ABSTRACT

Computer graphics has experienced a recent surge of data-centric approaches for photorealistic and controllable content creation. StyleGAN in particular sets new standards for generative modeling regarding image quality and controllability. However, StyleGAN’s performance severely degrades on large unstructured datasets such as ImageNet. StyleGAN was designed for controllability; hence, prior works suspect its restrictive design to be unsuitable for diverse datasets. In contrast, we find the main limiting factor to be the current training strategy. Following the recently introduced Projected GAN paradigm, we leverage powerful neural network priors and a progressive growing strategy to successfully train the latest StyleGAN3 generator on ImageNet. Our final model, StyleGAN-XL, sets a new state-of-the-art on large-scale image synthesis and is the first to generate images at a resolution of $1024^2$ at such a dataset scale. We demonstrate that this model can invert and edit images beyond the narrow domain of portraits or specific object classes. Code, models, and supplementary videos can be found at https://sites.google.com/view/stylegan-xl/.

KEYWORDS

Generative Adversarial Networks, Pretrained Models, Image Synthesis, Image Editing

1 INTRODUCTION

Computer graphics has long been concerned with generating photorealistic images at high resolution that allow for direct control over semantic attributes. Until recently, the primary paradigm was to create carefully designed 3D models which are then rendered using realistic camera and illumination models. A parallel line of research approaches the problem from a data-centric perspective. In particular, probabilistic generative models [Goodfellow et al. 2014; Song et al. 2021; van den Oord et al. 2017] have shifted the paradigm from designing assets to designing training procedures and datasets. Style-based GANs (StyleGANs) are a specific instance of these models, and they exhibit many desirable properties. They achieve high image fidelity [Karras et al. 2019, 2020b], fine-grained semantic control [Härkönen et al. 2020; Ling et al. 2021; Wu et al. 2021], and recently alias-free generation enabling realistic animation [Karras et al. 2021]. Moreover, they reach impressive photorealism on carefully curated datasets, especially of human faces. However, when trained on large and unstructured datasets like ImageNet [Deng et al. 2009], StyleGANs do not achieve satisfactory results yet. One other problem plaguing data-centric methods, in general, is that
they become prohibitively more expensive when scaling to higher resolutions as bigger models are required. Initially, StyleGAN [Karras et al. 2019] was proposed to explicitly disentangle factors of variations, allowing for better control and interpolation quality. However, its architecture is more restrictive than a standard generator network [Karras et al. 2018; Radford et al. 2016] which seems to come at a price when training on complex and diverse datasets such as ImageNet. Previous attempts at scaling StyleGAN and StyleGAN2 to ImageNet led to sub-par results [Grigoryev et al. 2022; Gwnern 2020], giving reason to believe it might be fundamentally limited for highly diverse datasets [Gwnern 2020].

BigGAN [Brock et al. 2019] is the state-of-the-art GAN model for image synthesis on ImageNet. The main factors for BigGANs success are larger batch and model sizes. However, BigGAN has not reached a similar standing as StyleGAN as its performance varies significantly between training runs [Karras et al. 2020a] and as it does not employ an intermediate latent space which is essential for GAN-based image editing [Abdal et al. 2021; Collins et al. 2020; Patashnik et al. 2021; Wu et al. 2021]. Recently, BigGAN has been superseded in performance by diffusion models [Dhariwal and Nichol 2021]. Diffusion models achieve more diverse image synthesis than GANs but are significantly slower during inference and prior work on GAN-based editing is not directly applicable. Following these arguments, successfully training StyleGAN on ImageNet has several advantages over existing methods.

The previously failed attempts at scaling StyleGAN raise the question of whether architectural constraints fundamentally limit style-based generators or if the missing piece is the right training strategy. Recent work by [Sauer et al. 2021] introduced Projected GANs which project generated and real samples into a fixed, pretrained feature space. Rephrasing the GAN setup this way leads to significant improvements in training stability, training time, and data efficiency. Leveraging the benefits of Projected GAN training might enable scaling StyleGAN to ImageNet. However, as observed by [Sauer et al. 2021], the advantages of Projected GANs only partially extend to StyleGAN on the unimodal datasets they investigated. We study this issue and propose architectural changes to address it. We then design a progressive growing strategy tailored to the latest StyleGAN3. These changes in conjunction with Projected GAN already allow surpassing prior attempts of training StyleGAN on ImageNet. To further improve results, we analyze the pretrained feature network used for Projected GANs and find that the two standard neural architectures for computer vision, CNNs and ViTs [Dosovitskiy et al. 2021], significantly improve performance when used jointly. Lastly, we leverage classifier guidance, a technique originally introduced for diffusion models to inject additional class-information [Dhariwal and Nichol 2021].

Our contributions culminate in a new state-of-the-art on large-scale image synthesis, pushing the performance beyond existing GAN and diffusion models. We showcase inversion and editing for ImageNet classes and find that Pivotal Tuning Inversion (PTI) [Roich et al. 2021], a powerful new inversion paradigm, combines well with our model and even embeds out-of-domain images smoothly into our learned latent space. Our efficient training strategy allows us to triple the parameters of the standard StyleGAN3 while reaching prior state-of-the-art performance of diffusion models [Dhariwal and Nichol 2021] in a fraction of their training time. It further enables us to be the first to demonstrate image synthesis on ImageNet-scale at a resolution of 1024^2 pixels. We will open-source our code and models upon publication.

2 BACKGROUND

We first introduce the main building blocks of our system: the StyleGAN3 generator [Karras et al. 2021] and Projected GAN’s [Sauer et al. 2021] feature projectors and multi-scale discriminators.

StyleGAN. This section describes style-based generators in general with a focus on the latest StyleGAN3 [Karras et al. 2021]. A StyleGAN generator consists of a mapping network \( G_m \) and a synthesis network \( G_s \). First, \( G_m \) maps a normally distributed latent code \( z \) to a style code \( w \). This style code \( w \) is then used for modulating the convolution kernels of \( G_s \) to control the synthesis process. The synthesis network \( G_s \) of StyleGAN3 starts from a spatial map defined by Fourier features [Tancik et al. 2020; Xu et al. 2021]. This input then passes through \( N \) layers of convolutions, non-linearities, and upsampling to generate an image. Each non-linearity is wrapped by an upsampling and downsampling operation to prevent aliasing. The low-pass filters used for these operations are carefully designed to balance image quality, antialiasing, and training speed. Concretely, their cutoff and stopband frequencies grow geometrically with network depth, the transition band half-widths are as wide as possible within the limits of the layer sampling rate, and only the last two layers are critically sampled, i.e., the filter cutoff equals the bandlimit. The number of layers \( N \) is 14, independent of the final output resolution.

Style mixing and path length regularization are methods for regularizing style-based generators. In style mixing, an image is generated by feeding sampled style codes \( w \) into different layers of \( G_s \) independently. Path length regularization encourages that a step of fixed size in latent space results in a corresponding fixed change in pixel intensity of the generated image [Karras et al. 2020b]. This inductive bias leads to a smoother generator mapping and has several advantages including fewer artifacts, more predictable training behavior, and better inversion.

Progressive growing was introduced by [Karras et al. 2018] for stable training at high resolutions but [Karras et al. 2020b] found that it can impair shift-equivariance. [Karras et al. 2021] observe that texture sticking artifacts are caused by a lack of equivariance and carefully design StyleGAN3 to prevent texture sticking. Hence, in this paper, as we build on StyleGAN3, we can revisit the idea of progressive growing to improve convergence speed and synthesis quality.

Projected GAN. The original adversarial game between a generator \( G \) and a discriminator \( D \) can be extended by a set of feature projectors \( \{P_l\} \) [Sauer et al. 2021]. The projectors map real images \( x \) and images generated by \( G \) to the discriminator’s input space. The Projected GAN objective is formulated as

\[
\min_G \max_{\{P_l\}} \sum_{l \in L} \left( \mathbb{E}_x [\log D_l(P_l(x))] + \mathbb{E}_z [\log(1 - D_l(P_l(G(z))))] \right)
\]
where \( \{D_i\} \) is a set of independent discriminators operating on different feature projections. The projectors consist of a pretrained feature network \( F \), cross-channel mixing (CCM) and cross-scale mixing (CSM) layers. The purpose of CCM and CSM is to prohibit the discriminators from focusing on only a subset of its input feature space which would result in mode collapse. Both modules employ differentiable random projections that are not optimized during GAN training. CCM mixes features across channels via random 1x1 convolutions, CSM mixes features across scales via residual random 3x3 convolution blocks and bilinear upsampling. The output of CSM is a feature pyramid consisting of four feature maps at different resolutions. Four discriminators operate independently on these feature maps. Each discriminator uses a simple convolutional architecture and spectral normalization [Miyato et al. 2018]. The depth of the discriminator varies depending on its input resolution, i.e., a spatially larger feature map corresponds to a deeper discriminator. Other than spectral normalization, Projected GANs do not use additional regularization such as gradient penalties [Mescheder et al. 2018]. Lastly, [Sauer et al. 2021] apply differentiable data-augmentation [Zhao et al. 2020] before \( F \) which improves Projected GAN’s performance independent of the dataset size.

[Sauer et al. 2021] evaluate several combinations of \( F \) and \( G \) and find an EfficientNet-Lite0 [Tan and Le 2019] and a FastGAN generator [Liu et al. 2021] to work especially well. When using a StyleGAN generator, they observe that the discriminators can quickly overpower the generator for suboptimal learning rates. The authors suspect that the generator might adapt too slowly due to its design which modulates feature maps with styles learned by a mapping network.

3 SCALING STYLEGAN TO IMAGENET

As mentioned before, StyleGAN has several advantages over existing approaches that work well on ImageNet. But a naive training strategy does not yield state-of-the-art performance [Grigoryev et al. 2022; Gwern 2020]. Our experiments confirm that even the latest StyleGAN3 does not scale well, see Fig. 1. Particularly at high resolutions, the training becomes unstable. Therefore, our goal is to train a StyleGAN3 generator on ImageNet successfully. Success is defined in terms of sample quality primarily measured by inception score (IS) [Salimans et al. 2016] and diversity measured by Fréchet Inception Distance (FID) [Heusel et al. 2017]. Throughout this section, we gradually introduce changes to the StyleGAN3 baseline (\( \text{Config-A} \)) and track the improvements in Table 1. First, we modify the generator and its regularization losses, adapting the latent space to work well with Projected GAN (\( \text{Config-B} \)) and for the class-conditional setting (\( \text{Config-C} \)). We then revisit progressive growing to improve training speed and performance (\( \text{Config-D} \)). Next, we investigate the feature networks used for Projected GAN training to find a well-suited configuration (\( \text{Config-E} \)). Lastly, we propose classifier guidance for GANs to provide class information via a pretrained classifier (\( \text{Config-F} \)). Our contributions enable us to train a significantly larger model than previously possible while requiring less computation than prior art. Our model is three times larger in terms of depth and parameter count than a standard StyleGAN3. However, to match the prior state-of-the-art performance of ADM [Dhariwal and Nichol 2021] at a resolution of 512^2 pixels, training the models on a single NVIDIA Tesla V100 takes 400 days compared to the previously required 1914 V100-days. We refer to our model as \textbf{StyleGAN-XL} (Fig. 2).

3.1 Adapting Regularization and Architectures

Training on a diverse and class-conditional dataset makes it necessary to introduce several adjustments to the standard StyleGAN configuration. We construct our generator architecture using layers of StyleGAN3-T, the translational-equivariant configuration of StyleGAN3. In initial experiments, we found the rotational-equivariant StyleGAN3-R to generate overly symmetric images on more complex datasets, resulting in kaleidoscope-like patterns.

Regularization. In GAN training, it is common to use regularization for both, the generator and the discriminator. Regularization improves results on uni-modal datasets like FFHQ [Karras et al. 2019] or LSUN [Yu et al. 2015], whereas it can be detrimental on multi-modal datasets [Brock et al. 2019; Gwern 2020]. Therefore, we aim to avoid regularization when possible. [Karras et al. 2021] find style mixing to be unnecessary for the latest StyleGAN3; hence, we also disable it. Path length regularization can lead to poor results on complex datasets [Gwern 2020] and is, per default, disabled for StyleGAN3 [Karras et al. 2021]. However, path length regularization is attractive as it enables high-quality inversion [Karras et al. 2020b]. We also observe unstable behavior and divergence when using path length regularization in practice. We found that this problem can be circumvented by only applying regularization after the model has been sufficiently trained, i.e., after 200k images. For the discriminator, following [Sauer et al. 2021], we use spectral normalization without gradient penalties. In addition, we blur all images with a Gaussian filter with \( \sigma = 2 \) pixels for the first 200k images. Discriminator blurring has been introduced in [Karras et al. 2021] for StyleGAN3-R. It prevents the discriminator from focusing on high frequencies early on, which we found beneficial across all settings we investigated.

Low-Dimensional Latent Space. As observed in [Sauer et al. 2021], Projected GANs work better with FastGAN [Liu et al. 2021] than with StyleGAN. One main difference between these generators is their latent space. StyleGAN’s latent space is comparatively high dimensional (FastGAN: \( \mathbb{R}^{190} \), BigGAN: \( \mathbb{R}^{128} \), StyleGAN: \( \mathbb{R}^{512} \)). Recent findings indicate that the intrinsic dimension of natural image datasets is relatively low [Pope et al. 2021], ImageNet’s dimension estimate is around 40. Accordingly, a latent code of size 512 is highly redundant, making the mapping network’s task harder at the beginning of training. Consequently, the generator is slow to adapt and cannot benefit from Projected GAN’s speed up. We therefore reduce StyleGAN’s latent code to 64 and now observe stable training in combination with Projected GAN, resulting in lower FID than the baseline (\( \text{Config-B} \)). We keep the original dimension of the style code \( w \in \mathbb{R}^{512} \) to not restrict the model capacity of the mapping network \( G_m \).

Pretrained Class Embeddings. Conditioning the model on class information is essential to control the sample class and improve overall performance. A class-conditional variant of StyleGAN was first proposed in [Karras et al. 2020a] for CIFAR10 [Krizhevsky
Figure 2: Training StyleGAN-XL. We feed a latent code \( z \) and class label \( c \) to the pretrained embedding and the mapping network \( G_m \) to generate style codes \( w \). The codes modulate the convolutions of the synthesis network \( G_s \). During training, we gradually add layers to double the output resolution for each stage of the progressive growing schedule. We only train the latest layers while keeping the others fixed. \( G_m \) is only trained for the initial 16\(^2\) stage and remains fixed for the higher-resolution stages. The synthesized image is upsampled when smaller than 224\(^2\) and passed through a CNN and a ViT and respective feature mixing blocks (CCM+CSM). At higher resolutions, the CNN receives the unaltered image while the ViT receives a downsampled input to keep memory requirements low but still utilize its global feedback. Finally, we apply eight independent discriminators on the resulting multi-scale feature maps. The image is also fed to classifier CLF for classifier guidance.

Table 1: Ablation Study on ImageNet 128\(^2\). Left: Results for different configurations after training for 15 V100-days. Right: Comparing combinations of different feature networks \( F \). Beginning from the base configuration using an EfficientNet-lite0 (EffNet), we add a second \( F \) with varying architecture type and pretraining objective (Class: Classification, Self: MoCo-v2 [Chen et al. 2020]).

| Configuration | FID ↓ | IS ↑ |
|---------------|-------|------|
| A StyleGAN3   | 53.57 | 15.30 |
| B + Projected GAN & small \( z \) | 22.98 | 57.62 |
| C + Pretrained embeddings | 20.91 | 35.79 |
| D + Progressive growing | 19.51 | 35.74 |
| E + ViT & CNN as \( F_{1,2} \) | 12.43 | 56.72 |
| F + CLF guidance (StyleGAN-XL) | **12.24** | **86.21** |

| Model          | Type   | Objective | FID ↓ | IS ↑ |
|----------------|--------|-----------|-------|------|
| EffNet CNN     | Class  |           | 19.51 | 35.74 |
| EffNet ResNet50| CNN    | Class     | 16.16 | 49.13 |
| EffNet ResNet50| CNN    | Class     | 18.53 | 38.26 |
| EffNet DeiT-M  | CNN    | ViT       | **12.43** | **56.72** |

et al. 2009] where a one-hot encoded label is embedded into a 512-dimensional vector and concatenated with \( z \). For the discriminator, class information is projected onto the last discriminator layer [Miyato and Koyama 2018]. We observe that Config-B tends to generate similar samples per class resulting in high IS. To quantify mode coverage, we leverage the recall metric [Kynkäänniemi et al. 2019] and find that Config-B achieves a low recall of 0.004. We hypothesize that the class embeddings collapse when training with Projected GAN. Therefore, to prevent this collapse, we aim to ease optimization of the embeddings via pretraining. We extract and spatially pool the lowest resolution features of an Efficientnet-lite0 [Tan and Le 2019] and calculate the mean per ImageNet class. The network has a low channel count to keep the embedding dimension small, following the arguments of the previous section. The embedding passes through a linear projection to match the size of \( z \) to avoid an imbalance. Both \( G_m \) and \( D_0 \) are conditioned on the embedding. During GAN training, the embedding and the linear projection are optimized to allow specialization. Using this configuration, we observe that the model generates diverse samples per class, and recall increases to 0.15 (Config-C). Note that for all configurations in this ablation, we restrict the training time to 15 V-100 days. Hence, the absolute recall is markedly lower compared to the fully trained models. Conditioning a GAN on pretrained features was also recently investigated by [Casanova et al. 2021]. In contrast to our approach, [Casanova et al. 2021] condition on specific instances, instead of learning a general class embedding.

3.2 Reintroducing Progressive Growing

Progressively growing the output resolution of a GAN was introduced by [Karras et al. 2018] for fast and more stable training. The original formulation adds layers during training to both \( G \) and \( D \) and gradually fades in their contribution. However, in a later work, it was discarded [Karras et al. 2020b] as it can contribute to texture sticking artifacts. Recent work by [Karras et al. 2021] finds that the primary cause of these artifacts is aliasing, so they redesign each layer of StyleGAN to prevent it. This motivates us to reconsider progressive growing with a carefully crafted strategy that aims to suppress aliasing as best as possible. Training first on very low resolutions, as small as 16\(^2\) pixels, enables us to break down the daunting task of training on high-resolution ImageNet into smaller subtasks. This idea is in line with the latest work on diffusion models [Dhariwal and Nichol 2021; Ho et al. 2022; Nichol and Dhariwal 2021; Saharia et al. 2021]. They observe considerable improvements in FID on ImageNet when using a two-stage model, i.e., stacking an independent low-resolution model and an upsampling model to generate the final image.

Commonly, GANs follow a rigid sampling rate progression, i.e., at each resolution, there is a fixed amount of layers followed by
an upsampling operation using fixed filter parameters. StyleGAN3 does not follow such a progression. Instead, the layer count is set to 14, independent of the output resolution, and the filter parameters of up- and downsampling operations are carefully designed for antialiasing under the given configuration. The last two layers are critically sampled to generate high-frequency details. When adding layers for the subsequent highest resolution, discarding the previously critically sampled layers is crucial as they would introduce aliasing when used as intermediate layers [Karras et al. 2021, 2020b]. Furthermore, we adjust the filter parameters of the added layers to adhere to the flexible layer specification of [Karras et al. 2021]; we refer to the supplementary for details. In contrast to [Karras et al. 2018] we do not add layers to the discriminator. Instead, to fully utilize the pretrained feature network $F$, we upsample both data and synthesized images to $F$’s training resolution ($224^2$ pixels) when training on smaller images.

We start progressive growing at a resolution of $16^2$ using 11 layers. Every time the resolution increases, we cut off 2 layers and add 7 new ones. Empirically, fewer layers result in worse performance; adding more leads to increased overhead and diminishing returns. For the final stage at $1024^2$, we add only 5 layers as the last two are not discarded. This amounts to 39 layers at the maximum resolution of $1024^2$. Instead of a fixed growing schedule, each stage is trained until FID stops decreasing. We find it beneficial to use a large batch size of 2048 on lower resolution ($16^2$ and $32^2$), similar to [Brock et al. 2019]. On higher resolutions, smaller batch sizes suffice ($64^2$ to $256^2$, $256$, $512^2$ to $1024^2$: 128). Once new layers are added, the lower resolution layers remain fixed to prevent mode collapse.

In our ablation study, FID improves only slightly (Config-D) compared to Config-C. However, the main advantage can be seen at high resolutions, where progressive growing drastically reduces training time. At resolution $512^2$, we reach the prior state-of-the-art (FID = 3.85) after 2 V100-days. This reduction is in contrast to other methods such as ADM, where doubling the resolution from $256^2$ to $512^2$ pixels corresponds to increasing training time from 393 to 1914 V100-days to find the best performing model$^1$. As our aim is not to introduce texture sticking artifacts, we measure $EQ-T$, a metric for determining translation equivariance [Karras et al. 2021], where higher is better. Config-C yields $EQ-T = 55$, while Config-D attains $EQ-T = 48$. This only slight reduction in equivariance shows that Config-D restricts aliasing almost as well as a configuration without growing. For context, architectures with aliasing yield $EQ-T \sim 15$.

### 3.3 Exploiting Multiple Feature Networks

An ablation study conducted in [Sauer et al. 2021] finds that most pretrained feature networks $F$ perform similarly in terms of FID when used for Projected GAN training regardless of training data, pretraining objective, or network architecture. However, the study does not answer if combining several $F$ is advantageous. Starting from the standard configuration, an EfficientNet-lite0, we add a second $F$ to inspect the influence of its pretraining objective (classification or self-supervision) and architecture (CNN or Vision Transformer (ViT) [Dosovitskiy et al. 2021]). The results in Table 1 show that an additional CNN leads to slightly lower FID. Combining networks with different pretraining objectives does not offer benefits over using two classifier networks. However, combining an EfficientNet with a ViT improves performance significantly. This result corroborates recent results in neural architecture literature, which find that supervised and self-supervised representations are similar [Grigg et al. 2021], whereas ViTs and CNNs learn different representations [Raghu et al. 2021]. Combining both architectures appears to have complementary effects for Projected GANs. We do not see significant improvements when adding more networks; hence, Config-E uses the combination of EfficientNet [Tan and Le 2019] and DeiT-base [Touvron et al. 2021].

### 3.4 Classifier Guidance for GANs

[Dhariwal and Nichol 2021] introduced classifier guidance to inject class information into diffusion models. Classifier guidance modifies each diffusion step at time step $t$ by adding gradients of a pretrained classifier $\nabla_x \log p_g(z, t)$ as an additional term to the generator loss and scale this term by a constant $\lambda$. For the classifier, we use DeiT-small [Touvron et al. 2021], which exhibits strong classification performance while not adding much overhead to the training. Similar to [Dhariwal and Nichol 2021], we observe a significant improvement in IS, indicating an increase in sample quality (Config-F). We find $\lambda = 8$ to work well empirically. Classifier guidance only works well on higher resolutions ($> 32^2$); otherwise, it leads to mode collapse. This is in contrast to [Dhariwal and Nichol 2021] who exclusively guide their low-resolution model. The difference stems from how guidance is applied: we use it for model training, whereas [Dhariwal and Nichol 2021] guide the sampling process.

### 4 RESULTS

In this section, we first compare StyleGAN-XL to the state-of-the-art approaches for image synthesis on ImageNet. We then evaluate the inversion and editing capabilities of StyleGAN-XL. As described above, we scale our model to a resolution of $1024^2$ pixels, which no prior work has attempted so far on ImageNet. The resolution of most images in ImageNet is lower. We therefore preprocess the data with a super-resolution network [Liang et al. 2021], see supplementary.

#### 4.1 Image Synthesis

Both our work and [Dhariwal and Nichol 2021] use classifier networks to guide the generator. To ensure the models are not inadvertently optimizing for FID and IS, which also utilize a classifier network, we propose random-FID (rFID). For rFID, we calculate the Fréchet distance in the $\text{pool}_3$ layer of a randomly initialized inception network [Szegedy et al. 2015]. The efficacy of random features for evaluating generative models has been demonstrated.

---

$^1$Note that these settings are not directly comparable as the stem of our model is pretrained, but the values should give a general sense of the order of magnitude.
we do not compare to baselines because of resource constraints as XL already achieves satisfactory inversion results using basic latent [Tov et al. 2021]. A common way to achieve low reconstruction and StyleGAN-XL attains FID images. We measure the FID between reconstructions and targets, BigGAN at PSNR optimization. For inversion on the ImageNet validation set at 512 are close to the original distribution of $W$. For example, that $W$’s sample diversity lies between BigGAN and ADM, making progress in closing the gap between these model types. BigGAN’s sample quality is the best among all compared approaches, which comes at the price of significantly lower recall. StyleGAN-XL allows for the truncation trick to improve sample fidelity, i.e., we can interpolate a sampled style code $w$ with the class-wise mean style code $\bar{w}$. We observe that for StyleGAN-XL, truncation does not increase precision, indicating that developing novel truncation methods for high-diversity GANs is an exciting research direction for future work. Interestingly, StyleGAN-XL attains high diversity across all resolutions, which can be attributed to our progressive growing strategy. Furthermore, this strategy enables us to scale to megapixel resolution successfully. Training at $1024^2$ for a single V100-day yields a noteworthy FID of 2.8. At this resolution, we do not compare to baselines because of resource constraints as they are prohibitively expensive to train. visualizes generated samples at increasing resolutions. Fig. 3 visualizes generated samples at increasing resolutions. In the supplementary, we show additional interpolations and qualitative comparisons to BigGAN and ADM.

4.2 Inversion and Manipulation

GAN-editing methods first invert a given image into latent space, i.e., find a style code $w$ that reconstructs the image as faithful as possible when passed through $G_w$. Then, $w$ can be manipulated to achieve semantically meaningful edits [Goetschalckx et al. 2019; Shen et al. 2020].

Inversion. Standard approaches for inverting $G_w$ use either latent optimization [Abdal et al. 2019; Creswell and Bharath 2019; Karras et al. 2020b] or an encoder [Alaluf et al. 2021; Perarnau et al. 2016; Tov et al. 2021]. A common way to achieve low reconstruction error is to use an extended definition of the latent space: $W^+$. For $W^+$ a separate $w$ is chosen for each layer of $G_w$. However, as highlighted by [Tov et al. 2021; Zhu et al. 2020], this extended definition achieves higher reconstruction quality in exchange for lower editability. Therefore, [Tov et al. 2021] carefully design an encoder to maintain editability by mapping to regions of $W^+$ that are close to the original distribution of $W$. We follow [Karras et al. 2020b] and use the original latent space $W$. We find that StyleGAN-XL already achieves satisfactory inversion results using basic latent optimization. For inversion on the ImageNet validation set at $512^2$, StyleGAN-XL yields PSNR = 13.5 on average, improving over BigGAN at PSNR = 10.8. Besides better pixel-wise reconstruction, StyleGAN-XL’s inversions are semantically closer to the target images. We measure the FID between reconstructions and targets, and StyleGAN-XL attains FID = 21.7 while BigGAN reaches FID = 47.5. For qualitative results, implementation details and additional metrics, we refer to the supplementary.

Given the results above, it is also possible to further refine the obtained reconstructions. [Roich et al. 2021] recently introduced pivotal tuning inversion (PTI). PTI uses an initial inverted style code as a pivot point around which the generator is finetuned. Additional regularization prevents altering the generator output far from the pivot. Combining PTI with StyleGAN-XL allows us to invert both in-domain (ImageNet validation set) and out-of-domain images almost precisely. At the same time, the generator output remains perceptually smooth, see Fig. 4.

Image Manipulation. Given the inverted images, we can leverage GAN-based editing methods [Härkönen et al. 2020; Kocasari et al. 2022; Shen and Zhou 2021; Spingarn et al. 2021; Vovynov and Babenko 2020] to manipulate the style code $w$. In Fig. 5 (Left), we first invert a given source image via latent space optimization. We can then apply a manipulation directions obtained by, e.g., GANspace [Härkönen et al. 2020]. Prior work [Jahanian et al. 2020] also investigates in-plane translation. This operation can be directly defined in the input grid of StyleGAN-XL. The input grid also allows performing extrapolation, see Fig. 5 (Left).

An inherent property of StyleGAN is the ability of style mixing by supplying the style codes of two samples to different layers of $G_s$, generating a hybrid image. This hybrid takes on different semantic properties of both inputs. Style mixing is commonly employed for instances of a single domain, i.e., combining two human portraits. StyleGAN-XL inherits this ability and, to a certain extent, even generates out-of-domain combinations between different classes, akin to counterfactual images [Sauer and Geiger 2021]. This technique works best for aligned samples, similar to StyleGAN’s originally favored setting, FFHQ. Curated examples are shown in Fig. 5 (Right).

5 LIMITATIONS AND FUTURE WORK

Our contributions allow StyleGAN to accomplish state-of-the-art high-resolution image synthesis on ImageNet. Furthermore, applying it to big and small unimodal datasets is straightforward, and we also achieve state-of-the-art performance on FFHQ and Pokemon at resolution $1024^2$, see supplementary. Exploring new editing methods and dataset generation [Chai et al. 2021; Li et al. 2022] using StyleGAN-XL are exciting future avenues. Furthermore, future work may tackle an even larger megapixel dataset. However, a larger yet diverse dataset is not available so far. Current large-scale, high-resolution datasets are of single object classes or contain many similar images [Fregin et al. 2018; Perot et al. 2020; Zhang et al. 2020]. In the supplementary, we discuss limitations of the current model, which should be addressed in the future.
Table 2: Image Synthesis on ImageNet. Empty cells indicate that the model was not available and the respective metric not evaluated in the original work.

| Model       | FID ↓ | sFID ↓ | rFID ↓ | IS ↑ | Pr ↑ | Rec ↑ | Model       | FID ↓ | sFID ↓ | rFID ↓ | IS ↑ | Pr ↑ | Rec ↑ |
|-------------|-------|--------|--------|------|------|-------|-------------|-------|--------|--------|------|------|-------|
| Resolution 128² |       |        |        |      |      |       | Resolution 256² |       |        |        |      |      |       |
| BigGAN      | 6.02  | 7.18   | 6.09   | 145.83 | 0.86 | 0.35  | BigGAN      | 49.20 |        |        |      |      |       |
| CDM         | 3.52  | 128.80 | 128.80 |      |      |       | CDM         | 4.88  | 158.70 | 158.70 |      |      |       |
| ADM-G       | 5.91  | 5.09   | 13.29  | 93.31 | 0.70 | 0.65  | ADM         | 10.94 | 6.02   | 125.78 | 100.98 | 0.69 | 0.63  |
| StyleGAN-XL | 1.81  | 3.82   | 1.82   | 200.55 | 0.77 | 0.55  | ADM-G-U     | 3.94  | 6.14   | 11.86  | 215.84 | 0.83 | 0.53  |
| Resolution 512² |       |        |        |      |      |       | Resolution 1024² |       |        |        |      |      |       |
| BigGAN      | 8.43  | 8.13   | 312.00 | 177.90 | 0.88 | 0.29  | StyleGAN-XL | 2.52  | 4.12   | 413.12 | 260.14 | 0.76 | 0.51  |
| ADM         | 23.24 | 10.19  | 561.32 | 58.06  | 0.73 | 0.60  | ADM         | 10.94 | 6.02   | 125.78 | 100.98 | 0.69 | 0.63  |
| ADM-G-U     | 3.85  | 5.86   | 210.83 | 221.72 | 0.84 | 0.53  | ADM-G-U     | 3.94  | 6.14   | 11.86  | 215.84 | 0.83 | 0.53  |
| StyleGAN-XL | 2.41  | 4.06   | 51.54  | 267.75 | 0.77 | 0.52  | ADM-G-U     | 3.94  | 6.14   | 11.86  | 215.84 | 0.83 | 0.53  |

Figure 3: Samples at Different Resolutions Using the Same w. The samples are generated by the models obtained during progressive growing. We upsample all images to 1024² using nearest-neighbor interpolation for visualization purposes. Zooming in is recommended.

Figure 4: Interpolations. StyleGAN-XL generates smooth interpolations between samples of different classes (Row 1 & Row 2). PTI allows inverting to the latent space with low distortion (outermost image, Row 3 & Row 4), and consistently embeds out-of-domain inputs, such as the one on the bottom right.
Figure 5: Image Editing and Style Mixing. Left: First, a given image is inverted via PTI [Roich et al. 2021]. Right: Given two images, we can mix their styles. This method works for samples of the same or similar classes, and to a certain extent, for distant classes. For this experiment, we utilize random samples instead of inversions.

Figure 6: Image Manipulation via Language. Given a random sample, we manipulate the image by following semantic directions in latent space found by StyleMC [Kocasari et al. 2022]. The latent space directions from top to bottom are: “smile”, “no stripes”, and “big eyes”.
ACKNOWLEDGMENTS

We acknowledge the financial support by the BMWi in the project "KI Delta Learning" (project number 19A19030). Andreas Geiger was supported by the ERC Starting Grant LEGO-3D (850533). We would like to thank Kashyap Chitta, Michael Niemeyer, and Božidar Antić for proofreading. Lastly, we would like to thank Vanessa Sauer for her general support.

REFERENCES

Rameen Abdal, Yipeng Qin, and Peter Wonka. 2019. ImageStyleGAN: How to Embed Images Into the StyleGAN Latent Space. In Proc. of the IEEE International Conf. on Computer Vision (ICCV). 4431–4440. https://doi.org/10.1109/ICCV.2019.00453

Rameen Abdal, Peihao Zhu, Niloj J. Mitra, and Peter Wonka. 2021. StyleFlow: Attribute-conditioned Exports of StyleGAN-Generated Images using Conditional Continuous Normalizing Flows. ACM Trans. on Graphics 40, 3 (2021), 21:1–21:21. https://doi.org/10.1145/3447684

Yuval Alaluf, Or Patashnik, and Daniel Cohen-Or. 2021. ReStyle: A Residual-Based StyleGAN Encoder via Iterative Refinement. Proc. of the IEEE International Conf. on Computer Vision (ICCV) aisb 2021.02969 (2021). https://arxiv.org/abs/2104.02969

Andrew Brock, Jeff Donahue, and Karen Simonyan. 2019. Large Scale GAN Training for High Fidelity Natural Image Synthesis. Proc. of the Conference on Learning Representations (ICLR). OpenReview.net. https://openreview.net/forum?id=B1seq0gFFm

Arantxa Casanova, Marlène Careil, Jakob Verbeek, and Michal Drozdzal. 2021. Improving baselines with Discrete Normalizing Flows. ACM Trans. on Graphics 40, 3 (2021), 21:1–21:21. https://doi.org/10.1145/3447684

Xinlei Chen, Haoqi Fan, and Ross Girshick. 2020. Inverting the Generator of a Generative Adversarial Network. In Proc. of the International Conf. on Learning Representations (ICLR). OpenReview.net. https://openreview.net/forum?id=ryQu7fY_B

Sanja Fidler. 2021. EditGAN: High-Precision Semantic Image Editing. arXiv.org. https://arxiv.org/abs/2112.08493

Alex Krizhevsky, Geoffrey Hinton, et al. 2009. Learning multiple layers of features from tiny images. (2009).

Timofey Grigorov, Andrey Vovnyov, and Artem Babenko. 2022. When, Why, and Which Pretrained GANs Are Useful? In Proc. of the International Conf. on Learning Representations (ICLR). OpenReview.net. https://openreview.net/forum?id=Hb3SHo5c1Bu

Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. 2020. Progressive Growing of GANs for Improved Quality, Stability, and Variation. In Proc. of the International Conf. on Learning Representations (ICLR). OpenReview.net. https://openreview.net/forum?id=HkB9z7zCeb

Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. 2020a. Training Generative Adversarial Networks with Limited Data. Advances in Neural Information Processing Systems (NeurIPS). https://proceedings.neurips.cc/paper/2020/hash/8300a9e67c7440759d7b3d03c61af3d-Abstract.html

Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. 2021. Alias-Free Generative Adversarial Networks. In Advances in Neural Information Processing Systems (NeurIPS).

Tero Karras, Samuli Laine, and Timo Aila. 2019. A Style-Based Generator Architecture for Generative Adversarial Networks. Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). Computer Vision Foundation / IEEE, 4401–4410. https://doi.org/10.1109/CVPR.2019.00453

Tim Salimans. 2022. Cascaded Diffusion Models for High Fidelity Image Generation. arXiv.org. https://arxiv.org/abs/2110.00528

Subject

Siegfried ‘22 Conference Proceedings, August 7–11, 2022, Vancouver, BC, Canada

Timofey Grigorov, Andrey Vovnyov, and Artem Babenko. 2022. When, Why, and Which Pretrained GANs Are Useful? In Proc. of the International Conf. on Learning Representations (ICLR). OpenReview.net. https://openreview.net/forum?id=Hb3SHo5c1Bu

Gwern. 2020. Making Anime Faces with StyleGAN. Retrieved January 16, 2022 from https://www.gwern.net/Faces/stylegan2-ext-modifications/

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. 2017. GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. Advances in Neural Information Processing Systems (NeurIPS). 6626–6637. https://proceedings.neurips.cc/paper/2017/hash/3a1e494707c6b6f60e6f587e16097492d6-Abstract.html

Jonathan Ho, Chiiwan Saharia, William Chan, David J. Fleet, Mohammad Norouzi, and Tim Salimans. 2022. Cascaded Diffusion Models for High Fidelity Image Generation. J. Mach. Learn. Res. 23 (2022), 47–1–47–33.

Ali Jahanian, Lucy Chai, and Phillip Isola. 2020. On the ‘steerability’ of generative adversarial networks. In Proc. of the International Conf. on Learning Representations (ICLR). OpenReview.net. https://openreview.net/forum?id=Hyl18TDxVb

Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. 2018. Progressive Growing of GANs for Improved Quality, Stability, and Variation. In Proc. of the International Conf. on Learning Representations (ICLR). OpenReview.net. https://openreview.net/forum?id=HkB9z7zCeb

Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. 2020a. Training Generative Adversarial Networks with Limited Data. Advances in Neural Information Processing Systems (NeurIPS). https://proceedings.neurips.cc/paper/2020/hash/8300a9e67c7440759d7b3d03c61af3d-Abstract.html

Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. 2021. Alias-Free Generative Adversarial Networks. In Advances in Neural Information Processing Systems (NeurIPS).

Tero Karras, Samuli Laine, and Timo Aila. 2019. A Style-Based Generator Architecture for Generative Adversarial Networks. Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). Computer Vision Foundation / IEEE, 4401–4410. https://doi.org/10.1109/CVPR.2019.00453

Umut Kocakarsi, Alara Dirik, Mert Tufikci, and Pınar Yanardag. 2022. StyleMC: Multi-Channel Based Fast Text-Guided Image Generation and Manipulation. Proc. of the IEEE Winter Conference on Applications of Computer Vision (WACV) (2022). https://arxiv.org/abs/2112.04849

Alex Krizhevsky, Geoffrey Hinton, et al. 2009. Learning multiple layers of features from tiny images. (2009).

Tomas Kynkännen, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. 2019. Improved Precision and Recall Metric for Assessing Generative Models. In Advances in Neural Information Processing Systems (NeurIPS). https://proceedings.neurips.cc/paper/2019/hash/02345e10bcb9b0d82bc7787f0311734797-Abstract.html

Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. 2021. Making Anime Faces with StyleGAN Gwern. 2020. Making Anime Faces with StyleGAN. Retrieved January 16, 2022 from https://www.gwern.net/Faces/stylegan2-ext-modifications/

Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. 2020. GANSpace: Discovering Interpretable GAN Controls. In Advances in Neural Information Processing Systems (NeurIPS). https://proceedings.neurips.cc/paper/2020/hash/8fde4326967adab6c4e5c6852855cc-5e-Abstract.html

Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. 2018. Spectral Normalization for Generative Adversarial Networks. In Advances in Neural Information Processing Systems (NeurIPS). 3777–3785. https://proceedings.neurips.cc/paper/2018/hash/6af9b275975fa9bbf212f959f05-Abstract.html

Bingchen Liu, Yizhe Zhu, Kunpeng Song, and Ahmed Elgammal. 2021. Towards Consistent Annotations. Proc. of the International Conf. on Learning Representations (ICLR). OpenReview.net. https://openreview.net/forum?id=Hyl18TDxVb

Bingchen Liu, Yizhe Zhu, Kunpeng Song, and Ahmed Elgammal. 2021. Towards Consistent Annotations. Proc. of the International Conf. on Learning Representations (ICLR). OpenReview.net. https://openreview.net/forum?id=Hyl18TDxVb

Takeru Miyato, Toshihiko Kataoka, Masanori Koyama, and Yuichi Yoshida. 2018. Spectral Normalization for Generative Adversarial Networks. In Proc. of the International Conf. on Learning Representations (ICLR). OpenReview.net. https://openreview.net/forum?id=Hyl18TDxVb

Takeru Miyato and Masanori Koyama. 2018. cGANs with Projection Discriminator. In Proc. of the International Conf. on Learning Representations (ICLR). OpenReview.net. https://openreview.net/forum?id=Hyl18TDxVb

Muhammad Ferjad Narem, Seong Joon Oh, Youngjun Uh, Yunjee Choi, and Jaeyoon Joo. 2020. Reliable Fidelity and Diversity Metrics for Generative Models. In Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13–18 July 2020, Virtual Event (Proceedings of Machine Learning Research, Vol. 119). 7176–7185.
