Score Matching Based Generative Model and Diffusion Model

김기웅
kwkim02@g.skku.edu
Computer Vision Core
2024/10/01

TRAIN AND TEST

Contents

- Score-Based Generative Model(NCSN)
- Diffusion Model(DDPM)
- Score-Based Generative Model through SDE

Contents

- Score-Based Generative Model(NCSN)
- Diffusion Model(DDPM)
- Score-Based Generative Model through SDE

Generative Modeling by Estimating Gradients of the Data Distribution

Yang Song
Stanford University
yangsong@cs.stanford.edu

Stefano Ermon
Stanford University
ermon@cs.stanford.edu

SCORE-BASED GENERATIVE MODELING THROUGH STOCHASTIC DIFFERENTIAL EQUATIONS

Yang Song*
Stanford University
yangsong@cs.stanford.edu

Jascha Sohl-Dickstein
Google Brain
jaschasd@google.com

Diederik P. Kingma
Google Brain
durk@google.com

Abhishek Kumar
Google Brain
abhishk@google.com

Stefano Ermon
Stanford University
ermon@cs.stanford.edu

Ben Poole
Google Brain
pooleb@google.com

TRAIN AND TEST

Score-Based Generative Model

Introduction

Generative Model

Likelihood-Based Method
- Specialized architecture
- Surrogate loss(ELBO)

GAN
- Unstable training (Adversarial)
- Unsuitable for
  high-dimensional data

Score-Based Generative Model

Introduction

Generative Model

Likelihood-Based Method
- Specialized architecture
- Surrogate loss(ELBO)

GAN
- Unstable training (Adversarial)
- Unsuitable for high-dimensional data

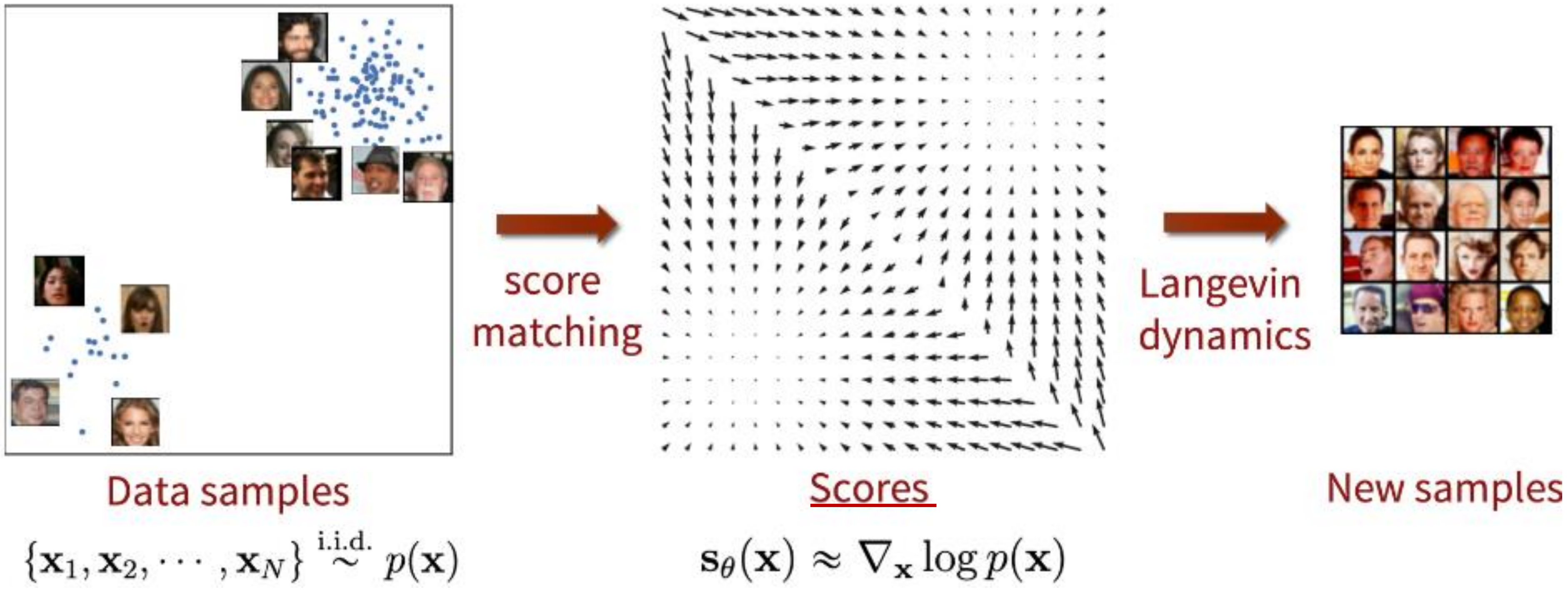
Score-Based Model
- ~~Adversarial training~~
- ~~MCMC sampling~~
- ~~Special architecture~~

Score-Based Generative Model

Overview

Score = Gradient of log(pdf)

$= \nabla_x logp(x)$



TRAIN AND TEST

Score-Based Generative Model

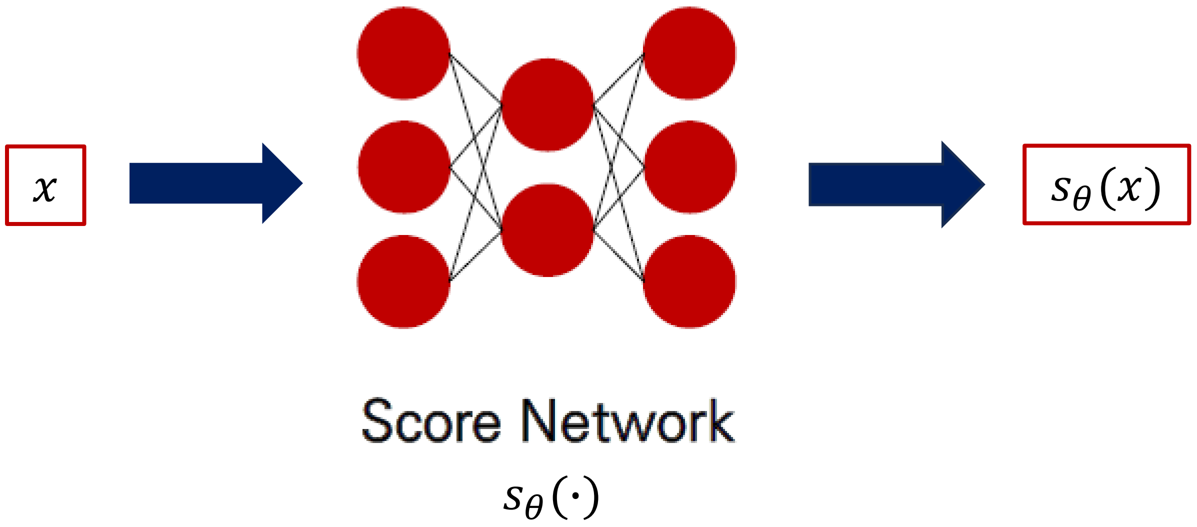
Score-Matching

- Score Network $s_\theta(\cdot) : R^D \rightarrow R^D$

  -U-Net in paper

- Training objective:
  MSE between network output and score

  -Network estimates score



Training objective

MSE($s_\theta(x),\ \nabla_x \log(p(x))$)

Score-Based Generative Model

Score-Matching

Training objective

$$\boxed{\text{MSE}(\ s_\theta(x),\ \nabla_x \log\big(p(x)\big))}$$

TRAIN AND TEST

Score-Based Generative Model

Score-Matching

Training objective

$$\boxed{\text{MSE}(\, s_\theta(x),\ \nabla_x \log(p(x)))}$$

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

Score-Based Generative Model

Score-Matching

Training objective

$$\boxed{\text{MSE}(\ s_\theta(x),\ \nabla_x \log(p(x)))}$$

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

Score-Matching

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

Using score matching, we can directly train a score network $\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x})$ to estimate $\nabla_{\mathbf{x}} \log p_{\text{data}}(\mathbf{x})$ training a model to estimate $p_{\text{data}}(\mathbf{x})$ first.

TRAIN AND TEST

Score-Based Generative Model

Score-Matching

Training objective

$\text{MSE}(\ s_\theta(x),\ \nabla_x \log(p(x)))$

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

Score-Matching

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

Jacobian Matrix(d*d)

Complex computation in high-dimension

TRAIN AND TEST

Score-Based Generative Model

Score-Matching

Training objective

$$\text{MSE}(\, s_\theta(x),\ \nabla_x \log(p(x)))$$

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

Score-Matching

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

Denoising Score Matching

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

Sliced score matching

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

TRAIN AND TEST

Score-Based Generative Model

Score-Matching

Training objective

$$\text{MSE}(\, s_\theta(x),\ \nabla_x \log(p(x)))$$

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

Score-Matching

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

Denoising Score Matching

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

Sliced score matching

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

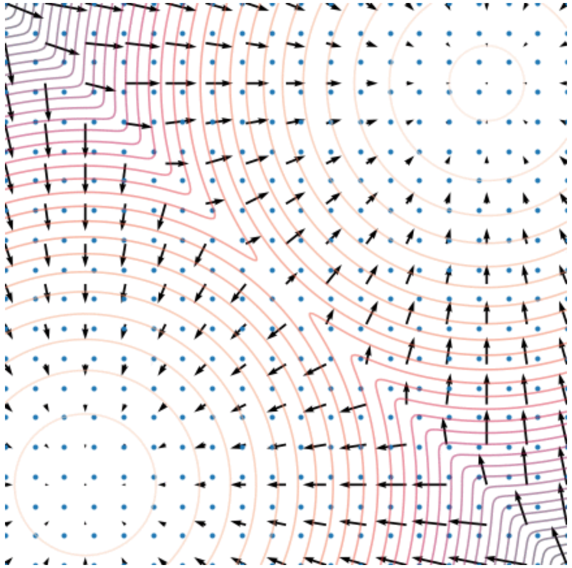
TRAIN AND TEST

Score-Based Generative Model

Langevin dynamics

- Using only score function
- Trained score network : $\mathbf{s}_\theta(\mathbf{x}) \approx \nabla_\mathbf{x} \log p(\mathbf{x})$
- ϵ : fixed step size
- z_t: noise $\sim N(0, I)$
- x_0= random noise
- Some error is negligible
  when ϵ is sufficiently small and T is sufficiently large

$$\tilde{\mathbf{x}}_t = \tilde{\mathbf{x}}_{t-1} + \frac{\epsilon}{2}\nabla_\mathbf{x} \log p(\tilde{\mathbf{x}}_{t-1}) + \sqrt{\epsilon}\,\mathbf{z}_t, \quad t = 0, 1, \cdots, \mathrm{T}$$



TRAIN AND TEST

Challenges of Score-Based Generative Model

Manifold Hypothesis

The manifold hypothesis states that data in the real world
tend to concentrate on low dimensional manifolds
embedded in a high dimensional space (a.k.a., the ambient space).

Difficulty

1. Since the score $\nabla_x \log(p(x))$ is a gradient taken in the ambient space, it is undefined when x is confined to a low dimensional manifold.

2. The score matching objective provides a consistent score estimator only when the support of the data distribution is the whole space.

TRAIN AND TEST

Challenges of Score-Based Generative Model

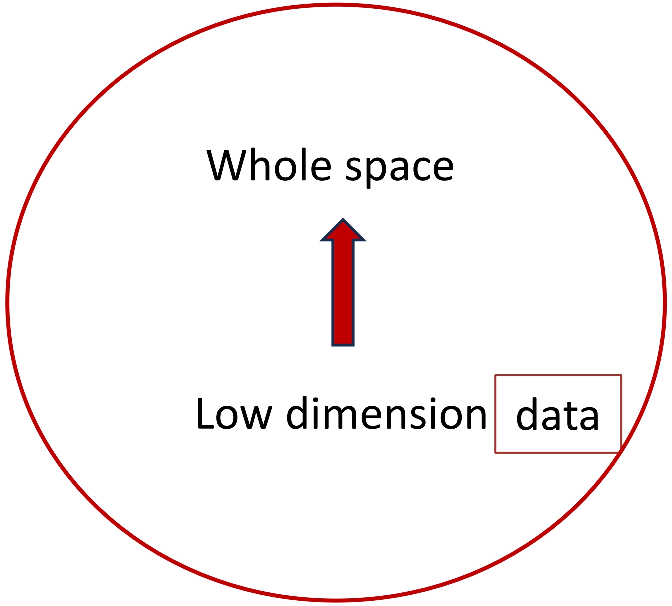
Manifold Hypothesis

The manifold hypothesis states that data in the real world tend to concentrate on low dimensional manifolds embedded in a high dimensional space (a.k.a., the ambient space).

Difficulty

1. Since the score $\nabla_x \log(p(x))$ is a gradient taken in the ambient space, it is undefined when x is confined to a low dimensional manifold.

2. The score matching objective provides a consistent score estimator only when the support of the data distribution is the whole space.



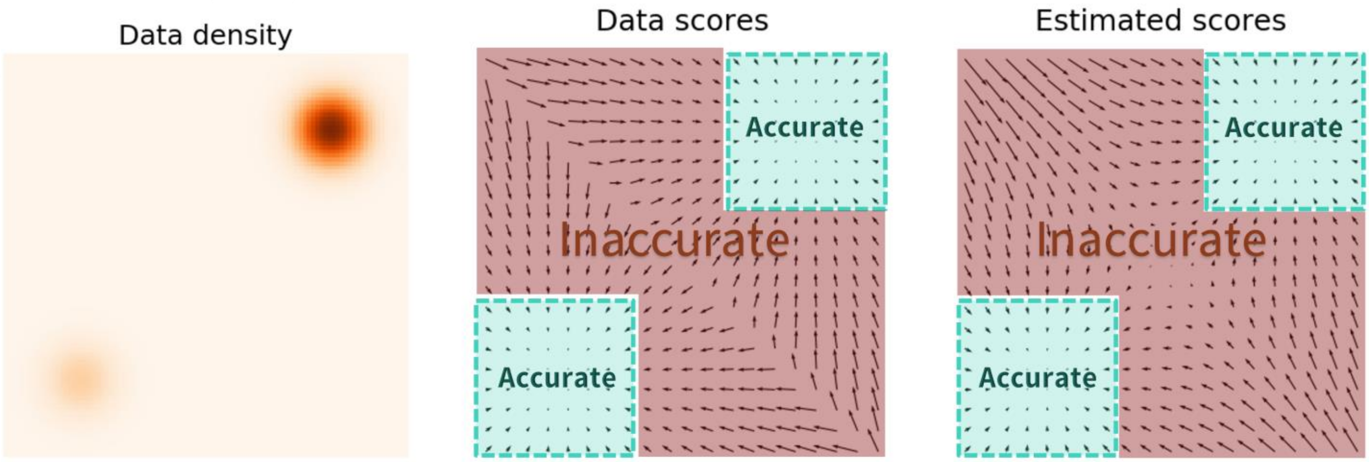
Challenges of Score-Based Generative Model

Low data density regions

Difficulty

Due to lack of data samples
1. Inaccurate score estimation with score matching
   lack of data samples

2. Slow mixing of Langevin dynamics



$$p_{data} = \frac{1}{5}N\big((-5,-5),I\big) + \frac{4}{5}N((5,5),I))$$

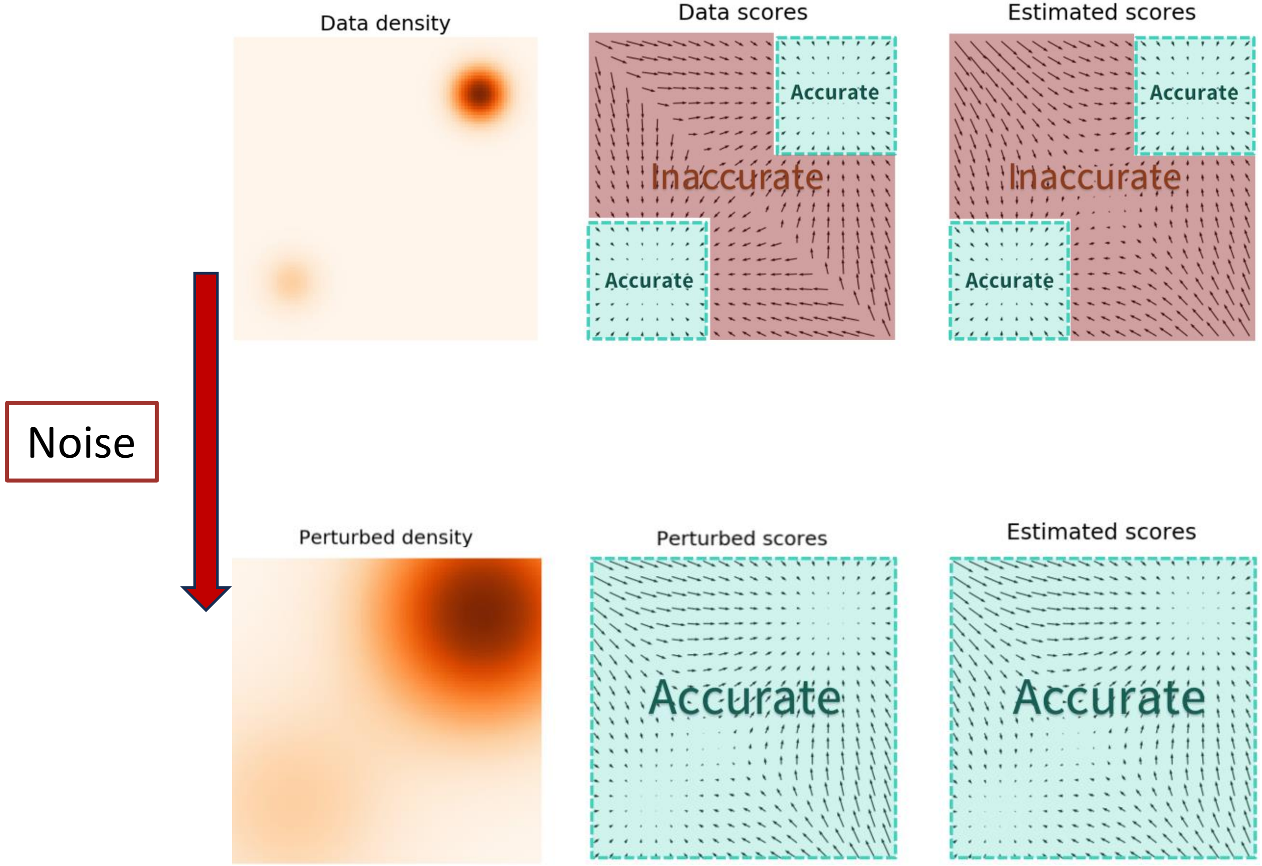
TRAIN AND TEST

Noise Conditional Score Networks(NCSN)

Problem-Solving

Adding multiple-level gaussian noise

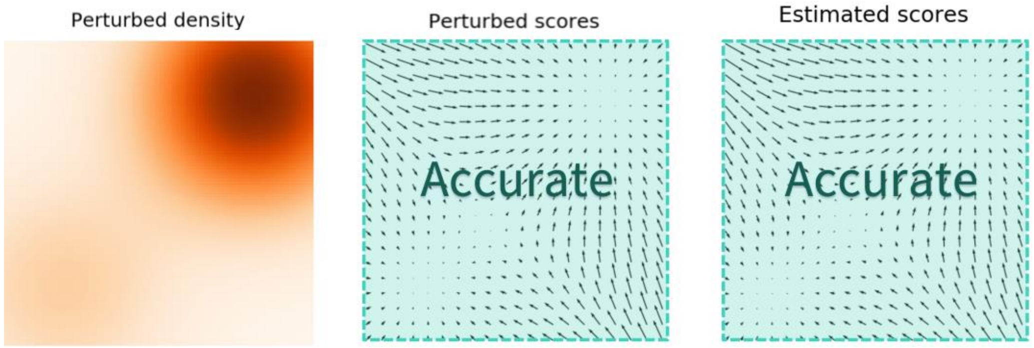
- Global – Distributing data to whole space
- Large – Filling low density regions
- Multiple level – improving mixing rate



Noise Conditional Score Networks(NCSN)

Denoising score-matching

- $q_\sigma(\tilde{x}|x)$: Gaussian noise
- $\tilde{x}$: data added noise (perturbing data)
- $q_\sigma(\tilde{\mathbf{x}}) \triangleq \int q_\sigma(\tilde{\mathbf{x}} \mid \mathbf{x}) p_{\text{data}}(\mathbf{x}) \mathrm{d}\mathbf{x}$: perturbed data distribution
- Small noise → $\mathbf{s}_{\boldsymbol{\theta}^*}(\mathbf{x}) = \nabla_{\mathbf{x}} \log q_\sigma(\mathbf{x}) \approx \nabla_{\mathbf{x}} \log p_{\text{data}}(\mathbf{x})$



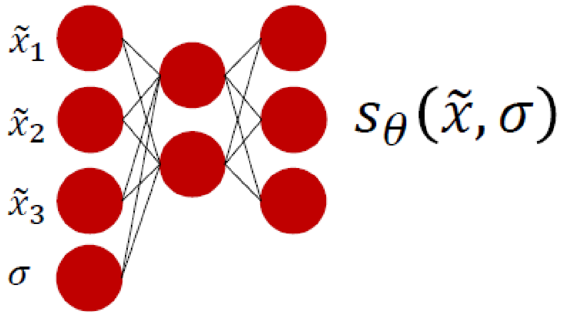
Perturbing an image with multiple scales of Gaussian noise

Noise Conditional Score Networks(NCSN)

Denoising score-matching

Multiple-level gaussian noise

- Large $\rightarrow$ small (like Fine-tunning)
  σ_1 $\qquad$ σ_L
- $q_\sigma(\tilde{\mathbf{x}} \mid \mathbf{x}) = \mathcal{N}(\tilde{\mathbf{x}} \mid \mathbf{x}, \sigma^2 I)$



Noise Conditional Score Network
(NCSN)

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

$$\Downarrow$$

$$\boxed{\frac{1}{2}\mathbb{E}_{p_{\text{data}}(\mathbf{x})}\mathbb{E}_{\tilde{\mathbf{x}}\sim\mathcal{N}(\mathbf{x},\sigma^2 I)}\left[\left\|\mathbf{s}_{\boldsymbol{\theta}}(\tilde{\mathbf{x}}, \underline{\sigma}) + \frac{\tilde{\mathbf{x}} - \mathbf{x}}{\underline{\sigma^2}}\right\|_2^2\right]}$$

Noise Conditional Score Networks(NCSN)

Annealed Langevin dynamics

$$\tilde{\mathbf{x}}_t = \tilde{\mathbf{x}}_{t-1} + \frac{\epsilon}{2}\nabla_{\mathbf{x}} \log p(\tilde{\mathbf{x}}_{t-1}) + \sqrt{\epsilon}\, \mathbf{z}_t$$

$$\tilde{\mathbf{x}}_t \leftarrow \tilde{\mathbf{x}}_{t-1} + \frac{\alpha_i}{2}\mathbf{s}_{\boldsymbol{\theta}}(\tilde{\mathbf{x}}_{t-1}, \underline{\sigma_i}) + \sqrt{\alpha_i}\, \mathbf{z}_t$$

Algorithm 1 Annealed Langevin dynamics.

Require: $\{\sigma_i\}_{i=1}^{L}, \epsilon, T.$

Initialize $\tilde{\mathbf{x}}_0$

for $i \leftarrow 1$ to L **do**

  $\boxed{\alpha_i \leftarrow \epsilon \cdot \sigma_i^2/\sigma_L^2}$ $\triangleright$ α_i is the step size.

  for $t \leftarrow 1$ to T **do**

    Draw $\mathbf{z}_t \sim \mathcal{N}(0, I)$

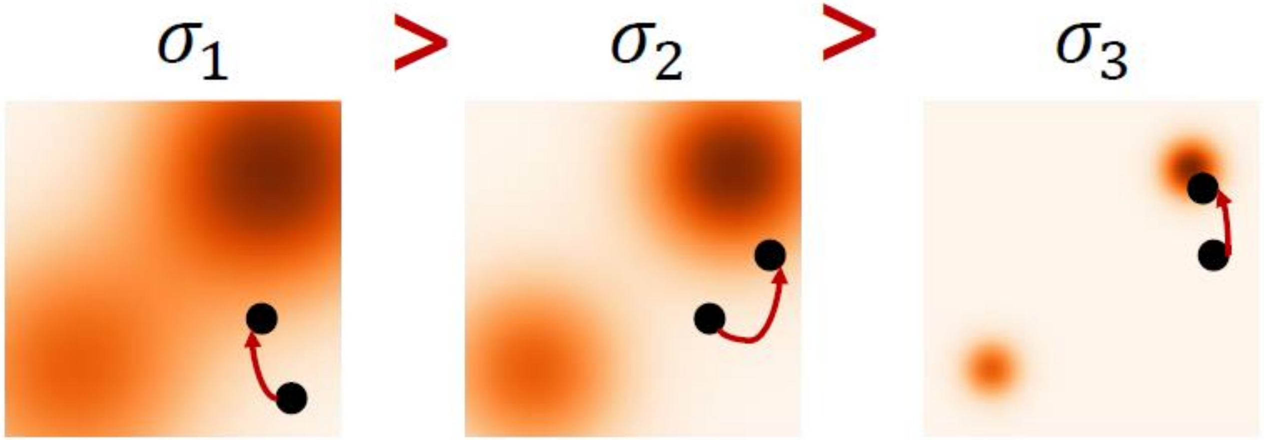
    $\tilde{\mathbf{x}}_t \leftarrow \tilde{\mathbf{x}}_{t-1} + \frac{\alpha_i}{2}\mathbf{s}_{\boldsymbol{\theta}}(\tilde{\mathbf{x}}_{t-1}, \sigma_i) + \sqrt{\alpha_i}\, \mathbf{z}_t$

  end for

  $\tilde{\mathbf{x}}_0 \leftarrow \tilde{\mathbf{x}}_T$

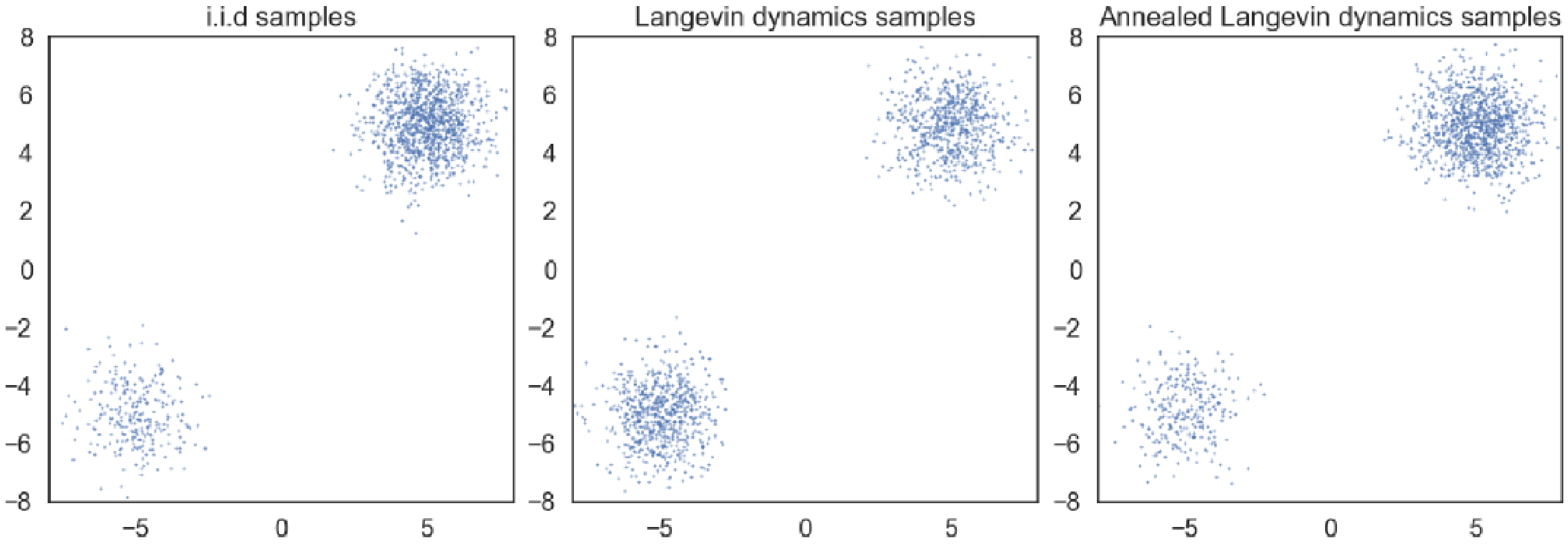
end for

return $\tilde{\mathbf{x}}_T$



Noise Conditional Score Networks(NCSN)

Annealed Langevin dynamics



(a) (b) (c)

Noise Conditional Score Networks(NCSN)

Experiment

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]	$7.86 \pm .07$	36.4
MoLM [45]	$7.90 \pm .10$	**18.9**
SNGAN [36]	$8.22 \pm .05$	21.7
ProgressiveGAN [25]	$8.80 \pm .05$	-
NCSN (Ours)	$\mathbf{8.87} \pm .12$	25.32
CIFAR-10 Conditional		
EBM [12]	8.30	37.9
SNGAN [36]	$8.60 \pm .08$	25.5
BigGAN [6]	**9.22**	**14.73**

Table 1: Inception and FID scores for CIFAR-10

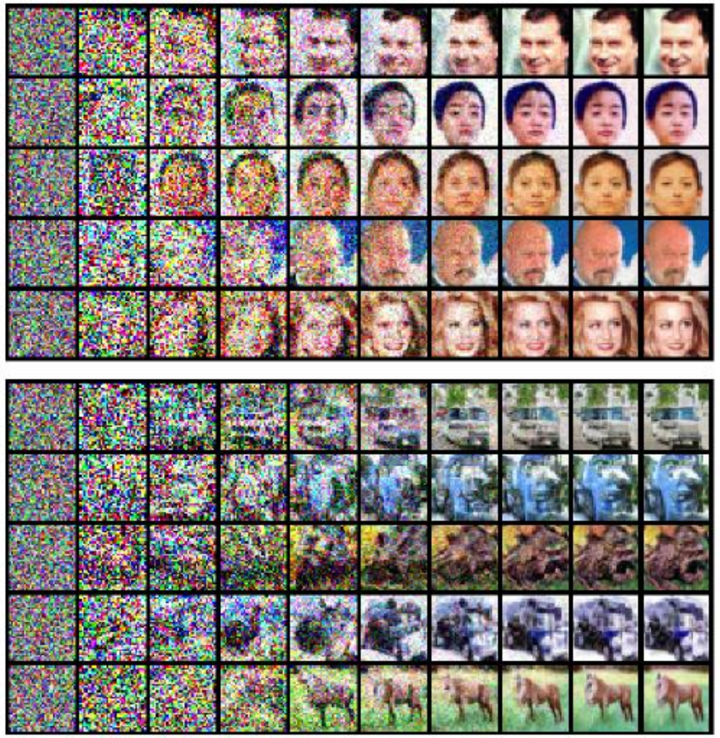


Figure 4: Intermediate samples of annealed Langevin dynamics.

Noise Conditional Score Networks(NCSN)

Experiment

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

Figure 5: Uncurated samples on MNIST, CelebA, and CIFAR-10 datasets.

Noise Conditional Score Networks(NCSN)

Experiment

Figure 6: Image inpainting on CelebA (**left**) and CIFAR-10 (**right**). The leftmost column of each figure shows the occluded images, while the rightmost column shows the original images.

TRAIN AND TEST

Denoising Diffusion Probability Model(DDPM)

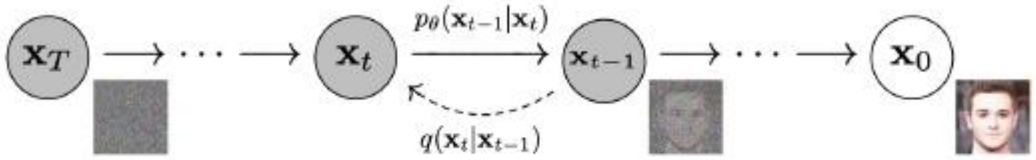
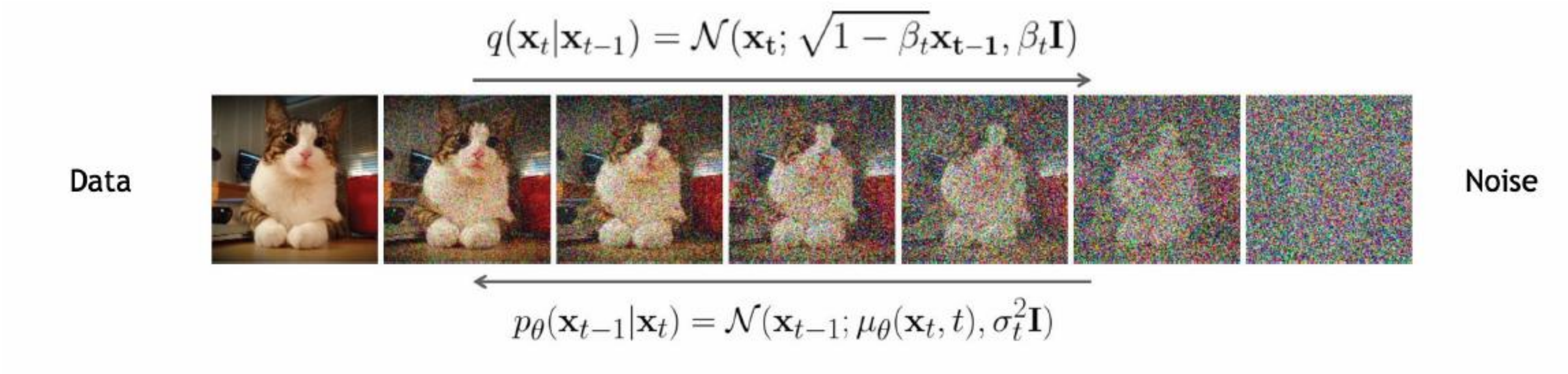


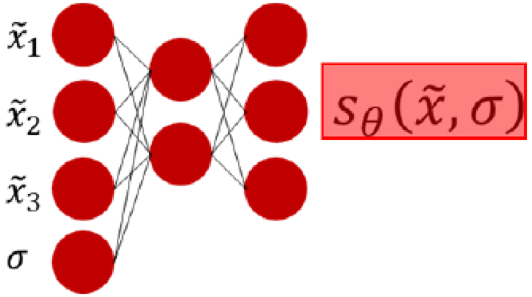
Figure 2: The directed graphical model considered in this work.



NCSN & DDPM

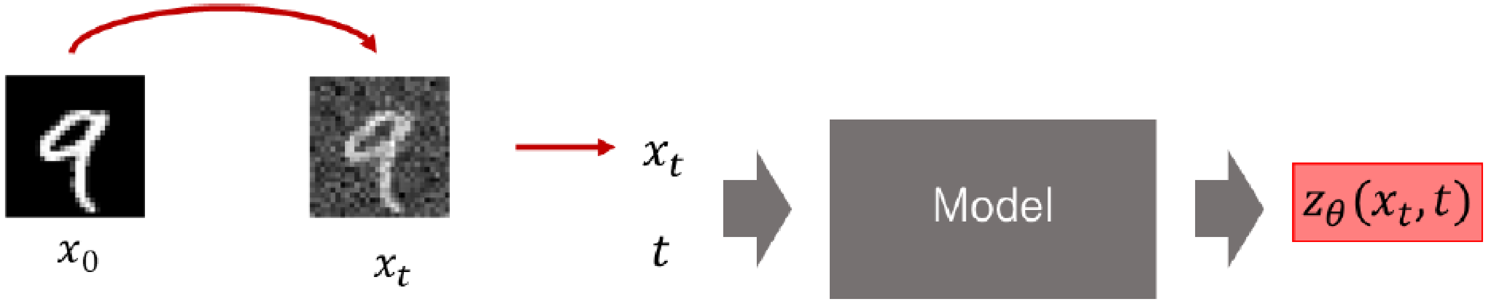
- Similar training objective

NCSN



$$\frac{1}{2} E_{q_\sigma(\tilde{x}|x) p_{data}(x)} \left[\left\| \boxed{s_\theta(\tilde{x}, \sigma)} - \frac{\boxed{\tilde{x} - x}}{\sigma^2} \right\|_2^2 \right]$$

DDPM



$$E_{x_0, z}[\| \boxed{z_t} - \boxed{z_\theta(x_t, t)} \|^2]$$

TRAIN AND TEST

NCSN & DDPM

- Similar new sampling

$$\boxed{\tilde{\mathbf{x}}_t \leftarrow \tilde{\mathbf{x}}_{t-1} + \frac{\alpha_i}{2}\mathbf{s}_{\boldsymbol{\theta}}(\tilde{\mathbf{x}}_{t-1}, \sigma_i) + \sqrt{\alpha_i}\,\mathbf{z}_t}$$

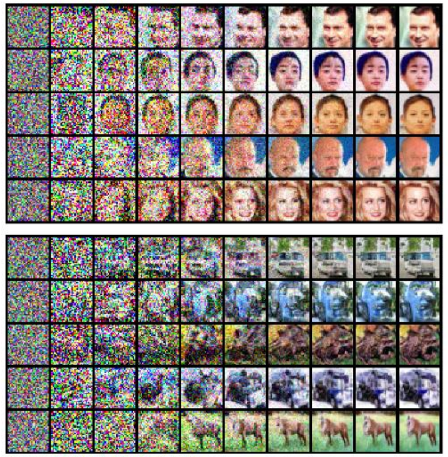
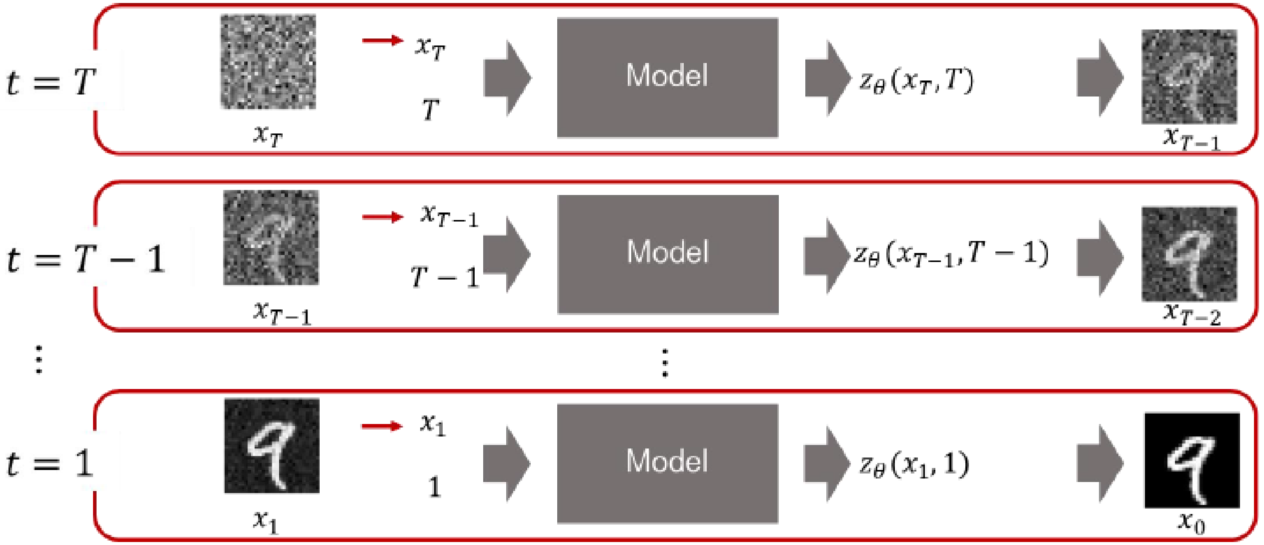


Figure 4: Intermediate samples of annealed Langevin dynamics.

NCSN



DDPM

Score-Based Generative Modeling Through SDEs

SCORE-BASED GENERATIVE MODELING THROUGH STOCHASTIC DIFFERENTIAL EQUATIONS

Yang Song*
Stanford University
yangsong@cs.stanford.edu

Jascha Sohl-Dickstein
Google Brain
jaschasd@google.com

Diederik P. Kingma
Google Brain
durk@google.com

Abhishek Kumar
Google Brain
abhishk@google.com

Stefano Ermon
Stanford University
ermon@cs.stanford.edu

Ben Poole
Google Brain
pooleb@google.com

Score-Based Generative Modeling Through SDEs

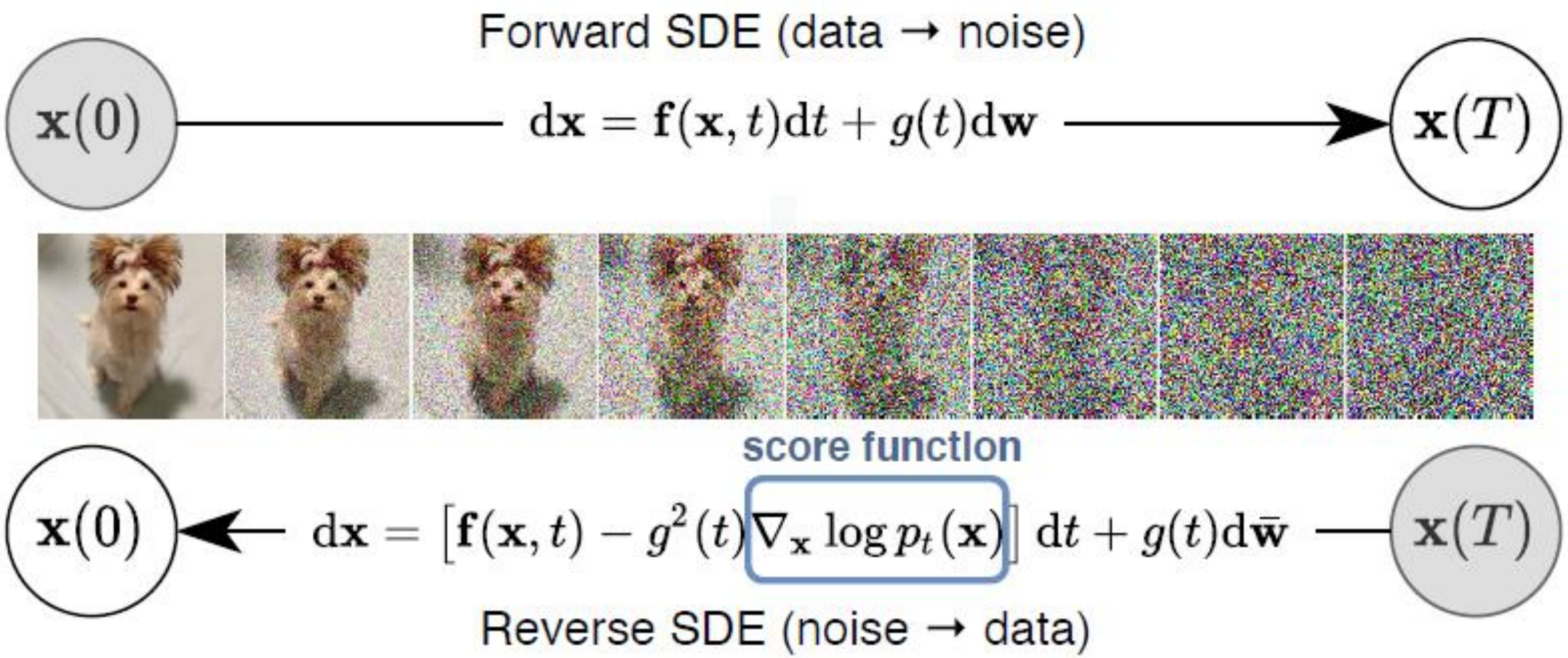
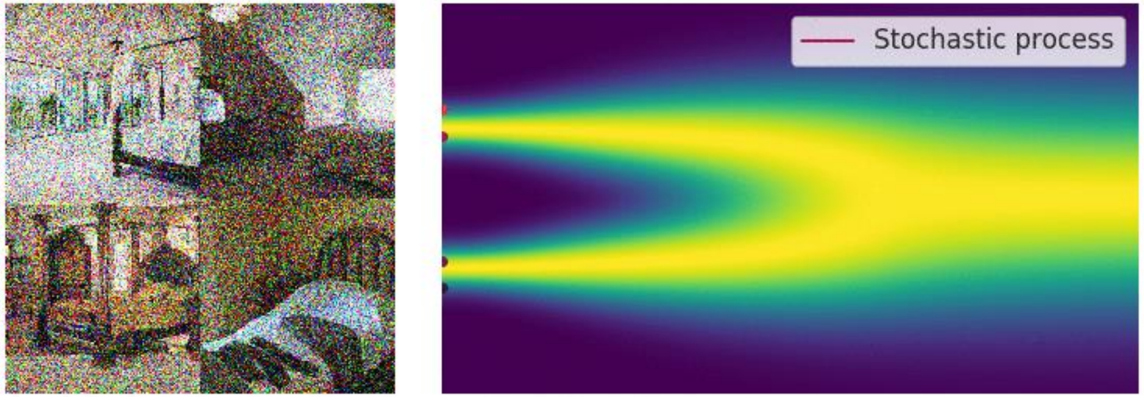


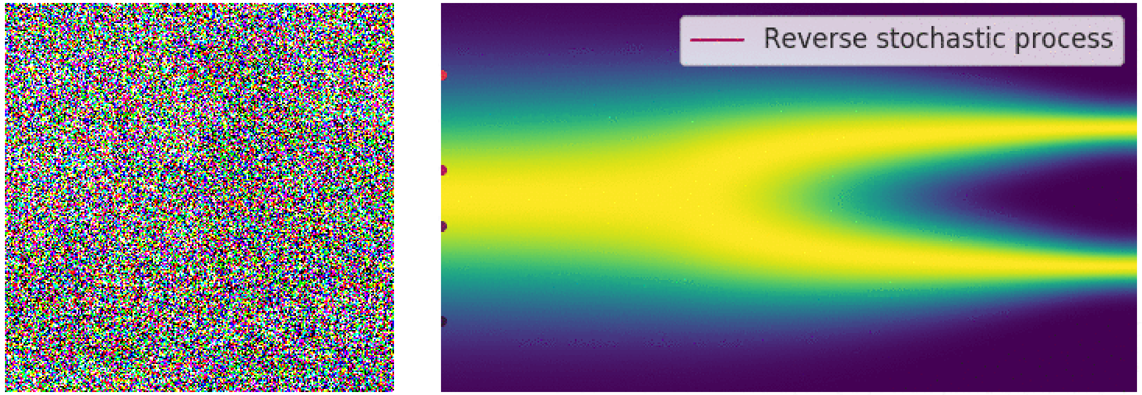
Figure 1: **Solving a reverse-time SDE yields a score-based generative model.** Transforming data to a simple noise distribution can be accomplished with a continuous-time SDE. This SDE can be reversed if we know the score of the distribution at each intermediate time step, $\nabla_{\mathbf{x}} \log p_t(\mathbf{x})$.

TRAIN AND TEST

Score-Based Generative Modeling Through SDEs



Perturbing data to noise with a continuous-time stochastic process.



Generate data from noise by reversing the perturbation procedure.

TRAIN AND TEST

Reference

- song, Y. (2021, May 21). *Generative Modeling by Estimating Gradients of the Data Distribution*. Yang-Song.Net. https://yang-song.net/blog/2021/score/

- Cho, H. (202, February 27). *[Open DMQA Seminar] Score-Based Generative Models and Diffusion Models*. Youtube. https://www.youtube.com/watch?v=d_x92vpIWFM&t=721s2

- Jang, Y. (2024, March 24). *Https://Process-Mining.Tistory.Com/211*. Tstory. https://process-mining.tistory.com/211

TRAIN AND TEST

TNT

TRAIN AND TEST