TransGAN Critique - Do Two Transformers Make One Strong GAN?

Khoja, Azfar
azfar.khoja@umontreal.ca

Peplowski, Alexander
alexander.peplowski@umontreal.ca

Abstract

In this project, we evaluate the work of Jiang et al. [2021b], who propose TransGAN, a GAN framework which eliminates the use of convolutional operations. We evaluate the performance of the TransGAN model by reproducing our own version of the model and by verifying that the performance is similar to that of the original implementation. As well, we perform an ablation study which evaluates the performance of each of the training tricks suggested by the original paper. Finally, we extend the TransGAN model by proposing and evaluating our own conditional implementation. The project also makes publicly available our own version of the TransGAN training loop, which has not been published by the original authors at the time of writing.

1 Project Overview

1.1 Introduction and Related Works

The convolution operator with its local receptive field has a strong inductive bias of feature locality and translation invariance and is the reason why CNNs are state of the art models in computer vision tasks. However CNNs need a sufficient number of layers to understand long range dependencies, which in turn leads to loss of resolution and fine details and introduces the difficulties in optimization.

Inspired by the recent work of Dosovitskiy et al. [2020a], pure transformer architectures achieve comparable image classification accuracy to state of the art convolutional networks by treating an image as a sequence of 16×16 visual words. Their results suggest that although transformers might lack the inherent biases of CNNs, given enough data, their ability to capture global contextual representation trumps inductive bias.

Although encouraging, it's still unclear whether transformers can replace convolutions for all computer vision tasks such as detection, segmentation and in this case generation which poses a high demand for spatial coherency in structure, color, and texture (compared to classification).

To work in this direction, the TransGAN paper by Jiang et al. [2021b] proposes a pilot study in building a GAN completely free of convolutions, using only a pure transformer-based architecture. Since their inception in Goodfellow et al. [2014], nearly every successful GAN implementation relies on CNN-based generators and discriminators. Convolutional layers, which have a strong inductive bias for natural images, contribute to the appealing visual results and diversity.

The objective of this project is to determine whether the transformer-based GAN architecture provides the spatial coherency in structure, color, and texture needed for generating images. To do this, we evaluate training methods by performing our own ablation study which follows the one proposed by Jiang et al. [2021b] and compare results using smallest capacity architecture. Finally, we extend the paper, proposing our own conditional GAN model based off of the TransGAN design.

1.2 TransGAN Model Architecture

We use the pre-LN transformer encoder as described in Xiong et al. [2020] as the building blocks for our model. It consists of a self-attention module followed by a linear layer with GELU non-linearity. We apply layer normalization before and a residual connection after each part. (Figure 1 right).

A Memory-Friendly Generator: Trivially generating an image with pixels as tokens would incur a large quadratic cost (32x32 pixels = 1024 sequence length for CIFAR-10) due to self-attention. Instead, as shown in Figure 1, we start with a low resolution feature map and iteratively upsample it at each stage until it meets the target resolution . Specifically the input random noise is passed through a linear layer to obtain a vector of size $H \times W \times C$. The vector is reshaped into a $H \times W$ resolution feature map (defaults to $H = W = 8$) with each point in the feature map a C-dimensional input token to the transformer combined with learned positional encodings.

To increase resolution, we upsample the feature map resolution while reducing the embedding dimension using PixelShuffle [Shi et al., 2016]. So, the 1D sequence of token embeddings are reshaped back to a feature map $(H \times W) \times C$ which gets upsampled to $(2H \times 2W) \times C/4$. Finally, it is flattened back to a 1D sequence of embedding tokens of length 4HW and embedding size of C/4, combined with positional encodings and input to the next stage. So at each stage the resolution is increased fourfold with a proportional decrease in embedding dimension. This ensures that feature size remains constant throughout the transformer blocks and keeps the generator memory-efficient. The final linear layer reduces the embedding dimension to 3 and reshapes it to obtain an RGB image

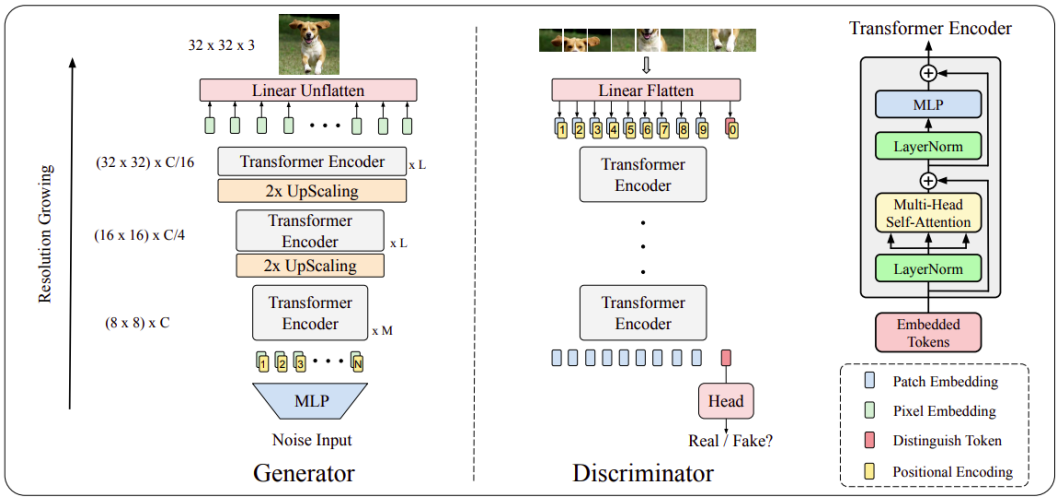

Figure 1: Left, **TransGAN Generator**. Right, **ViT Discriminator**. Figure from Jiang et al. [2021b]

ViT Discriminator: Since the discriminator is only required to differentiate between real and fake images, we use the vision transformer (ViT) as introduced in Dosovitskiy et al. [2020b]. As shown in Figure: 1 we semantically tokenize the input image into 8x8 patches which are converted into a 1D sequence of token embeddings through a linear flatten layer with an embedding dimension C that stays constant through all layers. Learned positional encoding are added to the token embeddings and the [CLS] token is inserted at the beginning of the sequence. After passing through subsequent layers, the output representation from the [CLS] token alone is used to classify images as real and fake.

1.3 Training Tricks

The vanilla TransGAN architecture naturally inherits the advantages of the global receptive field by using self-attention, but leads to poor performance (Table 2). To close the gap with existing CNN-based GANs the authors propose certain training "tricks" that considerably help to improve performance.

Data Augmentation: The latest work by Touvron et al. [2021] shows augmentations to be key in training data efficient vision transformers. Following this we study the impact of differential

augmentations by Zhao et al. [2020] in TransGAN. This involves introducing Translation, Cutout, Color, and horizontal flip augmentations at the Discriminator.

Co-Training with Self-Supervised Auxiliary Task: Motivated by the improvements in transformer performance brought upon by multi-task training in NLP (Devlin et al. [2019]), the co-training task of super-resolution prediction is introduced to GAN training. As shown in Figure 2, this involves providing down-sampled real images as low-resolution input to stage 2 of the generator. The upsampled output along with real image ground truths are used to calculate the auxiliary mean squared error loss for the generator. This loss is scaled by a factor of 50 and is shown to improve performance.

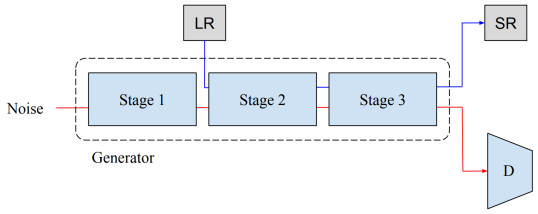

Figure 2: **Super-Resolution Co-Training Task**. From Jiang et al. [2021b]

Initialization for Self Attention: The idea is to encode inductive image biases by imitating the local receptive field of a convolutional kernel. This is done by introducing locality-aware initialization for the attention mask early during training and gradually reducing it over time so that self-attention is global. This will encourage pixels to attend to their neighborhood during first few epochs while still having the possibility of capturing global contextual information later.

Note: The left image in Figure 3 shows the illustration of the 2D attention mask from the original paper, however our investigation in the code reveals the mask is 1D and actually resembles the right image in Figure 3. So in reality, the mask forces pixels to attend to other pixels in a horizontal line. We followed up with the original authors to clarify this issue. They specify that the 2D illustration represents their expectation of self-attention magnitude after training but no experimental evidence is provided to justify the illustration.

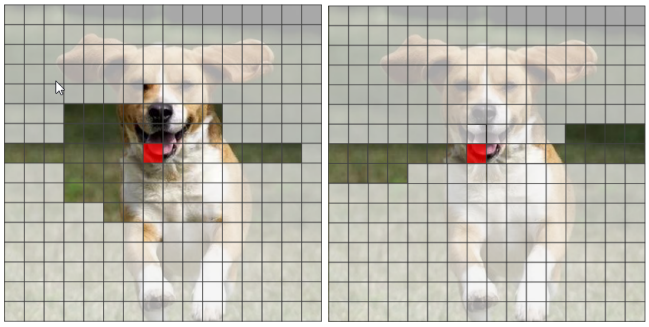

Figure 3: **Attention Initialization Masking**. Left, illustration of expected self-attention induced by an attention mask from Jiang et al. [2021b]. Right, our illustration of the actual attention mask implementation

Parameter Averaging: This technique maintains an exponential moving average of model parameter values. At inference time, the parameter values from the moving average are used for model evaluation. We observe that parameter averaging is used in the TransGAN public GitHub code base, but this trick from Salimans et al. [2016] is not mentioned in the paper. The update to the exponential moving average of parameter values is performed as per the equation:

$$\overline{\theta}_t = \alpha\theta_t + (1-\alpha)\overline{\theta}_{t-1}$$

Salimans et al. report that historical parameter averaging improves model performance. One possibility on why parameter averaging is effective is that it reduces bias towards the final minibatch seen through training, and adds more weight to the parameter state after previous minibatch iterations.

Conditional Loss: We extend TransGAN by creating a conditional variant based on the ACGAN implementation (Figure 4) by Odena et al. [2017]. The generator receives an embedding for the uniformly-sampled class label along with a random noise vector to synthesize an image. At the discriminator, the contextual embedding output of the [CLS] token is used for recognizing the correct source (real/fake) as well as identifying the correct class label.

This is achieved by adding the conditional log-likelihood term (defined below) to both the discriminator and the generator loss. This conditional variant learns a representation for the noise vector that is independent of class label.

$$L_C = \mathbb{E}[\log\ p(y = Y|X_{real})] + \mathbb{E}[\log\ p(y = Y|X_{fake})]$$

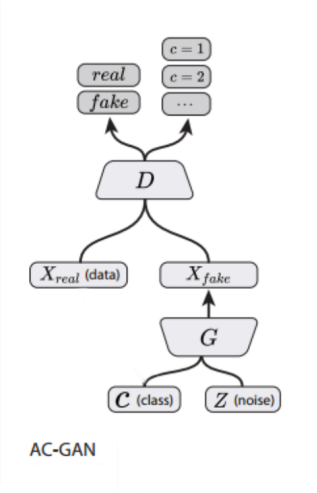

Figure 4: **ACGAN**

2 Experimental Setup

In this project, we propose that the results from the paper can be reproduced, but due to compute restrictions we focus our efforts on reproducing the training loop for the smallest configuration (TransGAN-S) with WGAN-GP loss (Gulrajani et al. [2017]) only on CIFAR-10. We also limit our training effort to a fixed number of training epochs. At the time of writing, the code for the training loop has not been published. However, the checkpoint for biggest configuration (TransGAN-XL) trained on the CIFAR-10 is available from Jiang et al. [2021a]. Table 1 highlights the difference between the models.

Ablation study: Recreating the original training loop described by the authors entails the implementation of each of the training tricks described previously. Following a successful baseline TransGAN-S implementation, we continue to implement all training enhancements proposed by the paper. To verify each training trick is correctly implemented and improves performance, we perform an ablation study adding each trick one at a time and limit the training effort to 150 generator epochs per experiment to

	Hyperparameters			
	Transformer Blocks / Stage			Emb. Size
Config	Stage 1	Stage 2	Stage 3	
TransGAN-S	5	2	2	384
TransGAN-XL	5	4	2	1024

Table 1: Comparison of TransGAN-S and TransGAN-XL architecture differences

reduce compute effort. However, for the complete model implementation, we do not restrict training effort.

Extension to Conditional Model As part of our ablation study, we extend the original work by modifying the architecture and the training loop to create a conditional variant of TransGAN, capable of generating images for a particular class. We continue to measure the overall performance of the model with these changes.

2.1 Practical Considerations

Collaboration Strategy GitHub is the primary platform used for code collaboration between the authors. As well, Overleaf is used for preparing reports and presentations. Our public Git repository is published on the GitHub website (Khoja and Peplowski [2021]).

Compute Resources We plan on using the smallest capacity model, TransGAN-S, for the basis of our experiments and limit training epochs to ensure that they can be completed in a timely manner.

We use Google Colab as a development and training platform as it provides free GPU resources with a constraint of 15GB RAM and a maximum time allocation of 12 hours.

In the course of this project, we also used the free credits available for new accounts on Google Cloud compute services for training the complete TransGAN-S model.

3 Results

Having sucessfully implemented the complete TransGAN-S conditional and unconditional models, we will analyze their results. In Figure 5 we compare the qualitative results of our models with the original author's results, and with samples from the training dataset. We observe that the images generated by our unconditional TransGAN-S is similar to the author's TransGAN-XL model. As well, for our conditional TransGAN-S model, we observe good intra-class diversity and inter-class separation. That is, there is a clear distinction when comparing images generated from different classes, and generated images within the same class seem distinct each other as well.

In Table 2 we show the results of our ablation study. This table includes the results from the original authors, our results with limited training epochs, and our unrestricted training run.

We observe that our results match those of the original authors. That is, we see that as each training trick is added, model performance improves. Note that, in the original paper, the authors did not discuss the impact of parameter averaging. That is, no ablation study on parameter averaging was performed and we presume that parameter averaging was included in all experiments. In our ablation study, we observe that parameter averaging plays an important role in final model performance.

Finally, we find that the conditional variant of TransGAN provides the best IS score overall, but the FID is no better than the unconditional model. Hence TransGAN can be extended to control class generation without impacting the baseline performance.

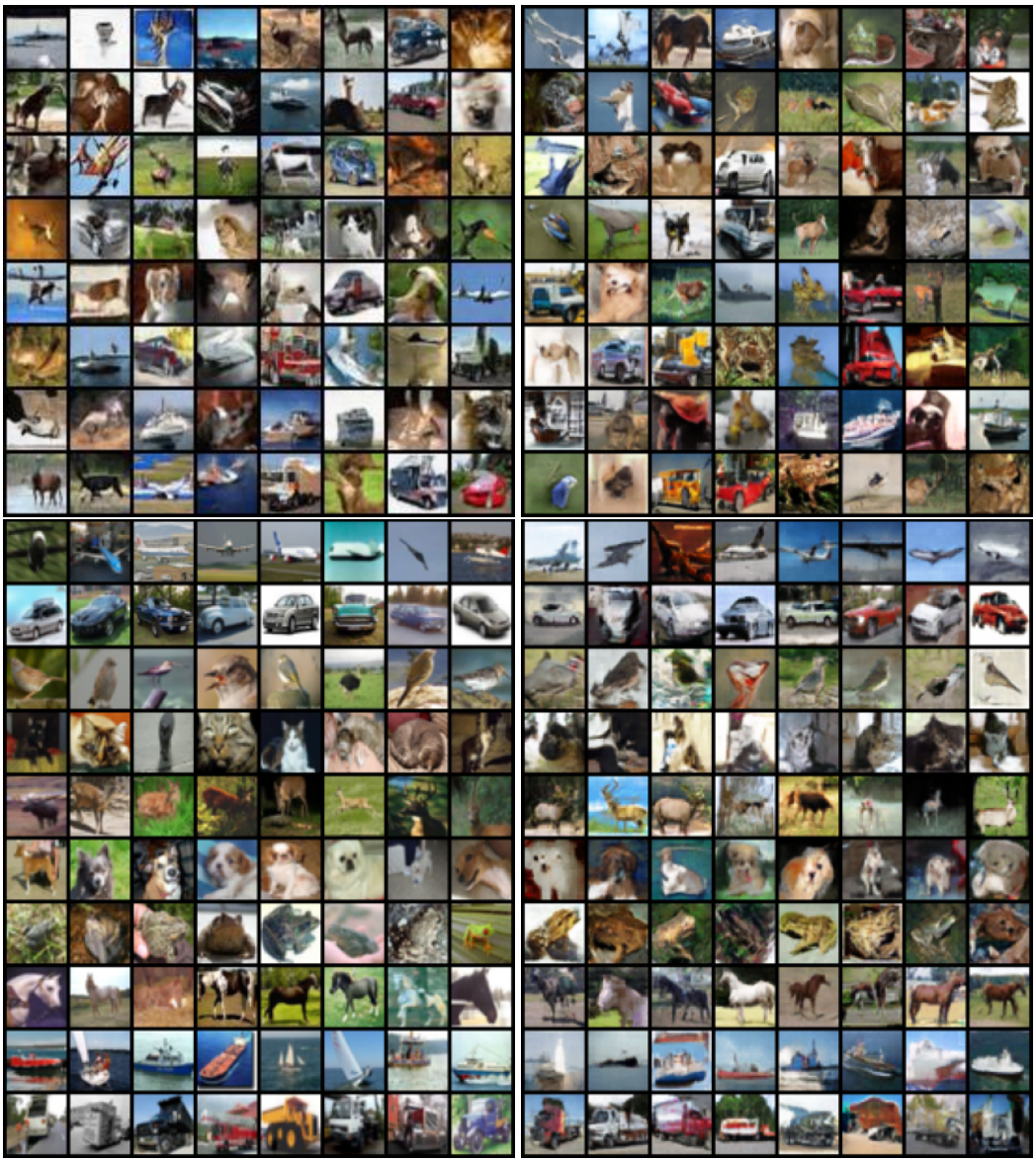

Figure 5: **Final Model Results** Clockwise from top-left: 1) TransGAN-XL at FID = 11.07 trained on CIFAR-10 from Jiang et al. [2021b]; 2) Our unconditional implementation of TransGAN-S at FID = 22.59; 3) Our conditional implementation of TransGAN-S at FID = 32.16; 4) Training images from CIFAR-10 dataset.

4 Conclusions

Looking at the results from our ablation study, we see that all training tricks proposed improve performance. Based on the magnitude of the performance improvement, data augmentation provides the most benefit, followed by parameter averaging. This supports the original author's suggestion along with [Dosovitskiy et al., 2020b] that transformer architectures need much more data to model spatial coherency required for images.

The original authors used four V100 GPUs and 2 days of training time for their TransGAN implementation. We demonstrate the implementation feasibility of the original paper by obtaining similar results (in terms of FID score and qualitative results) with fewer compute resources and using a lower-capacity model. Because not all hyperparameters are published, we propose that with enough compute for a thorough hyperparameter search we can reproduce the original results.

	Model Performance (FID/IS Scores)					
	Theirs		Ours[2]		Ours[3]	
Configuration	FID	IS	FID	IS	FID	IS
Vanilla TransGAN-S	41.41	$6.95 \pm .13$	77.84	$6.01 \pm .20$	-	-
Above + Data Augmentation	19.85	$8.15 \pm .14$	40.78	$6.48 \pm .18$	-	-
Above + Co-Training Task	19.12	$8.20 \pm .14$	40.10	$7.30 \pm .26$	-	-
Above + Attention Masking	**18.58**	**8.22 ± .12**	37.70	$7.32 \pm .17$	-	-
Above + Parameter Averaging	**18.58**[1]	**8.22 ± .12**[1]	33.49	$7.40 \pm .21$	**22.59**	**8.14 ± .25**
Above + Conditional Loss	-	-	34.52	**8.27 ± .16**	-	-

Table 2: Ablation study of TransGAN-S training configurations using CIFAR-10. Note 1: No ablation study on parameter averaging in original paper. Parameter averaging is included in all of the original author's trials. Note 2: With training limited to 150 generator epochs. Note 3: Without training limit.

Finally, we show that TransGAN can also be conditioned to generate images for a required class. By evaluating the qualitative results, we find that we have good results even when limiting training effort. For future work, we propose extending the training effort of the conditional TransGAN model to measure the benefit of adding the conditional loss, and to find a good hyperparameter weight associated with this loss.

In the course of this work, we discover that some items were left out from the original paper such as a discussion of parameter averaging which plays an important role in the final performance and details on the attention initialization. We clarify the difference between the authors' "expected" attention weights and the actual mask used to generate these weights.

Overall, this study showcases "TransGAN", a GAN framework based on pure transformers with carefully designed training techniques to successfully generate good quality images and provides an encouraging starting point for transformer architectures in computer vision. It also motivates the idea of transformers being "universal models" capable of being applied to diverse machine learning applications.

4.1 Acknowledgements

We follow the work of Jiang et al. [2021b]. Our contributions include:

- Development of a new version of the (unpublished) TransGAN training loop.
- Validation of the training tricks and ablation study for the TransGAN-S model.
- Extension of the TransGAN model by creating a conditional variant.
- Clarification of missing or unclear statements made in the original paper.

Credit goes to the original authors for providing a starting point for this project by releasing their work on GitHub which includes PyTorch code that implements the TransGAN model architecture. They also make some portions of the TransGAN training code available and provide some hints as to which hyperparameters were used in their experiments. We credit all respective authors for any code imported in our work. Our repository is available: here

References

J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2020a.

A. Dosovitskiy, L. Beyer, et al. An image is worth 16x16 words: Transformers for image recognition at scale, 2020b.

I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial networks, 2014.

I. Gulrajani, F. Ahmed, et al. Improved training of wasserstein gans, 2017.

Y. Jiang, S. Chang, and Z. Wang. Transgan: Two transformers can make one strong gan. `https://github.com/VITA-Group/TransGAN`, 2021a.

Y. Jiang, S. Chang, et al. Transgan: Two transformers can make one strong gan, 2021b.

A. Khoja and A. Peplowski. Transgan training loop. `https://github.com/azfarkhoja305/GANs`, 2021.

A. Odena, C. Olah, et al. Conditional image synthesis with auxiliary classifier gans, 2017.

T. Salimans, I. J. Goodfellow, et al. Improved techniques for training gans, 2016. URL `http://arxiv.org/abs/1606.03498`.

W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network, 2016.

H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou. Training data-efficient image transformers and distillation through attention, 2021.

R. Xiong, Y. Yang, D. He, K. Zheng, S. Zheng, C. Xing, H. Zhang, Y. Lan, L. Wang, and T.-Y. Liu. On layer normalization in the transformer architecture, 2020.

S. Zhao, Z. Liu, et al. Differentiable augmentation for data-efficient gan training, 2020.