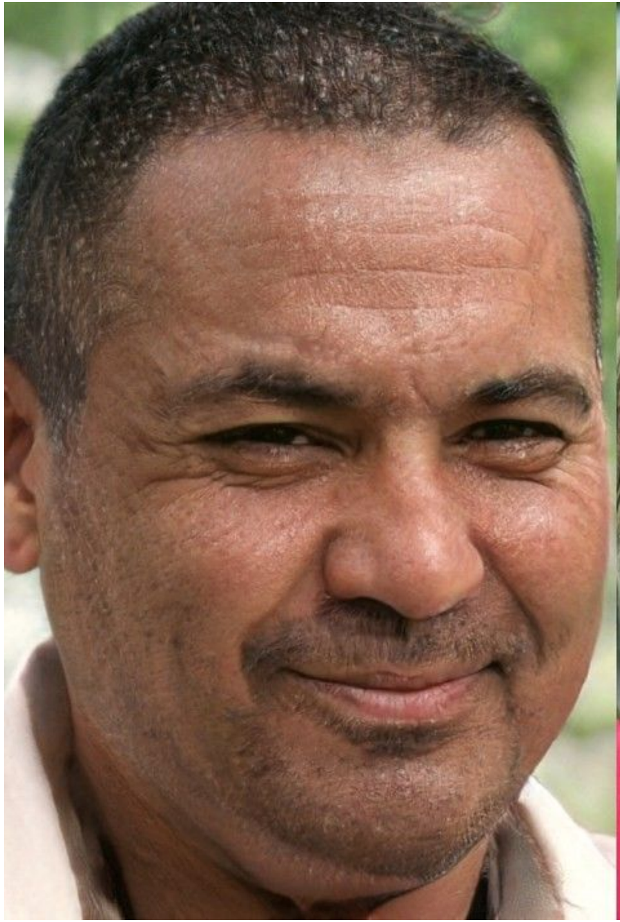
GANs

Fred Bunt, Jack Roddy



4.5 years of progress on faces

Unsupervised learning

- In supervised learning, we are trying to learn P(y|x)
    - For example, we learn to predict some class (y) given an input image (x)

Unsupervised learning

- In supervised learning, we are trying to learn P(y|x)
  - For example, we learn to predict some class (y) given an input image (x)

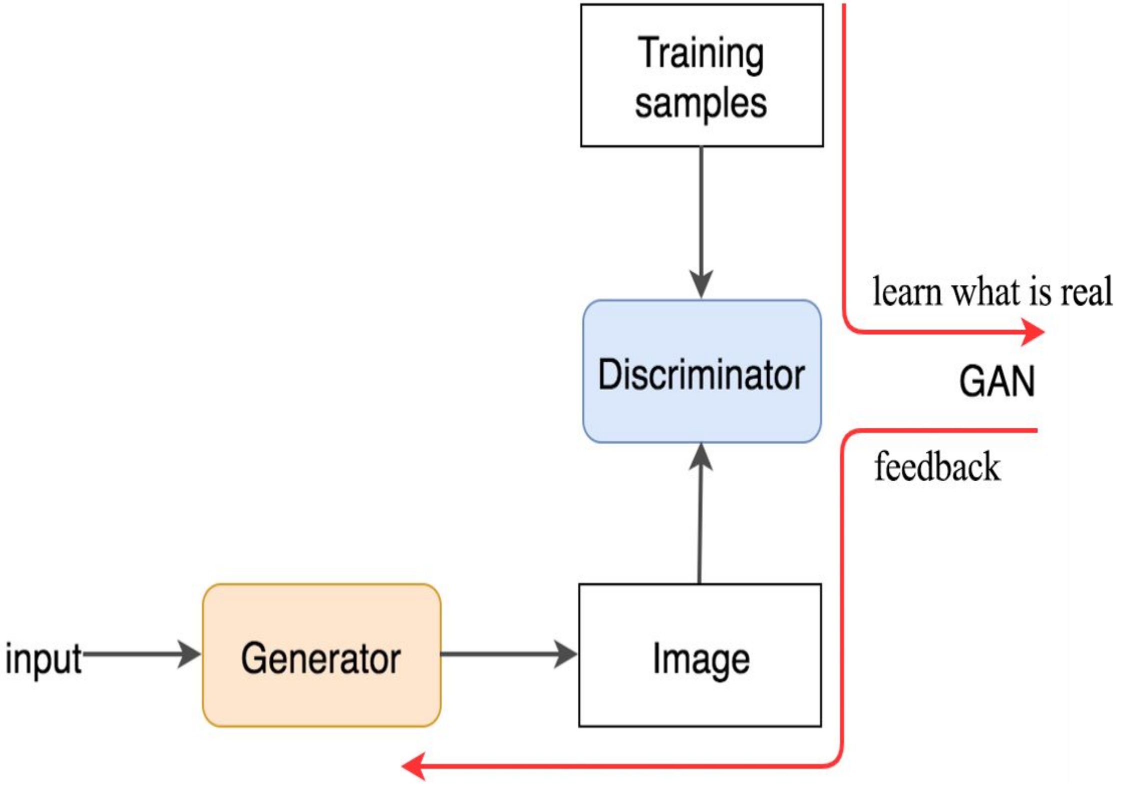
- In unsupervised learning, we are trying to learn P(x)
  - We are trying to approximate the distribution across images

What is a GAN?

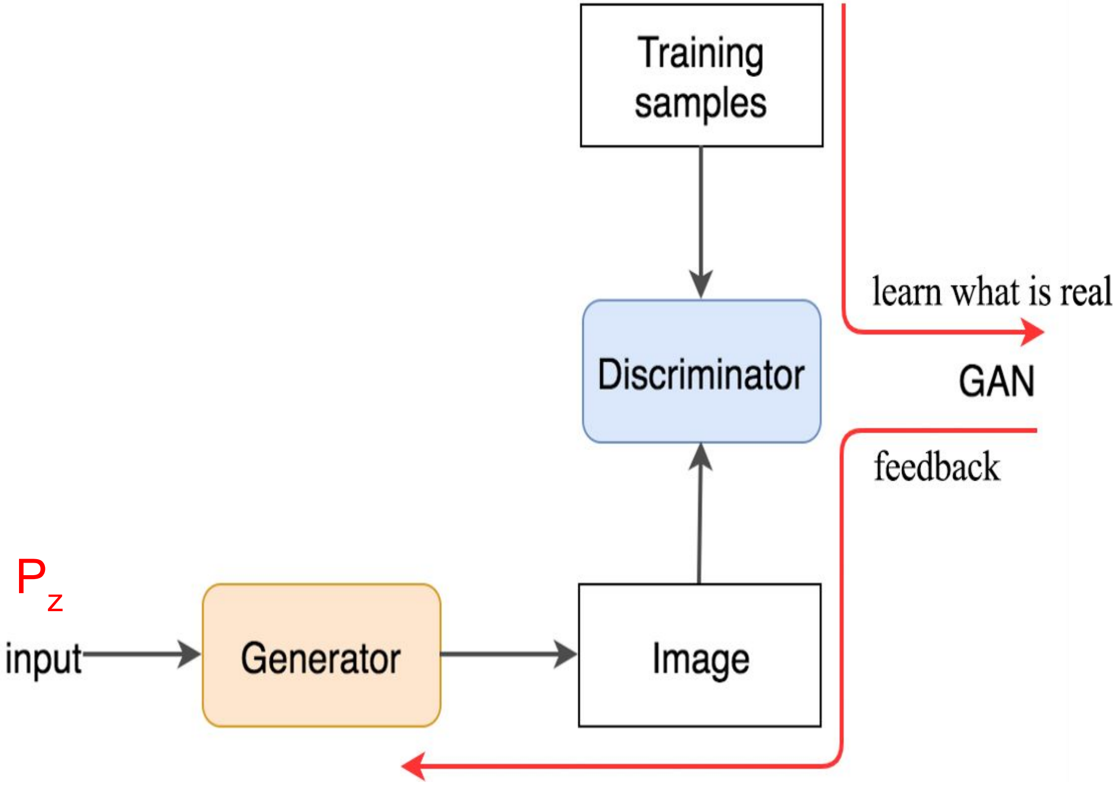
- Two networks trained together
  - One network is a “generator”
  - The other is a “discriminator”

What is a GAN?

- Two networks trained together
    - One network is a “generator”
    - The other is a “discriminator”
- The generator is like a counterfeiter, and the discriminator is like the police
- The generator tries to fabricate convincing output, and the discriminator tries to label output as real or fake

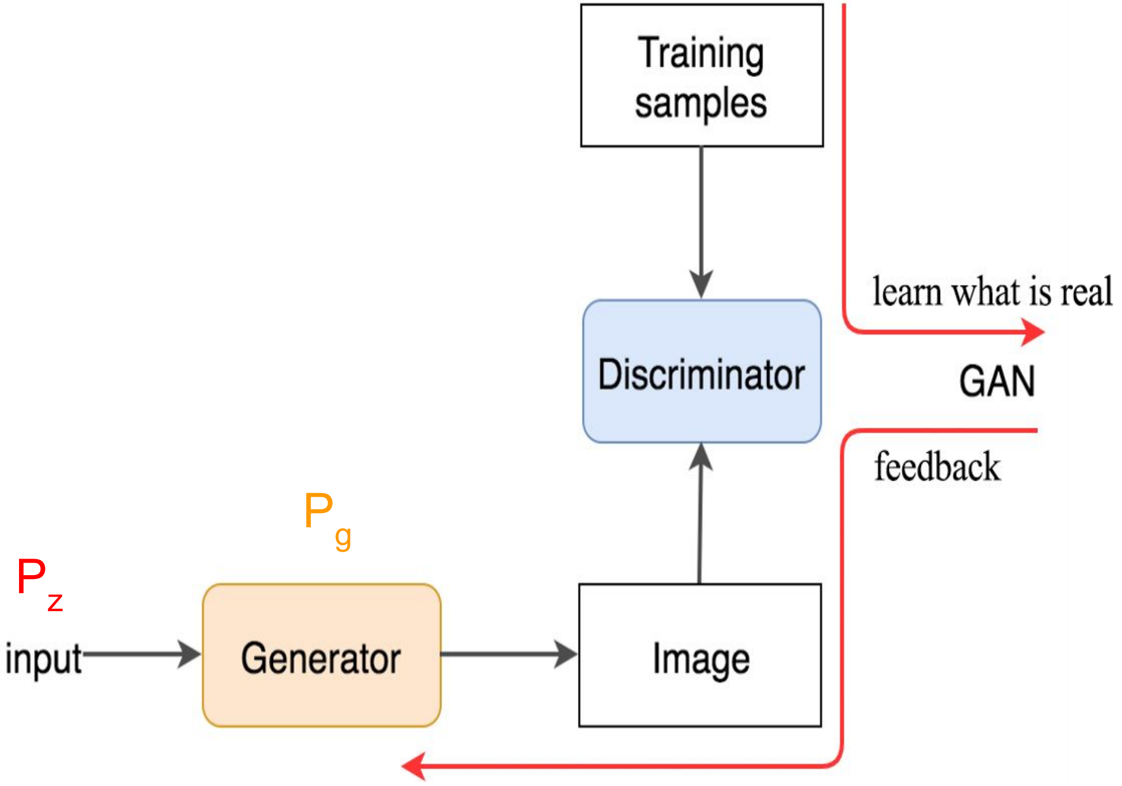

Training samples
learn what is real
Discriminator
GAN
feedback
input
Generator
Image

P_z Prior distribution on input noise variables

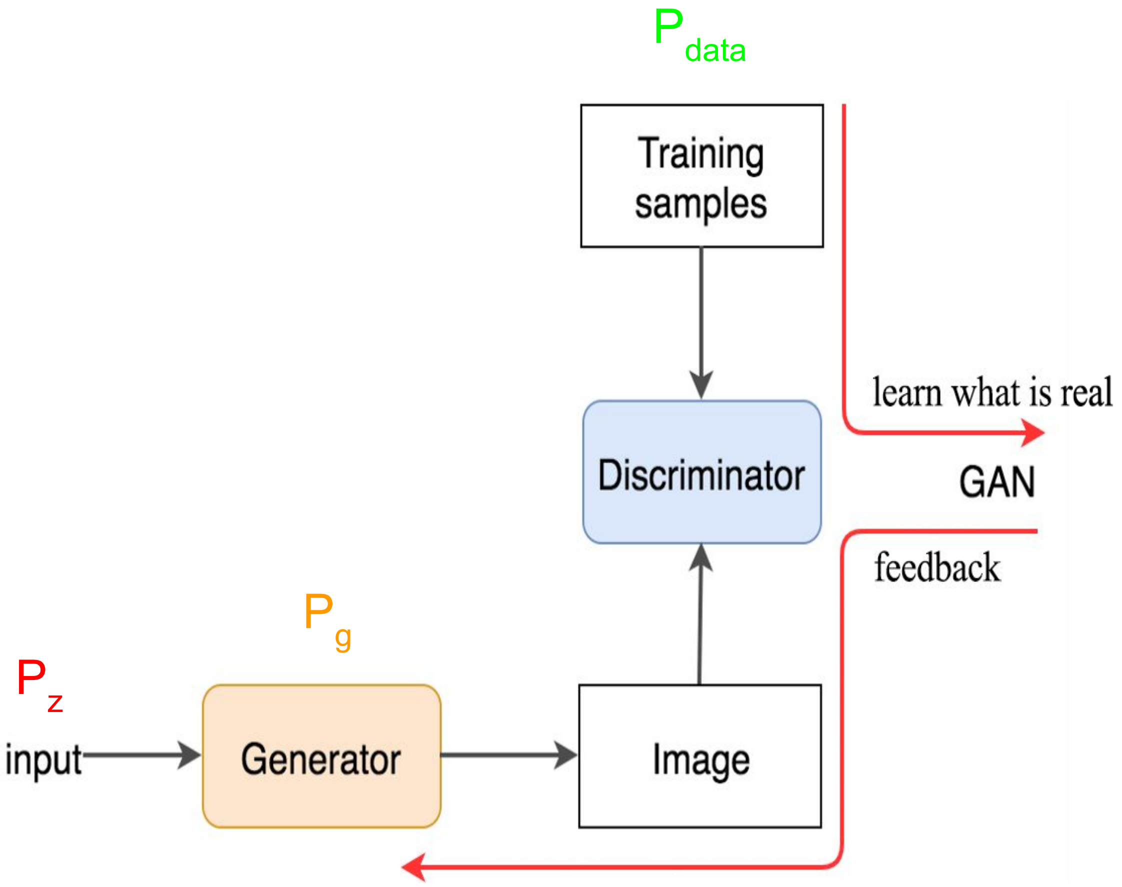


P_z Prior distribution on input noise variables

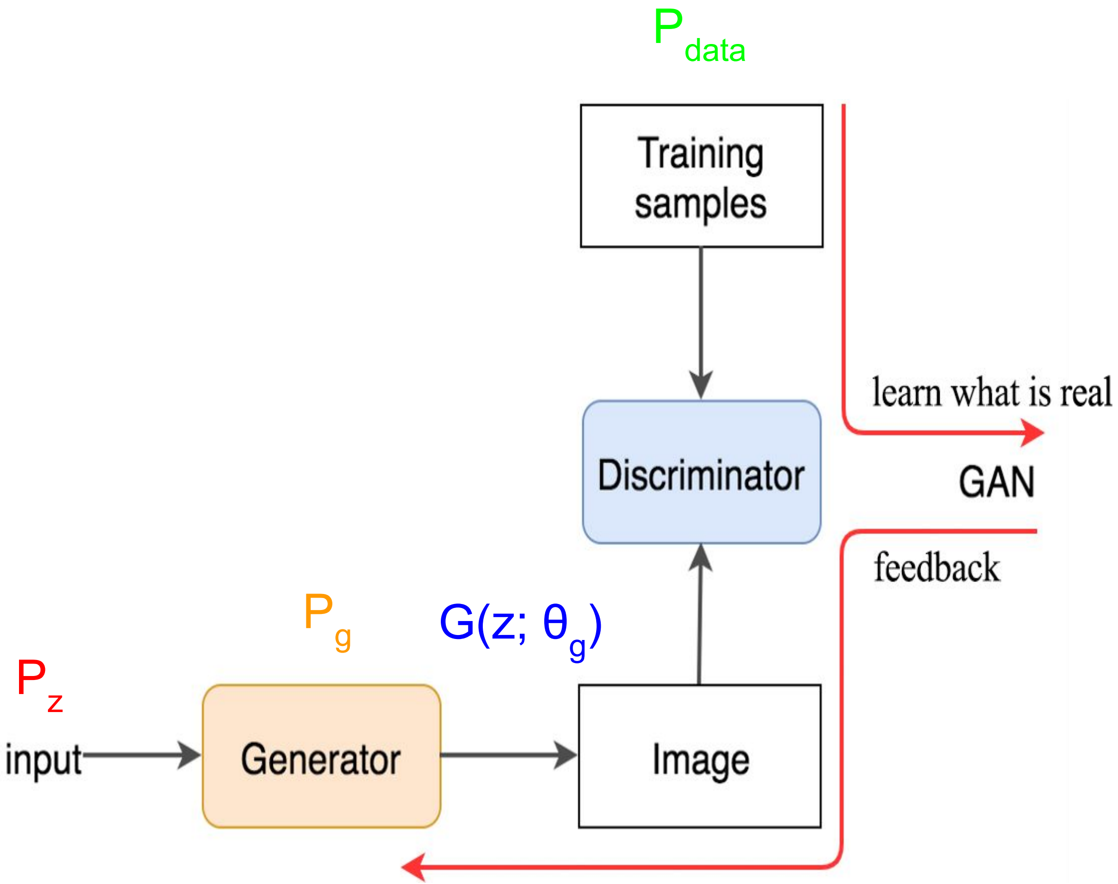
P_g The distribution that the generator learns



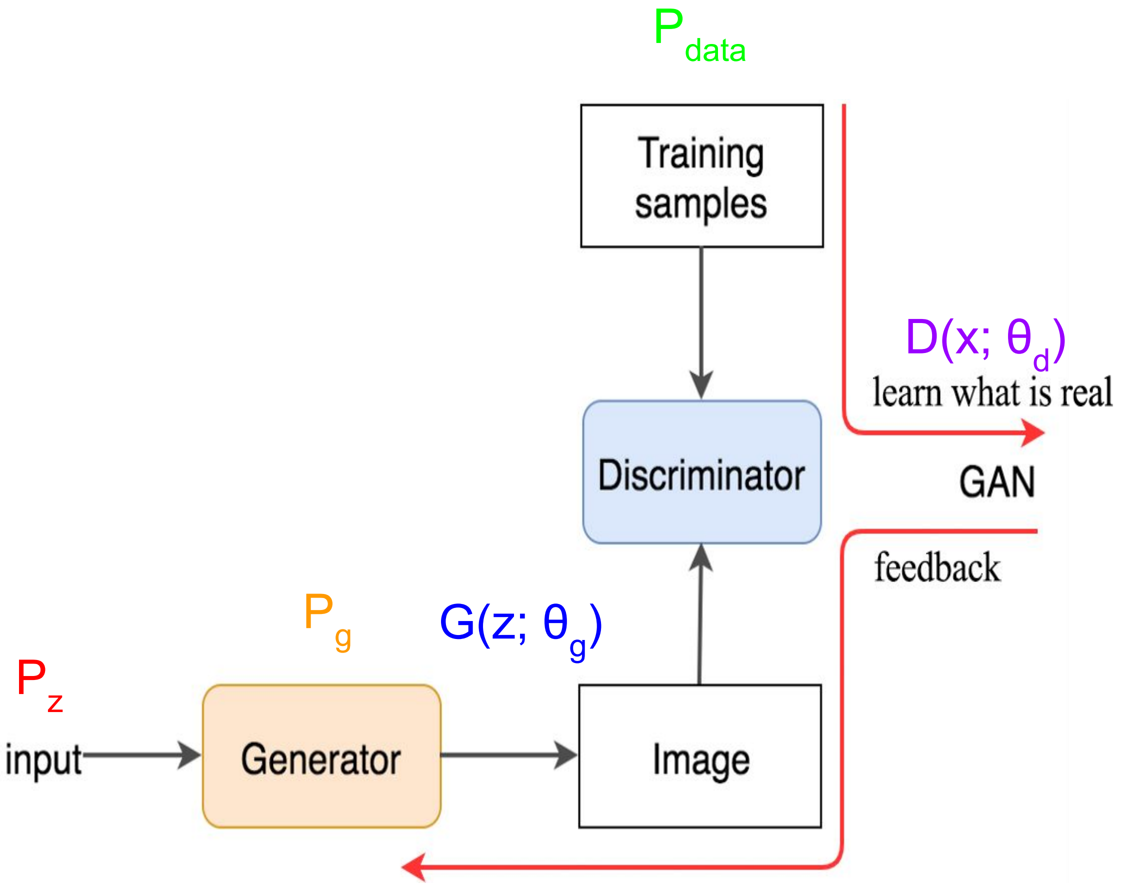
P_z	Prior distribution on input noise variables
P_g	The distribution that the generator learns
P_{data}	The true distribution of data that the generator tries to approximate



P_z	Prior distribution on input noise variables
P_g	The distribution that the generator learns
P_{data}	The true distribution of data that the generator tries to approximate
$G(z; \theta_g)$	The generator network, maps from z to image



P_z	Prior distribution on input noise variables
P_g	The distribution that the generator learns
P_{data}	The true distribution of data that the generator tries to approximate
$G(z; \theta_g)$	The generator network, maps from z to image
$D(x; \theta_d)$	The discriminator network, outputs scalar probability



Objective functions

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

Objective functions

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

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

$$\min_G V(G) = \mathbb{E}_{\boldsymbol{z} \sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$$

Discriminator objective

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

Discriminator objective

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

Discriminator objective

$\log D(\boldsymbol{x})$

- Log probability that x is a real image

Discriminator objective

$$\log D(\boldsymbol{x})$$

- Log probability that x is a real image

$$\mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})]$$

- Average of the log probability for all real images in our dataset

Discriminator objective

$$\log D(\boldsymbol{x})$$

- Log probability that x is a real image

$$\mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})]$$

- Average of the log probability for all real images in our dataset

$$\max_{D} V(D) = \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})]$$

- Try to maximize this expected value

Discriminator objective

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

Discriminator objective

$$\max_D V(D) = \underbrace{\mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})]}_{\text{recognize real images better}} + \underbrace{\mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]}_{\text{recognize generated images better}}$$

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

Generator objective

$$\min_G V(G) = \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$$

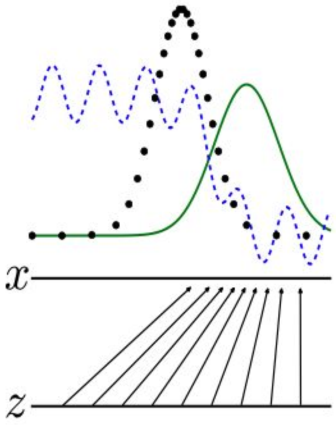
Generator objective

$$\min_G V(G) = \underline{\mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]}$$

Optimize G that can fool the discriminator the most.

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

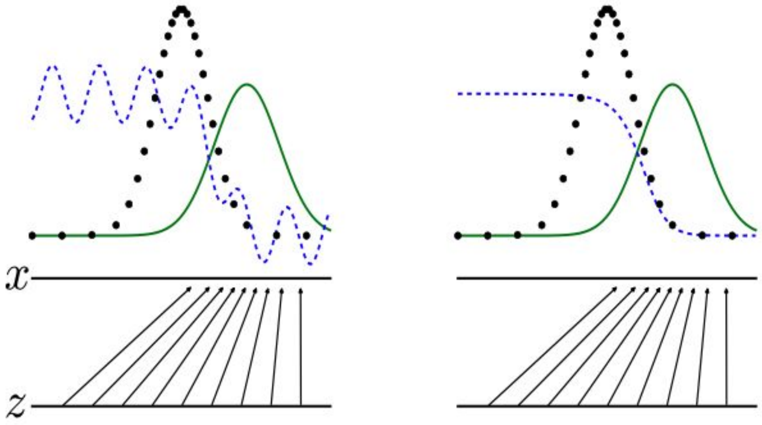
Training - initial conditions



(a)

- Generator poorly approximates the true distribution
- Discriminator doesn’t do the best job

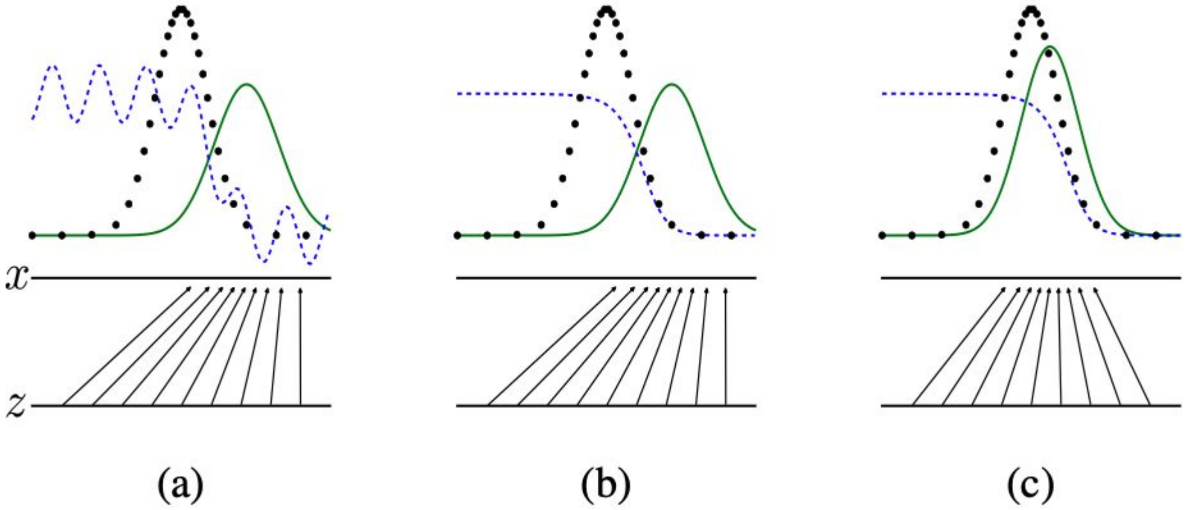
Training - discriminator



(a) (b)

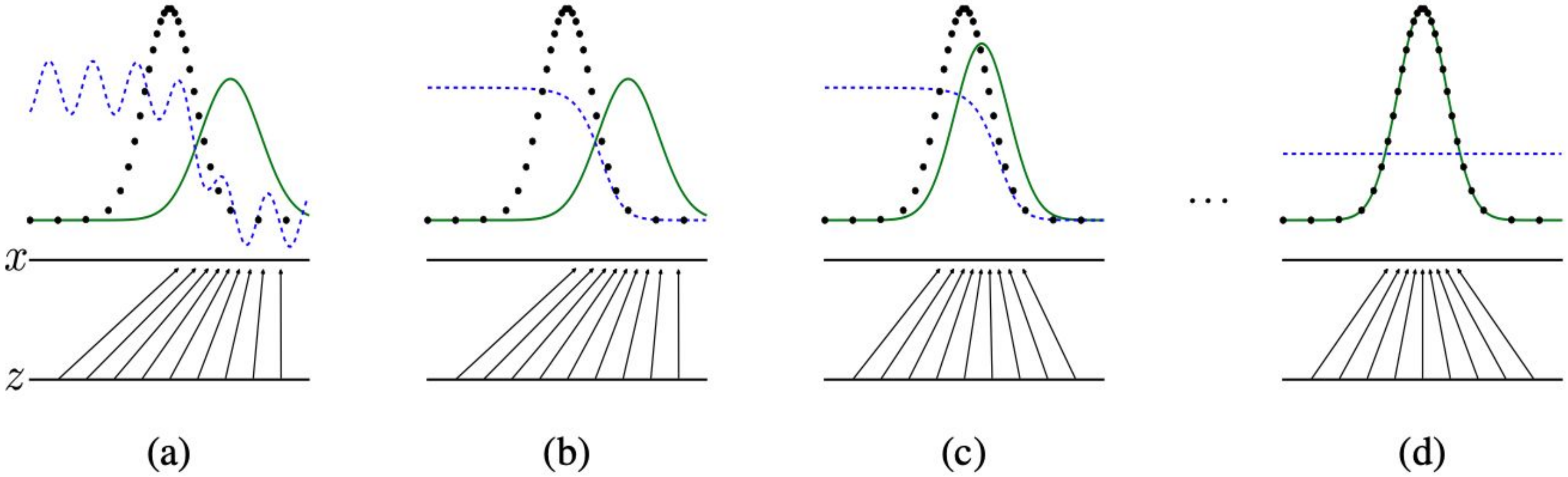
- Generator hasn’t changed
- Discriminator has been trained to better recognize fake images

Training - generator



- Generator trained to better approximate the true distribution
- Discriminator has not changed

Optimal solution



- Generator distribution is indistinguishable from the true distribution
- Discriminator can't tell the difference

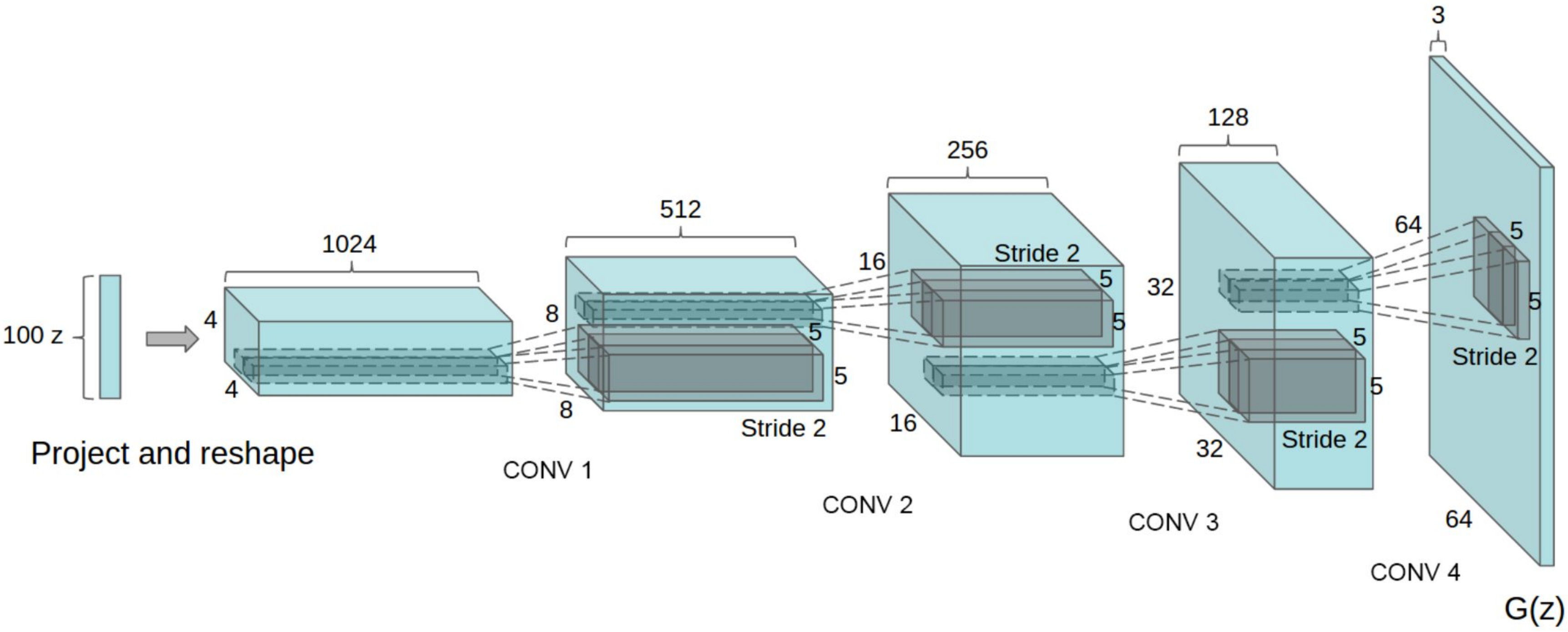
Thinking more about the latent input variables

- Why do we want to establish distinct, prior distributions over the input variables Z?

- What do the input variables eventually end up representing?

DCGAN

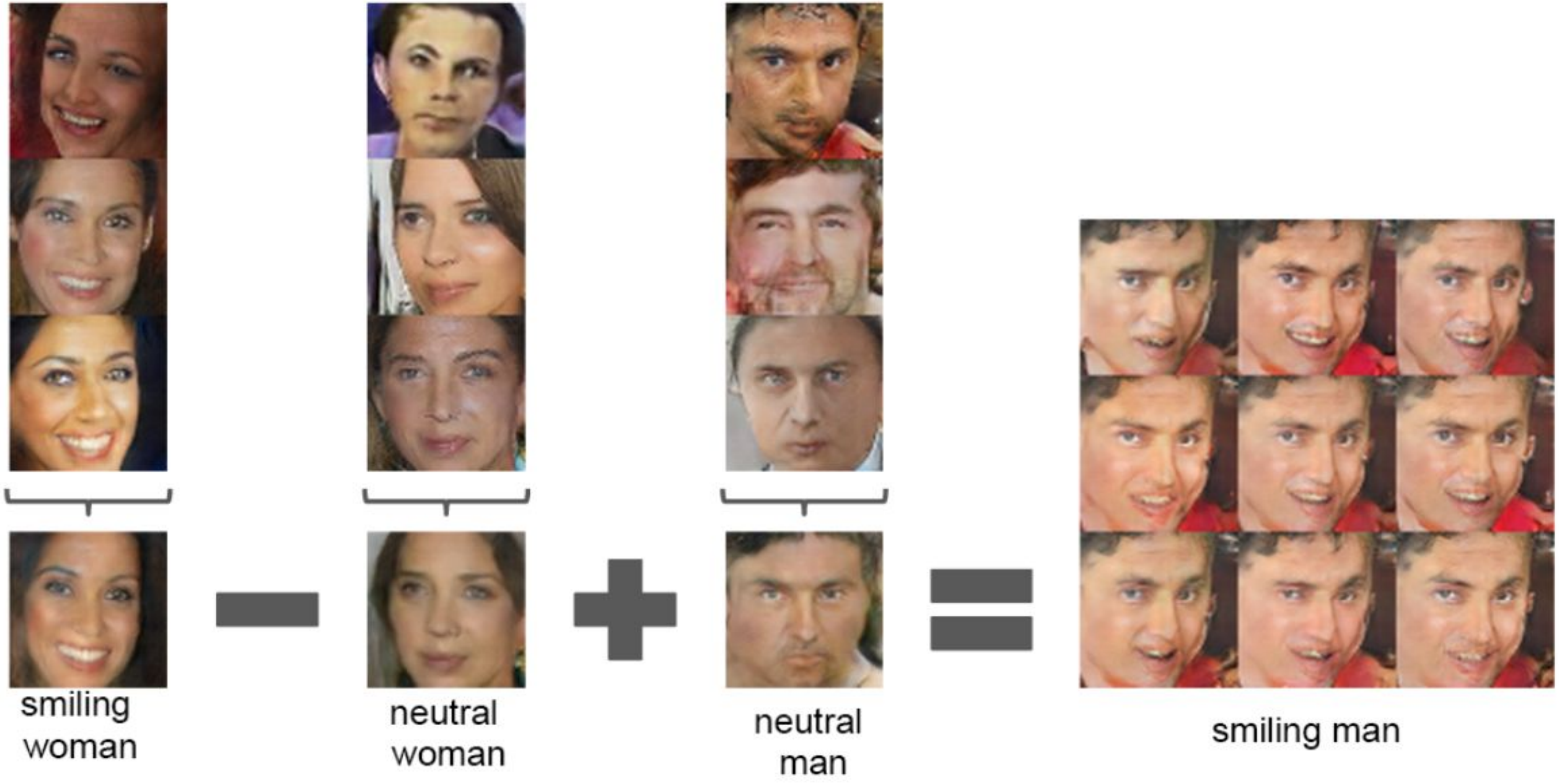
DCGAN architecture



Walking through the latent variable space

Walking through the latent variable space

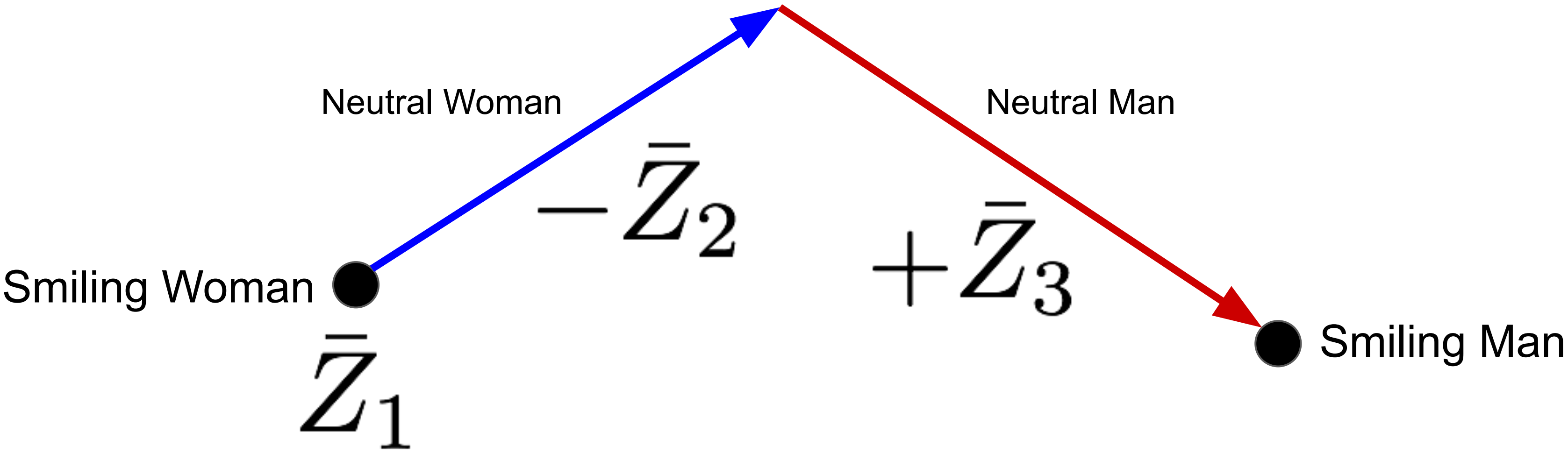
Latent Space Vector Arithmetic

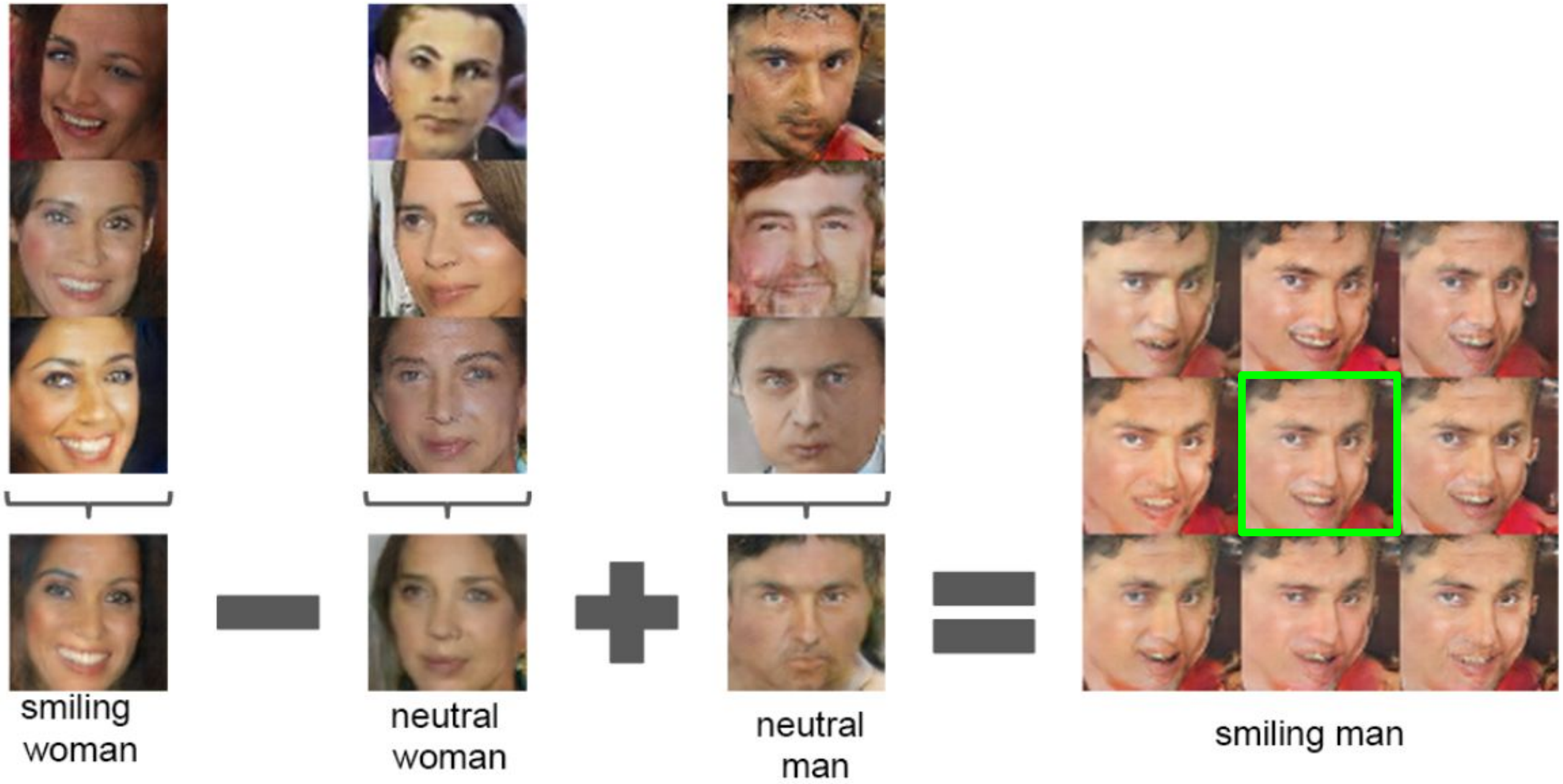


$$G\left(\bar{Z}_1 = \frac{1}{3}\sum_{i=1}^{3} Z_i\right) \quad G\left(\bar{Z}_2 = \frac{1}{3}\sum_{j=1}^{3} Z_j\right) \quad G\left(\bar{Z}_3 = \frac{1}{3}\sum_{k=1}^{3} Z_k\right)$$

$$G(\bar{Z}_1 - \bar{Z}_2 + \bar{Z}_3)$$

Latent space vector arithmetic

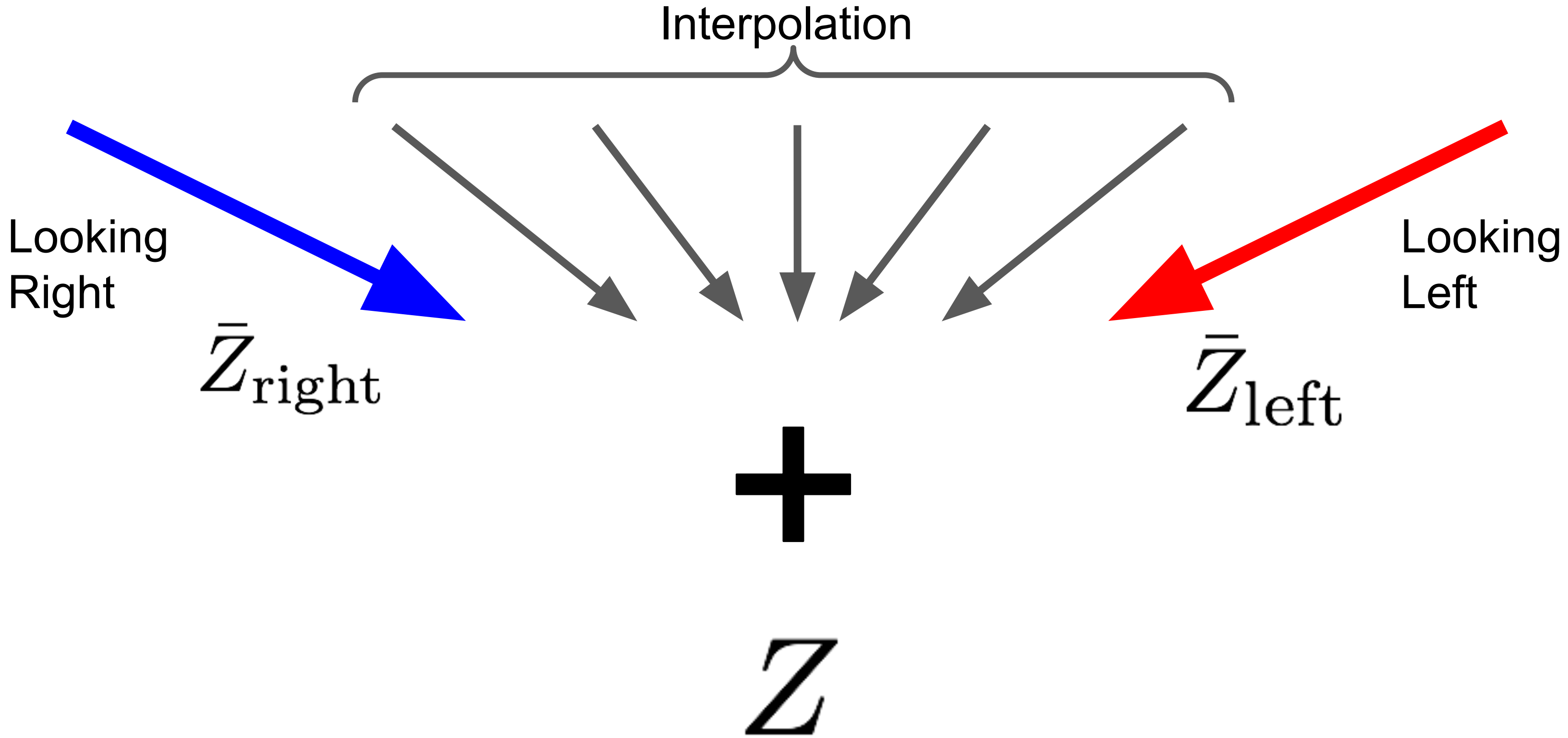




$$G\left(\bar{Z}_1 = \frac{1}{3}\sum_{i=1}^{3} Z_i\right) \quad G\left(\bar{Z}_2 = \frac{1}{3}\sum_{j=1}^{3} Z_j\right) \quad G\left(\bar{Z}_3 = \frac{1}{3}\sum_{k=1}^{3} Z_k\right) \qquad G(\bar{Z}_1 - \bar{Z}_2 + \bar{Z}_3)$$

Face Pose Transformation

Face Pose Transformation



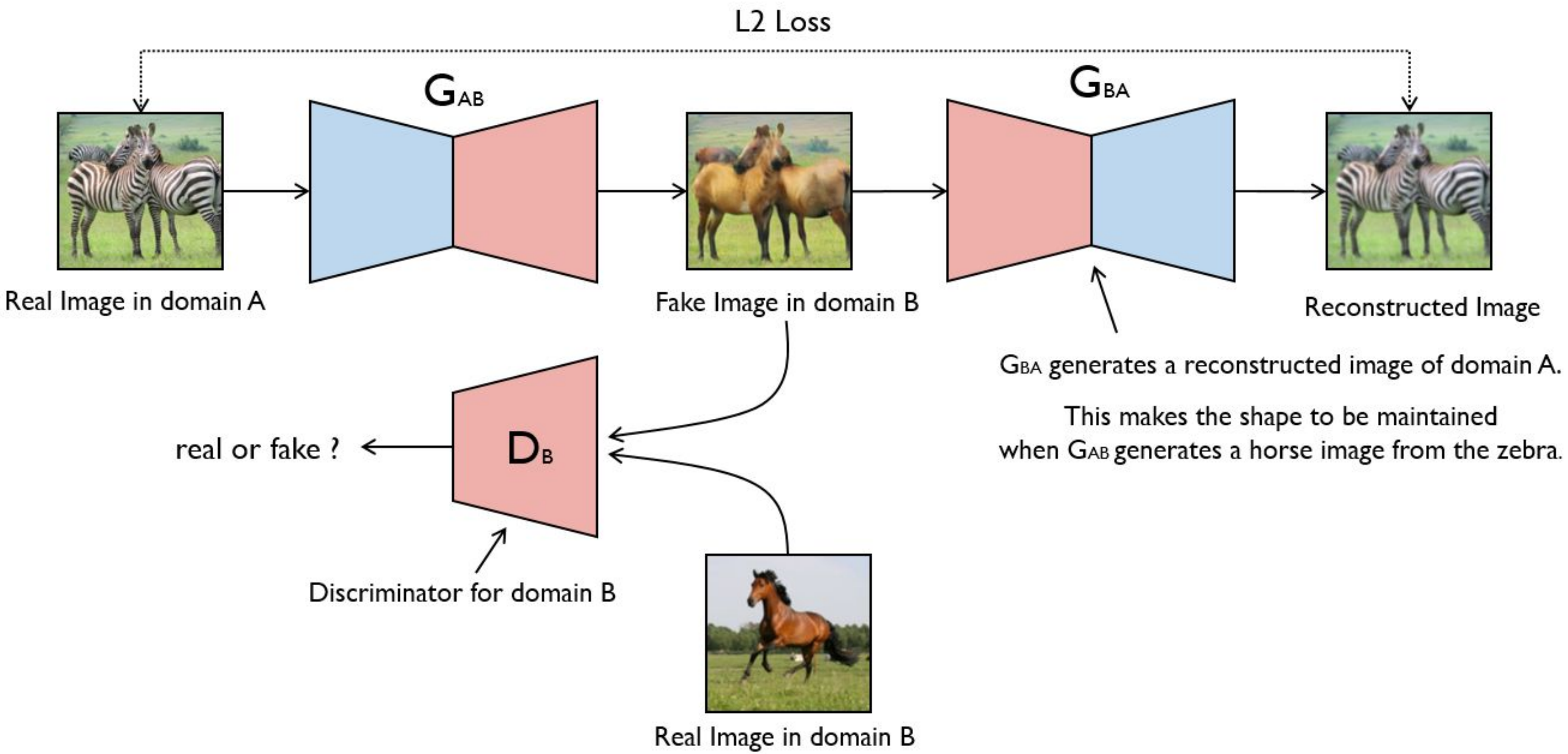
Reusable Representations

- Trained GAN on Imagenet-1k
- Extracted convolutional layers from discriminator
- Used them as input to new linear classifier model
- 82.8% accuracy on CIFAR-10

CycleGANs

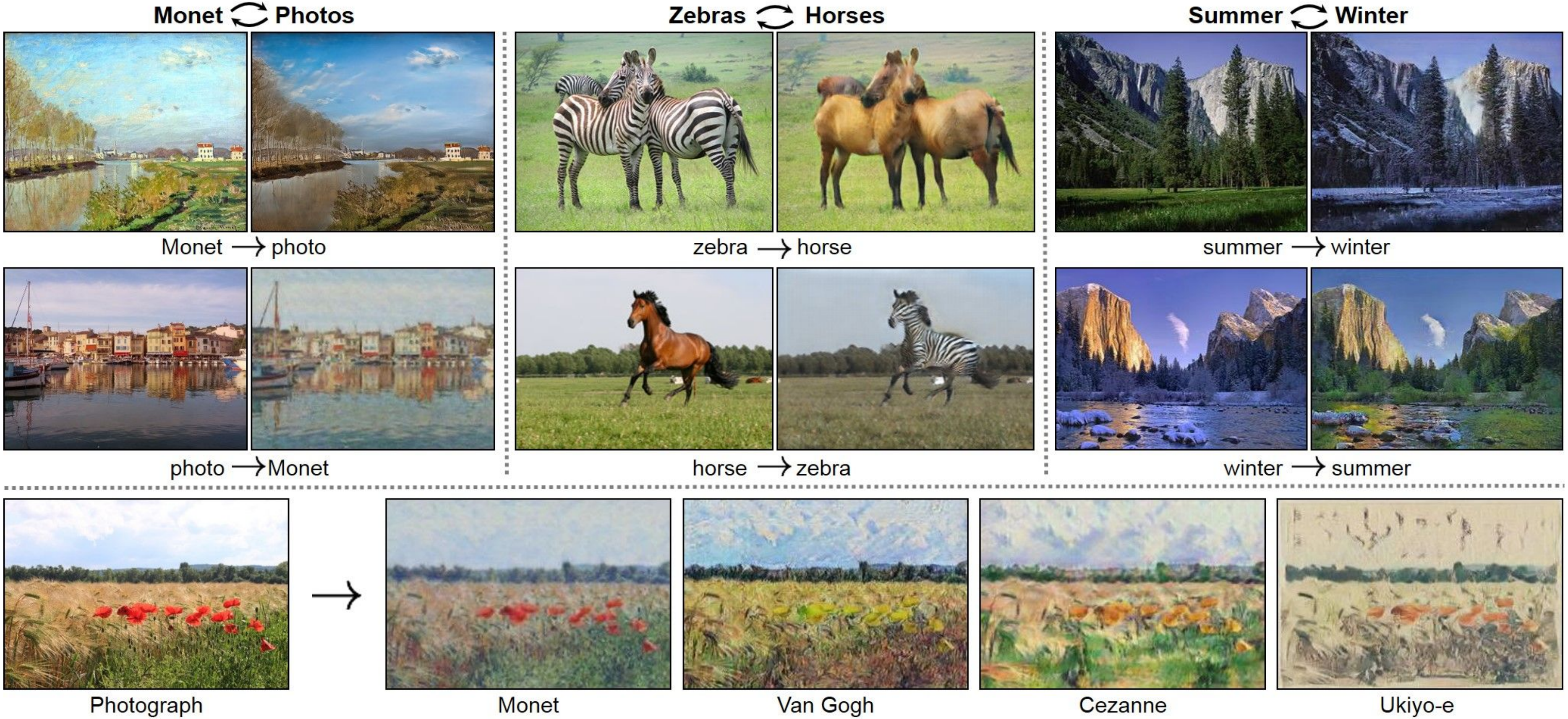
- Two generators
- Two discriminators
- Cyclic relation between the generators

CycleGAN



L2 Loss
G AB
G BA
Real Image in domain A
Fake Image in domain B
Reconstructed Image
G BA generates a reconstructed image of domain A.
This makes the shape to be maintained
when G AB generates a horse image from the zebra.
real or fake ?
D B
Discriminator for domain B
Real Image in domain B

CycleGANs


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

CycleGANs