Any-resolution Training for High-resolution Image Synthesis

Lucy Chai
MIT CSAIL
lrchai@mit.edu

Michael Gharbi
Adobe Research
mgharbi@adobe.com

Eli Shechtman
Adobe Research
elishe@adobe.com

Phillip Isola
MIT CSAIL
phillipi@mit.edu

Richard Zhang
Adobe Research
rizhang@adobe.com

Abstract

Generative models operate at fixed resolution, even though natural images come in a variety of sizes. As high-resolution details are downsampled away, and low-resolution images are discarded altogether, precious supervision is lost. We argue that every pixel matters and create datasets with variable-size images, collected at their native resolutions. Taking advantage of this data is challenging; high-resolution processing is costly, and current architectures can only process fixed-resolution data. We introduce *continuous-scale* training, a process that samples *patches* at random scales to train a new generator with variable output resolutions. First, conditioning the generator on a target scale allows us to generate higher resolutions images than previously possible, without adding layers to the model. Second, by conditioning on continuous coordinates, we can sample patches that still obey a consistent global layout, which also allows for scalable training at higher resolutions. Controlled FFHQ experiments show our method takes advantage of the multi-resolution training data better than discrete multi-scale approaches, achieving better FID scores and cleaner high-frequency details. We also train on other natural image domains including churches, mountains, and birds, and demonstrate arbitrary scale synthesis with both coherent global layouts and realistic local details, going beyond 2K resolution in our experiments. Our project page is available at: https://chail.github.io/anyres-gan/.

Keywords: Unconditional Image Synthesis, Generative Adversarial Networks, Continuous Coordinate Functions, Multi-scale Learning.

1 Introduction

The first step of typical generative modeling pipelines is to build a dataset with a fixed, target resolution. Images above the target resolution are downsampled, throwing away high-frequency details. Additionally, data of insufficient resolution is omitted, discarding structural information about low frequencies. Our simple insight is that this inconspicuous process wastes potentially learnable information. We propose to embrace the natural diversity of image sizes observed in-the-wild, processing them at their *native* resolution.

Relaxing the fixed-dataset assumption offers new, previously unexplored opportunities. One can potentially simultaneously learn global structure – for which large sets of readily-available low-resolution

Figure 1: Trained on a dataset of varied-size images, our unconditional generator learns to synthesize patches at continuous scales to match the distribution of real patches. Here, we render crops of the image at different resolutions, indicating the target resolution for each. We indicate the region of each crop in the top-left panel, which is the image directly sampled without scale input.

images suffice – and fine-scale details – where even a handful of high-resolution images may be adequate, especially given their internal recurrence [1]. This enables generating images at higher resolutions than previously possible, by adding in higher-resolution images to currently available fixed-size datasets.

This problem setting offers unique challenges, both in regards to modeling and scaling. First, one must generate images across multiple scales in order to compare with the target distribution. Naïvely downsampling the full-resolution image is suboptimal, as it is common to even have images of $16\times$ difference in scale in the dataset. Secondly, generating and processing high-resolution images offers scaling challenges. Modern training architectures at 1024 resolution already push the current hardware to the limits in memory and training time, and are unable to fully make use of images above that resolution.

To bypass these issues, we design a generator to synthesize image crops at arbitrary scales, hence, performing *any-resolution* training. We modify the state-of-the-art StyleGAN3 [2] architecture to take a grid of continuous coordinates, defined on a bounded domain, as well as a target scale, inspired by recent work in coordinate-based conditioning [3, 4, 5, 6]. By keeping the latent code constant and varying the crop coordinates and scale, our generator can output patches of the same image at various scales. This allows us to (1) efficiently generate at arbitrary scale, so that a discriminator critic can compare generations to a multi-resolution dataset, and (2) decouple high-resolution synthesis from increasing architecture size and prohibitive memory requirements.

We first experiment with our technique in a controlled setting on the FFHQ dataset [7], demonstrating efficient data usage by comparing our training using subsets of the dataset to using the entire full resolution dataset. We find minimal degradation (FIDs varying by 0.3), even at drastically skewed distributions – 98% of the dataset at $4\times$ lower resolution, and just 2% at a higher, mixed resolutions. Practically, this means we can leverage large-scale ($>$100k) lower-resolution datasets, such as LSUN Churches [8], Flickr Mountains [9], and Flickr Birds collected by us, and add a relatively small amount of high-resolution images ($\sim$ 6000), for continuous resolution synthesis beyond the 1024 resolution limit of current generators. To summarize, we:

- propose to use mixed-resolution datasets, embracing the natural distribution of images-in-the-wild.
- modify the generator to be amenable to such data, sampling patches at arbitrary scales during our *any-resolution* training procedure.
- demonstrate successful generations beyond 1024×1024, showing both fine details and coherent global structure, without the need for a larger and more expensive generator,
- introduce a variant of the FID metric that captures image statistics at multiple scales, thus accounting for the details of high resolutions.

2 Related Works

Unconditional image synthesis. Recent generative models including GANs [10, 11, 2, 12], Variational Autoencoders [13], diffusion models [14, 15, 16, 17, 18], and autoregressive models [19, 20, 21] such as transformers [22, 23, 24] are rapidly improving in quality. Of these, we focus on GANs, which offer state-of-the-art performance along with efficient inference and effective editing properties. A key area of innovation in GANs has been multi-resolution supervision during training. Works such as LapGAN [25], the Progressive/StyleGAN family [7, 11, 26, 2], MSG-GAN [27], and AnyCost-GAN [28] have demonstrated stable training by growing the generator with additional layers, progressively increasing the resolution by a factor two. Such a strategy even works for training single-image GANs [29, 30], based on the observation that images share statistics across scales. While several works [31, 32, 33] show data augmentations, such as small jitters in scale, can help stabilize training, they are processing the same, underlying fixed-resolution dataset. We draw upon the insights in these works for stable training, and seek to unlock training on an any-resolution dataset. Importantly, our generator does not use additional layers and can synthesize images at continuous scales, not only powers of two.

Coordinate-based functions. Coordinate-based encodings enable spatial conditioning and provide an inductive bias towards natural images [34]. Recent methods use point-based neural mappings to transform 2D or 3D coordinates to a color value for the purposes of unconditional generation [35, 2, 36, 5, 37], conditional generation [38], 3D view synthesis [3, 39, 40], or fitting arbitrary signals [6, 41]. By oversampling the coordinate grid, one can generate a larger image at inference time. However, because these models keep the same fixed-scale dataset assumption during training, the outputs struggle to offer additional high-frequency details without a high-frequency training signal. We draw upon the innovations in coordinate-based functions to sample patches at different scales and locations, enabling us to efficiently train on multiple scales. MS-PIE [42] and MS-PE [35] add positional encodings to allow for multi-scale synthesis, but they retain a global image discriminator designed to handle a single, fixed resolution of training images. Concurrently to our work, ScaleParty [43] also samples patches at arbitrary scales, but their goal is to generate at multiple scales while enforcing cross-scale consistency. In contrast, we focus on training with real images with arbitrary sizes which, in turn, lets us fully exploit the high quality of large images beyond standard datasets, and do so efficiently.

Extrapolation. One method of generating "infinite" resolution is extrapolating an image. Early texture synthesis works [44, 45, 46] focus on stationary textures. Recent advances [47] explore non-stationary textures, with large-scale structures and inhomogeneous patterns. Similar approaches operate by outpainting images, extending images beyond their boundaries in a conditional setting [48, 49, 50, 47]. Recent generative models synthesize large scenes [5, 51], typically casting synthesis as an outpainting problem [49, 50]. These methods are most effective for signals with a strong stationary component, such as landscapes, although extrapolation of structured scenes can be achieved in some domains [52].

Unlike textures, the images we wish to synthesize typically have a strong global structure. In a sense, we seek to extrapolate by "zooming in" or out, rather than "panning" beyond an image's boundaries.

Super-resolution. An alternative approach to generating high-resolution imagery would be to start with an off-the-shelf generative model and feed its outputs to a superresolution method [53, 6, 54, 55], possibly exploiting the self-similarity properties of images [56, 57, 58, 1]. Applying super-resolution models is challenging, in part because of the specific blur kernels super-resolution models are trained on [59]. Furthermore, though generations continue to improve, there remains a persistent domain shift between synthesized and real images [60, 61]. Finally, super-resolution is a local, conditional problem where the global structure is dictated by the low-resolution input, and optionally an additional high-resolution reference image [62, 63, 64, 65, 66]. We synthesize plausible images unconditionally, leveraging a *set* of high-resolution images to produce realistic global structure and coherent fine details.

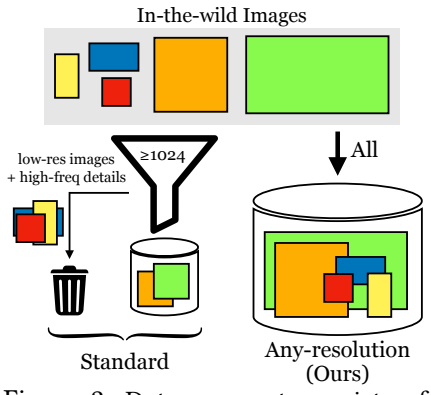

Figure 2: Data comes at a variety of scales. Traditional dataset construction filters out low-resolution images and downsamples high-resolution images to a fixed, training resolution. We aim to keep images at their original resolution.

3 Methods

In standard GAN training, all training images share a common fixed resolution, which matches the generator's output size. We seek to exploit the variety of image resolutions available in the wild, learning from pixels that are usually discarded, to enable high- and continuously-variable resolution synthesis. We achieve this by switching from the common fixed-resolution thinking, to a novel 'any-resolution' approach, where the original size of each training image is preserved (Fig 2). We introduce a new class of GAN generators that learn from this multi-resolution signal to synthesize images at any resolution (§ 3.1), and show how to train them by sampling patches at multiple scales to jointly supervise the global-structure and fine image details (§ 3.2).

3.1 Multi-resolution GAN

We design our approach to leverage state-of-the-art GANs. We keep the architecture of the discriminator largely unchanged. Because most discriminators operate at a fixed resolution, we modify the generator so it can synthesize images at any resolution, and receive the discriminator's fixed-resolution supervision. Our implementation builds on the recent StyleGAN3 framework [2], which is conditioned on a fixed coordinate grid. We modify this grid to handle any-resolution and patch-based synthesis..

Continuous-resolution generator. We treat each image as a continuous function defined on a bounded normalized coordinate domain $[0, 1] \times [0, 1]$. The generator G always generates patches at a fixed pixel resolution $p \times p$, but each patch implicitly corresponds to a square sub-region, centered at $\mathbf{v} \in [0, 1]^2$, of the larger image. Denoting the resolution of the larger image as $s \times s$, we have that the patch size is p/s in normalized coordinates (see Figure 3, left). During training, we sample patches from images at multiple scales s, either from the generator or from the multi-resolution dataset, before passing them to the fixed-resolution discriminator D. Formally, our generator takes three inputs: a regular grid of normalized continuous pixel coordinates $c_{\mathbf{v},s} \in \mathbb{R}^{p \times p \times 2}$, the resolution $s \in \mathbb{N}$ of the (implicit) larger image the patch is extracted from, and z, the latent code representing this larger image.

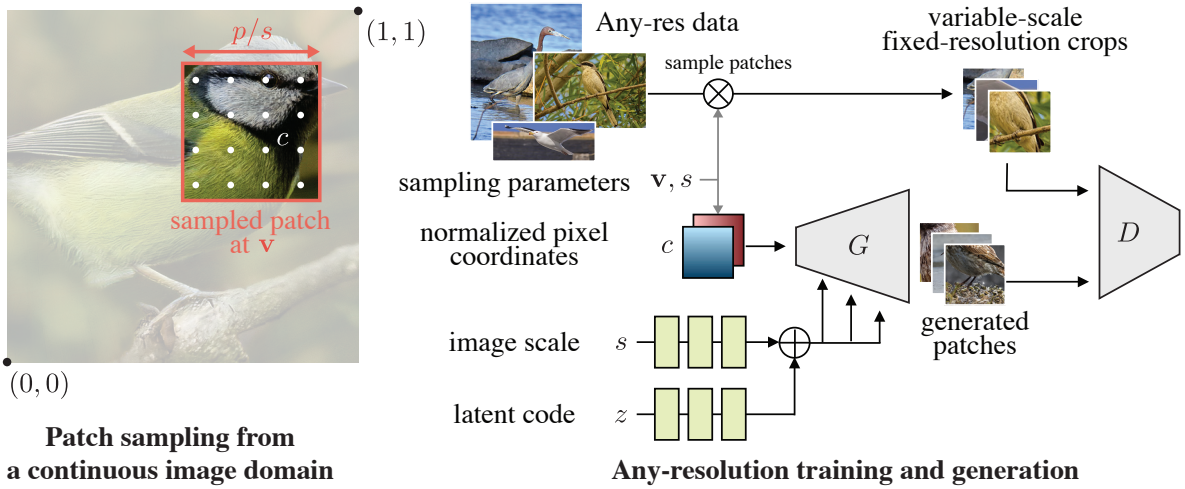

Figure 3: Overview. (Left) We parameterize images (real or synthetic) as continuous functions over a normalized domain from which we extract random patches at various scales s, but constant resolution p. (Right) To train, we sample crops at random scales and offsets $\mathbf{v}$ from the full-size images in our dataset. These same crops are sampled from the generator, by passing it a grid of the desired sample coordinate $c_{\mathbf{v},s}$, and injecting the image scale s through modulation, in addition to the latent code characterizing the complete image z.

It synthesizes the patch's pixel values at the sampled coordinates as:

$$G(z, c_{\mathbf{v},s}, s) = G(F(c_{\mathbf{v},s}); M(z, s)), \tag{3.1}$$

where F is a Fourier embedding of the continuous coordinates [2], and M is an auxiliary function that maps the latent code and sampling resolution into a set of modulation parameters for the StyleGAN3 generator (see § 3.3 for details). Our method therefore modifies two components from StyleGAN3. First, we replace the fixed coordinate grid with *patch-dependent* coordinates to train on variable-resolution images. Second, we append an auxiliary branch M to inject scale information throughout the generator.

At test time, we can generate images at arbitrarily high resolutions by sampling the full continuous domain $[0, 1] \times [0, 1]$ at the desired sampling rate. Theoretically, the maximum resolution is infinite, but in practice the amount of detail that the model can generate is determined by generator resolution p and the resolutions of the training images.

3.2 Two-phase training

We train our generator in two phases. In the first, we want the generator to learn to generate globally-coherent images. For this, we disable the patch sampling mechanism and pretrain the generator at a fixed scale, corresponding to the full continuous image domain. That is, we fix $s = p$, and $\mathbf{v} = (0.5, 0.5)$, which is equivalent to standard fixed-resolution GAN training. Both the coordinate tensor $c_{\mathbf{v},s}$ and the scale conditioning variable s are constant in this phase, so we simply refer to the image generated as $G_{\text{fixed}}(z)$ and follow the training procedure of StyleGAN3. In the second phase, we enable patch sampling for both the real and synthetic images and continue training the generator using variable-scale patches, so it learns to synthesize fine details at any resolution. We found that using a copy of the pretrained fixed-scale generator G_{fixed} as a teacher model helps stabilize training in this phase.

Global fixed-resolution pretraining. During the pretraing phase, we effectively resample all the training images to a fixed resolution $p \times p$, as in standard GAN training. Let $x \sim \mathcal{D}_{\text{fixed}}$ denote an image sampled from this fixed size dataset. We optimize a standard GAN objective with non-saturating logistic

loss and R_1 regularization on the discriminator:

$$V(D, G(z), x) = D(x) - D(G(z)), \quad R_1(D, x) = ||\nabla D(x)||^2,$$
$$G_{\text{fixed}} = \arg\min_G \max_D \mathbb{E}_{z,x\sim\mathcal{D}_{\text{fixed}}} V(D, G(z), x) - \frac{\lambda_{R_1}}{2} R_1(D, x). \tag{3.2}$$

We follow the recommended values for λ_{R_1}, depending on the the generator resolution p [2].

Mixed-resolution patch-based training. In the second phase, we enable multi-resolution sampling, alternating between extracting random crops from our any-resolution dataset and generating them with our continuous generator.

For synthetic patches, we sample a patch location $\mathbf{v}$ uniformly in the continuous domain $[0, 1] \times [0, 1]$; and an arbitrary image resolution $s \geq p$, corresponding to the implicit full image around the square patch. From those, we derive the sampling coordinate grid $c_{\mathbf{v},s}$, and synthesize the patch image $G(z, c_{\mathbf{v},s}, s)$, as described earlier.

For 'real' patches, we sample an image from our dataset. Because this image can have any resolution $s_{\text{im}} \geq p$, we crop it to a random square matching its smallest dimension, then Lanczos downsample this square to a random resolution $s \times s$ with $s_{\text{im}} \geq s \geq p$. Finally, we extract a random $p \times p$ crop from the downsampled image, recording its center $\mathbf{v}$. To preserve the generator's global coherence and continuous generation ability, we sample at the global scale $s = p$, $\mathbf{v} = (0.5, 0.5)$ with probability 50%. We found that image quality at the global resolution p degrades otherwise, and we refer to images generated at global resolution p as the "base image". Our any-resolution GAN optimizes the following objective during this phase:

$$G^* = \arg\min_G \min_D \mathbb{E}_{z,\{x,s,\mathbf{v}\}\sim\mathcal{D}} V(D, G(z, c_{\mathbf{v},s}, s), x) \\ + \lambda_{\text{teacher}} \mathcal{L}_{\text{teacher}}(G, G_{\text{fixed}}, z) - \frac{\lambda_{R_1}}{2} R_1(D, x). \tag{3.3}$$

We use $\lambda_{\text{teacher}} = 5$; other values offer slight tradeoffs between similarity to the base teacher model G_{fixed}, and FID score (see supplemental). $\mathcal{L}_{\text{teacher}}$ is an auxiliary loss to encourage faithfulness to the pretrained fixed-resolution generator G_{fixed}. The architecture of D remains the same as in the pretraining step; we found that modifying the discriminator setup did not further improve results (see supplemental).

Teacher model. For the second training phase above, we initialize G with the pretrained weights of G_{fixed}. Weights for discriminator D are also kept for fine-tuning. We keep a separate copy of G_{fixed} with frozen weights, the teacher, for additional supervision. We design a loss function that encourages the generated patch (at any resolution), to match the teacher's fixed-resolution output in the region corresponding to the patch, after downsampling and proper alignment. Formally, this loss is given by:

$$\mathcal{L}_{\text{teacher}}(G, G_{\text{fixed}}, z) = d\left(m \odot w_{\mathbf{v},s}(G(z, c_{\mathbf{v},s}, s)),\ m \odot G_{\text{fixed}}(z)\right), \tag{3.4}$$

where d is the sum of a pixel-wise ℓ_1 loss, and the LPIPS perceptual distance [67, 68]. The warp function $w_{\mathbf{v},s}$ transforms and resamples the generated high-resolution patch using a band-limited Lanczos kernel, to project it in the coordinate frames of the low-resolution, global image $G_{\text{fixed}}(z)$. Because the warped patch does not cover the entire image domain, we multiply it with a binary mask m to indicate the valid pixels, prior to computing loss d.

3.3 Implementation details

Scale-conditioning. In addition to the pixel location $c_{\mathbf{v},s}$, we also pass the global resolution information s to the generator. Knowledge of the global image scale is important to enable continuous scale

Table 1: Any-resolution datasets and generator settings. We build upon low-resolution (LR) datasets, and use it for fixed-size dataset pre-training. We add additional high-resolution (HR) images, of mixed resolutions. Note that the number of HR images is small (∼2-7% of LR size). Patches of size p are sampled from both subsets during training, with average sampled scale $\mathbb{E}[s]$.

Domain	**Dataset**							**Generator**		
	Source	# Imgs		Resolutions				Config	Resolutions	
		LR	HR	LR	HR_{min}	HR_{med}	HR_{max}		p	$\mathbb{E}[s]$
Faces	FFHQ	70,000	6000	256	512	819	1024	R	256	458
Churches	LSUN & Flickr	126,227	6253	256	1024	2836	18,000	T	256	1061
Birds	Flickr	112,266	7625	256	512	1365	2048	T	256	585
Mountains	Flickr	507,495	9361	1024	2049	3168	12,431	T	1024	1823

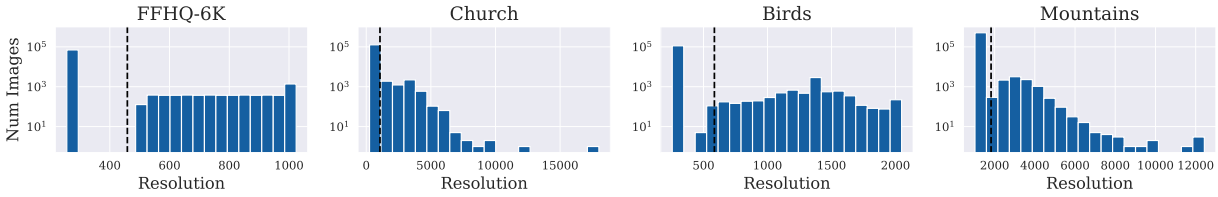

Figure 4: Training set size distributions. Y-axis is in log scale as most images are LR. The dotted line indicates the average sampled resolution during training $\mathbb{E}[s]$.

variations and proper anti-aliasing [6, 4]. We found it beneficial to explicitly inject this information into all intermediate layers of the generator. To achieve this, we use a dual modulation approach [69], embedding the latent code z and scale s separately using two independent sub-networks (we use the same mapping network architecture for each). The two outputs are summed to obtain a set of modulation parameters $M(z, s)$, used to modulate the main generator features. We preprocess s by using an affine remapping from the minimum and maximum resolution to $[-1, 1]$. Architectural details of the generator G and mapping network M, can be found in the supplemental.

Synthesizing large images. Our fully-convolutional generator can render image at arbitrary resolutions. But images larger than 1024×1024 require significant GPU memory. Equivalently, we can render non-overlapping tiles that we assemble into a larger image. Our patch-based multi-resolution training and the Fourier encoding of the spatial coordinates make the tile junctions seamless.

4 Experiments

We introduce a modified image quality metric that computes FID over multi-scale image patches without downsampling, which is largely correlated to the standard FID metric when ground truth high-resolution images are available (FFHQ), yet more sensitive to the quality of larger resolutions. We then compare our model to alternative approaches for variable scale synthesis, and super-resolution on other natural image domains (§ 4.1). Finally, we investigate variations of our model and training procedure to validate our design decisions (§ 4.2).

Data. Our method is general and can work on collections of any-resolution data. As such, when targeting high-resolution generation, rather than starting over, we can add additional high-resolution (HR) images to existing, fixed-size, low-resolution (LR) datasets. Our datasets and their statistics are listed in Tab. 1. Figure 4 shows the resolution distrbution in each dataset.

We begin initial experiments with a controlled setting of FFHQ, which contains 70K images at 1024 resolution. From these, we construct a varied-size dataset by (1) using 256 resolution for all images (2) downsampling a 5K subset between 512-1024 (uniformly distributed) and (3) add 1k subset at full 1024. The last step enables us to judiciously compare to methods that are limited to synthesizing images at strict powers of 2. We refer to this mixture as FFHQ6k. We use the full 1024 dataset as ground-truth for evaluation metrics.

In the remaining domains, we push current generation results to higher resolution by scraping HR images from Flickr. In cases where a standard fixed-size dataset is available (LSUN Churches [8] and Mountains [9]), we select the additional HR images to approximately match the LR domain. Our final generators synthesize realistic details despite the majority of the training set being LR. For Birds and Churches, $>$ 93% of the training set is at 256 resolution but our model maintains quality beyond 1024; for Mountains $>$ 98% of training set at 1024 but our model can generate beyond 2048. These categories cover a range between objects (but without the strong alignment of FFHQ) and outdoor scenes.

4.1 Continuous Multi-Scale Image Synthesis.

Qualitative examples. We show qualitative examples in Fig. 5. Our generated images preserve the fine details of HR structures, such as bricks, rocky slopes, feathers, or hair. Pushing the inference resolution towards and beyond the higher resolutions of training images, we find that textured surfaces typically deteriorate first, but our model preserves sharp edge boundaries when extrapolating (Fig. 6).

Patch-FID metric. In our setting, the standard FID pipeline [70] has some limitations: (1) the number of training images at high resolution is small (less than 10K images) which tends to bias the resulting FID estimate, (2) the high-resolution images have different sizes, while standard FID assumes all real images are the same size, and (3) FID downsamples images to 299, which itself can lead to artifacts [71] and ignores high-resolution details. Therefore, we propose a modification to the standard FID, which we dub 'patch-FID', that splits the HR dataset into 50K patches at random scales and locations, and computes FID on the real patches and generated patches with corresponding scales and locations. Crucially, this avoids downsampling the generated content. Our patch-FID is more sensitive to blurriness or artifacts at higher resolutions, resulting in larger absolute difference compared to standard FID at 1024 resolution, but it is largely correlated to standard FID when a full HR ground-truth is available (as in the FFHQ dataset, see Table 2). We abbreviate the standard FID metric computed at a given resolution as FID@res, and our patch-FID metric as pFID (lower numbers are better in both cases).

Alternative methods of multi-size training and generation. Our generator is encouraged to synthesize realistic high-resolution textures at training, even when the discriminator does not get to see the full image. While MS-PE [35] also enables continuous resolution synthesis, the discriminator learns only at a single resolution and the generator is not trained patch-wise. We find that this downsampling for the discriminator is detrimental to image quality at higher resolutions. Anycost-GAN[28] performs image synthesis at powers-of-two resolutions by adding additional synthesis blocks. For comparison purposes, we modify it to handle a multi-size dataset by downsampling images to the nearest power of two and training each layer only on the valid image subset. Compared to Anycost-GAN, our model is more data-efficient, due to weight sharing for generation at multiple scales. Anycost-GAN requires learning a separate module for each increase in resolution, thus resulting in artifacts at higher resolutions when fewer HR training images are available and higher FID scores (Tab. 2). Additionally, Anycost-GAN increases the generator and discriminator size for synthesis at higher resolutions, whereas our model incurs a constant training cost, regardless of the inference scale.

Comparison to super-resolution. Most super-resolution methods require LR/HR image pairs, whereas there is no ground-truth HR counterpart to a LR image synthesized by our fixed-scale generator G_{fixed}.

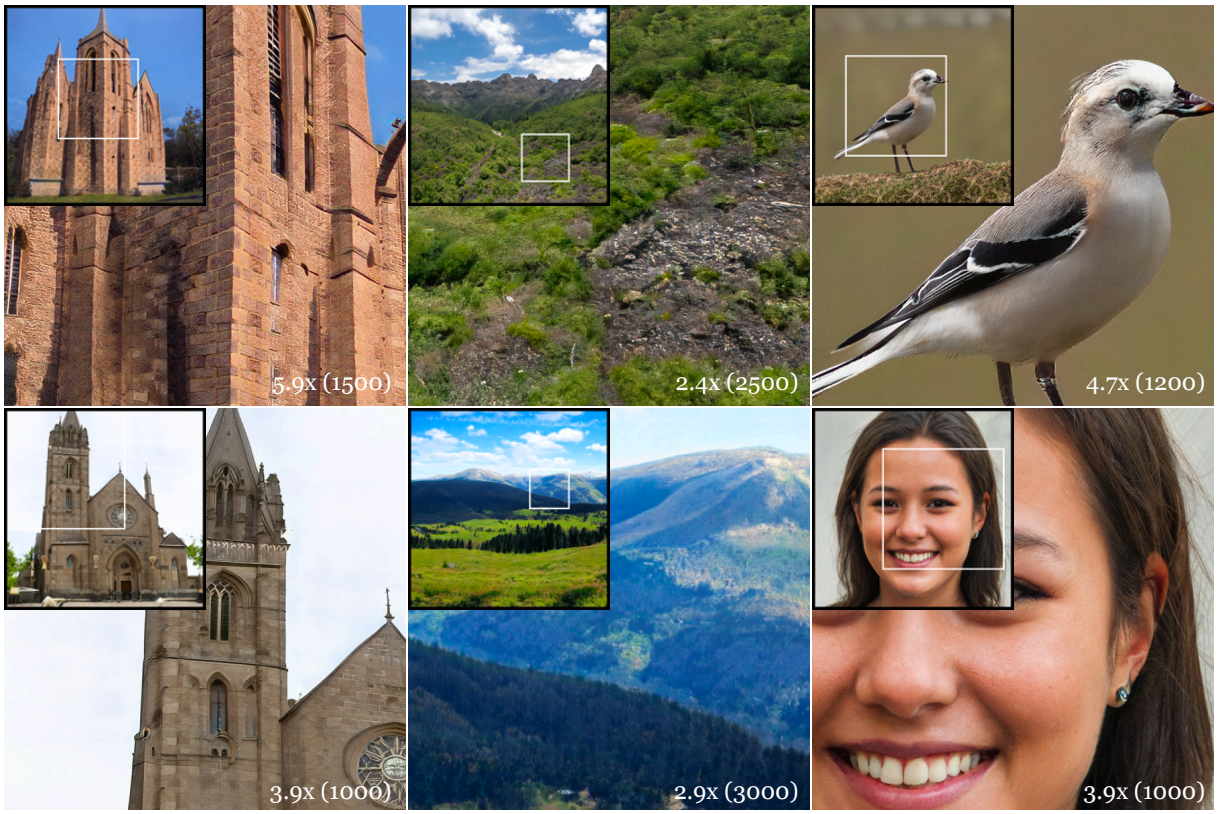

Figure 5: The inset shows the entire generated, high-resolution images (between 1000-3000 resolution), with enlarged regions of interest outlined in the white box. Note that our model can render the image (or any sub-region) at any resolution.

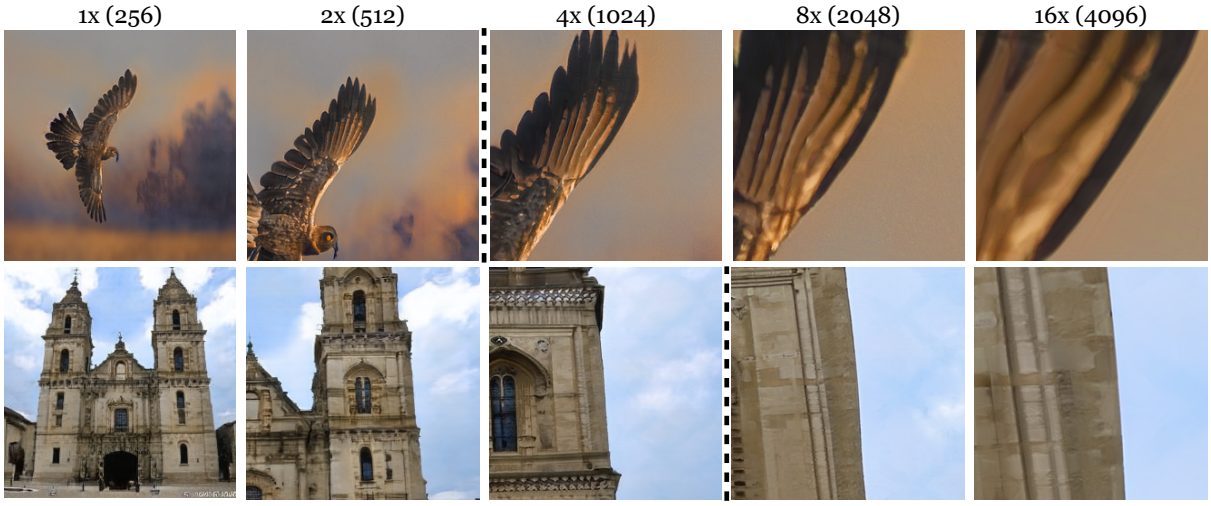

Figure 6: Extrapolation limits. We test the extrapolation capabilities of our model by specifying the inference scale s. Typically, textures such as bricks and feathers deteriorate first, but the model preserve edge boundaries well. The dotted line indicates when generation starts to exceed the average scale sampled in training $\mathbb{E}[s]$ (which is 585 for birds and 1061 for churches).

Table 2: Varied-size training and inference. Random-resize MS-PE [35] performs varied-size synthesis, but assumes a fixed-size dataset. AnyCost-GAN handles varied training at powers of 2. Our method directly utilizes training images at any size, achieving better results by FID. († = copied from paper)

FFHQ6K	FID@256	@512	@1024	pFID
MS-PE [35]†	6.75	30.41	–	–
Anycost [28]	4.24	5.94	6.47	18.39
Ours	3.27	3.92	3.95	2.94

Table 3: Comparison to super-resolution using patch-FID (pFID). For each domain, we compare our model to continuous-scale (LIIF) and fixed-scale super-resolution (Real-ESRGAN) models. Lower pFID suggests that our model can generate realistic details at high resolutions, not achievable with super-resolution alone.

	FFHQ6K	Church	Bird	Mountain
LIIF [6]	22.93	83.88	30.19	23.10
Real-ESRGAN [55]	16.92	23.04	16.10	19.05
Ours	2.94	12.18	6.81	8.76

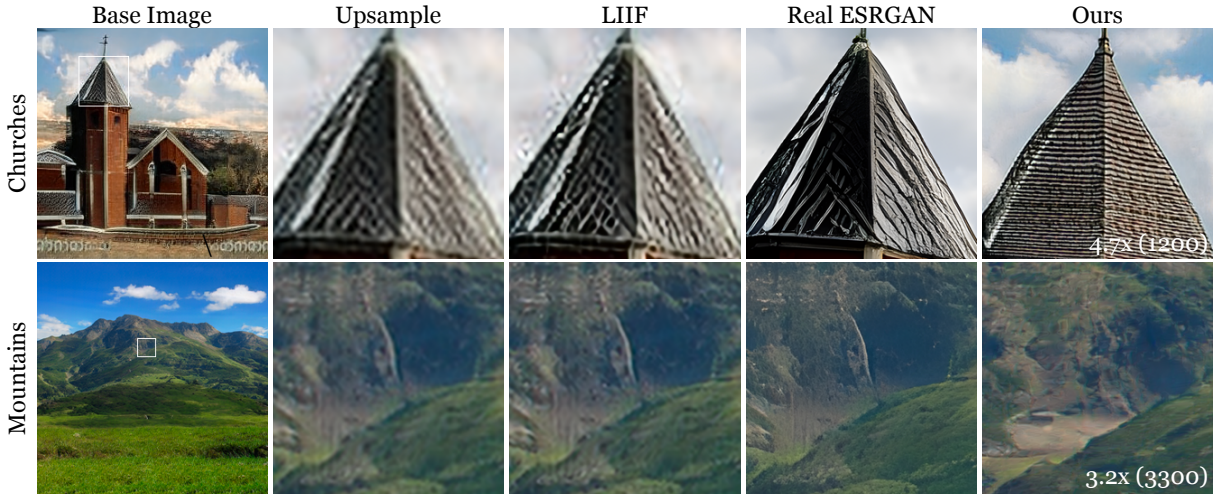

Figure 7: Super-resolution Comparisons. Qualitative comparisons of Lanczos upsampling a patch from the base image (upsample), continuous (LIIF [6]) and fixed-factor (Real ESRGAN [55]) super-resolution models, and our trained model. LIIF tends to amplify artifacts from the base image (e.g. the JPEG artifacts around the church). While Real-ESRGAN is better at suppressing artifacts, it tends to overly smooth surfaces or synthesize grid-like textures (mountain). Our model is not a super-resolution model; it can add additional details to the low-resolution image but tolerates slight distortions in structure which are regularized with the teacher weight.

The teacher regularization encourages similarity between G_{fixed} and G's outputs, but unlike super-resolution, this supervision occurs at low-resolution, which allows for variations in the fine details. Figure 7 compares our model to super-resolution methods applied to the output of G_{fixed}. Our method produces much sharper details than LIIF [6], a recent continuous-scale super-resolution technique, and cleaner images than the state-of-the-art Real-ESRGAN [55]. The latter is a fixed-resolution model, so we run it iteratively until exceeding a target resolution, and then Lanczos downsample the result to the target size. Real-ESRGAN's outputs are either overly smooth, or exhibit grid-like artifacts. Our method generates realistic textures, consistently improves upon the low-resolution output of G_{fixed}, and reaches a lower pFID (Tab. 3).

4.2 Model Variations

Using the full high-resolution FFHQ dataset as a benchmark, we investigate individual components of our architecture and training process. We train each model variation for 5M images and record metrics from the best FID@1024 checkpoint. We only report quantitative metrics in the main paper and refer to the supplemental for further evaluations and visual comparisons.

Impact of teacher regularization. Our full model uses an "inverse" teacher regularizer to encourages

Table 4: Multisize Training. Downsampling or upsampling all images to a common size, or using only the subset of the largest images, worsens FID compared to our training strategy.

	FID@256	@512	@1024	pFID
Resize down to 512	3.31	4.11	19.18	26.83
Resize up to 1024	3.46	13.43	4.86	6.65
Train 1024 subset	3.46	12.41	4.67	5.43
Multisize training	3.37	4.41	4.47	4.28

Table 5: Number of HR images. Our method is robust to a wide range of HR images, even when only 1K images at HR are available (<2% of the full ground-truth dataset).

	FID@256	@512	@1024	pFID
1k (1.4%)	3.43	5.13	4.38	3.73
5k (7.1%)	3.36	4.97	4.54	3.48
10k (14.2%)	3.46	4.96	4.65	3.54
70k (100%)	3.42	4.88	4.52	3.42

a downsampled HR patch to match the low-resolution teacher as described in 3.2. We also explored a variant with a "forward" teacher loss, in which the generated patch is encouraged to match the *upsampled* teacher output. This variation is qualitatively inferior and blurs details; it has worse FID at higher resolutions (see supplemental for details and visuals). Removing the teacher altogether improves pFID but degrades FID. Qualitatively the generated patches diverge significantly from the fixed-size global image. We hypothesize that the global change in structure negatively impacts overall image quality, causing global FID metrics to increase, but this cannot be captured from evaluating patches alone. We found $\lambda_{\text{teacher}} = 5$ to provide the best balance between global and local image quality, but we observe minimal differences in FID and pFID for other values, evidence that the model can tolerate a range of values for this parameter. See supplemental for a parameter sweep with full scores.

Removing scale conditioning degrades quality. We inject the scale information to intermediate layers of the generator through scale-conditioning. Adding this improves FID@1024 from 4.88 to 4.47, and pFID from 4.67 to 4.28.

Multi-size training improves fidelity at all scales. Our multi-size data pipeline lets our model learn to synthesize at continuous scales, which is a strictly more challenging than learning at a fixed scale. In Table 4, we investigate to what extent learning from images of varied sizes offers benefits over fixed-scale training on a smaller dataset. Visual comparisons can be found in supplemental. In a first alternative, using the same FFHQ-6K dataset, we resize all images down to 512 and train the model to generate patches at 512×512. In this case, the model performs well up to 512 scale, but does not generalize beyond (e.g., 1024) since it cannot exploit the information lost in downsampling. In two other variants, we train models for 1024 resolution in the first case by upsampling all images up to 1024, in the second by keeping only the 1K subset of images at 1024 resolution. Both variants are trained to output images specifically at 1024 resolution. The former approach (upsampling) increases blurriness. In the latter, FID@1024 remains worse than that of our multi-size training, which can take advantage of more data despite most of it being *smaller* than 1024.

Impact of number of HR images Due to the patch-based training procedure, we find that our model can be trained with a small fraction of HR images, compared to the 70k LR images in the dataset. In Table 5, we use progressively larger subsets of HR images: 1k, 5k, 10k. We found that the FID scores are largely similar (within 0.3) to using the entire 70k HR images. However, we observed that training with 1k or fewer HR images shows evidence of divergence during training (see supplemental). The training stabilizes by 5k HR images, So, for the remaining domains, we collect roughly 5K-10K images to construct the HR dataset.

4.3 Properties of Multi-Scale Generation

Correcting artifacts from low resolution. Because our model is not directly trained with corresponding LR and HR image pairs, we find that there can be small distortions between the upsampled base image and the HR generation from the same latent code. In some cases, this can be a desirable property (Fig. 8). For instance, the base generator on the birds dataset can struggle in synthesizing the eye of the

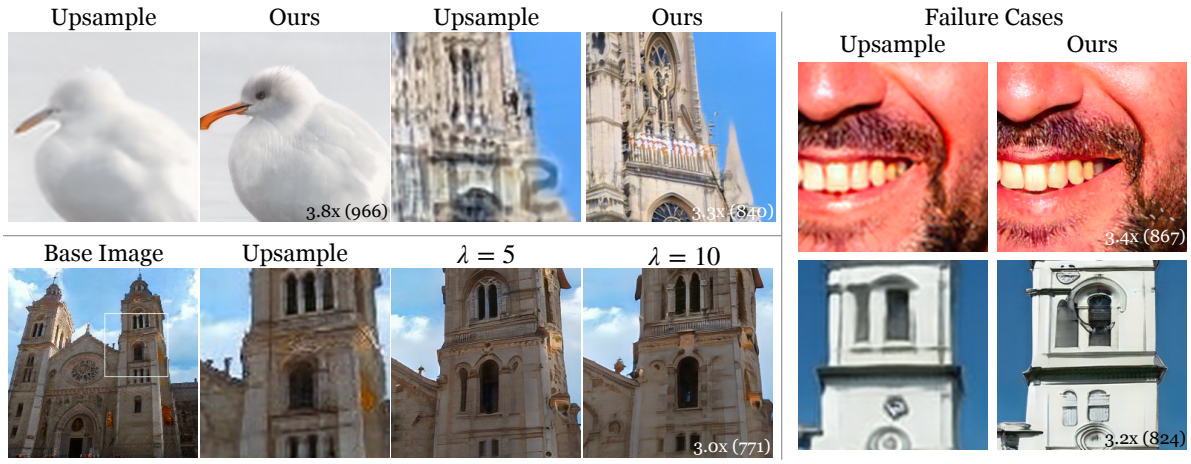

Figure 8: Model Properties and Failure Cases. As fine details can be more difficult to learn at low resolution, our model is capable of adding corrections when generating at higher resolutions. In the case of inconsistencies between the LR and HR data sources, the model deletes patterns that are not present in the HR dataset (*e.g.* watermarks). Both these properties rely on the fact that the models tolerates some distortions between different target scales, which can be adjusted by changing the teacher regularization weight. Failure cases include biases towards circular or ring-like structures, which may occur in beard patterns or around church windows.

bird, which is less apparent at low resolutions, but more salient at high resolution. Consequently, our HR generation will add the missing eye, and also synthesizes additional feather and beak details. In the churches domain, because the LR and HR datasets are collected separately, we find that the synthesized watermarks and JPEG artifacts at the base resolution disappear at higher resolution, because the HR dataset we used is of higher quality and does not have any watermark. The similarity between the LR and HR generations can be tuned using λ_{teacher} during training.

Failure Cases. Our model tends to inherit the artifacts from StyleGAN3, such as the generation of a centered front tooth in the FFHQ domain. In instances in which the generated image at base resolution contains uneven surfaces, the model may fail to fully mitigate them at higher resolutions. These artifacts are often subtle at the low resolution, but become more apparent when upsampling the base image or generating at a larger target scale. In some cases, our model also has a tendency to generate "watery" circular or ring-like artifacts (Fig. 8).

5 Conclusion

We propose an image synthesis approach that can train on images of varied resolution and perform inference at continuous resolutions. This lifts the fixed-resolution requirement of prior generative models, which discard important high-resolution details. To do this, we train a generator jointly on a low-resolution dataset to learn global structure, and on patches from the varied-size dataset to learn details. At inference time, we can synthesize an image at any resolution by supplying the appropriate coordinate grid and scale factor to the generator. We find that, by using training images at their native resolutions and a single model for continuous-resolution synthesis, our method can efficiently leverage information present in only a handful of high-resolution images to complement a large set of low-resolution images. Unlike previous works, this approach enables high-resolution synthesis without requiring a larger generator for training or large dataset of fixed-size, high-resolution images.

Acknowledgements. We thank Taesung Park for help in dataset collection. LC is supported by the National Science Foundation Graduate Research Fellowship under Grant No. 1745302 and Adobe Research Fellowship. This work was started while LC was an intern at Adobe Research.

References

[1] Assaf Shocher, Nadav Cohen, and Michal Irani. "zero-shot" super-resolution using deep internal learning. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 3118–3126, 2018.

[2] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks. In *Adv. Neural Inform. Process. Syst.*, volume 34, 2021.

[3] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *Eur. Conf. Comput. Vis.*, pages 405–421. Springer, 2020.

[4] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *Int. Conf. Comput. Vis.*, pages 5855–5864, 2021.

[5] Chieh Hubert Lin, Hsin-Ying Lee, Yen-Chi Cheng, Sergey Tulyakov, and Ming-Hsuan Yang. Infinitygan: Towards infinite-resolution image synthesis. *Int. Conf. Learn. Represent.*, 2021.

[6] Yinbo Chen, Sifei Liu, and Xiaolong Wang. Learning continuous image representation with local implicit image function. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 8628–8638, 2021.

[7] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *Int. Conf. Learn. Represent.*, 2018.

[8] Fisher Yu, Yinda Zhang, Shuran Song, Ari Seff, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015.

[9] Taesung Park, Jun-Yan Zhu, Oliver Wang, Jingwan Lu, Eli Shechtman, Alexei A. Efros, and Richard Zhang. Swapping autoencoder for deep image manipulation. In *Adv. Neural Inform. Process. Syst.*, 2020.

[10] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *Int. Conf. Learn. Represent.*, 2015.

[11] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019.

[12] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. In *Int. Conf. Learn. Represent.*, 2018.

[13] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[14] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *Int. Conf. Machine Learning*, pages 8162–8171. PMLR, 2021.

[15] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Adv. Neural Inform. Process. Syst.*, volume 33, pages 6840–6851, 2020.

[16] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *Int. Conf. Learn. Represent.*, 2021.

[17] Yang Song and Stefano Ermon. Improved techniques for training score-based generative models. In *Adv. Neural Inform. Process. Syst.*, volume 33, pages 12438–12448, 2020.

[18] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. volume 34, 2021.

[19] Hugo Larochelle and Iain Murray. The neural autoregressive distribution estimator. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 29–37. JMLR Workshop and Conference Proceedings, 2011.

[20] Aaron Van Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. In *Int. Conf. Machine Learning*, pages 1747–1756. PMLR, 2016.

[21] Aaron Van den Oord, Nal Kalchbrenner, Lasse Espeholt, Oriol Vinyals, Alex Graves, et al. Conditional image generation with pixelcnn decoders. In *Adv. Neural Inform. Process. Syst.*, volume 29, 2016.

[22] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Adv. Neural Inform. Process. Syst.*, volume 30, 2017.

[23] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In *Int. Conf. Machine Learning*, pages 1691–1703. PMLR, 2020.

[24] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 12873–12883, 2021.

[25] Emily Denton, Soumith Chintala, Arthur Szlam, and Rob Fergus. Deep generative image models using a laplacian pyramid of adversarial networks. In *Adv. Neural Inform. Process. Syst.*, 2015.

[26] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020.

[27] Animesh Karnewar and Oliver Wang. Msg-gan: Multi-scale gradients for generative adversarial networks. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 7799–7808, 2020.

[28] Ji Lin, Richard Zhang, Frieder Ganz, Song Han, and Jun-Yan Zhu. Anycost gans for interactive image synthesis and editing. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 14986–14996, 2021.

[29] Tamar Rott Shaham, Tali Dekel, and Tomer Michaeli. Singan: Learning a generative model from a single natural image. In *Int. Conf. Comput. Vis.*, pages 4570–4580, 2019.

[30] Assaf Shocher, Shai Bagon, Phillip Isola, and Michal Irani. Ingan: Capturing and retargeting the "dna" of a natural image. In *Int. Conf. Comput. Vis.*, pages 4492–4501, 2019.

[31] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. In *Adv. Neural Inform. Process. Syst.*, 2020.

[32] Zhengli Zhao, Zizhao Zhang, Ting Chen, Sameer Singh, and Han Zhang. Image augmentations for gan training. *arXiv preprint arXiv:2006.02595*, 2020.

[33] Shengyu Zhao, Zhijian Liu, Ji Lin, Jun-Yan Zhu, and Song Han. Differentiable augmentation for data-efficient gan training. In *Adv. Neural Inform. Process. Syst.*, volume 33, pages 7559–7570, 2020.

[34] Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. In *Adv. Neural Inform. Process. Syst.*, volume 33, pages 7537–7547, 2020.

[35] Jooyoung Choi, Jungbeom Lee, Yonghyun Jeong, and Sungroh Yoon. Toward spatially unbiased generative models. In *Int. Conf. Comput. Vis.*, 2021.

[36] Ivan Skorokhodov, Savva Ignatyev, and Mohamed Elhoseiny. Adversarial generation of continuous images. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 10753–10764, 2021.

[37] Ivan Anokhin, Kirill Demochkin, Taras Khakhulin, Gleb Sterkin, Victor Lempitsky, and Denis Korzhenkov. Image generators with conditionally-independent pixel synthesis. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 14278–14287, 2021.

[38] Tamar Rott Shaham, Michaël Gharbi, Richard Zhang, Eli Shechtman, and Tomer Michaeli. Spatially-adaptive pixelwise networks for fast image translation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 14882–14891, 2021.

[39] Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. Graf: Generative radiance fields for 3d-aware image synthesis. In *Adv. Neural Inform. Process. Syst.*, volume 33, pages 20154–20166, 2020.

[40] Eric R Chan, Marco Monteiro, Petr Kellnhofer, Jiajun Wu, and Gordon Wetzstein. pi-gan: Periodic implicit generative adversarial networks for 3d-aware image synthesis. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 5799–5809, 2021.

[41] Ishit Mehta, Michaël Gharbi, Connelly Barnes, Eli Shechtman, Ravi Ramamoorthi, and Manmohan Chandraker. Modulated periodic activations for generalizable local functional representations. In *Int. Conf. Comput. Vis.*, pages 14214–14223, 2021.

[42] Rui Xu, Xintao Wang, Kai Chen, Bolei Zhou, and Chen Change Loy. Positional encoding as spatial inductive bias in gans. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 13569–13578, 2021.

[43] Evangelos Ntavelis, Mohamad Shahbazi, Iason Kastanis, Radu Timofte, Martin Danelljan, and Luc Van Gool. Arbitrary-scale image synthesis. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022.

[44] Alexei A Efros and Thomas K Leung. Texture synthesis by non-parametric sampling. In *Int. Conf. Comput. Vis.*, volume 2, pages 1033–1038. IEEE, 1999.

[45] Alexei A Efros and William T Freeman. Image quilting for texture synthesis and transfer. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 341–346, 2001.

[46] Yonatan Wexler, Eli Shechtman, and Michal Irani. Space-time completion of video. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(3):463–476, 2007.

[47] Yang Zhou, Zhen Zhu, Xiang Bai, Dani Lischinski, Daniel Cohen-Or, and Hui Huang. Non-stationary texture synthesis by adversarial expansion. *ACM Trans. Graph.*, 2018.

[48] Piotr Teterwak, Aaron Sarna, Dilip Krishnan, Aaron Maschinot, David Belanger, Ce Liu, and William T Freeman. Boundless: Generative adversarial networks for image extension. In *Int. Conf. Comput. Vis.*, pages 10521–10530, 2019.

[49] Zongxin Yang, Jian Dong, Ping Liu, Yi Yang, and Shuicheng Yan. Very long natural scenery image prediction by outpainting. In *Int. Conf. Comput. Vis.*, pages 10561–10570, 2019.

[50] Ye Ma, Jin Ma, Min Zhou, Quan Chen, Tiezheng Ge, Yuning Jiang, and Tong Lin. Boosting image outpainting with semantic layout prediction. *arXiv preprint arXiv:2110.09267*, 2021.

[51] Yen-Chi Cheng, Chieh Hubert Lin, Hsin-Ying Lee, Jian Ren, Sergey Tulyakov, and Ming-Hsuan Yang. In&out: Diverse image outpainting via gan inversion. *arXiv preprint arXiv:2104.00675*, 2021.

[52] Yi Wang, Xin Tao, Xiaoyong Shen, and Jiaya Jia. Wide-context semantic image extrapolation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1399–1408, 2019.

[53] Xuecai Hu, Haoyuan Mu, Xiangyu Zhang, Zilei Wang, Tieniu Tan, and Jian Sun. Meta-sr: A magnification-arbitrary network for super-resolution. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1575–1584, 2019.

[54] Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Yu Qiao, and Chen Change Loy. Esrgan: Enhanced super-resolution generative adversarial networks. In *Proceedings of the European conference on computer vision (ECCV) workshops*, 2018.

[55] Xintao Wang, Liangbin Xie, Chao Dong, and Ying Shan. Real-esrgan: Training real-world blind super-resolution with pure synthetic data. In *Int. Conf. Comput. Vis.*, pages 1905–1914, 2021.

[56] Eli Shechtman and Michal Irani. Matching local self-similarities across images and videos. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1–8. IEEE, 2007.

[57] Daniel Glasner, Shai Bagon, and Michal Irani. Super-resolution from a single image. In *Int. Conf. Comput. Vis.*, pages 349–356. IEEE, 2009.

[58] Jia-Bin Huang, Abhishek Singh, and Narendra Ahuja. Single image super-resolution from transformed self-exemplars. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 5197–5206, 2015.

[59] Kai Zhang, Martin Danelljan, Yawei Li, Radu Timofte, Jie Liu, Jie Tang, Gangshan Wu, Yu Zhu, Xiangyu He, Wenjie Xu, et al. Aim 2020 challenge on efficient super-resolution: Methods and results. In *Eur. Conf. Comput. Vis.*, pages 5–40. Springer, 2020.

[60] Sheng-Yu Wang, Oliver Wang, Richard Zhang, Andrew Owens, and Alexei A Efros. Cnn-generated images are surprisingly easy to spot... for now. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020.

[61] Lucy Chai, David Bau, Ser-Nam Lim, and Phillip Isola. What makes fake images detectable? understanding properties that generalize. In *Eur. Conf. Comput. Vis.*, 2020.

[62] Haitian Zheng, Mengqi Ji, Haoqian Wang, Yebin Liu, and Lu Fang. Crossnet: An end-to-end reference-based super resolution network using cross-scale warping. In *Eur. Conf. Comput. Vis.*, pages 88–104, 2018.

[63] Fuzhi Yang, Huan Yang, Jianlong Fu, Hongtao Lu, and Baining Guo. Learning texture transformer network for image super-resolution. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 5791–5800, 2020.

[64] Liying Lu, Wenbo Li, Xin Tao, Jiangbo Lu, and Jiaya Jia. Masa-sr: Matching acceleration and spatial adaptation for reference-based image super-resolution. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 6368–6377, 2021.

[65] Yuming Jiang, Kelvin CK Chan, Xintao Wang, Chen Change Loy, and Ziwei Liu. Robust reference-based super-resolution via c2-matching. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 2103–2112, 2021.

[66] Bin Xia, Yapeng Tian, Yucheng Hang, Wenming Yang, Qingmin Liao, and Jie Zhou. Coarse-to-fine embedded patchmatch and multi-scale dynamic aggregation for reference-based super-resolution. *arXiv preprint arXiv:2201.04358*, 2022.

[67] Minyoung Huh, Richard Zhang, Jun-Yan Zhu, Sylvain Paris, and Aaron Hertzmann. Transforming and projecting images into class-conditional generative networks. In *Eur. Conf. Comput. Vis.*, 2020.

[68] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018.

[69] Shengyu Zhao, Jonathan Cui, Yilun Sheng, Yue Dong, Xiao Liang, Eric I Chang, and Yan Xu. Large scale image completion via co-modulated generative adversarial networks. In *Int. Conf. Learn. Represent.*, 2021.

[70] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. volume 30, 2017.

[71] Gaurav Parmar, Richard Zhang, and Jun-Yan Zhu. On aliased resizing and surprising subtleties in gan evaluation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022.

[72] Bart Thomee, David A Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. Yfcc100m: The new data in multimedia research. *Communications of the ACM*, 59(2):64–73, 2016.

[73] Xintao Wang, Yu Li, Honglun Zhang, and Ying Shan. Towards real-world blind face restoration with generative facial prior. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2021.

[74] Zongze Wu, Dani Lischinski, and Eli Shechtman. Stylespace analysis: Disentangled controls for stylegan image generation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 12863–12872, 2021.

[75] Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. Styleclip: Text-driven manipulation of stylegan imagery. In *Int. Conf. Comput. Vis.*, pages 2085–2094, October 2021.

Supplementary

In the supplementary, we first demonstrate an extension of our approach towards panorama generation (Section 6). In Section 7, we provide additional qualitative and quantitative comparisons to baselines (superresolution 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 8.

6 Panorama generation extension

Our default training setup assumes that we use low-resolution images is to learn global context, and patches from high-resolution images to learn details. An alternative setup to learn from patches directly, *even 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 for 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 9.

7 Experiments

7.1 Dataset Collection

To collect our varied-size dataset, we scrape image collections from Flickr photo groups (Tab. 6). In cases where a standard fixed-resolution dataset is available (*e.g.* , LSUN Churches [8]), 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 [72, 9], and to not violate license permissions, we will release the image IDs but not the images directly.

Table 6: Image sources for construction of our varied-resolution datasets.

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

7.2 Patch-FID

We describe additional 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. We modify the FID pipeline to avoid downsampling global images at higher resolutions to a fixed 299 pixel width. Instead, we sample patches of size p from real images at global scale s and locations $c_{\mathbf{v},s}$, and generate the corresponding patch $G\,(z,c_{\mathbf{v},s},s)$. 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

Figure 9: 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.

sensitive to image quality 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. 10). 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.

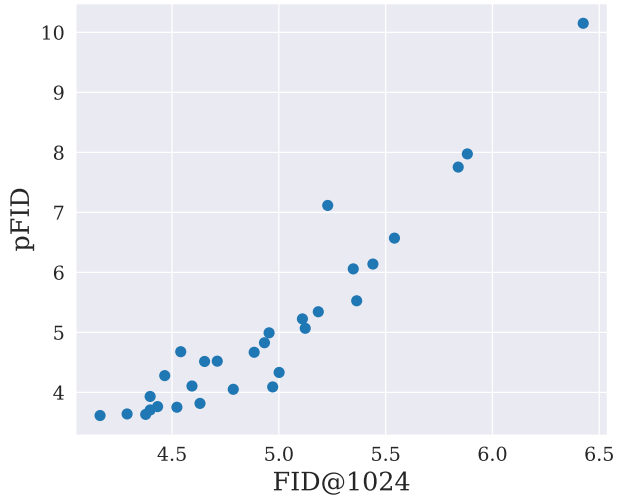

Figure 10: pFID vs FID@1024. The metrics are largely correlated, but FID@1024 downsamples images and assumes a fixed-size dataset. On other domains where a ground-truth HR dataset is not available, we primarily use pFID to measure sample quality.

7.3 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 7 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 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 ds-pFID, 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.

7.4 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 [28, 7, 25, 27] 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 [28] 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

Table 7: 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 [73], but the FID results were worse than that of Real-ESRGAN [55]

	FFHQ6K			**Church**		
	FID@256	FID@1024	pFID	FID@256	FID@1024	pFID
Fixed-Size	3.71	33.80	52.95	3.39	242.10	146.24
LIIF	–	7.05	22.93	–	18.66	83.88
Real-ESRGAN	–	19.04	16.92	–	12.26	23.04
Ours	3.27	3.95	2.94	3.59	8.08	12.18
	Mountain			**Birds**		
	FID@1024	ds-pFID	pFID	FID@256	FID@512	pFID
Fixed-Size	3.09	13.42	46.20	3.92	12.69	55.42
LIIF	–	4.55	23.10	–	7.29	30.19
Real-ESRGAN	–	7.60	19.05	–	8.51	16.10
Ours	3.09	4.20	8.76	3.80	6.33	6.81

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. 11.

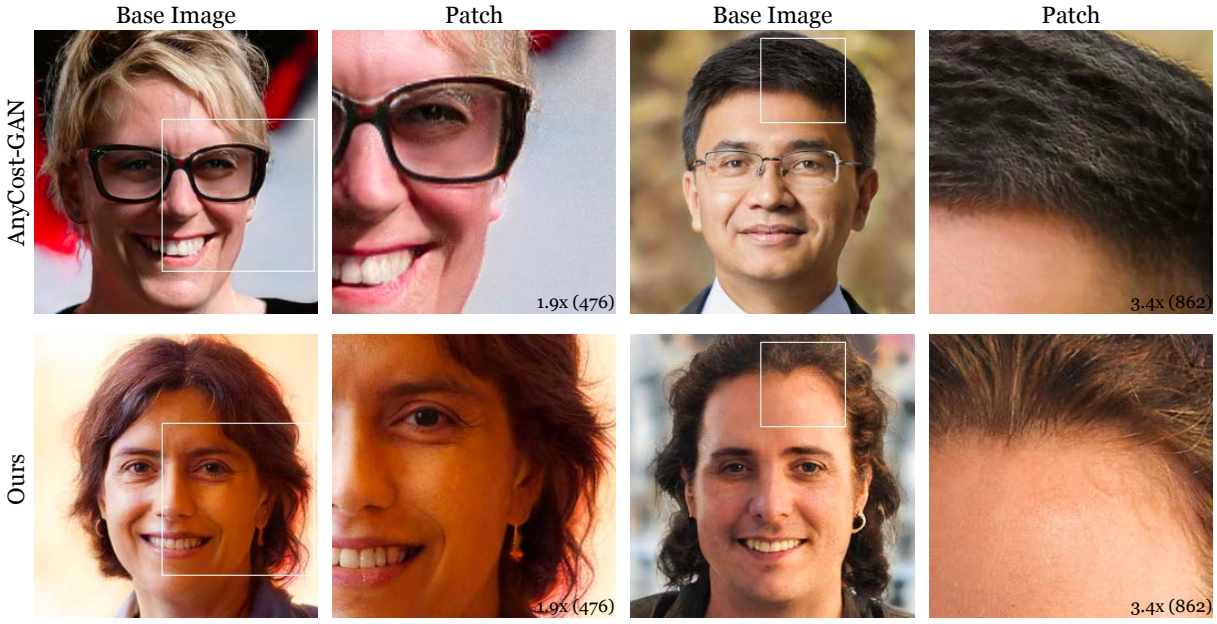

Figure 11: Using the same FFHQ6K dataset, we train an Anycost-GAN [28] and compare it to our model. While Anycost-GAN adds additional modules to increase synthesis resolution, our model shares weights across resolutions. Note that the output from Anycost-GAN contain more visual artifacts, particularly in finely textured regions such as hair.

7.5 Comparison to Oracle Generator

In Section 4 of the paper, we describe our experimental setup on the face domain, and in Table 5, we show competitive performance training on few HR images, even compared to an oracle trained 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.

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 generator on the full FFHQ dataset (Fig. 12). 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. 8. While the oracles can achieve lower FID, 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 allows us to take advantage of the varied resolutions of images in the wild.

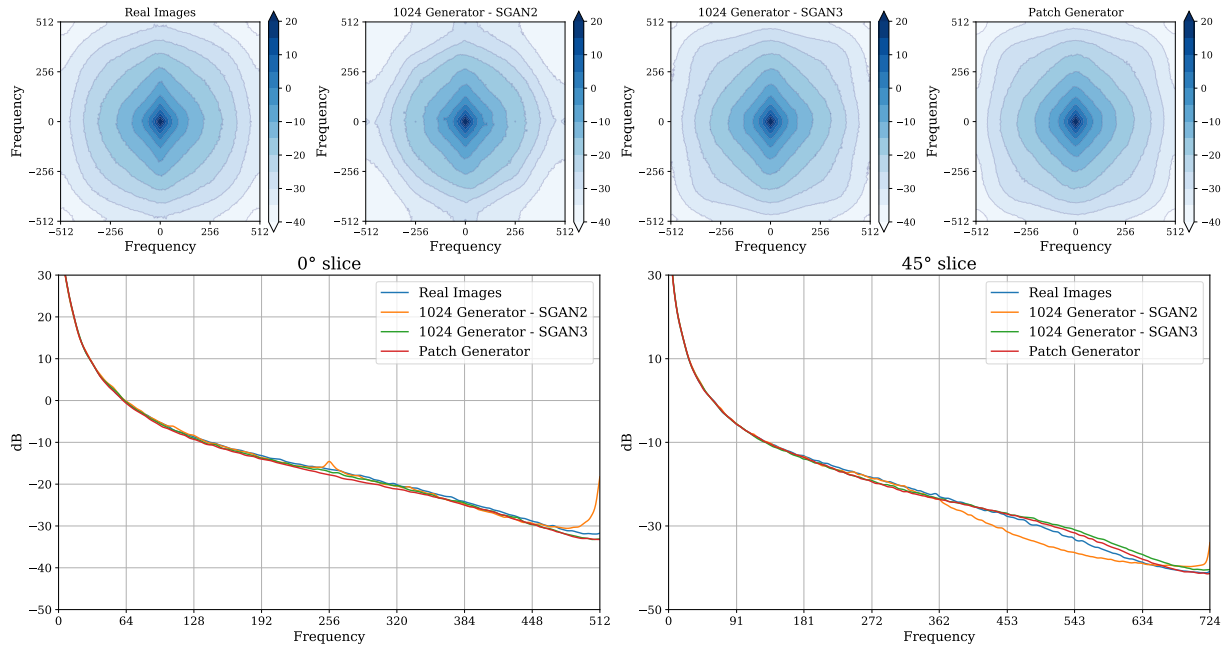

Figure 12: Comparison of frequency distribution. We plot the frequency spectrum of real images, StyleGAN2 and StyleGAN3 trained on the entire FFHQ dataset at 1024 resolution, and our Patch Generator which is trained on $p \times p$ patches of FFHQ6K (which contains approximately 1k images at 1024 resolution and 5k at lower resolutions). The frequency distributions are similar, suggesting that even a smaller generator is able to approximate fine textures well.

Table 8: Comparison of our patch generator (6k images, varied sizes) to oracle generators which train on the entire FFHQ dataset (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.

	FID@256	@512	@1024	pFID
SGAN2 Oracle	3.05	2.81	2.69	2.26
SGAN3 Oracle	3.52	3.23	3.06	2.44
Patch (Ours)	3.27	3.92	3.95	2.94

Table 9: Variations of teacher regularizer on FFHQ6k. The inverse teacher variant outperforms forward teacher regularization, and adding a scale-conditioning branch further improves FID at higher resolutions. Omitting the teacher improves pFID on local patches, but global image FID worsens.

	FID@256	@512	@1024	pFID
No teacher	5.50	5.93	7.13	3.17
Forward teacher	3.23	4.21	5.35	6.06
Inverse teacher	3.35	4.18	4.88	4.67
Inv + scale cond (Ours)	3.37	4.41	4.47	4.28

7.6 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.

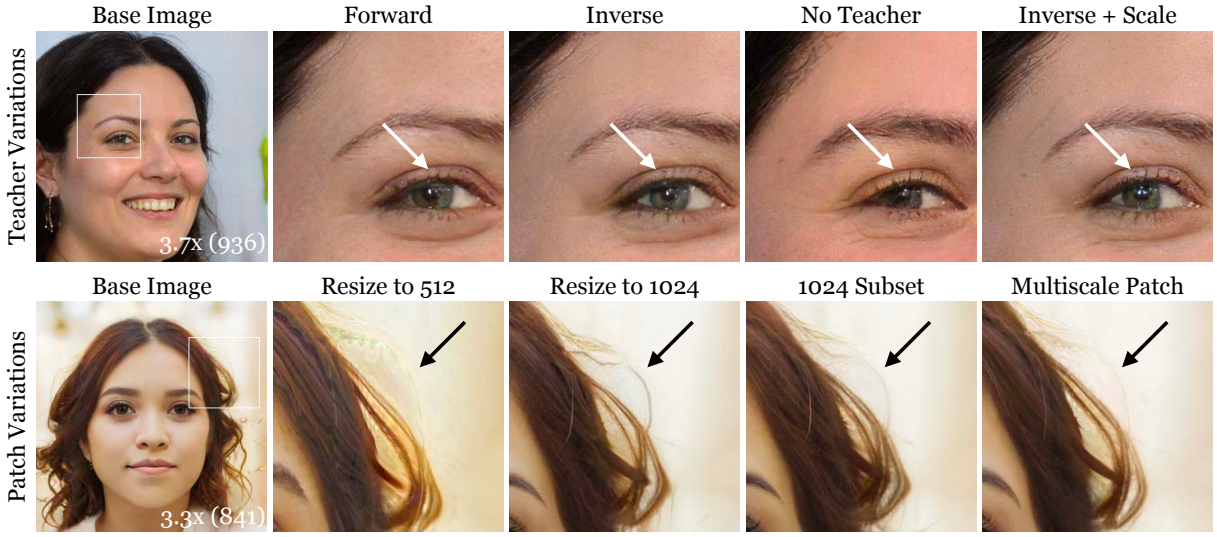

Figure 13: Qualitative examples of model variations. (Top) Using an inverse teacher loss and the scale conditioning branch generates sharper details while also preserving similarity to the base image. (Bottom) We compare our multi-size training approach to methods that do not take advantage of different 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.

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. 9 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. 13 (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 λ_{teacher} balances between

Table 10: 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.

	FID@256	@512	@1024	pFID	L1
$\lambda_{\text{teacher}} = 0$	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 11: We sample patches from the HR dataset at global resolutions between $(s_{\text{min}}, s_{\text{max}})$. Using the same model, but simply supplying patches from higher resolution images, improves the synthesis result at 1024 resolution.

	FID@256	@512	@1024	pFID
(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

local image quality and similarity to the base resolution image, where higher λ_{teacher} offers the most similarity to the base resolution with lower L1 difference, but lower λ_{teacher} improves pFID, suggesting better quality of the synthesized patches (Tab. 10).

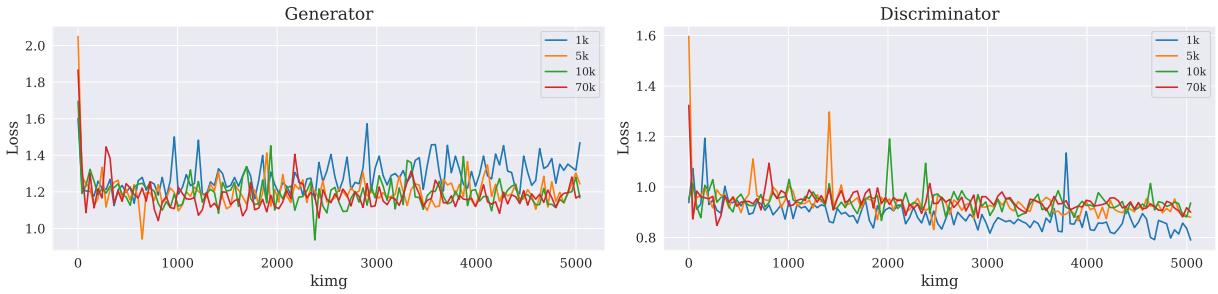

Figure 14: Number of training images. While FID numbers are similar, we find that using 1K HR images for training shows some evidence of divergence. The training dynamics of 5K images is similar to that of the full dataset.

Fixed-size vs Multi-size training. Fig 13 (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. 14). 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. 11, Fig. 15). 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 $p = 256$, and the model is jointly trained on the fixed-size dataset to preserve FID@256. Accordingly for the other domains, our sampled resolutions

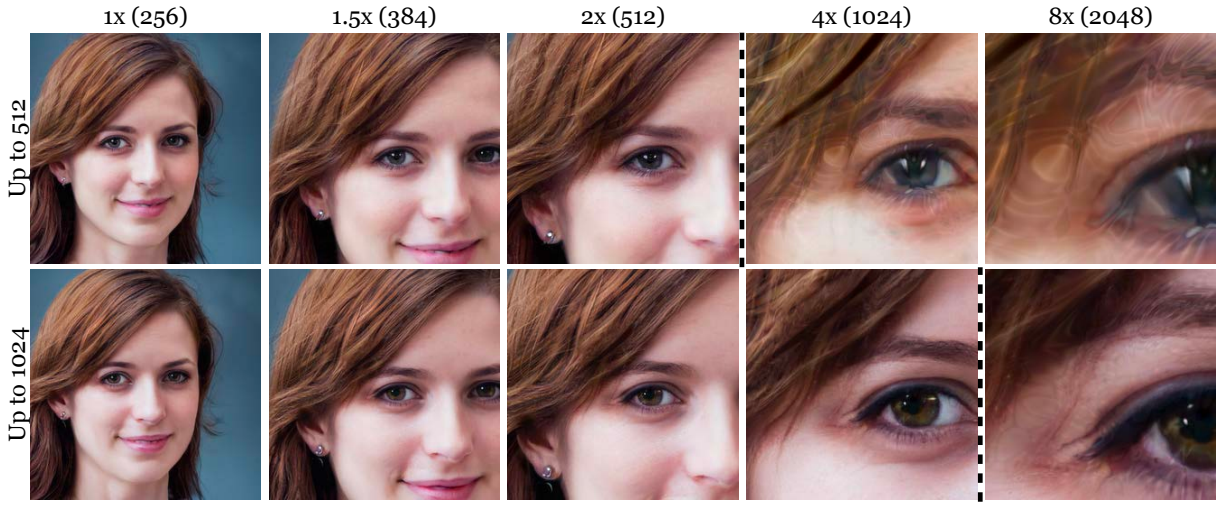

Figure 15: Impact of sampling resolutions. Using the same model architecture and FFHQ6k dataset, we sample images (top) from 256 to 512 resolution, and (bottom) from 512 to 1024 resolution. The dotted line indicates when inference resolution exceeds the maximum training resolution. Watery artifacts start to appear when extrapolation, but this can be tempered by simply training on patches from larger images.

for HR dataset range between the native resolution s_{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.

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. 12). (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 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 having 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.

Table 12: 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.

	FID@256	@512	@1024	pFID
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

7.7 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

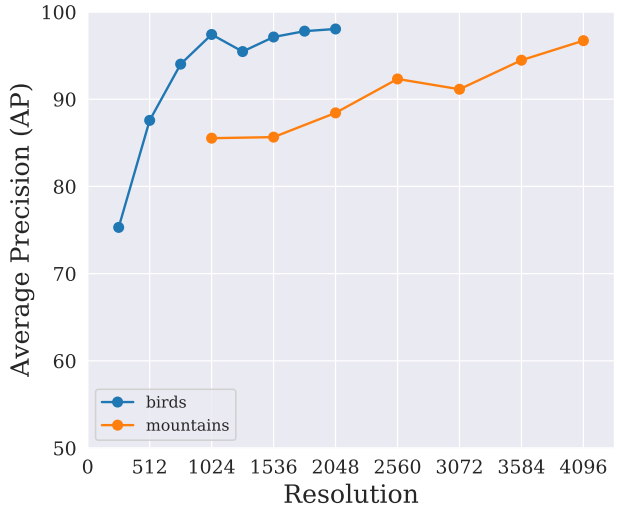

Figure 16: Detection score from [60] on our Birds and Mountains datasets. All scores are above chance of 50%. Note in both cases, the detectability of our network trends upwards with resolution.

from Wang et al. [60] on our Birds (generated $256 \rightarrow 2048$) and Mountains ($1024 \rightarrow 4096$) generators, across a large range of resolutions. As shown in Figure 16, 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.

8 Additional implementation details

Building off the StyleGAN3 [2] architecture, we describe our coordinate conditioning and scale modulation branch applied to enable generation of multi-scale patches during the second training phase.

8.1 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_{im}-by-s_{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.

$$x, s, \mathbf{v} = \text{Crop}(\text{Downsample}(x_i)) \tag{8.1}$$

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 homogeneous coordinates:

$$c_{\mathbf{v},s} = T_{\text{patch}} * c = \begin{bmatrix} \frac{p}{s} & 0 & \mathbf{v}_x \\ 0 & \frac{p}{s} & \mathbf{v}_y \\ 0 & 0 & 1 \end{bmatrix} * c \tag{8.2}$$

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:

$$F_{h,w}(c_{\mathbf{v},s}) = \sin\left(2\pi\, B c_{\mathbf{v},s} + \phi\right) \in \mathbb{R}^K, \tag{8.3}$$

8.2 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 into the range $[-1, 1]$ using:

$$\bar{s} = 2\frac{s - p}{s_{\text{max}} - p} - 1, \tag{8.4}$$

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.

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 [74, 75] to obtain the final modulation parameter $M(z, s)_k$ at layer k:

$$M(z, s)_k = (W_{z,k} * M_z(z) + b_{z,k}) + (W_{s,k} * M_s(s) + b_{s,k}) \tag{8.5}$$

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.

8.3 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 [2], 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.