

StyleSwin: Transformer-based GAN for High-resolution Image Generation

Bowen Zhang^{1*}Shuyang Gu¹Bo Zhang^{2†}Jianmin Bao²Dong Chen²Fang Wen²Yong Wang¹Baining Guo²¹University of Science and Technology of China²Microsoft Research Asia

Figure 1. Image samples generated by our StyleSwin on FFHQ 1024×1024 and LSUN Church 256×256 respectively.

Abstract

Despite the tantalizing success in a broad of vision tasks, transformers have not yet demonstrated on-par ability as ConvNets in high-resolution image generative modeling. In this paper, we seek to explore using pure transformers to build a generative adversarial network for high-resolution image synthesis. To this end, we believe that local attention is crucial to strike the balance between computational efficiency and modeling capacity. Hence, the proposed generator adopts Swin transformer in a style-based architecture. To achieve a larger receptive field, we propose double attention which simultaneously leverages the context of the local and the shifted windows, leading to improved generation quality. Moreover, we show that offering the knowledge of the absolute position that has been lost in window-based transformers greatly benefits the generation quality. The proposed StyleSwin is scalable to high resolutions, with both the coarse geometry and fine structures benefit from the strong expressivity of transformers. However, blocking artifacts occur during high-resolution synthesis because performing the local attention in a block-wise manner may break the spatial coherency. To solve this, we empirically investigate various solutions, among which we find that employing a wavelet discriminator to examine the spectral discrepancy effectively suppresses the artifacts. Extensive experiments show the superiority over prior transformer-based GANs, especially on high resolutions, e.g., $1024 \times$

1024 . The StyleSwin, without complex training strategies, excels over StyleGAN on CelebA-HQ 1024 , and achieves on-par performance on FFHQ- 1024 , proving the promise of using transformers for high-resolution image generation. The code and pretrained models are available at <https://github.com/microsoft/StyleSwin>.

1. Introduction

The state of image generative modeling has seen dramatic advancement in recent years, among which generative adversarial networks [17, 46] (GANs) offer arguably the most compelling quality on synthesizing high-resolution images. While early attempts focus on stabilizing the training dynamics via proper regularization [18, 19, 41, 51, 52] or adversarial loss designs [2, 30, 44, 50], remarkable performance leaps in recent prominent works mainly attribute to the architectural modifications that aim for stronger modeling capacity, such as adopting self-attention [71], aggressive model scaling [5], or style-based generators [34, 35]. Recently, drawn by the broad success of transformers in discriminative models [13, 37, 48], a few works [29, 42, 67, 72] attempt to use pure transformers to build generative networks in the hope that the increased expressivity and the ability to model long-range dependencies can benefit the generation of complex images, yet high-quality image generation, especially on high resolutions, remains challenging.

This paper aims to explore key ingredients when using transformers to constitute a competitive GAN for high-resolution image generation. The first obstacle is to tame the quadratic computational cost so that the network is scal-

*Author did this work during his internship at Microsoft Research Asia.

†Corresponding author.

able to high resolutions, *e.g.*, 1024×1024 . We propose to leverage Swin transformers [48] as the basic building block since the window-based local attention strikes a balance between computational efficiency and modeling capacity. As such, we could take advantage of the increased expressivity to characterize all the image scales, as opposed to reducing to point-wise multi-layer perceptrons (MLP) for higher scales [72], and the synthesis is scalable to high resolution, *e.g.*, 1024×1024 , with delicate details. Besides, the local attention introduces locality inductive bias so there is no need for the generator to relearn the regularity of images from scratch. These merits make a simple transformer network substantially outperform the convolutional baseline.

In order to compete with the state of the arts, we further propose three instrumental architectural adaptations. First, we strengthen the generative model capacity by employing the local attention in a style-based architecture [34], during which we empirically compare various style injection approaches for our transformer GAN. Second, we propose double attention in order to enlarge the limited receptive field brought by the local attention, where each layer attends to both the local and the shifted windows, effectively improving the generator capacity without much computational overhead. Moreover, we notice that Conv-based GANs implicitly utilize zero padding to infer the absolute positions, a crucial clue for generation, yet such feature is missing in the window-based transformers. We propose to fix this by introducing sinusoidal positional encoding [57] to each layer such that absolute positions can be leveraged for image synthesis. Equipped with the above techniques, the proposed network, dubbed as *StyleSwin*, starts to show advantageous generation quality on 256×256 resolution.

Nonetheless, we observe blocking artifacts when synthesizing high-resolution images. We conjecture that these disturbing artifacts arise because computing the attention independently in a block-wise manner breaks the spatial coherency. That is, while proven successful in discriminative tasks [48, 61], the block-wise attention requires special treatment when applied in synthesis networks. To tackle these blocking artifacts, we empirically investigate various solutions, among which we find that a wavelet discriminator [15] examining the artifacts in spectral domain could effectively suppress the artifacts, making our transformer-based GAN yield visually pleasing outputs.

The proposed *StyleSwin*, achieves state-of-the-art quality on multiple established benchmarks, *e.g.*, FFHQ, CelebA-HQ, and LSUN Church. In particular, our approach shows compelling quality on high-resolution image synthesis (Figure 1), achieving competitive quantitative performance relative to the leading ConvNet-based methods without complex training strategies. On CelebA-HQ 1024, our approach achieves an FID of 4.43, outperforming all the prior works including StyleGAN [34]; whereas on FFHQ-1024, we ob-

tain an FID of 5.07, approaching the performance of StyleGAN2 [35].

2. Related Work

High-resolution image generation. Image generative modeling has improved rapidly in the past decade [17, 24, 39, 40, 46, 60]. Among various solutions, generative adversarial networks (GANs) offer competitive generation quality. While early methods [2, 52, 54] focus on stabilizing the adversarial training, recent prominent works [5, 33–35] rely on designing architectures with enhanced capacity, which considerably improves generation quality. However, contemporary GAN-based methods adopt convolutional backbones which are now deemed to be inferior to transformers in terms of modeling capacity. In this paper, we are interested in applying the emerging vision transformers to GANs for high-resolution image generation.

Vision transformers. Recent success of transformers [6, 62] in NLP tasks inspires the research of vision transforms. The seminal work ViT [13] proposes a pure transformer-based architecture for image classification and demonstrates the great potential of transformers for vision tasks. Later, transformers dominate the benchmarks in a broad of discriminative tasks [12, 20, 48, 58, 61, 64, 65, 69]. However, the self-attention in transformer blocks brings quadratic computational complexity, which limits its application for high-resolution inputs. A few recent works [12, 48, 61] tackle this problem by proposing to compute self-attention in local windows, so that linear computational complexity can be achieved. Moreover, the hierarchical architecture makes them suitable to serve as general purpose backbones.

Transformer-based GANs. Recently, the research community begins to explore using transformers for generative tasks in the hope that the increased expressivity can benefit the generation of complex images. One natural way is to use transformers to synthesize pixels in an auto-regressive manner [7, 14], but the slow inference speed limits their practical usage. Recently a few works [29, 42, 67, 72] attempt to propose transformer-based GANs, yet most of these methods only support the synthesis up to 256×256 resolution. Notably, the HiT [72] successfully generates 1024×1024 images at the cost of reducing to MLPs in its high-resolution stages, hence unable to synthesize high-fidelity details as the Conv-based counterpart [34]. In comparison, our *StyleSwin* can synthesize fine structures using transformers, leading to comparable quality as the leading ConvNets on high-resolution synthesis.

3. Method

3.1. Transformer-based GAN architecture

We start from a simple generator architecture, as shown in Figure 2(a), which receives a latent variable $z \sim \mathcal{N}(0, I)$

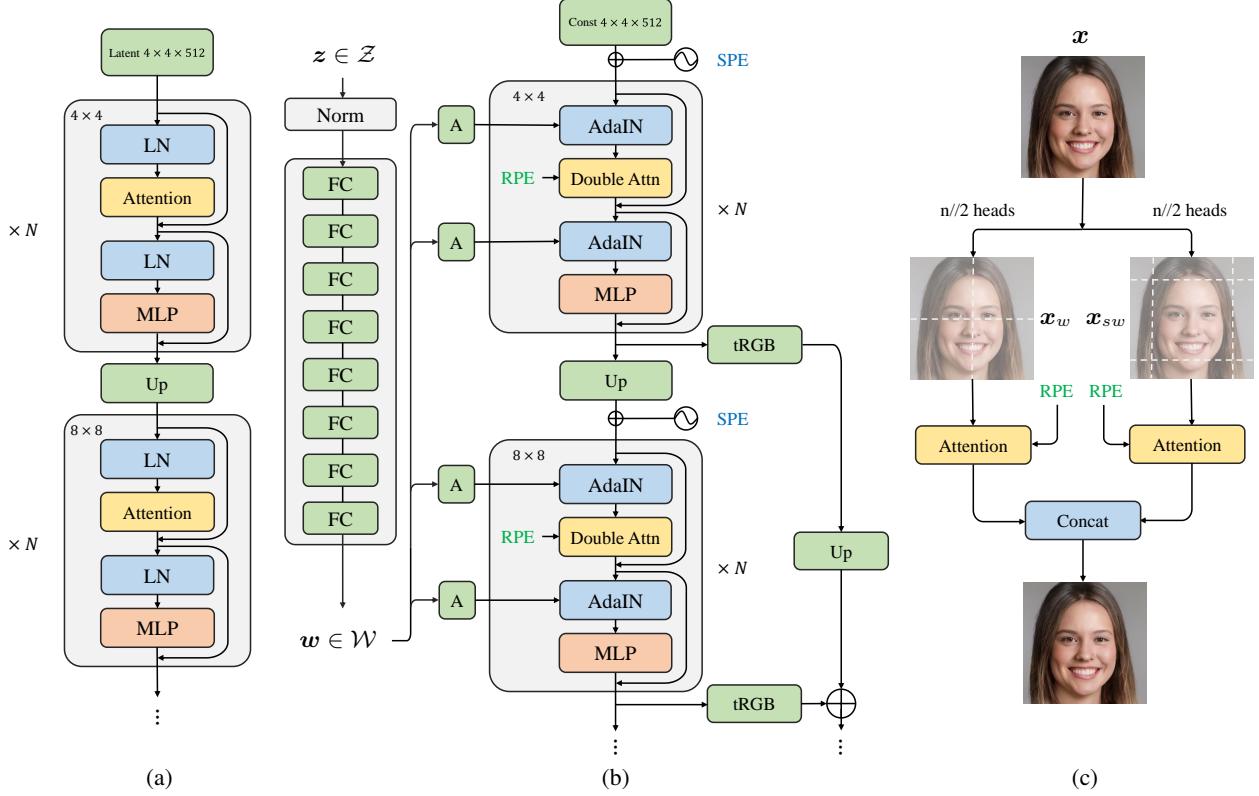


Figure 2. The architectures we investigate. (a) The baseline architecture is comprised of a series of transformer blocks hierarchically. (b) The proposed *StyleSwin* adopts style-based architecture, where the style codes derived from the latent code z modulate the feature maps of transformer blocks through style injection. (c) The proposed double attention enlarges the receptive field of transformer blocks by using split heads attending to the local and the shifted windows respectively.

as input and gradually upsamples the feature maps through a cascade of transformer blocks.

Due to the quadratic computational complexity, it is unaffordable to compute full-attention on high-resolution feature maps. We believe that local attention is a good way to achieve trade-off between computational efficiency and modeling capacity. We adopt Swin transformer [48] as the basic building block which computes multi-head self-attention (MSA) [62] locally in non-overlapping windows. To advocate the information interaction across adjacent windows, Swin transformer uses shifted window partition in alternative blocks. Specifically, given the input feature map $\mathbf{x}^l \in \mathbb{R}^{H \times W \times C}$ of layer l , the consecutive Swin blocks operate as follows:

$$\begin{aligned} \hat{\mathbf{x}}^l &= \text{W-MSA}(\text{LN}(\mathbf{x}^l)) + \mathbf{x}^l \\ \mathbf{x}^{l+1} &= \text{MLP}(\text{LN}(\hat{\mathbf{x}}^l)) + \hat{\mathbf{x}}^l \end{aligned} \left\{ \begin{array}{l} \text{regular window,} \\ \text{shifted window,} \end{array} \right. \quad (1)$$

where W-MSA and SW-MSA denote the window-based

multi-head self-attention under the regular and shifted window partitioning respectively, and LN stands for layer normalization. Since such block-wise attention induces linear computational complexity relative to the image size, the network is scalable to the high-resolution generation where the fine structures can be modeled by these capable transformers as well.

Since the discriminator severely affects the stability of adversarial training, we opt to use a Conv-based discriminator directly from [34]. In our experiment, we find that simply replacing the convolution with transformer blocks under this baseline architecture yields more stabilized training due to the improved model capacity. However, such naive architecture cannot make our transformer-based GAN compete with the state of the arts, so we make further studies which we introduce as follows.

Style injection. We first strengthen the model capability by adapting the generator to a style-based architecture [34, 35] as shown in Figure 2(b). We learn a non-linear mapping $f : \mathcal{Z} \rightarrow \mathcal{W}$ to map the latent code z from \mathcal{Z} space to \mathcal{W} space, which specifies the styles that are injected into the main synthesis network. We investigate the following style

Style injection methods	FID ↓
Baseline	15.03
AdaIN	6.34
AdaLN	6.95
AdaBN	> 100
AdaRMSNorm	7.43
Modulated MLP	7.09
Cross attention	6.59

Table 1. Comparison of different style injection methods on FFHQ-256. The style injection methods considerably improve the FID, among which the AdaIN leads to the best generation quality.

injection approaches:

- *AdaNorm* modulates the statistics (*i.e.*, mean and variance) of feature maps after normalization. We study multiple normalization variants, including instance normalization (IN) [59], batch normalization (BN) [26], layer normalization (LN) [3] and the recently proposed RMSnorm [70]. Since the RMSnorm removes the mean-centering of LN, we only predict the variance from the \mathcal{W} code.
- *Modulated MLP*: Instead of modulating feature maps, one can also modulate the weights of linear layers. Specifically, we rescale the channel-wise weight magnitude of the feed-forward network (FFN) within transformer blocks. According to [35], such style injection admits faster speed than AdaNorm.
- *Cross-attention*: Motivated by the decoder transformer [62], we explore a transformer-specific style injection in which the transformers additionally attend to the style tokens derived from the \mathcal{W} space. The effectiveness of this cross-attention is also validated in [72].

Table 1 shows that all the above style injection methods significantly boost the generative modeling capacity except that the training with AdaBN does not converge because the batch size is compromised for high-resolution synthesis. In comparison, AdaNorm brings more sufficient style injection possibly because the network could take advantage of the style information twice — in either the attention block and the FFN, whereas the modulated MLP and cross-attention make use of the style information once. We did not further study the hybrid of modulated MLP and cross-attention due to efficiency considerations. Furthermore, compared to AdaBN and AdaLN, AdaIN offers finer and more sufficient feature modulation as feature maps are normalized and modulated independently, so we choose AdaIN by default for our following experiments.

Double attention. Using local attention, nonetheless, sacrifices the ability to model long-range dependencies, which is pivotal to capture geometry [5, 71]. Let the window size

used by the Swin block be $\kappa \times \kappa$, then due to the shifted window strategy, the receptive field increases by κ in each dimension using one more Swin block. Suppose we use Swin blocks to process a 64×64 feature map and we by default choose $\kappa = 8$, then it takes $64/\kappa = 8$ transformer blocks to span over the entire feature map.

In order to achieve an enlarged receptive field, we propose *double attention* which allows a single transformer block to simultaneously attend to the context of the local and shifted windows. As illustrated in Figure 2(c), we split h attention heads into two groups: the first half of heads perform the regular window attention whereas the second half compute the shifted window attention, both of whose results are further concatenated to form the output. Specifically, we denote with \mathbf{x}_w and \mathbf{x}_{sw} the non-overlapping patches under the regular and shifted window partitioning respectively, *i.e.* $\mathbf{x}_w, \mathbf{x}_{sw} \in \mathcal{R}^{\frac{HW}{\kappa^2} \times \kappa \times \kappa \times C}$, then the double attention is formulated as,

$$\text{Double-Attention} = \text{Concat}(\text{head}_1, \dots, \text{head}_h) \mathbf{W}^O \quad (2)$$

where $\mathbf{W}^O \in \mathcal{R}^{C \times C}$ is the projection matrix used to mix the heads to output. The attention heads in Equation 2 can be computed as:

$$\text{head}_i = \begin{cases} \text{Attn}(\mathbf{x}_w \mathbf{W}_i^Q, \mathbf{x}_w \mathbf{W}_i^K, \mathbf{x}_w \mathbf{W}_i^V) & i \leq \lfloor \frac{h}{2} \rfloor \\ \text{Attn}(\mathbf{x}_{sw} \mathbf{W}_i^Q, \mathbf{x}_{sw} \mathbf{W}_i^K, \mathbf{x}_{sw} \mathbf{W}_i^V) & i > \lfloor \frac{h}{2} \rfloor \end{cases} \quad (3)$$

where $\mathbf{W}_i^Q, \mathbf{W}_i^K, \mathbf{W}_i^V \in \mathcal{R}^{C \times (C/h)}$ are query, key and value projection matrix for i -th head respectively. One can derive that the receptive field of each dimension increases by 2.5κ with one additional double attention block, which allows capturing larger context more efficiently. Still, for a 64×64 input, it now takes 4 transformer blocks to cover the entire feature map.

Local-global positional encoding. Relative positional encoding (RPE) adopted by the default Swin blocks encodes the relative position of pixels and has proven crucial for discriminative tasks [11, 48]. Theoretically, a multi-head local attention layer with RPE can express any convolutional layer of window-sized kernels [10, 43]. However, when substituting the convolutional layers with transformers that use RPE, one thing is rarely noticed: ConvNets could infer the absolute positions by leveraging the clues from the zero paddings [27, 36] yet such feature is missing in Swin blocks using RPE. On the other hand, it is essential to let the generator be aware of the absolute positions because the synthesis of a specific component, *e.g.*, mouth, highly depends on its spatial coordinate [1, 45].

In view of this, we propose to introduce sinusoidal position encoding [8, 62, 66] (SPE) on each scale, as shown in Figure 2(b). Specifically, after the scale upsampling, the

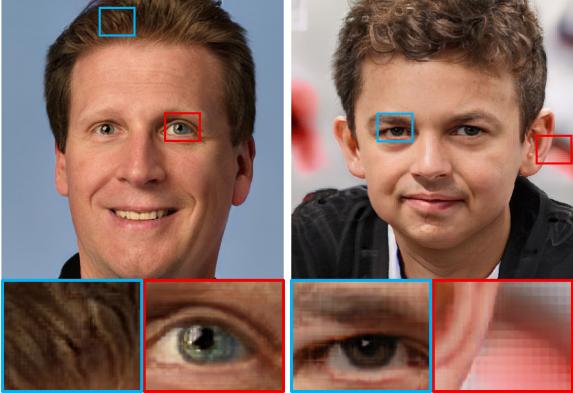


Figure 3. Blocking artifacts become obvious on 1024×1024 resolution. These artifacts correlate with the window size of local attentions.

feature maps are added with the following encoding:

$$\underbrace{[\sin(\omega_0 i), \cos(\omega_0 i), \dots]}_{\text{horizontal dimension}}, \underbrace{[\sin(\omega_0 j), \cos(\omega_0 j), \dots]}_{\text{vertical dimension}} \in \mathbb{R}^C, \quad (4)$$

where and $\omega_k = 1/10000^{2k}$ and (i, j) denotes the 2D location. We use SPE rather than learnable absolute positional encoding [13] because SPE admits translation invariance [63]. In practice, we make the best of RPE and SPE by employing them altogether: the RPE applied within each transformer block offers the relative positions within the local context, whereas the SPE introduced on each scale informs the global position.

3.2. Blocking artifact in high-resolution synthesis

While achieving state-of-the-art quality on synthesizing 256×256 images with the above architecture, directly applying it for higher resolution synthesis, e.g., 1024×1024 , brings blocking artifacts as shown in Figure 3, which severely affects the perceptual quality. Note that these are by no means the checkboard artifacts caused by the transposed convolution [53] as we use bilinear upsampling followed by anti-aliasing filters as [34].

We conjecture that the blocking artifacts are caused by the transformers. To verify this, we remove the attention operators starting from 64×64 and employ only MLPs to characterize the high-frequency details. This time we obtain artifact-free results. To be further, we find that these artifacts exhibit periodic patterns with a strong correlation with the window size of local attention. Hence, we are certain it is the window-wise processing that breaks the spatial coherency and causes the blocking artifacts. To simplify, one can consider the 1D case in Figure 4, where attention is computed locally in strided windows. For a continuous signal, the window-wise local attention is likely to produce a discontinuous output because the values within the same

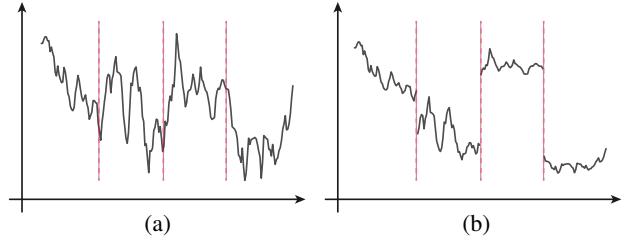


Figure 4. A 1D example illustrates that the window-wise local attention causes blocking artifacts. (a) Input continuous signal along with partitioning windows. (b) Output discontinuous signal after window-wise attention. For simplicity, we adopt one attention head with random projection matrices.

window tend to become uniform after the softmax operation, so the outputs of neighboring windows appear rather distinct. The 2D case is analogous to the JPEG compression artifacts caused by the block-wise encoding [47].

3.3. Artifact suppression

In the next, we discuss a few solutions to suppress the blocking artifacts.

Artifact-free generator. We first attempt to reduce artifacts by improving the generator.

- *Token sharing.* Blocking artifacts arise because there is an abrupt change of keys and values used by the attention computing across distinct windows, so we propose to make windows have shared tokens in a way like HaloNet [61]. However, artifacts are still noticeable since there always exist tokens exclusive to specific windows.
- Theoretically, *sliding window attention* [25] should lead to artifact-free results. Note that training the generator with sliding attention is too costly so we only adopt the sliding window for inference.
- *Reduce to MLPs on fine scales.* Just as [72], one can remove self-attention and purely rely on point-wise MLPs for fine structure synthesis at the cost of sacrificing the ability to model high-frequency details.

Artifact-suppression discriminator. Indeed, we observe blocking artifacts in the early training phase on 256×256 resolution, but they gradually fade out as training precedes. In other words, although the window-based attention is prone to produce artifacts, the generator does have the capability to offer an artifact-free solution. The artifacts plague the high-resolution synthesis because the discriminator fails to examine the high-frequency details. This enlightens us to resort to stronger discriminators for artifact suppression.

- *Patch discriminator* [28] possesses limited receptive field and can be employed to specifically penalize the local structures. Experiments show partial suppression of the blocking artifacts using a patch discriminator.

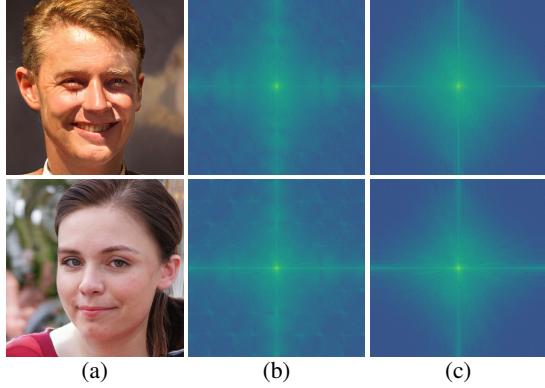


Figure 5. The Fourier spectrum of blocking artifacts. (a) Images with blocking artifacts. (b) The artifacts with periodic patterns can be clearly distinguished in the spectrum. (c) The spectrum of artifact-free images derived from the sliding window inference.

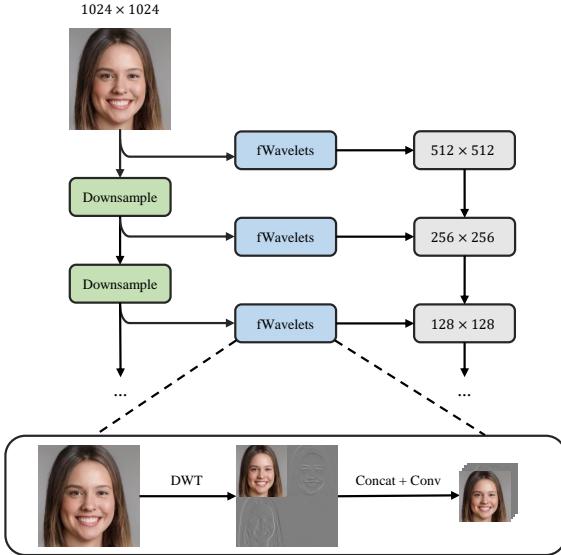


Figure 6. The wavelet discriminator suppresses the artifacts by examining the wavelet spectrum of the multi-scaled input.

- *Total variation annealing.* To advocate smooth outputs, we apply a large total variation loss at the beginning of training, aiming to suppress the network’s tendency to artifacts. The loss weight is then linearly decayed to zero towards the end of training. Though artifacts can be completely removed, such handcrafted constraint favors over-smoothed results and inevitably affects the distribution matching for high-frequency details.
- *Wavelet discriminator.* As shown in Figure 5, the periodic artifact pattern can be easily distinguished in the spectral domain. Inspired by this, we resort to a wavelet discriminator [15] to complement our spatial discriminator and we illustrate its architecture in Figure 6. The discriminator hierarchically downsamples the input image and on each scale examines the frequency discrepancy

Solutions	FID ↓	Remove artifacts?
Window-based attention	8.39	✗
Sliding window inference	12.08	✓
Token sharing	8.95	✗
MLPs after 64×64	12.69	✓
Patch discriminator	7.73	✗
Total variation annealing	12.79	✓
Wavelet discriminator	5.07	✓

Table 2. Comparison of the artifact suppression solutions on FFHQ-1024.

relative to real images after discrete wavelet decomposition. Such a wavelet discriminator works remarkably well in combating the blocking artifacts. Meanwhile, it does not bring any side-effects on distribution matching, effectively guiding the generator to produce rich details.

Table 2 compares the above artifact suppression methods, showing that there are four approaches that could totally remove the visual artifacts. However, sliding window inference suffers from the train-test gap, whereas MLPs fail to synthesize fine details on high-resolution stages, both of them leading to a higher FID score. On the other hand, the total variation with annealing still deteriorates the FID. In comparison, the wavelet-discriminator achieves the lowest FID score and yields the most visually pleasing results.

4. Experiments

4.1. Experiment setup

Datasets. We validate our StyleSwin on the following datasets: CelebA-HQ [32], LSUN Church [68], and FFHQ [34]. CelebA-HQ is a high-quality version of CelebA dataset [49] which contains 30,000 human face images of 1024 × 1024 resolution. FFHQ [34] is a commonly used dataset for high-resolution image generation. It contains 70,000 high-quality human face images with more variation of age, ethnicity and background, and has better coverage of accessories such as eyeglasses, sunglasses, hats, etc. We synthesize images on FFHQ and CelebA-HQ on either 256 × 256 and 1024 × 1024 resolutions. LSUN Church [68] contains around 126,000 church images in diverse architecture styles, on which we conduct experiments with 256 × 256 resolution.

Evaluation protocol. We adopt Fréchet Inception Distance (FID) [23] as the quantitative metric, which measures the distribution discrepancy between generated images and real ones. Lower FID scores indicate better generation quality. For FFHQ [34] and LSUN Church [68] datasets, we randomly sample 50,000 images from the original datasets as validation sets and calculate FID between the validation sets and 50,000 generated images. For CelebA-HQ [32], we cal-

Methods	FFHQ	CelebA-HQ	LSUN Church
StyleGAN2 [35]	3.62*	-	3.86
PG-GAN [32]	-	8.03	6.42
U-Net GAN [55]	7.63	-	-
INR-GAN [56]	9.57	-	5.09
MSG-GAN [31]	-	-	5.20
CIPS [1]	4.38	-	2.92
TransGAN [29]	-	9.60*	8.94
VQGAN [14]	11.40	10.70	-
HiT-B [72]	2.95*	3.39*	-
<i>StyleSwin</i>	2.81*	3.25*	2.95

Table 3. Comparison of state-of-the-art unconditional image generation methods on FFHQ, CelebA-HQ and LSUN Church of 256×256 resolution in terms of FID score (lower is better). The subscript (*) indicates that bCR is applied during training.

culated the FID between 30,000 generated images and all the training samples.

4.2. Implementation details

During training we use Adam solver [38] with $\beta_1 = 0.0$, $\beta_2 = 0.99$. Following TTUR [23], we set imbalanced learning rates, $5e-5$ and $2e-4$, for the generator and discriminator respectively. We train our model using the standard non-saturating GAN loss with R_1 gradient penalty [35] and stabilize the adversarial training by applying spectral normalization [52] on the discriminator. By default, we report all the results with the wavelet discriminator as shown in Figure 6. Using 8 32GB V100 GPUs, we are able to fit 32 images as one training batch for the training on 256×256 resolution and the batch size reduces to 16 on 1024×1024 resolution. For fair comparison with prior works, we report the FID with balanced consistency regularization (bCR) [75] on the FFHQ-256 and CelebA-HQ 256 datasets with the loss weight $\lambda_{\text{real}} = \lambda_{\text{fake}} = 10$. Similar to [72], we do not observe performance gain using bCR on higher resolutions. Note that we do not adopt complex training strategies, such as path length and mixing regularizations [34], as we wish to conduct studies on neat network architectures.

4.3. Main results

Quantitative results. We compare with state-of-the-art Conv-based GANs as well as the recent transformer-based methods. As shown in Table 3, our StyleSwin achieves state-of-the-art FID scores on all the 256×256 synthesis. In particular, on both FFHQ and LSUN Church datasets, StyleSwin outperforms StyleGAN2 [35]. Besides the impressive results on resolution 256×256 , the proposed StyleSwin shows a strong capability on high-resolution image generation. As shown in Table 4, we evaluate models on FFHQ and CelebA-HQ on the resolution of 1024×1024 ,

Methods	FFHQ-1024	CelebA-HQ 1024
StyleGAN ¹ [35] [34]	4.41	5.06
COCO-GAN	-	9.49
PG-GAN [32]	-	7.30
MSG-GAN [31]	5.80	6.37
INR-GAN [56]	16.32	-
CIPS [1]	10.07	-
HiT-B [72]	6.37	8.83
<i>StyleSwin</i>	5.07	4.43

Table 4. Comparison of state-of-the-art unconditional image generation methods on FFHQ and CelebA-HQ of resolution 1024×1024 in terms of FID score (lower is better). ¹We report the FID score of StyleGAN2 on FFHQ-1024 and that of StyleGAN on CelebA-HQ 1024. For fair comparison, we report results of StyleGAN2 without style-mixing and path regularization.

Model Configuration	FID \downarrow
A. Swin baseline	15.03
B. + Style injection	8.40
C. + Double attention	7.86
D. + Wavelet discriminator	6.34
E. + SPE	5.76
F. + Larger model	5.50
G. + bCR	2.81

Table 5. Ablation study conducted on FFHQ-256. Starting from the baseline architecture, we prove the effectiveness of each proposed component.

where the proposed StyleSwin also demonstrates state-of-the-art performance. Notably, we obtain the record FID score of 4.43 on CelebA-HQ 1024 dataset while considerably closing the gap with the leading approach StyleGAN2 without involving complex training strategies or additional regularization. Also, StyleSwin outperforms the transformer-based approach HiT by a large margin on 1024×1024 resolution, proving that the self-attention on high-resolution stages is beneficial to high-fidelity detail synthesis.

Qualitative results. Figure 7 shows the image samples generated by StyleSwin on FFHQ and CelebA-HQ of 1024×1024 resolution. Our StyleSwin shows compelling quality on synthesizing diverse images of different ages, backgrounds and viewpoints on the resolution of 1024×1024 . On top of face modeling, we show generation results of LSUN Church in Figure 8, showing StyleSwin is capable to model complex scene structures. Both the coherency of global geometry and the high-fidelity details all prove the advantages of using transformers among all the resolutions.

Ablation study. To validate the effectiveness of the proposed components, we conduct ablation studies in Table 5. Compared with the baseline architecture, we observe sig-



Figure 7. Image samples generated by our StyleSwin on (a) FFHQ 1024×1024 and (b) CelebA-HQ 1024×1024 .



Figure 8. Image samples generated by our StyleSwin on LSUN Church 256×256 .

nificant FID improvement thanks to the enhanced model capacity brought by the style injection. The double attention makes each layer leverage larger context at one time and further reduces the FID score. Wavelet discriminator brings a large FID improvement because it effectively suppresses the blocking artifacts and meanwhile brings stronger supervision for high-frequencies. In our experiment, we observe faster adversarial training when adopting the wavelet discriminator. Further, introducing sinusoidal positional encoding (SPE) on each generation scale effectively reduces the FID. Employing a larger model brings slight improvement and it seems that the model capacity of the current transformer structure is not the bottleneck. From Table 5 we see that bCR considerably improves the FID by 2.69, which coincides with the recent findings [29, 42, 72] that data augmentation is still vital in transformer-based GAN since transformers are data-hungry and prone to overfitting. However, we do not observe its effectiveness on higher resolutions, *e.g.*, 1024×1024 , and we leave regularization

Methods	#params	FLOPs
StyleGAN2 [35]	30.37M	74.27B
<i>StyleSwin</i>	40.86M	50.90B

Table 6. Comparison of the network parameters and FLOPs with StyleGAN2.

schemes for high-resolution synthesis to future work.

Parameters and Throughput. In Table 6, We compare the number of model parameters and FLOPs with StyleGAN2 for 1024×1024 synthesis. Although our approach has a larger model size, it achieves lower FLOPs than StyleGAN2, which means the method achieves competitive generation quality with less theoretical computational cost.

5. Conclusion

We propose StyleSwin, a transformer-based GAN for high-resolution image generation. The use of local attention is efficient to compute while attaining most modeling capability since the receptive field is largely compensated by double attention. Besides, we find one key feature is missing in transformer-based GANs — the generator is not aware of the position for patches under synthesis, so we introduce SPE for global positioning. Thanks to the increased expressivity, the proposed StyleSwin consistently outperforms the leading Conv-based approaches on 256×256 datasets. To solve the blocking artifacts on high-resolution synthesis, we propose to penalize the spectral discrepancy with a wavelet discriminator [15]. Ultimately, the proposed StyleSwin offers compelling quality on the resolution of 1024×1024 , which for the first time, approaches the best performed ConvNets. Our work hopefully incentives more studies on utilizing the capable transformers in generative modeling.

References

- [1] Ivan Anokhin, Kirill Demochkin, Taras Khakhulin, Gleb Sterkin, Victor Lempitsky, and Denis Korzenkov. Image generators with conditionally-independent pixel synthesis, 2020. [4](#) [7](#)
- [2] Martín Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. *ArXiv*, abs/1701.07875, 2017. [1](#) [2](#)
- [3] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. [4](#)
- [4] Mikołaj Bińkowski, Danica J Sutherland, Michael Arbel, and Arthur Gretton. Demystifying mmd gans. *arXiv preprint arXiv:1801.01401*, 2018. [14](#)
- [5] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *ArXiv*, abs/1809.11096, 2019. [1](#) [2](#) [4](#)
- [6] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020. [2](#)
- [7] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pre-training from pixels. In *International Conference on Machine Learning*, pages 1691–1703. PMLR, 2020. [2](#)
- [8] Jooyoung Choi, Jungbeom Lee, Yonghyun Jeong, and Sungroh Yoon. Toward spatially unbiased generative models. *arXiv preprint arXiv:2108.01285*, 2021. [4](#)
- [9] Min Jin Chong and David Forsyth. Effectively unbiased fid and inception score and where to find them. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6070–6079, 2020. [14](#)
- [10] Jean-Baptiste Cordonnier, Andreas Loukas, and Martin Jaggi. On the relationship between self-attention and convolutional layers. *ArXiv*, abs/1911.03584, 2020. [4](#)
- [11] Zihang Dai, Hanxiao Liu, Quoc V Le, and Mingxing Tan. Coatnet: Marrying convolution and attention for all data sizes. *arXiv preprint arXiv:2106.04803*, 2021. [4](#)
- [12] Xiaoyi Dong, Jianmin Bao, Dongdong Chen, Weiming Zhang, Nenghai Yu, Lu Yuan, Dong Chen, and Baining Guo. Cswin transformer: A general vision transformer backbone with cross-shaped windows, 2021. [2](#)
- [13] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ArXiv*, abs/2010.11929, 2021. [1](#) [2](#) [5](#)
- [14] Patrick Esser, Robin Rombach, and Björn Ommer. Taming transformers for high-resolution image synthesis, 2020. [2](#) [7](#)
- [15] Rinon Gal, Dana Cohen, Amit Bermano, and Daniel Cohen-Or. Swagan: A style-based wavelet-driven generative model, 2021. [2](#) [6](#) [8](#)
- [16] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256. JMLR Workshop and Conference Proceedings, 2010. [12](#)
- [17] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, 2014. [1](#) [2](#) [12](#)
- [18] Shuyang Gu, Jianmin Bao, Dong Chen, and Fang Wen. Giqa: Generated image quality assessment. In *European Conference on Computer Vision*, pages 369–385. Springer, 2020. [1](#)
- [19] Ishaan Gulrajani, Faruk Ahmed, Martín Arjovsky, Vincent Dumoulin, and Aaron C. Courville. Improved training of wasserstein gans. In *NIPS*, 2017. [1](#)
- [20] Kai Han, An Xiao, Enhua Wu, Jianyuan Guo, Chunjing Xu, and Yunhe Wang. Transformer in transformer. *arXiv preprint arXiv:2103.00112*, 2021. [2](#)
- [21] Boris Hanin and David Rolnick. How to start training: The effect of initialization and architecture, 2018. [12](#)
- [22] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning, 2020. [12](#)
- [23] 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, 2018. [6](#) [7](#) [12](#)
- [24] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *arXiv preprint arXiv:2006.11239*, 2020. [2](#)
- [25] Han Hu, Zheng Zhang, Zhenda Xie, and Stephen Lin. Local relation networks for image recognition, 2019. [5](#)
- [26] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015. [4](#)
- [27] Md Amirul Islam, Sen Jia, and Neil DB Bruce. How much position information do convolutional neural networks encode? *arXiv preprint arXiv:2001.08248*, 2020. [4](#)
- [28] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. *CVPR*, 2017. [5](#)
- [29] Yifan Jiang, Shiyu Chang, and Zhangyang Wang. Transgan: Two transformers can make one strong gan. *arXiv preprint arXiv:2102.07074*, 2021. [1](#) [2](#) [7](#) [8](#)
- [30] Alexia Jolicoeur-Martineau. The relativistic discriminator: a key element missing from standard gan. *ArXiv*, abs/1807.00734, 2019. [1](#)
- [31] Animesh Karnewar and Oliver Wang. Msg-gan: Multi-scale gradients for generative adversarial networks. *arXiv preprint arXiv:1903.06048*, 2019. [7](#)
- [32] 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. [6](#) [7](#) [12](#)
- [33] Tero Karras, Miika Aittala, Samuli Laine, Erik Häkkinen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks. *arXiv preprint arXiv:2106.12423*, 2021. [2](#)

- [34] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks, 2019. 1, 2, 3, 5, 6, 7, 12
- [35] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8107–8116, 2020. 1, 2, 3, 4, 7, 8
- [36] Osman Semih Kayhan and Jan C van Gemert. On translation invariance in cnns: Convolutional layers can exploit absolute spatial location. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14274–14285, 2020. 4
- [37] Salman Hameed Khan, Muzammal Naseer, Munawar Hayat, Syed Waqas Zamir, Fahad Shahbaz Khan, and Mubarak Shah. Transformers in vision: A survey. *ArXiv*, abs/2101.01169, 2021. 1
- [38] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017. 7
- [39] Diederik P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. *arXiv preprint arXiv:1807.03039*, 2018. 2
- [40] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 2
- [41] Karol Kurach, Mario Lucic, Xiaohua Zhai, Marcin Michalski, and Sylvain Gelly. A large-scale study on regularization and normalization in gans. In *ICML*, 2019. 1
- [42] Kwonjoon Lee, Huiwen Chang, Lu Jiang, Han Zhang, Zhiowen Tu, and Ce Liu. Vitgan: Training gans with vision transformers. *ArXiv*, abs/2107.04589, 2021. 1, 2, 8
- [43] Shanda Li, Xiangning Chen, Di He, and Cho-Jui Hsieh. Can vision transformers perform convolution? *ArXiv*, abs/2111.01353, 2021. 4
- [44] Jae Hyun Lim and J. C. Ye. Geometric gan. *ArXiv*, abs/1705.02894, 2017. 1
- [45] Chieh Hubert Lin, Chia-Che Chang, Yu-Sheng Chen, Da-Cheng Juan, Wei Wei, and Hwann-Tzong Chen. Cocogan: Generation by parts via conditional coordinating. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4512–4521, 2019. 4
- [46] Ming-Yu Liu, Xun Huang, Jiahui Yu, Ting-Chun Wang, and Arun Mallya. Generative adversarial networks for image and video synthesis: Algorithms and applications. *arXiv preprint arXiv:2008.02793*, 2020. 1, 2
- [47] Shizhong Liu and Alan C Bovik. Efficient dct-domain blind measurement and reduction of blocking artifacts. *IEEE Transactions on Circuits and Systems for Video Technology*, 12(12):1139–1149, 2002. 5
- [48] 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, 2021. 1, 2, 3, 4
- [49] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015. 6
- [50] Xudong Mao, Qing Li, Haoran Xie, Raymond Y. K. Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2813–2821, 2017. 1
- [51] Lars M. Mescheder, Andreas Geiger, and Sebastian Nowozin. Which training methods for gans do actually converge? In *ICML*, 2018. 1, 12
- [52] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *ArXiv*, abs/1802.05957, 2018. 1, 2, 7, 12
- [53] Augustus Odena, Vincent Dumoulin, and Chris Olah. Deconvolution and checkerboard artifacts. *Distill*, 1(10):e3, 2016. 5
- [54] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015. 2
- [55] Edgar Schonfeld, Bernt Schiele, and Anna Khoreva. A u-net based discriminator for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8207–8216, 2020. 7
- [56] Ivan Skorokhodov, Savva Ignatyev, and Mohamed Elhoseiny. Adversarial generation of continuous images, 2021. 7
- [57] Matthew Tancik, Pratul P Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *arXiv preprint arXiv:2006.10739*, 2020. 2
- [58] 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*, pages 10347–10357. PMLR, 2021. 2
- [59] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016. 4
- [60] Aaron Van Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. In *International Conference on Machine Learning*, pages 1747–1756. PMLR, 2016. 2
- [61] Ashish Vaswani, Prajit Ramachandran, Aravind Srinivas, Niki Parmar, Blake Hechtman, and Jonathon Shlens. Scaling local self-attention for parameter efficient visual backbones. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12894–12904, 2021. 2, 5
- [62] Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *ArXiv*, abs/1706.03762, 2017. 2, 3, 4
- [63] Benyou Wang, Lifeng Shang, Christina Lioma, Xin Jiang, Hao Yang, Qun Liu, and Jakob Grue Simonsen. On position embeddings in bert. In *International Conference on Learning Representations*, 2020. 5

- [64] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. *arXiv preprint arXiv:2102.12122*, 2021. [2](#)
- [65] Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. Cvt: Introducing convolutions to vision transformers. *arXiv preprint arXiv:2103.15808*, 2021. [2](#)
- [66] Rui Xu, Xintao Wang, Kai Chen, Bolei Zhou, and Chen Change Loy. Positional encoding as spatial inductive bias in gans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13569–13578, 2021. [4](#)
- [67] Rui Xu, Xiangyu Xu, Kai Chen, Bolei Zhou, and Chen Change Loy. Stransgan: An empirical study on transformer in gans. *ArXiv*, abs/2110.13107, 2021. [1](#), [2](#)
- [68] Fisher Yu, Yinda Zhang, Shuran Song, Ari Seff, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015. [6](#)
- [69] Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Zihang Jiang, Francis EH Tay, Jiashi Feng, and Shuicheng Yan. Tokens-to-token vit: Training vision transformers from scratch on imagenet. *arXiv preprint arXiv:2101.11986*, 2021. [2](#)
- [70] Biao Zhang and Rico Sennrich. Root mean square layer normalization. In *NeurIPS*, 2019. [4](#)
- [71] Han Zhang, Ian J. Goodfellow, Dimitris N. Metaxas, and Augustus Odena. Self-attention generative adversarial networks. In *ICML*, 2019. [1](#), [4](#)
- [72] Long Zhao, Zizhao Zhang, Ting Chen, Dimitris N. Metaxas, and Hang Zhang. Improved transformer for high-resolution gans. *ArXiv*, abs/2106.07631, 2021. [1](#), [2](#), [4](#), [5](#), [7](#), [8](#)
- [73] Nanxuan Zhao, Zhirong Wu, Rynson WH Lau, and Stephen Lin. What makes instance discrimination good for transfer learning? *arXiv preprint arXiv:2006.06606*, 2020. [12](#)
- [74] Shengyu Zhao, Zhijian Liu, Ji Lin, Jun-Yan Zhu, and Song Han. Differentiable augmentation for data-efficient gan training, 2020. [12](#)
- [75] Zhengli Zhao, Sameer Singh, Honglak Lee, Zizhao Zhang, Augustus Odena, and Han Zhang. Improved consistency regularization for gans, 2020. [7](#), [12](#)

Appendix A. Implementation Details

We train the StyleSwin using the standard non-saturating logistic GAN loss [17] with R_1 gradient penalty [51]. Specifically, the discriminator is trained to measure the realism of image samples whereas the generator is trained to generate samples that the discriminator mistakenly recognizes as real ones. In addition, the R_1 regularization term penalizes the gradient on real data to advocate the local stability. The training loss can be formulated as:

$$\begin{aligned}\mathcal{L}_D &= -\mathbb{E}_{x \sim P_x} [\log(D(x))] - \mathbb{E}_{z \sim P_z} [\log(1 - D(G(z)))] + \gamma \cdot \mathbb{E}_{x \sim P_x} [\|\nabla_x D(x)\|_2^2], \\ \mathcal{L}_G &= -\mathbb{E}_{z \sim P_z} [\log(D(G(z)))].\end{aligned}$$

In practice, we perform R_1 gradient penalty every 16 iterations and the corresponding weight γ varies for different datasets.

The training follows the TTUR strategy [23] in which the discriminator adopts a $4 \times$ larger learning rate than the generator. We linearly decay the learning rate to 0 from the LR decay start iteration for training all datasets except CelebA-HQ 1024. We apply spectral normalization [52] upon discriminator to ensure its Lipschitz continuity. The transformers are initialized with a truncated normal distribution [21] with zero mean and standard deviation of 0.02. For the convolution 1×1 used in tRGB layers, we use Glorot initialization [16] with a gain of 0.02. We use an exponential moving average of weights of generator [32] when sampling image, with a decay rate of 0.9978 following [34].

When synthesizing 256×256 resolution images of FFHQ and CelebA-HQ, the training benefits from balanced consistency regularization (bCR) [75]. Specifically, images are augmented by $\{\text{Flipping}, \text{Color}, \text{Translation}, \text{Cutout}\}$ of probability $\{0.5, 1.0, 1.0, 1.0\}$ as in DiffAug [74]. *Translation* is performed within $[-1/8, 1/8]$ of the image size, and random squares of half image size are masked when applying *Cutout*.

We implement the StyleSwin using Pytorch and conduct experiments with Tesla V100 GPUs. Training on 1024×1024 resolution takes about 14 days using 8 32GB GPUs. The hyper-parameters in the experiments are summarized in Table 7.

	FFHQ-256	CelebA-HQ 256	LSUN Church 256	FFHQ-1024	CelebA-HQ 1024
Training iteration	32.0M	25.6M	48M	25.6M	25.6M
Number of GPUs	8	8	8	16	16
Batch size	32	32	32	32	32
Learning rate of D	$2e - 4$	$2e - 4$	$2e - 4$	$2e - 4$	$2e - 4$
Learning rate of G	$5e - 5$	$5e - 5$	$5e - 5$	$5e - 5$	$5e - 5$
LR decay start iteration	24.8M	16M	41.6M	19.2M	-
R_1 regularization γ	10	5	5	10	10
bCR	✓	✓	✗	✗	✗

Table 7. Experiment settings for different datasets.

Appendix B. Detailed Architecture

StyleSwin starts from a constant input of size $4 \times 4 \times 512$ and hierarchically upsamples the feature map with transformer blocks. We use two transformer blocks to model each resolution scale. The detailed model architecture is shown in Table 8. “Double attn, 512-d, 4-w, 16-h” indicates a double attention block with a channel dimension of 512, window size of 4, and 16 attention heads. “Bilinear upsampling, 512-d” indicates a bilinear upsampling layer followed by feedforward MLPs with an output dimension of 512.

Appendix C. The Modeling Capacity of Double Attention

In order to prove the improved expressivity of the proposed double attention, we train an autoencoder for image reconstruction. Specifically, we adopt a conv-based encoder — a ResNet-50 pretrained from MoCo [22] such that both the low-level and high-level information are well preserved in the 16×16 feature map [73]. The latent feature map is further fed into the decoder for image reconstruction. The decoder adopts transformer blocks, which hierarchically upsamples the latent feature map and reconstructs the input. No style injection module is needed and we replace AdaIN with layer normalization. The decoder adopts either the vanilla Swin attention block or the proposed double attention. The autoencoders are trained with L_1 loss. Figure 9 shows the training loss curve of the two autoencoders. One can see that the decoder with double attention shows faster convergence and yields lower reconstruction loss, indicating that the decoder that leverages enhanced receptive field shows stronger generative capacity.

Input size	StyleSwin-256	StyleSwin-1024
4×4	$\left\{ \begin{array}{l} \text{Double attn, 512-d, 4-w, 16-h} \\ \text{MLP, 512-d} \end{array} \right\} \times 2$	$\left\{ \begin{array}{l} \text{Double attn, 512-d, 4-w, 16-h} \\ \text{MLP, 512-d} \end{array} \right\} \times 2$
	Bilinear upsampling, 512-d	Bilinear upsampling, 512-d
8×8	$\left\{ \begin{array}{l} \text{Double attn, 512-d, 8-w, 16-h} \\ \text{MLP, 512-d} \end{array} \right\} \times 2$	$\left\{ \begin{array}{l} \text{Double attn, 512-d, 8-w, 16-h} \\ \text{MLP, 512-d} \end{array} \right\} \times 2$
	Bilinear upsampling, 512-d	Bilinear upsampling, 512-d
16×16	$\left\{ \begin{array}{l} \text{Double attn, 512-d, 8-w, 16-h} \\ \text{MLP, 512-d} \end{array} \right\} \times 2$	$\left\{ \begin{array}{l} \text{Double attn, 512-d, 8-w, 16-h} \\ \text{MLP, 512-d} \end{array} \right\} \times 2$
	Bilinear upsampling, 512-d	Bilinear upsampling, 512-d
32×32	$\left\{ \begin{array}{l} \text{Double attn, 512-d, 8-w, 16-h} \\ \text{MLP, 512-d} \end{array} \right\} \times 2$	$\left\{ \begin{array}{l} \text{Double attn, 512-d, 8-w, 16-h} \\ \text{MLP, 512-d} \end{array} \right\} \times 2$
	Bilinear upsampling, 512-d	Bilinear upsampling, 256-d
64×64	$\left\{ \begin{array}{l} \text{Double attn, 512-d, 8-w, 16-h} \\ \text{MLP, 512-d} \end{array} \right\} \times 2$	$\left\{ \begin{array}{l} \text{Double attn, 256-d, 8-w, 8-h} \\ \text{MLP, 256-d} \end{array} \right\} \times 2$
	Bilinear upsampling, 256-d	Bilinear upsampling, 128-d
128×128	$\left\{ \begin{array}{l} \text{Double attn, 256-d, 8-w, 8-h} \\ \text{MLP, 256-d} \end{array} \right\} \times 2$	$\left\{ \begin{array}{l} \text{Double attn, 128-d, 8-w, 4-h} \\ \text{MLP, 128-d} \end{array} \right\} \times 2$
	Bilinear upsampling, 128-d	Bilinear upsampling, 64-d
256×256	$\left\{ \begin{array}{l} \text{Double attn, 128-d, 8-w, 4-h} \\ \text{MLP, 128-d} \end{array} \right\} \times 2$	$\left\{ \begin{array}{l} \text{Double attn, 64-d, 8-w, 4-h} \\ \text{MLP, 64-d} \end{array} \right\} \times 2$
		Bilinear upsampling, 32-d
512×512		$\left\{ \begin{array}{l} \text{Double attn, 32-d, 8-w, 4-h} \\ \text{MLP, 32-d} \end{array} \right\} \times 2$
		Bilinear upsampling, 16-d
1024×1024		$\left\{ \begin{array}{l} \text{Double attn, 16-d, 8-w, 4-h} \\ \text{MLP, 16-d} \end{array} \right\} \times 2$

Table 8. The detailed generator architecture of StyleSwin-256 and StyleSwin-1024.

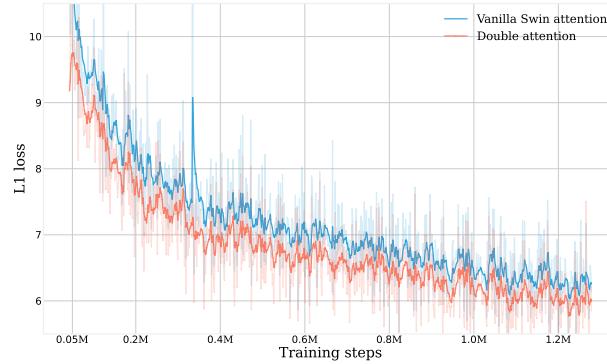


Figure 9. Image reconstruction training loss of autoencoders. The autoencoder adopts a fixed conv-based encoder and transformer-based decoder and is trained with \mathcal{L}_1 loss. The decoder with double attention shows improved modeling capacity over the vanilla Swin attention.

Appendix D. Additional Quantitative Evaluation

To further demonstrate StyleSwin’s strong ability to model complex scenes and materials, we train our model on a subset of LSUN Car, which achieves comparable performance to state-of-the-art StyleGAN2. We also present additional quantitative evaluation results in terms of KID [4] and FID-Inf [9] on all evaluation datasets, comparing to StyleGAN2. The detailed measures are presented in Table 9 and Table 10.

Methods	FFHQ-256			Church-256			CelebAHQ-256			Car-256		
	FID	KID $\times 10^{-3}$	FID-Inf	FID	KID $\times 10^{-3}$	FID-Inf	FID	KID $\times 10^{-3}$	FID-Inf	FID	KID $\times 10^{-3}$	FID-Inf
StyleGAN2	3.62	1.45	1.37	3.86	1.71	1.53	-	-	-	4.32	1.63	1.60
<i>StyleSwin</i>	2.81	0.54	0.83	2.95	1.02	1.44	3.25	0.61	1.36	4.35	1.53	1.80

Table 9. Evaluation results comparing to StyleGAN2 on resolution 256 in terms of FID, KID and FID-Inf.

Methods	FFHQ-1024			CelebAHQ-1024		
	FID	KID $\times 10^{-3}$	FID-Inf	FID	KID $\times 10^{-3}$	FID-Inf
StyleGAN2 ¹	4.41	1.22	1.57	5.17	1.71	1.53
<i>StyleSwin</i>	5.07	2.07	2.13	4.43	1.42	2.08

Table 10. Evaluation results comparing to StyleGAN2 on resolution 1024 in terms of FID, KID and FID-Inf. ¹We report the metrics of StyleGAN2 on FFHQ-1024 and that of StyleGAN on CelebA-HQ 1024.

Appendix E. More Qualitative Results

Latent code interpolation. To explore the property of the learned latent space of StyleSwin, we randomly sample two latent codes in the latent space and perform linear interpolation between them. As shown in Figure 10, our StyleSwin could produce smooth, meaningful image morphing with respect to different styles like gender, poses, and eyeglasses.

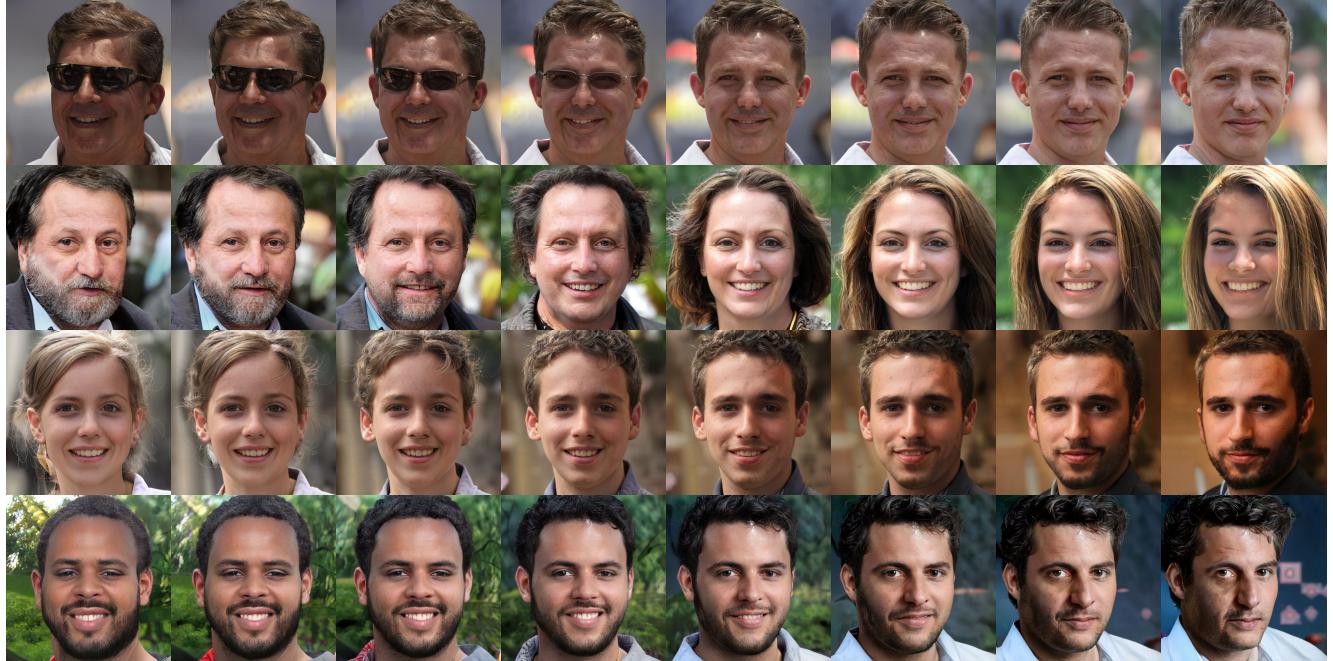


Figure 10. Latent interpolation results of the left-most and the right-most images on FFHQ 1024 \times 1024.

Additional image samples. We provide additional image samples generated by our StyleSwin. Figure 11 and Figure 12 show the impressive synthetic face images of FFHQ-1024 and CelebA-HQ 1024 with diverse viewpoints, backgrounds, and

accessories, which illustrate the strong capacity of the proposed StyleSwin. Image samples of LSUN Church 256 and LSUN Car 256 are shown in Figure 13 and Figure 14, showing that our StyleSwin is capable to synthesize complex scenes with coherent structures and complicated materials with high-quality light effects.

Appendix F. Responsible AI Considerations

Our work does not directly modify the exiting images which may alter the identity or expression of the people. We discourage the use of our work in such applications as it is not designed to do so. We have quantitatively verified that the proposed method does not show evident disparity, on gender and ages as the model mostly follows the dataset distribution, however, we encourage additional care if you intend to use the system on certain demographic groups. We also encourage use of fair and representative data when training on customized data. We caution that the high-resolution images produced by our model may potentially be misused for impersonating humans and viable solutions so avoid this include adding tags or watermarks when distributing the generated photos.

Appendix G. Discussion of Limitation

Although, as stated in the main article, StyleSwin’s theoretical FLOPs are smaller than StyleGAN2, there is a gap between the theoretical FLOPs and the throughput in practice. The throughput of StyleGAN2 and StyleSwin are 40.05 imgs/sec and 11.05 imgs/sec respectively on a single V100 GPU. This is primarily due to the fact that vision transformers have not been sufficiently optimized as ConvNets (e.g. using CuDNN), and we believe future optimization will democratize the usage of transformers as they exhibit lower theoretical FLOPs. Besides, bCR is not effective on 1024×1024 , which we leave for further study.



Figure 11. Image samples of FFHQ 1024 × 1024.



Figure 12. Image samples of CelebA-HQ 1024 × 1024.



Figure 13. Image samples of LSUN Church 256×256 .



Figure 14. Image samples of LSUN Car 256 × 256.