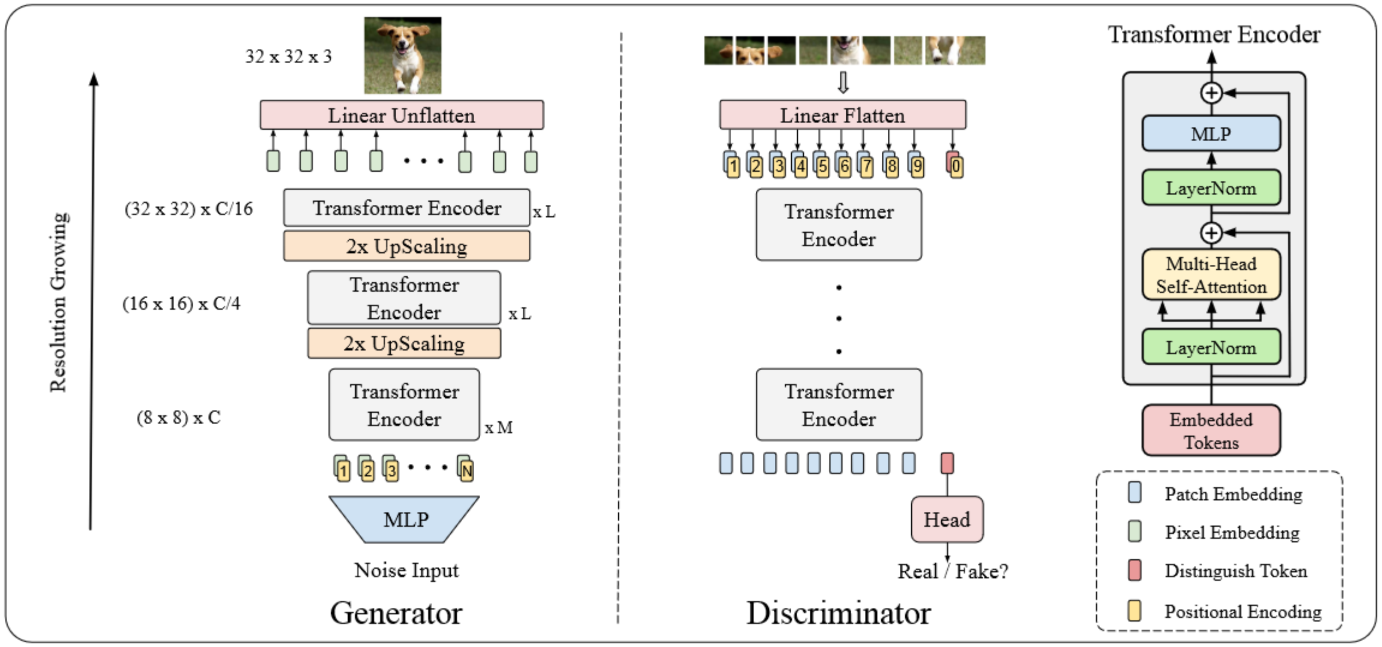
TransGAN: Two Transformers Can Make One Strong GAN

Yifan Jiang[1] Shiyu Chang[2] Zhangyang Wang[1]

CVPR 2021



[1]University of Texas at Austin,
[2]MIT–IBM Watson AI Lab

Presented by
Ahmet Sarigun

Figure from paper

Outline

- Introduction
- What is Transformers?
- Background and related work
- Methods
- Experiments
- Results
- Conclusion

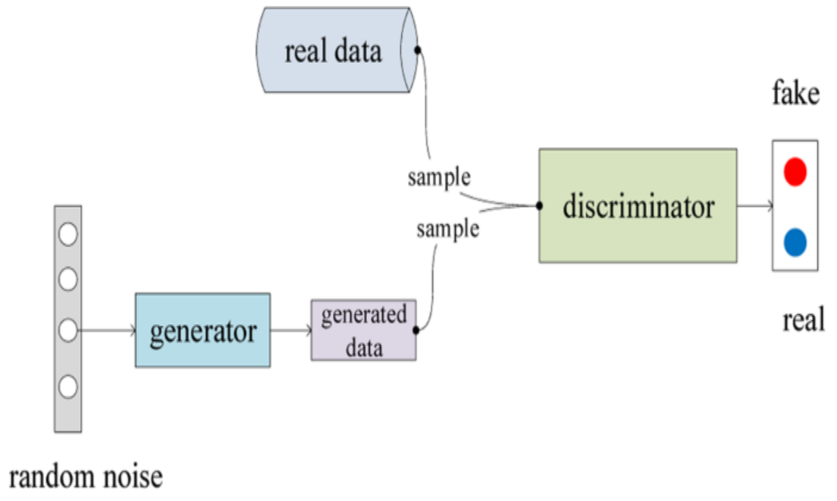


Figure from link

Introduction

- To learn the generator's distribution p_g over data x, firstly, input noise variables $p_z(z)$, defined.
- Representing a mapping to data space as $G(z;\theta_g)$, where G is a differentiable function represented by a multilayer perceptron with parameters θ_g.
- Second multilayer perceptron $D(x;\,\theta_d)$ that outputs a single scalar. $D(x)$ represents the probability that x came from the data rather than p_g.

$$\min_{G}\max_{D} V(D,G) = \mathbb{E}_{x\sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}\left[\log\left(1- D\left(G(z)\right)\right)\right]$$

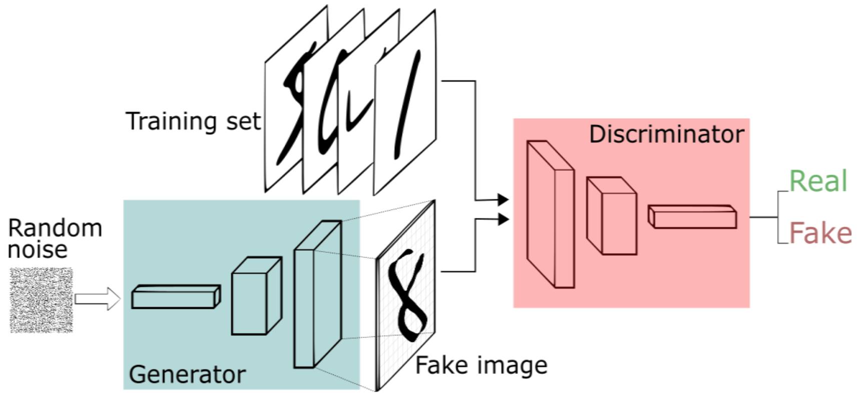


Figure from link

Introduction

❖ The original GAN used fully-connected networks and can only generate small images.

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

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

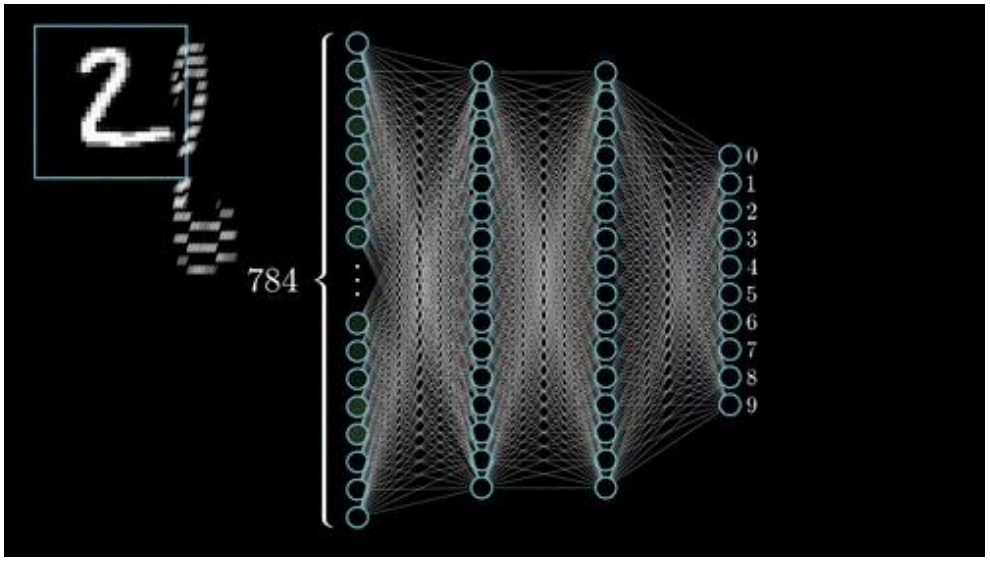


Figure from link

Introduction

- DCGAN (Radford et al., 2015) was the first to scale up GANs using CNN architectures, which allowed for stable training for higher resolution and deeper generative models.

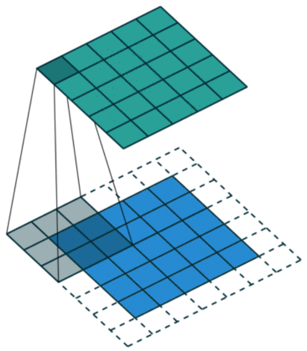


Figure from link

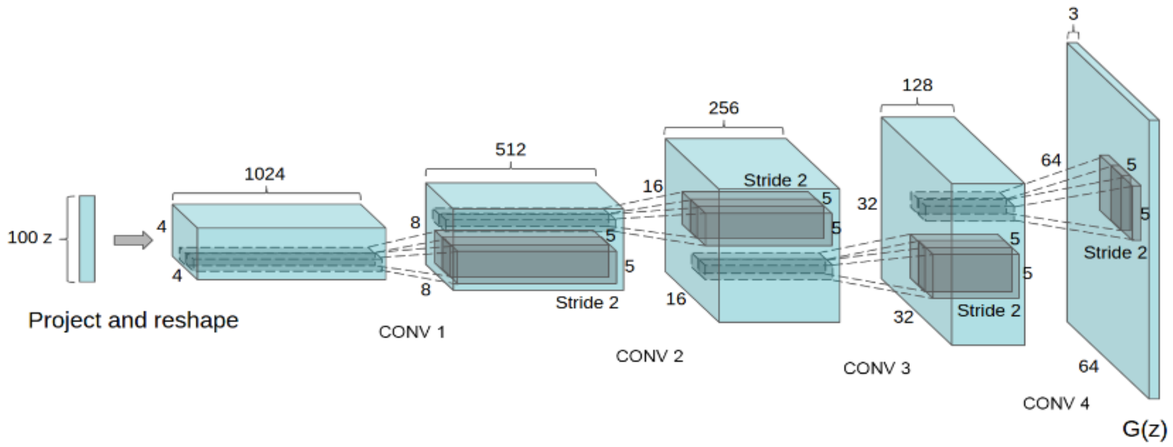


Figure 1: DCGAN generator used for LSUN scene modeling. A 100 dimensional uniform distribution Z is projected to a small spatial extent convolutional representation with many feature maps. A series of four fractionally-strided convolutions (in some recent papers, these are wrongly called deconvolutions) then convert this high level representation into a 64×64 pixel image. Notably, no fully connected or pooling layers are used.

Figure from link

Radford et al., 2015; Unsupervised representation learning with deep convolutional generative adversarial networks

Introduction

- In the computer vision domain, nearly every successful GAN relies on CNN-based generators and discriminators.
- Convolutions, with the strong inductive bias for natural images, crucially contribute to the appealing visual results and rich diversity achieved by modern GANs.
- Fundamentally, a convolution operator has a local receptive field, and hence CNNs cannot process long-range dependencies unless passing through a sufficient number of layers.
- However, that could cause the loss of feature resolution and fine details, in addition to the difficulty of optimization.

Can We Build a Strong GAN Completely Free of Convolutions?

❖ Not only arising from intellectual curiosity, but also of practical relevance.

❖ Inspired by the emerging trend of using Transformer architectures for computer vision tasks (Carion et al., 2020;Zeng et al., 2020; Dosovitskiy et al., 2020).

❖ Transformers (Vaswani et al., 2017; Devlin et al., 2018) have prevailed in natural language processing (NLP), and lately, start to perform comparably or even better than their CNN competitors in a variety of vision benchmarks.

Carion et al., 2020; End-to-End object detection with transformers
Zeng et al., 2020; Learning joint spatial-temporal transformations for video inpaiting
Dosovitsky et al., 2020; An image is worth 16x16 words: Transformers for image recognition at scale
Vaswani et al., 2017; Attention is all you need
Devlin et al., 2018; Bert: Pre-training of deep bidirectional transformers for language understanding

Can We Build a Strong GAN Completely Free of Convolutions?

- The charm of the transformer to computer vision lies in at least two-fold:

  I. it has strong representation capability and is free of human-defined inductive bias. In comparison, CNNs exhibit a strong bias towards feature locality, as well as spatial invariance due to sharing filter weights across all locations;

  II. the transformer architecture is general, conceptually simple, and has the potential to become a powerful “universal” model across tasks and domains (Dosovitskiy et al., 2020).It can get rid of many ad-hoc building blocks commonly seen in CNN-based pipelines (Carion et al., 2020).

Carion et al., 2020; End-to-End object detection with transformers
Dosovitsky et al., 2020; An image is worth 16x16 words: Transformers for image recognition at scale

What is Transformers?

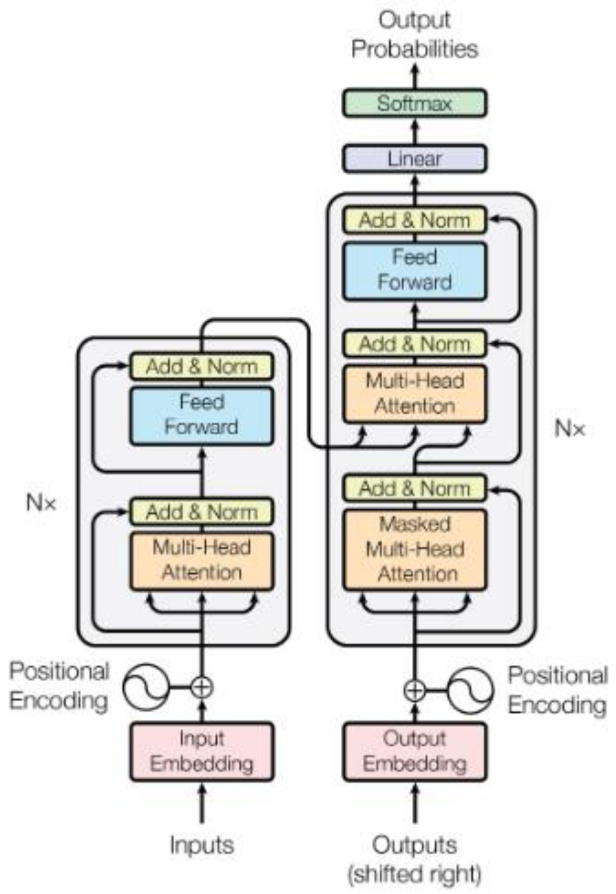


Figure 1: The Transformer - model architecture.

Figure from [link]

What is Transformers?

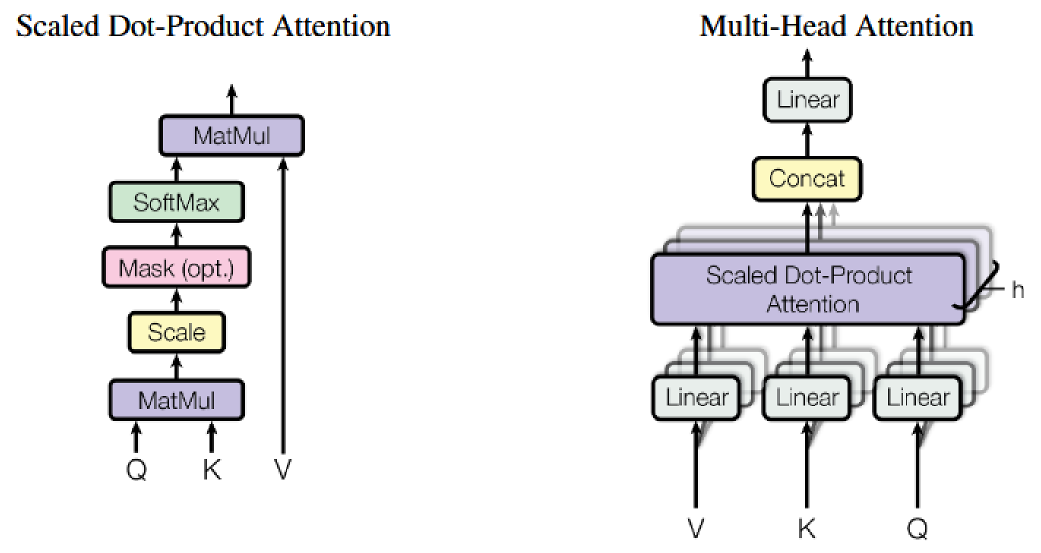


Figure 2: (left) Scaled Dot-Product Attention. (right) Multi-Head Attention consists of several attention layers running in parallel.

$$Attention(Q,K,V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Figure from link

What is Transformers?

- Multi-head attention allows the model to jointly attend to information from different representation subspaces at different positions. With a single attention head, averaging inhibits this.

$$MultiHead(Q,K,V) = Concat(head_1, .., head_h)W^o$$

$$\text{where } head_i = Attention(QW_i^Q, KW_i^K, VW_i^V)$$

- Where the projections are parameter matrices $W_i^Q \in \mathbb{R}^{d_{model} \times d_k}, W_i^K \in \mathbb{R}^{d_{model} \times d_k}, W_i^V \in \mathbb{R}^{d_{model} \times d_k}$ and $W_i^O \in \mathbb{R}^{d_{model} \times d_k}$.

Background and related work

Vision Transformer (ViT)

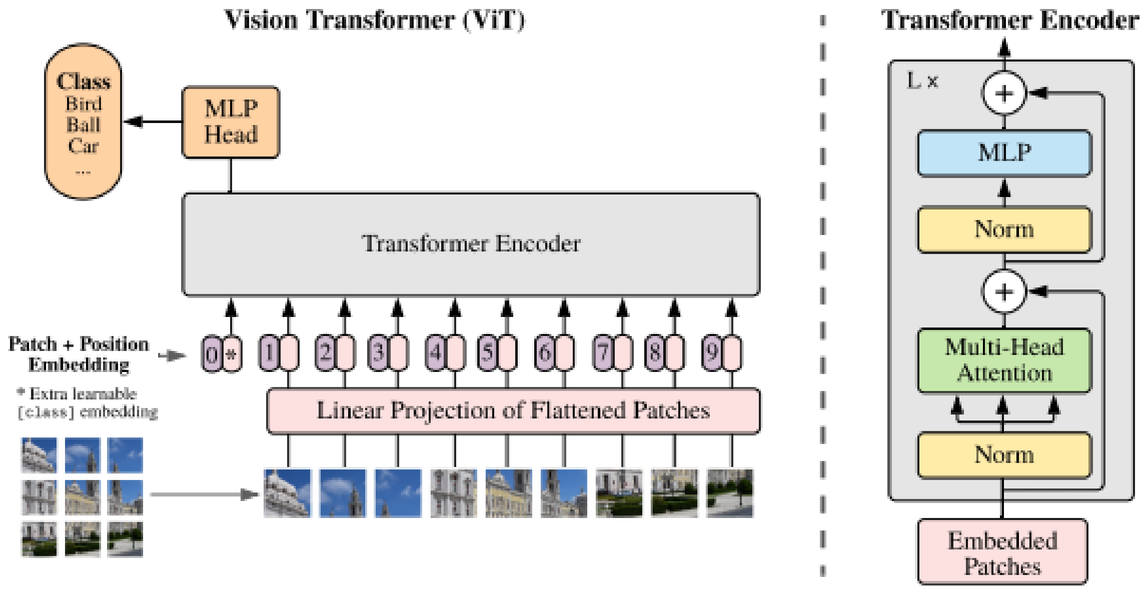
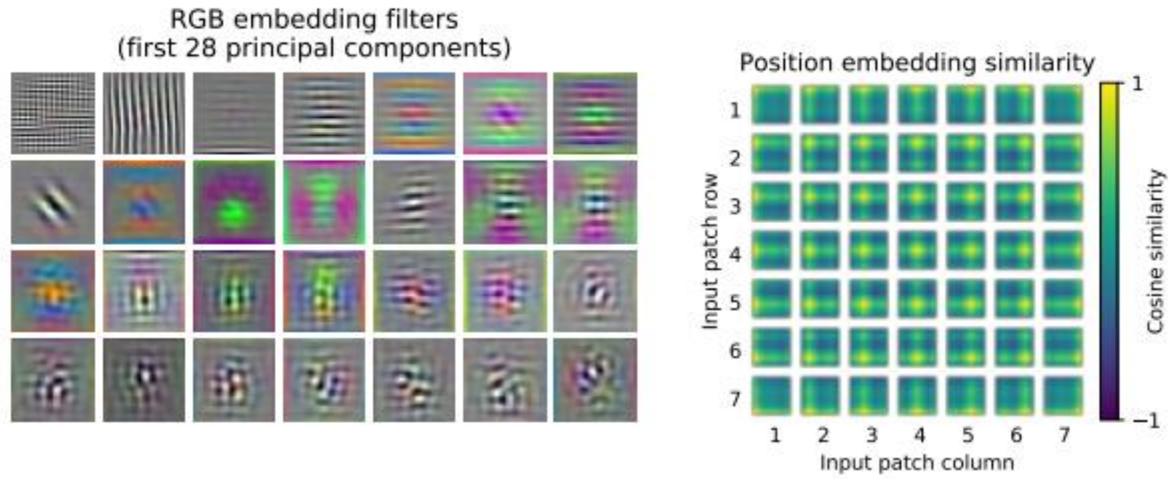


Figure 1: Model overview. We split an image into fixed-size patches, linearly embed each of them, add position embeddings, and feed the resulting sequence of vectors to a standard Transformer encoder. In order to perform classification, we use the standard approach of adding an extra learnable "classification token" to the sequence. The illustration of the Transformer encoder was inspired by Vaswani et al. (2017).

Figure from link

Background and related work

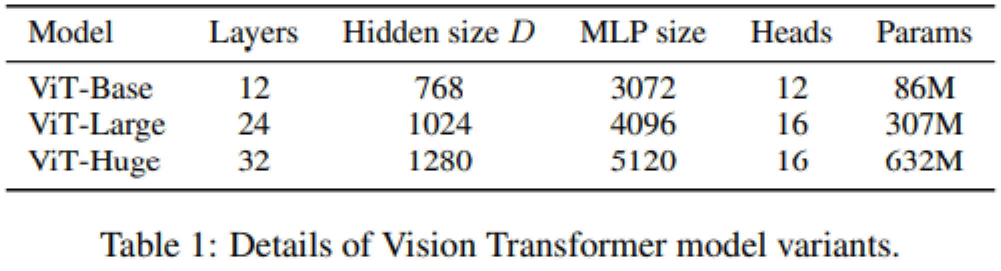
Vision Transformer (ViT)



Left: Filters of the initial linear embedding of RGB values of ViT-L/32.
Right: Similarity of position embeddings of ViT-L/32. Tiles show the cosine similarity between the position embedding of the patch with the indicated row and column and the position embeddings of all other patches.

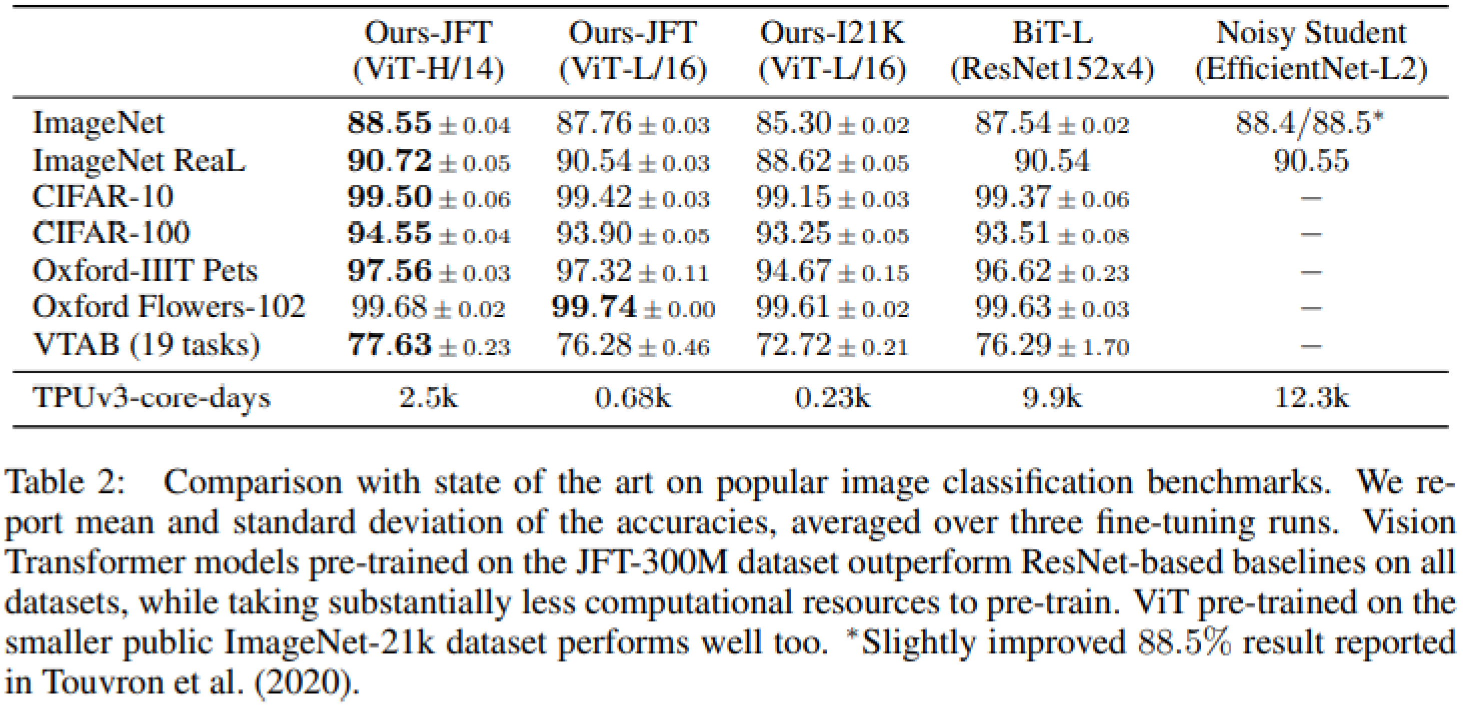
Tables from [link]()

Background and related work



Model	Layers	Hidden size D	MLP size	Heads	Params
ViT-Base	12	768	3072	12	86M
ViT-Large	24	1024	4096	16	307M
ViT-Huge	32	1280	5120	16	632M

Table 1: Details of Vision Transformer model variants.



	Ours-JFT (ViT-H/14)	Ours-JFT (ViT-L/16)	Ours-I21K (ViT-L/16)	BiT-L (ResNet152x4)	Noisy Student (EfficientNet-L2)
ImageNet	**88.55** ± 0.04	87.76 ± 0.03	85.30 ± 0.02	87.54 ± 0.02	88.4/88.5*
ImageNet ReaL	**90.72** ± 0.05	90.54 ± 0.03	88.62 ± 0.05	90.54	90.55
CIFAR-10	**99.50** ± 0.06	99.42 ± 0.03	99.15 ± 0.03	99.37 ± 0.06	–
CIFAR-100	**94.55** ± 0.04	93.90 ± 0.05	93.25 ± 0.05	93.51 ± 0.08	–
Oxford-IIIT Pets	**97.56** ± 0.03	97.32 ± 0.11	94.67 ± 0.15	96.62 ± 0.23	–
Oxford Flowers-102	99.68 ± 0.02	**99.74** ± 0.00	99.61 ± 0.02	99.63 ± 0.03	–
VTAB (19 tasks)	**77.63** ± 0.23	76.28 ± 0.46	72.72 ± 0.21	76.29 ± 1.70	–
TPUv3-core-days	2.5k	0.68k	0.23k	9.9k	12.3k

Table 2: Comparison with state of the art on popular image classification benchmarks. We report mean and standard deviation of the accuracies, averaged over three fine-tuning runs. Vision Transformer models pre-trained on the JFT-300M dataset outperform ResNet-based baselines on all datasets, while taking substantially less computational resources to pre-train. ViT pre-trained on the smaller public ImageNet-21k dataset performs well too. *Slightly improved 88.5% result reported in Touvron et al. (2020).

Tables from link

Methods

TransGAN: Two Transformers Can Make One Strong GAN

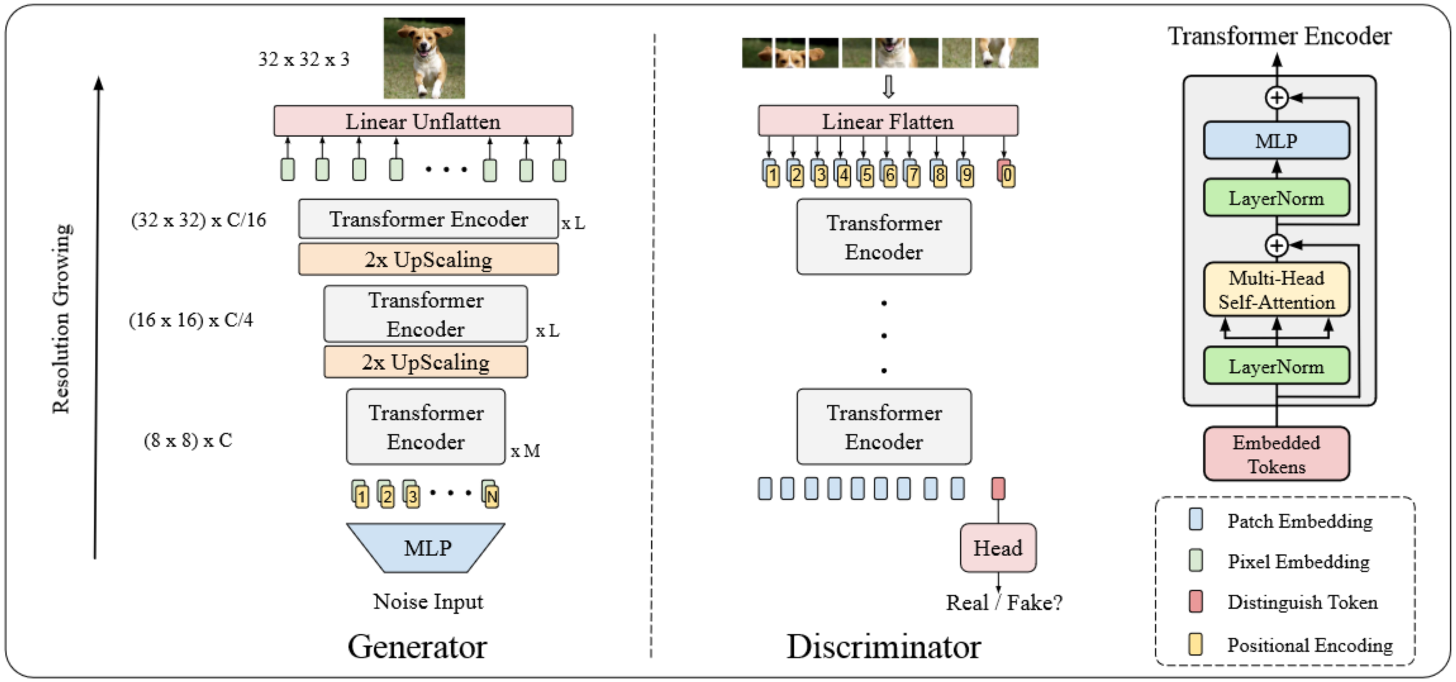


Figure 1. The pipeline of the pure transform-based generator and discriminator of TransGAN. Here $H = W = 8$ and $H_T = W_T = 32$. We show 9 patches for discriminator as an example while in practice we use 8×8 patches across all datasets.

Figure from paper

Up Scaling

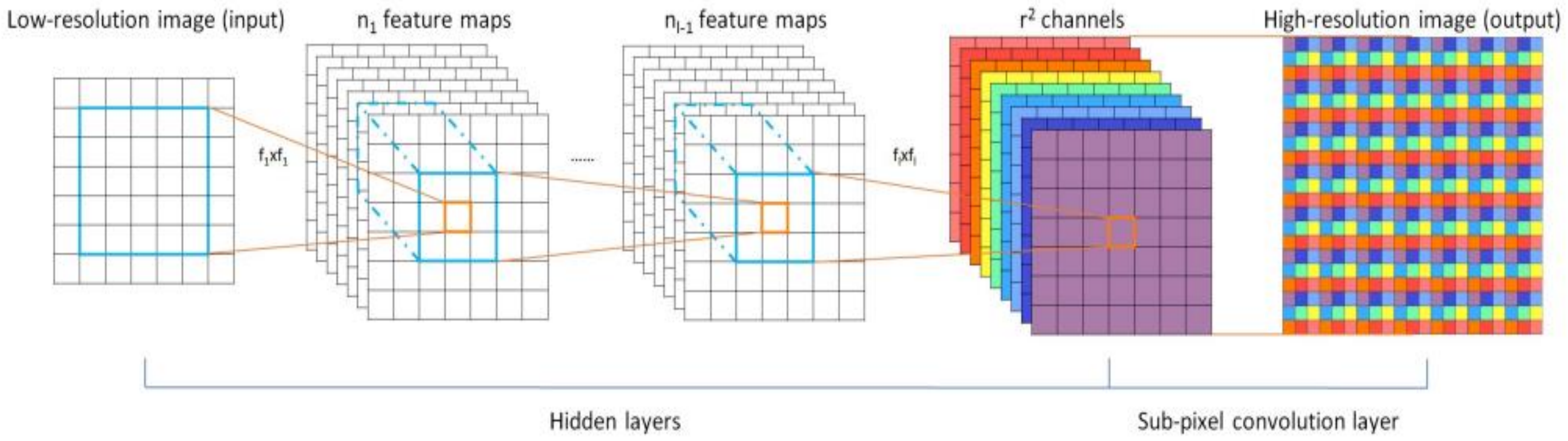


Figure 1. The proposed efficient sub-pixel convolutional neural network (ESPCN), with two convolution layers for feature maps extraction, and a sub-pixel convolution layer that aggregates the feature maps from LR space and builds the SR image in a single step.

Figure from link

Methods

TransGAN: Two Transformers Can Make One Strong GAN

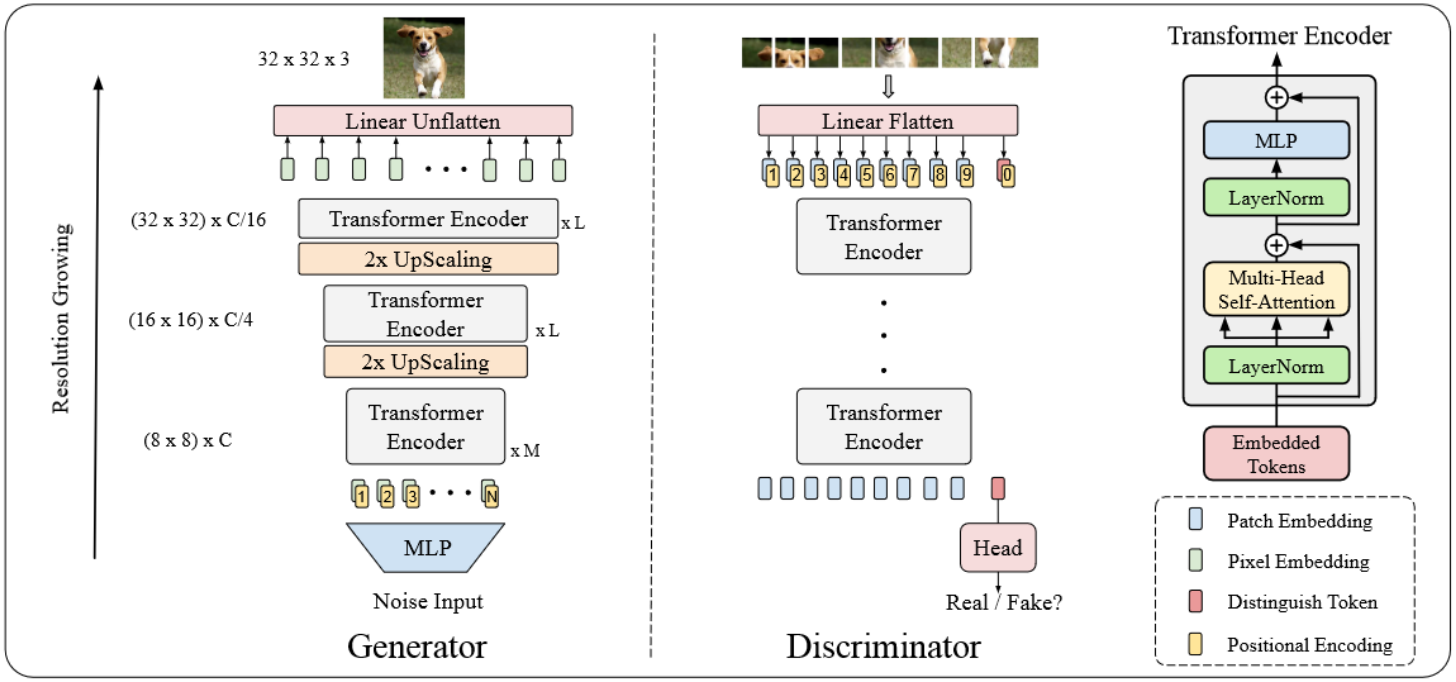


Figure 1. The pipeline of the pure transform-based generator and discriminator of TransGAN. Here $H = W = 8$ and $H_T = W_T = 32$. We show 9 patches for discriminator as an example while in practice we use 8×8 patches across all datasets.

Figure from paper

Data Augmentation

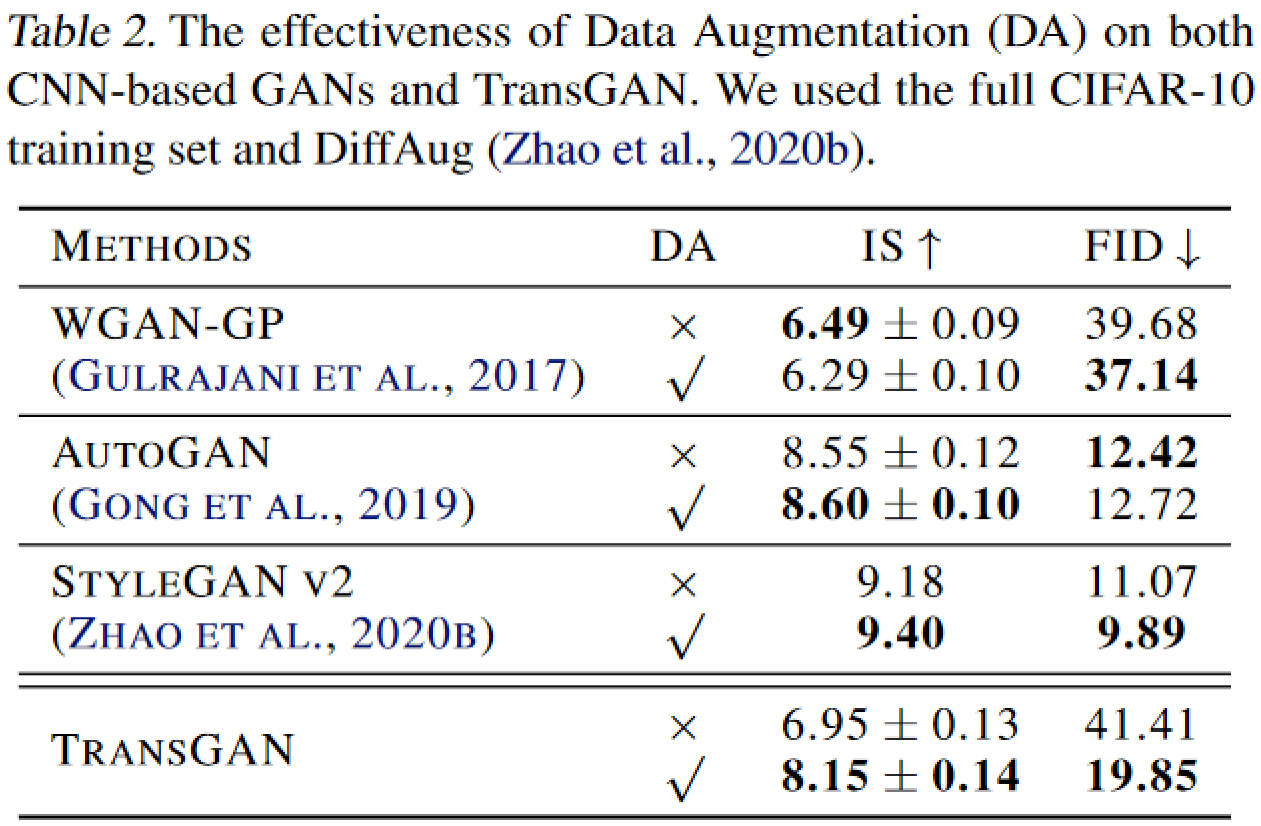


Table 2. The effectiveness of Data Augmentation (DA) on both CNN-based GANs and TransGAN. We used the full CIFAR-10 training set and DiffAug (Zhao et al., 2020b).

Methods	DA	IS ↑	FID ↓
WGAN-GP (Gulrajani et al., 2017)	×	**6.49** ± 0.09	39.68
	√	6.29 ± 0.10	**37.14**
AutoGAN (Gong et al., 2019)	×	8.55 ± 0.12	**12.42**
	√	**8.60 ± 0.10**	12.72
StyleGAN v2 (Zhao et al., 2020b)	×	9.18	11.07
	√	**9.40**	**9.89**
TransGAN	×	6.95 ± 0.13	41.41
	√	**8.15 ± 0.14**	**19.85**

Table from paper

Co-Training with Self-Supervised Auxiliary Task

- An auxiliary task of super resolution, in addition to the GAN loss. This task comes "for free", since we can just treat the available real images as high-resolution, and down sample them to obtain low-resolution counterparts.

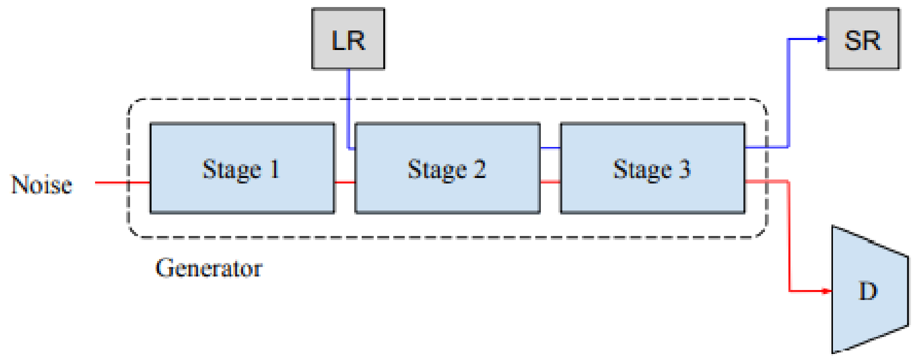


Figure 2. Co-training the transformer G with an auxiliary task of super resolution. "LR" and "SR" represent low-resolution input and high-resolution output respectively.

Figure from paper

Co-Training with Self-Supervised Auxiliary Task

- The generator loss is added with a auxiliary term $\lambda * L_{SR}$, where L_{SR} is the mean-square-error (MSE) loss and the coefficient λ is empirically set as 50.

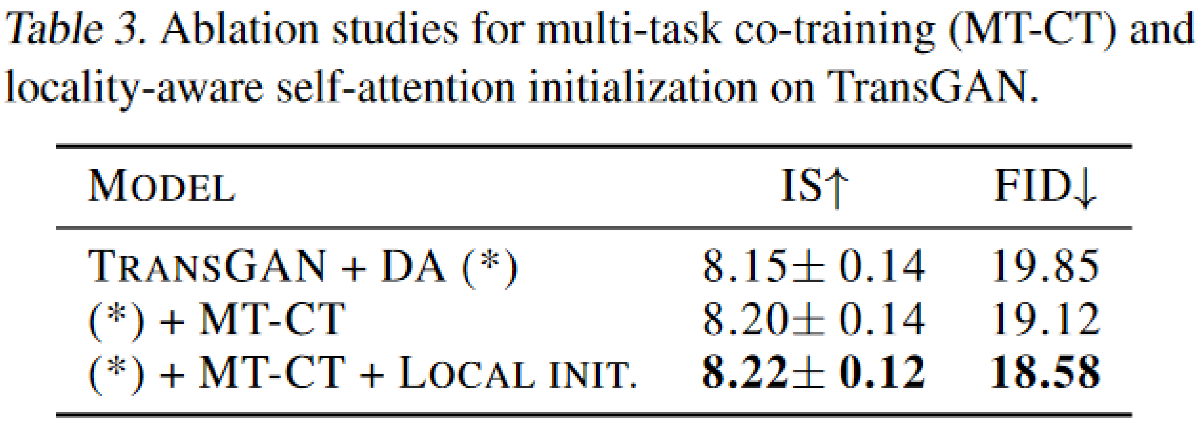


Table 3. Ablation studies for multi-task co-training (MT-CT) and locality-aware self-attention initialization on TransGAN.

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**

Table from paper

Locality-Aware Initialization for Self-Attention

- (Dosovitskiy et al., 2020) observed that transformers still tend to learn convolutional structures from images.
- Therefore, a meaningful question arises as, can we efficiently encode inductive image biases while still retaining the flexibility of transformers?
- Introduced a mask by which each query is only allowed to interact with its local neighbors that are not “masked”.

Locality-Aware Initialization for Self-Attention

- Different from previous methods (Daras et al., 2020; Parmar et al., 2018; Child et al., 2019; Beltagy et al., 2020) during training we gradually reduce the mask until diminishing it, and eventually the self-attention is fully global.

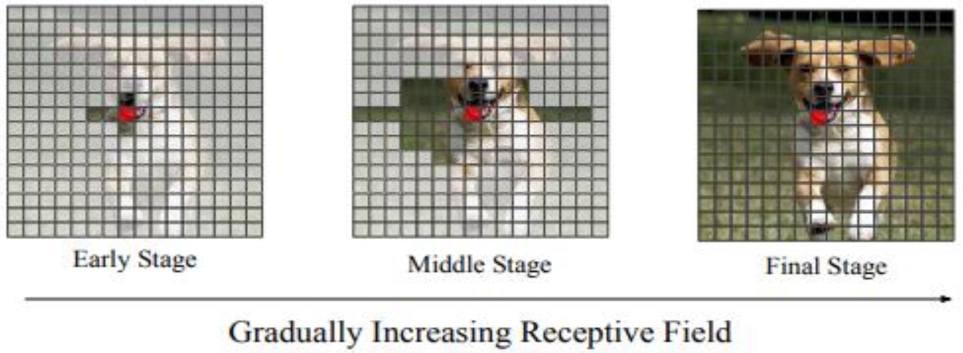


Figure 3. Locality-aware initialization for self-attention. The red block indicates a query location, the transparent blocks are its allowable key locations to interact with, and the gray blocks indicate the masked region. TransGAN gradually increases the allowable region during the training process.

Figure from paper

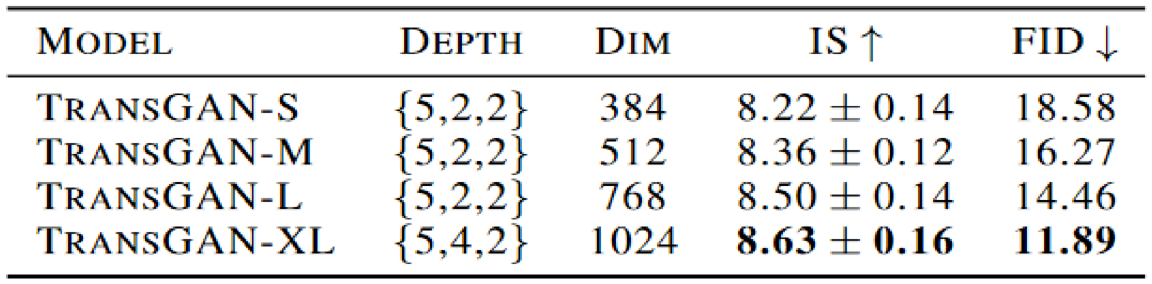
Locality-Aware Initialization for Self-Attention

- During training the mask gradually reduced until diminishing it, and eventually the self-attention is fully global.
- A localized self-attention (Parmar et al., 2018) is most helpful at the early training stage, but can hurt the later training stage and the final achievable performance.
- We consider this locality-aware initialization as a regularizer that comes for the early training dynamics and then gradually fades away (Golatkar et al., 2019).

Parmar et al., 2018; Image transformer
Golatkar et al., 2019 ; Time matters in regularizing deep networks: Weight decay and data augmentation affect early learning dynamics, matter little near convergence.

Scaling up to Large Models

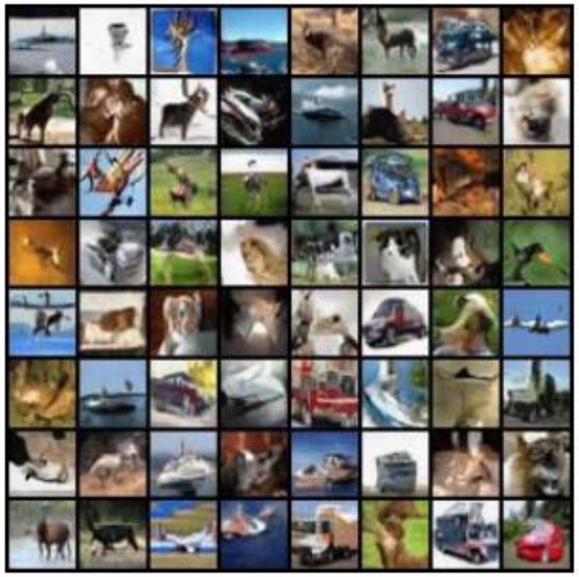
Table 4. Scaling-up the model size of TransGAN on CIFAR-10. Here “Dim” represents the embedded dimension of transformer and “Depth” is the number of transformer encoder block in each stage.



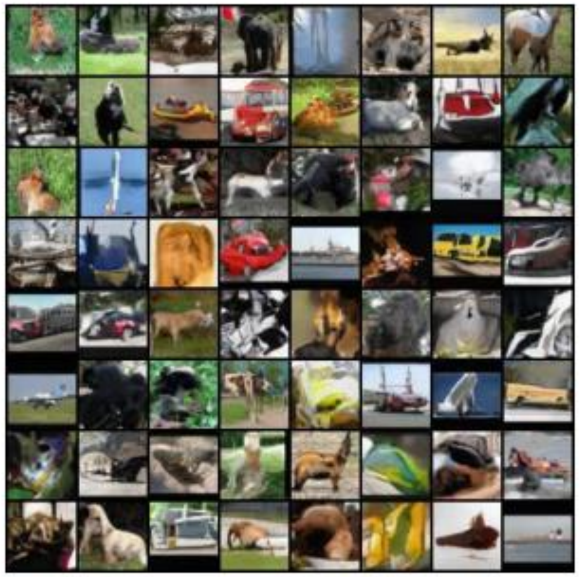
MODEL	DEPTH	DIM	IS ↑	FID ↓
TRANSGAN-S	{5,2,2}	384	8.22 ± 0.14	18.58
TRANSGAN-M	{5,2,2}	512	8.36 ± 0.12	16.27
TRANSGAN-L	{5,2,2}	768	8.50 ± 0.14	14.46
TRANSGAN-XL	{5,4,2}	1024	$\mathbf{8.63 \pm 0.16}$	**11.89**

Tables from paper

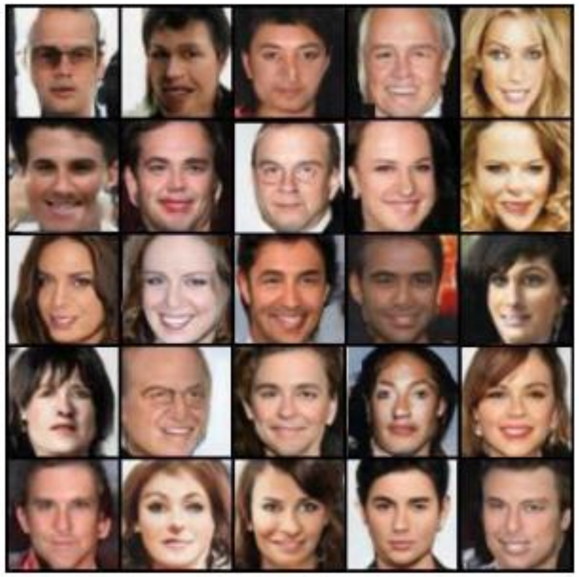
Experiments & Results



(a) Cifar-10



(b) STL-10



(c) CelebA 64 x 64

Figure 4. Visual Results of TransGAN on CIFAR-10, STL-10, and CelebA 64 $\times$ 64.

Figure from paper

Experiments & Results

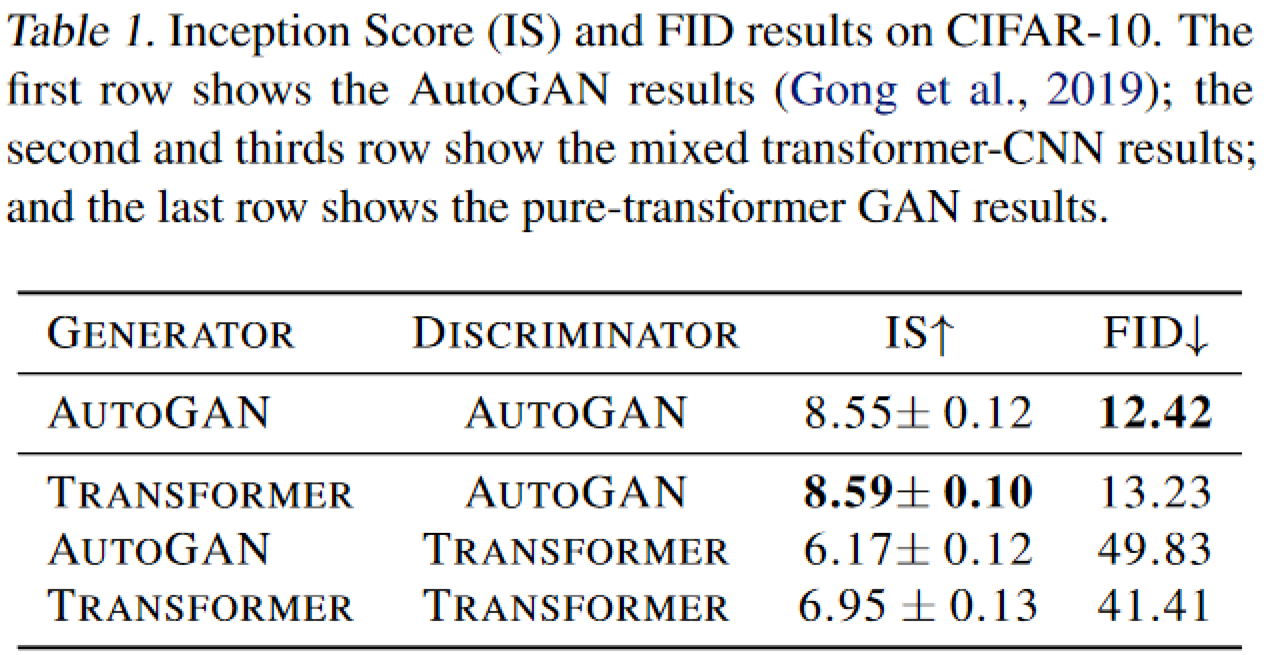


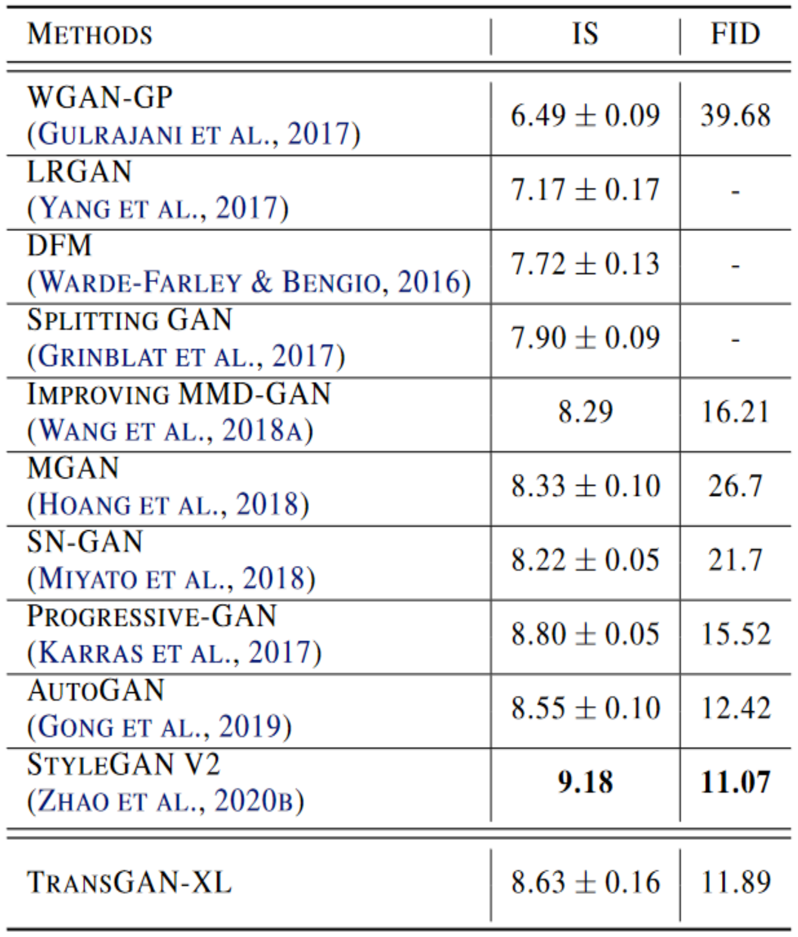
Table 1. Inception Score (IS) and FID results on CIFAR-10. The first row shows the AutoGAN results (Gong et al., 2019); the second and thirds row show the mixed transformer-CNN results; and the last row shows the pure-transformer GAN results.

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

Table from paper

Experiments & Results

Table 5. Unconditional image generation results on CIFAR-10.



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

Tables from paper

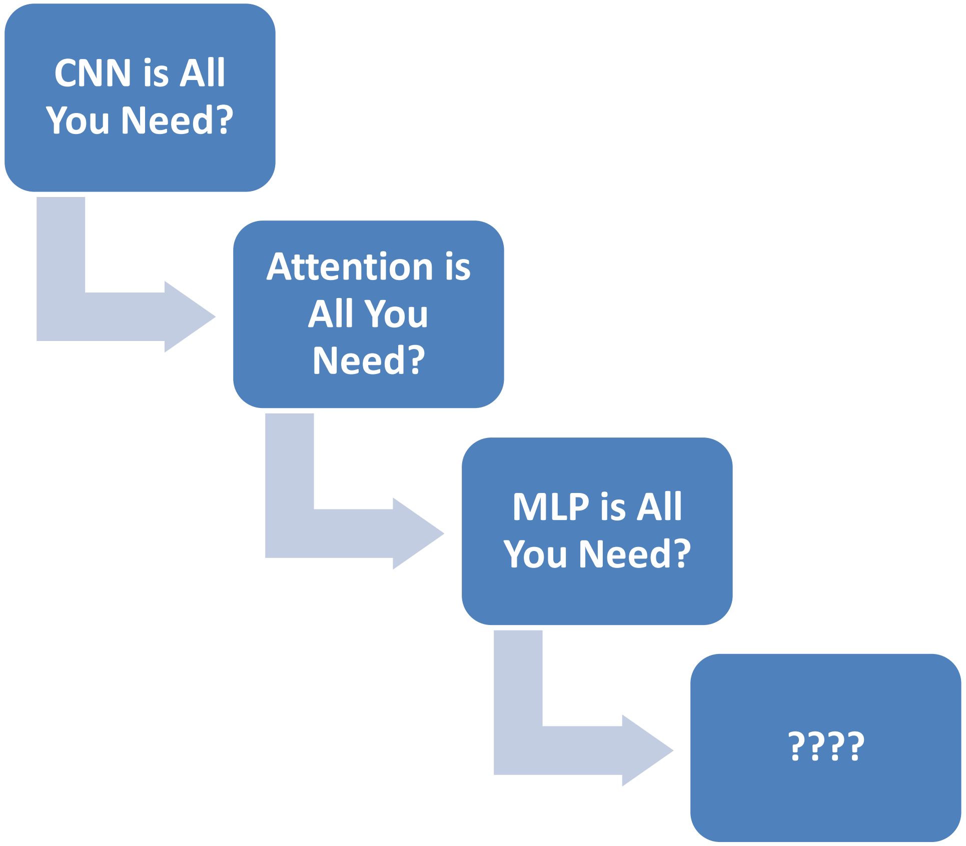
Conclusion

- **Model Architecture:**
  - build the first GAN using purely transformers and no convolution. To avoid over-whelming memory overheads,
  - Memory-friendly generator and a patch-level discriminator created, both transformer-based without bells and whistles.
  - Trans-GAN can be effectively scaled up to larger models.

Conclusion

- **Training Technique:**
  - to train TransGAN better, ranging from **data augmentation**, **multi-task co-training for generator with self-supervised auxiliary loss, and localized initialization for self-attention**.
- **Performance:**
  - TransGAN achieves highly competitive performance compared to current **state-of-the-art CNN**-based GANs.

Conclusion


CNN is All You Need?
Attention is All You Need?
MLP is All You Need?
????

Thank you for listening!

+/-/?/!