AY 2024/2025

EE6222 Machine Vision

Video (Part I)

Cheng Jun

Institute for Infocomm Research, A*STAR

Address: 1 Fusionopolis Way, 138632

Email: cheng_jun@i2r.a-star.edu.sg; cheng.jun@ntu.edu.sg;

About me

Education:

- Bachelor: Electronic Engineering and Information Science, University of Science and Technology of China
- PhD: EEE, Nanyang Technological University

Research:

- Medical Image Computing
- Robotics
- Computer Vision
- Machine Learning

Project:

- Towards Realistic Deep Learning for 3D Vision", MTC Programmatic Fund, 2023-2026.
- Learning-based Approach to High Pitch Helical CT Reconstruction and Validation with Cardiac / abdominal CT scan, RCA with General Electric, 2023-2025
- AI-based OCT Angiography for Non-Invasive Capillary Blood Flow Imaging", A*STAR AI3 HTPO Seed Fund, 2023-2024

https://samjcheng.github.io/

Copyright Issues

- Some materials in the slides are copyrighted by respective publishers and authors.
- The lecture slides should be used for your own personal studies only.
- No part of these slides may be recorded, reproduced or distributed in any form or any means, without the prior written permission of the authors.

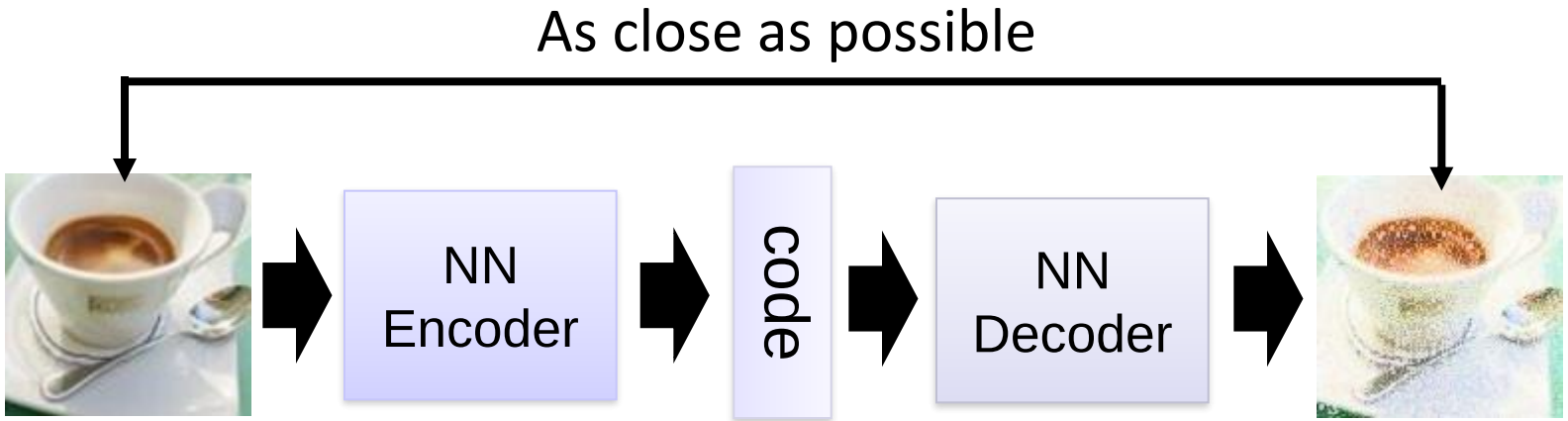
Content

- 1 Video: Generation
- 2 Video: Analysis/Enhancement
- 3 3D: Keypoint and Structure from Motion
- 4 3D: Stereo Matching and Others.

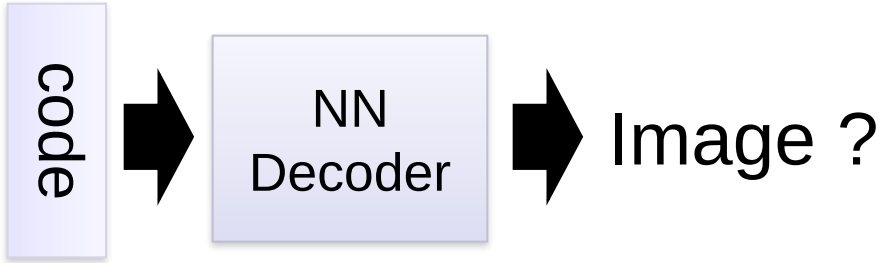
Video Generation


CLICK
BBC
NEWS
WH
.GOV

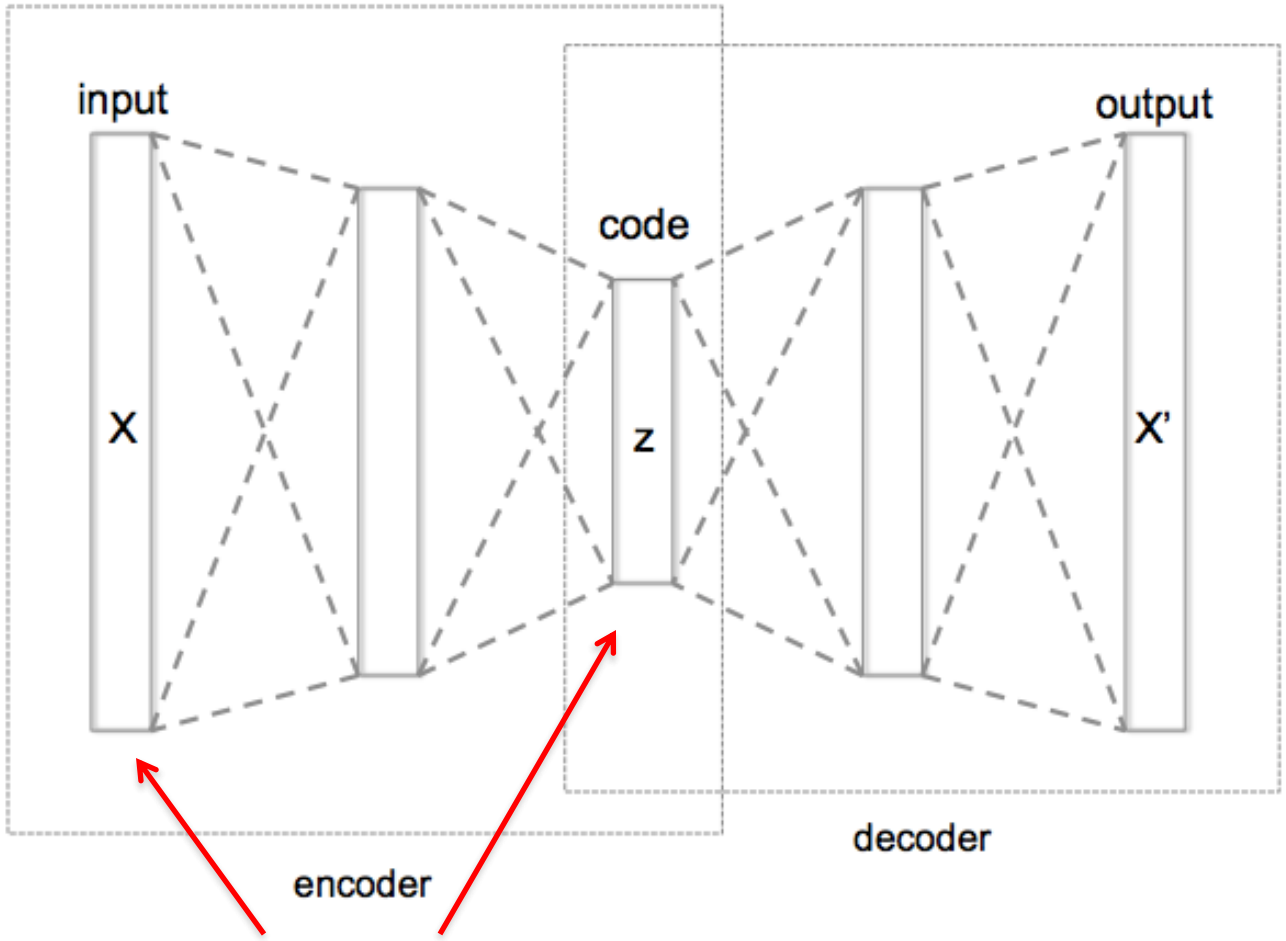
Autoencoder



Randomly generate a vector as code



Autoencoder with 3 fully connected layers

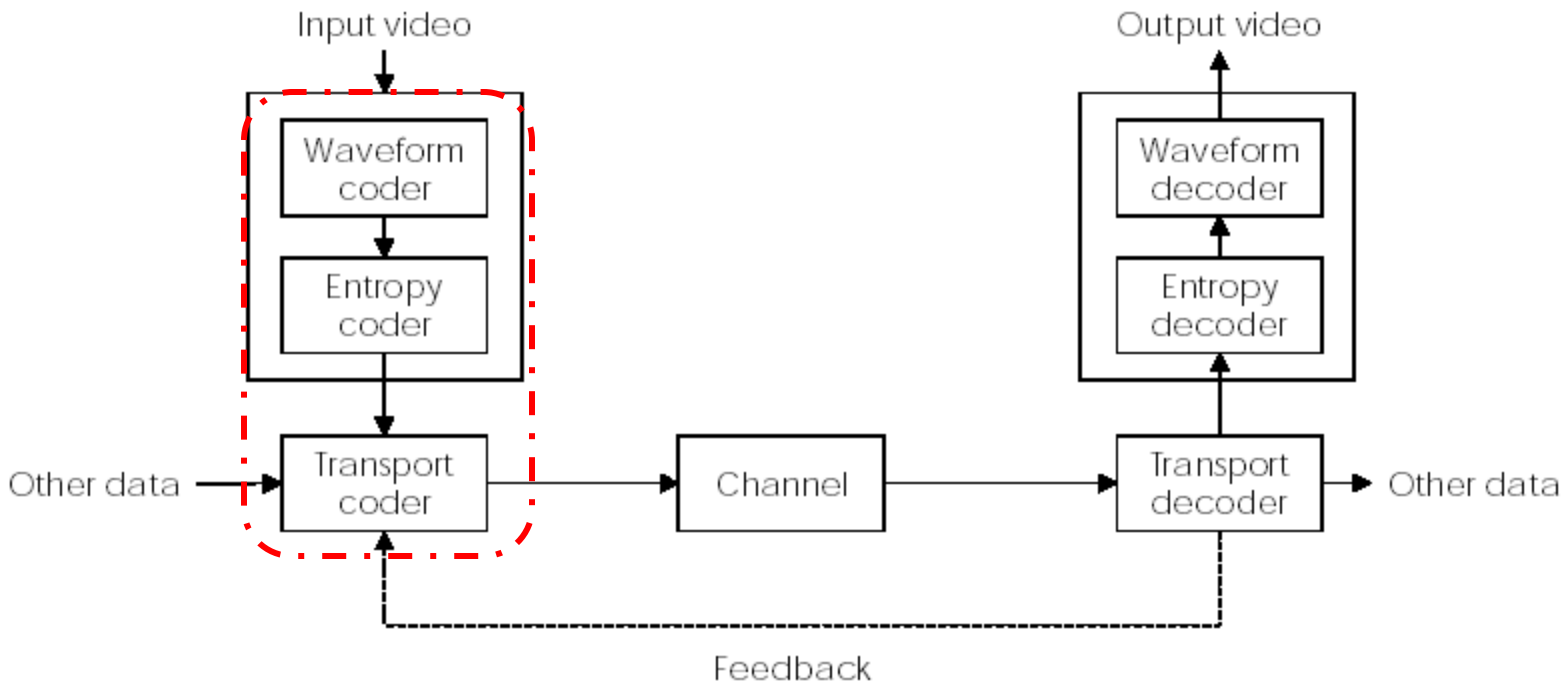


Large → small, learn to compress

Recall that

Video Communication System

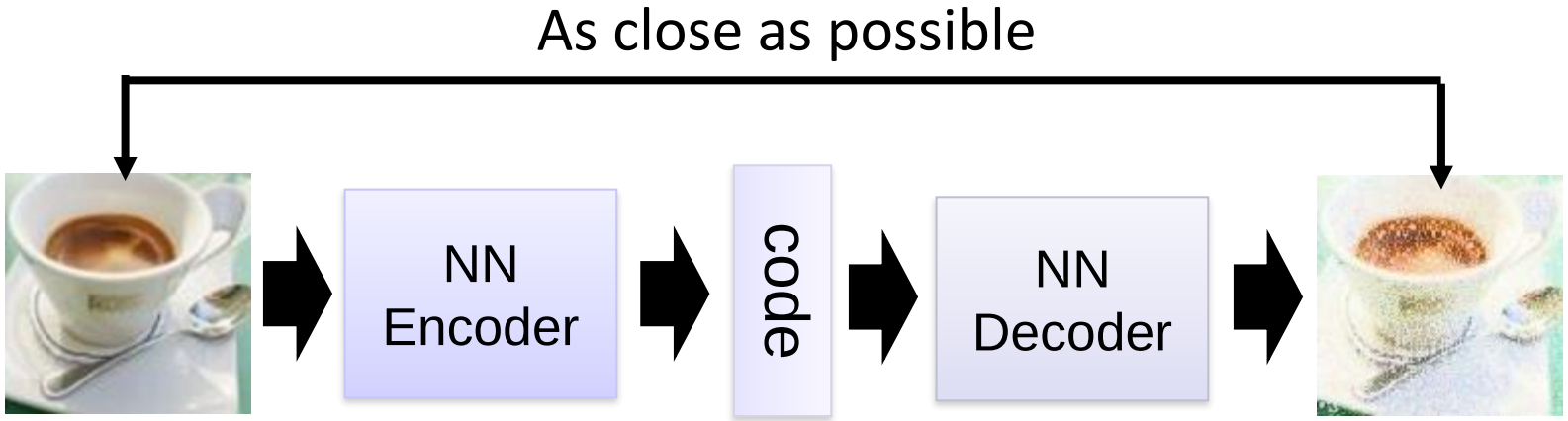
- A functional block diagram for a video communication



A quick test

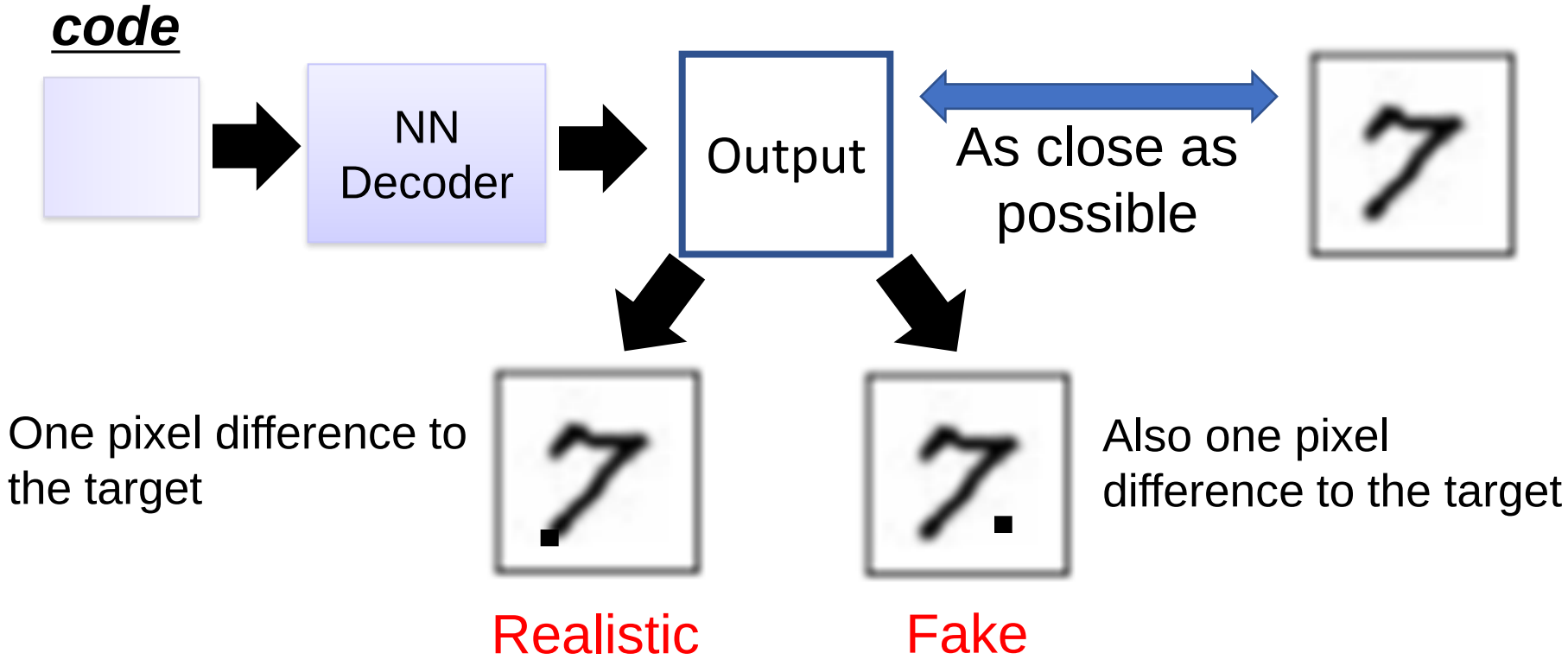
Why we can restore most of the information from the ‘code’

A. The original image has a lot of redundance
B. Trained auto-encoder contain domain knowledge on how to reconstruct the data.

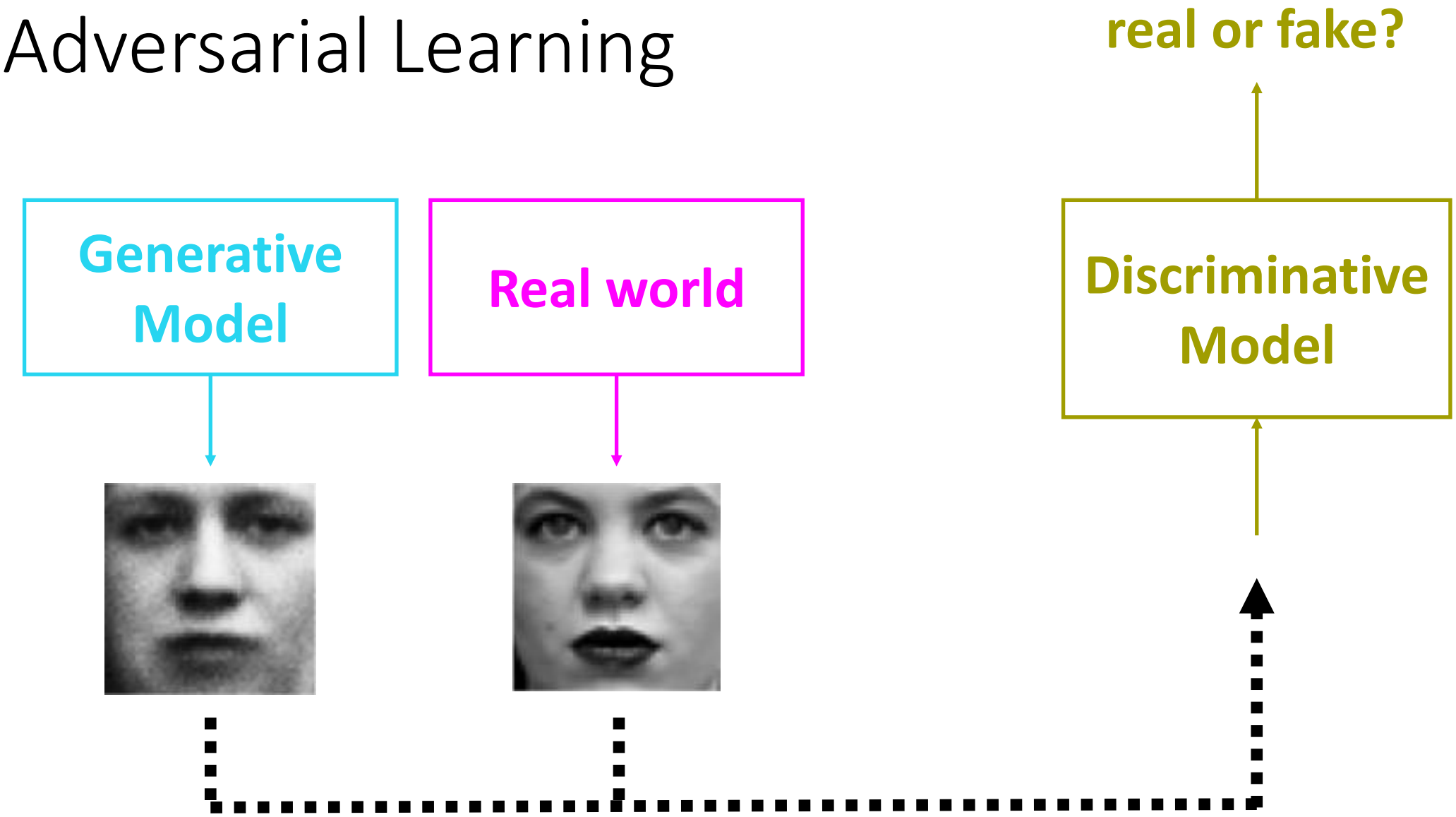


Problems of Autoencoder

- It does not really try to simulate real images



Adversarial Learning

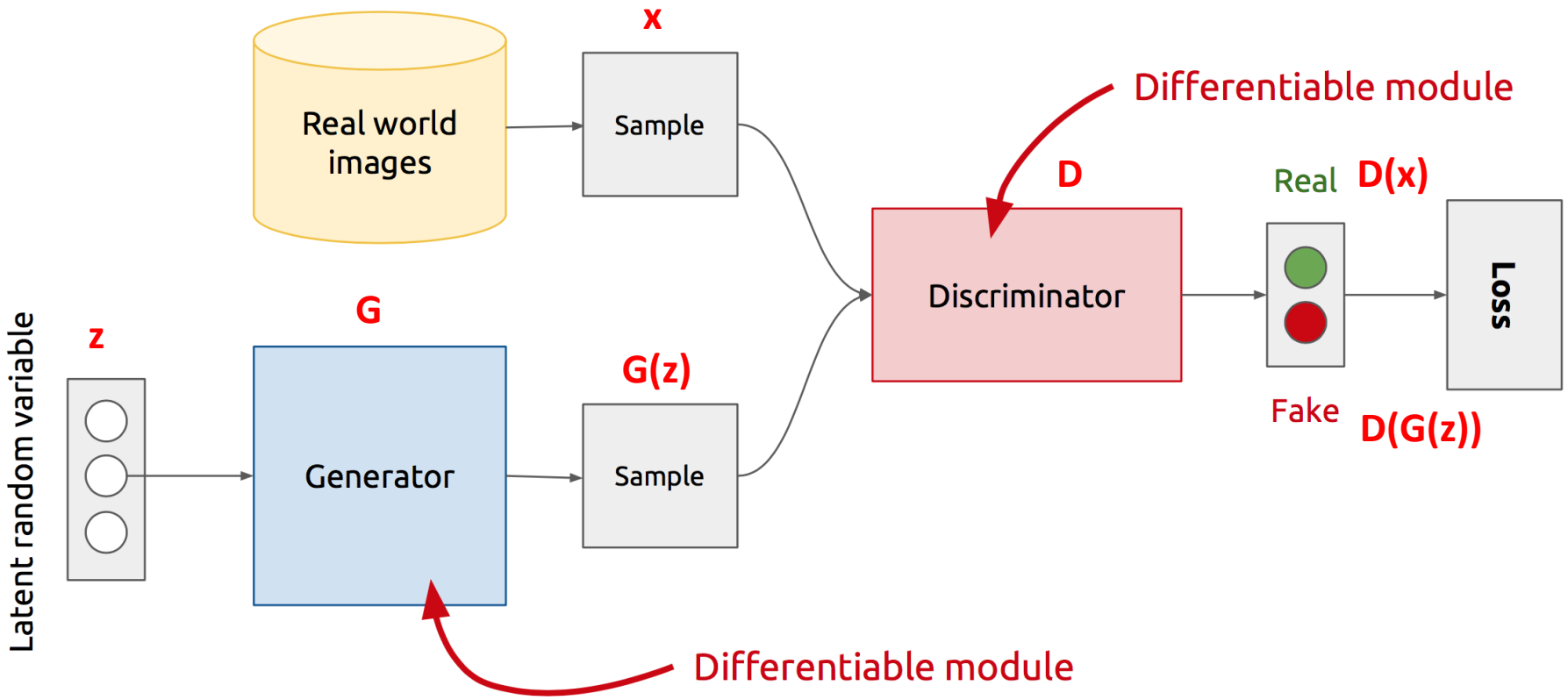


Generative Adversarial Networks (GANs)

- **Generative**
  - Learn a generative model
- **Adversarial**
  - Trained in an adversarial setting
- **Networks**
  - Use Deep Neural Networks

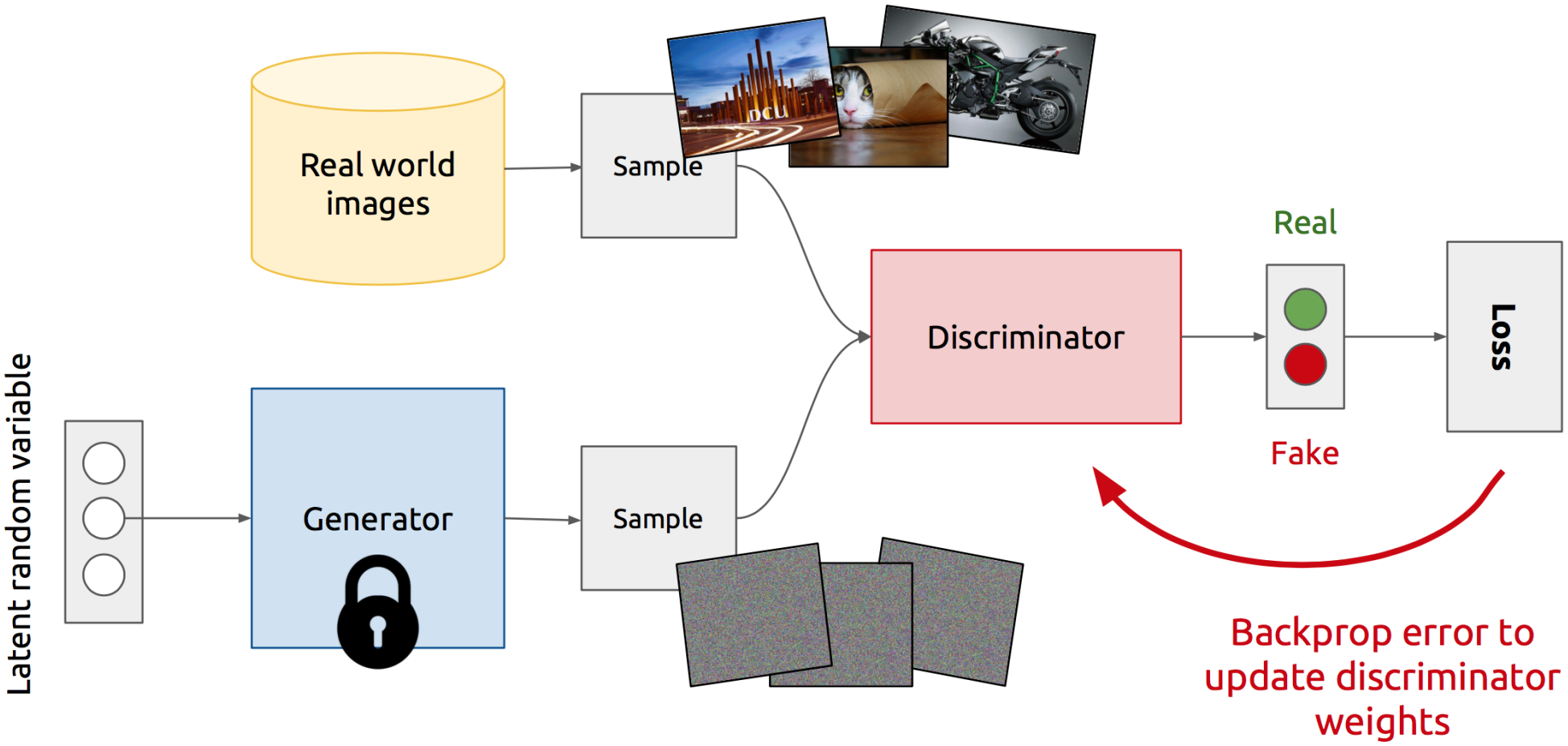
"The most interesting idea in the last ten years in machine learning" LeCun Yann

Structure of GANs

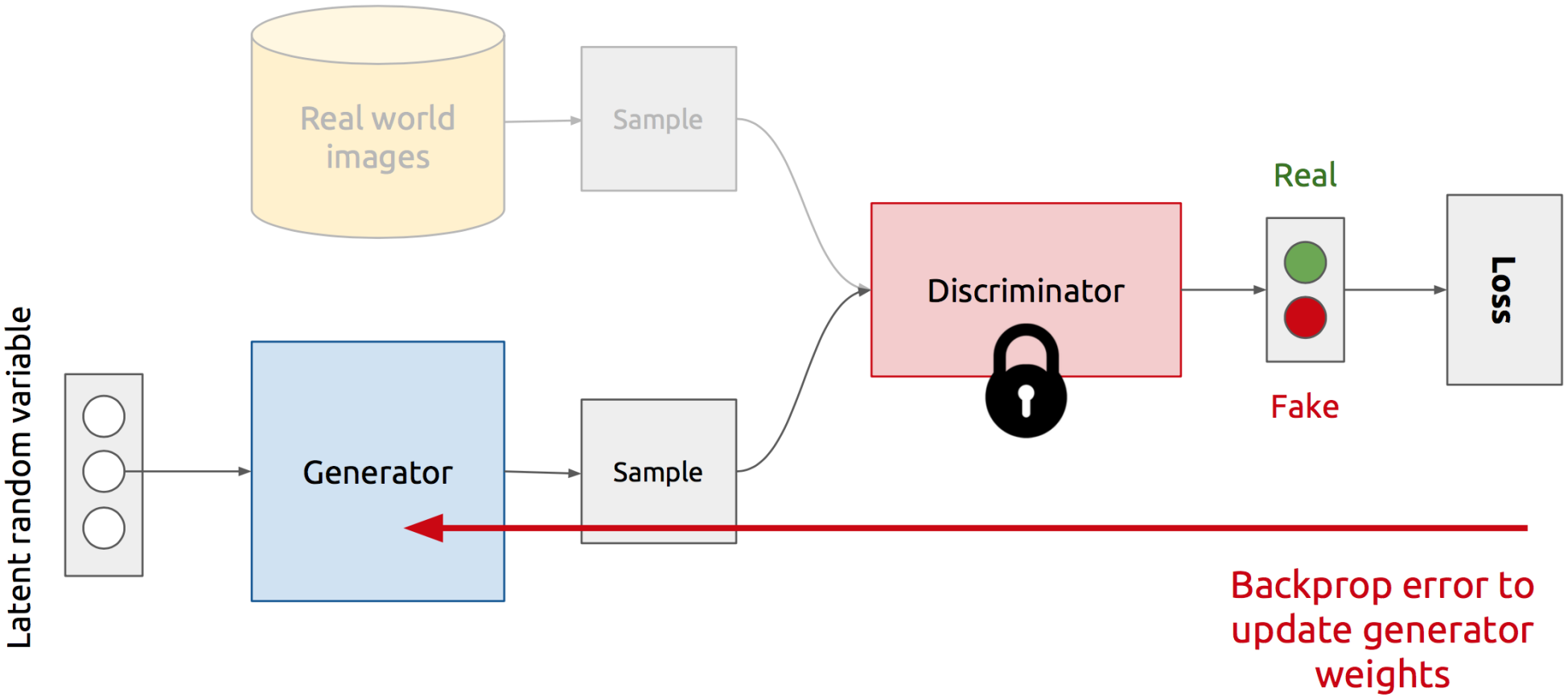


- **Z** is some random noise (Gaussian/Uniform).
- **Z** can be thought as the latent representation of the image.

Training Discriminator



Training Generator



GAN's formulation

$$\min_G \max_D V(D,G)$$

- It is formulated as a **minimax game**, where:
  - The Discriminator is trying to maximize its reward $\boldsymbol{V(D,G)}$
  - The Generator is trying to minimize Discriminator's reward (or maximize its loss)

$$V(D,G) = \mathbb{E}_{x\sim p(x)}[\log D(x)] + \mathbb{E}_{z\sim q(z)}\left[\log\left(1 - D(G(z))\right)\right]$$

- The Nash equilibrium of this particular game is achieved at:
  - $P_{data}(x) = P_{gen}(x) \;\; \forall x$
  - $\mathrm{D}(x) = \frac{1}{2} \;\; \forall x$

Algorithm 1 Minibatch stochastic gradient descent training of generative adversarial nets. The number of steps to apply to the discriminator, k, is a hyperparameter. We used $k = 1$, the least expensive option, in our experiments.

for number of training iterations **do**

for k steps **do**

- Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
- Sample minibatch of m examples $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(m)}\}$ from data generating distribution $p_{\text{data}}(\boldsymbol{x})$.
- Update the discriminator by ascending its stochastic gradient:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(\boldsymbol{x}^{(i)}\right) + \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right) \right].$$

end for

- Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
- Update the generator by descending its stochastic gradient:

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right).$$

end for

The gradient-based updates can use any standard gradient-based learning rule. We used momentum in our experiments.

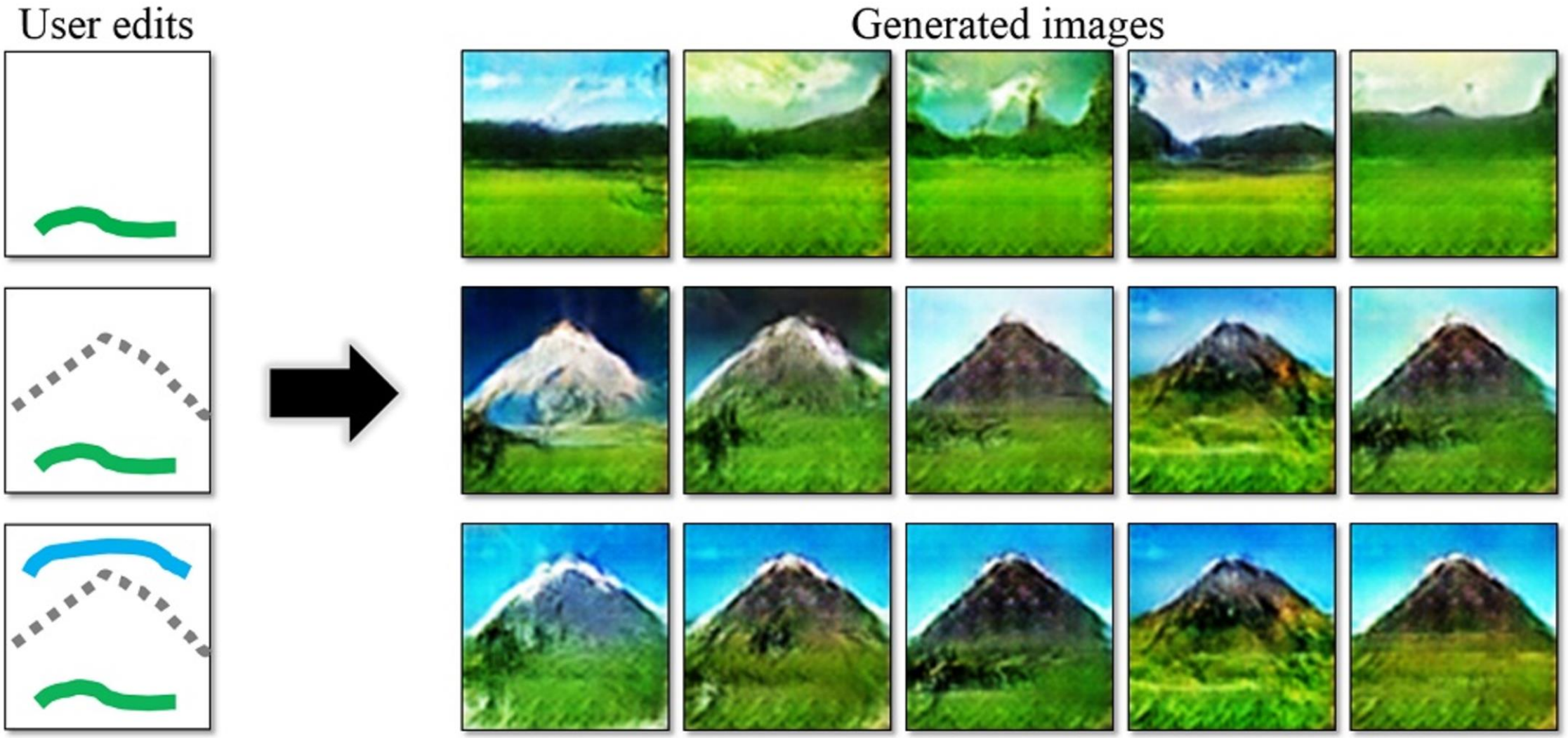
Discriminator updates

Generator updates

Magic of GANs

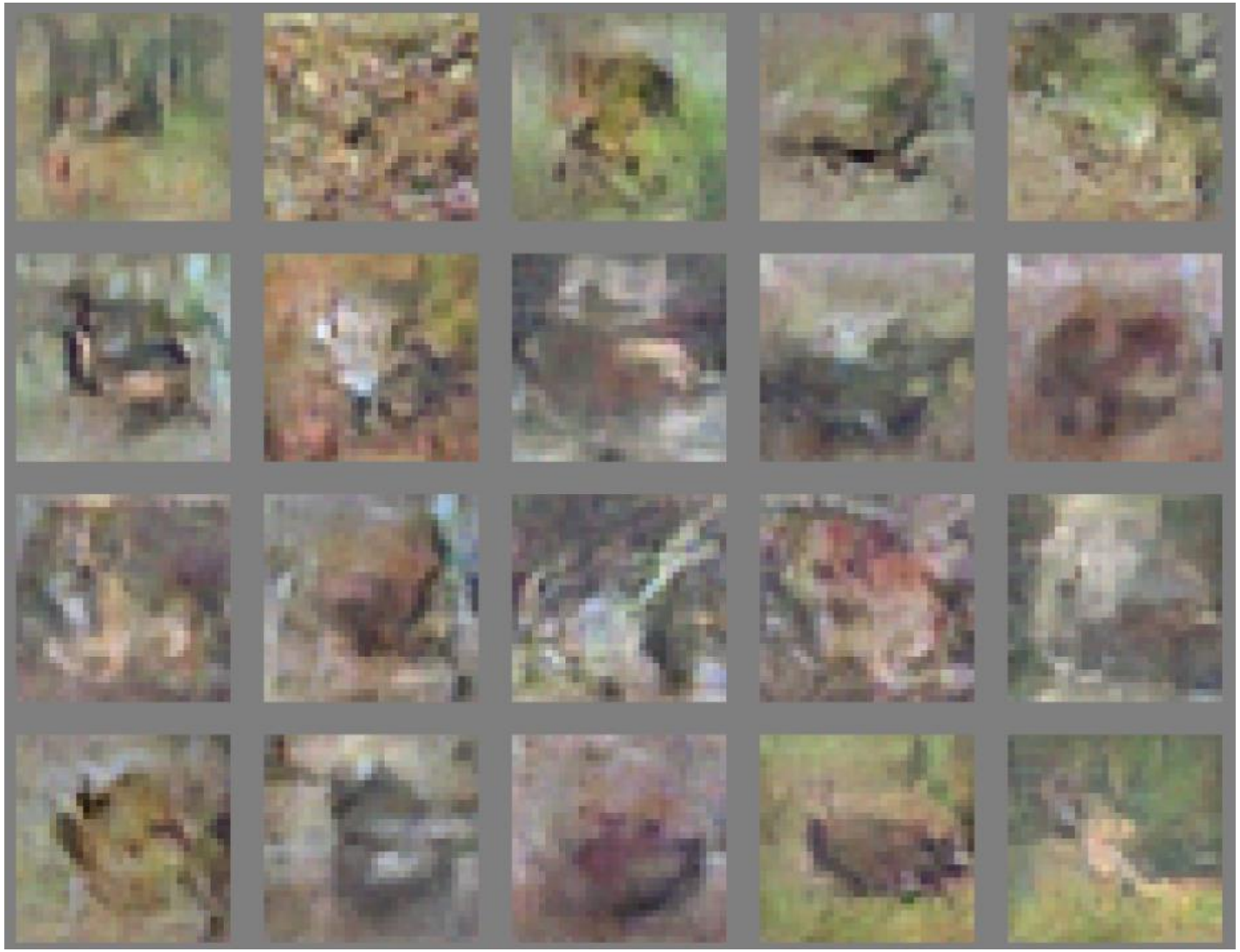
Which one is computer generated?

Magic of GANs



Faces

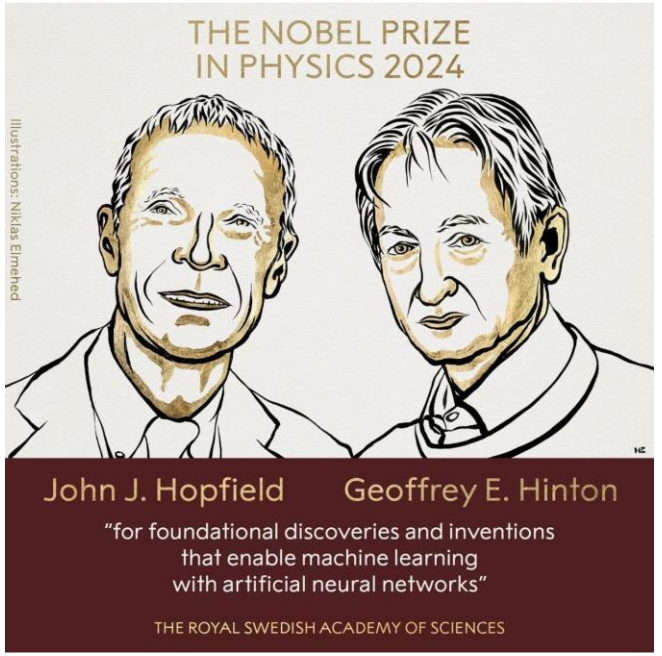
CIFAR



DCGAN

Advantages of GANs

- **Plenty of existing work on Deep Generative Models**
  - Boltzmann Machine
  - Deep Belief Nets
  - Variational AutoEncoders (VAE)



- **Why GANs?**
  - Sampling (or generation) is straightforward.
  - Robust to Overfitting since Generator never sees the training data.
  - Empirically, GANs are good at capturing the modes of the distribution.

Normal model vs GAN

- **Deep Learning models (in general) involve a single player**
  - The player tries to maximize its reward (minimize its loss).
  - Use SGD (with Backpropagation) to find the optimal parameters.
  - SGD has convergence guarantees (under certain conditions).
  - **Problem:** With non-convexity, we might converge to local optima.

$$\min_G L(G)$$

- **GANs instead involve two (or more) players**
  - Discriminator is trying to maximize its reward.
  - Generator is trying to minimize Discriminator's reward.

$$\min_G \max_D V(D, G)$$

  - SGD was not designed to find the Nash equilibrium of a game.
  - **Problem:** We might not converge to the Nash equilibrium at all.

Non-convergence

$$\min_x \max_y V(x, y)$$

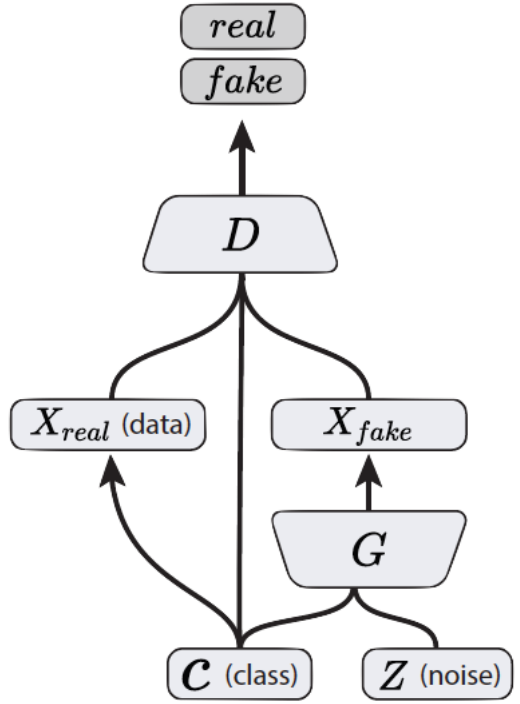
Let $V(x, y) = xy$

• State 1:	x > 0	y > 0	V > 0	Increase y	Decrease x
• State 2:	x < 0	y > 0	V < 0	Decrease y	Decrease x
• State 3:	x < 0	y < 0	V > 0	Decrease y	Increase x
• State 4 :	x > 0	y < 0	V < 0	Increase y	Increase x
• State 5:	x > 0	y > 0	V > 0 == State 1	Increase y	Decrease x

Conditional GANs

- Simple modification to the original GAN framework that conditions the model on *additional information* for better multi-modal learning.

- Lends to many practical applications of GANs when we have explicit *supervision* available.



Conditional GAN
(Mirza & Osindero, 2014)

Image Credit: Figure 2 in Odena, A., Olah, C. and Shlens, J., 2016. Conditional image synthesis with auxiliary classifier GANs. *arXiv preprint arXiv:1610.09585*.

Conditional GANs

MNIST digits generated conditioned on their class label.

$[1, 0, 0, 0, 0, 0, 0, 0, 0, 0] \rightarrow$

$[0, 1, 0, 0, 0, 0, 0, 0, 0, 0] \rightarrow$

$[0, 0, 1, 0, 0, 0, 0, 0, 0, 0] \rightarrow$

$[0, 0, 0, 1, 0, 0, 0, 0, 0, 0] \rightarrow$

$[0, 0, 0, 0, 1, 0, 0, 0, 0, 0] \rightarrow$

$[0, 0, 0, 0, 0, 1, 0, 0, 0, 0] \rightarrow$

$[0, 0, 0, 0, 0, 0, 1, 0, 0, 0] \rightarrow$

$[0, 0, 0, 0, 0, 0, 0, 1, 0, 0] \rightarrow$

$[0, 0, 0, 0, 0, 0, 0, 0, 1, 0] \rightarrow$

$[0, 0, 0, 0, 0, 0, 0, 0, 0, 1] \rightarrow$

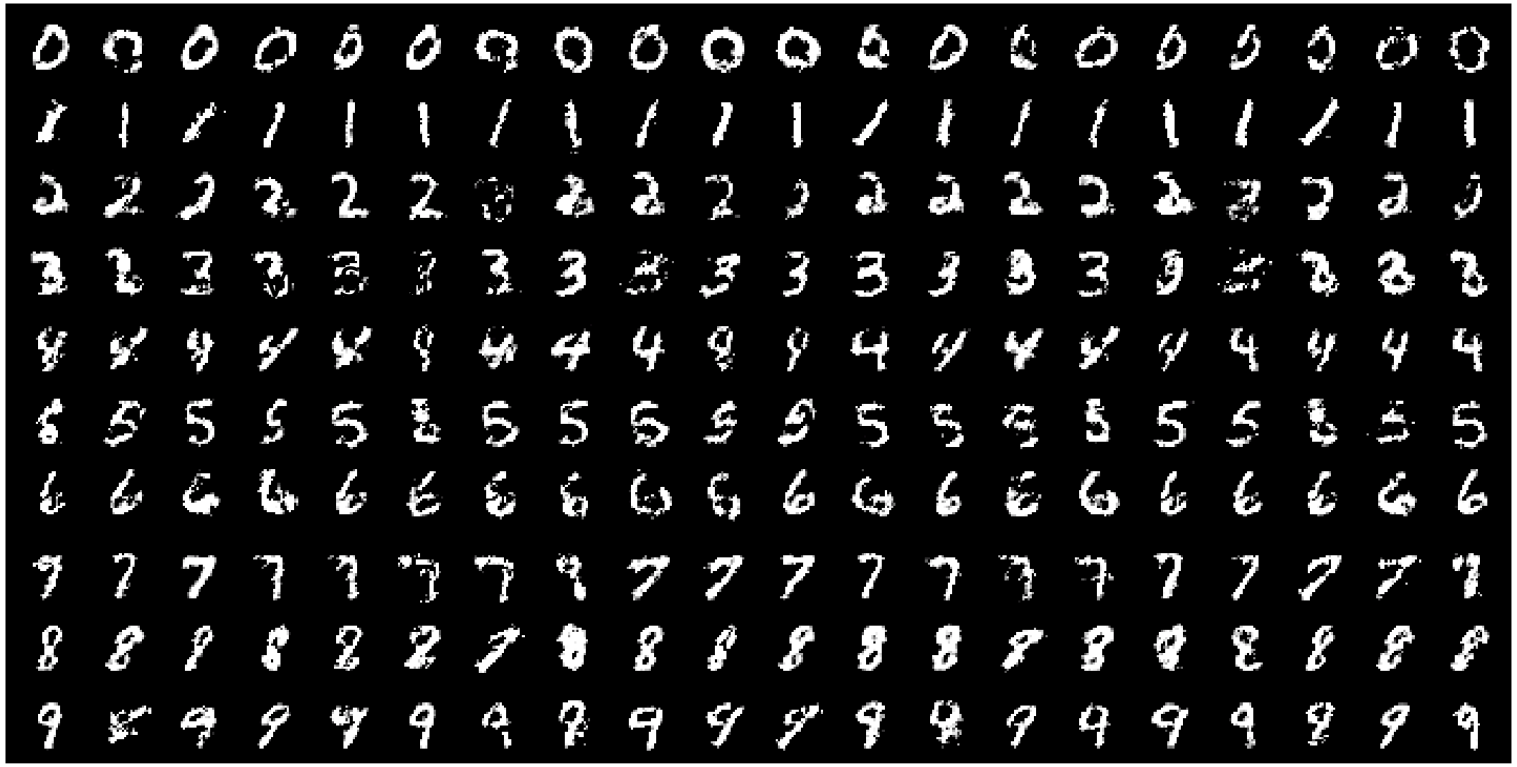
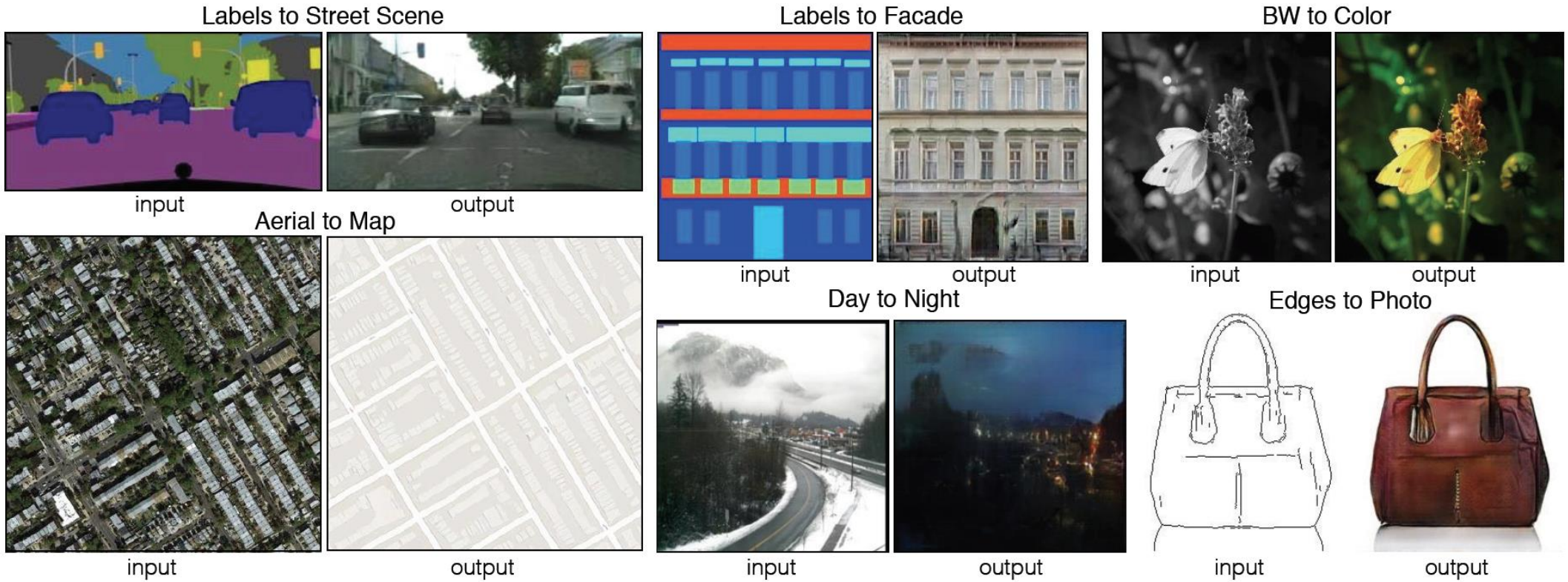
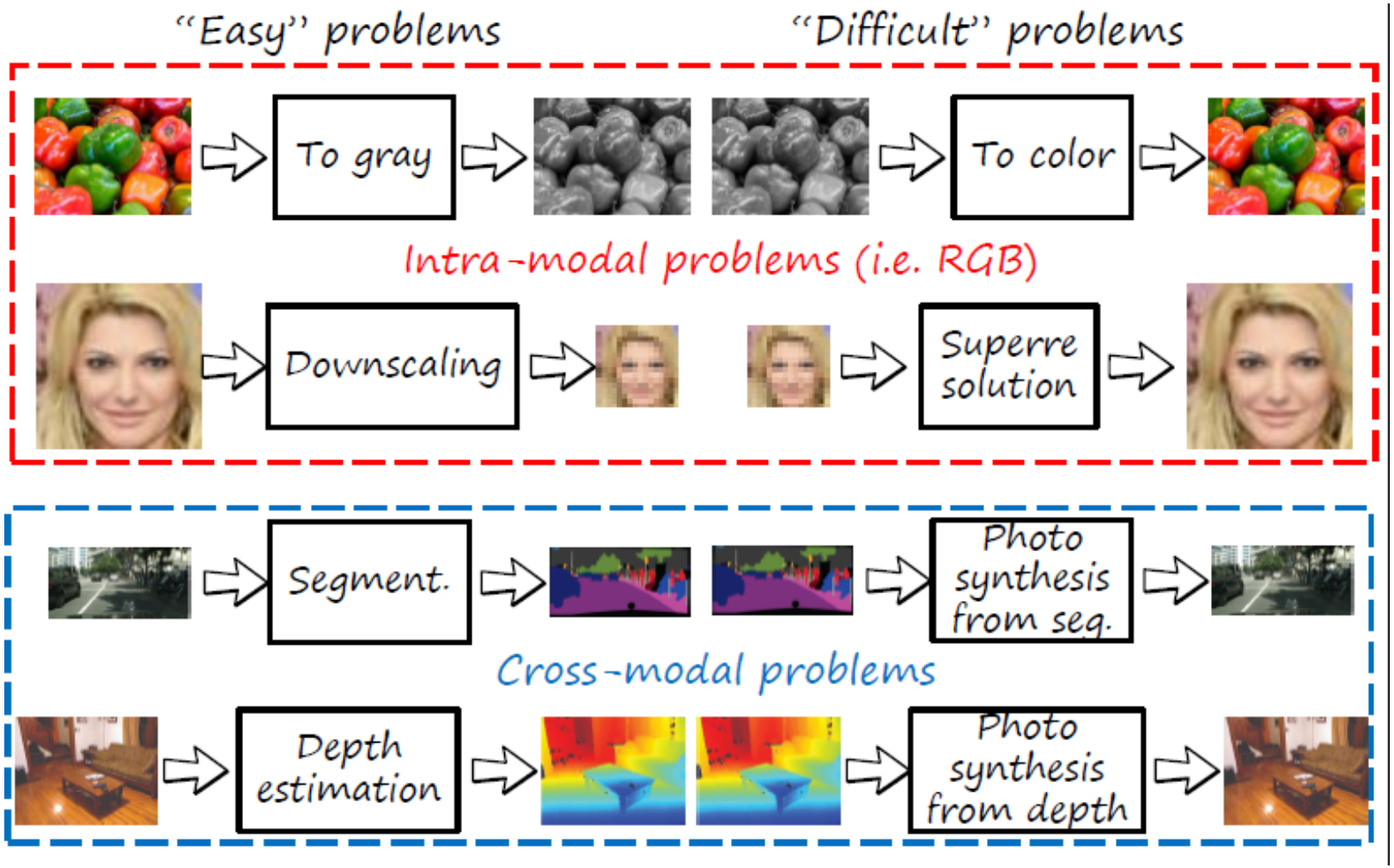


Figure 2 in the original paper.

Image-to-image translation

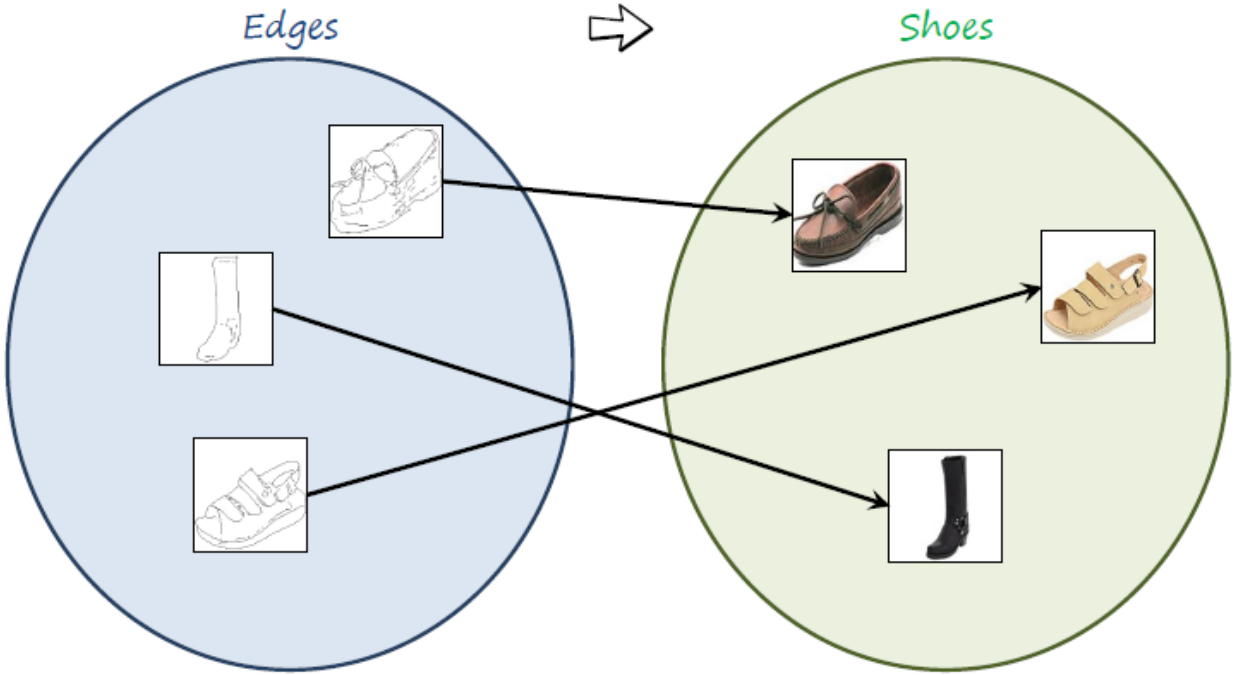


Paired Image to Image Translation

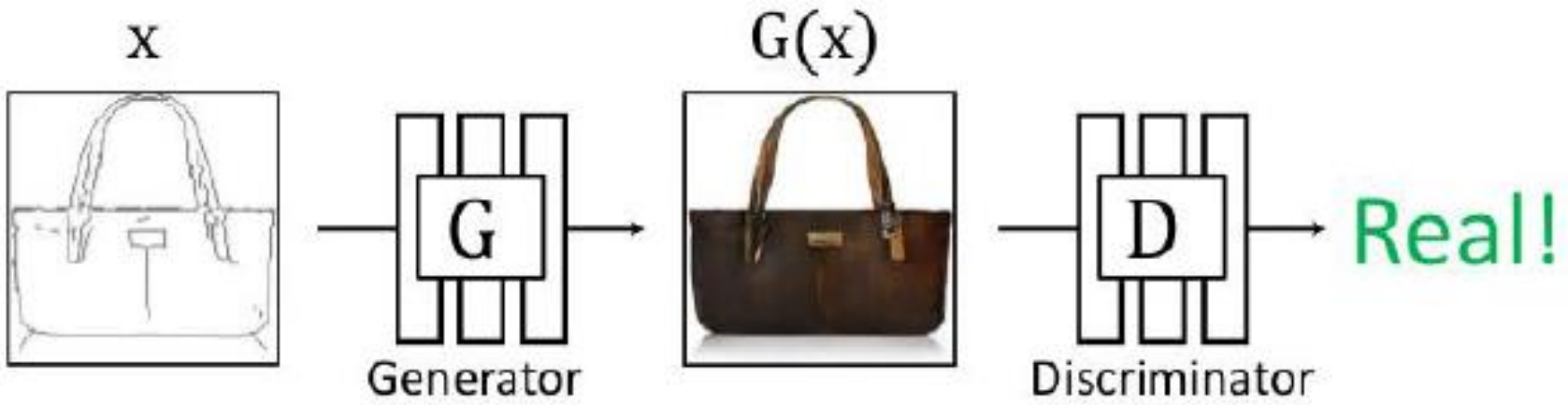


Paired Image to Image Translation

- One to One Mapping



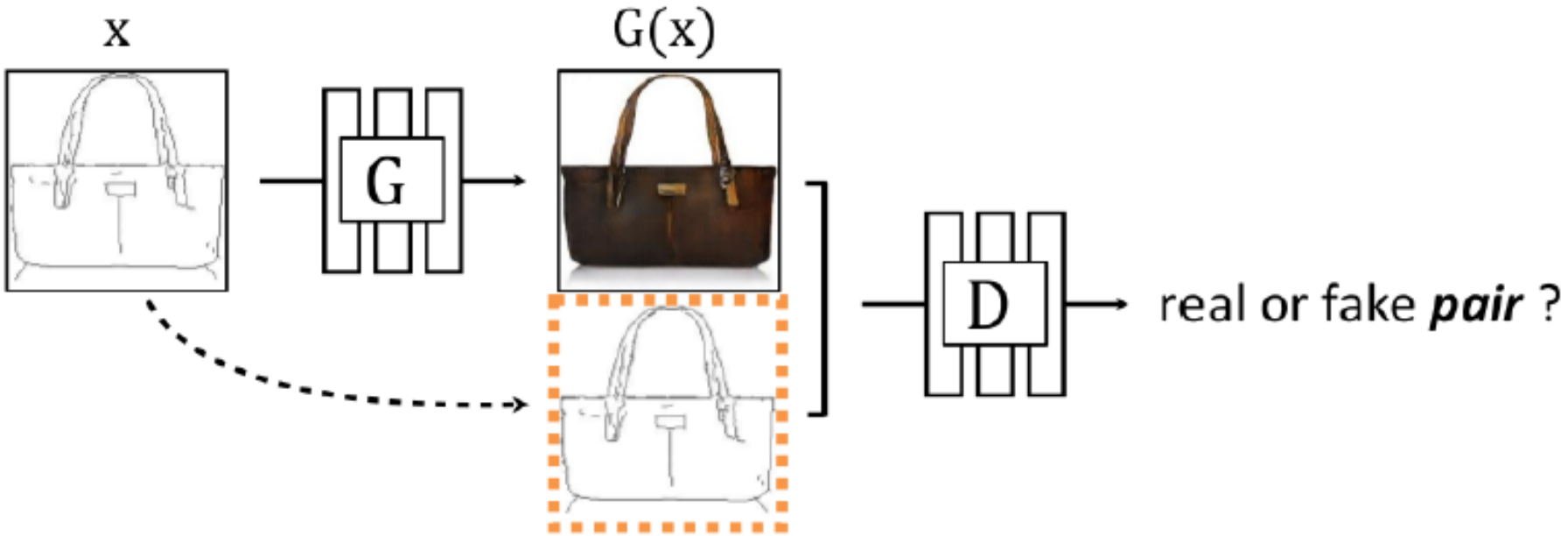
Is this conditional GAN?



$$\min_G \max_D \mathbb{E}_{x,y}[\log D(G(x)) + \log(1 - D(y))]$$

This is conventional GAN!

Pix2Pix

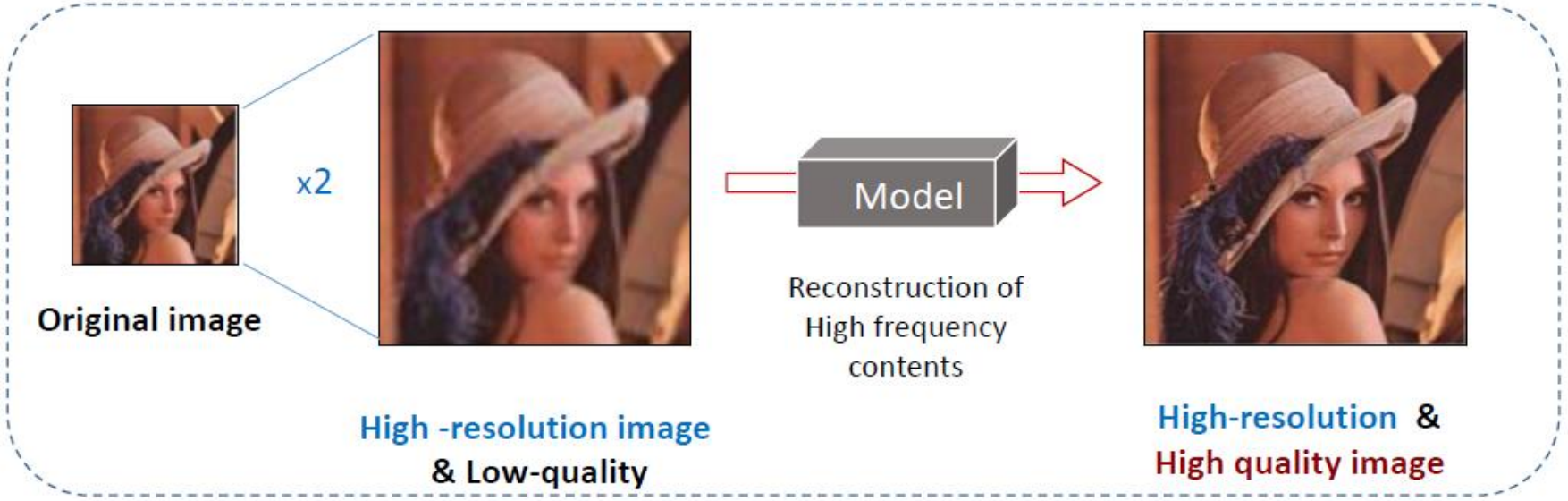


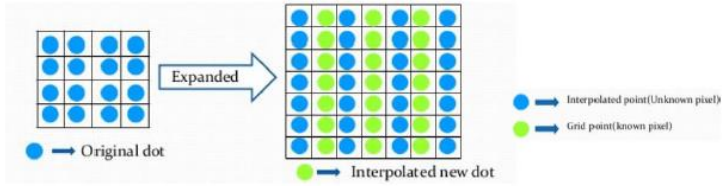
$$\min_G \max_D \mathbb{E}_{x,y}[\log D(x, G(x)) + \log(1 - D(x, y))]$$

Isola et al. "**Image-to-image translation with conditional adversarial networks**". CVPR 2017

[Link to an interactive demo of this paper]

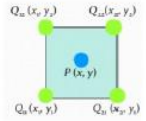
Super Resolution



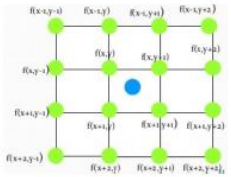


Nearest Neighbor

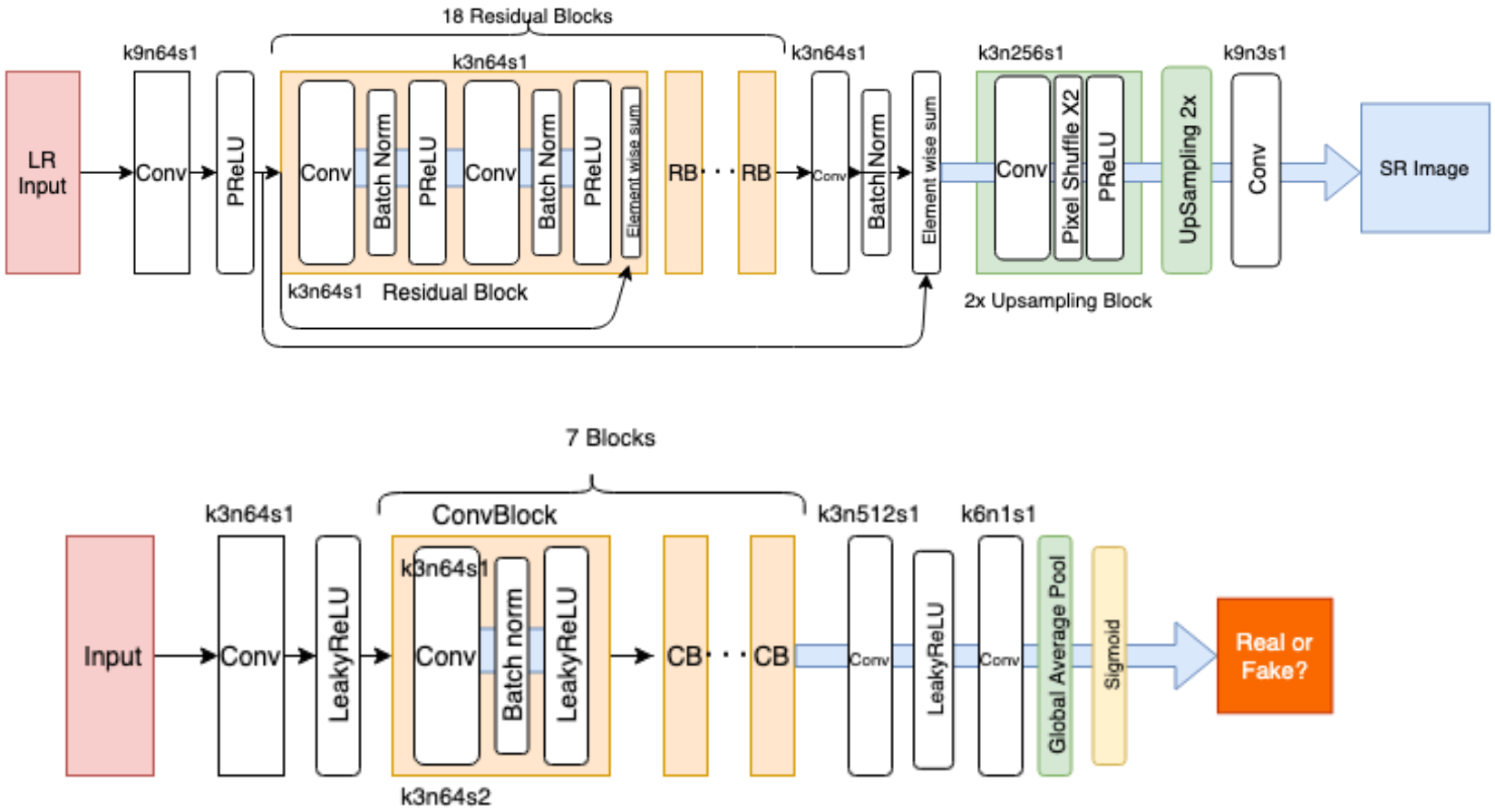
Bilinear



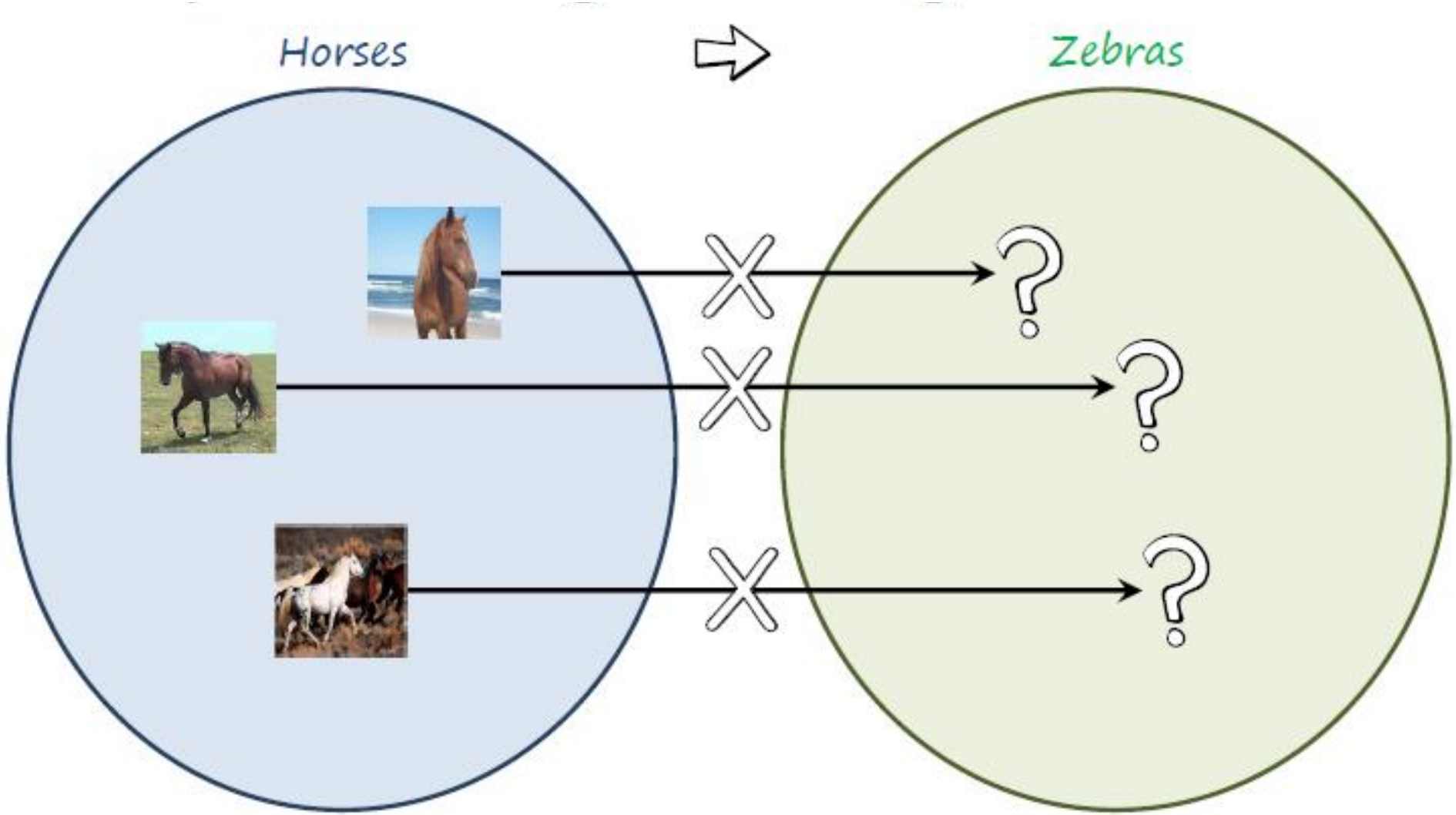
Bicubic



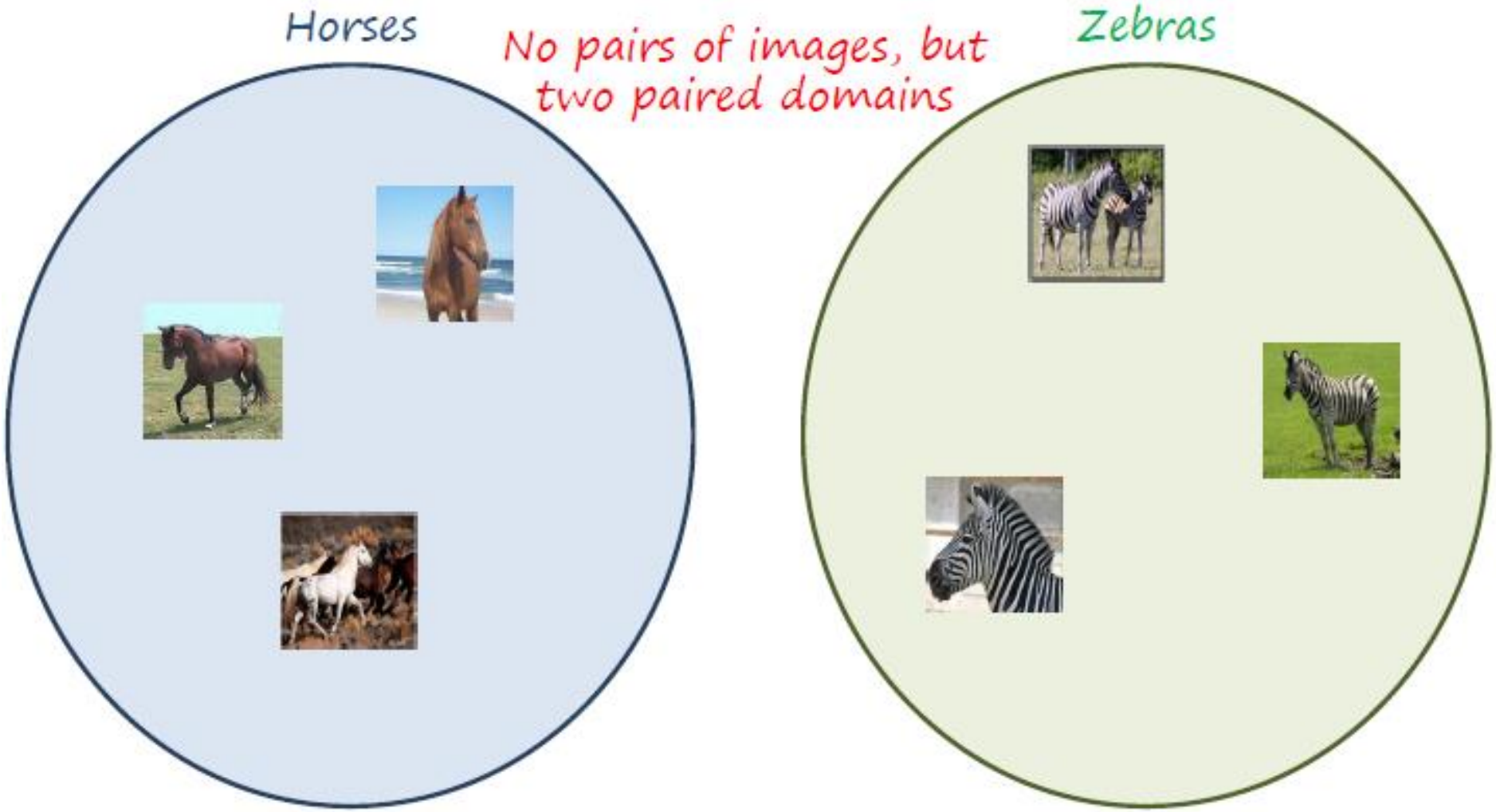
Super Resolution GAN



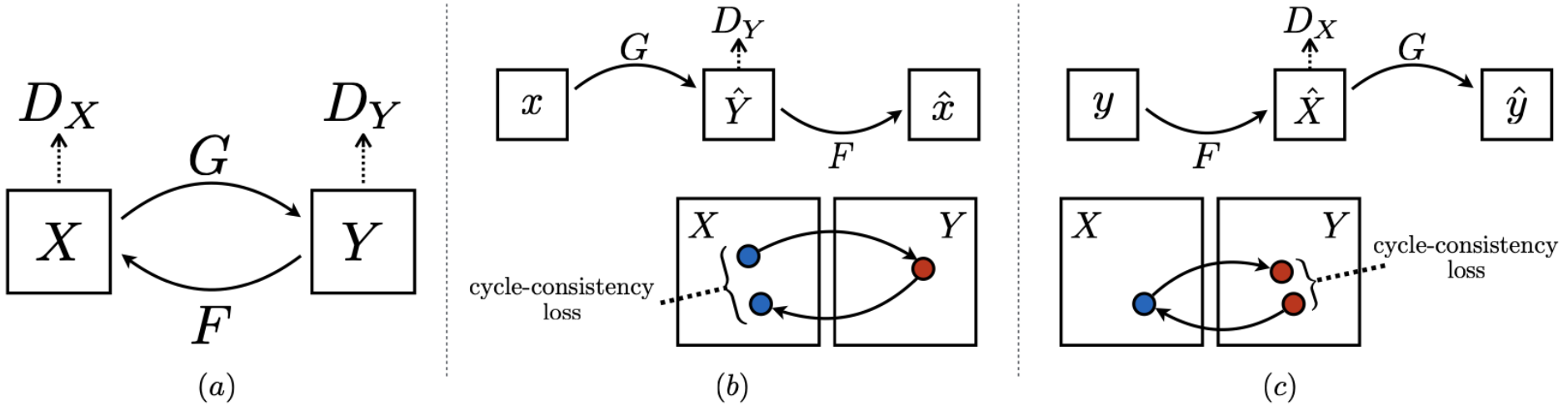
Unpaired Image to Image Translation



Unpaired Image to Image Translation



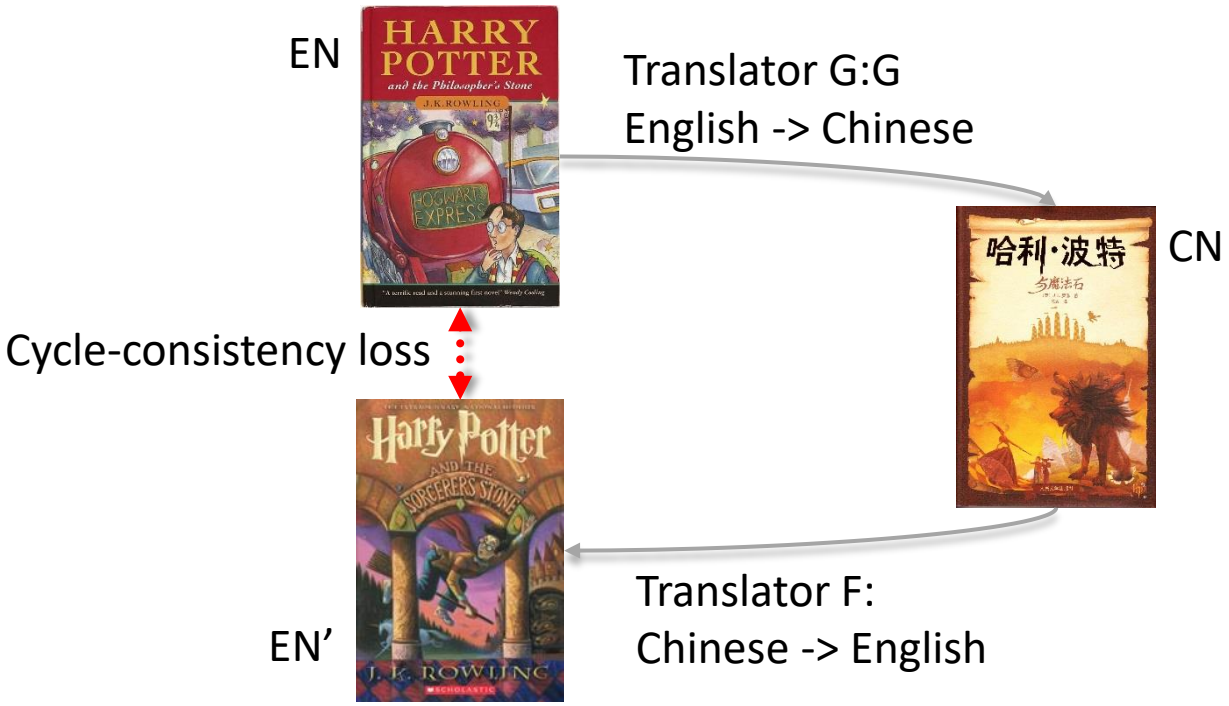
CycleGAN



The model contains two mapping functions G : X → Y and F : Y → X, and associated adversarial discriminators. It has two cycle consistency losses that capture the intuition that if it translates from one domain to the other and back again it should arrive at where it started: (b) forward cycle-consistency loss: x → G(x) → F (G(x)) ≈ x, and (c) backward cycle-consistency loss: y → F (y) → G(F (y)) ≈ y

Zhu et al., "Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks," *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2242-2251.

Cycle Consistency



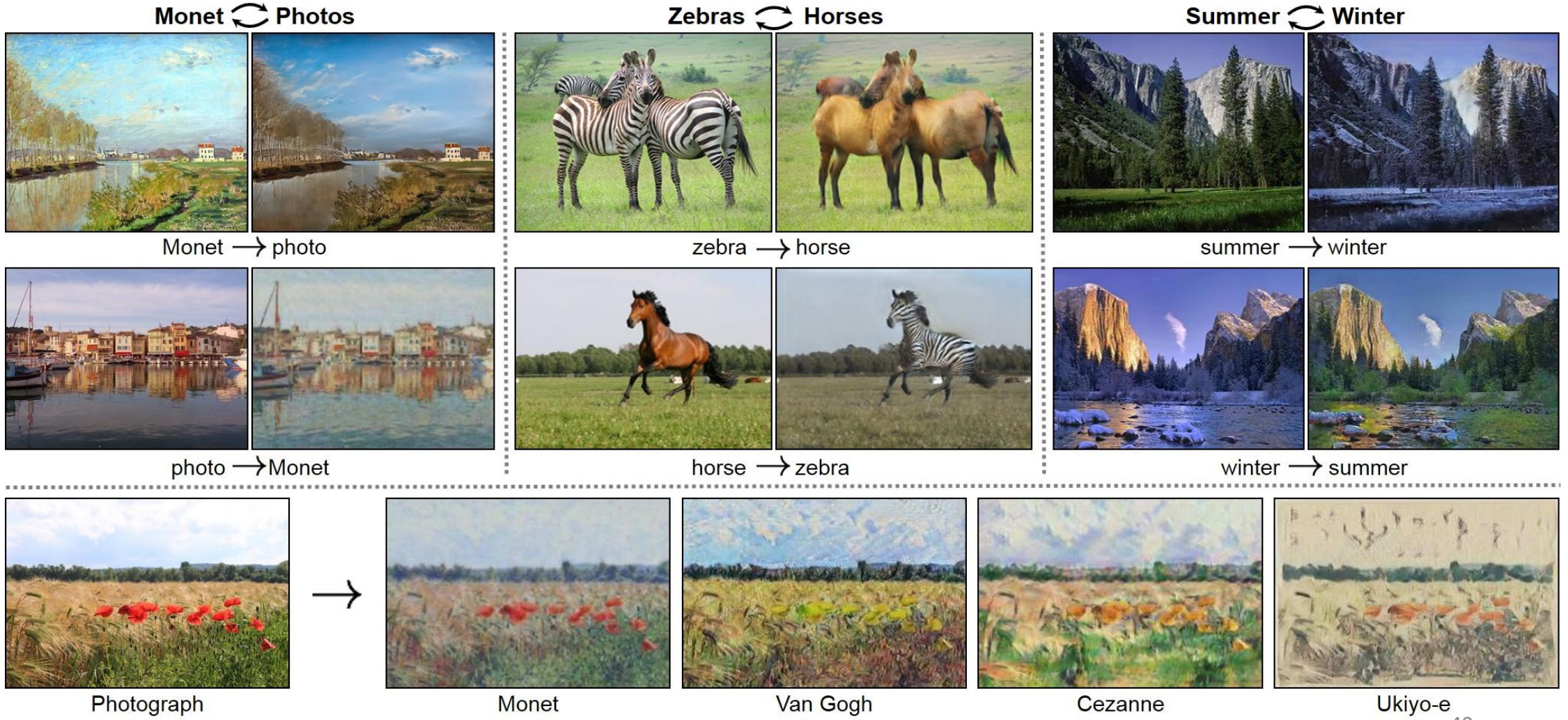
Obviously, we expected

EN -> G(EN) -> F(G(EN)) ~ EN

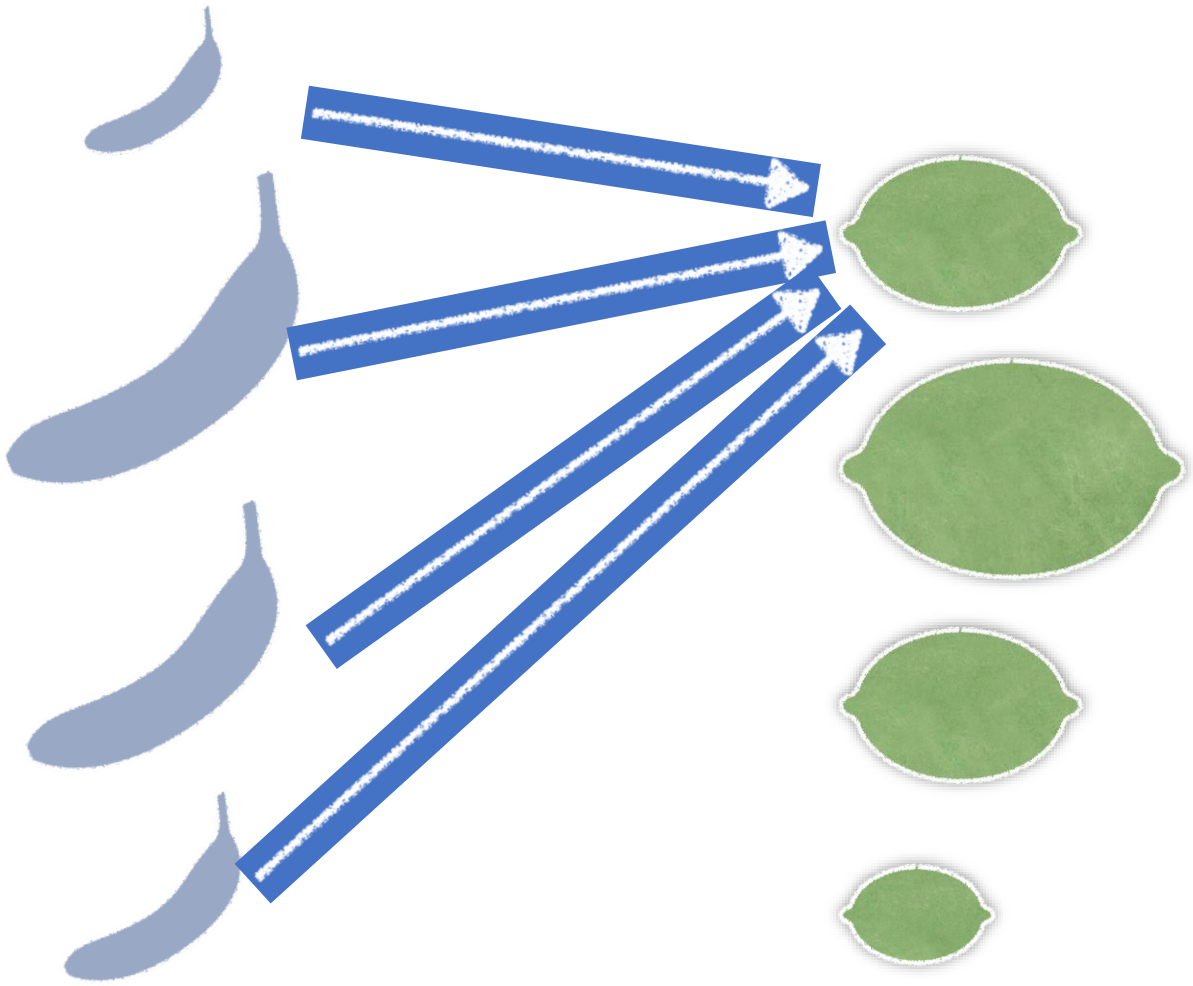
CN EN'

Formalized,
if G: X -> Y, F: Y -> X; expect,
X -> G(X) -> F(G(X)) ~ X

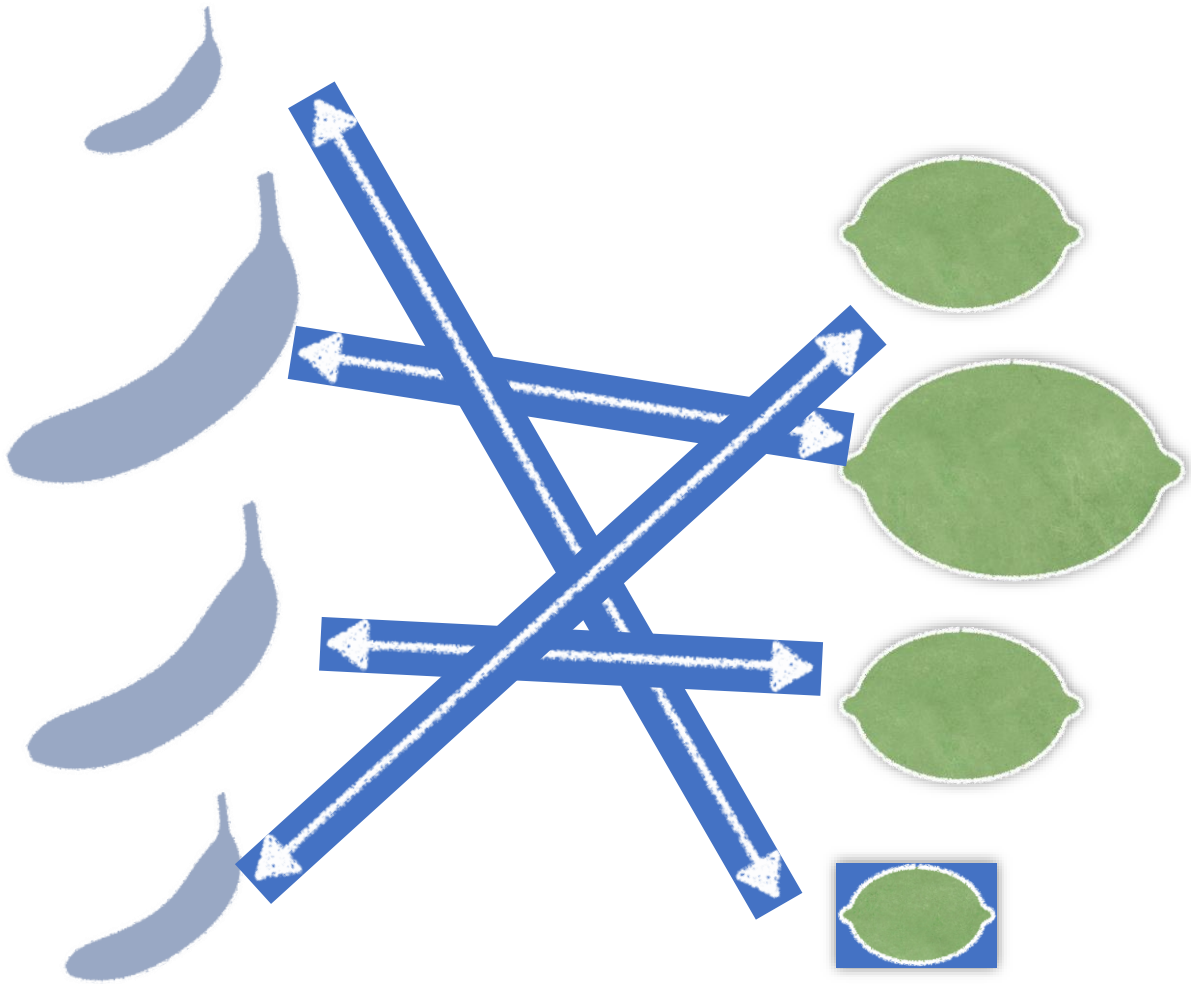
CycleGAN


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

mode collapse

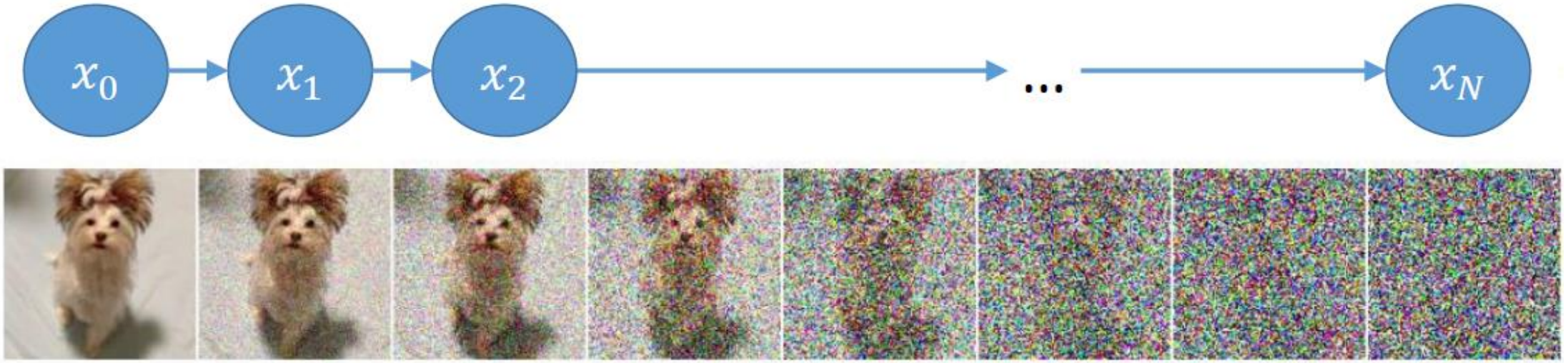


one to one mapping



Diffusion Probabilistic Models

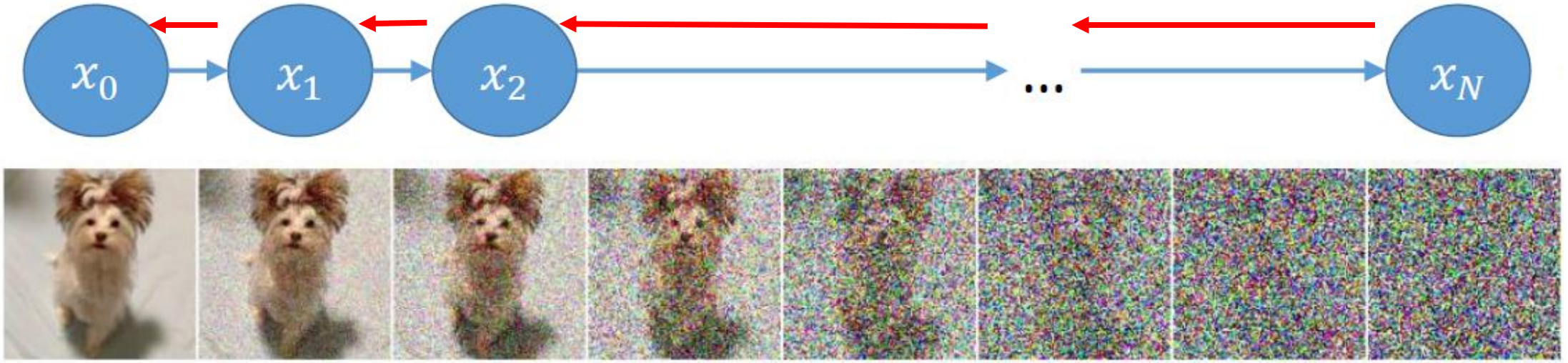
- Diffusion process gradually adds noise to data



Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. In Proceedings of the 34th International Conference on Neural Information Processing Systems (NIPS '20). Curran Associates Inc., Red Hook, NY, USA, Article 574, 6840–6851.

Diffusion Probabilistic Models

- Diffusion process in the reverse direction ⇔denoising process



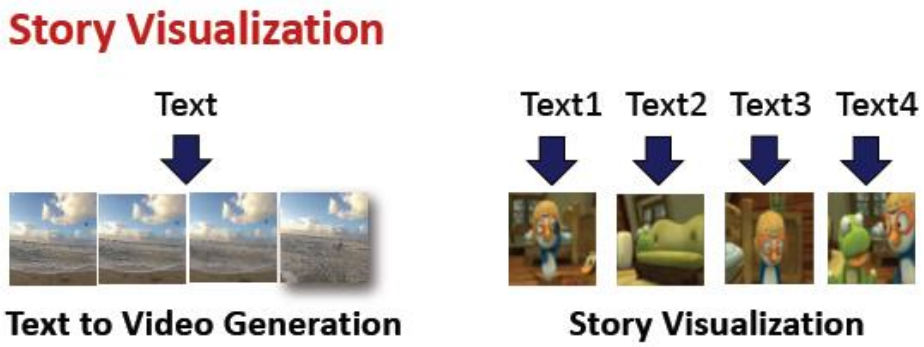
Video GAN

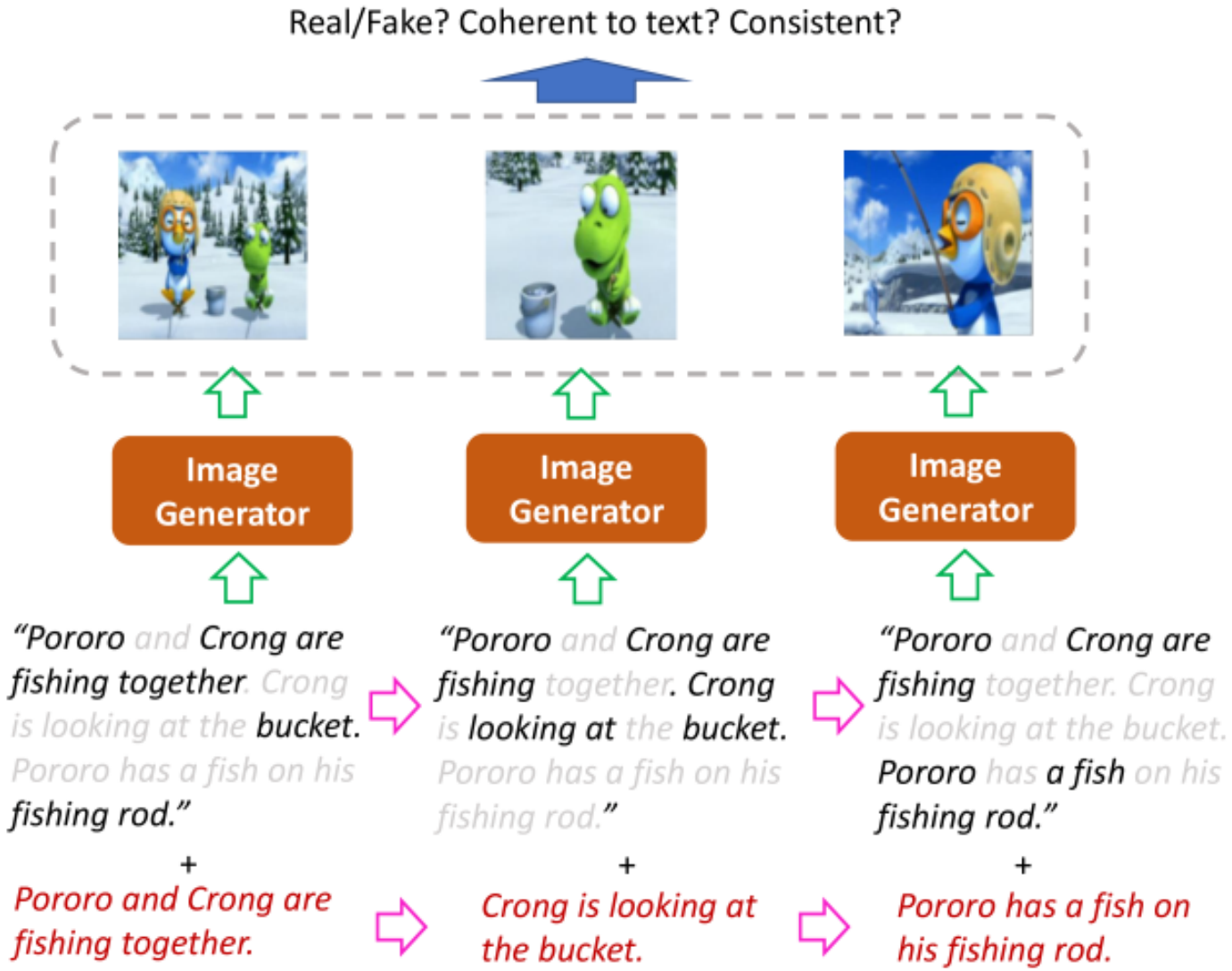
Three main types:

- GANs frameworks that use RNN architectures to address the time-series nature of video data.
- Two-stream architecture video GANs, where each stream considers a different aspect of the video.
- Progressive video GANs models in which initial frames are first generated, and the generated data is fed into another generator to produce an enhanced result.

Text to Video

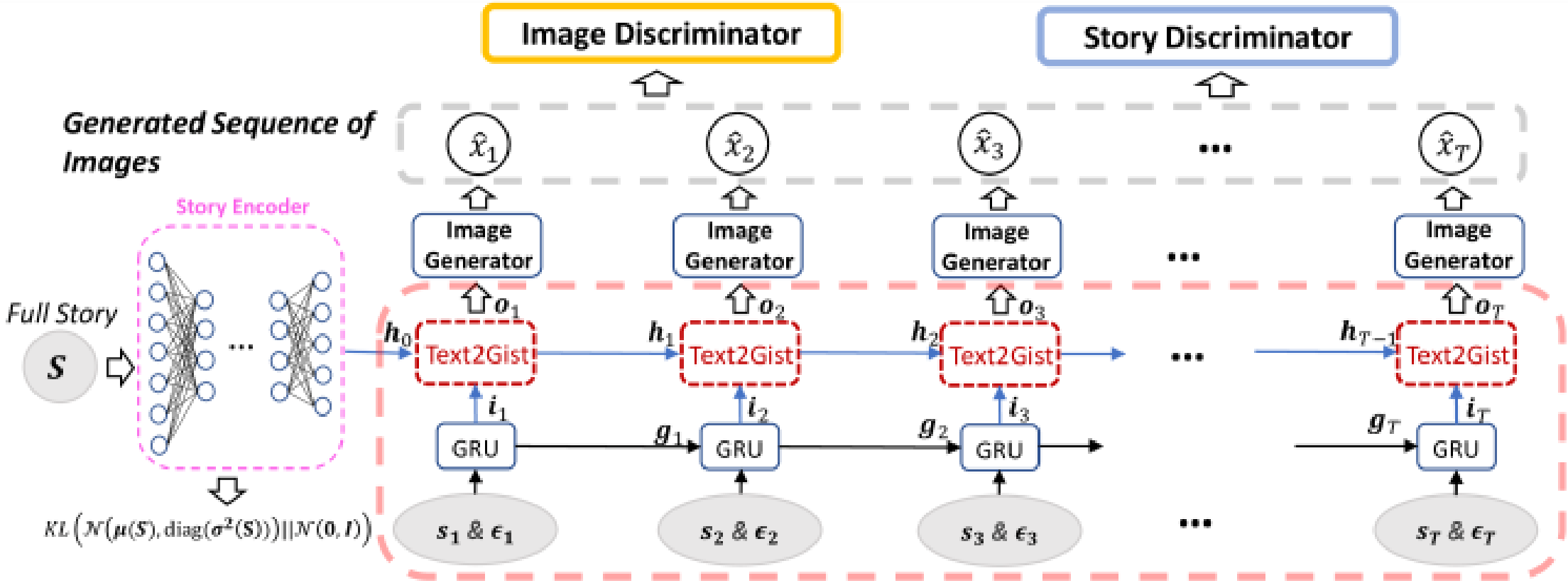
Challenge: The generated image sequence must consistently and coherently depict the whole story and maintain the logic of the storyline





StoryGAN: A Sequential Conditional GAN for Story Visualization

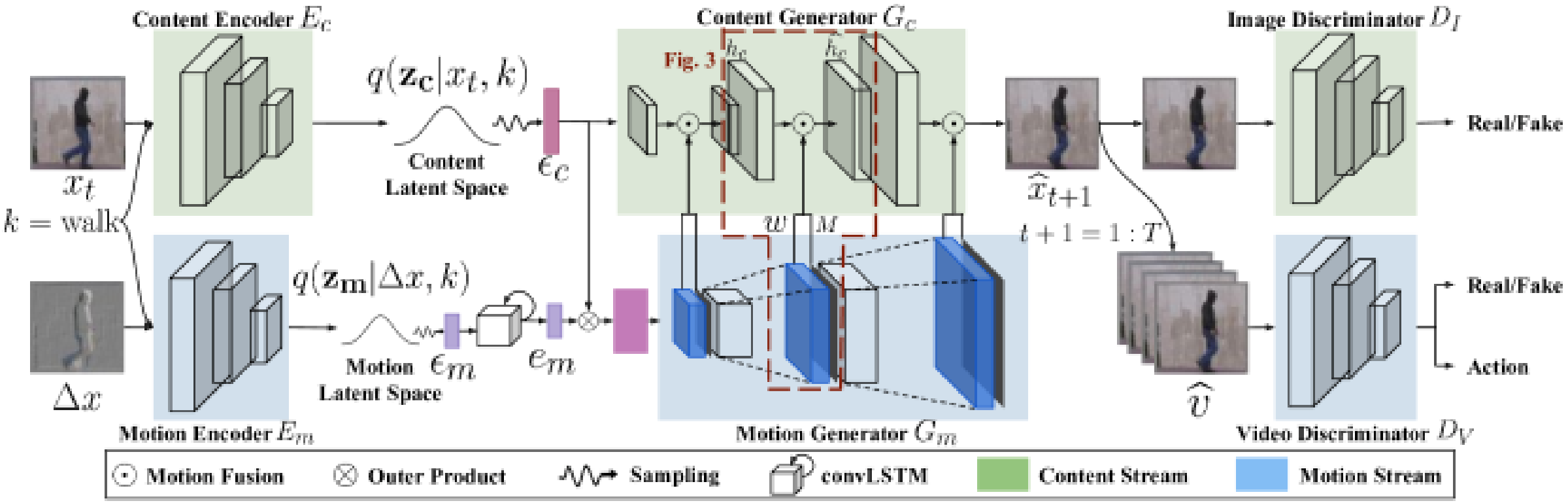
StoryGAN uses a gated recurrent unit (GRU) RNN to incorporate the previously generated images into the current sentence's image generation.



] Y. Li et al., "StoryGAN: A Sequential Conditional GAN for Story Visualization," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 6329-6338.

Two-Stream VAN for Video Generation

Generates a video from an action label and a disentangled noise vector



Content Stream + Motion Stream

X. Sun, H. Xu, and K. Saenko, "TwoStreamVAN: Improving Motion Modeling in Video Generation", WACV, 2020.

Sora: Video generation models as world simulators

https://openai.com/index/video-generation-models-as-world-simulators/

Transformers, Diffusion model, DeepFake, ….

Summary

- GANs are generative models that are implemented using two types of stochastic neural network modules: **Generators** and **Discriminators**.
- **Generator** tries to generate samples from random noise or condition as input
- **Discriminator** tries to distinguish the samples from Generator and samples from the real data distribution.
- Both networks are trained adversarially to fool the other component. In this process, both models become better at their respective tasks.

Assignment

Suppose that you receive a task to develop a deep learning network that can generate fake VIDEOs of Donald Trump.

Describe your methods in no more than two pages (A4, font 11), including but not limited to:

1) Data
2) Network architecture
3) Any other resources you need
4) Explain your design and thoughts

Please submit your solutions to course site in PDF format by 13 April 2025.

Online submission system close after 13 April 2025