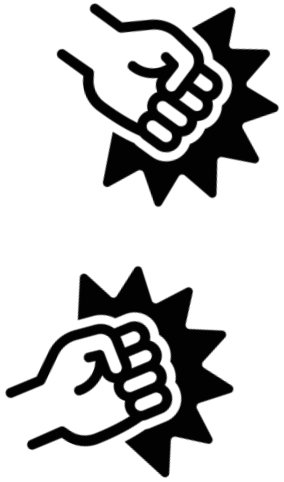
Two Transformers Can Make One Strong GAN

TransGAN

김영민 김지수 이다인

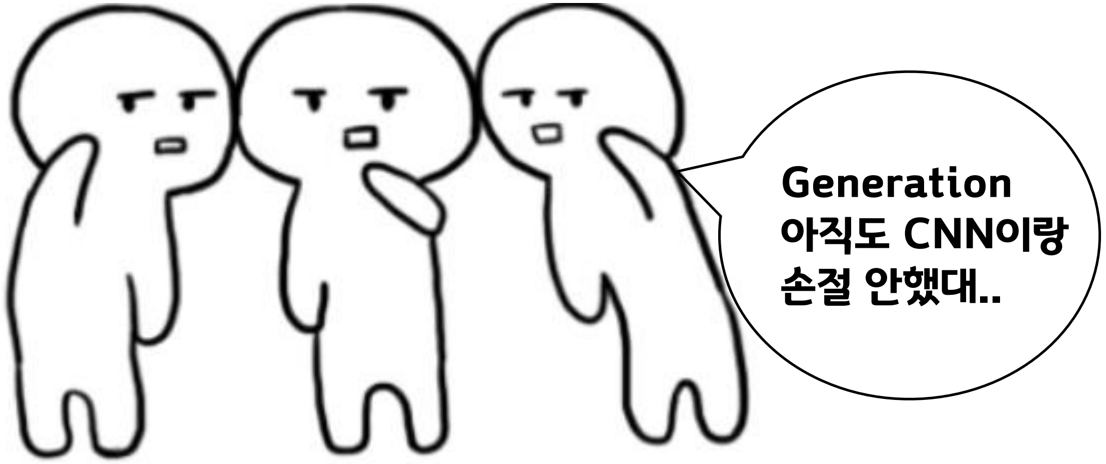
1. 최근 동향

Transformer

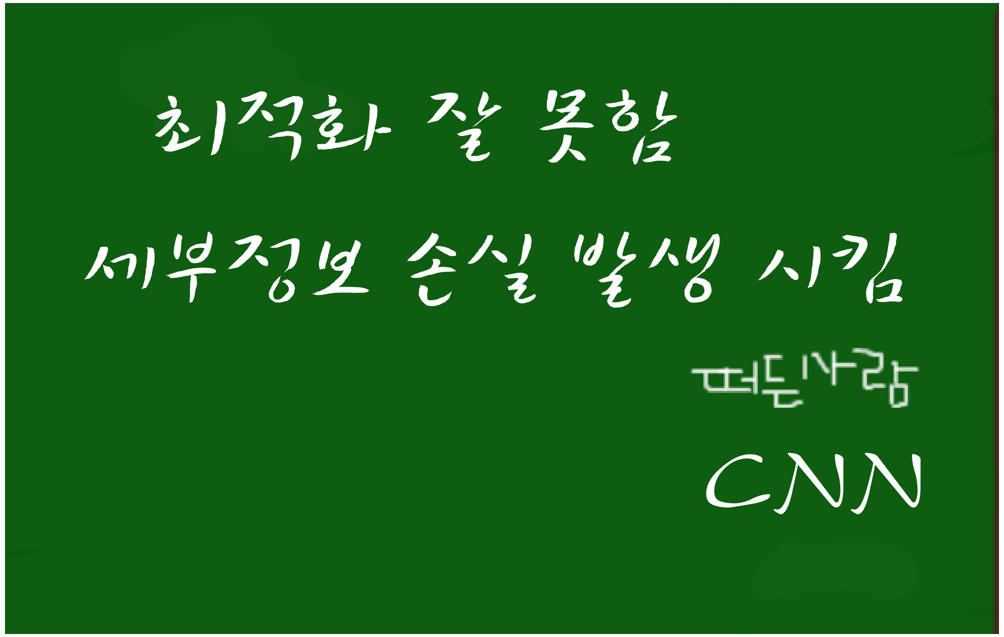


MLP

멸종 위기 CNN



Classification Segmentation Detection



2. Contribution

1. Convolution layer 멈춰!

GAN completely
free of convolution

2. memory-friendly
Transformer

Generator based on
Memory friendly
Transformer

3. SOTA!

STL-10
FID,IS 1st

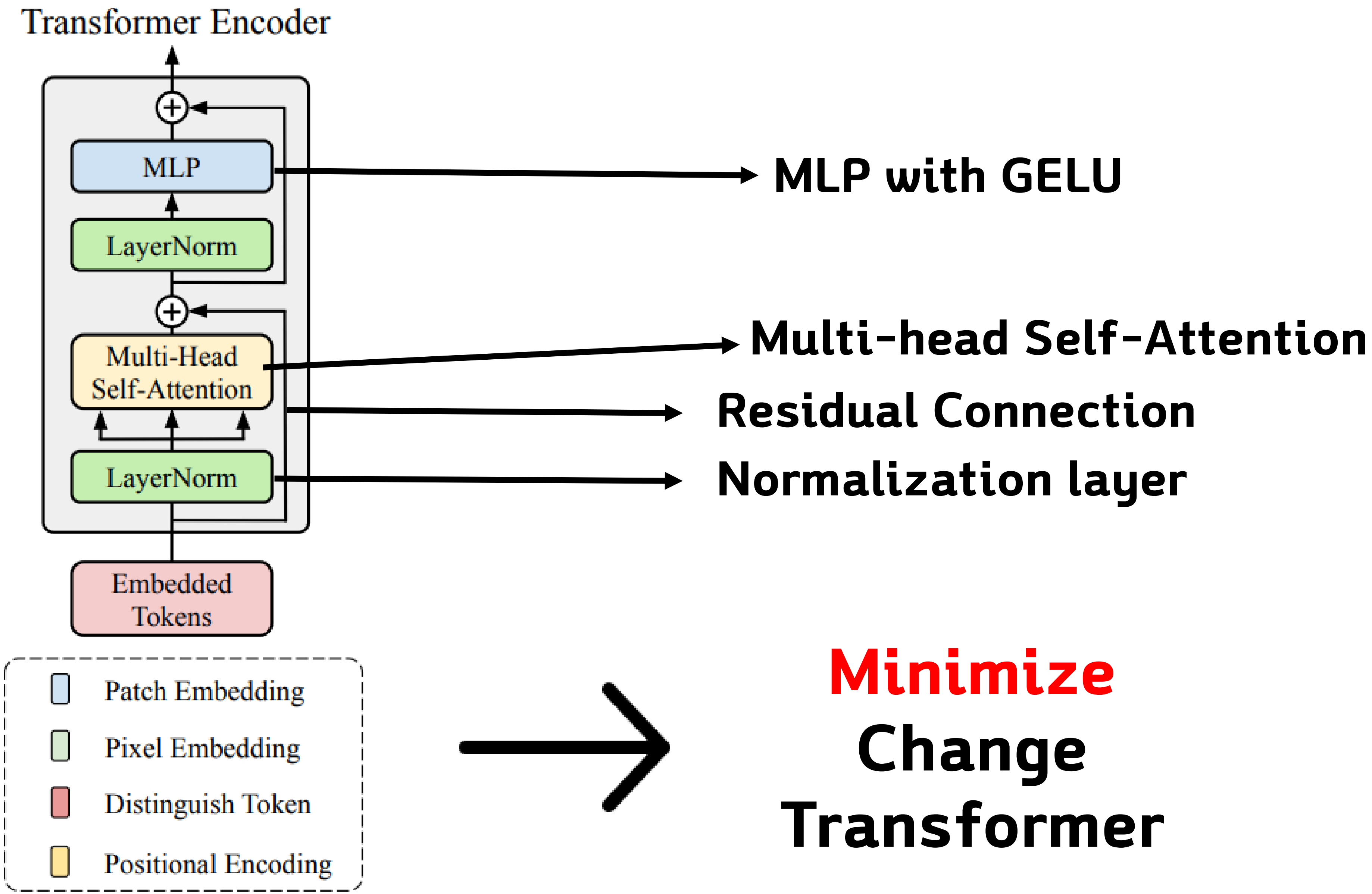
3. Charm of Transformer

1. It has strong representation capability and is free of human-defined inductive bias.

2. the transformer architecture is general, conceptually simple and has the potential to become a powerful "universal" model across tasks and domains

4. A Journey Towards GAN with Pure Transformers

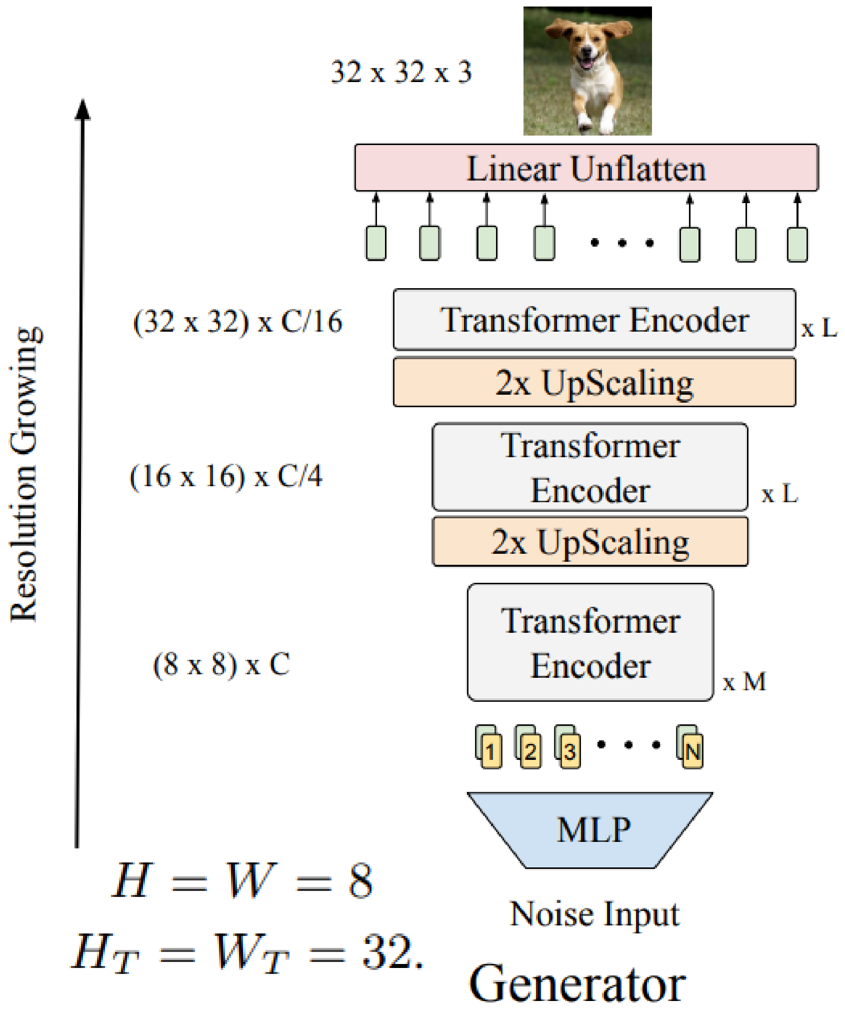
Transformer Encoder As Basic Block



Minimize Change Transformer

4. A Journey Towards GAN with Pure Transformers

Memory-Friendly Generator



$H = W = 8$

$H_T = W_T = 32.$

PixelShuffle method

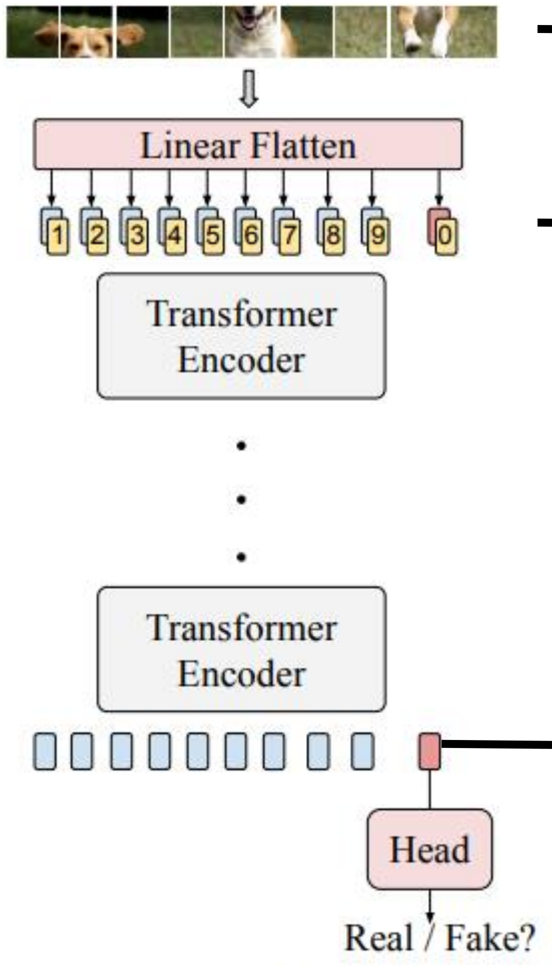
멈춤 조건 $W_T = W, H_T = H$

Pseudo code

```
While W_t == W and H_t == H:
    W *= 2
    H *= 2
    C *= 1/4
    Resolution = W * H * C
Resolution = W_t * H_t * 3
```

4. A Journey Towards GAN with Pure Transformers

Tokenized-Input for Discriminator



Discriminator

Patch-Level Tokenizing

Reshape 1d sequence

[CLS] token → **Predict Real/Fake**

4. A Journey Towards GAN with Pure Transformers

Result

Dataset = CIFAR-10

GENERATOR	DISCRIMINATOR	IS↑	FID↓
AUTOGAN	AUTOGAN	8.55± 0.12	**12.42**
TRANSFORMER	AUTOGAN	**8.59± 0.10**	13.23
AUTOGAN	TRANSFORMER	6.17± 0.12	49.83
TRANSFORMER	TRANSFORMER	6.95 ± 0.13	41.41

→ **Paper method**

Transformer G -> Good Performance

But, Transformer D -> Bad Performance

5. Data Augmentation is Crucial for TransGAN

Few-shot

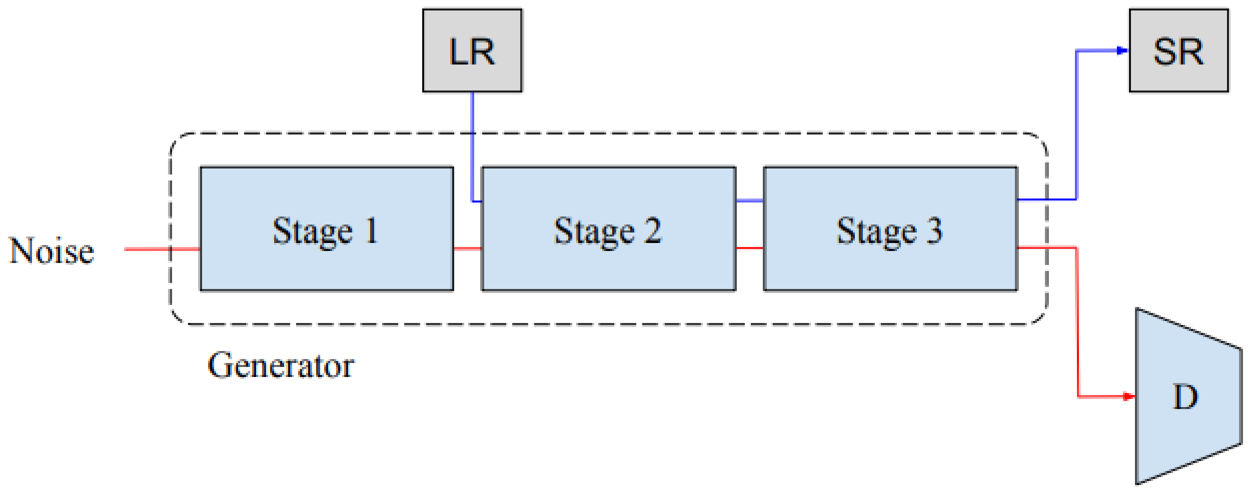
좋은 생성 모델로 만든 가짜 이미지로 데이터 증강을 하는 방법

METHODS	DA	IS ↑	FID ↓
WGAN-GP	×	**6.49** ± 0.09	39.68
(GULRAJANI ET AL., 2017)	√	6.29 ± 0.10	**37.14**
AUTOGAN	×	8.55 ± 0.12	**12.42**
(GONG ET AL., 2019)	√	**8.60 ± 0.10**	12.72
STYLEGAN V2	×	9.18	11.07
(ZHAO ET AL., 2020B)	√	**9.40**	**9.89**
TRANSGAN	×	6.95 ± 0.13	41.41
	√	**8.15 ± 0.14**	**19.85**

Data Augmentation을 통해 눈에 띄는 성능 향상!

6. Co-Training with Self-Supervised Auxiliary Task

MT-CT **Multi-Task Co-Training**



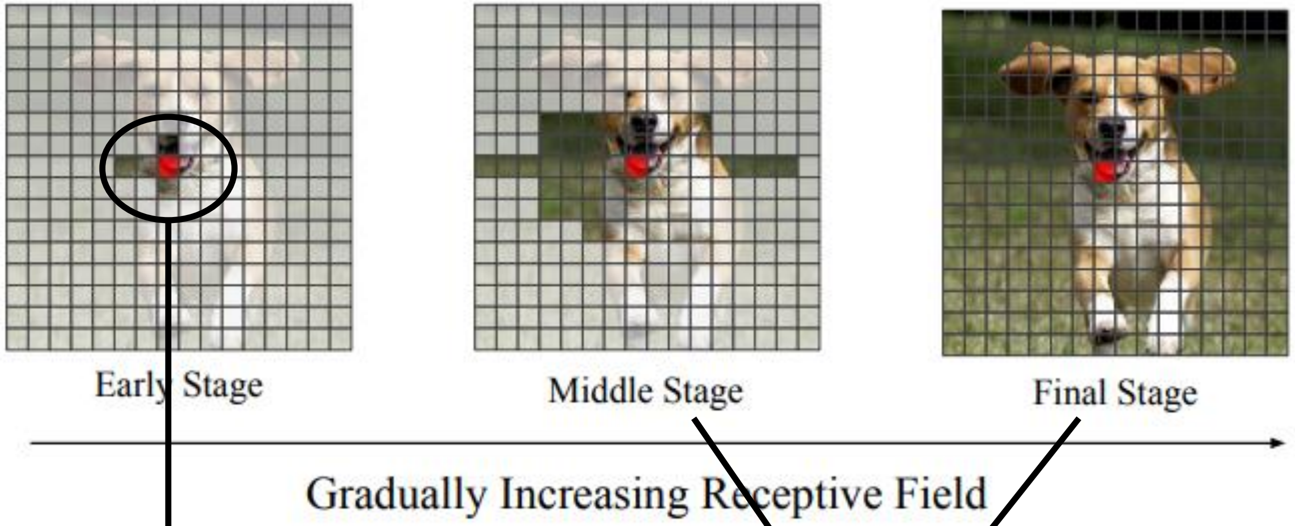
GAN loss + Supper Resolution 보조 task

MODEL	IS↑	FID↓
TRANSGAN + DA (*)	8.15± 0.14	19.85
(*) + MT-CT	8.20± 0.14	19.12

7. Locality-Aware Initialization for Self-Attention

Local Initialization without Convolution

Regularizer 역할



우선순위를 갖게 함

Provides necessary details

Provide More detail & noise

7. Locality-Aware Initialization for Self-Attention

Result

MODEL	IS↑	FID↓
TRANSGAN + DA (*)	8.15± 0.14	19.85
(*) + MT-CT	8.20± 0.14	19.12
(*) + MT-CT + LOCAL INIT.	**8.22± 0.12**	**18.58**

8. Scaling up to Large Model

Model

S : 384x384

M : 512 x 512

L : 768 x 768

XL : 1024x1024

Model	Depth	Dim	IS ↑	FID ↓
TransGAN-S	{5,2,2}	384	8.22 ± 0.14	18.58
TransGAN-M	{5,2,2}	512	8.36 ± 0.12	16.27
TransGAN-L	{5,2,2}	768	8.50 ± 0.14	14.46
TransGAN-XL	{5,4,2}	1024	$\mathbf{8.63 \pm 0.16}$	**11.89**

Number of Encoder Block

9. Comparison with State-of-the-art GANs

CIFAR-10

FID 2등!

METHODS	IS	FID
WGAN-GP (GULRAJANI ET AL., 2017)	6.49 ± 0.09	39.68
LRGAN (YANG ET AL., 2017)	7.17 ± 0.17	-
DFM (WARDE-FARLEY & BENGIO, 2016)	7.72 ± 0.13	-
SPLITTING GAN (GRINBLAT ET AL., 2017)	7.90 ± 0.09	-
IMPROVING MMD-GAN (WANG ET AL., 2018A)	8.29	16.21
MGAN (HOANG ET AL., 2018)	8.33 ± 0.10	26.7
SN-GAN (MIYATO ET AL., 2018)	8.22 ± 0.05	21.7
PROGRESSIVE-GAN (KARRAS ET AL., 2017)	8.80 ± 0.05	15.52
AUTOGAN (GONG ET AL., 2019)	8.55 ± 0.10	12.42
STYLEGAN V2 (ZHAO ET AL., 2020B)	**9.18**	**11.07**
TRANSGAN-XL	8.63 ± 0.16	11.89

STL-10

SOTA!!

IS, FID 1등!

METHODS	IS ↑	FID ↓
DFM (WARDE-FARLEY & BENGIO, 2016)	8.51 ± 0.13	-
D2GAN (NGUYEN ET AL., 2017)	7.98	-
PROBGAN (HE ET AL., 2019)	8.87 ± 0.09	47.74
DIST-GAN (TRAN ET AL., 2018)	-	36.19
SN-GAN (MIYATO ET AL., 2018)	9.16 ± 0.12	40.1
IMPROVING MMD-GAN (WANG ET AL., 2018A)	9.23 ± 0.08	37.64
AUTOGAN (GONG ET AL., 2019)	9.16 ± 0.12	31.01
ADVERSARIALNAS-GAN (GAO ET AL., 2020)	9.63 ± 0.19	26.98
TRANSGAN-XL	$\mathbf{10.10 \pm 0.17}$	**25.32**

감사합니다