# Any-resolution Training for High-resolution Image Synthesis: Supplementary Materials

Lucy Chai[1], Michaël Gharbi[2], Eli Shechtman[2]
Phillip Isola[1], and Richard Zhang[2]

[1]MIT [2]Adobe Research

In the supplementary, we first demonstrate an extension of our approach towards panorama generation (Section 1). In Section 2, we provide additional qualitative and quantitative comparisons to baselines (super-resolution methods, discrete-resolution and oracle generators), explore additional model variations, and investigate the detectability of our method using an off-the-shelf forensics method. We provide implementation details in Section 3.

## 1 Panorama generation extension

Our default training setup assumes that we use low-resolution images to learn global context and patches from high-resolution images to learn details. An alternative setup to learn from patches directly, *without ever knowing the entire global context*. One such scenario is panorama generation, where a large-scale dataset would be much more difficult to obtain than single images. We investigate this setup on the Mountains domain, in which the generator is tasked with synthesizing a panorama from landscape images, without training directly on panoramas. Accordingly, we modify the $[0, 1] \times [0, 1]$ coordinate grid to $[-\pi, \pi] \times [0, 1]$, and enforce continuity on the endpoints by using a sine and cosine encoding prior to Fourier feature embedding. At training time, we sample a "slice" of the coordinate grid for generation corresponding to a random viewing angle, but at inference time the entire panorama can be synthesized by specifying the full grid of coordinates. In this case, we find that it is important to use a *cross-frame* discriminator, in which the discriminator straddles the boundary between two generated slices to enable seamless boundaries in the panorama. Qualitative results are shown in Fig 2. At inference time, we can spatially interpolate the $w$ latent code with arbitrary spacing, which generates seamless infinite landscapes.

Fig. 1: Panorama generation from patches. We modify our training framework to train without the global image context. We map our coordinate grid to $[-\pi, \pi] \times [0, 1]$ and use a cross-frame discriminator to enable seamless transitions between patches. The model is trained with FOV $= 60°$. The vertical white line indicates a full $360°$ revolution.

Fig. 2: By spatially adjusting the latent code only at inference time, the same model is capable of generating infinite landscapes, shown on multiple lines here for visibility.

## 2 Experiments

### 2.1 Dataset Collection

To collect our varied-size dataset, we scrape image collections from Flickr photo groups (Tab. 1). In cases where a standard fixed-resolution dataset is available (*e.g.* LSUN Churches [14]), we seek to find photos that approximately match the domain of the standard dataset. Due to domain mismatches between LSUN and the photos scraped from Flickr, we manually filter the collected images to approximately match the LSUN domain, which remains tractable for the few thousand HR images used in the patch-based training phase. As is standard practice [10,7], and to not violate license permissions, we will release the image IDs but not the images directly.

Table 1: Image sources for construction of our varied-resolution datasets.

| Domain | Flickr Source |
|---|---|
| Church | Church Exteriors |
| Mountains | Mountains Anywhere |
| Birds | Birding in the Wild |

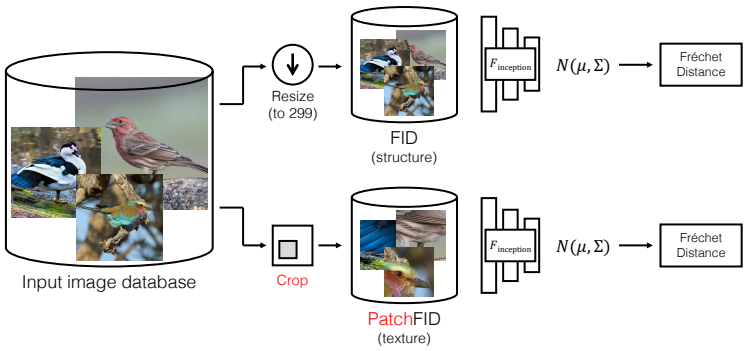


Fig. 3: Preprocessing for FID and patch-FID (pFID). FID evaluates global structure, but downsamples all images to a common 299px size which ignores higher resolution details. pFID takes images crops instead rather than downsampling to capture texture realism at higher resolutions. We use both to measure structure and texture properties.

### 2.2 Patch-FID

We describe details of our Patch FID metric, introduced in Section 4.1 of the main paper. The metric is aimed at better capturing the realism of details at high resolution by avoiding downsampling to capture texture details. In our setting, we have a smaller number of real images present at various resolutions. Standard FID, which focuses on global structure, does not capture these varied-resolution details. We modify the FID pipeline to avoid downsampling global images at higher resolutions to a fixed 299 pixel width. Instead, we randomly sample patches of size $p$ from real images at global scale $s$ and locations $c_{\mathbf{v},s}$, and generate the corresponding patch $G\ (z,c_{\mathbf{v},s},s)$. The number of samples is crucial to getting an accurate estimate in FID calculation [1], and 50,000 is typically used. A benefit of this patch-sampling procedure also means that we can obtain the necessary large number of patches for FID computation, more than the available number of real images in the HR dataset. In the Mountains generator, where the patch size $p$ is larger than 299, we subsequently also select a random 299-pixel crop from the patches to compute the image features. Because this avoids downsampling the generated content, we find that it is more sensitive to image quality

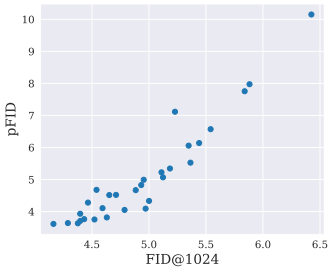


Fig. 4: pFID vs FID@1024. The metrics are largely correlated, but FID@1024 generates images at 1024 resolution and then downsamples images, assuming a fixed-size dataset. On other domains where a ground-truth HR dataset is not available, we primarily use pFID to measure sample quality.

Table 2: Alternative FID evaluation metrics. We primarily use pFID, which avoids downsampling synthesized content and evaluates multiscale patches, as an evaluation metric. Here, we also report FID at the base resolution from the result of fixed-size pretraining (Fixed-Size), and compare to global FID metrics after downsampling the HR images to a common size. On FFHQ, we also tried applying GFP-GAN [12] which is a facial super-resolution model.

| | **FFHQ6K** | | | **Church** | | | **Birds** | | | **Mountain** | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | FID | | pFID | FID | | pFID | FID | | pFID | FID | pFID | |
| | 256 | 1024 | random | 256 | 1024 | random | 256 | 512 | random | 1024 | down | random |
| Fixed-Size | 3.71 | 33.80 | 52.95 | 3.39 | 242.10 | 146.24 | 3.92 | 12.69 | 55.42 | 3.09 | 13.42 | 46.20 |
| Upsample | – | 11.70 | 17.29 | – | 14.21 | 80.48 | – | 7.67 | 30.57 | – | 4.53 | 20.00 |
| LIIF | – | 7.05 | 22.93 | – | 18.66 | 83.88 | – | 7.29 | 30.19 | – | 4.55 | 23.10 |
| Real-ESRGAN | – | 19.04 | 16.92 | – | 12.26 | 23.04 | – | 8.51 | 16.10 | – | 7.60 | 19.05 |
| GFP-GAN | – | 19.15 | 16.27 | – | – | – | – | – | – | – | – | – |
| Ours | 3.34 | 4.06 | 2.96 | 3.84 | 6.98 | 9.89 | 3.78 | 6.29 | 6.52 | 3.14 | 4.33 | 7.99 |

at high resolutions. Using the full FFHQ dataset as ground-truth, we find that our patch-FID metric is largely correlated to the standard FID numbers at 1024 resolution (Fig. 4). Therefore, we use Patch-FID as a metric of sample quality on our datasets collected from Flickr, when a full high-resolution dataset of images all at the same resolution is not available.

### 2.3 Additional quantitative results

In Table 3 of the main text, we report comparisons of our method and off-the-shelf super-resolution methods using the patch-FID metric. We report additional metrics in Table 2 here, including the FID at base resolution (the result of the pretraining step), and FID at a higher resolution after downsampling all images in the HR dataset (between 5k-10k images, which is lower than the typical 50k used to compute FID) to a common size. Notably, the base resolution FID is largely similar before and after patch-based training, and in the case of FFHQ and Birds, patch-based training at higher resolutions even improves the low-resolution FID. Without direct multi-scale training, however, the fixed-size model obtained from the pretraining step does not naturally generalize to higher resolutions. We find that our pFID metric is more discriminative to differences in image quality at higher resolutions. In particular, the LIIF super-resolution model tends to obtain better FID@1024 (corresponding to super-resolving generated images to 1024 resolution and computing FID) compared to Real-ESRGAN, but the outputs are visually blurry. Because our pFID does not perform downsampling, it can better capture this blurriness, reflected in an increased pFID. In another variant, we compute pFID on patches of size 1024 synthesized by the Mountain generator, and then subsequently downsample them, which we denote as pFID (down), rather than cropping. Again, we find that this downsampling operation can obscure image deterioration at higher resolution, producing artificially lower FID scores compared to pFID computed without downsampling.

### 2.4 Comparison to powers-of-two synthesis

Using the same set of generator weights, our model can synthesize images at a specified scale by simply providing the corresponding $s$ and $c_{\mathbf{v},s}$ inputs. On the other hand, other methods for multi-resolution synthesis [6,4,2,3] generate images that are iteratively enlarged by a factor of two, by adding additional network layers. These methods are typically introduced to impove training stability. For this baseline, we modify the recent Anycost-GAN [6] framework to fit our varied-size training setting. Specifically, we downsample all images in FFHQ6K to the nearest power of two, and train the corresponding network layers only on the appropriate subset of data. To generate at any resolution below 1024, we take the nearest model output that is larger than the target resolution, and apply Lanczos downsampling. Similar to our approach, we start with a pretrained model at 256 resolution, and initialize both the generator and discriminator with pretrained weights. Because each increase in output resolution involves training additional weights, and the number of images at a given resolution decreases as resolution increases, we find that this training approach yields visual artifacts at higher resolutions, shown in Fig. 5.

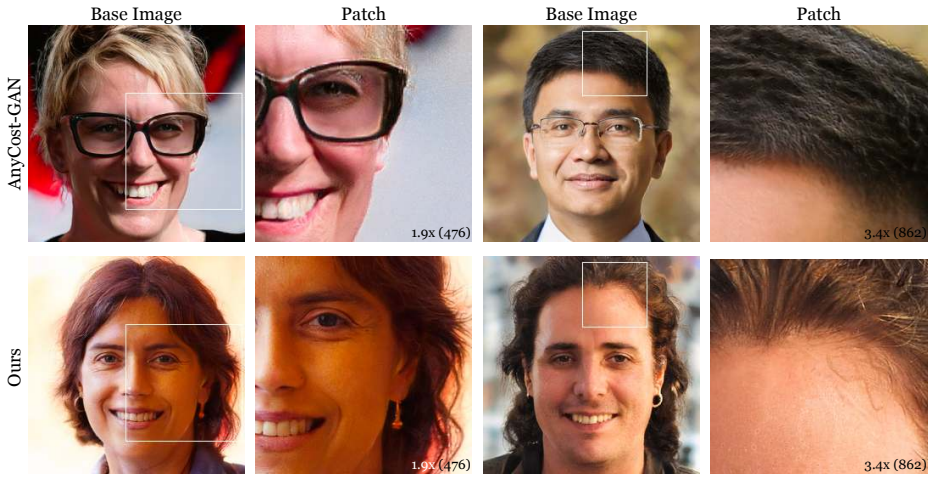


Fig. 5: Using the same FFHQ6K dataset, we train an Anycost-GAN [6] and compare it to our model. While Anycost-GAN adds additional modules to increase synthesis resolution, our model shares weights across resolutions. Note that the output from Anycost-GAN contain more visual artifacts, particularly in finely textured regions such as hair.

### 2.5 Comparison to Oracle Generator

In Section 4 of the paper, we describe our experimental setup on the face domain, and in Table 5 in the paper, we show competitive performance training on few HR

Table 3: Comparison of our patch generator (6k images, varied sizes) to oracle generators which train on the entire FFHQ dataset (70k images, 1024 resolution). Although the oracle generators attain better FID, our method enables synthesis at continuous resolutions and can train without assuming that all images are resized to a common resolution. We also include pFID at the maximum 1024 scale.

| | FID | | | pFID | |
|---|---|---|---|---|---|
| | 256 | 512 | 1024 | random | 1024 |
| SGAN2 Oracle | 3.05 | 2.81 | 2.69 | 2.26 | 4.83 |
| SGAN3 Oracle | 3.54 | 3.23 | 3.06 | 2.44 | 4.29 |
| Patch (Ours) | 3.34 | 3.71 | 4.06 | 2.96 | 8.01 |

images, even compared training on the whole HR dataset. We provide additional details and visualizations here.

We use the FFHQ dataset as a collection of 70k high-resolution ground-truth images. To simulate more "in-the-wild" settings, we use a fraction (6k) of HR images for patch-based training with a generator of size $p = 256$, where only 1k of the images are the full 1024 resolution, and the remainder are uniformly downsampled between 512 and 1024 prior to training. As a comparison, we also evaluate two oracle models that train a generator directly for the $s = 1024$ global image, using the entire 70K images in the FFHQ dataset, and Lanczos downsample the result for FID computations at other resolutions. We also evaluate a variant of pFID, by holding the scale *fixed* at the maximal resolution and randomly sampling crop locations (pFID 1024), in addition to our original pFID metric that randomly samples both scale and location (pFID random). Note that this evaluation is only possible for FFHQ controlled setting, as all images are present at the maximal 1024 size.

Despite being trained to generate patches, our generator can approximately match the frequency content in real images, and that of a StyleGAN3 model trained for 1024 resolution generation on the full FFHQ dataset (Fig. 6). While StyleGAN2 achieves better FID than StyleGAN3, we find that it has a different frequency profile that is less similar to that of real images. We compare the FID of these oracle models with our continuous patch model in Tab. 3. While the oracles can achieve lower FID and pFID variants, we note that training the oracle assumes that a sufficient number of high-resolution images of the same size are available, and trains the model specifically for a fixed resolution, whereas we employ mixed resolution training on fewer than 10% of the full HR dataset. Our training strategy therefore allows us to take advantage of the varied resolutions of images in the wild, which is not possible in the oracle setting.

### 2.6 Additional Model Variations

In Section 4.2 of the main text, we describe and study variations of our model. Here, we provide additional quantitative and qualitative results and study additional factors.

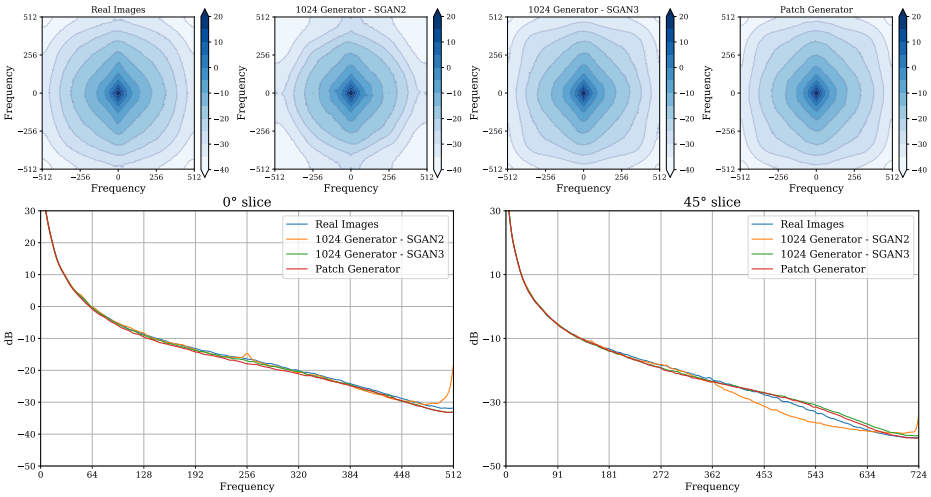


Fig. 6: Comparison of frequency distribution. We plot the frequency spectrum of real images, StyleGAN2 and StyleGAN3 trained on the entire FFHQ dataset at 1024 resolution, and our Patch Generator which is trained on $p \times p$ patches of FFHQ6K (which contains approximately 1k images at 1024 resolution and 5k at lower resolutions). The frequency distributions are similar, suggesting that even a smaller generator is able to approximate fine textures well.

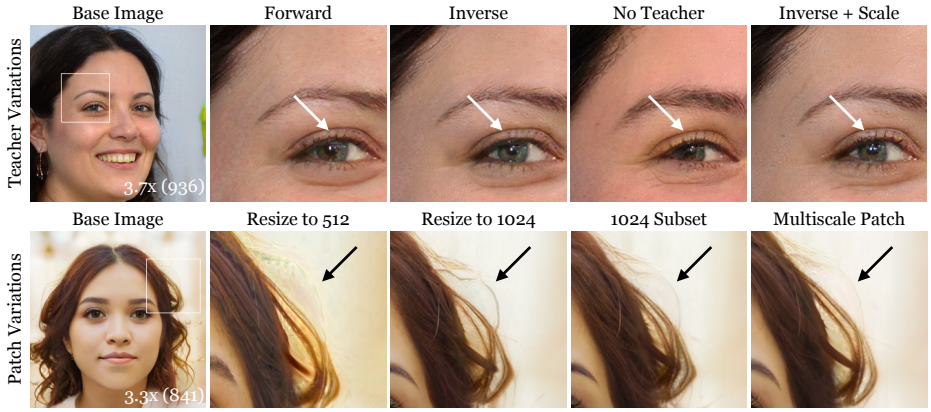


Fig. 7: Qualitative examples of model variations. (Top) Using an inverse teacher loss and the scale conditioning branch generates sharper details while also preserving similarity to the base image. (Bottom) We compare our multi-size training approach to methods that do not take advantage of different image sizes and instead train for a fixed resolution. Fixed-resolution training cannot generalize to other resolutions, and upsampling images leads to blurring. Our final model is able learn from mixed-resolution training images and also synthesize at arbitrary resolutions.

Table 4: We evaluate additional precision and recall metrics [9], and their corresponding patch variants, for our model, naive upsampling, and super-resolution methods on the FFHQ dataset. We also include pFID at a fixed 1024 scale evaluated in Tab. 3.

| | Precision | | Recall | | pFID |
|---|---|---|---|---|---|
| | 1024 | Patch | 1024 | Patch | 1024 |
| Upsample | 0.68 | 0.77 | 0.37 | 0.20 | 31.70 |
| Real-ESRGAN | 0.40 | 0.54 | 0.51 | 0.47 | 20.04 |
| GFP-GAN | 0.48 | 0.62 | 0.34 | 0.32 | 20.58 |
| Ours | 0.69 | 0.68 | 0.47 | 0.55 | 8.01 |

Table 5: Variations of teacher regularizer on FFHQ6k. The inverse teacher regularization outperforms forward regularization, and adding a scale-conditioning branch further improves higher resolutions. Omitting the teacher harms global structure. (*) indicates default setting.

| | FID | | | pFID |
|---|---|---|---|---|
| | 256 | 512 | 1024 | random |
| Forward teacher | 3.23 | 4.21 | 5.35 | 6.06 |
| Inverse teacher | 3.35 | 4.18 | 4.88 | 4.67 |
| No teacher | 5.50 | 5.93 | 7.13 | 3.17 |
| Inverse + scale (*) | 3.37 | 4.41 | 4.47 | 4.28 |

Table 6: Teacher regularization weight trades off between improved detail synthesis (pFID) and global realism (full image FIDs). We choose an in-between value ($\lambda_{\text{teacher}} = 5$); this value can be adjusted based on desired similarity to the base resolution. (*) indicates our default setting.

| | FID | | | pFID | L1 |
|---|---|---|---|---|---|
| | 256 | 512 | 1024 | random | |
| $\lambda_{\text{teacher}} = 0$ | 5.50 | 5.93 | 7.13 | 3.17 | 0.16 |
| $\lambda_{\text{teacher}} = 2$ | 3.42 | 4.58 | 5.46 | 3.15 | 0.10 |
| $\lambda_{\text{teacher}} = 5$ (*) | 3.37 | 4.41 | 4.47 | 4.28 | 0.08 |
| $\lambda_{\text{teacher}} = 10$ | 3.46 | 4.25 | 4.61 | 5.39 | 0.07 |

Table 7: We sample patches from the HR dataset at global resolutions between $(s_{\min}, s_{\max})$. The same model architecture trained on patches from higher resolution images improves the synthesis result at 1024 resolution. (*) indicates our default setting.

| | FID | | | pFID |
|---|---|---|---|---|
| | 256 | 512 | 1024 | random |
| (256, 512) | 5.20 | 5.92 | 19.01 | 35.66 |
| (256, 1024) | 3.43 | 4.16 | 4.61 | 4.19 |
| (512, 1024) (*) | 3.28 | 4.04 | 4.16 | 3.61 |

**Alternative metrics** In addition to FID and pFID, we also report precision and recall [9] for our model, naive upsampling, and super-resolution models (Tab. 4). Super-resolution obtains lower precision (suggesting out-of-distribution results) and similar or lower recall. Upsampling obtains similar or better precision, but lower recall (suggesting that is does not sufficiently cover the real image distribution). Here, we also include pFID measured at the maximum resolution for FFHQ (1024).

**Variations on teacher regularization.** In the main text, we introduce variations on the teacher regularization including "forward" and "inverse" loss formulations, and discarding the teacher regularization all-together. Tab. 5 shows the FID comparisons of these three variants, in which the "inverse" loss obtains the best FID scores at the highest 1024 resolution. Adding the scale-conditioning branch to inject scale information throughout the generator further improves FID@1024 and pFID. We show qualitative examples in Fig. 7 (top), where the inverse teacher with scale-conditioning input can synthesize the cleanest details while still being similar to the base image.

As default, we set $\lambda_{\text{teacher}} = 5$ during the patch-based training phase. Changing $\lambda_{\text{teacher}}$ balances between local image quality and similarity to the base resolution image, where higher $\lambda_{\text{teacher}}$ offers the most similarity to the base resolution with lower L1 difference, but lower $\lambda_{\text{teacher}}$ improves pFID, suggesting better quality of the synthesized patches (Tab. 6).

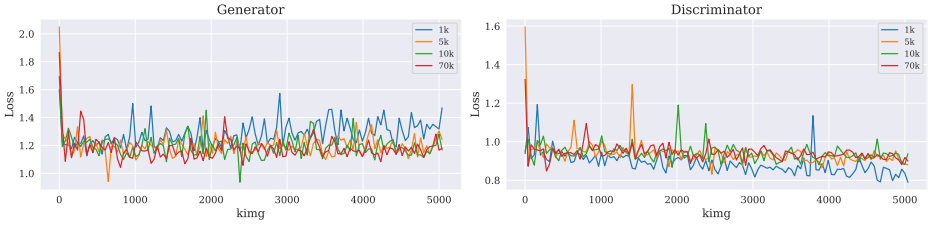


Fig. 8: Number of training images. While FID numbers are similar, we find that using 1K HR images for training shows some evidence of divergence. The training dynamics of 5K images is similar to that of the full dataset.

**Fixed-size vs Multi-size training.** Fig 7 (bottom) shows an example of a synthesized patch comparing our multi-size training to strategies of fixed-size training. Fixed-size training does not naturally generalize to other sizes, causing deterioration in image quality when sampled at resolutions not equal to the training resolution. Upsampling the training images to a common resolution introduces blurriness in the synthesized output. The result of training on only the subset of images at 1024 resolution looks qualitatively similar to that of multi-scale training, but multi-scale training attains better FID metrics and is able to use more images for training.

**Changing the number of training images.** While the model FID scores remain largely similar (within a range of 0.3) when training on 1k to 70k high-resolution images, we found that using 1k images showed some evidence of training divergence (Fig. 8). On the other hand, the training trajectory of using 5k images looks largely similar to that of using the full HR dataset (70k) images. Therefore, when collecting images for the remaining domains, we aim to collect between 5k-10k images to construct the HR dataset.

**Investigating the impact of sampling resolutions.** Our FFHQ6k dataset contains images between 512 and 1024 resolution, and during training the images are randomly downsampled from their native resolution, and can be optionally clipped at an upper resolution. Here, we conduct experiments to study the effects of these sampling ranges. When training the model on resolutions $s$ sampled between 256 and 512, the image quality declines by 1024 resolution at inference time and contains visual artifacts (Tab. 7, Fig. 9). Taking the same image and model architecture, but instead training on resolutions between 256 and 1024 offers better FID@1024, and sampling from 512 to 1024 resolution further improves FID@1024. As before, all models are trained on patches of size $p = 256$,

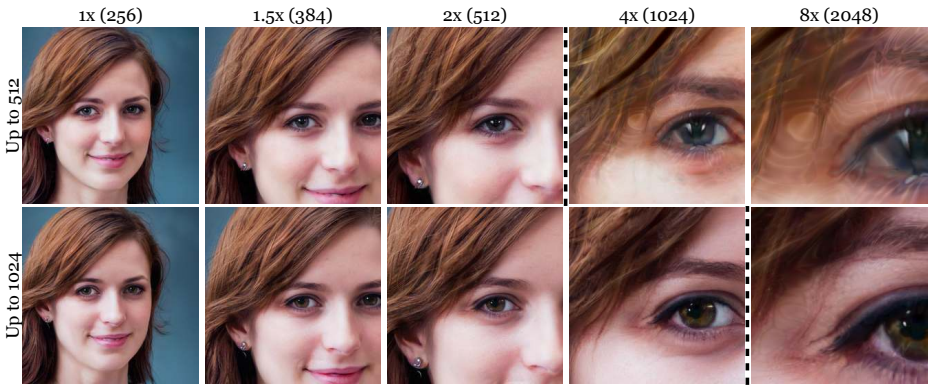


Fig. 9: Impact of sampling resolutions. Using the same model architecture and FFHQ6k dataset, we sample images (top) from 256 to 512 resolution, and (bottom) from 512 to 1024 resolution. The dotted line indicates when inference resolution exceeds the maximum training resolution. Watery artifacts start to appear when extrapolation, but this can be tempered by simply training on patches from larger images.

and the model is jointly trained on the fixed-size dataset to preserve FID@256. Accordingly for the other domains, our sampled resolutions for HR dataset range between the native resolution $s_{\text{im}}$ and the minimum resolution of the HR images. These results suggest that the synthesized resolution can be dictated by the training images; simply adding patches from higher resolution images can allow the same model to better synthesize at a higher resolution.We also tried an experiment using a separately trained smaller 128px patch generator and the same 512 to 1024 resolution patch sampling scheme, but obtained worse FID (8.38 at 1024 resolution compared to 4.16 for our default model); we hypothesize this is because may be due to worse FID from the initial pretraining phase that carries over to the patch training phase (5.14 compared to 3.71 for our default model).

**Changing the discriminator.** Our final model introduces changes to the generator, but keeps the same discriminator from the initial pretraining step. During patch-training, the discriminator must also learn to distinguish between real and synthesized patches. Here, we investigate alternatives of changing the discriminator setup (Tab. 8). (1) We remove sampling from the LR dataset, now causing the discriminator to focus entirely on patches. This causes pFID to improve but the remaining global FIDs to worsen. In particular, this allows the

Table 8: Discriminator variations. Our default discriminator, which jointly trains globally on the LR dataset and patches from the HR dataset attains the best FID metrics. Other changes to the discriminator did not improve performance. (*) indicates our default setting.

| | FID | | | pFID |
|---|---|---|---|---|
| | 256 | 512 | 1024 | random |
| Default Discriminator (*) | 3.28 | 4.04 | 4.16 | 3.61 |
| No Base Resolution | 9.96 | 4.23 | 4.69 | 2.92 |
| Two Discriminators | 3.82 | 4.63 | 5.34 | 3.38 |
| Scale-conditioned Discriminator | 31.81 | 71.33 | 89.38 | 120.06 |

generator to forget how to synthesize at the base resolution, causing a large increase in FID@256. The impact of sampling from the base resolution and the teacher regularization have similar outcomes: both encourage global coherence, but the teacher has a stronger effect than base resolution training. (2) We also try adding a second discriminator so that one focuses entirely on the global low-resolution image, and the other entirely on patches. Both discriminators are initialized with the result from pretraining, but we find that this setting leads to suboptimal metrics, compared to using a single discriminator. (3) We inject scale information into the discriminator following a similar method as the generator via weight modulation. In this case, the training becomes unstable as the discriminator is able to out-compete the generator.

**Removing the pretraining step.** Our final model is first pretrained at a fixed, smaller resolution before varied-size patch training is enabled. This pretraining phase encourages global coherence and also serves as the teacher model later during patch-based training. We conduct an experiment in which the initial pretraining step is omitted, and the model is trained on randomly sampled patches from the start of training. When trained with the same number of HR image patches, the model without global pretraining suffers in both structure and texture – FID for 1024px generated images is 26.78 and pFID is 8.64 – compared to our original model which performs global pretraining at low resolution – FID at 1024px is 4.50 and pFID is 3.46 after 10M training images.

### 2.7 Detectability

A concern with improved image generation is the potential for more convincing deceiving images, particularly those of higher resolution, which is the focus of our work. We use the off-the-shelf detector from Wang et al. [11] on our Birds (generated 256 → 2048) and Mountains (1024 → 4096) generators, across a large range of resolutions. As shown in Figure 10, the scores are well above chance (50%) across both datasets and resolutions. Interestingly, the curve generally trends upwards, indicating that while higher resolution images may look more natural, they are also easier to detect.

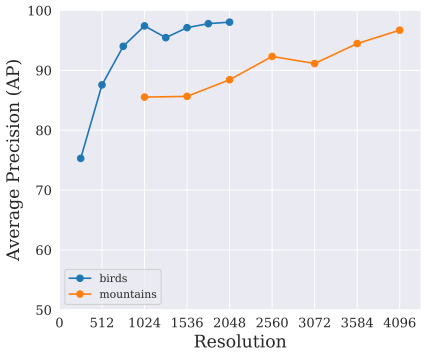


Fig. 10: Detection score from [11] on our Birds and Mountains datasets. All scores are above chance of 50%. Note in both cases, the detectability of our network trends upwards with resolution.

## 3 Additional implementation details

Building off the StyleGAN3 [5] architecture, we describe our coordinate conditioning and scale modulation branch applied to enable generation of multi-scale patches during the second training phase.

### 3.1 Patch-based training

**Extracting patches from varied size images** From our dataset of images $\mathcal{D}$, we sample an image $x_i \in \mathbb{R}^{H_i \times W_i \times 3} \sim \mathcal{D}$, with short-side $s_{\text{im}} = \min(H_i, W_i)$, and take an $s_{\text{im}}$-by-$s_{\text{im}}$ square crop. We then Lanczos downsample the image to an intermediate resolution $s \in [p, s_{\text{im}}]$, which provides "free" additional views from the same image, without introducing image corruptions.

Next, we sample a random crop of size $p$ and record the sampling location $v \in \mathbb{R}^2$. To summarize this procedure, we obtain a patch $x \in \mathbb{R}^{p \times p \times 3}$ from these two operations, while saving the sampled image resolution $s$ and patch center location $\mathbf{v} = (\mathbf{v}_x, \mathbf{v}_y) \in [0, 1]^2$ for later use.

$$x, s, \mathbf{v} = \text{Crop}(\text{Downsample}(x_i)) \tag{1}$$

**Synthesizing patches from the generator** Given a set of patches from the image dataset, the generator is tasked with synthesizing images at the corresponding patch locations. To transform the normalized coordinate domain $[0, 1] \times [0, 1]$ into patch coordinates, we apply a transformation matrix to each 2D location $c$ in homogenous coordinates:

$$c_{\mathbf{v},s} = T_{\text{patch}} * c = \begin{bmatrix} \frac{p}{s} & 0 & \mathbf{v}_x \\ 0 & \frac{p}{s} & \mathbf{v}_y \\ 0 & 0 & 1 \end{bmatrix} * c \tag{2}$$

Following StyleGAN3, these transformed coordinates are then encoded as $K$ random Fourier channels by multiplying by frequencies $B \in \mathbb{R}^{K \times 2}$ and adding phases $\phi \in \mathbb{R}^K$. For patch synthesis, the Fourier feature extraction at index $(h, w)$ becomes:

$$F_{h,w}(c_{\mathbf{v},s}) = \sin\left(2\pi\, B c_{\mathbf{v},s} + \phi\right) \in \mathbb{R}^K, \tag{3}$$

### 3.2 Scale-conditioning branch

As individual coordinate positions $c_{\mathbf{v},s}$ do not directly convey scale information, we found it beneficial additionally incorporate the scale input to intermediate layers of the generator. To do this, we first normalize the target scale $s$ to $[0, 1]$ using:

$$\bar{s} = \frac{s - p}{s_{\max} - p} \tag{4}$$

where $s_{\max}$ is selected from dataset statistics and is only present as a normalization factor, but does not clip the upper synthesis bound during inference. Empirically, we found that adding a small offset factor (we use 0.1) to the normalized target scale $\bar{s}$ allows for smoother interpolations between resolutions by avoiding a discontinuity at zero.

We then encode $\bar{s}$ using a parallel mapping network of identical architecture to the latent mapping network $M(z)$, and add the two inputs after undergoing

a layer-specific affine transformation into style-space [13,8] to obtain the final modulation parameter $M(z,s)_k$ at layer $k$:

$$M(z,s)_k = (W_{z,k} * M_z(z) + b_{z,k}) + (W_{s,k} * M_s(s) + b_{s,k}) \tag{5}$$

Because the modulation parameter is a multiplicative factor on the network weights and the scale-conditioning portion is added only during the secondary patch-wise training step, we initialize $b_{z,k} = \mathbf{1}$ and $b_{s,k} = \mathbf{0}$ to allow the network to smoothly transition between the initial pretraining step and secondary patch-based training.

### 3.3 Training procedure

We train our models on four to eight V100 GPUs with 16GB memory. By sampling fixed-size patches, the memory and compute footprint remain constant during training. For FFHQ, we finetune our initial fixed-scale generator from the pretrained FFHQ-U model [5], which reaches a minimum FID within 4M training images. In the remaining domains, we perform the pretraining step from scratch, retaining the checkpoint with the lowest FID, computed over 25M image samples, before continuing with the second, mixed-resolution training phase. Our training procedure is compatible with both 3×3 and 1×1 kernel sizes in StyleGAN3 (T & R configurations, respectively). For the patch-based training step, we proceed with the model configuration that reaches the best FID in pretraining, which is typically Config T with the exception of the FFHQ domain.

## References

1. Chong, M.J., Forsyth, D.: Effectively unbiased fid and inception score and where to find them. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 6070–6079 (2020) 4
2. Denton, E., Chintala, S., Szlam, A., Fergus, R.: Deep generative image models using a laplacian pyramid of adversarial networks. In: Adv. Neural Inform. Process. Syst. (2015) 6
3. Karnewar, A., Wang, O.: Msg-gan: Multi-scale gradients for generative adversarial networks. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 7799–7808 (2020) 6
4. Karras, T., Aila, T., Laine, S., Lehtinen, J.: Progressive growing of gans for improved quality, stability, and variation. In: Int. Conf. Learn. Represent. (2018) 6
5. Karras, T., Aittala, M., Laine, S., Härkönen, E., Hellsten, J., Lehtinen, J., Aila, T.: Alias-free generative adversarial networks. In: Adv. Neural Inform. Process. Syst. vol. 34 (2021) 12, 14
6. Lin, J., Zhang, R., Ganz, F., Han, S., Zhu, J.Y.: Anycost gans for interactive image synthesis and editing. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 14986–14996 (2021) 6
7. Park, T., Zhu, J.Y., Wang, O., Lu, J., Shechtman, E., Efros, A.A., Zhang, R.: Swapping autoencoder for deep image manipulation. In: Adv. Neural Inform. Process. Syst. (2020) 3

8. Patashnik, O., Wu, Z., Shechtman, E., Cohen-Or, D., Lischinski, D.: Styleclip: Text-driven manipulation of stylegan imagery. In: Int. Conf. Comput. Vis. pp. 2085–2094 (October 2021) 14
9. Sajjadi, M.S., Bachem, O., Lucic, M., Bousquet, O., Gelly, S.: Assessing generative models via precision and recall. In: Adv. Neural Inform. Process. Syst. vol. 31 (2018) 9
10. Thomee, B., Shamma, D.A., Friedland, G., Elizalde, B., Ni, K., Poland, D., Borth, D., Li, L.J.: Yfcc100m: The new data in multimedia research. Communications of the ACM **59**(2), 64–73 (2016) 3
11. Wang, S.Y., Wang, O., Zhang, R., Owens, A., Efros, A.A.: Cnn-generated images are surprisingly easy to spot... for now. In: IEEE Conf. Comput. Vis. Pattern Recog. (2020) 12
12. Wang, X., Li, Y., Zhang, H., Shan, Y.: Towards real-world blind face restoration with generative facial prior. In: IEEE Conf. Comput. Vis. Pattern Recog. (2021) 5
13. Wu, Z., Lischinski, D., Shechtman, E.: Stylespace analysis: Disentangled controls for stylegan image generation. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 12863–12872 (2021) 14
14. Yu, F., Zhang, Y., Song, S., Seff, A., Xiao, J.: Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. arXiv preprint arXiv:1506.03365 (2015) 3