2024.11.15

arXiv > cs > arXiv:1812.04948

Search...

Help

Computer Science > Neural and Evolutionary Computing

[Submitted on 12 Dec 2018 (v1), last revised 29 Mar 2019 (this version, v3)]

A Style-Based Generator Architecture for Generative Adversarial Networks

Tero Karras, Samuli Laine, Timo Aila

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.

Comments: CVPR 2019 final version
Subjects: **Neural and Evolutionary Computing (cs.NE)**; Machine Learning (cs.LG); Machine Learning (stat.ML)
Cite as: arXiv:1812.04948 **[cs.NE]**
(or arXiv:1812.04948v3 **[cs.NE]** for this version)
https://doi.org/10.48550/arXiv.1812.04948

Submission history

From: Samuli Laine [view email]
[v1] Wed, 12 Dec 2018 13:59:43 UTC (22,070 KB)
[v2] Wed, 6 Feb 2019 14:58:00 UTC (22,070 KB)
[v3] Fri, 29 Mar 2019 11:08:46 UTC (22,071 KB)

Style GAN; disentanglement ; Style mixing

Yeon Su Park

Pukyong National University
Division of Computer Engineering And Artificial Intelligence
Medical AI Lab.

Introduction

Problem of exisiting model

- The resolution and quality of images produced by generative methods especially generative adversarial networks (GAN) have seen rapid improvement recently.

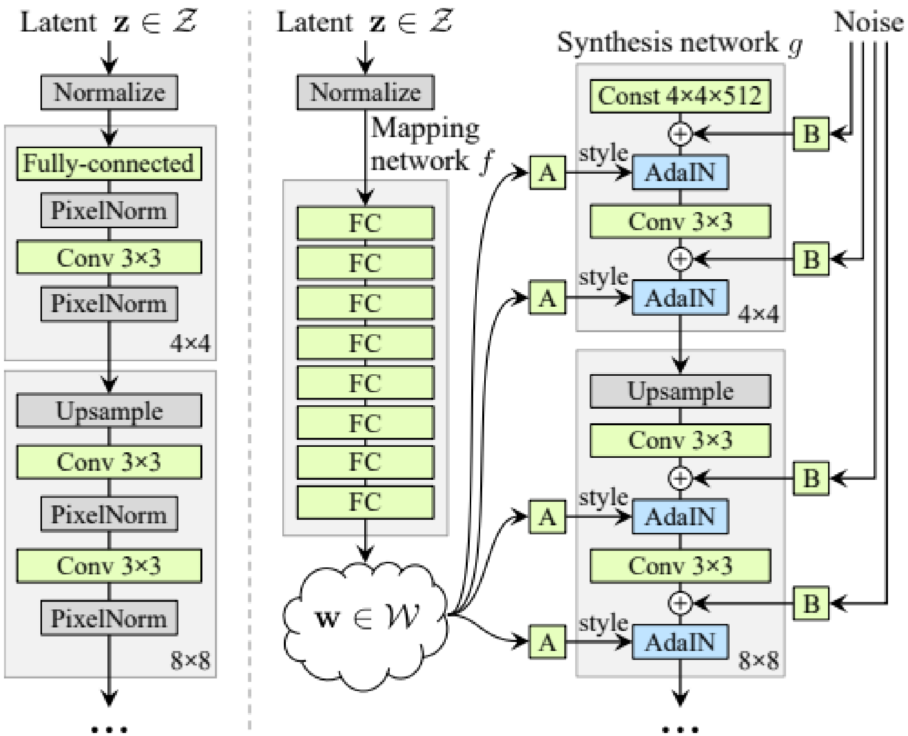
- The properties of the latent space are also poorly understood

- the commonly demonstrated latent space interpolations provide no quantitative way to compare different generators against each other

- In this paper, Motivated by style transfer literature, we re-design the generator architecture in a way that exposes novel ways to control the image synthesis process.

Style-based generator

Structure of the StyleGAN



(a) Traditional (b) Style-based generator

Figure 1. While a traditional generator [30] feeds the latent code though the input layer only, we first map the input to an intermediate latent space $\mathcal{W}$, 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. Here "A" stands for a learned affine transform, and "B" applies learned per-channel scaling factors to the noise input. The mapping network f consists of 8 layers and the synthesis network g consists of 18 layers — two for each resolution ($4^2 - 1024^2$). The output of the last layer is converted to RGB using a separate 1×1 convolution, similar to Karras et al. [30]. Our generator has a total of 26.2M trainable parameters, compared to 23.1M in the traditional generator.

$$\mathrm{AdaIN}(\mathbf{x}_i, \mathbf{y}) = \mathbf{y}_{s,i} \frac{\mathbf{x}_i - \mu(\mathbf{x}_i)}{\sigma(\mathbf{x}_i)} + \mathbf{y}_{b,i}, \qquad (1)$$

$$\mathrm{AdaIN}(x, y) = \sigma(y)\left(\frac{x - \mu(x)}{\sigma(x)}\right) + \mu(y)$$

Style-based generator

Structure of the StyleGAN

- **AdaIN**
  - each feature map x_i is normalized separately, and then scaled and biased using the corresponding scalar components from style y
  - AdaIN is particularly well suited for our purposes due to its efficiency and compact representation.
  - w => affine transform = y = (yb, ys)

- **Noise inputs**
  - to generate stochastic detail
  - single-channel images consisting of uncorrelated Gaussian noise.
  - a dedicated noise image are fed to each layer of the synthesis network.

Properties of the style-based generator

Style mixing

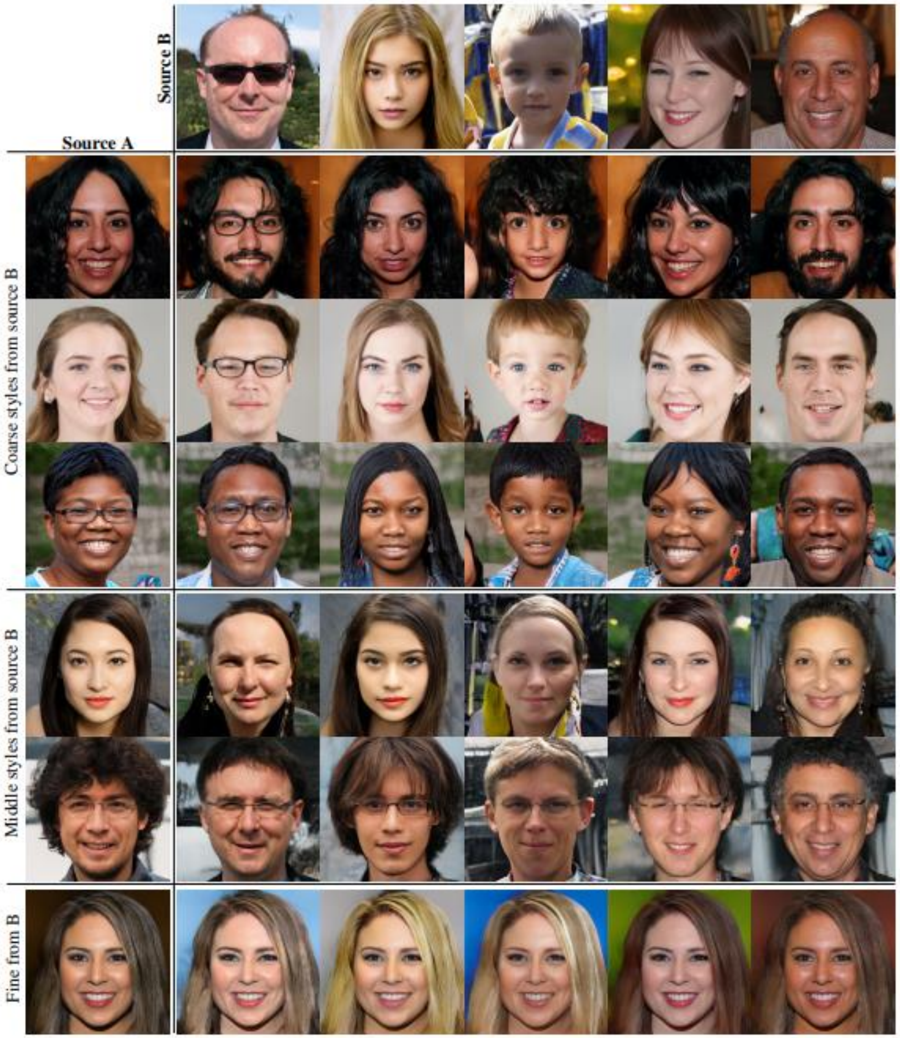


Figure 3. Two sets of images were generated from their respective latent codes (sources A and B); the rest of the images were generated by copying a specified subset of styles from source B and taking the rest from source A. Copying the styles corresponding to coarse spatial resolutions ($4^2 - 8^2$) brings high-level aspects such as pose, general hair style, face shape, and eyeglasses from source B, while all colors (eyes, hair, lighting) and finer facial features resemble A. If we instead copy the styles of middle resolutions ($16^2 - 32^2$) from B, we inherit smaller scale facial features, hair style, eyes open/closed from B, while the pose, general face shape, and eyeglasses from A are preserved. Finally, copying the fine styles ($64^2 - 1024^2$) from B brings mainly the color scheme and microstructure.

- mixing regularization
  - Introduced to further encourage the styles to localize.
  - 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.

- Coarse - high-level aspects
- Middle - we inherit
- smaller scale facial features
- Fine - mainly the color scheme and microstructure

Part 3,

Properties of the style-based generator

Stochastic variation

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

Figure 4. 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.

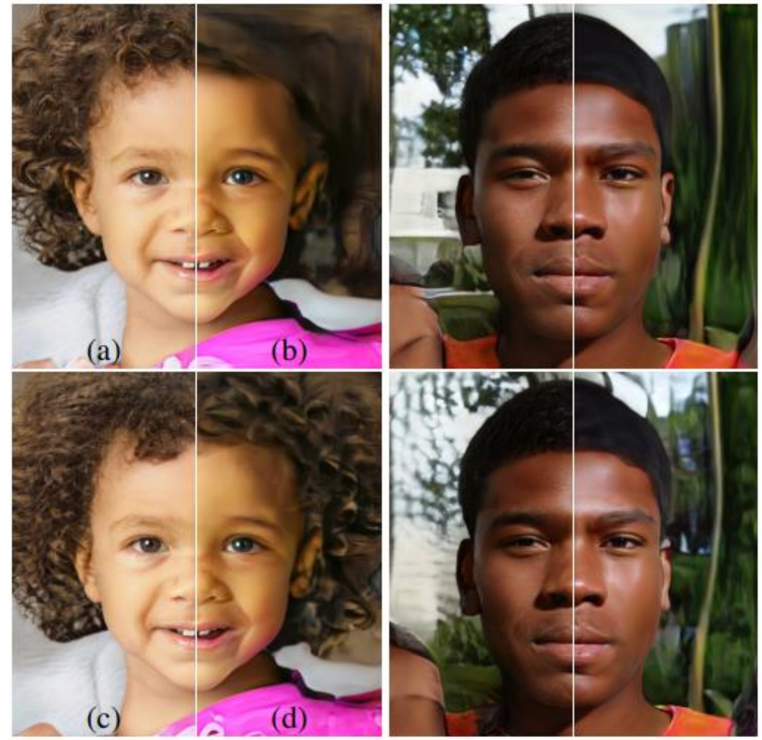


Figure 5. Effect of noise inputs at different layers of our generator. (a) Noise is applied to all layers. (b) No noise. (c) Noise in fine layers only (64^2 – 1024^2). (d) Noise in coarse layers only (4^2 – 32^2). We can see that the artificial omission of noise leads to featureless "painterly" look. 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.

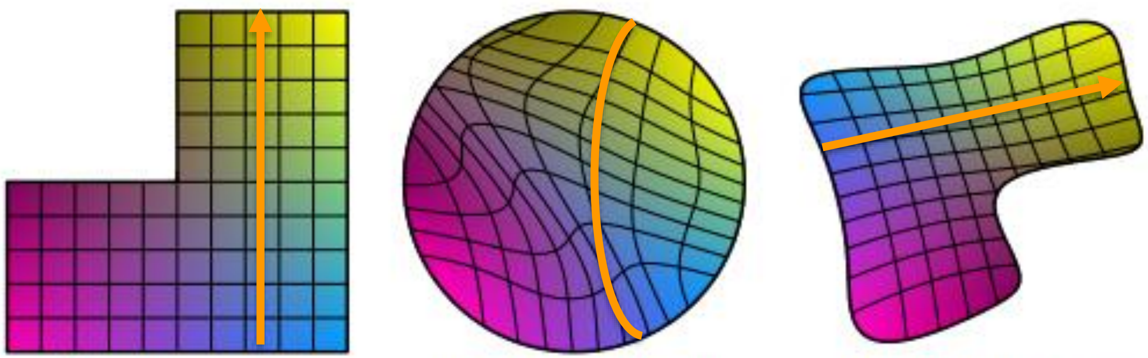
Properties of the style-based generator

Separation of global effects from stochasticity

- the style affects the entire image because complete feature maps are scaled and biased with the same values.
- Therefore, global effects such as pose, lighting, or background style can be controlled coherently.
- Meanwhile, the noise is added independently to each pixel and is thus ideally suited for controlling stochastic variation.
- If the network tried to control, e.g., pose using the noise, that would lead to spatially inconsistent decisions that would then be penalized by the discriminator. Thus the network learns to use the global and local channels appropriately, without explicit guidance.

Part 4,
Disentanglement studies
Perceptual path length

- Disentanglement
  - a latent space that consists of linear subspaces, each of which controls one factor of variation

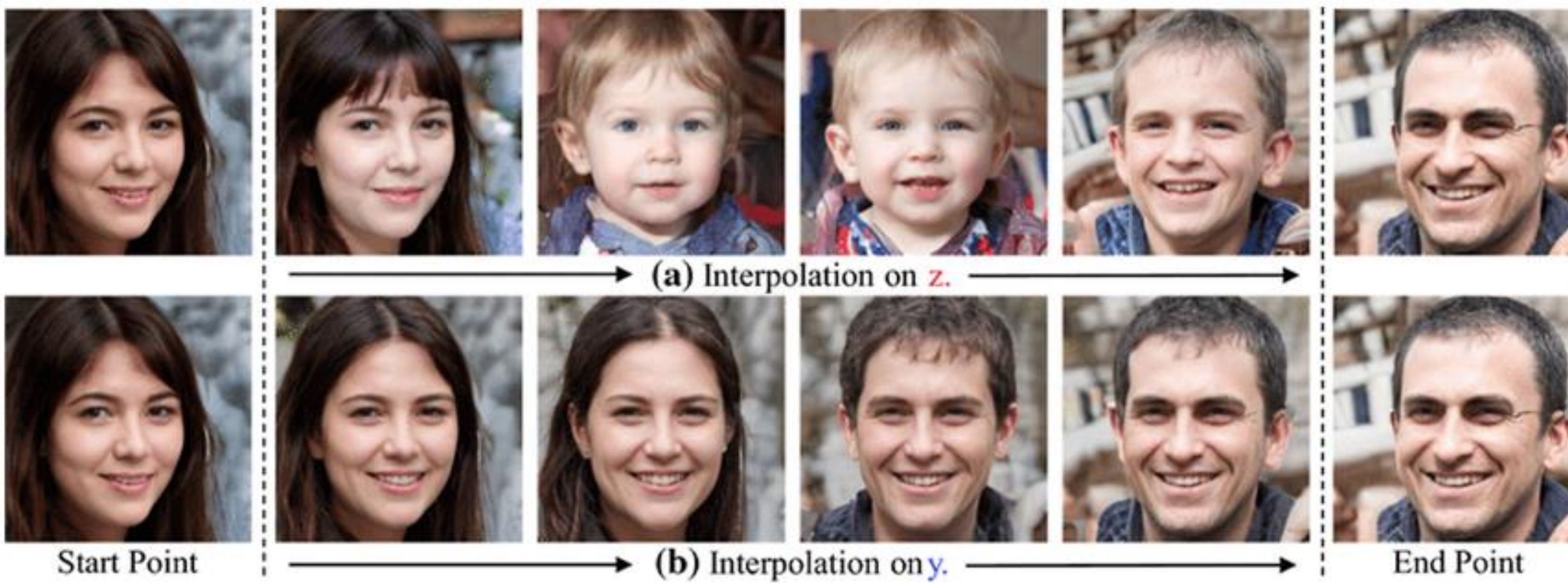


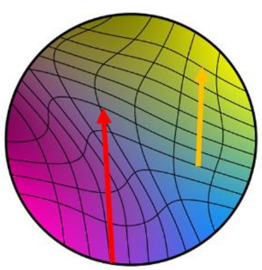
(a) Distribution of features in training set (b) Mapping from $\mathcal{Z}$ to features (c) Mapping from $\mathcal{W}$ to features

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.

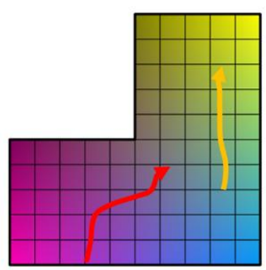
Disentanglement studies

Perceptual path length





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



(a) Distribution of features in training set

$$l_{\mathcal{Z}} = \mathbb{E}\left[\frac{1}{\epsilon^2} d\big(G(\mathrm{slerp}(\mathbf{z}_1, \mathbf{z}_2;\, t)),\; G(\mathrm{slerp}(\mathbf{z}_1, \mathbf{z}_2;\, t+\epsilon))\big)\right]$$

$$l_{\mathcal{W}} = \mathbb{E}\left[\frac{1}{\epsilon^2} d\big(g(\mathrm{lerp}(f(\mathbf{z}_1), f(\mathbf{z}_2);\, t)),\; g(\mathrm{lerp}(f(\mathbf{z}_1), f(\mathbf{z}_2);\, t+\epsilon))\big)\right]$$

Disentanglement studies

Linear separability

- If a latent space is sufficiently disentangled, it should be possible to find direction vectors that consistently correspond to individual factors of variation.
- W is consistently better separable than Z, suggesting a less entangled representation.

Method		Path length full	Path length end	Separa-bility
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.

Method		FID	Path length full	Path length end	Separa-bility
B Traditional 0	$\mathcal{Z}$	5.25	412.0	415.3	10.78
Traditional 8	$\mathcal{Z}$	4.87	896.2	902.0	170.29
Traditional 8	$\mathcal{W}$	4.87	324.5	212.2	6.52
Style-based 0	$\mathcal{Z}$	5.06	283.5	285.5	9.88
Style-based 1	$\mathcal{W}$	4.60	219.9	209.4	6.81
Style-based 2	$\mathcal{W}$	4.43	**217.8**	199.9	6.25
F Style-based 8	$\mathcal{W}$	**4.40**	234.0	**195.9**	**3.79**

Table 4. The effect of a mapping network in FFHQ. The number in method name indicates the depth of the mapping network. We see that FID, separability, and path length all benefit from having a mapping network, and this holds for both style-based and traditional generator architectures. Furthermore, a deeper mapping network generally performs better than a shallow one.

Conclusion

- **Superiority of Style-Based Designs:**
  - styleGAN is better than traditional GAN generator architecture is in every way

- **Seperation and linearity :**
  - the separation of high-level attributes and stochastic effects within the model.
  - They believe that studying the linearity of the intermediate latent space contributes to a better understanding and enhanced control over GAN synthesis.

- **Potential for Training Regularization :**
  - We note that our average path length metric could easily be used as a regularizer during training

- **Future work :**
  - In general, we expect that methods for directly shaping the intermediate latent space during training will provide interesting avenues for future work.