Copyright Notice

deeplearning.ai

GAN Improvements

Outline

- How GANs have improved
- State of the art methods for improving GANs performance

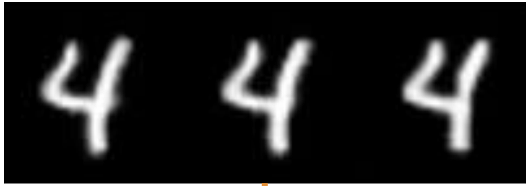
GANs Over Time

Ian Goodfellow
@goodfellow_ian

4.5 years of GAN progress on face generation.
arxiv.org/abs/1406.2661 arxiv.org/abs/1511.06434
arxiv.org/abs/1606.07536 arxiv.org/abs/1710.10196
arxiv.org/abs/1812.04948

Main Improvements: (1) Stability

High standard deviation



Low standard deviation

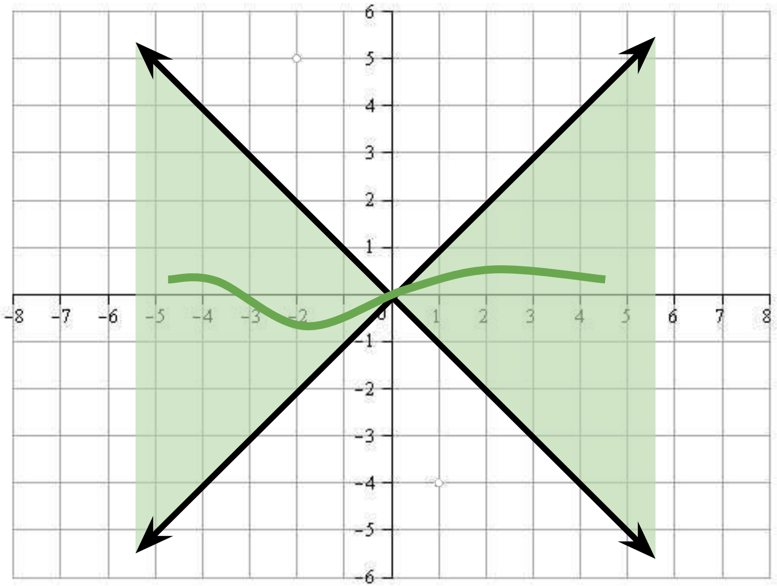
Use batch standard deviation to encourage diversity

Main Improvements: (1) Stability

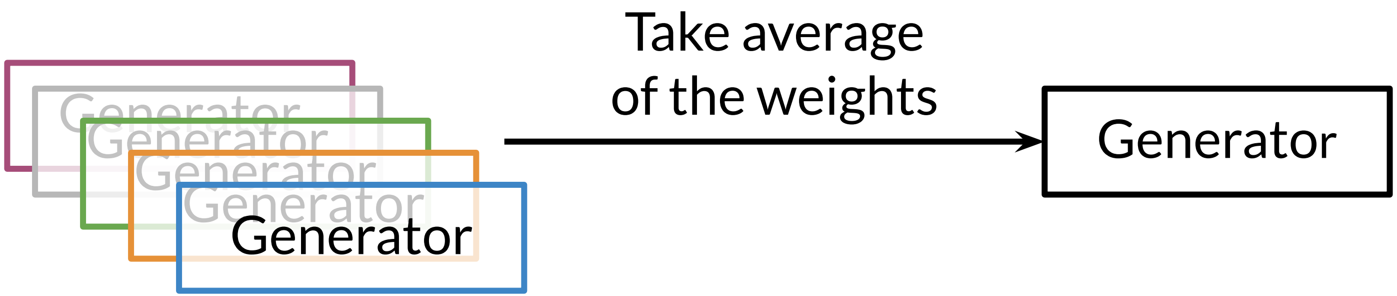
Improve stability by enforcing 1-Lipschitz continuity

E.g. WGAN-GP and Spectral Normalization

∇ : gradient



Main Improvements: (1) Stability



Generator at
different iterations

Main Improvements: (1) Stability

No averaging

Exponential averaging

Use moving average for smoother results

Available from: https://arxiv.org/abs/1806.04498v2

Main Improvements: (1) Stability

Progressive growing gradually trains GAN at increasing resolutions

Available from: https://arxiv.org/abs/1710.10196

Main Improvements: (2) Capacity

Larger models can use
higher resolution images

Main Improvements: (3) Diversity

Available from: https://github.com/NVlabs/stylegan

Summary

- GANs have improved because of:
    - Stability - longer training and better images
    - Capacity - larger models and higher resolution images
    - Diversity - increasing variety in generated images

deeplearning.ai

StyleGAN Overview

Outline

- StyleGAN achievements
- What styles are
- Introduction to StyleGAN architecture and components

StyleGAN Goals

1. Greater **fidelity** on high-resolution images
2. Increased *diversity* of outputs
3. More control over image features

Greater Fidelity

Not fooling anyone

I’m shook

(Left) Available from: https://arxiv.org/abs/1406.2661
(Right) Available from: https://github.com/NVlabs/stylegan

Increased Diversity

Available from: https://arxiv.org/abs/1812.04948

Increased Diversity

More Feature Control

Hair color/style →

← Glasses

Available from: https://arxiv.org/abs/1812.04948

Style in GANs

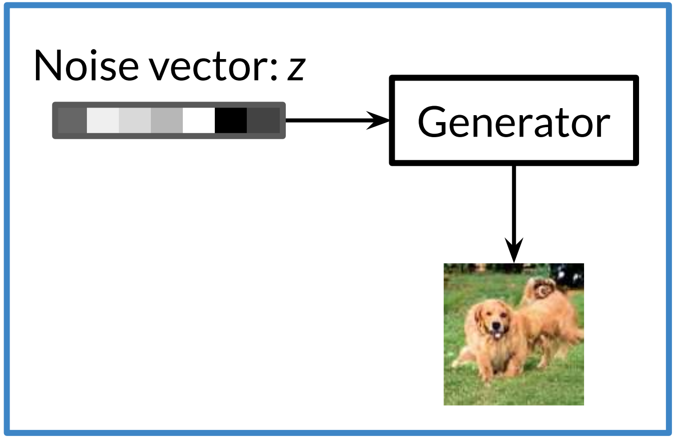
Style = variation in an image

Early styles are coarser like face shape

Later styles are finer like hair wisps

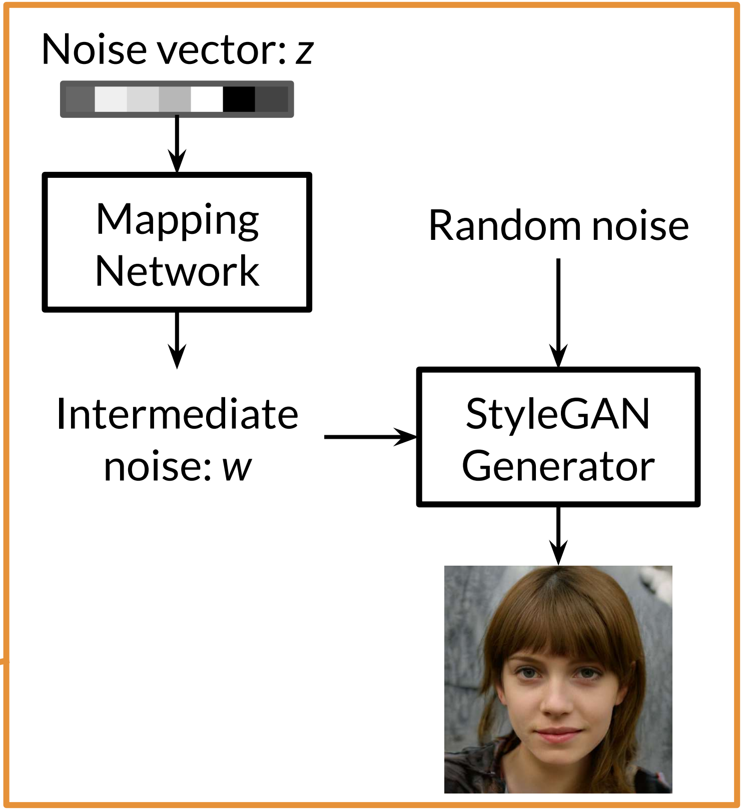
Available from: https://arxiv.org/abs/1812.04948

The Style-Based Generator

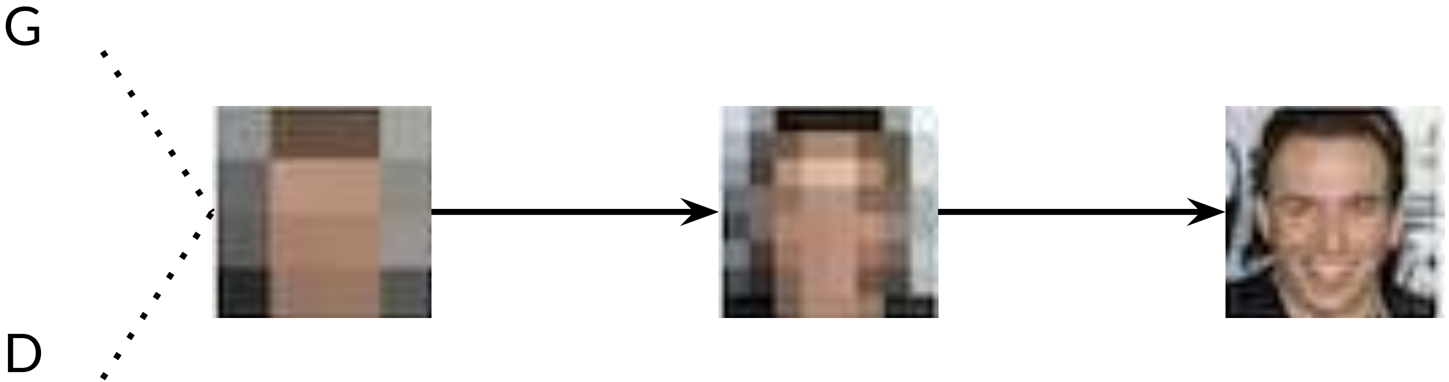


Traditional architecture

StyleGAN architecture



Progressive Growing



Available from: https://arxiv.org/abs/1710.10196

Summary

- StyleGAN's **goals**:
  - Greater fidelity, increased diversity, improved control over features
- Style is any variation in the image
- **Main components** of StyleGAN:
  - Progressive growing
  - Noise mapping network
  - Adaptive instance normalization (AdaIN)

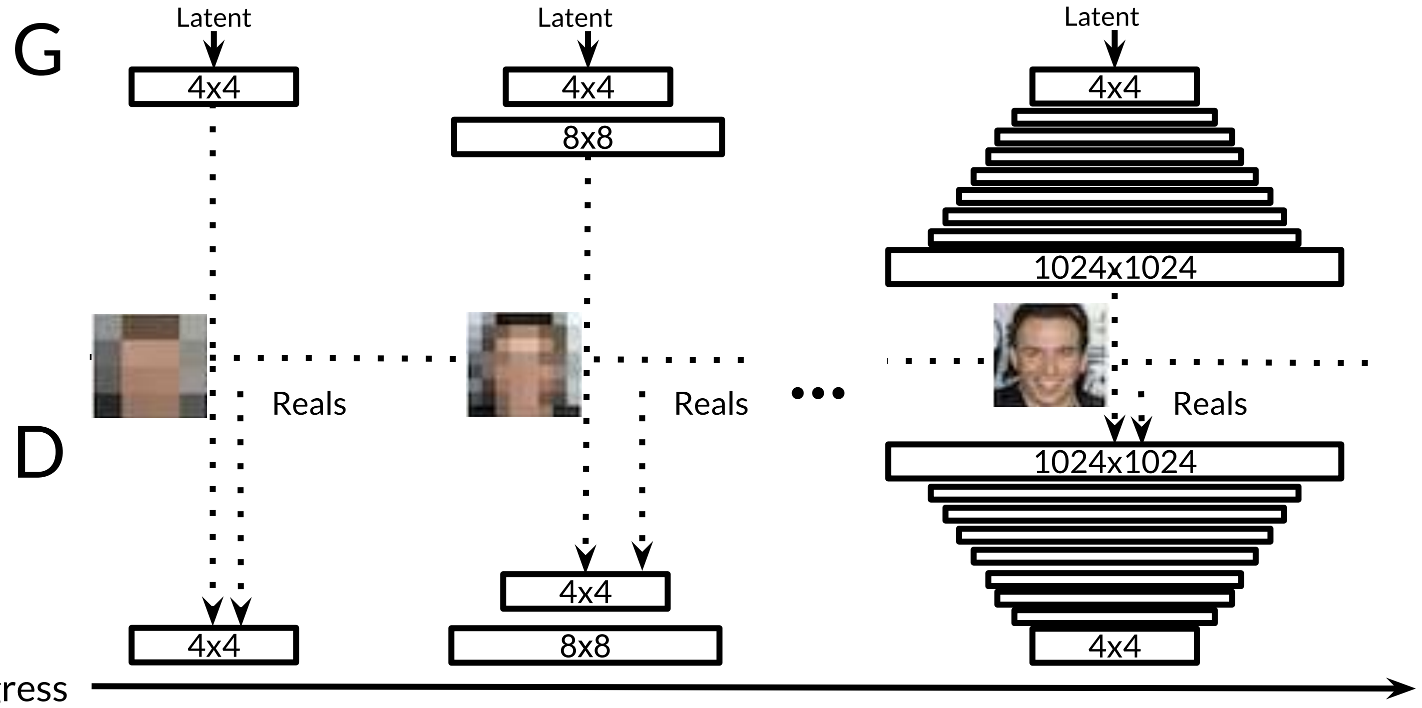
deeplearning.ai

Progressive Growing

Outline

- Progressive growing intuition and motivation
- How to implement it

Progressive Growing

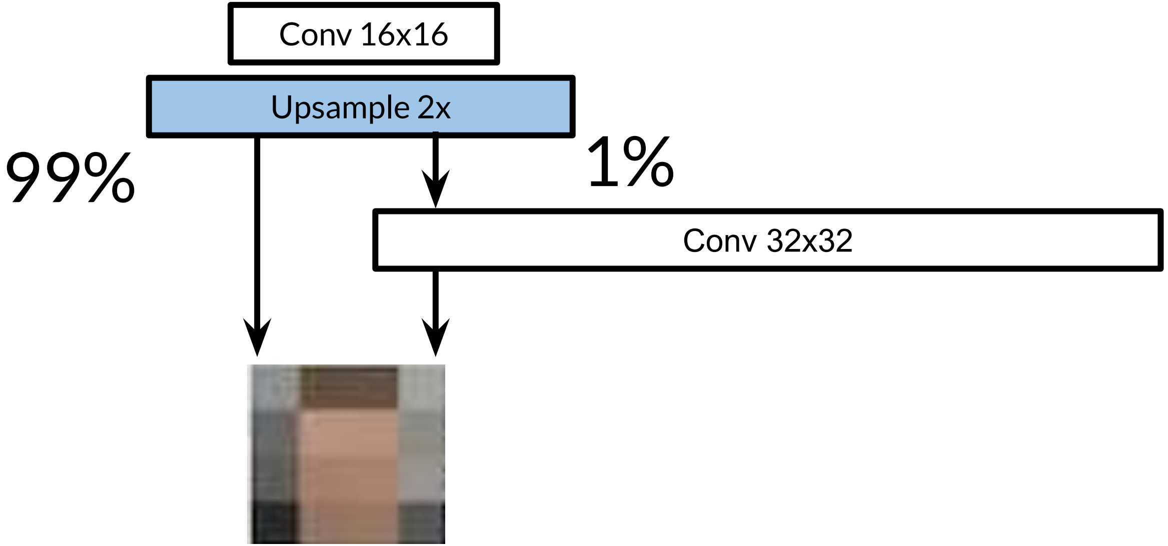


Based on: https://arxiv.org/abs/1710.10196

Progressive Growing in Action

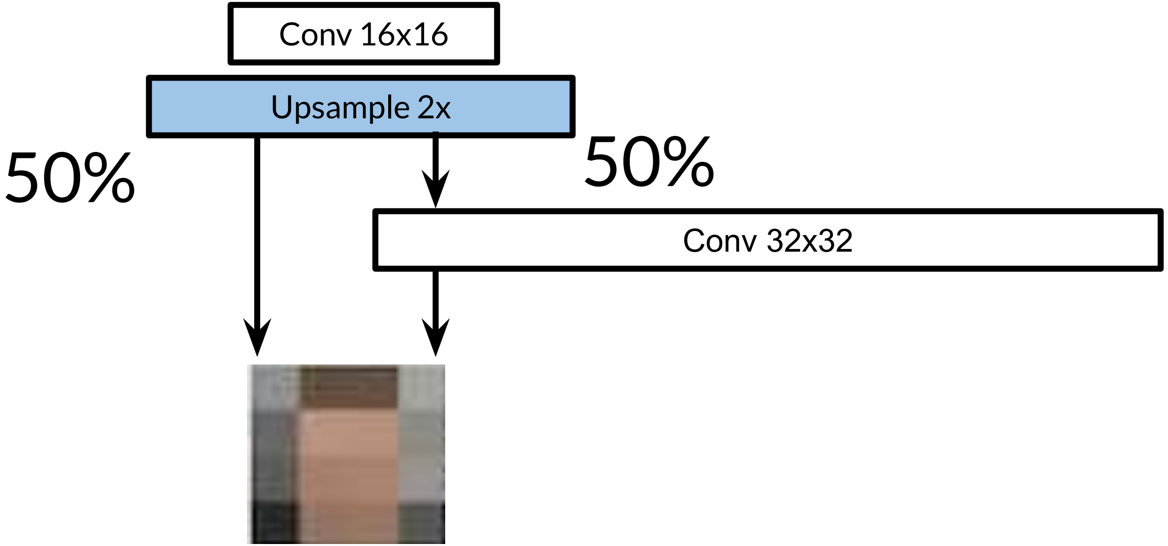
Available from: https://www.gwern.net/images/gan/2019-03-16-stylegan-facestraining.mp4

Progressive Growing: Generator



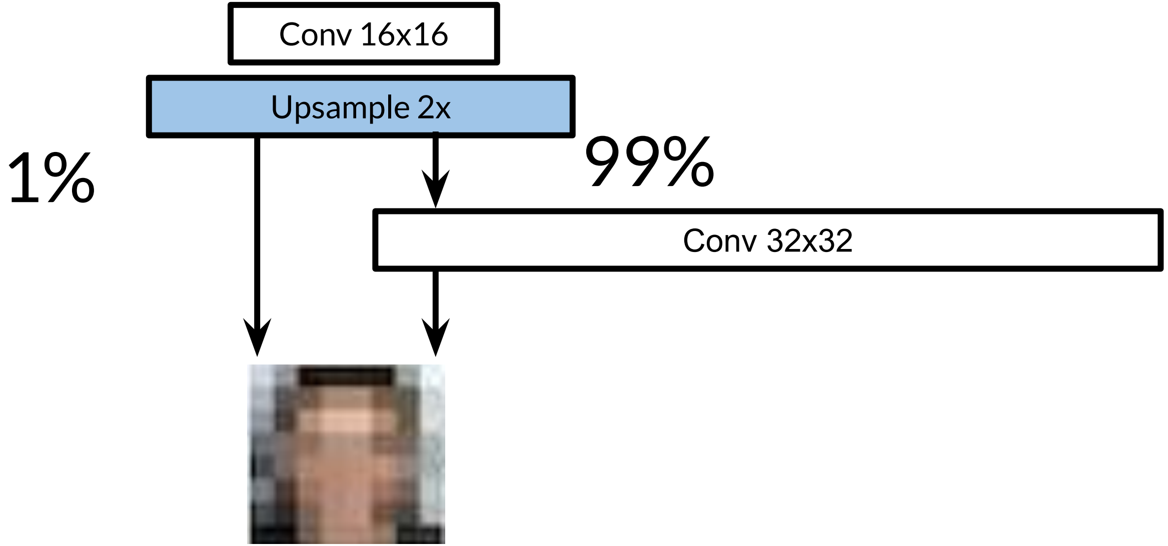
Based on: https://arxiv.org/abs/1710.10196

Progressive Growing: Generator



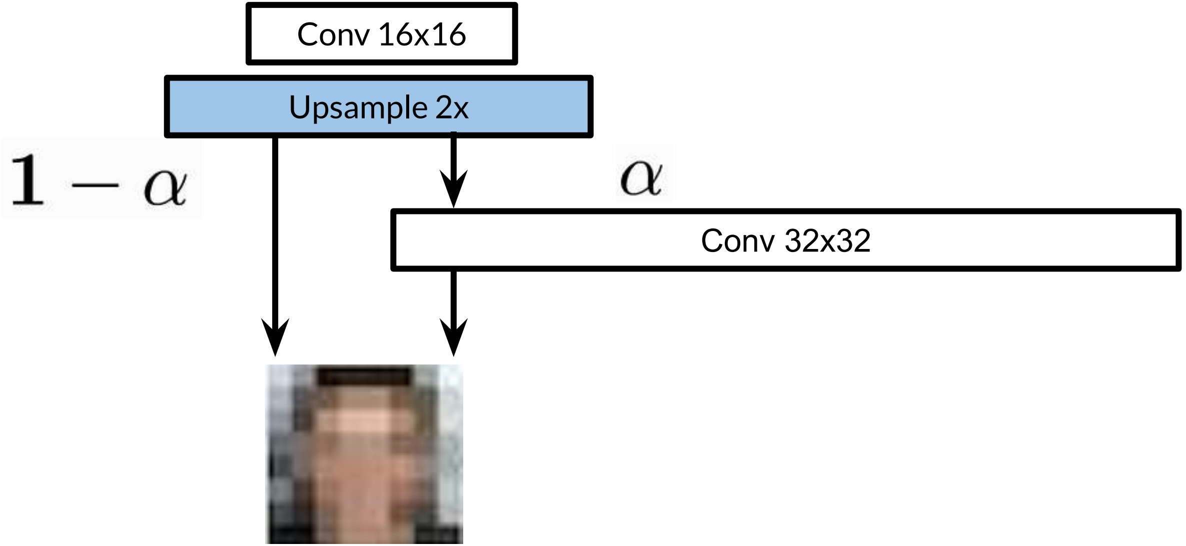
Based on: https://arxiv.org/abs/1710.10196

Progressive Growing: Generator



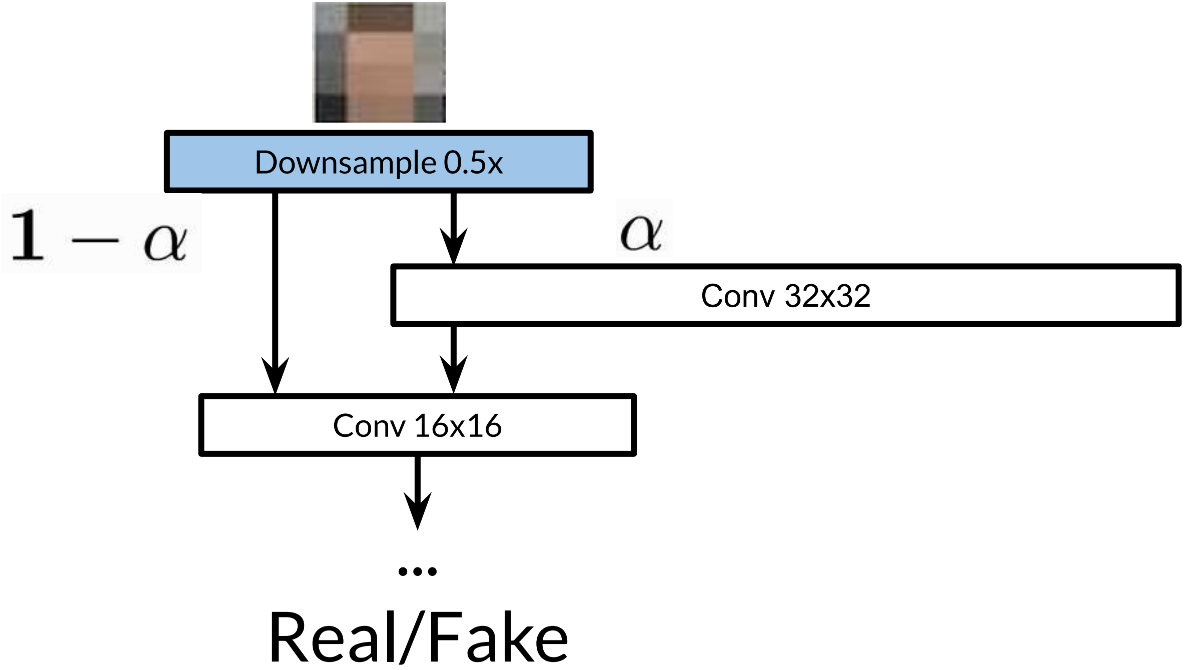
Based on: https://arxiv.org/abs/1710.10196

Progressive Growing: Generator



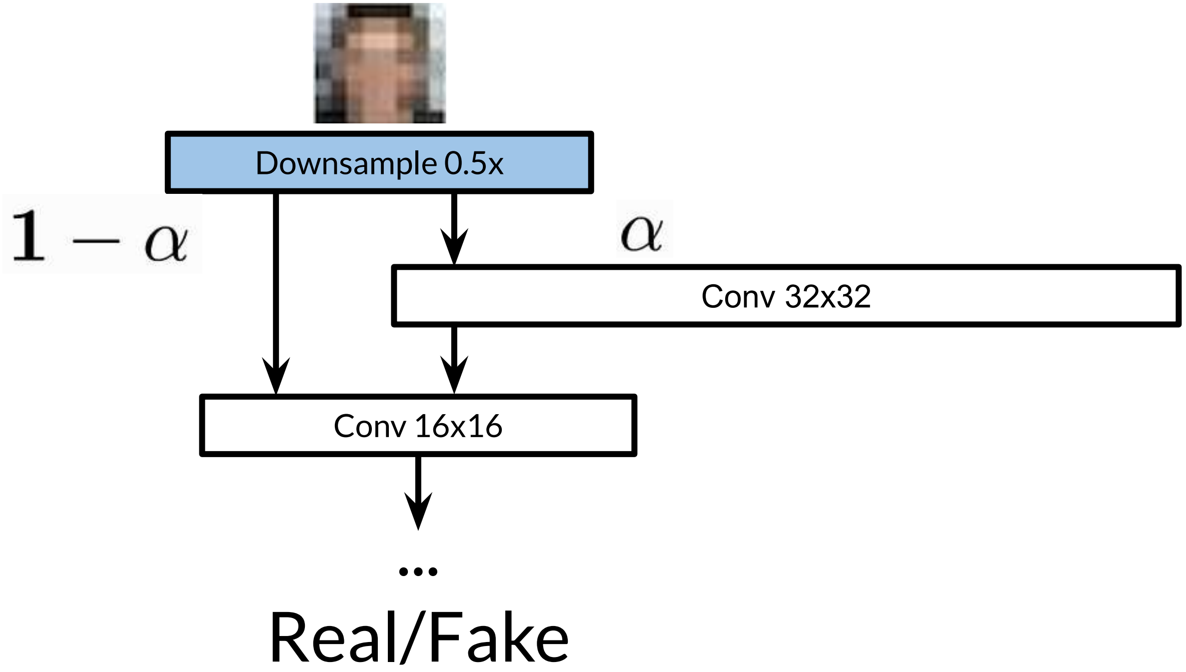
Based on: https://arxiv.org/abs/1710.10196

Progressive Growing: Discriminator



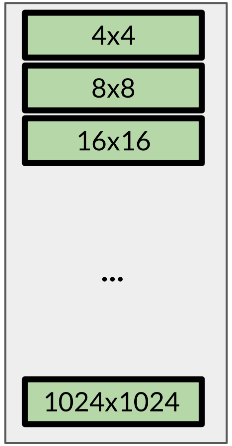
Based on: https://arxiv.org/abs/1710.10196

Progressive Growing: Discriminator



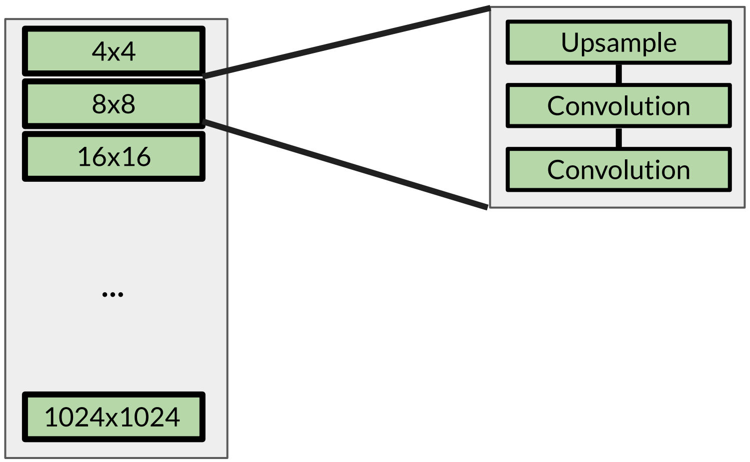
Based on: https://arxiv.org/abs/1710.10196

Progressive Growing in Context



Based on: https://arxiv.org/abs/1812.04948

Progressive Growing in Context



Based on: https://arxiv.org/abs/1812.04948

Summary

- Progressive growing gradually doubles image resolution
- Helps with faster, more stable training for higher resolutions

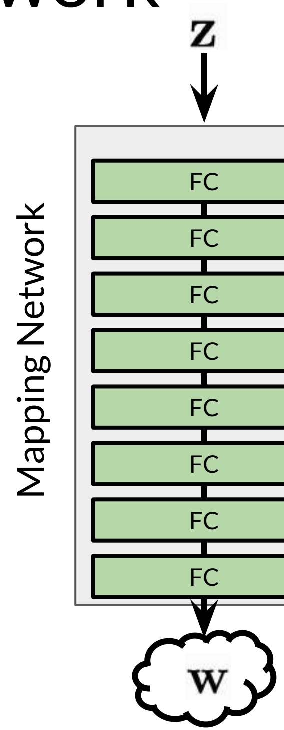
deeplearning.ai

Noise Mapping Network

Outline

- Noise mapping network structure
- Motivation behind the noise mapping network
- Where its output $\mathbf{w}$ goes

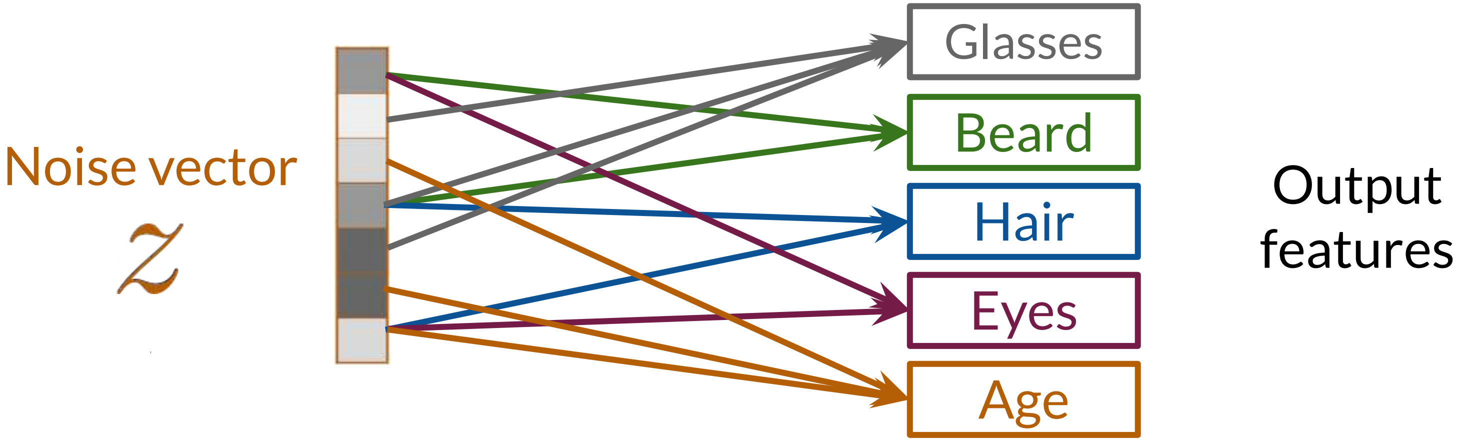
Noise Mapping Network



Intermediate noise vector

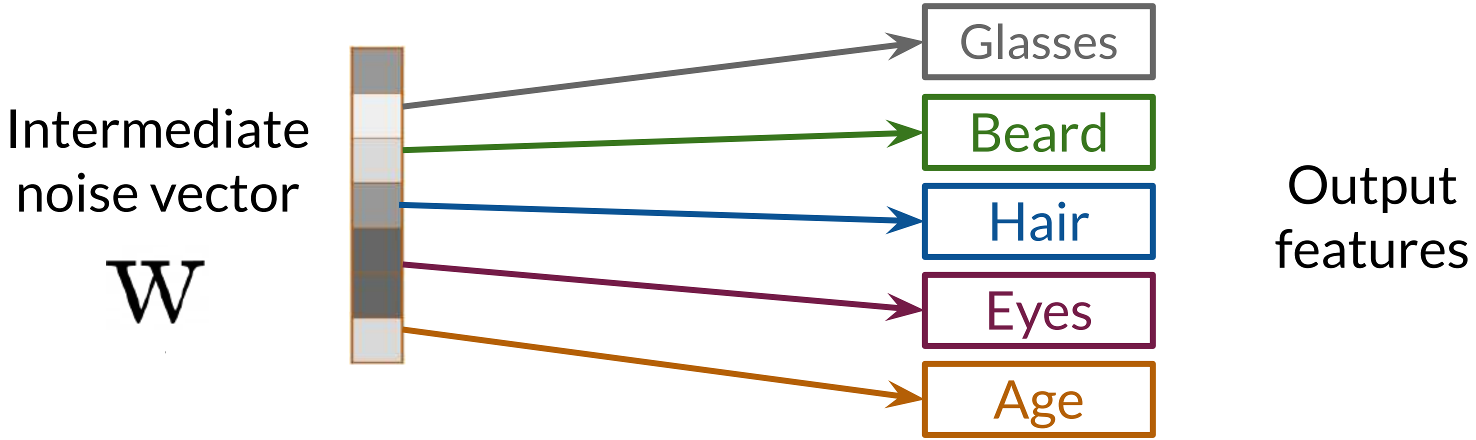
Based on: https://arxiv.org/abs/1812.04948

Remember: Z-Space Entanglement



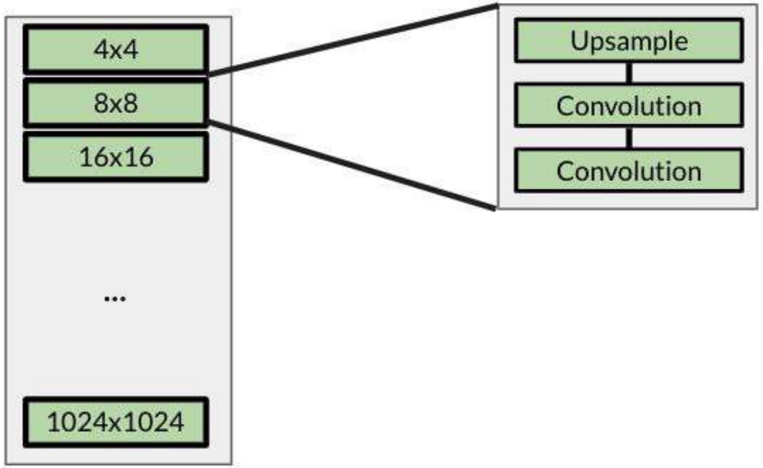
Not possible to control single output features

W-Space: Less Entangled



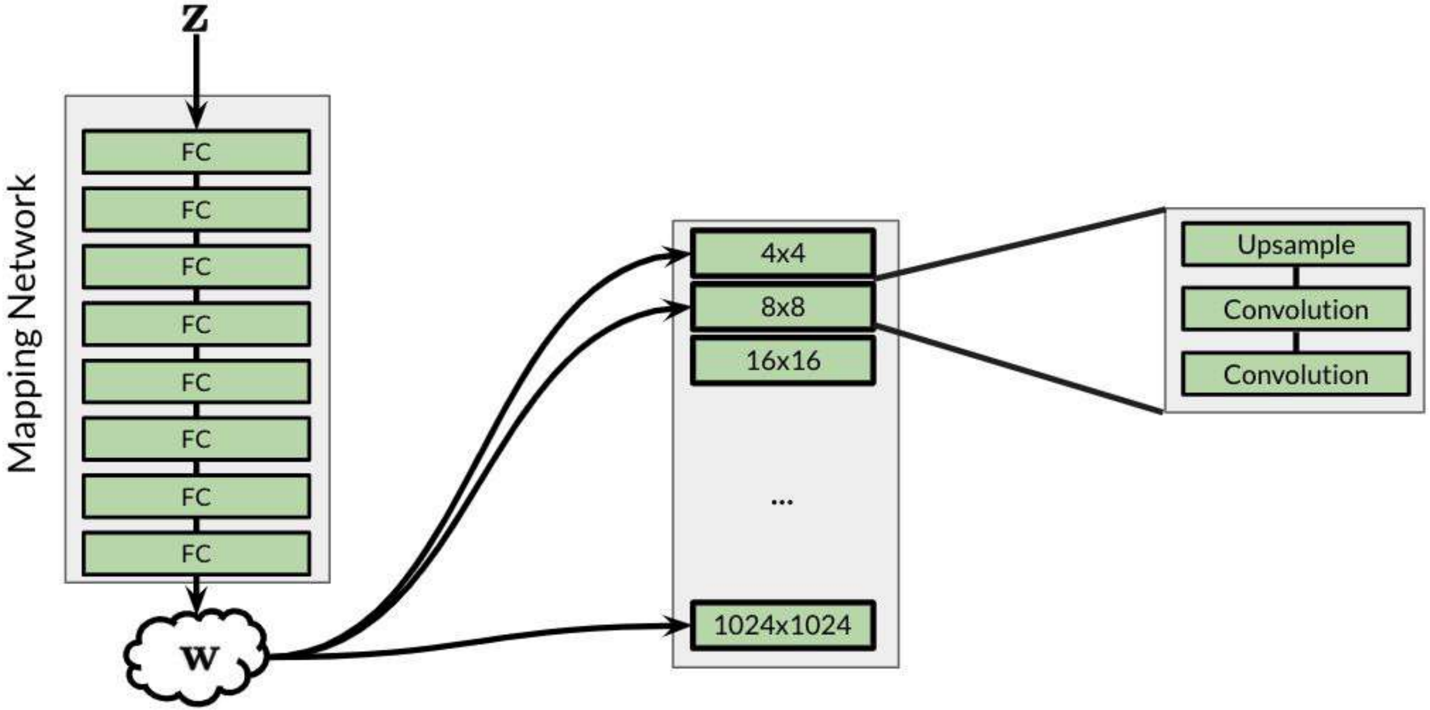
More possible to control single output features

Mapping Network in Context



Based on: https://arxiv.org/abs/1812.04948

Mapping Network in Context



Based on: https://arxiv.org/abs/1812.04948

Summary

- Noise mapping allows for a more disentangled noise space
- The intermediate noise vector W is used as input to the generator

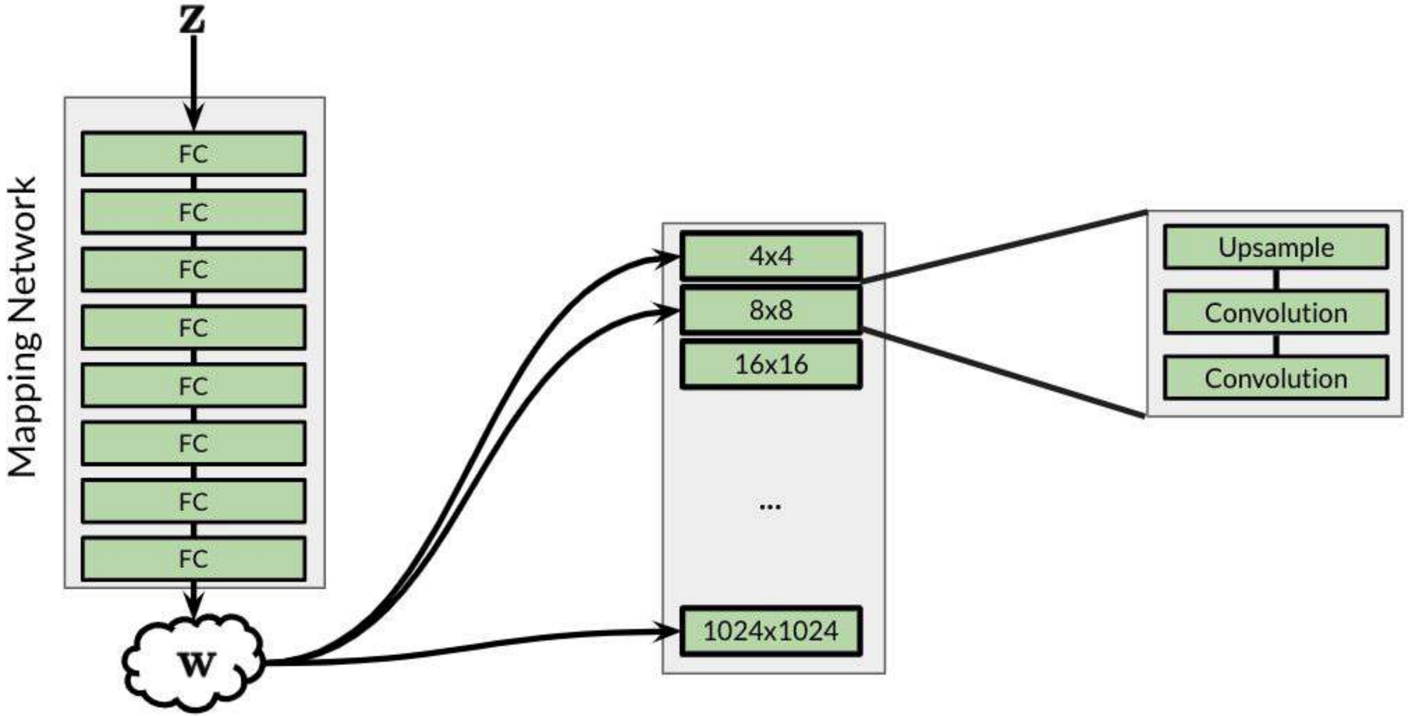
deeplearning.ai

Adaptive Instance Normalization (AdaIN)

Outline

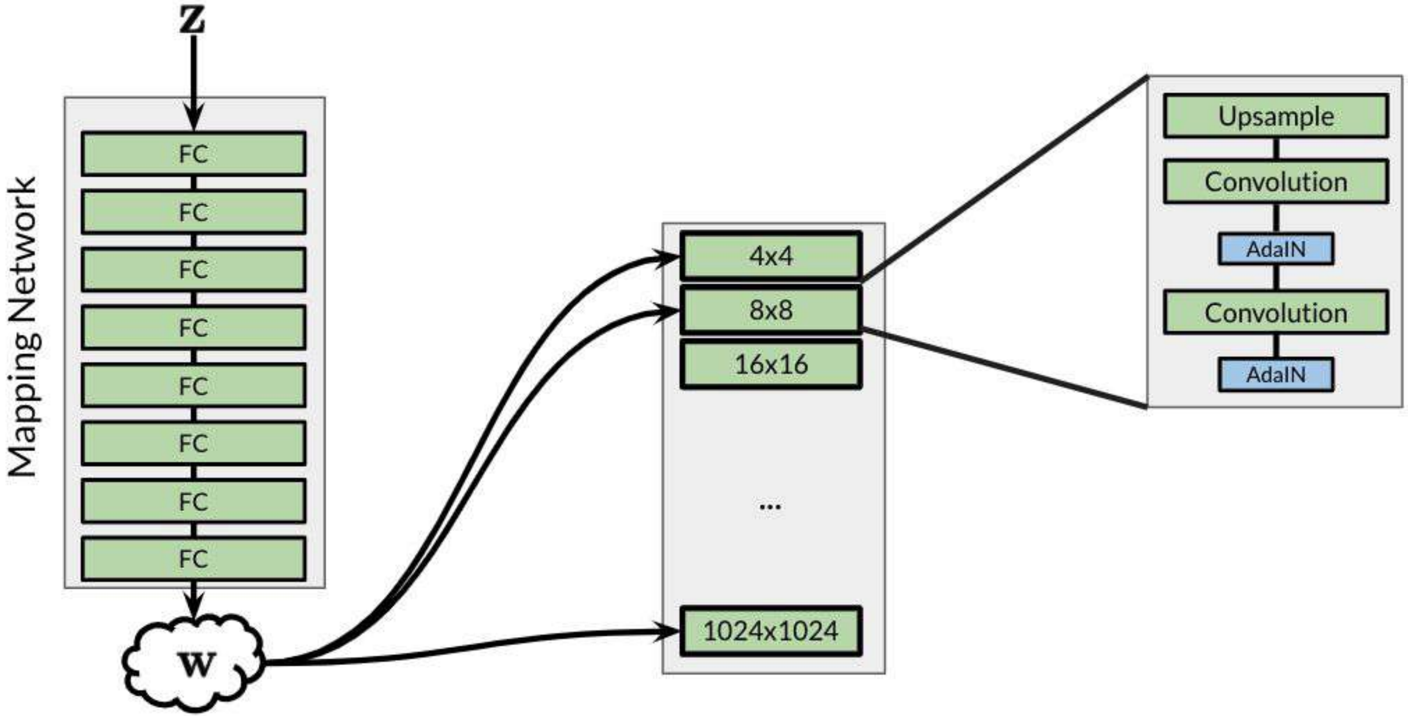
- Instance Normalization
- Adaptive Instance Normalization (AdaIN)
- Where and why AdaIN is used

AdaIN in Context



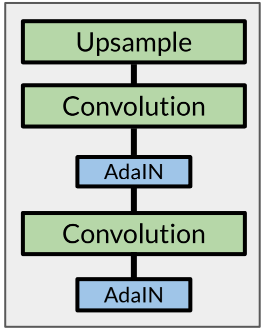
Based on: https://arxiv.org/abs/1812.04948

AdaIN in Context



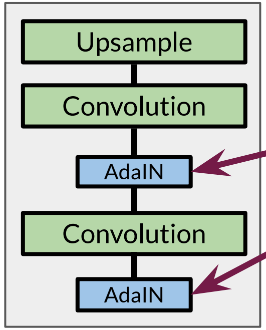
Based on: https://arxiv.org/abs/1812.04948

AdaIN



Based on: https://arxiv.org/abs/1812.04948

AdaIN

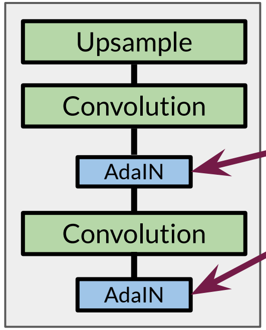


$$\frac{\mathbf{x} - \mu(\mathbf{x})}{\sigma(\mathbf{x})}$$

Step 1: Normalize convolution outputs

Based on: https://arxiv.org/abs/1812.04948

AdaIN



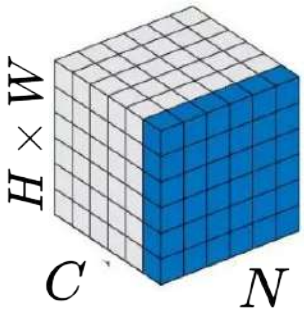
$$\frac{\mathbf{x} - \mu(\mathbf{x})}{\sigma(\mathbf{x})}$$

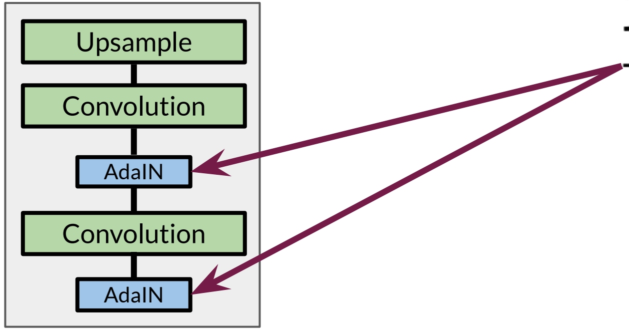
Step 1: Normalize convolution outputs using **Instance Normalization**

Based on: https://arxiv.org/abs/1812.04948

AdaIN

Batch norm





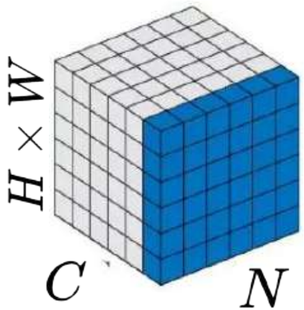
$$\frac{\mathbf{x} - \mu\left(\mathbf{x}\right)}{\sigma\left(\mathbf{x}\right)}$$

Step 1: Normalize convolution outputs using **Instance Normalization**

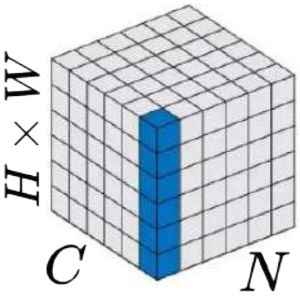
(Left) Available from: https://medium.com/syncedreview/facebook-ai-proposes-group-normalization-alternative-to-batch-normalization-fb0699bffae7
(Right) Based on: https://arxiv.org/abs/1812.04948

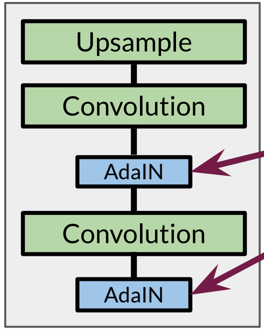
AdaIN

Batch norm



Instance norm





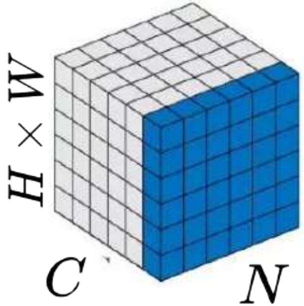
$$\frac{\mathbf{x} - \mu(\mathbf{x})}{\sigma(\mathbf{x})}$$

Step 1: Normalize convolution outputs using **Instance Normalization**

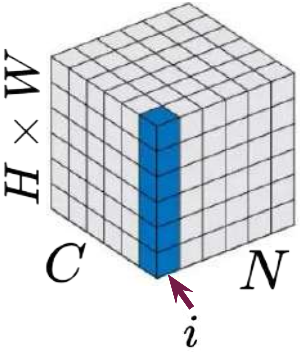
(Left) Available from: https://medium.com/syncedreview/facebook-ai-proposes-group-normalization-alternative-to-batch-normalization-fb0699bffae7
(Right) Based on: https://arxiv.org/abs/1812.04948

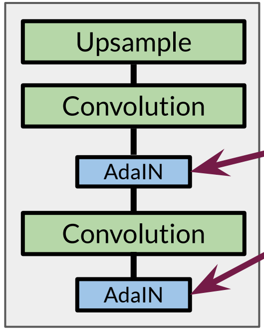
AdaIN

Batch norm



Instance norm



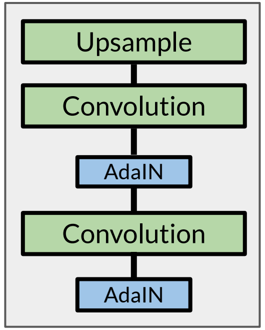


$$\frac{\mathbf{x}_i - \mu(\mathbf{x}_i)}{\sigma(\mathbf{x}_i)}$$

Step 1: Normalize convolution outputs using **Instance Normalization**

(Left) Available from: https://medium.com/syncedreview/facebook-ai-proposes-group-normalization-alternative-to-batch-normalization-fb0699bffae7
(Right) Based on: https://arxiv.org/abs/1812.04948

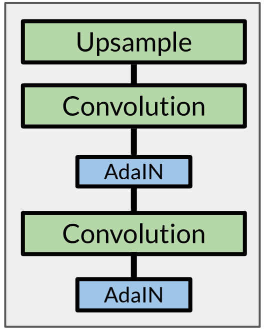
AdaIN



Step 2: Apply adaptive styles

Based on: https://arxiv.org/abs/1812.04948

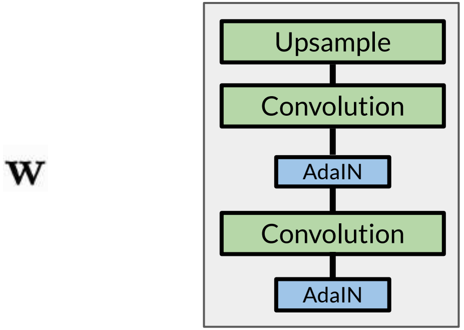
AdaIN



Step 2: Apply adaptive styles **using the intermediate noise vector**

Based on: https://arxiv.org/abs/1812.04948

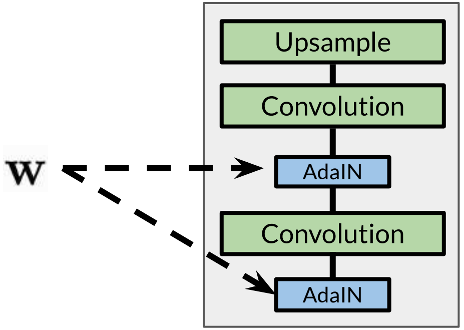
AdaIN



Step 2: Apply adaptive styles **using the intermediate noise vector**

Based on: https://arxiv.org/abs/1812.04948

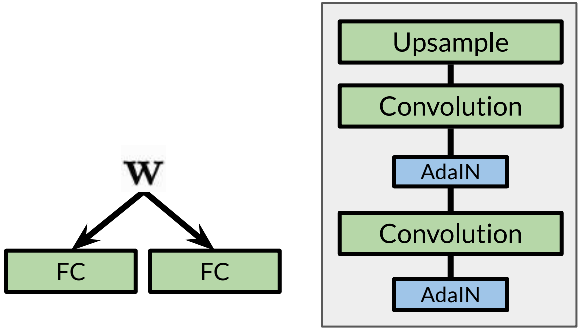
AdaIN



Step 2: Apply adaptive styles **using the intermediate noise vector**

Based on: https://arxiv.org/abs/1812.04948

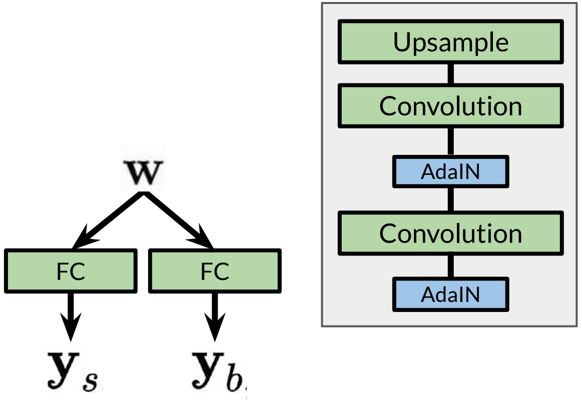
AdaIN



Step 2: Apply adaptive styles **using the intermediate noise vector**

Based on: https://arxiv.org/abs/1812.04948

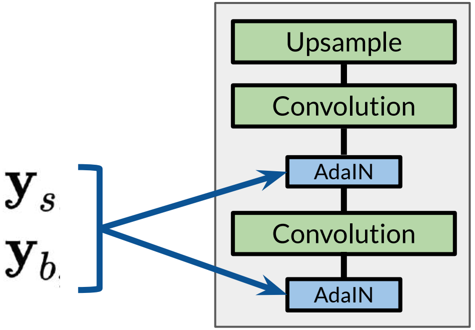
AdaIN



Step 2: Apply adaptive styles **using the intermediate noise vector**

Based on: https://arxiv.org/abs/1812.04948

AdaIN



Step 2: Apply adaptive styles **using the intermediate noise vector**

Based on: https://arxiv.org/abs/1812.04948

AdaIN

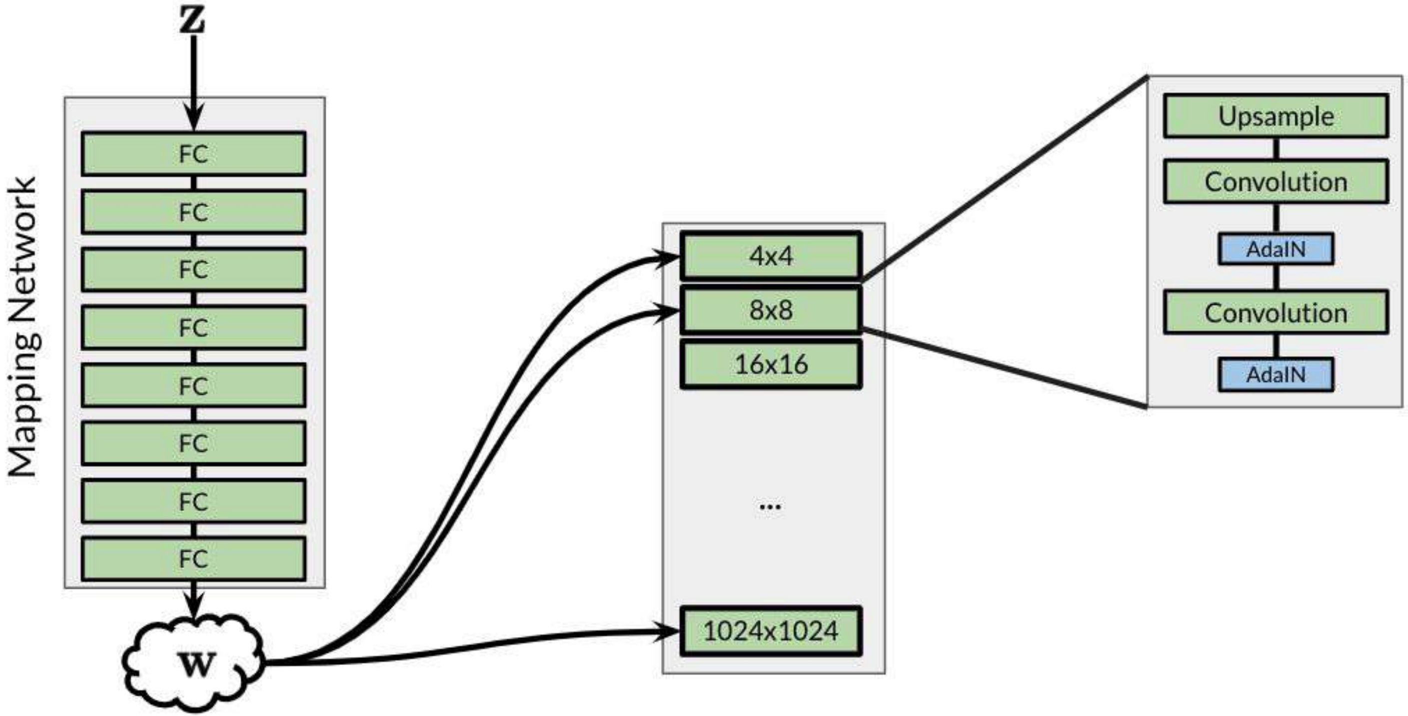
$$\mathrm{AdaIN}(\mathbf{x}_i, \mathbf{y}) = \mathbf{y}_{s,i} \underbrace{\frac{\mathbf{x}_i - \mu(\mathbf{x}_i)}{\sigma(\mathbf{x}_i)}}_{\text{Step 1: Instance normalization}} + \mathbf{y}_{b,i}$$

AdaIN

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

Step 2: Adaptive styles

AdaIN in Context



Based on: https://arxiv.org/abs/1812.04948

Summary

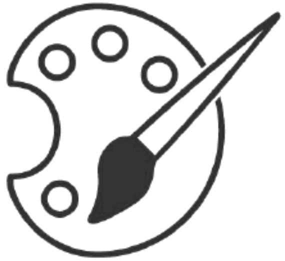
- AdaIN transfers style information onto the generated image from the intermediate noise vector $\mathbf{w}$
- Instance Normalization is used to normalize individual examples before apply style statistics from $\mathbf{w}$

deeplearning.ai

Style Mixing & Stochastic Noise

Outline

- Controlling coarse and fine styles with StyleGAN
- Style mixing for increased diversity during training/inference
- Stochastic noise for additional variation

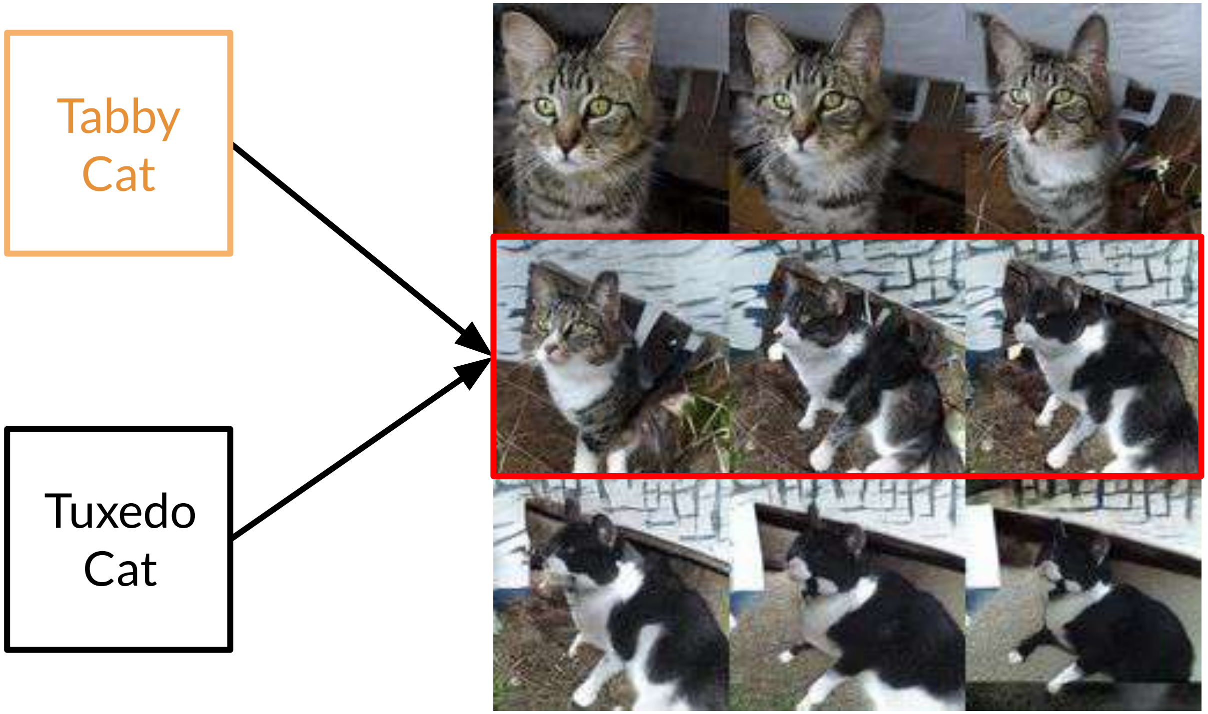


Style Mixing

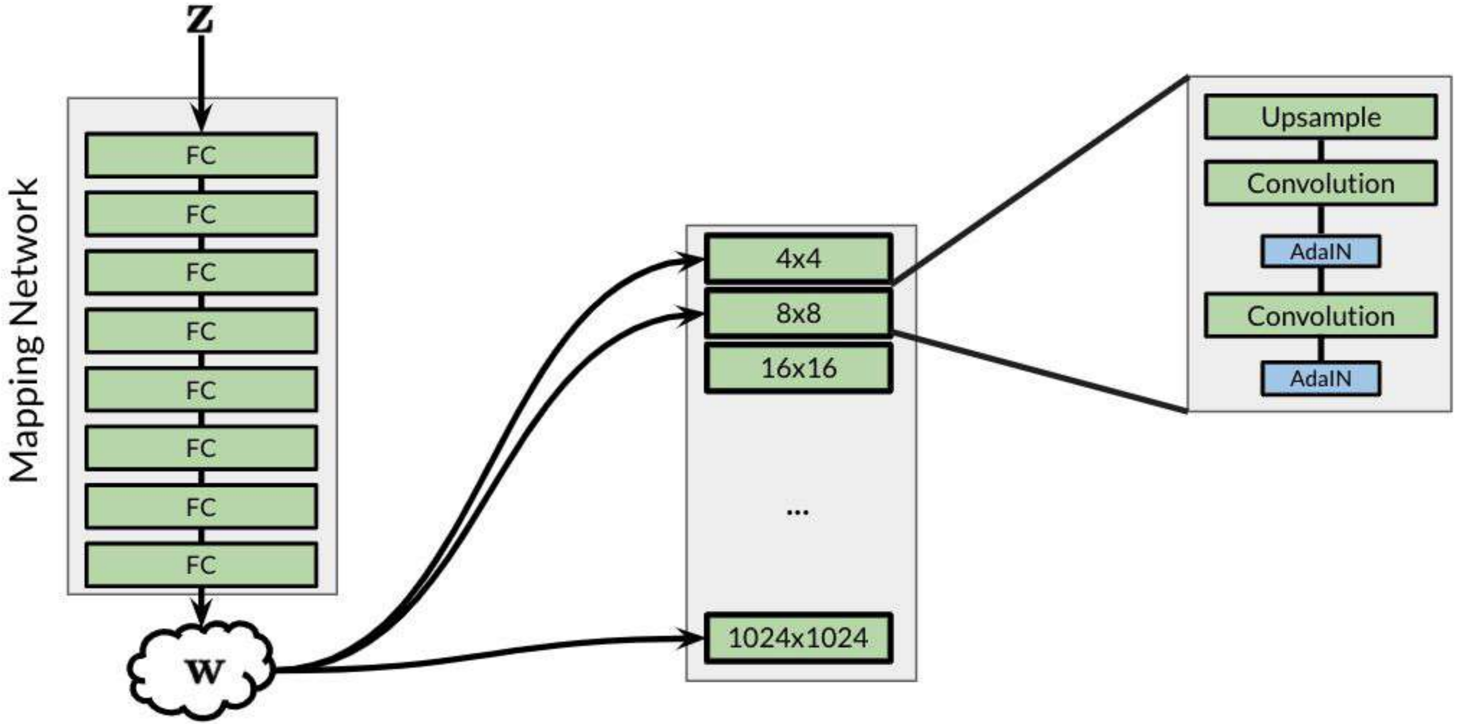
Tabby Cat

Tuxedo Cat

Style Mixing

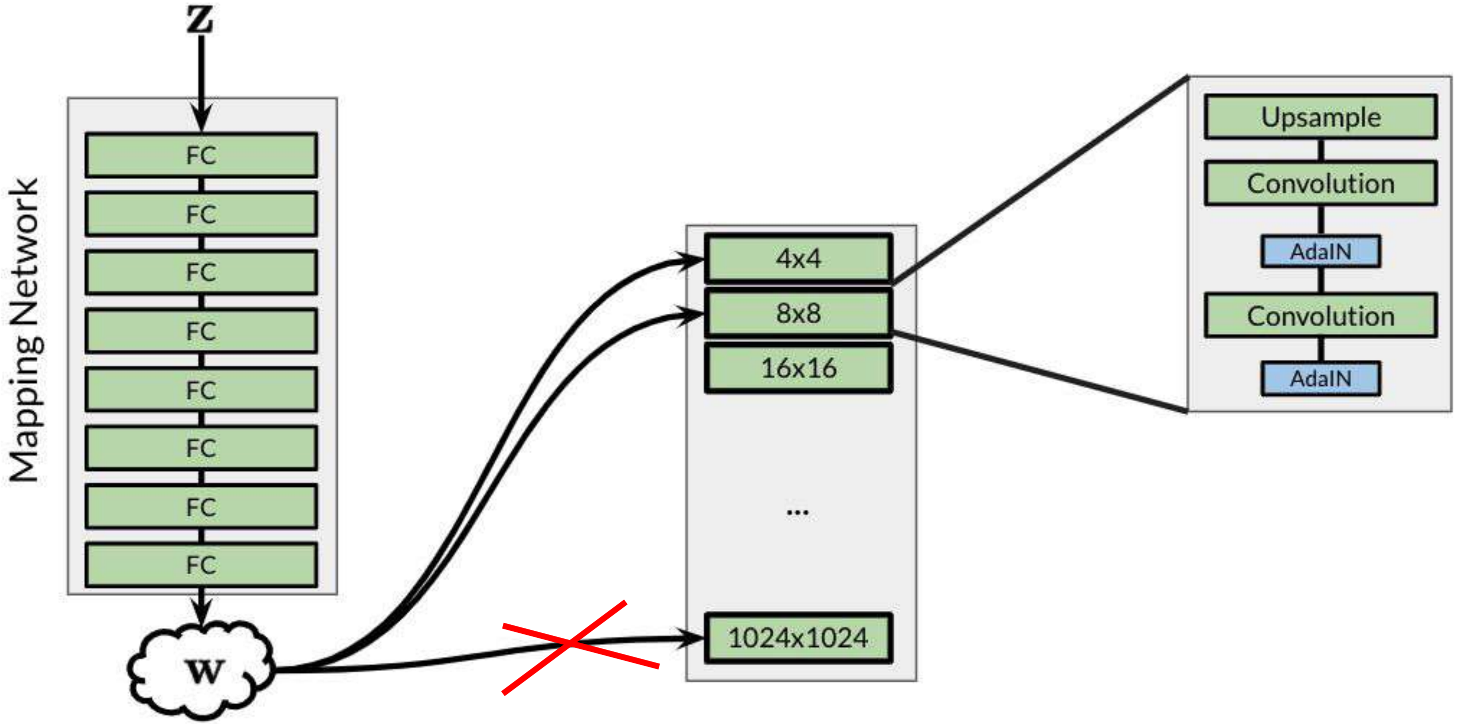


Style Mixing in Context



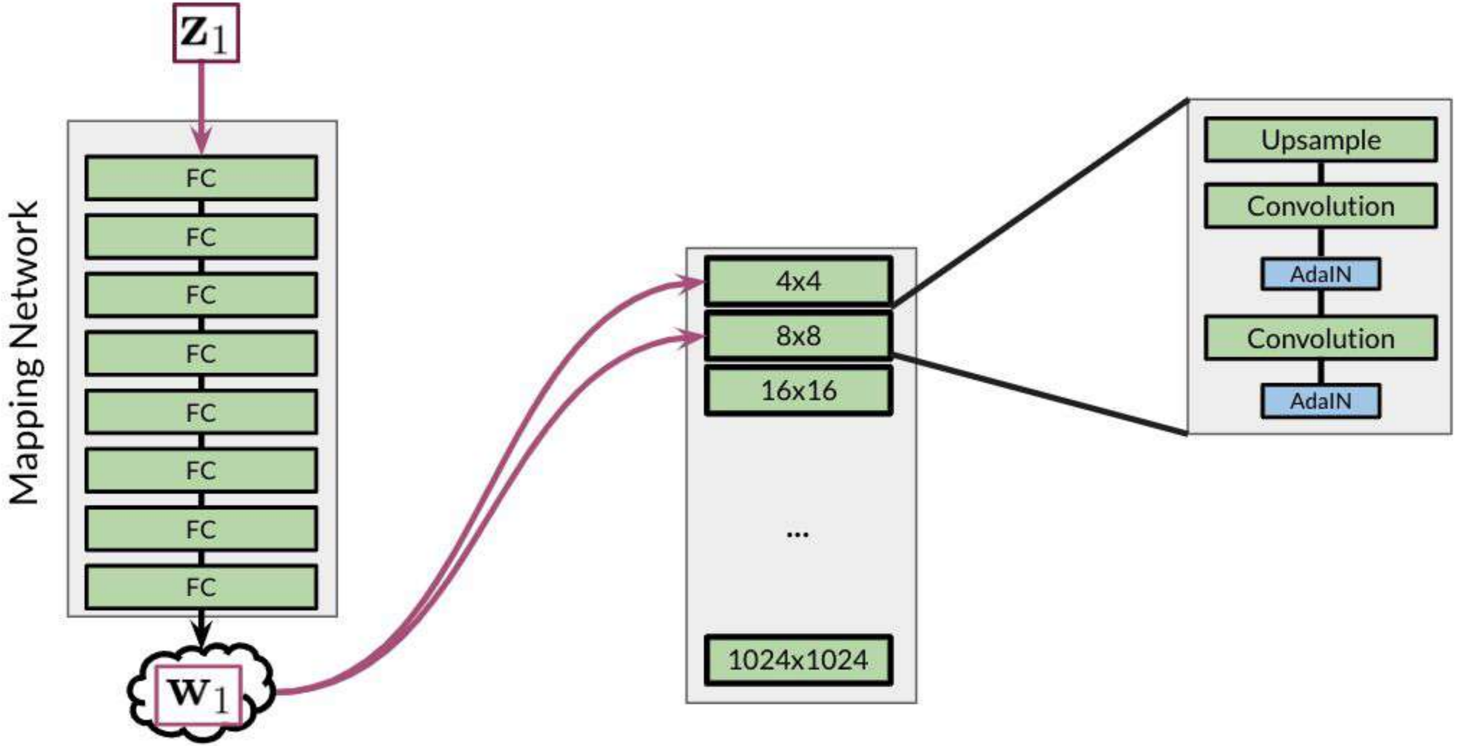
Based on: https://arxiv.org/abs/1812.04948

Style Mixing in Context



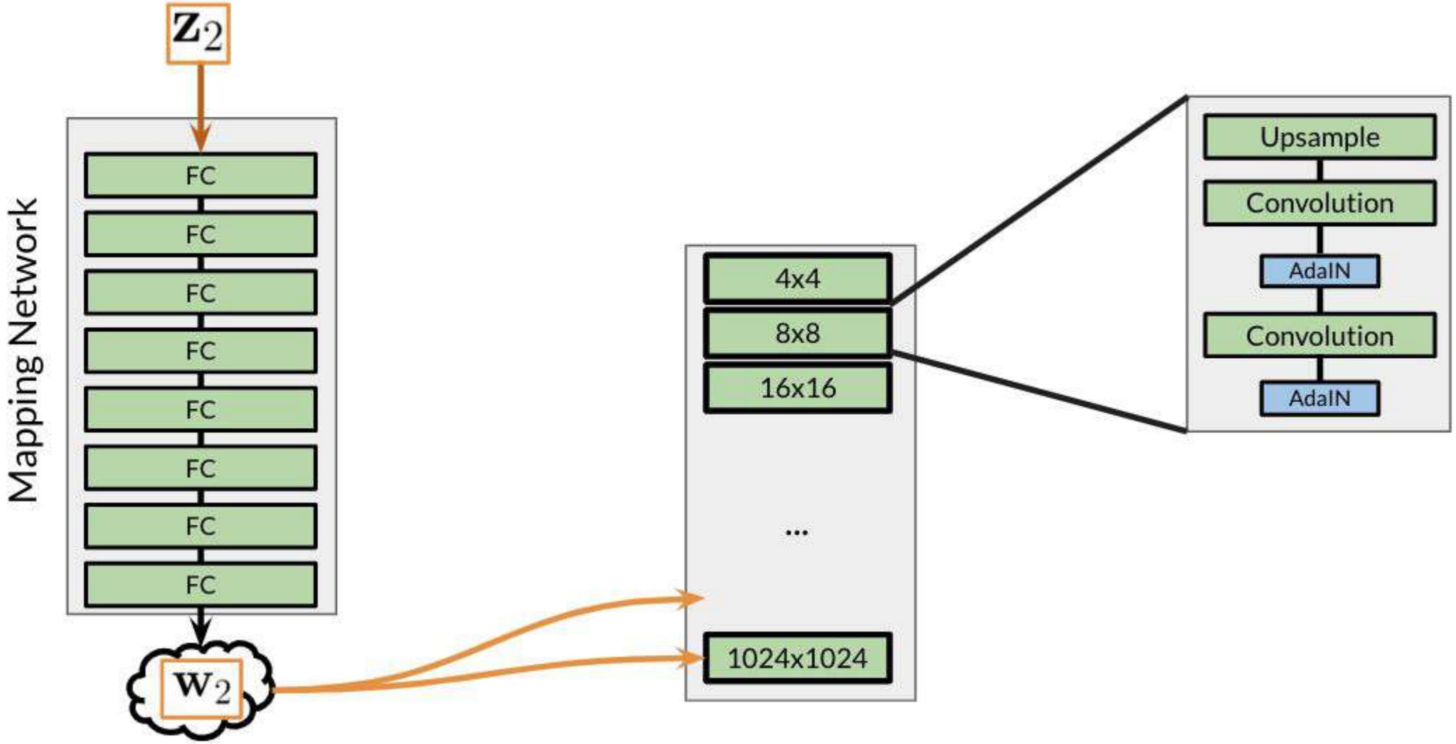
Based on: https://arxiv.org/abs/1812.04948

Style Mixing in Context



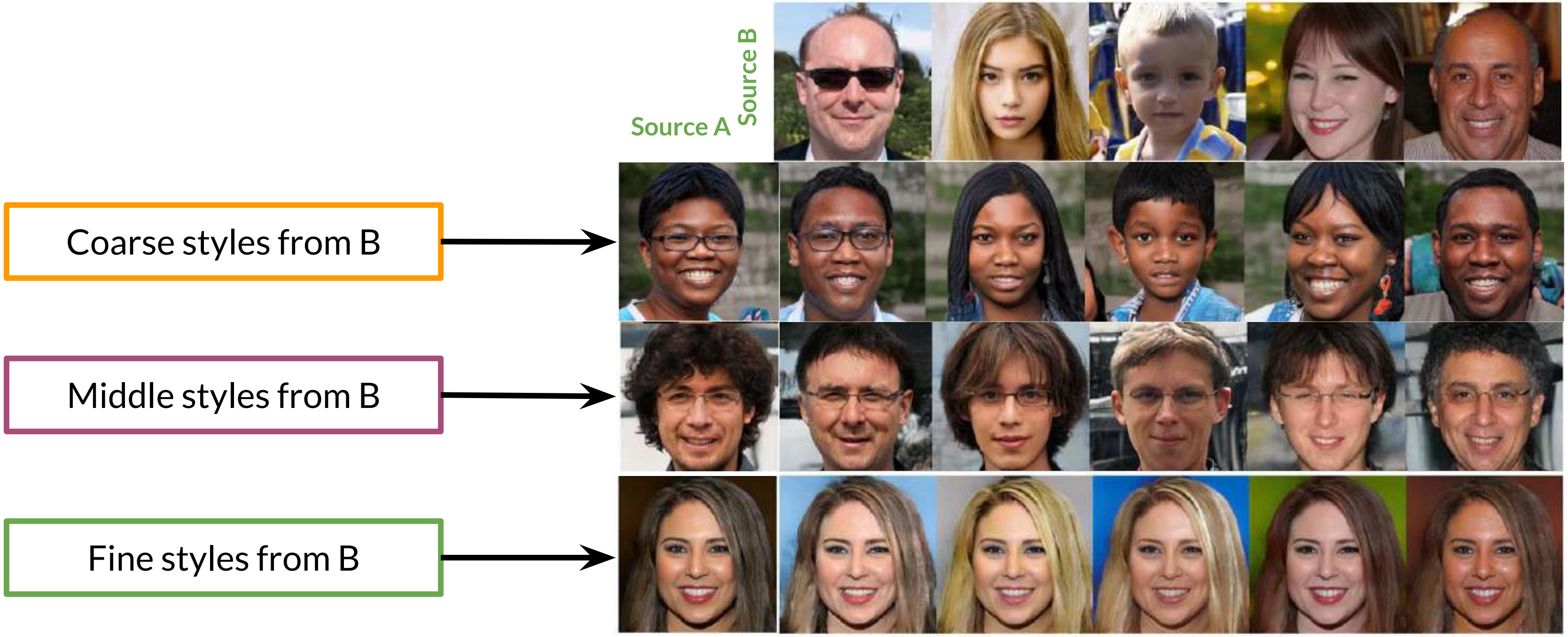
Based on: https://arxiv.org/abs/1812.04948

Style Mixing in Context



Based on: https://arxiv.org/abs/1812.04948

Style Mixing



Available from: https://arxiv.org/abs/1812.04948

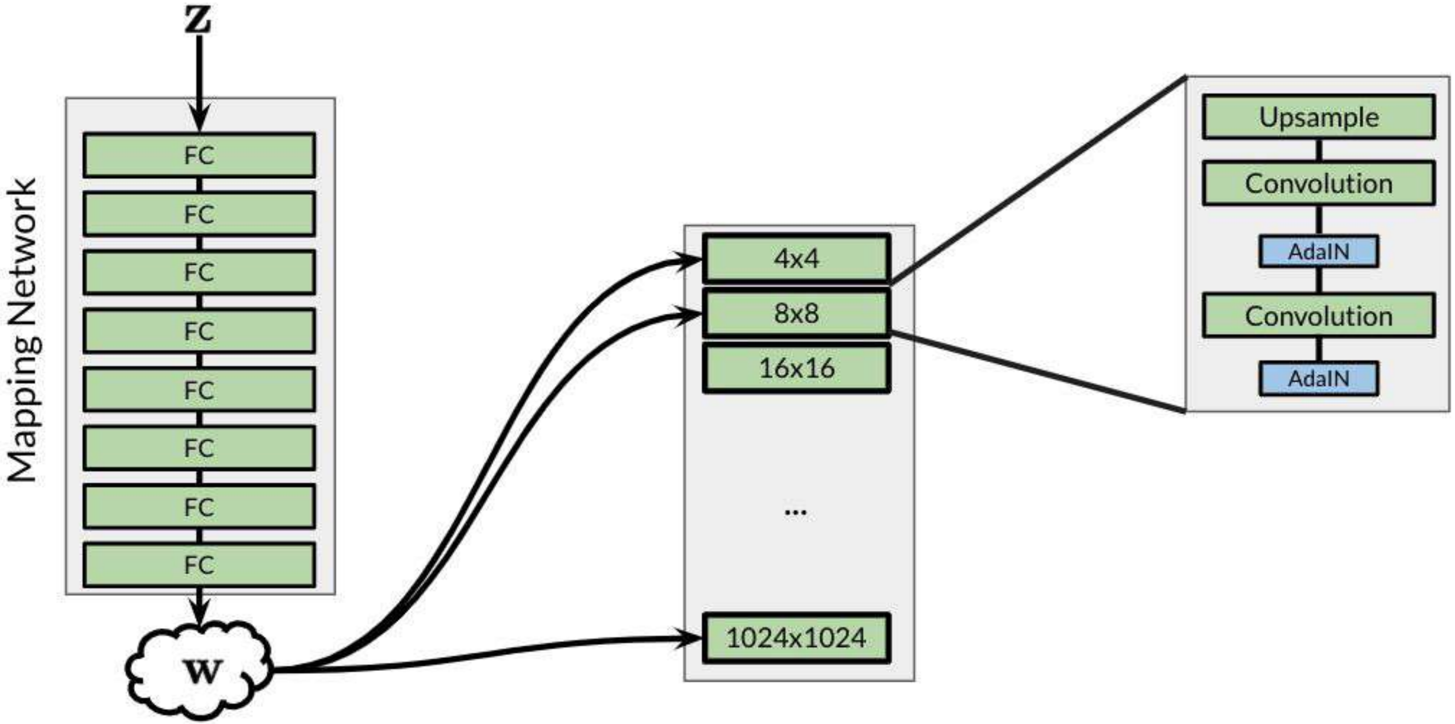
Stochastic Variation

Fine
layers

Coarse
layers

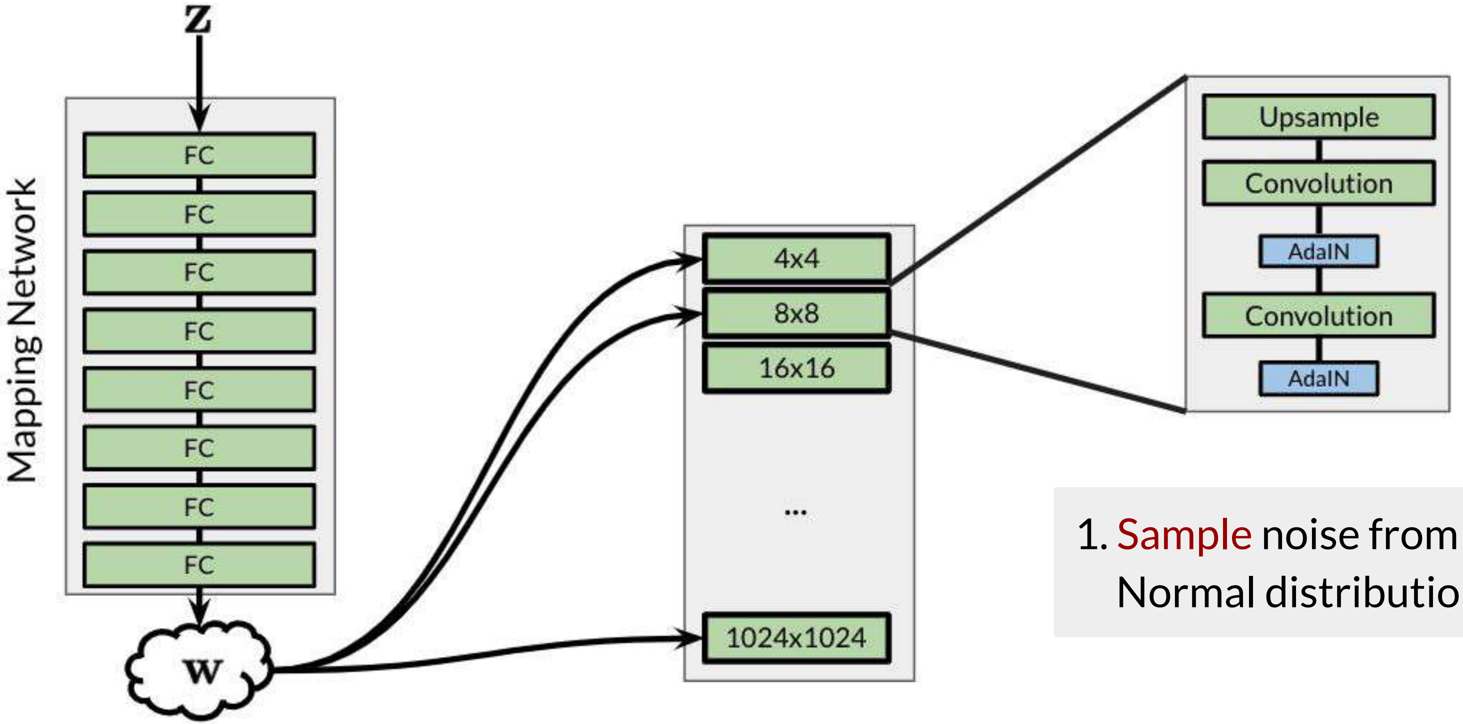
Available from: https://arxiv.org/abs/1812.04948

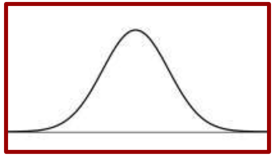
Stochastic Noise in Context



Based on: https://arxiv.org/abs/1812.04948

Stochastic Noise in Context

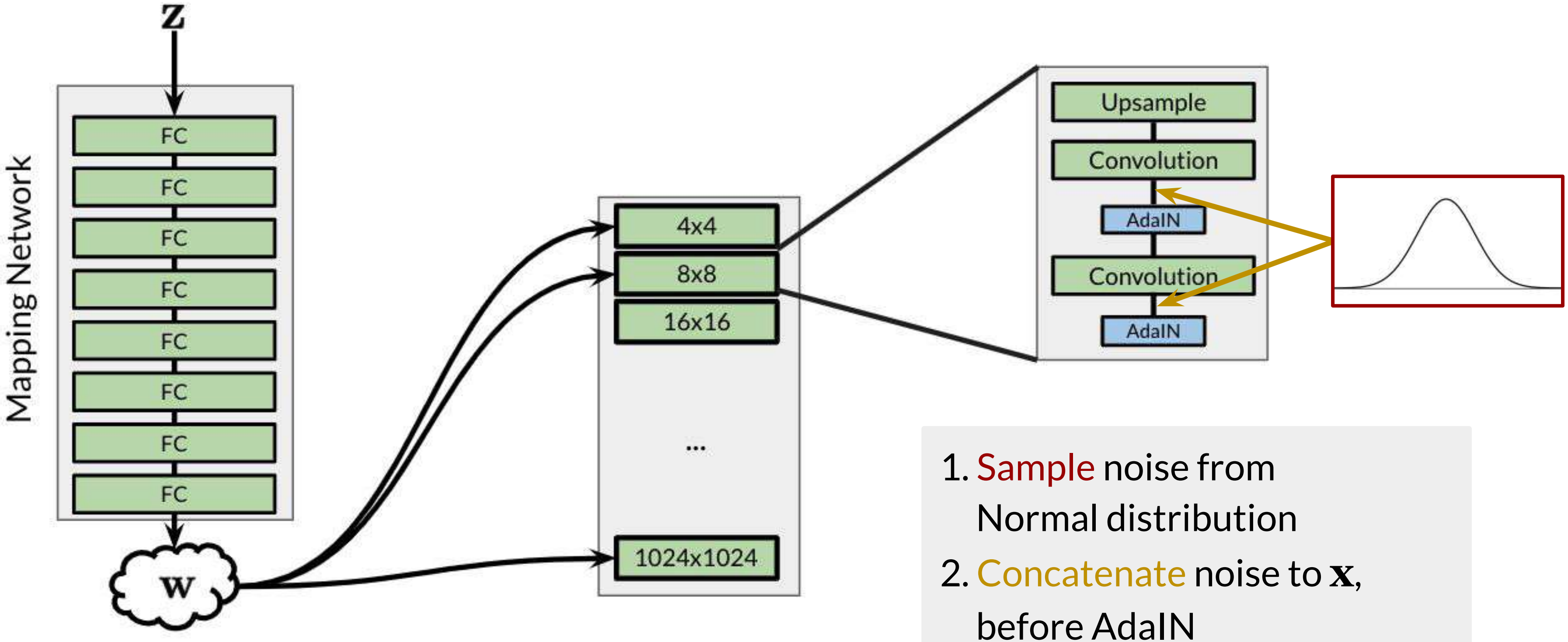




1. Sample noise from Normal distribution

Based on: https://arxiv.org/abs/1812.04948

Stochastic Noise in Context



Based on: https://arxiv.org/abs/1812.04948

1. Sample noise from Normal distribution
2. Concatenate noise to $\mathbf{x}$, before AdaIN

Stochastic Variation

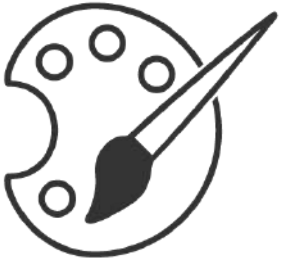
Small details: hair strands, wrinkles, etc.

Different extra noise values create stochastic variation

Available from: https://arxiv.org/abs/1812.04948

Summary

- Style mixing increases diversity that the model sees during training
- Stochastic noise causes small variations to output
- Coarse or fineness depends where in the network style or noise is added
    - Earlier for coarser variation
    - Later for finer variation

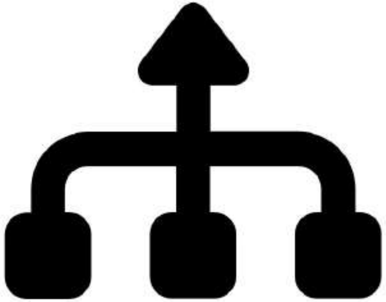


deeplearning.ai

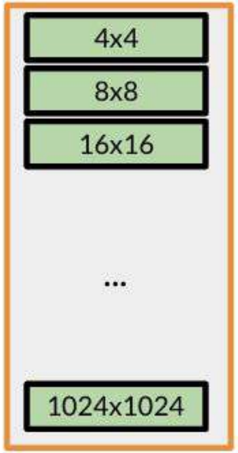
Putting It All Together

Outline

- Putting all the StyleGAN components together!

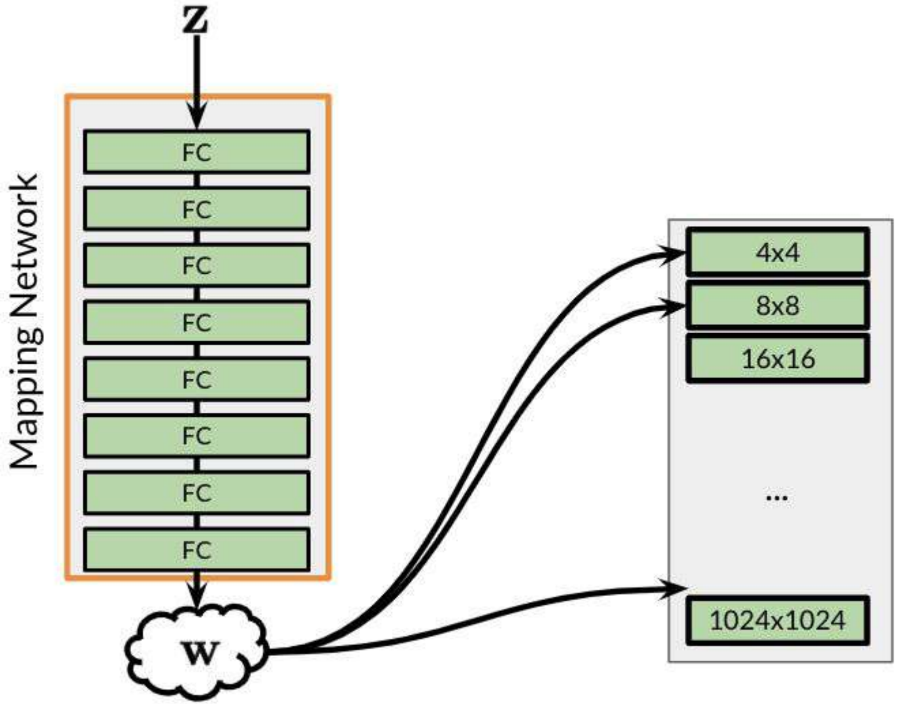


StyleGAN Architecture: Progressive Growing



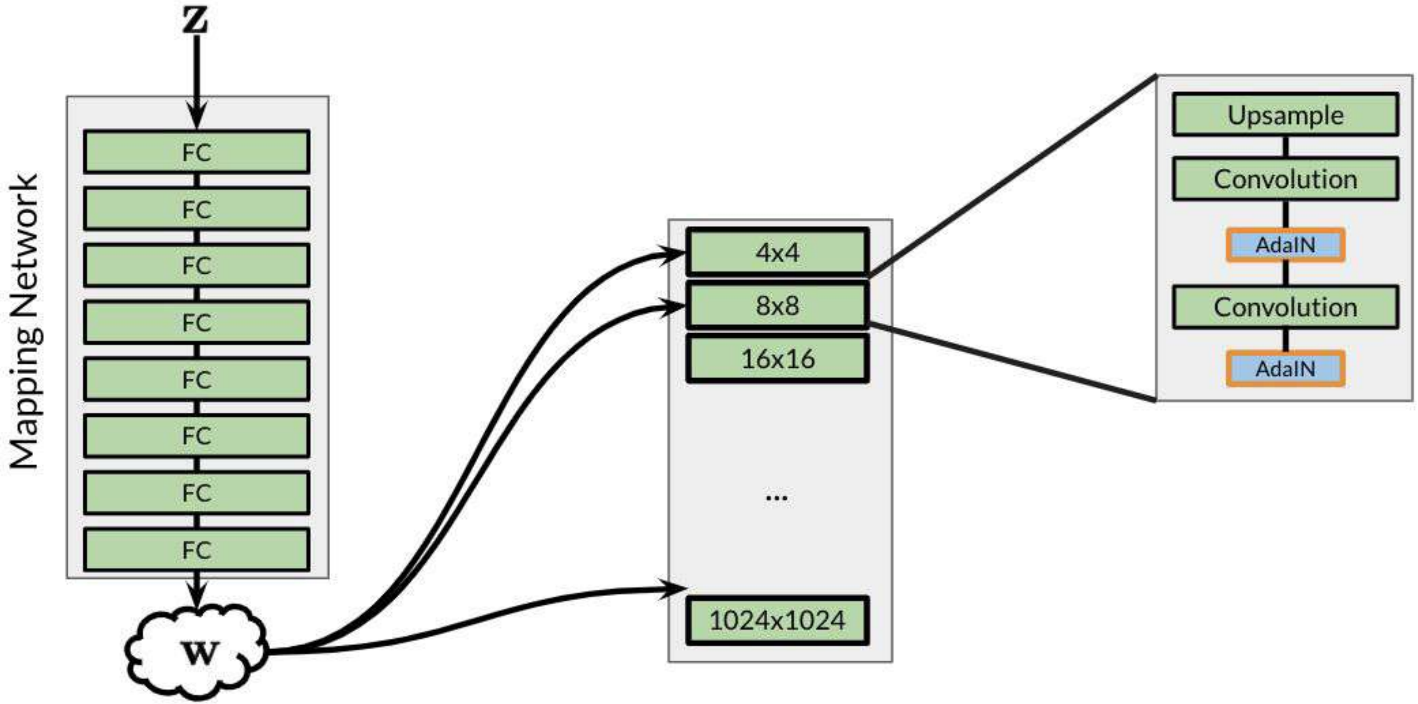
Based on: https://arxiv.org/abs/1812.04948

StyleGAN Architecture: Noise Mapping Network



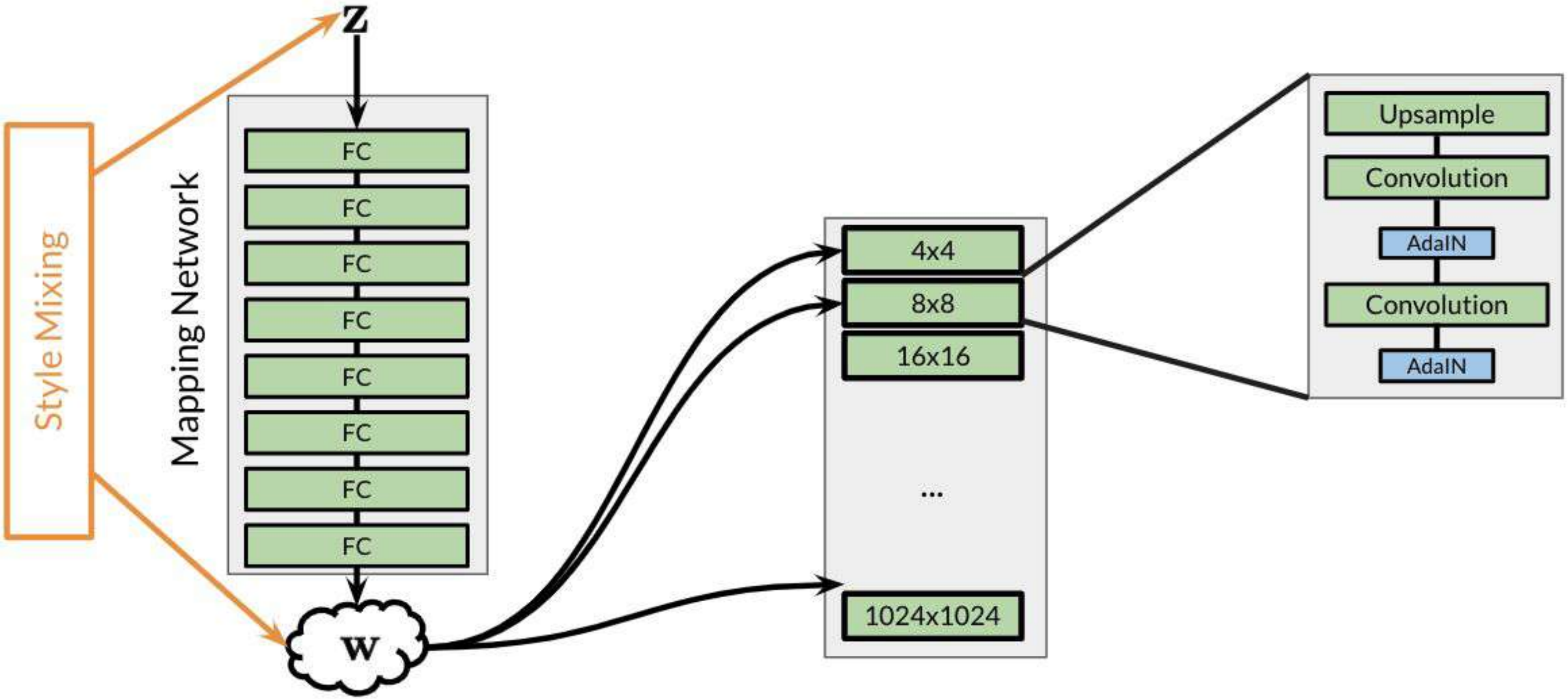
Based on: https://arxiv.org/abs/1812.04948

StyleGAN Architecture: AdaIN



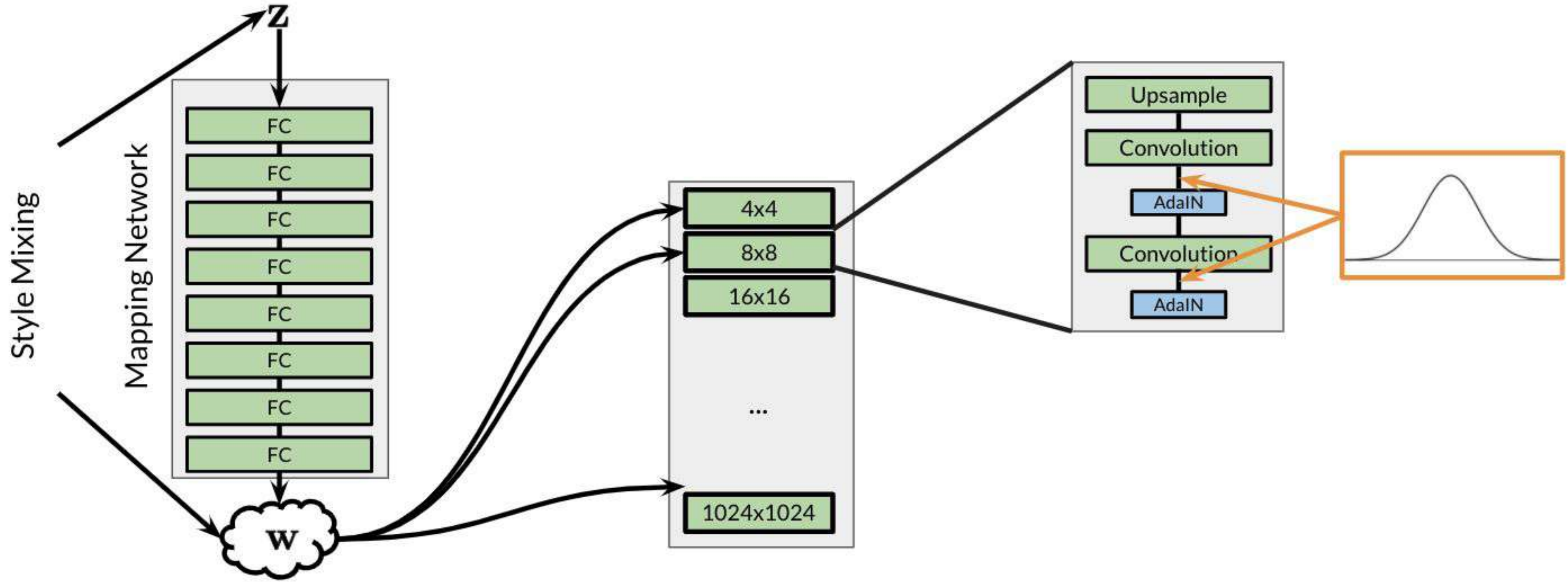
Based on: https://arxiv.org/abs/1812.04948

StyleGAN Architecture: Style Mixing



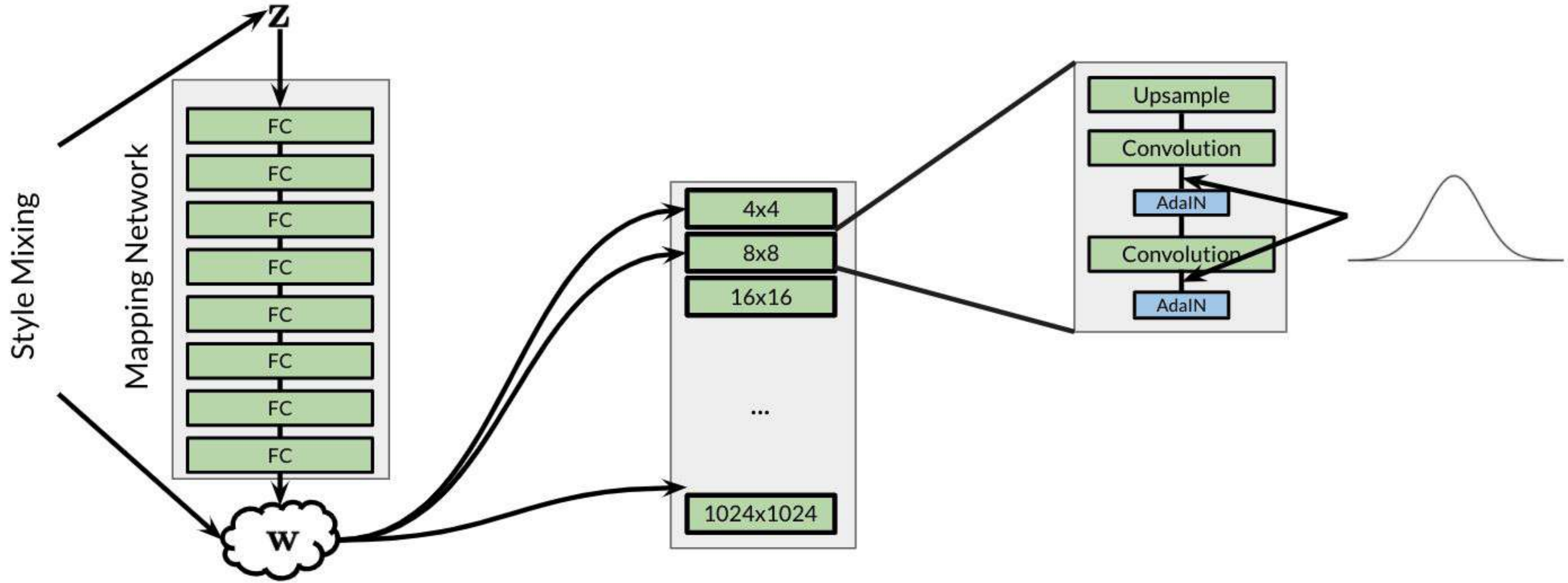
Based on: https://arxiv.org/abs/1812.04948

StyleGAN Architecture: Stochastic Noise



Based on: https://arxiv.org/abs/1812.04948

StyleGAN Architecture: That's a Wrap!



Based on: https://arxiv.org/abs/1812.04948

Summary

- Main components of StyleGAN:
  - Progressive Growing
  - Noise Mapping Network
  - AdaIN
  - Style Mixing
  - Stochastic Noise