Improving Sampling Speed of Diffusion Models

Open DMQA Seminar
2023.02.10

조한샘

고려대학교
KOREA UNIVERSITY

CONTENTs

발표자 소개

- 조한샘
  - ✓ Data Mining & Quality Analytics Lab
  - ✓ 석·박통합과정 (2020.09~)
- 관심 연구 분야
  - ✓ Deep Generative Models
- Contact
  - ✓ chosam95@korea.ac.kr

Introduction

AI Image Generator

미술대회 우승까지 한 'AI 그림'…단순 표절일 뿐 vs 새로운 예술 도구

2022.09.10 08:00

이윤정 기자

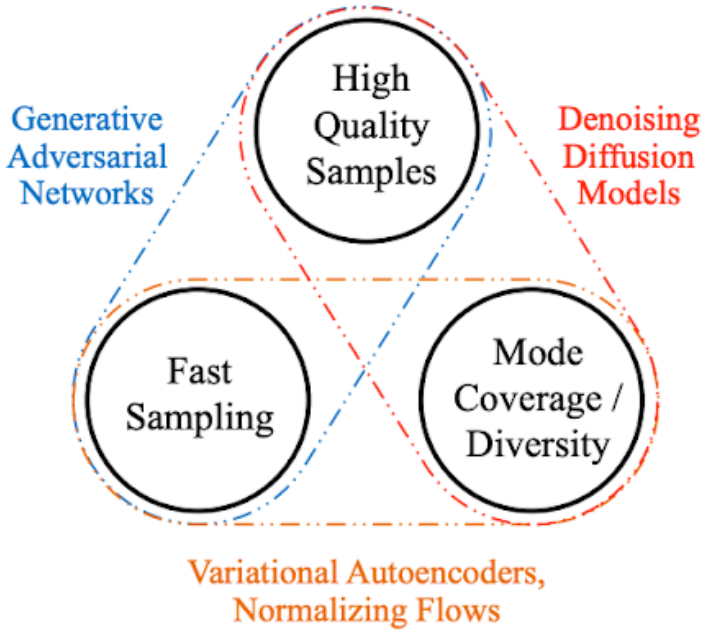
AI가 제작한 작품 <스페이스 오페라 극장> 옆에 미국 콜로라도 주립 박람회 미술대회 우승 표시가 붙어 있다. 제이슨 엘런 디스코드 갈무리·연합뉴스

https://m.khan.co.kr/economy/economy-general/article/202209100800001#c2b
https://www.youtube.com/watch?v=Q38vjk9uZkM

Introduction

Generative Learning Trilemma

- **High-quality sampling**: 모델이 좋은 퀄리티의 이미지를 생성
- **Mode coverage / Diversity**: 다양한 이미지를 생성
- **Fast sampling**: 이미지를 빠르게 생성



https://developer.nvidia.com/blog/improving-diffusion-models-as-an-alternative-to-gans-part-1/

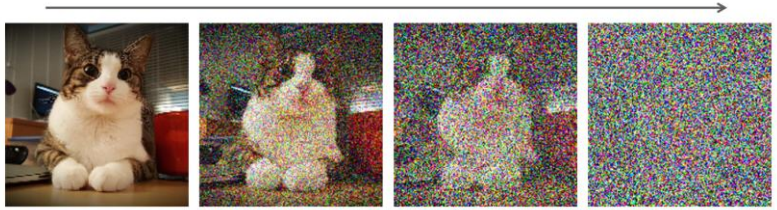
Diffusion Models

Denoising Diffusion Probabilistic Models (DDPM)

- **Forward Process**: Data (x_0) → Noise (x_T) / **Fixed**
- **Reverse Process**: Noise (x_T) → Data (x_0) / **Learning**
- 노이즈를 제거하는 reverse process를 학습 → 랜덤 노이즈로부터 데이터 생성 가능

Forward Process $q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I})$

x_0 (Data)



x_T (Noise)

$x_T \sim N(0,1)$

Reverse Process $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \sigma_t^2\mathbf{I})$

https://developer.nvidia.com/blog/improving-diffusion-models-as-an-alternative-to-gans-part-1/

Diffusion Models

Denoising Diffusion Probabilistic Models (DDPM)

- Training: 각 시점에서 제거할 노이즈에 대해 학습
- Sampling: 각 시점 마다 노이즈 제거하며 데이터 생성

Algorithm 1 Training

1: **repeat**
2: $\mathbf{x}_0 \sim q(\mathbf{x}_0)$
3: $t \sim \text{Uniform}(\{1, \dots, T\})$
4: $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
5: Take gradient descent step on
$$\nabla_\theta \left\| \epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, t) \right\|^2$$
6: **until** converged

Algorithm 2 Sampling

1: $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
2: **for** $t = T, \dots, 1$ **do**
3: $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\mathbf{z} = \mathbf{0}$
4: $\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon_\theta(\mathbf{x}_t, t) \right) + \sigma_t \mathbf{z}$
5: **end for**
6: **return** $\mathbf{x}_0$

Diffusion Models

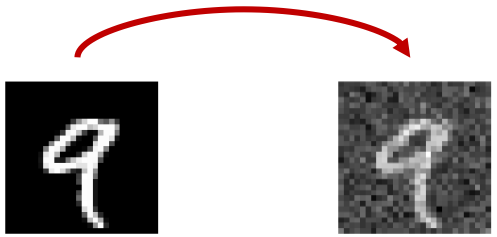
Algorithm 1 Training

1: **repeat**
2: $\mathbf{x}_0 \sim q(\mathbf{x}_0)$
3: $t \sim \text{Uniform}(\{1, \dots, T\})$
4: $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
5: Take gradient descent step on
$\nabla_\theta \left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t) \right\|^2$
6: **until** converged

Denoising Diffusion Probabilistic Models (DDPM)

- Training: 각 시점에서 제거할 노이즈에 대해 학습
- Forward process는 모든 시점을 거치지 않음 → 효율적으로 학습 가능

$$x_t = \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon$$



x_0 x_t

Model

$\epsilon_\theta(x_t, t)$

$\epsilon \sim N(0,1)$

$E_{x_0,\epsilon}[\|\epsilon - \epsilon_\theta(x_t, t)\|^2]$

Model output

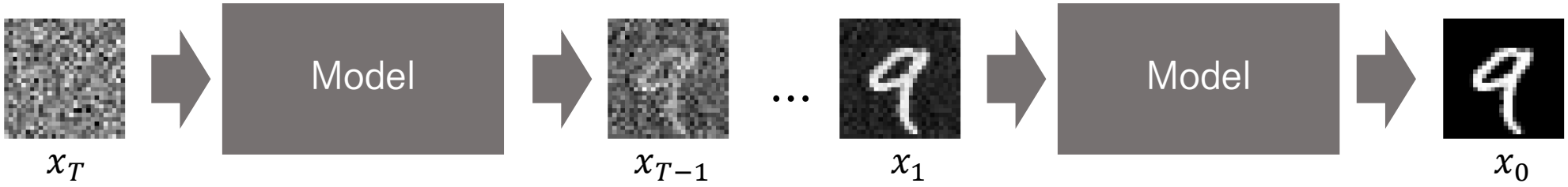
Diffusion Models

Algorithm 2 Sampling

1: $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
2: **for** $t = T, \dots, 1$ **do**
3: $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\mathbf{z} = \mathbf{0}$
4: $\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) \right) + \sigma_t \mathbf{z}$
5: **end for**
6: **return** $\mathbf{x}_0$

Denoising Diffusion Probabilistic Models (DDPM)

- Sampling: 각 시점 마다 노이즈 제거하며 데이터 생성
- Reverse process는 모든 timestep을 거침 → sampling speed 느림



Diffusion Models

Score-based Generative Models and Diffusion Models

2022년 2월 11일 오전 2:01 / 조회수: 4170

REFERENCES

Score-based Generative Models and Diffusion Models.pdf

INFORMATION

2022년 2월 11일 오후 1시 ~ 온라인 비디오 시청 (YouTube)

발표자:  조한샘

TOPIC

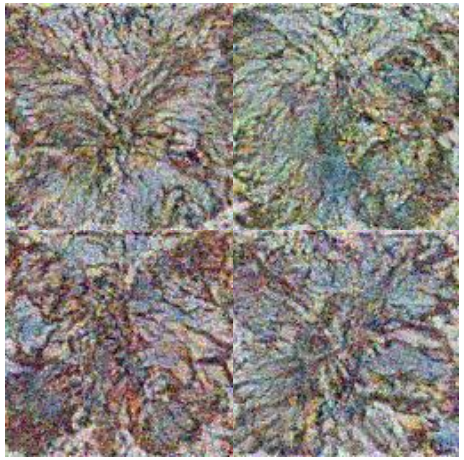
Score-based Generative Models and Diffusion Models

Improving Sampling Speed of Diffusion Models

Simple Idea

- Sampling step 중간을 뛰어 넘자 (Respacing)
- 기존: 1000 → 999 → 998 → … → 1 → 0 (T=1000)
- Speed↑: 1000 → 998 → 996 → … → 2 → 0 (T=500)

<T=1000>

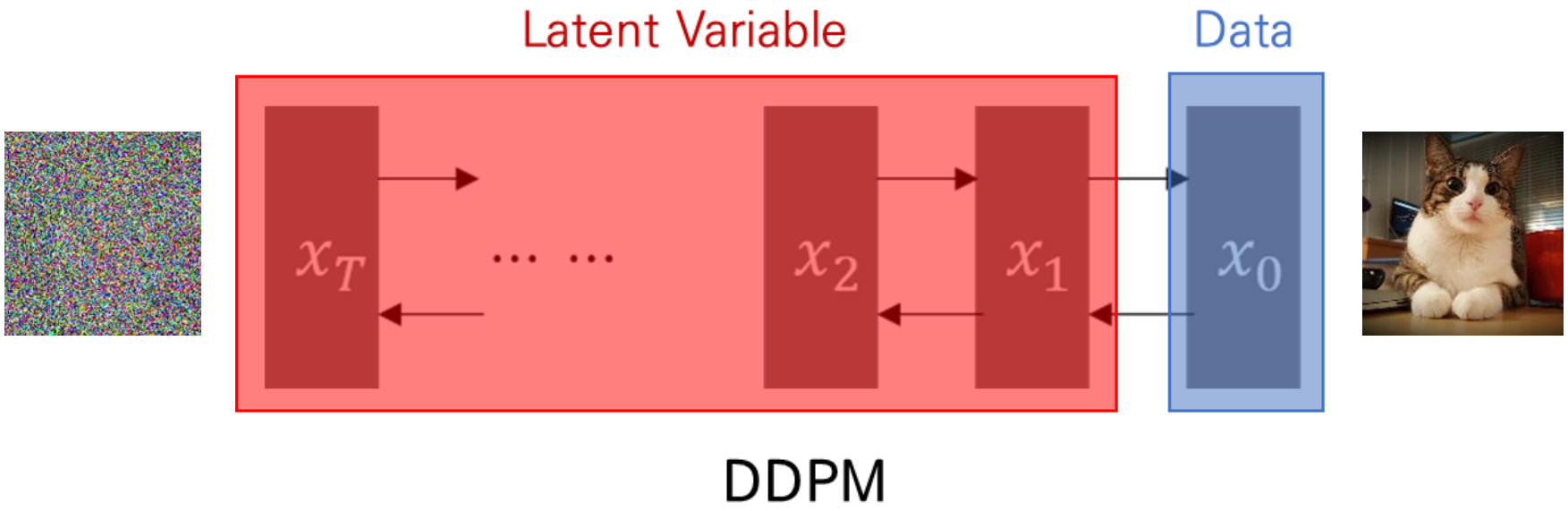


<T=500>

Improving Sampling Speed of Diffusion Models

Simple Idea

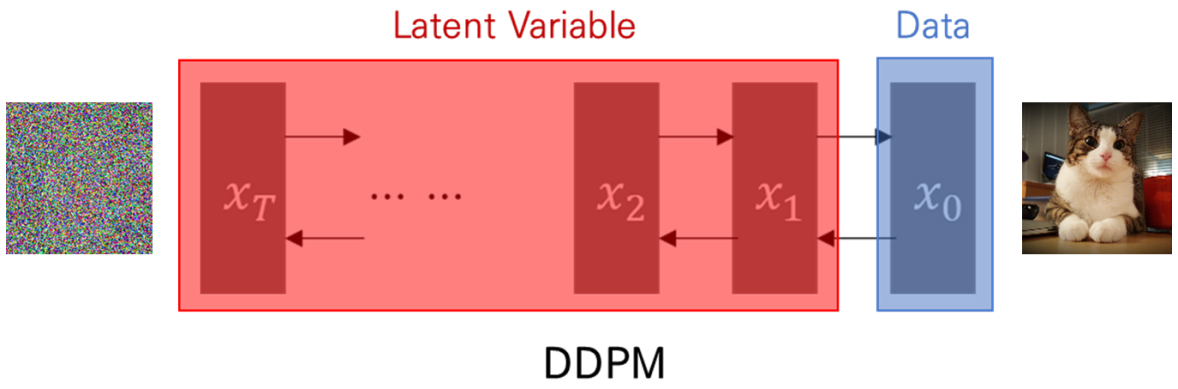
- DDPM은 forward와 reverse process를 **Markov chain**으로 정의됨
- Markov chain: 이전 시점의 변수에만 영향을 받는 random process
  - $P(X_{n+1}|X_n) = P(X_{n+1}|X_n, X_{n-1}, \dots, X_0)$
- Random process: 확률 변수들의 나열



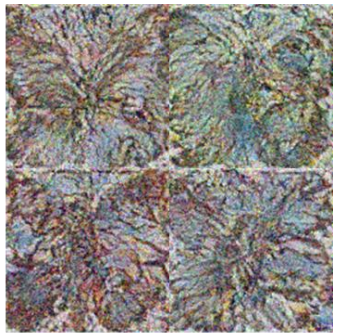
Improving Sampling Speed of Diffusion Models

Simple Idea

- 기존: 1000 → 999 → 998 → … → 1 → 0 (T=1000) / **Markov chain O**
- Speed↑: 1000 → 998 → 996 → … → 2 → 0 (T=500) / **Markov chain X**



<T=1000>



<T=500>

Improving Sampling Speed of Diffusion Models

Denoising Diffusion Implicit Models (DDIM)

- ICLR 2021
- 2023년 02월 10일 기준: 325회 인용

Published as a conference paper at ICLR 2021

DENOISING DIFFUSION IMPLICIT MODELS

Jiaming Song, Chenlin Meng & Stefano Ermon
Stanford University
{tsong,chenlin,ermon}@cs.stanford.edu

ABSTRACT

Denoising diffusion probabilistic models (DDPMs) have achieved high quality image generation without adversarial training, yet they require simulating a Markov chain for many steps in order to produce a sample. To accelerate sampling, we present denoising diffusion implicit models (DDIMs), a more efficient class of iterative implicit probabilistic models with the same training procedure as DDPMs. In DDPMs, the generative process is defined as the reverse of a particular Markovian diffusion process. We generalize DDPMs via a class of non-Markovian diffusion processes that lead to the same training objective. These non-Markovian processes can correspond to generative processes that are deterministic, giving rise to implicit models that produce high quality samples much faster. We empirically demonstrate that DDIMs can produce high quality samples $10\times$ to $50\times$ faster in terms of wall-clock time compared to DDPMs, allow us to trade off computation for sample quality, perform semantically meaningful image interpolation directly in the latent space, and reconstruct observations with very low error.

Improving Sampling Speed of Diffusion Models

Denoising Diffusion Implicit Models (DDIM)

- Non-Markovian으로 forward, reverse process 정의
- Forward: $q_\sigma(x_t|x_{t-1}, x_0) = \frac{q_\sigma(x_{t-1}|x_t,x_0)q_\sigma(x_t|x_0)}{q_\sigma(x_{t-1}|x_t,x_0)}$
- Reverse: $q_\sigma\left(x_{t-1}|x_t, f_\theta^t(x_t)\right)$
- x_0와 $f_\theta^t(x_t)$ 가 조건으로 주어짐 → **Non-Markovian**

- 새롭게 정의한 reverse process를 활용해 loss function을 유도 (DDIM Appendix B.)
  - DDPM의 loss function과 동일
  - **DDPM으로 학습된 모델 활용 가능**
  - **Non-Markovian reverse process를 통해 sampling 속도 향상**

Improving Sampling Speed of Diffusion Models

Denoising Diffusion Implicit Models (DDIM)

- Reverse: $q_\sigma(x_{t-1}|x_t, f_\theta^t(x_t))$
- $f_\theta^t(x_t)$: 노이즈 이미지x_t를 기반으로 깨끗한 이미지 x_0예측
- x_0와 x_t를 기반으로 x_{t-1}을 예측

  → non-Markovian이기 때문에 **sampling 속도 향상**
- σ_t를 0으로 설정함으로써 **deterministic**한 sampling 진행

Reverse Process $$x_{t-1} = \sqrt{\bar{\alpha}_{t-1}}\left(\frac{x_t - \sqrt{1-\bar{\alpha}_t}\cdot\epsilon_\theta^t(x_t)}{\sqrt{\bar{\alpha}_t}}\right) + \sqrt{1-\bar{\alpha}_{t-1}-\sigma_t^2}\cdot\epsilon_\theta^t(x_t) + \sigma_t\epsilon_t$$

$f_\theta^t(x_t) : x_t \rightarrow x_0$ 예측

Improving Sampling Speed of Diffusion Models

Denoising Diffusion Implicit Models (DDIM)

- Speed ↑ : 1000 → 990 → 980 → … → 10 → 0 (T=100)
- DDIM은 step수가 줄어들어도 좋은 성능을 유지

DDPM (1000 step)

DDIM (1000 step)

DDIM (100 step)

Improving Sampling Speed of Diffusion Models

Tackling the Generative Learning Trilemma with Denoising Diffusion GANs

- ICLR 2022
- 2023년 02월 10일 기준: 82회 인용

Published as a conference paper at ICLR 2022

TACKLING THE GENERATIVE LEARNING TRILEMMA WITH DENOISING DIFFUSION GANS

Zhisheng Xiao*
The University of Chicago
zxiao@uchicago.edu

Karsten Kreis
NVIDIA
kkreis@nvidia.com

Arash Vahdat
NVIDIA
avahdat@nvidia.com

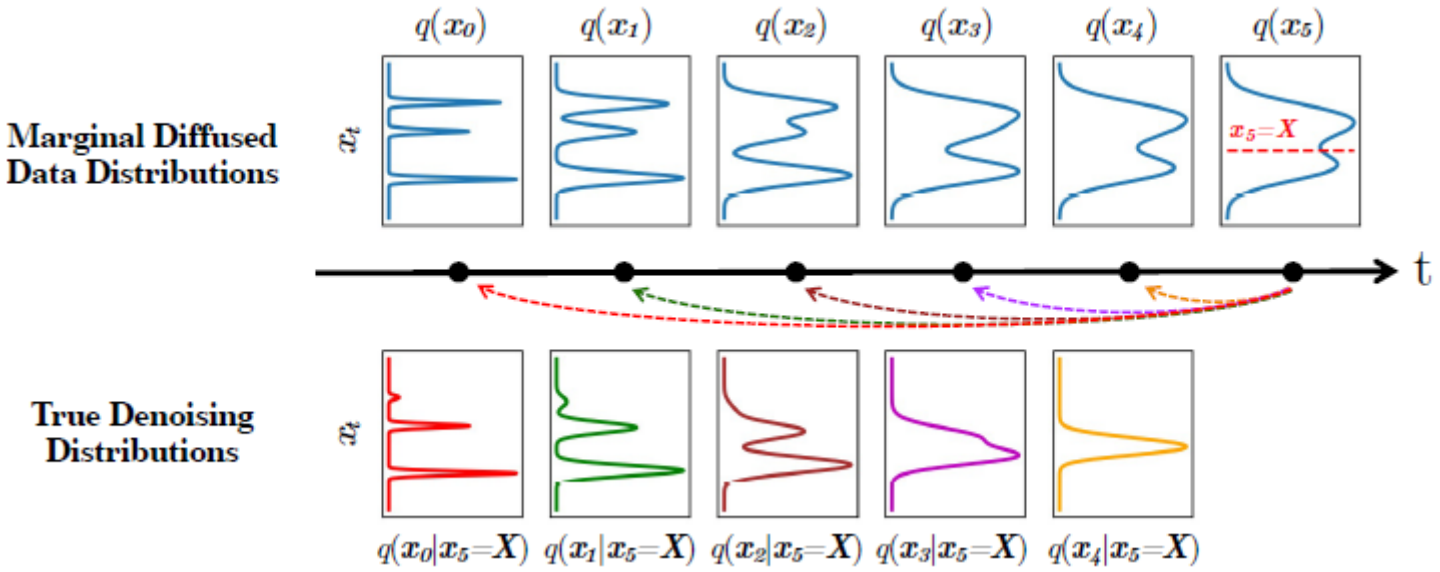
ABSTRACT

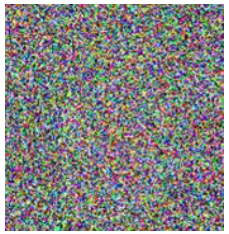
A wide variety of deep generative models has been developed in the past decade. Yet, these models often struggle with simultaneously addressing three key requirements including: high sample quality, mode coverage, and fast sampling. We call the challenge imposed by these requirements *the generative learning trilemma*, as the existing models often trade some of them for others. Particularly, denoising diffusion models have shown impressive sample quality and diversity, but their expensive sampling does not yet allow them to be applied in many real-world applications. In this paper, we argue that slow sampling in these models is fundamentally attributed to the Gaussian assumption in the denoising step which is justified only for small step sizes. To enable denoising with large steps, and hence, to reduce the total number of denoising steps, we propose to model the denoising distribution using a complex multimodal distribution. We introduce *denoising diffusion generative adversarial networks (denoising diffusion GANs)* that model each denoising step using a multimodal conditional GAN. Through extensive evaluations, we show that denoising diffusion GANs obtain sample quality and diversity competitive with original diffusion models while being 2000× faster on the CIFAR-10 dataset. Compared to traditional GANs, our model exhibits better mode coverage and sample diversity. To the best of our knowledge, denoising diffusion GAN is the first model that reduces sampling cost in diffusion models to an extent that allows them to be applied to real-world applications inexpensively. Project page and code: https://nvlabs.github.io/denoising-diffusion-gan.

Improving Sampling Speed of Diffusion Models

Denoising Diffusion GAN (DDGAN)

- DDPM의 reverse process는 Gaussian으로 정의
- Reverse process의 **step이 커지려면** reverse process는 **non-Gaussian multimodal**
- Gaussian이라는 가정이 깨져야 한다





Improving Sampling Speed of Diffusion Models

Denoising Diffusion GAN (DDGAN)

- DDPM: Markov chain, **Gaussian**
- DDIM: Non-Markovian, **Gaussian**

DDPM $p_\theta(x_{t-1}|x_t) = N(\mu_\theta(x_t,t),\sigma_t I)$ $where$ $\mu_\theta(x_t,t) = \frac{1}{\sqrt{\bar{\alpha}_t}}\left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(x_t,t)\right)$

DDIM $p_\theta(x_{t-1}|x_t) = q_\sigma\left(x_{t-1}\middle|x_t, f_\theta^t(x_t)\right) = N\left(\sqrt{\bar{\alpha}_{t-1}}f_\theta^t(x_t) + \sqrt{1-\bar{\alpha}_{t-1}-\sigma_t^2}\cdot\epsilon_\theta^t(x_t), \sigma_t I\right)$

Improving Sampling Speed of Diffusion Models

Denoising Diffusion GAN (DDGAN)

- DDGAN은 Generator를 통해서 x_0를 예측
- 동일한 x_t를 입력 받아도 서로 다른 x_0를 예측하도록 모델에 랜덤성 부여
- DDIM : $x_t \rightarrow x_0$ 예측이 **deterministic** / reverse process 고정 → **Gaussian**
- DDGAN : $x_t \rightarrow x_0$ 예측이 **stochastic** / reverse process 고정X → **Non-Gaussian**

DDIM

$$p_\theta(x_{t-1}|x_t) = q_\sigma\left(x_{t-1}|x_t, f_\theta^t(x_t)\right) = N\left(\sqrt{\bar{\alpha}_{t-1}} f_\theta^t(x_t) + \sqrt{1-\bar{\alpha}_{t-1}-\sigma_t^2}\cdot\epsilon_\theta^t(x_t), \sigma_t I\right)$$

DDGAN

$$p_\theta(x_{t-1}|x_t) = q_\sigma(x_{t-1}|x_t, G_\theta(x_t, z, t)) = N\left(\sqrt{\bar{\alpha}_{t-1}} G_\theta(x_t, z, t) + \sqrt{1-\bar{\alpha}_{t-1}-\sigma_t^2}\cdot\epsilon_\theta^t(x_t), \sigma_t I\right)$$

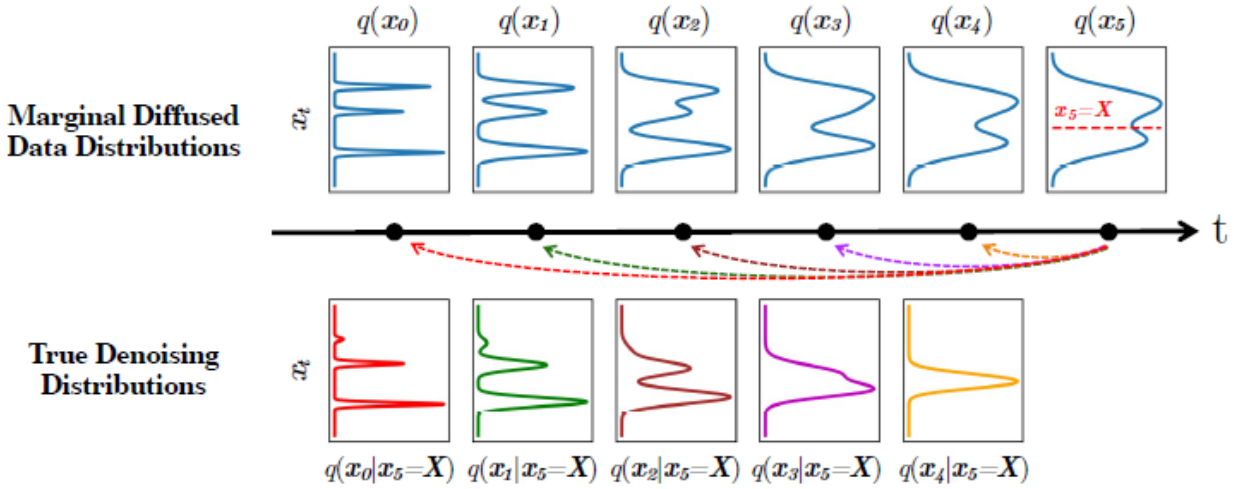
Improving Sampling Speed of Diffusion Models

Denoising Diffusion GAN (DDGAN)

큰 step size

→ non-Gaussian reverse process

→ Generator를 통해 $x_t \rightarrow x_0$ stochastic하게 예측 후 x_{t-1} 샘플링

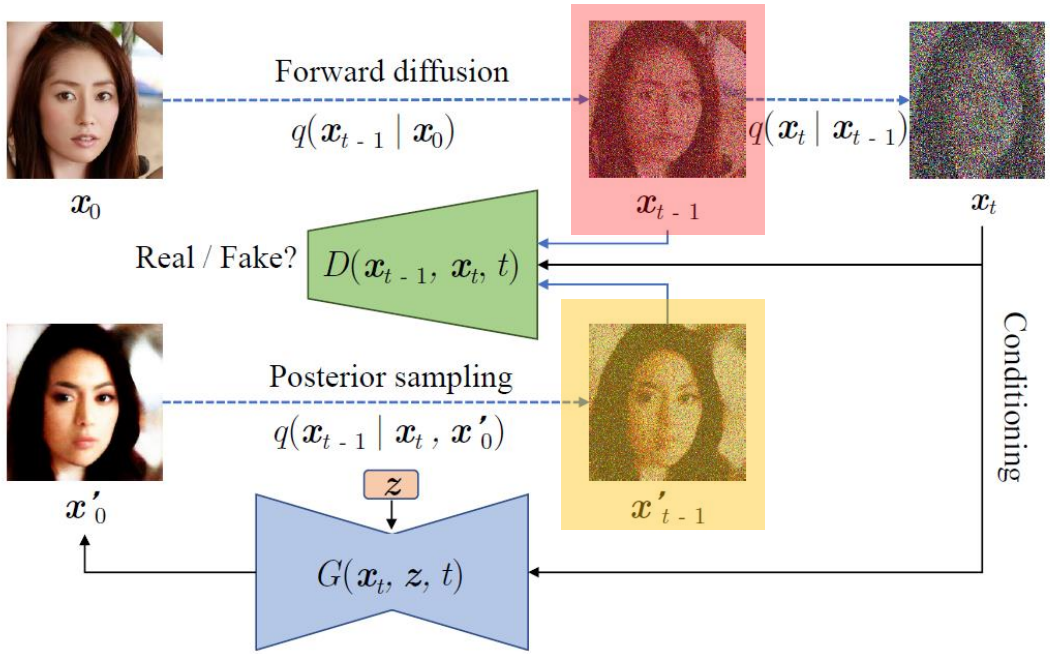


$$p_\theta(x_{t-1}|x_t) = q_\sigma(x_{t-1}|x_t, G_\theta(x_t, z, t)) = N\left(\sqrt{\bar{\alpha}_{t-1}} G_\theta(x_t, z, t) + \sqrt{1-\bar{\alpha}_{t-1}-\sigma_t^2}\cdot \epsilon_\theta^t(x_t), \sigma_t I\right)$$

Improving Sampling Speed of Diffusion Models

Denoising Diffusion GAN (DDGAN)

- x_{t-1} : 데이터에서 얻어진 노이즈 이미지
- x'_{t-1} : Generator가 생성한 노이즈 이미지
- Discriminator 진짜 가짜 구분 → Generator가 $x_t \rightarrow x_{t-1}$로 가는 reverse process 분포 학습



Improving Sampling Speed of Diffusion Models

Denoising Diffusion GAN (DDGAN)

- Experiments

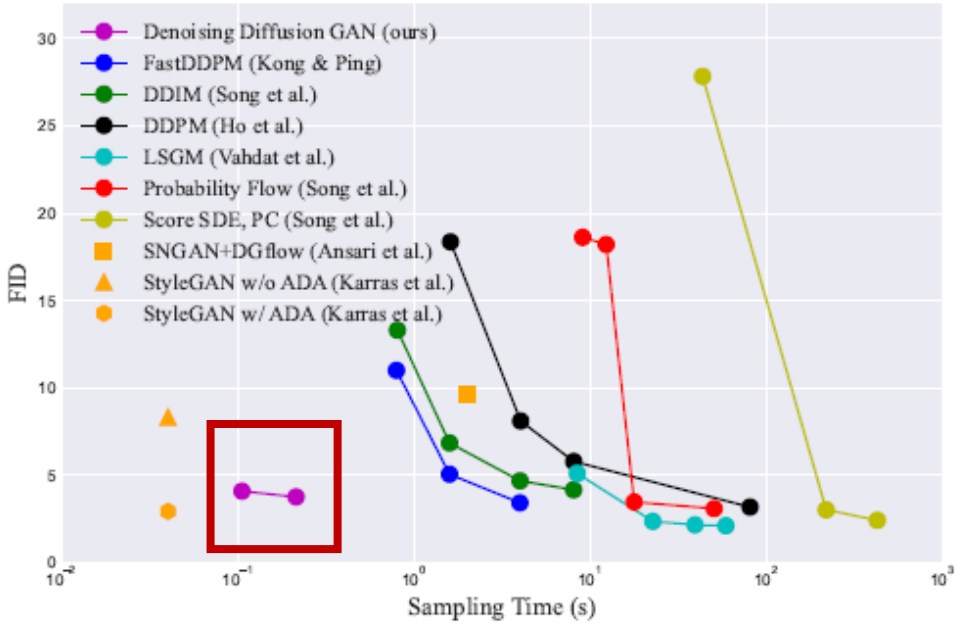


Figure 4: Sample quality vs sampling time trade-off.

Table 1: Results for unconditional generation on CIFAR-10.

Model	IS↑	FID↓	Recall↑	NFE ↓	Time (s) ↓
Denoising Diffusion GAN (ours), T=4	9.63	3.75	0.57	4	0.21
DDPM (Ho et al., 2020)	9.46	3.21	0.57	1000	80.5
NCSN (Song & Ermon, 2019)	8.87	25.3	-	1000	107.9
Adversarial DSM (Jolicoeur-Martineau et al., 2021b)	-	6.10	-	1000	-
Likelihood SDE (Song et al., 2021b)	-	2.87	-	-	-
Score SDE (VE) (Song et al., 2021c)	9.89	2.20	0.59	2000	423.2
Score SDE (VP) (Song et al., 2021c)	9.68	2.41	0.59	2000	421.5
Probability Flow (VP) (Song et al., 2021c)	9.83	3.08	0.57	140	50.9
LSGM (Vahdat et al., 2021)	9.87	2.10	0.61	147	44.5
DDIM, T=50 (Song et al., 2021a)	8.78	4.67	0.53	50	4.01
FastDDPM, T=50 (Kong & Ping, 2021)	8.98	3.41	0.56	50	4.01
Recovery EBM (Gao et al., 2021)	8.30	9.58	-	180	-
Improved DDPM (Nichol & Dhariwal, 2021)	-	2.90	-	4000	-
VDM (Kingma et al., 2021)	-	4.00	-	1000	-
UDM (Kim et al., 2021)	10.1	2.33	-	2000	-
D3PMs (Austin et al., 2021)	8.56	7.34	-	1000	-
Gotta Go Fast (Jolicoeur-Martineau et al., 2021a)	-	2.44	-	180	-
DDPM Distillation (Luhman & Luhman, 2021)	8.36	9.36	0.51	1	-
SNGAN (Miyato et al., 2018)	8.22	21.7	0.44	1	-
SNGAN+DGflow (Ansari et al., 2021)	9.35	9.62	0.48	25	1.98
AutoGAN (Gong et al., 2019)	8.60	12.4	0.46	1	-
TransGAN (Jiang et al., 2021)	9.02	9.26	-	1	-
StyleGAN2 w/o ADA (Karras et al., 2020a)	9.18	8.32	0.41	1	0.04
StyleGAN2 w/ ADA (Karras et al., 2020a)	9.83	2.92	0.49	1	0.04
StyleGAN2 w/ Diffaug (Zhao et al., 2020)	9.40	5.79	0.42	1	0.04
Glow (Kingma & Dhariwal, 2018)	3.92	48.9	-	1	-
PixelCNN (Oord et al., 2016b)	4.60	65.9	-	1024	-
NVAE (Vahdat & Kautz, 2020)	7.18	23.5	0.51	1	0.36
IGEBM (Du & Mordatch, 2019)	6.02	40.6	-	60	-
VAEBM (Xiao et al., 2021)	8.43	12.2	0.53	16	8.79

Improving Sampling Speed of Diffusion Models

Denoising Diffusion GAN (DDGAN)

- Experiments

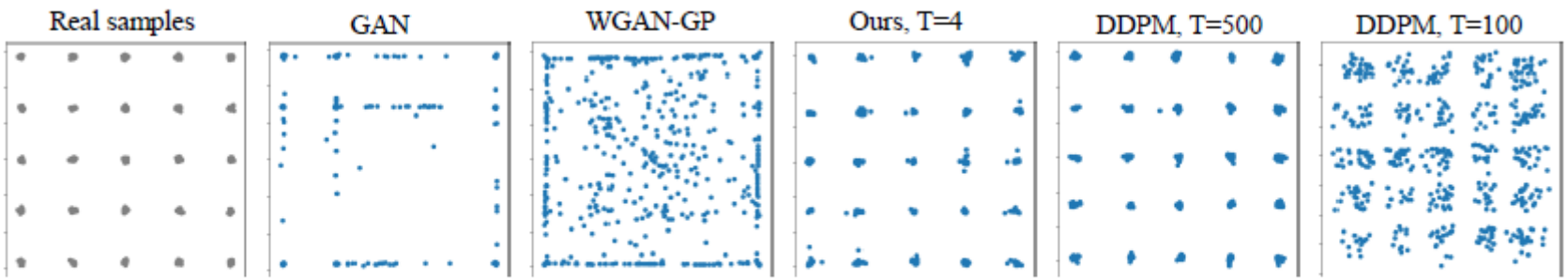


Figure 6: Qualitative results on the 25-Gaussians dataset.

Improving Sampling Speed of Diffusion Models

Progressive Distillation for Fast Sampling of Diffusion Models

- ICLR 2022
- 2023년 02월 10일 기준: 87회 인용

Published as a conference paper at ICLR 2022

PROGRESSIVE DISTILLATION FOR FAST SAMPLING OF DIFFUSION MODELS

Tim Salimans & Jonathan Ho
Google Research, Brain team
{salimans,jonathanho}@google.com

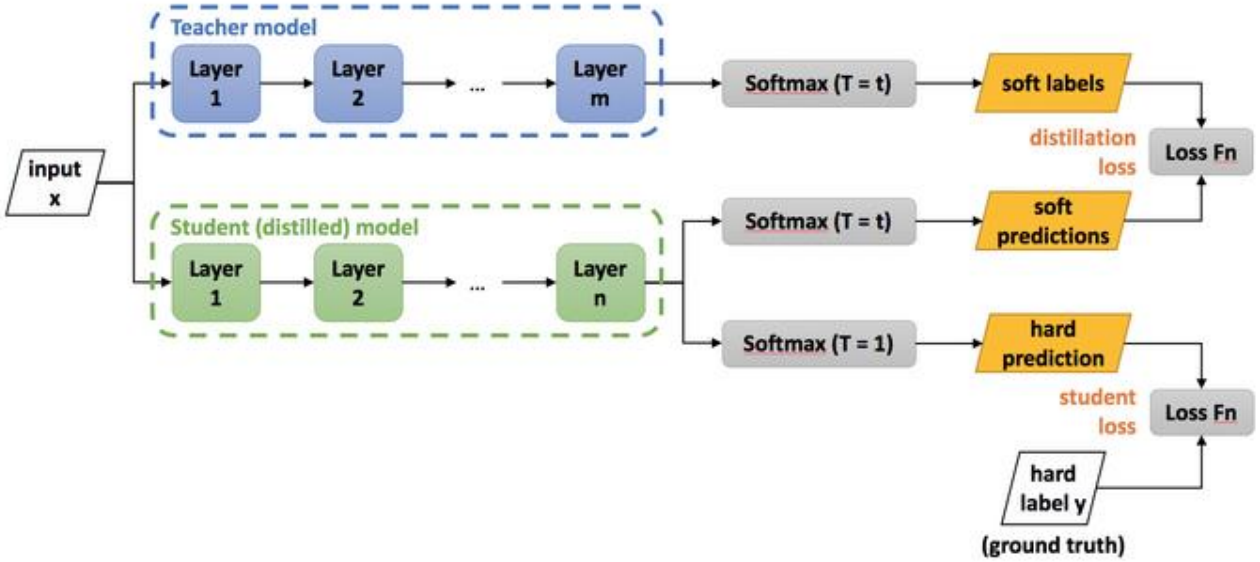
ABSTRACT

Diffusion models have recently shown great promise for generative modeling, outperforming GANs on perceptual quality and autoregressive models at density estimation. A remaining downside is their slow sampling time: generating high quality samples takes many hundreds or thousands of model evaluations. Here we make two contributions to help eliminate this downside: First, we present new parameterizations of diffusion models that provide increased stability when using few sampling steps. Second, we present a method to distill a trained deterministic diffusion sampler, using many steps, into a new diffusion model that takes half as many sampling steps. We then keep progressively applying this distillation procedure to our model, halving the number of required sampling steps each time. On standard image generation benchmarks like CIFAR-10, ImageNet, and LSUN, we start out with state-of-the-art samplers taking as many as 8192 steps, and are able to distill down to models taking as few as 4 steps without losing much perceptual quality; achieving, for example, a FID of 3.0 on CIFAR-10 in 4 steps. Finally, we show that the full progressive distillation procedure does not take more time than it takes to train the original model, thus representing an efficient solution for generative modeling using diffusion at both train and test time.

Improving Sampling Speed of Diffusion Models

Progressive Distillation for Fast Sampling of Diffusion Models

- Teacher model: 파라미터 수가 많고 task에 대해서 높은 성능을 내는 모델
- Student model: 파라미터 수가 적고 teacher model의 지식을 전달받는 모델
- Student model의 출력이 teacher model의 출력과 유사해지면서 task를 수행하도록 학습 진행

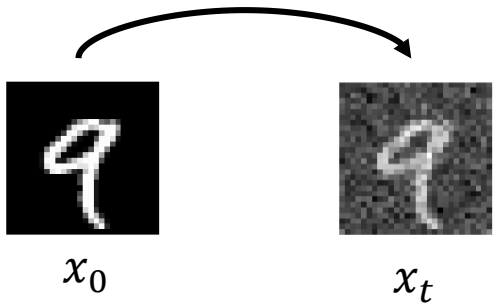


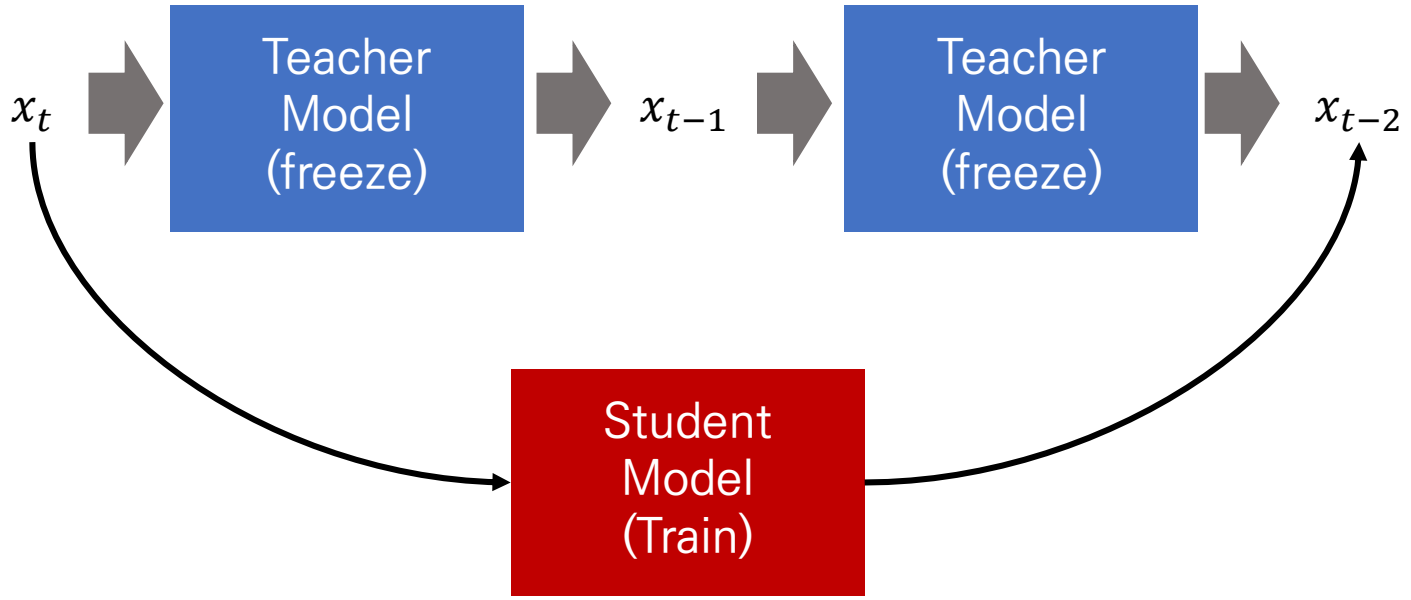
https://intellabs.github.io/distiller/knowledge_distillation.html

Improving Sampling Speed of Diffusion Models

Progressive Distillation for Fast Sampling of Diffusion Models

- Teacher model: 학습된 diffusion model
- Student model: Teacher 모델로 부터 출발해서 학습되는 모델
- Teacher model의 2step reverse process를 student 모델이 예측

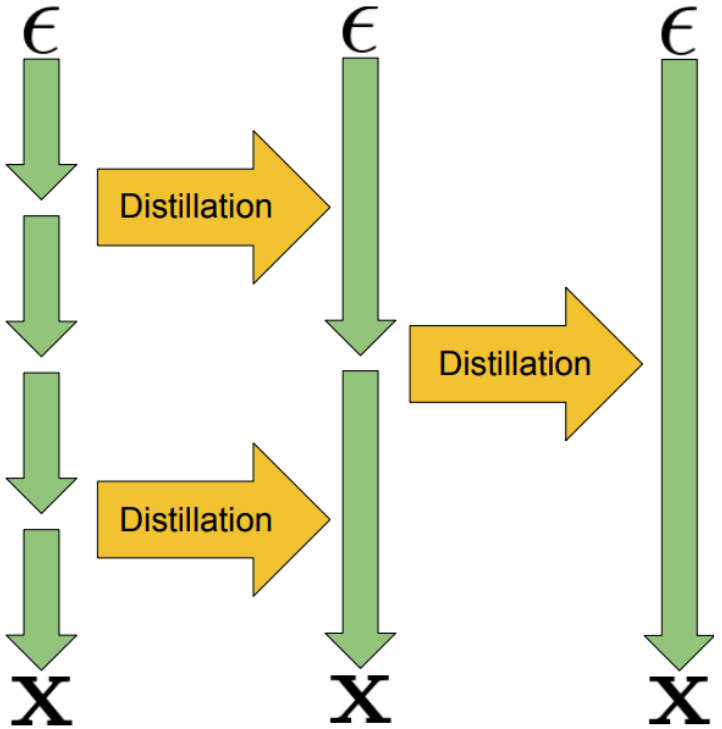




Improving Sampling Speed of Diffusion Models

Progressive Distillation for Fast Sampling of Diffusion Models

- 한번의 distillation을 통해 reverse process의 timestep 절반으로 감소
- 여러 번의 distillation을 통해 원하는 timestep까지 감소 가능



Improving Sampling Speed of Diffusion Models

Progressive Distillation for Fast Sampling of Diffusion Models

- 작은 timestep으로 좋은 sample quality 유지

(a) 256 sampling steps (b) 4 sampling steps (c) 1 sampling step

Improving Sampling Speed of Diffusion Models

On Distillation of Guided Diffusion Models

- NeurIPS 2022 Workshop
- 2023년 02월 10일 기준: 8회 인용
- Progressive distillation + **Classifier-free guidance**

On Distillation of Guided Diffusion Models

Chenlin Meng*
Stanford University
chenlin@cs.stanford.edu

Robin Rombach
Stability AI & LMU Munich
robin@stability.ai

Ruiqi Gao
Google Research, Brain Team
ruiqig@google.com

Diederik P. Kingma
Google Research, Brain Team
durk@google.com

Stefano Ermon
Stanford University
ermon@cs.stanford.edu

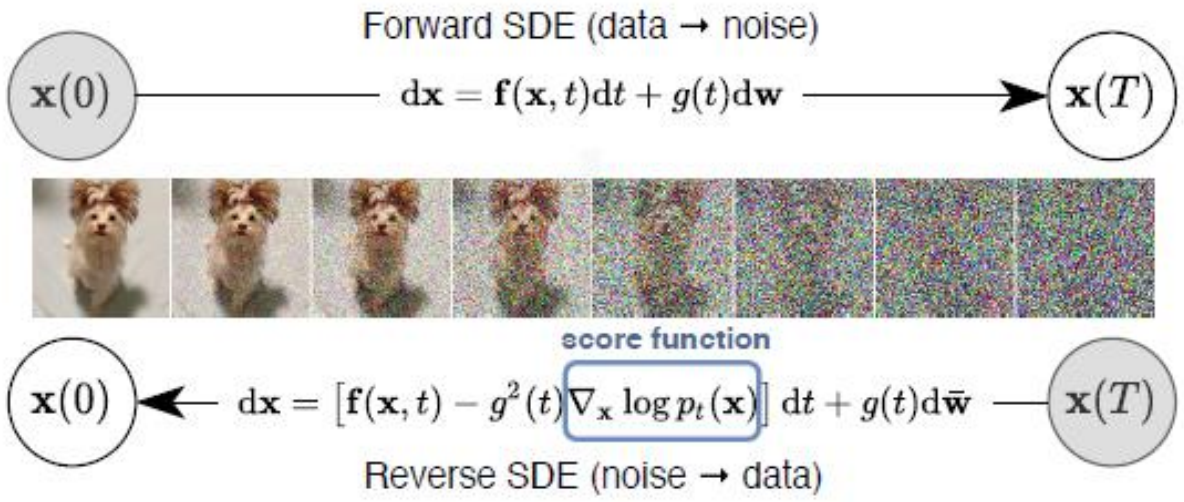
Jonathan Ho
Google Research, Brain Team
jonathanho@google.com

Tim Salimans
Google Research, Brain Team
salimans@google.com

Improving Sampling Speed of Diffusion Models

Score-based Generative Models through SDE

- DDPM의 timestep을 continuous 하게 확장
- 학습된 **unconditional diffusion model과 classifier**가 있으면 **conditional generation** 가능



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

$$logp_t(x|y) = logp_t(y|x) + logp_t(x) - logp_t(y)$$

$$\nabla_x logp_t(x|y) = \nabla_x logp_t(y|x) + \nabla_x logp_t(x)$$

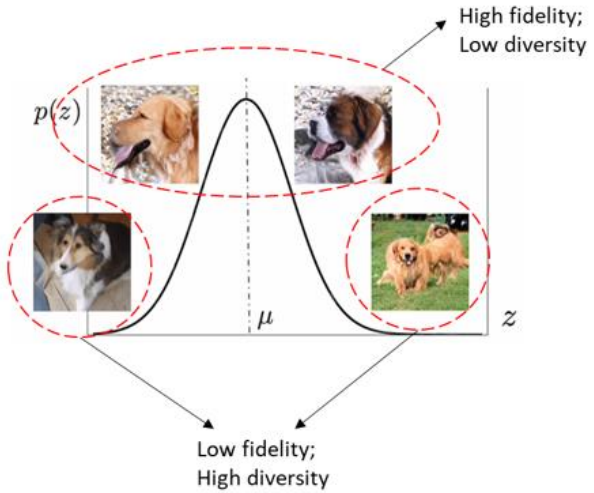
Classifier gradient

Unconditional Model output

Improving Sampling Speed of Diffusion Models

Classifier Guidance

- Classifier guidance: 생성되는 이미지의 diversity과 sample quality 사이 trade-off 관계를 조절 (truncation trick in GANs)
- Condition 쪽에 큰 가중치를 주면 좋은 퀄리티의 sample 생성 가능
- $\nabla_x log p_t(x|y) = \nabla_x log p_t(x) + \nabla_x log p_t(y|x)$
- $\nabla_x log p_t(x|y) = \nabla_x log p_t(x) + s \cdot \nabla_x log p_t(y|x)$



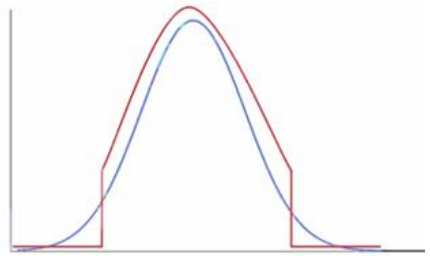


Figure 3: Samples from an unconditional diffusion model with classifier guidance to condition on the class "Pembroke Welsh corgi". Using classifier scale 1.0 (left; FID: 33.0) does not produce convincing samples in this class, whereas classifier scale 10.0 (right; FID: 12.0) produces much more class-consistent images.

Improving Sampling Speed of Diffusion Models

Classifier-free Guidance

- Classifier-free guidance: Classifier 없이 guidance를 적용하자
- Conditional 모델과 unconditional 모델을 활용해 classifier guidance 근사
- $\nabla_x log p_t(x|y) = (1+w)\nabla_x log p_t(x|y) - w \cdot \nabla_x log p_t(x|\phi)$
- W=0 → conditional model
- W↑ → diversity↓, fidelity↑
- W↓ → diversity↑, fidelity↓

Algorithm 1 Joint training a diffusion model with classifier-free guidance

Require: p_{uncond}: probability of unconditional training

1: **repeat**
2: $(\mathbf{x}, \mathbf{c}) \sim p(\mathbf{x}, \mathbf{c})$ ▷ Sample data with conditioning from the dataset
3: $\mathbf{c} \leftarrow \varnothing$ with probability p_{uncond} ▷ Randomly discard conditioning to train unconditionally
4: $\lambda \sim p(\lambda)$ ▷ Sample log SNR value
5: $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
6: $\mathbf{z}_\lambda = \alpha_\lambda \mathbf{x} + \sigma_\lambda \epsilon$ ▷ Corrupt data to the sampled log SNR value
7: Take gradient step on $\nabla_\theta \left\| \epsilon_\theta(\mathbf{z}_\lambda, \mathbf{c}) - \epsilon \right\|^2$ ▷ Optimization of denoising model
8: **until** converged

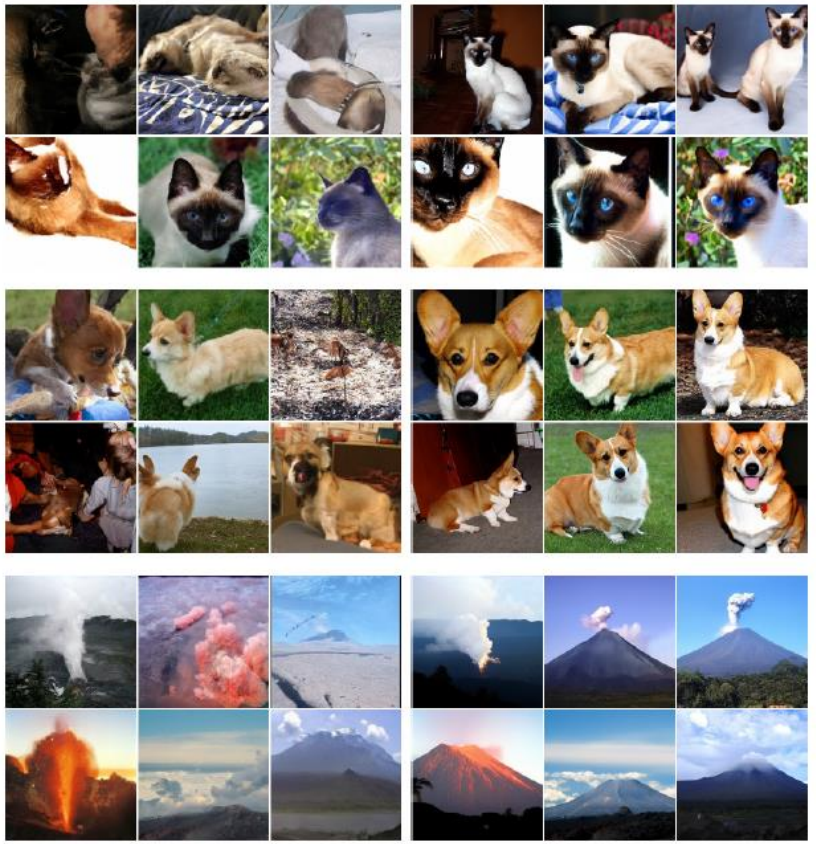


Figure 3: Classifier-free guidance on 128x128 ImageNet. Left: non-guided samples, right: classifier-free guided samples with $w = 3.0$. Interestingly, strongly guided samples such as these display saturated colors. See Fig. 8 for more.

Improving Sampling Speed of Diffusion Models

On Distillation of Guided Diffusion Models

- Classifier-free guidance를 적용하기

  → Conditional model과 unconditional model 모두에 대해서 forward 필요

  → Sampling time*2

- Progressive distillation을 적용해 classifier-free guidance의 sampling speed 개선

Improving Sampling Speed of Diffusion Models

On Distillation of Guided Diffusion Models

- Step1: Classifier-free guidance의 output에 대한 distillation 진행

  → 하나의 모델로 classifier-free guidance output 예측

Algorithm 1 Step-one distillation

Require: Trained classifier-free guidance teacher model $[\hat{\mathbf{x}}_{c,\boldsymbol{\theta}}, \hat{\mathbf{x}}_{\boldsymbol{\theta}}]$
Require: Data set $\mathcal{D}$
Require: Loss weight function $\omega()$
while not converged **do**

$\mathbf{x} \sim \mathcal{D}$	▷ Sample data
$t \sim U[0, 1]$	▷ Sample time
$w \sim U[w_{\min}, w_{\max}]$	▷ Sample guidance
$\epsilon \sim N(0, I)$	▷ Sample noise
$\mathbf{z}_t = \alpha_t \mathbf{x} + \sigma_t \epsilon$	▷ Add noise to data
$\lambda_t = \log[\alpha_t^2 / \sigma_t^2]$	▷ log-SNR
$\hat{\mathbf{x}}^w_{\boldsymbol{\theta}}(\mathbf{z}_t) = (1 + w)\hat{\mathbf{x}}_{c,\boldsymbol{\theta}}(\mathbf{z}_t) - w\hat{\mathbf{x}}_{\boldsymbol{\theta}}(\mathbf{z}_t)$	▷ Compute target
$L_{\boldsymbol{\eta}_1} = \omega(\lambda_t)\|\hat{\mathbf{x}}^w_{\boldsymbol{\theta}}(\mathbf{z}_t) - \hat{\mathbf{x}}_{\boldsymbol{\eta}_1}(\mathbf{z}_t, w)\|_2^2$	▷ Loss
$\boldsymbol{\eta}_1 \leftarrow \boldsymbol{\eta}_1 - \gamma \nabla_{\boldsymbol{\eta}_1} L_{\boldsymbol{\eta}_1}$	▷ Optimization

end while

Classifier-free guidance (teacher)

Student

Improving Sampling Speed of Diffusion Models

On Distillation of Guided Diffusion Models

- Step2: Progressive distillation 진행 → sampling step 줄이기

Algorithm 2 Step-two distillation for deterministic sampler

Require: Trained teacher model $\hat{\mathbf{x}}_{\boldsymbol{\eta}}(\mathbf{z}_t, w)$
Require: Data set $\mathcal{D}$
Require: Loss weight function $\omega()$
Require: Student sampling steps N
  for K iterations **do**
    $\boldsymbol{\eta}_2 \leftarrow \boldsymbol{\eta}$ ▷ Init student from teacher
    while not converged **do**
      $\mathbf{x} \sim \mathcal{D}$
      $t = i/N, \;\; i \sim Cat[1, 2, \ldots, N]$
      $w \sim U[w_{\text{min}}, w_{\text{max}}]$ ▷ Sample guidance
      $\epsilon \sim N(0, I)$
      $\mathbf{z}_t = \alpha_t \mathbf{x} + \sigma_t \epsilon$
      `# 2 steps of DDIM with teacher`
      $t' = t - 0.5/N, \quad t'' = t - 1/N$
      $\mathbf{z}^w_{t'} = \alpha_{t'} \hat{\mathbf{x}}_{\boldsymbol{\eta}}(\mathbf{z}_t, w) + \frac{\sigma_{t'}}{\sigma_t}(\mathbf{z}_t - \alpha_t \hat{\mathbf{x}}_{\boldsymbol{\eta}}(\mathbf{z}_t, w))$
      $\mathbf{z}^w_{t''} = \alpha_{t''} \hat{\mathbf{x}}_{\boldsymbol{\eta}}(\mathbf{z}^w_{t'}, w) + \frac{\sigma_{t''}}{\sigma_{t'}}(\mathbf{z}^w_{t'} - \alpha_{t'} \hat{\mathbf{x}}_{\boldsymbol{\eta}}(\mathbf{z}^w_{t'}, w))$
      $\tilde{\mathbf{x}}^w = \frac{\mathbf{z}^w_{t''} - (\sigma_{t''}/\sigma_t)\mathbf{z}_t}{\alpha_{t''} - (\sigma_{t''}/\sigma_t)\alpha_t}$ ▷ Teacher $\hat{\mathbf{x}}$ target
      $\lambda_t = \log[\alpha_t^2/\sigma_t^2]$
      $L_{\boldsymbol{\eta}_2} = \omega(\lambda_t) \| \tilde{\mathbf{x}}^w - \hat{\mathbf{x}}_{\boldsymbol{\eta}_2}(\mathbf{z}_t, w) \|_2^2$
      $\boldsymbol{\eta}_2 \leftarrow \boldsymbol{\eta}_2 - \gamma \nabla_{\boldsymbol{\eta}_2} L_{\boldsymbol{\eta}_2}$
    end while
    $\boldsymbol{\eta} \leftarrow \boldsymbol{\eta}_2$ ▷ Student becomes next teacher
    $N \leftarrow N/2$ ▷ Halve number of sampling steps
  end for

Step1 student model (teacher)

Student

Improving Sampling Speed of Diffusion Models

On Distillation of Guided Diffusion Models

- Samples

Improving Sampling Speed of Diffusion Models

On Distillation of Guided Diffusion Models

- Samples

Input Mask Result 1 Result 2

Image inpainting (2 steps)

Input Output (different styles)

Image to image translation (3 steps)

Conclusion

Improving Sampling Speed of Diffusion Models

- **DDIM**

  DDPM의 reverse process는 Markovian → step size를 키우기 어려움

  Non-Markovian forward, reverse process를 활용해 sampling 속도 개선

- **DDGAN**

  Reverse process에서 step 사이즈가 커지기 위해서는 non-Gaussian 분포를 따라야 한다

  GAN을 활용해 $x_t \rightarrow x_{t-1}$로 가는 non-Gaussian reverse process 분포 학습

- **Progressive Distillation**

  Teacher model: 학습된 diffusion model

  Student model: timestep이 절반으로 감소한 diffusion model

  지속적인 distillation을 통해 timestep 감소 가능