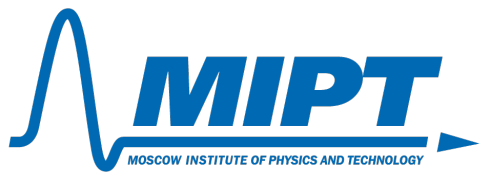
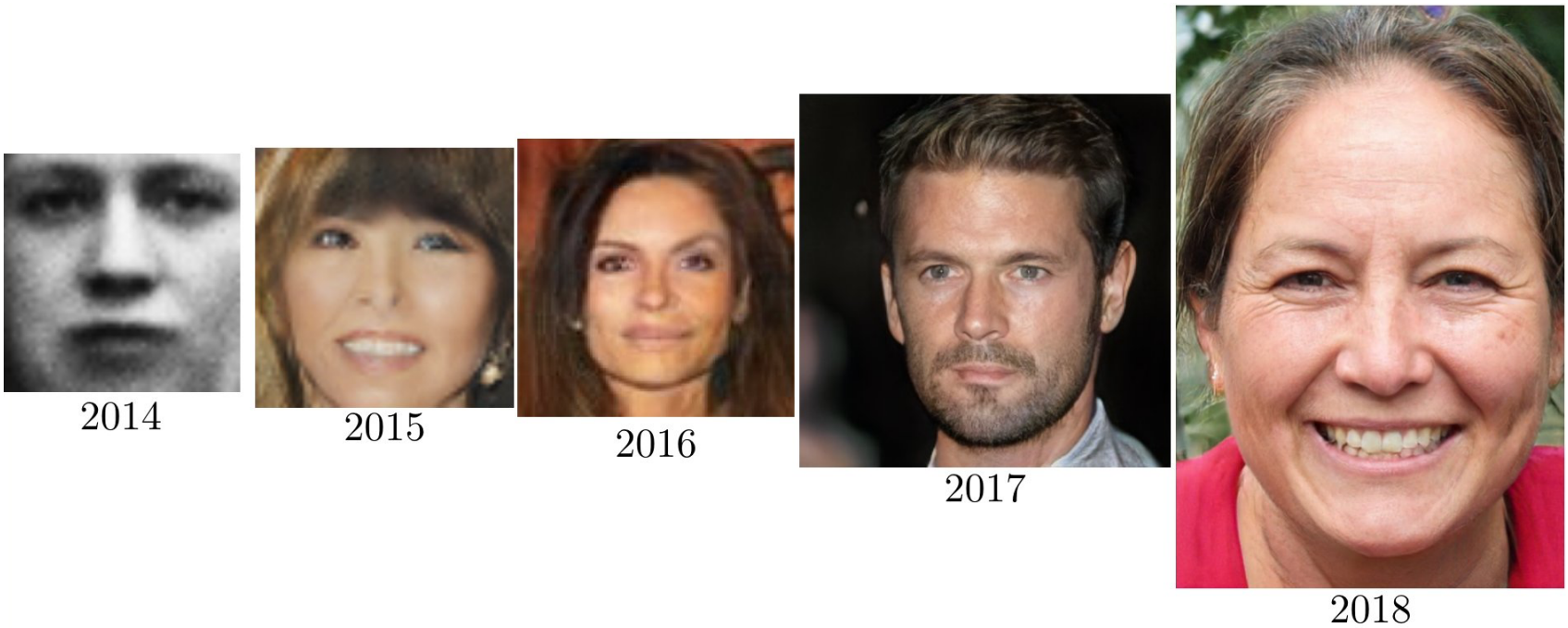
Deep Generative Models
GAN milestones

Petr Mokrov (slides by Roman Isachenko)



Autumn, 2023

Evolution of GANs



- **Standard GAN** https://arxiv.org/abs/1406.2661
- **DCGAN** https://arxiv.org/abs/1511.06434
- **CoGAN** https://arxiv.org/abs/1606.07536
- **ProGAN** https://arxiv.org/abs/1710.10196
- **StyleGAN** https://arxiv.org/abs/1812.04948

Evolution of GANs

2019 2021

- **StyleGAN2** https://arxiv.org/abs/1912.04958
- **StyleGAN3** https://nvlabs.github.io/stylegan3/

Self-Attention GAN

- Convolutional layers process the information in a local neighborhood.
- Using convolutional layers alone is computationally inefficient for modeling long-range dependencies in images.

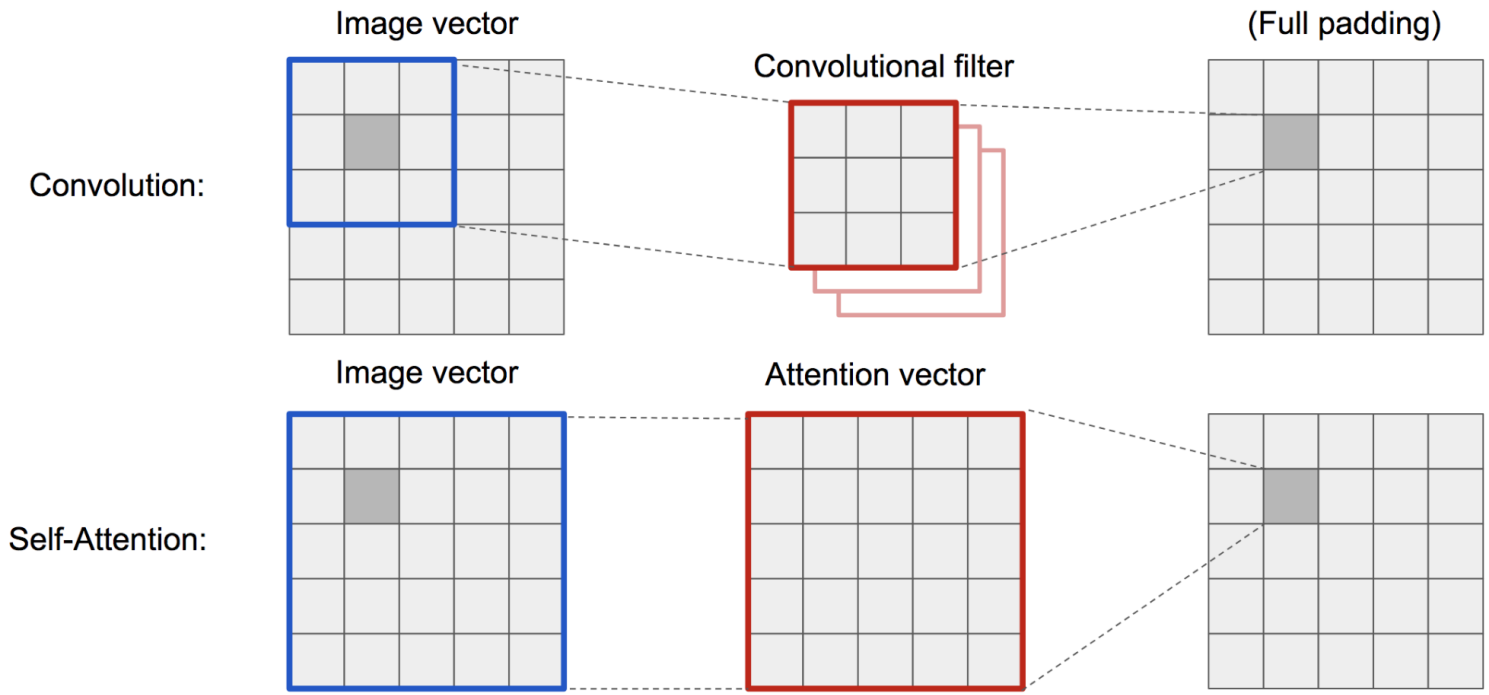
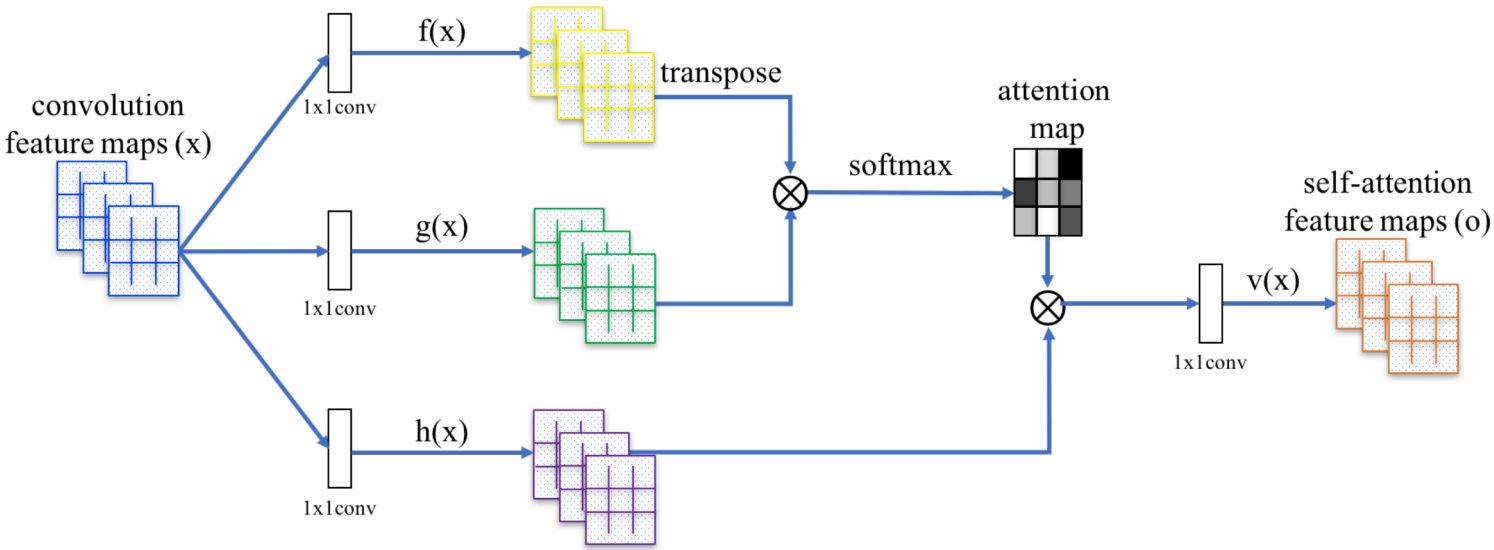


image credit: https://lilianweng.github.io/lil-log/2018/06/24/attention-attention.html

Self-Attention GAN



- $\mathbf{x}$ – feature vector for one feature location.
- N – number of feature locations.

$$\mathbf{f}(\mathbf{x}) = \mathbf{W}_f\mathbf{x}, \quad \mathbf{g}(\mathbf{x}) = \mathbf{W}_g\mathbf{x}, \quad \mathbf{h}(\mathbf{x}) = \mathbf{W}_h\mathbf{x}, \quad \mathbf{v}(\mathbf{x}) = \mathbf{W}_v\mathbf{x}$$

$$s_{ij} = \mathbf{f}(\mathbf{x}_i)^T\mathbf{g}(\mathbf{x}_j), \quad a_{ij} = \frac{\exp s_{ij}}{\sum_{i=1}^{N} \exp s_{ij}}, \quad \mathbf{o}_j = \mathbf{v}\left(\sum_{i=1}^{N} a_{ij}\mathbf{h}(\mathbf{x}_i)\right)$$

Zhang H. et al. Self-Attention Generative Adversarial Networks, 2018

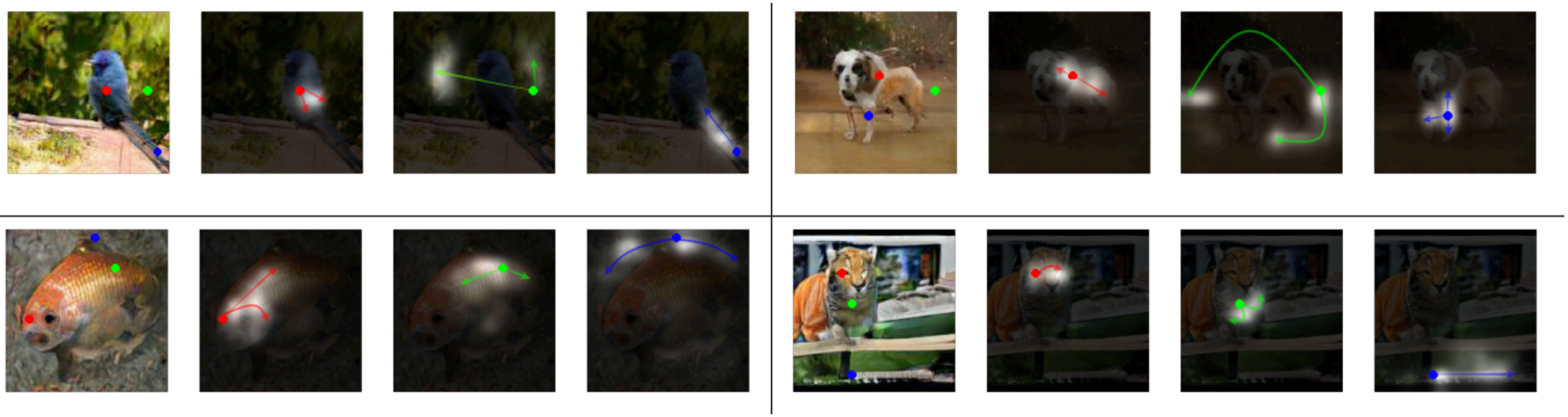
Self-Attention GAN

Technical Details

- Hinge loss for training.
- SpectralNorm in both the generator and the discriminator.
- Separate learning rates for the generator and the discriminator.

Model	Inception Score	Intra FID	FID
AC-GAN (Odena et al., 2017)	28.5	260.0	/
SNGAN-projection (Miyato & Koyama, 2018)	36.8	92.4	27.62*
SAGAN	**52.52**	**83.7**	**18.65**

Visualization of attention maps



Zhang H. et al. Self-Attention Generative Adversarial Networks, 2018

BigGAN

Technical Details

- Hinge loss.
- Self-Attention GAN baseline.
- **Orthogonal regularization**

$$\|\mathbf{W}^T\mathbf{W} - \mathbf{I}\|^2 \quad \Rightarrow \quad \|\mathbf{W}^T\mathbf{W} - \text{diag}(\mathbf{W}^T\mathbf{W})\|^2$$

- **Truncation trick.** Components of $\mathbf{z} \sim \mathcal{N}(0, \mathbf{I})$ which fall outside a predefined range are resampled.

Batch	Ch.	Param (M)	Shared	Skip-z	Ortho.	Itr $\times 10^3$	FID	IS
256	64	81.5	SA-GAN Baseline			1000	18.65	52.52
512	64	81.5	✗	✗	✗	1000	15.30	58.77($\pm$1.18)
1024	64	81.5	✗	✗	✗	1000	14.88	63.03($\pm$1.42)
2048	64	81.5	✗	✗	✗	732	12.39	76.85($\pm$3.83)
2048	96	173.5	✗	✗	✗	295($\pm$18)	9.54($\pm$0.62)	92.98($\pm$4.27)
2048	96	160.6	✓	✗	✗	185($\pm$11)	9.18($\pm$0.13)	94.94($\pm$1.32)
2048	96	158.3	✓	✓	✗	152($\pm$7)	8.73($\pm$0.45)	98.76($\pm$2.84)
2048	96	158.3	✓	✓	✓	165($\pm$13)	8.51($\pm$0.32)	99.31($\pm$2.10)
2048	64	71.3	✓	✓	✓	371($\pm$7)	10.48($\pm$0.10)	86.90($\pm$0.61)

Brock A., Donahue J., Simonyan K. Large Scale GAN Training for High Fidelity Natural Image Synthesis, 2018

BigGAN

Samples (512x512)

Interpolations

Brock A., Donahue J., Simonyan K. Large Scale GAN Training for High Fidelity Natural Image Synthesis, 2018

Progressive Growing GAN

Problems with HR image generation

- Disjoint manifolds $\Rightarrow$ gradient problem.
- Small minibatch $\Rightarrow$ training instability.

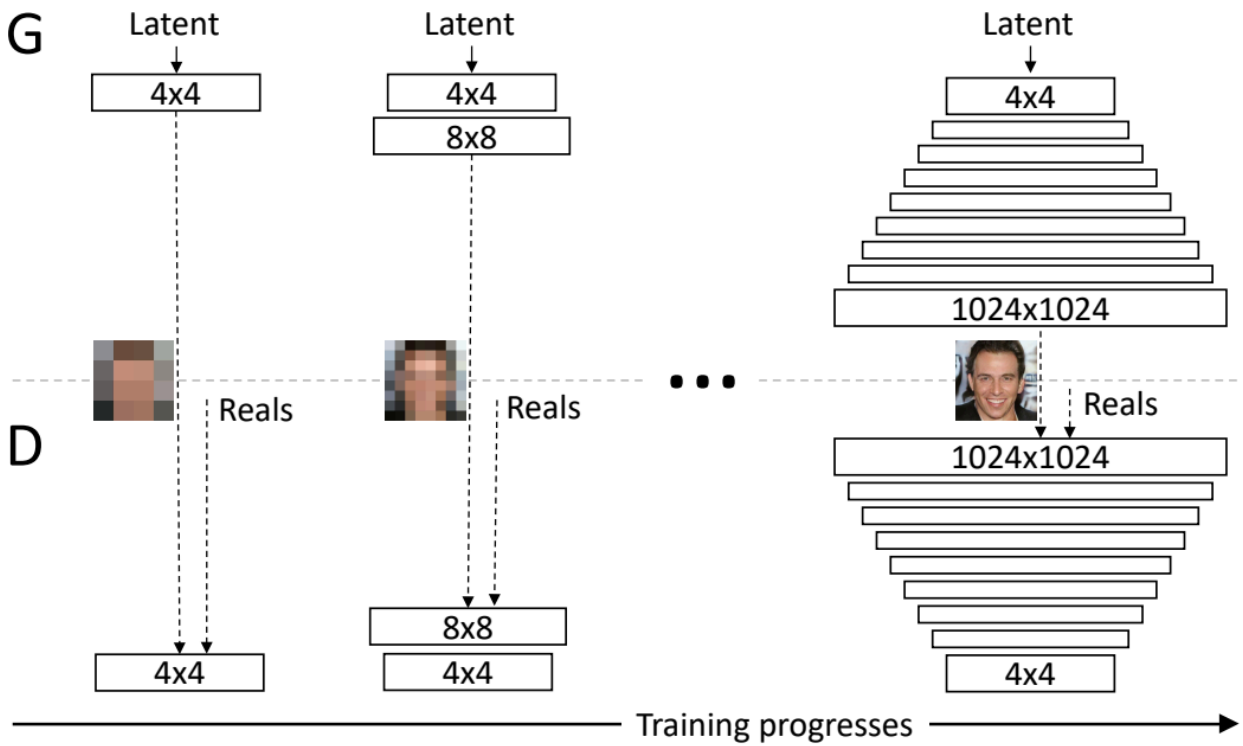
Samples (1024x1024)

Karras T. et al. Progressive Growing of GANs for Improved Quality, Stability, and Variation, 2017

Progressive Growing GAN

Grow both the generator and discriminator progressively, new layers will introduce higher-resolution details as the training progresses.

- Train GAN which generate 4x4 images (2 convs for G and D).
- Add upsampling layers to G, downsampling layers to D.
- Train GAN which generate 8x8 images.
- etc.



Karras T. et al. Progressive Growing of GANs for Improved Quality, Stability, and Variation, 2017

StyleGAN

- Generating of HR images is hard.
- Progressive growing greatly simplifies the task.
- The ability to control specific features of the generated image is very limited.

Face image features

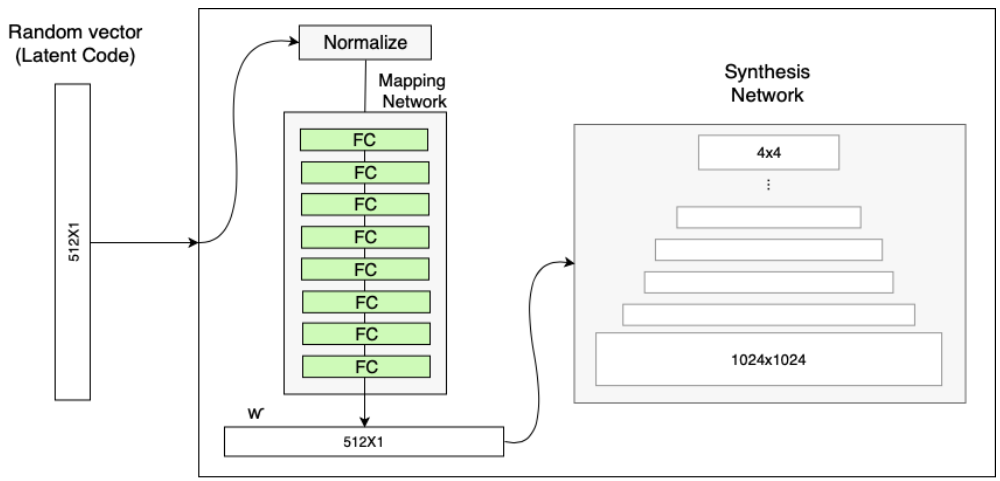
- Coarse (pose, general hair style, face shape). Resolution $4^2 - 8^2$.
- Middle (finer facial features, hair style, eyes open/closed). Resolution $16^2 - 32^2$.
- Fine (color scheme (eye, hair and skin) and micro features). Resolution $64^2 - 1024^2$.

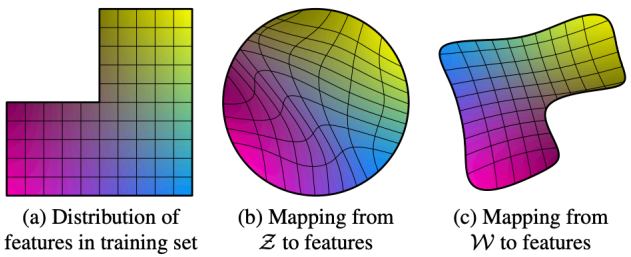
Karras T., Laine S., Aila T. A Style-Based Generator Architecture for Generative Adversarial Networks, 2018

StyleGAN

Step 1: Mapping Network

- Generator input is likely to be **disentangled**. Each component of input vector **z** should be responsible for one generative factor.
- Mapping network $f : \mathcal{Z} \to \mathcal{W}$ is used to reduce correlations between components of **z**.





(a) Distribution of features in training set

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

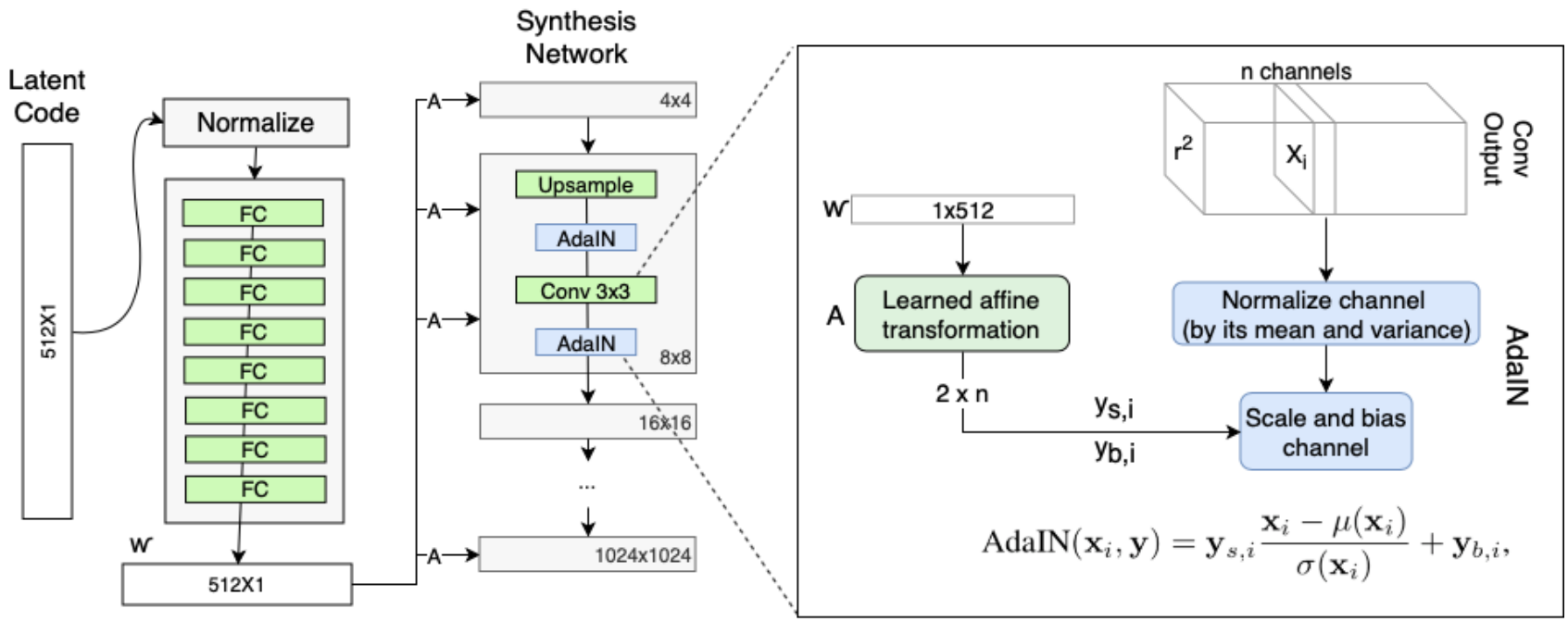
(c) Mapping from $\mathcal{W}$ to features

Karras T., Laine S., Aila T. A Style-Based Generator Architecture for Generative Adversarial Networks, 2018

StyleGAN

Step 2: Style modulation

- Adaptive Instance Normalization transfers the **w** vector to the synthesis Network.
- The module is added to each resolution to define the visual expression of the features.

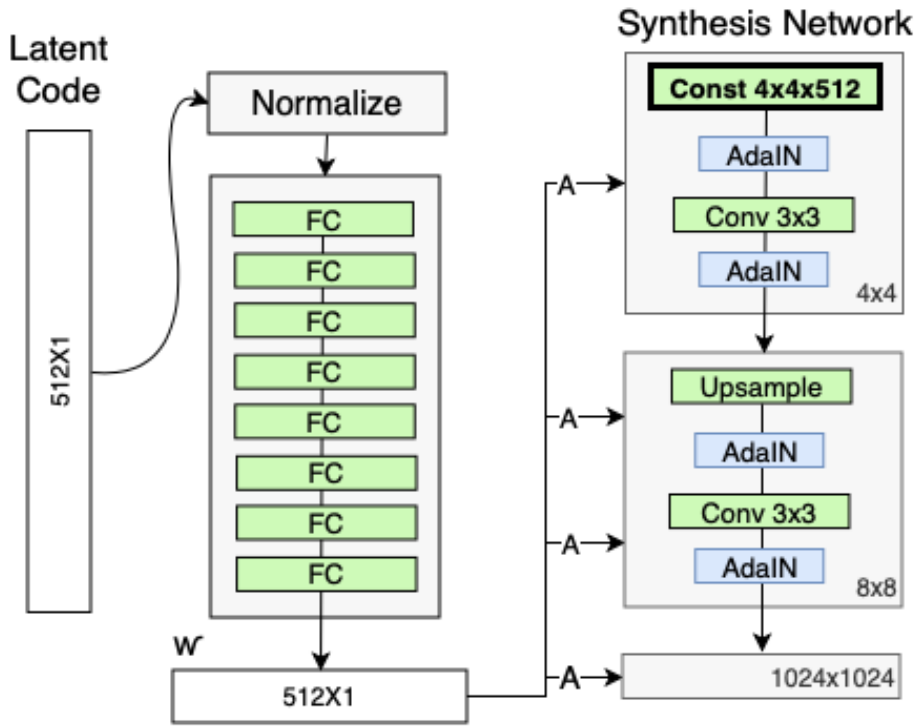


Karras T., Laine S., Aila T. A Style-Based Generator Architecture for Generative Adversarial Networks, 2018

StyleGAN

Step 3: Remove traditional input

Mapping network provides stochasticity to different stages of the synthesis network. Input of the synthesis network is a trainable vector.

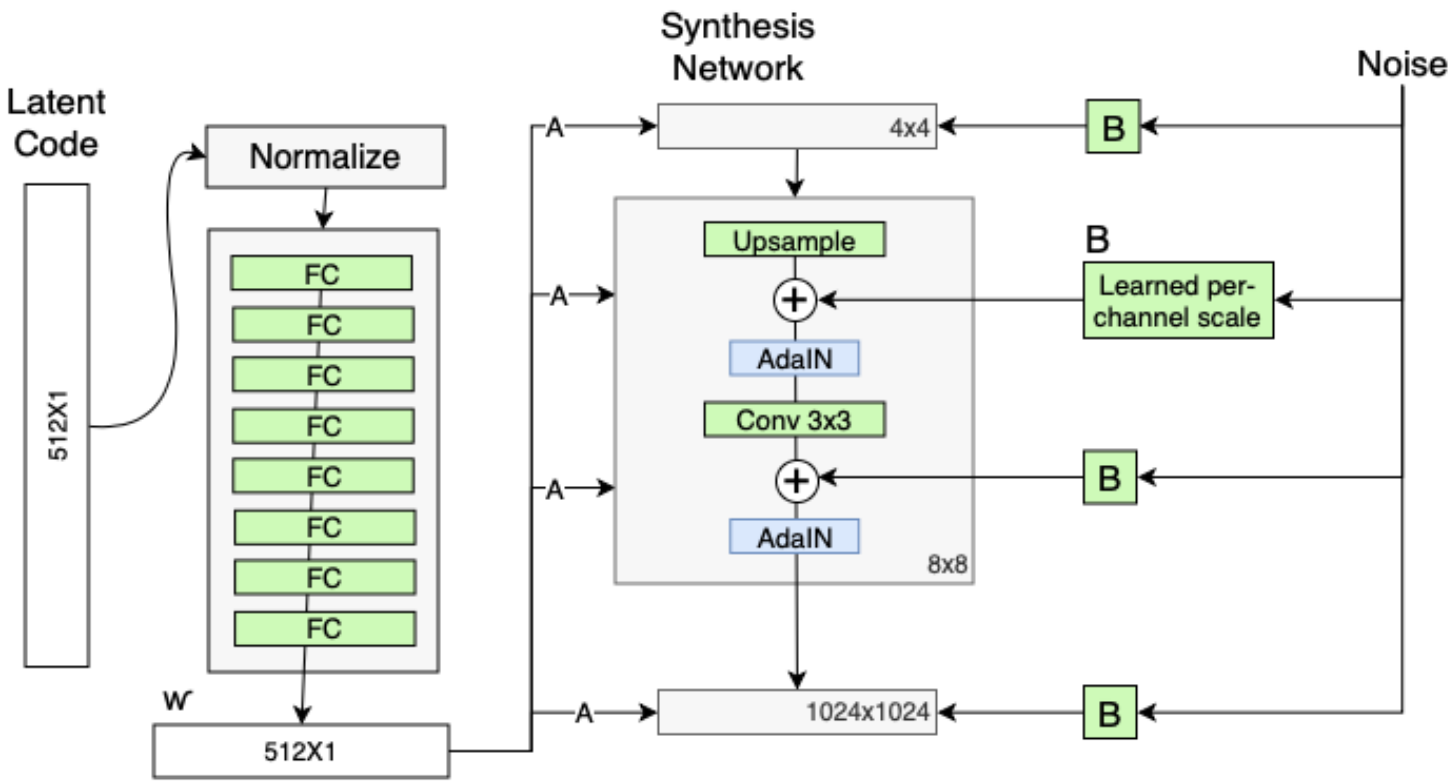


Karras T., Laine S., Aila T. A Style-Based Generator Architecture for Generative Adversarial Networks, 2018

StyleGAN

Step 4: Stochastic variation

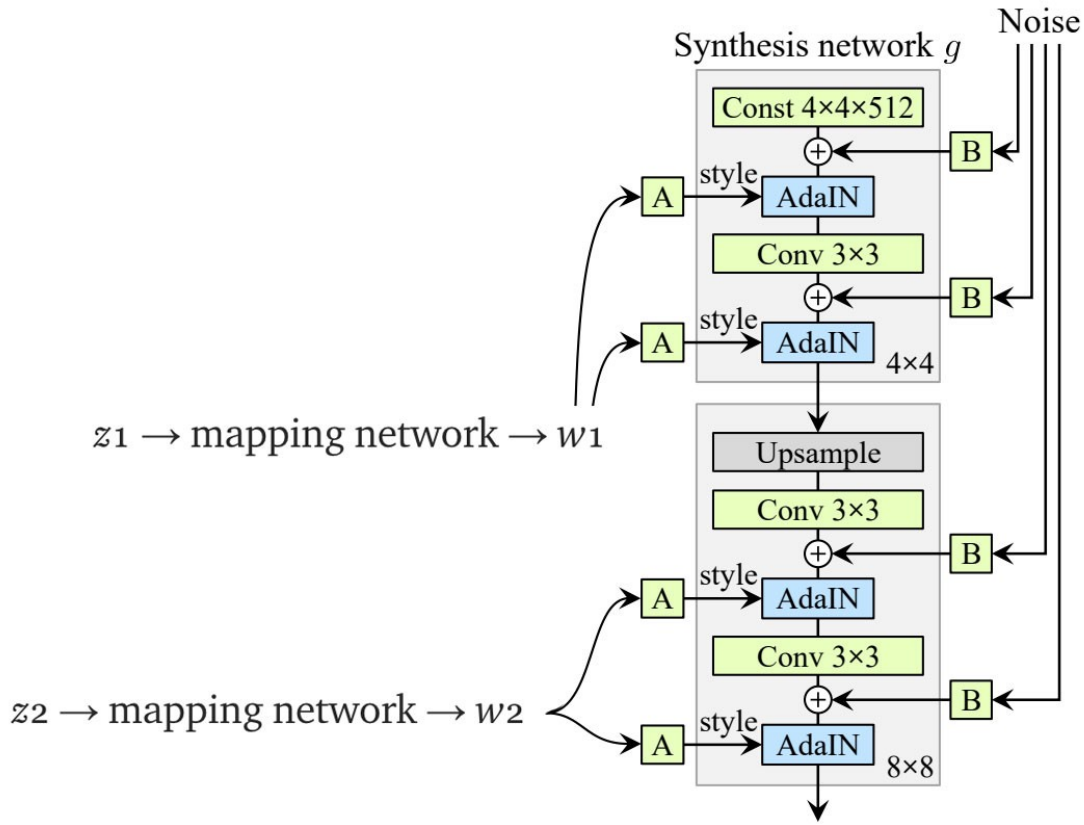
Inject random noise to add small aspects, such as freckles, exact placement of hairs, wrinkles, features which make the image more realistic and increase the variety of outputs.



Karras T., Laine S., Aila T. A Style-Based Generator Architecture for Generative Adversarial Networks, 2018

StyleGAN

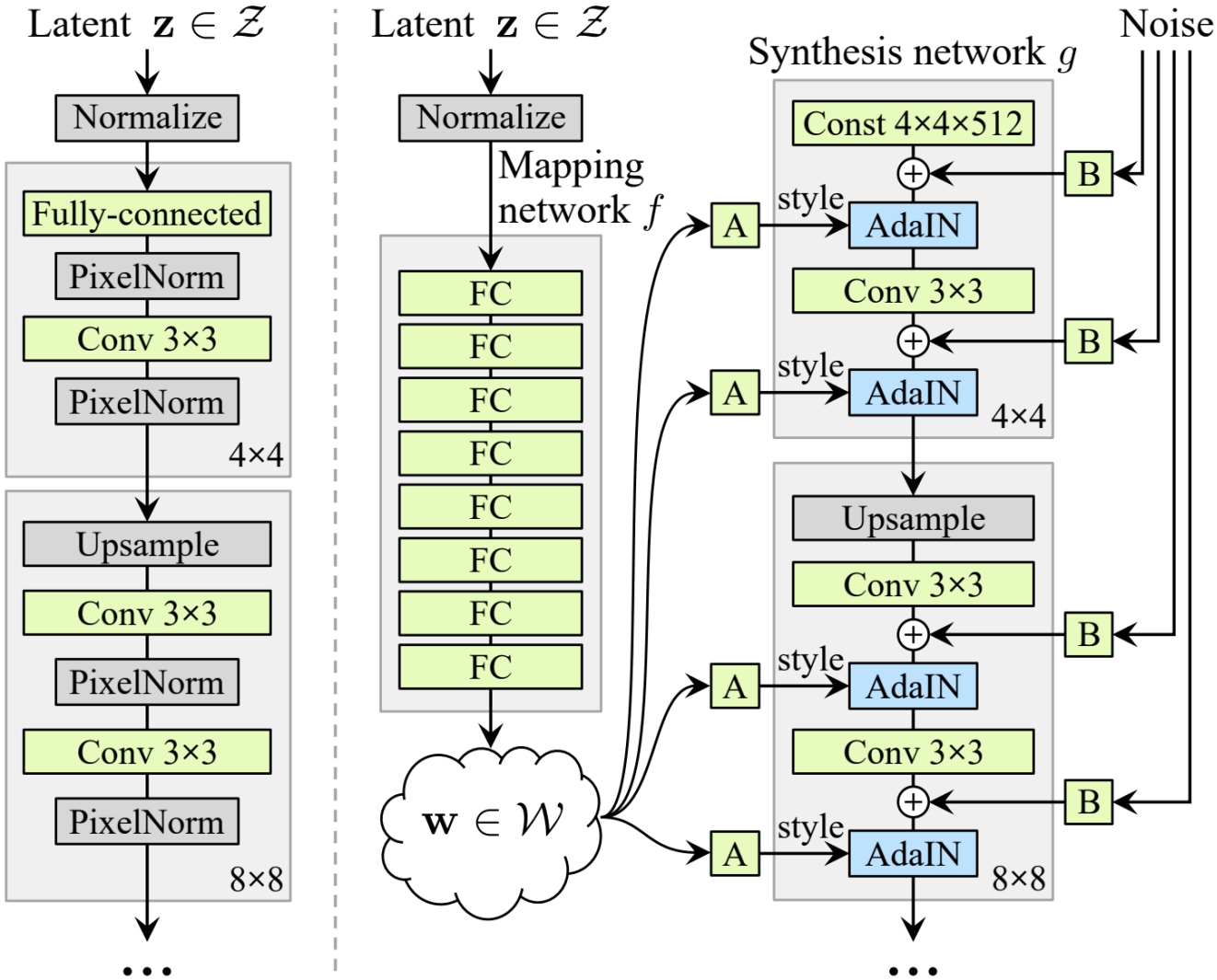
Step 5: Style Mixing



- Makes different levels of synthesis network to be independent.
- Allows to couple diffirent styles.

Karras T., Laine S., Aila T. A Style-Based Generator Architecture for Generative Adversarial Networks, 2018

StyleGAN



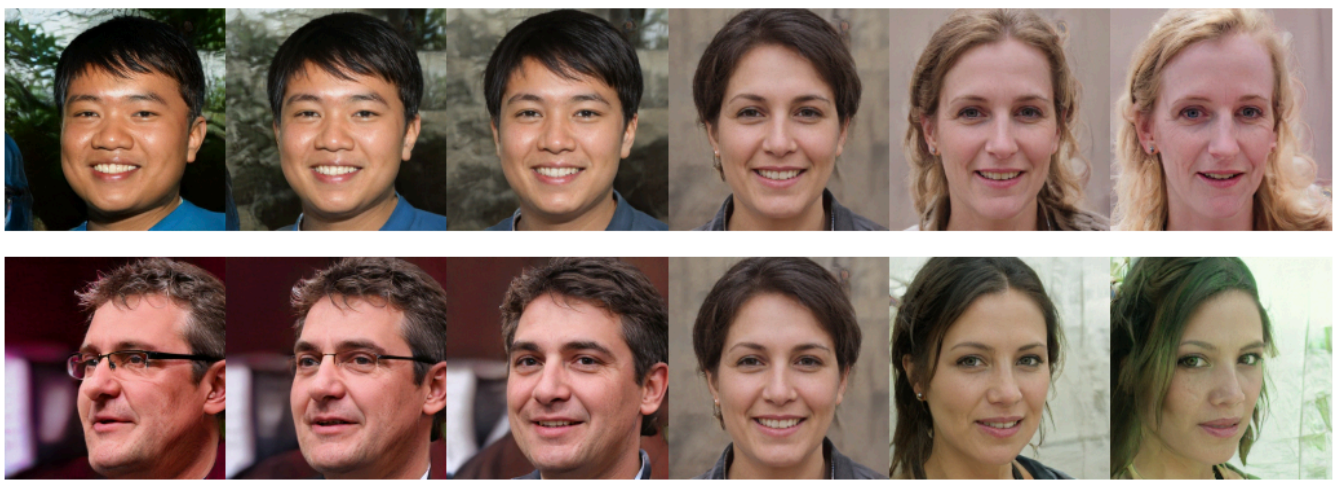
Karras T., Laine S., Aila T. A Style-Based Generator Architecture for Generative Adversarial Networks, 2018

StyleGAN

Truncation trick

$$\mathbf{w}' = \hat{\mathbf{w}} + \psi \cdot (\mathbf{w} - \hat{\mathbf{w}}), \quad \hat{\mathbf{w}} = \mathbb{E}_{\mathbf{z}} p(f(\mathbf{z}))$$

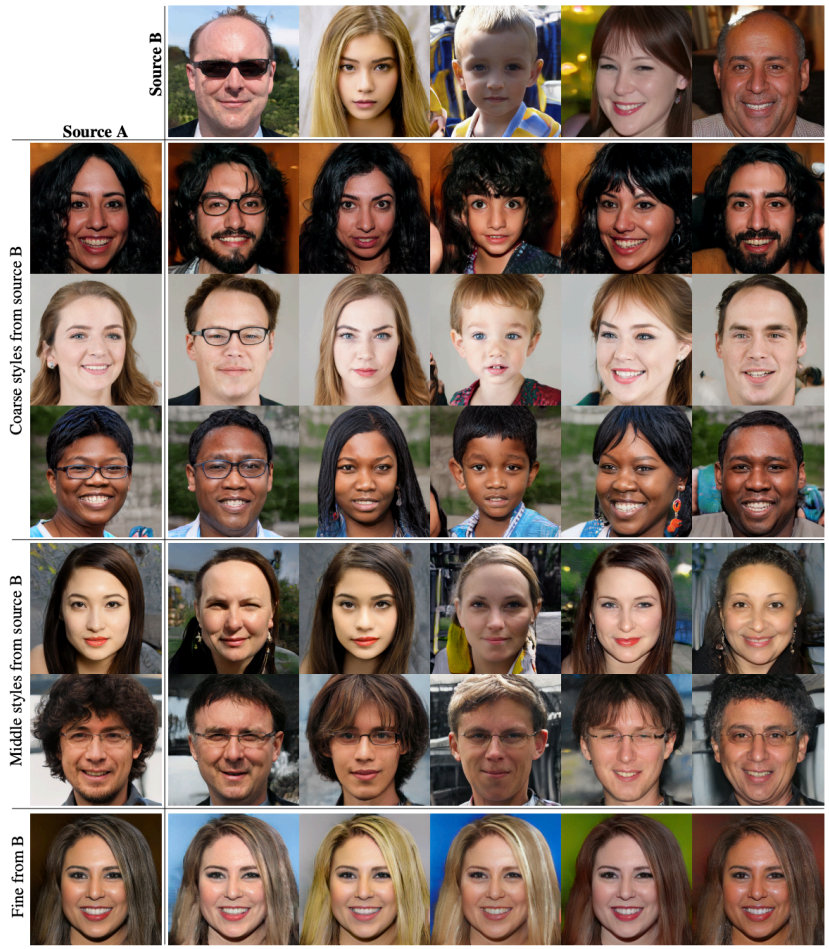
- Constant ψ is a tradeoff between diversity and fidelity.
- $\psi = 0.7$ is used for most of the results.
- Truncation is done only at the low-resolution layers.



Karras T., Laine S., Aila T. A Style-Based Generator Architecture for Generative Adversarial Networks, 2018

StyleGAN

Style Mixing demonstration



Karras T., Laine S., Aila T. A Style-Based Generator Architecture for Generative Adversarial Networks, 2018

StyleGAN

Results

	Method	CelebA-HQ	FFHQ
A	Baseline Progressive GAN [30]	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**

Samples (1024x1024)

Karras T., Laine S., Aila T. A Style-Based Generator Architecture for Generative Adversarial Networks, 2018

StyleGAN

Links

Video: https://www.youtube.com/watch?v=kSLJriaOumA
Code: https://github.com/rosinality/style-based-gan-pytorch
Coursera course: https://www.coursera.org/learn/build-better-generative-adversarial-networks-gans

---

Karras T., Laine S., Aila T. A Style-Based Generator Architecture for Generative Adversarial Networks, 2018