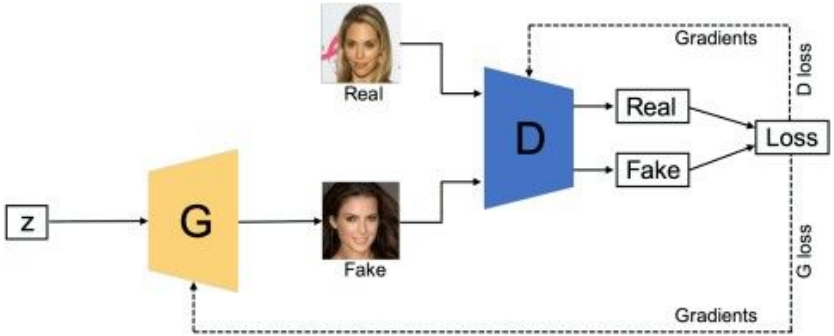
Chapter 3
Generative Adversarial Networks

Three Perspective on GANs

1. Structured loss

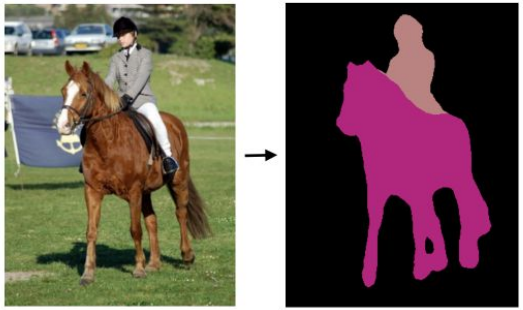
2. Generative model

3. Domain-level supervision / mapping



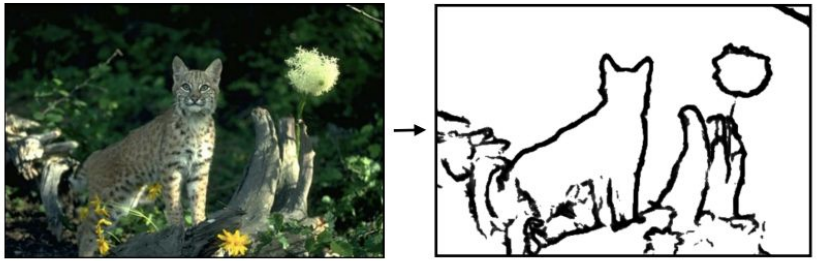
Data Prediction Problems ("Structured Prediction")

Object labeling



[Long et al. 2015, …]

Edge Detection



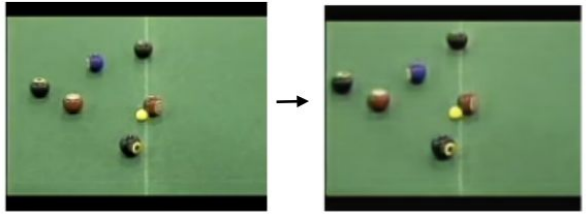
[Xie et al. 2015, …]

Text-to-photo

"this small bird has a pink breast and crown…"

[Reed et al. 2014, …]

Future frame prediction



[Mathieu et al. 2016, …]

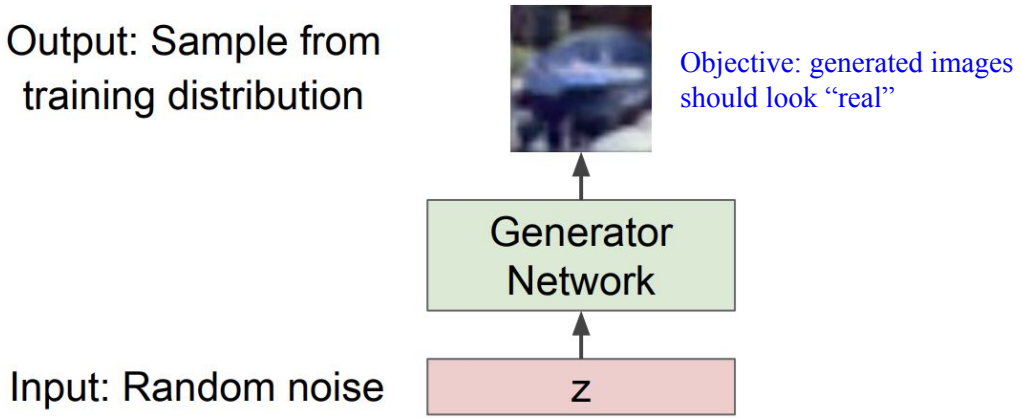
Data Prediction Problems ("Structured Prediction")

Problem: Want to sample from complex, high-dimensional training distribution.
Answer: No direct way to do this!

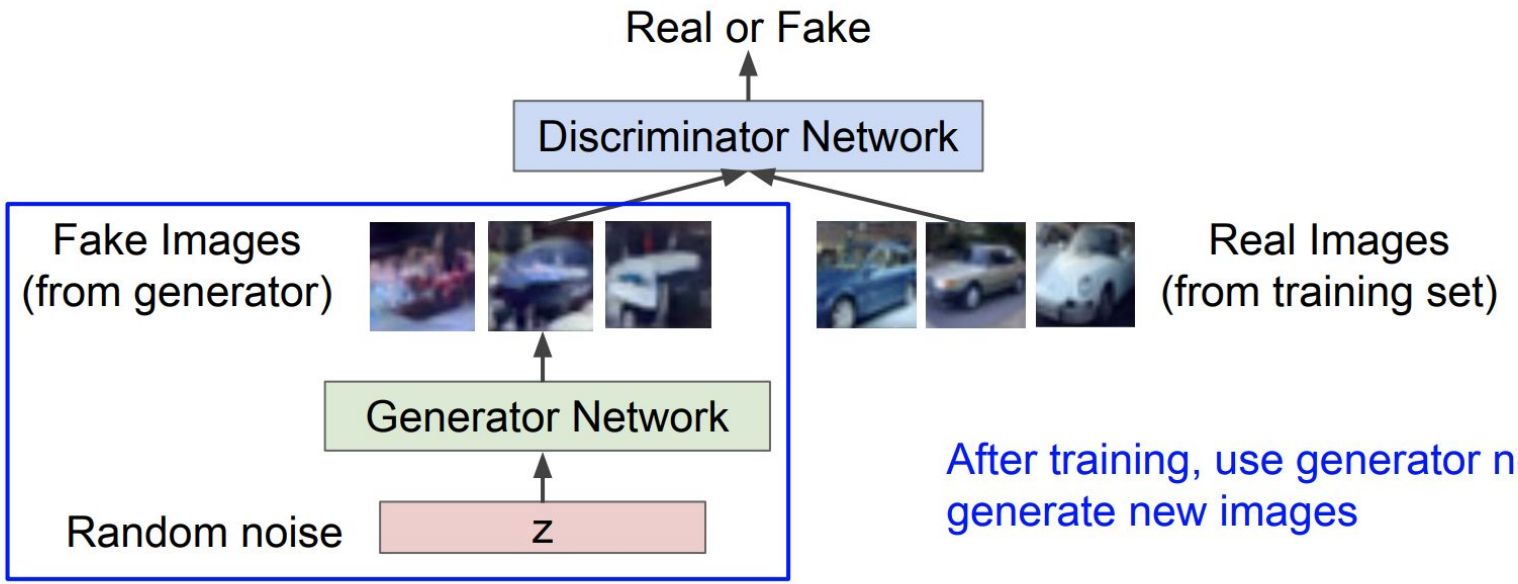
Solution: Sample from a simple distribution we can easily sample from, e.g. random noise. Learn transformation to training distribution.

But we don't know which sample z maps to which training image => can't learn by reconstructing training images.

Solution: Use a discriminator network to tell whether the generate image is within data distribution ("real") or not.

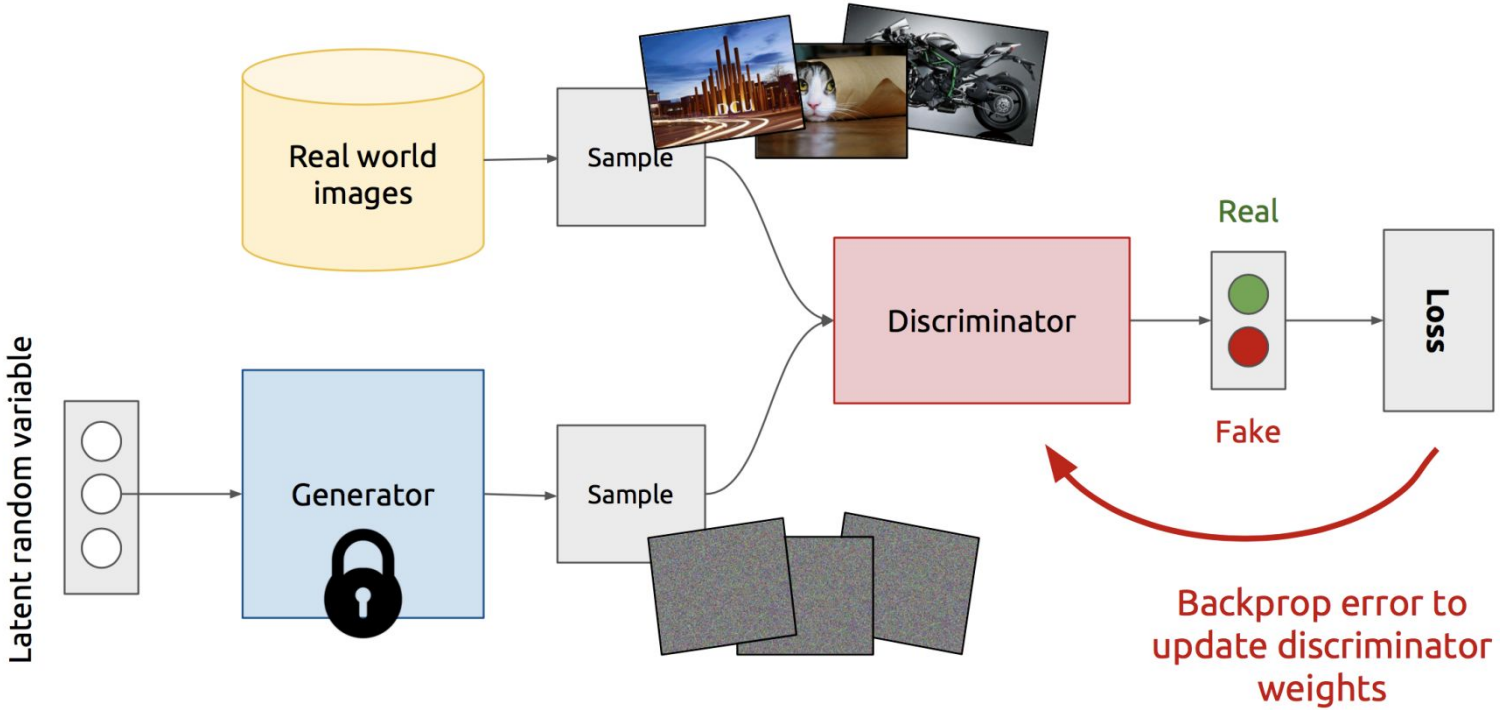


GAN: Architecture

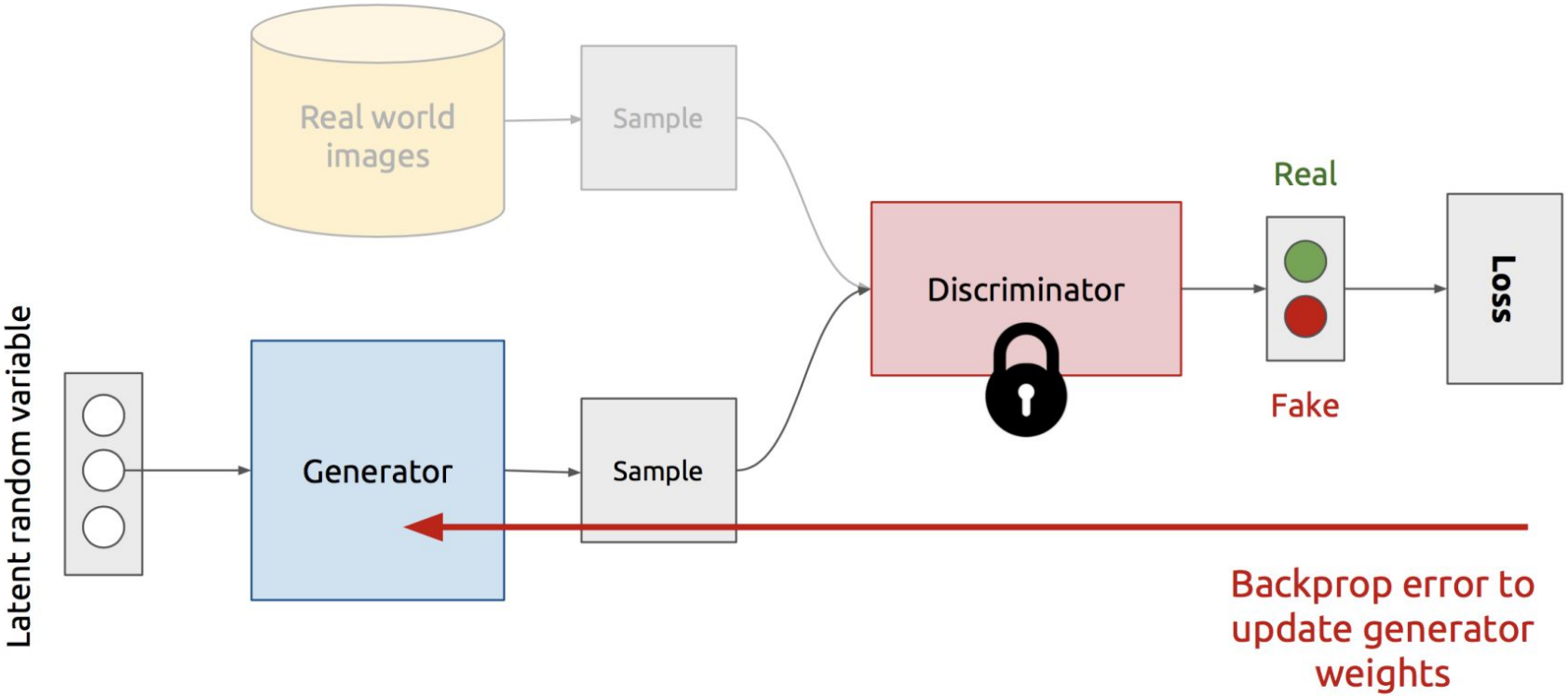


After training, use generator network to generate new images

Training Discriminator



Training Generator



Training GAN: Two player Game

Discriminator network: try to distinguish between real and fake images
Generator network: try to fool the discriminator by generating real-looking images

Train jointly in **minimax game**

Minimax objective function:

Discriminator outputs likelihood in (0,1) of real image

$$\min_{\theta_g} \max_{\theta_d} \left[\mathbb{E}_{x \sim p_{data}} \log D_{\theta_d}(x) + \mathbb{E}_{z \sim p(z)} \log(1 - D_{\theta_d}(G_{\theta_g}(z))) \right]$$

Discriminator output for real data x

Discriminator output for generated fake data G(z)

- Discriminator (θd) wants to maximize objective such that D(x) is close to 1 (real) and D(G(z)) is close to 0 (fake)
- Generator (θg) wants to minimize objective such that D(G(z)) is close to 1 (discriminator is fooled into thinking generated G(z) is real)

Training GAN: Two player Game

Minimax objective function:

$$\min_{\theta_g} \max_{\theta_d} \left[\mathbb{E}_{x \sim p_{data}} \log D_{\theta_d}(x) + \mathbb{E}_{z \sim p(z)} \log(1 - D_{\theta_d}(G_{\theta_g}(z))) \right]$$

Alternate between:

1. **Gradient ascent** on discriminator

$$\max_{\theta_d} \left[\mathbb{E}_{x \sim p_{data}} \log D_{\theta_d}(x) + \mathbb{E}_{z \sim p(z)} \log(1 - D_{\theta_d}(G_{\theta_g}(z))) \right]$$

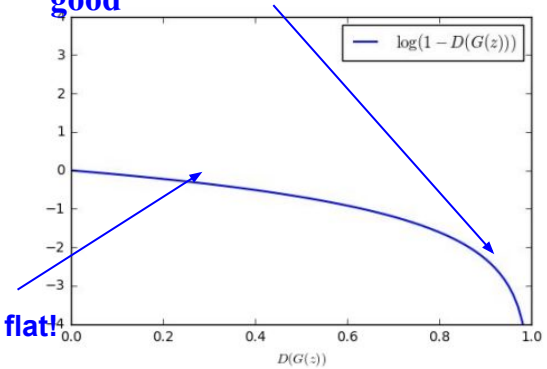
2. **Gradient descent** on generator

$$\min_{\theta_g} \mathbb{E}_{z \sim p(z)} \log(1 - D_{\theta_d}(G_{\theta_g}(z)))$$

In practice, optimizing this generator objective does not work well!

When sample is likely fake, want to learn from it to improve generator (move to the right on X axis).

Gradient signal dominated by region where sample is already good



But gradient in this region is relatively flat!

Training GAN: Two player Game

Minimax objective function:

$$\min_{\theta_g} \max_{\theta_d} \left[\mathbb{E}_{x \sim p_{data}} \log D_{\theta_d}(x) + \mathbb{E}_{z \sim p(z)} \log(1 - D_{\theta_d}(G_{\theta_g}(z))) \right]$$

Alternate between:

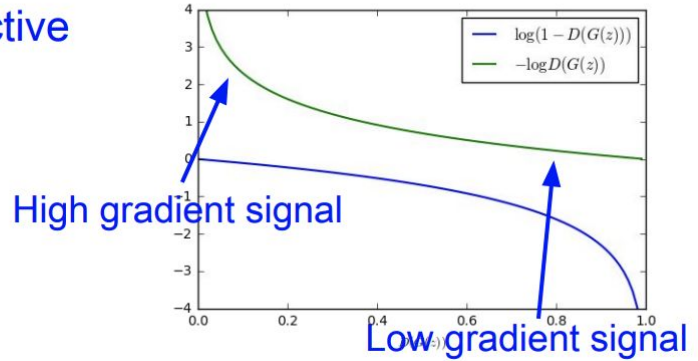
1. **Gradient ascent** on discriminator

$$\max_{\theta_d} \left[\mathbb{E}_{x \sim p_{data}} \log D_{\theta_d}(x) + \mathbb{E}_{z \sim p(z)} \log(1 - D_{\theta_d}(G_{\theta_g}(z))) \right]$$

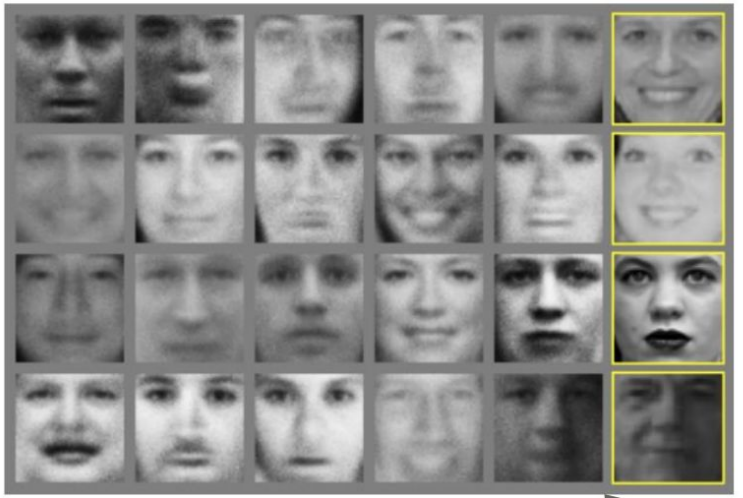
2. **Instead: Gradient ascent** on generator, different objective

$$\max_{\theta_g} \mathbb{E}_{z \sim p(z)} \log(D_{\theta_d}(G_{\theta_g}(z)))$$

Instead of minimizing likelihood of discriminator being correct, now maximize likelihood of discriminator being wrong.
Same objective of fooling discriminator, but now higher gradient signal for bad samples => works much better! Standard in practice.

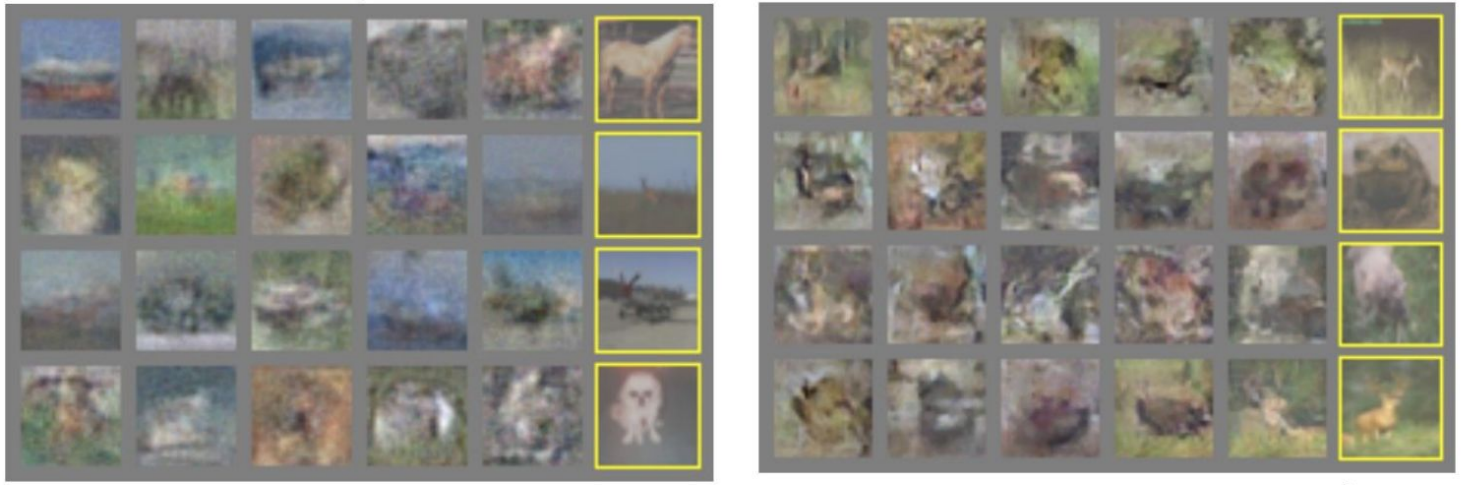


Generated Samples



Nearest neighbor from training set

Generated Samples (CIFAR-10)



Nearest neighbor from training set

GANs: Convolutional Architectures

Generator is an upsampling network with fractionally-strided convolutions
Discriminator is a convolutional network

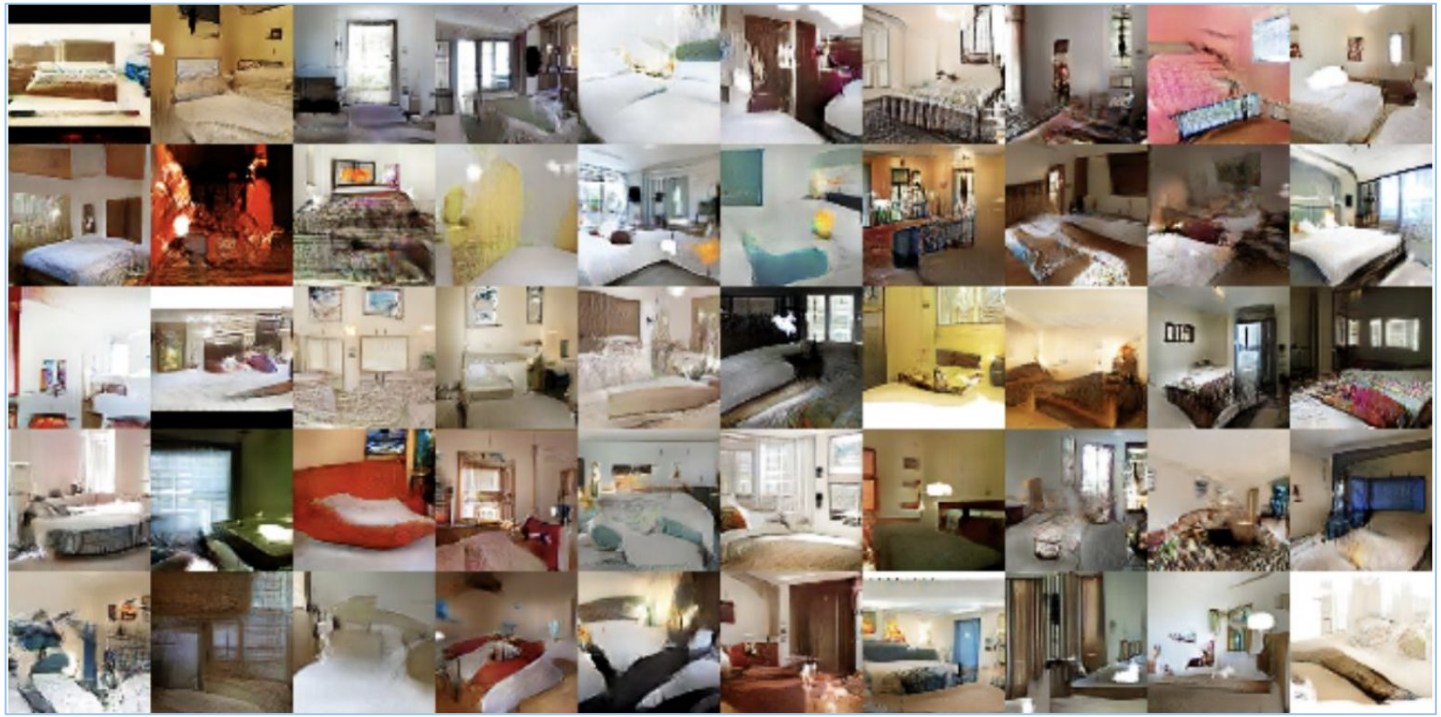
Architecture guidelines for stable Deep Convolutional GANs

- 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 et al, “Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks”, ICLR 2016

GANs: Convolutional Architectures

Samples from the model look much better!



LSUN Bedrooms dataset

2017: Explosion of GANs

See also: https://github.com/soumith/ganhacks for tips and tricks for trainings GANs

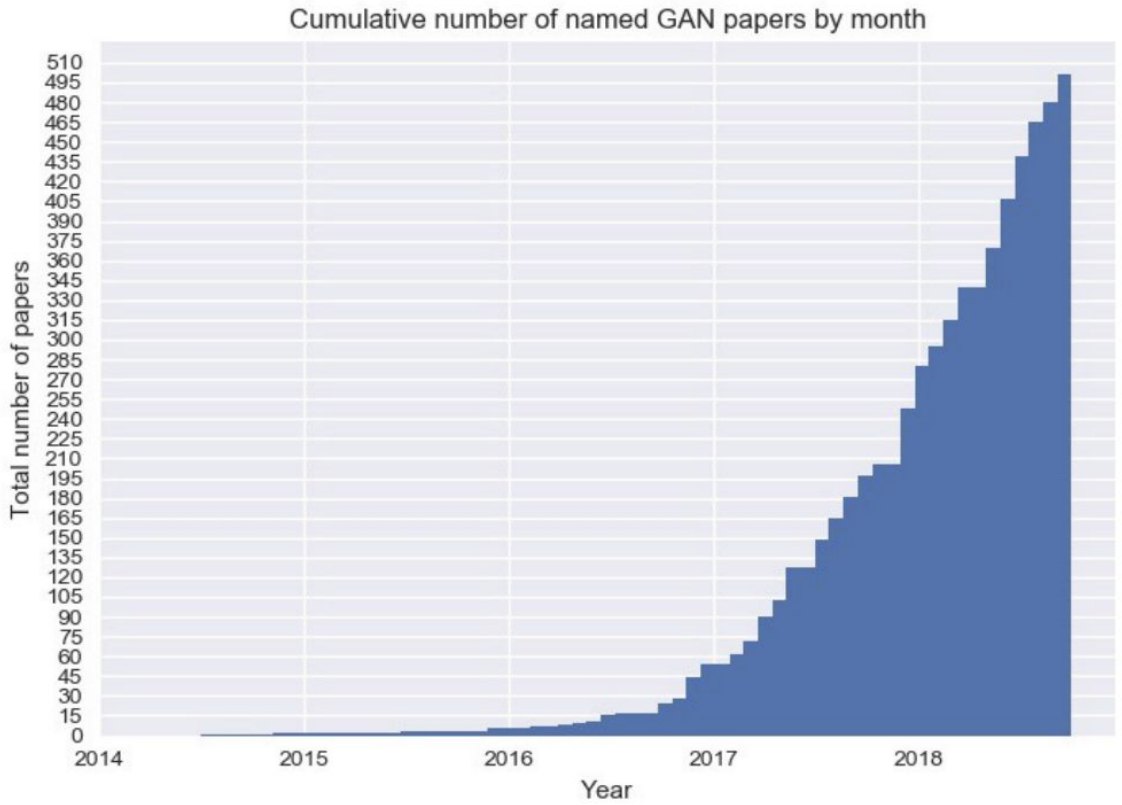
"The GAN Zoo"

- 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

https://github.com/hindupuravinash/the-gan-zoo

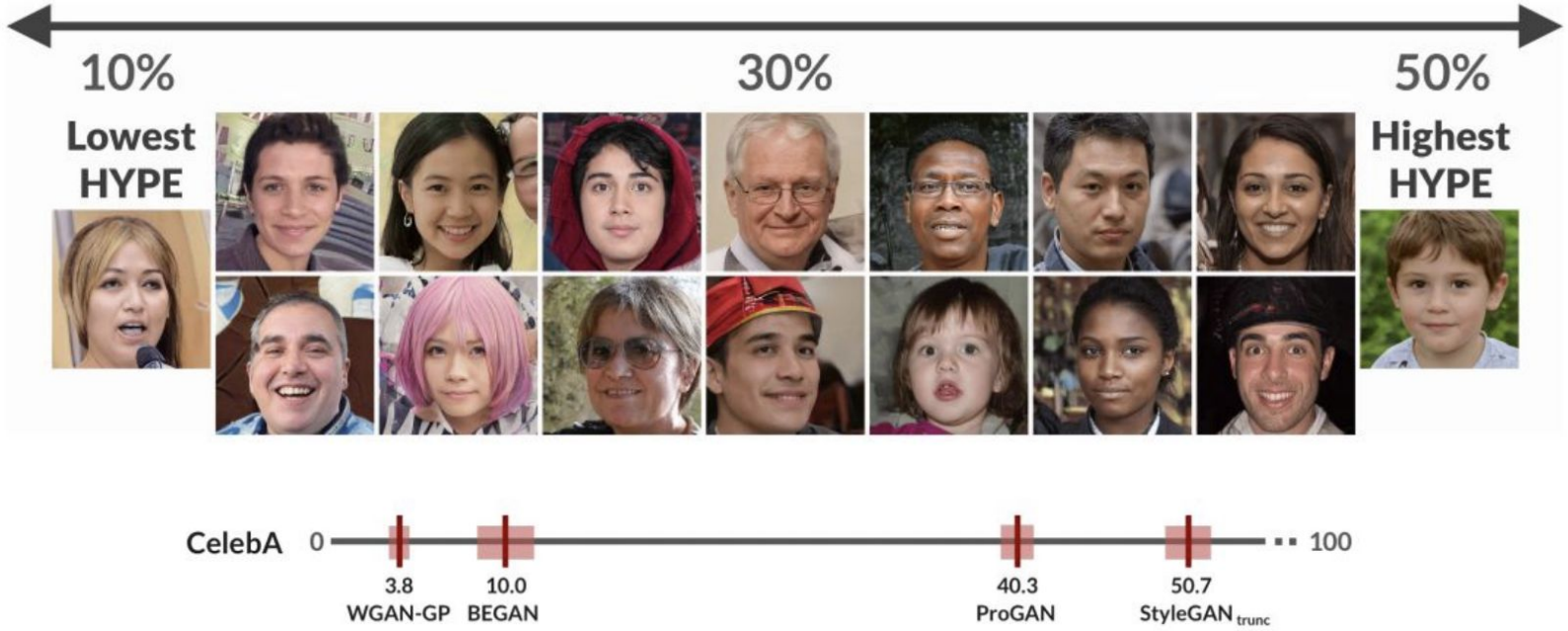
Why to care about GAN!

Introduced in 2014, Approximately 500 GAN papers as of September 2018!



HYPE: Human eYe Perceptual Evaluations

hype.stanford.edu



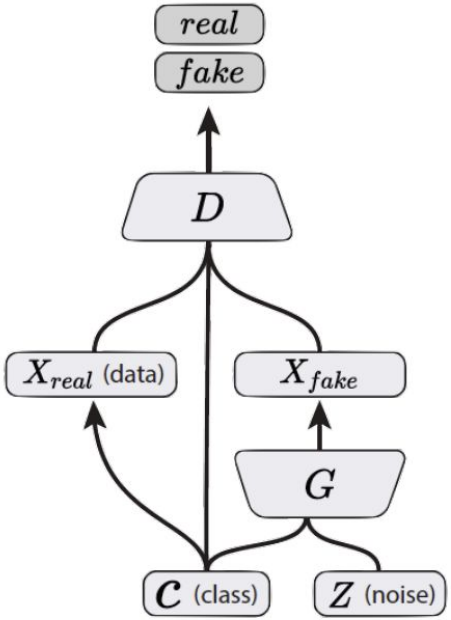
Zhou, Gordon, Krishna et al. HYPE: Human eYe Perceptual Evaluations, NeurIPS 2019

Exercise

1. In face aging cGAN, why do we need *identity preservation*?
2. Draw a block diagram of a Conditional GAN for Super-Resolution.
3. If the generator becomes too strong too quickly, what problem can occur during GAN training?
4. In GAN training, the generator minimizes while the discriminator maximizes the objective. What kind of problem does this setup represent in game theory?
5. Why is the generator in DCGAN designed as an upsampling network with fractionally-strided convolutions, while the discriminator uses standard convolutions?
6. GANs sometimes produce very similar outputs regardless of input noise (mode collapse). Why does mode collapse happen, and how could modifying the discriminator training help reduce it?
7. When the discriminator becomes too strong, the generator struggles to improve. Why does this lead to vanishing gradients for the generator, and what training trick can stabilize this?

Conditional GAN (cGAN)

- Simple modification to the original GAN framework that conditions the model on additional information for better multimodal learning.
- Lends to many practical applications of GANs when we have explicit supervision available.



Conditional GAN
(Mirza & Osindero, 2014)

MNIST digits generated conditioned on their class label.

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

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

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

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

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

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

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

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

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

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

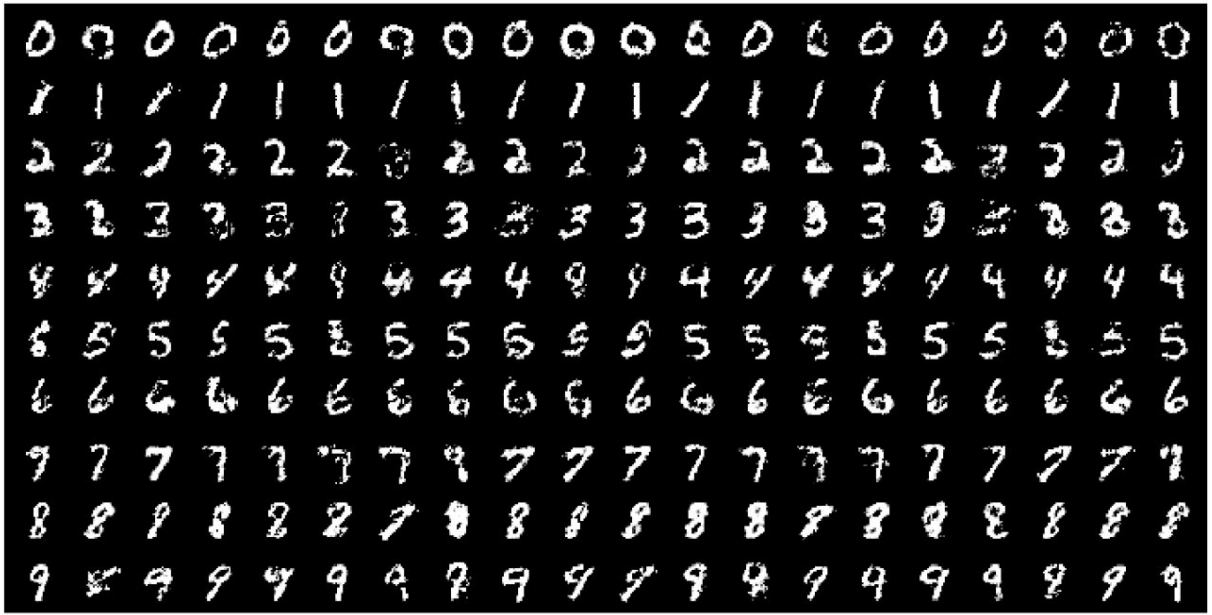


Figure 2 in the original paper.

Mirza, Mehdi, and Simon Osindero. Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784 (2014).

Image-to-Image Translation

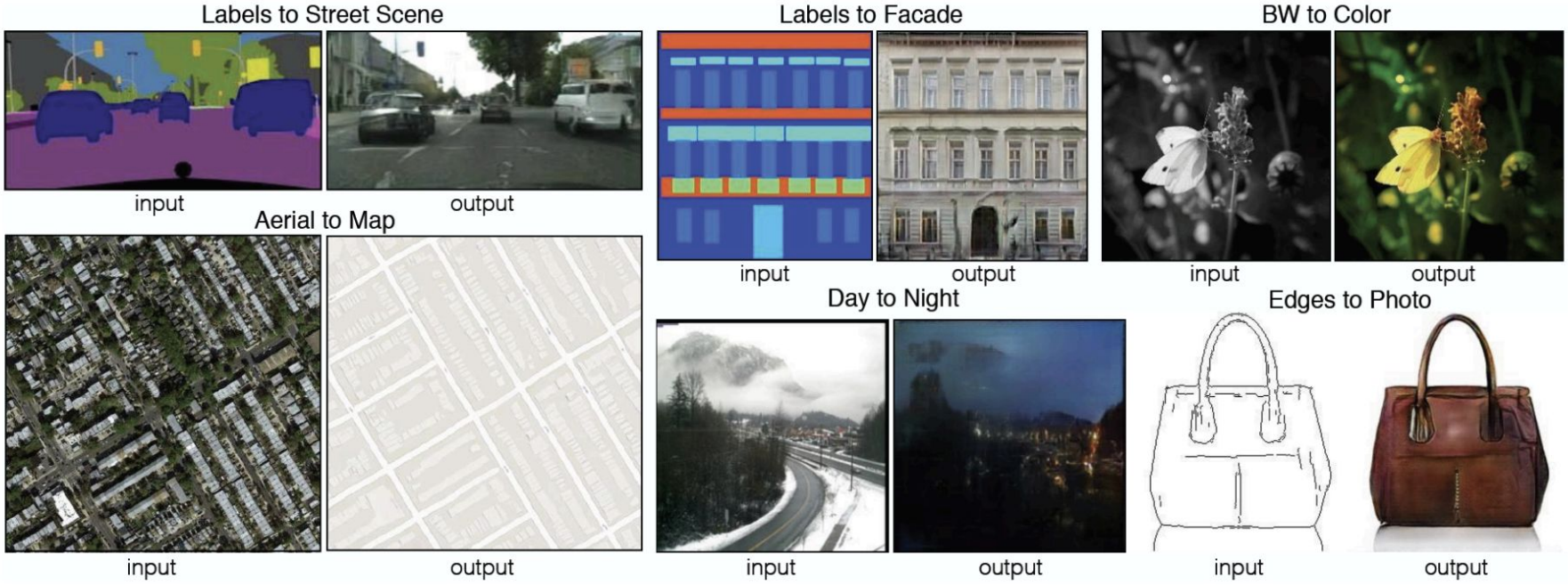


Figure 1 in the original paper.

Image-to-Image Translation

- **Architecture:** DCGAN-based architecture
- Training is conditioned on the images from the source domain.
- Conditional GANs provide an effective way to handle many complex domains without worrying about designing structured loss functions explicitly.

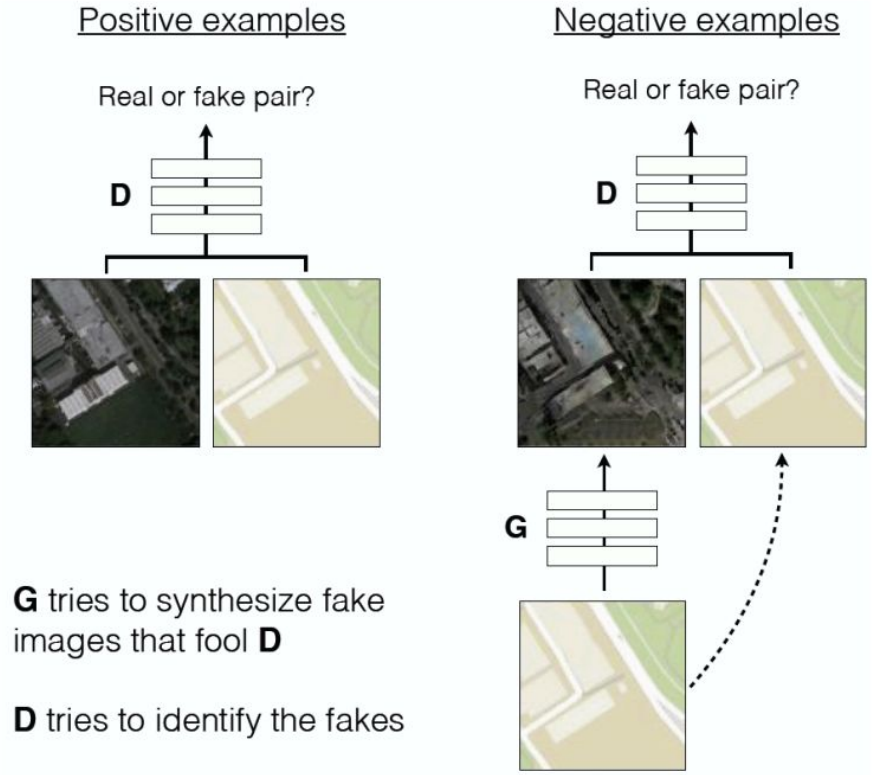


Figure 2 in the original paper.

Text-to-Image Synthesis

- Given a text description, generate images closely associated.
- Uses a conditional GAN with the generator and discriminator being condition on "dense" text embedding.

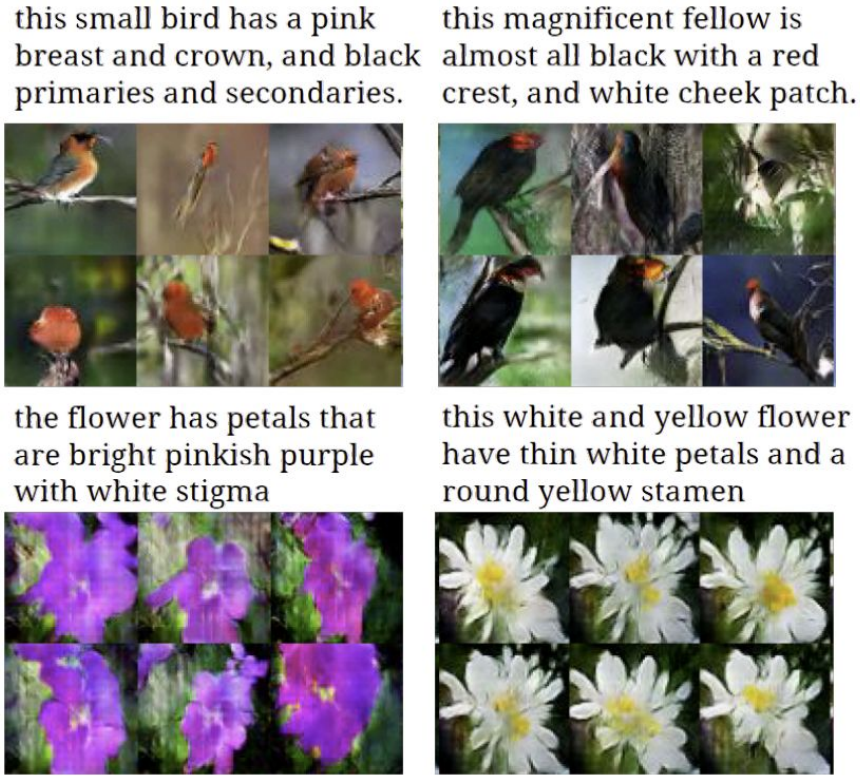


Figure 1 in the original paper.

Text-to-Image Synthesis

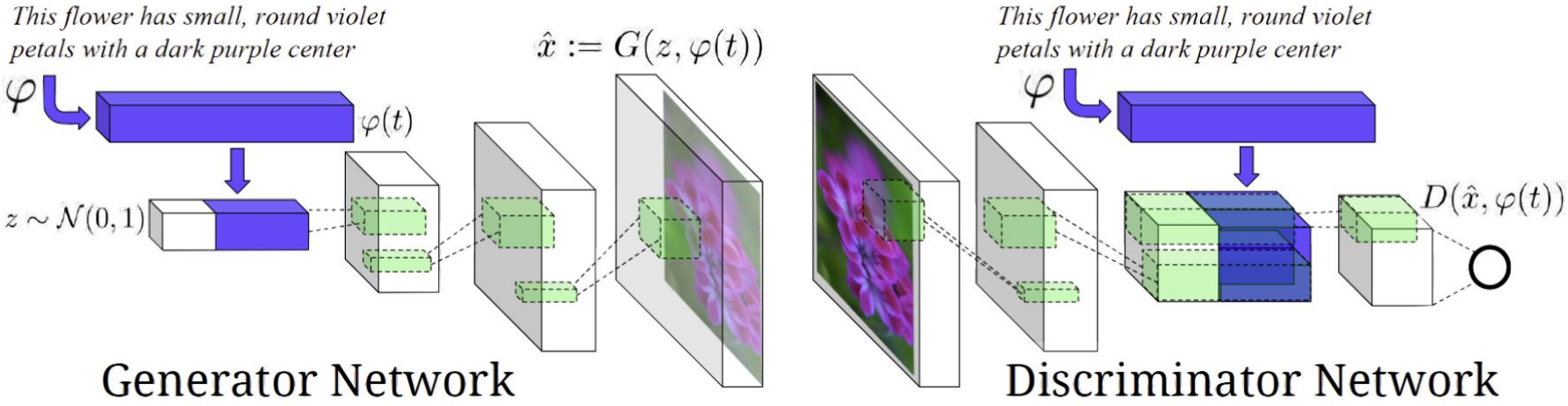


Figure 2 in the original paper.

Positive Example:
Real Image, Right Text

Negative Examples:
Real Image, Wrong Text
Fake Image, Right Text

Reed, S., Akata, Z., Yan, X., Logeswaran, L., Schiele, B., & Lee, H. "**Generative adversarial text to image synthesis**". ICML (2016).

Face Aging with Conditional GANs (Inference Phase)

- **Differentiating Feature:** Uses an Identity Preservation Optimization using an auxiliary network to get a better approximation of the latent code (z*) for an input image.
- Latent code is then conditioned on a discrete (one-hot) embedding of age categories.

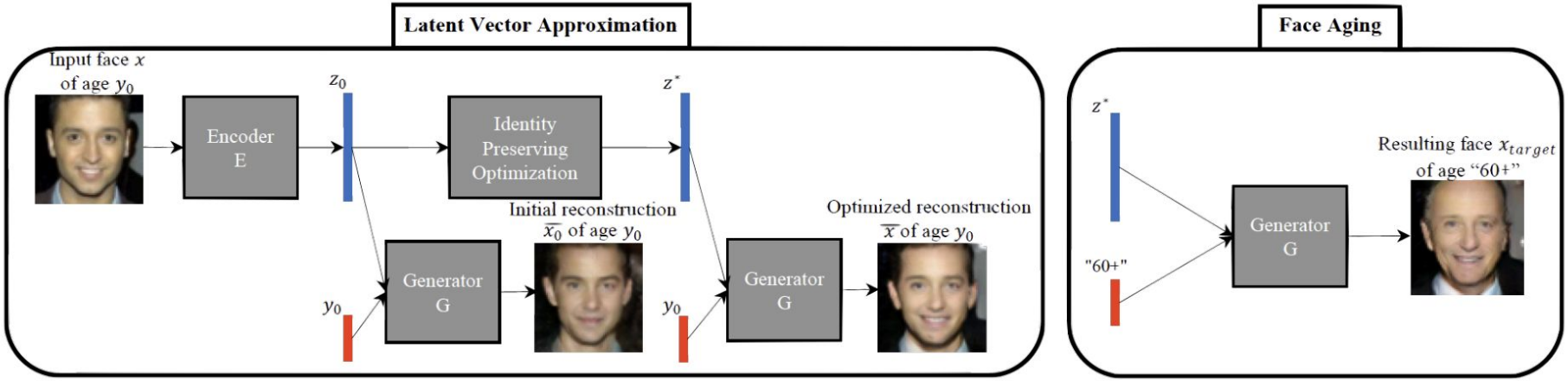


Figure 1 in the original paper.

Antipov, G., Baccouche, M., & Dugelay, J. L. (2017). "**Face Aging With Conditional Generative Adversarial Networks**". arXiv preprint arXiv:1702.01983.

Face Aging with Conditional GANs

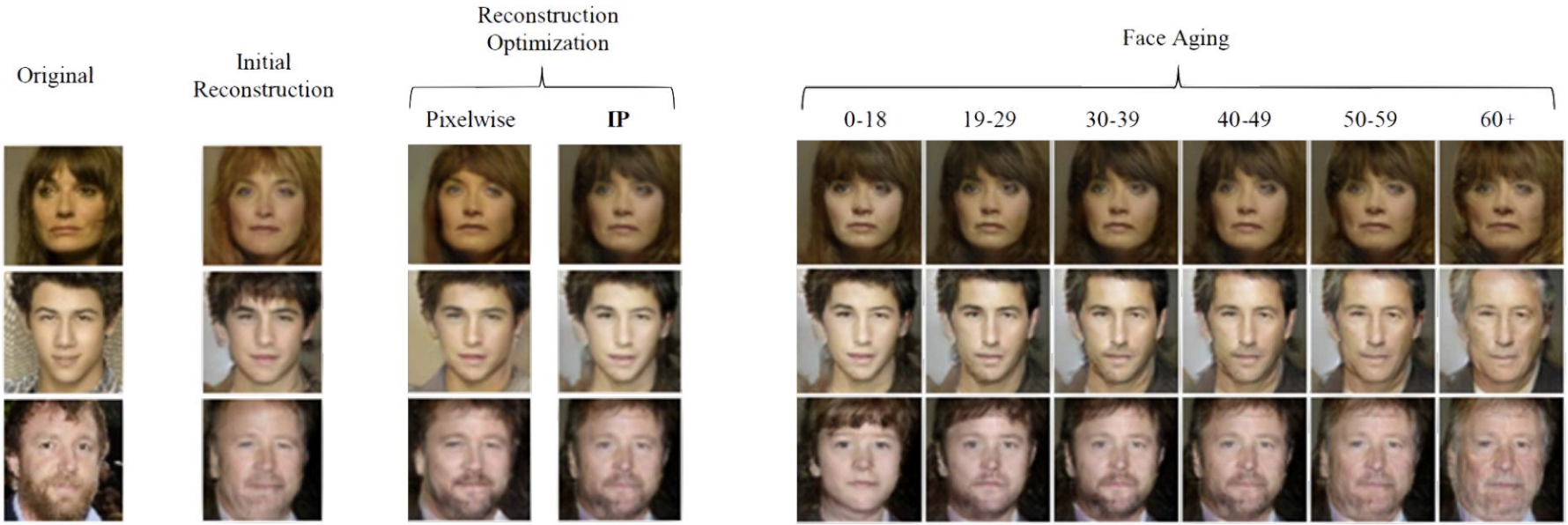


Figure 3 in the original paper.

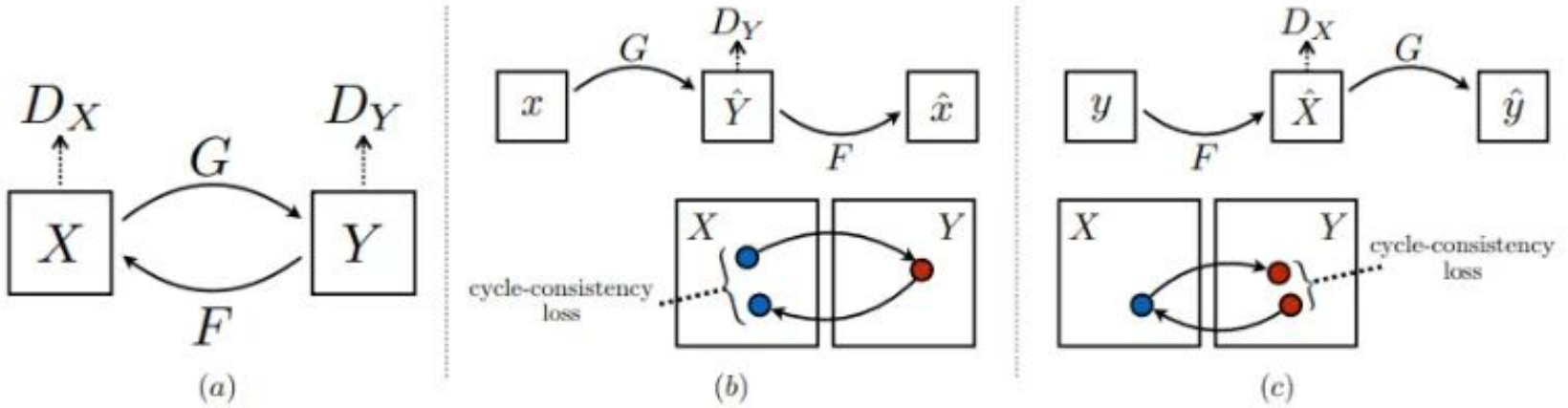
Antipov, G., Baccouche, M., & Dugelay, J. L. (2017). "**Face Aging With Conditional Generative Adversarial Networks**". arXiv preprint arXiv:1702.01983.

Or maybe you want to put a smile on Agent 42's face with the virally popular Faceapp.

image-to-image translation

Cycle GAN

- Learns transformations across domains with unpaired data.
- It works by learning the mapping between two different image domains, such as photographs and sketches, by training a Generative Adversarial Network (GAN) on a dataset.



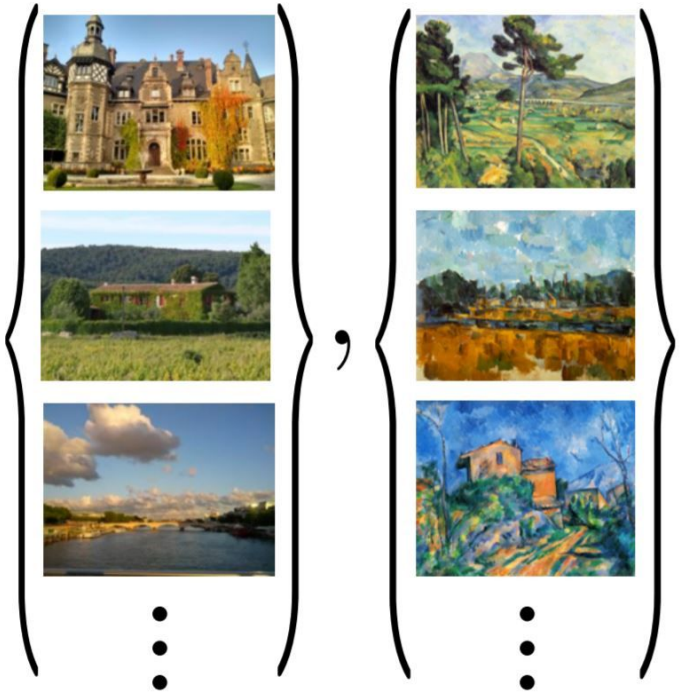
Unpaired Image-to-Image Translation

Paired

x_i y_i

Unpaired

X Y



no one-to-one correspondence between them.

Cycle GAN: Architecture (Forward Cycle)

Start

Original Images

Input_A

Discriminator A

Generator A2B

Decision [0,1]

Generated_B

Discriminator B

Decision [0,1]

Original Images

cycle-consistency loss:

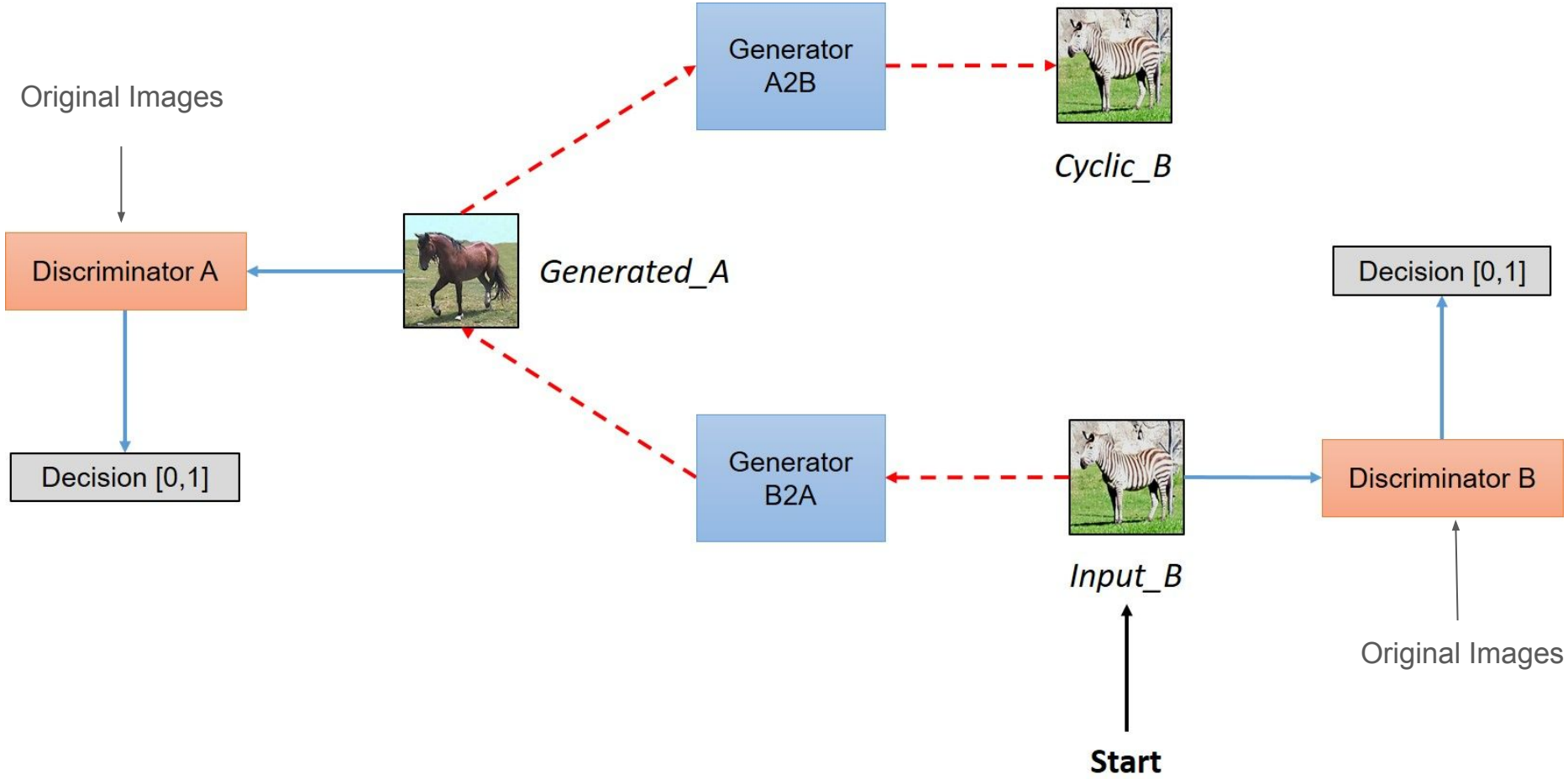
$$L_{cyc} = ||Cyclic\_A - Input\_A||_1$$

Cyclic_A

Generator B2A

To enforce cycle consistency, the generated zebra (Generated_B) is passed back through the other generator, Generator B2A.

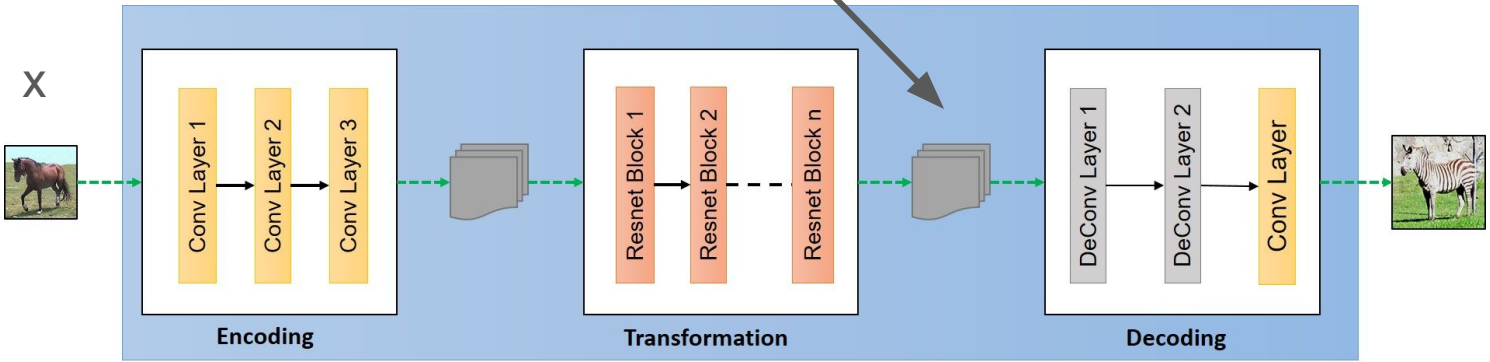
Cycle GAN: Architecture (Reverse Cycle)



High Level Structure of Generator

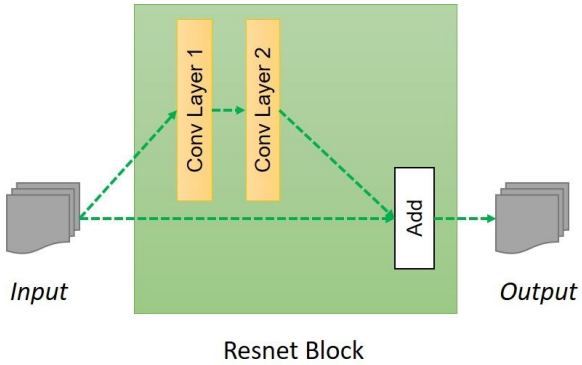
$$\text{Output} = F(x) + x$$

F(x)= transformation learned by the conv layers

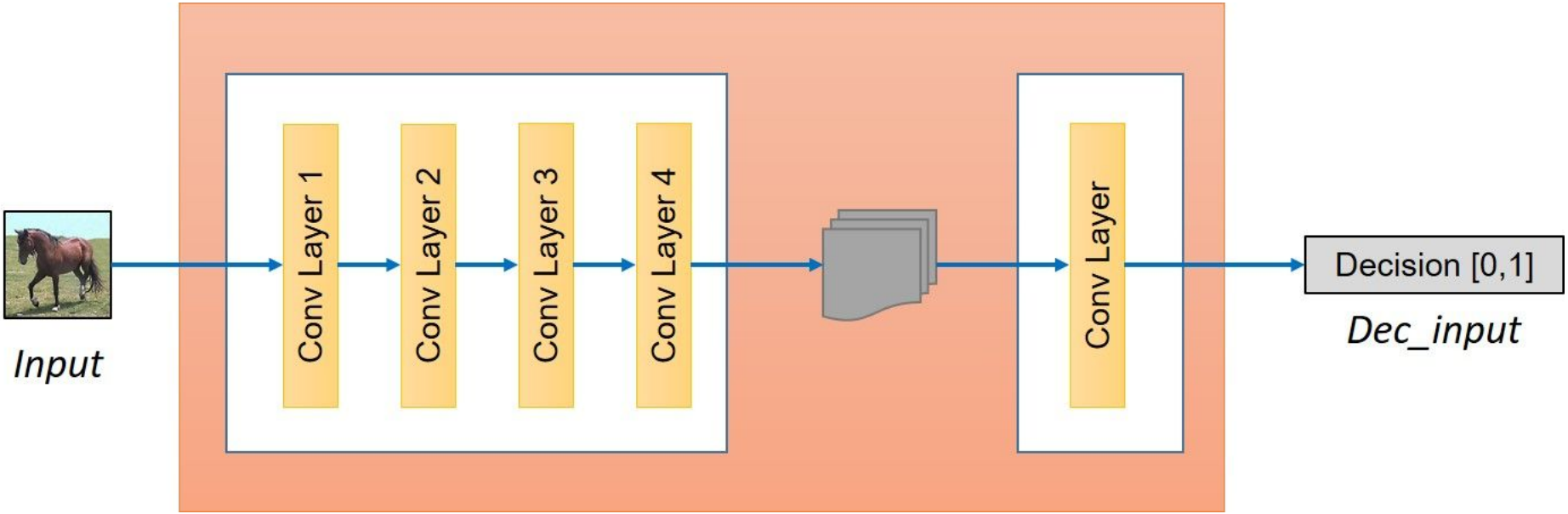


Input feature map → Conv Layer 1 → Conv Layer 2 → Add skip connection (Input + Output) → Result.

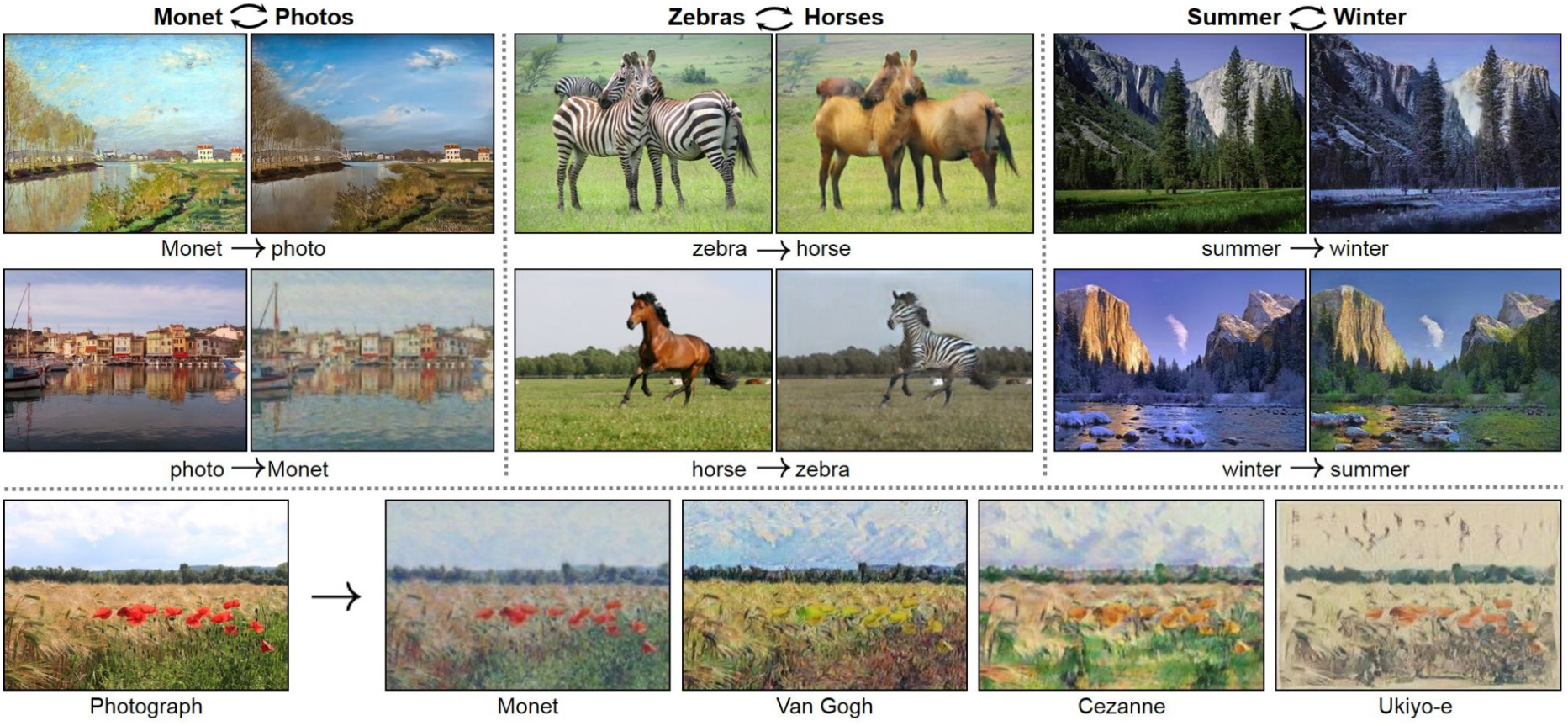
Keeps identity information from the original while allowing transformation.



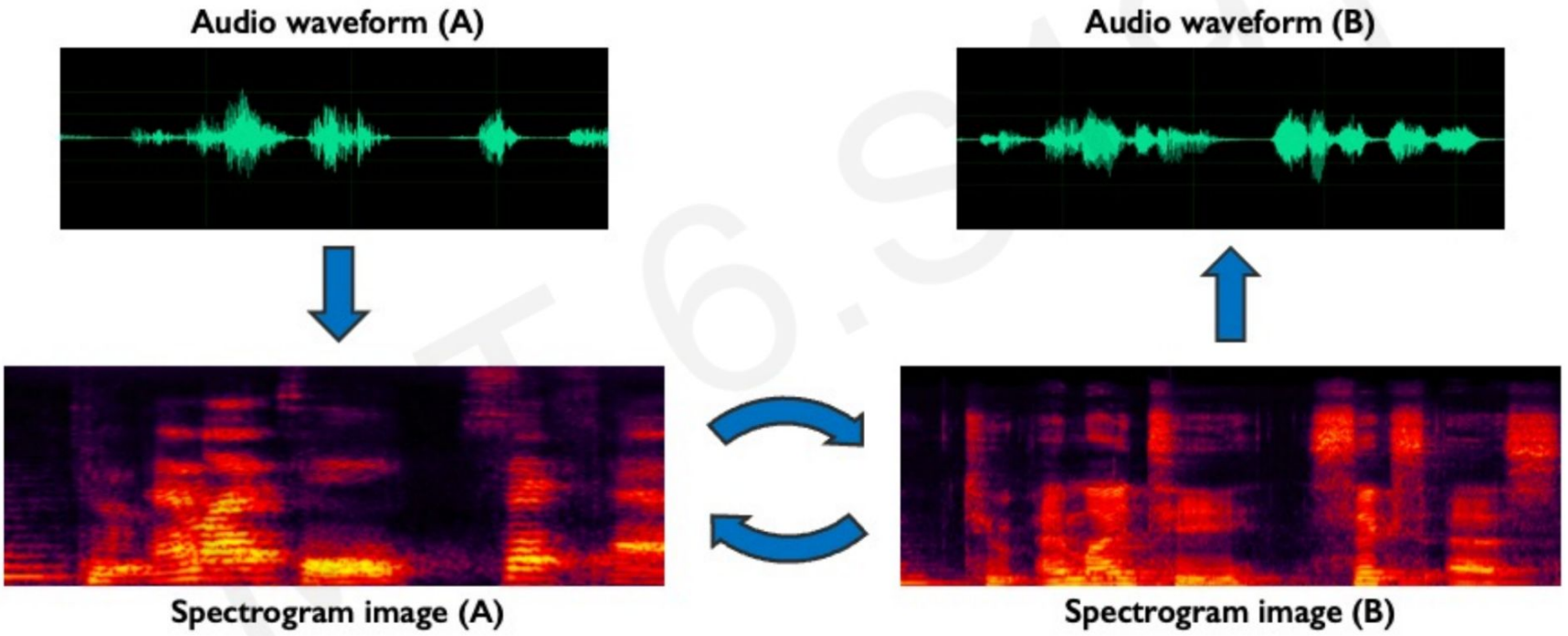
High Level Structure of Discriminator

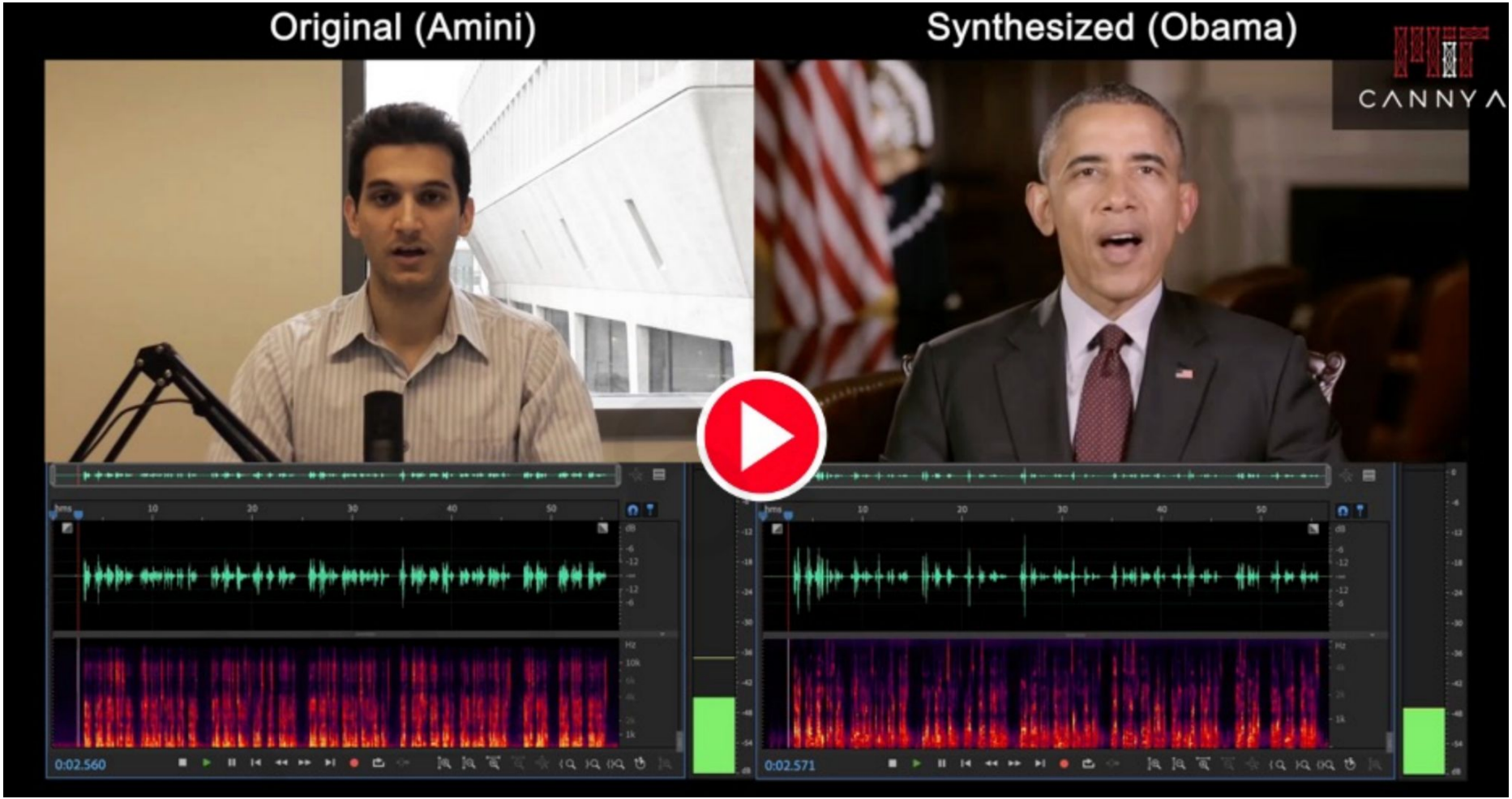


Application: Image-to-Image Translation



Application: Speech Transformation




Original (Amini)
Synthesized (Obama)
CANNYA
0:02.560
0:02.571

Coupled GAN (CoGAN)

- Designed for joint image generation across two different domains without paired training data.

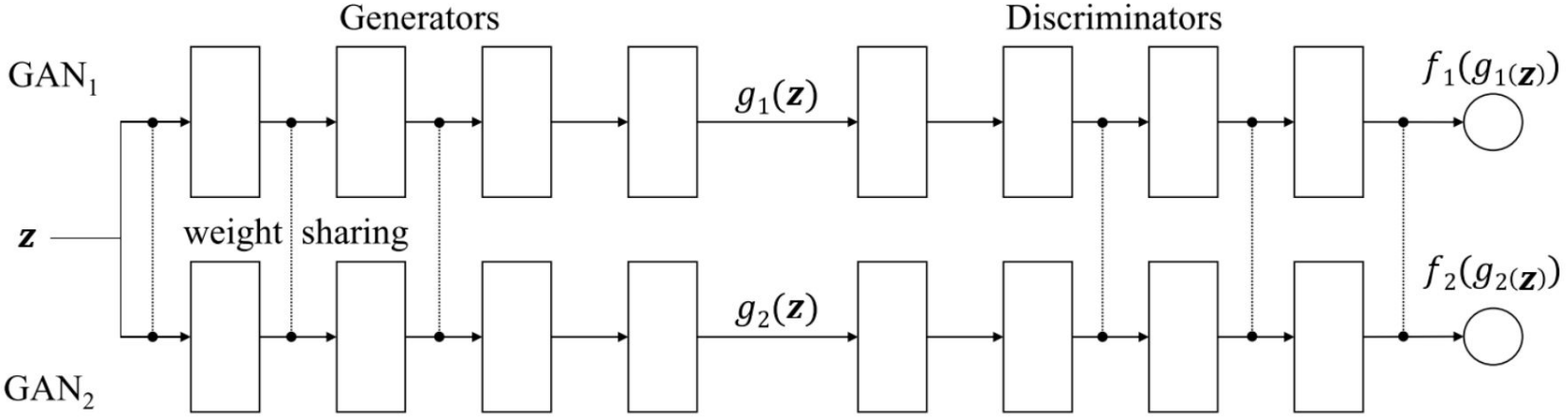


Figure 1 of the original paper.

Generators share some of their weights in the early layers.
Reason:

- Early layers learn high-level abstract representations (shapes, structures, poses).
- These should be common across domains (e.g., both a color face and a sketch have the same pose and facial structure).

Liu, Ming-Yu, and Oncel Tuzel. "Coupled generative adversarial networks". NIPS (2016).

Generators

- Both $g1$ and $g2$ are realized as multilayer perceptrons (MLP)

$$g_1(\mathbf{z}) = g_1^{(m_1)}\left(g_1^{(m_1-1)}\left(\dots g_1^{(2)}\left(g_1^{(1)}(\mathbf{z})\right)\right)\right), \quad g_2(\mathbf{z}) = g_2^{(m_2)}\left(g_2^{(m_2-1)}\left(\dots g_2^{(2)}\left(g_2^{(1)}(\mathbf{z})\right)\right)\right)$$

- where $g(i)1$ and $g(i)2$ are the ith layers of $g1$ and $g2$ and $m1$ and $m2$ are the numbers of layers in $g1$ and $g2$.
- Through layers of perceptron operations, the generative models gradually decode information from more abstract concepts to more material details.
- The first layers decode high-level semantics and the last layers decode low-level details.
- No constraints are enforced to the last layers.

Discriminator

- The discriminative models map an input image to a probability score, estimating the likelihood that the input is drawn from a true data distribution.
- The first layers of the discriminative models extract low-level features, while the last layers extract high-level features.
- Similar to generator, the last layers are weight shared.

But it is later found out that it does not help much on the quality of the synthesized images. But still, the weight sharing is used.

This is because ***the weight-sharing constraint in the discriminators helps reduce the total number of parameters in the network, though it is not essential for learning a joint distribution.***

Learning

- In the game, there are two teams and each team has two players.

$$\max_{g_1,g_2} \min_{f_1,f_2} V(f_1, f_2, g_1, g_2), \quad \text{subject to} \quad \boldsymbol{\theta}_{g_1^{(i)}} = \boldsymbol{\theta}_{g_2^{(i)}}, \quad \text{for } i = 1, 2, ..., k$$

$$\boldsymbol{\theta}_{f_1^{(n_1-j)}} = \boldsymbol{\theta}_{f_2^{(n_2-j)}}, \quad \text{for } j = 0, 1, ..., l-1$$

- Same as GAN, CoGAN can be trained by back propagation with the alternating gradient update steps.

image-to-image translation