Super Pixel

1st Dhruv Kabariya
Under Prof. Mehul Raval
Ahmedabad University
Ahmedabad, India
dhruv.k1@ahduni.edu.in

2nd Shivam Thakker
Under Prof. Mehul Raval
Ahmedabad University
Ahmedabad, India
shivam.t1@ahduni.edu.in

3rd Pranav Gandhi
Under Prof. Mehul Raval
Ahmedabad University
Ahmedabad, India
pranav.g@ahduni.edu.in

***Abstract*—The primary aim of this project is to construct a high-resolution (HR) image from a corresponding low-resolution (LR) input image. The key pain point behind the super-resolution is to recover the finer texture details when super-resolved at large upscaling factor. We will be using TransGAN which is a GAN architecture which is completely free of convolutions, using only pure transformer-based architectures. It consists of a memory-friendly transformer-based generator that progressively increases feature resolution, and correspondingly a multi-scale discriminator to capture simultaneously semantic contexts and low-level textures. Also in order to the improve memory bottleneck further, and to scale up TransGAN to high-resolution, the new module of grid self-attention is introduced. Unique training method is also introduced including a series of techniques that can tackle the training instability issues of TransGAN, such as data augmentation, modified normalization, and relative position encoding generation. Specifically, TransGAN sets the new state-of-the-art inception score of 10.43 and FID of 18.28 on STL-10. It also achieves the inception score of 9.02 and FID of 9.26 on CIFAR-10, and 5.28 FID on CelebA 128×128. When it comes to higher-resolution (e.g. 256 × 256) generation tasks, such as on CelebAHQ and LSUN-Church, TransGAN continues to produce diverse visual examples with high fidelity and reasonable texture details.**

***Index Terms*—Generative Adversarial Network(GAN), Super Resolution, TransGAN, Grid self-attention, multi-scale discriminator**

I. Introduction

Using convolutional neural networks (CNNs) as GAN backbones is rarely challenged. The original GAN used fully-connected networks and can only generate small images. DCGAN was the first to scale up GANs using CNN architectures, which allowed for stable training for higher resolution and deeper generative models. Since then, in the computer vision domain, every successful GAN relies on CNN-based generators and discriminators. Most of the CNN based models focus on increasing the architecture design such as residual learning and dense connections. Using the above methods, the performance has increased significantly but it suffers from two basic problems. First the interactions between images and convolution kernels are content-independent. Means they are using same convolution kernel to restore different image regions which may not be the suitable choice. Second, under the principle of local processing, convolution is not effective for long-range dependency modelling.

Many things are tried to overcome above challenges and improve GAN's. Some of the research improving GANs examines their neural architectures which reported a large-scale study of GANs and observed that when serving as (generator) backbones, popular neural architectures perform comparably well across the considered datasets. Their ablation study suggested that most of the variations applied in the ResNet family resulted in very marginal improvements. Nevertheless, neural architecture search (NAS) was later introduced to GANs and suggests enhanced backbone designs are also important for improving GANs, just like for other computer vision tasks.

We will be using GAN completely free of convolutions, using only pure transformer-based architectures and also explaining it's architecture in this report further.

A. Abbreviations and Acronyms

- h - height
- w - width
- c - channel size or dimension
- p - patch size

II. Contribution

In the veiw of the challenges mentioned under problem formulation section in using CNN based GANs, pure transformer-based GAN architectures, called TransGAN is used. A common approach may directly stack multiple transformer blocks from raw pixel inputs. But the problem with that is that it would scale poorly due to memory explosion.

Instead, we start with a memory-friendly transformer-based generator by gradually increasing the feature map resolution in each stage. Improvements in the discriminator consists of multi-scale structure that takes patches of varied size as inputs, which balances between capturing global contexts and local details. It also enhances memory efficiency.

A new module called grid self-attention is introduced, that increases the memory bottleneck further when scaling up TransGAN to high-resolution generation (e.g. 256 × 256). Our contributions are below:-

- Novel Architecture Design: TransGAN network, a memory-friendly generator and a multi-scale discriminator, and is further equipped with a new grid self-attention mechanism. These architectural components are thoughtfully designed to balance memory efficiency, global feature statistics, and local fine details with spatial variances.

- New Training Method: Data augmentation, modifying layer normalization, and adopting relative position encoding, for both generator and discriminator are leveraged here.
- Performance and Scalability: TransGAN achieves high performance in comparision to current GANs. Specifically, it sets the new state-of-the-art inception score of 10.43 and FID score of 18.28 on STL-10. It also reaches competitive 9.02 inception score and 9.26 FID on CIFAR-10, and 5.28 FID score on CelebA 128 × 128, respectively. Meanwhile, we also evaluate TransGAN on higher-resolution (e.g., 256 × 256) generation tasks, where TransGAN continues to yield diverse and impressive visual examples.

III. Problem Formulation

- Creating a model which will convert Low resolution image into High resolution by taking care of every small pixel level detail.
- Minimizing the Mean Squared Error(MSE) between the produced High Resolution image and the original image.
- Designing a deep neural network architecture for generating High Resolution image
- 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 is inefficient, and could cause the loss of feature resolution and fine details, in addition to the difficulty of optimization.
- Given well-designed CNN-based architectures, training GANs is notoriously unstable and prone to mode collapse.
- Training vision transformers are also known to be tedious, heavy, and data-hungry. Hence we will try to develop a TransGAN network which is convolution free which will help in feature resolution and capturing semantic contexts and low-level textures.

IV. Network Model

As shown in figure 1 is the pipeline of the pure transform-based generator and discriminator of TransGAN. We take 256 × 256 resolution image generation task as a example to illustrate the main procedure. Here patch size p is set to 32 as an example for the convenience of illustration, while practically the patch size is normally set to be no more than 8×8 which will depend on dataset. To start with, we choose the transformer encoder as our basic block and try to make minimal changes. An encoder is a composition of two parts. The first part is constructed by a multi-head self-attention module and the second part is a feed-forward MLP with GELU non-linearity.

Memory-Friendly Generator

- The transformer encoders take embedding token words as inputs and calculate the interaction between each token recursively.
- If we use a low resolution image (e.g. 32 x 32) and then converting it into 1D which would be a 1024 long sequence which would be a heavy computation task and would prohibit the scalability of higher resolution. Hence inspired by CNN based GANs, we will iteratively upscale the resolution at multiple stages.
- As we can see in figure 1, our generator consists of multiple stages and each stage stacks several transformer blocks. We gradually increase the feature map resolution by stages until it meets the target resolution H × W.
- The generator takes the random noise as its input, and passes it through a multiple-layer perceptron (MLP) to a vector of length H0 × W0 × C. The vector is reshaped into a H0 × W0 resolution feature map (by default we use H0 = W0 = 8), each point a C-dimensional embedding. This "feature map" is next treated as a length-64 sequence of C-dimensional tokens, combined with the learnable positional encoding
- To scale up to higher-resolution images, we insert an upsampling module after each stage, consisting of a reshaping and resolution-upscaling layer.
- For lower-resolution stages (resolution lower than 64 × 64), the upsampling module firstly reshapes the 1D sequence of token embedding back to a 2D feature map $X_i \in$ H$_i$ × W_i × C and then adopts the bicubic layer to upsample its resolution while the embedded dimension is kept unchanged, resulting in the output of $2H_i$ × $2W_i$ × C .After that, the 2D feature map Xi is again reshaped into the 1D sequence of embedding tokens. For higher-resolution stages, we replace the bicubic upscaling layer with the pixelshuffle module, which upsamples the resolution of feature map by 2× ratio and also reduces the embedding dimension to a quarter of the input.
- We repeat multiple stages until it reaches the target resolution (H, W), and then we will project the embedding dimension to 3 and obtain the RGB image of H×W×3.

Multi-Scale Discriminator

- Here we are simply tokenizing the input image in a coarser patch-level, where each patch can be regarded as a "word".
- We observe that the patch splitting rule plays a crucial role, where large patch size sacrifices low-level texture details, and smaller patch size results in a longer sequence that costs more memory. The above dilemma motivates our design of multi-scale discriminator below.
- As shown in Figure 1(right), a multi-scale discriminator is designed to take varying size of patches as inputs, at its different stages. We firstly split the input images of H×W×3 into three different sequences by choosing different patch sizes (P , 2P , 4P). The longest sequence ($\frac{H}{P}$ x $\frac{W}{P}$) × 3 is linearly transformed to ($\frac{H}{P}$ x $\frac{W}{P}$) × $\frac{C}{4}$ and then combined with the learnable position encoding to serve as the input of the first stage , where $\frac{C}{4}$ is the embedded dimension size.

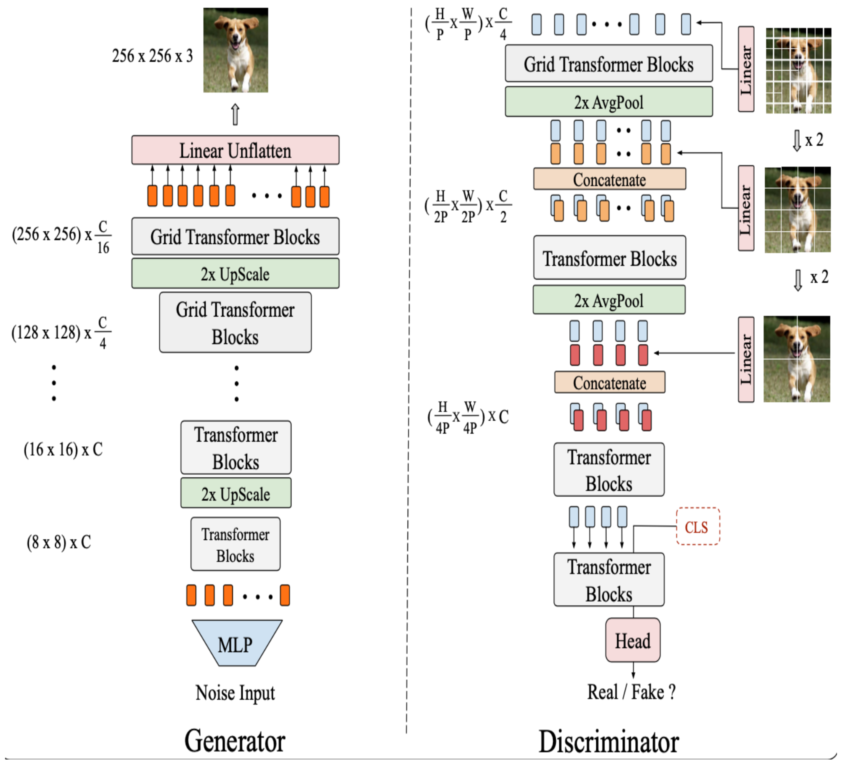

Fig. 1. Network Model TransGAN

- Similarly,the second and third sequences are linearly transformed to ($\frac{H}{2P}$ x $\frac{W}{2P}$) $\times$ $\frac{C}{4}$ and ($\frac{H}{4P}$ x $\frac{W}{4P}$) $\times$ $\frac{C}{2}$, and then separately joined into the second and third stages. Thus these three different sequences are able to extract both the semantic structure and texture details.
- We reshape the 1D-sentence to 2D feature map and adopt Average Pooling layer to down sample the feature map resolution, between each stage. And at the end of these blocks, a [cls] token is appended at the beginning of the 1D sequence and then taken by the classification head to output the real/fake prediction.

Grid Self Attention

- A Scalable Variant of Self-Attention for Image Generation.
- As we can see in figure 2, the grid self-attention partitions the full-size feature map into several non-overlapped grids, instead of calculating the correspondence between a given token and all other tokens. And then inside each local grid, token interactions are calculated.
- grid self-attention on high-resolution stages (resolution higher than 32 $\times$ 32) is added while keeping standard self-attention in low-resolution stages, as shown in figure 3, and hence balance between local details and global awareness is maintained.
- We replace Standard Self-Attention with Grid Self-Attention when the resolution is higher than 32 $\times$ 32 and the grid size is set to be 16 $\times$ 16 by default.

Training Method

- Data Augmentation: Due to removing human design bias, these transformer based architectures are very data hungry. To remove this roadblock, data augmentation was revealed as very useful. Just using differential augmentation with just three basic operators which are Translation, Cutout, and Color leads to surprising performance improvement for TransGAN. While CNN based GAN's does not benefit from these operators at all.

- Relative Position Encoding: It exploits lags instead of absolute positions. Single head of self attention layer:-

$$\text{Attention(Q, K, V)} = \text{softmax}(\ (\ \text{QK}^T/\sqrt{d_k} * V))$$

where Q,K,V of size (H×W)×C represent query, key, value matrices. H,W,C denotes the height, width, embedded dimension of the input feature map. The difference in coordinate between each query and key on H axis lies in the range of [(H 1), H 1], and similar for W axis. Relative position can be represented by a parameterized matrix by considering both H and W axis is of size (2H -1) x (2W - 1). The relative position encoding E is taken from matrix M and added to the attention map QKT as bias term, shown as below:-

$$\text{Attention(Q, K, V)} = \text{softmax}((\ (\ \text{QK}^T/\sqrt{d_k} + E)V)$$

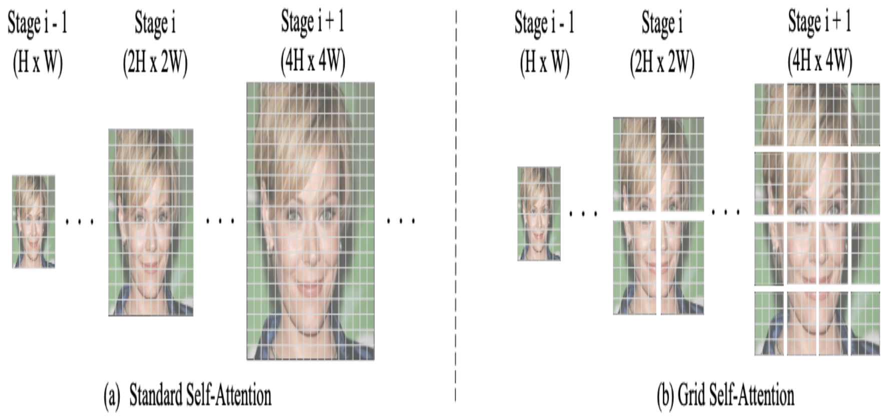

Fig. 2. Standard Self Attention Vs. Grid Self Attention

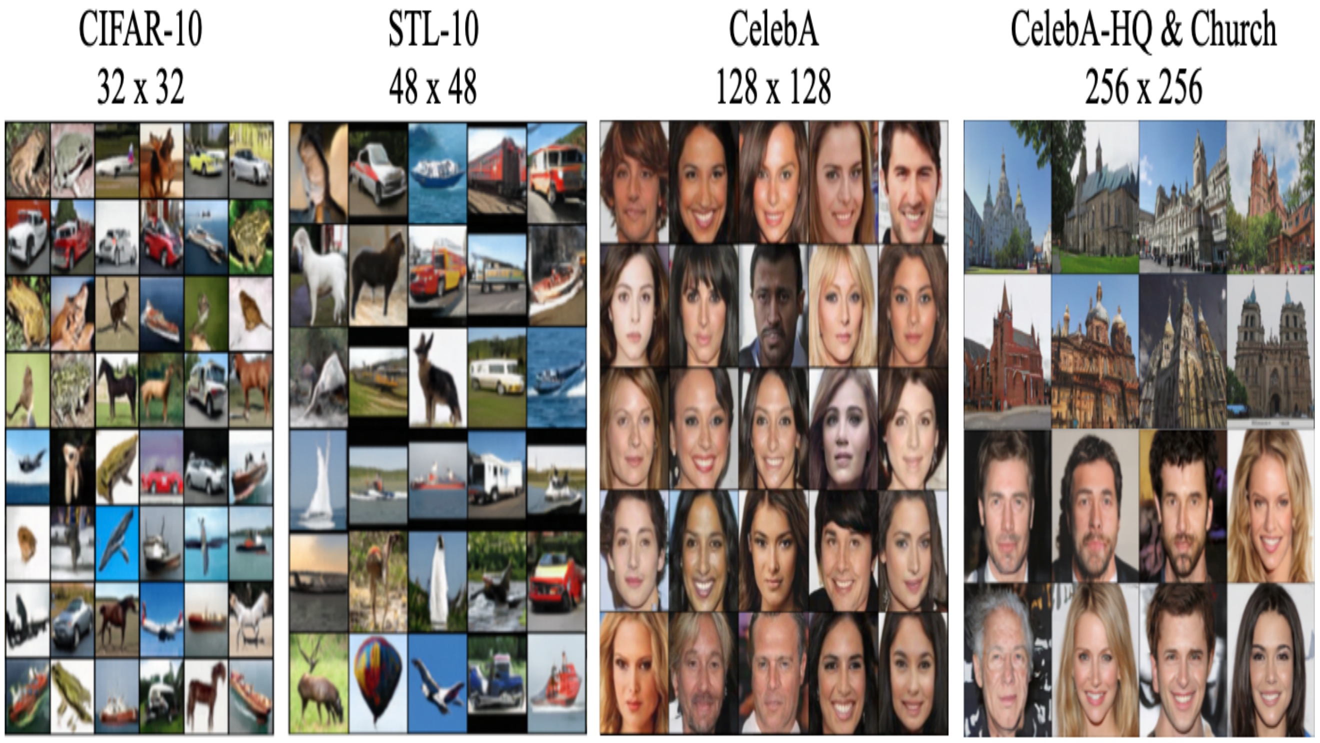

Fig. 3. Representative visual results produced by TransGAN on different datasets, as resolution grows from 32 × 32 to 256 × 256

Methods	**CIFAR-10**		**STL-10**		**CelebA**
	IS↑	FID↓	IS↑	FID↓	FID↓
WGAN-GP [1]	6.49 ± 0.09	39.68	-	-	-
SN-GAN [48]	8.22 ± 0.05	-	9.16 ± 0.12	40.1	-
AutoGAN [18]	8.55 ± 0.10	12.42	9.16 ± 0.12	31.01	-
AdversarialNAS-GAN [18]	8.74 ± 0.07	10.87	9.63 ± 0.19	26.98	-
Progressive-GAN [16]	8.80 ± 0.05	15.52	-	-	7.30
COCO-GAN [74]	-	-	-	-	5.74
StyleGAN-V2 [69]	9.18	11.07	10.21* ± 0.14	20.84*	5.59*
StyleGAN-V2 + DiffAug. [69]	**9.40**	9.89	10.31*± 0.12	19.15*	5.40*
TransGAN	9.02 ± 0.12	**9.26**	**10.43** ± 0.16	**18.28**	**5.28**

Fig. 4. Effectiveness of Data Augmentation on both CNN-based GANs and TransGAN. We use the full CIFAR-10 training set and DiffAug

Relative position encoding learns a stronger "relationship" between local contents, bringing important performance gains in large-scale cases.

- Modified Normalization: Instead of default use of layer normalization, we replace it with a token-wise scaling layer to prevent the magnitudes in transformer blocks from being too high.

V. Results

Fig. 2. shows that grid self attention mechanism wherein we divide the high resolution image into grids and then process it separately in order to get better results. As you can see in figure 3, the results produced by TransGAN on different datasets:- CIFAR-10, STL-10, CelebA and CelebA-HQ and Church. It takes care of every fine texture of the image and give us the super resolved image. Figure 4 shows the performance of TransGAN in comparision to other GAN's on CIFAR-10, STL-10 and CelebA datasets.

VI. Conclusion

Looking at the results and their comparison with the different models we can say that we can create and successfully use GAN which are free from any convolutions. And effectiveness of our TransGAN is also good in comparision to GAN's with convolution. Also this model has been checked across various datasets and has given promising results and a new way along building transformers, free of convolutions.

References

[1] Jiang, Y., Chang, S., Wang, Z. (2021). Transgan: Two pure transformers can make one strong gan, and that can scale up. Advances in Neural Information Processing Systems, 34.

[2] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Advances in neural information processing systems, pages 5998–6008, 2017.

[3] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. arXiv preprint arXiv:1710.10196, 2017.

[4] Emily Denton, Soumith Chintala, Arthur Szlam, and Rob Fergus. Deep generative image models using a laplacian pyramid of adversarial networks. arXiv preprint arXiv:1506.05751, 2015.

[5] Yunyang Xiong, Zhanpeng Zeng, Rudrasis Chakraborty, Mingxing Tan, Glenn Fung, Yin Li, and Vikas Singh. Nystromformer: A nystrom-based algorithm for approximating self-attention. arXiv preprint arXiv:2102.03902, 2021.

[6] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In Advances in neural information processing systems, pages 5767–5777, 2017.