Practical GAN Training

Neural Networks Design And Application

Tricks

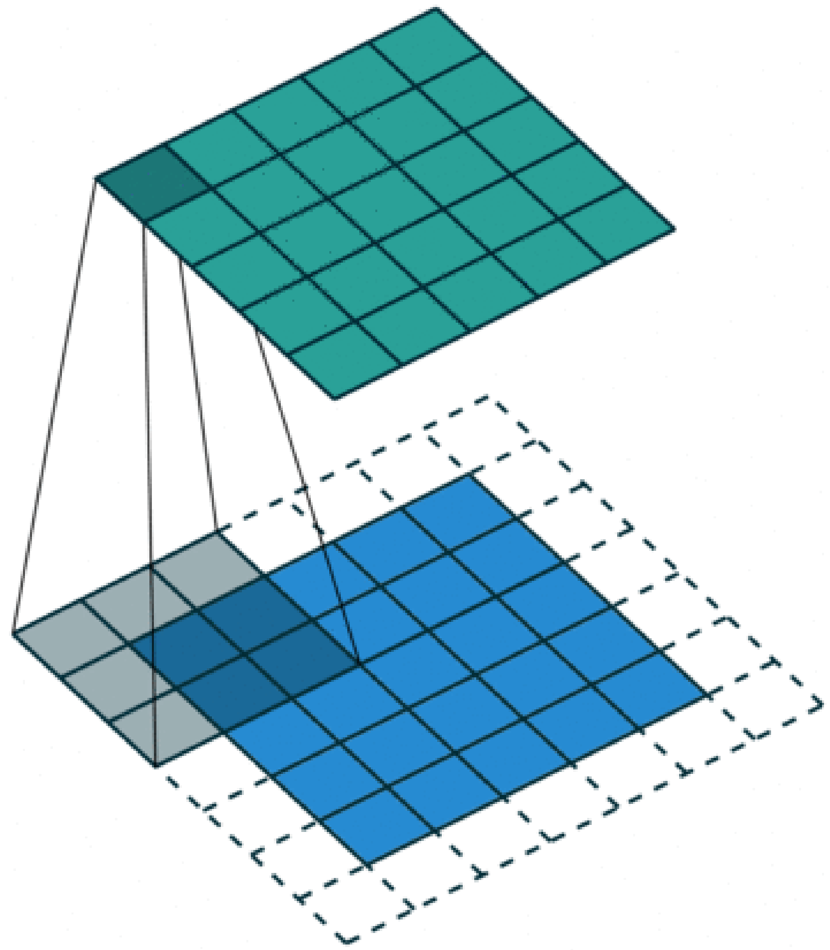
Architecture guidelines for stable Deep Convolutional GANs

- Replace any pooling layers with strided convolutions (discriminator) and fractional-strided convolutions (generator).
- Use batchnorm in both the generator and the discriminator.
- Remove fully connected hidden layers for deeper architectures.
- Use ReLU activation in generator for all layers except for the output, which uses Tanh.
- Use LeakyReLU activation in the discriminator for all layers.

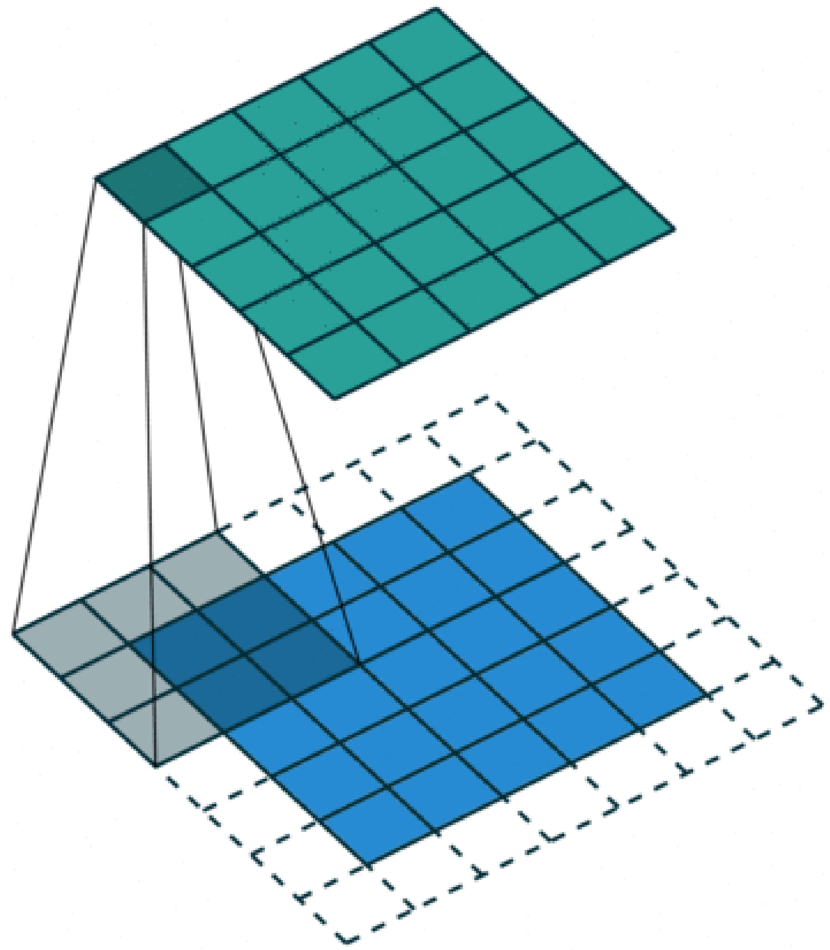
Radford, Alec, Luke Metz, and Soumith Chintala. "Unsupervised representation learning with deep convolutional generative adversarial networks." *arXiv preprint arXiv:1511.06434* (2015). https://arxiv.org/pdf/1511.06434.pdf

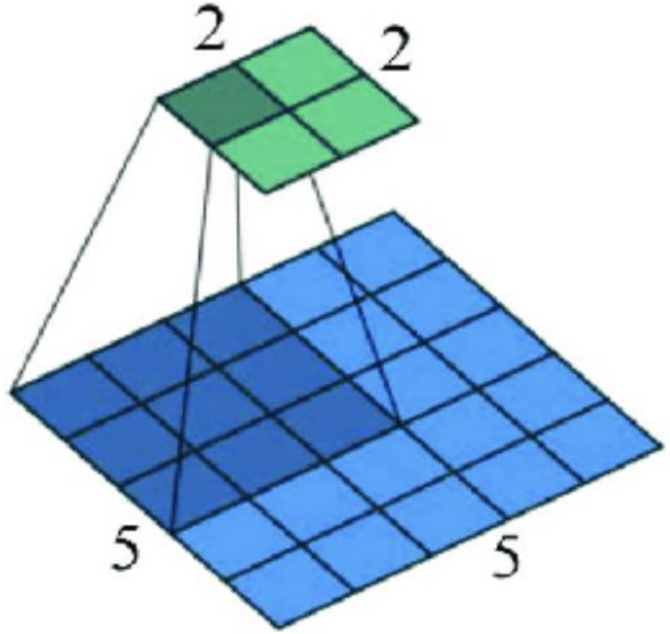
https://machinelearningmastery.com/how-to-train-stable-generative-adversarial-networks/

Replace pooling with strided conv layer



Replace pooling with strided conv layer





Stride 2 Padding 0

Strided Convolution

Batch normalization [BN]

Input: Values of x over a mini-batch: $\mathcal{B} = \{x_{1\ldots m}\}$;
Parameters to be learned: γ, β

Output: $\{y_i = \mathrm{BN}_{\gamma,\beta}(x_i)\}$

$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i \qquad \text{// mini-batch mean}$$

$$\sigma_{\mathcal{B}}^2 \leftarrow \frac{1}{m}\sum_{i=1}^{m} (x_i - \mu_{\mathcal{B}})^2 \qquad \text{// mini-batch variance}$$

$$\widehat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}} \qquad \text{// normalize}$$

$$y_i \leftarrow \gamma \widehat{x}_i + \beta \equiv \mathrm{BN}_{\gamma,\beta}(x_i) \qquad \text{// scale and shift}$$

Algorithm 1: Batch Normalizing Transform, applied to activation x over a mini-batch.

Rescaling for a batch

A linear model as output:
There are two learnable parameters

LeNet-5 in 1999

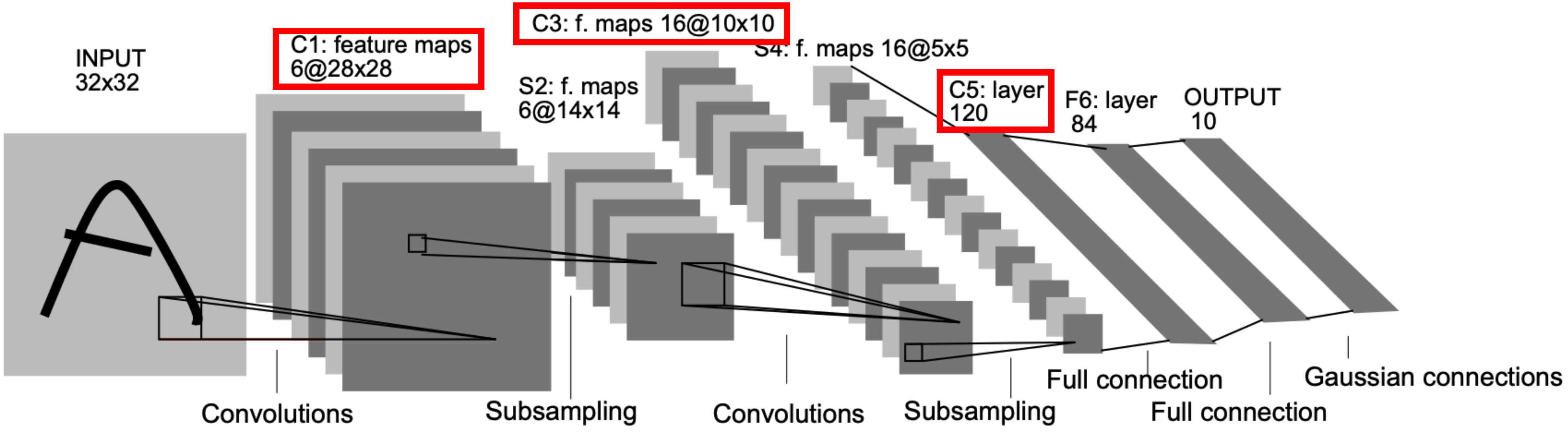


Fig. 1. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

LeCun, Yann, Patrick Haffner, Léon Bottou, and Yoshua Bengio. "Object recognition with gradient-based learning." In *Shape, contour and grouping in computer vision*, pp. 319-345. Springer, Berlin, Heidelberg, 1999.

Remove full connected layer

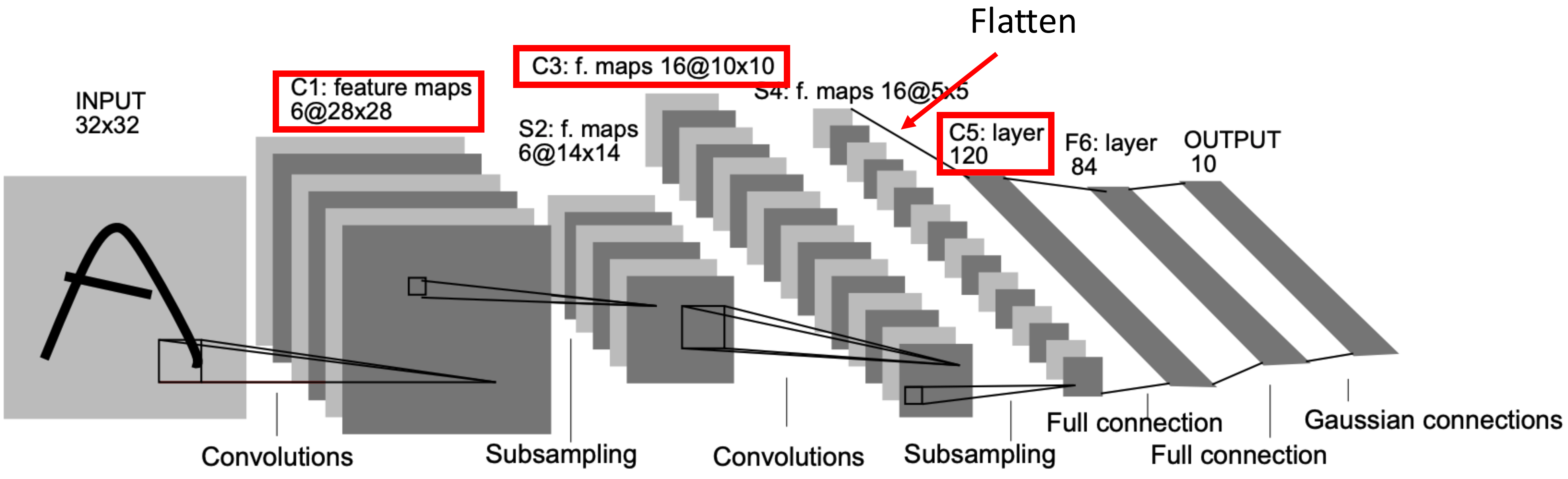


Fig. 1. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

LeCun, Yann, Patrick Haffner, Léon Bottou, and Yoshua Bengio. "Object recognition with gradient-based learning." In *Shape, contour and grouping in computer vision*, pp. 319-345. Springer, Berlin, Heidelberg, 1999.

Remove full connected layer

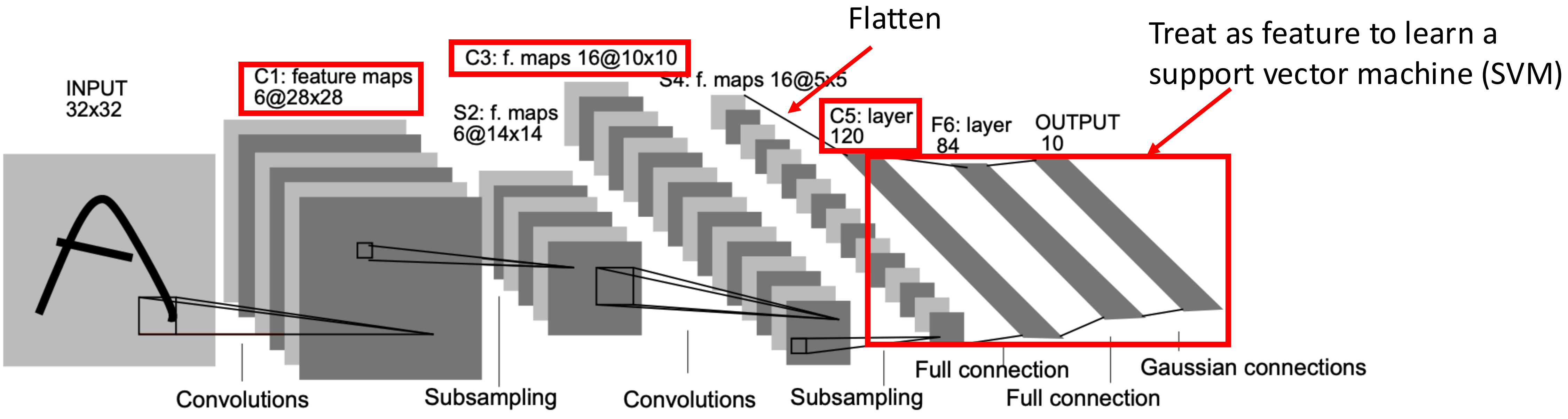


Fig. 1. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

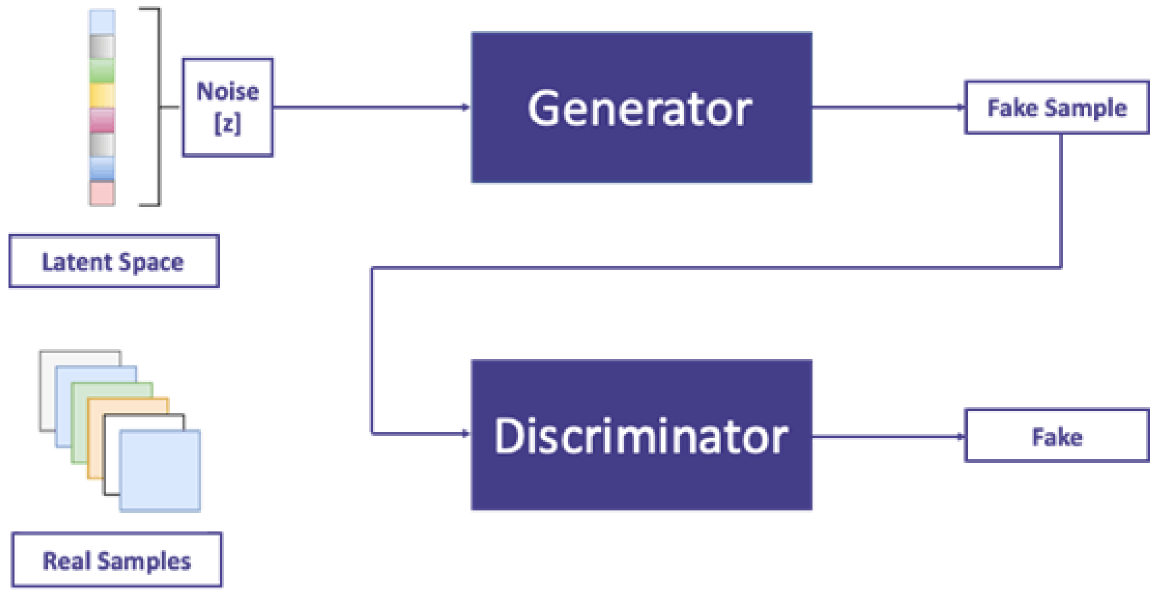
LeCun, Yann, Patrick Haffner, Léon Bottou, and Yoshua Bengio. "Object recognition with gradient-based learning." In *Shape, contour and grouping in computer vision*, pp. 319-345. Springer, Berlin, Heidelberg, 1999.

Architectures

- CycleGAN
- StyleGAN
- …

https://machinelearningmastery.com/tour-of-generative-adversarial-network-models/
https://neptune.ai/blog/6-gan-architectures

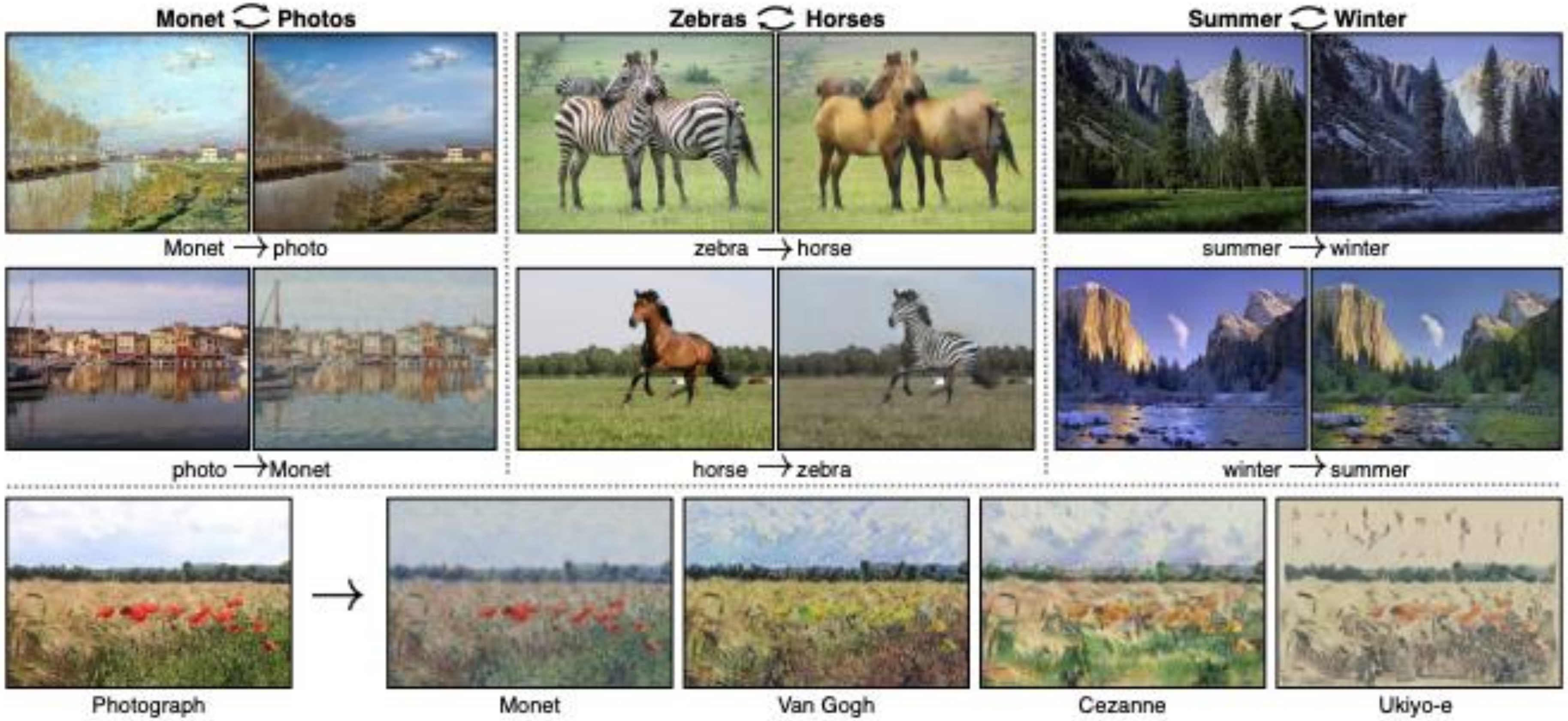
GAN



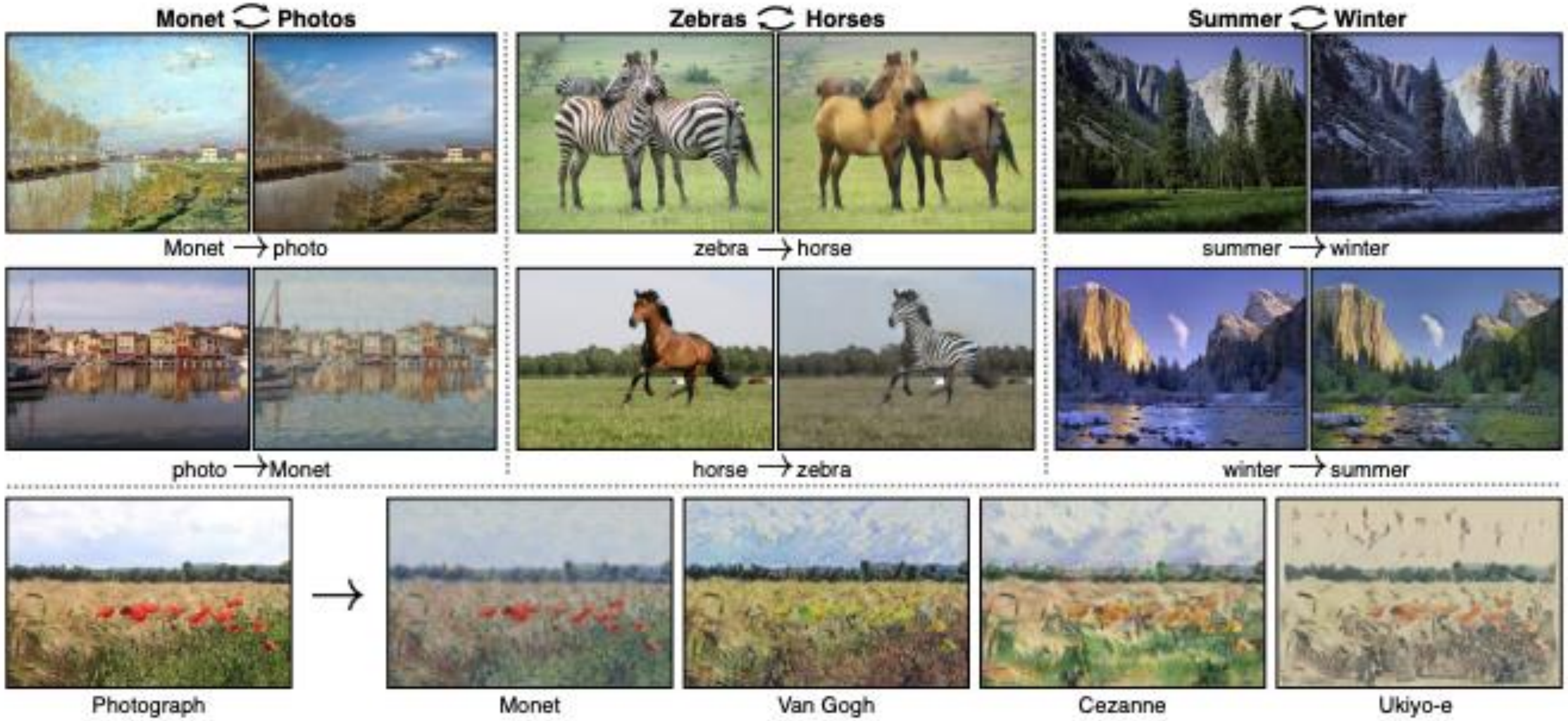
$$\min_G \max_D V(D, G) = \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z} \sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))].$$

Image credit https://neptune.ai/blog/6-gan-architectures

CycleGAN

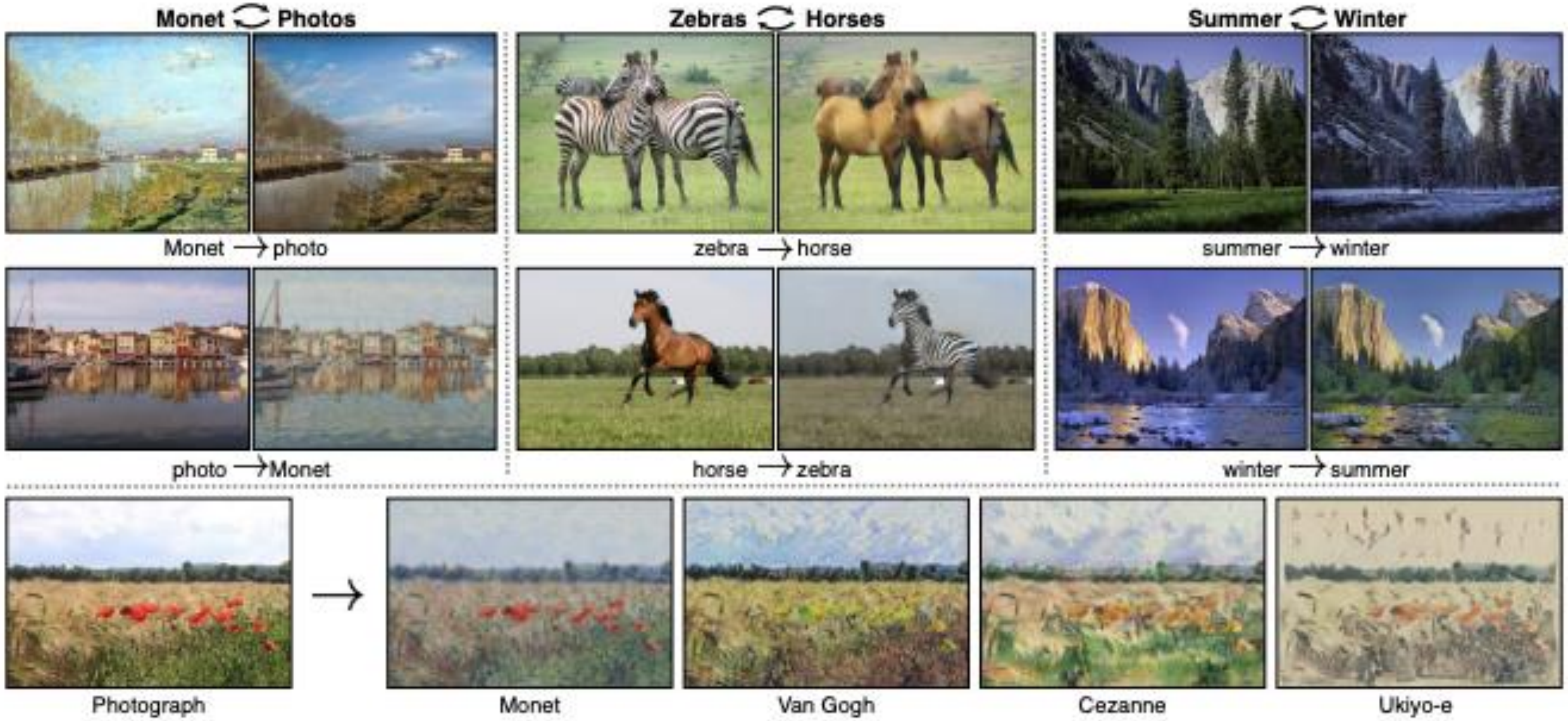

Monet ⟲ Photos
Zebras ⟲ Horses
Summer ⟲ Winter
Monet → photo
zebra → horse
summer → winter
photo → Monet
horse → zebra
winter → summer
Photograph
Monet
Van Gogh
Cezanne
Ukiyo-e

CycleGAN



Q1: content changed?

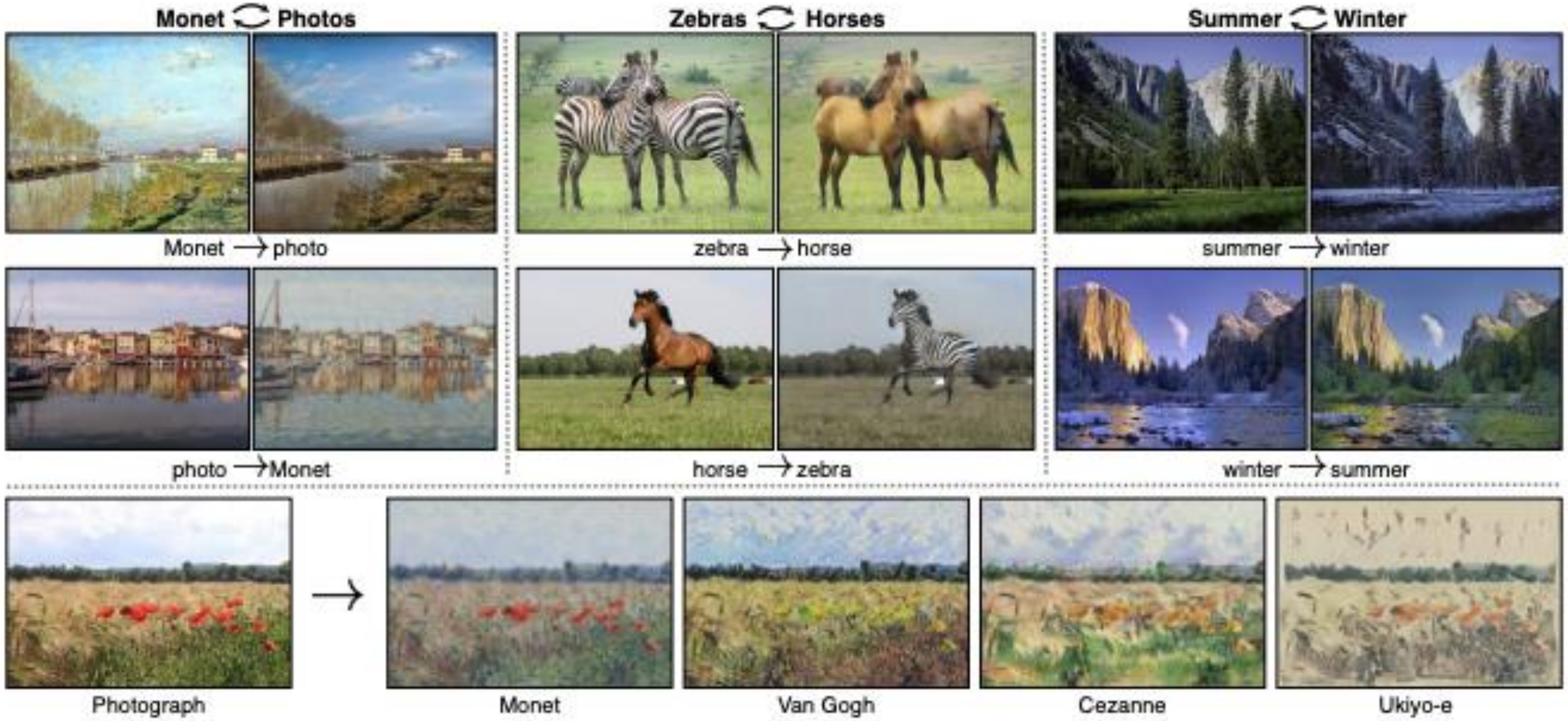
CycleGAN



Q1: content changed?

Q2: what changed?

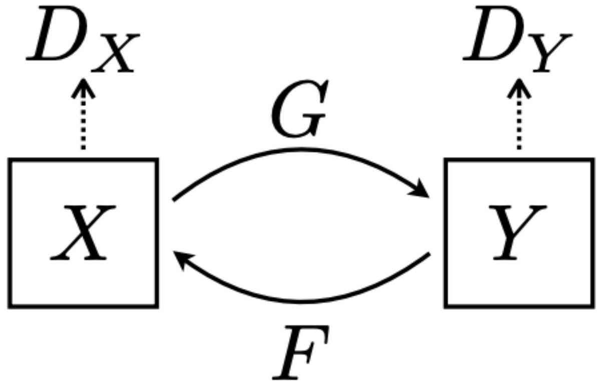
CycleGAN



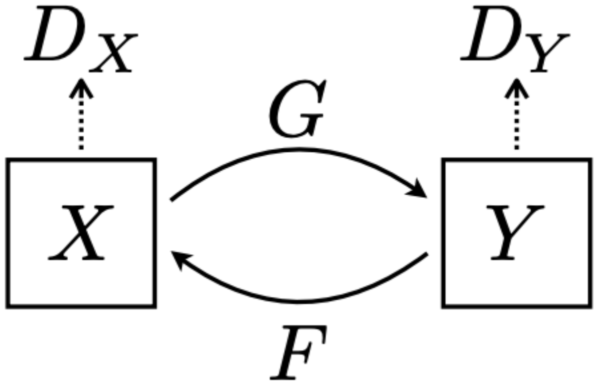
Q1: content changed?

Q2: what changed?→image-to-image translation

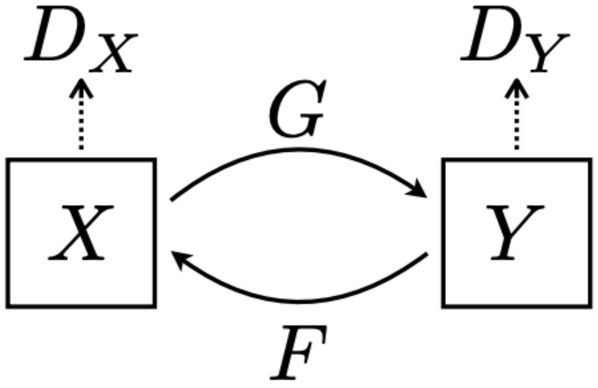
CycleGAN

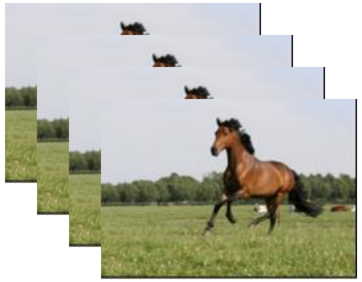


CycleGAN

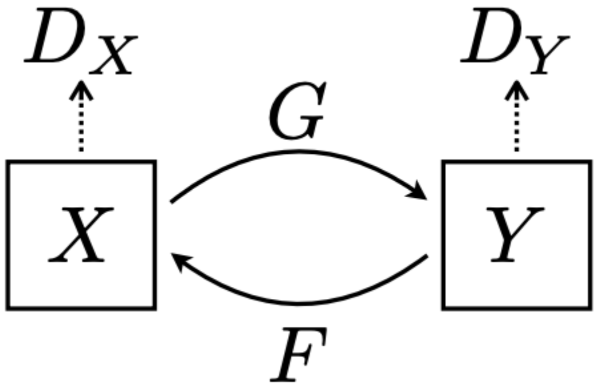


CycleGAN

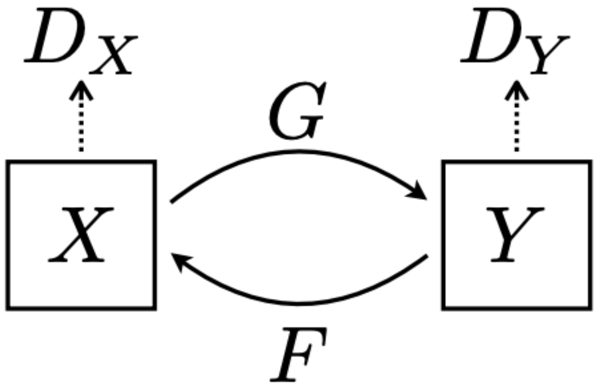


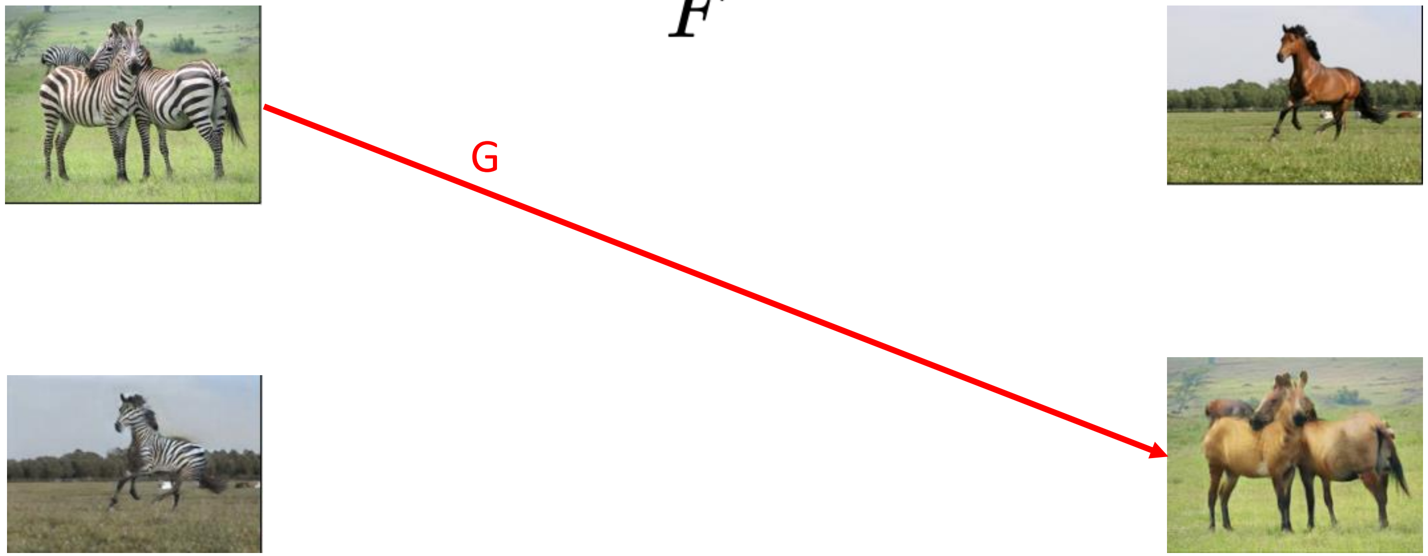


CycleGAN

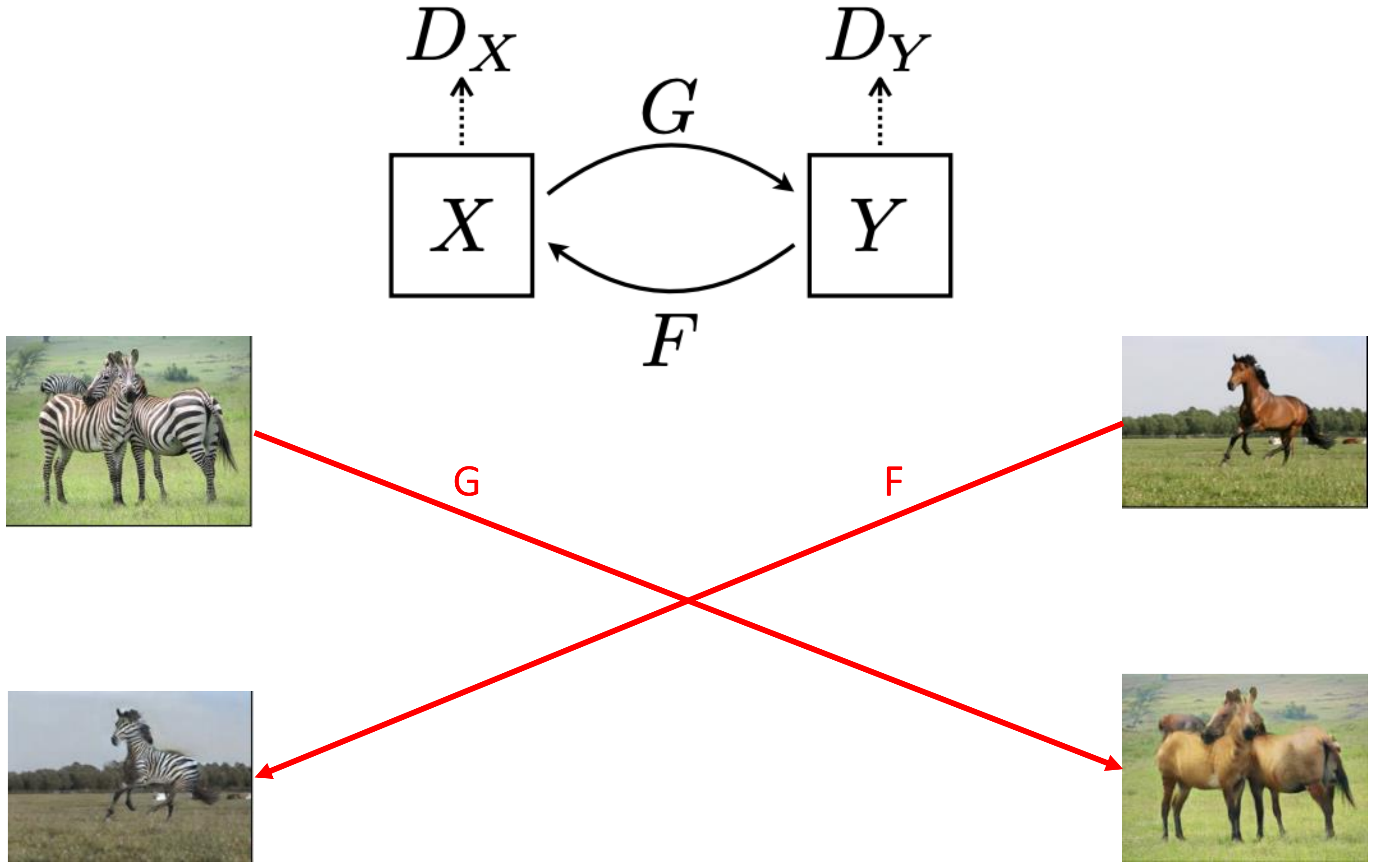


CycleGAN

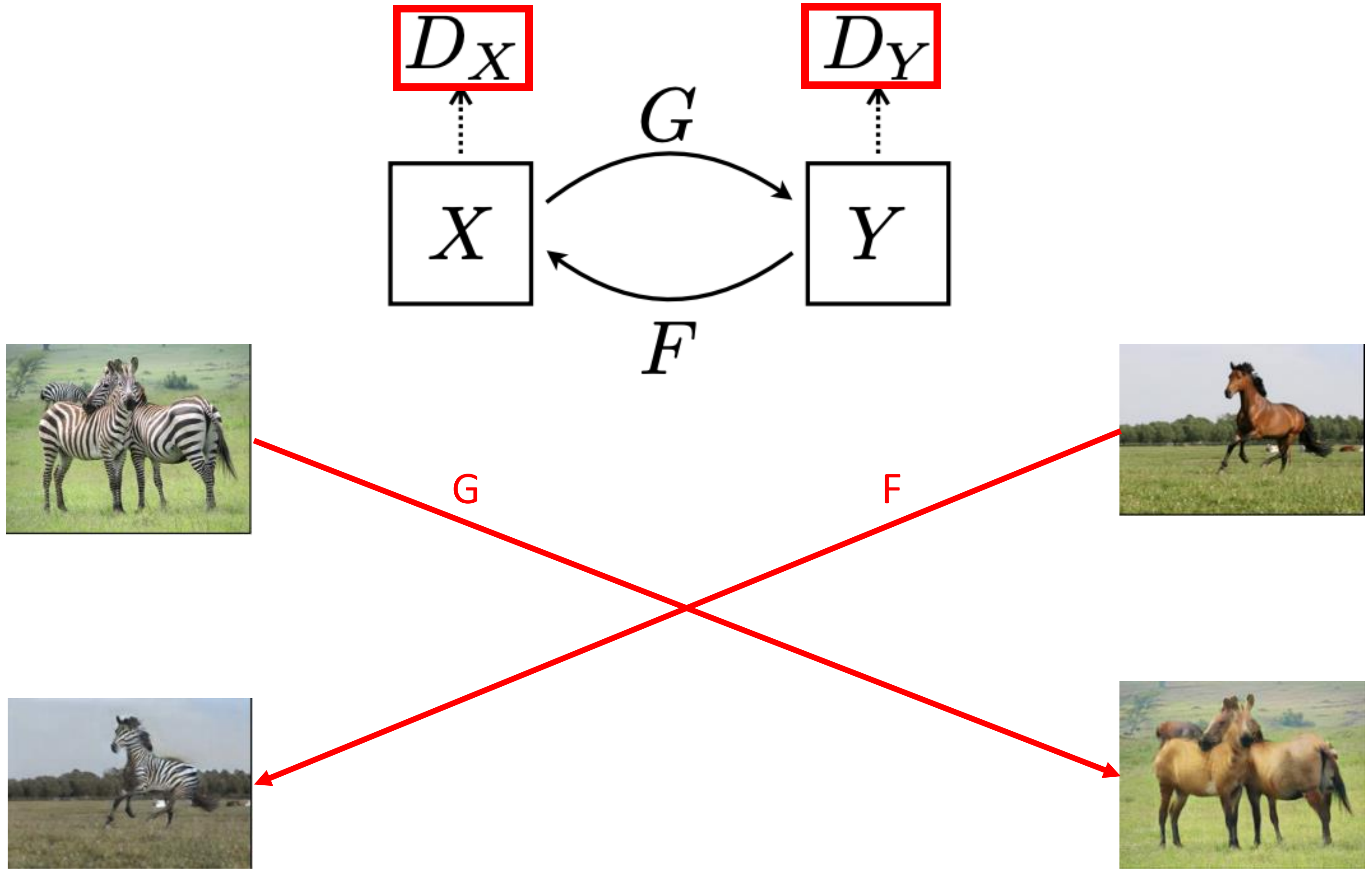




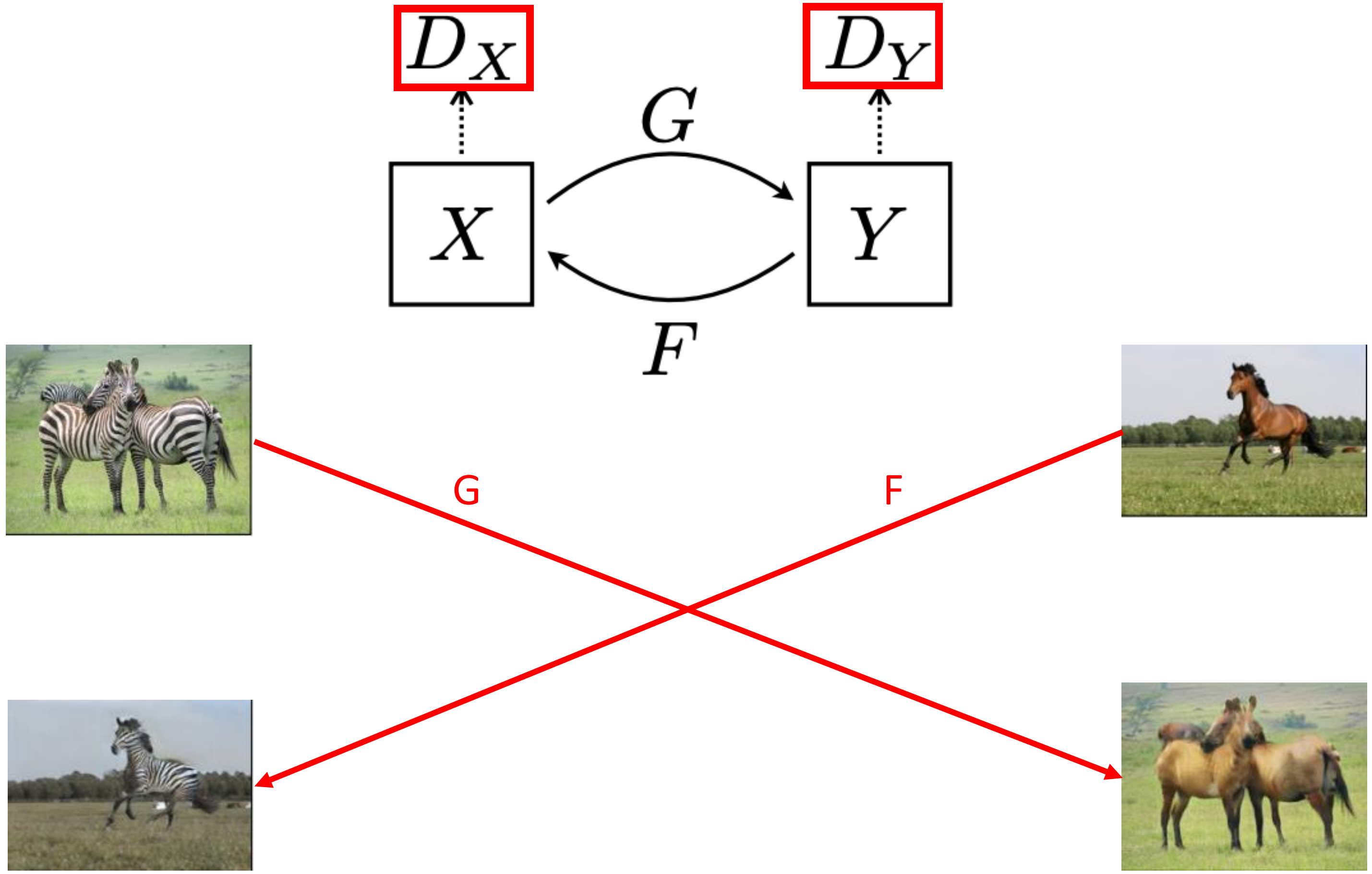
CycleGAN



CycleGAN

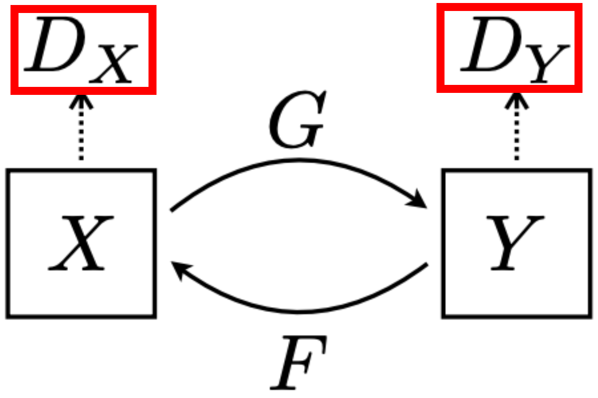


CycleGAN

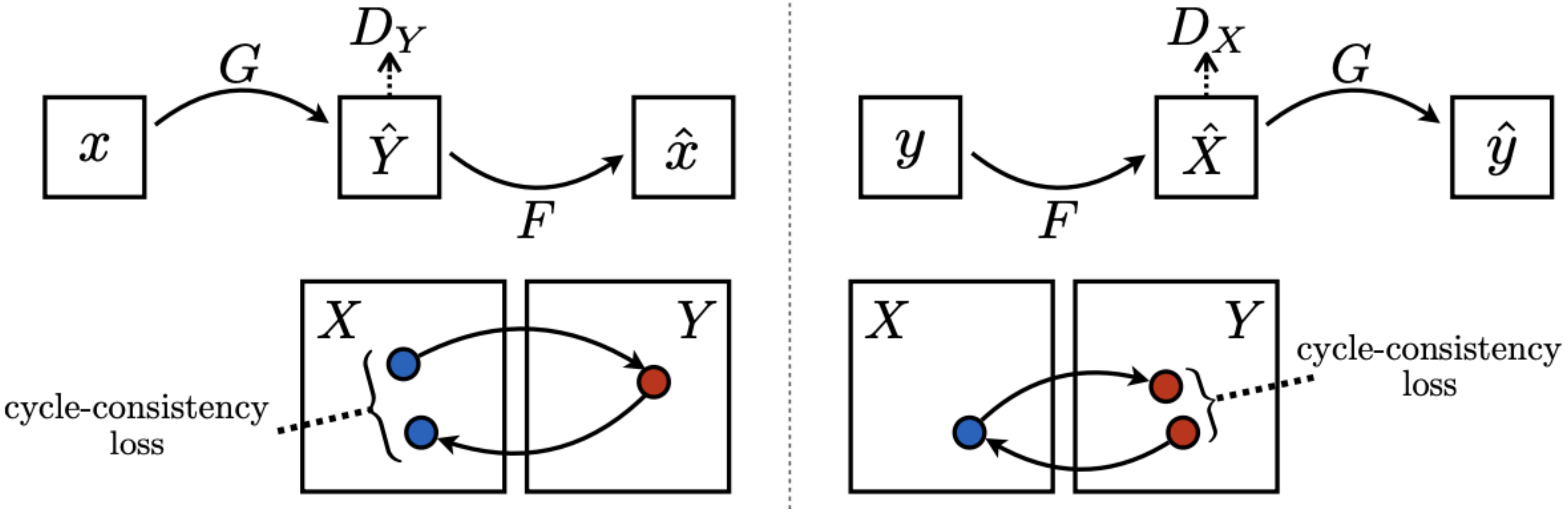


Q: how to ensure the content is not changed?

CycleGAN

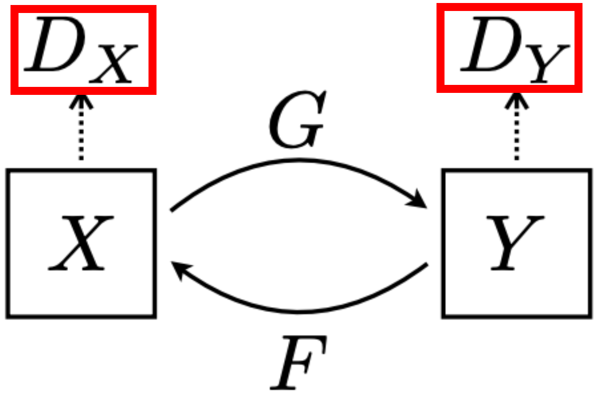


D_X
D_Y
G
X
Y
F

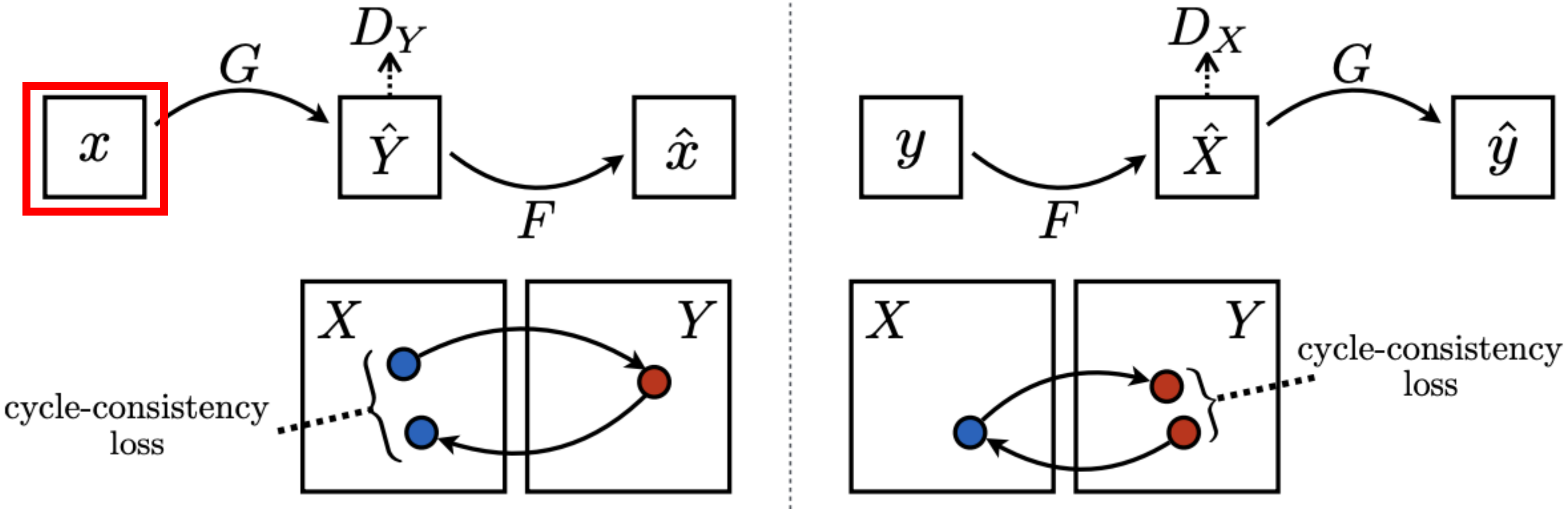


G
D_Y
x
$\hat{Y}$
$\hat{x}$
F
X
Y
cycle-consistency loss
D_X
G
y
$\hat{X}$
$\hat{y}$
F
X
Y
cycle-consistency loss

CycleGAN

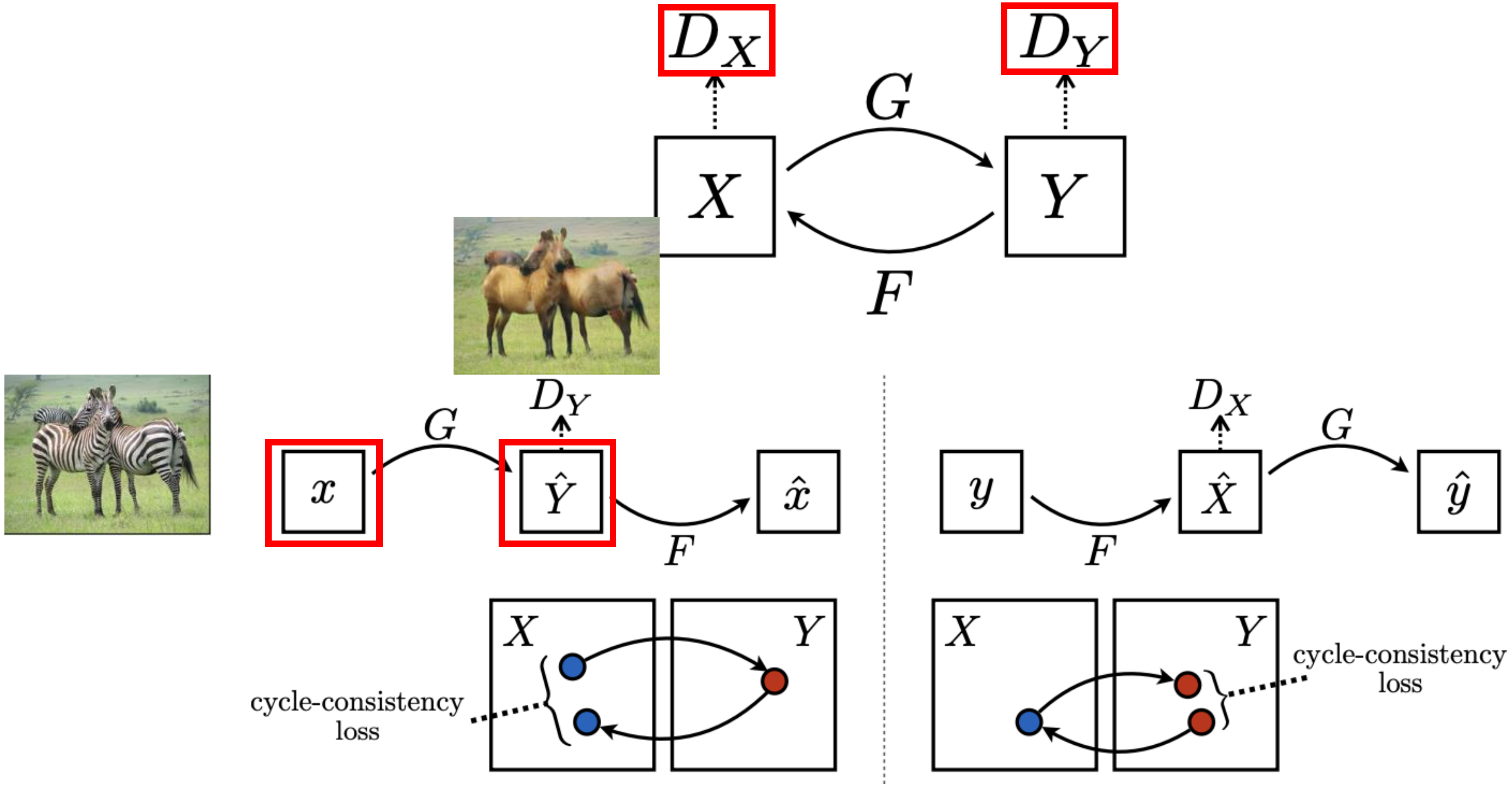


D_X
D_Y
G
X
Y
F



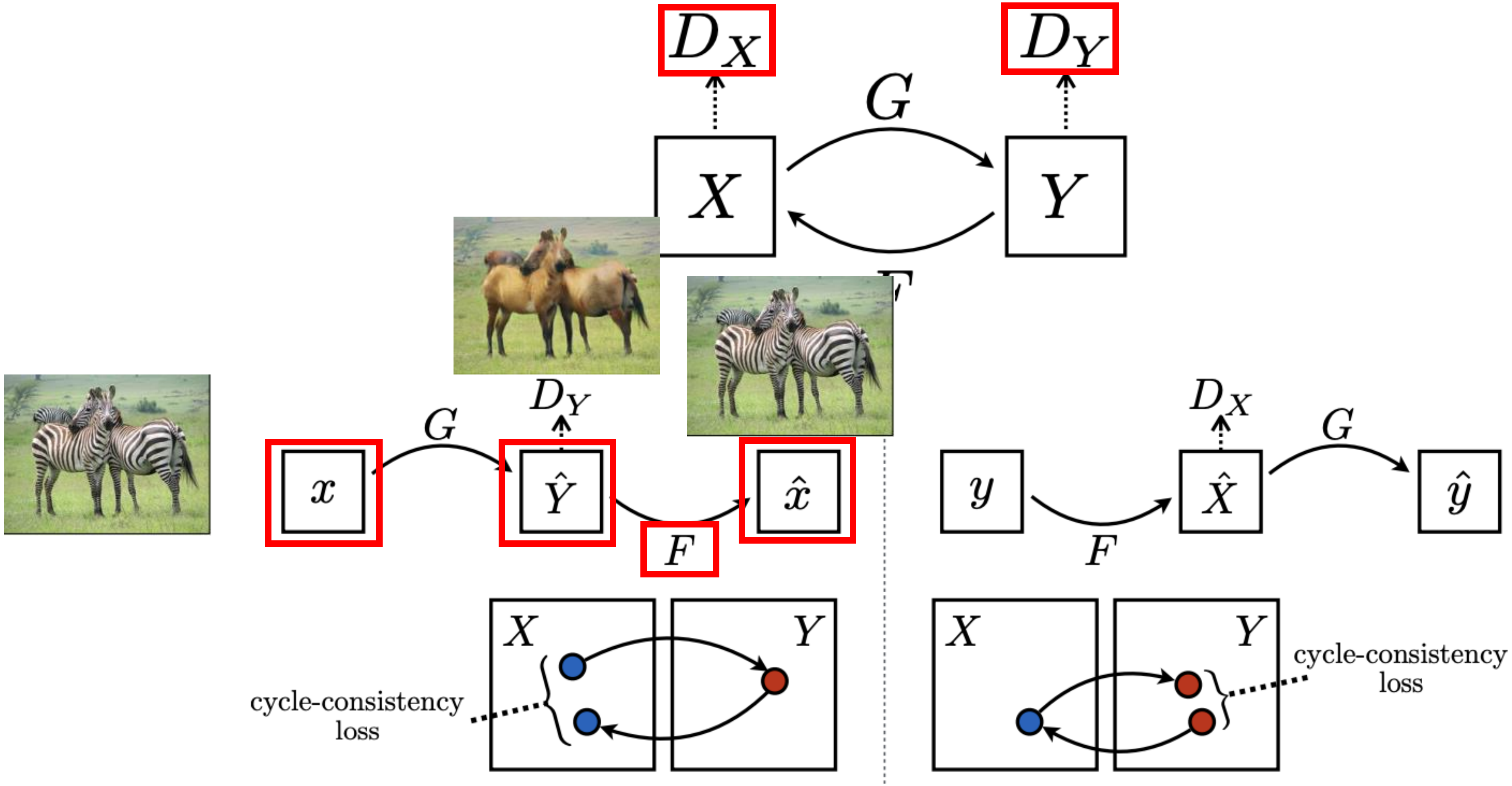
x
G
D_Y
$\hat{Y}$
F
$\hat{x}$
y
F
D_X
$\hat{X}$
G
$\hat{y}$
X
Y
cycle-consistency
loss
X
Y
cycle-consistency
loss

CycleGAN



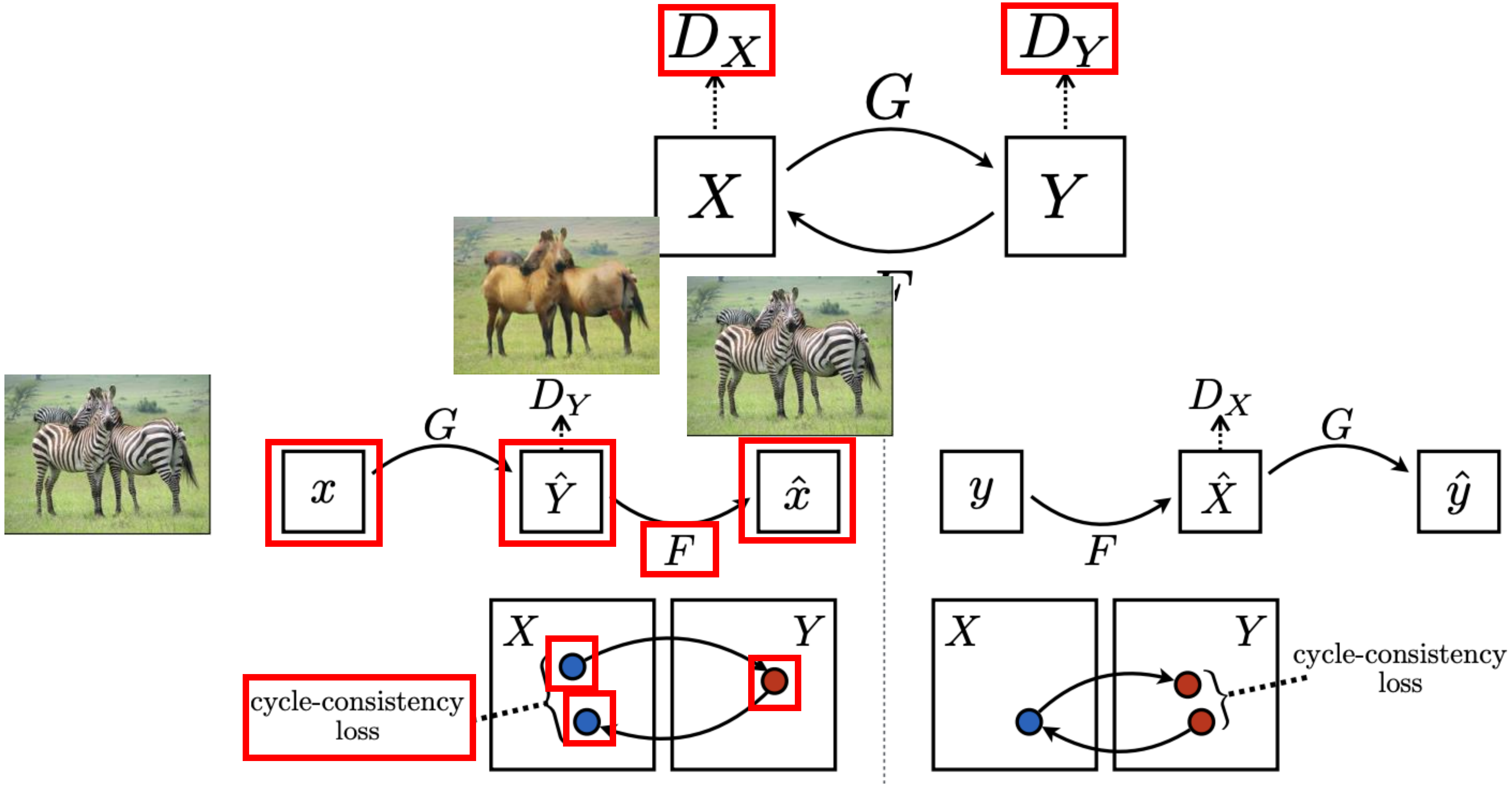
D_X
D_Y
G
X
Y
F
x
D_Y
$\hat{Y}$
$\hat{x}$
y
D_X
$\hat{X}$
$\hat{y}$
cycle-consistency loss
cycle-consistency loss

CycleGAN



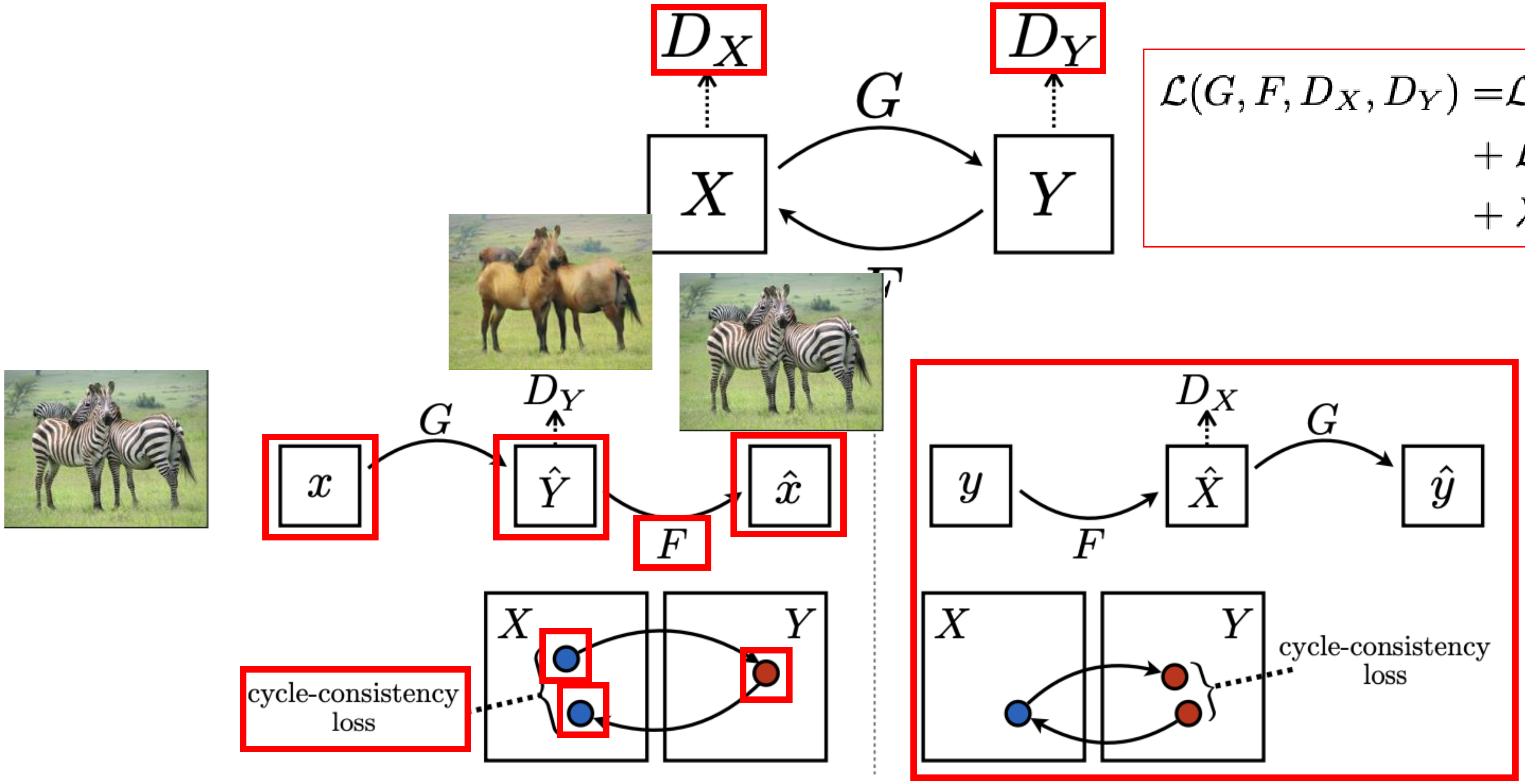
D_X
D_Y
G
X
Y
F
x
G
D_Y
$\hat{Y}$
F
$\hat{x}$
y
F
D_X
$\hat{X}$
G
$\hat{y}$
X
Y
cycle-consistency loss
X
Y
cycle-consistency loss

CycleGAN



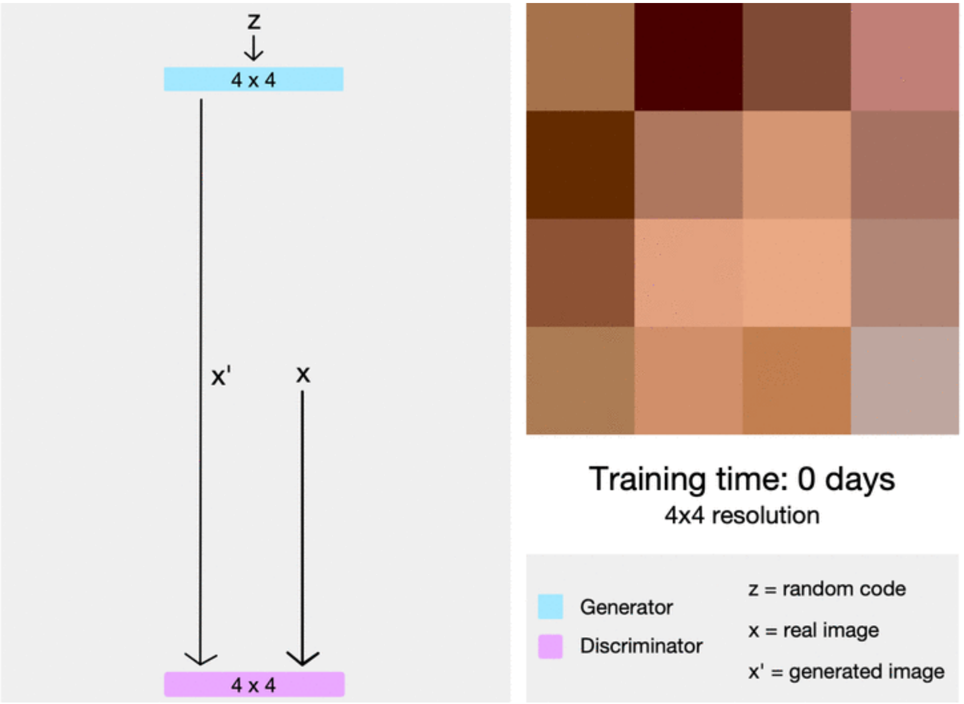
D_X
D_Y
G
X
Y
F
x
G
D_Y
$\hat{Y}$
F
$\hat{x}$
y
F
D_X
$\hat{X}$
G
$\hat{y}$
X
Y
cycle-consistency loss
X
Y
cycle-consistency loss

CycleGAN



$$
\begin{aligned}
\mathcal{L}(G, F, D_X, D_Y) = & \mathcal{L}_{\text{GAN}}(G, D_Y, X, Y) \\
& + \mathcal{L}_{\text{GAN}}(F, D_X, Y, X) \\
& + \lambda \mathcal{L}_{\text{cyc}}(G, F),
\end{aligned}
$$

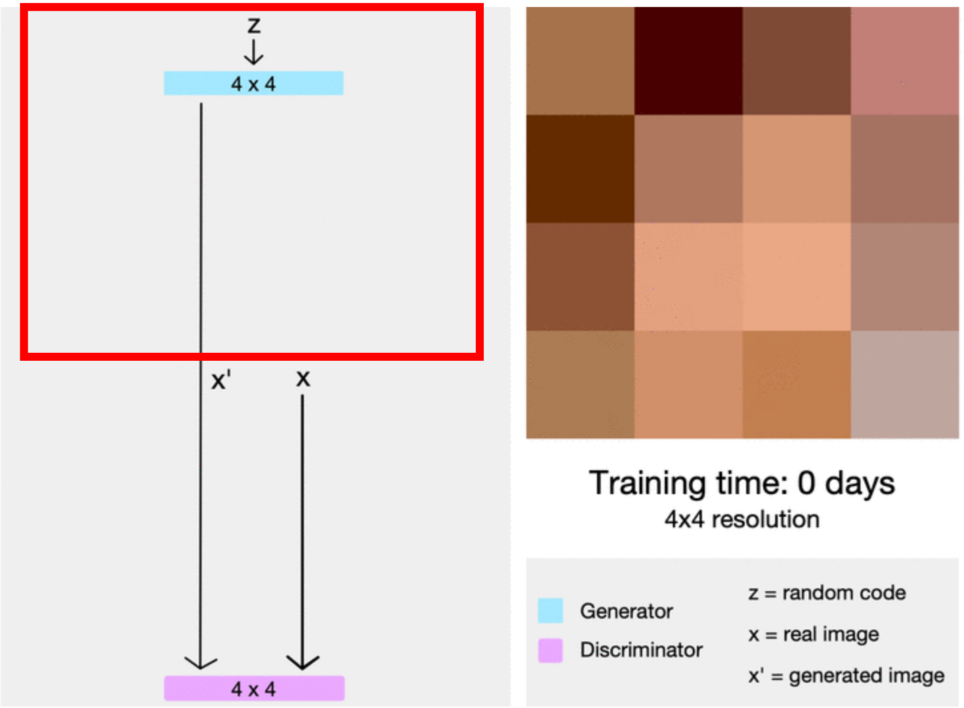
StyleGAN



Progressive GAN https://arxiv.org/pdf/1710.10196.pdf

https://towardsdatascience.com/explained-a-style-based-generator-architecture-for-gans-generating-and-tuning-realistic-6cb2be0f431

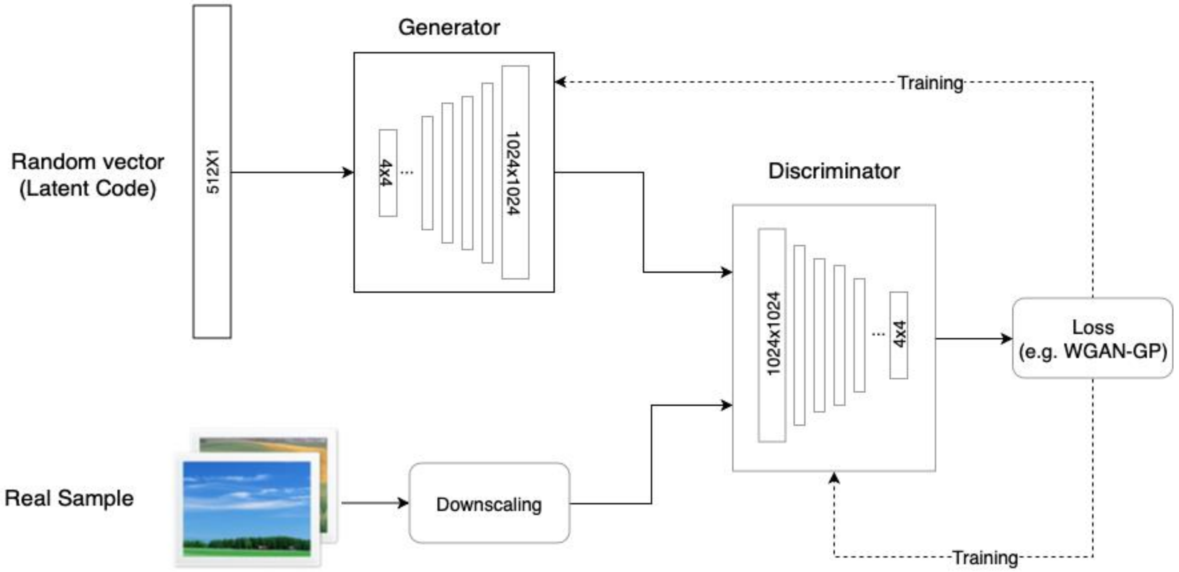
StyleGAN



Progressive GAN https://arxiv.org/pdf/1710.10196.pdf

https://towardsdatascience.com/explained-a-style-based-generator-architecture-for-gans-generating-and-tuning-realistic-6cb2be0f431

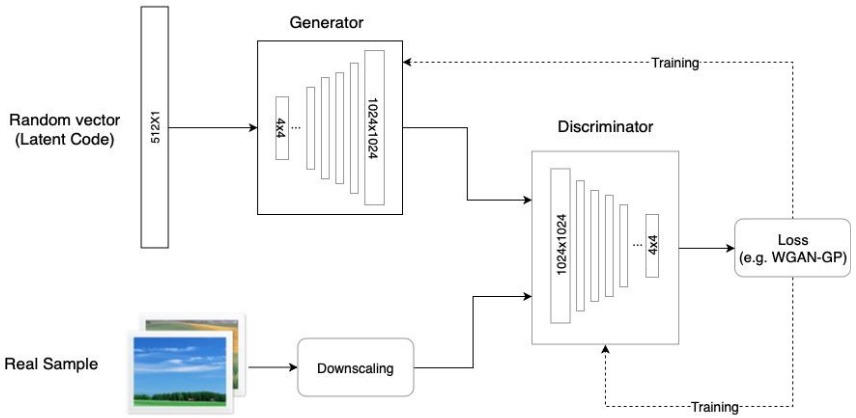
StyleGAN



Progressive GAN https://arxiv.org/pdf/1710.10196.pdf

https://towardsdatascience.com/explained-a-style-based-generator-architecture-for-gans-generating-and-tuning-realistic-6cb2be0f431

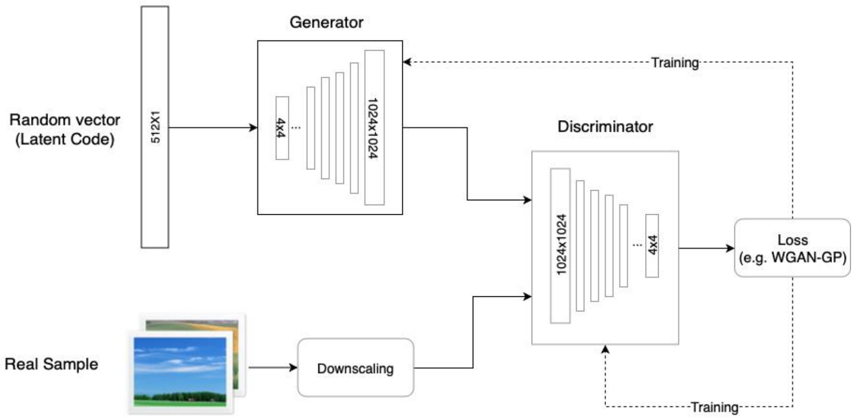
StyleGAN



1. Coarse - resolution of up to 82 - affects pose, general hair style, face shape, etc
2. Middle - resolution of 162 to 322 - affects finer facial features, hair style, eyes open/closed, etc.
3. Fine - resolution of 642 to 10242 - affects color scheme (eye, hair and skin) and micro features.

Karras, Tero, Samuli Laine, and Timo Aila. "A style-based generator architecture for generative adversarial networks." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4401-4410. 2019. https://arxiv.org/pdf/1812.04948.pdf
https://towardsdatascience.com/explained-a-style-based-generator-architecture-for-gans-generating-and-tuning-realistic-6cb2be0f431

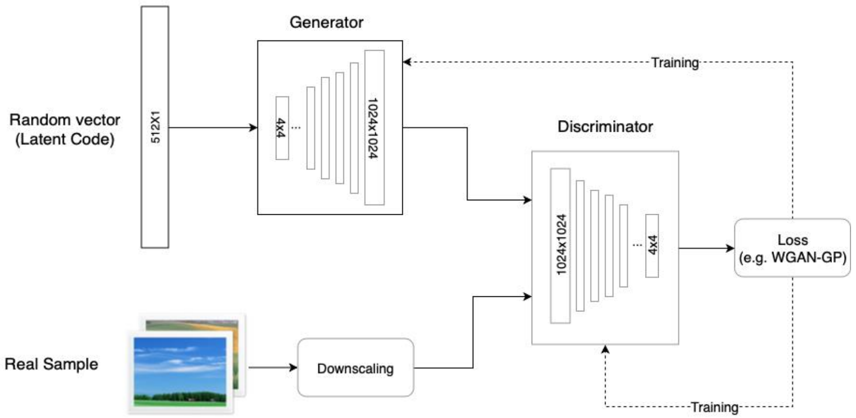
StyleGAN



1. Coarse - resolution of up to 82 - affects pose, general hair style, face shape, etc
2. Middle - resolution of 162 to 322 - affects finer facial features, hair style, eyes open/closed, etc.
3. Fine - resolution of 642 to 10242 - affects color scheme (eye, hair and skin) and micro features.

Karras, Tero, Samuli Laine, and Timo Aila. "A style-based generator architecture for generative adversarial networks." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4401-4410. 2019. https://arxiv.org/pdf/1812.04948.pdf
https://towardsdatascience.com/explained-a-style-based-generator-architecture-for-gans-generating-and-tuning-realistic-6cb2be0f431

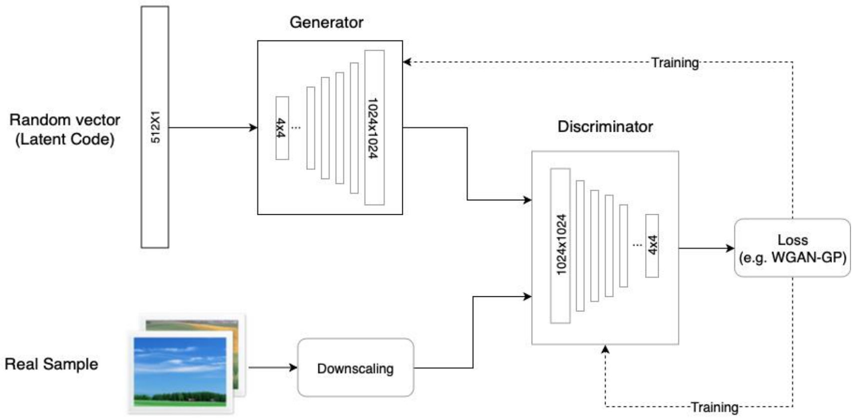
StyleGAN



1. Coarse - resolution of up to 82 - affects pose, general hair style, face shape, etc
2. Middle - resolution of 162 to 322 - affects finer facial features, hair style, eyes open/closed, etc.
3. Fine - resolution of 642 to 10242 - affects color scheme (eye, hair and skin) and micro features.

Karras, Tero, Samuli Laine, and Timo Aila. "A style-based generator architecture for generative adversarial networks." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4401-4410. 2019. https://arxiv.org/pdf/1812.04948.pdf
https://towardsdatascience.com/explained-a-style-based-generator-architecture-for-gans-generating-and-tuning-realistic-6cb2be0f431

StyleGAN

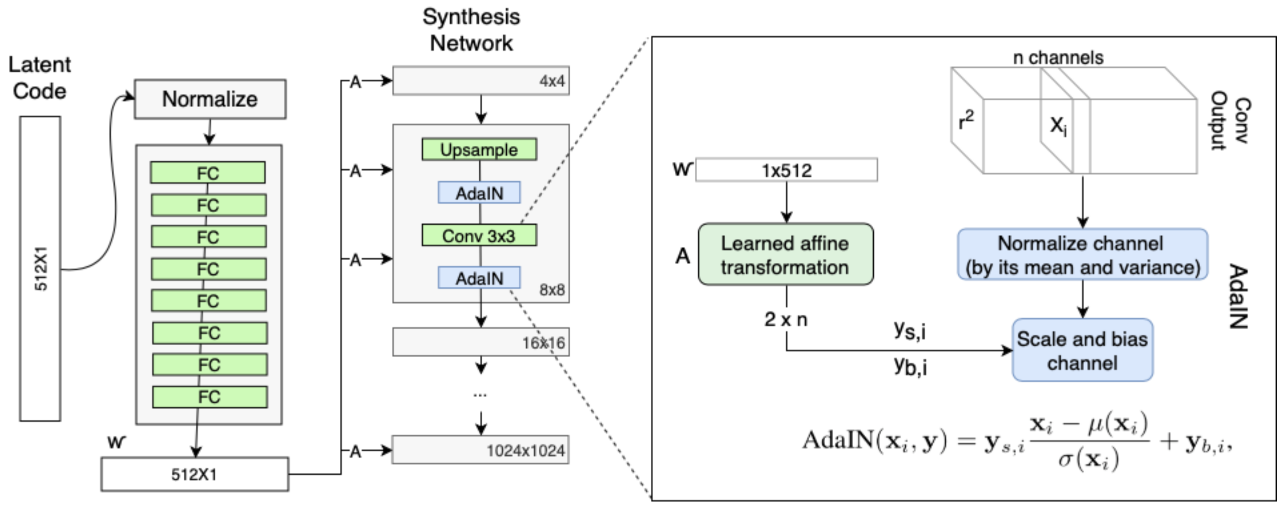


Q: can we control these styles?

1. Coarse - resolution of up to 82 - affects pose, general hair style, face shape, etc
2. Middle - resolution of 162 to 322 - affects finer facial features, hair style, eyes open/closed, etc.
3. Fine - resolution of 642 to 10242 - affects color scheme (eye, hair and skin) and micro features.

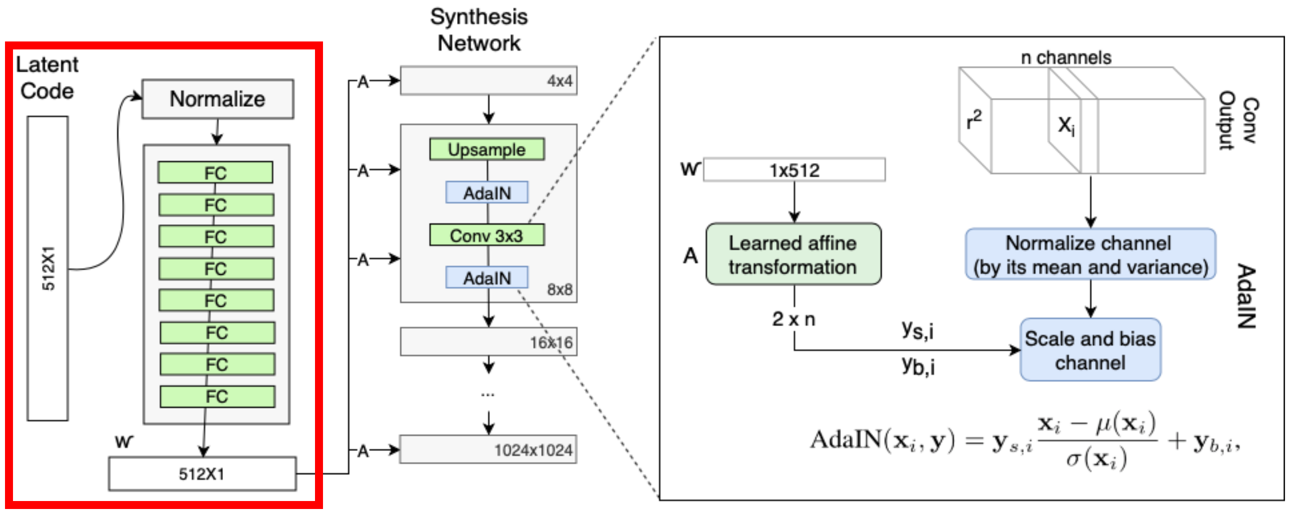
Karras, Tero, Samuli Laine, and Timo Aila. "A style-based generator architecture for generative adversarial networks." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4401-4410. 2019. https://arxiv.org/pdf/1812.04948.pdf
https://towardsdatascience.com/explained-a-style-based-generator-architecture-for-gans-generating-and-tuning-realistic-6cb2be0f431

StyleGAN



Karras, Tero, Samuli Laine, and Timo Aila. "A style-based generator architecture for generative adversarial networks." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4401-4410. 2019. https://arxiv.org/pdf/1812.04948.pdf
https://towardsdatascience.com/explained-a-style-based-generator-architecture-for-gans-generating-and-tuning-realistic-6cb2be0f431

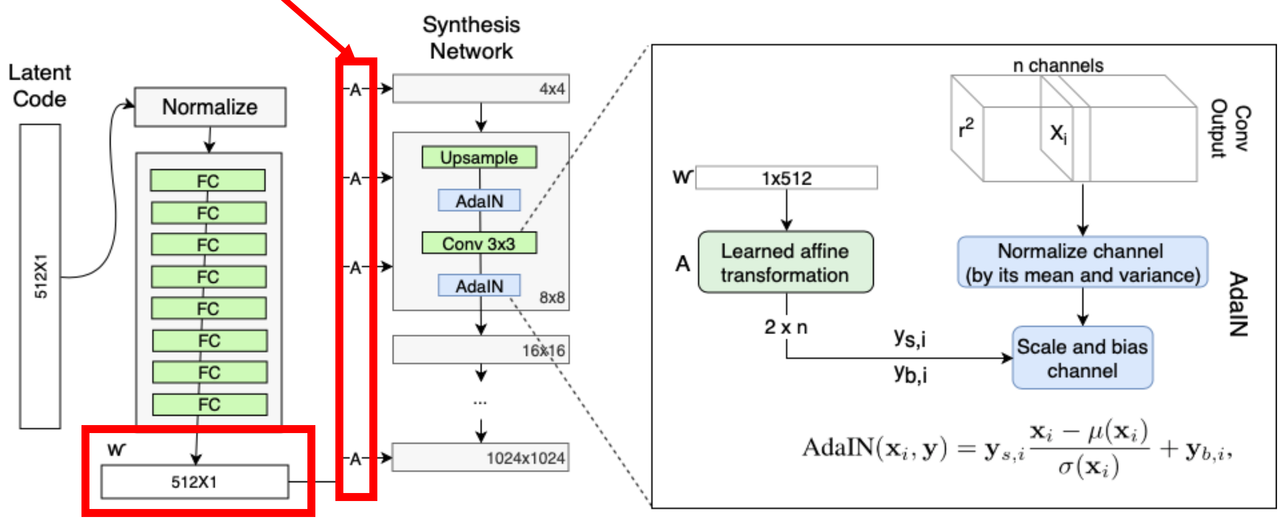
StyleGAN



Karras, Tero, Samuli Laine, and Timo Aila. "A style-based generator architecture for generative adversarial networks." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4401-4410. 2019. https://arxiv.org/pdf/1812.04948.pdf
https://towardsdatascience.com/explained-a-style-based-generator-architecture-for-gans-generating-and-tuning-realistic-6cb2be0f431

StyleGAN

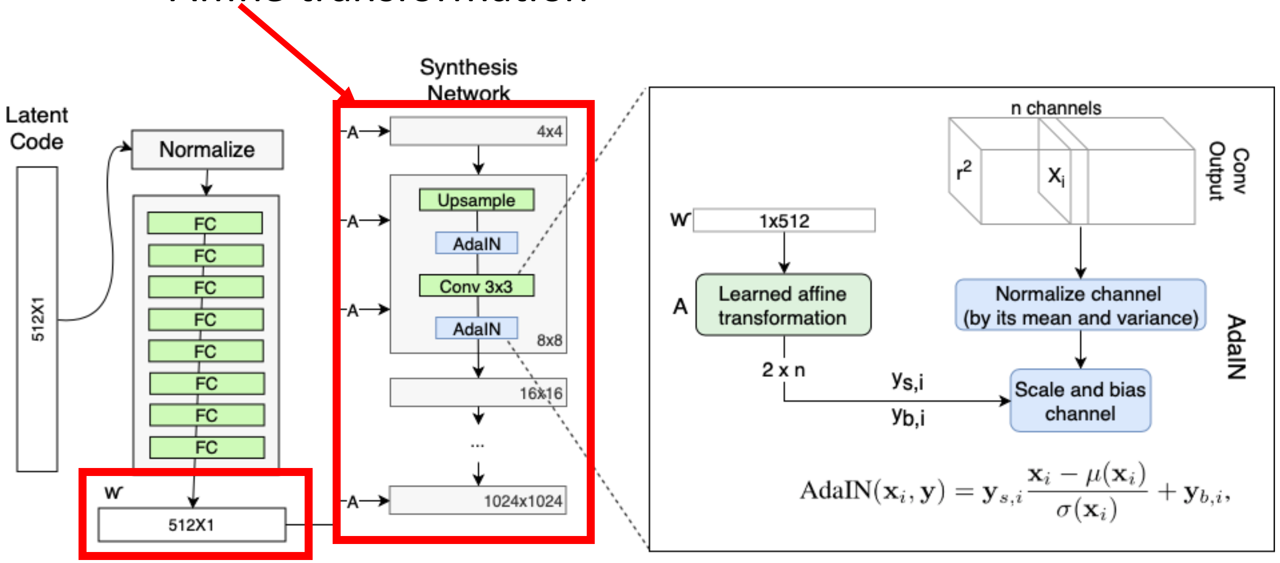
Affine transformation



Karras, Tero, Samuli Laine, and Timo Aila. "A style-based generator architecture for generative adversarial networks." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4401-4410. 2019. https://arxiv.org/pdf/1812.04948.pdf
https://towardsdatascience.com/explained-a-style-based-generator-architecture-for-gans-generating-and-tuning-realistic-6cb2be0f431

StyleGAN

Affine transformation

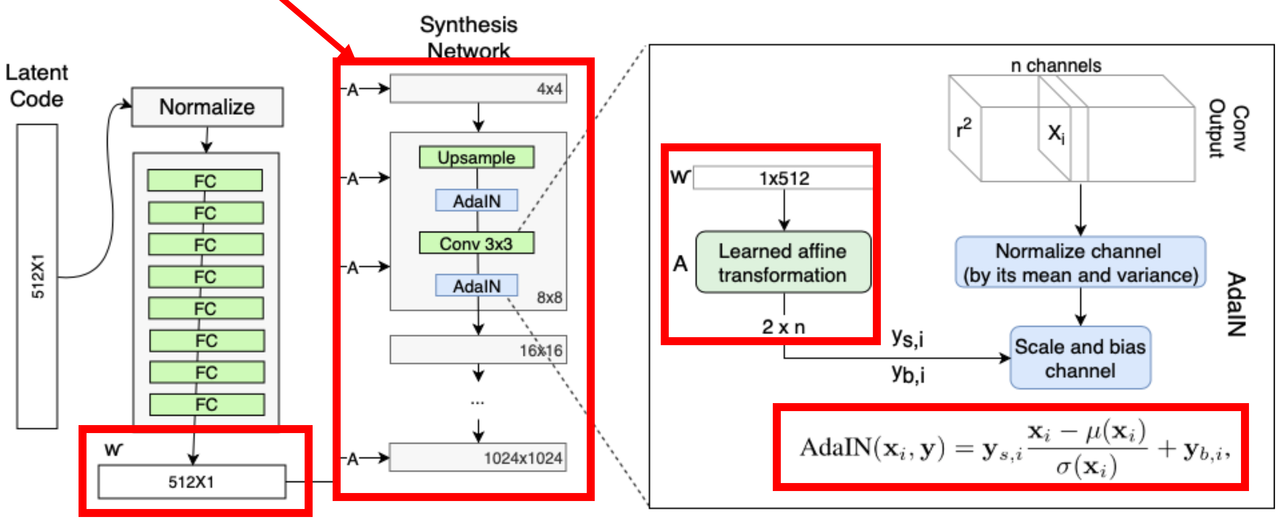


Karras, Tero, Samuli Laine, and Timo Aila. "A style-based generator architecture for generative adversarial networks." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4401-4410. 2019. https://arxiv.org/pdf/1812.04948.pdf

https://towardsdatascience.com/explained-a-style-based-generator-architecture-for-gans-generating-and-tuning-realistic-6cb2be0f431

StyleGAN

Affine transformation

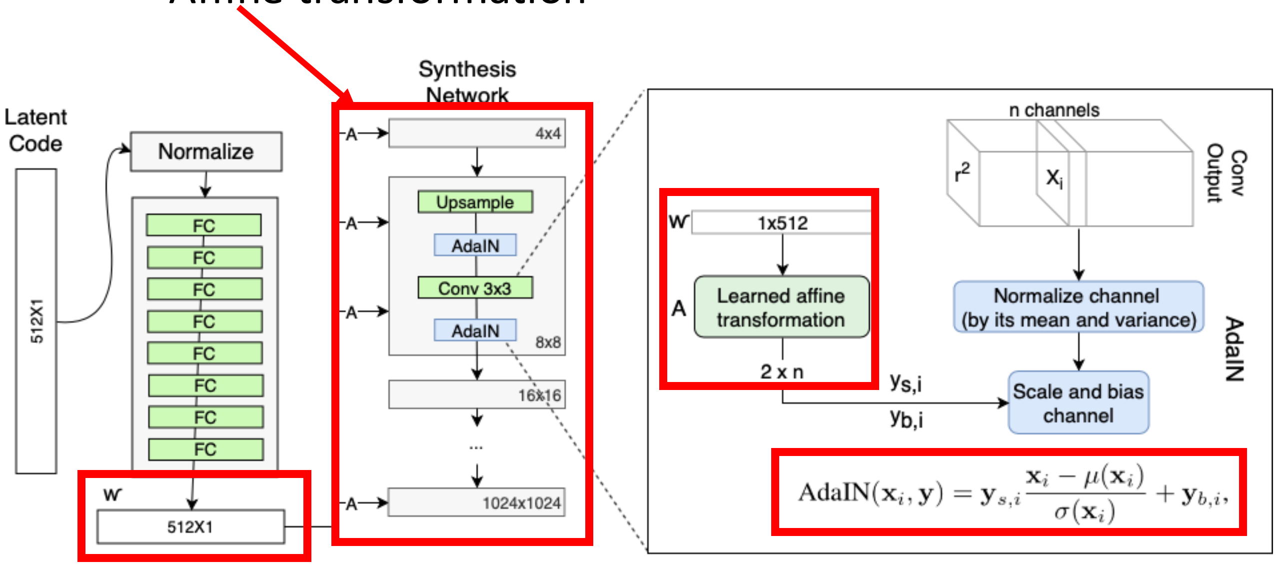


Karras, Tero, Samuli Laine, and Timo Aila. "A style-based generator architecture for generative adversarial networks." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4401-4410. 2019. https://arxiv.org/pdf/1812.04948.pdf

https://towardsdatascience.com/explained-a-style-based-generator-architecture-for-gans-generating-and-tuning-realistic-6cb2be0f431

StyleGAN

Affine transformation



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

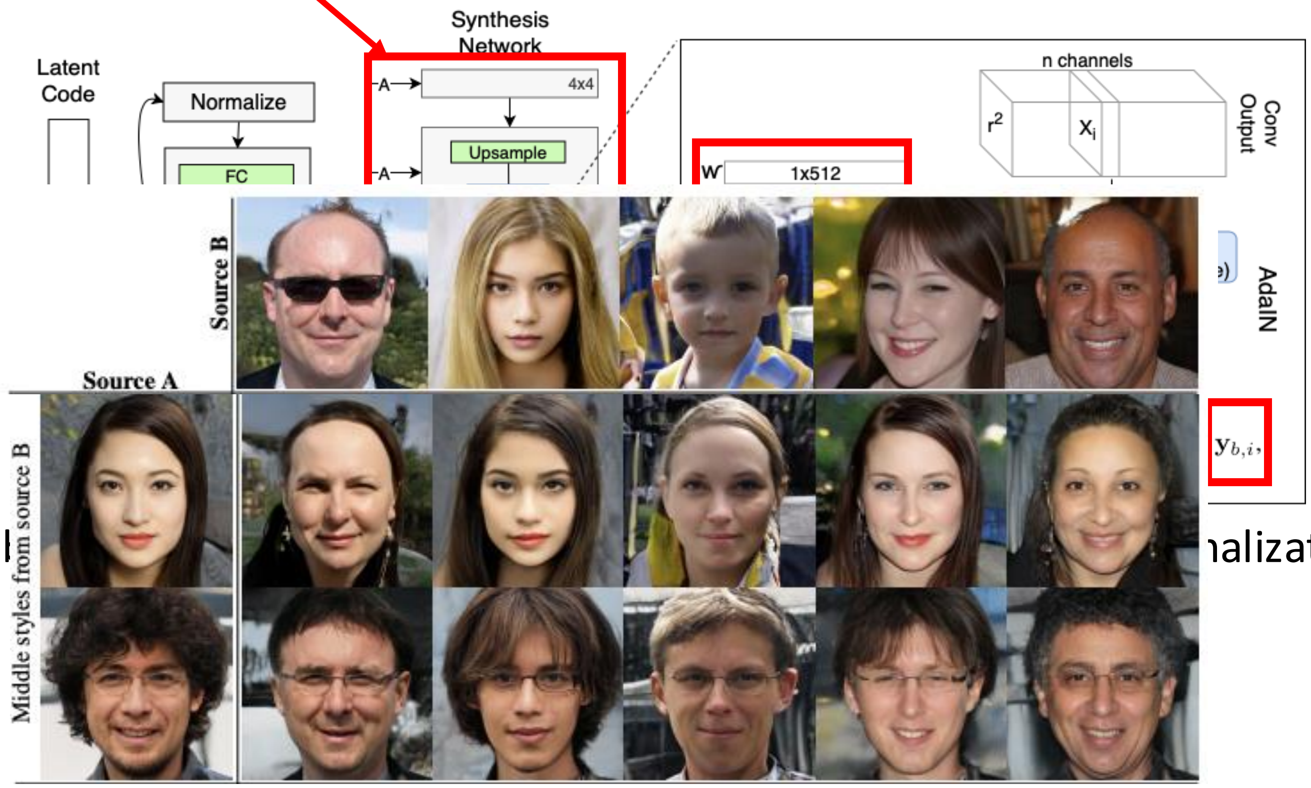
ADAptive Instance Normalization

Karras, Tero, Samuli Laine, and Timo Aila. "A style-based generator architecture for generative adversarial networks." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4401-4410. 2019. https://arxiv.org/pdf/1812.04948.pdf

https://towardsdatascience.com/explained-a-style-based-generator-architecture-for-gans-generating-and-tuning-realistic-6cb2be0f431

StyleGAN

Affine transformation

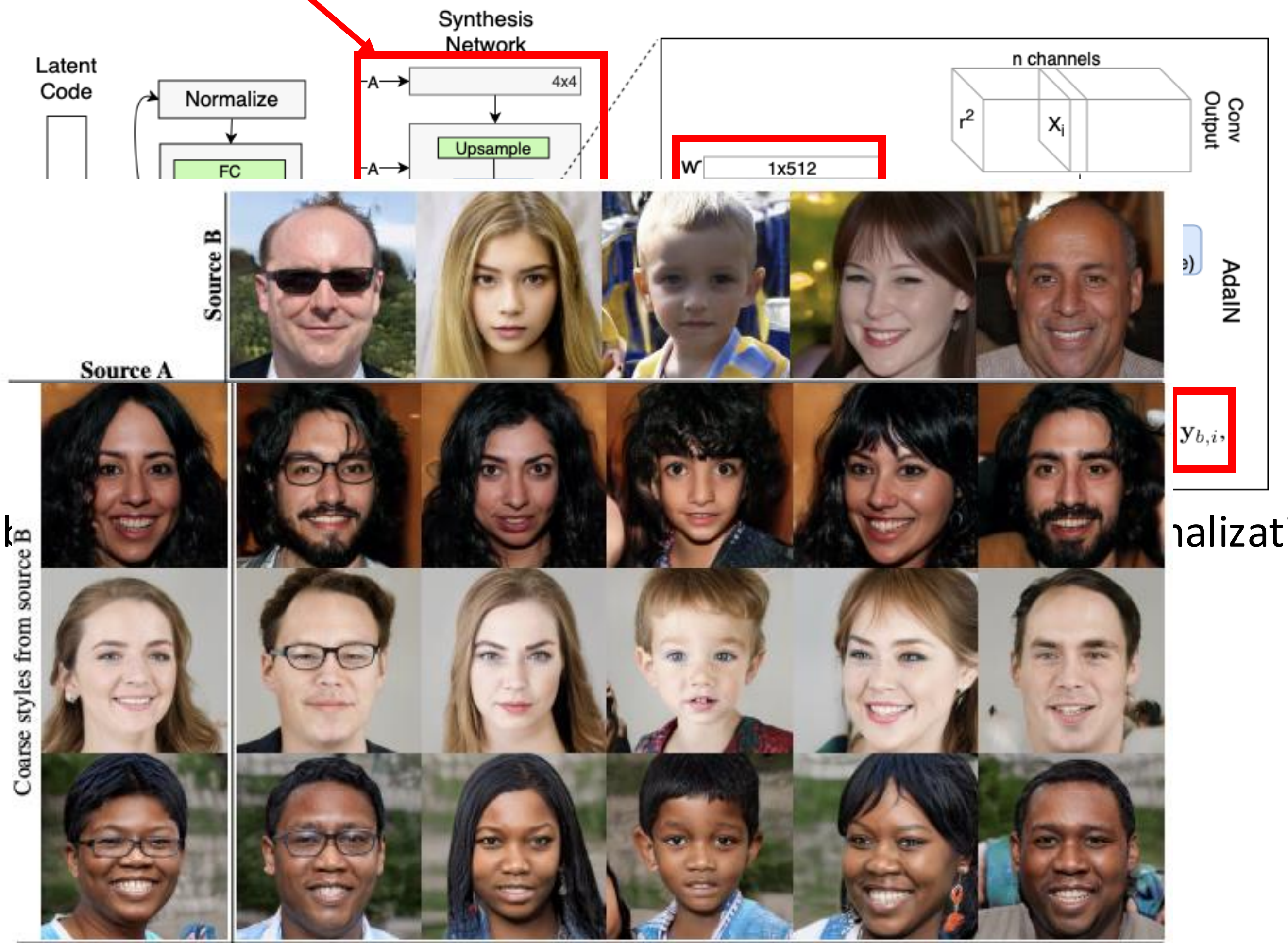


Style mixing: what al ıalization
recourses at various

Karras, Tero, Samuli Laine, and Timo Aila. "A style-based generator architecture for generative adversarial networks." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4401-4410. 2019. https://arxiv.org/pdf/1812.04948.pdf
https://towardsdatascience.com/explained-a-style-based-generator-architecture-for-gans-generating-and-tuning-realistic-6cb2be0f431

StyleGAN

Affine transformation

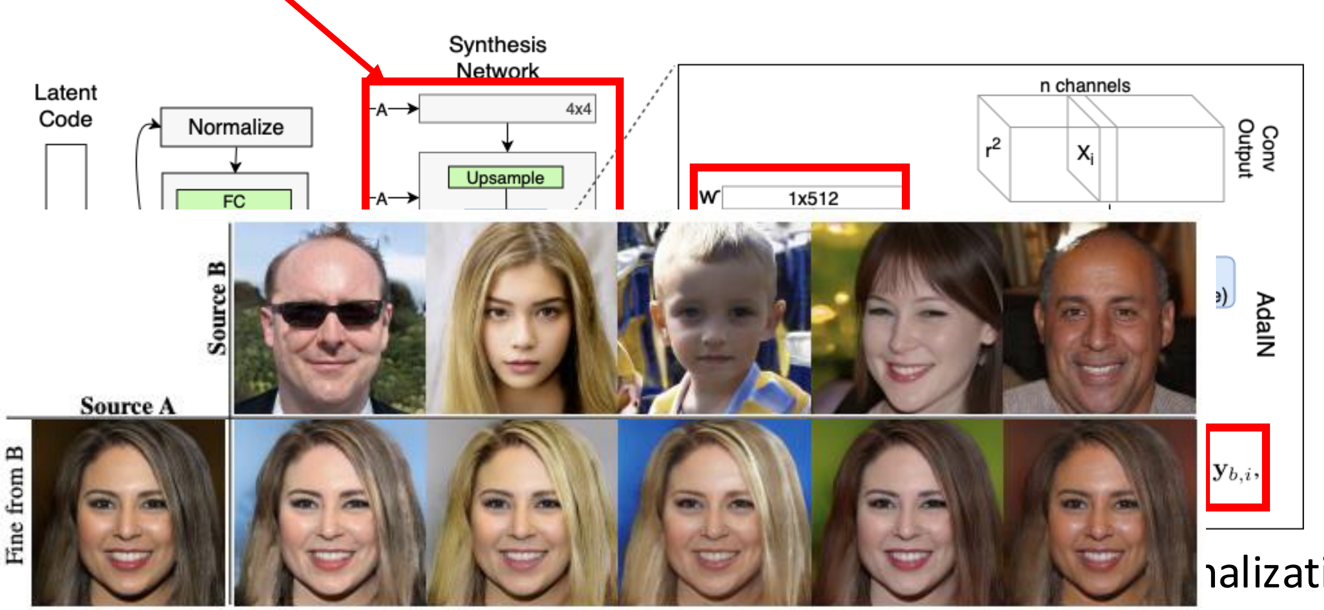


Style mixing: what al nalization
recourses at various

https://towardsdatascience.com/explained-a-style-based-generator-architecture-for-gans-generating-and-tuning-realistic-6cb2be0f431

StyleGAN

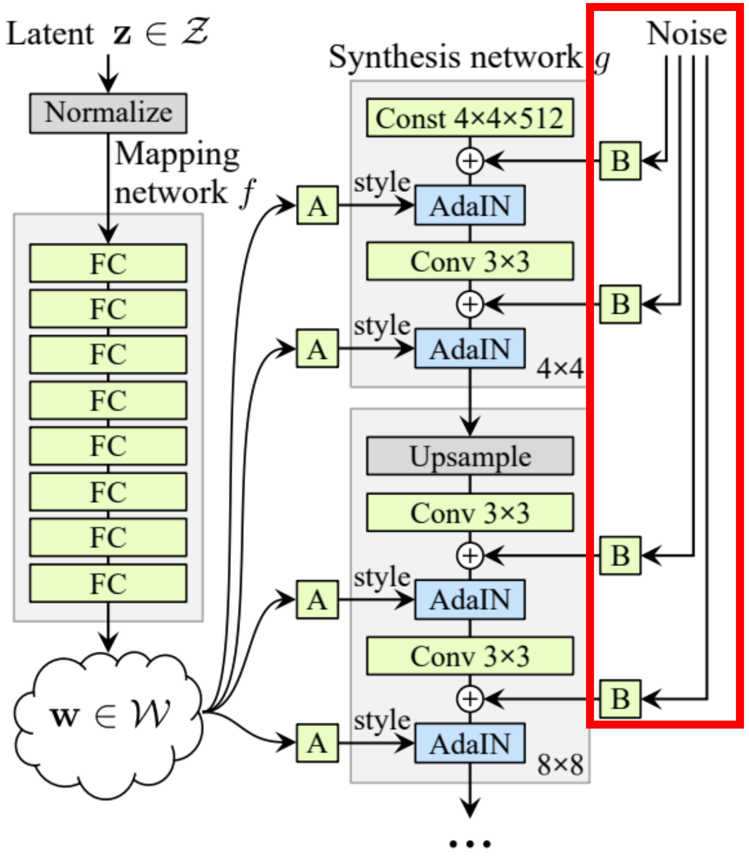
Affine transformation



Style mixing: what a ... ıalization
recourses at various layers?

https://towardsdatascience.com/explained-a-style-based-generator-architecture-for-gans-generating-and-tuning-realistic-6cb2be0f431

StyleGAN



(b) Style-based generator

Stochastic variation

StyleGAN


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

variation

(b) Style-based generator