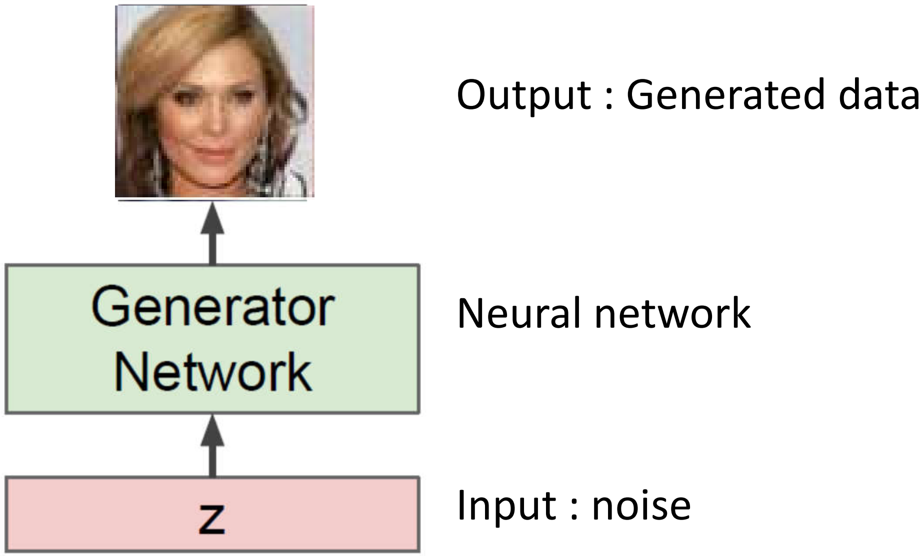
Introduction to the Generative Adversarial Networks (GAN)

Blaise Hanczar

Generative Adversarial Network (GAN)

- Neural network that generates data from noise that looks real
- Does not look for an explicit distribution of the real data
- One of the most popular methods in deep learning (2014)



Data Generation

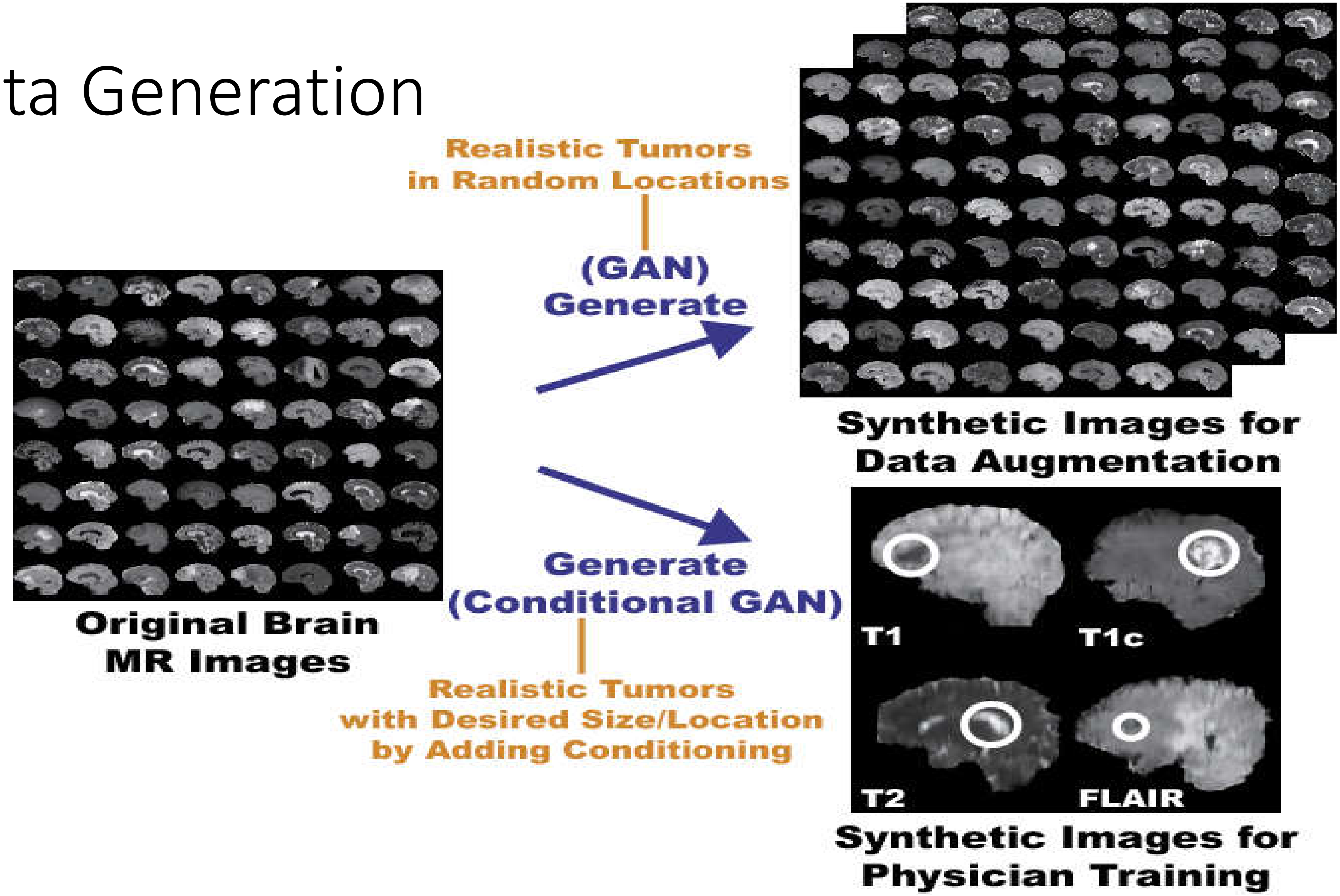
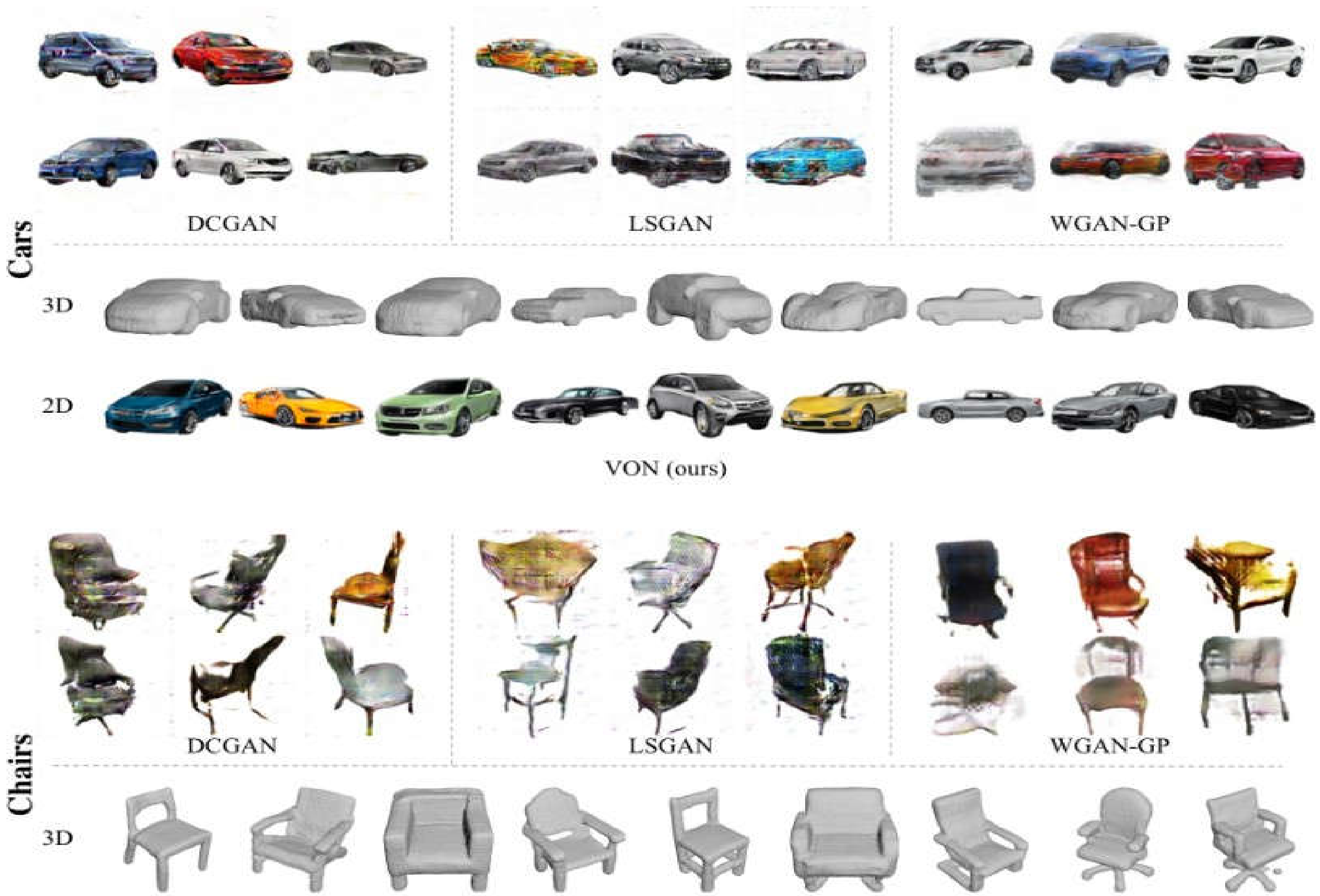
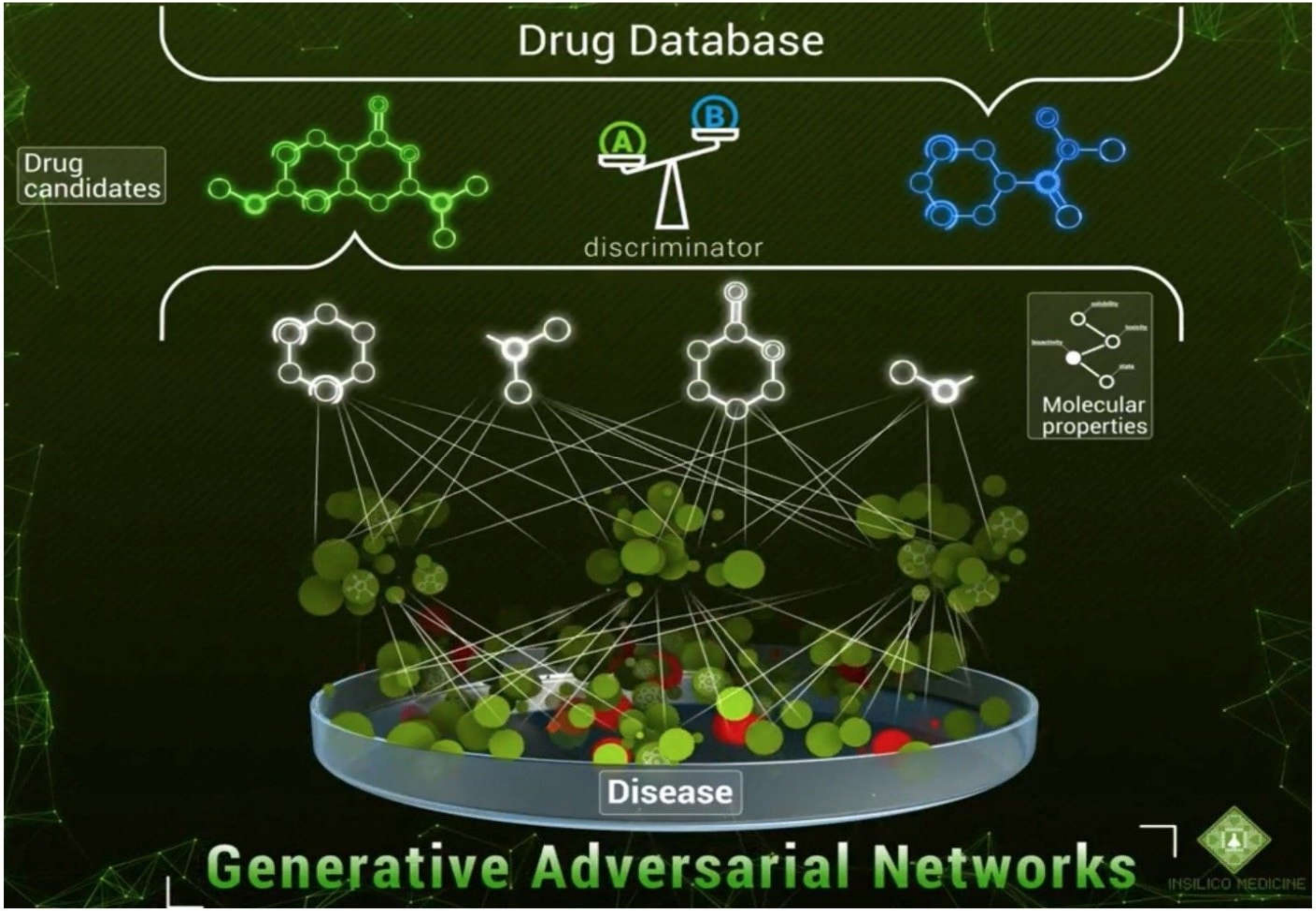


Image colorization

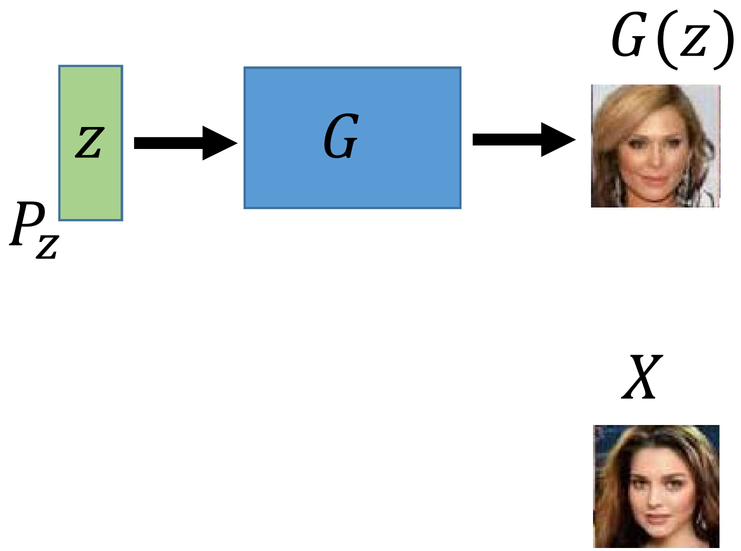
Image Editing

3D object generation



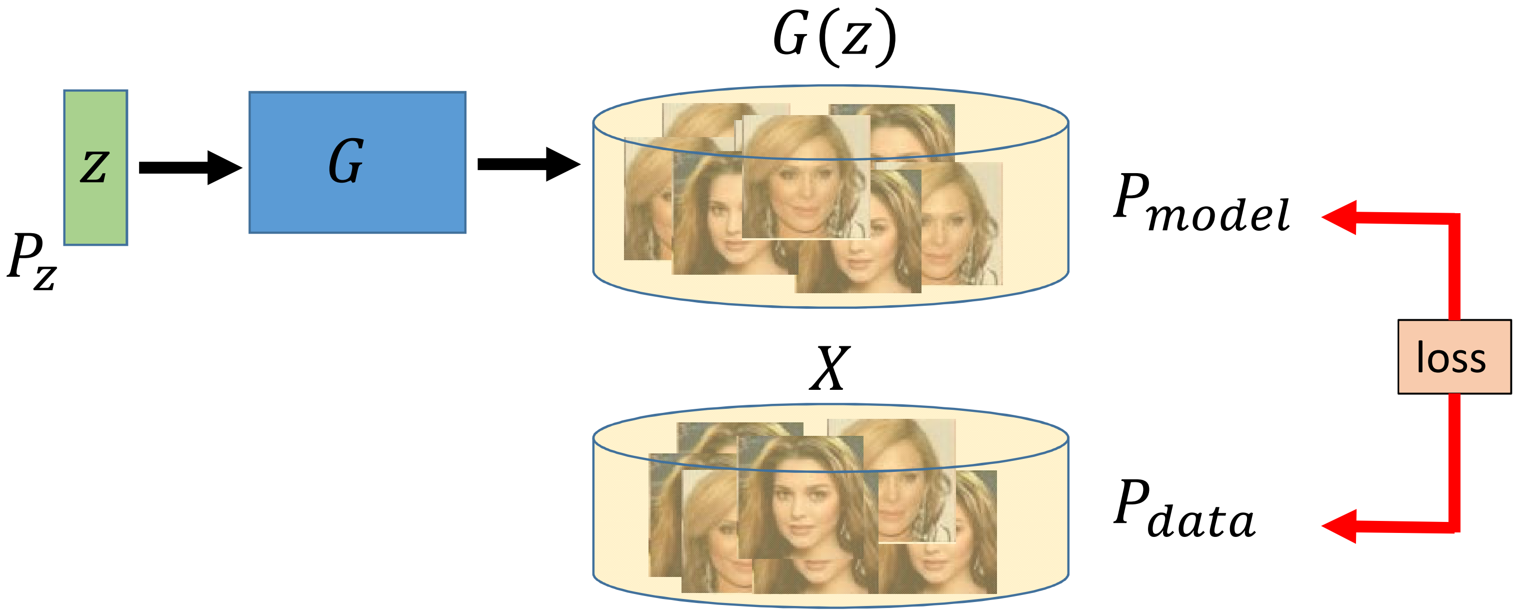

Drug Database
Drug candidates
A
B
discriminator
Molecular properties
Disease
Generative Adversarial Networks
INSILICO MEDICINE

Generative Models



- Does not look for an explicit distribution of the real data
- Generate data from noise (random distribution)
- Learn a neural nework that projects noise into the real data distribution

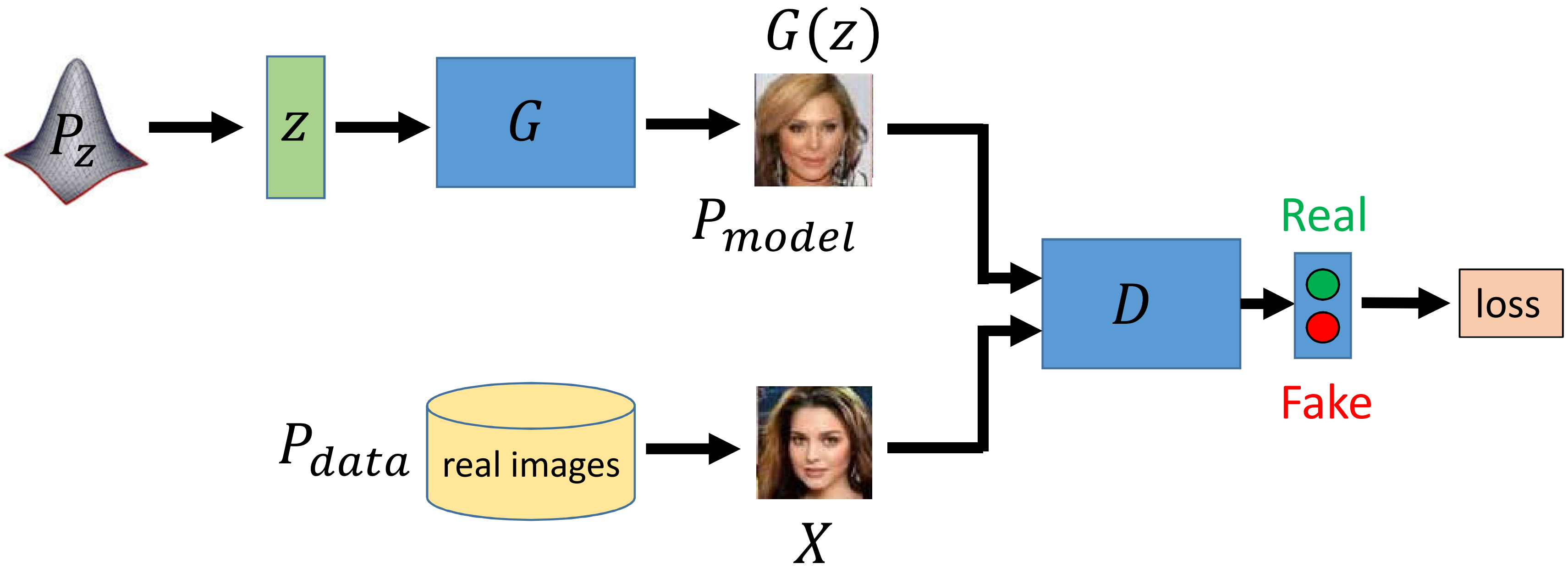
Generative Models



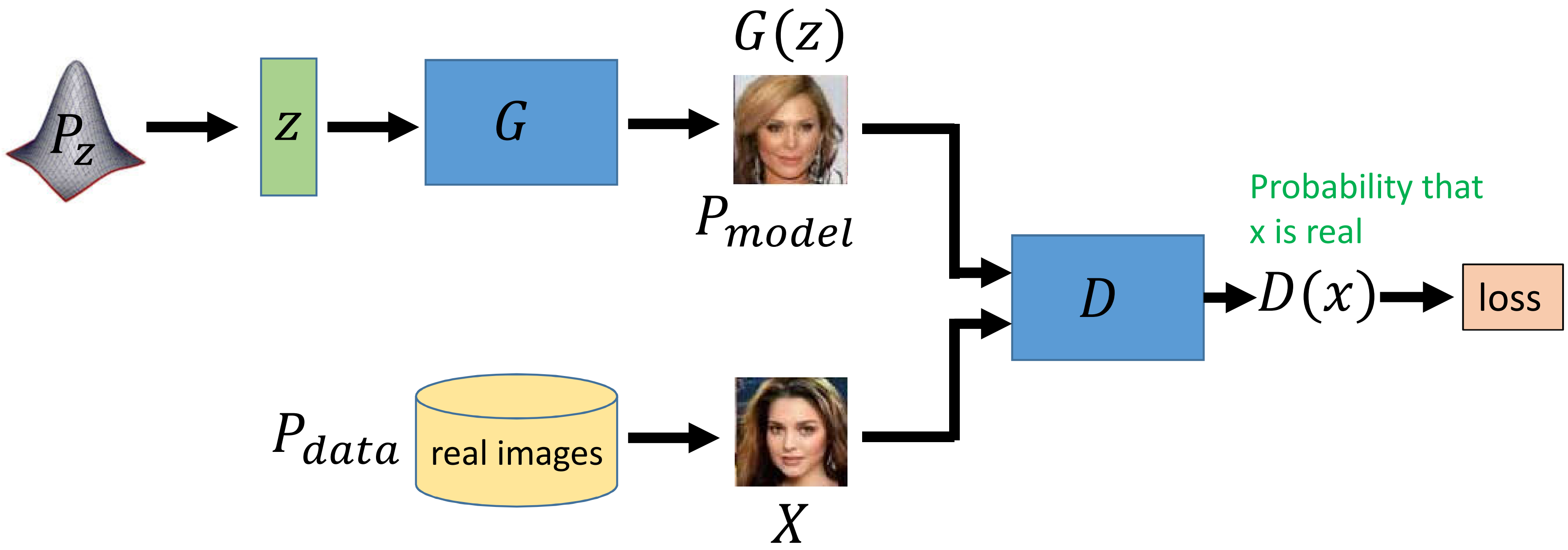
Objective : $P_{data} = P_{model}$

Learning : Minimizing the distance between P_{data} and P_{model}

Generative Adversarial Networks

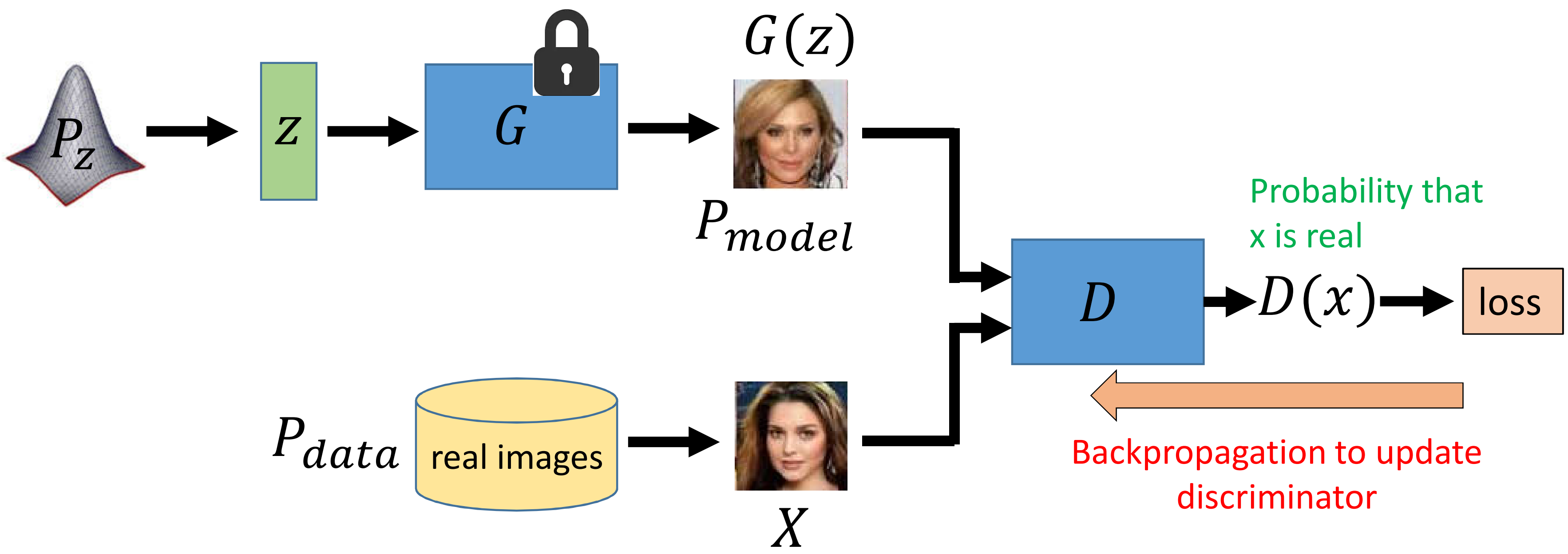


Generative Adversarial Networks



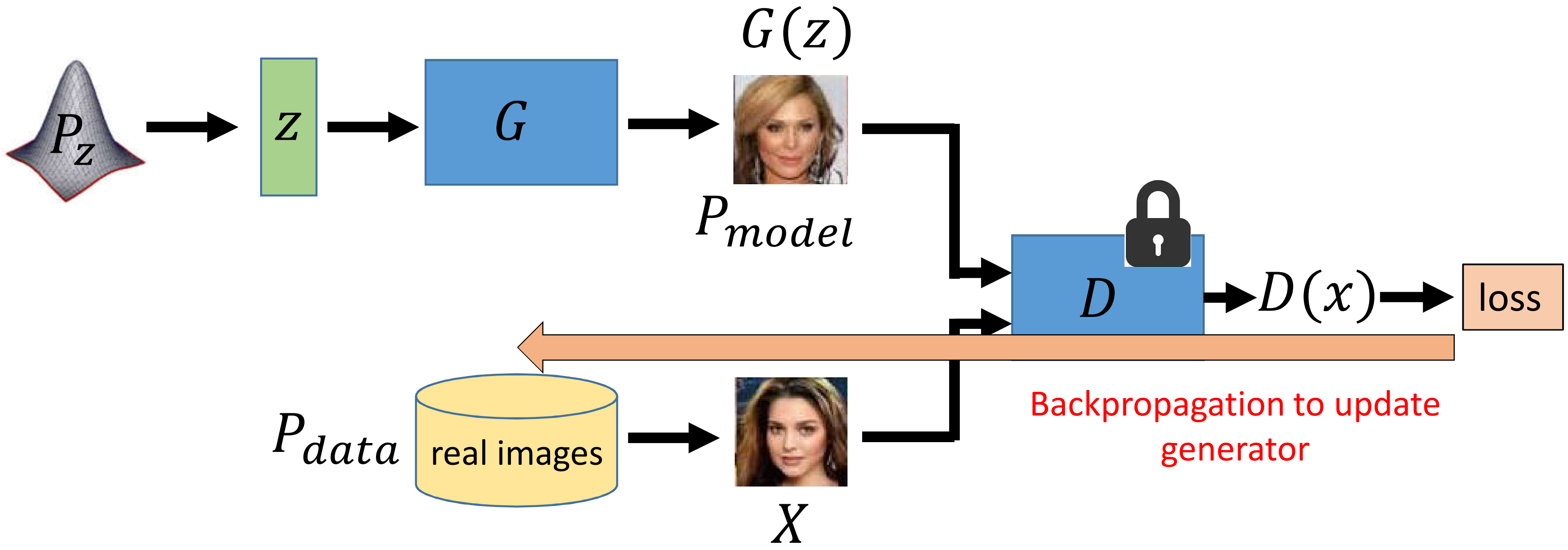
$$max_D min_G \left[\mathbb{E}_{x \sim p_{data}} \log \underbrace{D(x)}_{\text{Output of the discriminator for real data}} + \mathbb{E}_{z \sim p_z} \log(1 - \underbrace{D(G(z))}_{\text{Output of the discriminator for artificial data}})\right]$$

Training Discriminator



$$max_D\left[\mathbb{E}_{x\backsim p_{data}} \log \underbrace{D(x)}_{\text{Output of the discriminator for real data}} + \mathbb{E}_{z\backsim p_z} \log(1-\underbrace{D(G(z))}_{\text{Output of the discriminator for artificial data}})\right]$$

Training Generator



$$min_G\left[\mathbb{E}_{x\sim p_{data}} \log \underbrace{D(x)}_{\text{Output of the discriminator for real data}} + \mathbb{E}_{z\sim p_z} \log(1 - \underbrace{D(G(z))}_{\text{Output of the discriminator for artificial data}})\right]$$

GAN's formulation

A 2-players game, where:

- The discriminator is trying to maximize $L(D, G)$
- The generator is trying to minimize $L(D, G)$

$$(D^*, G^*) = max_D min_G L(D, G)$$

$$L(D, G) = \left[\mathbb{E}_{x \backsim p_d} \log D(x) + \mathbb{E}_{z \backsim p_z} \log(1 - D(G(z)))\right]$$

The Nash equilibrium of this particular game is achieved at:

- $P_{data} = P_{model}$
- $D(x) = \frac{1}{2} \; \forall x$

GAN's loss function

$$L(D,G) = \mathbb{E}_{x \backsim p_d} \log D(x) + \mathbb{E}_{z \backsim p_z} \log(1 - D(G(z)))$$
$$= \mathbb{E}_{x \backsim p_d} \log D(x) + \mathbb{E}_{x \backsim p_m} \log(1 - D(x))$$

- For a given G, what is the optimal D ?

$$D^* = argmax_D(L(D,G))$$

$$\frac{\partial L(D,G)}{\partial D(x)} = 0 \quad \Longrightarrow \quad D^*(x) = \frac{p_d(x)}{p_d(x) + p_m(x)}$$ Bayes classifier

GAN's loss function

- Given D^*, what is the optimal G ?

$$G^* = argmin_D(L(D^*, G))$$

Kullback-Leiber divergence :

$$KL(p_1, p_2) = \mathbb{E}_{x \backsim p_1} log \frac{p_1(x)}{p_2(x)}$$

Jensen-Shannon divergence :

$$JS(p_1, p_2) = \frac{1}{2} KL\left(p_1, \frac{p_1 + p_2}{2}\right) + \frac{1}{2} KL\left(p_2, \frac{p_1 + p_2}{2}\right)$$

$$L = \mathbb{E}_{x \backsim p_d} \log \frac{p_m(x)}{\frac{1}{2}(p_d(x) + p_m(x))} + \mathbb{E}_{x \backsim p_m} \log \frac{p_d(x)}{\frac{1}{2}(p_d(x) + p_m(x))} - 2log2$$

$$L = \frac{1}{2} KL\left(p_m || \frac{p_d + p_m}{2}\right) + \frac{1}{2} KL\left(p_d || \frac{p_d + p_m}{2}\right) - 2log2$$

$$L = 2\, JS(p_m, p_d) - 2log2$$

GAN Training

Objective : $max_D min_G \left[\mathbb{E}_{x \backsim p_d} \log D(x) + \mathbb{E}_{z \backsim p_z} \log(1 - D(G(z)))\right]$

Alternate the training of the two networks :

1. Discriminator : Gradient ascent

$$max_D \left[\mathbb{E}_{x \backsim p_d} \log D(x) + \mathbb{E}_{z \backsim p_z} \log(1 - D(G(z)))\right]$$

2. Generator : Gradient descent

$$min_G \left[\mathbb{E}_{z \backsim p_z} \log(1 - D(G(z)))\right]$$

Apprentissage GAN : un jeu à deux joueurs

Objective : $max_D min_G \left[\mathbb{E}_{x \backsim p_d} \log D(x) + \mathbb{E}_{z \backsim p_z} \log(1 - D(G(z)))\right]$

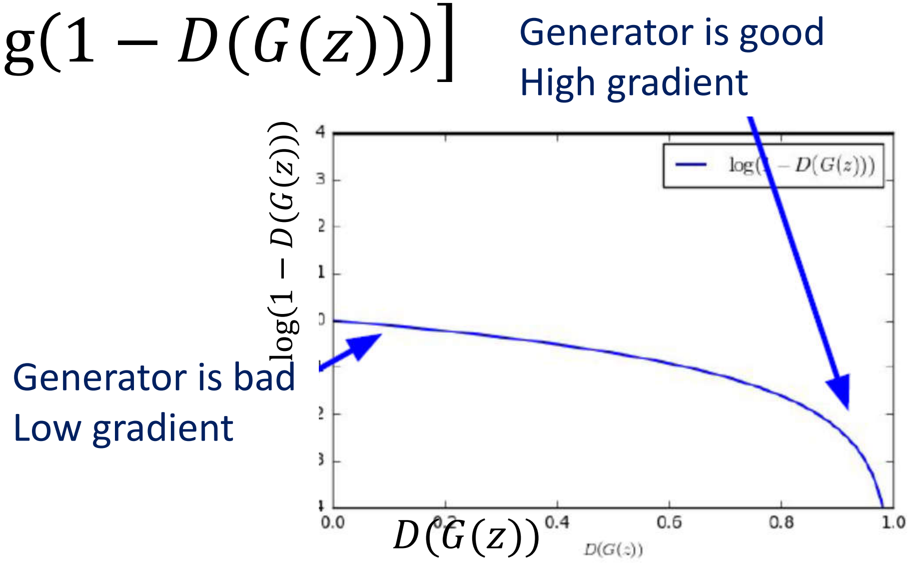
Alternate the training of the two networks :

1. Discriminator : Gradient ascent

$$max_D\left[\mathbb{E}_{x \backsim p_d} \log D(x) + \mathbb{E}_{z \backsim p_z} \log(1 - D(G(z)))\right]$$

2. Generator : Gradient descent

$$min_G\left[\mathbb{E}_{z \backsim p_z} \log(1 - D(G(z)))\right]$$



Apprentissage GAN : un jeu à deux joueurs

Objective : $max_D min_G \left[\mathbb{E}_{x\backsim p_d} \log D(x) + \mathbb{E}_{z\backsim p_z} \log(1 - D(G(z)))\right]$

Alternate the training of the two networks :

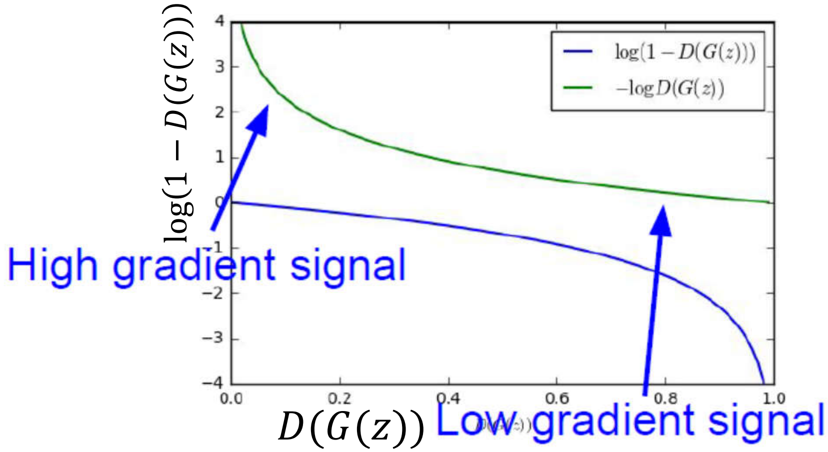
1. Discriminator : Gradient ascent

$$max_D \left[\mathbb{E}_{x\backsim p_d} \log D(x) + \mathbb{E}_{z\backsim p_z} \log(1 - D(G(z)))\right]$$

2. Generator : Gradient ascent

$$\cancel{min_G \left[\mathbb{E}_{z\backsim p_z} \log(1 - D(G(z)))\right]}$$

$$max_G \left[\mathbb{E}_{z\backsim p_z} \log(D(G(z)))\right]$$



Apprentissage GAN : un jeu

Objective : $max_D min_G \left[\mathbb{E}_{x\backsim p_d} \log D(x) + \mathbb{E}\right.$

Alternate the training of the two networks :

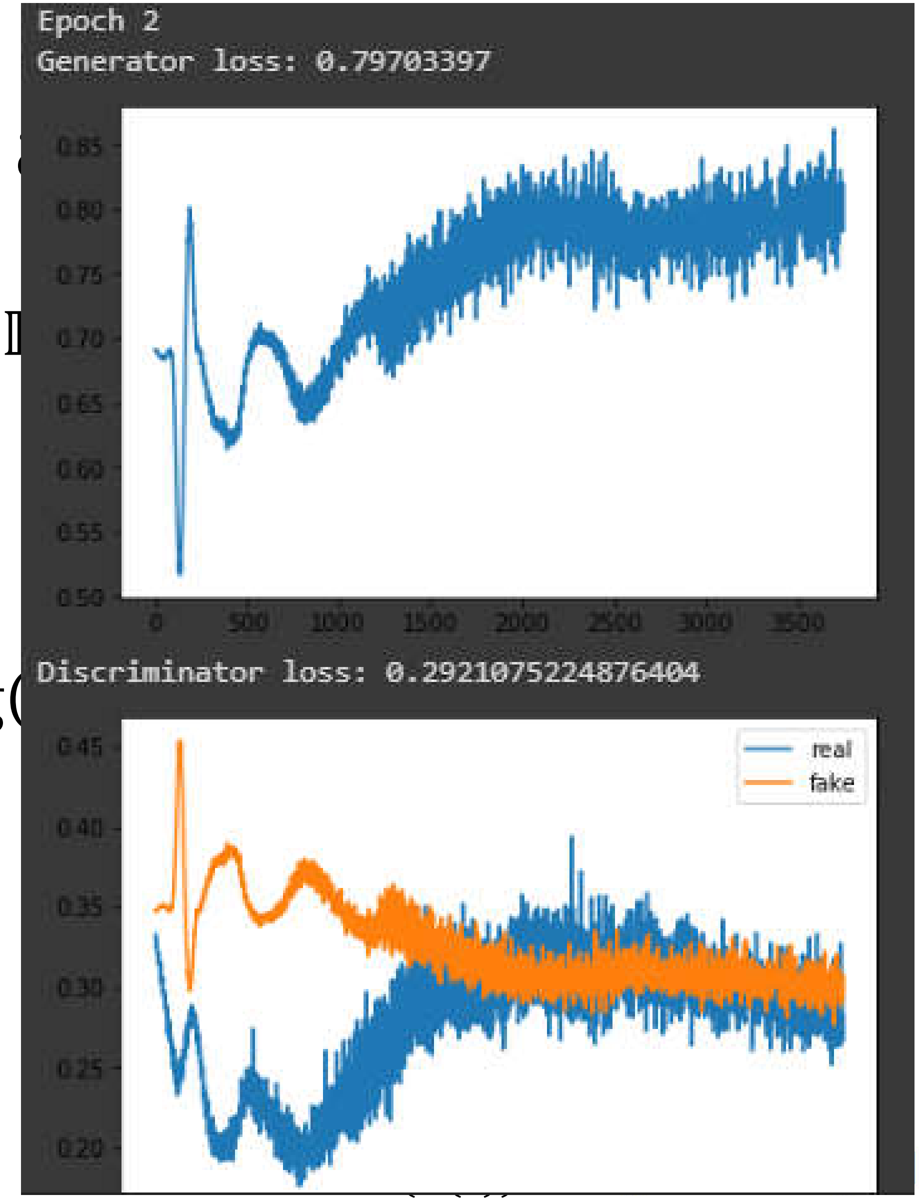
1. Discriminator : Gradient ascent

$$max_D\left[\mathbb{E}_{x\backsim p_d} \log D(x) + \mathbb{E}_{z\backsim p_z} \log(\right.$$

2. Generator : Gradient ascent

$$\cancel{min_G\left[\mathbb{E}_{z\backsim p_z} \log(1 - D(G(z)))\right]}$$

$$max_G\left[\mathbb{E}_{z\backsim p_z} \log(D(G(z)))\right]$$



Unstable training

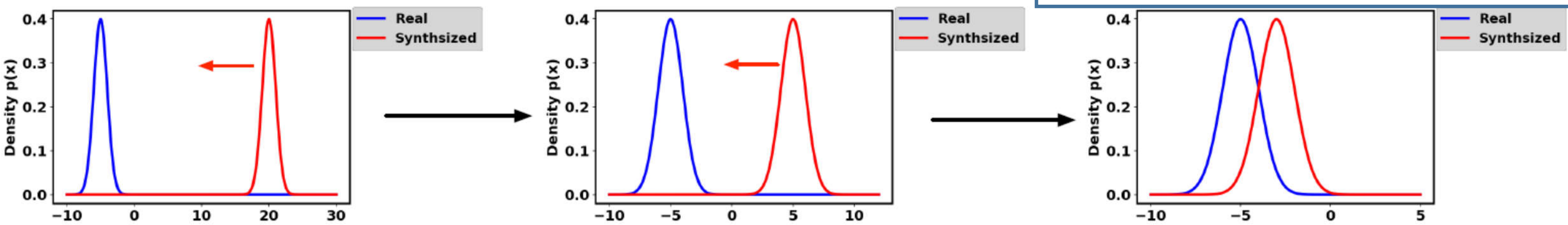
Unstable training caused by JS divergence

Kullback-Leiber divergence :

$$KL(p_1, p_2) = \mathbb{E}_{x \sim p_1} log \frac{p_1(x)}{p_2(x)}$$

Jensen-Shannon divergence :

$$JS(p_m, p_d) = \frac{1}{2} KL\left(p_m, \frac{p_m + p_d}{2}\right) + \frac{1}{2} KL\left(p_d, \frac{p_m + p_d}{2}\right)$$



$JS(p_m, p_d) = 0{,}693$ $JS(p_m, p_d) = 0{,}693$ $JS(p_m, p_d) = 0{,}336$

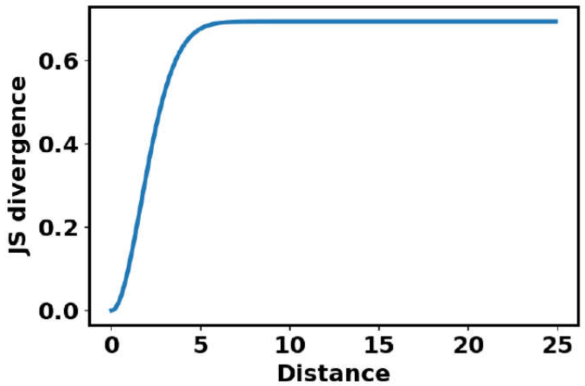
- With an optimal discriminator
- If no overlap between P_m and P_d

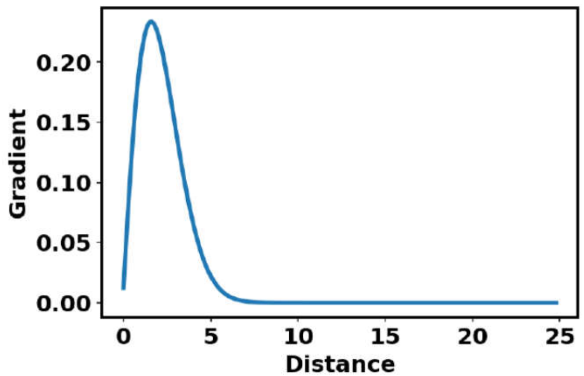
➢ $JS(p_m, p_d) = log2$

➢ $L = 2\, JS(p_m, p_d) - 2log2 = 0$

➢ Gradient is null

➢ No training on G





Dropping mode

- With D^*, $KL(p_m||p_d)$ can be reformulated as :

$$KL(p_m||p_d) = \mathbb{E}_{x \backsim p_m} \log(1 - D^*(x)) - \mathbb{E}_{x \backsim p_m} \log D^*(x)$$

- G tries to minimizes $-\mathbb{E}_{x \backsim p_m} \log D^*(x)$

$$-\mathbb{E}_{x \backsim p_m} \log D^*(x) = KL(p_m||p_d) - \mathbb{E}_{x \backsim p_m} \log\big(1 - D^*(x)\big)$$

- In introducing the $L = 2\, JS(p_m||p_d) - 2log2$

$$-\mathbb{E}_{x \backsim p_m} \log D^*(x) = KL(p_m||p_d) - 2\, JS(p_m||p_d) - 2log2 + \mathbb{E}_{x \backsim p_d} \log D^*(x)$$

- G tries both to minimize $KL(p_m||p_d)$ and to maximize $JS(p_m||p_d)$

Mode collapse

- KL divergence is an unsymmetrical distribution measure

$$if \quad p_m(x) \to 0, p_d(x) \to 1 \quad then \quad KL(p_m||p_d) \to 0$$

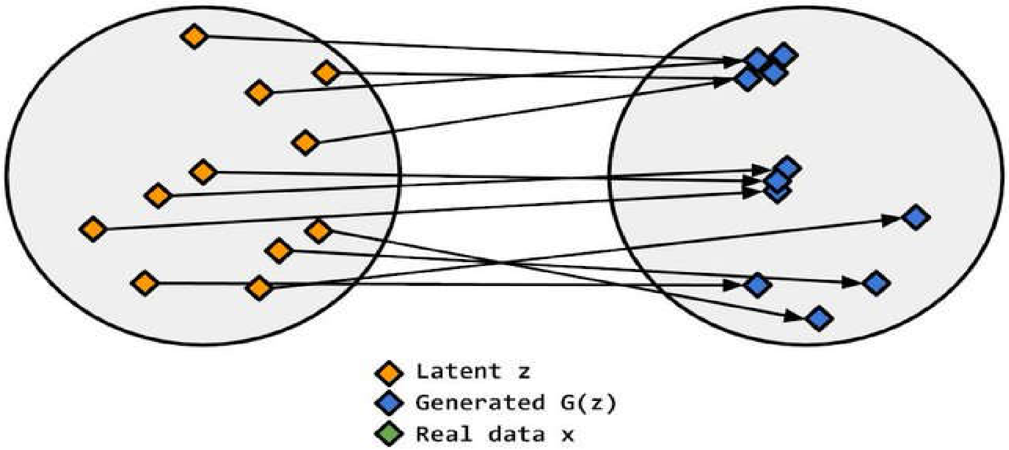
  - G does not produce very plausible data, tiny penalization
  - ➢ Generated data lack the diversity

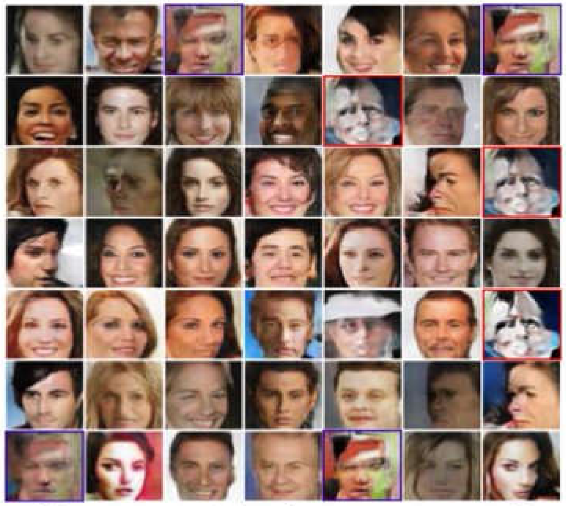
$$if \quad p_m(x) \to 1, p_d(x) \to 0 \quad then \quad KL(p_m||p_d) \to +\infty$$

  - G produces implausible data, large penalization
  - ➢ Generated data are not accurate

- G will prefer the first case

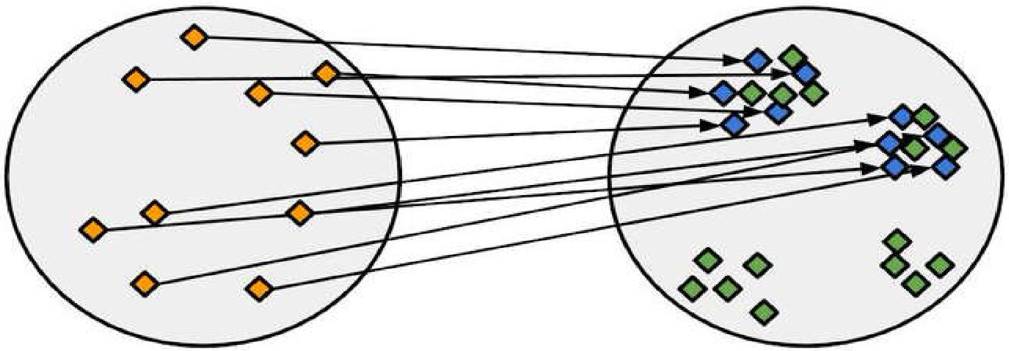
Mode Collapse: Many z to ~one x



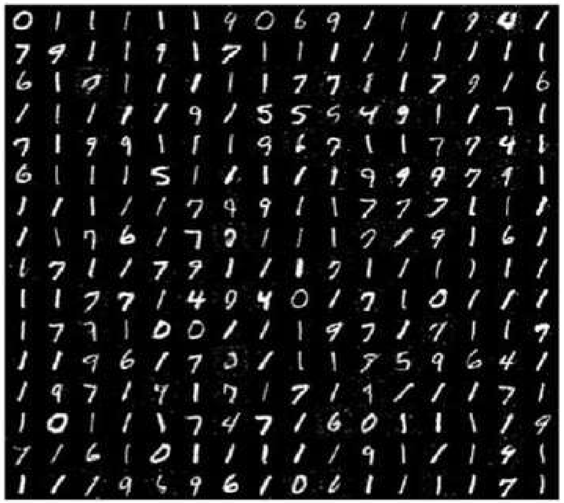


Mode Collapse on CelebA (Source: Geometric GAN)

Mode Dropping: Real Data not in G(z)



- Mode collapse and Mode Dropping can co-occur
- Complete training collapse is a separate phenomenon, for which extreme Mode Dropping and/or Mode Collapse are often symptoms



Mode Dropping on MNIST (Source:TripletGAN)

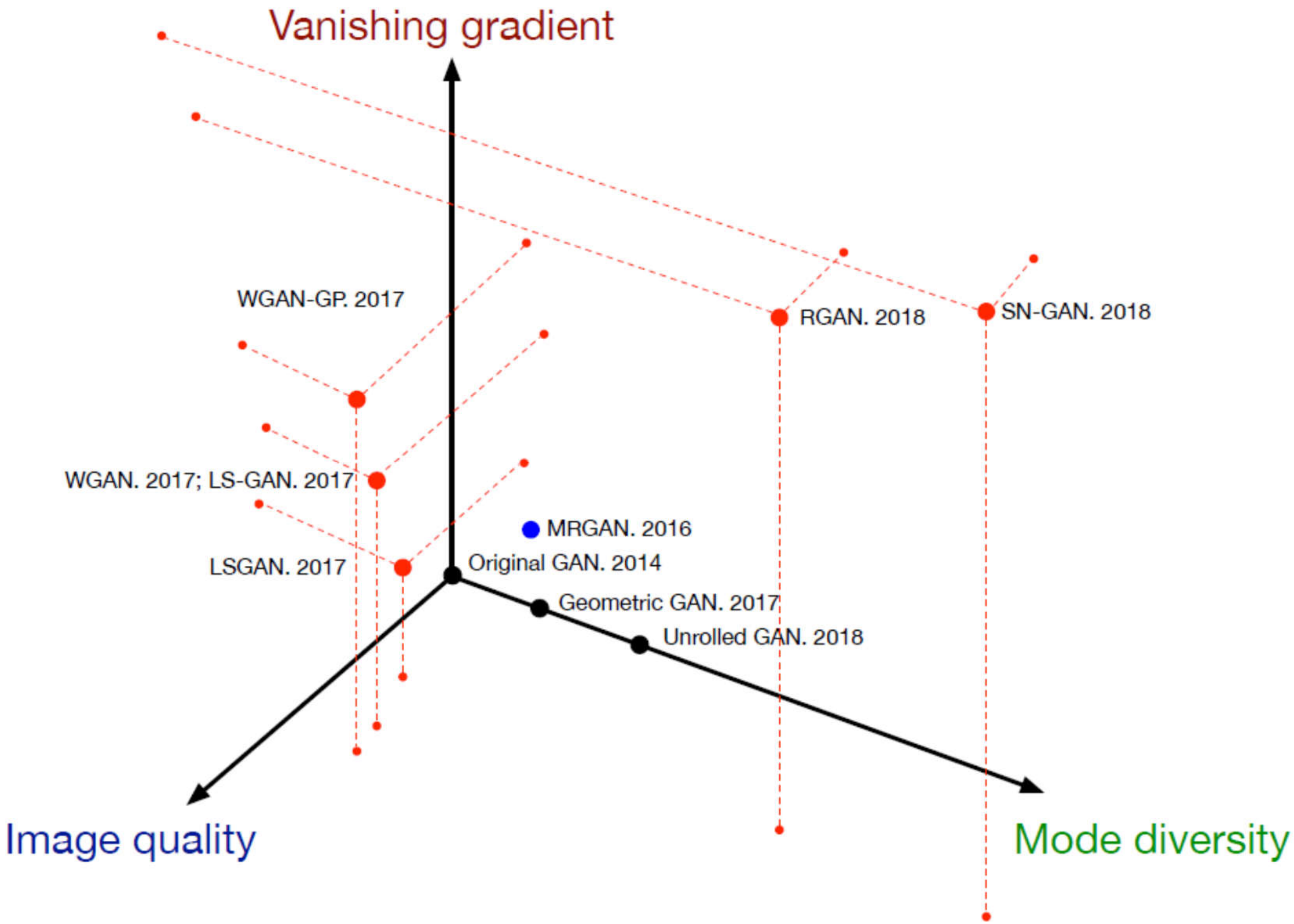
Nombreuses variantes

- GAN - Generative Adversarial Networks
- 3D-GAN - Learning a Probabilistic Latent Space of Object Shapes via 3D Generative-Adversarial Modeling
- acGAN - Face Aging With Conditional Generative Adversarial Networks
- AC-GAN - Conditional Image Synthesis With Auxiliary Classifier GANs
- AdaGAN - AdaGAN: Boosting Generative Models
- AEGAN - Learning Inverse Mapping by Autoencoder based Generative Adversarial Nets
- AffGAN - Amortised MAP Inference for Image Super-resolution
- AL-CGAN - Learning to Generate Images of Outdoor Scenes from Attributes and Semantic Layouts
- ALI - Adversarially Learned Inference
- AM-GAN - Generative Adversarial Nets with Labeled Data by Activation Maximization
- AnoGAN - Unsupervised Anomaly Detection with Generative Adversarial Networks to Guide Marker Discovery
- ArtGAN - ArtGAN: Artwork Synthesis with Conditional Categorial GANs
- b-GAN - b-GAN: Unified Framework of Generative Adversarial Networks
- Bayesian GAN - Deep and Hierarchical Implicit Models
- BEGAN - BEGAN: Boundary Equilibrium Generative Adversarial Networks
- BiGAN - Adversarial Feature Learning
- BS-GAN - Boundary-Seeking Generative Adversarial Networks
- CGAN - Conditional Generative Adversarial Nets
- CaloGAN - CaloGAN: Simulating 3D High Energy Particle Showers in Multi-Layer Electromagnetic Calorimeters with Generative Adversarial Networks
- CCGAN - Semi-Supervised Learning with Context-Conditional Generative Adversarial Networks
- CatGAN - Unsupervised and Semi-supervised Learning with Categorical Generative Adversarial Networks
- CoGAN - Coupled Generative Adversarial Networks
- Context-RNN-GAN - Contextual RNN-GANs for Abstract Reasoning Diagram Generation
- C-RNN-GAN - C-RNN-GAN: Continuous recurrent neural networks with adversarial training
- CS-GAN - Improving Neural Machine Translation with Conditional Sequence Generative Adversarial Nets
- CVAE-GAN - CVAE-GAN: Fine-Grained Image Generation through Asymmetric Training
- CycleGAN - Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks
- DTN - Unsupervised Cross-Domain Image Generation
- DCGAN - Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks
- DiscoGAN - Learning to Discover Cross-Domain Relations with Generative Adversarial Networks
- DR-GAN - Disentangled Representation Learning GAN for Pose-Invariant Face Recognition
- DualGAN - DualGAN: Unsupervised Dual Learning for Image-to-Image Translation
- EBGAN - Energy-based Generative Adversarial Network
- f-GAN - f-GAN: Training Generative Neural Samplers using Variational Divergence Minimization
- FF-GAN - Towards Large-Pose Face Frontalization in the Wild
- GAWWN - Learning What and Where to Draw
- GeneGAN - GeneGAN: Learning Object Transfiguration and Attribute Subspace from Unpaired Data
- Geometric GAN - Geometric GAN
- GoGAN - Gang of GANs: Generative Adversarial Networks with Maximum Margin Ranking
- GP-GAN - GP-GAN: Towards Realistic High-Resolution Image Blending
- IAN - Neural Photo Editing with Introspective Adversarial Networks
- iGAN - Generative Visual Manipulation on the Natural Image Manifold
- IcGAN - Invertible Conditional GANs for image editing
- ID-CGAN - Image De-raining Using a Conditional Generative Adversarial Network
- Improved GAN - Improved Techniques for Training GANs
- InfoGAN - InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets
- LAGAN - Learning Particle Physics by Example: Location-Aware Generative Adversarial Networks for Physics Synthesis
- LAPGAN - Deep Generative Image Models using a Laplacian Pyramid of Adversarial Networks

2014

2019

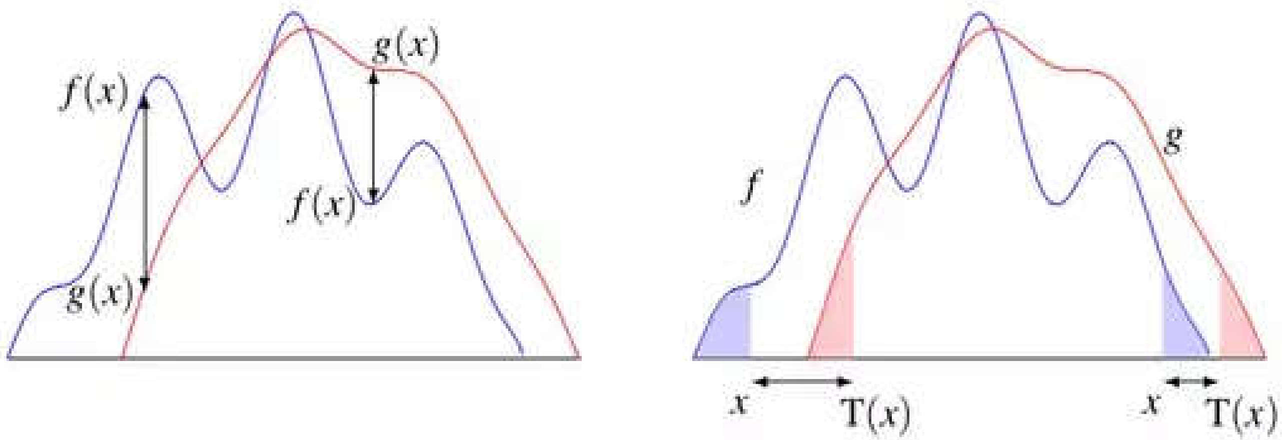
Loss Variant GAN



Wasserstein GAN (WGAN)

- Wasserstein distance computes the distance between 2 distributions based on the optimal transport theory

$$W(p_d, p_m) = inf_{\gamma \in \Pi(p_d, p_m)} \mathbb{E}_{(x,y) \sim \gamma} \|x - y\|$$



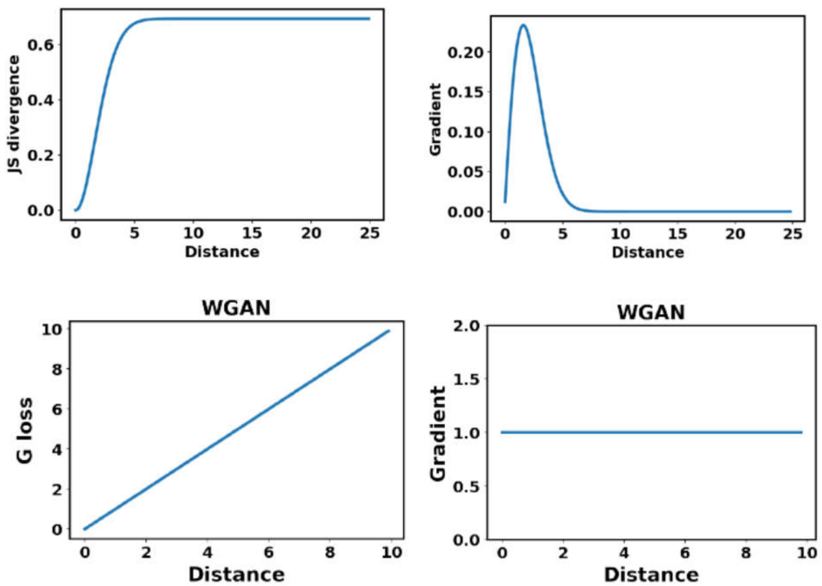
Wasserstein GAN (WGAN)

- Wasserstein distance computes the distance between 2 distributions based on the optimal transport theory

$$W(p_d, p_m) = inf_{\gamma \in \Pi(p_d, p_m)} \mathbb{E}_{(x,y)\sim\gamma} \|x - y\|$$

- W distance is intractable ! But can be estimated
- The discriminator is use to estimate the W distance
  - D_w : regression problem, sigmoid output layer

$$L_G = -\mathbb{E}_{z \backsim p_z} \log(D_w\,(G(z)))$$



Spectral Normalization GAN (SN-GAN)

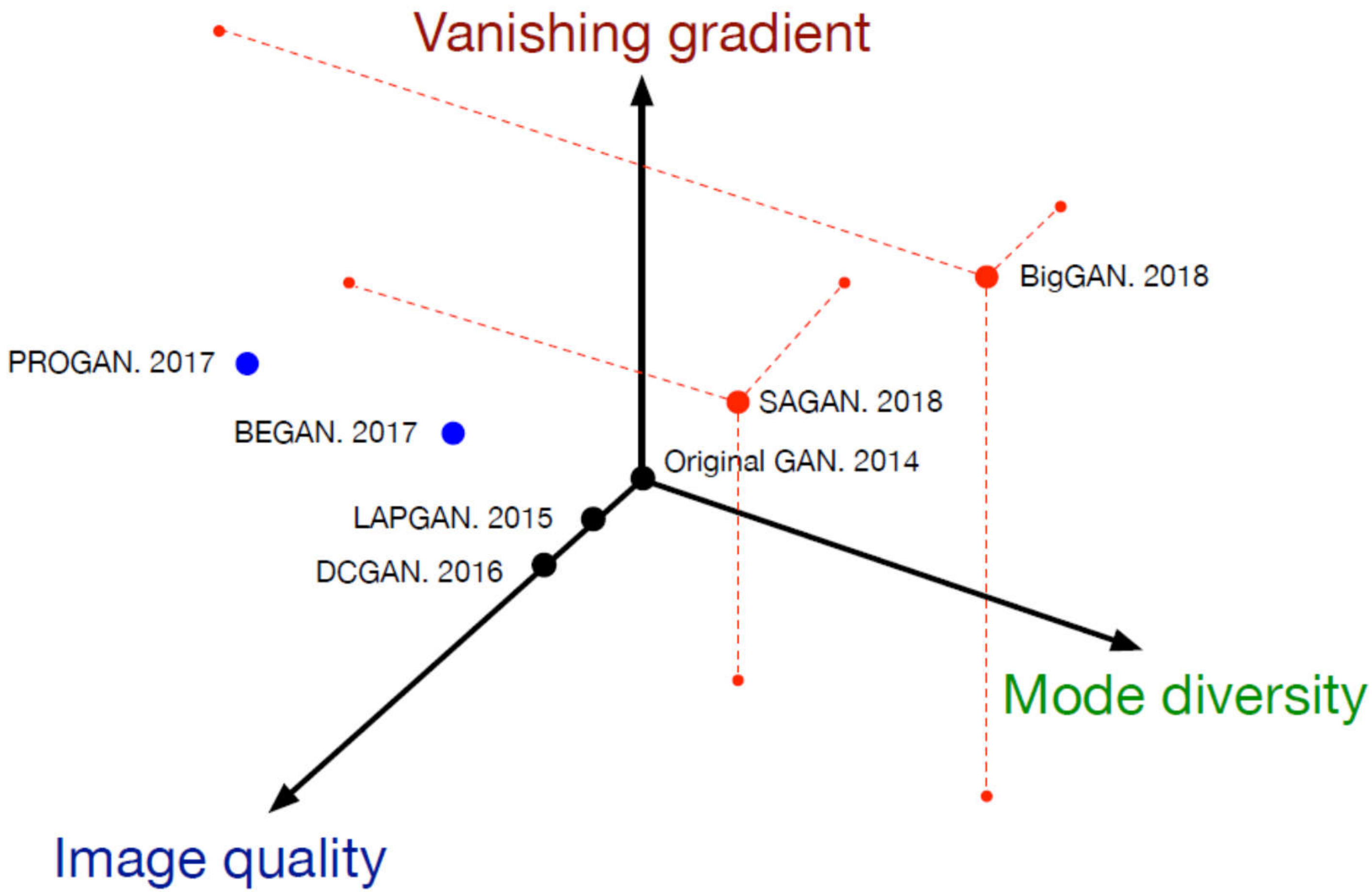
- Weight normalization to stabilize the training of D
- D should be a K-Lipshitz continous function

$$\widetilde{W}_{SN}(W) = \frac{W}{\sigma(W)}$$

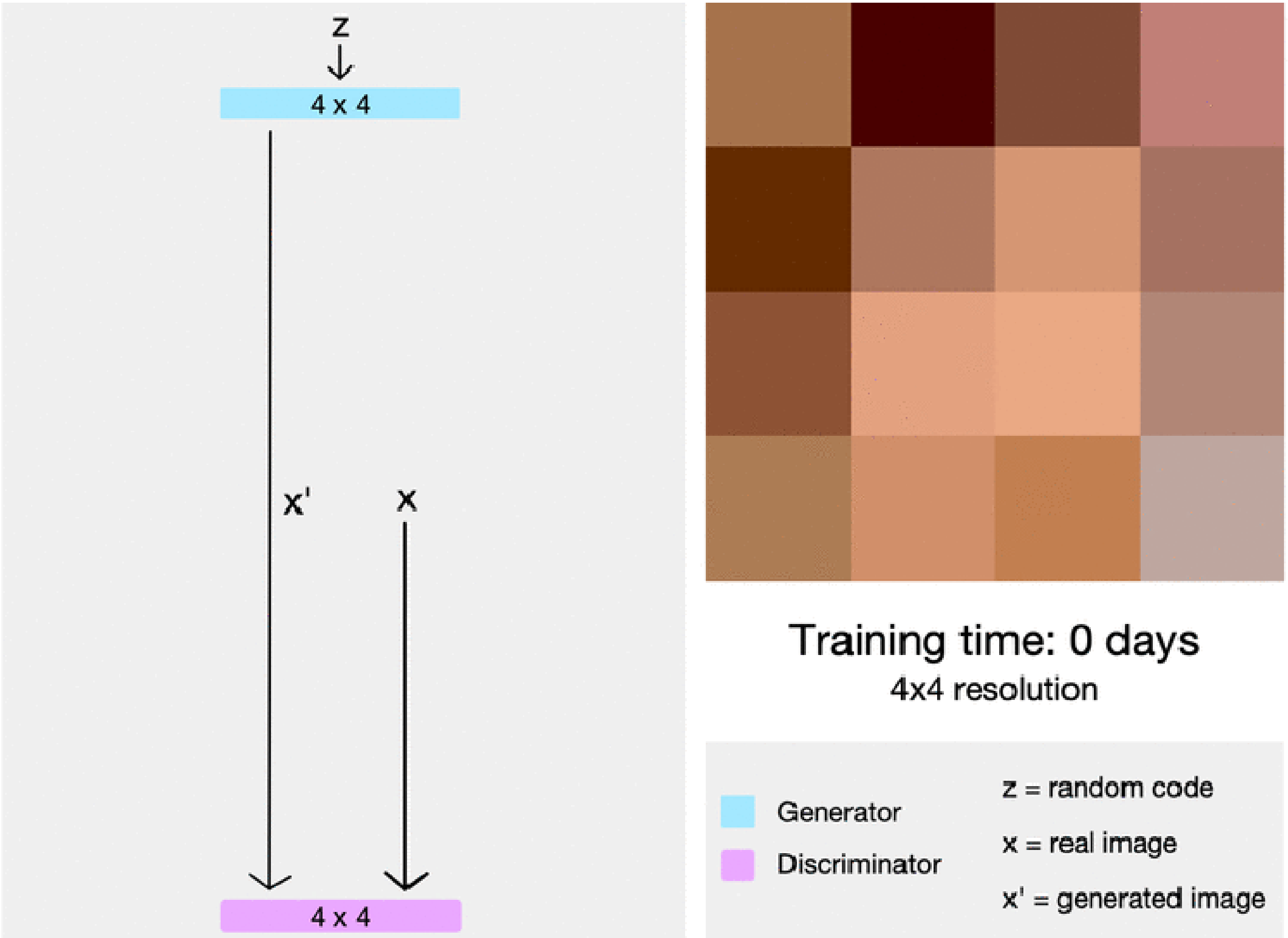
With W the weights matrix of D
$\sigma(W)$ the L_2 matrix norm of W

- Computationally light and easily applied to other GAN

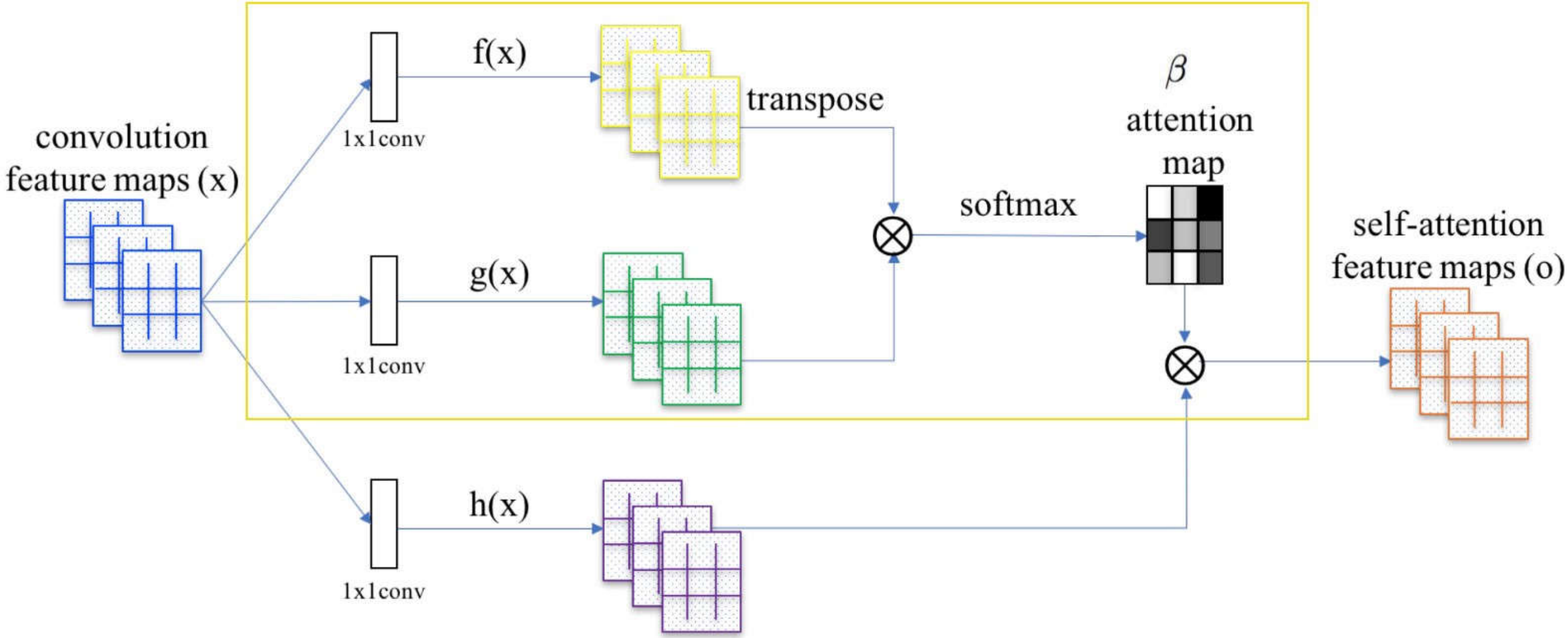
Architecture variant GAN



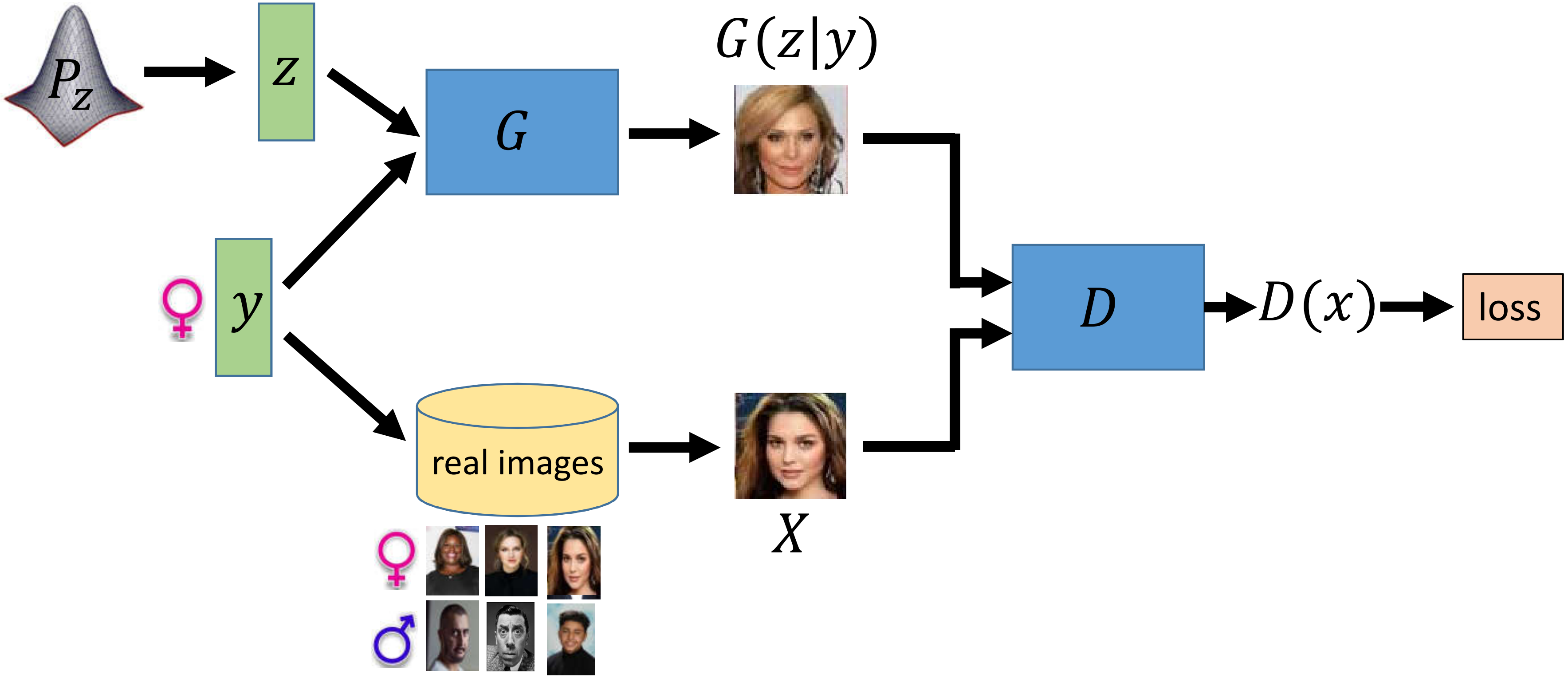
Progressive GAN (PROGAN)



Self Attention GAN (SAGAN)

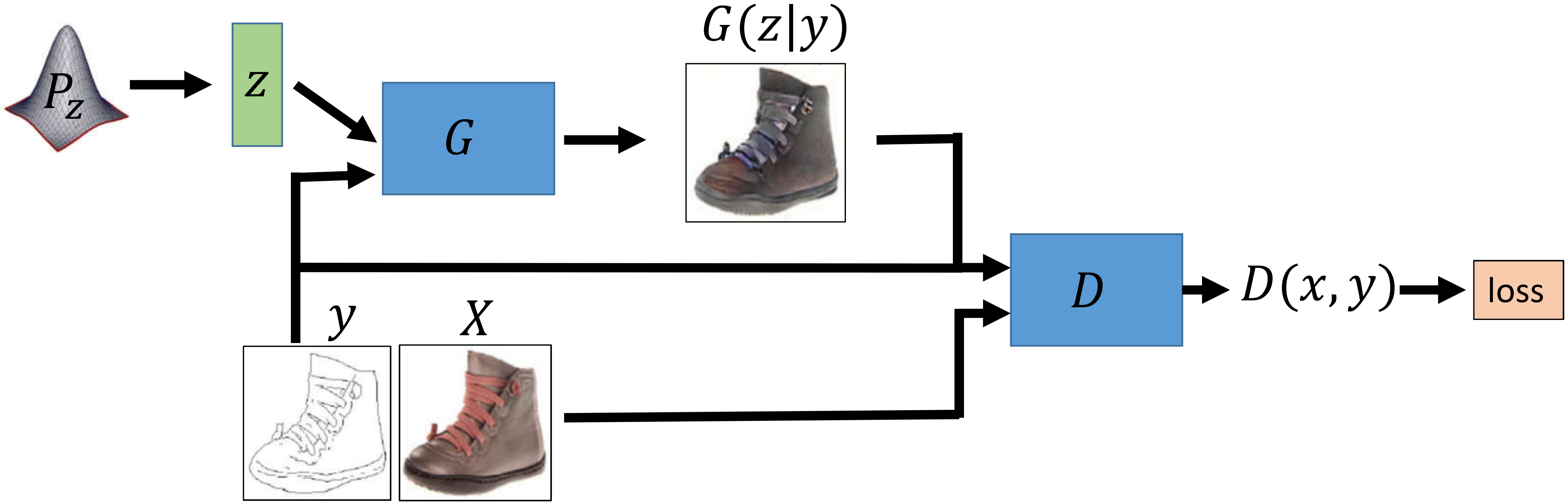


Conditional GAN (cGAN)



$$L(D,G) = \left[\mathbb{E}_{x \backsim p_{data}} \log D(x|y) + \mathbb{E}_{z \backsim p_z} \log(1 - D(G(z|y)))\right]$$

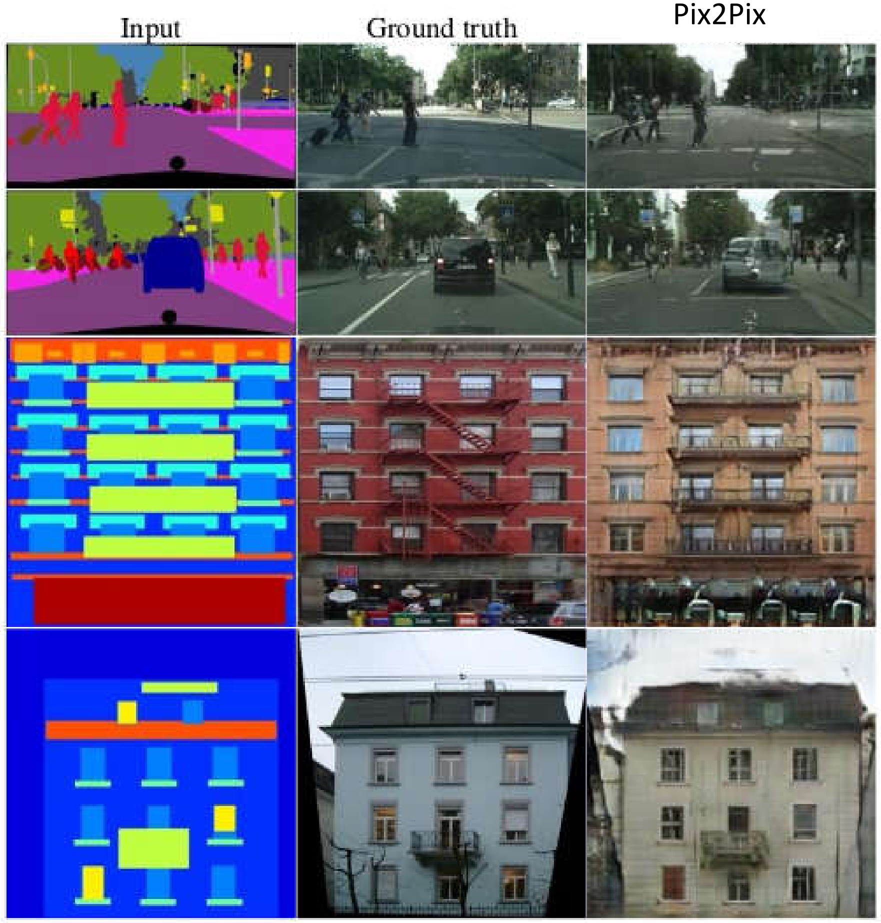
cGAN for image translation (Pix2Pix)



Real pair of images

$$L(D,G) = \left[\mathbb{E}_{x \backsim p_{data}} \log D(x, y) + \mathbb{E}_{z \backsim p_z} \log(1 - D(G(z|y), y))\right]$$

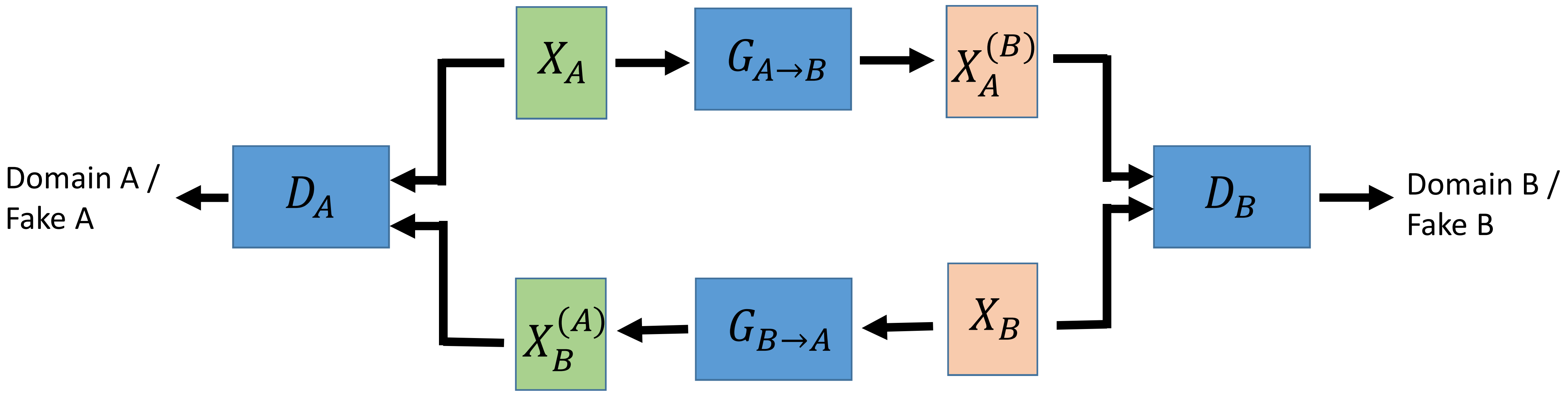
cGAN for image translation (Pix2Pix)



The small bird has a red head with feathers that fade from red to gray from head to tail

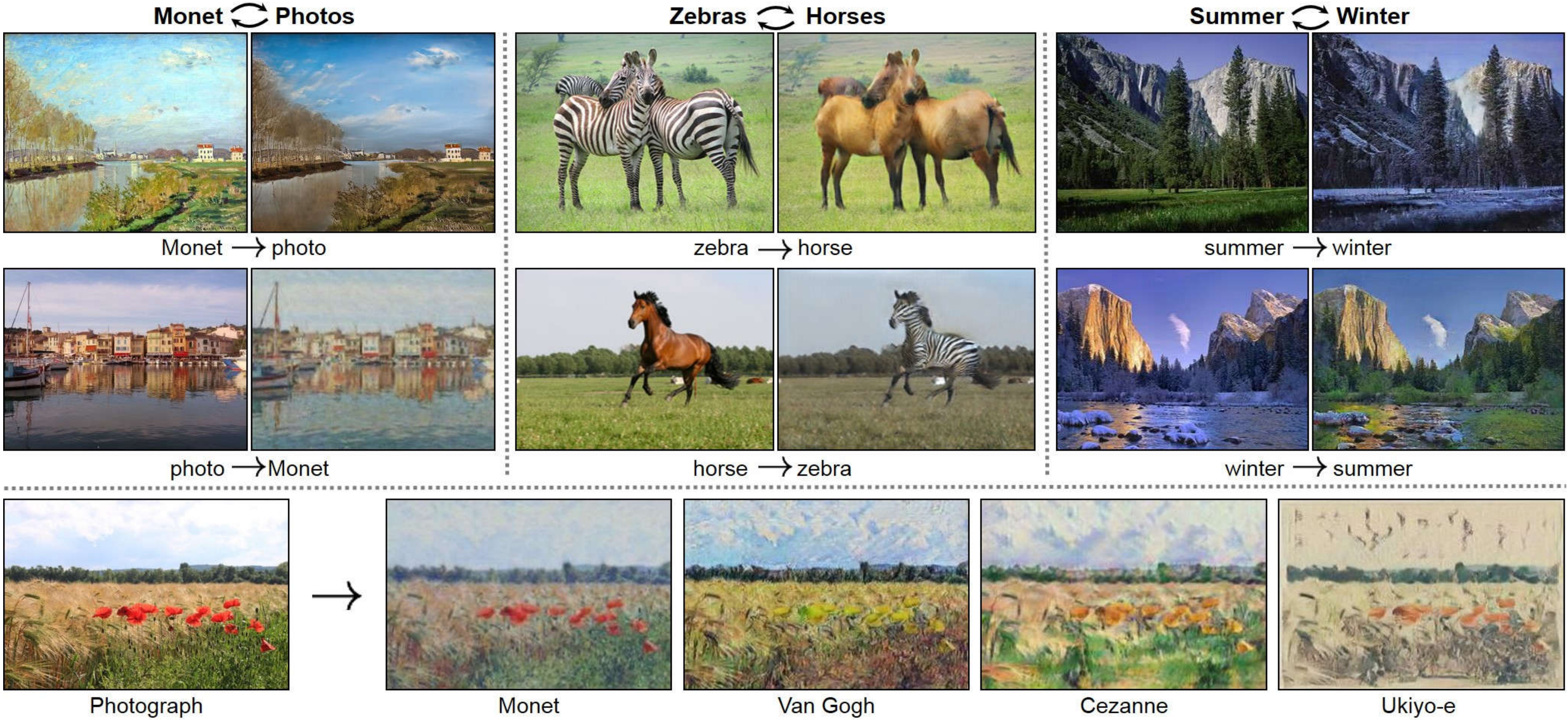
This bird is black with green and has a very short beak

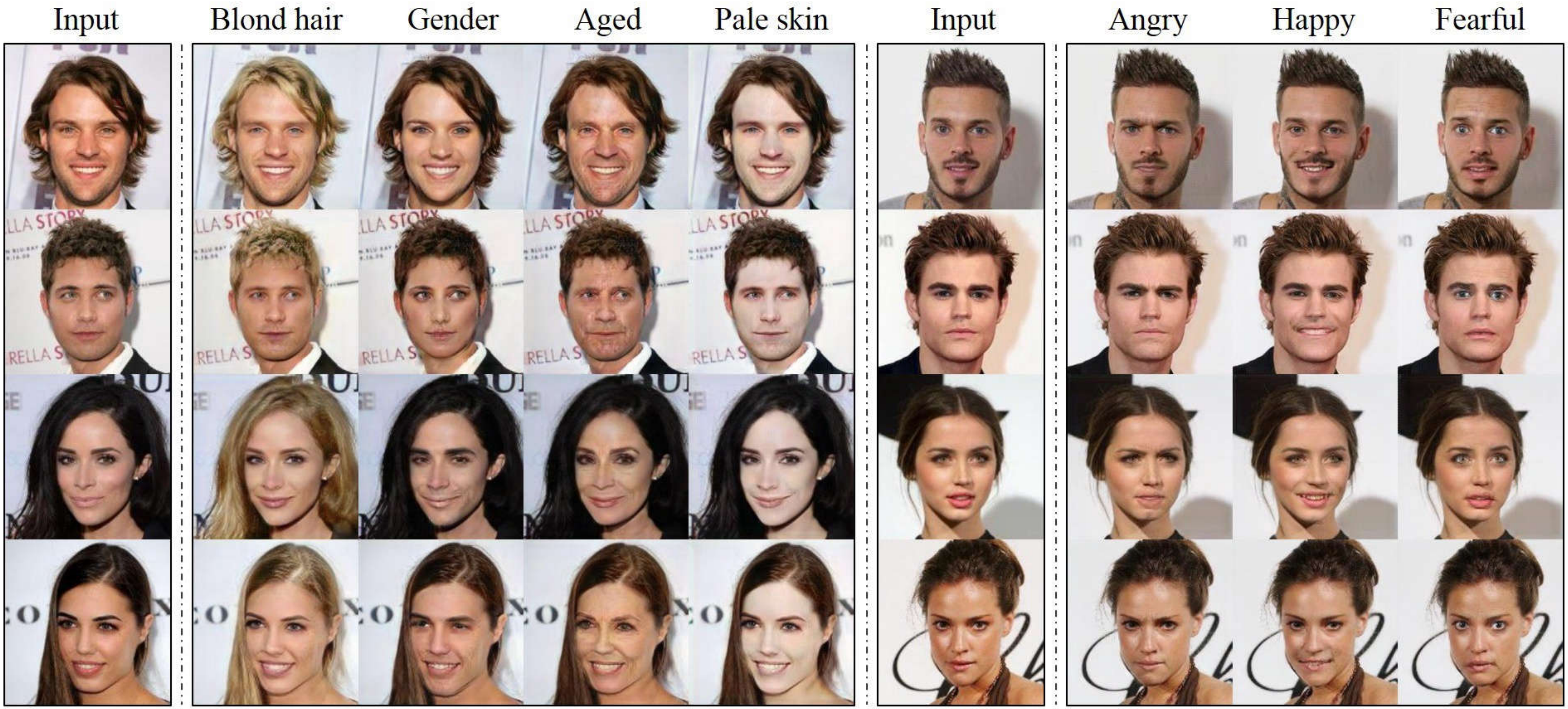
Cycle GAN



$$min_{G_{A\to B},G_{B\to A}} max_{D_A,D_B} \quad L_{GAN}(G_{A\to B}) + L_{GAN}(G_{B\to A}) \; + \mathbb{E}_{x\backsim p_{d_A}} \overbrace{\|G_{A\to B}(G_{B\to A}(x)) - x\|}^{\text{Cycle consistency on domain A}}$$

$$+ \mathbb{E}_{x\backsim p_{d_B}} \underbrace{\|G_{B\to A}(G_{A\to B}(x)) - x\|}_{\text{Cycle consistency on domain B}}$$


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


Input
Blond hair
Gender
Aged
Pale skin
Input
Angry
Happy
Fearful

Other applications

- Semi-supervised learning
- Anomaly detection
- Representation learning
- Model interpretation
- Transfer learning
- Security of predictive models