

# **Classifier Guidance**

---

**: Diffusion Modes Beat GANs on Image Synthesis**

김재원

---

1

# INTRODUCTION

# 1. Introduction

## GAN vs Diffusion

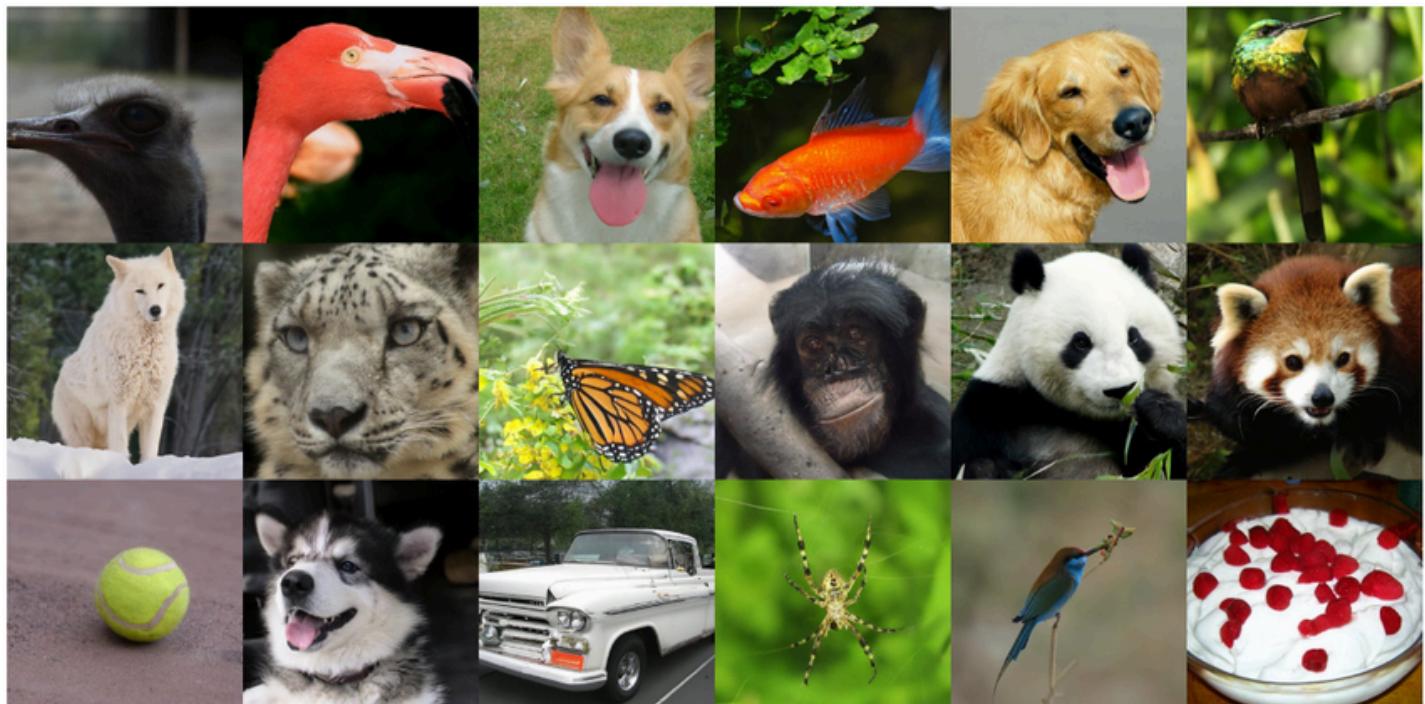


Figure 1: Selected samples from our best ImageNet 512×512 model (FID 3.85)

- Diffusion의 단점: GAN보다 낮은 sampling quality

ImageNet과 같이 복잡한 Dataset에 대한 sampling quality가 GAN에 비해 상대적으로 낮음

- How Diffusion Beat GANs?  
OpenAI에서 어떻게 Diffusion을 이용하여 BigGAN (SOTA)보다 더 좋은 Sampling Quality를 달성하는지 설명

- Unconditional / Conditional Generation

# 1. Introduction

## Why diffusion is not good enough?

- Diffusion은 GAN보다 sample diversity / training stability 모두 높음
- 그럼에도 불구하고 다른 Probabilistic Generative Model보다 좋은 성능을 내지 못하는 이유?

1)GAN은 오랜시간동안 연구가 진행됨 (Optimal Architecture, Hyperparameter set)

2)GAN은 fieldity가 높은 대신 diversity가 낮음 (trade - off)

## So...?

- 저자들은 GAN의 장점(fieldity, optimal Architecture)을 Diffusion에 이용하자고 주장

---

2

## BACKGROUND

## 2. Background

### DDPM Baseline

**Improved DDPM (Trainable Variance) + DDIM (for faster sampling)**

#### Improved DDPM

- 적은 time step을 통해 high sampling quality
- Training Method의 수정 (Algorithm 1)

#### DDIM

- DDPM 동일한 Marginal Distribution  $p(x_t | x_0)$ 를 가지도록 하는 Non-Markovian Forward Process  $p(x_{t-1} | x_t, x_0)$  설정
- Sampling Process의 가속화 (Algorithm 2)

## 2. Background

### 2.1) Improvements

#### DDIM

50 step 보다 작은 sequence로 sample을 생성할 경우 DDIM 사용

#### Improved DDPM

$$L_{\text{hybrid}} = L_{\text{simple}} + \lambda L_{\text{vlb}}$$

Hybrid Loss를 사용

$$L_{\text{vlb}} := L_0 + L_1 + \dots + L_{T-1} + L_T \quad (4)$$

$$L_0 := -\log p_\theta(x_0|x_1) \quad (5)$$

$$L_{t-1} := D_{KL}(q(x_{t-1}|x_t, x_0) \parallel p_\theta(x_{t-1}|x_t)) \quad (6)$$

$$L_T := D_{KL}(q(x_T|x_0) \parallel p(x_T)) \quad (7)$$

#### Trainable Variance

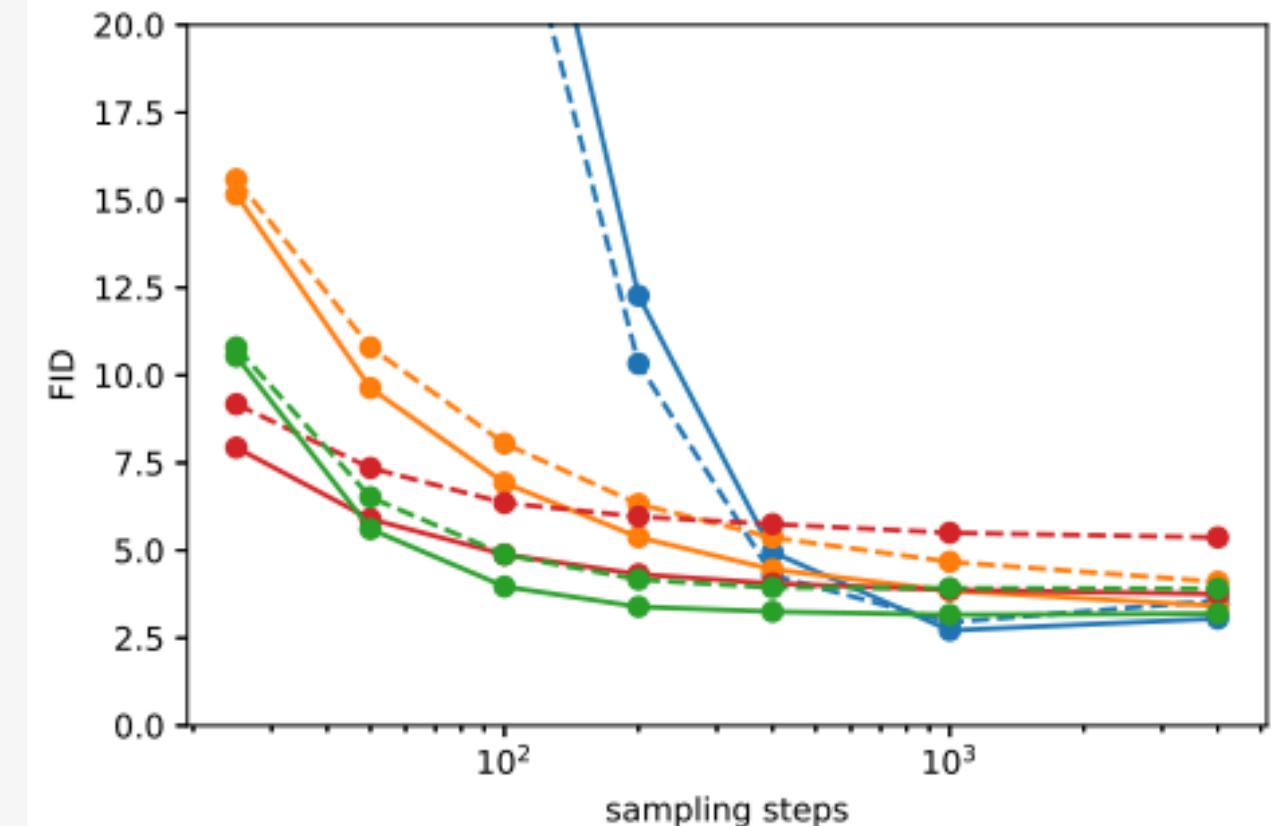
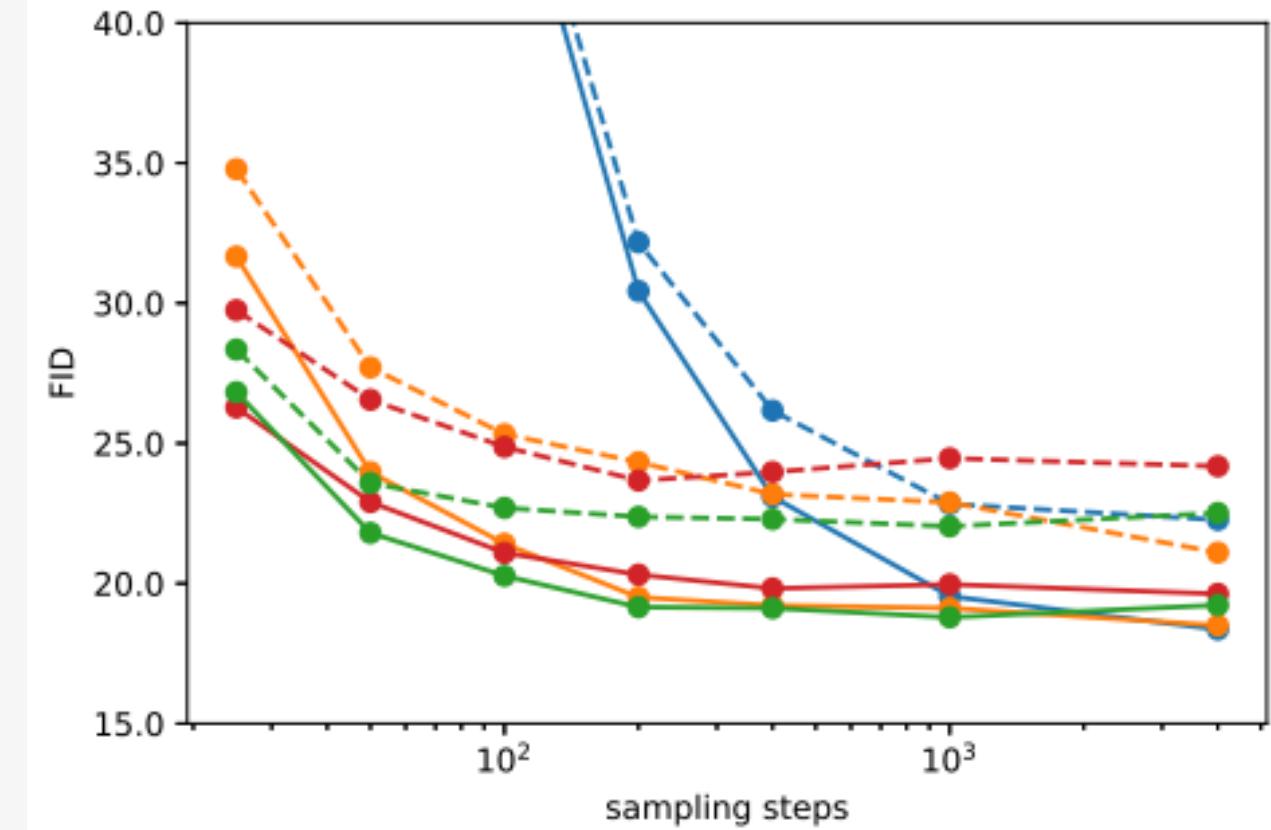
$$\Sigma_\theta(x_t, t) = \exp(v \log \beta_t + (1 - v) \log \tilde{\beta}_t)$$

## 2. Background

### 2.1) Improvements

DDIM

50 step보다 작게 sampling할 경우 Sample Quality가 더 좋음



## 2. Background

### 2.2) Sample Quality Metrics

#### Inception Score (fidelity, Quantity Measure)

$$IS(G) = \exp(\mathbb{E}_{x \sim G}(D_{KL}(p(y|x, p(y))))$$

- 생성된 이미지만 사용하여 성능을 평가 (높을수록 good)
- $p(y)$ : 생성되는 sample이 전체 class에 대해 고르게 나타나는가
- $p(y|x)$ : 생성된 sample의 quality를 측정
- 각 class 별로 다양한 sample을 생성하지 못함 (**Mode Collapse**같은 문제 확인 불가)

#### FID (Quantity Measure)

$$FID(x, g) = \| \mu_x - \mu_g \|_2^2 + Tr(\Sigma_x + \Sigma_g - 2(\Sigma_x \Sigma_g)^{1/2})$$

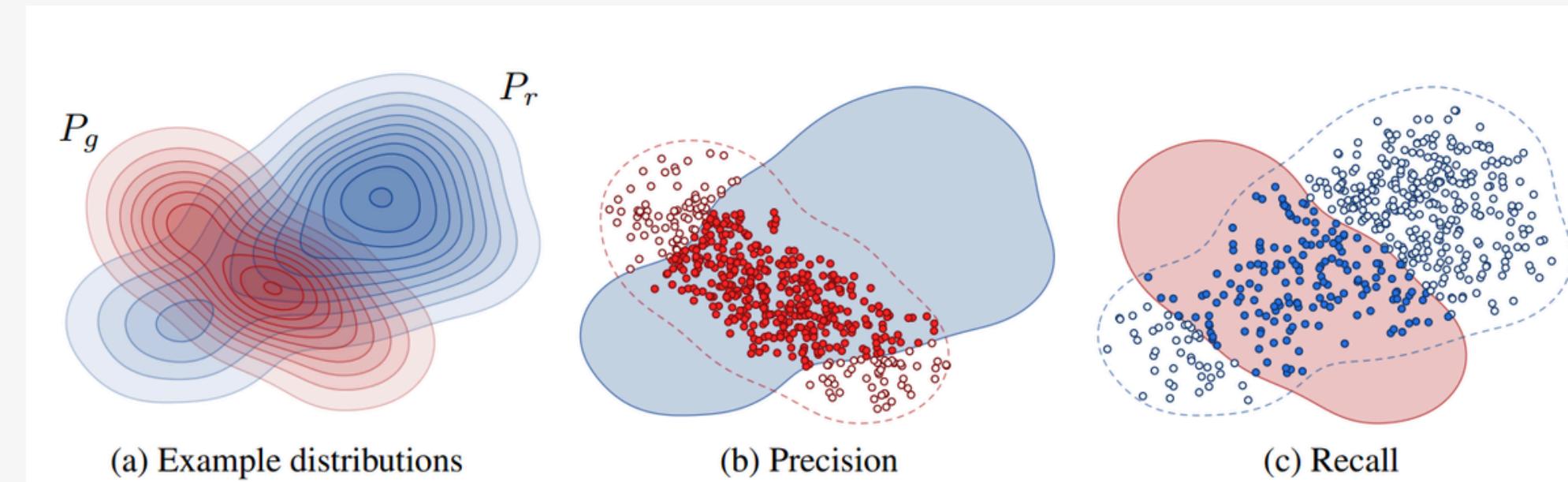
- 실제 이미지와 생성된 이미지 모두 사용 (낮을수록 good)
- Inception Network를 활용하여 Layer에서의 feature 사용, Multivariate Gaussian 모델링

#### Precision, Recall Metric

## 2. Background

### 2.2) Sample Quality Metrics

#### Precision, Recall Metric



- sample fieldity(precision), sample diversity(recal)을 분리
- 실제 sample distribution:  $P_r$
- 예측한 sample distribution:  $P_g$

#### Precision (fieldity)

$$(Precision) = \frac{TP}{TP + FP}$$

- 모델이 생성한 샘플 중 실제 샘플의 분포 내에 들어가는 정도를 측정하는 것이 샘플링 성능이랑 관련

#### Recall (diversity)

$$(Recall) = \frac{TP}{TP + FN}$$

- 실제 분포의 샘플 중 모델이 생성한 샘플에 들어가는 정도를 측정하는 것이 샘플링 다양성과 관련

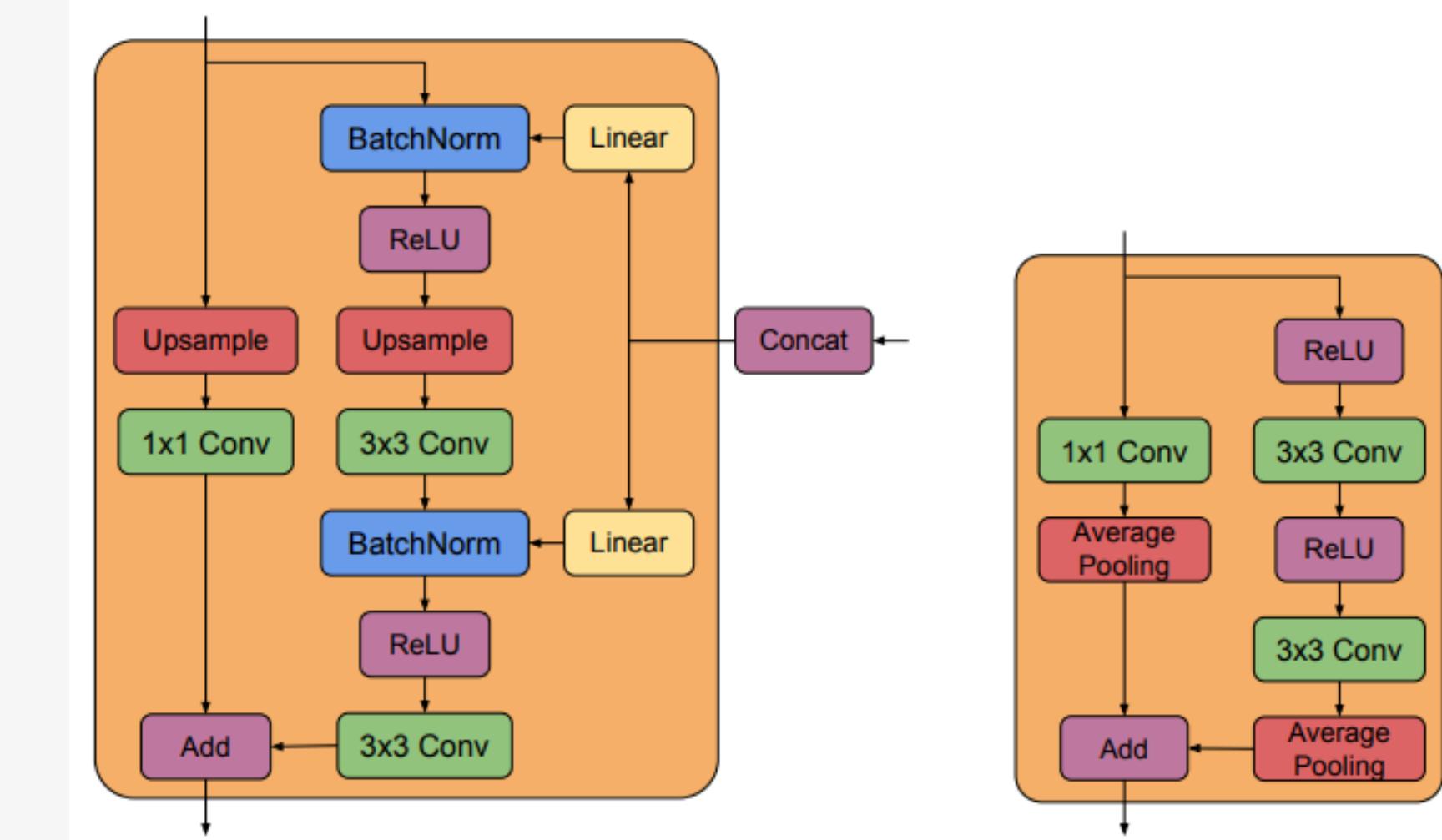
---

3

## ARCHITECTURE IMPROVEMENTS

### 3. Architecture Improvements

- Diffusion Network 구조에 따른 충분한 Research의 부재
- Diffusion Model의 Sampling Quality를 높일 수 있는 구조



BigGAN Architecture

- Depth(네트워크 깊이) 대비 Width(채널 수)를 늘린다. 이때 모델 크기는 상대적으로 일정하게 유지하게끔 증가시킨다.
- Attention head의 갯수를 늘린다(베이스라인이 되는 UNet의 residual block에 attention이 들어간다).
- Attention을 원래  $16 \times 16$ 의 feature map level에만 적용했었는데, 이걸  $32 \times 32, 8 \times 8$ 의 feature map에도 적용한다.
- Activation upsampling 및 downsampling 시에 BigGAN의 residual block을 사용한다.
- Residual connection을  $\frac{1}{\sqrt{2}}$  만큼 수행한다.

### 3. Architecture Improvements

ImageNet 128×128 크기의 이미지에 대해 256의 batch size, 250의 sampling step으로 통일하고 FID를 기준으로 실험을 진행

Channels	Depth	Heads	Attention resolutions	BigGAN up/downsample	Rescale resblock	FID 700K	FID 1200K
160	2	1	16	X	X	15.33	13.21
128	4	4	32,16,8		✓	-0.21	-0.48
						-0.54	-0.82
						-0.72	-0.66
						-1.20	-1.21
160	2	4	32,16,8	✓	✓	0.16	0.25
					X	<b>-3.14</b>	<b>-3.00</b>

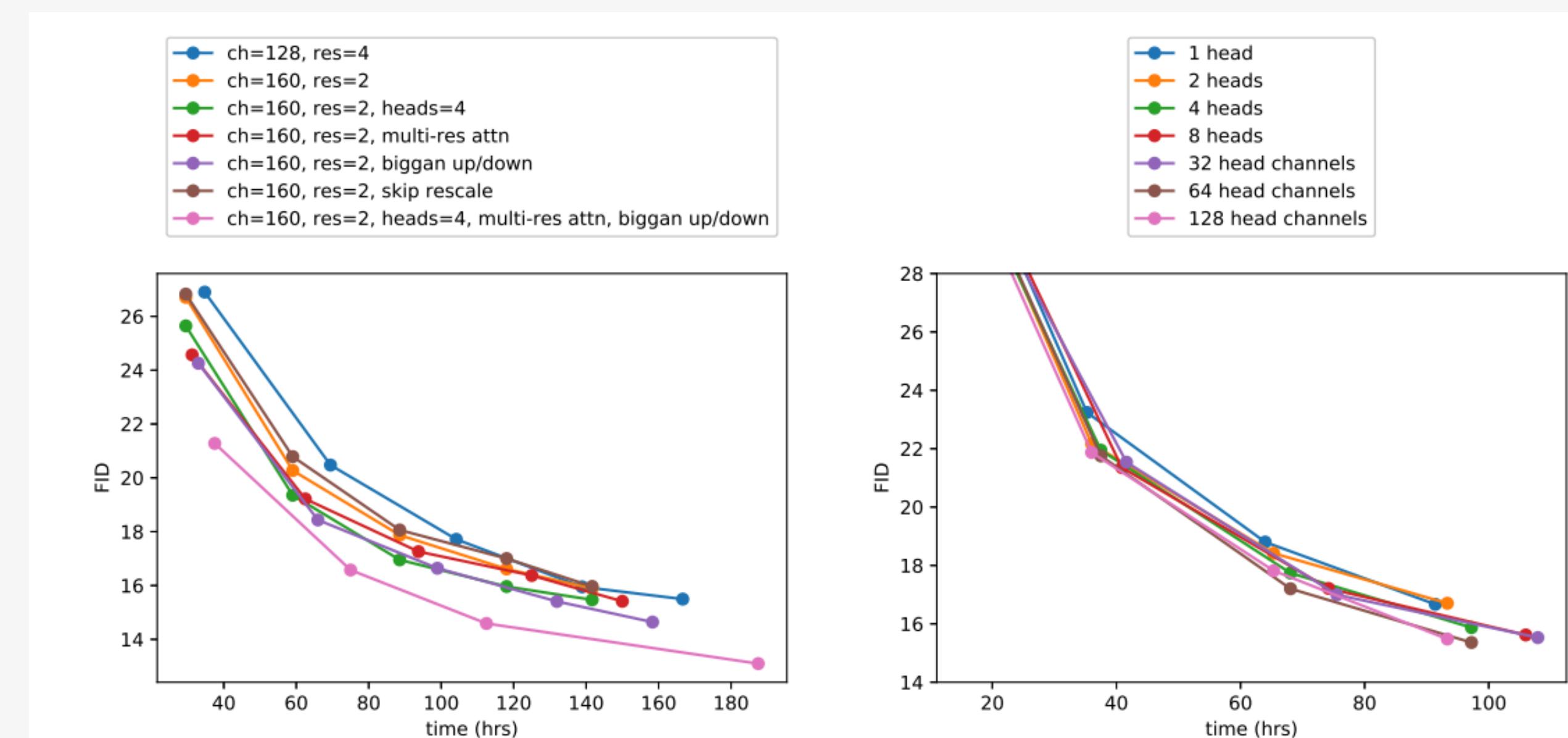
Number of heads	Channels per head	FID
1		14.08
2		-0.50
4		-0.97
8		-1.17
	32	-1.36
	64	-1.03
	128	-1.08

위 테이블에서는 rescaling 부분을 제외하고 모든 구조적 제안이 FID 성능을 높이는데 기여하는 것을 볼 수 있다.

### 3. Architecture Improvements

아래 그래프에서 보게 되면 depth를 증가시키는 선택 또한 성능 향상에 도움이 되는 경향을 보았지만,

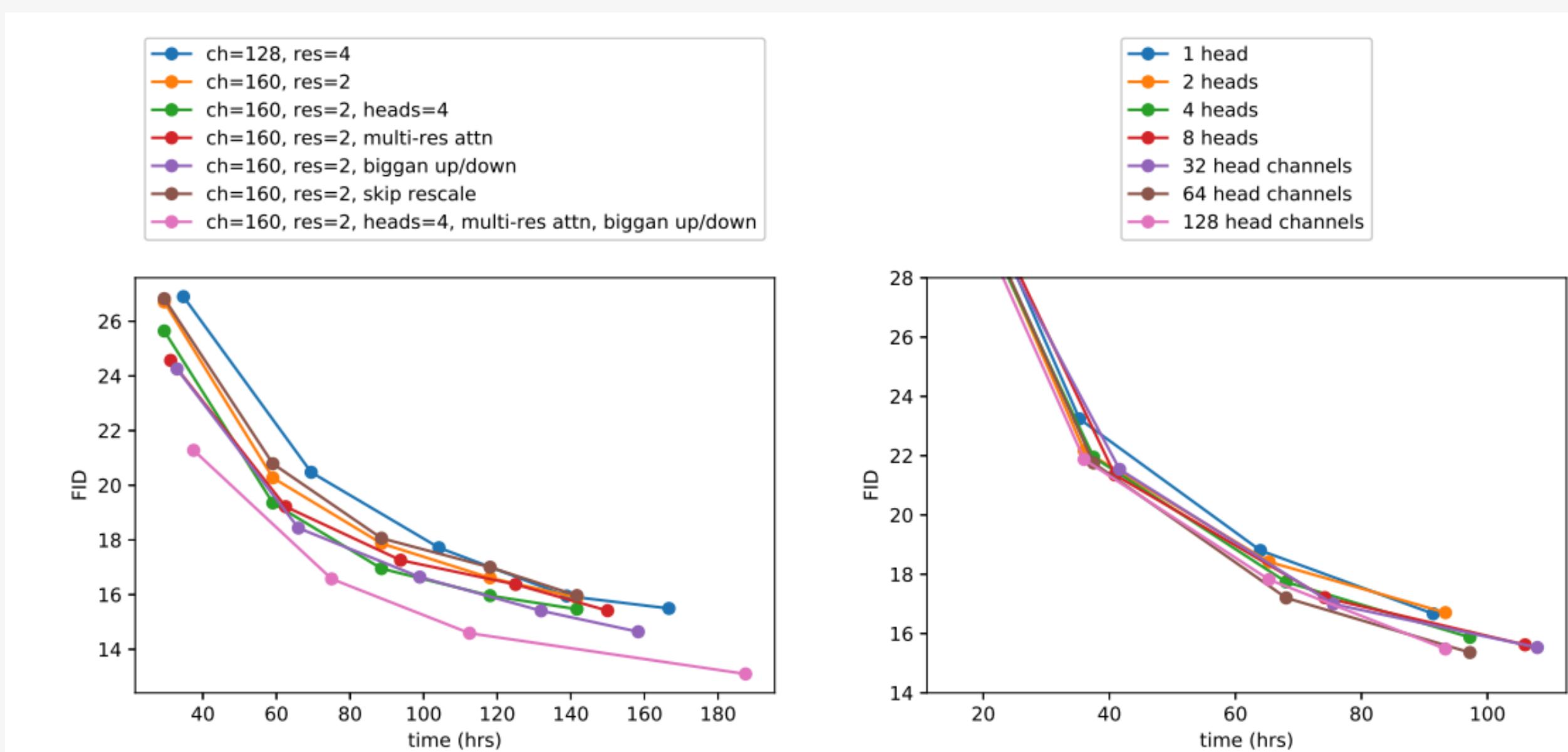
학습 시간이 지나치게 증가한다는 문제 때문에 더이상 실험을 진행하지 않음



# 3. Architecture Improvements

# attention configuration에 대한 실험

- 실험 결과를 보면 head의 개수를 늘리고 각 head의 channel 수를 줄이는 것이 가장 좋은 FID를 보여줌
  - 그래프에서 확인해보면 64 channel을 사용할 때가 학습 속도 면에서 가장 성능 효율이 좋았기 때문에 이를 사용
  - 신기하게도 이러한 구조적 장점(성능 경향성)은 transformer의 구조와 동일



### 3. Architecture Improvements

#### 3.1) Adaptive GroupNorm

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

time step과 class embedding을 각 residual block에 stylization

- Hidden Layer activation:  $h$
- time step, class embedding의 linear projection:  $[y_s, y_b]$
- StyleGAN과 대체로 비슷 (GroupNorm 부분만 제외)

Ablation study를 통해 결정한 최종 Archiecture

- 각 resolution마다 2개의 residual block(BigGAN)을 가지며, width도 resolution에 맞게 조정됨
- Attention head마다 64의 channel 수를 가지는데, resolution 32, 16, 8에 모두 attention layer가 있음
- BigGAN residual block을 upsampling, downsampling할 때 사용하며 AdaGN이 들어가서 timestep과 class embedding을 넣어줌

---

# 4

## CLASSIFIER GUIDANCE

## 4. Classifier Guidance

### 4.1 Conditional Reverse Noising Process

각 noised image에 대해서 사전 학습된 pre-trained classifier network  $p_\phi(y|x_t, t)$  diffusion process에 완전히 explicit한 정보이기 때문에 다음과 같이 normalizing factor Z에 대해 constant 취급

(자세한 증명은 논문 Appendix에 있음)

$$p_{\theta, \phi}(x_t | x_{t+1}, y) = Z p_\theta(x_t | x_{t+1}) p_\phi(y | x_t)$$

일반적인 unconditional diffusion process는 다음과 같이 정의된다.

$$\log p_\theta(x_t | x_{t+1}) = -\frac{1}{2} (x_t - \mu)^\top \Sigma^{-1} (x_t - \mu) + C$$

이때  $\log p_\phi(y|x_t, t)$ 가 가지는 curvature가  $\log p_\theta(x_t | x_{t+1})$ 이 가지는 curvature 보다 매우 작게 된다.

(계수가  $1/2\Sigma$  이기 때문, diffusion process에서  $\Sigma$ 는 매우 작은 값)

## 4. Classifier Guidance

### 4.1 Conditional Reverse Noissing Process

$x_t = \mu$ 인 점(diffusion reverse process의 quadratic function 꼭짓점)에서 Classifier Guidance부분을 Taylor Series의 1차 근사를 통해 나타낸다.

$$\log p_\phi(y|x_t) \approx \log p_\phi(y|x_t)|_{x_t=\mu} + (x_t - \mu) \nabla_{x_t} \log p_\phi(y|x_t)|_{x_t=\mu} = (x_t - \mu)g + C_1$$

$\mu$ 가 sampling되는 부분인데, 여기서  $p_\theta$  대비  $p_\phi$ 가 가지는 곡률이 상대적으로 매우 작기에 식 전개 가능

이때  $g$ 는  $x_t = \mu$ 일 때 classifier에서의 log-likelihood gradient이다.

$$\begin{aligned}\log(p_\theta(x_t|x_{t+1})p_\phi(y|x_t)) &\approx -\frac{1}{2}(x_t - \mu)^\top \Sigma^{-1}(x_t - \mu) + (x_t - \mu)g + C_2 \\ &= -\frac{1}{2}(x_t - \mu - \Sigma g)^\top \Sigma^{-1}(x_t - \mu - \Sigma g) + \frac{1}{2}g^\top \Sigma g + C_2 \\ &= -\frac{1}{2}(x_t - \mu - \Sigma g)^\top \Sigma^{-1}(x_t - \mu - \Sigma g) + C_3 \\ &= \log p(z) + C_4, \quad z \sim \mathcal{N}(\mu + \Sigma g, \Sigma)\end{aligned}$$

## 4. Classifier Guidance

### 4.1 Conditional Reverse Noising Process

Classifier에 의한 Guidance는 sampling할 때 gradient 방향을 틀어준다고 생각하자

---

**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  $\mathcal{N}(0, \mathbf{I})$   
**for all**  $t$  from  $T$  to 1 **do**  
     $\mu, \Sigma \leftarrow \mu_\theta(x_t), \Sigma_\theta(x_t)$   
     $x_{t-1} \leftarrow$  sample from  $\mathcal{N}(\mu + s\Sigma \nabla_{x_t} \log p_\phi(y|x_t), \Sigma)$   
**end for**  
**return**  $x_0$

---

## 4. Classifier Guidance

### 4.2) Conditional Sampling for DDIM

DDIM과 같이 Deterministic한 sampling을 하는 경우에는 이전의 식 사용 불가

(수식 전개 과정 상 Markov-process를 가정)

$$x_{t-1} = \sqrt{\bar{\alpha}_{t-1}} \left( \frac{x_t - \sqrt{1 - \bar{\alpha}_t} \epsilon_\theta^{(t)}(x_t)}{\sqrt{\bar{\alpha}_t}} \right) + \sqrt{1 - \bar{\alpha}_{t-1} - \sigma_t^2} \cdot \epsilon_\theta^{(t)}(x_t) + \sigma_t z, \quad z \sim \mathcal{N}(0, I)$$

predicted  $x_0$       direction pointing to  $x_t$       random noise

deterministic DDIM은  $x_0$ 로부터  $x_t$ 를 예측하는 형태로 샘플링이 진행되다보니  $x_t$ 에 대한 classifier gradient를 적용할 수 없음

따라서 DDPM과 Score-based Model의 관련성을 이용하여 식 전개!

DDPM Sampling -> Score estimation function ( $s_\theta$ 에 대한 확률 미분방정식)

$$x_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left( x_t - \frac{\beta_t}{\sqrt{1 - \bar{\alpha}_t}} \epsilon_\theta(x_t, t) \right) + \sigma_t z$$

$$x_{t-1} = \frac{1}{\sqrt{\alpha_t}} (x_i + \beta_i s_{\theta^*}(x_i, i)) + \sqrt{\beta_i} z_i$$

## 4. Classifier Guidance

### 4.2) Conditional Sampling for DDIM

따라서 score function을 time t에 대해 정리

$$\nabla_{x_t} \log p_\theta(x_t) = -\frac{1}{\sqrt{1-\bar{\alpha}_t}} \epsilon_\theta(x_t)$$

$$\begin{aligned}\nabla_{x_t} \log(p_\theta(x_t)p_\phi(y|x_t)) &= \nabla_{x_t} \log p_\theta(x_t) + \nabla_{x_t} \log p_\phi(y|x_t) \\ &= -\frac{1}{\sqrt{1-\bar{\alpha}_t}} \epsilon_\theta(x_t) + \nabla_{x_t} \log p_\phi(y|x_t)\end{aligned}$$

epsilon을 다음과 같이 새롭게 정의 가능

$$\hat{\epsilon}_\theta(x_t) := \epsilon_\theta(x_t) - \sqrt{1-\bar{\alpha}_t} \nabla_{x_t} \log p_\phi(y|x_t)$$

## 4. Classifier Guidance

### 4.3) Scaling Classifier Gradients

Classifier pφ에 의한 score guide를 주기 위해서는 classification model을 학습시켜야 함.

- Classifier architecture는 UNet model의 downsampling 부분에서 추출된 feature map에 attention pooling( $8 \times 88 \times 8$ )을 통해 최종 output을 추출
- Classifier는 각 노이즈 스텝에 대해 분류할 수 있어야 하므로 각각의 time step에 대한 noised input을 학습
- 학습 이후에는 앞서 언급한 gradient 영향을 주면서 샘플링을 진행

## 4. Classifier Guidance

### 4.3) Scaling Classifier Gradients

초반 unconditional ImageNet model(class condition을 따로 embedding으로 주지 않은 네트워크)로 실험했을 때, classifier guidance  $s_s$ 를 11보다 크게 하지 않으면 원하는 class의 샘플이 나올 확률이 절반으로 뚝 떨어지는 것을 확인하였고, 심지어 이 확률로 샘플을 만들어도 시각적으로 그다지 해당 클래스의 범주에 속하지 않는 것을 확인하였다.

예컨데 “Pembroke Welsh corgi”의 class에 대한 scale을 1.0으로 주었을 때(좌측) 제대로 생성되지 않던 웰시코기 이미지가 10.0으로 키웠을 때 유의미하게 좋아지는 것을 볼 수 있다.



## 4. Classifier Guidance

### 4.3) Scaling Classifier Gradients

표에서 주목할 점은 unconditional model에 classifier guidance를 충분히 큰 값으로 주게 되면(guidance = 10.0) conditional model에 필적하는 FID 및 IS를 보여주는 것을 확인할 수 있다.

추가로 언급한 내용 중에 low resolution image를 condition으로 하는 2-stage diffusion process를 사용했을 때 BigGAN의 성능을 넘어선 것을 알 수 있는데, 여전히 샘플링 속도가 문제가 된다는 점과 classifier training으로부터 자유롭지 않기 때문에 labeled sample에 한정된다는 문제가 발생한다.

Conditional	Guidance	Scale	FID	sFID	IS	Precision	Recall
✗	✗		26.21	<b>6.35</b>	39.70	0.61	0.63
✗	✓	1.0	33.03	6.99	32.92	0.56	<b>0.65</b>
✗	✓	10.0	<b>12.00</b>	10.40	<b>95.41</b>	<b>0.76</b>	0.44
✓	✗		10.94	6.02	100.98	0.69	<b>0.63</b>
✓	✓	1.0	<b>4.59</b>	<b>5.25</b>	186.70	0.82	0.52
✓	✓	10.0	9.11	10.93	<b>283.92</b>	<b>0.88</b>	0.32

