A Style-Based Generator Architecture for GAN

Tero Karras, Samuli Laine, and Timo Aila from NVIDIA

Compiled by Hongming Shan

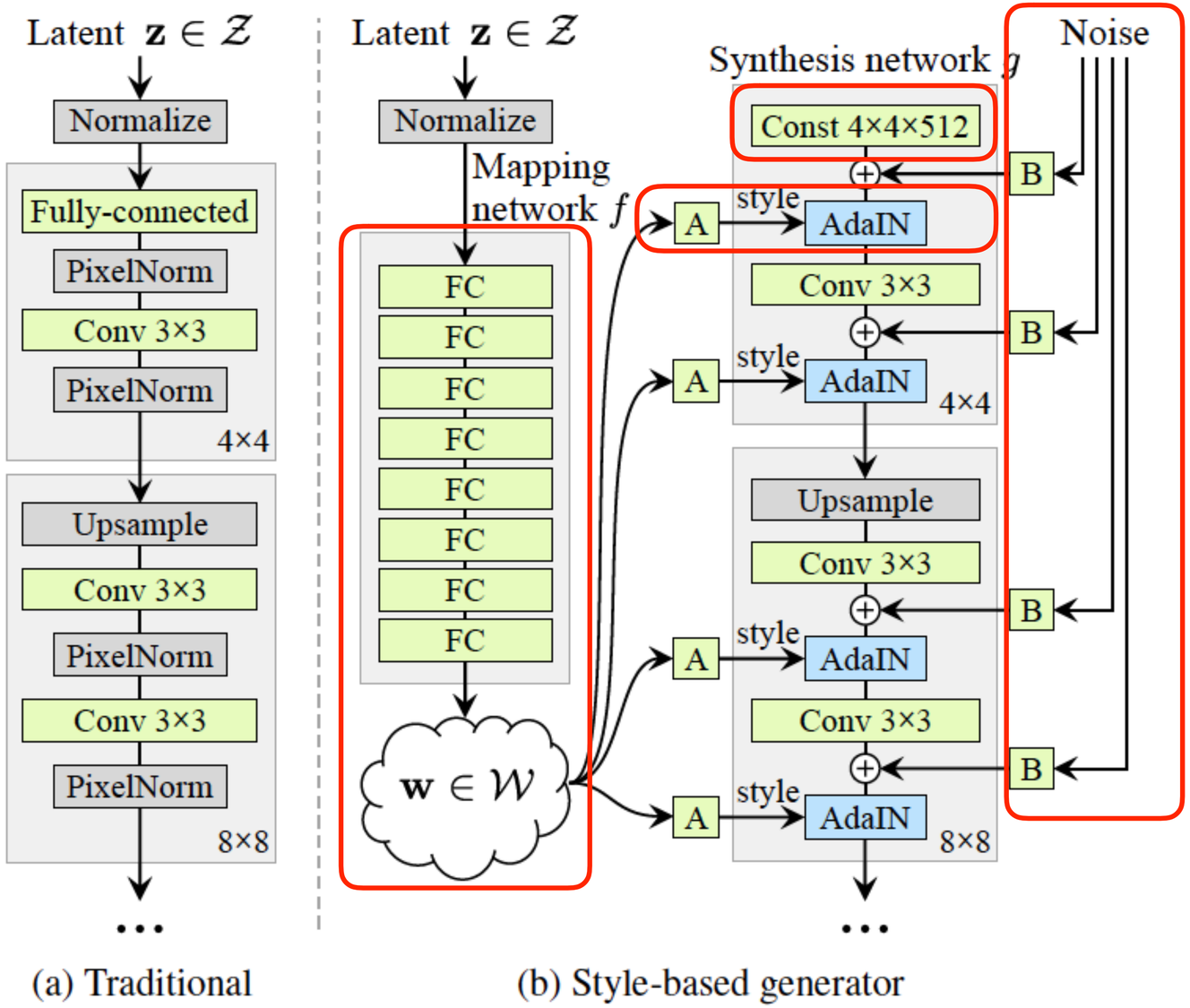
Video

Contributions

- We propose an **alternative generator architecture** for generative adversarial networks, borrowing from style transfer literature.
- The new architecture leads to an **automatically learned, unsupervised separation** of **high-level attributes** (e.g., pose and identity when trained on human faces) and **stochastic variation** in the generated images (e.g., freckles, hair), and it enables **intuitive, scale-specific** control of the synthesis.
- The new generator improves the state-of-the-art in terms of traditional distribution quality metrics, leads to demonstrably **better interpolation properties**, and also better **disentangles the latent factors of variation**.
- To **quantify interpolation quality and disentanglement**, we propose two new, automated methods that are applicable to any generator architecture.
- Finally, we introduce **a new, highly varied and high-quality dataset** of human faces.

Style-based Generator

Network Structure



(a) Traditional (b) Style-based generator

- We first map the input to an intermediate latent space W, which then controls the generator through adaptive instance normalization (AdaIN) at each convolution layer.
- Gaussian noise is added after each convolution, before evaluating the nonlinearity.
- “A” stands for a learned affine transform
- “B” applies learned per-channel scaling factors to the noise input.
- The output of the last layer is converted to RGB using a separate 1x1 convolution

Style input and Noise input

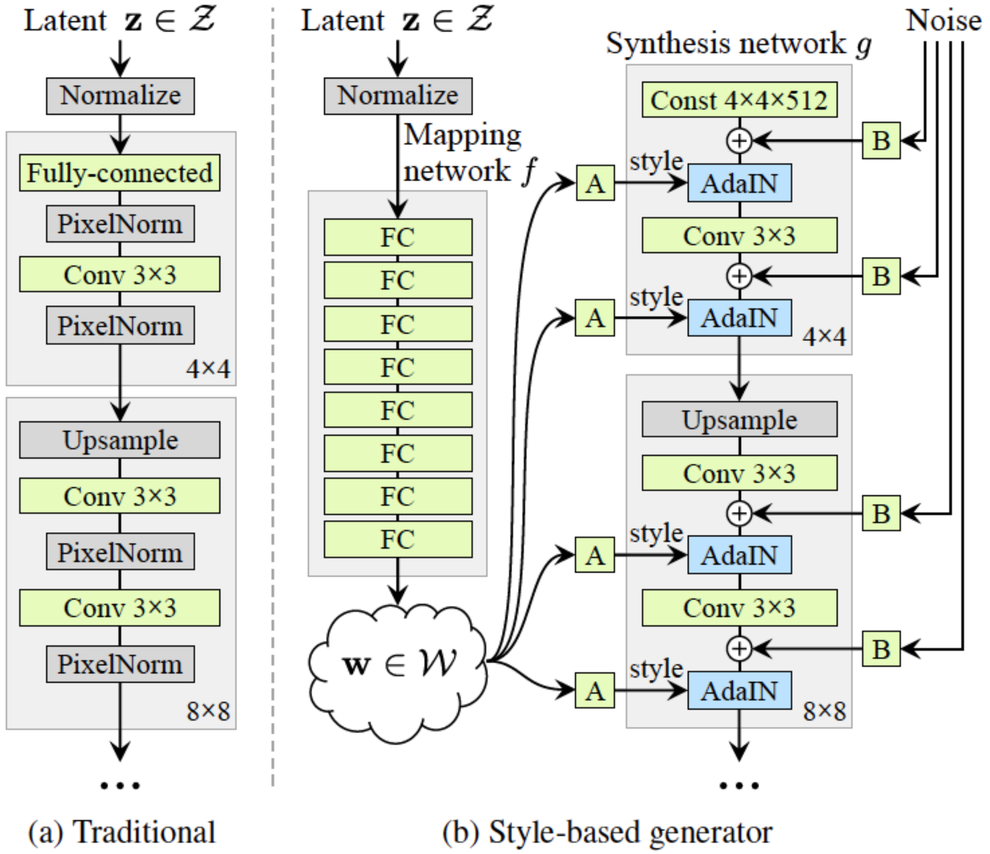
- **Learned affine transformations** then specialize *w* to styles *y = (y_s; y_b)* that control **adaptive instance normalization (AdaIN)** operations after each convolution layer of the synthesis network *g*

- **Adaptive instance normalization**:

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

- Noise input: single-channel images consisting of **uncorrelated Gaussian noise**, and we feed a dedicated noise image to each layer of the synthesis network. The noise image **is broadcasted to all feature maps** using **learned per-feature scaling factors** and then added to the output of the corresponding convolution

Quality of generated images



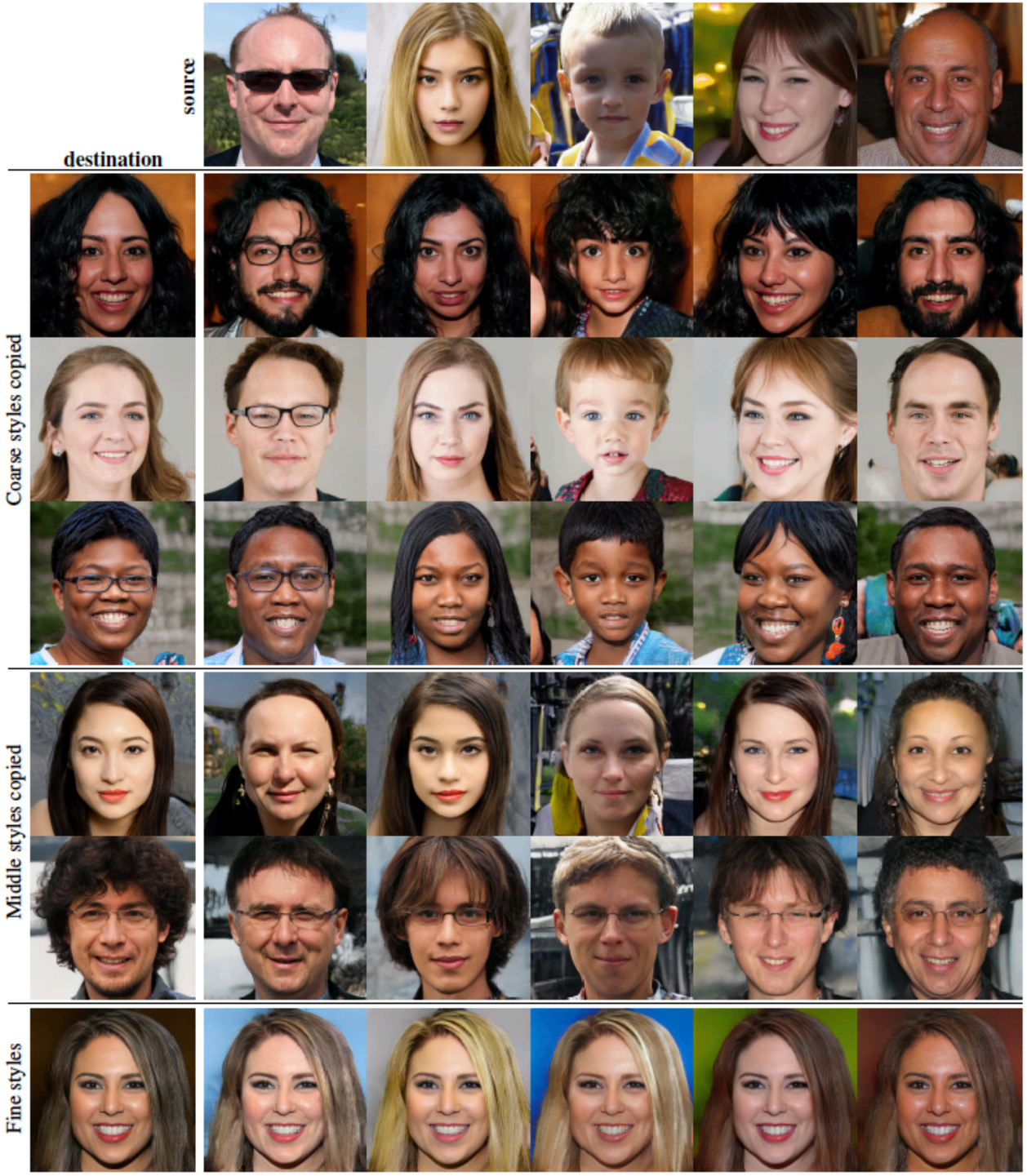
(a) Traditional (b) Style-based generator

	Method	CelebA-HQ	FFHQ
A	Baseline Progressive GAN [29]	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.

Style mixing regularization

- To further encourage the styles to localize, we employ mixing regularization , where a given percentage of images are generated using **two random latent codes** instead of **one** during training.
- When generating such an image, we **simply switch from one latent code to another**—an operation we refer to as style mixing —at a randomly selected point in the synthesis network.
- two latent codes z1, z2 through the mapping network, and have the corresponding w1,w2 control the styles so that w1 applies before the crossover point and w2 after it.
- This regularization technique prevents the network from assuming that adjacent styles are correlated.



- Visualizing the effect of styles in the generator by having the styles produced by one latent code (source) override a subset of the styles of another one (destination).

- Overriding the styles of layers corresponding to **coarse spatial resolutions** (4^2 – 8^2), high-level aspects such as pose, general hair style, face shape, and eyeglasses get copied from the source, while all colors (eyes, hair, lighting) and finer facial features of the destination are retained.

- If we instead copy **the styles of middle layers** (16^2 – 32^2), we inherit smaller scale facial features, hair style, eyes open/closed from the source, while the pose, general face shape, and eyeglasses from the destination are preserved.

- Finally, copying **the styles corresponding to fine resolutions** (64^2 – 1024^2) brings mainly the color scheme and microstructure from the source.

Mixing regularization

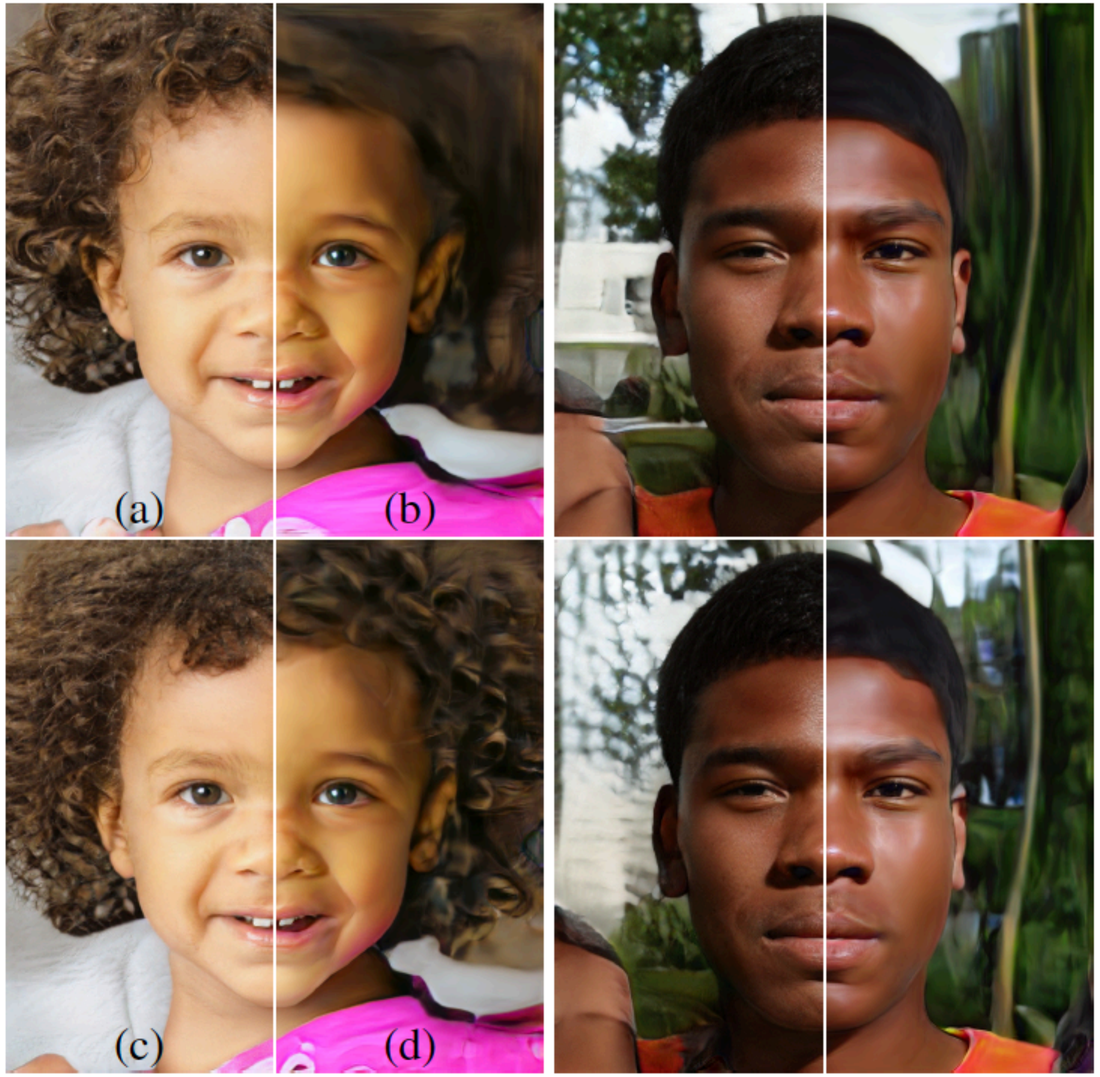
	Mixing regularization	Number of latents during testing 1	2	3	4
E	0%	4.42	8.22	12.88	17.41
	50%	4.41	6.10	8.71	11.61
F	90%	**4.40**	**5.11**	6.88	9.03
	100%	4.83	5.17	**6.63**	**8.40**

Table 2. FIDs in FFHQ for networks trained by enabling the mixing regularization for different percentage of training examples. Here we stress test the trained networks by randomizing 1 . . . 4 latents and the crossover points between them. Mixing regularization improves the tolerance to these adverse operations significantly. Labels E and F refer to the configurations in Table 1.

Stochastic variation

(a) Generated image (b) Stochastic variation (c) Standard deviation

- Examples of stochastic variation.
- (a) Two generated images.
- (b) Zoom-in with different realizations of input noise. While the overall appearance is almost identical, individual hairs are placed very differently.
- (c) Standard deviation of each pixel over 100 different realizations, highlighting which parts of the images are affected by the noise.
- The main areas are the hair, silhouettes, and parts of background, but there is also interesting stochastic variation in the eye reflections.
- Global aspects such as identity and pose are unaffected by stochastic variation.

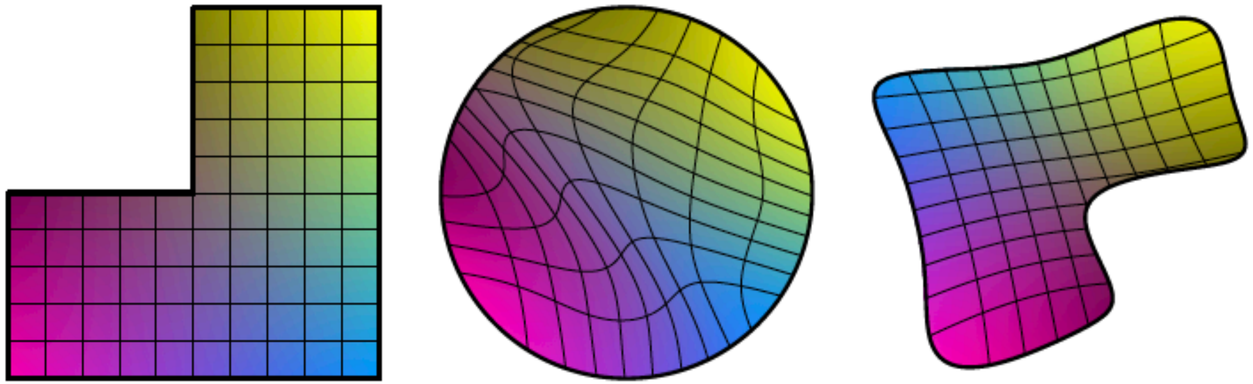


- Effect of noise inputs at different layers of our generator.
- (a) Noise is applied to all layers.
- (b) No noise.
- (c) Noise in fine layers only (64^2 – 1024^2).
- (d) Noise in coarse layers only (4^2 – 32^2).
- We can see that the artificial omission of noise leads to featureless “painterly” look. Coarse noise causes large-scale curling of hair and appearance of larger background features, while the fine noise brings out the finer curls of hair, finer background detail, and skin pores.

Disentanglement studies

- A common goal of disentanglement is a latent space that consists of linear subspaces, each of which controls one factor of variation.

- However, the sampling probability of each combination of factors in Z needs to match the corresponding density in the training data.



(a) Distribution of features in training set

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

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

- 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 virtually non-existent.
- (b) This forces the mapping from Z to image features to become curved so that the forbidden combination disappears in Z to prevent the sampling of invalid combinations.
- (c) However, the learned mapping from Z to W is able to “undo” much of the warping.

Quantifying disentanglement

- Perceptual path length
  - Calculate perceptual distance over a subdivision (linear segment)
- Linear separation

Style scale (Truncation trick)

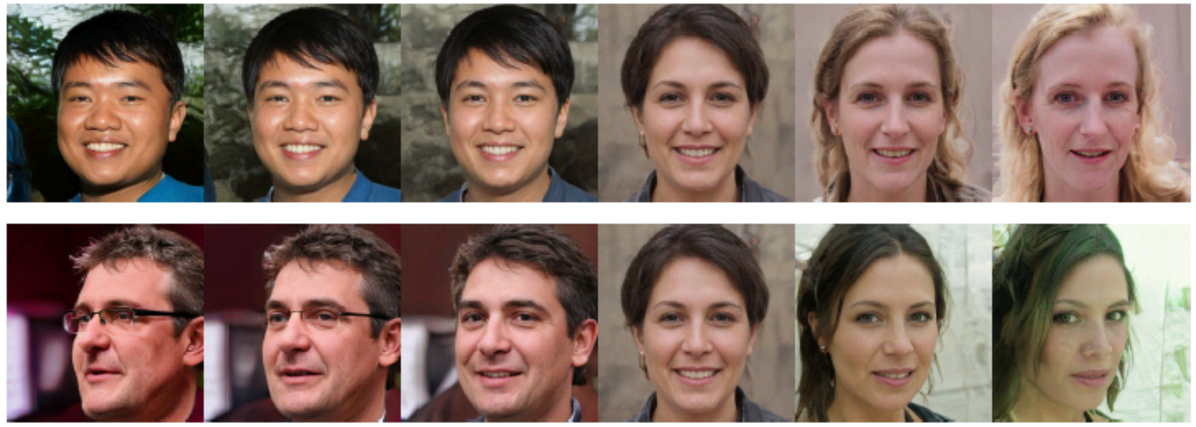


Figure 8. The effect of truncation trick as a function of style scale ψ. When we fade $\psi \to 0$, all faces converge to the "mean" face of FFHQ. This face is similar for all trained networks, and the interpolation towards it never seems to cause artifacts. By applying negative scaling to styles, we get the corresponding opposite or "anti-face". It is interesting that various high-level attributes often flip between the opposites, including viewpoint, glasses, age, coloring, hair length, and often gender.

We can follow a similar strategy. To begin, we compute the center of mass of $\mathcal{W}$ as $\bar{\mathbf{w}} = \mathbb{E}_{\mathbf{z} \sim P(\mathbf{z})}[f(\mathbf{z})]$. In case of FFHQ this point represents a sort of an average face (Figure 8, $\psi = 0$). We can then scale the deviation of a given $\mathbf{w}$ from the center as $\mathbf{w}' = \bar{\mathbf{w}} + \psi(\mathbf{w} - \bar{\mathbf{w}})$, where $\psi < 1$. While Brock et al. [5] observe that only a subset of networks is amenable to such truncation even when orthogonal regularization is used, truncation in $\mathcal{W}$ space seems to work reliably even without changes to the loss function.

Summary

- It is becoming clear that the traditional GAN generator architecture is in every way inferior to a style-based design.

- Our investigations to the separation of high-level attributes and stochastic effects, as well as the linearity of the intermediate latent space will prove fruitful in improving the understanding and controllability of GAN synthesis.