STransGAN: An Empirical Study on Transformer in GANs

Anonymous authors
Paper under double-blind review

snowbird Maltese dog mushroom valley

Figure 1: Left: uncurated 256×256 results generated for FFHQ with our STrans-G. Right: 128×128 conditional samples from our STrans-G with AdaBN-T injecting IMAGENET class information.

Abstract

Transformer becomes prevalent in computer vision, especially for high-level vision tasks. However, deploying Transformer in the generative adversarial network (GAN) framework is still an open yet challenging problem. In this paper, we conduct a comprehensive empirical study to investigate the intrinsic properties of Transformer in GAN for high-fidelity image synthesis. Our analysis highlights the importance of feature locality in image generation. We first investigate the effective ways to implement local attention. We then examine the influence of residual connections in self-attention layers and propose a novel way to reduce their negative impacts on learning discriminators and conditional generators. Our study leads to a new design of Transformers in GAN, a convolutional neural network (CNN)-free generator termed as *STrans-G*, which achieves competitive results in both unconditional and conditional image generations. The Transformer-based discriminator, *STrans-D*, also significantly reduces its gap against the CNN-based discriminators. Models and code will be publicly available.

1 Introduction

Transformers (Vaswani et al., 2017) have shown remarkable performance in natural language processing, and lately on computer vision, thanks to the exceptional representation learning capability of self-attention layers. The success has led a further exploration to replace the commonly-used convolutional neural network (CNN) backbone with Transformers in generative adversarial networks (GAN) (Goodfellow et al., 2014) for image synthesis. Early attempts (Jiang et al., 2021; Lee et al., 2021) revealed that it is particularly challenging to design an effective Transformer-based GAN. In particular, a direct application of Transformers previously designed for classification tasks in GAN often leads to inferior synthesis performance to the CNN counterparts. For instance, the GAN model with a ViT backbone (Jiang et al., 2021) only achieves a Fréchet inception distance (FID) (Heusel et al., 2017) of 8.92 in 64×64 CELEBA dataset, compared to the FID of 3.16 achieved by the CNN-based StyleGAN2 (Karras et al., 2020b). In addition, the Transformer structure makes GAN training unstable, making it to heavily rely on the manual tuning of hyper-parameters.

In this paper, we aim to understand the intrinsic behavior of Transformers in GAN, aiming to reduce the performance gap between the Transformer-based GANs and those based on CNN backbones. We investigate not only unconditional image synthesis but also the less-explored conditional setting. Our study provides the following practices and design principles of adopting Transformer in GANs:

1) *Locality is essential* - Locality of feature extraction has been shown pivotal to efficiency and performance of Transformer in image classification (Li et al., 2021; d'Ascoli et al., 2021). We observe that such a rule-of-thumb is also applicable to GAN-based image generation. In particular, we find that global self-attention operations (Vaswani et al., 2017; Dosovitskiy et al., 2020) as practiced in existing Transformer-based GANs (Jiang et al., 2021) are detrimental to the synthesis performance as well as computationally prohibitive for high-resolution image generation. We highlight several possible ways of adding locality to Transformers. Among these methods, Swin layer (Liu et al., 2021) proves to be the most effective building block to offer the locality inductive bias.

2) *Mind the residual connections in the discriminator* - Transformer employs a residual connection around each sub-layer of self-attention and the pointwise fully connected layer. Through a detailed analysis of norm ratios, we find that residual connections tend to dominate the information flow in a Transformer-based discriminator. Consequently, sub-layers that perform self-attention and fully connected operations in the discriminator are inadvertently bypassed, causing inferior quality and slow convergence during training. We address this problem by replacing each residual connection with a skip-projection layer, which better retains the information flow in the residual blocks.

3) *Transformer-specific strategy for placing conditional normalization* - We observe that conventional approaches of injecting conditional class information do not work well for Transformer-based conditional GAN. The main culprit lies in the large information flow through the residual connections in the Transformer generator. If the conditional information is injected within the main branch[1], it will largely be ignored and contribute little to the final outputs. We present a viable way of adopting conditional normalization layers in the trunk, which helps retain conditional information throughout the Transformer generator.

Taking the principles above into consideration, we successfully mitigate the gaps between Transformer-based GAN and contemporary CNN-based GANs, which were deemed difficult in previous studies. All the design choices can be implemented easily without elaborative architectural modifications to the commonly-used Transformer models. The resulting model, called *STransGAN*, achieves a state-of-the-art FID of 2.03 in CELEBA dataset and reaches a competitive FID of 4.84 in FFHQ-256 dataset. In our attempt of using Transformers for conditional synthesis, the presented model improves the inception score from 10.14 to 11.62 in the challenging CIFAR10 dataset and achieves competitive performance in IMAGENET1K (Fig. 1). It is noteworthy that this study presents the first successful application of Transformer-based GAN under the conditional setting.

2 RELATED WORK

Vision Transformer. Due to its success in NLP, Transformer (Vaswani et al., 2017) has obtained growing attention in the computer vision community (Lin et al., 2021; Carion et al., 2020; Xu & Loy, 2021). ViT (Dosovitskiy et al., 2020) and subsequent studies (Touvron et al., 2021; Zhou et al., 2021; Rao et al., 2021) successfully transfer the Transformer architecture to the image domain. Transformer-based models have been proven effective in multiple high-level tasks, *e.g.*, image detection (Carion et al., 2020; Zhu et al., 2020) and segmentation (Zheng et al., 2021; Strudel et al., 2021). However, the standard Transformer with heavy computational costs prevents its application on high-resolution images. To reduce the computational costs, recent studies propose to use local attention (Liu et al., 2021; Zhang et al., 2021), which shows superior performance. There have been a few recent efforts in image restoration (Chen et al., 2021; Liang et al., 2021) and image synthesis (Esser et al., 2021) with Transformer. Nevertheless, they merely apply the Transformer block as an intermediate component, while our generator is a CNN-free model with only Transformer blocks.

Generative Adversarial Networks. Convolutional GANs (Radford et al., 2015; Arjovsky et al., 2017) are dominant in the literature. Karras et al. (2018) develop a progressive growing architecture to facilitate higher-resolution synthesis. The state-of-the-art StyleGAN2 (Karras et al., 2019; 2020b) offer an architecture to disentangle the style information. In the conditional GANs, the attention mechanism (Zhang et al., 2019; Brock et al., 2018; Kang & Park, 2020) has been widely adopted in convolutional generators to better capture the shape priors and long-range correspondence. However, these works merely regard the self-attention block as an auxiliary module. In this study, we carefully study the role and impact of self-attention blocks in Transformers when the architecture is used as the generator and discriminator.

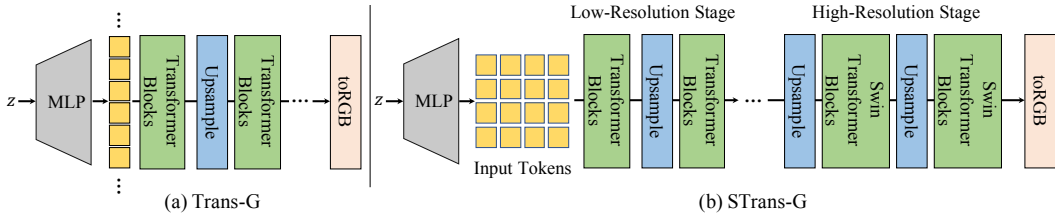[1]Also known as the residual mapping path in He et al. (2016).

Figure 2: (a) illustrates the overall structure of the baseline generator, Trans-G, consisting of standard Transformer blocks. (b) An overview of STrans-G that adopts localized attention modules in high-resolution stages.

Transformer in GANs. The most related work to ours are several pilot studies (Jiang et al., 2021; Zhao et al., 2021; Lee et al., 2021; Park & Kim, 2021), which also attempt to devise Transformer-based GANs. Apart from the difference in adopting Swin Transformer, we present a detailed comparison between these concurrent works and ours in Tab. 1. In this work, we conduct a more comprehensive study containing the conditional generation setting. Besides, we investigate the phenomenon of imbalanced feature flow in Transformer blocks and its impacts to the discriminators and conditional generators.

Table 1: Comparison between concurrent work and ours. We summarize if these studies investigate the generator (Gen.) or/and discriminator (Disc.) design. Additionally, we show whether unconditional (Uncond.) and conditional (Cond.) synthesis are supported.

Methods	Architecture		Sampling	
	Gen.	Disc.	Uncond.	Cond.
TwoTransGANs (Jiang et al., 2021)	✓	✓	✓	
HiT (Zhao et al., 2021)	✓		✓	
ViTGANs (Lee et al., 2021)	✓	✓	✓	
StyleFormer (Park & Kim, 2021)	✓		✓	
Ours	✓	✓	✓	✓

3 METHODOLOGY

While Transformer has shown impressive performance in representation learning (Vaswani et al., 2017; Xie et al., 2021; Carion et al., 2020), early attempts to adopt it for generative modeling mostly lead to dissatisfying results. In the following, we investigate why the existing models fail to generate high-quality images, and further discuss useful practices for designing competitive Transformer-based GANs under both conditional and unconditional settings.

3.1 TRANSFORMER-BASED GENERATOR

Trans-G Baseline. To design the generator, we start from a straightforward baseline structure, Trans-G, which is composed of standard vision Transformer blocks, as shown in Fig. 2(a). Unlike traditional CNN-based models, Trans-G reshapes the spatial feature map (H_0, W_0, C) into a sequence of input tokens $(H_0 \times W_0, C)$. By stacking upsampling operation and Transformer blocks, Trans-G gradually increases the resolution of the feature map until it meets the target scale $H \times W$. In such a design, the global attention module in Transformer blocks enhances the modeling ability of long-range dependencies, and several recent studies based on it (Lee et al., 2021; Park & Kim, 2021) present promising results in low-resolution synthesis.

Samples generated by Trans-G, however, often contain severe artifacts and unrealistic details, leading to poor visual quality (Fig. 3(a)). In particular, we train Trans-G with a strong CNN-based discriminator adopted in StyleGAN2-ADA (Karras et al., 2020b;a) on 64×64 FFHQ dataset (Karras et al., 2019). As shown in Fig. 3(a), while Trans-G can synthesize reasonable global structures in almost all the samples, the generator fails to synthesize realistic high-frequency textures in local regions causing unpleasant artifacts, *e.g.*, the white points in the facial areas.

Given the presence of such high-frequency artifacts, we believe the root cause is the attention modules in the high-resolution stages. Consequently, we first analyze the last high-resolution attention map closest to the final image feature. The yellow arrows in Fig. 3(c) point to the pixel with the highest attention score *w.r.t.* the query pixel, *i.e.*, the starting point of the arrow. As can be observed, the self-attention module in Trans-G aggregates features from an unrelated distant region, *e.g.*, the query pixel in the facial region attends to a background pixel. Such a phenomenon suggests the possible pitfall of introducing global attention in the high-resolution feature stage.
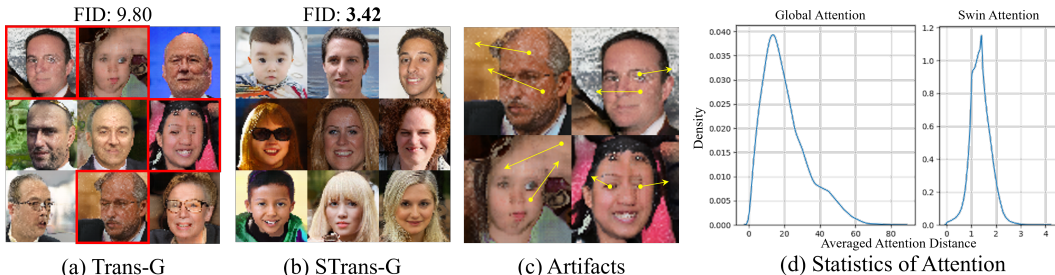
| FID: 9.80 | FID: **3.42** | | Global Attention | Swin Attention |
| (a) Trans-G | (b) STrans-G | (c) Artifacts | (d) Statistics of Attention | |

Figure 3: (a, b) show 64×64 samples from Trans-G and STrans-G, respectively. (c) We analyze the failure cases from Trans-G. Given a query pixel at the starting point, the yellow arrow in (c) points to the pixels obtaining the highest attention score in the last attention layer. (d) We plot the distribution of the averaged attention distance, indicating the spatial distance from the query token position to the location that modules attend to. The statistics are calculated in the last stage of the generator.

We further study the behavior of the self-attention module by computing the average distance between the query token position and the locations it attends to. Figure 3(d) shows the statistics collected by sampling 10k images. Without any strict constraints, the global attention module always attends to pixels with a medium distance (10-20 pixels) that cannot offer direct guidance in synthesizing local textures. Importantly, a significant amount of samples in Trans-G even attend to unreasonably distant pixels (>32 pixels) in the highest scale, which can easily result in unpleasant noisy artifacts as shown in Fig. 3(c).

STrans-G. The analysis above suggests that those Transformer blocks in high-resolution stages should follow the locality nature of image data rather than perform feature reassembly in an unconstrained manner. The finding motivates us to explore the effect of various local attention mechanisms (Liu et al., 2021; Zhang et al., 2021; Jiang et al., 2021) in generating realistic high-resolution images.

After trying different local attention modules (Jiang et al., 2021; Zhao et al., 2021) extensively, we select the Swin architecture (Liu et al., 2021) as the building block in high-resolution stages starting from 16×16 pixels (Fig. 2(b)). In the first several stages, since feature maps have a small spatial dimension, we directly apply the standard Transformer block (Dosovitskiy et al., 2020) to capture the global structure. Swin not only promotes locality with its non-overlapped local windows, but it also helps attention exchange with shifted window attention. In addition, it does not introduce more parameters, and it keeps a linear computational complexity ($\mathcal{O}(HW)$) in various input resolutions. All such designs make the Swin layer an attractive option for a Transformer-based GAN.

After switching to the Swin architecture with a window size of $M = 4$, we observe a significant improvement in both qualitative and quantitative results, as shown in Fig. 3(b). The unpleasant artifacts disappear and the FID is improved by 65% over Trans-G. Figure 3(d) presents the distribution of the attention distance in Swin attention modules from the last stage of the generator. Interestingly, the Swin architecture tends to focus on nearby regions within only a vicinity of two pixels.

Relationship with Concurrent Work. Several concurrent studies (Zhao et al., 2021; Jiang et al., 2021) also introduce local attention mechanisms into GANs. These prior studies propose local attention mostly from a computational perspective, *i.e.*, to avoid the quadratic complexity of global attention modules. This goal differs significantly from our motivation of reducing artifacts and improving image quality. Specifically, Jiang et al. (2021) propose the grid self-attention to aggregate features in a 16×16 window. However, the grid attention lacks connections between local regions. Information from different local regions thus cannot be effectively communicated, leading to unpleasant results. While HiT (Zhao et al., 2021) remedies this issue with dilated attention, it brings $\mathcal{O}((HW)^{1.5})$ computational complexity that is unaffordable for higher resolutions. Thus, HiT can only apply Transformer blocks in low-resolution stages, while the synthesis in high-resolution stages cannot benefit from the self-attention mechanism. In contrast, STrans-G uses the shift-window scheme in Swin blocks, allowing effective information propagation between neighboring regions while restricting the computations in local windows. In Sec. 4.3, we provide a comprehensive comparison among different local attention mechanisms and a systematic analysis of their effects on image generation. Note that we do not claim contributions on the Swin architecture. Instead, this study focuses on
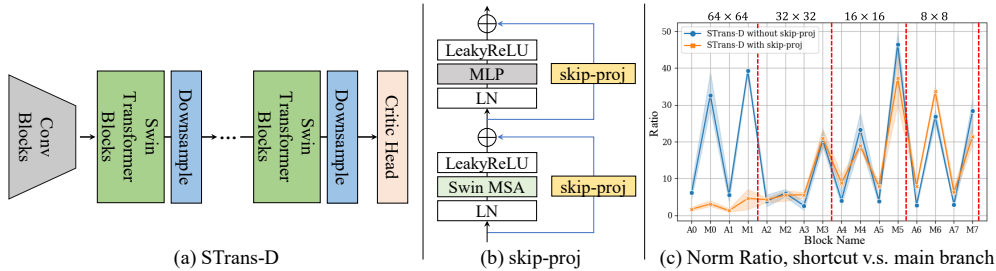
(a) STrans-D   (b) skip-proj   (c) Norm Ratio, shortcut v.s. main branch

Figure 4: (a) The overall architecture of STrans-D. In (b), we illustrate the detailed architecture of the Swin Transformer block with **skip-proj** in STrans-D. (c) presents the norm ratio between the shortcut (blue path in (b)) and the main branch that contains self-attention and MLP blocks. 'Ai' and 'Mj' denotes the i-th attention block and the j-th MLP block, respectively. The red vertical line in (c) represents the position of a downsampling operation. At the top of (c), we offer the resolution of the features in the current stage. We train each configuration with three different random seeds. The error bar on each curve represents the variance across different runs.

investigating the benefits of adopting local attention in Transformer-based GANs and the underlying reasons.

3.2 TRANSFORMER-BASED DISCRIMINATOR

Similar to STrans-G, we also use Swin layers for the discriminator, termed as STrans-D. Nevertheless, directly applying Swin in the discriminator leads to unstable adversarial training in our experiments, which often results in degenerated image qualities. Next, we summarize our experience to improve the baseline model of STrans-D and present a novel skip-projection scheme as a remedy.

Empirical Strategies for STrans-D. First, instead of starting with a patch embedding module as in most vision Transformers (Dosovitskiy et al., 2020), we adopt a lightweight convolutional block to downsample the original input by $4\times$ and project the image tensor to an arbitrary dimension. Compared to patch embedding, the convolutional token extractor adopts overlapped patches, keeping more detailed information. Second, we adopt the equalized learning rate (Karras et al., 2018) in all attention modules and MLPs. This is motivated by the slow and unsatisfactory convergence of Transformer blocks in the discriminator when they use a small learning rate to stabilize its training. We resolve this issue by setting a large learning rate for the whole discriminator and introduce a special scaler to multiply the learnable parameters from Transformer blocks at runtime. Additionally, we replace GeLU (Hendrycks & Gimpel, 2016) with LeakyReLU (Maas et al., 2013) and add a nonlinear activation function at the end of the attention and MLP modules. More details about these improvements can be found in Appendix A.1.2.

Skip-Projection. Despite the empirical improvements above, there is still a significant performance gap between the Transformer-based STrans-D and the discriminator of StyleGAN2. To solve this problem, we perform a careful analysis of the STrans-D, and find it is the residual connections in Transformer blocks that lead to an undesirable information flow, making the feedback from the discriminator degrades. In particular, we plot the norm ratio (Raghu et al., 2021) of the residual connections, which is defined as $||x||/||f(x)||$, where x denotes the features from the shortcut and $f(x)$ represents the transformation of x from the main branch. The norm ratio is collected from the checkpoint with the best FID and thus faithfully reflects the information flow in the network.

As shown in Fig. 4(c), STrans-D (blue curve) has an extremely high ratio in the 64×64 stage, indicating that the information almost flows through the direct shortcuts instead of the sub-layers containing the self-attention and MLP blocks. It is from 32×32 stages that STrans-D starts processing features with the main branch. This deviates from our expectation that a discriminator should have paid more attention to the high-resolution features containing plentiful details, so that its feedback can better guide high-quality synthesis.

To avoid such an unhealthy behavior, we apply a skip-projection layer, termed as **skip-proj**, which performs a linear projection in the residual connection (Fig. 4(b)). The linear projection layer adaptively adjusts the scale of features in residual connections, allowing the discriminator to experience a more reasonable information flow. The comparison in Fig. 4(c) proves that skip-proj prevents the
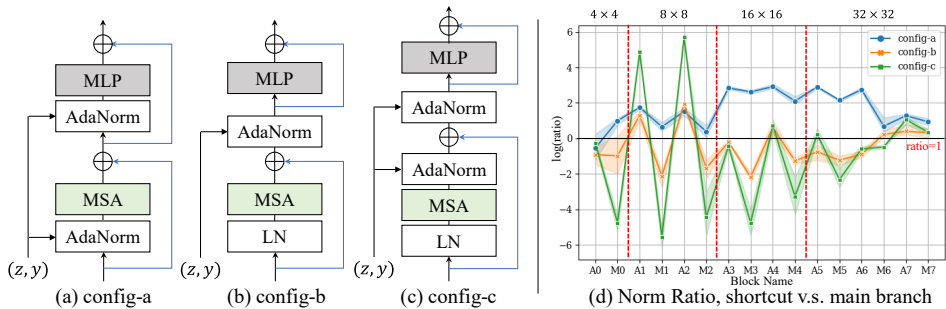
5

Figure 5: (a)-(c) show three alternative designs of adopting AdaNorm in a Transformer block. (d) presents the norm ratio between the shortcut (blue path in (a)-(c)) and the main branch. 'Ai' and 'Mj' denote the i-th attention block and the j-th MLP block, respectively. The red vertical line in (d) represents the position of an upsampling operation. At the top of (d), we offer the resolution of the features in the current stage. We train each configuration with three different random seeds. The error bar on each curve represents the variance across different runs.

escalation of the norm ratios in early stages. Benefiting from the high-resolution features, STrans-D with skip-proj effectively reduces the gap against the StyleGAN2 discriminator. In Sec. 4.2, we show how STrans-D evolves step by step and verify the effectiveness of skip-proj in speeding up convergence and improving final performance.

3.3 CONDITIONAL IMAGE GENERATION

We also study Transformer-based GAN for conditional image generation. Related to our previous discussions on skip-proj, we find that the direct shortcut in Transformer blocks also has a profound influence on the design of conditional generators.

In convolutional GANs, the category condition is typically injected by AdaNorm on the main branch (Dumoulin et al., 2017; Huang & Belongie, 2017; Park et al., 2019):

$$\text{AdaNorm}(x, y, z) = \gamma(y, z) \cdot \text{Norm}(x) + \beta(y, z), \tag{1}$$

where x, y and z are input features, category conditions, and noise vectors respectively. The $\text{Norm}(\cdot)$ can be realized with different normalization layers, *e.g.*, Layer normalization (Ba et al., 2016), Batch Normalization (Ioffe & Szegedy, 2015), and Instance Normalization (Ulyanov et al., 2016) (correspondingly, the AdaNorm is called as AdaLN, AdaBN, and AdaIN). $\gamma(\cdot)$ and $\beta(\cdot)$ modulate the normalized feature $\text{Norm}(x)$ according to the conditional label y and noise vector z.

Our attempt of adopting AdaLN directly in Transformer blocks (config-a in Fig. 5(a)) fails. In particular, we find the FID stops decreasing in the early stage. To trace the source of failure, we plot the norm ratio of this baseline configuration. As shown in Fig. 5(d), there exist several blocks with a large norm ratio, indicating that some AdaNorm layers in the main branch contribute little to the intermediate features, leading to a loss of conditional information. To guarantee the injection of conditional information, a simple solution is to apply the AdaNorm in the trunk, as shown in config-b of Fig. 5(b). In this way, the features from both the shortcut and the MLP branch are ensured to contain class information. Interestingly, config-b brings the norm ratio below the baseline config-a.

Config-b, however, is not a stable solution as it causes undesirable mode collapse in several training cases. Since mode collapse reflects a failure in modeling global structures, we focus on the early attention blocks which compute global attention. As shown in Fig. 5(c), after adopting another AdaNorm at the end of the attention block (config-c), we find mode collapse disappears in the early stage. As shown in Fig. 5(d), config-c leads to an extremely high ratio in the second and third attention blocks, hence avoiding collapsed global structures from the main branch. We hypothesize this is mainly due to $\gamma(\cdot)$ and $\beta(\cdot)$ in Eq. (1), which can serve as a gating function to mask undesirable signals. Meanwhile, the lower norm ratio in config-c (green curve in Fig. 5(d)) suggests that the model learns to depend more on the MLP blocks to mitigate the impacts of bypassing the global attention modules.

Table 2: (a, b) present the comparison of FID with popular unconditional methods.[2] In conditional generation, (c, d) show the comparison with CNN-based conditional models.[3] STrans-G in FFHQ 256^2 only contains **20M** parameters, while StyleGAN2 and HiT have 30M and 46M parameters in the generator. ↓ indicates lower the better, and ↑ indicates higher the better.

(a) CELEBA 64^2

Method	FID ↓
HDCGAN	8.44
StyleGAN2-ADA	3.16
UDM	2.78
TransGAN	5.01
ViTGAN	3.74
StyleFormer	3.66
STrans-G	**2.03**

(b) FFHQ 256^2

Method	FID ↓
U-Net GAN	7.63
VQGAN	11.40
INR-GAN	9.57
CIPS	4.38
StyleGAN2-ADA	3.62
HiT	**2.95**
STrans-G	4.84

(c) CIFAR10 32^2

Method	FID ↓	IS ↑
ProjGAN	17.50	8.62
BigGAN	14.73	9.22
MultiHinge	6.40	9.68
FQGAN	5.59	8.48
Mix-MHingeGAN	3.60	10.21
StyleGAN2-ADA	**2.42**	10.14
STrans-G	2.77	**11.62**

(d) IMAGENET 128^2

Method	FID ↓	IS ↑
ProjGAN	27.62	36.80
SAGAN	18.65	52.52
YLG	15.94	57.22
ContraGAN	19.69	31.10
FQ-GAN	13.77	54.36
BigGAN	**8.70**	**98.80**
STrans-G	16.05	47.17

In summary, Fig. 5(c) depicts the structure that we found useful for adopting AdaNorm in Transformer blocks. We call it as ***AdaNorm-T***, and correspondingly AdaIN-T/AdaBN-T if IN/BN is used. Note that the AdaNorm after the attention modules is only adopted in the global attention blocks, as we find applying it in Swin attention blocks does not further improve the performance. The effectiveness of AdaNorm-T is verified on the CIFAR10 and IMAGENET datasets in Sec. 4.1.

4 EXPERIMENTS

Implementation Details. To reduce computational costs, we set the channel expansion ratio to 2 for the MLP modules in Transformer blocks. The input token dimension is 512. We adopt four attention heads by default. We select Adam (Kingma & Ba, 2014) optimizer with $\beta_1 = 0, \beta_2 = 0.99$ to train our generators and discriminators. STrans-G and STrans-D are optimized with a learning rate of 0.0001 and 0.002, respectively. All of our models are trained on 8 Tesla V100 GPUs in PyTorch (Paszke et al., 2019). We evaluate our models with FID and Inception Score (IS) (Salimans et al., 2016), following the implementation in Karras et al. (2020a). More details about models and training pipelines can be found in the following sections and Appendix A.1.

4.1 STRANS-G FOR UNCONDITIONAL AND CONDITIONAL GENERATION

Setup. To evaluate the effectiveness of STrans-G, we first adopt strong CNN-based discriminators to pair with STrans-G in both unconditional and conditional settings. For experiments in FFHQ 256^2 (Karras et al., 2019) and CIFAR10 (Krizhevsky, 2009), we use the popular discriminator proposed in StyleGAN2-ADA (Karras et al., 2020a), and follow its training configuration. Note that the path regularization loss is removed during optimizing STrans-G to reduce the training costs. For CELEBA (Liu et al., 2015), we apply the same training configuration with FFHQ 256^2. In the experiments for IMAGENET (Krizhevsky et al., 2012), as we find the StyleGAN2 discriminator always leads to an early stopping in training with such large-scale data, we switch to the discriminator proposed by Brock et al. (2018) and follow its setting with a batch size of 2048. To efficiently inject class information, we adopt AdaIN-T in CIFAR10 and AdaBN-T in IMAGENET.

Results. In Tab. 2, we compare STrans-G with state-of-the-art CNN-based and Transformer-based models. In unconditional generation, STrans-G significantly outperforms all previous methods in CELEBA 64^2. It also achieve competitive performance in the high-resolution setting of FFHQ 256^2. It is noteworthy that the gap between HiT (Zhao et al., 2021) and STrans-G is mainly caused by the model size and large training batches. In fact, STrans-G only contains **20M** parameters, while StyleGAN2 and HiT have 30M and 46M parameters in the generator, respectively. In addition, HiT applies a large batch size of 256 using TPUs, while STrans-G's batch size is 64 by default.

For conditional image generation, with the proposed AdaIN-T layer, STrans-G improves the SOTA Inception Score (IS) (Salimans et al., 2016) from 10.14 to 11.62 on CIFAR10, as shown in Tab. 2(c).

---

[2]Unconditional methods: HDCGAN (Curtó et al., 2017), StyleGAN2-ADA (Karras et al., 2020a), UDM (Kim et al., 2021), U-NetGAN (Schonfeld et al., 2020), VQGAN (Esser et al., 2021), INR-GAN (Skorokhodov et al., 2021), CIPS (Anokhin et al., 2021), TransGANs (Jiang et al., 2021), ViTGANs (Lee et al., 2021), StyleFormer (Park & Kim, 2021), and HiT (Zhao et al., 2021).

[3]CNN-based conditional models: ProjGAN (Miyato & Koyama, 2018), SAGAN (Zhang et al., 2019), MultiHinge (Kavalerov et al., 2019), YLG (Daras et al., 2020), FQGAN (Zhao et al., 2020b), Mix-MHingeGAN (Tang, 2020), ContraGAN (Kang & Park, 2020), and BigGAN (Brock et al., 2018).
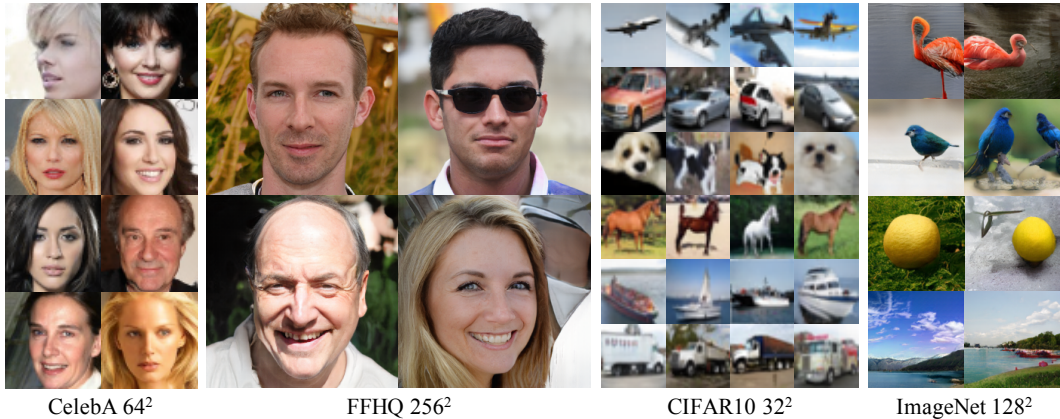
| CelebA 64^2 | FFHQ 256^2 | CIFAR10 32^2 | ImageNet 128^2 |

Figure 6: Exemplar images generated from STrans-G in unconditional and conditional settings.

Configuration	FFHQ	CELEBA
a. Swin Baseline	20.19	14.63
b. + Conv Extractor	14.10	11.87
c. + Equalized LR	8.95	6.13
d. + LeakyReLU	8.63	4.44
e. + skip-proj	5.98	3.41
f. StyleGAN2-D	4.84	2.03

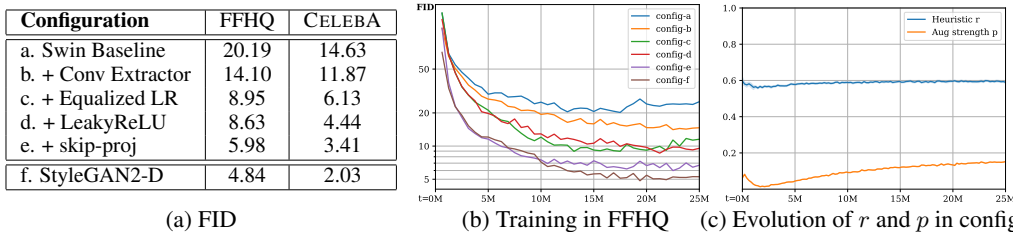| (a) FID | (b) Training in FFHQ | (c) Evolution of r and p in config-e |

Figure 7: (a) The FID for various discriminator configurations described in Sec. 3.2. (b) We plot the evaluation FID during the training stage for each configuration in (a). (c) In FFHQ 256^2, we present the evolution of two indicators proposed in ADA (Karras et al., 2020a), *i.e.*, overfitting heuristic r, and augmentation strength p.

Since CIFAR10 is widely adopted as a limited data benchmark (Karras et al., 2020a), this result also suggests the robustness of STrans-G in modeling real distributions with limited data. In the IMAGENET evaluation, we observe a considerable gap between STrans-G and the state-of-the-art CNN-based BigGAN model. The results suggest room for improvement in Transformer-based GAN compared to the extensively tuned CNN-based models. Nonetheless, for the first time, our study shows the potential of Transformers in the conditional generation setting. Figure 6 presents unconditional and conditional samples generated from STrans-G. The visual quality of these generated samples suggests the great potential of using pure Transformer blocks in GANs. Additional qualitative results, including interpolation in the latent space, can be found in Appendix A.2.

4.2 STRANS-D

Setup. Here, we study the effectiveness of STrans-D by pairing it with the unconditional STrans-G. The conditional discriminator can be easily implemented by adding conditional projection (Miyato & Koyama, 2018) in the critic head (Fig. 4(a)), which is not relevant to the Transformer backbone. Thus, in this work, we only conduct experiments in the unconditional generation. The training schedule is kept the same as the StyleGAN2 discriminator (Sec. 4.1) to guarantee a fair comparison. More implementation details about STrans-D can be found in Appendix A.1.2.

Results. Fig. 7(a) shows how we improve the initial Swin-based discriminator baseline step by step. Meanwhile, we also plot the evaluation FID over the training process in Fig. 7(b), so that we can conveniently compare the convergence speed.

As the original patch embedding module in vision Transformer takes non-overlapped patches as input, it is not efficient in extracting input tokens. Switching to a convolutional feature extractor can bring a clear improvement of FID in Fig. 7(a). We then adopt the equalized learning rate to adjust the learning rate for the Transformer blocks at runtime. Thanks to this scheme, we can train the discriminator with a large learning rate, speeding up the convergence and decreasing the FID. In addition, adopting LeakyReLU also slightly improves the performance. Finally, the proposed skip-proj significantly reduces this gap (variant-e in Fig. 7(a)) and accelerates the convergence in the early stage (Fig. 7(b)).

Table 3: (a) We study the effects of adopting different attention mechanisms in STtrans-G. 'Trans' indicates the standard self-attention module (Dosovitskiy et al., 2020), while 'Grid' (Jiang et al., 2021) and 'MultiAxis' (Zhao et al., 2021) are recent studies applying the localized idea in attention modules. For a fair comparison, we also add relative positional encoding (Liu et al., 2021) for the standard Transformer blocks, denoted by '+Rel. Pos.'. (b) We compare different architectural choices and normalization layers in AdaNorm-T. The architectures are depicted in Fig. 5(a-c), while 'LN' and 'IN' denote layer normalization and instance normalization, respectively.

(a) Attention Compraison

Attention	CELEBA FID↓	FFHQ FID↓	CIFAR10 FID↓	CIFAR10 IS↑
Trans	3.77	–	4.01	9.25
+Rel. Pos.	3.70	–	4.25	9.11
Grid	3.63	6.52	3.82	9.31
MultiAxis	3.60	5.63	3.22	10.12
Swin	**2.03**	**4.84**	**2.77**	**11.62**

(b) AdaNorm-T in CIFAR10

Architecture	Norm	FID↓	IS↑
config-a	LN	4.94	9.53
	IN	5.78	8.13
config-b	LN	13.30	5.27
	IN	10.14	5.48
config-c	LN	3.99	9.72
	IN	2.77	11.62

We highlight that STrans-D is not prone to the overfitting problem. As introduced in Karras et al. (2020a), the overfitting of a discriminator can be characterized by two indicators: the overfitting heuristic r and augmentation strength p. If a discriminator meets the overfitting problem, p should increase quickly in training and ends up with a value larger than 0.2, while r should reach a value close to 1. We plot the evolution of r and p over the course of training with variant-e in Figure 7(c). The heuristic r is always below 0.6 and the augmentation strength p ends up with a small probability, suggesting that STrans-D has not met the overfitting problem. We further discuss the effects of differentiable data augmentation in Appendix A.1.3.

4.3 FURTHER ANALYSIS

Different Local Attention Mechanisms. STrans-G is a general framework where we can switch to various attention mechanisms. Here, we study the effects of different possible choices based on the framework of STrans-G. Table 3(a) presents the influence of different attention mechanisms on the final performance. With global attention, the standard Transformer block consistently yields inferior image qualities, compared to the others with local attention mechanisms. Grid attention (Jiang et al., 2021) computes self-attention within non-overlapped windows but lacks an efficient way to bridge each local region. MultiAxis (Zhao et al., 2021) adopts a sparsed global attention to enhance communication among local patches. Nevertheless, its design breaks the locality in high-resolution stages while introducing heavy computational costs. Thus, these models do not give satisfying results as shown in Tab. 3(a), and we select the Swin architecture as our default choice of the attention mechanism for high-fidelity image synthesis.

Ablation Study on AdaNorm-T. In Tab. 3(b), we study the effects of different architectural choices and normalization layers in AdaNorm-T. As shown in Fig. 5(a), config-a only injects conditional information in residual blocks, which may be ignored by the direct shortcuts. Thus, it often causes inferior results compared with the config-c in Tab. 3(b). Wile config-b (Fig. 5(b)) guarantees the injection of conditional information, it often leads to mode collapse in the early training stages, resulting in extremely high FID. By adopting an additional AdaNorm at the end of the attention blocks, AdaNorm-T (Fig. 5(c)) avoids early stopping in training and achieves much better performance (config-c in Tab. 3(b)). Besides, the choice of normalization layers also influences the quality of conditional generation. We note that the choice of normalization layers highly depends on the dataset. More discussions are provided in the Appendix A.1.1.

5 CONCLUSION

We have conducted comprehensive empirical study on deploying Transformer in GANs. Through studying the intrinsic properties of self-attention layers, we observed the crucial role of feature locality in high-quality image synthesis. Swin layer, the popular attention block for classification network, shows promising potential in Transformer-based GAN. Our analysis also revealed the residual connections, which were assumed to be 'safe and useful', could harm the performance of Transformer-based discriminator and conditional generator. The resulting STransGAN achieves high-quality image generation on a wide variety of benchmarks, which can serve as a simple yet strong baseline for future research. This work also demystifies the early failures of using Transformers for GANs and paves the promising direction of Transformer-based generative modeling.

6 ETHICS AND REPRODUCIBILITY STATEMENT

Ethics Statement. This work improves the image generation quality, which can facilitate the movie industry and virtual reality. However, there could be potentially inappropriate applications of this technique, lying in various forms of disinformation, *e.g.*, synthesizing fake portraits in different scenes (Perov et al., 2020). Our work may unintentionally make these applications more convincing or deceiving. Viable solutions to prevent misuse and negative effects include developing robust methods to detect fake content generated by deep models (Tolosana et al., 2020) and adopting model watermarking (Yu et al., 2020).

Reproducibility Statement. We have provided the necessary implementation details in Sec. 3, Sec. 4, and Appendix A.1. In addition, we also include more results with different settings of STransGAN in Appendix A.1. Our training pipeline follows standard training configurations in current literature and does not include extra tricks. The models and codes will be publicly available.

REFERENCES

Ivan Anokhin, Kirill Demochkin, Taras Khakhulin, Gleb Sterkin, Victor Lempitsky, and Denis Korzhenkov. Image generators with conditionally-independent pixel synthesis. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. *International Conference on Machine Learning*, 2017.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *Advances in Neural Information Processing Systems Workshop*, 2016.

Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. In *International Conference on Learning Representations*, 2018.

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision*, 2020.

Hanting Chen, Yunhe Wang, Tianyu Guo, Chang Xu, Yiping Deng, Zhenhua Liu, Siwei Ma, Chunjing Xu, Chao Xu, and Wen Gao. Pre-trained image processing transformer. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

Joachim D Curtó, Irene C Zarza, Fernando De La Torre, Irwin King, and Michael R Lyu. High-resolution deep convolutional generative adversarial networks. *arXiv preprint arXiv:1711.06491*, 2017.

Giannis Daras, Augustus Odena, Han Zhang, and Alexandros G Dimakis. Your local gan: Designing two dimensional local attention mechanisms for generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14531–14539, 2020.

Stéphane d'Ascoli, Hugo Touvron, Matthew Leavitt, Ari Morcos, Giulio Biroli, and Levent Sagun. Convit: Improving vision transformers with soft convolutional inductive biases. In *International Conference on Machine Learning*, 2021.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *International Conference on Learning Representations*, 2020.

Vincent Dumoulin, Jonathon Shlens, and Manjunath Kudlur. A learned representation for artistic style. *International Conference on Learning Representations*, 2017.

Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, 2014.

Boris Hanin and David Rolnick. How to start training: The effect of initialization and architecture. *Advances in Neural Information Processing Systems*, 2018.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*, 2017.

Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *IEEE International Conference on Computer Vision*, 2017.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *International Conference on Machine Learning*, 2015.

Yifan Jiang, Shiyu Chang, and Zhangyang Wang. Transgan: Two transformers can make one strong gan. *arXiv preprint arXiv:2102.07074*, 2021.

Minguk Kang and Jaesik Park. Contragan: Contrastive learning for conditional image generation. *Advances in Neural Information Processing Systems*, 2020.

Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *International Conference on Learning Representations*, 2018.

Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. *Advances in Neural Information Processing Systems*, 2020a.

Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020b.

Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks. *arXiv preprint arXiv:2106.12423*, 2021.

Ilya Kavalerov, Wojciech Czaja, and Rama Chellappa. cgans with multi-hinge loss. *arXiv preprint arXiv:1912.04216*, 2019.

Dongjun Kim, Seungjae Shin, Kyungwoo Song, Wanmo Kang, and Il-Chul Moon. Score matching model for unbounded data score. *arXiv preprint arXiv:2106.05527*, 2021.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, 2012.

Kwonjoon Lee, Huiwen Chang, Lu Jiang, Han Zhang, Zhuowen Tu, and Ce Liu. Vitgan: Training gans with vision transformers. *arXiv preprint arXiv:2107.04589*, 2021.

Yawei Li, Kai Zhang, Jiezhang Cao, Radu Timofte, and Luc Van Gool. Localvit: Bringing locality to vision transformers. *IEEE International Conference on Computer Vision*, 2021.

Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. Swinir: Image restoration using swin transformer. In *IEEE Conference on International Conference on Computer Vision Workshops*, 2021.

Kevin Lin, Lijuan Wang, and Zicheng Liu. End-to-end human pose and mesh reconstruction with transformers. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *IEEE International Conference on Computer Vision*, 2021.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *IEEE International Conference on Computer Vision*, 2015.

Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. Rectifier nonlinearities improve neural network acoustic models. In *International Conference on Machine Learning*, 2013.

Takeru Miyato and Masanori Koyama. cgans with projection discriminator. *arXiv preprint arXiv:1802.05637*, 2018.

Jeeseung Park and Younggeun Kim. Styleformer: Transformer based generative adversarial networks with style vector. *arXiv preprint arXiv:2106.07023*, 2021.

Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, 2019.

Ivan Perov, Daiheng Gao, Nikolay Chervoniy, Kunlin Liu, Sugasa Marangonda, Chris Umé, Mr Dpfks, Carl Shift Facenheim, Luis RP, Jian Jiang, et al. Deepfacelab: A simple, flexible and extensible face swapping framework. *arXiv preprint arXiv:2005.05535*, 2020.

Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *International Conference on Learning Representations*, 2015.

Maithra Raghu, Thomas Unterthiner, Simon Kornblith, Chiyuan Zhang, and Alexey Dosovitskiy. Do vision transformers see like convolutional neural networks? *arXiv preprint arXiv:2108.08810*, 2021.

Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. Dynamicvit: Efficient vision transformers with dynamic token sparsification. *arXiv preprint arXiv:2106.02034*, 2021.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in Neural Information Processing Systems*, 2016.

Edgar Schonfeld, Bernt Schiele, and Anna Khoreva. A u-net based discriminator for generative adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

Ivan Skorokhodov, Savva Ignatyev, and Mohamed Elhoseiny. Adversarial generation of continuous images. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for semantic segmentation. *IEEE International Conference on Computer Vision*, 2021.

Shichang Tang. Lessons learned from the training of gans on artificial datasets. *IEEE Access*, 2020.

Ruben Tolosana, Ruben Vera-Rodriguez, Julian Fierrez, Aythami Morales, and Javier Ortega-Garcia. Deepfakes and beyond: A survey of face manipulation and fake detection. *Information Fusion*, 2020.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, 2021.

Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *Advances in Neural Information Processing Systems*, 2016.

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*, 2017.

Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *arXiv preprint arXiv:2105.15203*, 2021.

Rui Xu, Xintao Wang, Kai Chen, Bolei Zhou, and Chen Change Loy. Positional encoding as spatial inductive bias in gans. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

Xiangyu Xu and Chen Change Loy. 3D human texture estimation from a single image with transformers. In *IEEE International Conference on Computer Vision*, 2021.

Ning Yu, Vladislav Skripniuk, Sahar Abdelnabi, and Mario Fritz. Artificial fingerprinting for generative models: Rooting deepfake attribution in training data. *arXiv preprint arXiv:2007.08457*, 2020.

Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. In *International Conference on Machine Learning*, 2019.

Zizhao Zhang, Han Zhang, Long Zhao, Ting Chen, and Tomas Pfister. Aggregating nested transformers. *arXiv preprint arXiv:2105.12723*, 2021.

Long Zhao, Zizhao Zhang, Ting Chen, Dimitris N Metaxas, and Han Zhang. Improved transformer for high-resolution gans. *arXiv preprint arXiv:2106.07631*, 2021.

Shengyu Zhao, Zhijian Liu, Ji Lin, Jun-Yan Zhu, and Song Han. Differentiable augmentation for data-efficient gan training. *Advances in Neural Information Processing Systems*, 2020a.

Yang Zhao, Chunyuan Li, Ping Yu, Jianfeng Gao, and Changyou Chen. Feature quantization improves gan training. *arXiv preprint arXiv:2004.02088*, 2020b.

Zhengli Zhao, Sameer Singh, Honglak Lee, Zizhao Zhang, Augustus Odena, and Han Zhang. Improved consistency regularization for gans. *Association for the Advancement of Artificial Intelligence*, 2020c.

Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

Daquan Zhou, Bingyi Kang, Xiaojie Jin, Linjie Yang, Xiaochen Lian, Zihang Jiang, Qibin Hou, and Jiashi Feng. Deepvit: Towards deeper vision transformer. *arXiv preprint arXiv:2103.11886*, 2021.

Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *International Conference on Learning Representations*, 2020.

A   APPENDIX

A.1   IMPLEMENTATION DETAILS AND DISCUSSION

In this section, we present the implementation details of STransGAN and offer discussions about several details. All of our models are trained on 8 Tesla V100 GPUs in PyTorch (Paszke et al., 2019). The models and related codes will be made publicly available.

A.1.1   STRANS-G

As shown in Fig. 2(b), in the unconditional setting, STrans-G adopts a latent code $z \sim \mathcal{N}(0, I)$ as the input, and then a linear projection layer projects it into a $(4 \times 4 \times C)$ feature map. Each spatial vector in this low-resolution feature is regarded as an input token to the following Transformer blocks. To inject positional information (Karras et al., 2021; Xu et al., 2021), we add a learnable positional encoding to the input tokens. STrans-G gradually increases the resolution of the feature map through multiple Transformer blocks and upsampling operations. Except for the 64×64 stage, we choose the bilinear upsampling operation in each stage. In the 64^2 scale, we adopt pixel shuffle (Shi et al., 2016) operator to upsample features and reduce the channel dimension. The last 'toRGB' block generates image tensors with three channels. As shown in Fig. 8(a), we let the MLP module output a three-channel feature and add a linear layer on the skip connection to match the channel dimension.

In STrans-G, we initialize all of the weights with a truncated normal distribution (Hanin & Rolnick, 2018), following the original setting used in Liu et al. (2021). The number of attention heads and the window size in Swin blocks are both set to 4 by default. Different from the current vision Transformer, STrans-G only expands the channel dimensions in MLP blocks by 2 times. Following the architectural design in StyleGAN, we adopt two Transformer blocks inside each stage in both generators and discriminators.
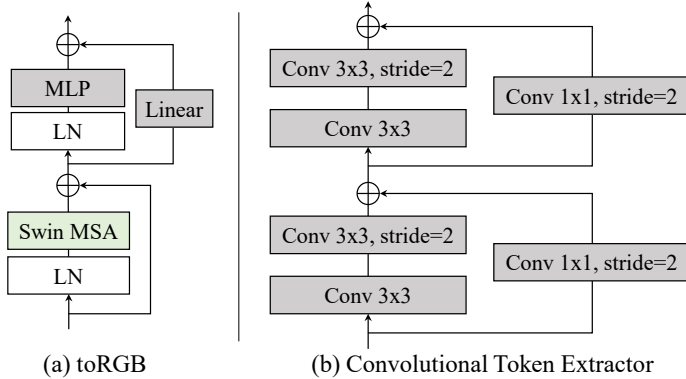

(a) toRGB   (b) Convolutional Token Extractor

Figure 8: (a) illustrates the 'toRGB' layers used in our STrans-G. (b) Architecture of the convolutional residual block used to extract input tokens for discriminators.

Table 4: (a) We report the FID from STrans-G with different window sizes in the attention module. (b) We compare different generators in terms of the number of parameters.

(a) Effects of Window Size

Window Size	CELEBA	FFHQ
2	7.92	11.35
4	**2.03**	4.98
8	2.85	5.23
16	3.51	6.14
Tuning	–	**4.84**

(b) Comparison of Model Size

Method	Dataset	Parameters (million)	FID
StyleGAN2		30.03	3.62
HiT (Zhao et al., 2021)	FFHQ256	46.22	2.95
our STrans-G		19.91	4.84
BigGAN ch=64		31.98	10.48
BigGAN ch=96	IMAGENET128	70.43	8.51
our STrans-G		30.95	16.05

Norm	Dataset	FID	IS
IN	CIFAR10	2.77	11.62
BN		5.61	9.55
IN	IMAGENET128	116.26	7.50
BN		16.05	47.17

(a) Normalization in AdaNorm-T   (b) Training in CIFAR10   (c) Training in IMAGENET
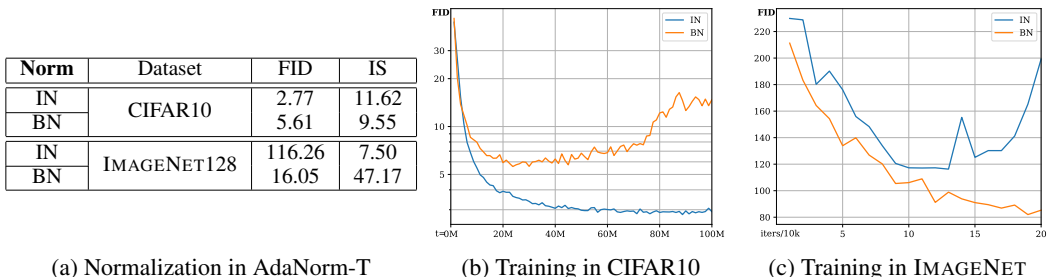
Figure 9: (a) We show how different normalization layers in AdaNorm-T influence the quality of conditional generation in different datasets. To further show the influence in the training process, (b) plots the FID curve during training in CIFAR10. In (c), we only plot the training process at the early stage to show the differences between AdaIN-T and AdaBN-T in IMAGENET.

Window Size. In our experiments, we notice that the window size in the Swin architecture can influence the generation quality. As shown in Tab. 4(a), too small window size causes a significant drop in performance. A large window size like 16 also increases the FID while introducing heavier computational costs. Consequently, we choose 4 as the default window size. We also note that a special tuning in the architecture can lead to better performance in the FFHQ dataset, as shown in Tab. 4(a). In detail, we adopt a window size of 8 in the stages from 16×16 to 64×64 and a window size of 4 in the stages from 128×128 to 256×256. The motivation behind this design is that the layers in lower resolutions need a larger window size to learn global structures. In the higher resolutions, it is more important to keep the locality with a small window size.

Model Size. Table 4(b) presents a comparision of the model size among different methods. Due to the limits of computational resources, STrans-G contains fewer parameters than current popular methods, *e.g.*, StyleGAN2 (Karras et al., 2020a), and BigGAN (Brock et al., 2018). The conclusions in Zhao et al. (2021) and Brock et al. (2018) prove that a larger model always brings gains in the final performance. Thus, the gap between STrans-G and SOTA methods partially comes from the model size. We will further try a larger model once the computational resource meets the requirements.

Normalization in AdaNorm-T. The choice of normalization layers in AdaNorm-T is important and related to the dataset. In this work, we adopt instannce normalization (AdaIN-T) in CIFAR10 and batch normalization (AdaBN-T) in IMAGENET. Figure 9 presents how the normalization layers influence the final performance and the training process in the conditional generator, STrans-G. In CIFAR10 with limited data, if we adopt AdaBN-T in STrans-G, the FID starts to deteriorate at the early stage. In contrast, in IMAGENET with large-scale data, AdaIN-T always leads to an unstable training and model divergence. We think that it is because batch normalization layers need amounts of data and large batch sizes to model accurate statistics in the training process. Instance normalization adopting each sample as input cannot benefit from the large batch size in IMAGENET. The quantitative results in Fig. 9(a) also demonstrate that the choice of normalization layers highly relies on the property of the datasets.

A.1.2 STRANS-D

Hybrid Design. We directly adopt the residual downsampling blocks in the StyelGAN2 discriminator as our feature extractor generating input tokens. As shown in the Fig. 8(b), each convolutional block downsamples the input features by $2\times$ times. We apply two successive convolutional blocks outputting the input tokens in a $4\times$ downsampled scale. By stacking Swin Transformer blocks with bilinear downsampling operation, the high-resolution input tokens are gradually downsampled to $(4 \times 4 \times C)$ feature. Note that we discard the original patch merging module in Liu et al. (2021) to avoid a large channel dimension. Containing a convolutional layer and several linear layers, the critic head in Fig. 4(a) outputs the final prediction value.

Equalized Learning Rate. The equalized learning rate (EqLR) is originally designed for scaling the weight at runtime instead of applying careful weight initialization, like $\mathcal{N}(0, 0.02)$. In StyleGAN, Karras et al. (2019) extend EqLR to the style mapping network to adjust the learning rate separately. In STrans-D, we observe that the Transformer blocks need a small learning rate to keep a stable

Table 5: We present the influence of differentiable data augmentation on the final FID. With STrans-G as the generator, we also compare the behavior of StyleGAN2 discriminator ('CNN') and our STrans-D ('Trans').

Dataset	Disc	ADA	
		w/o	w/
CELEBA	CNN	2.41	2.03
	Trans	3.32	3.16
FFHQ	CNN	5.24	4.84
	Trans	6.22	5.98
CIFAR10	CNN	4.07	2.77

training procedure. However, a slow learning speed in the discriminator always brings slow convergence speed in the generator and unsatisfactory generation qualities, which is also a well-known practice (Heusel et al., 2017) in current literature. Thus, to keep the whole discriminator trained with a large learning rate, we adopt EqLR in the sub-layers of the attention and pointwise fully connected operations. In precise, we rescale the learnable weights w_i by multiplying a small scaler c_{lr} in the forward pass:

$$\hat{w}_i = c_{lr} \cdot w_i. \tag{2}$$

Then, the rescaled $\hat{w}_i$ will be used in the subsequent operations like the linear projection.

In our experiments, without EqLR, the config-a and config-b in Fig 7(a) are trained with a small learning rate of 0.0001. Otherwise, the adversarial training will diverge in the early stages. After adopting EqLR, we apply a normal learning rate of 0.002 in the whole discriminator.

Initialization. In STrans-D, we initialize the weights in the convolutional feature extractor and the critic head with standard $\mathcal{N}(0, I)$ distribution. For the Swin Transformer blocks, we still use the truncated normalization following the original implementation. However, we slightly increase the value range of the initialization strategy, because the equalized learning rate module in these blocks contains a learning rate scaler of 0.1. In detail, we select $\mathcal{N}(0, 0.2)$ as the default distribution and truncated the value in the range of $[-10, 10]$.

A.1.3 EFFECTS OF DIFFERENTIABLE DATA AUGMENTATION.

Recent concurrent works (Jiang et al., 2021; Lee et al., 2021; Zhao et al., 2021) claim that differentiable data augmentation (Karras et al., 2020a; Zhao et al., 2020a;c) plays a more important role in Transformer-based generative models than in CNN-based ones. Different from these findings, as shown in Tab. 5, STransGAN with Swin architecture can achieve competitive performance without differentiable data augmentation in regular datasets, *i.e.*, CELEBA, and FFHQ. Even if in CIFAR10 where the training data is highly limited, removing differentiable data augmentation will not cause any serious drop of FID. This implies that our STransGAN can learn the complex image distribution in a data-efficient way.

A.1.4 EVALUATION

For FFHQ and CELEBA dataset, we calculate the Fréchet inception distance (FID) between 50k generated samples and 50k real samples randomly selected from the training set. Following the implementation in Brock et al. (2018), we compute FID with 50k generated samples and the full training set in CIFAR10 and IMAGENET. We also sample 50k images to evaluate IS metric. Note that we report the IS score with the checkpoint obtaining the best FID in all of our experiments.

A.2 ADDITIONAL RESULTS

In Fig. 11, Fig. 10, Fig. 12, and Fig. 13, we present additional samples from STrans-G in FFHQ, CELEBA, CIFAR10, and IMAGENET respectively. We only adopt the truncation tricks for random sampling in IMAGENET dataset. As shown in the visualization results, the local window attention mechanism in the Swin architecture will not bring boundary artifacts in the synthesized images. Meanwhile, the visually pleasant results indicate that STrans-G provides a promising direction of generating high-quality images with pure Transformer models. Figure 14 presents the results for the interpolation of the latent space from STrans-G trained in FFHQ 256^2.

Figure 10: Unconditional FFHQ 256^2 samples from our STrans-G.

Figure 11: Unconditional CELEBA 64^2 samples from our STrans-G.
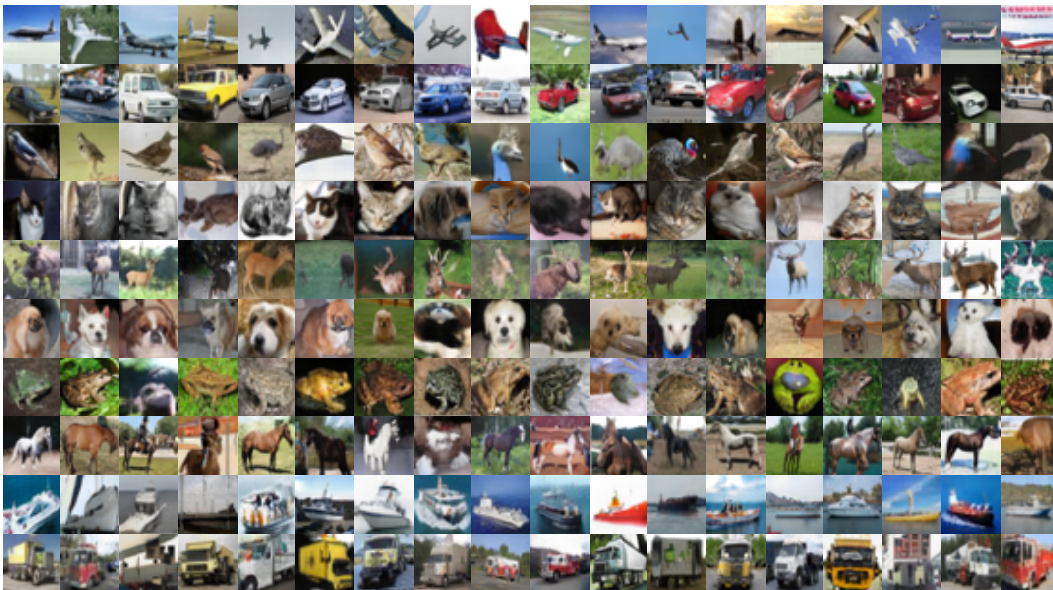

Figure 12: Conditional CIFAR10 32^2 samples from our STrans-G with AdaIN-T. Each row presents the samples from one category in CIFAR10.
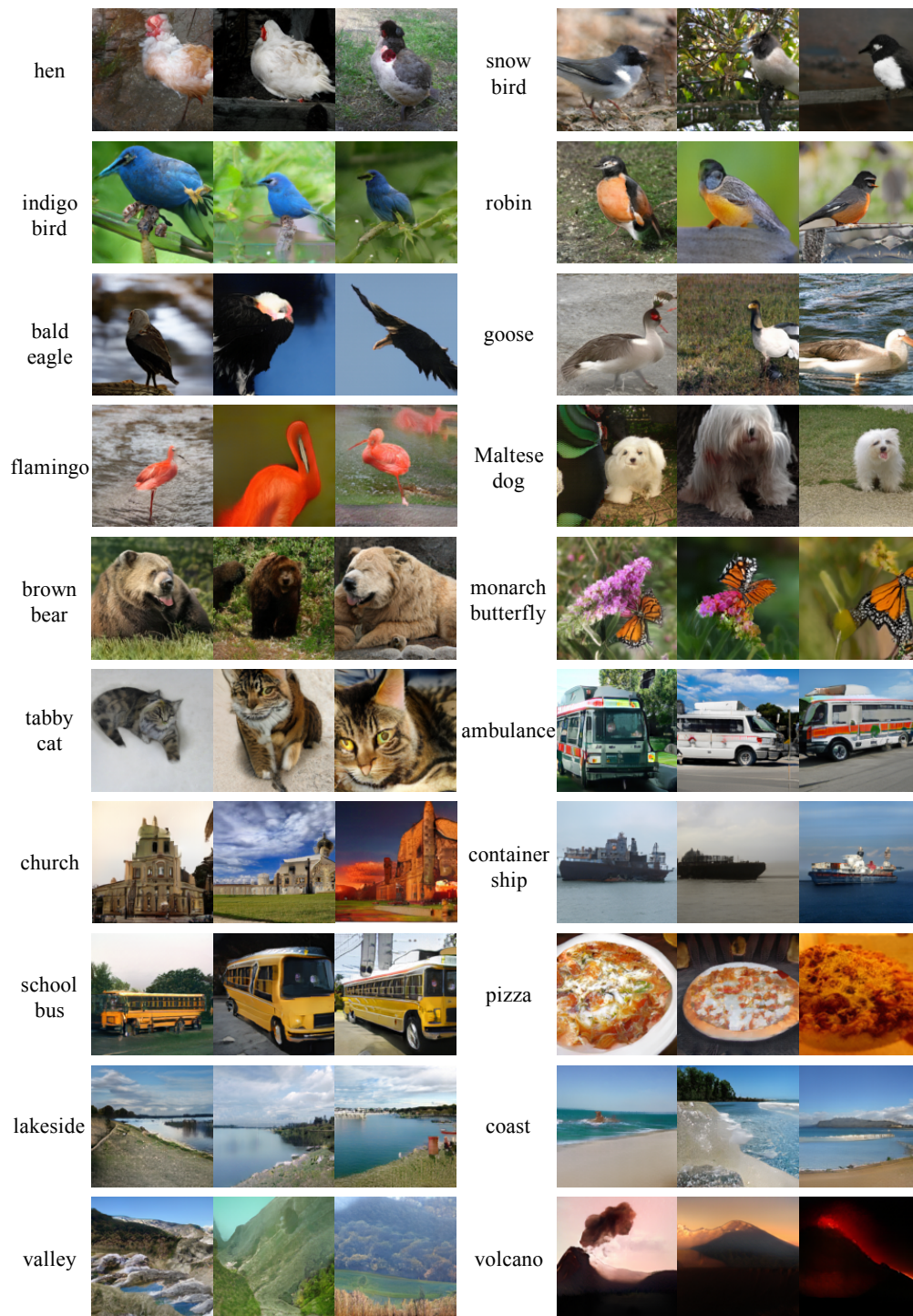
Figure 13: Conditional IMAGENET 128^2 samples from our STrans-G with AdaBN-T. We adopt a truncation threshhold of 0.5 when performing random sampling.

Figure 14: Examples for latent space interpolation in FFHQ 256^2.