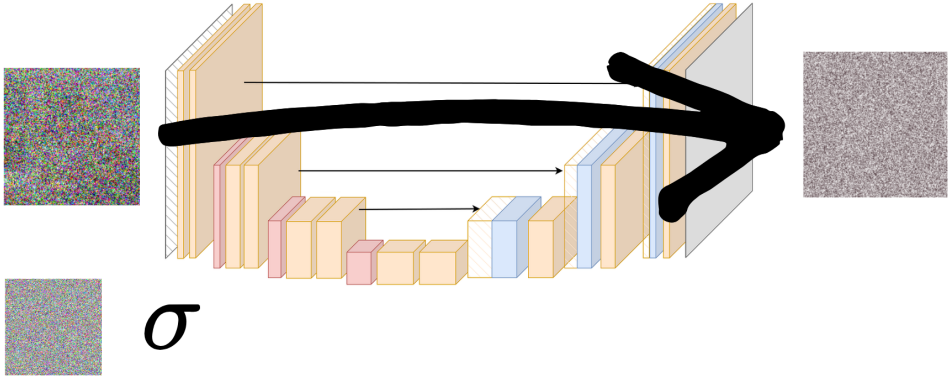
NCSN – Example



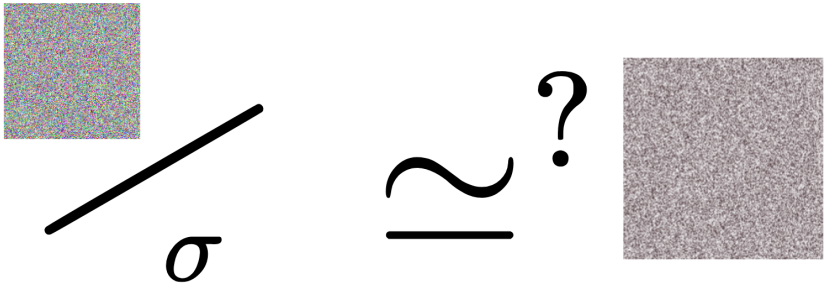
NCSN – Example



NCSN – Example

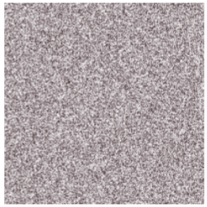


NCSN – Example

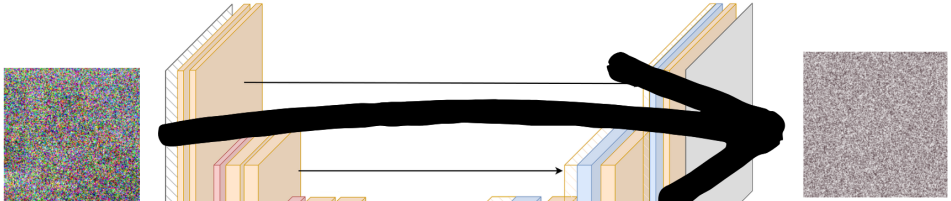


NCSN – Example

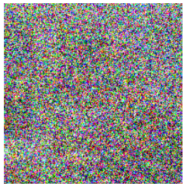
$$\epsilon / \sigma \simeq ?$$

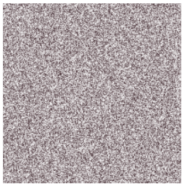


NCSN – Example

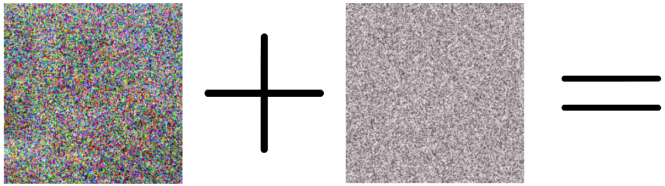


NCSN – Example

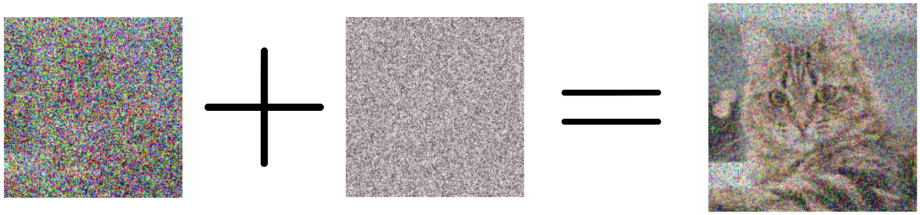




NCSN – Example



NCSN – Example



Generate with NCSN

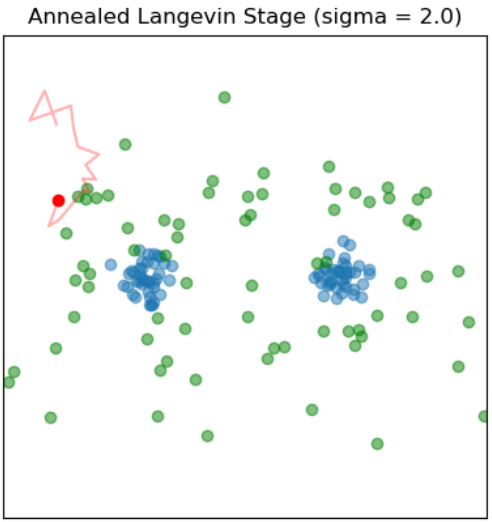
Annealed Langevin Dynamics.

Initialize x_0
for $i \leftarrow 1$ to L **do**
  $\alpha_i \leftarrow \epsilon \cdot \sigma_i^2 / \sigma_L^2$
  for $t \leftarrow 1$ to T **do**
    $\mathbf{z}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
    $x_t \leftarrow x_{t-1} + \alpha_i \mathbf{s}(x_{t-1}, \sigma_i) + \sqrt{\alpha_i}\, \mathbf{z}_t$
  end for
  $x_0 \leftarrow x_T$
end for
return x_L

Generate with NCSN

Annealed Langevin Dynamics.

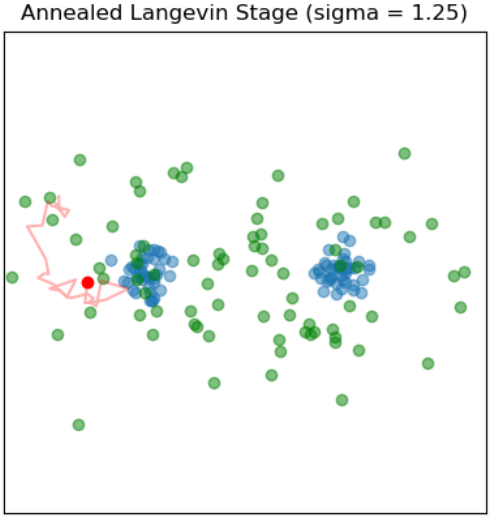
1: Initialize x_0
2: **for** $i \leftarrow 1$ to L **do**
3: $\quad \alpha_i \leftarrow \epsilon \cdot \sigma_i^2 / \sigma_L^2$
4: $\quad$ **for** $t \leftarrow 1$ to T **do**
5: $\quad\quad \mathbf{z}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
6: $\quad\quad x_t \leftarrow x_{t-1} + \alpha_i \mathbf{s}(x_{t-1}, \sigma_i) + \sqrt{\alpha_i}\, \mathbf{z}_t$
7: $\quad$ **end for**
8: $\quad x_0 \leftarrow x_T$
9: **end for**
10: **return** x_L



Generate with NCSN

Annealed Langevin Dynamics.

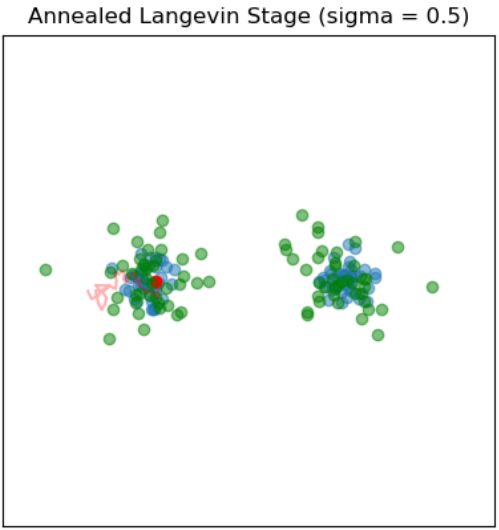
1: Initialize x_0
2: **for** $i \leftarrow 1$ to L **do**
3: $\quad \alpha_i \leftarrow \epsilon \cdot \sigma_i^2 / \sigma_L^2$
4: $\quad$ **for** $t \leftarrow 1$ to T **do**
5: $\quad\quad \mathbf{z}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
6: $\quad\quad x_t \leftarrow x_{t-1} + \alpha_i \mathbf{s}(x_{t-1}, \sigma_i) + \sqrt{\alpha_i}\, \mathbf{z}_t$
7: $\quad$ **end for**
8: $\quad x_0 \leftarrow x_T$
9: **end for**
10: **return** x_L



Generate with NCSN

Annealed Langevin Dynamics.

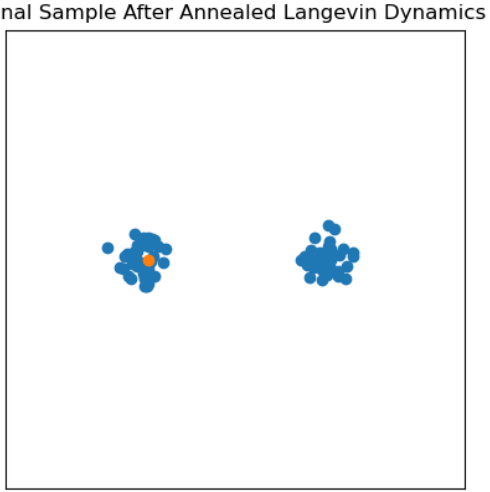
Initialize x_0
for $i \leftarrow 1$ to L **do**
  $\alpha_i \leftarrow \epsilon \cdot \sigma_i^2 / \sigma_L^2$
  for $t \leftarrow 1$ to T **do**
    $\mathbf{z}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
    $x_t \leftarrow x_{t-1} + \alpha_i \mathbf{s}(x_{t-1}, \sigma_i) + \sqrt{\alpha_i}\, \mathbf{z}_t$
  end for
  $x_0 \leftarrow x_T$
end for
return x_L



Generate with NCSN

Annealed Langevin Dynamics.

Initialize x_0
for $i \leftarrow 1$ to L **do**
  $\alpha_i \leftarrow \epsilon \cdot \sigma_i^2 / \sigma_L^2$
  for $t \leftarrow 1$ to T **do**
    $\mathbf{z}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
    $x_t \leftarrow x_{t-1} + \alpha_i \mathbf{s}(x_{t-1}, \sigma_i) + \sqrt{\alpha_i}\, \mathbf{z}_t$
  end for
  $x_0 \leftarrow x_T$
end for
return x_L



NCSN – Training

Training

Input: dataset D, noise levels $\sigma_1 > \sigma_2 > ... > \sigma_L$
Initialize network parameters θ

while not converged **do:**
  $x \leftarrow$ sample from D
  $\sigma \leftarrow$ sample uniformly from $\sigma_1, ..., \sigma_L$
  $\epsilon \leftarrow$ sample from $\mathcal{N}(\mathbf{0}, \mathbf{I})$
  $\hat{x} \leftarrow x + \sigma\epsilon$
  $y \leftarrow -\epsilon/\sigma$
  $\mathcal{L} \leftarrow ||s_\theta(\hat{x}, \sigma) - y||^2$
  update network parameters θ
end while

Stefano Ermon

Yang Song

elicsir
Elevating Italian Computer Science Innovation and Research

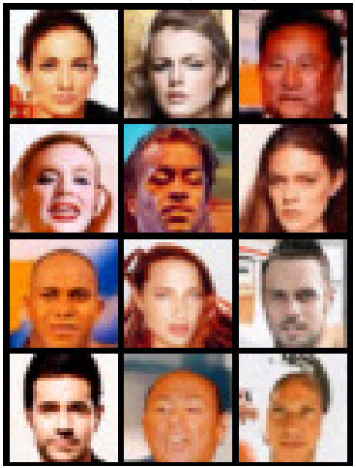
Stefano Ermon

Yang Song

Gianfranco Bilardi

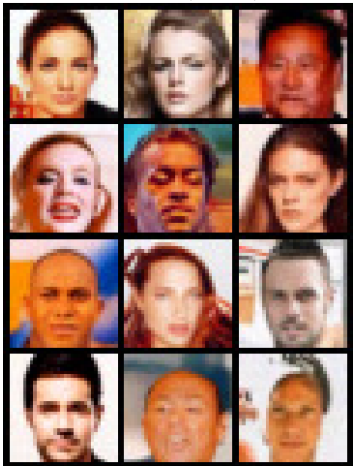
Evolution of Diffusion Models

NCSN (2019)



Evolution of Diffusion Models

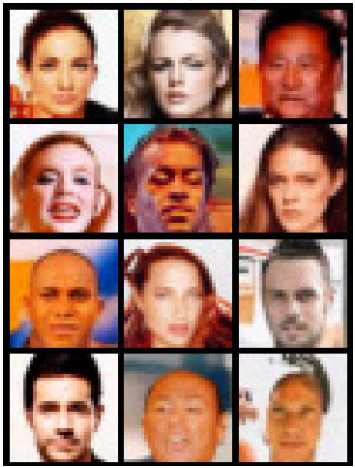
NCSN (2019)



DDPM (2020)

Evolution of Diffusion Models

NCSN (2019)



DDPM (2020)

Latent Diffusion (2022)

References

Foundations

Hinton, G. E., & Sejnowski, T. J. (1985). *Learning and Relearning in Boltzmann Machines.*
LeCun, Y., Chopra, S., & Hadsell, R. (2006). *A Tutorial on Energy-Based Learning.*

Score Matching

Hyvärinen, A. (2005). *Estimation of Non-Normalized Statistical Models by Score Matching.*
Vincent, P. (2010). *A Connection Between Score Matching and Denoising Autoencoders.*

Noise Conditional Score Networks

Song, Y. & Ermon, S. (2019). *Generative Modeling by Estimating Gradients of the Data Distribution.*

Thank you!

Hyvärinen Score Matching Result

$$\phi^\star = \min_\phi \mathbb{E}_{p_{\text{data}}} [\|s_\phi(x) - s_{\text{data}}(x)\|^2]$$

Hyvärinen Score Matching Result

$$\phi^\star = \min_\phi \mathbb{E}_{p_{\text{data}}} \left[\| s_\phi(x) - s_{\text{data}}(x) \|^2 \right]$$

$$\| s_\phi(x) - s_{\text{data}}(x) \|^2 = s_\phi(x)^2 + s_{\text{data}}(x)^2 - 2\, s_\phi(x)\, s_{\text{data}}(x)$$

Hyvärinen Score Matching Result

$$s_{\text{data}}(x) = \nabla_x \log p_{\text{data}}(x)$$

$$\begin{aligned}
\mathbb{E}_{p_{\text{data}}}\left[-2\, s_\phi(x)\, s_{\text{data}}(x)\right] &= -2 \int_{-\infty}^{\infty} s_\phi(x)\, s_{\text{data}}(x)\, p_{\text{data}}(x)\, dx \\
&= -2 \int_{-\infty}^{\infty} s_\phi(x)\, \frac{\nabla_x p_{\text{data}}(x)}{p_{\text{data}}(x)}\, p_{\text{data}}(x)\, dx \\
&= -2 \int_{-\infty}^{\infty} s_\phi(x)\, \nabla_x p_{\text{data}}(x)\, dx
\end{aligned}$$

Hyvärinen Score Matching Result

$$-2 \int_{-\infty}^{\infty} s_\phi(x)\, \nabla_x p_{\text{data}}(x)\, dx$$

$$= - \int_{-\infty}^{\infty} s_\phi(x)\, \nabla_x p_{\text{data}}(x)\, dx$$

$$= - \left[s_\phi(x)\, p_{\text{data}}(x) \right] \Big|_{-\infty}^{\infty} + \int_{-\infty}^{\infty} \text{div}\left(s_\phi(x) \right)\, p_{\text{data}}(x)\, dx$$

$$= - \left[s_\phi(x)\, p_{\text{data}}(x) \right] \Big|_{-\infty}^{\infty} + \mathbb{E}_{p_{\text{data}}} \left[\text{div}(s_\phi(x)) \right]$$

$\left[s_\phi(x)\, p_{\text{data}}(x) \right] \Big|_{-\infty}^{\infty} \to 0$