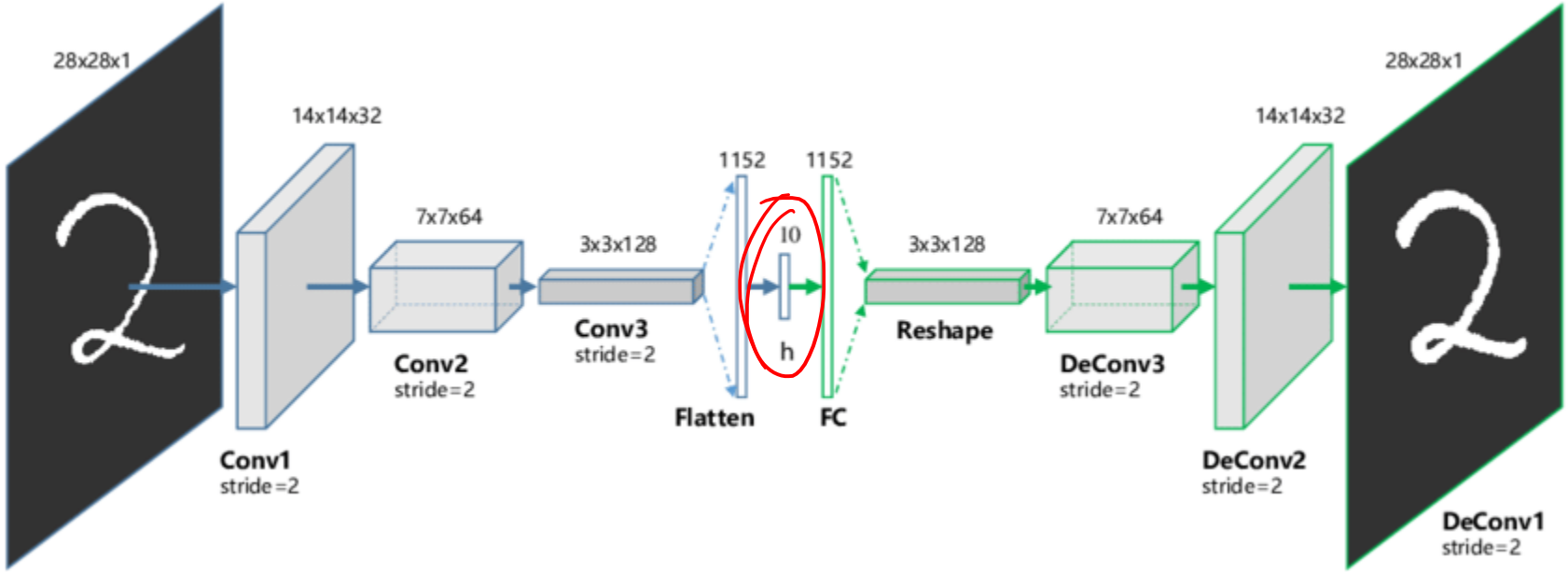
Generative Adversarial Network (GAN)

Tuan Nguyen - AI4E

Outline

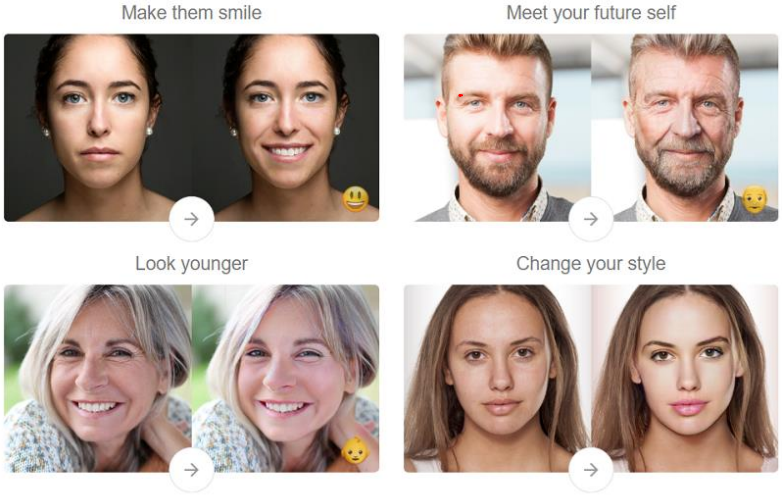
- Autoencoder review
- What is GAN?
- GAN model
- GAN loss function
- Image to image translation
- GAN application

Autoencoder



GAN introduction

- Introduced by J. Goodfellow in 2014
- Yann LeCun, VP and Chief AI Scientist, Facebook talked about GAN: “The most interesting idea in the last 10 years in Machine Learning”



What is GAN?

GAN

generative models

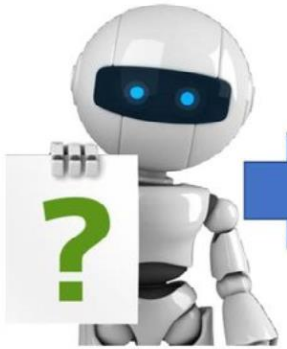
VAE

1

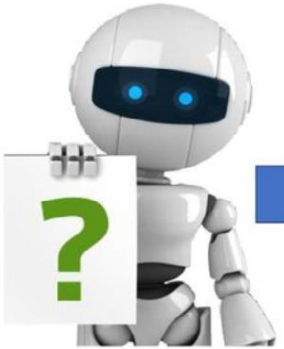
2

→ 3

Generation



Writing Poems?



Drawing?

GAN progressive

sinh ảnh mặt, ảnh thời nhiều.
img 2 img translation
deepfake.

DCGAN
11/2015

EBGAN-PT
9/2016

BEGAN
3/2017
128 × 128

Progressive GAN
10/2017
1024 x 1024

?

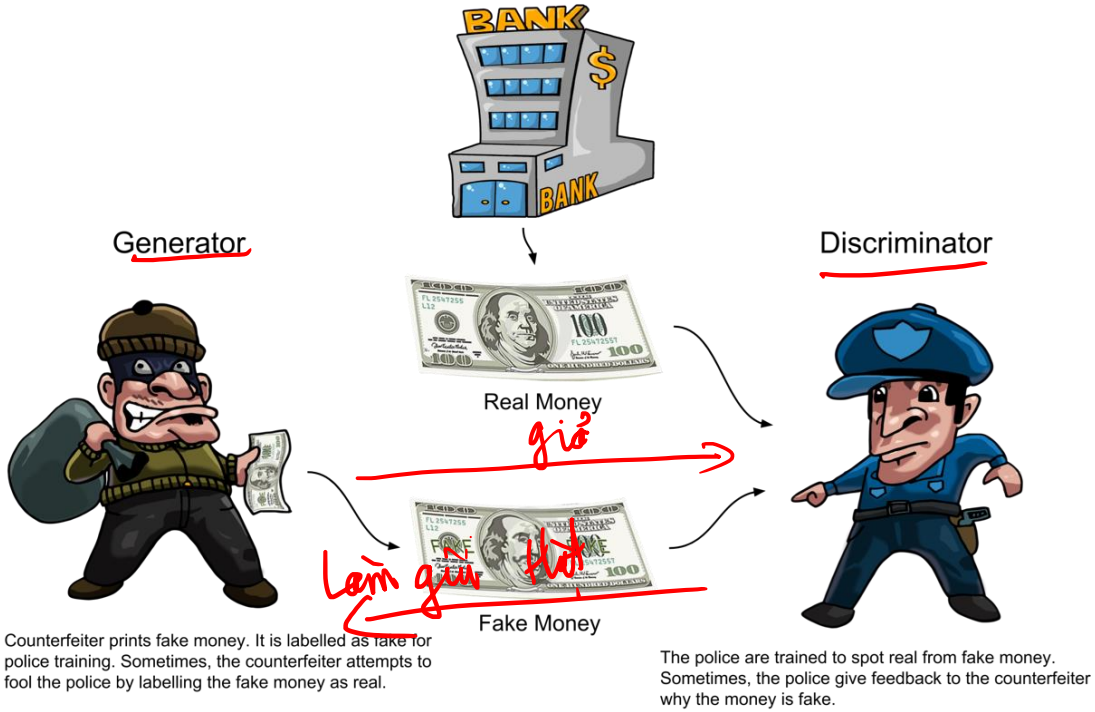
202x

Introduction to GAN

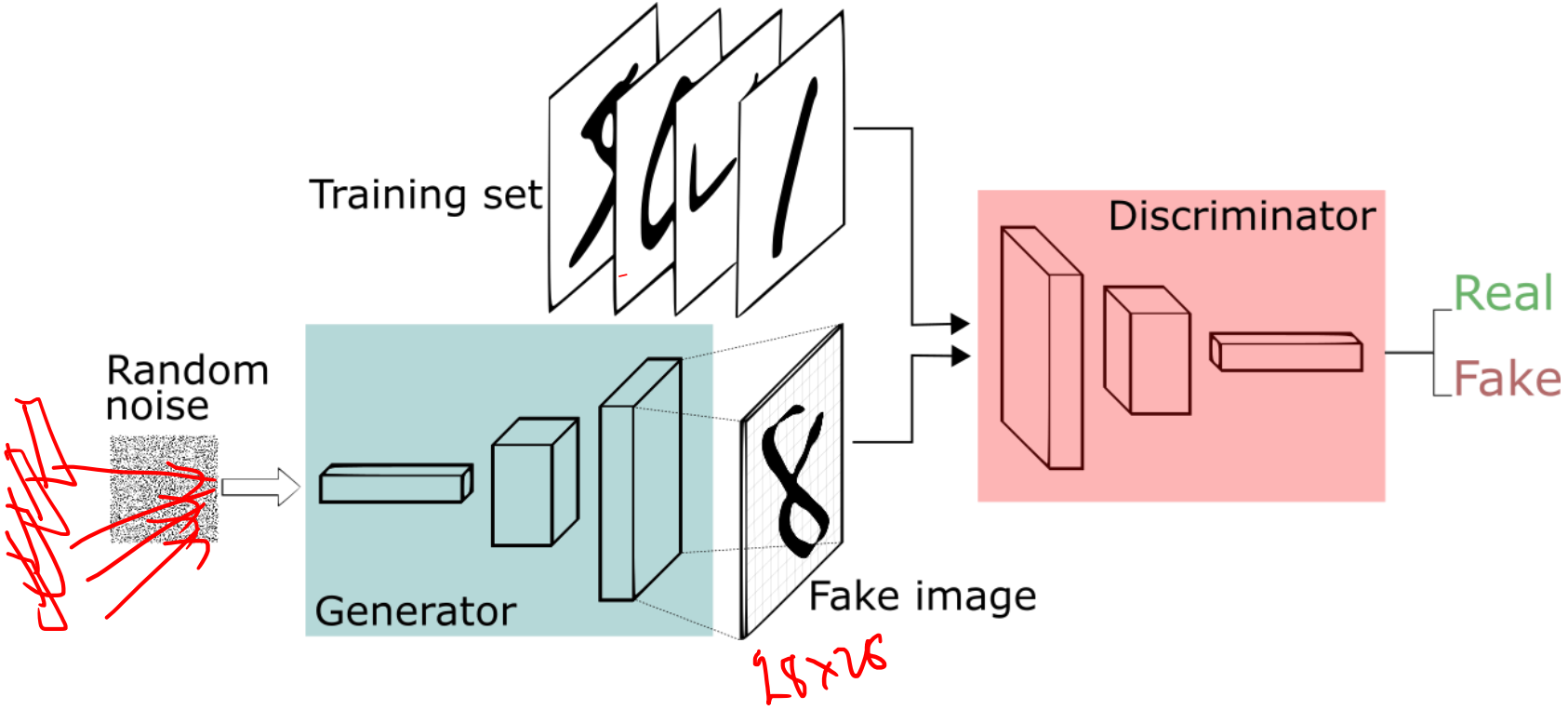
https://www.youtube.com/watch?v=mUfJOQKdtAk&t=52s

GAN component

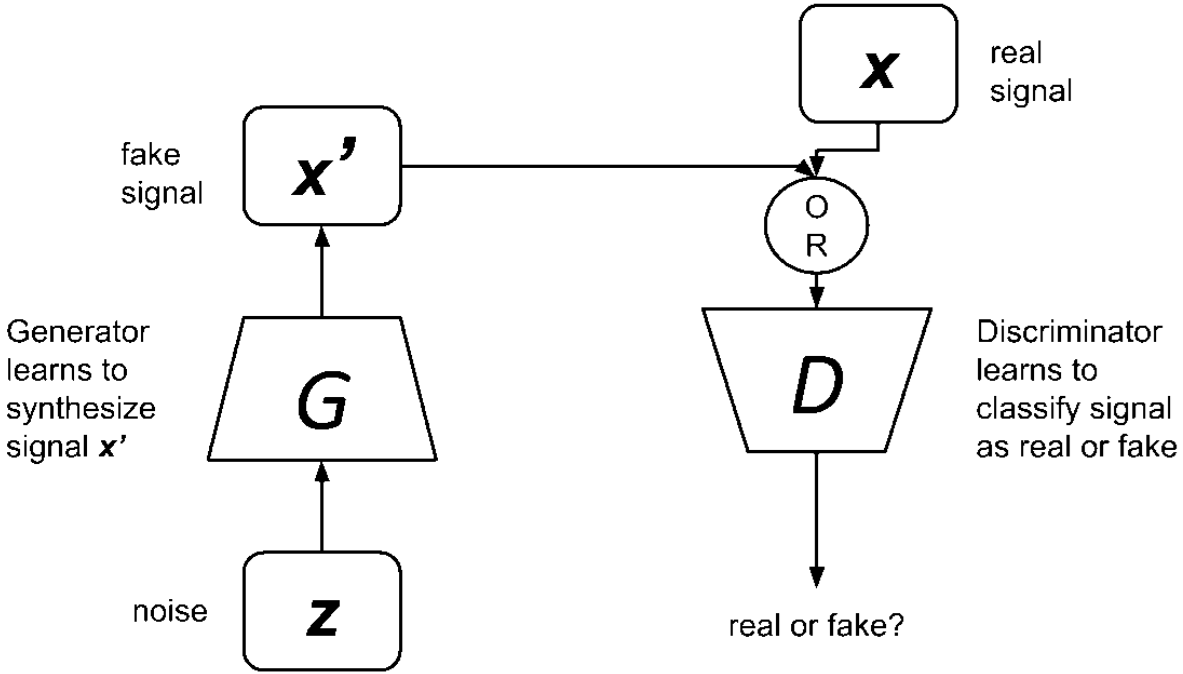
: Làm tiền giả giống với tiền thật ở ngân hàng.



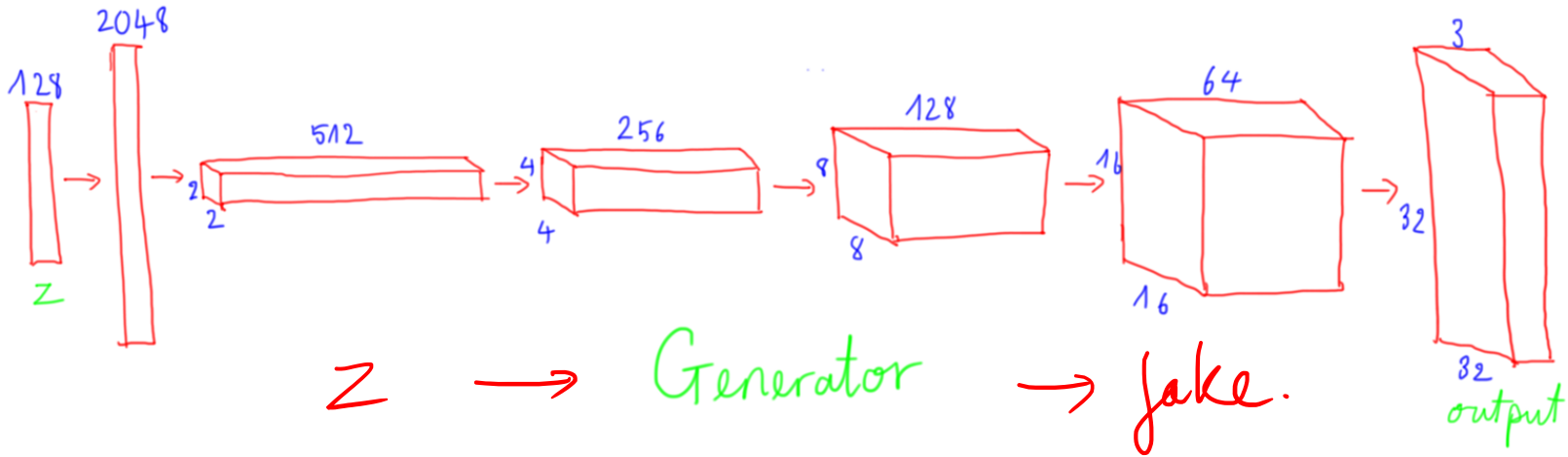
GAN model


Training set
Discriminator
Real
Fake
Random noise
Generator
Fake image
28x28

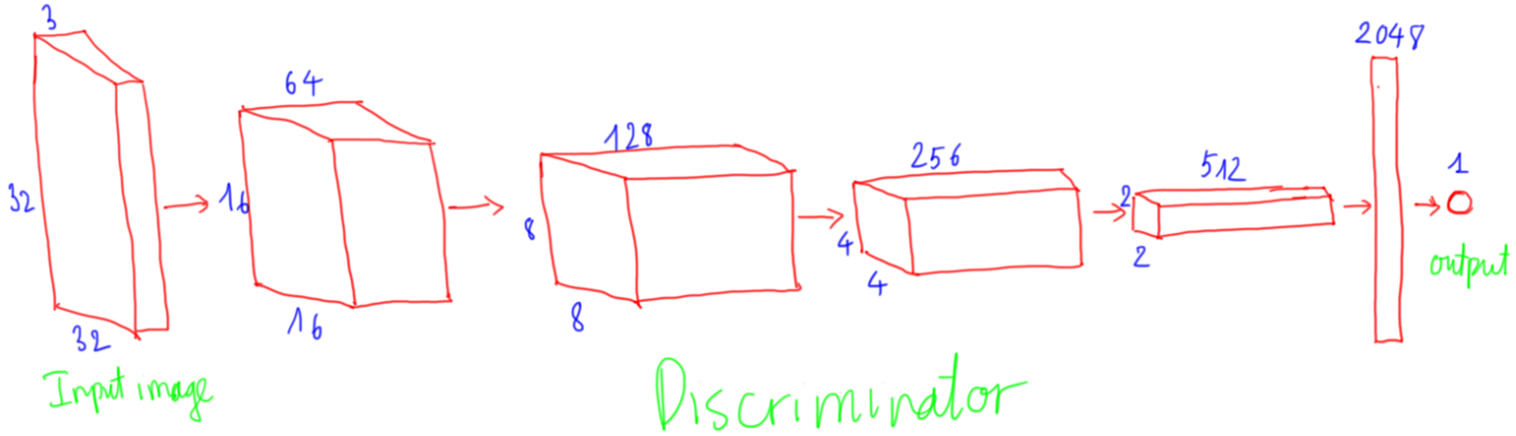
GAN model


x
real
signal
fake
signal
x'
O
R
Generator
learns to
synthesize
signal x'
G
D
Discriminator
learns to
classify signal
as real or fake
noise
z
real or fake?

Generator


cifar-10.
128
z
2048
512
2
2
256
4
4
128
8
8
64
16
16
3
32
32
z → Generator → fake.
output

Discriminator

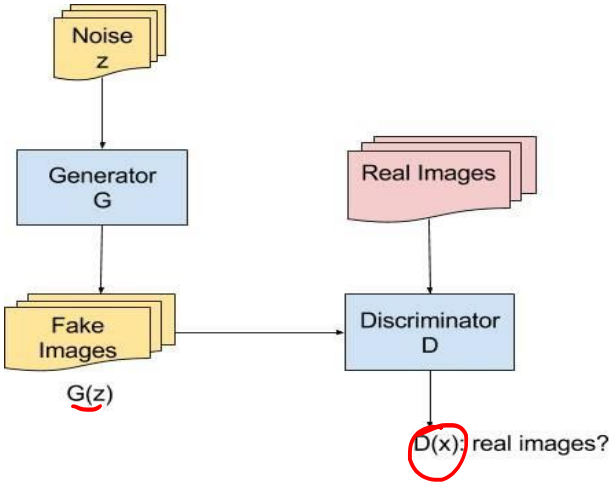


Loss function

fake : $g(z)$.
real : x
: $D(x)$. , $D(G(z))$

D : $D(x) \to 1$
$D(G(z)) \to 0$.

G : $D(G(z)) \to 1$.

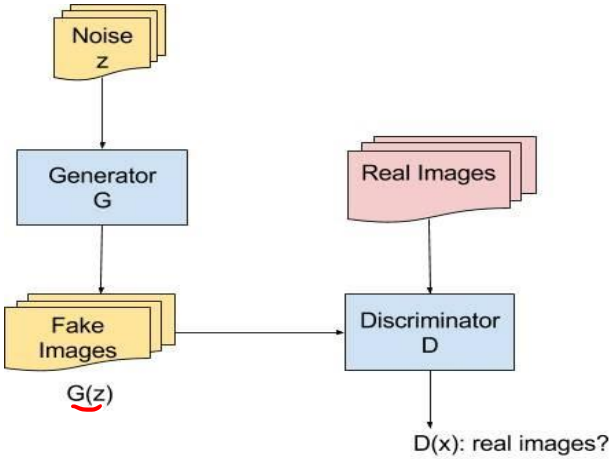


$0 < D(x) < 1$.

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

Loss function

fake : $g(z)$.
real : x
: $D(x)$. , $D(G(z))$



$D(G(z)) \to 1$

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

Training

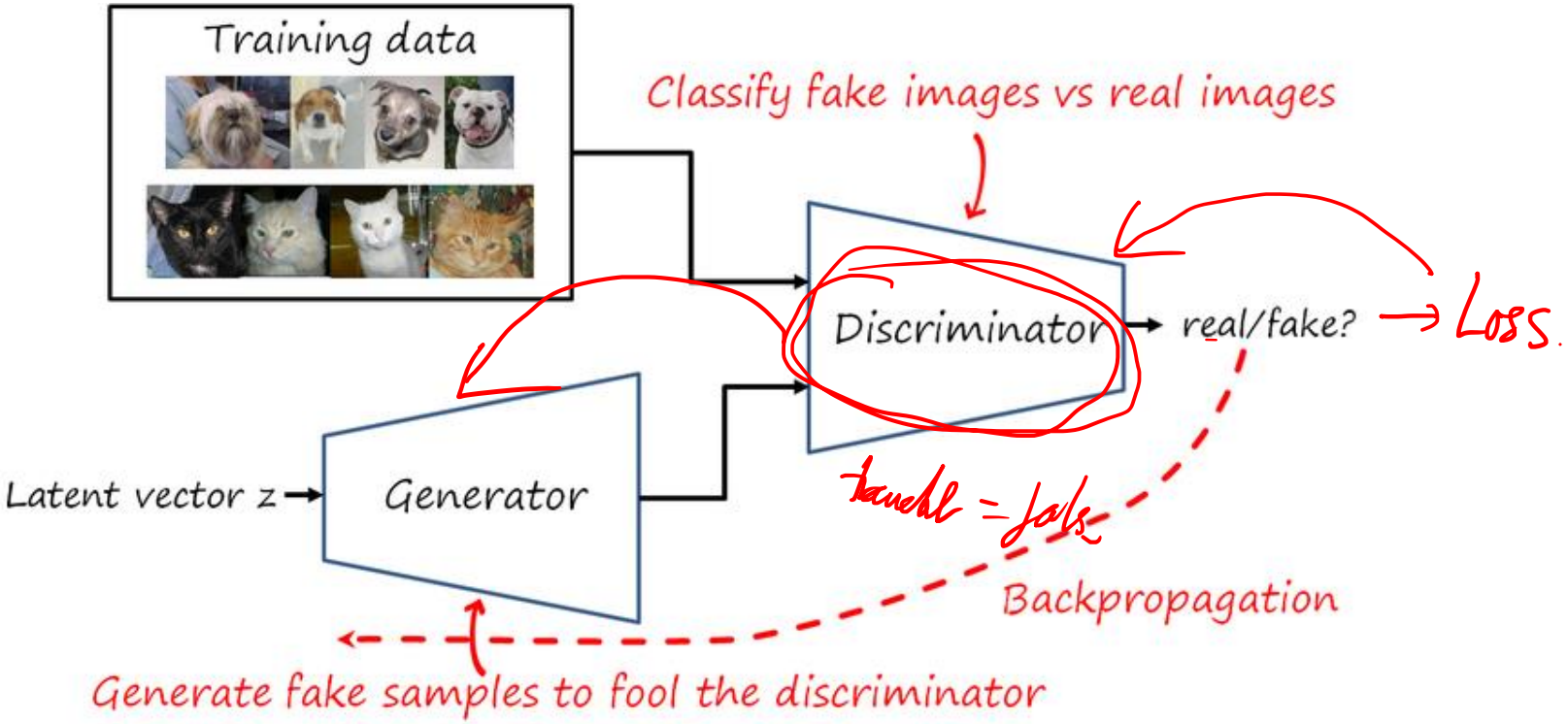
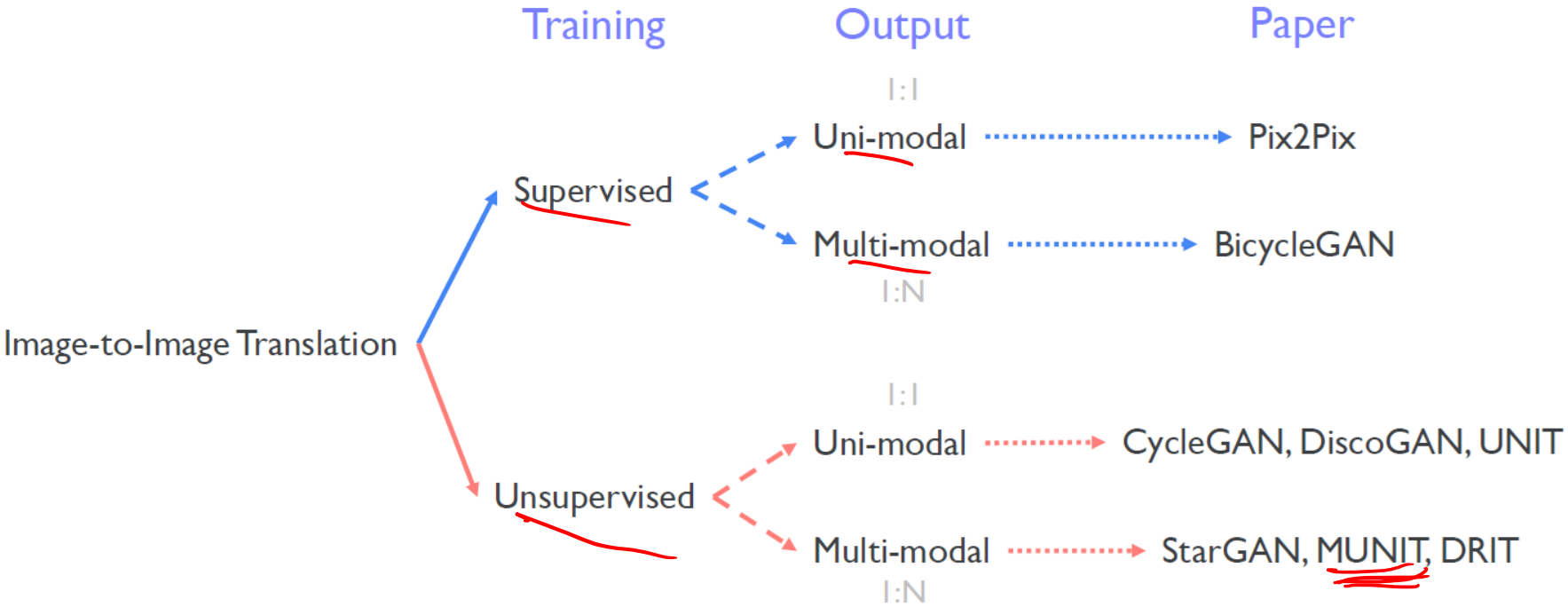

Training data
Classify fake images vs real images
Discriminator
real/fake?
Loss.
Latent vector z
Generator
Backpropagation
Generate fake samples to fool the discriminator

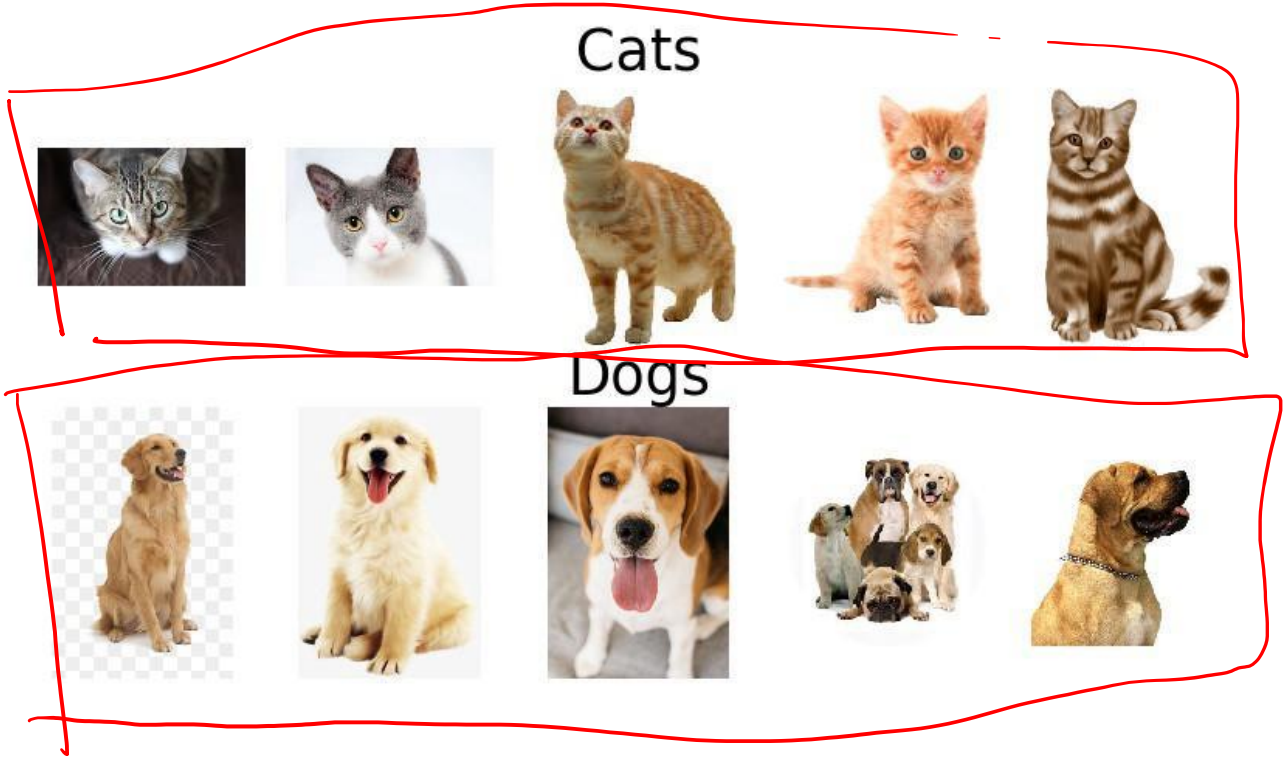
Image Translation

Image to image translation

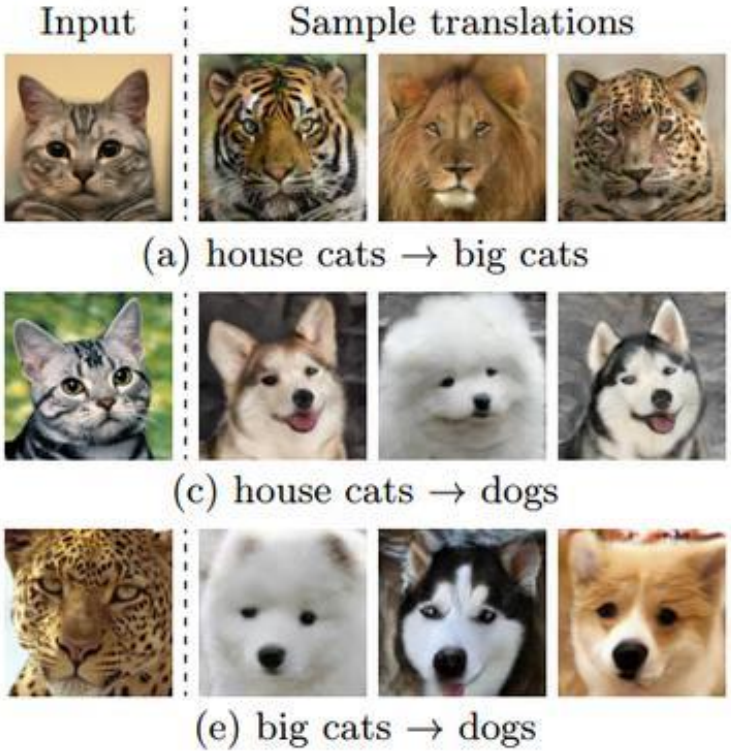


Supervised image to image translation

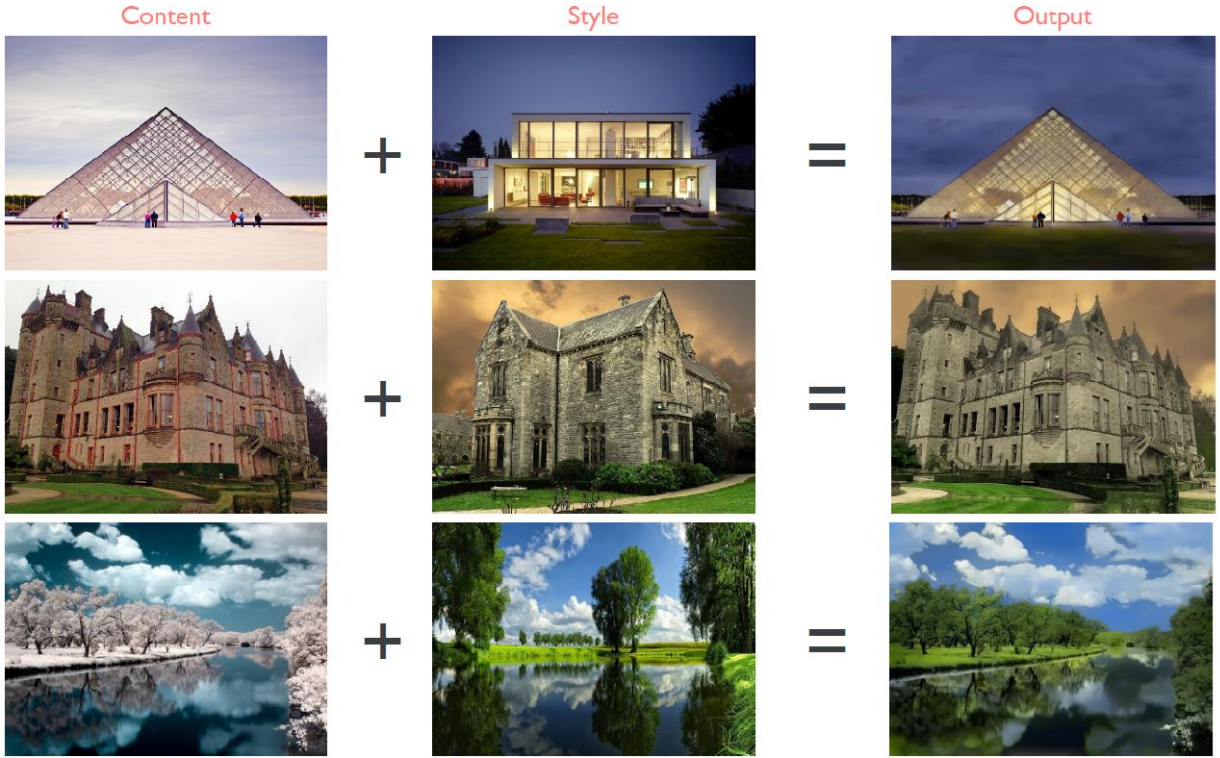
Unsupervised image to image translation



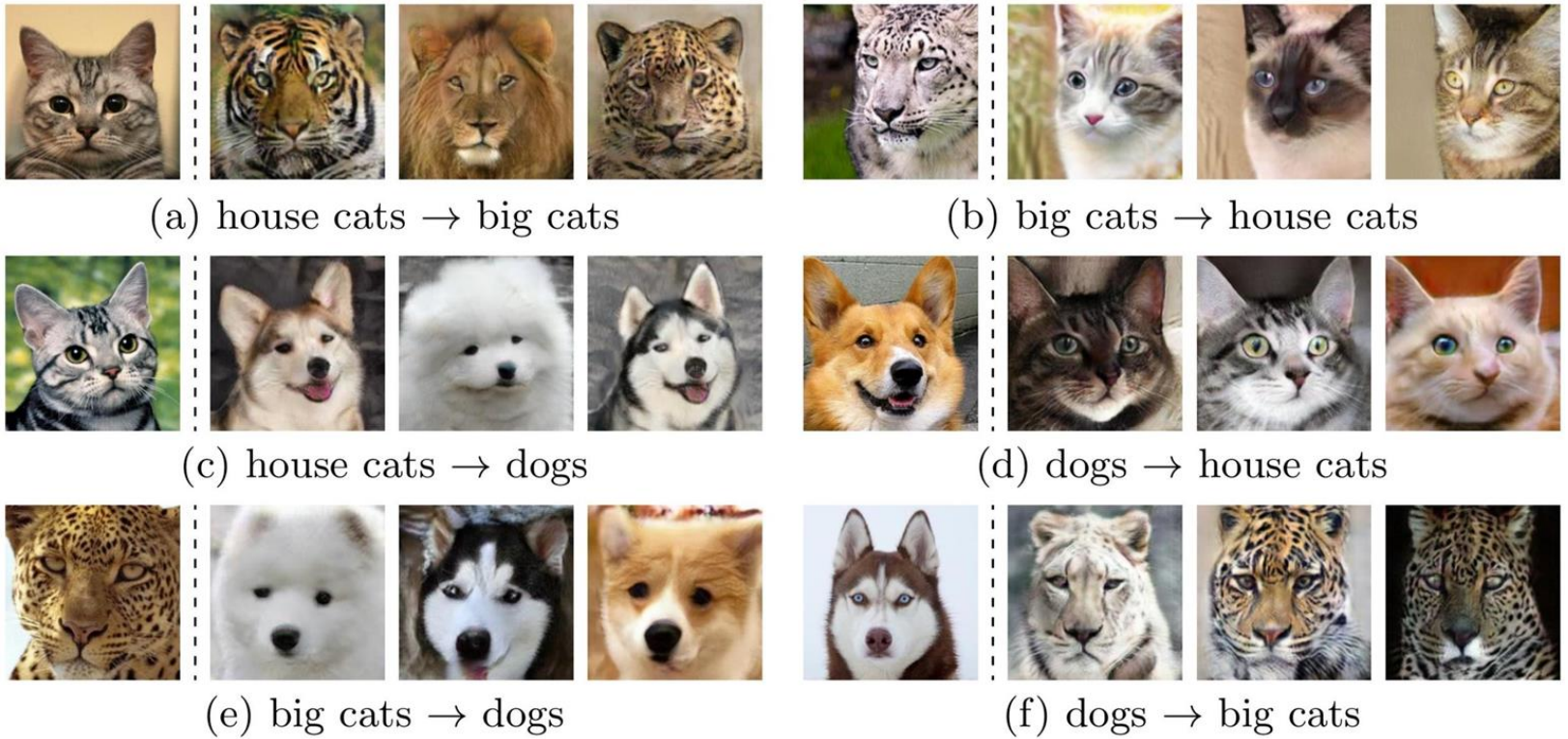
Multimodel


Input
Sample translations
(a) house cats → big cats
(c) house cats → dogs
(e) big cats → dogs

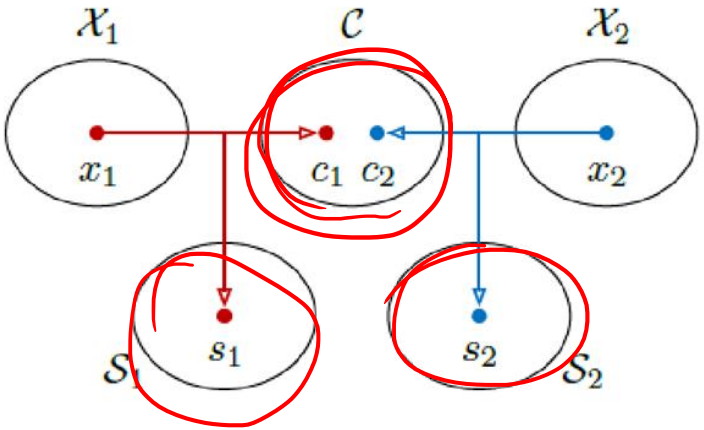
Style and content



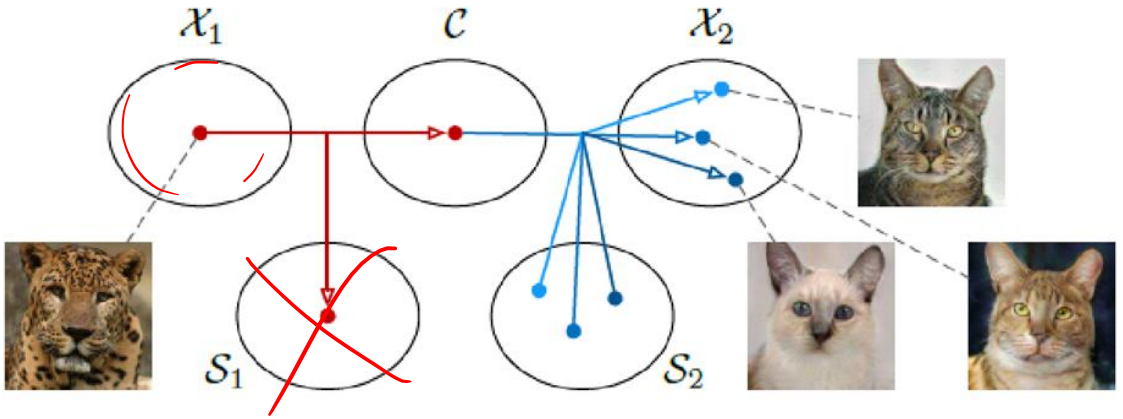
MUNIT


(a) house cats → big cats
(b) big cats → house cats
(c) house cats → dogs
(d) dogs → house cats
(e) big cats → dogs
(f) dogs → big cats

MUNIT

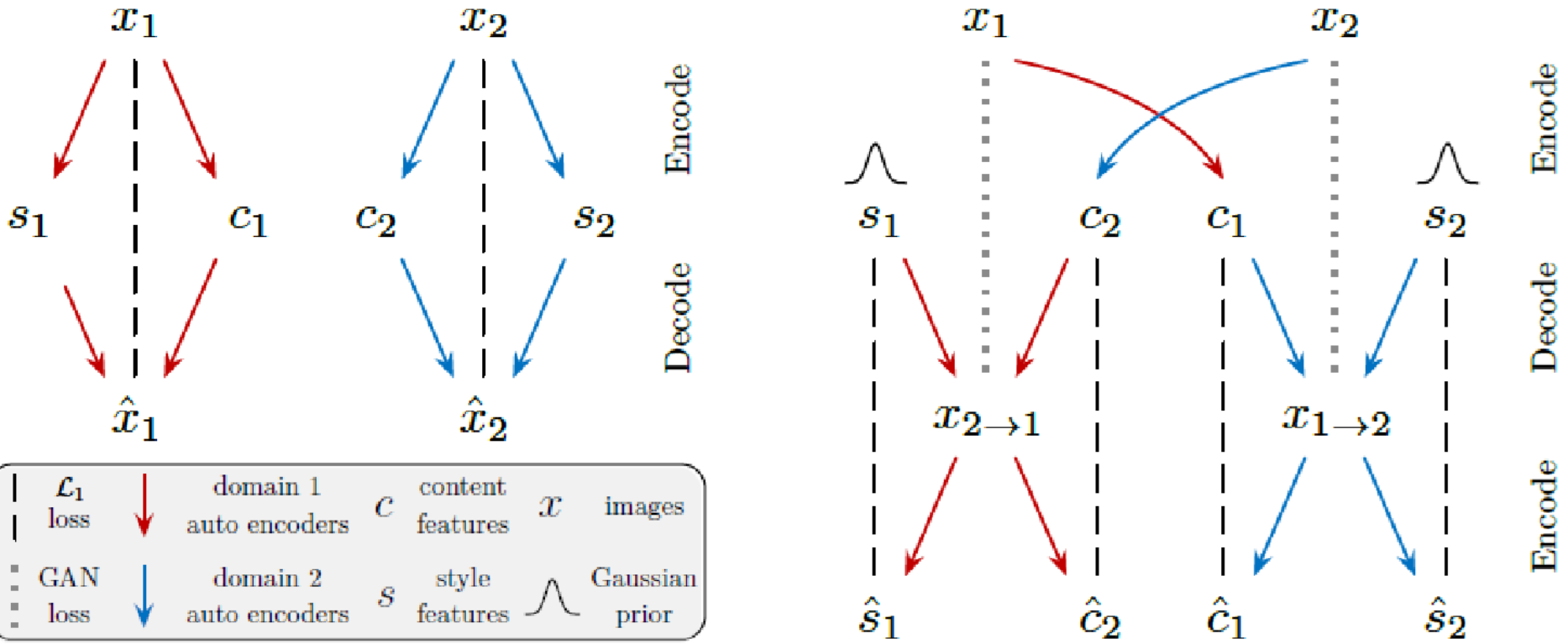


(a) Auto-encoding



(b) Translation

MUNIT


x_1
x_2
s_1
c_1
c_2
s_2
Encode
Decode
$\hat{x}_1$
$\hat{x}_2$
$\mathcal{L}_1$ loss
domain 1 auto encoders
c content features
x images
GAN loss
domain 2 auto encoders
s style features
Gaussian prior
$x_{2\to1}$
$x_{1\to2}$
$\hat{s}_1$
$\hat{c}_2$
$\hat{c}_1$
$\hat{s}_2$

Loss

- Image reconstruction

$$\mathcal{L}_{\text{recon}}^{x_1} = \mathbb{E}_{x_1 \sim p(x_1)}[||G_1(E_1^c(x_1), E_1^s(x_1)) - x_1||_1]$$

- Laten Reconstruction

$$\mathcal{L}_{\text{recon}}^{c_1} = \mathbb{E}_{c_1 \sim p(c_1), s_2 \sim q(s_2)}[||E_2^c(G_2(c_1, s_2)) - c_1||_1]$$

$$\mathcal{L}_{\text{recon}}^{s2} = \mathbb{E}_{c_1 \sim p(c_1), s_2 \sim q(s_2)}[||E_2^s(G_2(c_1, s_2)) - s_2||_1]$$

- GAN loss

$$\mathcal{L}_{\text{GAN}}^{x_2} = \mathbb{E}_{c_1 \sim p(c_1), s_2 \sim q(s_2)}[\log(1 - D_2(G_2(c_1, s_2)))] + \mathbb{E}_{x_2 \sim p(x_2)}[\log D_2(x_2)]$$

- Total loss

$$\min_{E_1, E_2, G_1, G_2} \max_{D_1, D_2} \mathcal{L}(E_1, E_2, G_1, G_2, D_1, D_2) = \mathcal{L}_{\text{GAN}}^{x_1} + \mathcal{L}_{\text{GAN}}^{x_2} +$$

$$\lambda_x(\mathcal{L}_{\text{recon}}^{x_1} + \mathcal{L}_{\text{recon}}^{x_2}) + \lambda_c(\mathcal{L}_{\text{recon}}^{c_1} + \mathcal{L}_{\text{recon}}^{c_2}) + \lambda_s(\mathcal{L}_{\text{recon}}^{s_1} + \mathcal{L}_{\text{recon}}^{s_2})$$

Model

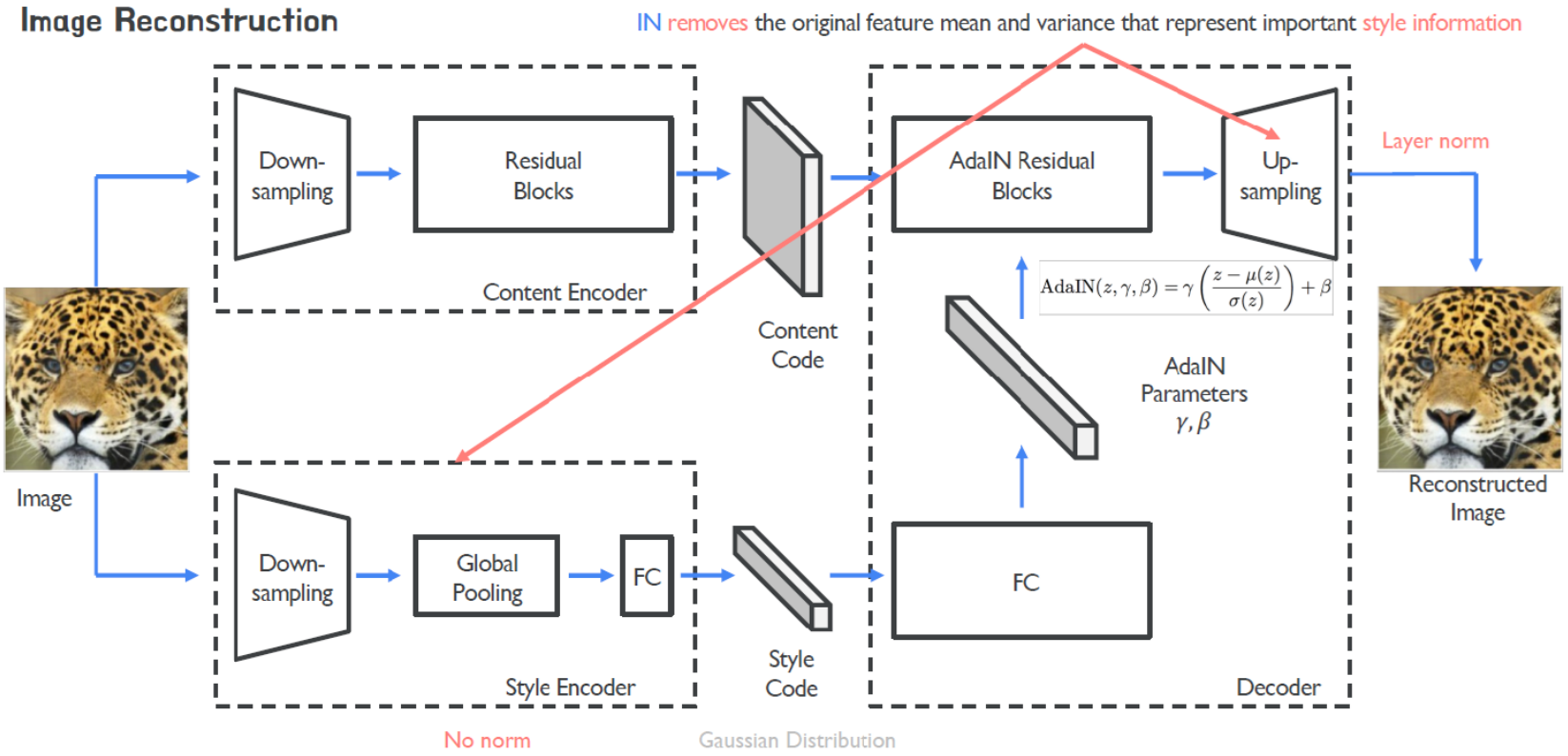
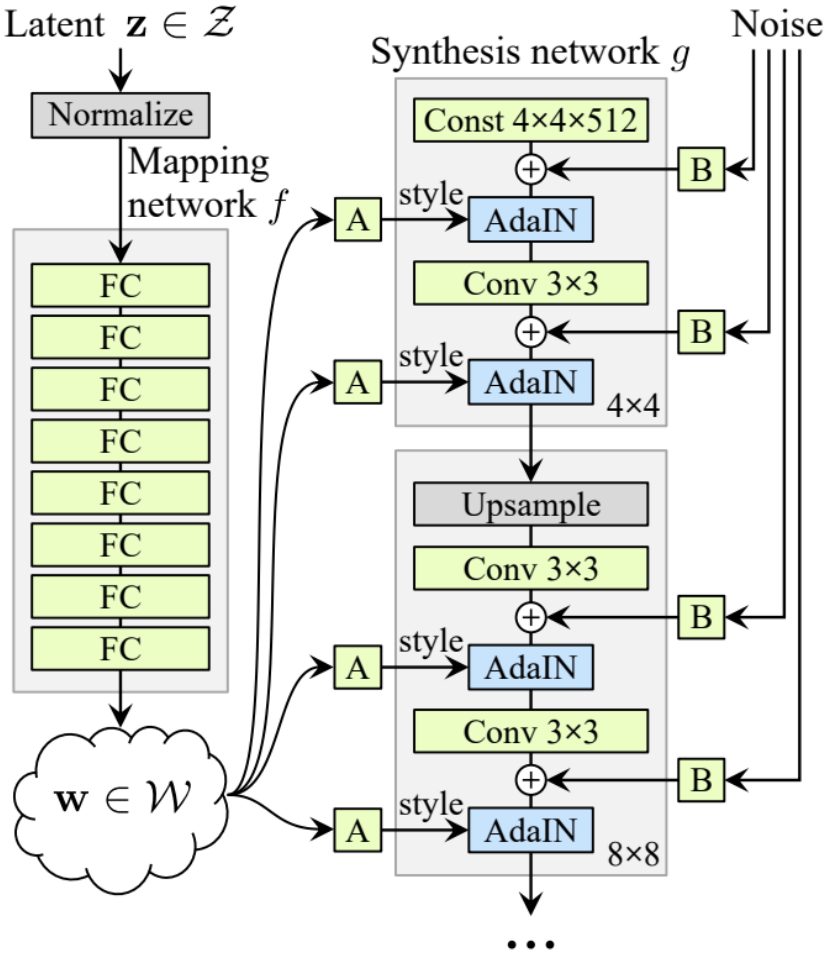


Image Reconstruction
IN removes the original feature mean and variance that represent important style information
Down-sampling
Residual Blocks
Content Encoder
Content Code
AdaIN Residual Blocks
Up-sampling
Layer norm
$\mathrm{AdaIN}(z,\gamma,\beta)=\gamma\left(\frac{z-\mu(z)}{\sigma(z)}\right)+\beta$
AdaIN Parameters γ,β
Image
Reconstructed Image
Down-sampling
Global Pooling
FC
Style Code
FC
Style Encoder
Decoder
No norm
Gaussian Distribution

AdaIN


Latent $\mathbf{z} \in \mathcal{Z}$
Normalize
Mapping network f
FC
FC
FC
FC
FC
FC
FC
FC
$\mathbf{w} \in \mathcal{W}$
Synthesis network g
Noise
Const 4×4×512
B
style
A
AdaIN
Conv 3×3
4×4
Upsample
8×8
...

Generation

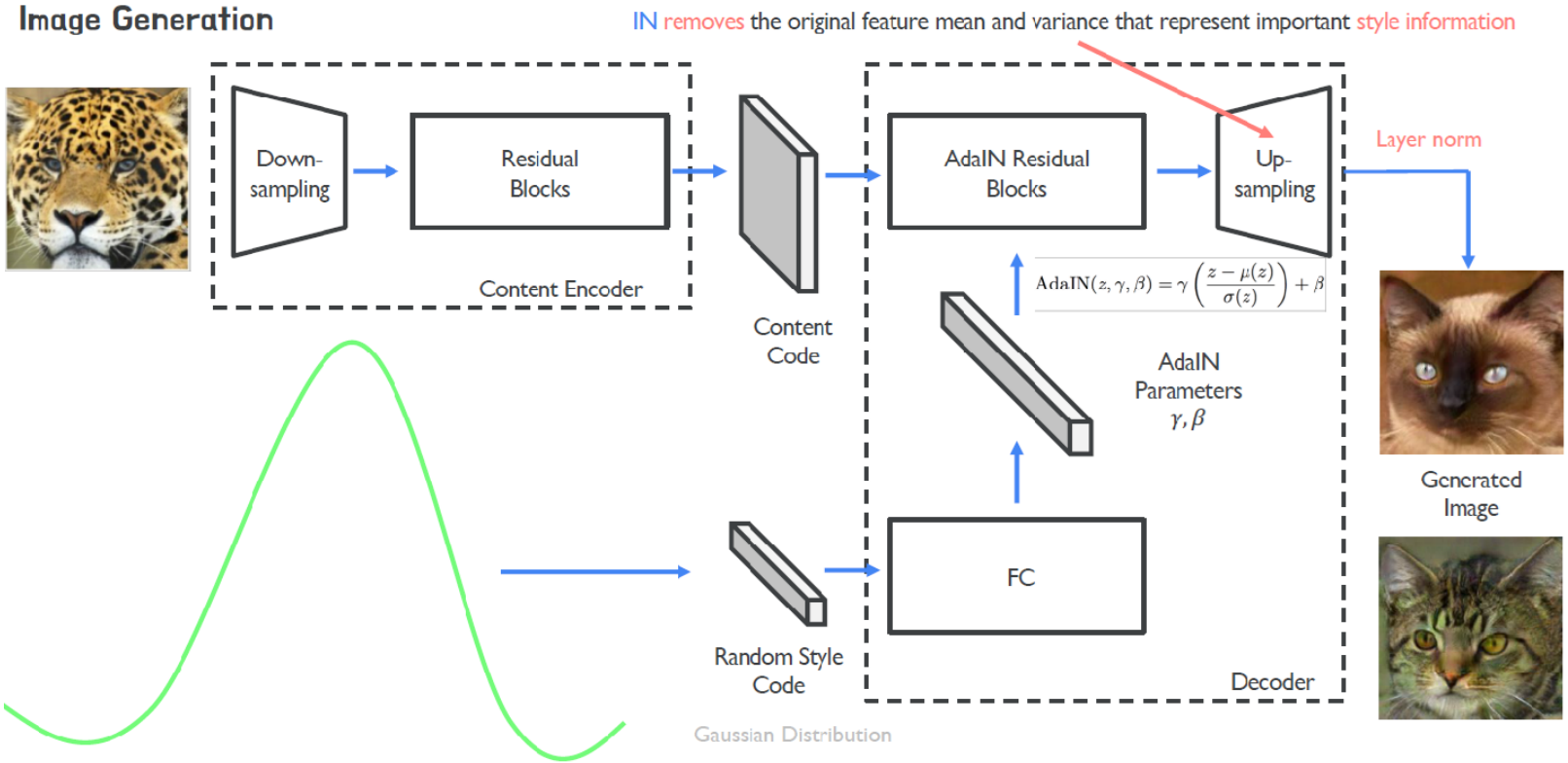


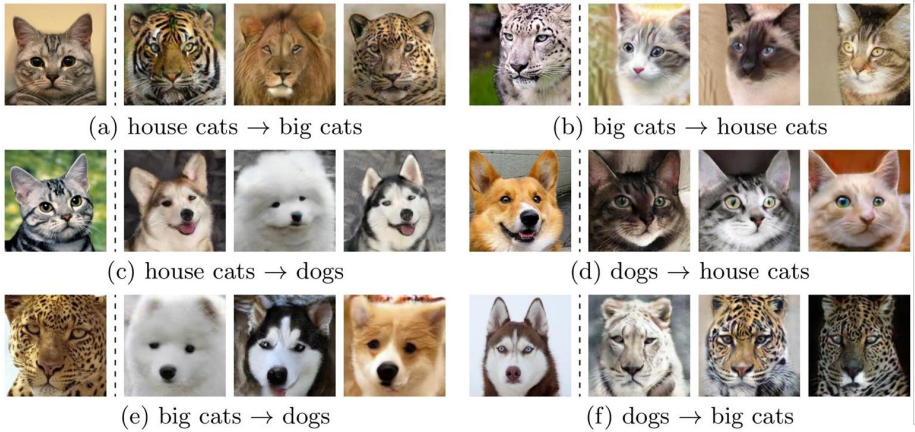
Image Generation
IN removes the original feature mean and variance that represent important style information
Down-sampling
Residual Blocks
Content Encoder
Content Code
AdaIN Residual Blocks
Up-sampling
Layer norm
$\text{AdaIN}(z, \gamma, \beta) = \gamma\left(\frac{z - \mu(z)}{\sigma(z)}\right) + \beta$
AdaIN Parameters γ, β
Generated Image
FC
Random Style Code
Decoder
Gaussian Distribution

Japanese to Korean

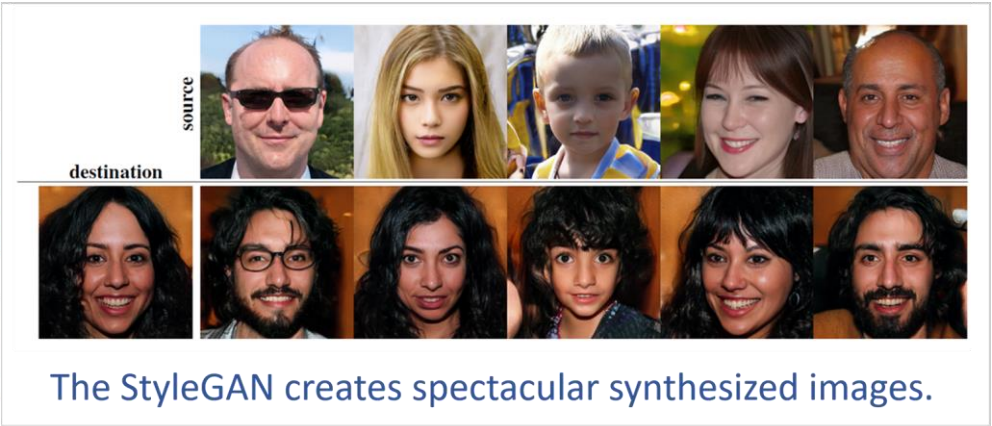
Korean to Japanese

Problems?

Image to image translation



High resolution images



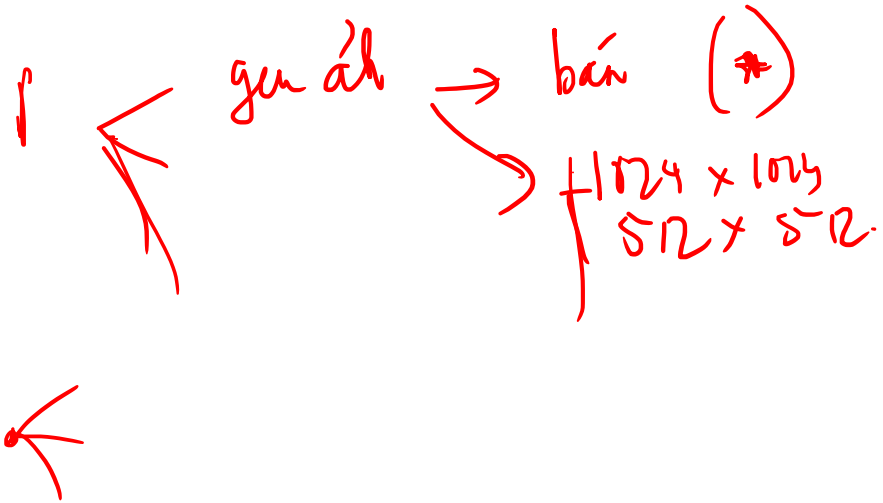
High resolution image to image translation

Japanese

Korean

Application of GANs

https://nttuan8.com/gioi-thieu-series-gan-generative-adversarial-networks/



Q&A


@LateNightSeth
THANK YOU!