

StyleGAN-XL: Scaling StyleGAN to Large Diverse Datasets

AXEL SAUER, KATJA SCHWARZ, and ANDREAS GEIGER

University of Tübingen and Max Planck Institute for Intelligent Systems, Tübingen, Germany

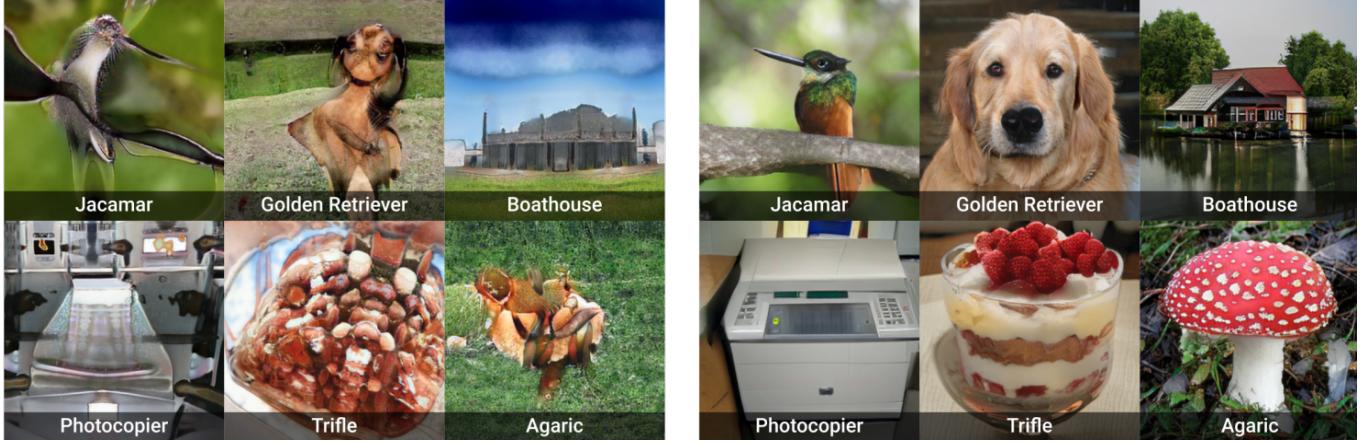


Fig. 1. Class-conditional samples generated by StyleGAN3 (left) and StyleGAN-XL (right) trained on ImageNet at resolution 256^2 .

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/>.

CCS Concepts: • Computing methodologies → Learning latent representations; Image manipulation; Computer graphics; Neural networks.

Additional Key Words and Phrases: Generative Adversarial Networks, Pre-trained Models, Image Synthesis, Image Editing

ACM Reference Format:

Axel Sauer, Katja Schwarz, and Andreas Geiger. 2022. StyleGAN-XL: Scaling StyleGAN to Large Diverse Datasets. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Proceedings (SIGGRAPH ’22 Conference Proceedings)*, August 7–11, 2022, Vancouver, BC, Canada. ACM, New York, NY, USA, 19 pages. <https://doi.org/10.1145/3528233.3530738>

Publication rights licensed to ACM. ACM acknowledges that this contribution was authored or co-authored by an employee, contractor or affiliate of a national government. As such, the Government retains a nonexclusive, royalty-free right to publish or reproduce this article, or to allow others to do so, for Government purposes only.
SIGGRAPH ’22 Conference Proceedings, August 7–11, 2022, Vancouver, BC, Canada
© 2022 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-9337-9/22/08...\$15.00
<https://doi.org/10.1145/3528233.3530738>

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; Gwern 2020], giving reason to believe it might be fundamentally limited for highly diverse datasets [Gwern 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

$$\min_G \max_{\{D_l\}} \sum_{l \in \mathcal{L}} \left(\mathbb{E}_x [\log D_l(P_l(x))] + \mathbb{E}_z [\log(1 - D_l(P_l(G(z))))] \right) \quad (1)$$

where $\{D_l\}$ 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 1×1 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 naïve 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 (**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 (**Config-B**) and for the class-conditional setting (**Config-C**). We then revisit progressive growing to improve training speed and performance (**Config-D**). Next, we investigate the feature networks used for Projected GAN training to find a well-suited configuration (**Config-E**). Lastly, we propose classifier guidance for GANs to provide class information via a pretrained classifier (**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 **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}^{100} , 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 \mathbf{z} to 64 and now observe stable training in combination with Projected GAN, resulting in lower FID than the baseline (**Config-B**). We keep the original dimension of the *style code* $\mathbf{w} \in \mathbb{R}^{512}$ 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 et al. 2009] where a one-hot encoded label is embedded into a 512-dimensional vector and concatenated with \mathbf{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 [Kynkänniemi 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

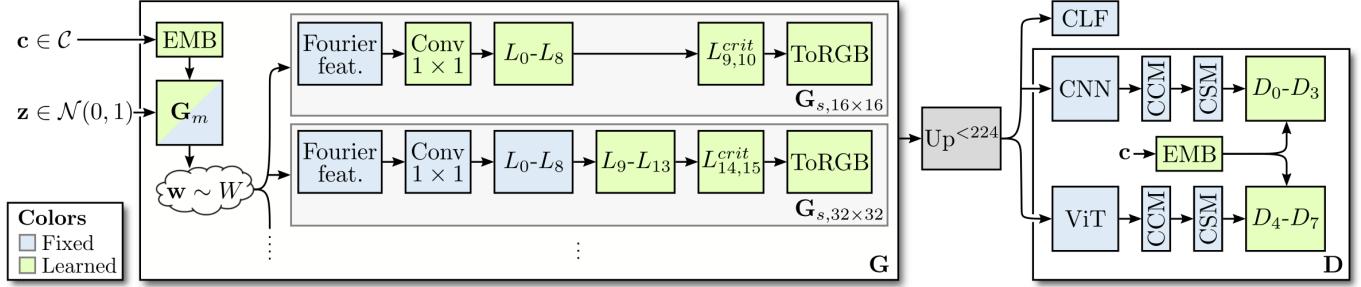


Fig. 2. **Training StyleGAN-XL.** We feed a latent code z and class label c to the pretrained embedding and the mapping network G_m to generate style codes w . The codes modulate the convolutions of the synthesis network G_s . During training, we gradually add layers to double the output resolution for 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 128².** 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 ↑
	F₁	F₂	F₁	F₂		
EffNet		CNN		Class	19.51	35.74
EffNet	ResNet50	CNN	CNN	Class	16.16	49.13
EffNet	ResNet50	CNN	CNN	Self	18.53	38.26
EffNet	DeiT-M	CNN	ViT	Class	12.43	56.72

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_i 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 model¹. As our aim is not to introduce texture sticking artifacts, we measure $EQ\text{-}T$, a metric for determining translation equivariance [Karras et al. 2021], where higher is better. **Config-C** yields $EQ\text{-}T = 55$, while **Config-D** attains $EQ\text{-}T = 48$. 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 $EQ\text{-}T \sim 15$.

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

¹Note 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.

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 $\nabla_{x_t} \log p_\phi(\mathbf{c}|x_t, t)$. The best results are obtained by applying guidance on class-conditional models and scaling the classifier gradients by a constant $\lambda > 1$. This combination indicates that our model may also profit from classifier guidance, even though it already receives class information via embeddings.

We first pass the generated image \mathbf{x} through a pretrained classifier CLF to predict the class label c_i . We then add a cross-entropy loss $\mathcal{L}_{CE} = -\sum_{i=0}^C c_i \log CLF(x_i)$ as an additional term to the generator loss and scale this term by a constant λ . 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 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 in [Naeem et al. 2020]. Furthermore, we report sFID [Nash et al. 2021] to assess spatial structure. Lastly, sample fidelity and diversity are evaluated via precision and recall [Kynkäanniemi et al. 2019].

In Table 2, we compare StyleGAN-XL to the currently strongest GAN model (BigGAN-deep [Brock et al. 2019]) and diffusion models (CDM [Ho et al. 2022], ADM [Dhariwal and Nichol 2021]) on ImageNet. The values for ADM are calculated with and without additional methods (Upsampling \mathbf{U} and Classifier Guidance \mathbf{G}). For

StyleGAN2, we report numbers by [Grigoryev et al. 2022]. We find that StyleGAN-XL substantially outperforms all baselines across all resolutions in FID, sFID, rFID, and IS. An exception is recall, according to which StyleGAN-XL’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 increase 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 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_s . Then, w can be manipulated to achieve semantically meaningful edits [Goetschalckx et al. 2019; Shen et al. 2020].

Inversion. Standard approaches for inverting G_s 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: \mathcal{W}^+ . For \mathcal{W}^+ a separate w is chosen for each layer of G_s . 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 \mathcal{W}^+ that are close to the original distribution of \mathcal{W} . We follow [Karras et al. 2020b] and use the original latent space \mathcal{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; Voynov 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 following, we discuss limitations of the current model, which should be addressed in the future.

Architectural Limitations. First, StyleGAN-XL is three times larger than StyleGAN3, constituting a higher computational overhead when used as a starting point for finetuning. Therefore, it will be worth exploring GAN distillation methods [Chang and Lu 2020] that trade-off performance for model size. Second, we find StyleGAN3, and consequently, StyleGAN-XL, harder to edit, e.g., high-quality edits via \mathcal{W} are noticeably easier to achieve with StyleGAN2. As already observed in [Karras et al. 2021], StyleGAN3’s semantic controllability is reduced for the sake of equivariance. However, techniques using the *StyleSpace* [Wu et al. 2021], e.g., StyleMC [Kocasari et al. 2022], tend to yield better results in our experiments, confirming the findings of concurrent work by [Alaluf et al. 2022]. Furthermore, we remark that our framework can also easily be used with StyleGAN2 layers.

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	StyleGAN2	49.20					
CDM	3.52	128.80		128.80			BigGAN	6.95	7.36	75.24	202.65	0.87	0.28
ADM	5.91	5.09	13.29	93.31	0.70	0.65	CDM	4.88	158.70		158.70		
ADM-G	2.97	5.09	3.80	141.37	0.78	0.59	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
							StyleGAN-XL	2.30	4.02	7.06	265.12	0.78	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-G-U	3.85	5.86	210.83	221.72	0.84	0.53							
StyleGAN-XL	2.41	4.06	51.54	267.75	0.77	0.52							

Fig. 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.Fig. 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.

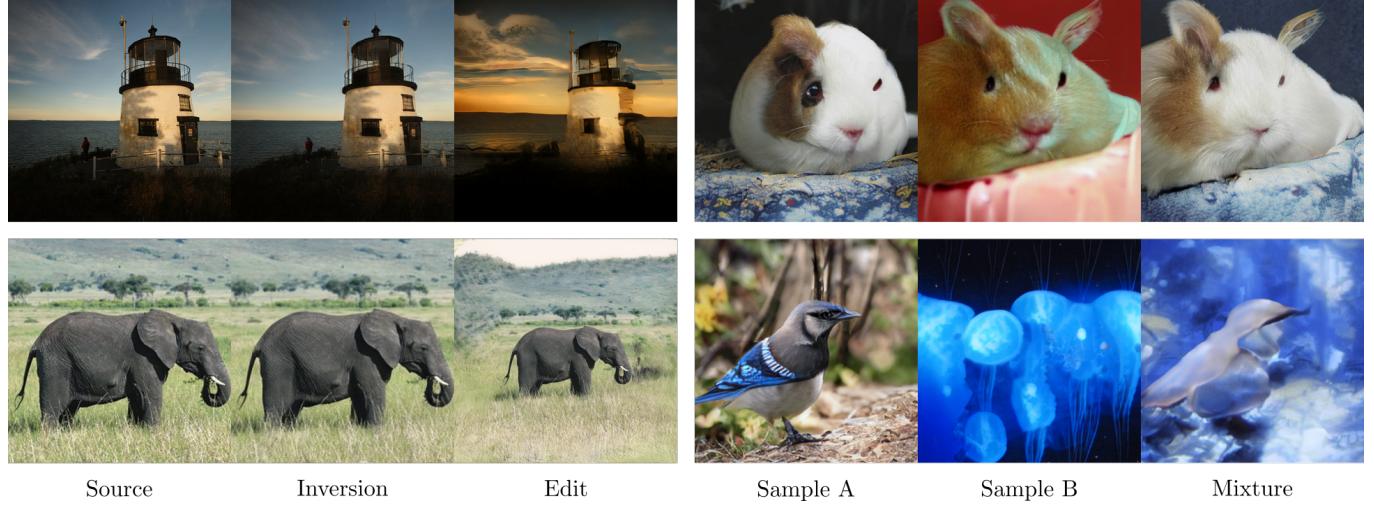


Fig. 5. Image Editing and Style Mixing. Left: First, a given image is inverted via PTI [Roich et al. 2021]. Right: Given two images, we can mix their styles. This method works for samples of the same or similar classes, and to a certain extent, for distant classes. For this experiment, we utilize random samples instead of inversions.

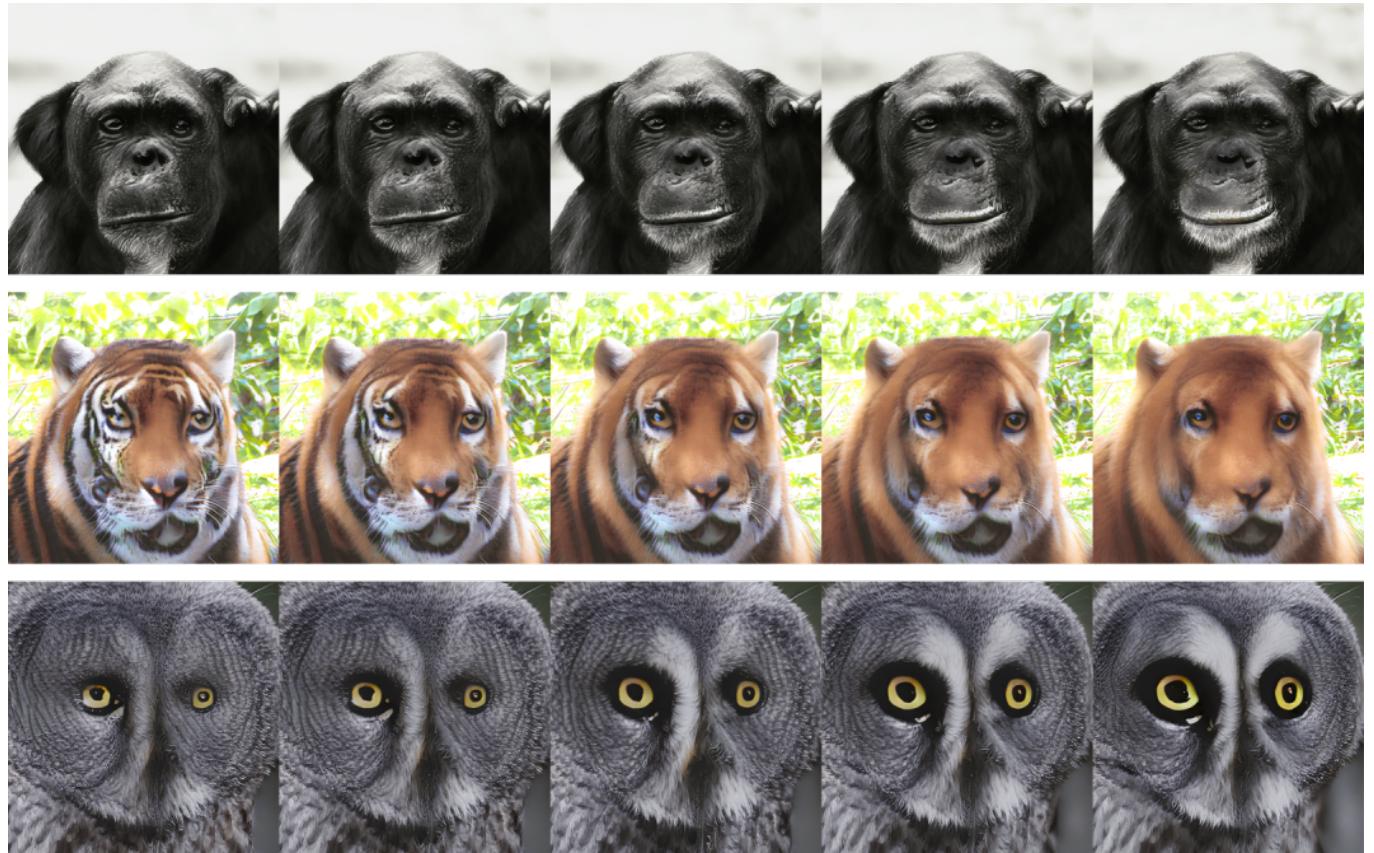


Fig. 6. Image Manipulation via Language. Given a random sample, we manipulate the image by following semantic directions in latent space found by StyleMC [Kocasari et al. 2022]. The latent space directions from top to bottom are: "smile", "no stripes", and "big eyes".

ACKNOWLEDGMENTS

We acknowledge the financial support by the BMWi in the project KI Delta Learning (project number 19A19013O). 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. Image2StyleGAN: 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, Niloy J. Mitra, and Peter Wonka. 2021. StyleFlow: Attribute-conditioned Exploration of StyleGAN-Generated Images using Conditional Continuous Normalizing Flows. *ACM Trans. on Graphics* 40, 3 (2021), 21:1–21:21. <https://doi.org/10.1145/3447648>
- 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)* abs/2104.02699 (2021). <https://arxiv.org/abs/2104.02699>
- Yuval Alaluf, Or Patashnik, Zongze Wu, Asif Zamir, Eli Shechtman, Dani Lischinski, and Daniel Cohen-Or. 2022. Third Time's the Charm? Image and Video Editing with StyleGAN3. *arXiv.org* abs/2201.13433 (2022). <https://arxiv.org/abs/2201.13433>
- Andrew Brock, Jeff Donahue, and Karen Simonyan. 2019. Large Scale GAN Training for High Fidelity Natural Image Synthesis. In *Proc. of the International Conf. on Learning Representations (ICLR)*. OpenReview.net. <https://openreview.net/forum?id=B1xsqj09Fm>
- Arantxa Casanova, Marlène Careil, Jakob Verbeek, Michal Drozdzal, and Adriana Romero-Soriano. 2021. Instance-Conditioned GAN. In *Advances in Neural Information Processing Systems (NeurIPS)*.
- Lucy Chai, Jun-Yan Zhu, Eli Shechtman, Phillip Isola, and Richard Zhang. 2021. Ensembling With Deep Generative Views. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. Computer Vision Foundation / IEEE, 14997–15007. https://openaccess.thecvf.com/content/CVPR2021/html/Chai_Ensembling_With_Deep_Generative_Views_CVPR_2021_paper.html
- Ting-Yun Chang and Chi-Jen Lu. 2020. TinyGAN: Distilling BigGAN for Conditional Image Generation. In *Proc. of the Asian Conf. on Computer Vision (ACCV) (Lecture Notes in Computer Science, Vol. 12625)*. Hiroshi Ishikawa, Cheng-Lin Liu, Tomás Pajdla, and Jianbo Shi (Eds.), 509–525. https://doi.org/10.1007/978-3-030-69538-5_31
- Xinlei Chen, Haoqi Fan, and Ross Girshick. 2020. Improved baselines with momentum contrastive learning. *arxiv.org*:2003.04297
- Edo Collins, Raja Bala, Bob Price, and Sabine Siessstrunk. 2020. Editing in Style: Uncovering the Local Semantics of GANs. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. Computer Vision Foundation / IEEE, 5770–5779. <https://doi.org/10.1109/CVPR42600.2020.00581>
- Antonia Creswell and Anil Anthony Bharath. 2019. Inverting the Generator of a Generative Adversarial Network. *IEEE Trans. Neural Networks Learn. Syst.* 30, 7 (2019), 1967–1974. <https://doi.org/10.1109/TNNLS.2018.2875194>
- Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. ImageNet: A large-scale hierarchical image database. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, 248–255. <https://doi.org/10.1109/CVPR.2009.5206848>
- Prafulla Dhariwal and Alex Nichol. 2021. Diffusion Models Beat GANs on Image Synthesis. (2021).
- 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. 2021. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *Proc. of the International Conf. on Learning Representations (ICLR)*. OpenReview.net. <https://openreview.net/forum?id=YicbFdNTTy>
- Andreas Fregin, Julian Müller, Ulrich Krebel, and Klaus Dietmayer. 2018. The DriveU Traffic Light Dataset: Introduction and Comparison with Existing Datasets. In *Proc. IEEE International Conf. on Robotics and Automation (ICRA)*. IEEE, 3376–3383. <https://doi.org/10.1109/ICRA.2018.8460737>
- Lore Goetschalckx, Alex Andonian, Aude Oliva, and Phillip Isola. 2019. GANalyze: Toward Visual Definitions of Cognitive Image Properties. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*. IEEE, 5743–5752. <https://doi.org/10.1109/ICCV.2019.00584>
- Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. 2014. Generative Adversarial Nets. In *Advances in Neural Information Processing Systems (NeurIPS)*.
- Zoubin Ghahramani, Max Welling, Corinna Cortes, Neil D. Lawrence, and Kilian Q. Weinberger (Eds.). 2672–2680. <https://proceedings.neurips.cc/paper/2014/hash/5ca3e9b122f618f06494c97biafcfb3-Abstract.html>
- Tom George Grigg, Dan Busbridge, Jason Ramapuram, and Russ Webb. 2021. Do Self-Supervised and Supervised Methods Learn Similar Visual Representations? *arxiv.org*:2110.00528
- Timofey Grigoryev, Andrey Voynov, and Artem Babenko. 2022. When, Why, and Which Pretrained GANs Are Useful?. In *Proc. of the International Conf. on Learning Representations (ICLR)*. <https://openreview.net/forum?id=4Ycr8oeCoh>
- 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/6fe43269967adb64ec6149852b5cc3e-Abstract.html>
- 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. In *Advances in Neural Information Processing Systems (NeurIPS)*. 6626–6637. <https://proceedings.neurips.cc/paper/2017/hash/8a1d694707eb0fefef65871369074926d-Abstract.html>
- Jonathan Ho, Chitwan 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=HylsTT4FvB>
- 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=Hk99zCeAb>
- Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. 2020a. Training Generative Adversarial Networks with Limited Data. In *Advances in Neural Information Processing Systems (NeurIPS)*. <https://proceedings.neurips.cc/paper/2020/hash/8d30aa96e72440759f74bd2306c1fa3d-Abstract.html>
- Tero Karras, Miika Aittala, Samuli Laine, Eriik 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. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. Computer Vision Foundation / IEEE, 4401–4410. <https://doi.org/10.1109/CVPR.2019.00453>
- Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. 2020b. Analyzing and Improving the Image Quality of StyleGAN. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. Computer Vision Foundation / IEEE, 8107–8116. <https://doi.org/10.1109/CVPR42600.2020.00813>
- Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *Proc. of the International Conf. on Learning Representations (ICLR)*.
- Umut Kocasari, Alara Dirik, Mert Tiftikci, and Pinar 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/2112.08493>
- Alex Krizhevsky, Geoffrey Hinton, et al. 2009. Learning multiple layers of features from tiny images. (2009).
- Tuomas Kynkänniemi, 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/0234c510bc6d908b28c70ff313743079-Abstract.html>
- Daiqing Li, Huan Ling, Seung Wook Kim, Karsten Kreis, Adela Barriuso, Sanja Fidler, and Antonio Torralba. 2022. BigDatasetGAN: Synthesizing ImageNet with Pixel-wise Annotations. *arXiv.org* (2022). <https://arxiv.org/abs/2201.04684>
- Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. 2021. SwinIR: Image Restoration Using Swin Transformer. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV) Workshops*. IEEE, 1833–1844. <https://doi.org/10.1109/ICCVW54120.2021.000210>
- Huan Ling, Karsten Kreis, Daiqing Li, Seung Wook Kim, Antonio Torralba, and Sanja Fidler. 2021. EditGAN: High-Precision Semantic Image Editing. *arXiv.org* abs/2111.03186 (2021). <https://arxiv.org/abs/2111.03186>
- Bingchen Liu, Yizhe Zhu, Kunpeng Song, and Ahmed Elgammal. 2021. Towards Faster and Stabilized GAN Training for High-fidelity Few-shot Image Synthesis. In *Proc. of the International Conf. on Learning Representations (ICLR)*. OpenReview.net. <https://openreview.net/forum?id=1Fqg133qRaI>
- Lars M. Mescheder, Andreas Geiger, and Sebastian Nowozin. 2018. Which Training Methods for GANs do actually Converge?. In *Proc. of the International Conf. on Machine learning (ICML) (Proceedings of Machine Learning Research, Vol. 80)*. PMLR, 3478–3487. <http://proceedings.mlr.press/v80/mescheder18a.html>

- Takeru Miyato, Toshiki 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=B1QRgziT>
- 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=ByS1VpgRZ>
- Muhammad Ferjad Naeem, Seong Joon Oh, Youngjung Uh, Yunjey Choi, and Jaejun Yoo. 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. <http://proceedings.mlr.press/v119/naeem20a.html>
- Charlie Nash, Jacob Menick, Sander Dieleman, and Peter W. Battaglia. 2021. Generating images with sparse representations. In *Proc. of the International Conf. on Machine learning (ICML)*. <http://proceedings.mlr.press/v139/nash21a.html>
- Alexander Quinn Nichol and Prafulla Dhariwal. 2021. Improved Denoising Diffusion Probabilistic Models. In *Proc. of the International Conf. on Machine learning (ICML) (Proceedings of Machine Learning Research, Vol. 139)*. PMLR, 8162–8171. <http://proceedings.mlr.press/v139/nichol21a.html>
- Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. 2021. StyleCLIP: Text-Driven Manipulation of StyleGAN Imagery. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*. 2085–2094.
- Guim Perarnau, Joost van de Weijer, Bogdan C. Raducanu, and Jose M. Álvarez. 2016. Invertible Conditional GANs for image editing. [arXiv.org/abs/1611.06355](https://arxiv.org/abs/1611.06355) (2016). <http://arxiv.org/abs/1611.06355>
- Etienne Perot, Pierre de Tournemire, Davide Nitti, Jonathan Masci, and Amos Sironi. 2020. Learning to Detect Objects with a 1 Megapixel Event Camera. In *Advances in Neural Information Processing Systems (NeurIPS)*. <https://proceedings.neurips.cc/paper/2020/hash/c213877427b46fa96cfcc39e837cce-Abstract.html>
- Phillip Pope, Chen Zhu, Ahmed Abdelkader, Micah Goldblum, and Tom Goldstein. 2021. The Intrinsic Dimension of Images and Its Impact on Learning. In *Proc. of the International Conf. on Learning Representations (ICLR)*. OpenReview.net. <https://openreview.net/forum?id=Xjk19XzGq2j>
- Alec Radford, Luke Metz, and Soumith Chintala. 2016. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. In *Proc. of the International Conf. on Learning Representations (ICLR)*, Yoshua Bengio and Yann LeCun (Eds.). <http://arxiv.org/abs/1511.06434>
- Maithra Raghu, Thomas Unterthiner, Simon Kornblith, Chiyuan Zhang, and Alexey Dosovitskiy. 2021. Do Vision Transformers See Like Convolutional Neural Networks? In *Advances in Neural Information Processing Systems (NeurIPS)*.
- Daniel Roich, Ron Mokady, Amit H. Bermano, and Daniel Cohen-Or. 2021. Pivotal Tuning for Latent-based Editing of Real Images. [arXiv.org/abs/2106.05744](https://arxiv.org/abs/2106.05744)
- Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J. Fleet, and Mohammad Norouzi. 2021. Image Super-Resolution via Iterative Refinement. arxiv.org/abs/2104.07636
- Tim Salimans, Ian J. Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. 2016. Improved Techniques for Training GANs. In *Advances in Neural Information Processing Systems (NeurIPS)*. 2226–2234. <https://proceedings.neurips.cc/paper/2016/hash/8a3363abe792db2d8761d6403605aeb7-Abstract.html>
- Axel Sauer, Kashyap Chitta, Jens Müller, and Andreas Geiger. 2021. Projected GANs Converge Faster. In *Advances in Neural Information Processing Systems (NeurIPS)*.
- Axel Sauer and Andreas Geiger. 2021. Counterfactual Generative Networks. In *Proc. of the International Conf. on Learning Representations (ICLR)*. OpenReview.net. <https://openreview.net/forum?id=BxewfAYMmjw>
- Yujun Shen, Ceyuan Yang, Xiaonan Tang, and Bolei Zhou. 2020. InterFaceGAN: Interpreting the Disentangled Face Representation Learned by GANs. *IEEE Trans. Pattern Anal. Mach. Intell.* 44, 4 (2020), 2004–2018. <https://doi.org/10.1109/TPAMI.2020.3034267>
- Yujun Shen and Bolei Zhou. 2021. Closed-Form Factorization of Latent Semantics in GANs. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. 1532–1540. https://openaccess.thecvf.com/content/CVPR2021/html/Shen_Closed-Form_Factorization_of_Latent_Semantics_in_GANs_CVPR_2021_paper.html
- Jiaming Song, Chenlin Meng, and Stefano Ermon. 2021. Denoising Diffusion Implicit Models. In *Proc. of the International Conf. on Learning Representations (ICLR)*. <https://openreview.net/forum?id=St1giarCHLP>
- Nurit Spingarn, Ron Banner, and Tomer Michaeli. 2021. GAN "Steerability" without optimization. In *Proc. of the International Conf. on Learning Representations (ICLR)*. OpenReview.net. https://openreview.net/forum?id=zDy_nQCXij
- Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E. Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. Going deeper with convolutions. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, 1–9. <https://doi.org/10.1109/CVPR.2015.7298594>
- Mingxing Tan and Quoc V. Le. 2019. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In *Proc. of the International Conf. on Machine learning (ICML) (Proceedings of Machine Learning Research, Vol. 97)*. Kamalika Chaudhuri and Ruslan Salakhutdinov (Eds.). PMLR, 6105–6114. <http://proceedings.mlr.press/v97/tan19a.html>
- Matthew Tancik, Pratul P. Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nitin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T. Barron, and Ren Ng. 2020. Fourier Features Let Networks Learn High Frequency Functions in Low Dimensional Domains. In *Advances in Neural Information Processing Systems (NeurIPS)*. <https://proceedings.neurips.cc/paper/2020/hash/55053683268957697aa39fba6f231c68-Abstract.html>
- Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. 2021. Training data-efficient image transformers & distillation through attention. In *Proc. of the International Conf. on Machine learning (ICML) (Proceedings of Machine Learning Research, Vol. 139)*. PMLR, 10347–10357. <http://proceedings.mlr.press/v139/touvron21a.html>
- Omer Tov, Yuval Alaluf, Yotam Nitzan, Or Patashnik, and Daniel Cohen-Or. 2021. Designing an Encoder for StyleGAN Image Manipulation. *ACM Trans. on Graphics* 40, 4 (2021). <https://doi.org/10.1145/3450626.3459838>
- Aäron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. 2017. Neural Discrete Representation Learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (Eds.). 6306–6315. <https://proceedings.neurips.cc/paper/2017/hash/7a98af17e63a0ac09c2e96d03992fb-Abstract.html>
- Andrey Voynov and Artem Babenko. 2020. Unsupervised Discovery of Interpretable Directions in the GAN Latent Space. In *Proc. of the International Conf. on Machine learning (ICML)*. PMLR, 9786–9796. <http://proceedings.mlr.press/v119/voynov20a.html>
- Daniel Watson, Jonathan Ho, Mohammad Norouzi, and William Chan. 2021. Learning to Efficiently Sample from Diffusion Probabilistic Models. *CoRR* abs/2106.03802 (2021). <https://arxiv.org/abs/2106.03802>
- Zongze Wu, Dani Lischinski, and Eli Shechtman. 2021. StyleSpace Analysis: Disentangled Controls for StyleGAN Image Generation. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. Computer Vision Foundation / IEEE, 12863–12872. https://openaccess.thecvf.com/content/CVPR2021/html/Wu_StyleSpace_Disentangled_Controls_for_StyleGAN_Image_Generation_CVPR_2021_paper.html
- Rui Xu, Xintao Wang, Kai Chen, Bolei Zhou, and Chen Change Loy. 2021. Positional Encoding As Spatial Inductive Bias in GANs. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. Computer Vision Foundation / IEEE, 13569–13578. https://openaccess.thecvf.com/content/CVPR2021/html/Xu_Positional-Encoding_As_Spatial_Inductive_Bias_in_GANs_CVPR_2021_paper.html
- Fisher Yu, Yinda Zhang, Shuran Song, Ari Seff, and Jianxiong Xiao. 2015. LSUN: Construction of a Large-scale Image Dataset using Deep Learning with Humans in the Loop. arxiv.org/abs/1506.03365
- Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. 2018. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. Computer Vision Foundation / IEEE Computer Society, 586–595. <https://doi.org/10.1109/CVPR.2018.00068>
- Xucong Zhang, Seonwook Park, Thabo Beeler, Derek Bradley, Siyu Tang, and Otmar Hilliges. 2020. ETH-XGaze: A Large Scale Dataset for Gaze Estimation Under Extreme Head Pose and Gaze Variation. In *Proc. of the European Conf. on Computer Vision (ECCV) (Lecture Notes in Computer Science, Vol. 12350)*. Springer, 365–381. https://doi.org/10.1007/978-3-03-58558-7_22
- Shengyu Zhao, Zhijian Liu, Ji Lin, Jun-Yan Zhu, and Song Han. 2020. Differentiable Augmentation for Data-Efficient GAN Training. In *Advances in Neural Information Processing Systems (NeurIPS)*. <https://proceedings.neurips.cc/paper/2020/hash/55479c55ebd1ef3f125f1337100388-Abstract.html>
- Jiapeng Zhu, Yujun Shen, Deli Zhao, and Bolei Zhou. 2020. In-Domain GAN Inversion for Real Image Editing. In *Proc. of the European Conf. on Computer Vision (ECCV)*, Andre Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm (Eds.). Springer, 592–608. https://doi.org/10.1007/978-3-03-58520-4_35
- In this supplemental document, we elaborate on increasing the resolution of ImageNet to one megapixel, compare to the baseline on a class containing humans, and specify the implementation details of our approach. The supplemental video shows additional samples and interpolations. We use the same mathematical notation as in the paper.

A PREPROCESSING IMAGENET

An initial challenge is the lack of high-resolution data; the mean resolution of ImageNet is 469×387. Similar to the procedure used for generating CelebA-HQ [Karras et al. 2018], we preprocess the whole



Fig. 7. **Imagenet Classes Containing Humans.** Samples for BigGAN and ADM are taken from [Dhariwal and Nichol 2021].

Table 3. **Inference speed comparison.** We measure the time required for a forward pass with batch size 1 in V100-seconds. ADM uses classifier guidance.

Model	Inference Time ↓		
	Res. 128 ²	Res. 256 ²	Res. 512 ²
ADM	27.07	40.26	91.54
StyleGAN-XL	0.05	0.07	0.10

dataset with SwinIR-Large [Liang et al. 2021], a recent model for real-world image super-resolution. Of course, a trivial way of achieving good performance on this dataset would be to draw samples from a 256² generative model and passing it through SwinIR. However, SwinIR adds significant computational overhead as it is 60 times slower than our upsampling stack. Furthermore, this way, StyleGAN-XL’s weights can be used for initialization when finetuning on other high-resolution datasets. Lastly, combining StyleGAN-XL and SwinIR would impair translation equivariance.

B CLASSES OF UNALIGNED HUMANS

We observe that ADM [Dhariwal and Nichol 2021] generates more convincing human faces than StyleGAN-XL and BigGAN. Both GANs can synthesize realistic faces; however, the main challenge in this setting is that the dataset is unstructured, and the humans are not aligned. [Brock et al. 2019] remarked the particular challenge of classes containing details to which human observers are more sensitive. We show examples in Fig. 7.

C INFERENCE SPEED

GANs generate samples in a single forward pass, unlike diffusion models that must be applied several hundred or thousand times to generate a sample. Table 3 compares StyleGAN-XL to ADM. We find that StyleGAN-XL is several orders of magnitude faster. In defense of diffusion models, speeding up their sampling is an active area of research, and novel techniques [Watson et al. 2021] may be able to reduce this gap in the future.

D RESULTS ON UNIMODAL DATASETS

StyleGAN-XL is designed to enable training on large and diverse datasets. However, applying it to big and small unimodal datasets is straightforward. In contrast to the configuration for ImageNet, we begin with ten layers at the lowest stage and add two layers per

Table 4. **Results on Unimodal Datasets.**

Model	FID	Model	FID
FFHQ 1024²			Pokémon 1024²
StyleGAN2	2.70	FastGAN	56.46
StyleGAN3	2.79	Projected GAN	33.96
StyleGAN-XL	2.02	StyleGAN-XL	25.47

Table 5. **Inversion Results.** The metrics are computed between the inversions obtained by the model and the reconstruction targets.

Model	MSE ↓	PSNR ↑	SSIM ↑	FID ↓
BigGAN	0.10	10.85	0.26	47.48
StyleGAN-XL	0.06	13.45	0.33	21.73

resolution stage. Furthermore, we do not employ classifier guidance. Table 4 reports the results for both datasets at resolution 1024², StyleGAN-XL achieves state-of-the-art performance on both.

E ADDITIONAL QUALITATIVE RESULTS

In the following, we present additional qualitative results. Fig. 8 shows additional interpolations between samples from different classes. Fig. 10 and Fig. 11 show samples on FFHQ 1024² and Pokemon 1024² respectively. Lastly, we compare BigGAN, ADM, and StyleGAN-XL on different ImageNet classes. For a fair comparison, we do not use truncation or classifier guidance. Instead, we show images with the largest logits given by a VGG16 which corresponds to individual image quality.

F IMPLEMENTATION DETAILS

Inversion. Following [Karras et al. 2020b], we use basic latent optimization in \mathcal{W} for inversion. Given a target image, we first compute its average style code $\bar{\mathbf{w}}$ by running 10000 random latent codes \mathbf{z} and target specific class samples \mathbf{c} through the mapping network. As the class label of the target image is unknown, we pass it to a pretrained classifier. We then use the classifier logits as a multinomial distribution to sample \mathbf{c} . In our experiments, we use Deit-base [Touvron et al. 2021] as a classifier, but other choices are possible. At the beginning of optimization, we initialize $\mathbf{w} = \bar{\mathbf{w}}$. The components of \mathbf{w} are the only trainable parameters. The optimization runs for 1000 iterations using the Adam optimizer [Kingma and Ba 2015] with default parameters. We optimize the LPIPS [Zhang et al. 2018] distance between the target image and the generated image. For StyleGAN-XL, the maximum learning rate is $\lambda_{max} = 0.05$. It is ramped up from zero linearly during the first 50 iterations and ramped down to zero using a cosine schedule during the last 250 iterations. For BigGAN, we empirically found $\lambda_{max} = 0.001$ and a ramp-down over the last 750 iterations to yield the best results. All inversion experiments are performed at resolution 512² and computed on 5k images (10% of the validation set). We report the results in Table 5 and show qualitative results in Fig. 9.

Table 6. Results on ImageNet at Lower Resolutions..

Model	FID ↓		
	Res. 16 ²	Res. 32 ²	Res. 64 ²
StyleGAN-XL	0.73	1.10	1.51

Training StyleGAN3 on ImageNet. For training StyleGAN3, we use the official PyTorch implementation². The results in Fig. 1 are computed with the StyleGAN3-R configuration on resolution 256² until the discriminator has seen 10 million images. We find that StyleGAN3-R and StyleGAN3-T converge to similar FID without any changes to their training paradigm. The run with the best FID score was selected from three runs with different random seeds. We use a channel base of 16384 and train on 8 GPUs with total batch size 256, $\gamma = 0.256$. The remaining settings are chosen according to the default configuration of the code release. For the ablation study in Table 1, we use the StyleGAN3-T configuration as baseline since StyleGAN-XL builds upon the translational-equivariant layers of StyleGAN3. We train on 4 GPUs with total batch size 256 and batch size 32 per GPU, $\gamma = 0.25$, and disable augmentation.

Training & Evaluation. For all our training runs, we do not use data amplification via *x-flips* following [Karras et al. 2020b]. Furthermore, we evaluate all metrics using the official StyleGAN3 codebase. For the baseline values in Table 2 we report the numbers of [Dhariwal and Nichol 2021]. The official codebase of ADM³ provides files containing 50k samples for ADM and BigGAN. We utilize the provided samples to compute rFID. Following [Dhariwal and Nichol 2021], we compute precision and recall between 10k real samples and 50k generated samples. Table 6 reports the results on ImageNet at lower resolutions.

Layer configurations. We start progressive growing at resolution 16² using 11 layers. The layer specifications are computed according to [Karras et al. 2021] and remain fixed for the remaining training. For the next stage, at resolution 32², we discard the last 2 layers and add 7 new ones. The specifications for the new layers are computed according to [Karras et al. 2021] for a model with resolution 32² and 16 layers. Continuing this strategy up to resolution 1024² yields the flexible layer specification of StyleGAN-XL in Fig. 15.

²<https://github.com/NVlabs/stylegan3.git>³<https://github.com/openai/guided-diffusion>



Fig. 8. **Interpolations.** StyleGAN-XL generates smooth interpolations between samples of different classes.

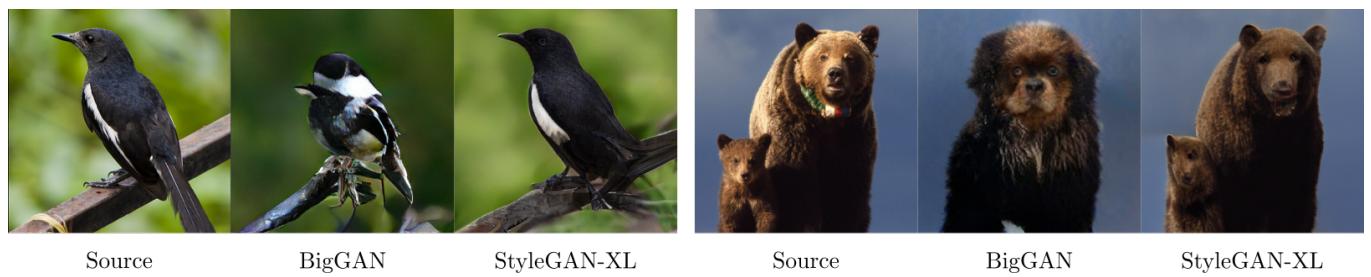


Fig. 9. **Inversion of a Given Source Image.** For BigGAN, we invert to its latent space z , for StyleGAN-XL we invert to style codes w .

Fig. 10. Samples on FFHQ 1024².

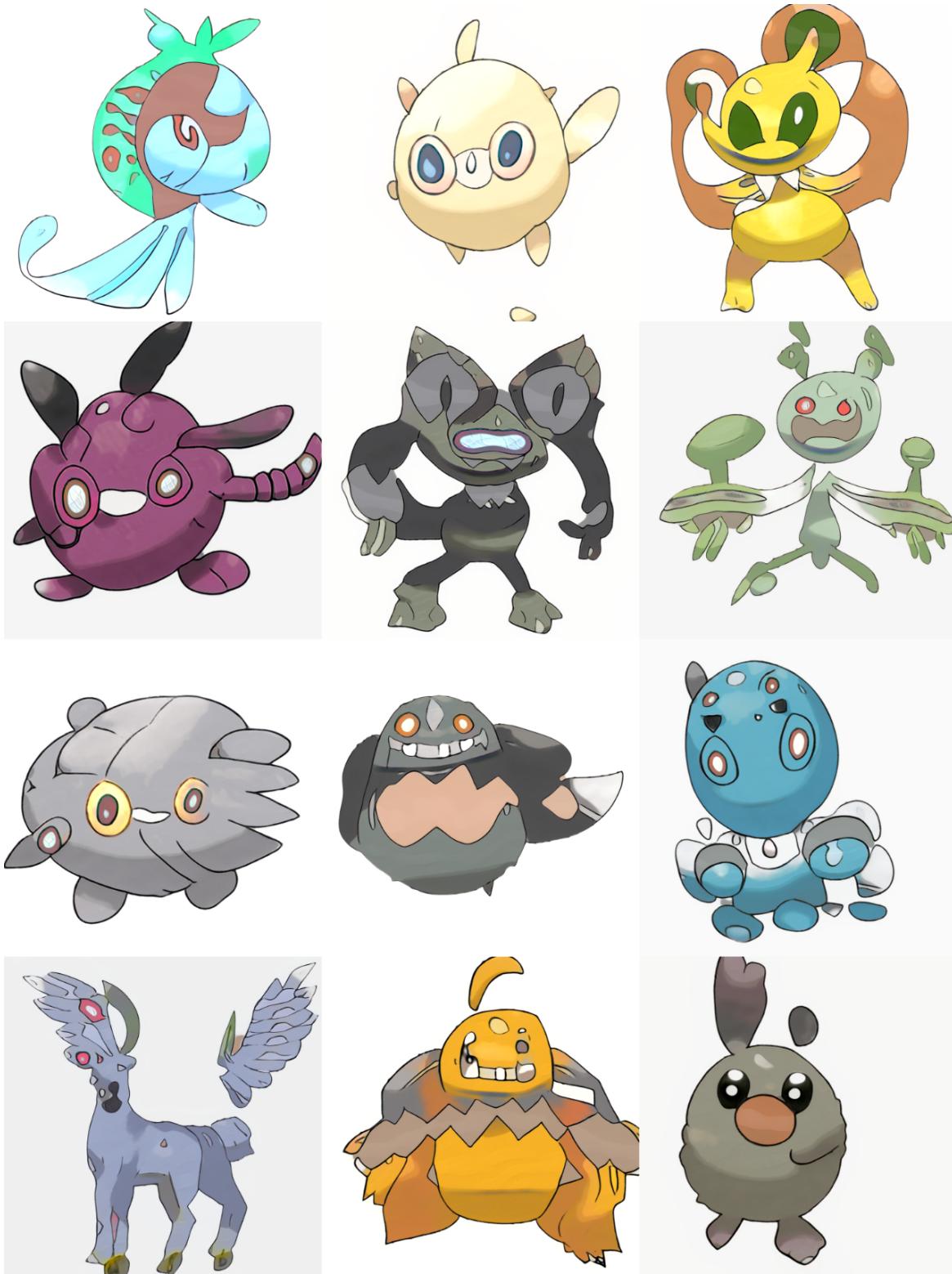
Fig. 11. Samples on Pokemon 1024².



Fig. 12. Qualitative Comparison on ImageNet 256². We compare BigGAN (left column), ADM (middle column), and StyleGAN-XL (right column). Classes from top to bottom: pizza, valley, daisy, dough, comic book.

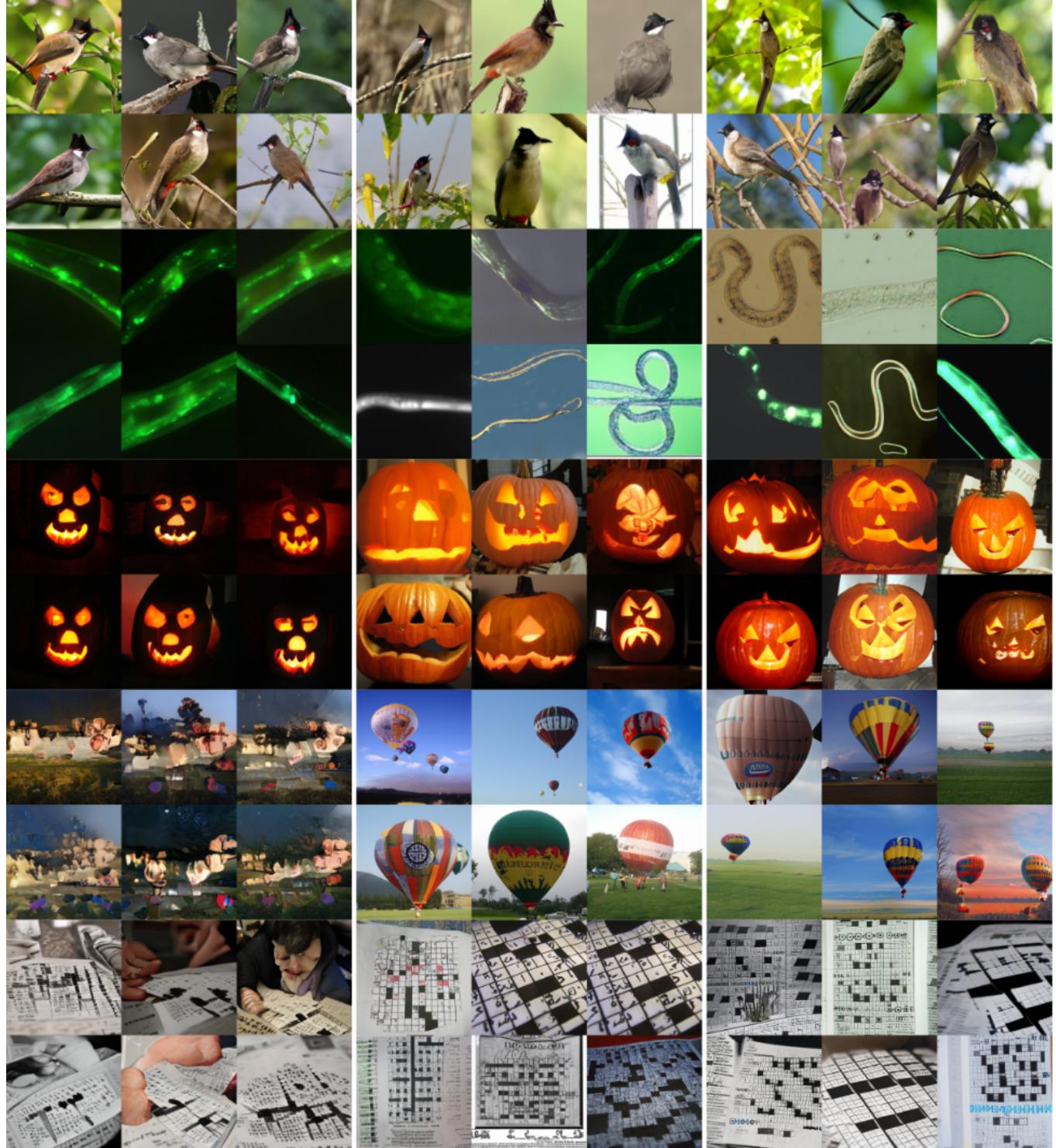


Fig. 13. **Qualitative Comparison on ImageNet 256².** We compare BigGAN (left column), ADM (middle column), and StyleGAN-XL (right column). Classes from top to bottom: bulbul, nematode, jack-o'-lantern, balloon, crossword puzzle.



Fig. 14. Qualitative Comparison on ImageNet 256². We compare BigGAN (left column), ADM (middle column), and StyleGAN-XL (right column). Classes from top to bottom: agaric, orange, Tibetan mastiff, espresso, paddlewheel.

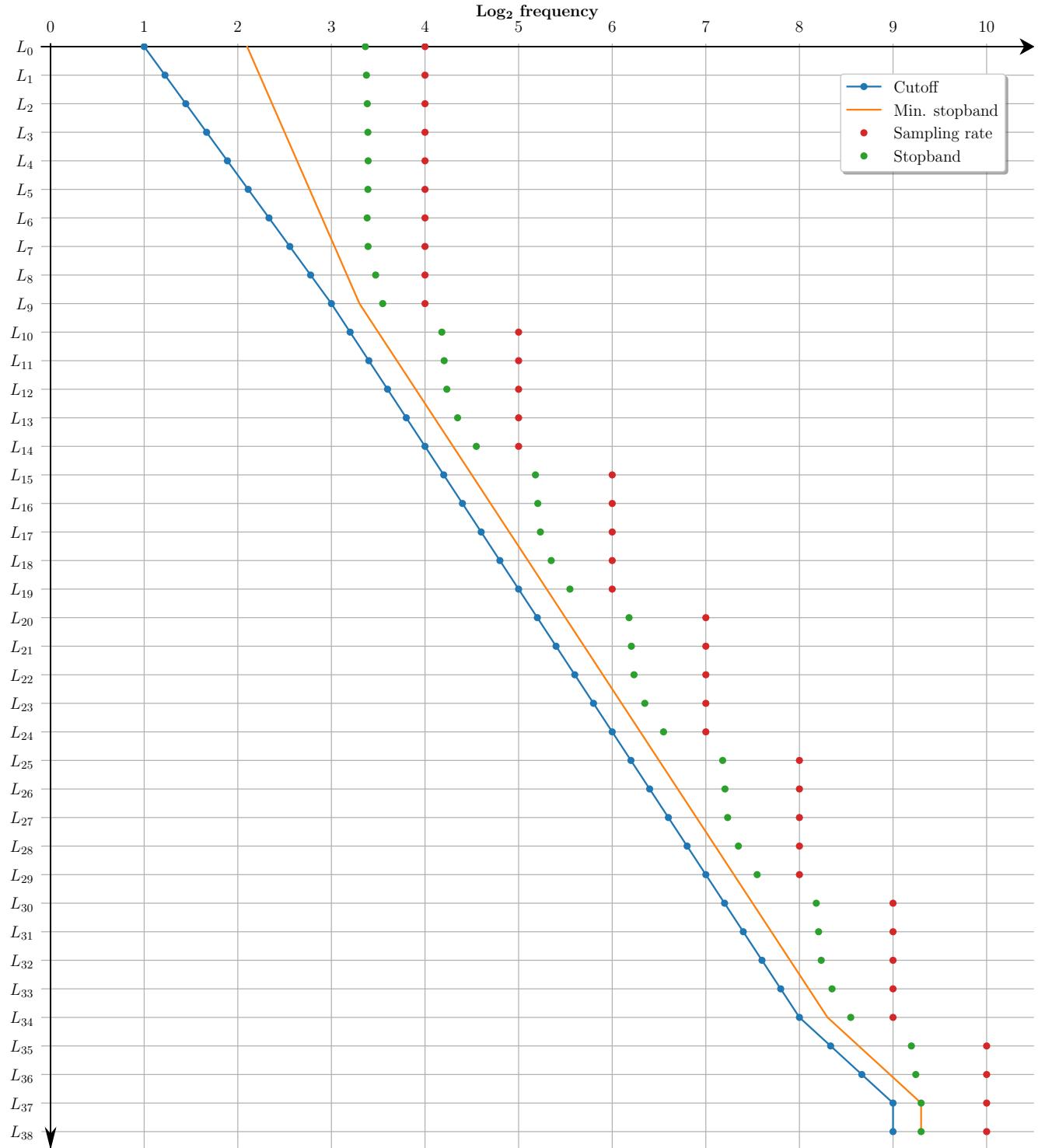


Fig. 15. **Flexible Layer Specification of Stylegan-XL.** StyleGAN-XL consists of 39 layers at resolution 1024^2 . Cutoff (blue) and minimum acceptable stopband frequency (orange) obey geometric progression over the layers; sampling rate (red) and actual stopband (green) are computed according to our design constraints.