2022 하계 세미나

Diffusion Models

Sogang University
Vision & Display Systems Lab, Dept. of Electronic Engineering

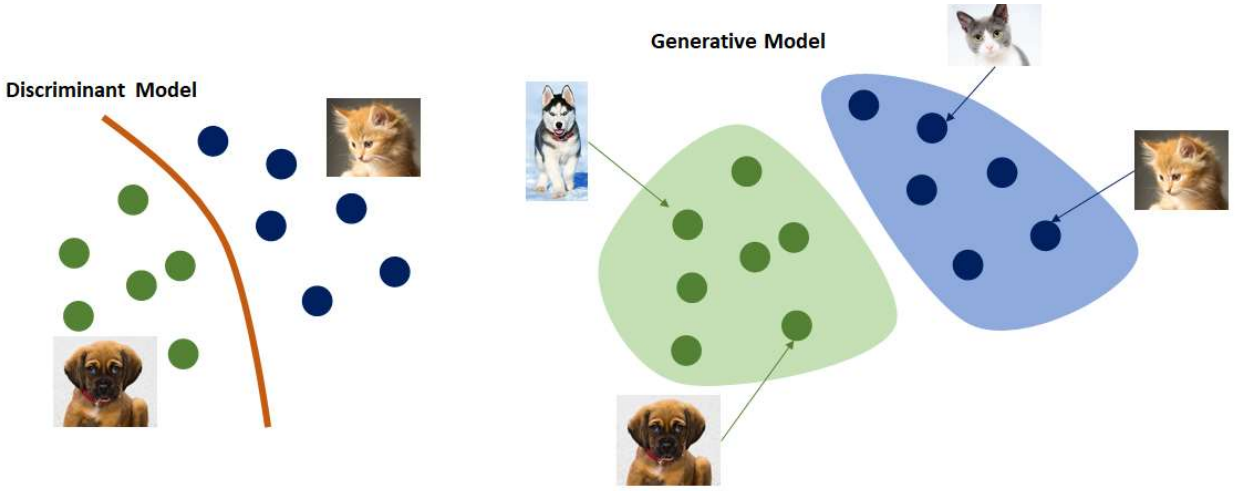
Presented by
이정훈

Outline

- Background
  - Discriminative vs. Generative model
  - Diffusion model
- Diffusion model
  - Diffusion Models beat GANs on Image Synthesis (NIPS 2021)
- Conclusion

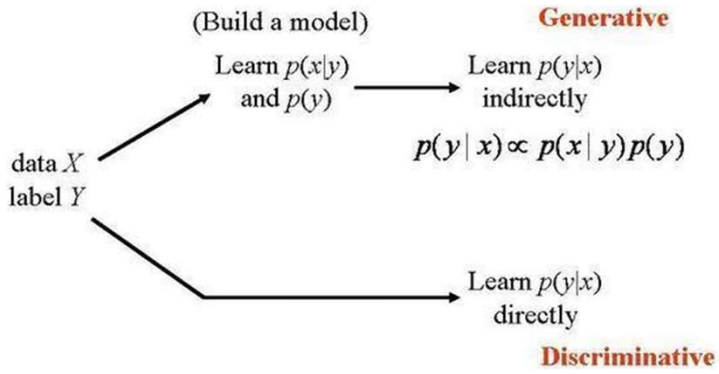
Background

- Discriminative vs. Generative
  - Discriminative model
    - 데이터 X가 주어졌을 때 레이블 Y가 나타날 조건부확률 p(Y|X)를 직접적으로 반환하는 모델
      - X의 레이블을 잘 구분하는 decision boundary를 학습하는 것이 목표임
      - 레이블 정보가 있어야 하기 때문에 지도학습 범주에 속함
  - Generative model
    - 데이터 X가 생성되는 과정을 두 개의 확률모형 p(Y), p(X|Y)으로 정의하고, Bayes Rule을 사용해 p(Y|X)를 간접적으로 도출하는 모델
      - 범주의 분포(distribution)를 학습하는 것이 목표이며, p(X|Y)을 구축하기 때문에 이 모델을 활용해 X를 샘플링 할 수 있음

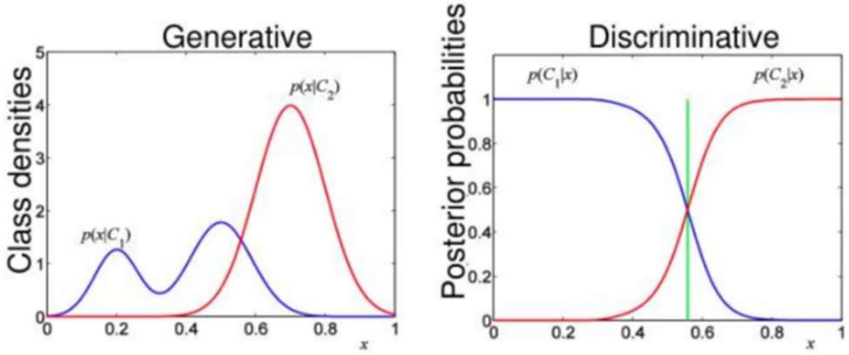


Background

- Discriminative vs. Generative
  - Generative model은 사후확률(Posteriori Probability)을 간접적으로 도출하고, Discriminative model은 직접적으로 도출

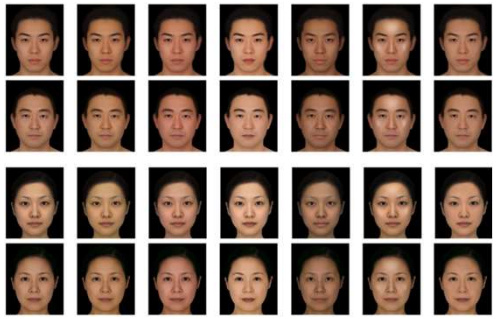


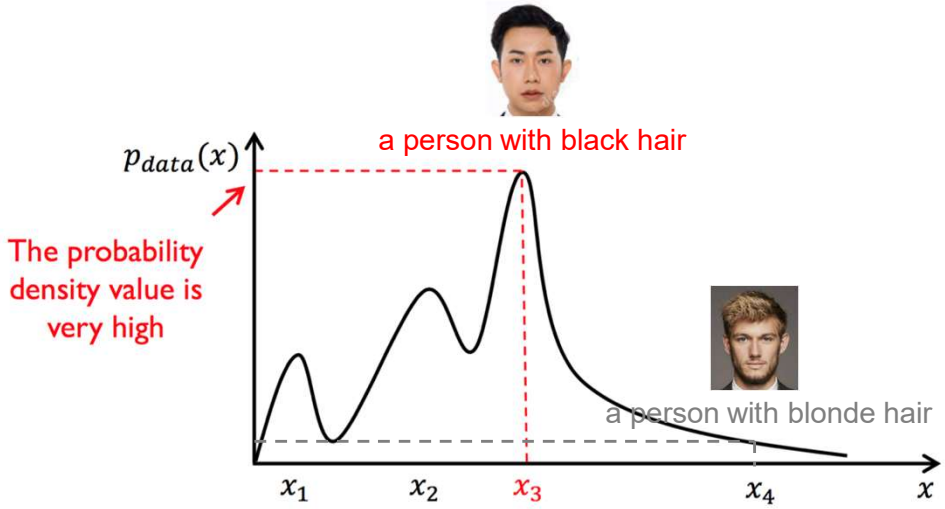
  - Generative model은 데이터 범주의 분포(distribution)를 학습하고, Discriminative model은 결정경계(Decision Boundary)를 학습



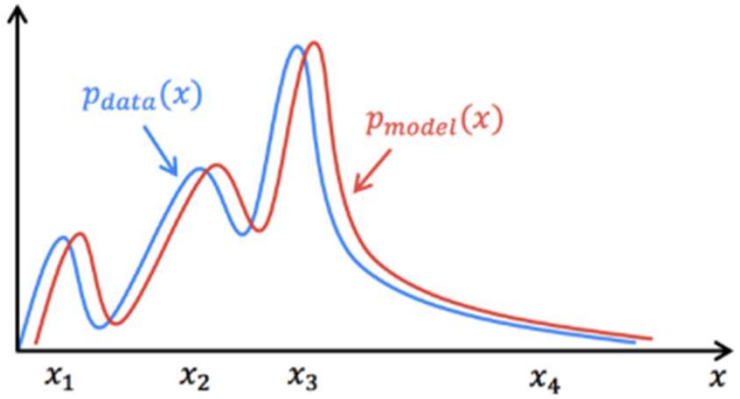
Background

- Generative Model
  - Data distribution
    - Dataset : Asian face (mostly black hair)





  - Generative model의 목적은 데이터 distribution을 학습하는 것
    - 구축하고자 하는 모델에 데이터를 넣으면 실제 데이터의 확률에 가깝게 반환하도록 학습



Background

- Generative Model
  - 주어진 데이터 x가 샘플링 된 분포(distribution)를 평가(estimate)하는 모델
    - 모델의 output이 distribution을 결정 짓는 변수이거나, 주어진 데이터가 나올 확률
    - Model을 이용해서 주어진 test data의 likelihood 값을 계산하거나, 새로운 data 생성 가능 (학습한 data의 distribution을 따르는 새로운 data를 만들어내는 모델)
  - Generative model의 성능은 주어진 데이터 x의 likelihood p(x) 값을 얼마나 잘 예측하는지, 생성된 데이터 x의 quality가 얼마나 우수한지로 판단

- Generative Model 비교
  - Auto-regressive models
  - VAEs
  - Energy Based Models
  - GANs
  - Normalizing Flows
  - Diffusion

Background

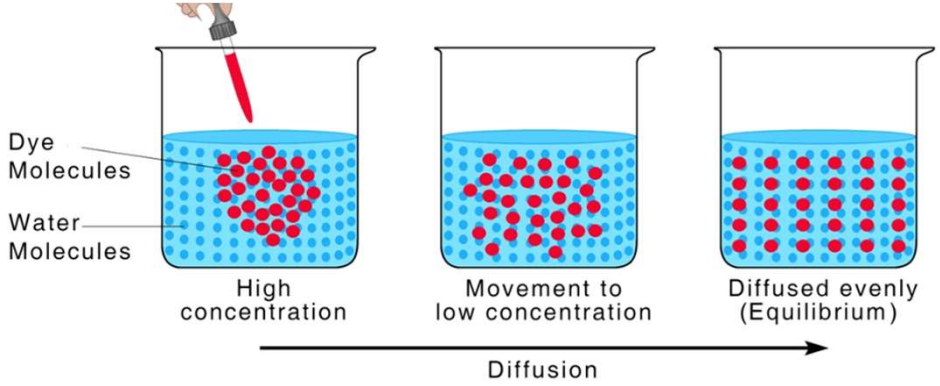
- Generative Model 비교
  - VAEs
    - 데이터의 확률분포 학습을 위해 encoder를 통해 관측된 데이터 x를 받아 잠재변수 z를 만들고, 임의의 z 값을 decoder에 넣어 이를 복원 또는 다양한 데이터 생성 가능
  - GANs
    - Discriminator와 Generator를 서로 adversarial 하게 학습시켜서 데이터를 생성하는 모델
  - Normalizing Flows
    - Simple한 분포 p(z)에서 복잡한 데이터 분포 p(x)로 가는 invertible mapping 함수를 이용해 distribution을 estimate하는 모델
  - Diffusion
    - 데이터 x에서 점점 noise를 추가해서 noise data로 만들고, noise data에서 데이터 x로 돌아오는 과정을 학습해 distribution을 estimate하는 모델

✓ 생성 모델을 통해 간단한 분포(z)를 특정한 패턴을 갖는 분포로 변환

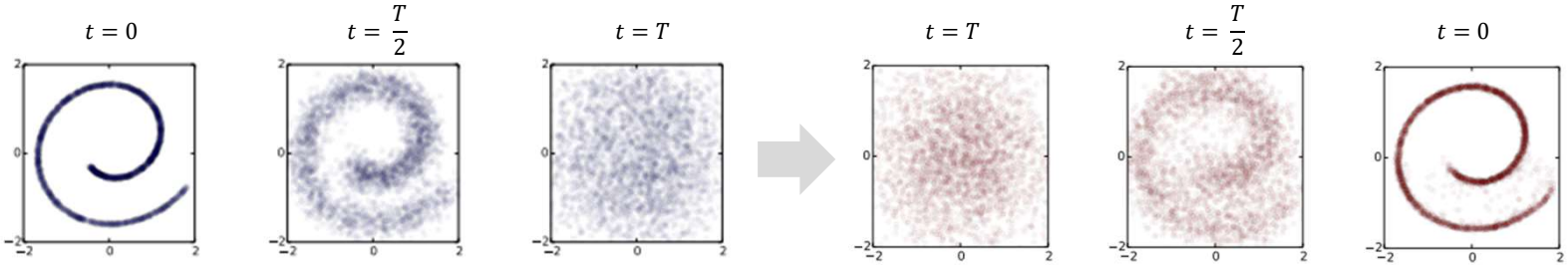
✓ 주어진 입력 데이터로부터 latent variable(z)를 얻고, 이를 변환하는 역량을 학습

Background

- What is diffusion
  - Diffusion is the net movement of anything generally from a region of higher concentration to a region of lower concentration [Wikipedia]



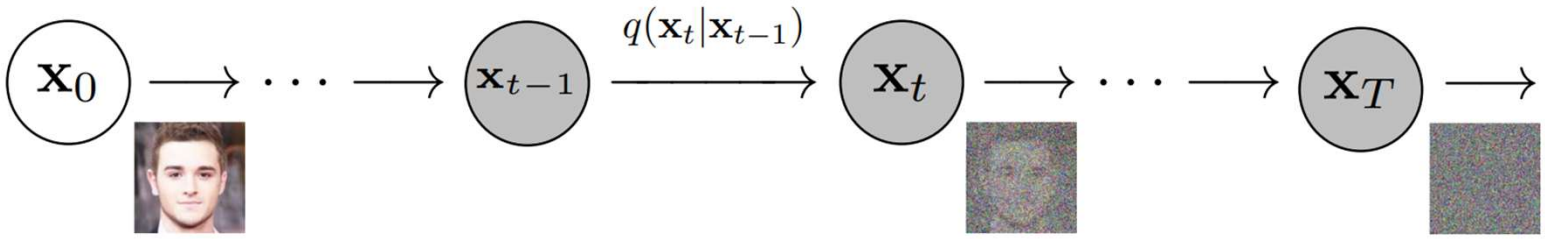
  - Diffusion model은 Thermodynamics에서 아이디어를 착안, 본 논문[1]에서 처음 제안됨
    - The data distribution undergoes Gaussian diffusion, which gradually transforms it into Gaussian noise → undergoes a Gaussian diffusion process, and is transformed back into the data distribution



Background

- Diffusion Model
  - 주어진 데이터 x에 gaussian noise를 점점 추가하고 (diffusion process), noise data에서 noise를 제거하여 이를 다시 복원하는 조건부 PDF (reverse process)를 학습
  - Diffusion process : 데이터(x_0) + 노이즈 → 랜덤 노이즈(x_T)
    - 주어진 데이터 x에 time step마다 noise를 추가하는 것으로, X_0에서 noise를 추가하면서 X_T로 가고 각 step에서 noise를 추가
      - Size of gaussian noise, β_t, is pre-defined, and noise gets bigger over the time-step

$$q(X_t \mid X_{t-1}) = \mathrm{N}(X_t ; \mu_{X_{t-1}}, \Sigma_{X_{t-1}}) = \mathrm{N}(X_t ; \sqrt{1-\beta_t}\, X_{t-1}, \beta_t)$$

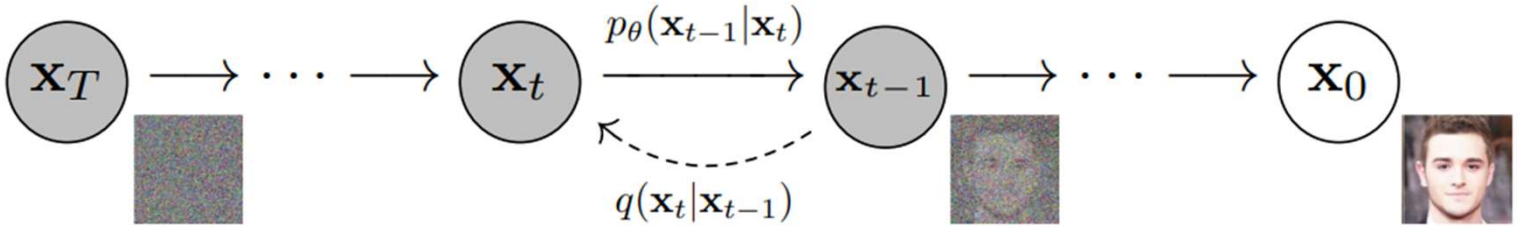


isotropic gaussian distribution

Background

- Diffusion Model
  - Diffusion process는 사전에 정의한 noise 크기(β)에 의해 모평균과 분포를 정의
  - Reverse process : 랜덤 노이즈(x_T) + denoising → 데이터(x_0)
    - 학습된 데이터의 패턴 복원(denoising)을 통해 원래의 데이터 x로 돌아오는 과정
    - 모델을 통해 reverse diffusion process를 표현하는 distribution을 학습
      - 각 t 시점의 조건부 gaussian 분포의 평균(μ_t)과 분산(Σ_t) 학습 필요

$$p_\theta(X_{0:T}) = p(X_T) \prod_{t=1}^{T} q(X_{t-1} \mid X_t), \quad p_\theta(X_{t-1} \mid X_t) = \mathrm{N}(X_{t-1}\,;\, \underbrace{\mu_\theta(X_t, t), \Sigma_\theta(X_t, t)}_{\text{학습 대상}})$$



✓ 모델이 예측한 평균/분산이 diffusion process에서 정의한 각 t 시점의 β(noise) 기반의 평균/분산과 유사해지도록 objective function을 구성

Background

- Diffusion Model
  - Loss
    - Diffusion model의 목적은 데이터의 분포 예측이고, 이는 주어진 데이터 x의 likelihood를 잘 계산하는 것으로 판단
      - 목적식은 log likelihood of x를 잘 계산하는 것이고, 이 목적식으로부터 loss 도출 가능

noise x_t의 분포 / noise x_t의 분포 / diffusion process의 조건부 gaussian 분포 / reverse process의 조건부 gaussian 분포

$$Loss_{diffusion} = \underbrace{D_{KL}(q(z \mid x_0) \,||\, P_\theta(z))}_{\text{두 gaussian distribution 간 KL divergence 최소화}} + \sum_{t=2}^{T} \underbrace{D_{KL}(q(x_{t-1} \mid x_t, x_0) \,||\, P_\theta(x_{t-1} \mid x_t))}_{\text{Denoising Process KL divergence 최소화}} - \underbrace{E_q[\log P_\theta(x_0 \mid x_1)]}_{\text{Latent variable } x_1\text{에서 원본 } x_0\text{를 추정하는 확률 모델의 parameter 최적화}}$$

- Reverse process ($P_\theta(x_{t-1} \mid x_t)$)는 Diffusion process ($q(x_{t-1} \mid x_t, x_0)$)를 approximate 하도록 학습

Diffusion Models Beat GANs[1]

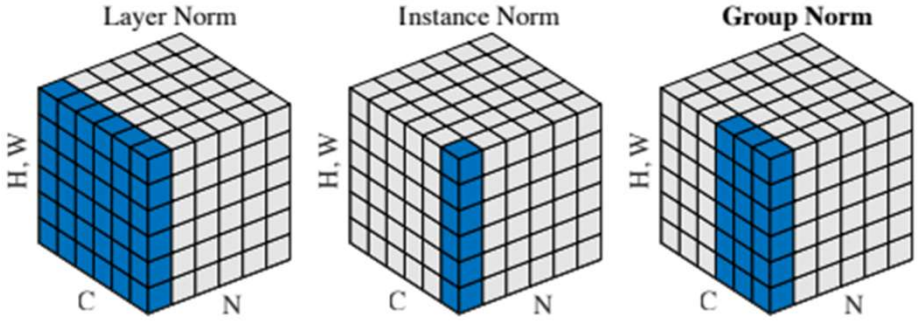
- (Simple) architectural improvement
  - Architecture can give a substantial boost to sample quality on much larger and more diverse datasets at a higher resolution
    - Increase in the number of attention heads (multi head attention)
    - Use of attention at 32x32, 16x16, and 8x8 resolutions (multi resolution) rather than only at 16x16
    - Use of the BigGAN [2] residual block for upsampling and downsampling the activations

	LSUN	ImageNet 64	ImageNet 128	ImageNet 256	ImageNet 512
Diffusion steps	1000	1000	1000	1000	1000
Noise Schedule	linear	cosine	linear	linear	linear
Model size	552M	296M	422M	554M	559M
Channels	256	192	256	256	256
Depth	2	3	2	2	2
Channels multiple	1,1,2,2,4,4	1,2,3,4	1,1,2,3,4	1,1,2,2,4,4	0.5,1,1,2,2,4,4
Heads			4		
Heads Channels	64	64		64	64
Attention resolution	32,16,8	32,16,8	32,16,8	32,16,8	32,16,8
BigGAN up/downsample	✓	✓	✓	✓	✓
Dropout	0.1	0.1	0.0	0.0	0.0
Batch size	256	2048	256	256	256
Iterations	varies*	540K	4360K	1980K	1940K
Learning Rate	1e-4	3e-4	1e-4	1e-4	1e-4

Table 11: Hyperparameters for diffusion models. *We used 200K iterations for LSUN cat, 250K for LSUN horse, and 500K for LSUN bedroom.

Diffusion Models Beat GANs[1]

- (Simple) architectural improvement
  - Adaptive Group Normalization (AdaGN)
    - Feature map들을 몇 개의 group으로 묶고, 그 안에서 normalization을 하는 방식
    - LN and IN have limited success in visual recognition, for which GN presents better results



[Facebook AI Research, 2018]

$$\mathrm{AdaGN}(h, y) = y_s\ \mathrm{GroupNorm}(h) + y_b$$

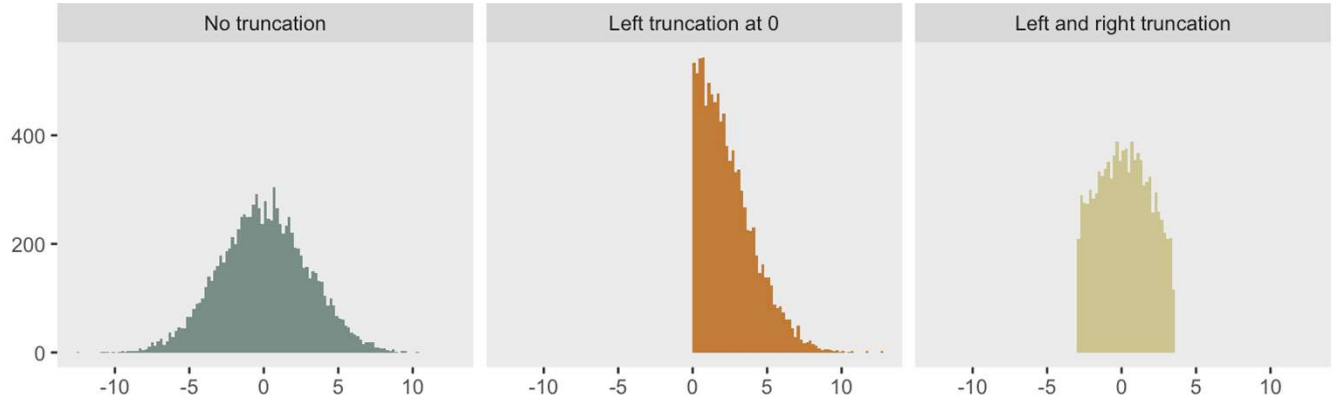
h에 GN 적용 후 scale and shift 해주는 방식

time step & class embedding을 residual block에 넣어 줌으로써 성능(FID)을 개선하였음

h : the intermediate activations of the residual block following the first convolution

Diffusion Models Beat GANs[1]

- Classifier guidance
  - GANs hold the state-of-the-art on most image generation tasks as measured by sample quality metrics such as FID, Inception Score and Precision
  - The gap between diffusion models and GANs
    - GANs are able to trade off diversity for fidelity, producing high quality samples but not covering the whole distribution
      - GAN에서 truncation trick(the latent vector is sampled from a truncated normal distribution)을 쓰면 naturally leads to a decrease in diversity but an increase in fidelity
        - ✓ Truncation trick : 실제로 학습 데이터의 분포를 고려하면, density가 낮은 부분의 경우 학습 후 표현이 잘 되지 않음. 즉, 생성기가 제대로 학습을 하지 못하는 부분을 방지하기 위하여 쓰는 방법으로 이는 학습 중에 적용하는 것이 아닌 학습이 완료된 네트워크의 input을 제어하는 방법



Diffusion Models Beat GANs[1]

- Classifier guidance
  - 이미지 생성 프로세스를 특정 class를 만들도록 guide 한다는 의도
  - 이 방법으로 diversity와 fidelity trade-off를 통해 생성된 이미지 성능 향상 가능
  - Classifier $p_\phi(y \mid x_t)$를 노이즈 이미지 x_t에 학습하고,
  - 이에 대한 $\nabla_{x_t} \log p_\phi(y \mid x_t)$를 이용해서 diffusion sampling process에 임의의 class label y에 대한 guide를 줌

Algorithm 1 Classifier guided diffusion sampling, given a diffusion model $(\mu_\theta(x_t), \Sigma_\theta(x_t))$, classifier $p_\phi(y|x_t)$, and gradient scale s.

Input: class label y, gradient scale s
$x_T \leftarrow$ sample from $\mathcal{N}(0, \mathbf{I})$ #정규분포에서 noise를 샘플링
for all t from T to 1 **do**
  $\mu, \Sigma \leftarrow \mu_\theta(x_t), \Sigma_\theta(x_t)$ # μ랑 Σ를 가지는 normal distribution을 샘플링
  $x_{t-1} \leftarrow$ sample from $\mathcal{N}(\mu + s\Sigma \nabla_{x_t} \log p_\phi(y|x_t), \Sigma)$ # μ를 classifier의 probability를 maximize하는 방향으로 shift
end for
return x_0

Diffusion Models Beat GANs[1]

- Classifier guidance
  - Scaling classifier gradient

$$s \cdot \nabla_x \log p(y|x) = \nabla_x \log \frac{1}{Z} p(y|x)^s$$

  - Using a larger gradient scale focuses more on the modes of the classifier, which is potentially desirable for producing higher fidelity (but less diverse) samples
    - $s > 1$, classifier의 distribution이 더 sharp해지는 효과를 낳고, classifier가 더 강해지면서 guidance도 더 강해지는 효과를 줌

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.

Diffusion Models Beat GANs[1]

- Results
  - Models can obtain the best FID on each task, and the best sFID on all but one task
  - For higher resolution ImageNet, we observe that classifier guidance allows our models to substantially outperform the best GANs

Model	FID	sFID	Prec	Rec
LSUN Bedrooms 256×256				
DCTransformer† [42]	6.40	6.66	0.44	**0.56**
DDPM [25]	4.89	9.07	0.60	0.45
IDDPM [43]	4.24	8.21	0.62	0.46
StyleGAN [27]	2.35	6.62	0.59	0.48
ADM (dropout)	**1.90**	**5.59**	**0.66**	0.51
LSUN Horses 256×256				
StyleGAN2 [28]	3.84	6.46	0.63	0.48
ADM	2.95	**5.94**	0.69	**0.55**
ADM (dropout)	**2.57**	6.81	**0.71**	**0.55**
LSUN Cats 256×256				
DDPM [25]	17.1	12.4	0.53	0.48
StyleGAN2 [28]	7.25	**6.33**	0.58	0.43
ADM (dropout)	**5.57**	6.69	**0.63**	**0.52**
ImageNet 64×64				
BigGAN-deep* [5]	4.06	3.96	**0.79**	0.48
IDDPM [43]	2.92	**3.79**	0.74	0.62
ADM	2.61	**3.77**	0.73	0.63
ADM (dropout)	**2.07**	4.29	0.74	**0.63**

Model	FID	sFID	Prec	Rec
ImageNet 128×128				
BigGAN-deep [5]	6.02	7.18	**0.86**	0.35
LOGAN† [68]	3.36			
ADM	5.91	**5.09**	0.70	**0.65**
ADM-G (25 steps)	5.98	7.04	0.78	0.51
ADM-G	**2.97**	**5.09**	0.78	0.59
ImageNet 256×256				
DCTransformer† [42]	36.51	8.24	0.36	**0.67**
VQ-VAE-2†‡ [51]	31.11	17.38	0.36	0.57
IDDPM‡ [43]	12.26	5.42	0.70	0.62
SR3†‡ [53]	11.30			
BigGAN-deep [5]	6.95	7.36	**0.87**	0.28
ADM	10.94	6.02	0.69	0.63
ADM-G (25 steps)	5.44	5.32	0.81	0.49
ADM-G	**4.59**	**5.25**	0.82	0.52
ImageNet 512×512				
BigGAN-deep [5]	8.43	8.13	**0.88**	0.29
ADM	23.24	10.19	0.73	**0.60**
ADM-G (25 steps)	8.41	9.67	0.83	0.47
ADM-G	**7.72**	**6.57**	0.87	0.42

Precision : relative to fidelity, Recall : relative to diversity

Diffusion Models Beat GANs[1]

- Results
  - The samples are of similar perceptual quality, the diffusion model contains more modes than the GAN

Figure 6: Samples from BigGAN-deep with truncation 1.0 (FID 6.95, left) vs samples from our diffusion model with guidance (FID 4.59, middle) and samples from the training set (right).

Diffusion Models Beat GANs[1]

- Conclusion
  - Contribution
    - Diffusion models can obtain better sample quality than state-of-the-art GANs
    - Classifier guidance technique allows to achieve high sample quality on class-conditional tasks
      - The scale of the classifier gradients can be adjusted to trade off diversity for fidelity
  - Limitation
    - Still slower than GANs at sampling time due to the use of multiple denoising steps
      - Distill the sampling process into a single step model (faster than single-step likelihood-based models)
    - Proposed classifier guidance technique is currently limited to labeled datasets
      - No effective strategy for trading off diversity for fidelity on unlabeled datasets

감사합니다.