Analyzing and Improving the Image Quality of StyleGAN

펀디멘탈 팀: 송헌, 고형권, 김동희, 김창연, 이민경, 이재윤

발표: 고형권 (hyungkwonko@gmail.com)

Progressive GAN (2018 ICLR) recap

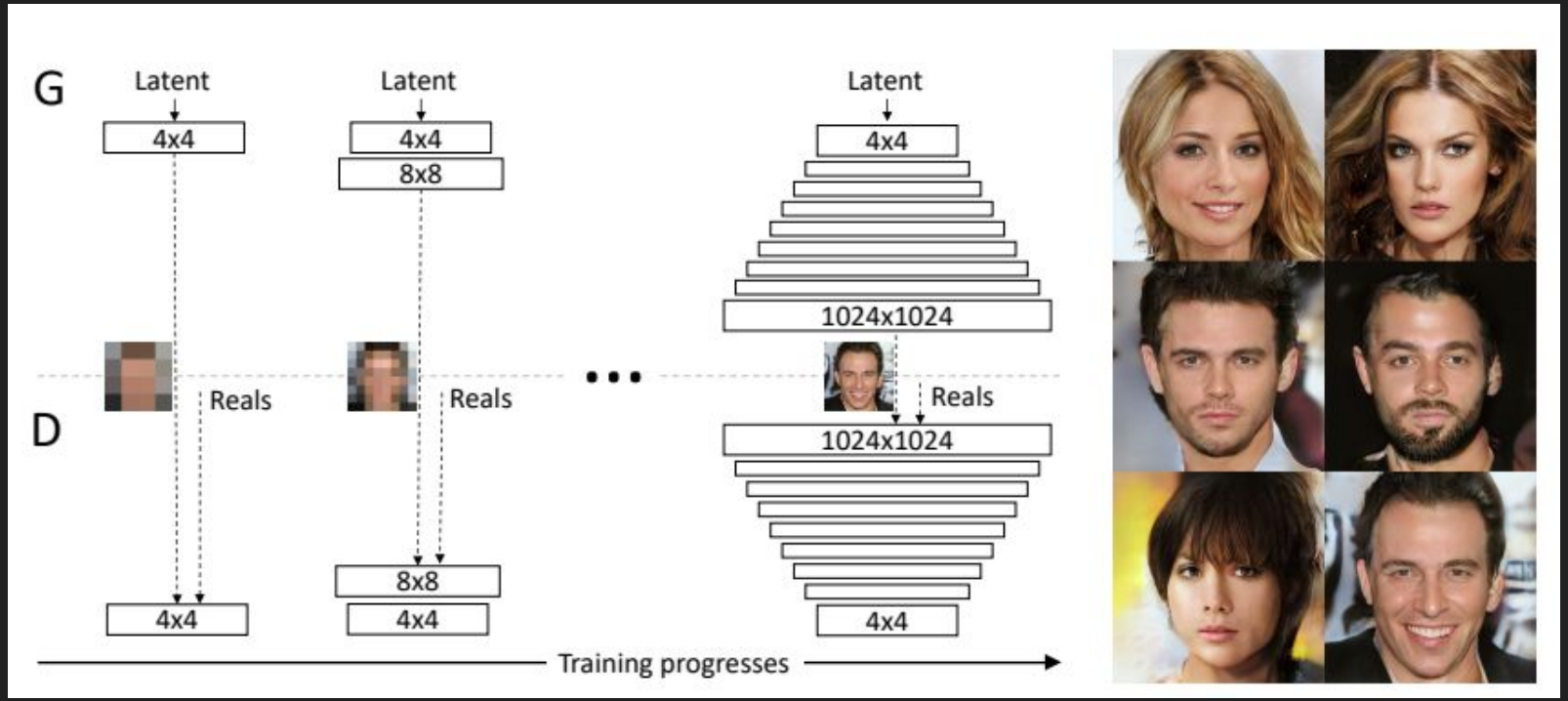


Image from original paper

StyleGAN (2019 CVPR) recap

	Method	CelebA-HQ	FFHQ
A	Baseline Progressive GAN [28]	7.79	8.04
B	+ Tuning (incl. bilinear up/down)	6.11	5.25
C	+ Add mapping and styles	5.34	4.85
D	+ Remove traditional input	5.07	4.88
E	+ Add noise inputs	**5.06**	4.42
F	+ Mixing regularization	5.17	**4.40**

Table 1. Fréchet inception distance (FID) for various generator designs (lower is better). In this paper we calculate the FIDs using 50,000 images drawn randomly from the training set, and report the lowest distance encountered over the course of training.

Image from original paper

StyleGAN (2019 CVPR) recap

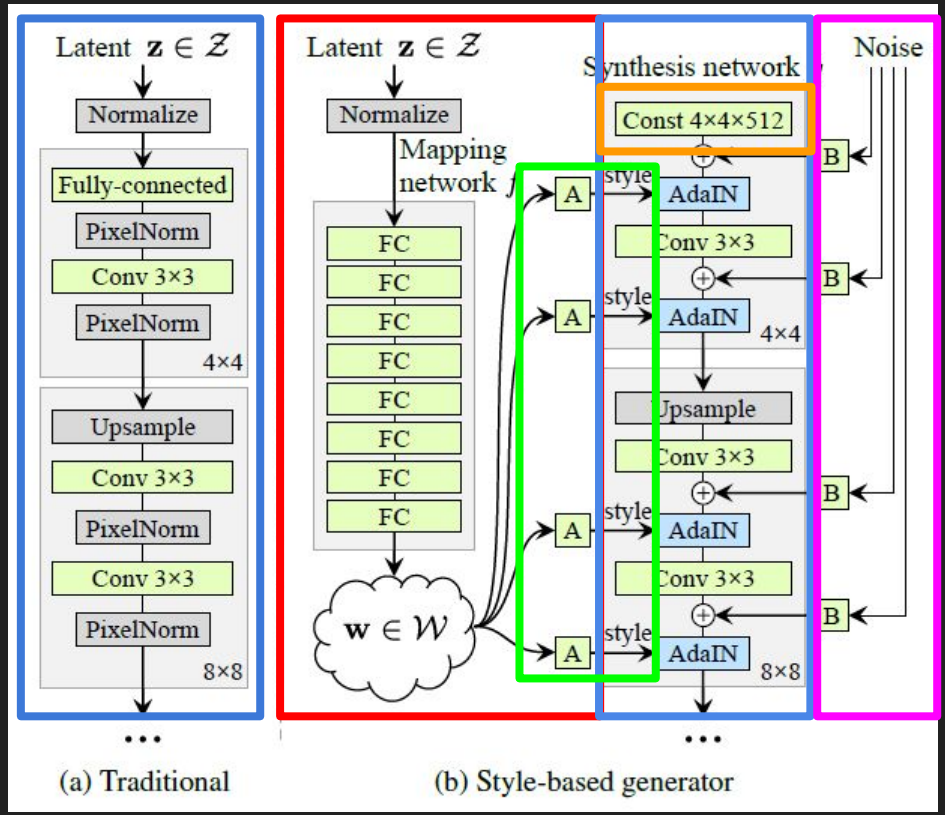
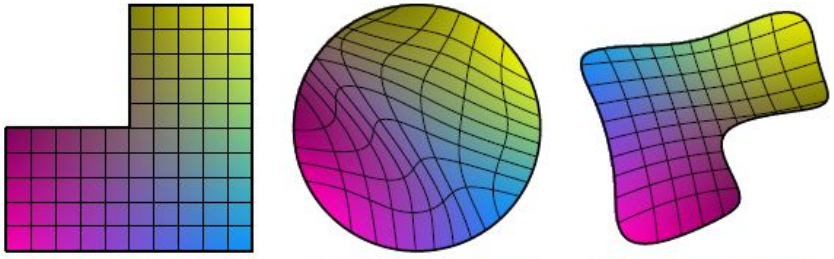


Image from original paper

StyleGAN (2019 CVPR) recap



(a) Distribution of features in training set

(b) Mapping from $\mathcal{Z}$ to features

(c) Mapping from $\mathcal{W}$ to features

Figure 6. Illustrative example with two factors of variation (image features, e.g., masculinity and hair length). (a) An example training set where some combination (e.g., long haired males) is missing. (b) This forces the mapping from $\mathcal{Z}$ to image features to become curved so that the forbidden combination disappears in $\mathcal{Z}$ to prevent the sampling of invalid combinations. (c) The learned mapping from $\mathcal{Z}$ to $\mathcal{W}$ is able to "undo" much of the warping.

	Method	Path length full	Path length end	Separability
B	Traditional generator $\mathcal{Z}$	412.0	415.3	10.78
D	Style-based generator $\mathcal{W}$	446.2	376.6	3.61
E	+ Add noise inputs $\mathcal{W}$	**200.5**	**160.6**	3.54
	+ Mixing 50% $\mathcal{W}$	231.5	182.1	**3.51**
F	+ Mixing 90% $\mathcal{W}$	234.0	195.9	3.79

Table 3. Perceptual path lengths and separability scores for various generator architectures in FFHQ (lower is better). We perform the measurements in $\mathcal{Z}$ for the traditional network, and in $\mathcal{W}$ for style-based ones. Making the network resistant to style mixing appears to distort the intermediate latent space $\mathcal{W}$ somewhat. We hypothesize that mixing makes it more difficult for $\mathcal{W}$ to efficiently encode factors of variation that span multiple scales.

Image from original paper

Pretrained models for different datasets

Awesome Pretrained StyleGAN

A collection of pre-trained StyleGAN models trained on different datasets at different resolution.

For the equivalent collection for StyleGAN 2, see this repo

Table of Contents

- Models
  - car (config-e)
  - car (config-f)
  - cat
  - church
  - faces (FFHQ config-e)
  - faces (FFHQ config-e 256x256)
  - faces (FFHQ config-f)
  - faces (FFHQ config-f 512x512)
  - horse
  - Imagenet
  - WikiArt
  - Anime portraits
  - microscope images

https://github.com/justinpinkney/awesome-pretrained-stylegan2

Q & A

StyleGAN2 (2020 CVPR)

- *In this paper, we focus all analysis solely on W, as it is the relevant latent space from the synthesis network's point of view.*

StyleGAN2 (2020 CVPR)

Configuration	FFHQ, 1024×1024				LSUN Car, 512×384			
	FID ↓	Path length ↓	Precision ↑	Recall ↑	FID ↓	Path length ↓	Precision ↑	Recall ↑
A Baseline StyleGAN [21]	4.40	212.1	**0.721**	0.399	3.27	1484.5	**0.701**	0.435
B + Weight demodulation	4.39	175.4	0.702	0.425	3.04	862.4	0.685	0.488
C + Lazy regularization	4.38	158.0	0.719	0.427	2.83	981.6	0.688	0.493
D + Path length regularization	4.34	**122.5**	0.715	0.418	3.43	651.2	0.697	0.452
E + No growing, new G & D arch.	3.31	124.5	0.705	0.449	3.19	471.2	0.690	0.454
F + Large networks (StyleGAN2)	**2.84**	145.0	0.689	**0.492**	**2.32**	**415.5**	0.678	**0.514**
Config A with large networks	3.98	199.2	0.716	0.422	–	–	–	–

Table 1. Main results. For each training run, we selected the training snapshot with the lowest FID. We computed each metric 10 times with different random seeds and report their average. *Path length* corresponds to the PPL metric, computed based on path endpoints in $\mathcal{W}$ [21], without the central crop used by Karras et al. [21]. The FFHQ dataset contains 70k images, and the discriminator saw 25M images during training. For LSUN CAR the numbers were 893k and 57M. ↑ indicates that higher is better, and ↓ that lower is better.

Image from original paper

The Problem of StyleGAN (Droplet artifacts)

Feature map 64^2 | Feature map 128^2 | Feature map 256^2 | Feature map 512^2 | Generated image

Image from original paper

The Problem of StyleGAN (Droplet artifacts)

youtube.com/watch?v=c-NJtV9Jvp0

B. Weight Demodulation

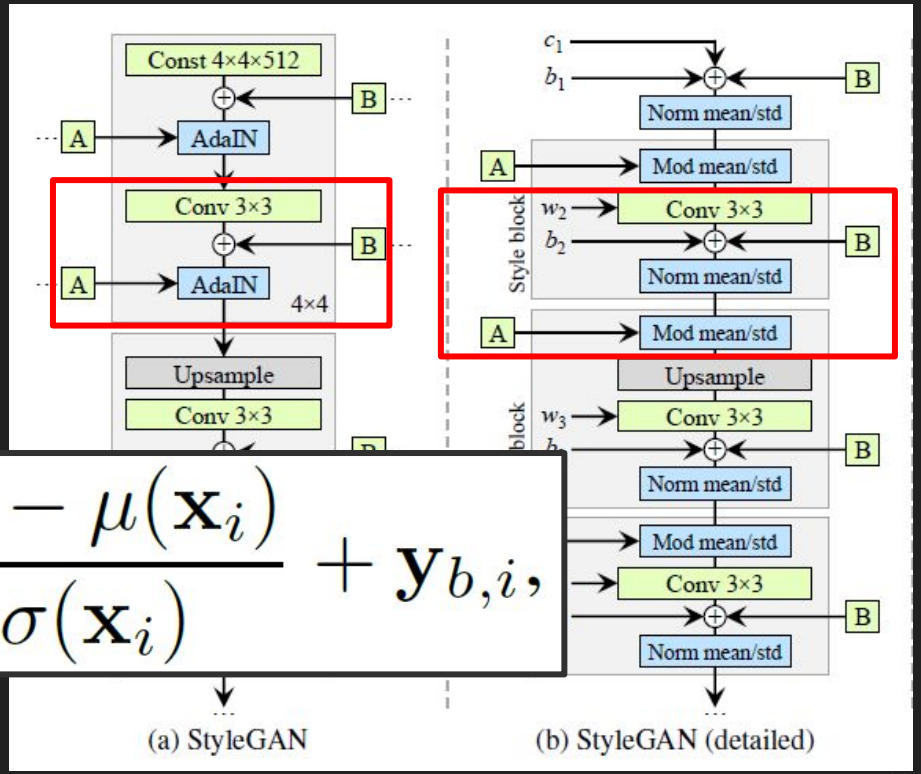
- AdaIN operation normalizes the mean and variance of each feature map separately, thereby potentially **destroying any information found in the magnitudes of the features relative to each other.**

- How to fix? **Get rid of AdaIN structure**

B. Weight Demodulation

- BEFORE (StyleGAN)

- W → affine transform → A



$$\mathrm{AdaIN}(\mathbf{x}_i, \mathbf{y}) = \mathbf{y}_{s,i} \frac{\mathbf{x}_i - \mu(\mathbf{x}_i)}{\sigma(\mathbf{x}_i)} + \mathbf{y}_{b,i},$$

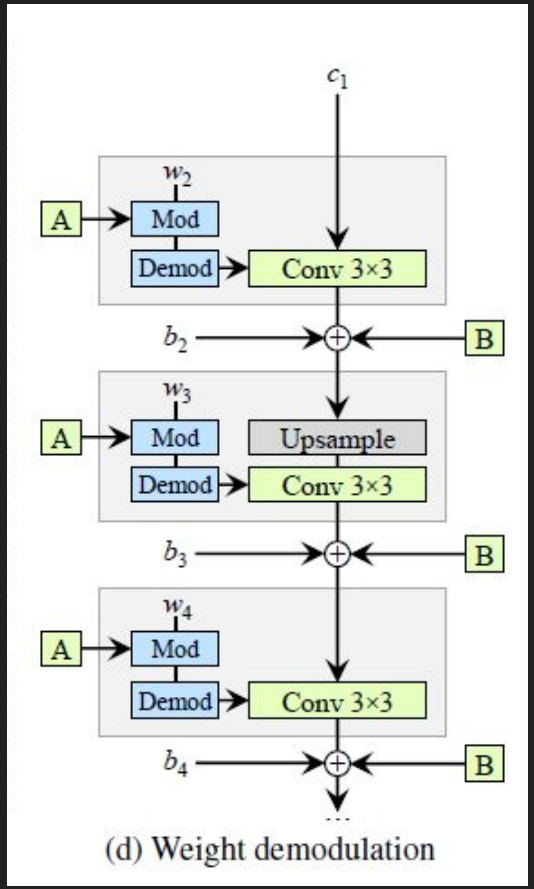
Image from original paper

B. Weight Demodulation

- AFTER (StyleGAN2)

$$w'_{ijk} = s_i \cdot w_{ijk}, \qquad \sigma_j = \sqrt{\sum_{i,k} {w'_{ijk}}^2},$$

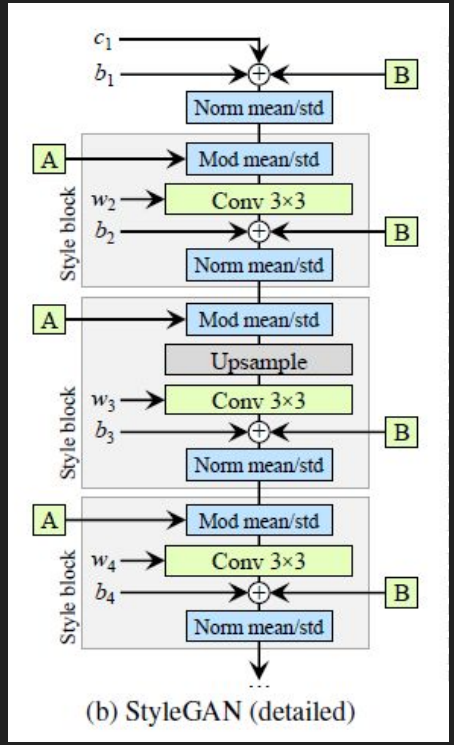
$$w''_{ijk} = w'_{ijk} \Big/ \sqrt{\sum_{i,k} {w'_{ijk}}^2 + \epsilon},$$



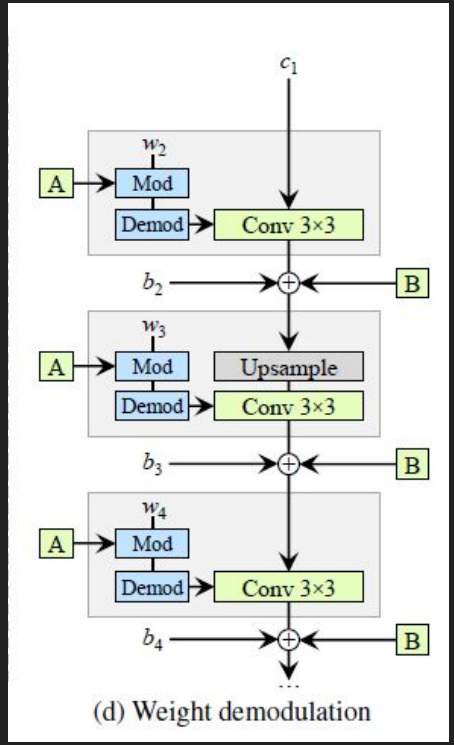
(d) Weight demodulation

Image from original paper

B. Weight Demodulation



(b) StyleGAN (detailed)



(d) Weight demodulation

Image from original paper

The Result of Weight Demodulation

youtube.com/watch?v=c-NJtV9Jvp0

The Result of Weight Demodulation (in style mixing)

youtube.com/watch?v=c-NJtV9Jvp0

Q & A

Perceptual Path Length score

- Perceptual Path Length score (PPL): Sampled from latent-space vectors using interpolation → distance calculation using features of VGG

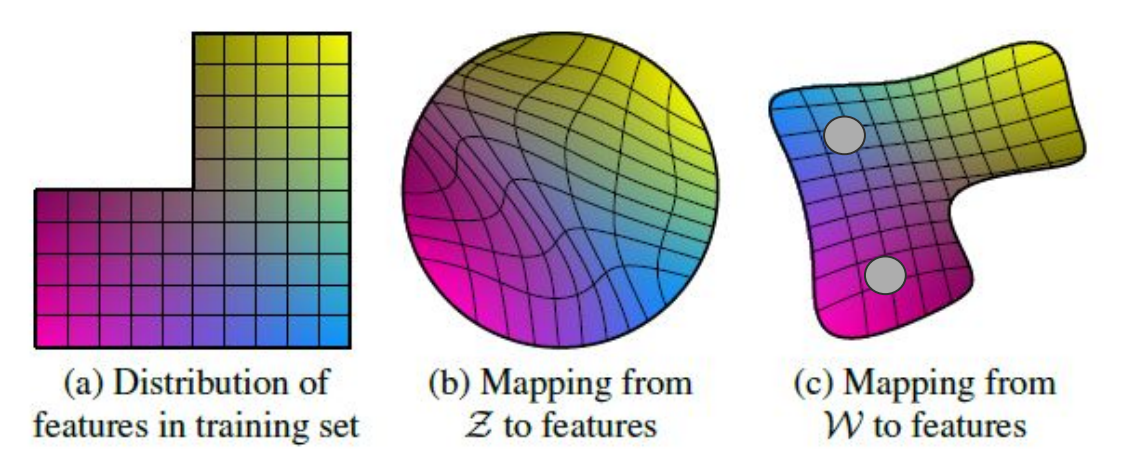


Image from original paper

Perceptual Path Length score

(a) Low PPL scores (b) High PPL scores

Figure 4. Connection between perceptual path length and image quality using baseline StyleGAN (config A) with LSUN CAT. (a) Random examples with low PPL ($\leq 10^{\text{th}}$ percentile). (b) Examples with high PPL ($\geq 90^{\text{th}}$ percentile). There is a clear correlation between PPL scores and semantic consistency of the images.

Image from original paper

Why LOW PPL == GOOD Quality?

- *NOT immediately obvious…*
- *Hypothesis:*

  GOOD image region (STRETCHED, large latent space)

  BAD image region (SQUEEZED, small latent space)

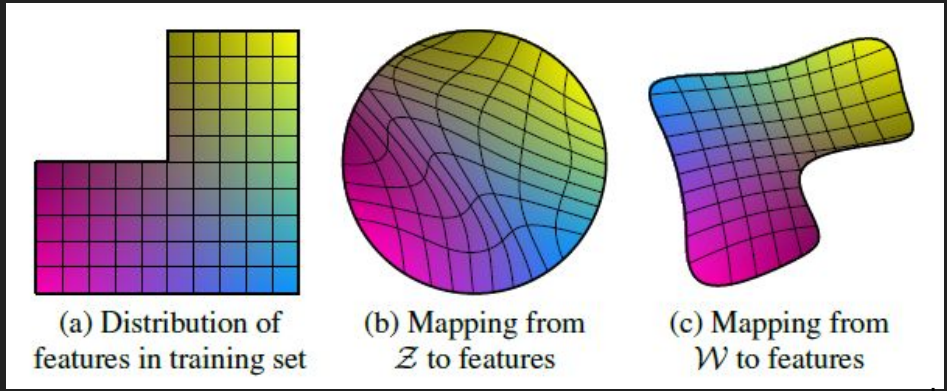
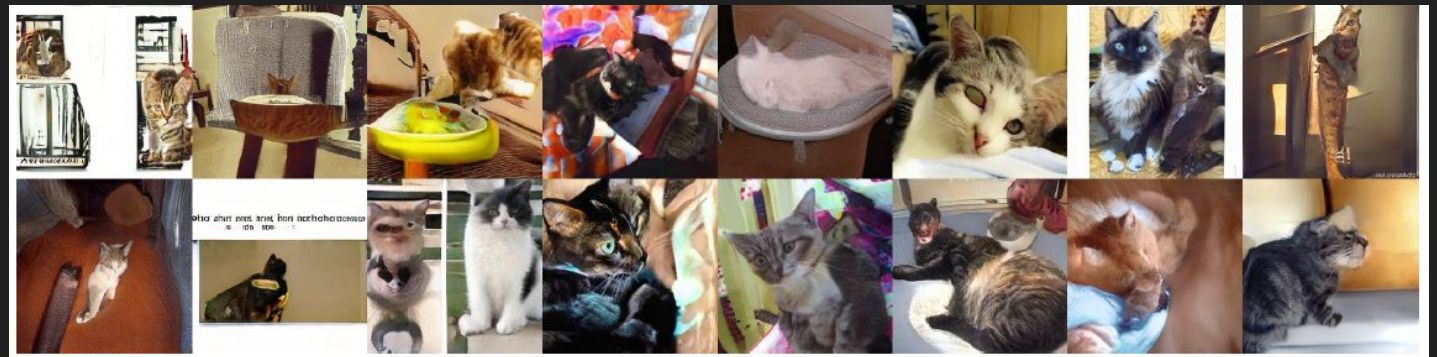


Image from original paper

Perceptual Path Length score



Model 1: FID = 8.53, P = 0.64, R = 0.28, PPL = 924

Model 2: FID = 8.53, P = 0.62, R = 0.29, PPL = 387

Why better than FID and Precision & Recall?

[2]We believe that the key to the apparent inconsistency lies in the particular choice of feature space rather than the foundations of FID or P&R. It was recently discovered that classifiers trained using ImageNet [35] tend to base their decisions much more on texture than shape [12], while humans strongly focus on shape [28]. This is relevant in our context because

(a) Texture image

81.4%	**Indian elephant**
10.3%	indri
8.2%	black swan

(b) Content image

71.1%	**tabby cat**
17.3%	grey fox
3.3%	Siamese cat

(c) Texture-shape cue conflict

63.9%	**Indian elephant**
26.4%	indri
9.6%	black swan

Figure 1: Classification of a standard ResNet-50 of **(a)** a texture image (elephant skin: only texture cues); **(b)** a normal image of a cat (with both shape and texture cues), and **(c)** an image with a texture-shape cue conflict, generated by style transfer between the first two images.

Geirhos et al., 2019 ICLR (oral)

D. Path Length Regularization

- *We would like to encourage that a fixed-size step in W results in a non-zero, fixed-magnitude change in the image.*
- *We can measure the deviation from this ideal empirically by stepping into random directions in the image space and observing the corresponding w gradients.*

$$\mathcal{L}_{\text{pl}} = \mathbb{E}_{\mathbf{w}}\mathbb{E}_{\mathbf{y}}\left(\left\|\mathbf{J}_{\mathbf{w}}^{T}\mathbf{y}\right\|_2 - a\right)^2, \quad (6)$$

where $\mathbf{y} \in \mathbb{R}^M$ is a unit normal distributed random variable in the space of generated images (of dimension $M = 3wh$, namely the RGB image dimensions), $\mathbf{J}_{\mathbf{w}} \in \mathbb{R}^{M \times L}$ is the Jacobian matrix of the generator function $g : \mathbb{R}^L \mapsto \mathbb{R}^M$ at a latent space point $\mathbf{w} \in \mathbb{R}^L$, and $a \in \mathbb{R}$ is a global value that expresses the desired scale of the gradients.

The constant a is set dynamically during optimization as the long-running exponential moving average of the lengths $\|\mathbf{J}_{\mathbf{w}}^{T}\mathbf{y}\|_2$, allowing the optimization to find a suitable global scale by itself.

Augustus Odena, Jacob Buckman, Catherine Olsson, Tom B. Brown, Christopher Olah, Colin Raffel, and Ian Goodfellow. Is generator conditioning causally related to GAN performance? CoRR, abs/1802.08768, 2018.

C. Lazy Regularization

- *As the resulting regularization term is somewhat expensive to compute, we first describe a general optimization that applies to any regularization technique.*
- *… performed only once every 16 mini-batches*

Q & A

The Problem of StyleGAN (Phase artifacts)

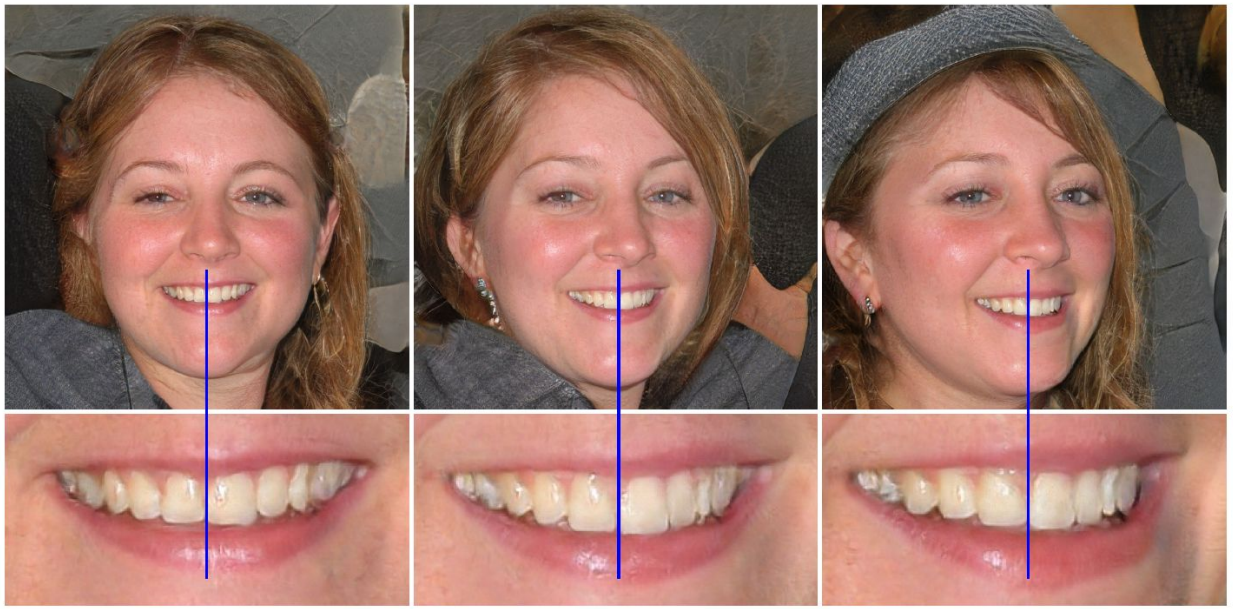


Figure 6. Progressive growing leads to "phase" artifacts. In this example the teeth do not follow the pose but stay aligned to the camera, as indicated by the blue line.

Image from original paper

The Problem of StyleGAN (Phase artifacts)

youtube.com/watch?v=c-NJtV9Jvp0

The Problem of StyleGAN (Phase artifacts)

- *… We believe the problem is that in progressive growing each resolution serves momentarily as the output resolution,* ***forcing it to generate maximal frequency details,*** *which then leads to the trained network to have excessively high frequencies in the intermediate layers, ...*

MSG-GAN (2020 CVPR)

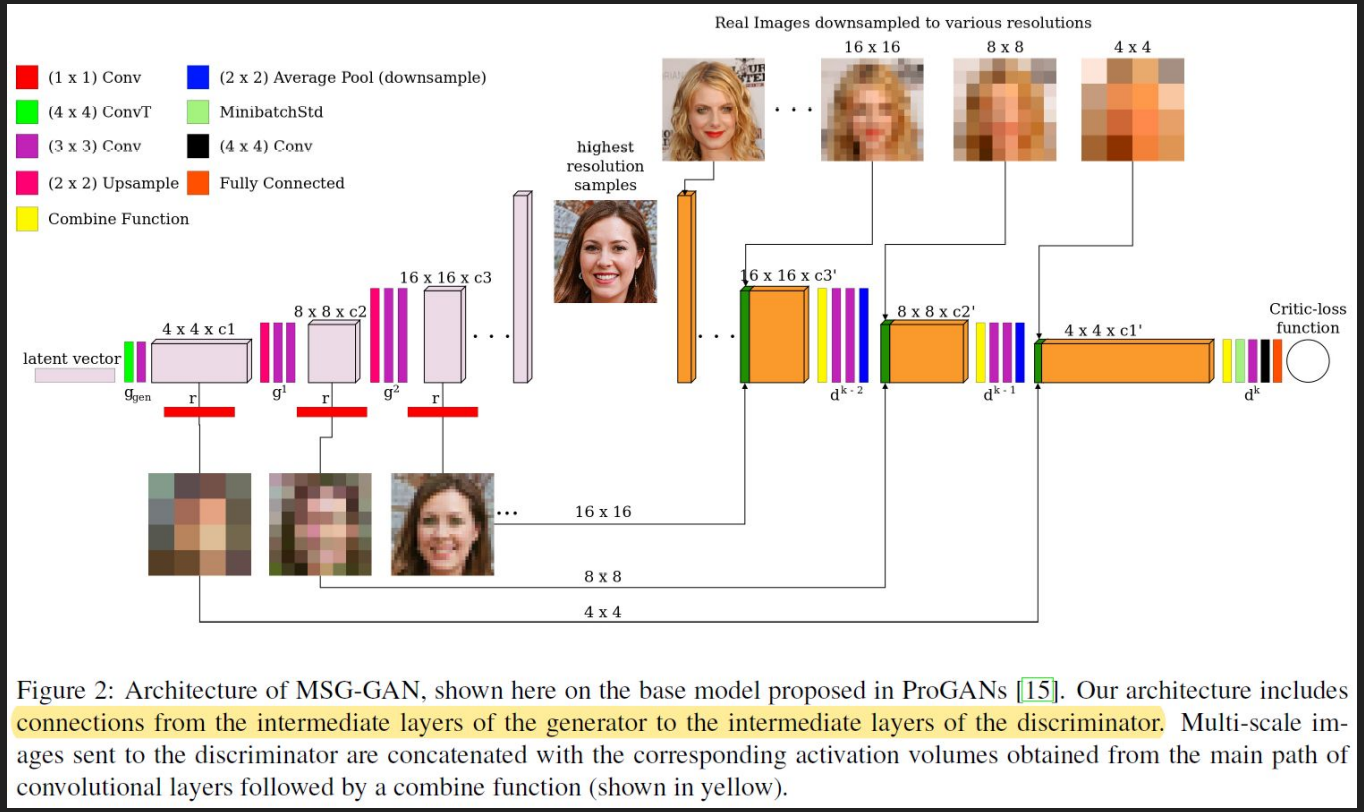


Figure 2: Architecture of MSG-GAN, shown here on the base model proposed in ProGANs [15]. Our architecture includes connections from the intermediate layers of the generator to the intermediate layers of the discriminator. Multi-scale images sent to the discriminator are concatenated with the corresponding activation volumes obtained from the main path of convolutional layers followed by a combine function (shown in yellow).

Image from original paper

E. No Growing, New G & D Architecture

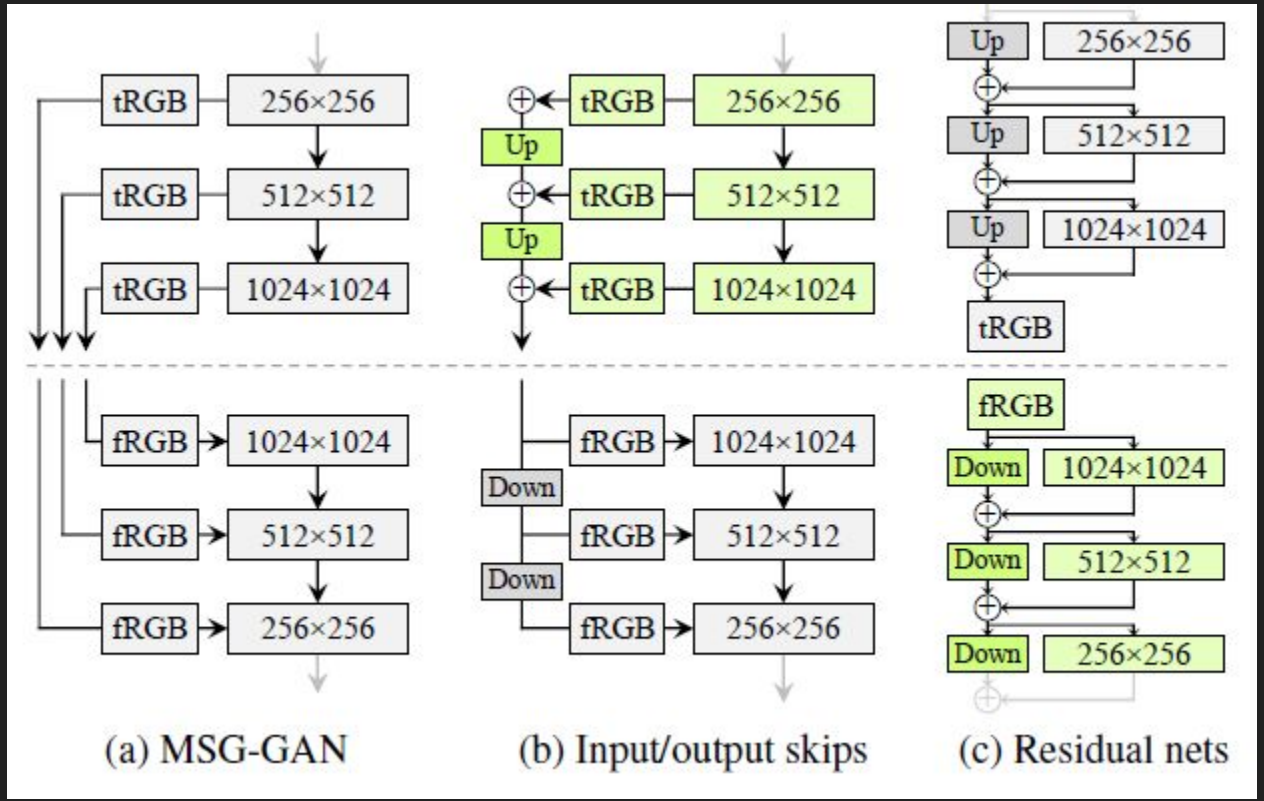


Image from original paper

E. No Growing, New G & D Architecture

FFHQ	D original		D input skips		D residual	
	FID	PPL	FID	PPL	FID	PPL
G original	4.32	265	4.18	235	3.58	269
G output skips	4.33	169	3.77	127	**3.31**	**125**
G residual	4.35	203	3.96	229	3.79	243

LSUN Car	D original		D input skips		D residual	
	FID	PPL	FID	PPL	FID	PPL
G original	3.75	905	3.23	758	3.25	802
G output skips	3.77	544	3.86	**316**	3.19	471
G residual	3.93	981	3.40	667	**2.66**	645

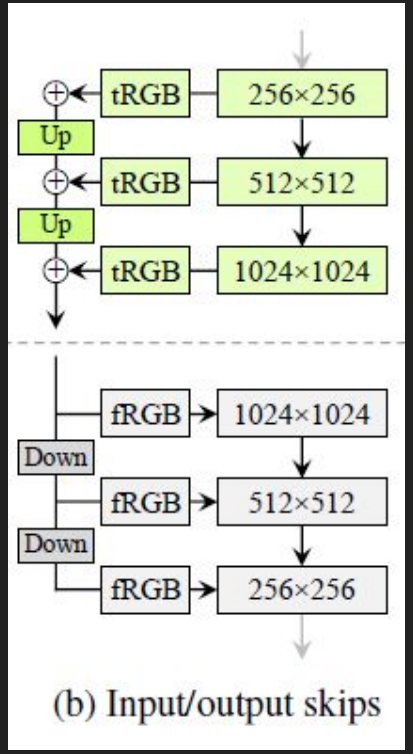
Table 2. Comparison of generator and discriminator architectures without progressive growing. The combination of generator with output skips and residual discriminator corresponds to configuration E in the main result table.

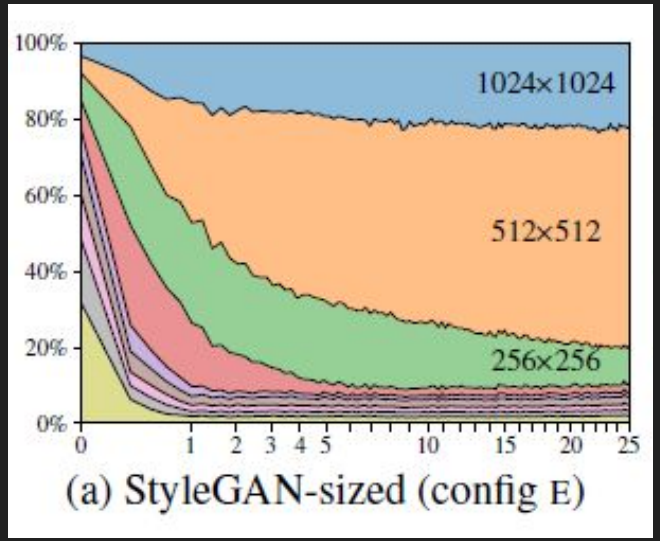
Image from original paper

The Result of New Architecture

youtube.com/watch?v=c-NJtV9Jvp0

F. Large Networks for Better Performance



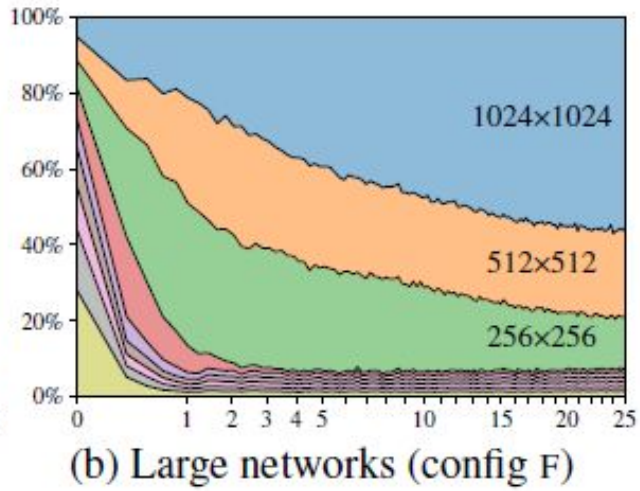


… To verify this, we inspected the generated images manually and noticed that they generally lack some of the pixel-level detail...

Image from original paper

F. Large Networks for Better Performance

[4]We double the number of feature maps in resolutions 64^2–1024^2 while keeping other parts of the networks unchanged. This increases the total number of trainable parameters in the generator by 22% (25M → 30M) and in the discriminator by 21% (24M → 29M).



(b) Large networks (config F)

Image from original paper

Deep Fake Detection Algorithm

- How can we detect generated / synthesized image?

StyleGAN — generated images | StyleGAN2 — generated images | StyleGAN2 — real images

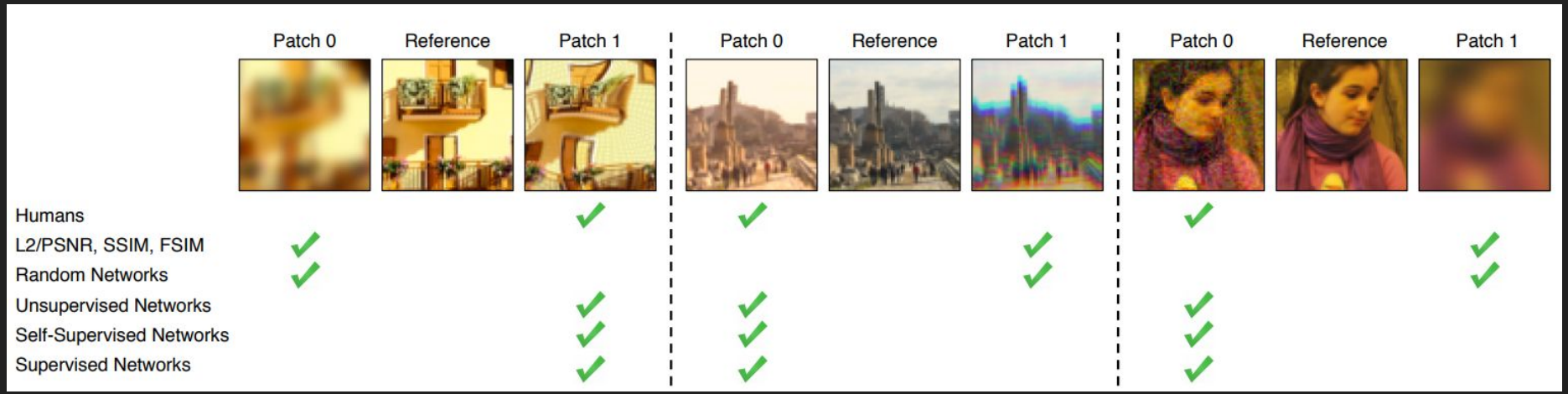
Figure 9. Example images and their projected and re-synthesized counterparts. For each configuration, top row shows the target images and bottom row shows the synthesis of the corresponding projected latent vector and noise inputs. With the baseline StyleGAN, projection often finds a reasonably close match for generated images, but especially the backgrounds differ from the originals. The images generated using StyleGAN2 can be projected almost perfectly back into generator inputs, while projected real images (from the training set) show clear differences to the originals, as expected. All tests were done using the same projection method and hyperparameters.

Image from original paper

Deep Fake Detection Algorithm

youtube.com/watch?v=c-NJtV9Jvp0

How to do this? LPIPS distance



Zhang, R., Isola, P., Efros, A. A., Shechtman, E., & Wang, O. (2018). The unreasonable effectiveness of deep features as a perceptual metric. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 586-595).

How to do this?

- (for StyleGAN) it appears that the mapping from W to images is **too complex for this to succeed reliably in practice.**

- We find it encouraging that StyleGAN2 makes source attribution easier even though the image quality has improved significantly.

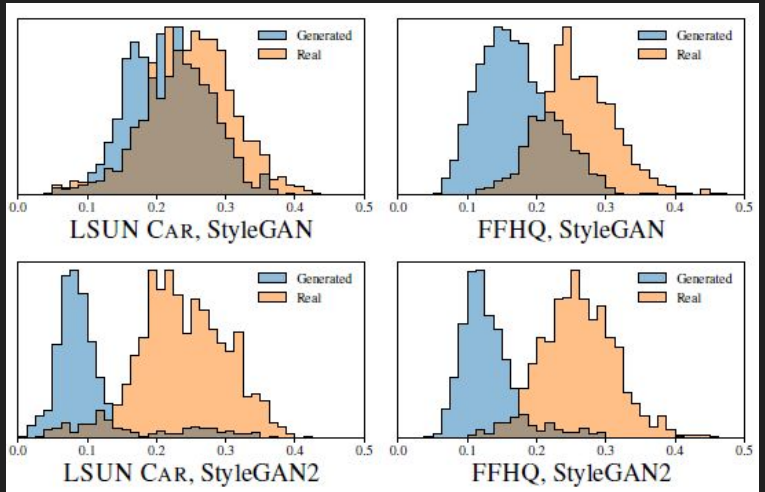


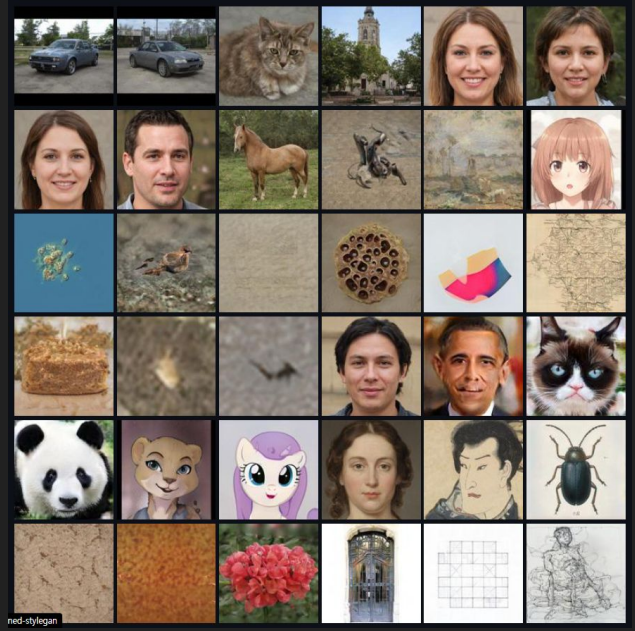
Figure 10. LPIPS distance histograms between original and projected images for generated (blue) and real images (orange). Despite the higher image quality of our improved generator, it is much easier to project the generated images into its latent space $\mathcal{W}$. The same projection method was used in all cases.

Image from original paper

Many pre-trained models exist

A collection of pre-trained StyleGAN2 models trained on different datasets at different resolution.

See this repo for pretrained models for StyleGAN 1



- CIFAR 10
- CIFAR 100
- faces (FFHQ slim 256x256)
- obama
- grumpy cat
- panda
- fursona
- my little pony
- painting faces
- ukiyoe faces
- beetles
- textures
- more abstract art
- flowers
- Doors
- floor plans
- figure drawings

- cat
- church
- faces (FFHQ config-e)
- faces (FFHQ config-e 256x256)
- faces (FFHQ config-f)
- faces (FFHQ config-f 512x512)
- horse
- Imagenet
- WikiArt
- Anime portraits
- microscope images
- wildlife
- modern art
- trypophobia
- Abstract art
- Maps
- cakes

https://github.com/justinpinkney/awesome-pretrained-stylegan2

Outro

Item	GPU years (Volta)	Electricity (MWh)
Initial exploration	20.25	58.94
Paper exploration	13.71	31.49
FFHQ config F	0.23	0.68
Other runs in paper	7.20	16.77
Backup runs left out	4.73	12.08
Video, figures, etc.	0.31	0.82
Public release	4.62	10.82
Total	51.05	131.61

Table 5. Computational effort expenditure and electricity consumption data for this project. The unit for computation is GPU-years on a single NVIDIA V100 GPU — it would have taken approximately 51 years to execute this project using a single GPU.

Performance optimizations We profiled our training runs extensively and found that — in our case — the default primitives for image filtering, up/downsampling, bias addition, and leaky ReLU had surprisingly high overheads in terms of training time and GPU memory footprint. This motivated us to optimize these operations using hand-written CUDA kernels. We implemented filtered up/downsampling as a single fused operation, and bias and activation as another one. In configuration E at 1024^2 resolution, our optimizations improved the overall training time by about 30% and memory footprint by about 20%.

Q & A