StyleGAN Family

Xuzhe Dang

GAN Overview

Application

- Digital Art
  - StyleGAN, CycleGAN
- Super Resolution
  - SRGAN, SOUP-GAN
- Object Detection
  - Perceptual GAN
- Reinforcement Learning
  - GAIL

GAN Overview

Supplement

- If images are normalized between (-1, 1), the last layer's active function is tanh
- If images are normalized between(0, 1), the last layer's active function is sigmoid

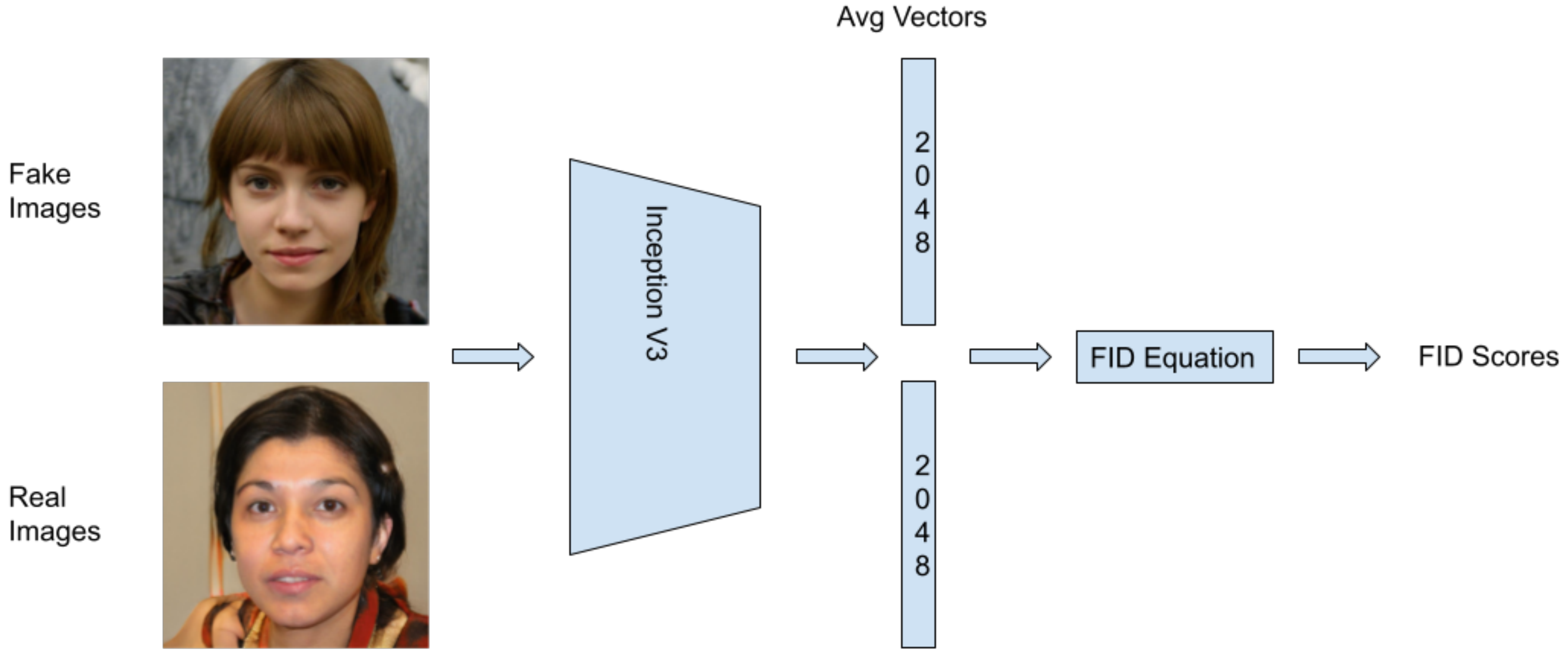
GAN Overview

Image Quality Metric

- FID Score
  - Frechet Inception Distance
  - Compare the distribution of generated images with the distribution of real images that were used to train the generator
  - Use a pre-trained Inception V3 to get feature(Remove the MLP layers)

GAN Overview

Image Quality Metric



$$FID = ||\mu_r - \mu_g||^2 + Tr(\sum_r + \sum_g - 2(\sum_r \sum_g)^{\frac{1}{2}})$$

ProGAN

Motivation

- Three Image Generation Methods
  - Autoregression Model: PixelRNN - Slow
  - VAE - Blurry Result
  - GAN - Small Resolution and Limit Variation
- Why is GANs hard to generate high resolution images ?
  - Gradient Problem (Odena et al., 2017)
  - Small Batch Size

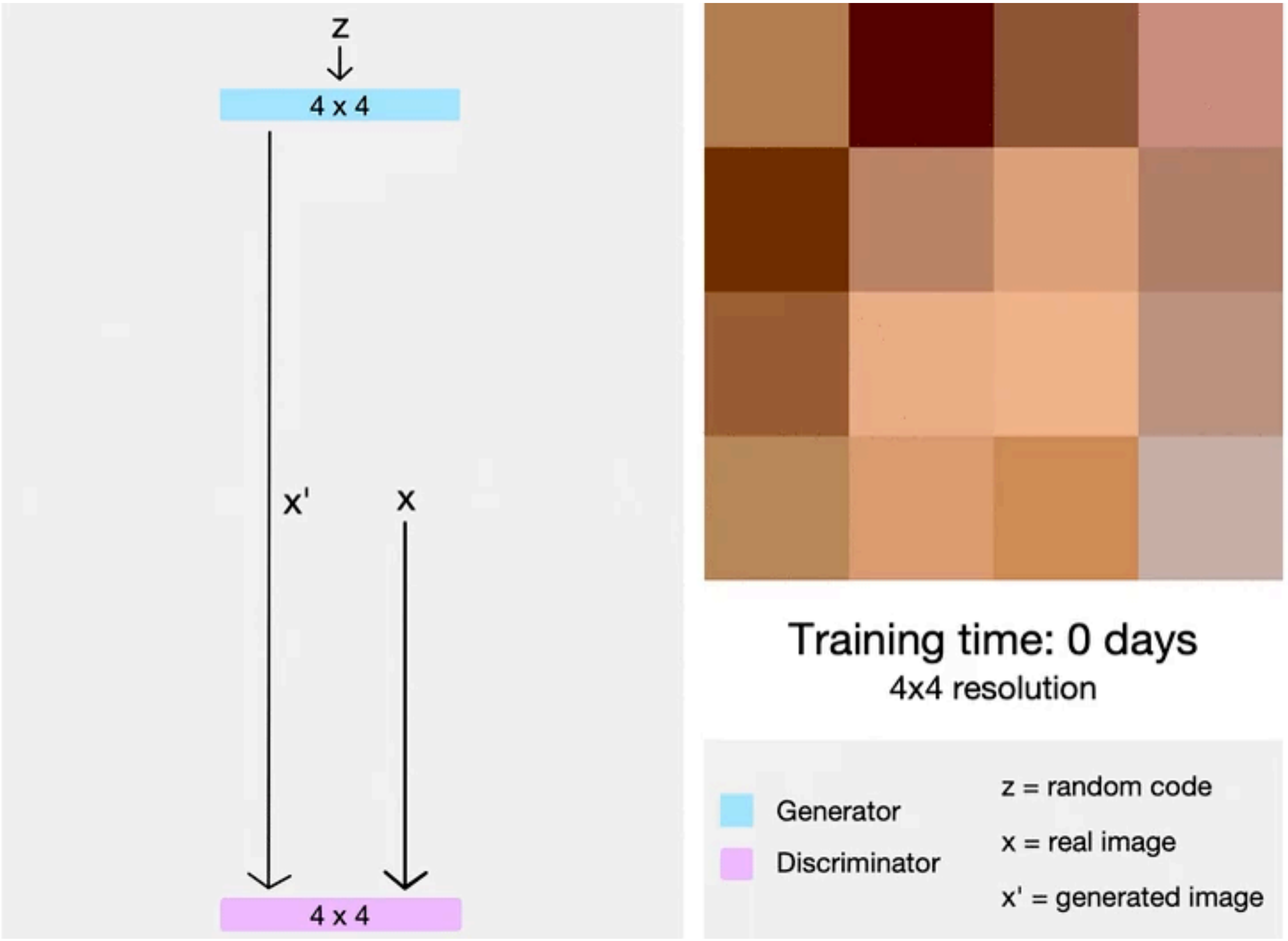
ProGAN

Overview

- In 2017, Nividia proposed ProGAN (Progressive GAN)
- Key to Generate High :
  - Train both generator and discriminator in a progressive growth way

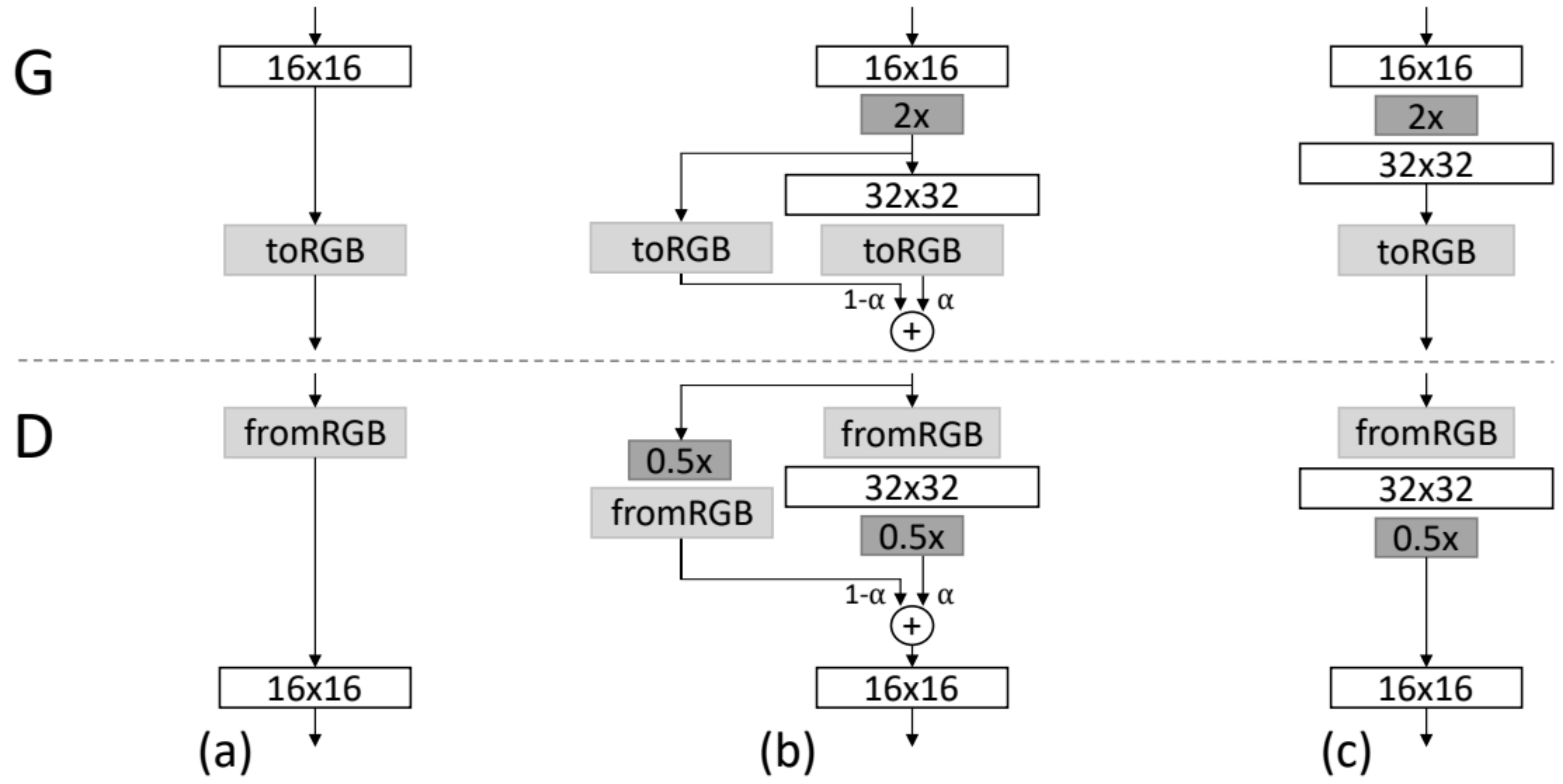
ProGAN

Progressive Growth



ProGAN

Progressive Growth



Smooth Fade In: During the training, α will increase from 0 to 1

ProGAN

Progressive Growth

- Benefits:
  - More stable: Each layer increase the difficult slowly, like how human learn math
    - Low layers: Pose, Face Shape
    - Medium Layers: Hair Style, Eye Postion
    - High Layers: Eyes Color, Skin
  - Reduce Train Time
    - 2–6 Times Faster

ProGAN

Results

Mao et al. (2016b) (128 × 128) Gulrajani et al. (2017) (128 × 128) Our (256 × 256)

StyleGAN

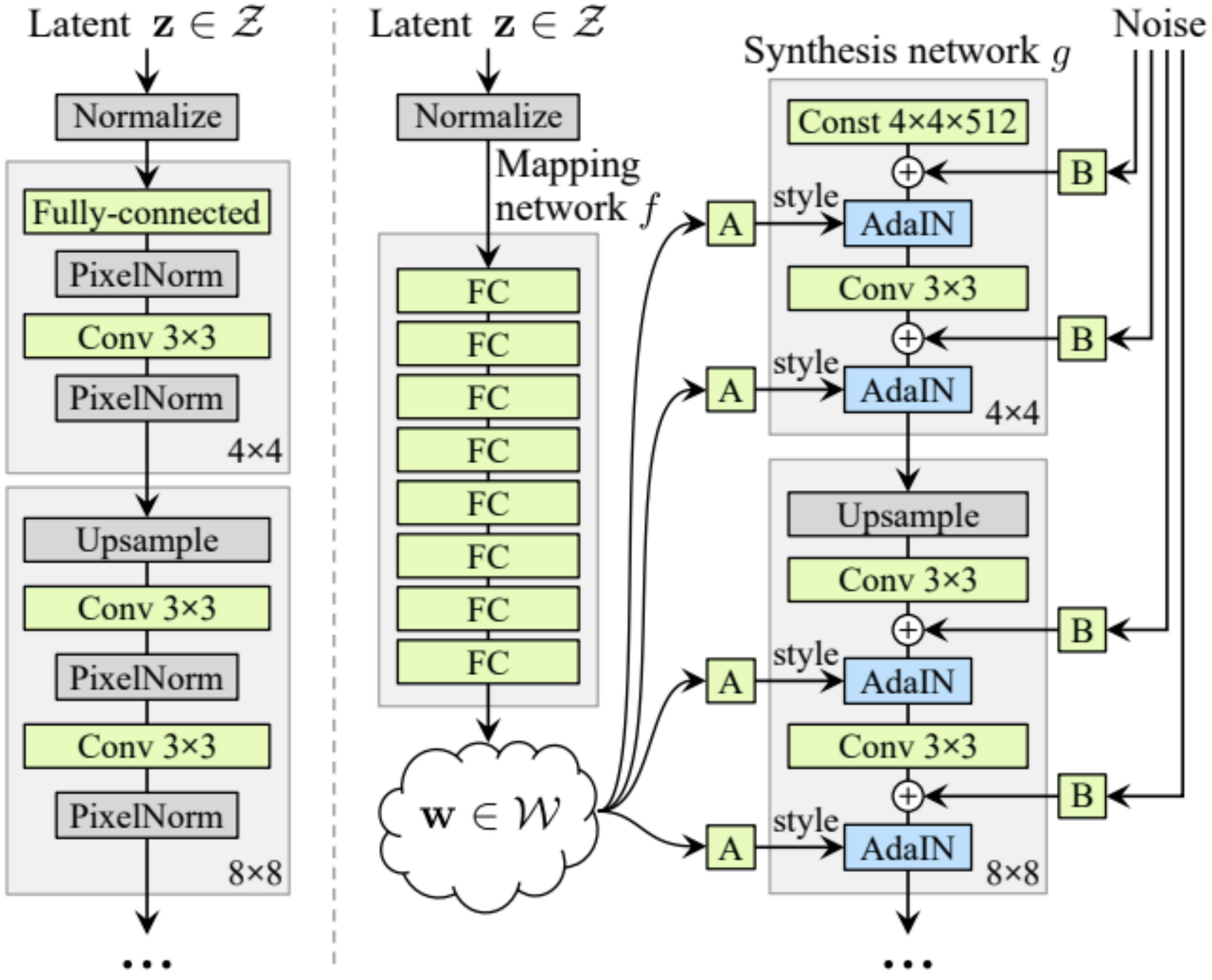
Overview

- Disadvantages with ProGAN:
  - Like other GANs, ProGAN has a poor ability to control specified aspects of generated images.
- Contributions:
  - Style-based Generator
  - Methods to Measure the Disentanglement
  - A High Quality Human Face Dataset

StyleGAN

Style-based Generator

- Use a learned const matrix as initial input
- Mapping Netoworks:
  - Latent Z -> Latent W
- AdaIn:
  - Specialize w to styles y = (ys, yb)
  - Mix the style into images
  - $AdaIn(x_i, y) = y_{s,i}\frac{x_i - \mu(x_i)}{\sigma(x_i)} + y_{b,i}$
- Add Noise into images



(a) Traditional

(b) Style-based generator

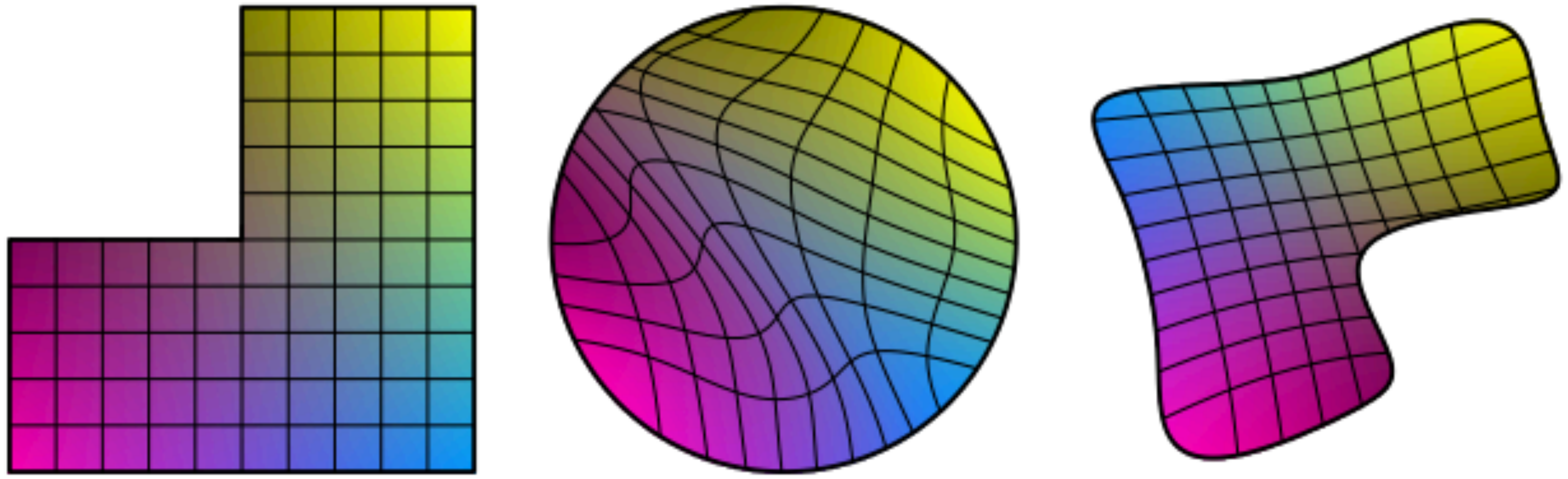
StyleGAN

Style-based Generator

- Why convert latent Z into intermedia latent W?
- Disentanglement
  - A latent space that consists of linear subspaces
  - Each subspaces controls one factor of variation
- Sampling probability of each combination of factors in latent Z need match the corresponding density in the training data
- Intermedia Latent W is sampled from a by learned function rather than a fixed distribution

StyleGAN

Style-based Generator



(a) Distribution of features in training set

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

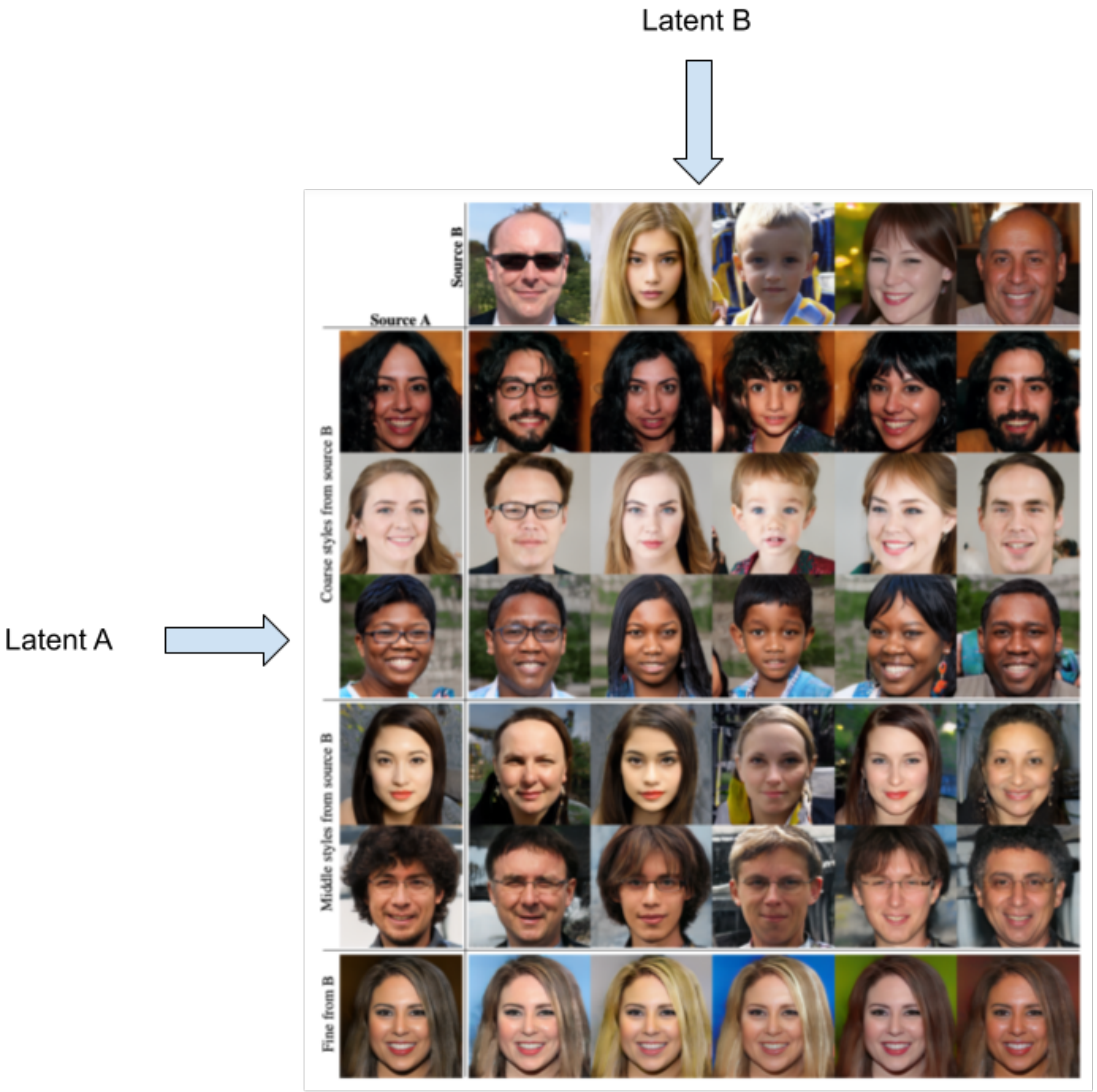
(c) Mapping from $\mathcal{W}$ to features

StyleGAN

Style-based Generator

- Mixing regularization
  - During the train, image generated from a mixing of two latent codes
  - Encourage the styles to localize
- FID Scores of applying 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**


Latent B
Latent A
Source B
Source A
Course styles from source B
Middle styles from source B
Fine from B

StyleGAN

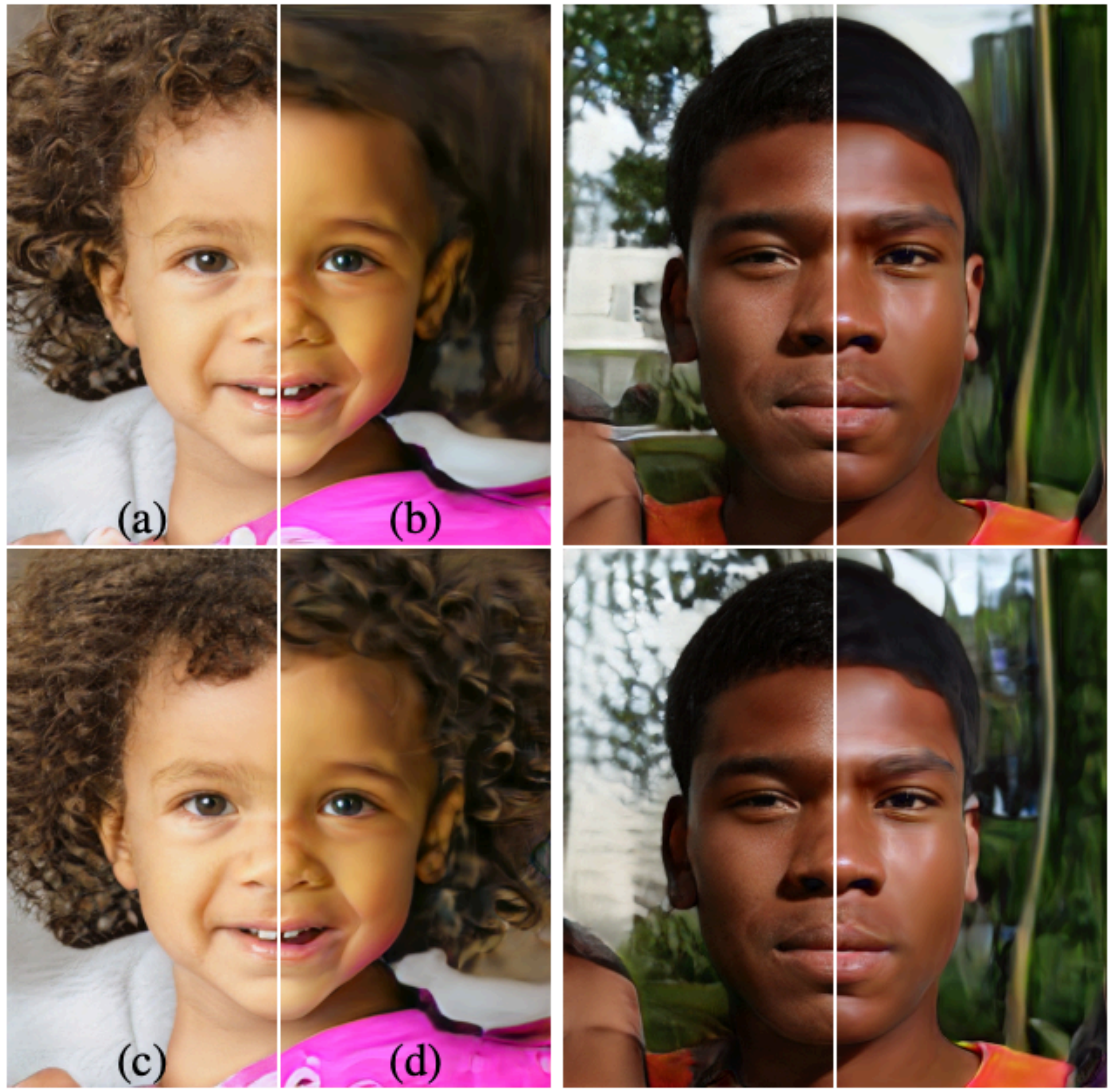
Style-based Generator

- Stochastic variation
  - There are lots of aspects in human portraits can be consider stochastic
  - Freckles, skin pores or placement of hairs
- StyleGAN introduce these stochastic aspects with noise

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

StyleGAN

Style-based Generator



StyleGAN

Methods to Measure the Disentanglement

- StyleGAN introduce two methods to measure the Disentanglement
  - Perceptual path length
  - Linear separabilit

StyleGAN

Perceptual path length

- Latent space interpolations
  - Interpolation of latent-space vectors may yield surprisingly non-linear changes in the image, which means the latent space is entangled and the factors of variation are not properly separated.

StyleGAN

Perceptual path length

- https://github.com/NVlabs/stylegan/blob/master/metrics/perceptual_path_length.py
- Perceptual path length
  - Whether the image changes smoothly in "perceptual"
- Equations for compute perceptual path length
  - $L_z = E[\frac{1}{\epsilon^2} d(G(slerp(z_1, z_2; t)), G(slerp(z_1, z_2; t + \epsilon)))]$

    $L_w = E[\frac{1}{\epsilon^2} d(G(slerp(f(z_1), f(z_2); t)), G(slerp(f(z_1), f(z_2); t + \epsilon)))]$

StyleGAN

Linear separabilit

- https://github.com/NVlabs/stylegan/blob/master/metrics/linear_separability.py
- Linear separabilit
    - If latent space is well disentangled, it would be easier to do classification in the latent space (Easily train some simple classifier, for example SVM)
    - Train SVM on latent space
    - Train CNN on images from latent spcace
    - Compare the outputs' difference of two classifiers

StyleGAN

Linear separabilit

- Train auxiliary classification networks for a number of binary attribute Linear separability
- Get 100,000 labeled latent-space vector from the auxiliary classification networks
- Fit a linear SVM for each attribute to predict the label based on the latent-space point —z for traditional and w for style-based — and classify the points by this plane.
- Compute the final with equation
  - $exp(\sum_i H(Y_i | X_i))$

StyleGAN

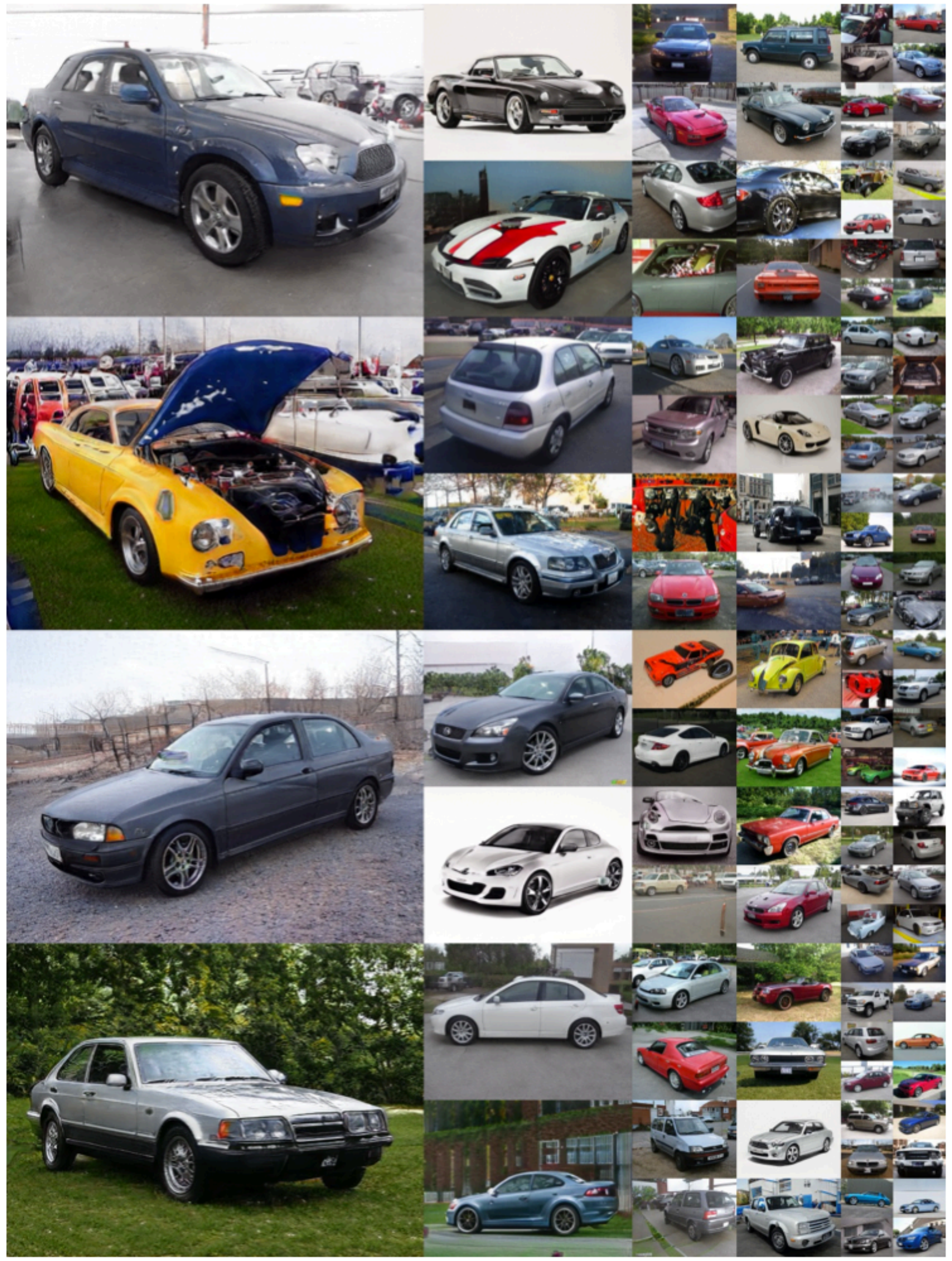
Results

	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

	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**

StyleGAN

Results



StyleGAN 2

Motivation

- Problem of image generated from StyleGAN
  - Water droplet -like artifacts in some generated images from StyleGAN; both images and features

StyleGAN 2

Overview

- Try to improve the generated images' quality by applying several methods to StyleGAN
- Key changes:
  - Modify the architecture of generator
  - Lazy regularization
  - Path length regularization
  - Remove Progressive growth

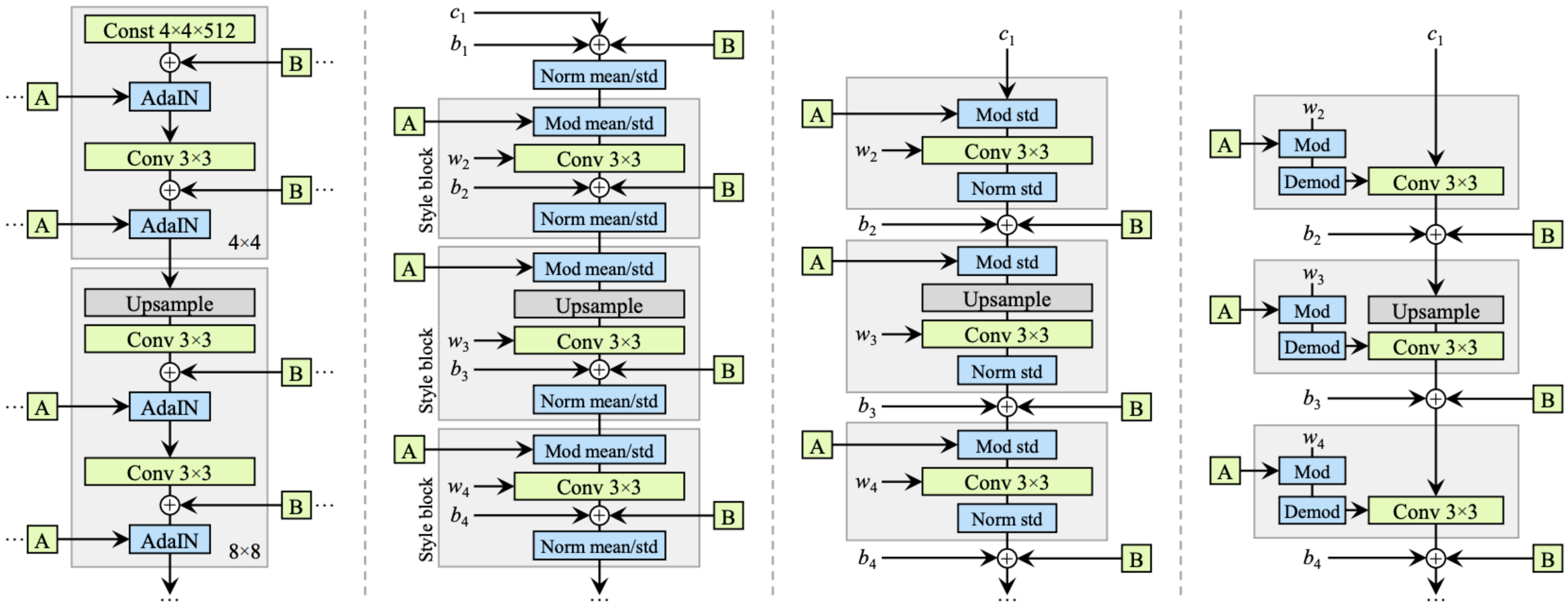
StyleGAN 2

StyleGAN2 Generator

- What cause the water droplet -like artifacts?
- Hypothesis
  - The normalize operation of AdaIn will destroy the information of features relative to each other
  - With AdaIn, Generator creating a strong, localized spike that dominates the statistics, then the generator can effectively scale the signal as it likes elsewhere.
  - Experiments support this hypothesis

StyleGAN 2

StyleGAN2 Generator



(a) StyleGAN (b) StyleGAN (detailed) (c) Revised architecture (d) Weight demodulation

StyleGAN 2

Lazy regularization and Path length regularization

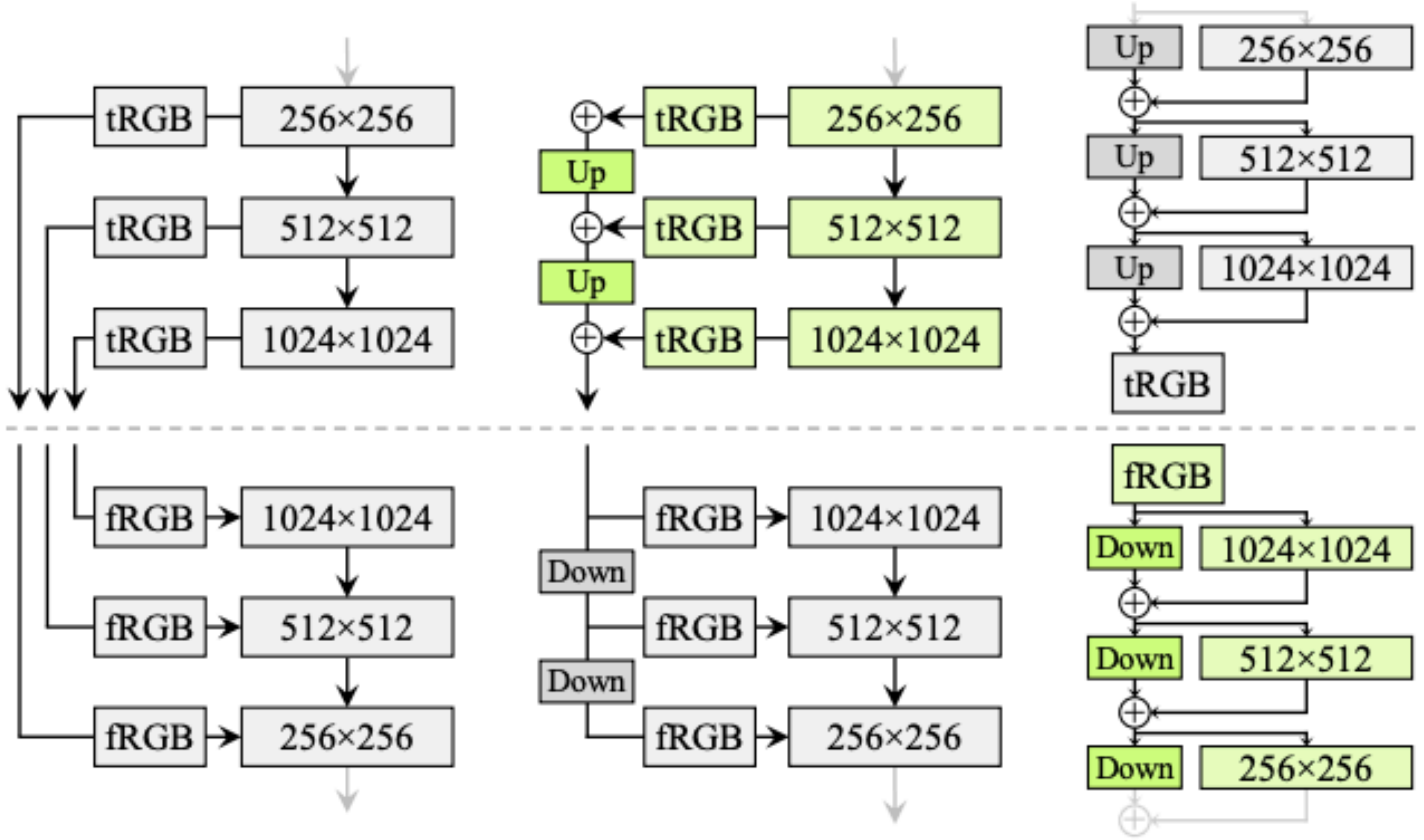
- Lazy regularization
  - Normally, loss function and regularization are computed at same time.
  - In StyleGAN2, to reduce the computational cost and memory use, the regularization computed in every 16 steps.
- Path length regularization
  - Image distances of the continuous linear interpolation points should be similar
  - $E_{w,y\ N(0,I)}(||J_W^T y||_2 - a)^2$

StyleGAN 2

Problem with Progressive Growth

StyleGAN 2

Different Generators and Discriminator without Progressive Growth



(a) MSG-GAN (b) Input/output skips (c) Residual nets

StyleGAN 2

Problem with Progressive Growth

	Configuration	FFHQ, 1024×1024				LSUN Car, 512×384			
		FID ↓	Path length ↓	Precision ↑	Recall ↑	FID ↓	Path length ↓	Precision ↑	Recall ↑
A	Baseline StyleGAN [24]	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	–	–	–	–

StyleGAN 2

Problem with Progressive Growth

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

StyleGAN 2

Project image to latent code

- Image2StyleGAN: How to Embed Images Into the StyleGAN Latent Space
- It's a reversed way of image generation.
  - Latent code -> Image
  - Image -> Latent code
- Could be used to compare similarity of two images

StyleGAN 2

Project image to latent code

- Method 1
  - Train an auto encoder to encode images to latent space
  - Fast, but poor generalization ability
- Method 2
  - Optimize a random latent code
  - Slow, but good generalization ability

StyleGAN 2

Results

Demo Time

Q&A

Thank You