Introduction of
Generative Adversarial Networks

Hao Dong

2019

Peking University

Generative Models vs. Computer Graphics

- Generate data (e.g., image) in computer

Computer Graphics

Ball(color=yellow, position=(50, 100),…)
Ball(color=red, position=(30, 75),…)
Ball(color=blue, position=(30, 125),…)
…
…

Computer Vision

“description”

“observation”

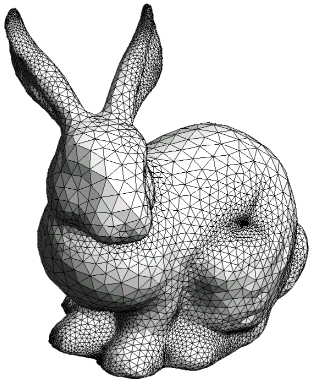
Generative Models vs. Computer Graphics

- **Statistical Generative Models** are **data-driven** methods

Computer Graphics | **Statistical Generative Models**

Prior Knowledge | **Data**

Material, Physical Modeling, Lighting ….



Generative Models vs. Computer Graphics

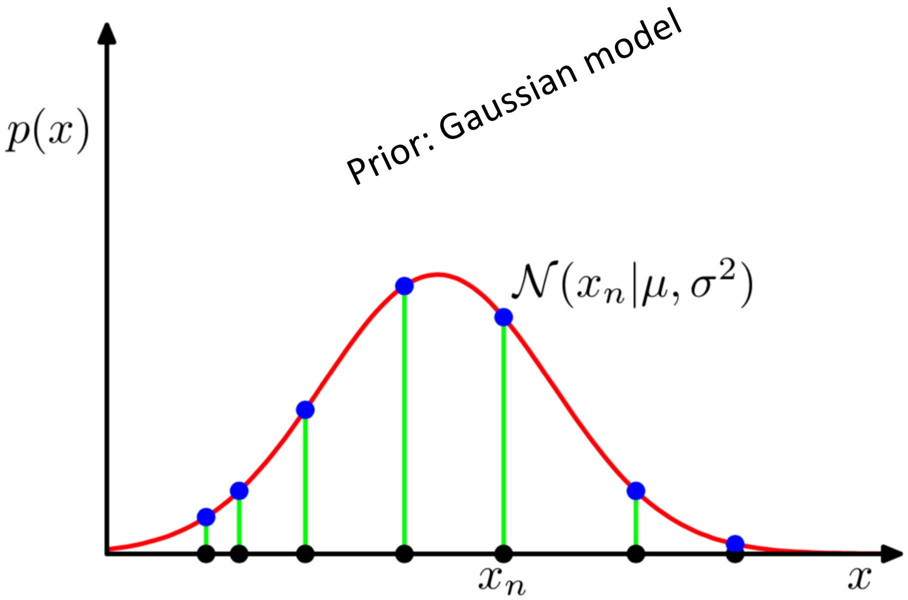
- **Computer Graphics**
  - Purely based on prior knowledge
  - Difficult to scale and generalize
  - Development is time-consuming

- **Machine Learning/Deep Learning**
  - Reduce the need of prior knowledge
  - Learn from data

- **Statistical/Deep Generative Models** still need some prior knowledge …
  - loss function, learning method, architecture, prior distribution (e.g., Gaussian)

Generative Models vs. Computer Graphics

- **Statistical/Deep Generative Models**



- Given data samples
- Learn the probability distribution $p(x)$

So that

- It is generative because new data samples can be sampled from $p(x)$

$$x_{new} \sim p_x$$

Generative Models vs. Computer Graphics

- **Statistical/Deep Generative Models**

The data distribution can be high-dimensional, like images

- Given data samples
- Learn the probability distribution $p(x)$

So that

- It is generative because new data samples can be sampled from $p(x)$

$$x_{new} \sim p_x$$

Data Representation

p_{data}

$$\mathbf{x}^j \sim p_{data}$$
$$j = 1,2, \dots |\mathcal{D}|$$

- dataset $\mathcal{D}$
- data distribution p_{data}
- model parameters $\theta \in \mathcal{M}$

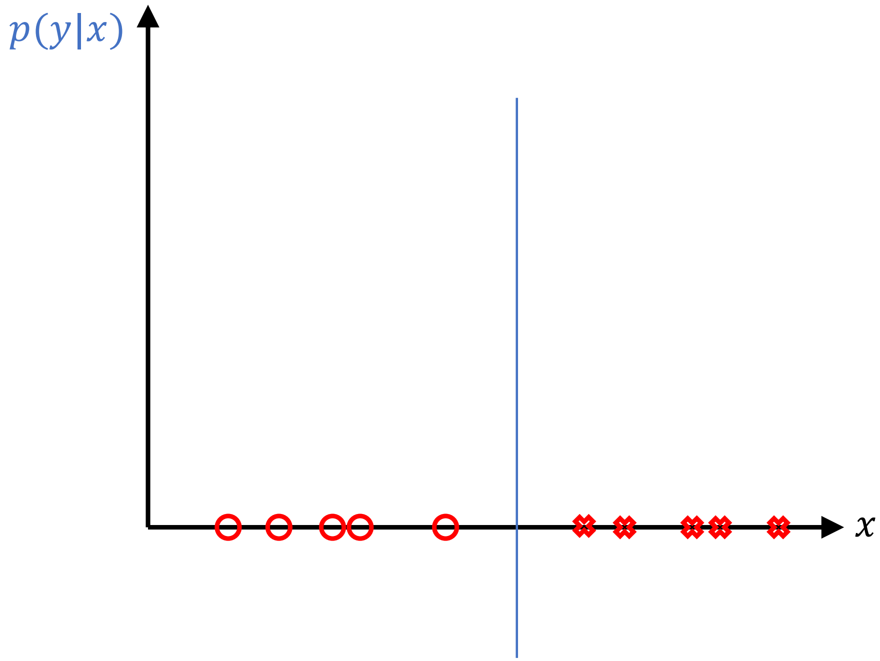
- **We want to learn a probability distribution $p(\mathrm{x})$ over $\mathbf{x}$**

1. **Generation (sampling):** $\mathbf{x}_{new} \sim p(\mathrm{x})$
2. **Density Estimation:** $p(\mathrm{x})$ high if $\mathbf{x}$ looks like a cat
3. **Unsupervised Representation Learning:**
   Discovering the underlying structure from the data distribution (e.g., ears, nose, eyes …)

Discriminative vs. Generative

Discriminative models: classify data

finding the **decision boundary** $P(Y|X)$

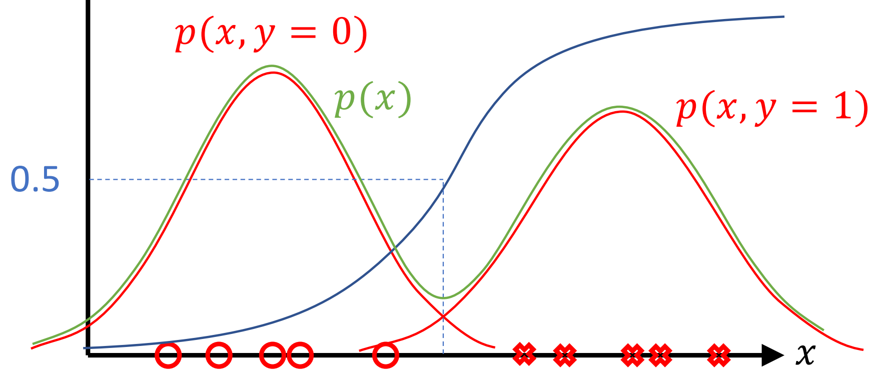


Generative models: generate data

finding **joint distribution** $P(Y, X)$

$$Conditional = \frac{Joint}{Marginal}$$

$$p(y=1|x) = \frac{p(x, y=1)}{p(x)}$$



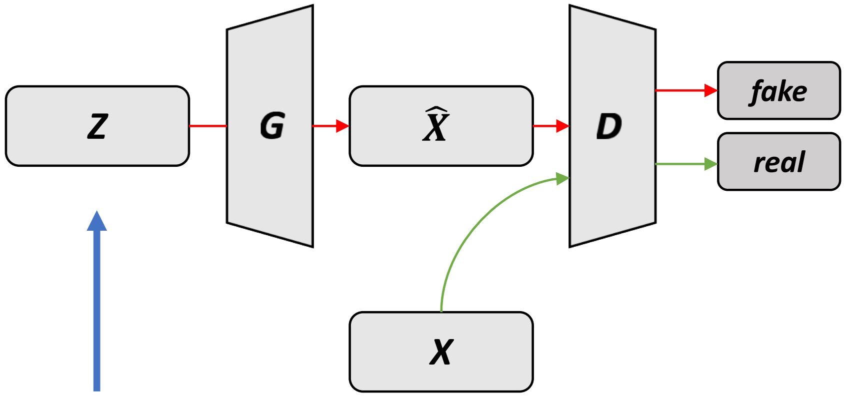
Note: Generative models can perform both generative and discriminative tasks

Introduction of Generative Adversarial Networks (GAN)

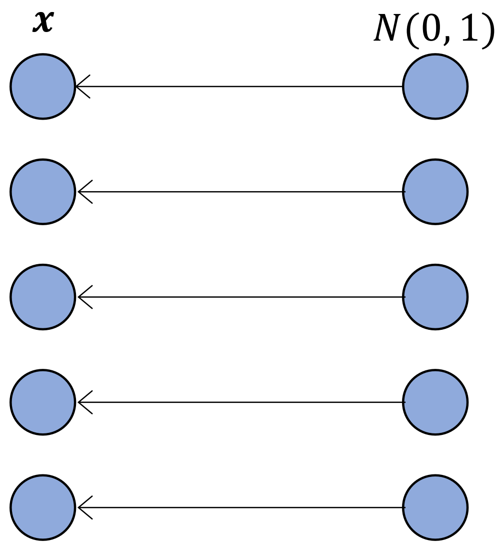
- Vanilla GAN
- GAN with Encoder
- Summary

Vanilla GAN

Vanilla GAN



Normal/~~uniform~~
distribution



Unidirectional Mapping

GAN: map a distribution to another distribution

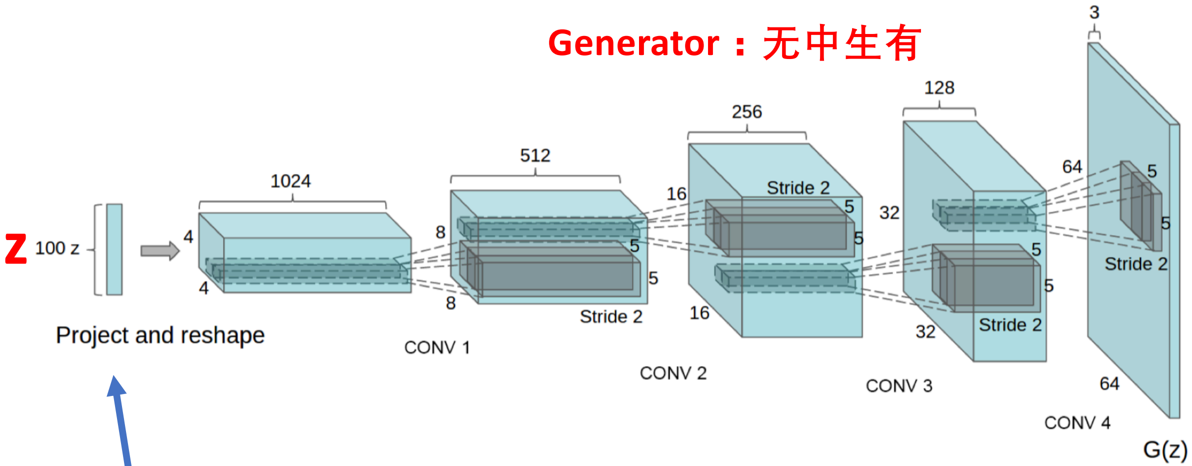
$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{data}}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_z}[\log(1 - D(G(\boldsymbol{z}))]$$

$$\mathcal{L}_D = -\boxed{\mathbb{E}_{\boldsymbol{x}\sim p_{data}}[\log D(\boldsymbol{x})]} - \boxed{\mathbb{E}_{\boldsymbol{z}\sim p_z}[\log(1 - D(G(\boldsymbol{z}))]}$$

$$\mathcal{L}_G = -\boxed{\mathbb{E}_{\boldsymbol{z}\sim p_z}[\log D(G(\boldsymbol{z}))]}$$

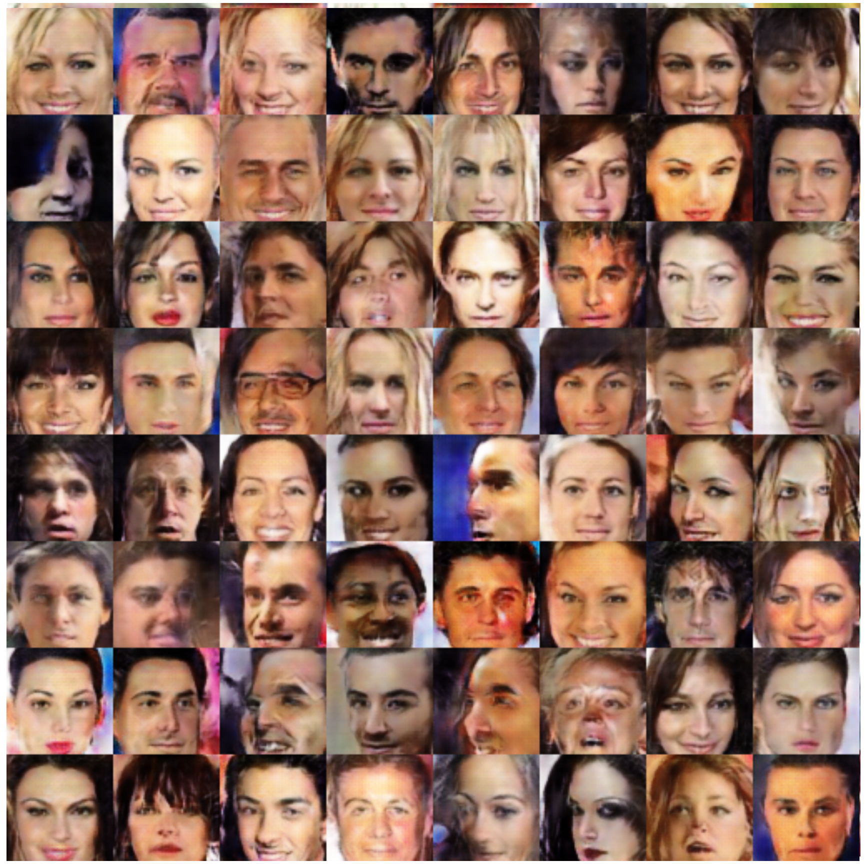
Vanilla GAN – Deep Convolutional GAN (DCGAN)

- Using the power of CNN



Normal distribution
z = 100 values

$\widehat{X}$



Vanilla GAN – Deep Convolutional GAN (DCGAN)

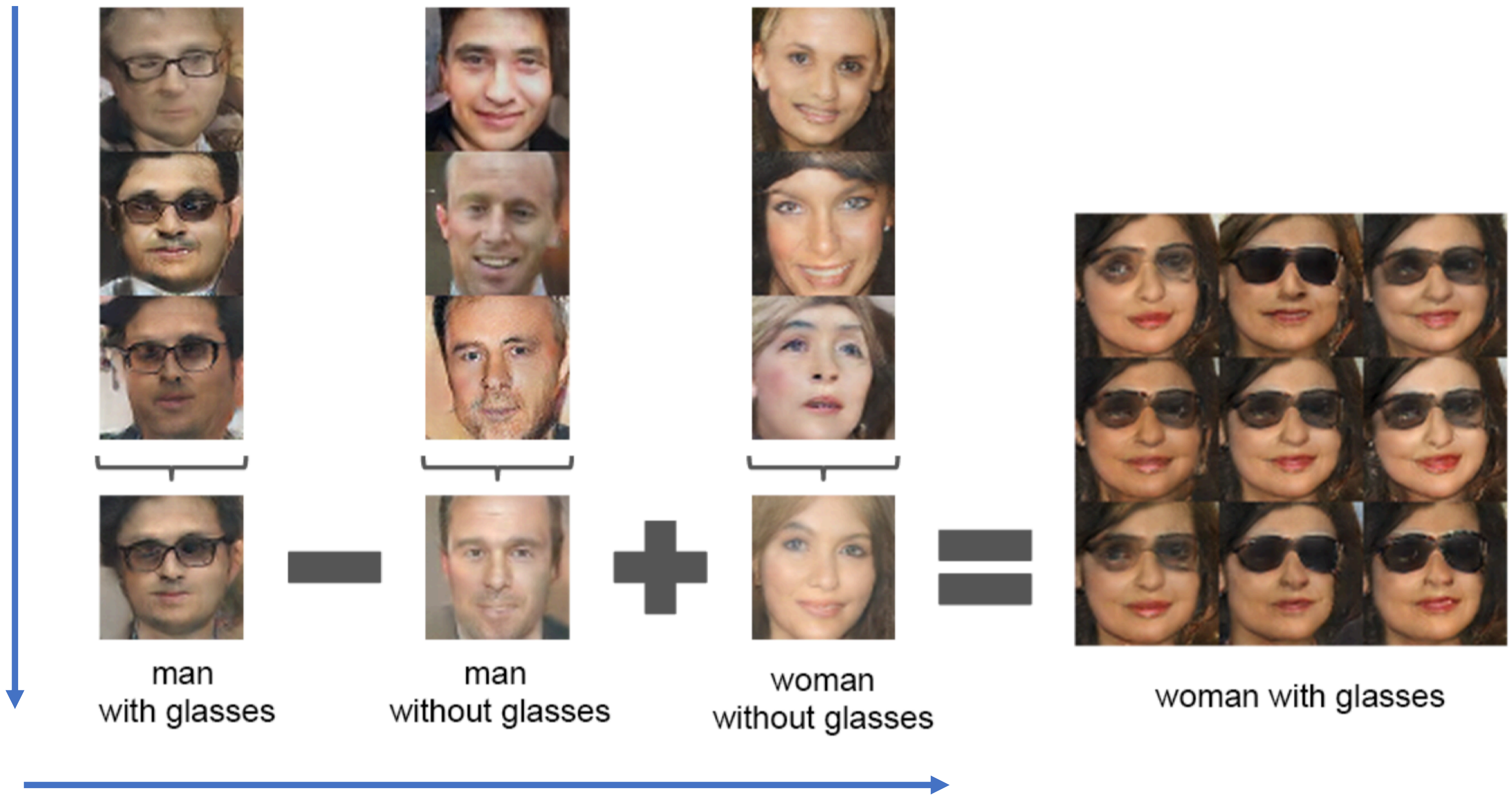
- Latent representation z

Interpolation

Noise vector Z_A → **Noise vector Z_B**

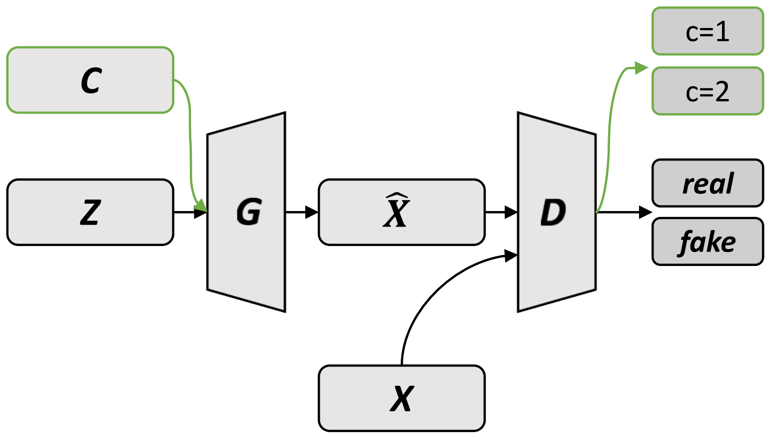
Vanilla GAN – Deep Convolutional GAN (DCGAN)

- Latent representation z



Vanilla GAN -- Conditional GAN

- Auxiliary Classifier GANs



$$\mathcal{L}_D = \mathbb{E}_{x\sim p_{data}}[logD_x(x)] + \mathbb{E}_{z\sim p_z}[log(1 - D_x(G(z,c)))]$$

$$\mathbb{E}_{x\sim p_{data}}[logD_c(x)] + \mathbb{E}_{z\sim p_z}[log(1 - D_c(G(z,c)))]$$

$$\mathcal{L}_G = \mathbb{E}_{x\sim p_{data}}[logD_x(G(z,c))] + \mathbb{E}_{z\sim p_z}[logD_c(G(z,c))]$$

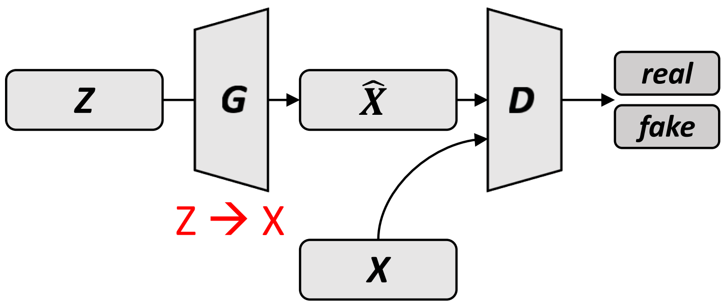
monarch butterfly | goldfinch | daisy

Multi-modal problem: one problem has multiple solutions
P(z, c)

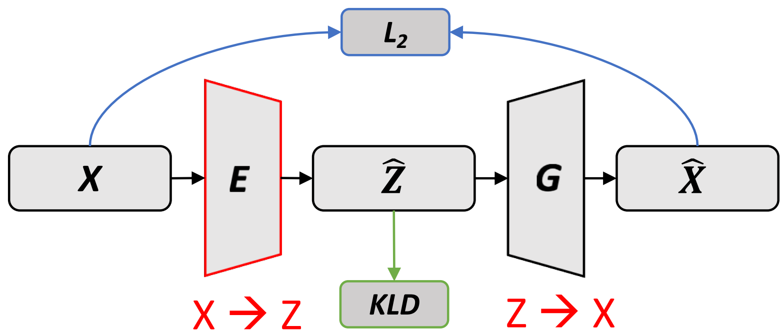
GAN with Encoder

GAN with Encoder – Vanilla GAN vs VAE

Vanilla GAN

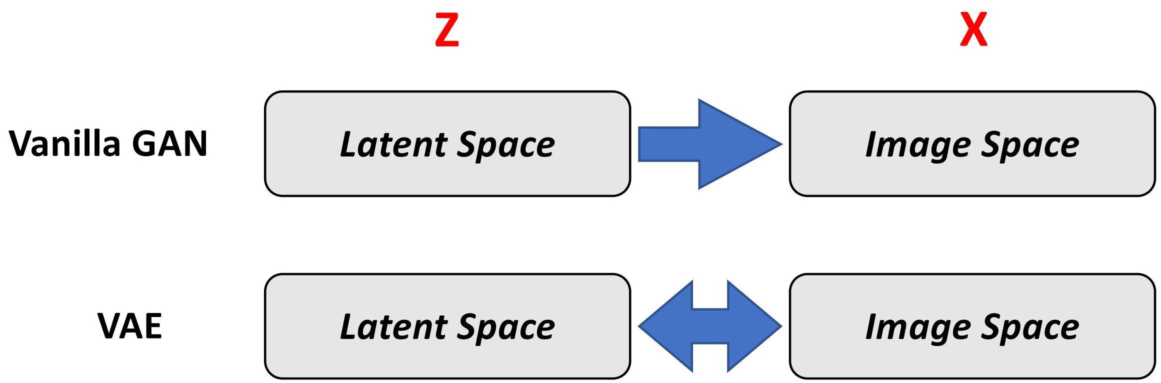


VAE variational autoencoder



VAE has an Encoder that can map x to z

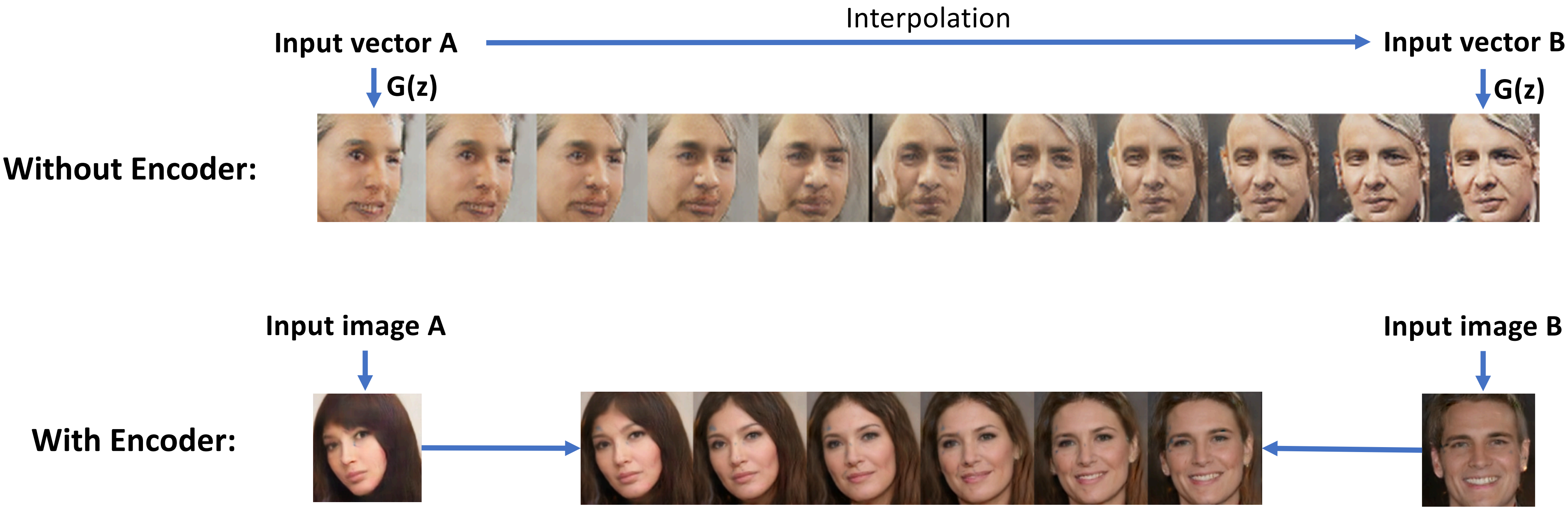
GAN with Encoder – Vanilla GAN vs VAE



- VAE = **G**enerator + **E**ncoder
- Vanilla GAN = **G**enerator + **D**iscriminator
- Better GAN = **G**enerator + **D**iscriminator + **Encoder**

GAN with Encoder – Why Encoder

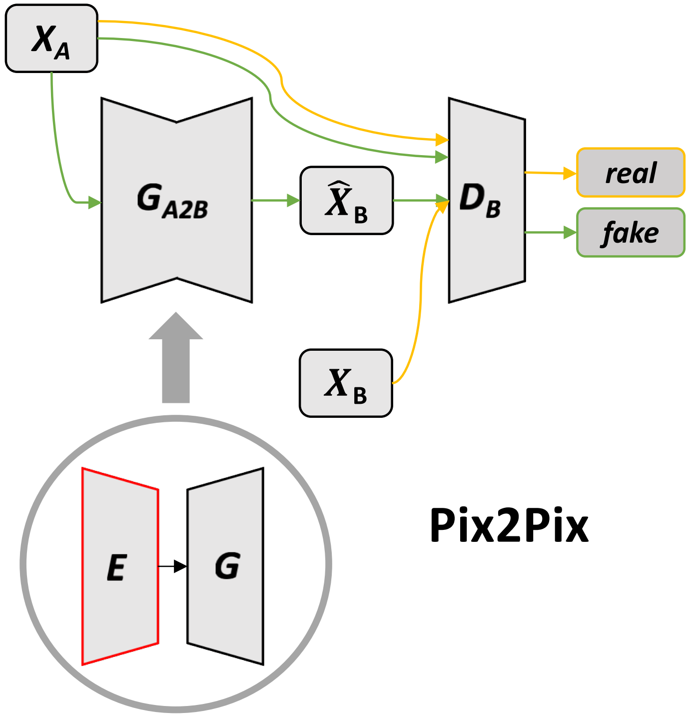
- Encoder allows GAN to receive images == More applications



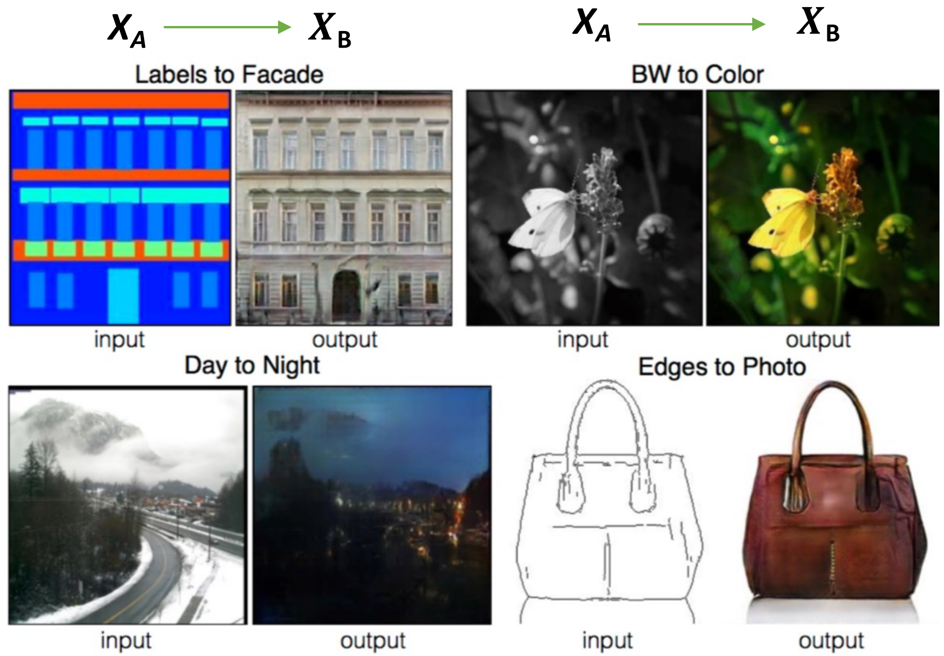
GAN with Encoder – Encoder as a part of the Generator

- Supervised image-to-image “translation”

Inputs can be images



Pix2Pix



$$\mathcal{L}_D = \boxed{\mathbb{E}_{x\sim p_{data}}[logD(x_A, x_B)]} + \boxed{\mathbb{E}_{x\sim p_{data}}[log(1 - D(x_A, G(x_A)))]}$$

$$\mathcal{L}_G = \boxed{\mathbb{E}_{x\sim p_{data}}[logD(x_A, G(x_A))]}$$

Image-to-Image Translation with Conditional Adversarial Networks. *P. Isola, J. Zhu et al. CVPR 2017.*

GAN with Encoder – Encoder as a part of the Generator

- Supervised image-to-image "translation"

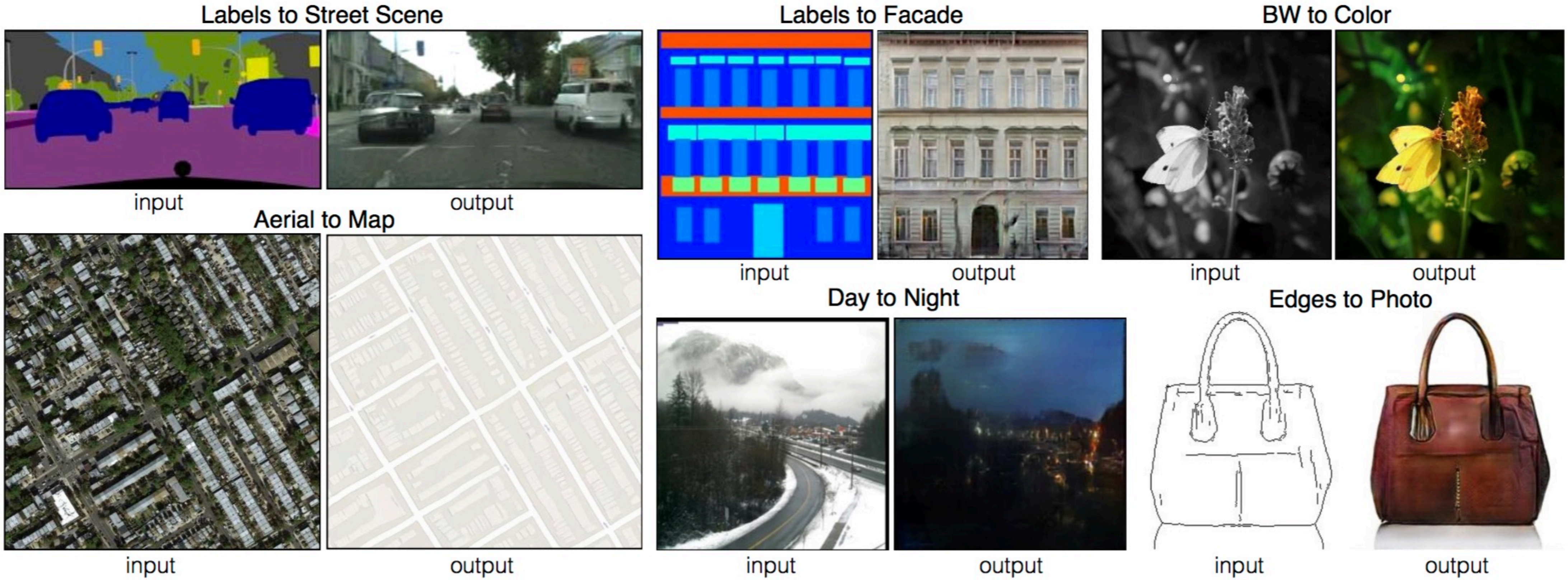


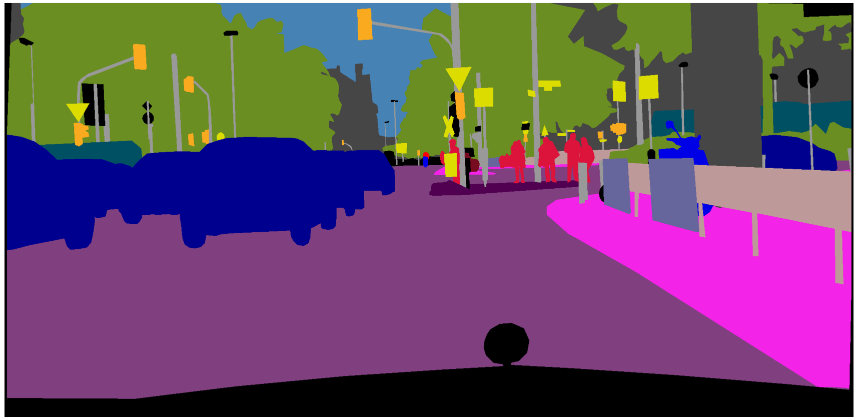
Image-to-Image Translation with Conditional Adversarial Networks. *P. Isola, J. Zhu et al. CVPR 2017.*

GAN with Encoder – Encoder as a part of the Generator

- Supervised image-to-image "translation"

Input Output

Input labels



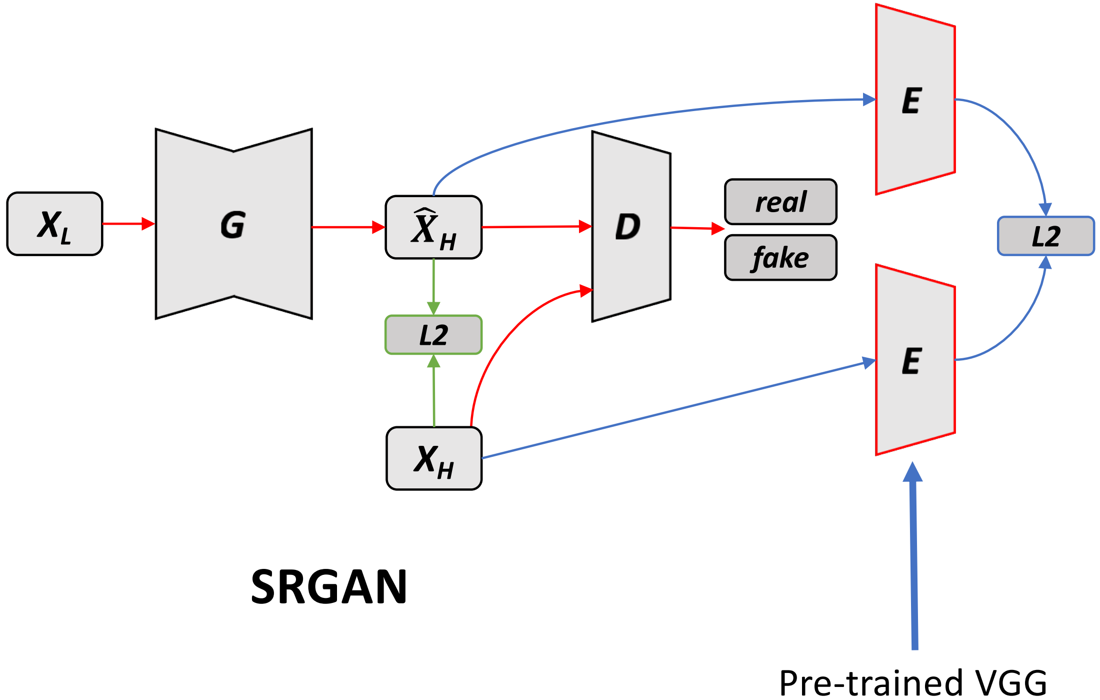
Synthesized image

Image-to-Image Translation with Conditional Adversarial Networks. *P. Isola, J. Zhu et al. CVPR 2017.*

GAN with Encoder – Encoder as the Feature Extractor

- Supervised image super resolution

Better feature reconstruction



SRGAN

Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network. *C. Ledig, L. Theis et al. CVPR 2017.*

GAN with Encoder – Encoder as the Feature Extractor

- Supervised image super resolution

Input Output

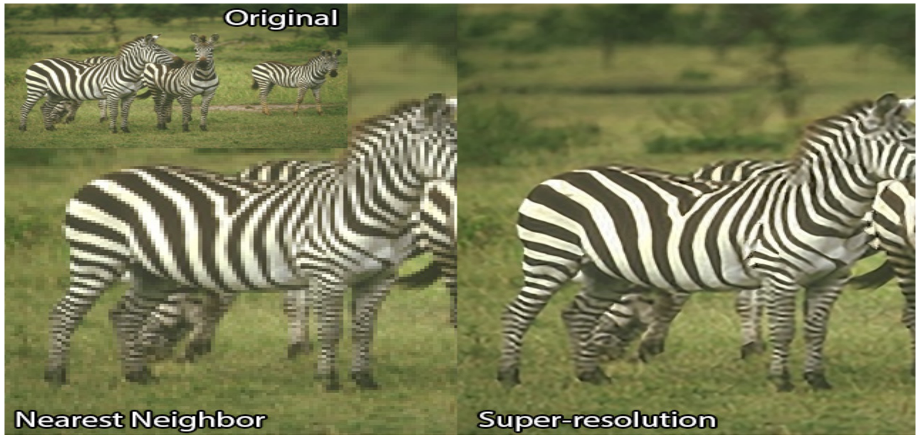
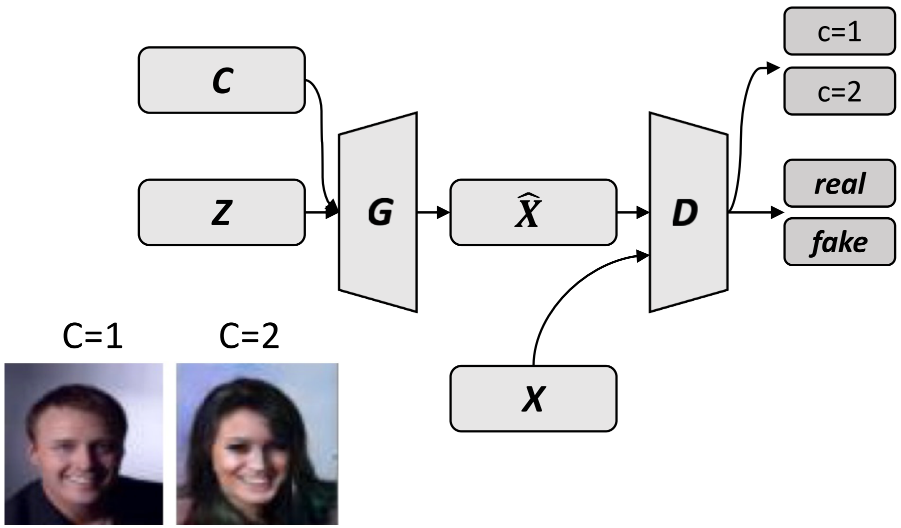


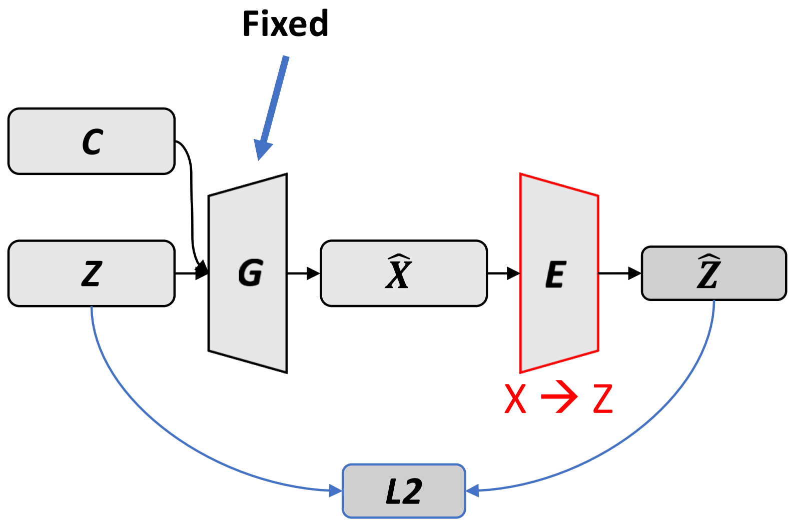
Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network. *C. Ledig, L. Theis et al. CVPR 2017.*

GAN with Encoder – Learn the mapping from x to z like VAE

- Unsupervised image-to-image translation



Given an ACGAN

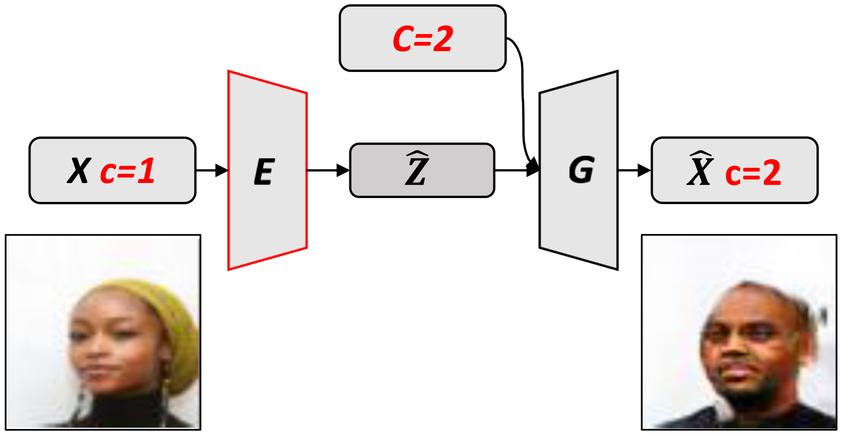


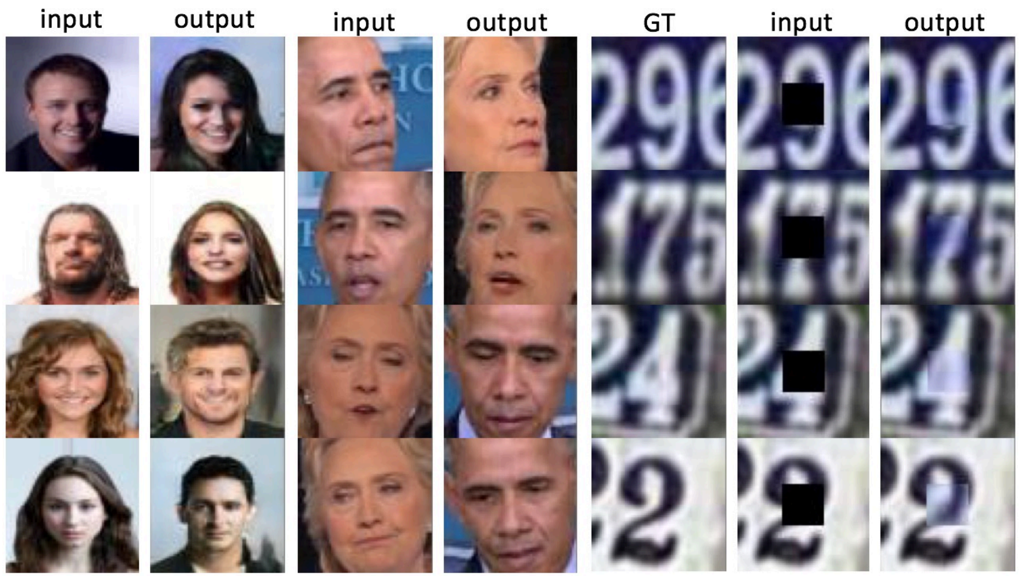
Learning the Encoder in a Brute Force Way

Unsupervised Image-to-Image Translation with Generative Adversarial Networks. *H. Dong, P. Neekhara et al. arXiv 2017.*

GAN with Encoder – Learn the mapping from x to z

- Unsupervised image-to-image translation





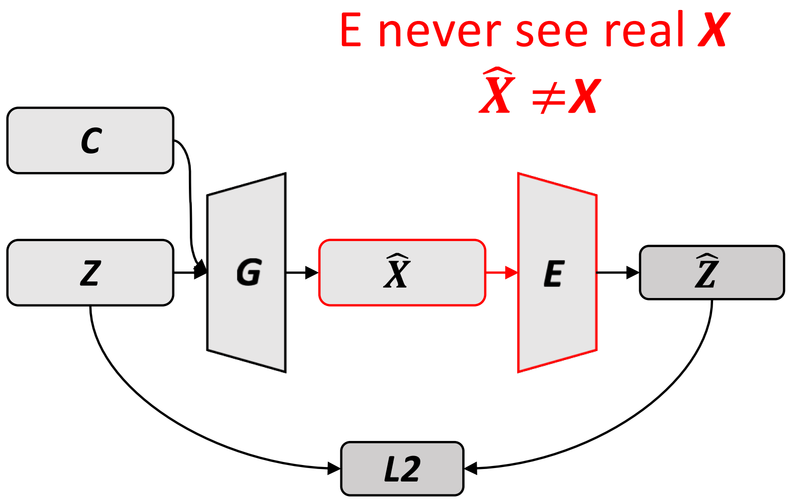
Gender transformation

Face swapping

Image inpainting

GAN with Encoder – Learn the mapping from x to z

- Limitation of the brute force method : Encoder never see real data samples

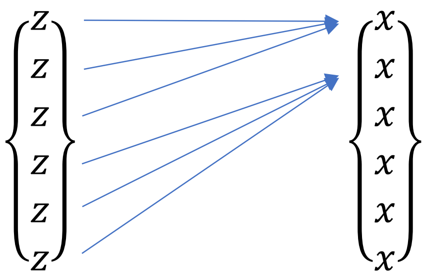


GAN with Encoder – Learn the mapping from x to z

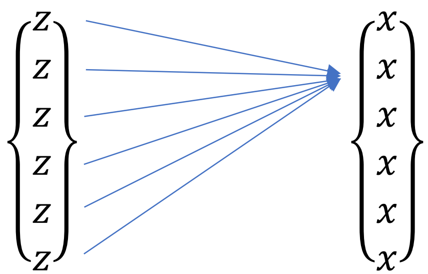
- Limitation of the brute force method : Mode Collapse

E never see real $\boldsymbol{X}$

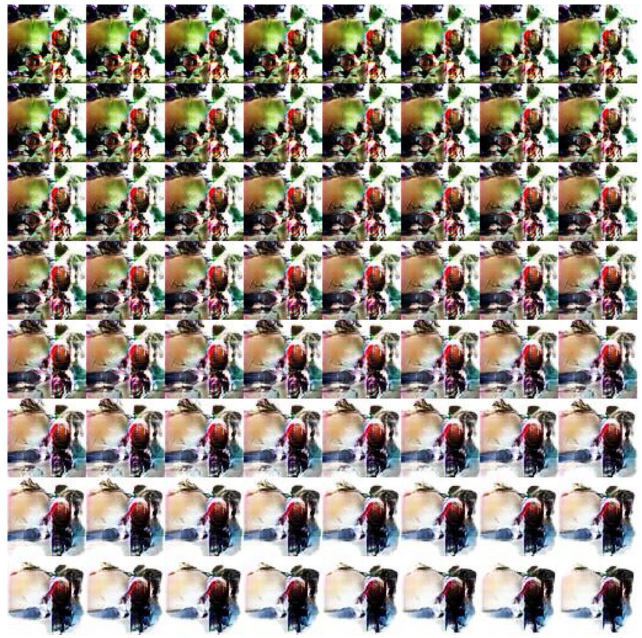
$\widehat{\boldsymbol{X}} \neq \boldsymbol{X}$



G can only synthesis some part of the dataset x



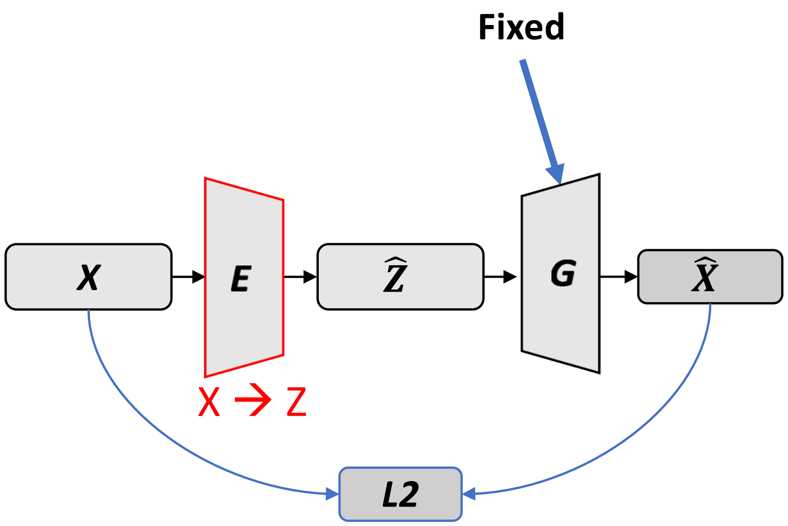
G can only synthesis one data



Examples of GAN collapse

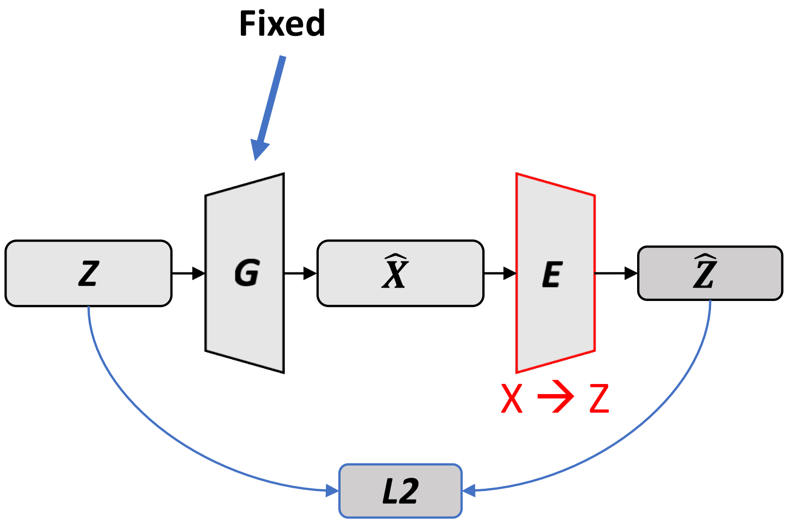
GAN with Encoder – Learn the mapping from x to z

- Limitation of the brute force method : Encoder never see real data samples

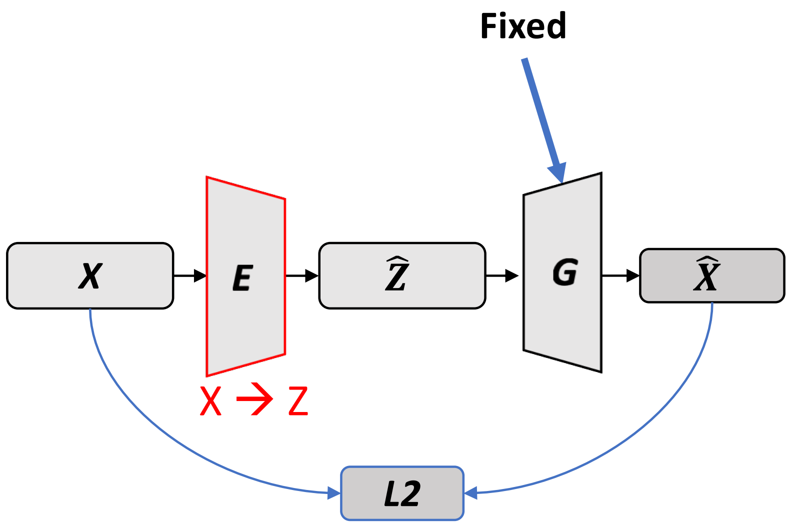


GAN with Encoder – Learn the mapping from x to z

- Only work well if only if the fake distribution == the real distribution, but it is impossible in practice.



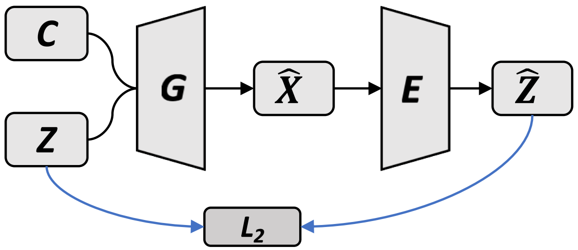
Method 1



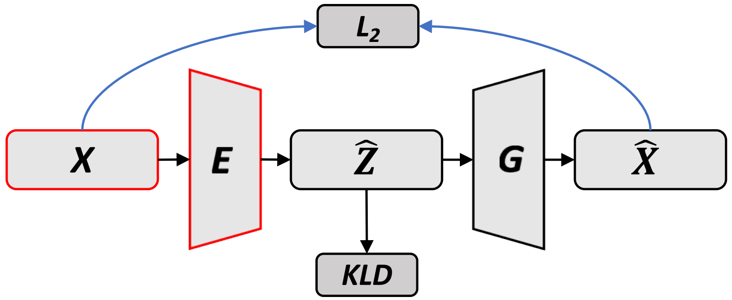
Method 2

GAN with Encoder -- Learn the Encoder from Real Data

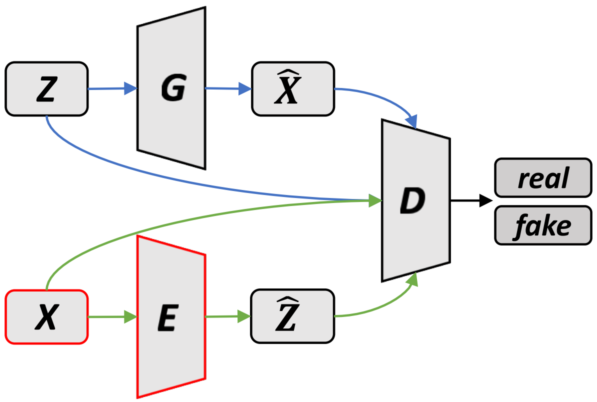
- Learn the Encoder



Brute Force

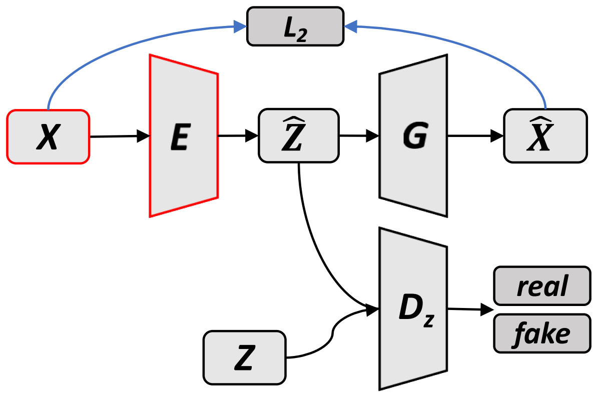


VAE



BiGAN

Bidirectional GAN



AAE

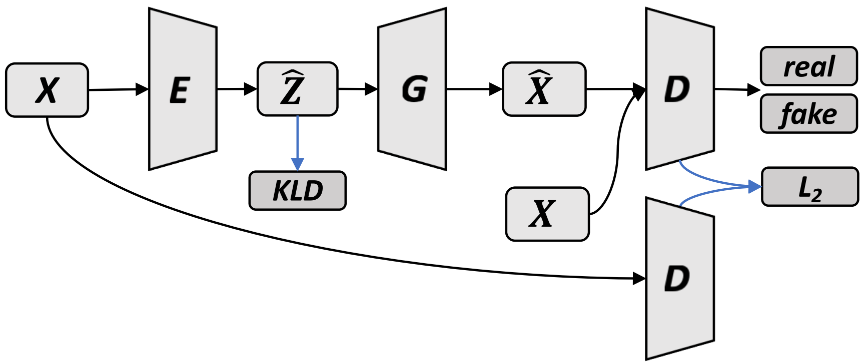
Adversarial Autoencoder

GAN with Encoder -- Learn the Encoder from Real Data

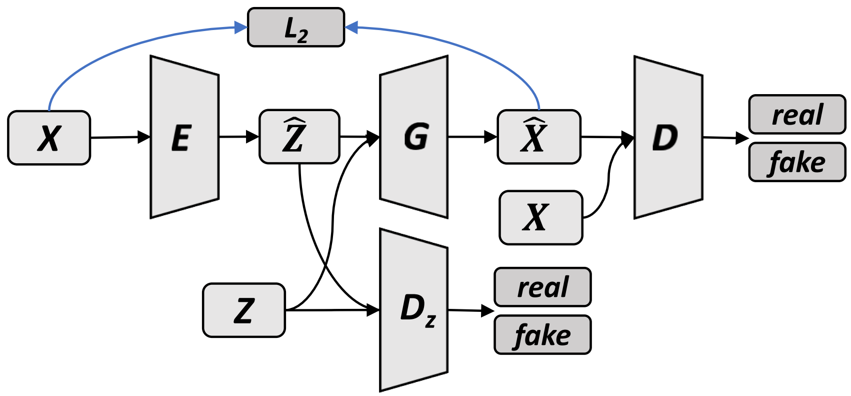
北京大學 PEKING UNIVERSITY

- Learn the Encoder

Avoid GAN collapse



VAE+GAN

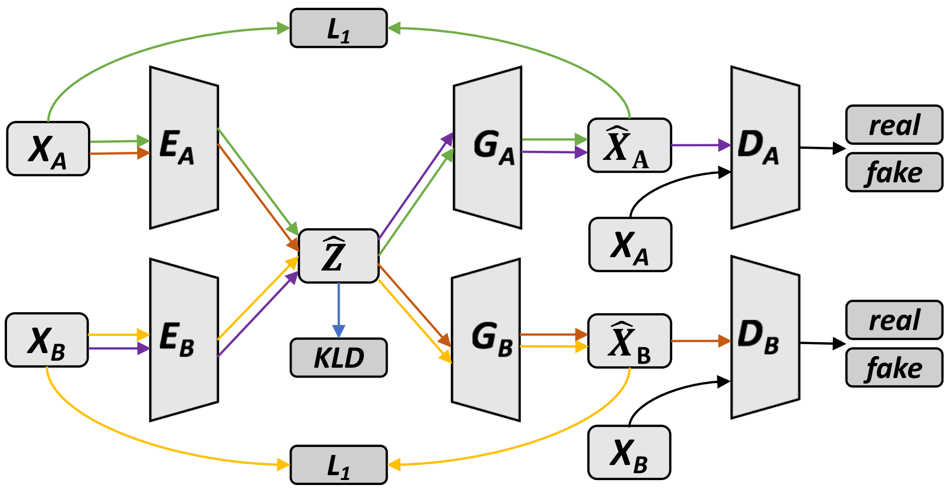


α-GAN

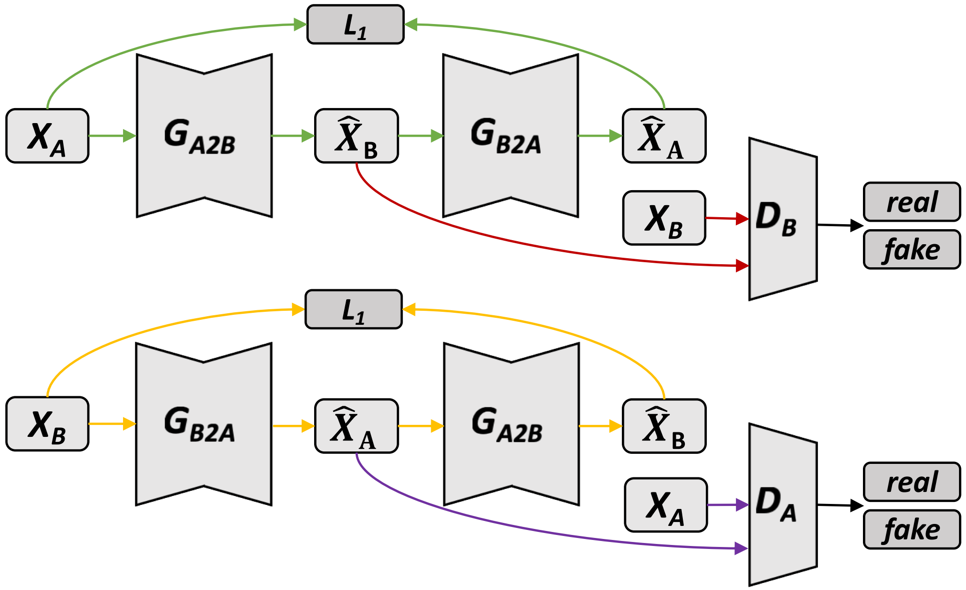
Discriminator as the feature extractor

- Training the *G* and *E* in Autoencoder way can force the G to be able to generate all *X, avoiding GAN collapse*

GAN with Encoder -- Unsupervised Image-to-Image Translation



UNIT
Learn the Encoder **Explicitly**

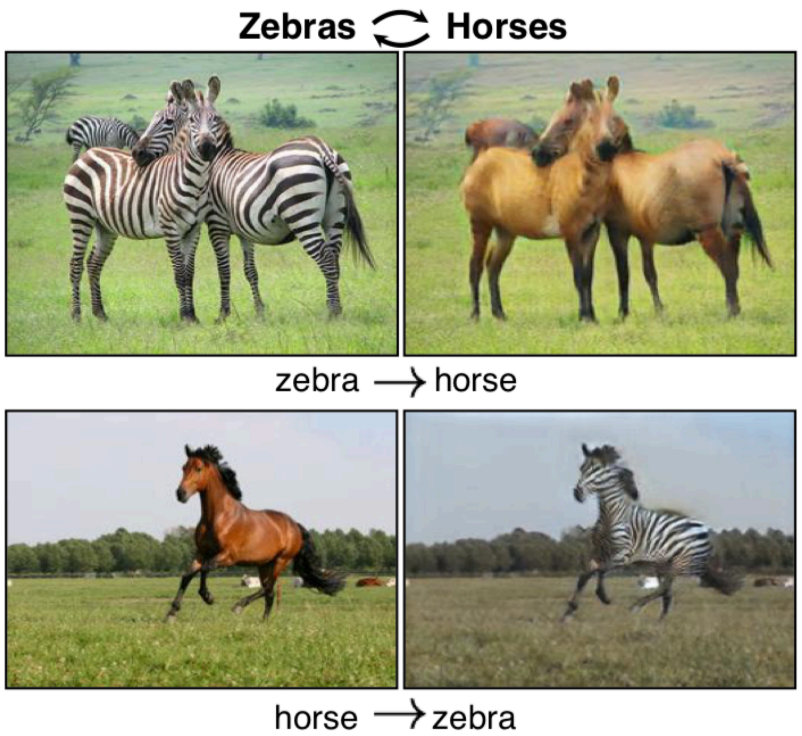


CycleGAN
Learn the Encoder **Implicitly**

Unsupervised image-to-image translation networks. *M.Y. Liu, T. Breuel, J. Kautz. NIPS. 2017*
Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. *J. Zhu, T. Park et al. ICCV 2017.*

GAN with Encoder -- Unsupervised Image-to-Image Translation

Liu et al.
Learn the Encoder **Explicitly**



CycleGAN
Learn the Encoder **Implicitly**

Unsupervised image-to-image translation networks. *M.Y. Liu, T. Breuel, J. Kautz. NIPS. 2017*
Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. *J. Zhu, T. Park et al. ICCV 2017.*

GAN with Encoder -- Unsupervised Image-to-Image Translation

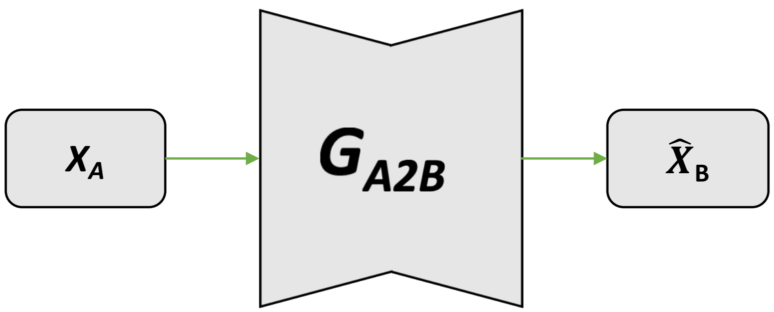
Input GTA5 CG

Output image with German street view style

Unsupervised image-to-image translation networks. *M.Y. Liu, T. Breuel, J. Kautz. NIPS. 2017*
Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. *J. Zhu, T. Park et al. ICCV 2017.*

GAN with Encoder – Learn Encoder in Implicit or Explicit Ways?

- Simple normal distribution is difficult to model complex images
- 3D tensors can contain more spatial information than vectors
- Many applications do not need interpolation



- Image inpainting
- Image super resolution
- Image-to-image translation
- ….

Summary

PEKING UNIVERSITY

Summary

This talk:

- GAN : G + D → G + D + E
- Learning E from real data is important
- Autoencoder can help to avoid mode collapse
- Learning E implicitly is becoming more and more popular
- Explicit E is still important for representation learning
- The E can be extended to text and any other data type

GAN applications:

- Image-to-image: Pix2Pix→CycleGAN→Attention CycleGAN
- Text-to-image: GAN-CLS → StackGAN → StackGAN++
- Text+image to image: …
- Video-to-video: …

Questions

Questions

- Q1. 如何解决GAN中，输入的normal distribution太简单的问题？
- Q2. 为什么G、D要来回对抗训练，而不是完全训练好D后再训练G？
- Q3. G是不是真的能创造出训练数据中没有的数据？