M.Tech Program

Advanced Industry Integrated Programs

Jointly offered by University and LTIMindTree

Generative AI Fundamentals: Understanding Generative Adversarial Networks

Knowledge partner

LTIMindtree

Implementation partner

L&T EduTech

Modules to cover….

1. Build Basic Generative Adversarial Networks

2. **Build Better Generative Adversarial Networks**

3. Apply Generative Adversarial Networks

Build Better Generative Adversarial Network

Build Better Generative Adversarial Networks

GAN Evaluation

1. **No Universal Gold Standard**

   Discriminator overfits to to the generator it was trained with.

2. **Fidelity vs. Diversity Diversity**

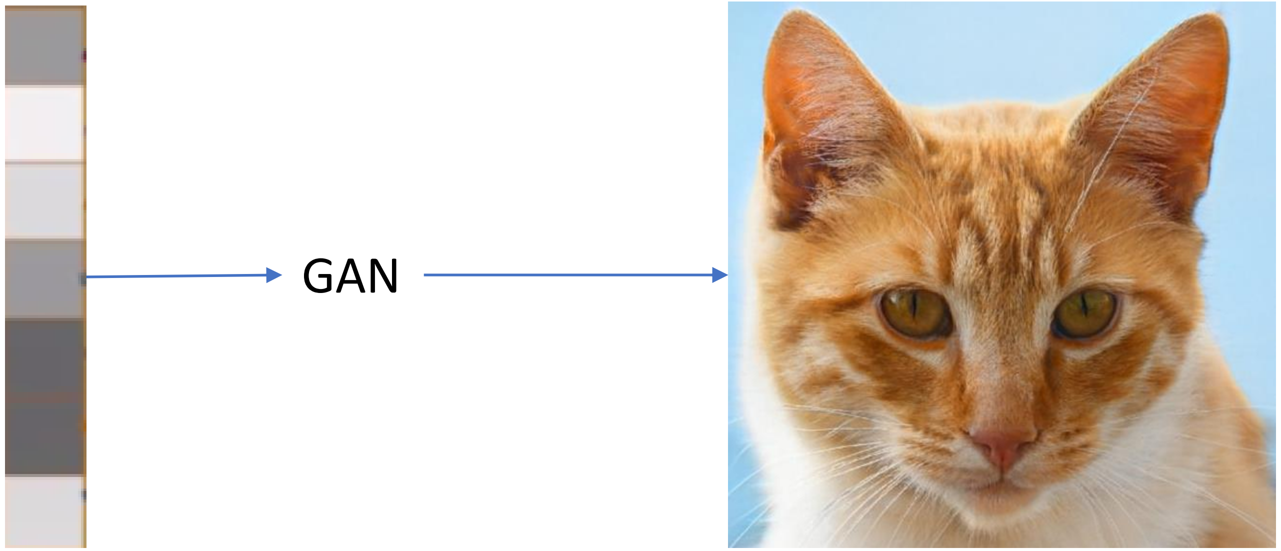
   Measuring both image quality and variety is difficult.

3. **No Ground Truth**

   There is no definitive way to judge the "realness" of generated images.

Build Better Generative Adversarial Networks

GAN Evaluation



- No ground truth to evaluation
- We can’t use the discriminator to evaluate the generator, that will cause the over fitting issue

Build Better Generative Adversarial Networks

Understand the Challenges of evaluating GANs

Fidelity

Quality of Images

How well the generator creates realistic images

Diversity

Variety of images

The variety of images the generator can produce.

Build Better Generative Adversarial Networks

Understand the Challenges of evaluating GANs

Comparing Real and Fake Images

Pixel Distance

Simple but insufficient for evaluating GANs.

Build Better Generative Adversarial Networks

Understand the Challenges of evaluating GANs

Pixel Distance

$$\left| \underbrace{\begin{bmatrix} 200 & 50 & 200 \\ 200 & 50 & 200 \\ 200 & 50 & 200 \end{bmatrix}}_{\text{Real}} - \underbrace{\begin{bmatrix} 200 & 50 & 200 \\ 200 & 50 & 200 \\ 200 & 50 & 200 \end{bmatrix}}_{\text{Fake}} \right| = \underbrace{\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}}_{\text{Absolute difference}}$$

Build Better Generative Adversarial Networks

Understand the Challenges of evaluating GANs

Pixel Distance Issue – shift of pixels

$$\left| \underbrace{\begin{matrix} 200 & 50 & 200 \\ 200 & 50 & 200 \\ 200 & 50 & 200 \end{matrix}}_{\textit{Real}} - \underbrace{\begin{matrix} 50 & 200 & 200 \\ 50 & 200 & 200 \\ 50 & 200 & 200 \end{matrix}}_{\textbf{Fake}} \right|$$

Build Better Generative Adversarial Networks

Understand the Challenges of evaluating GANs

Comparing Real and Fake Images

Pixel Distance

Simple but insufficient for evaluating GANs.

Feature Distance

More reliable for comparing real and fake images.

Build Better Generative Adversarial Networks

Understand the Challenges of evaluating GANs

Feature Distance

Real

2 eyes,
2 droopy ears,
1 nose, ...

Fake

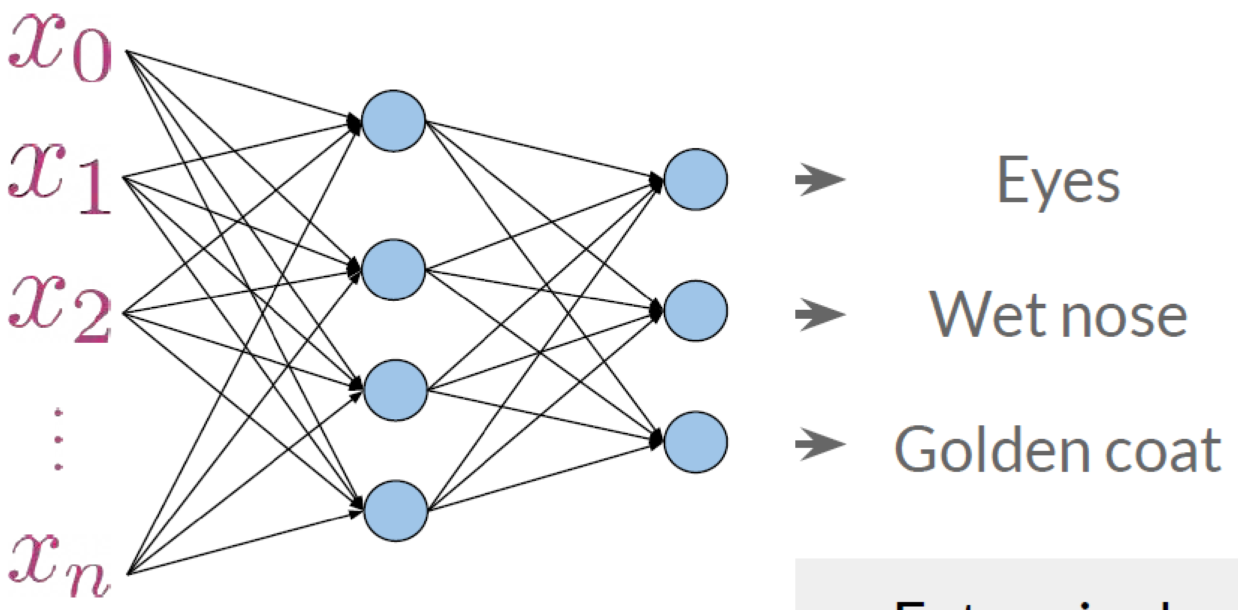
2 eyes,
1 droopy ear,
5 legs,
1 nose, ...

Compare features!

Build Better Generative Adversarial Networks

Understand the Challenges of evaluating GANs

Feature Extraction

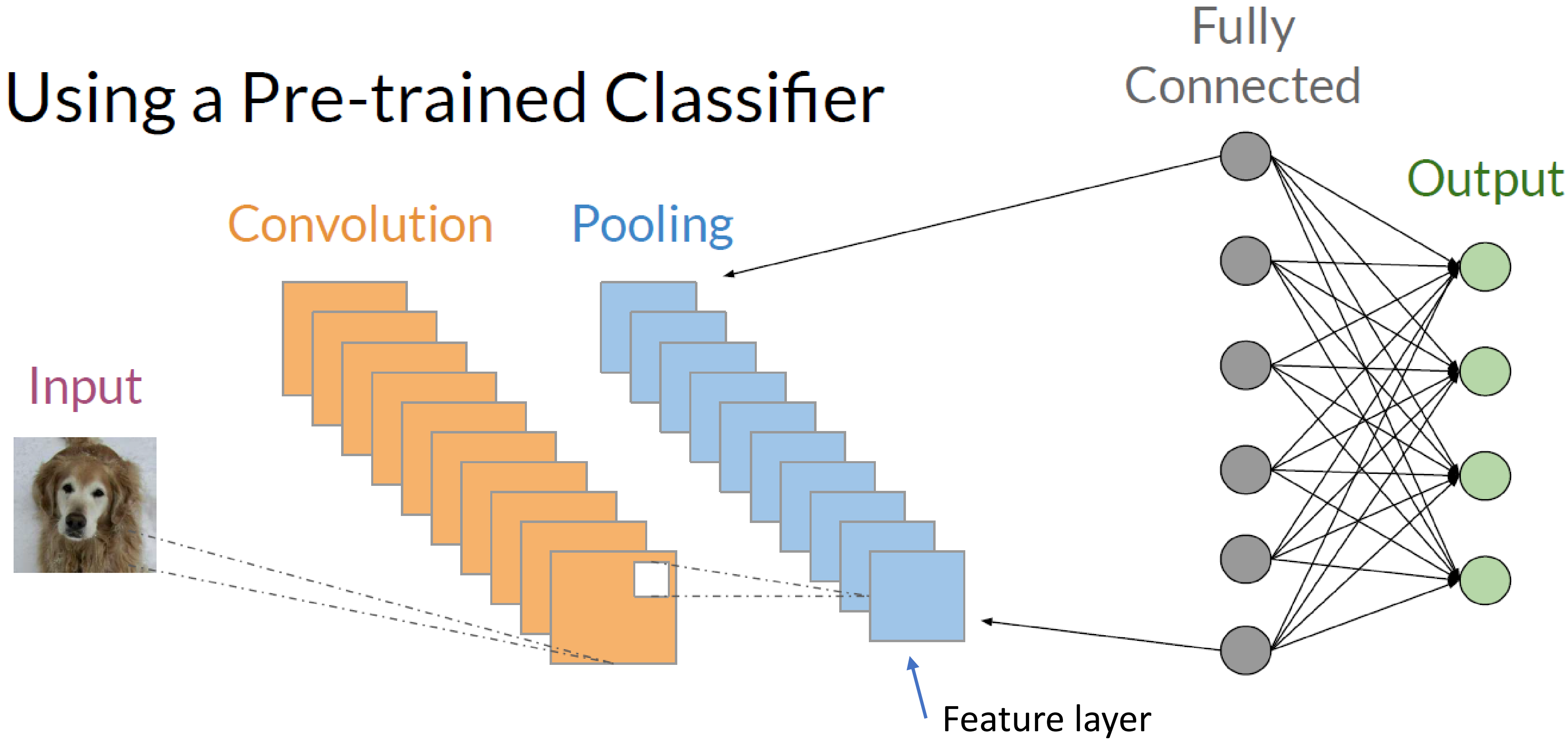


Extensively pre-trained classifiers available to use

Build Better Generative Adversarial Networks

Understand the Challenges of evaluating GANs

Feature Extraction



Build Better Generative Adversarial Networks

Understand the Challenges of evaluating GANs

Feature Extraction

1 **Pre-Trained Classifiers**

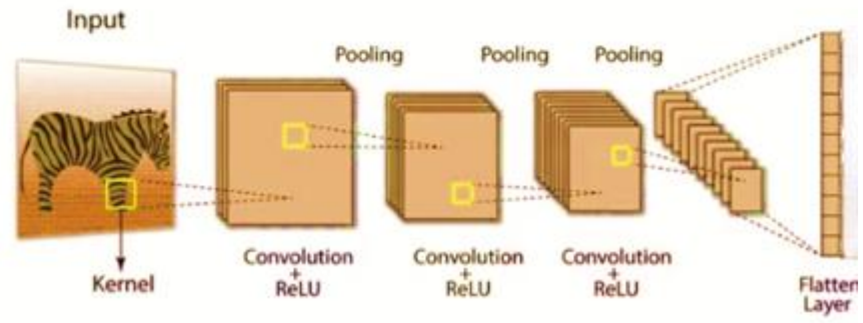
Use last layer for most information, earlier layers for primitives.

2 **ImageNet**

Large dataset of 14M images across 20K categories.

3 **Inception-v3**

Common feature extractor pre-trained on ImageNet with output layer cut off.



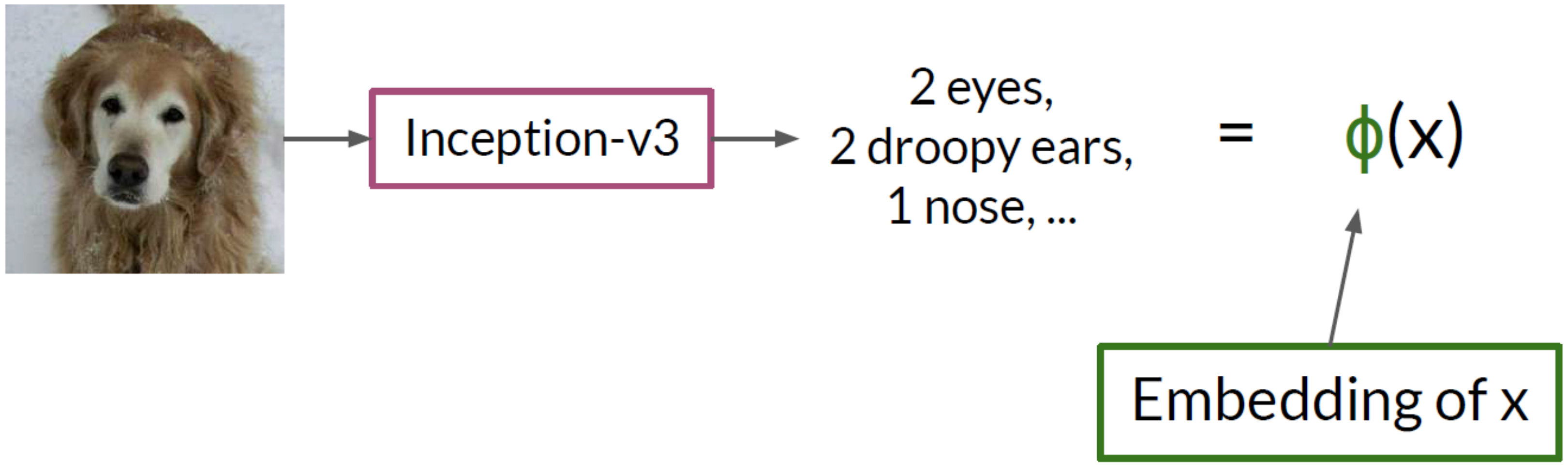
Source: https://www.researchgate.net/publication/

Build Better Generative Adversarial Networks

Understand the Challenges of evaluating GANs

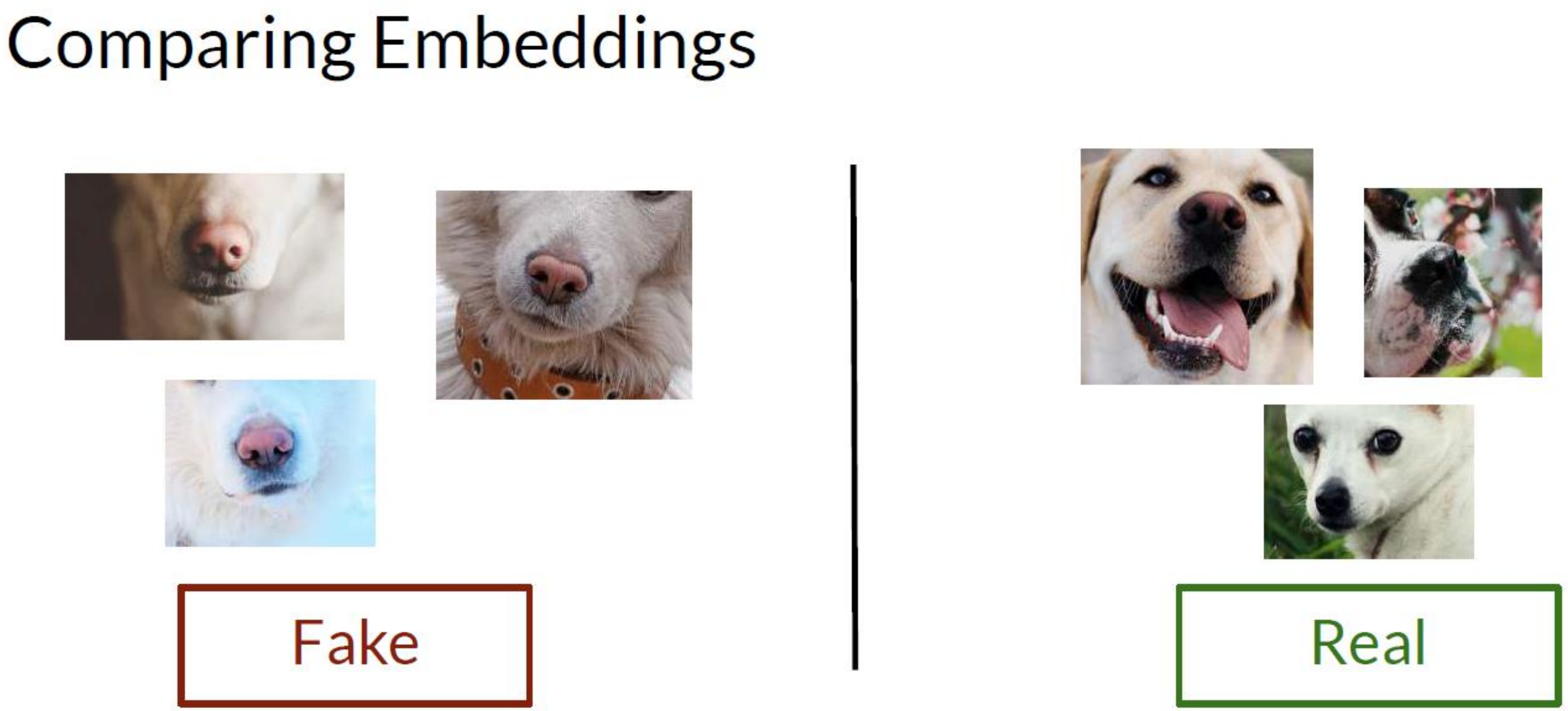
Feature Extraction

Embeddings



Build Better Generative Adversarial Networks

Understand the Challenges of evaluating GANs

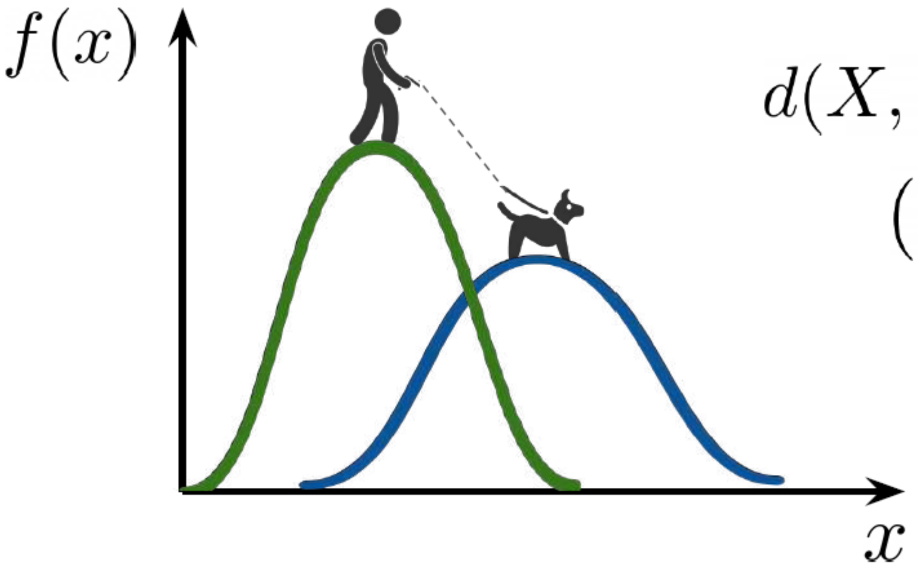


Build Better Generative Adversarial Networks

Understand the Challenges of evaluating GANs

FID

Fréchet Distance Between Normal Distributions



$$d(X,Y) = (\mu_X - \mu_Y)^2 + (\sigma_X - \sigma_Y)^2$$

Mean

Standard deviation

Build Better Generative Adversarial Networks

Understand the Challenges of evaluating GANs

FID

Multivariate Normal Fréchet Distance

Univariate Normal Fréchet Distance =

$$(\mu_X - \mu_Y)^2 + (\sigma_X - \sigma_Y)^2$$

Multivariate Normal Fréchet Distance =

$$\|\mu_X - \mu_Y\|^2 + \mathrm{Tr}\left(\Sigma_X + \Sigma_Y - 2\sqrt{\Sigma_X \Sigma_Y}\right)$$

Build Better Generative Adversarial Networks

Understand the Challenges of evaluating GANs

FID

Fréchet Inception Distance (FID)

Lower FID = closer distributions

$$\text{FID} = \|\mu_X - \mu_Y\|^2 + \text{Tr}\left(\Sigma_X + \Sigma_Y - 2\sqrt{\Sigma_X \Sigma_Y}\right)$$

Real and fake embeddings are two multivariate normal distributions

Use *large sample size* to reduce noise

Build Better Generative Adversarial Networks

Fréchet Inception Distance method using embedding

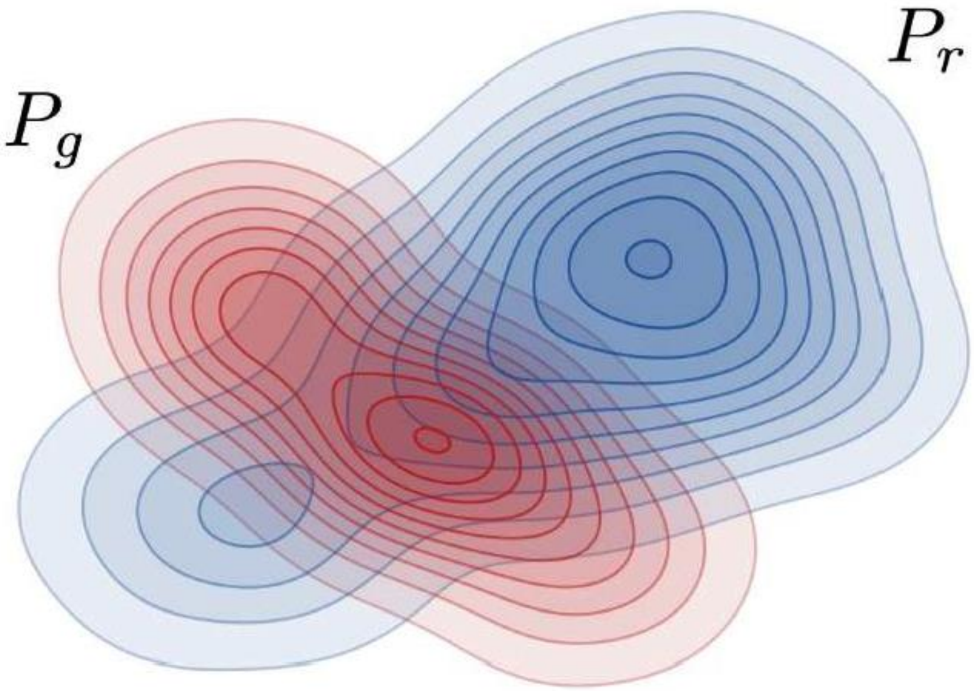
Shortcomings of FID

- Relies on a pre-trained model that might miss some details.
- Not as good with small amounts of data.
- Performs slowly.
- Uses only mean and covariance.

Build Better Generative Adversarial Networks

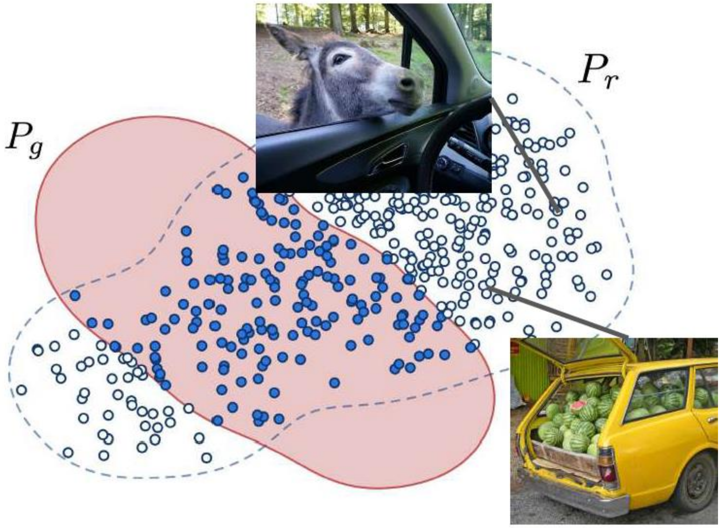
Recall and Precision

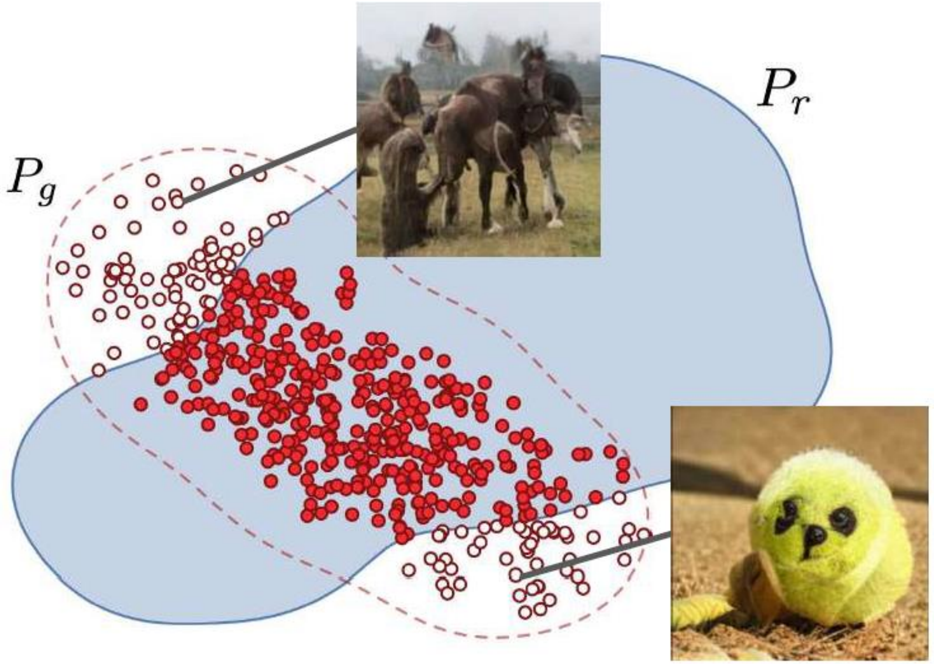
Precision and Recall



Build Better Generative Adversarial Networks

Recall and Precision





Recall

Measures diversity, generator tends to excel.

Precision

Measures fidelity, truncation trick can improve.

Build Better Generative Adversarial Networks

Learn about the advantages and Disadvantages of different GAN performance Measures

Fréchet Inception Distance (FID):

- Advantages:
1. Captures both image quality and diversity.
2. Robust to dataset variations.
- Disadvantages:
1. Computationally expensive.
2. Requires pre-trained inception model.

Precision and Recall:

- Advantages:
1. Evaluates specific aspects of image quality.
2. Provides insights into mode collapse.
- Disadvantages:
1. Depends on threshold selection.
2. May not fully capture image quality.

Implement the Fréchet Inception Distance FID

Method using embeddings to assess the accuracy of GANs

Build Better Generative Adversarial Networks

Fréchet Inception Distance method using embedding

Knowledge Check

When calculating the FID score, what type of statistical properties are compared between real and generated images?

a) First-order properties

b) Second-order properties

c) Both first and second-order properties

d) Third-order properties

Build Better Generative Adversarial Networks

Fréchet Inception Distance method using embedding

Knowledge Check

When calculating the FID score, what type of statistical properties are compared between real and generated images?

a) First-order properties (Mean)

b) Second-order properties (Covariance)

c) Both first and second-order properties

d) Third-order properties (Skewness)

GAN Disadvantages

Build Better Generative Adversarial Networks

GAN Disadvantages and Bias

Lack of Intrinsic Evaluation Metrics

GANs lack intrinsic evaluation metrics, making it difficult to assess the quality of the generated outputs.

Unstable Training

The training process of GANs can be unstable, leading to challenges in convergence and model performance.

No Density Estimation

GANs do not provide a way to estimate the underlying probability distribution of the data, which can be a limitation in certain applications.

Difficulty in Inverting

Inverting the GAN generator to obtain the original input from the generated output is not straightforward.

Instability During Training

Despite the high fidelity of the generated outputs, there can be instability during the training process of GANs.

Build Better Generative Adversarial Networks

Find out the disadvantages of GANS When compared to other generative models

Alternatives of GAN

Variational Autoencoders (VAEs)

VAEs are an alternative to GANs that aim to model the underlying probability distribution of the data, providing density estimation and invertibility.

Autoregressive Models

Autoregressive models rely on previous pixels to generate the next pixel, offering a different approach to generative modeling.

Flow Models

Flow models use invertible mappings to transform the data, providing a way to model the underlying probability distribution.

Build Better Generative Adversarial Networks

Find out the disadvantages of GANS When compared to other generative models

Modeling P(feature | class)

These alternative approaches to GANs try to model the conditional probability of features given a class, rather than directly generating the data.

$$\underbrace{\xi}_{\text{Noise}}, \underbrace{Y}_{\text{Class}} \to \underbrace{X}_{\text{Features}}$$

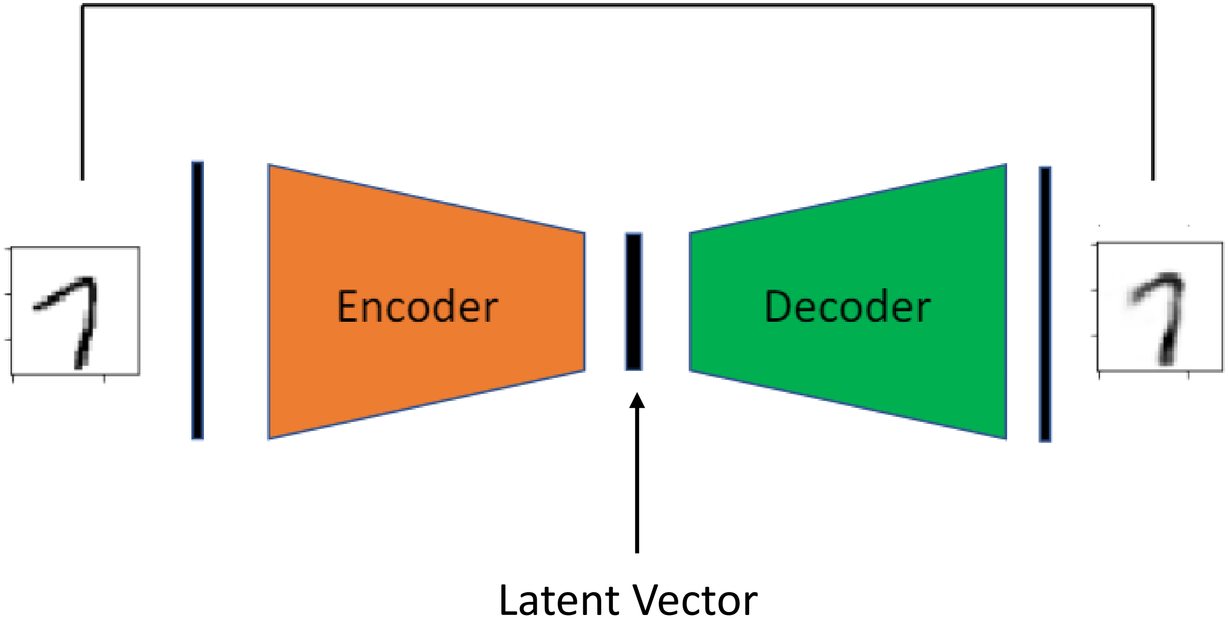
$$P(X|Y)$$

Build Better Generative Adversarial Networks

Find out the disadvantages of GANS When compared to other generative models – VAEs – Traditional Auto Encoder

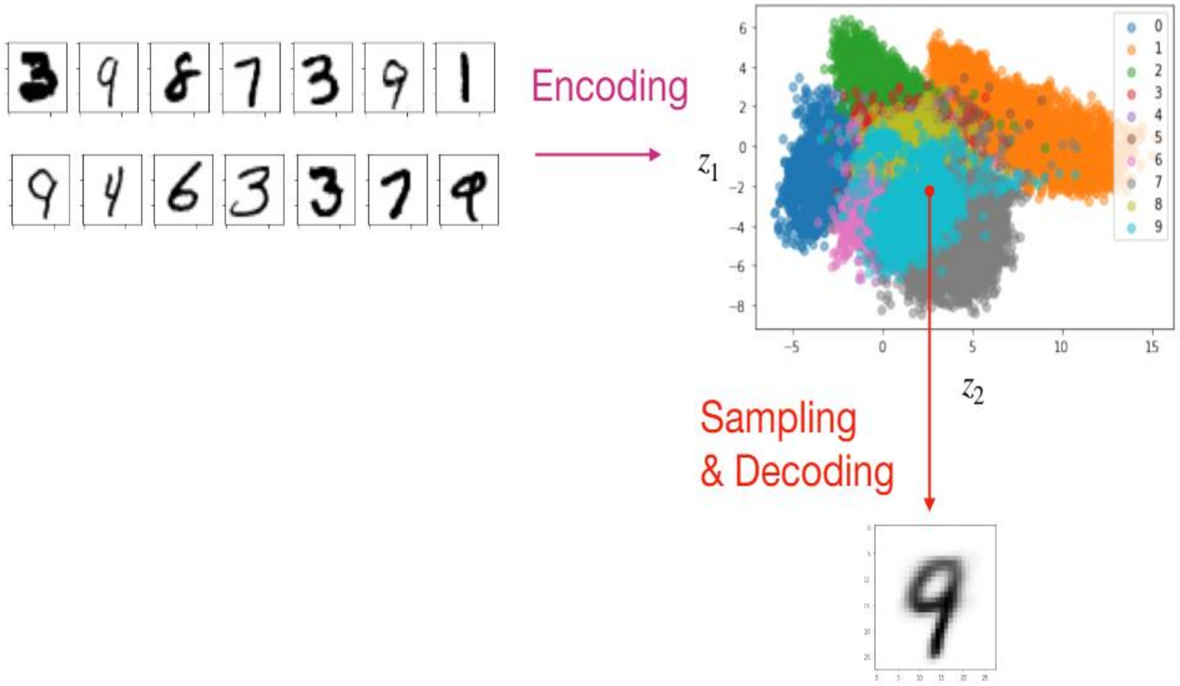
Minimize squared error loss:

$$\mathcal{L} = ||\mathbf{x} - Dec(Enc(\mathbf{x}))||_2^2$$



Build Better Generative Adversarial Networks

Find out the disadvantages of GANS When compared to other generative models – VAEs – Traditional Auto Encoder



Challenge: regular autoencoders are difficult to sample from, because

1. oddly shaped distribution, hard to sample in a balanced way
2. distribution not centered at (0, 0)
3. distribution not necessarily continuous (hard to see here in 2D, but a big problem in higher dimensional latent spaces)

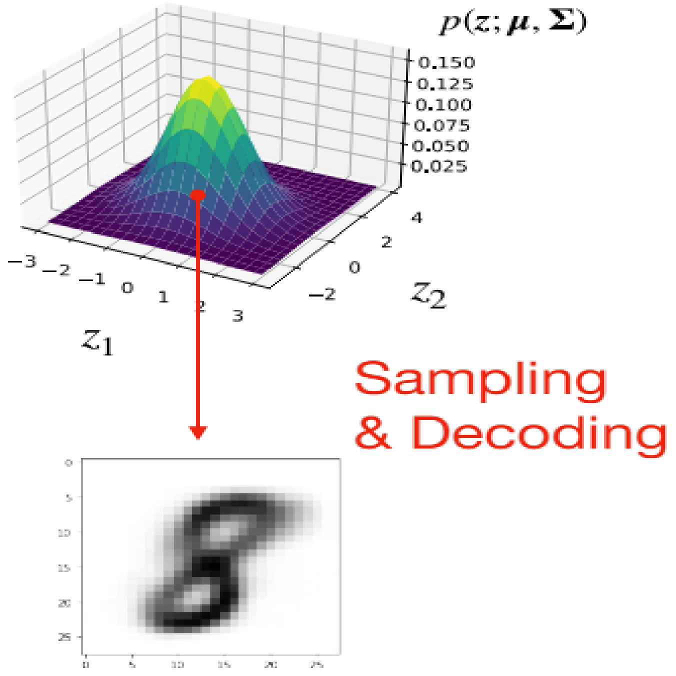
Build Better Generative Adversarial Networks

Find out the disadvantages of GANS When compared to other generative models – VAEs

Sampling → $z = \boldsymbol{\mu} + \boldsymbol{\sigma} \cdot \boldsymbol{\epsilon}$

Where $\boldsymbol{\sigma}^2 = \begin{pmatrix} \sigma_1^2 \\ \sigma_2^2 \end{pmatrix}$

$$\epsilon_1, \epsilon_2 \sim N(0,1)$$



- VAE's assume a diagonal covariance matrix (no interaction between the features).
- Thus, we only need a mean and a variance vector, no covariance matrix

https://www.youtube.com/watch?v=YgSWrafXI8U

Build Better Generative Adversarial Networks

Discover the Pros/cons of these models

Variational Autoencoders (VAEs)

The Opposite of GANs

VAEs are the exact opposite of GANs, as they focus on density estimation and provide a stable training process.

Invertible and Stable

VAEs are invertible and have a stable training process, but they may produce lower-quality results compared to GANs.

Build Better Generative Adversarial Networks

Discover the Pros/cons of these models

Variational Autoencoders (VAEs)

Encoder → Latent Space → Decoder

Build Better Generative Adversarial Networks

Discover the Pros/cons of these models

Difference Between VAE and GAN

1

VAE

VAEs encode real images, then decode them back.

2

GAN

GANs create an entirely new one without knowing the real ones.

Build Better Generative Adversarial Networks

Discover the Pros/cons of these models

Difference Between VAE and GAN

Real image → Encoder → Latent Space → Decoder

Noise vector → Generator → Discriminator

Build Better Generative Adversarial Networks

Discover the Pros/cons of these models

Autoregressive Models

Pixel-by-Pixel Generation

Autoregressive models rely on previous pixels to generate the next pixel, offering a unique approach to generative modeling.

resource: https://arxiv.org/pdf/1606.05328

Build Better Generative Adversarial Networks

Discover the Pros/cons of these models

Flow Models

Invertible Mappings

Flow models use invertible mappings to transform the data, providing a way to model the underlying probability distribution.

Build Better Generative Adversarial Networks

Discover the Pros/cons of these models

Hybrid Models

- The VAE-GAN combines the training workflows of both VAEs and GANs.
- During training, the decoder acts as both a VAE decoder and a GAN generator.
- This hybrid model aims to benefit from the strengths of both approaches, producing high-quality images while maintaining flexibility in latent space representation

Build Better Generative Adversarial Networks

Discover the Pros/cons of these models

Knowledge Check

What type of models depend on preceding pixels to generate subsequent pixels, providing a distinct method for generative modelling?

a) Autoregressive models

b) Flow models

c) GANs (Generative Adversarial Networks)

d) Variational Autoencoders (VAEs)

Build Better Generative Adversarial Networks

Discover the Pros/cons of these models

Knowledge Check

What type of models depend on preceding pixels to generate subsequent pixels, providing a distinct method for generative modelling?

a) Autoregressive models

b) Flow models

c) GANs

d) Variational Autoencoders

Learn about the many places where bias in machine learning can come from-why it's important and an approach to identify it in GANs

Build Better Generative Adversarial Networks

Bias in Machine Learning

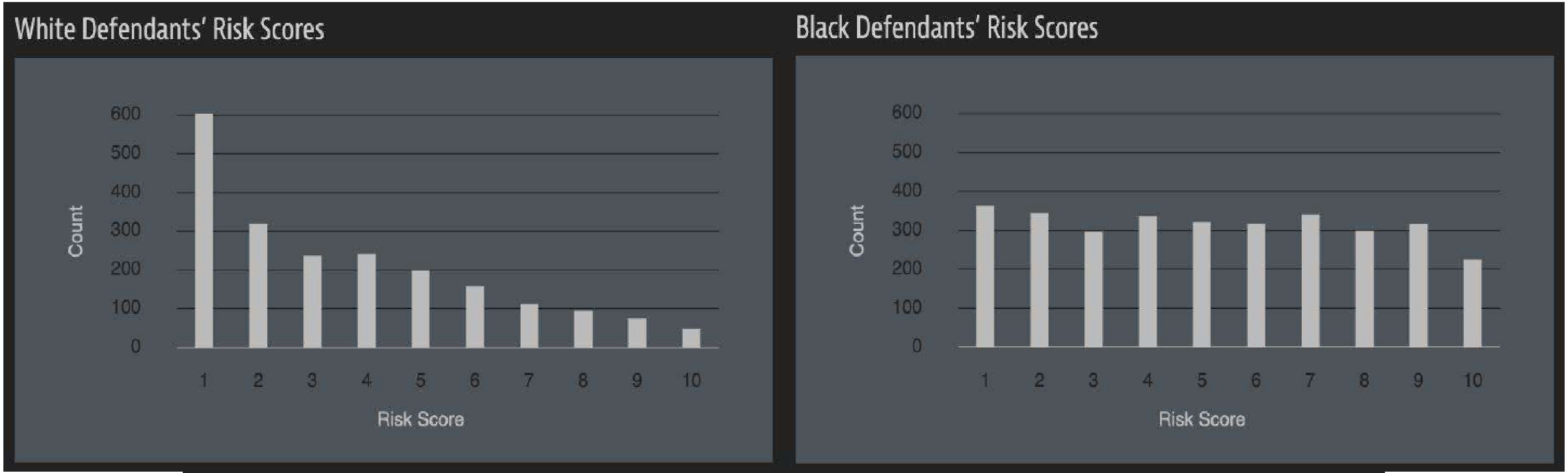
- Machine bias is a significant issue in risk assessment.
- Computerized algorithms are used to calculate the likelihood of an individual committing a crime in the future.
- This process of risk assessment is often flawed, with biases inherent in the data and algorithms used.

Build Better Generative Adversarial Networks

Bias in Machine Learning

Build Better Generative Adversarial Networks

Bias in Machine Learning



These charts show that scores for white defendants were skewed toward lower-risk categories. Scores for black defendants were not. (Source: ProPublica analysis of data from Broward County, Fla.)

Build Better Generative Adversarial Networks

COMPAS Algorithm

Purpose

The COMPAS algorithm was developed to reduce bias in pretrial sentencing, but it has been found to be significantly biased itself.

Efficiency

The efficiency of the COMPAS algorithm is just just 20%, meaning it is not not very effective in achieving its goal of reducing bias.

Transparency

The COMPAS algorithm uses a proprietary calculation that is not available to the public and has not been properly validated.

Build Better Generative Adversarial Networks

Risks of Risk Assessment Software

1

Disproportionate Impact

Machine learning bias in risk assessment software has a disproportionately negative effect on historically underserved populations.

2

Validation Challenges

It is difficult to validate the accuracy and fairness of risk assessment software.

3

Overlooked Considerations

Risk assessment software often misses important considerations about the people it is evaluating.

Build Better Generative Adversarial Networks

Defining Fairness

No Single Definition

There is no single definition for fairness when it comes to machine learning models.

Data Fairness

Fairness can mean that the data used is unbiased across different groups.

Equality of Odds

Fairness can also mean that there is a similar chance of correct or incorrect generation for all groups.

Demographic Parity

Fairness can focus on the overall distribution of generated data matching across groups, without considering the quality of generated data within each group.

Build Better Generative Adversarial Networks

Bias in Training phase

1

Biased Training Data

Bias can arise from a lack of variation or inherent biases in the data used to train machine learning models.

2

Examples of Bias

Examples include fewer images of people of color or biases from scraping celebrity images.

3

Mitigating Bias

Considerations for mitigating bias include ensuring source and demographic diversity in data collection and the impact of labeler diversity on labeled data.

Build Better Generative Adversarial Networks

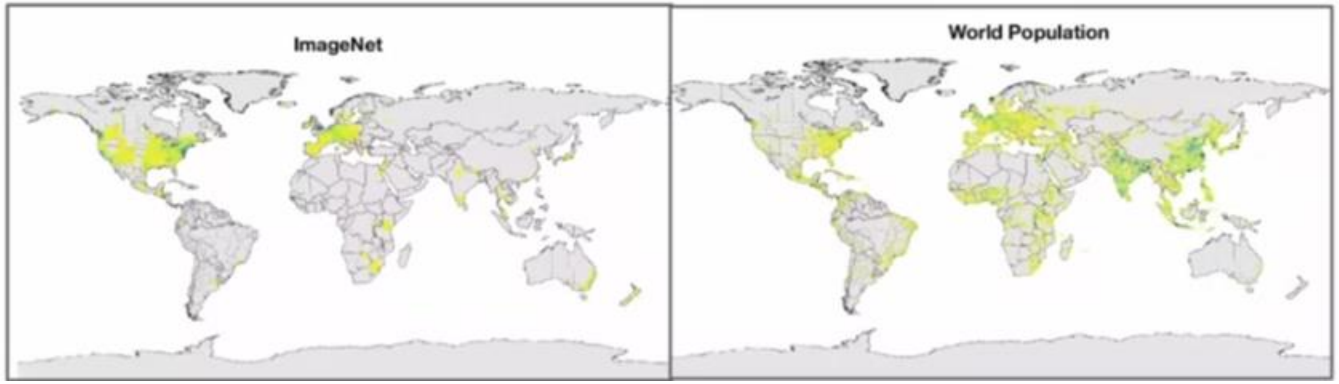
Societal Bias in Evaluation

Cultural Differences

Societal biases can influence the evaluation methods used to assess machine learning models.

Geographic Imbalance

The ImageNet dataset, for example, has a geographic imbalance that can impact the performance of classification models.



Resource: https://arxiv.org/pdf/1906.02659

Build Better Generative Adversarial Networks

Bias in Model Architecture

Programmer Diversity

The diversity of programmers and their views can influence the design of machine learning models.

Impact on Outputs

The impact of programmers' views can be seen in the generated images, such as the effect of loss functions in GANs on skin color.

Defining 'Right' and 'Wrong'

Challenges arise in defining what is 'right' and 'wrong' in the context of generative models.

Build Better Generative Adversarial Networks

Pulse

`

- Pulse uses StlyeGan for upsampling pixelated images.

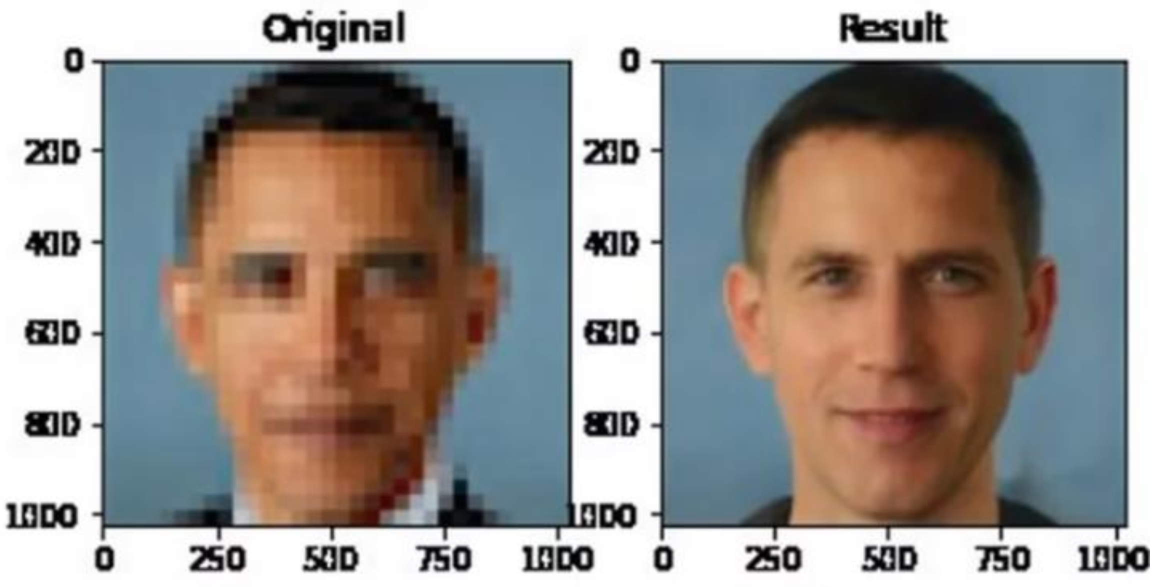
- Issues with other public being transformed into whiter version

- Efforts to mitigate bias in GANs through adversarial loss.

Build Better Generative Adversarial Networks

Pulse

Example of pixelated Obama photo is upsampled to a white man



Courtesy: @hardmaru on Twitter

Build Better Generative Adversarial Networks

Addressing Bias in Machine learning

1 **Pervasive Issue**

Bias is a pervasive issue in machine learning models that must be addressed.

2 **Awareness of Bias**

It is important to be aware of bias in different stages of model development.

3 **Combating Bias**

Researchers and practitioners have a responsibility to combat bias in their day-to-day work with machine learning models.

StyleGAN and Advancement

Build Better Generative Adversarial Networks

GAN Improvements

1. **Stability**

   Expanded training on high-resolution images

2. **Capacity**

   Increased model complexity for superior resolution

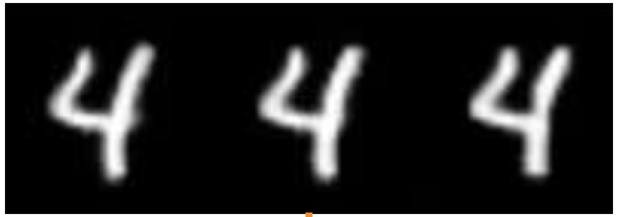
3. **Diversity**

   Increasing variety

Build Better Generative Adversarial Networks

GAN Improvements - Stability

High standard deviation



Low standard deviation

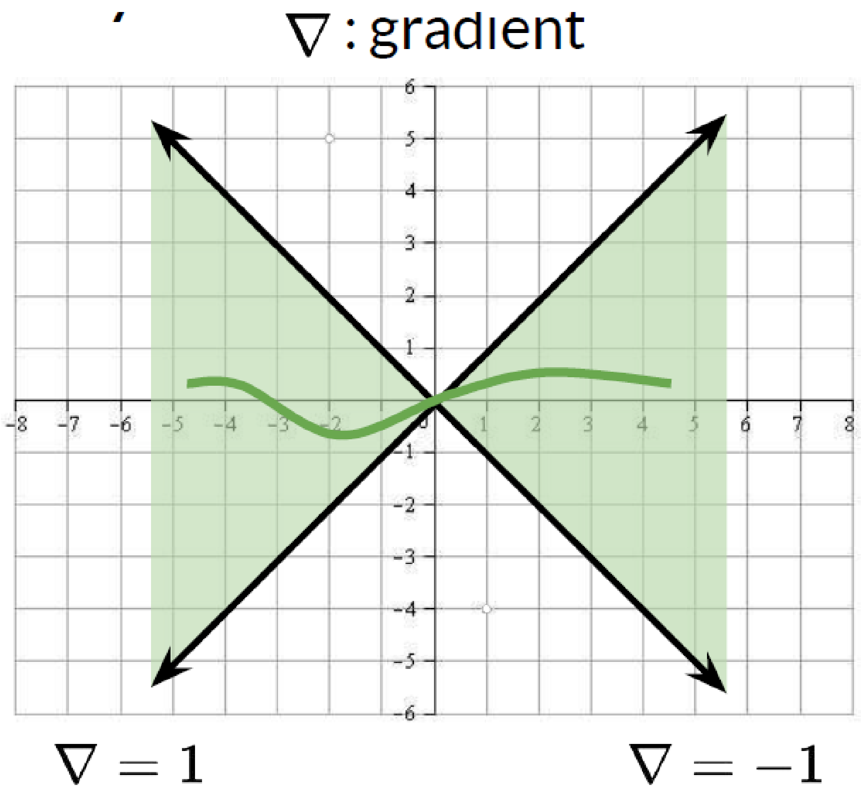
Use batch standard deviation to encourage diversity

Build Better Generative Adversarial Networks

GAN Improvements - Stability

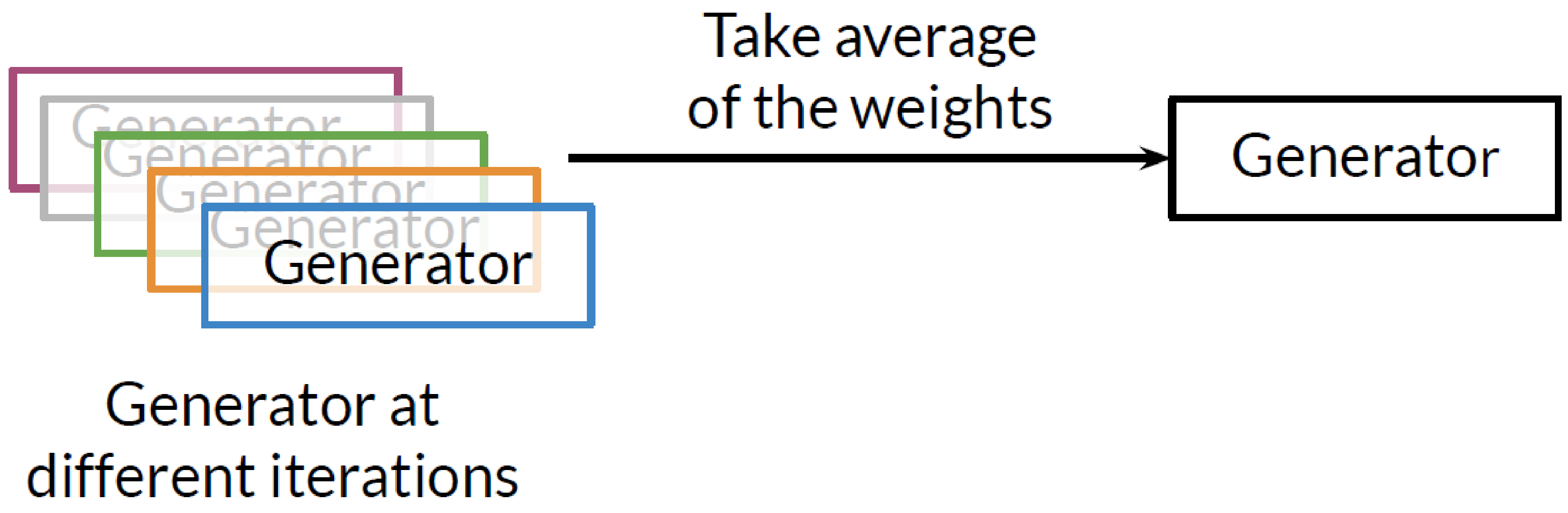
Improve stability by enforcing 1-Lipschitz continuity

E.g. WGAN-GP and Spectral Normalization



Build Better Generative Adversarial Networks

Stability



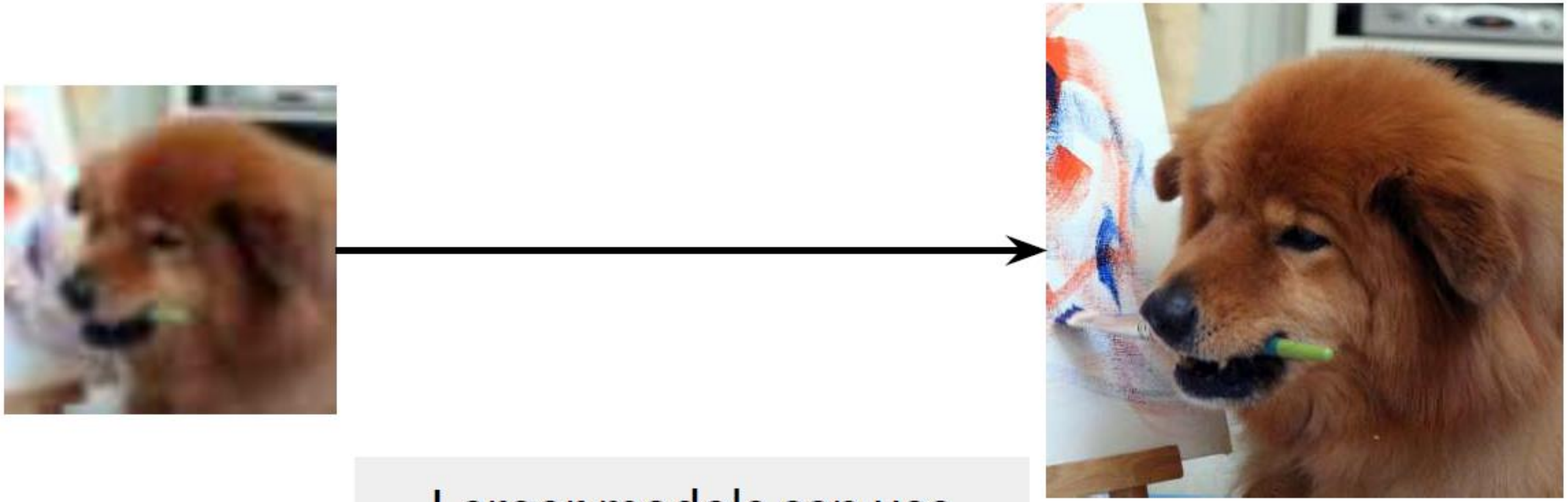
Build Better Generative Adversarial Networks

Stability

Progressive growing gradually trains GAN at increasing resolutions

Build Better Generative Adversarial Networks

Capacity



Larger models can use higher resolution images

Build Better Generative Adversarial Networks

Diversity

Build Better Generative Adversarial Networks

GAN Training Stability

- The first major improvement was to GAN training stability. Mode collapse is a key issue where the generator is stuck in a local minimum, producing little variation.
- Standard deviation across samples can indicate mode collapse. Low variation suggests mode collapse.
- Passing this information to the discriminator helps in providing feedback to the generator to avoid mode collapse.

Build Better Generative Adversarial Networks

Capacity And Diversity

Larger Capacity

GANs have larger capacity due to improved hardware and high-resolution datasets like FFHQ.

Improved Architectures

DC-GANs and other architectures cater to these datasets, leading to better quality outputs.

Increased Diversity

Increased diversity in generated images is achieved through larger datasets with more variety, preventing mode collapse and modeling various human faces.

Style GANs

Build Better Generative Adversarial Networks

StyleGAN Goals

Greater fidelity

Style GAN aims to generate images with higher fidelity, capturing more realistic and detailed visual elements.

Increased diversity of output

Style GAN seeks to increase the diversity of generated images, allowing for a wider range of unique and varied outputs.

More control over image feature

Style GAN provides users with more control over the specific features and attributes of the generated images.

Build Better Generative Adversarial Networks

StyleGAN Goals

Greater Fidelity

Not fooling anyone

I'm shook

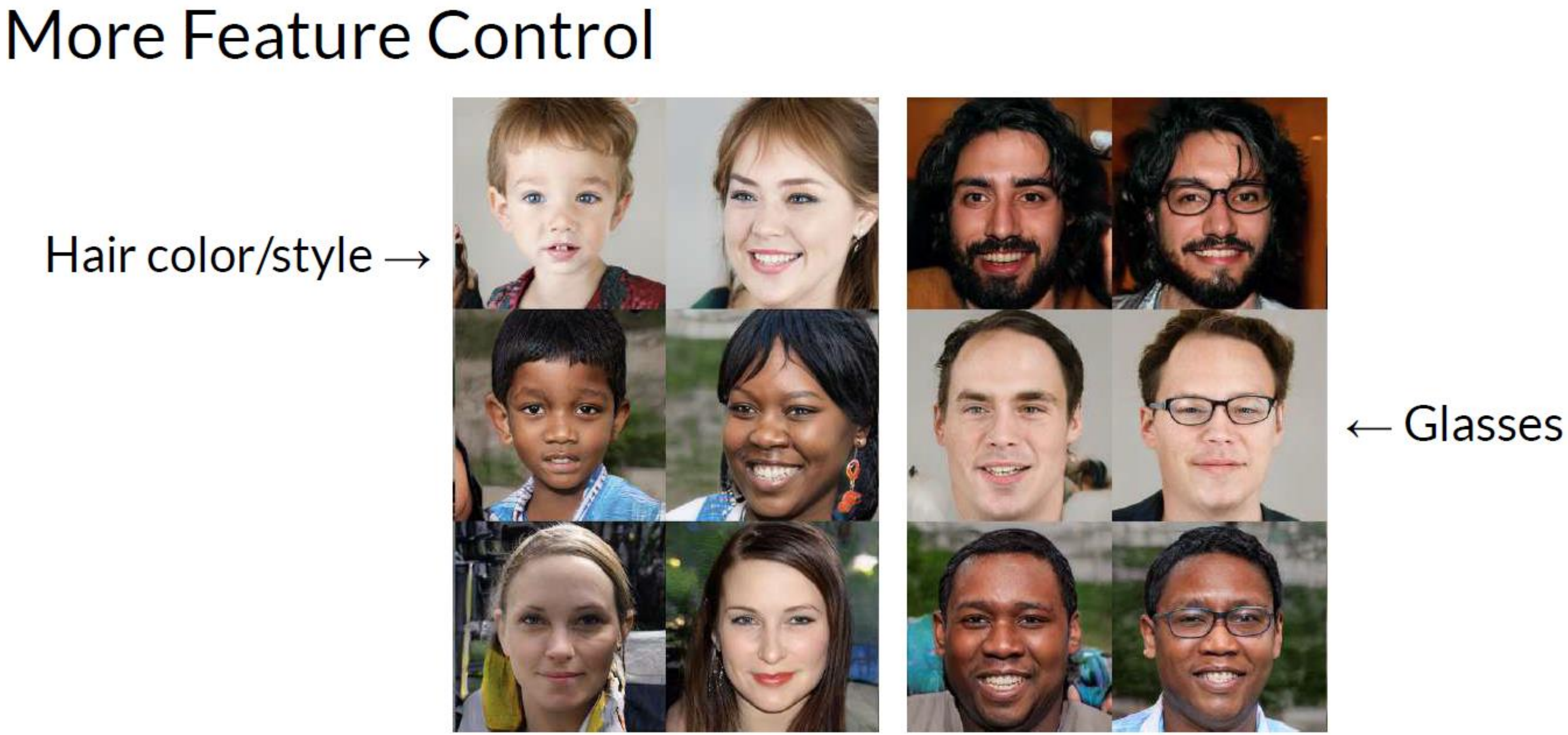
Build Better Generative Adversarial Networks

StyleGAN Goals

Increased Diversity

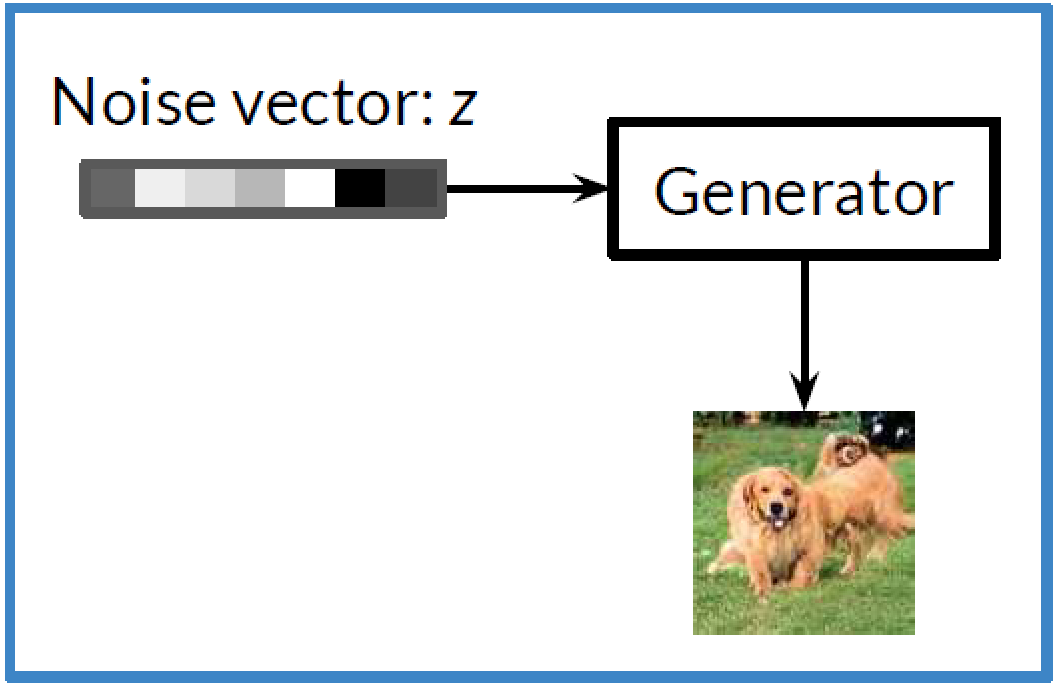
Build Better Generative Adversarial Networks

StyleGAN Goals

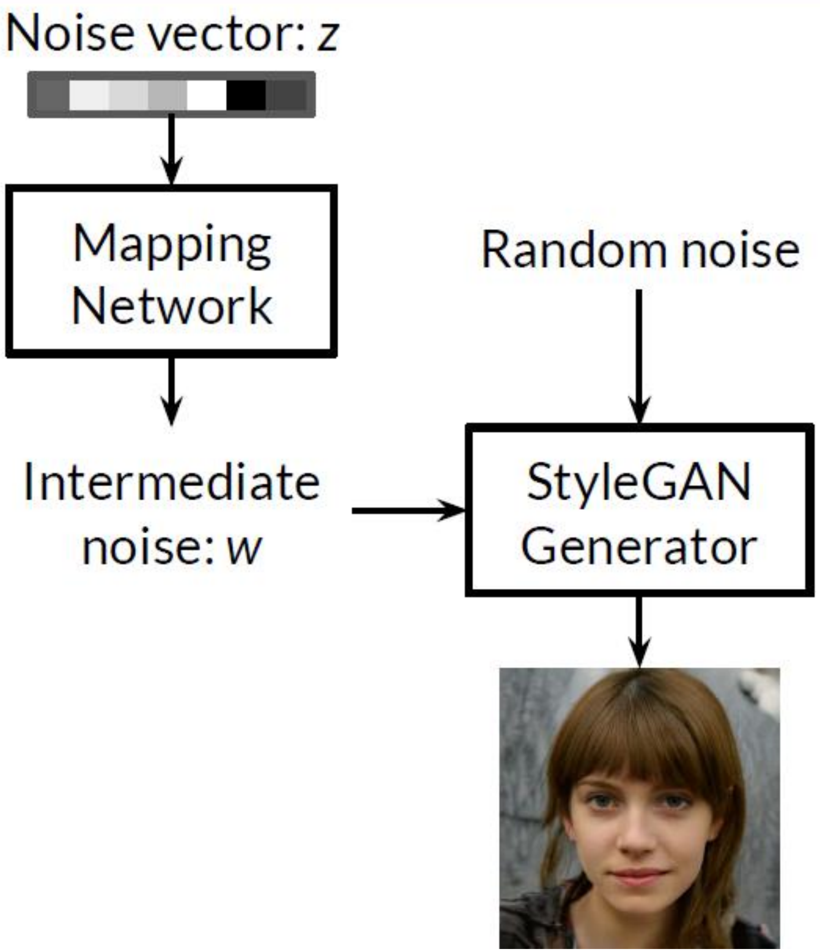


Build Better Generative Adversarial Networks

StyleGAN Architecture



Traditional architecture



Build Better Generative Adversarial Networks

The Style based Generator

Mapping network

The mapping network is a key component of the style-based generator. It takes a random input and maps it to an intermediate latent space.

Intermediate noise

The intermediate noise is a crucial input to the generator. It allows for more fine-grained control over the generated output.

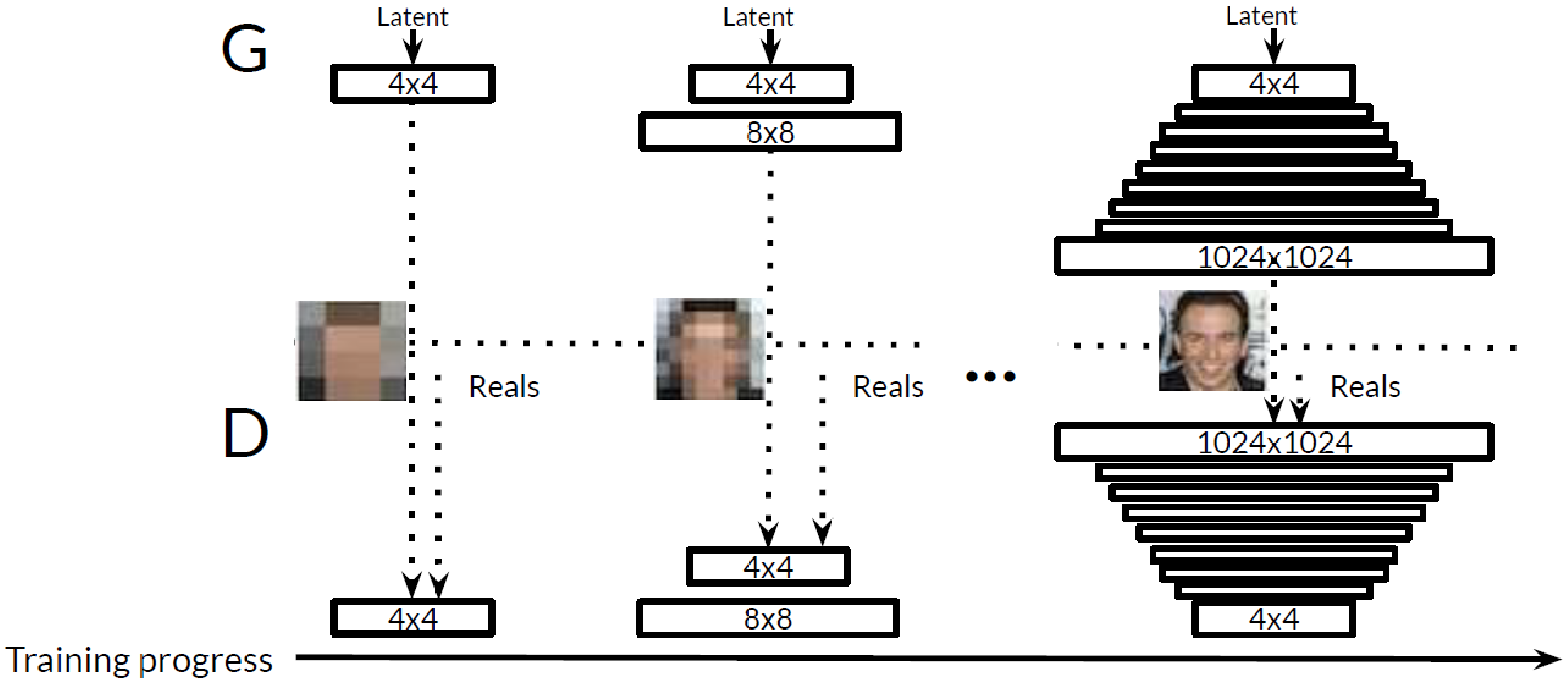
Random noise

The random noise is the initial input to the mapping network. It provides the starting point for the generator to create new, unique content.

Understand how StyleGAN improves upon previous models and implements the components and the techniques associated with StyleGAN

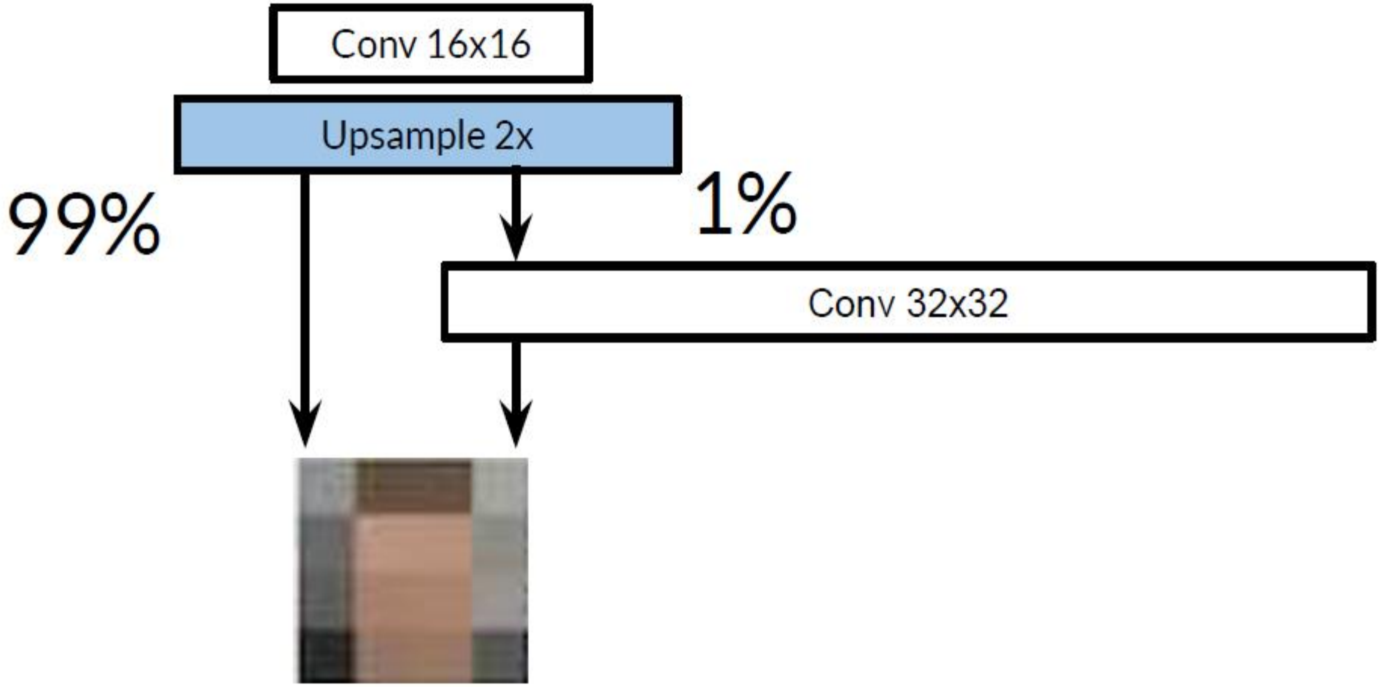
Build Better Generative Adversarial Networks

Progressive Growing



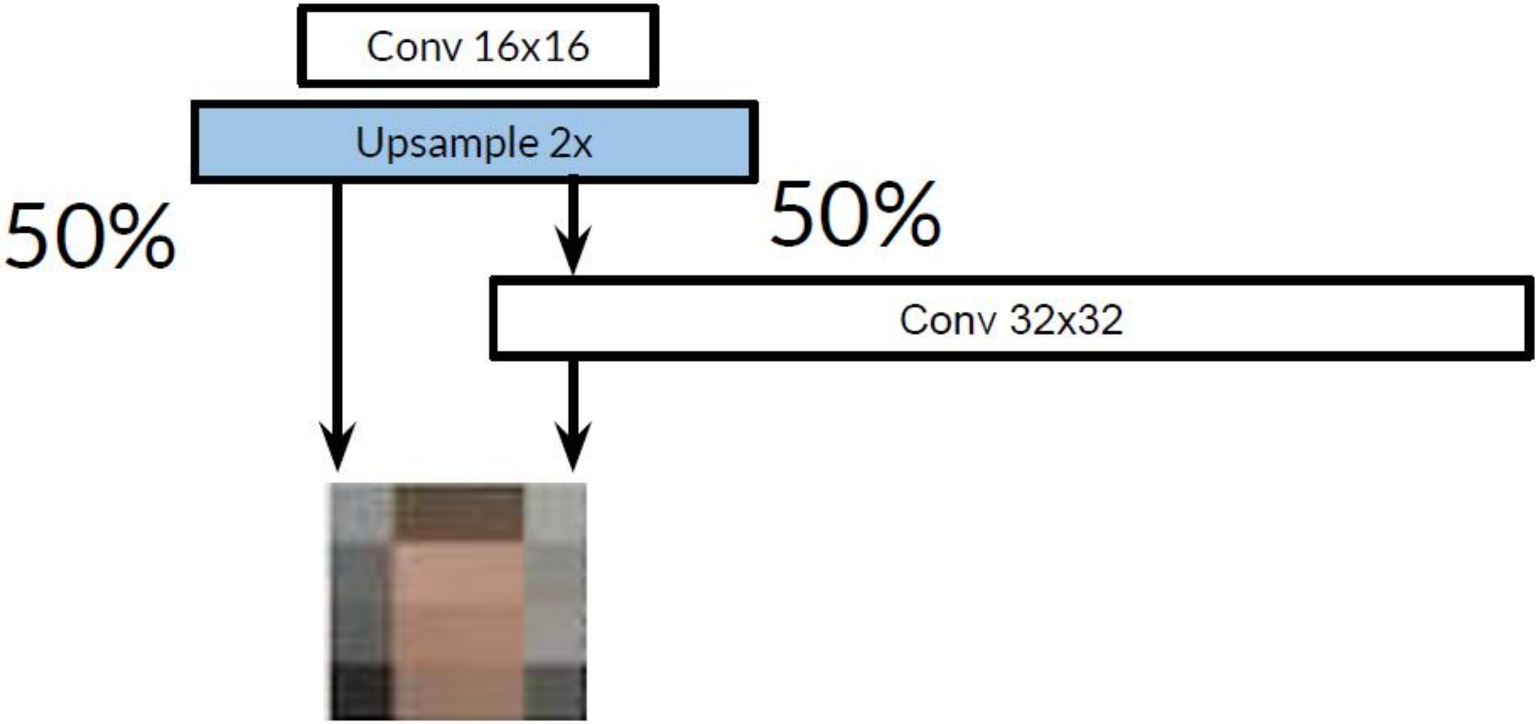
Build Better Generative Adversarial Networks

Progressive Growing - Generator



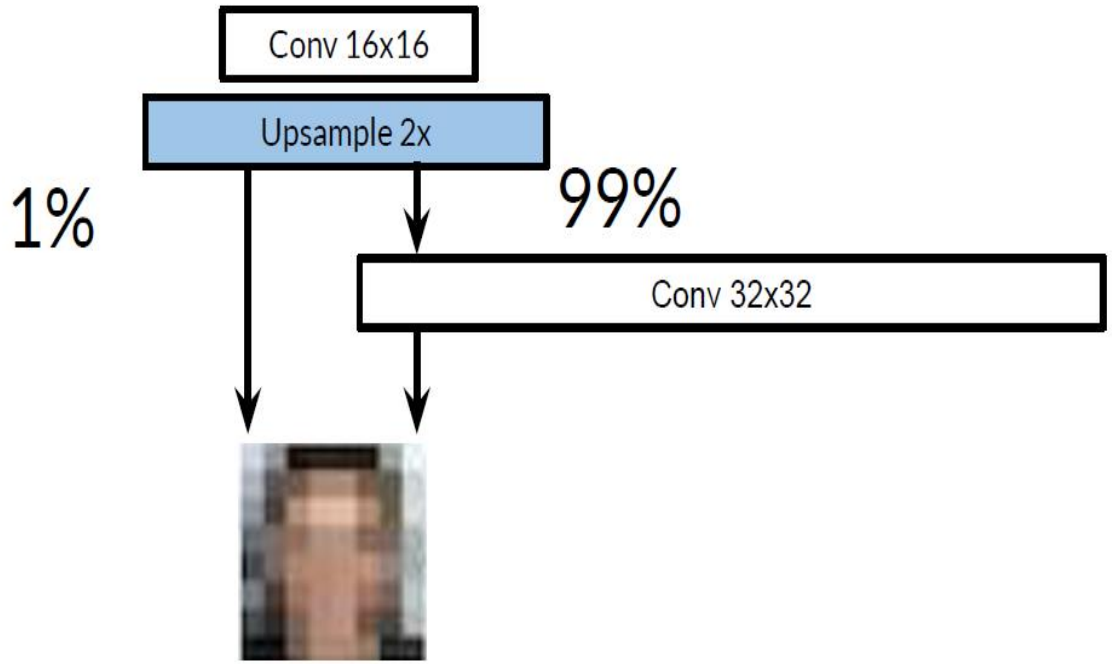
Build Better Generative Adversarial Networks

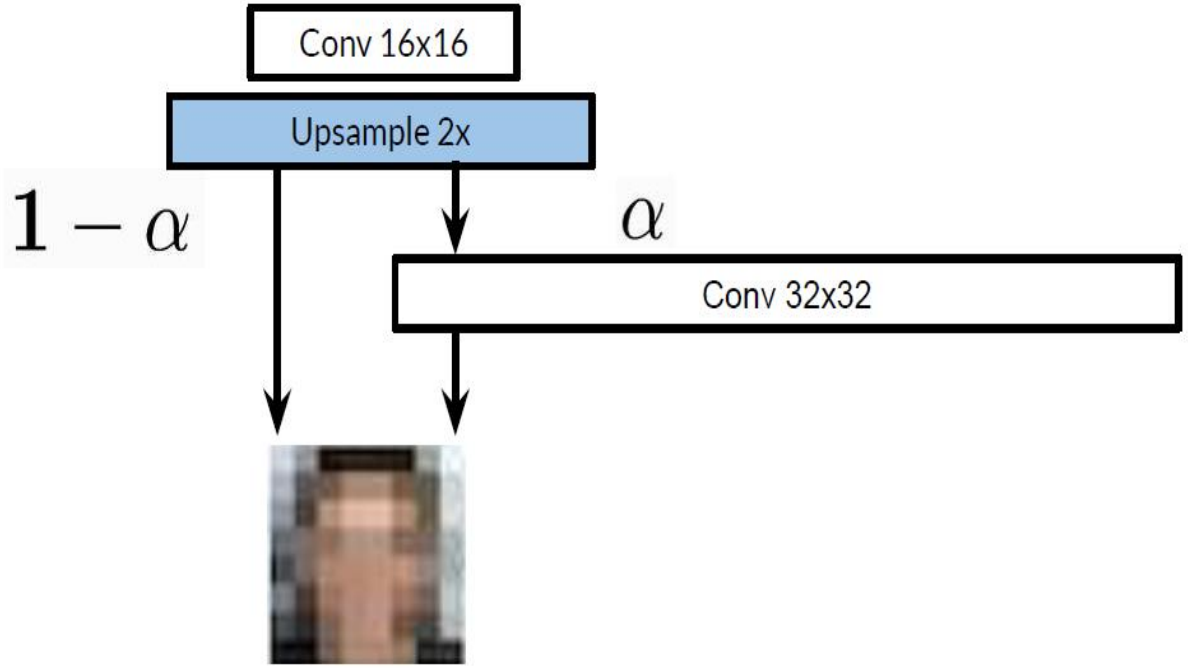
Progressive Growing - Generator



Build Better Generative Adversarial Networks

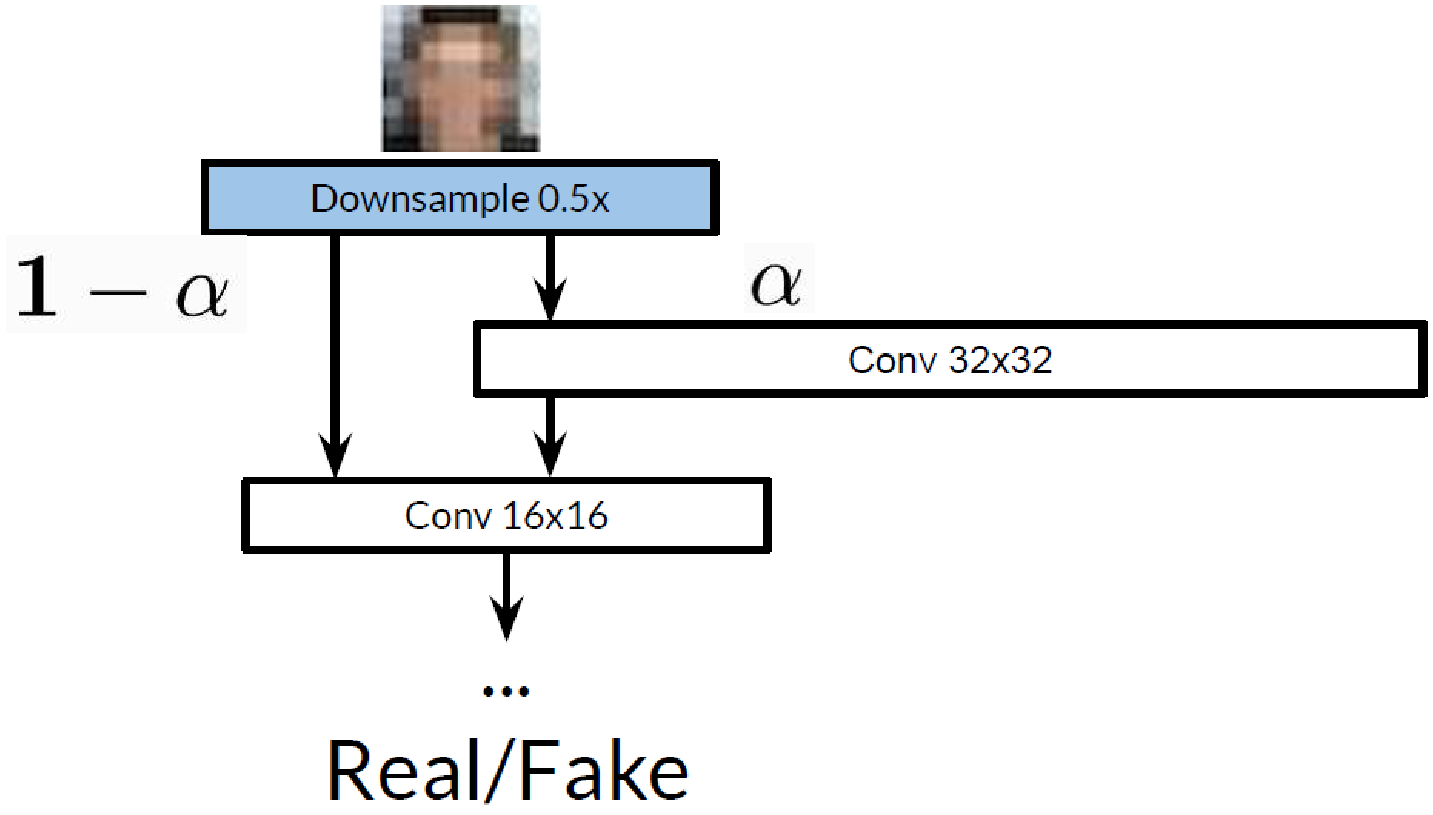
Progressive Growing - Generator





Build Better Generative Adversarial Networks

Progressive Growing - Discriminator



Build Better Generative Adversarial Networks

Progressive Growing

1

Slow and steady

Got to take it slow for better results

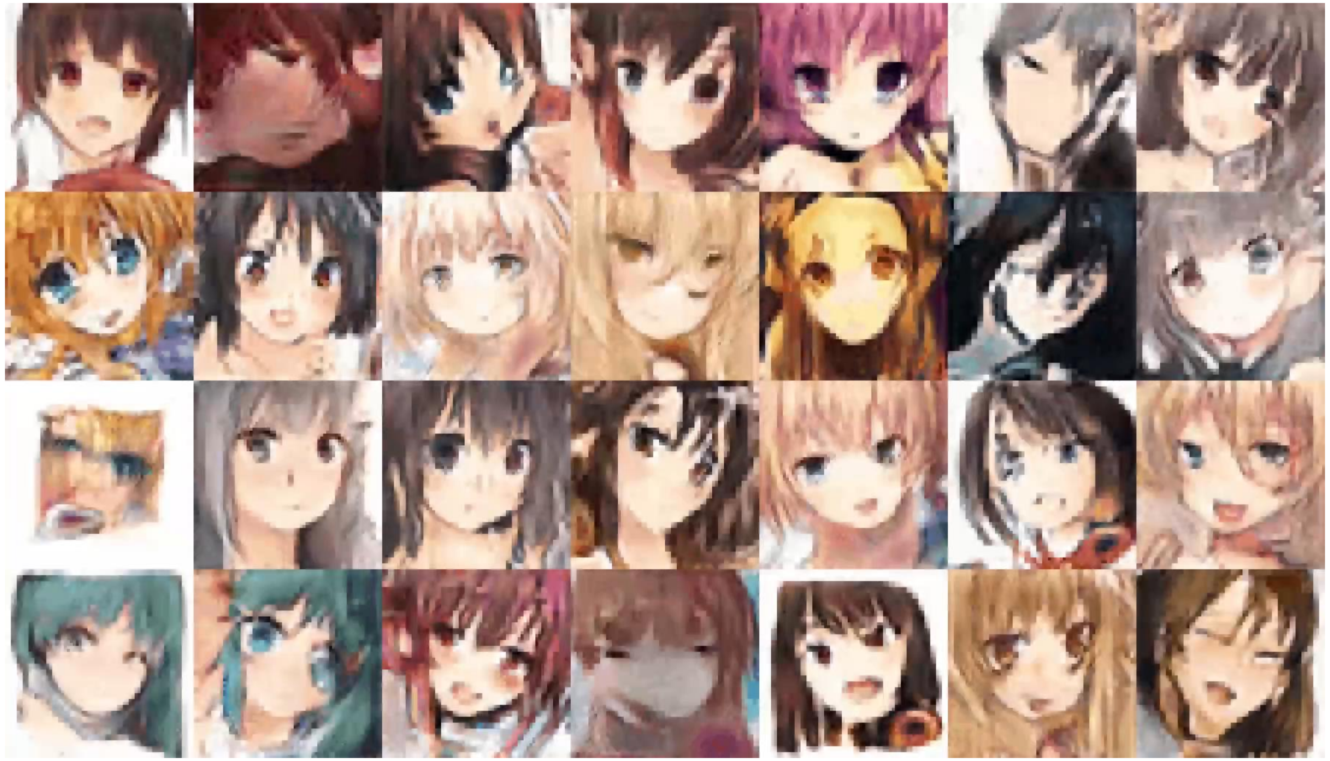
2

Dependence on parameters

To increase the dependence of learn parameters

Build Better Generative Adversarial Networks

Progressive Growing



Build Better Generative Adversarial Networks

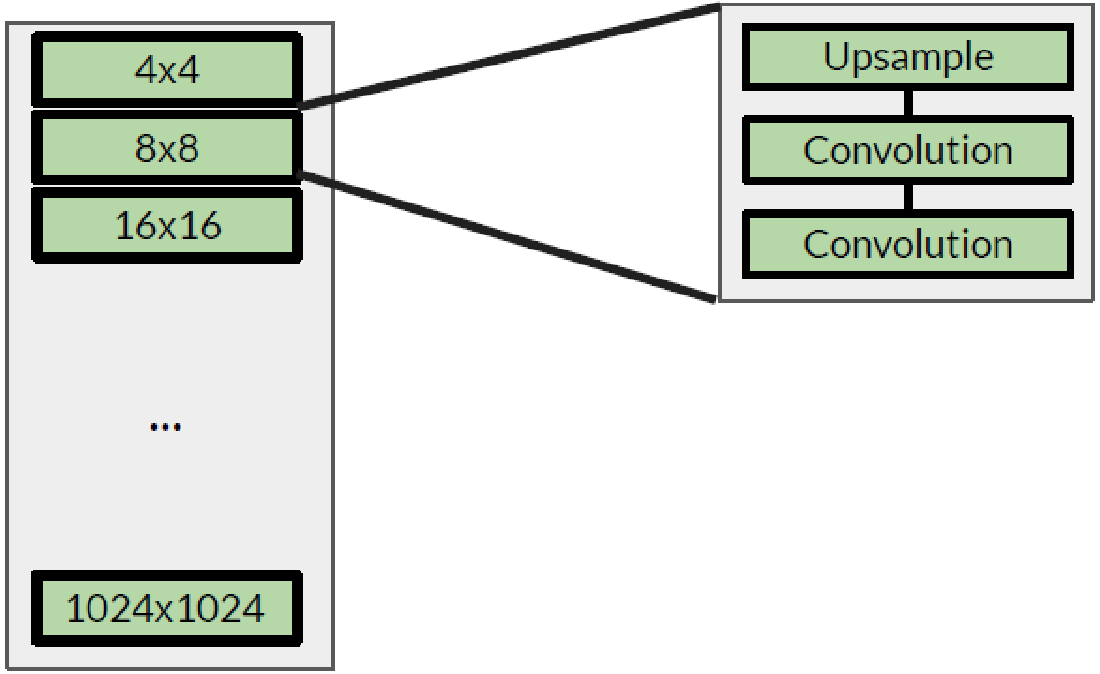
Progressive in Context

Double Image Resolution

Progressively growing the GAN model allows for doubling the image resolution, which helps with faster and more stable training for higher resolution outputs.

Faster, More Stable Training

The progressive approach to growing the GAN model facilitates faster and more stable training, especially when generating high resolution images.



Build Better Generative Adversarial Networks

Understanding StyleGAN

Knowledge Check

How does the generator's stability relate to its input weights during different iterations?

a) The generator maintains a consistent input weight average across iterations

b) The input weights of the generator fluctuate drastically over time

c) The generator's stability is unaffected by changes in input weights

d) The generator's input weights are updated randomly at each iteration

Build Better Generative Adversarial Networks

Understanding StyleGAN

Knowledge Check

How does the generator's stability relate to its input weights during different iterations?

a) The generator maintains a consistent input weight average across iterations

b) The input weights of the generator fluctuate drastically over time

c) The generator's stability is unaffected by changes in input weights

d) The generator's input weights are updated randomly at each iteration

L&T EduTech

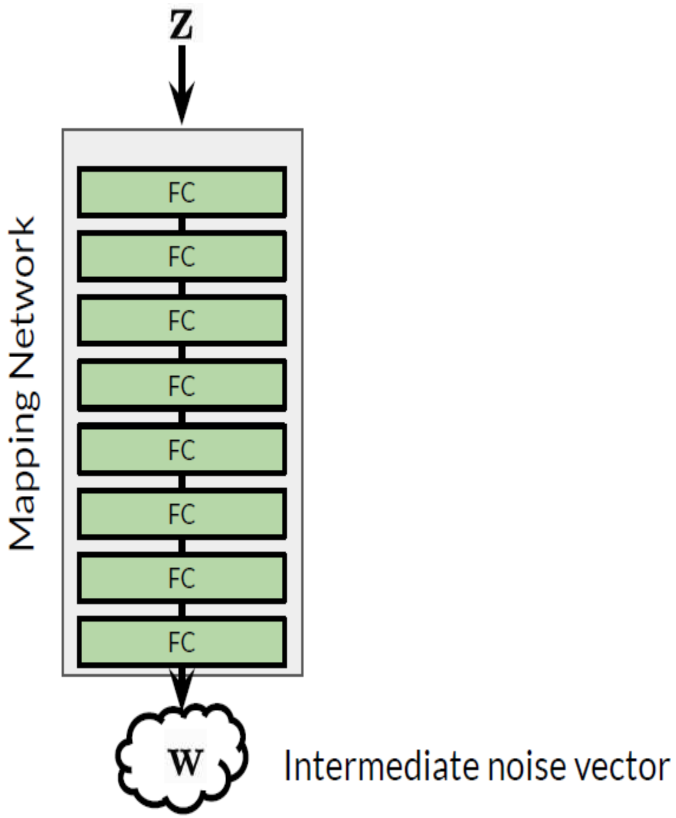
LTIMindtree

Build Better Generative Adversarial Networks

Noise Mapping Network

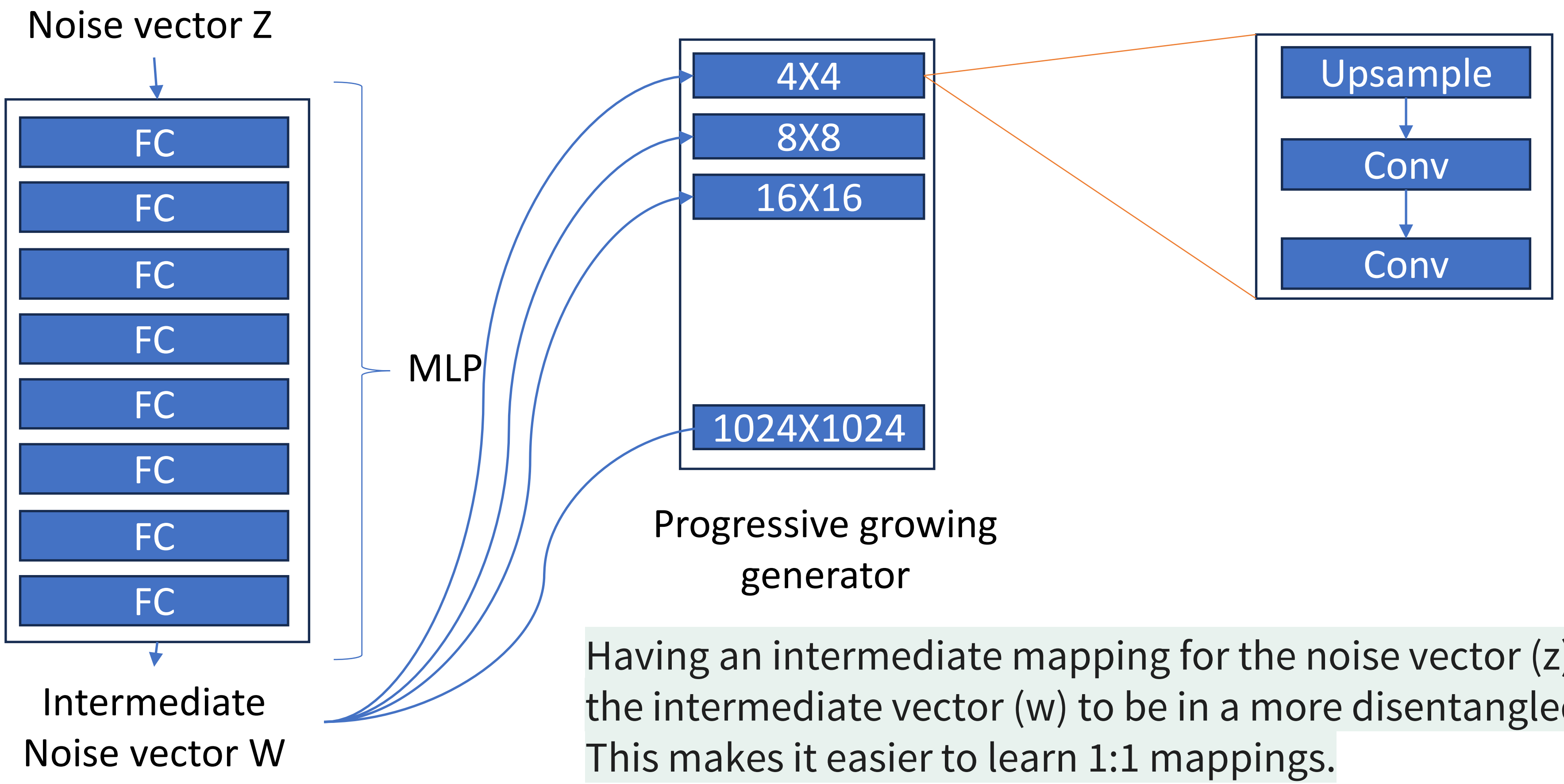
The noise mapping network doesn't change the value of the vector, but rather the mapping would get you a more disentangled representation

This allows for more fine-grained control and manipulation of the generated output, as the noise vector can be mapped to a more meaningful and interpretable latent space.



Build Better Generative Adversarial Networks

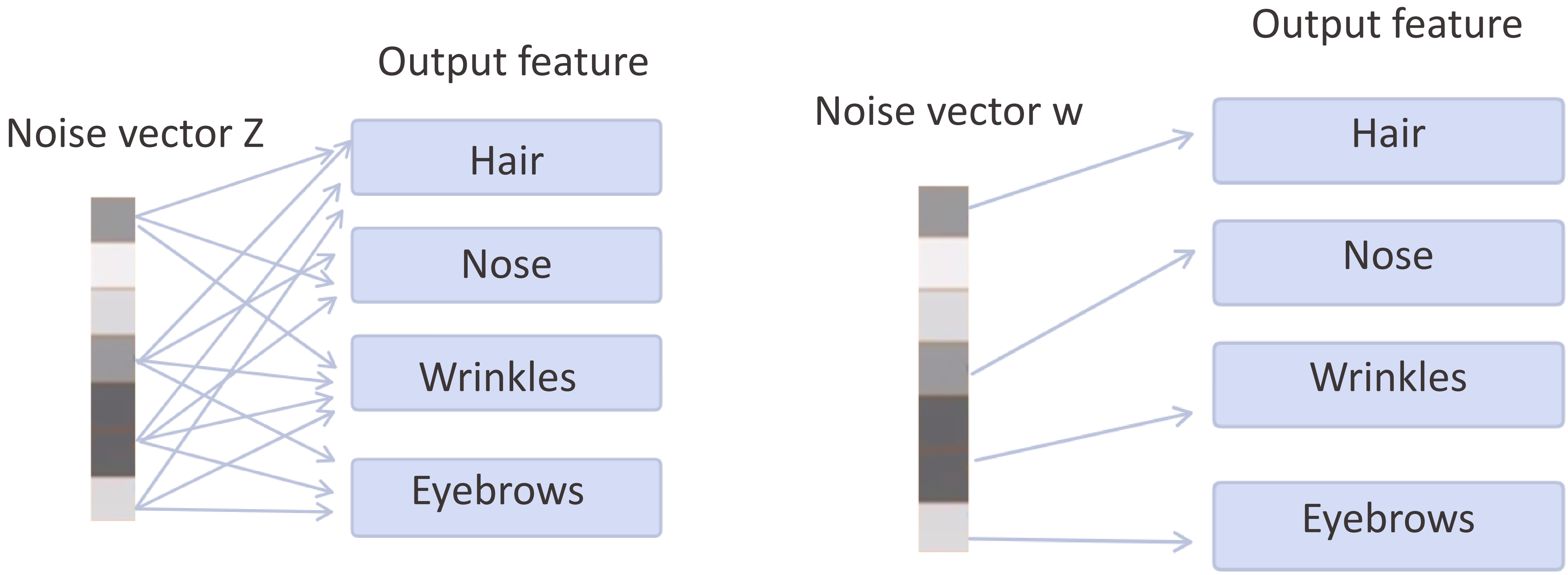
Noise Mapping Vector Architecture



Having an intermediate mapping for the noise vector (z) allows the intermediate vector (w) to be in a more disentangled space. This makes it easier to learn 1:1 mappings.

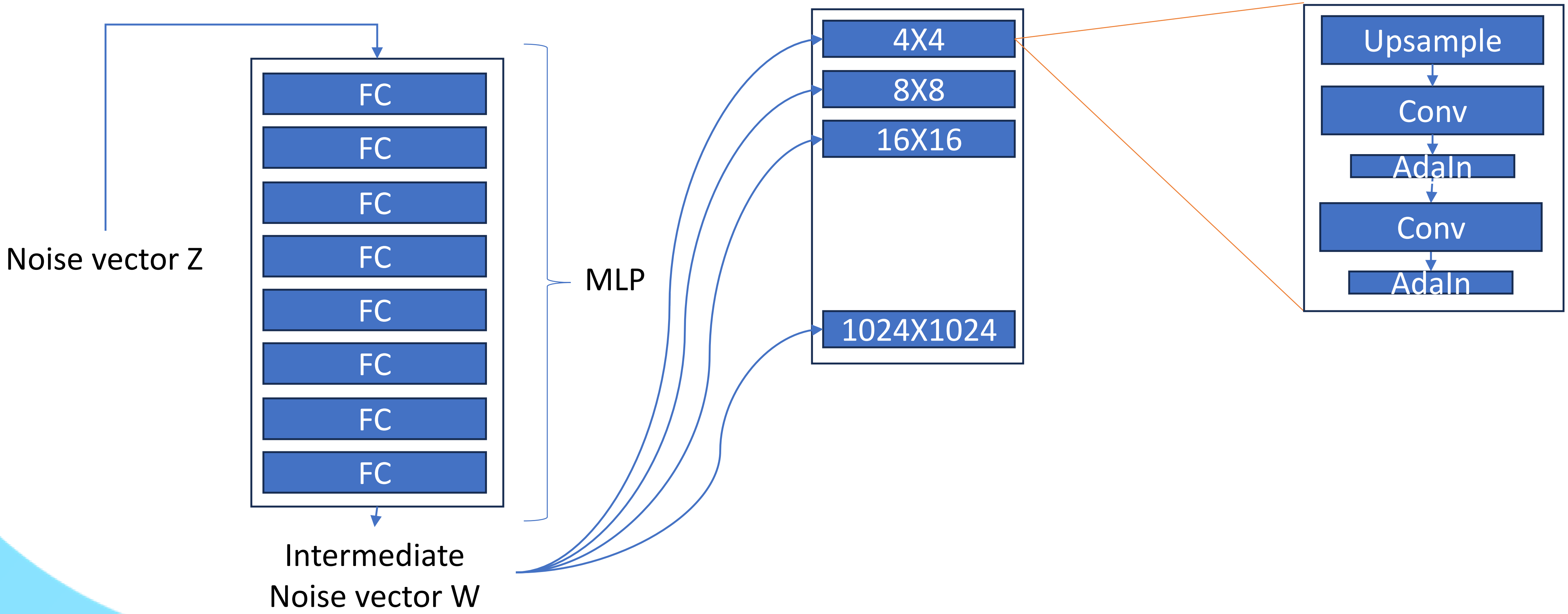
Build Better Generative Adversarial Networks

Understanding Z-Space and W-Space



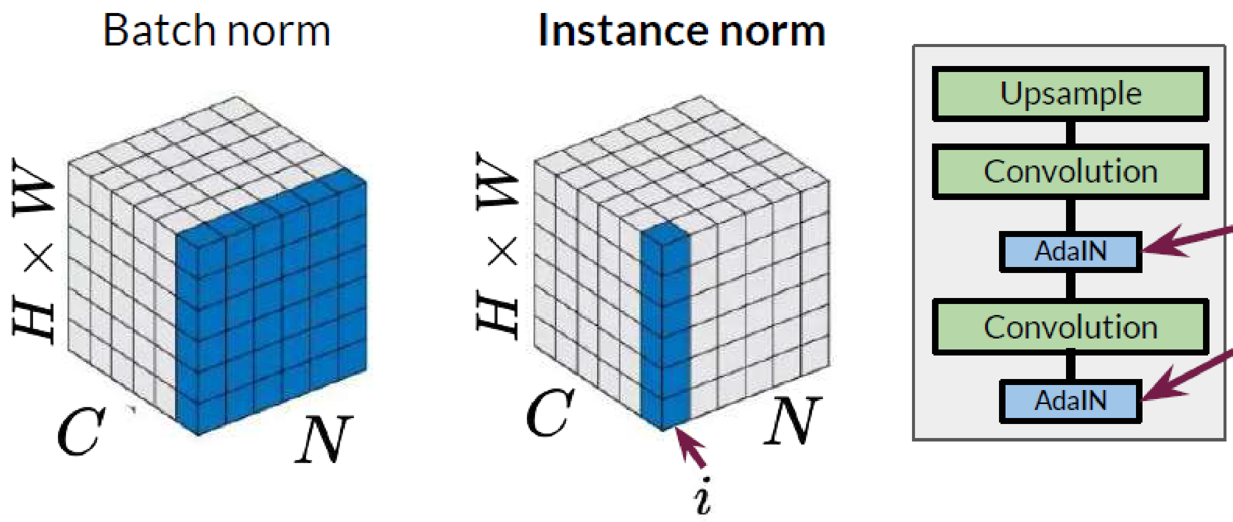
Build Better Generative Adversarial Networks

Adaptive Instance (AdaIN)



Build Better Generative Adversarial Networks

Adaptive Instance (AdaIN)

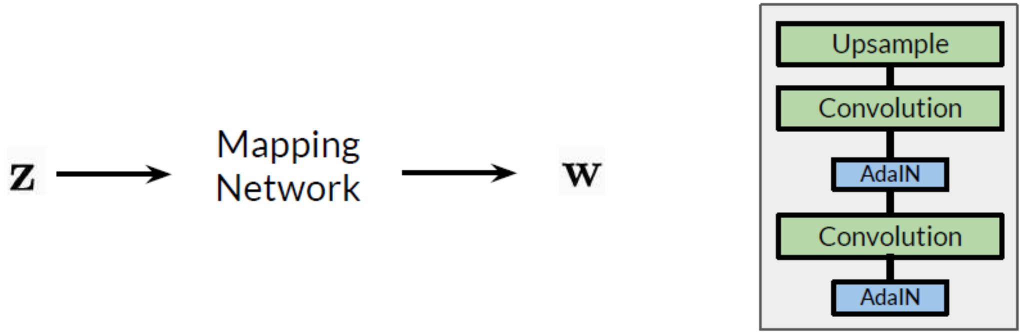


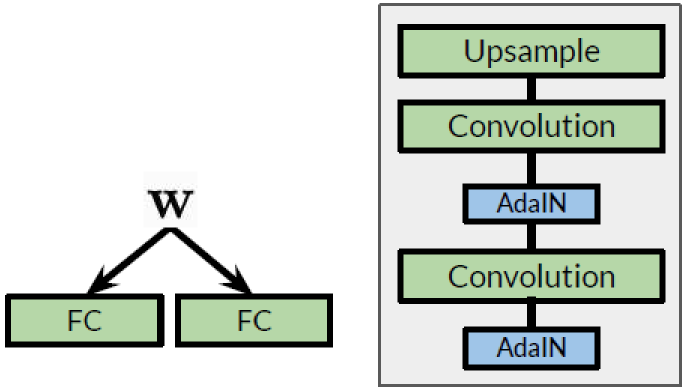
$$\frac{\mathbf{x}_i - \mu(\mathbf{x}_i)}{\sigma(\mathbf{x}_i)}$$

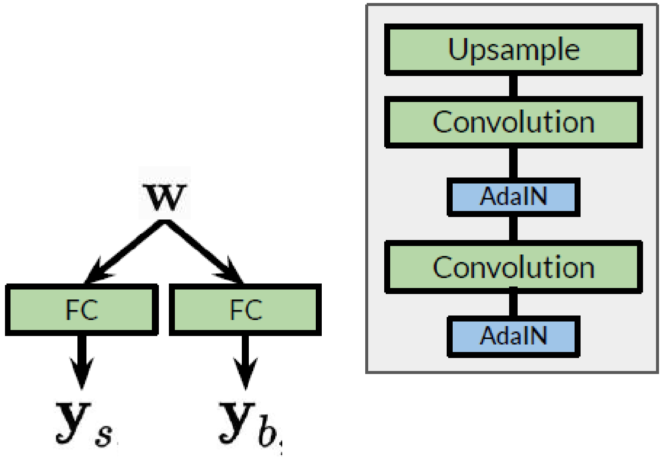
Step 1: Normalize convolution outputs using **Instance Normalization**

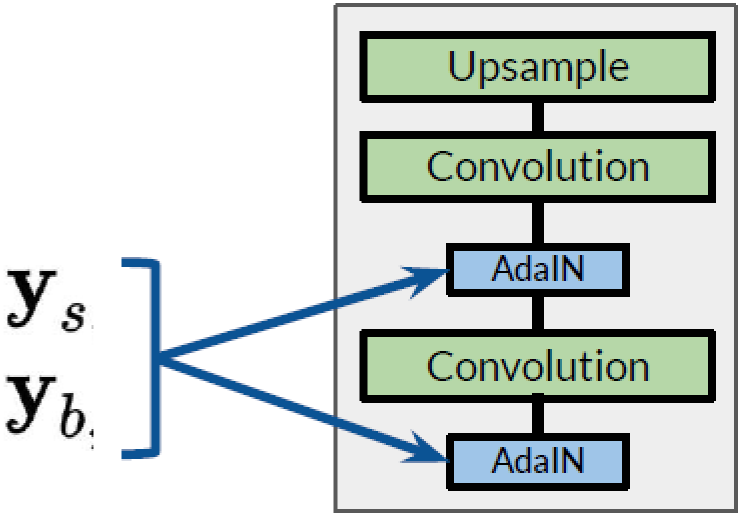
Build Better Generative Adversarial Networks

Adaptive Instance (AdaIN)









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

Build Better Generative Adversarial Networks

Adaptive Instance (AdaIN)

Normalization

AdaIN normalizes the convolution outputs using instance normalization, ensuring that each example is normalized independently.

Adaptive Styles

AdaIN then applies adaptive styles to the normalized features using the intermediate noise vector, allowing for the transfer of style information onto the generated image.

$$AdaIN(x_i, y) = y_{s,i} \frac{x_i - \mu(x_i)}{\sigma(x_i)} + y_{b,i}$$

Build Better Generative Adversarial Networks

The Power of AdaIN

1. **Style Transfer**

   AdaIN transfers style information from the intermediate noise vector onto the generated image, allowing for the seamless integration of learned style characteristics.

2. **Instance Normalization**

   Instance normalization is used to normalize individual examples before applying the style statistics from the W space.

3. **Targeted Style Application**

   AdaIN blocks facilitate the targeted transfer of learned style information from the intermediate noise vector onto the generated image.

Currently the most state-of-the-art GAN with powerful capabilities

Build Better Generative Adversarial Networks

Style Mixing

Flexible Manipulation

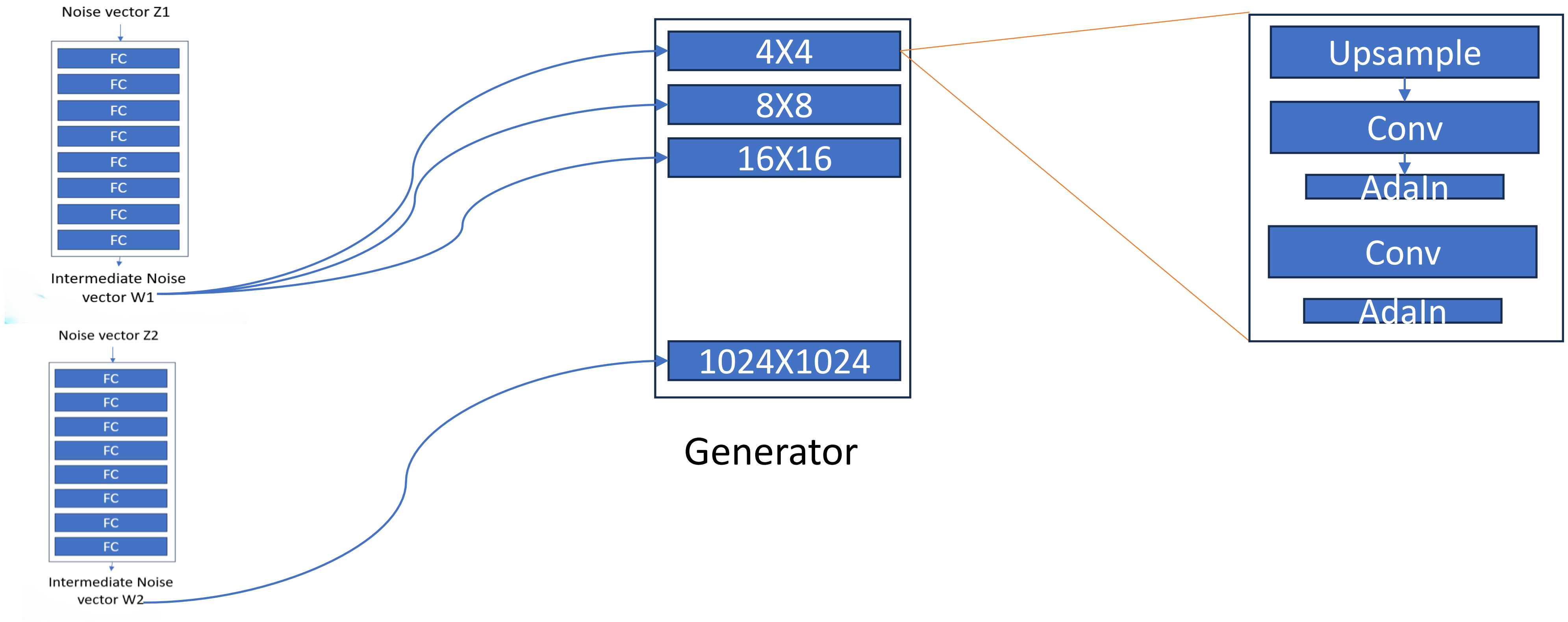
Style mixing allows for the manipulation of any point in the generated image, increasing the diversity of the output.

Coarse to Fine Control

The level of control, whether coarse or fine, depends on where the network style or noise is added, with earlier layers affecting coarse features and later layers affecting finer details.

Build Better Generative Adversarial Networks

Style Mixing



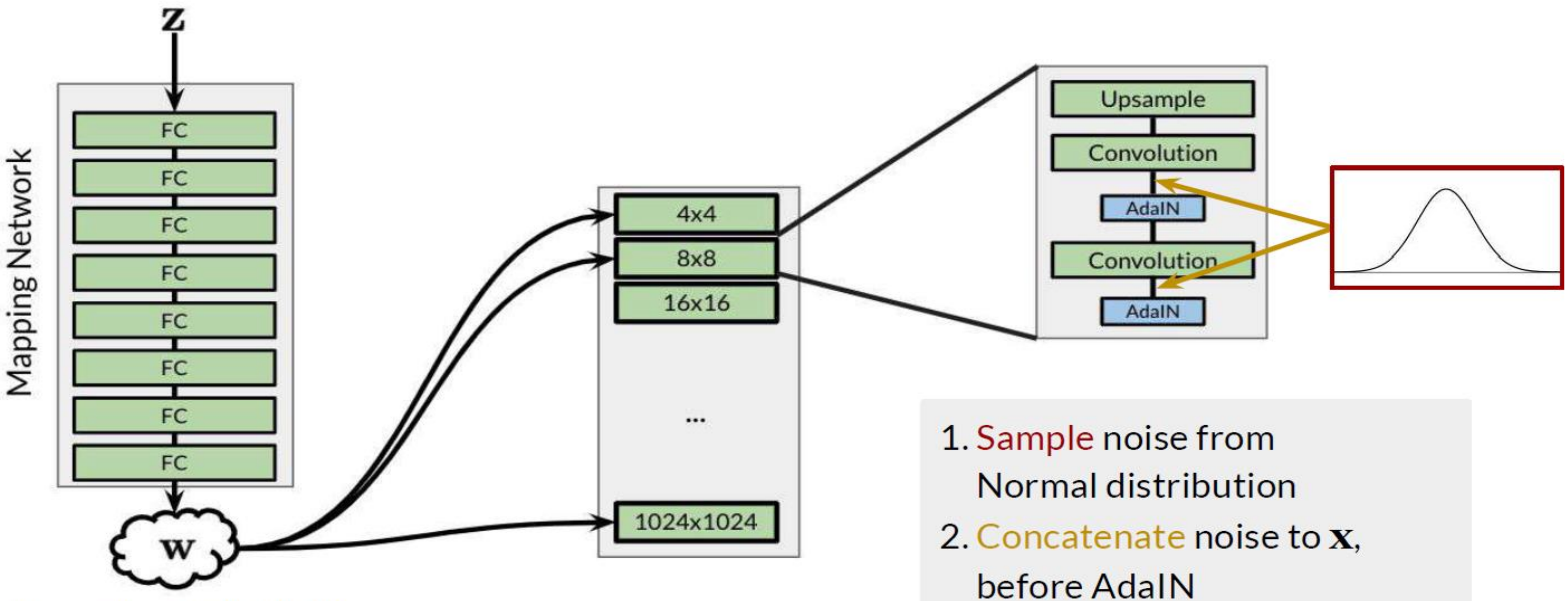
Build Better Generative Adversarial Networks

Style Mixing

Source: https://arxiv.org/abs/1812.04948

Build Better Generative Adversarial Networks

Stochastic Noise in Context



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

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

Build Better Generative Adversarial Networks

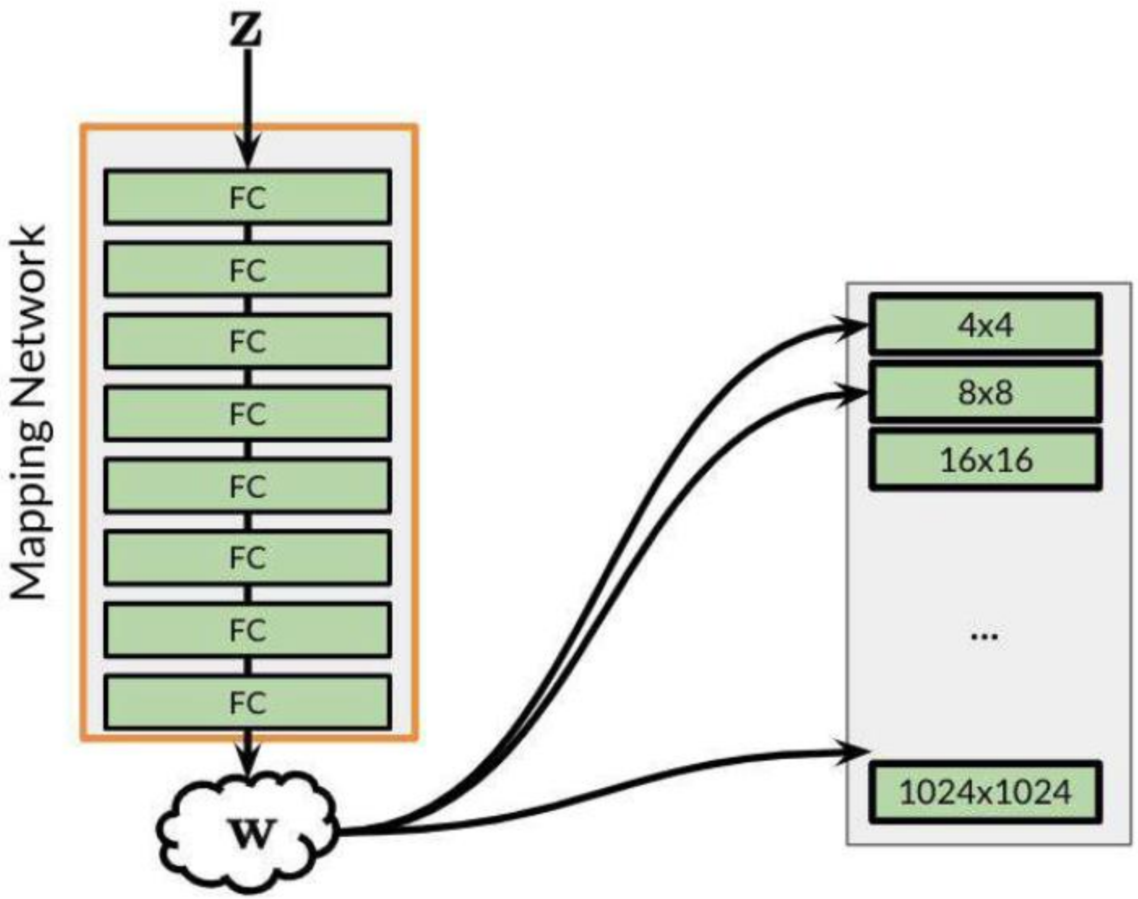
Stochastic Variation

Source: https://arxiv.org/abs/1812.04948

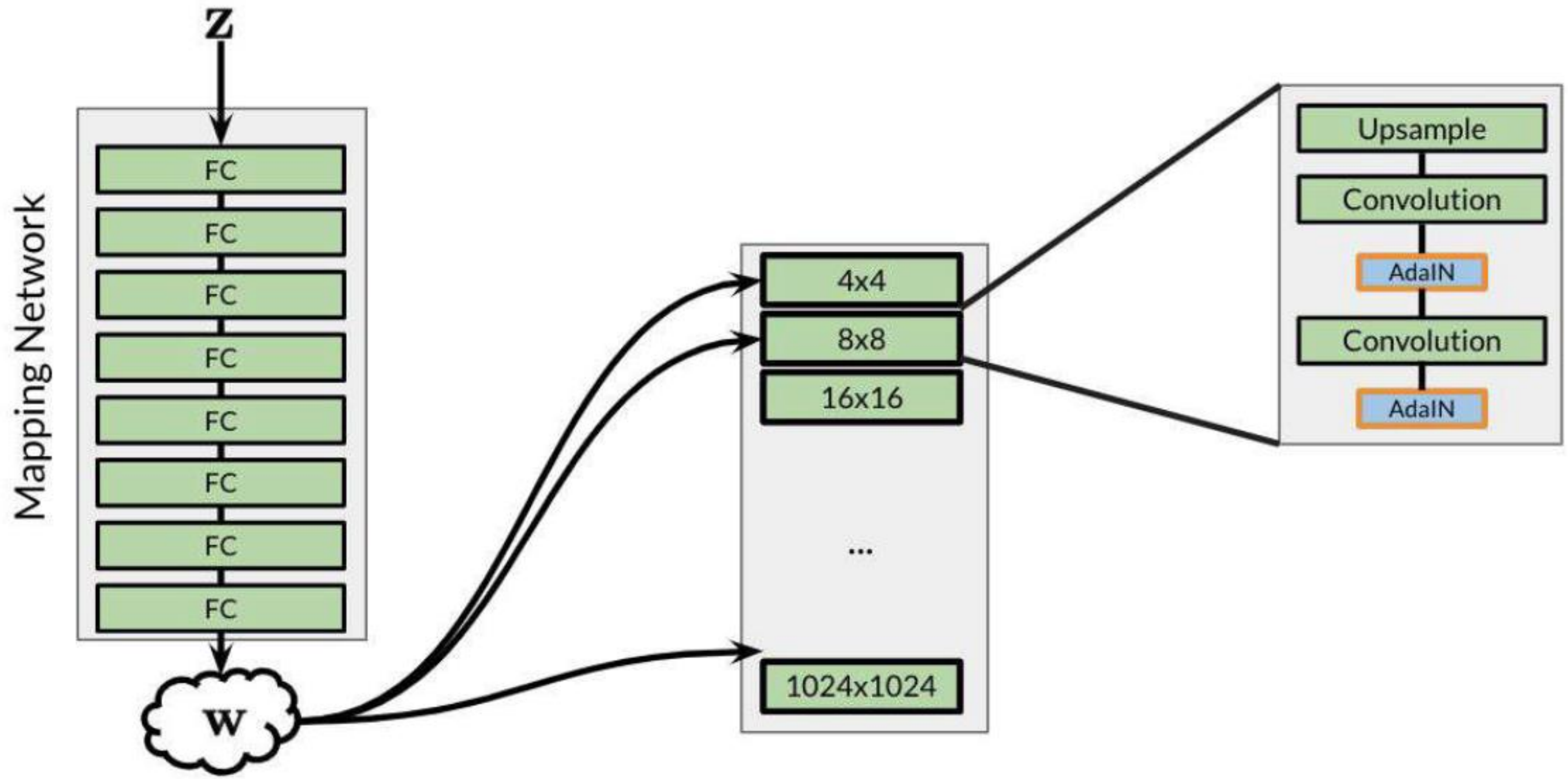
Putting everything together for Style GAN

StyleGAN Architecture: Progressive Growing

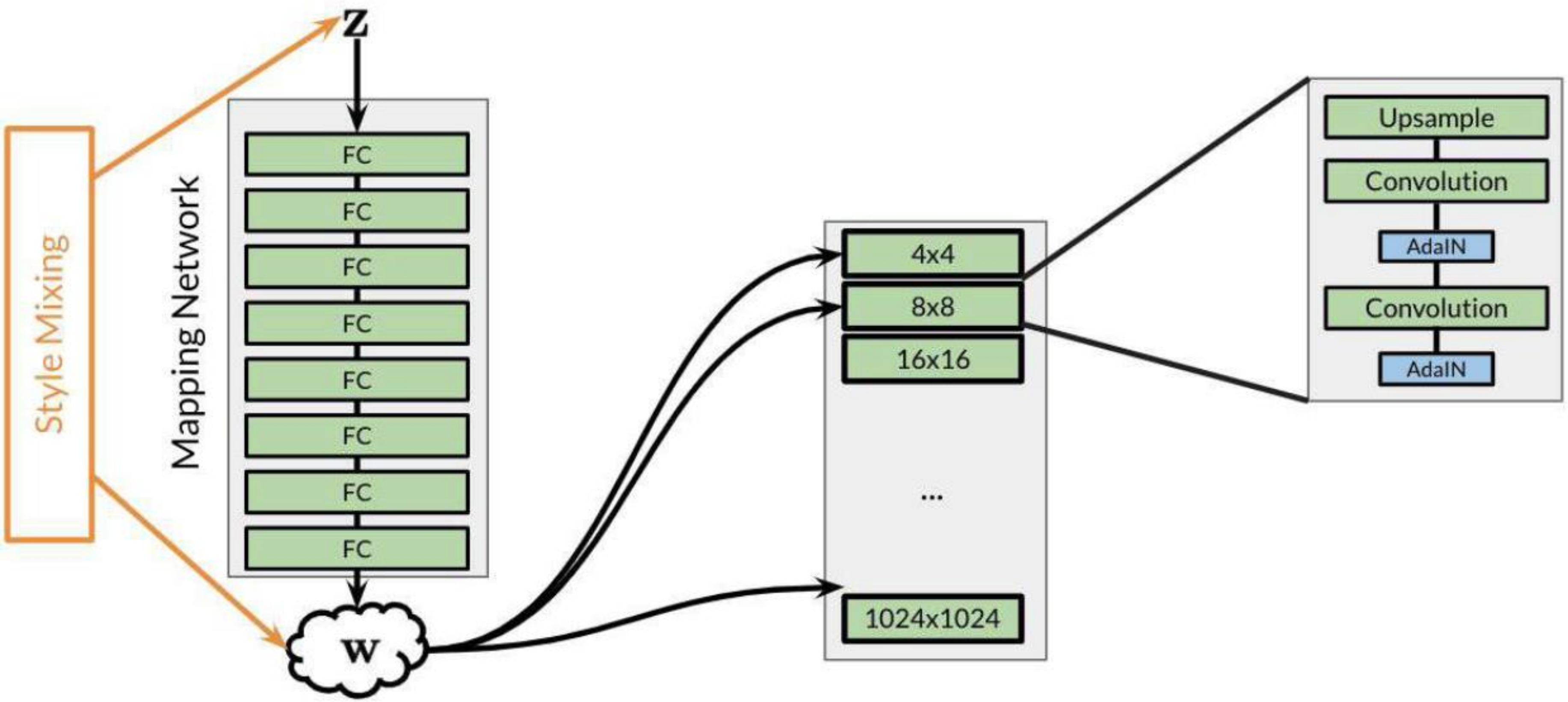
StyleGAN Architecture: Noise Mapping Network



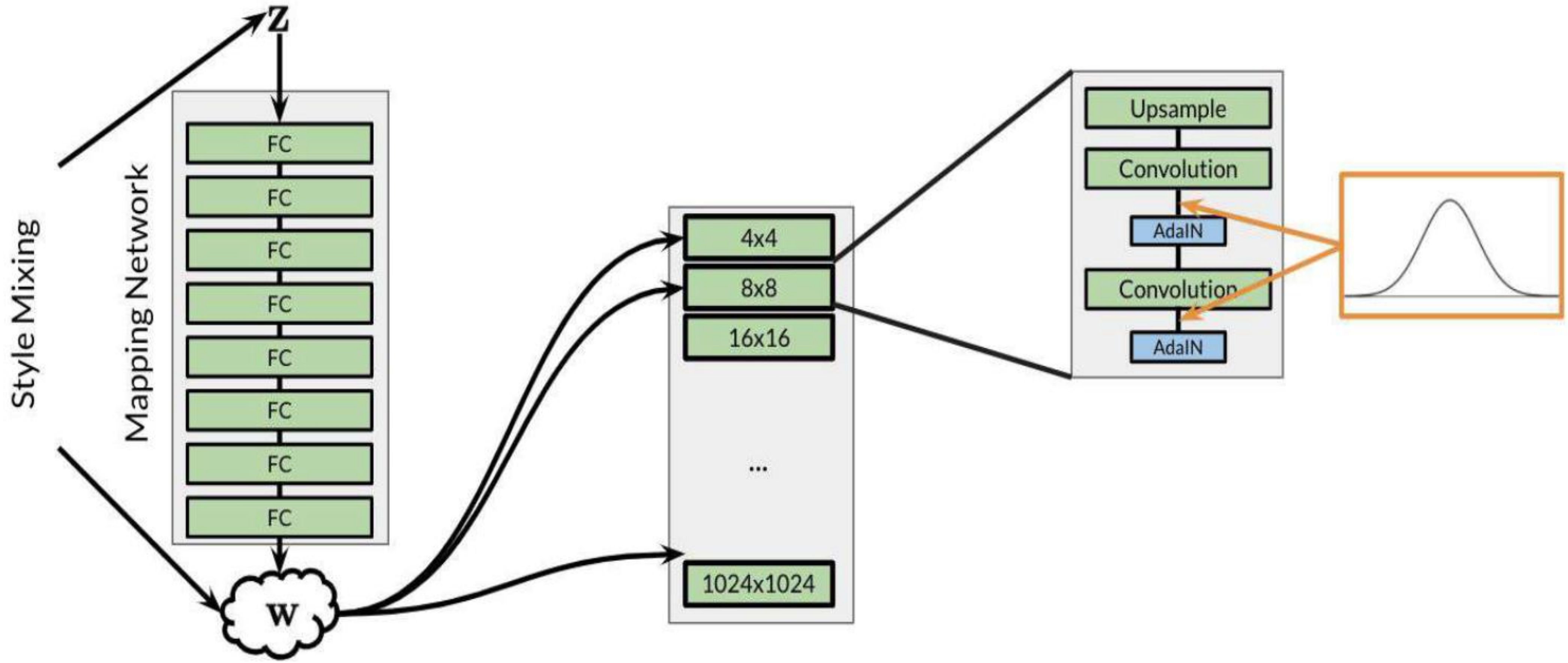
StyleGAN Architecture: AdaIN



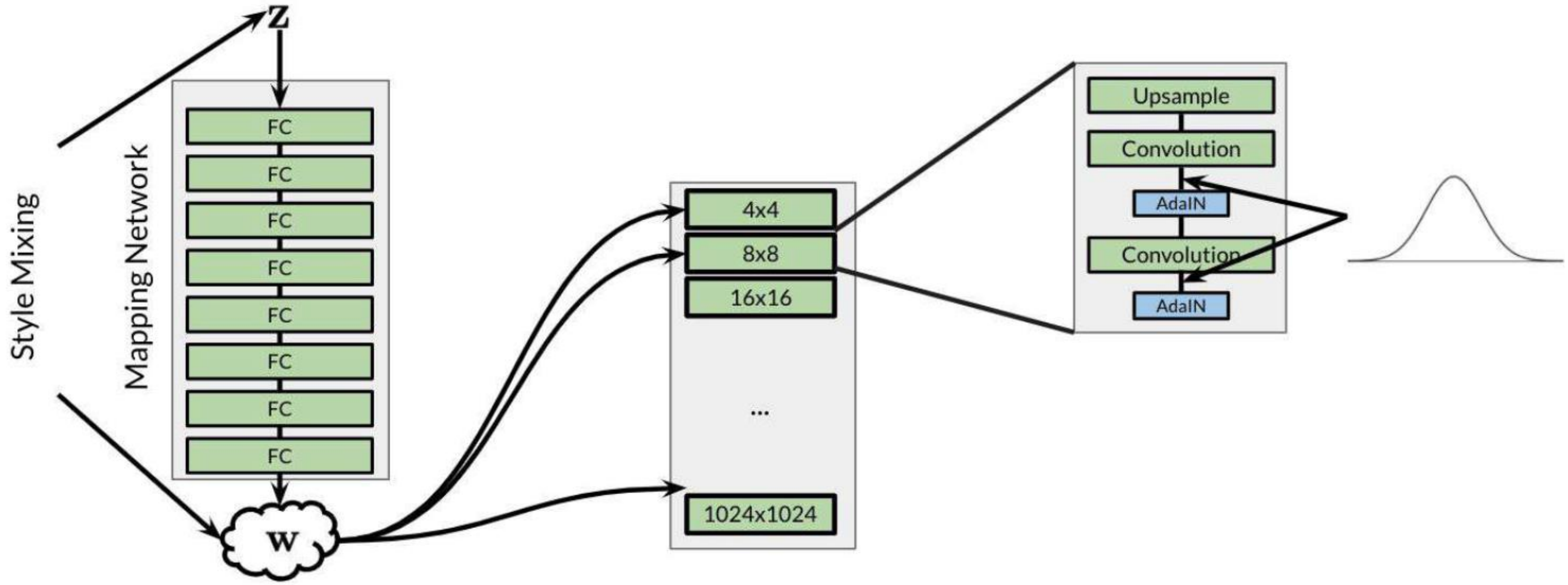
StyleGAN Architecture: Style Mixing



StyleGAN Architecture: Stochastic Noise



StyleGAN Architecture: That's a Wrap!



?

Thank You !!!