Generative Models


2014
2015
2016
2017
2018
2019
2020
2021

StableDiffusion

What I cannot create,
I do not understand

~~Autoencoder~~ *Not a generative model*

Variational Autoencoder (VAE)

Generative Adversarial Networks (GANs)

Flow Model

Diffusion Model

~~Autoencoder~~ *Not a generative model*

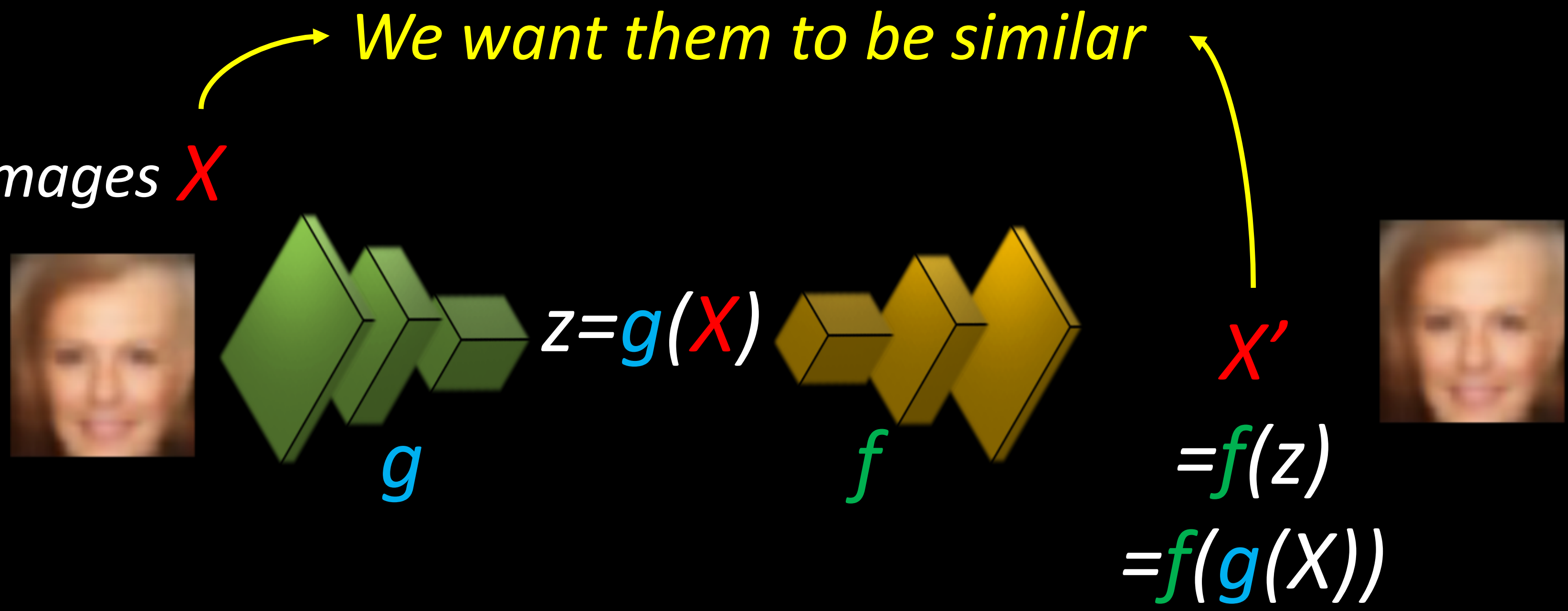
Variational Autoencoder (VAE) *Elegant*

Generative Adversarial Networks (GANs) *Easy to control, hard to train*

Flow Model *Elegant but hard to control*

Diffusion Model *Create highest quality but computationally costly*

Autoencoder



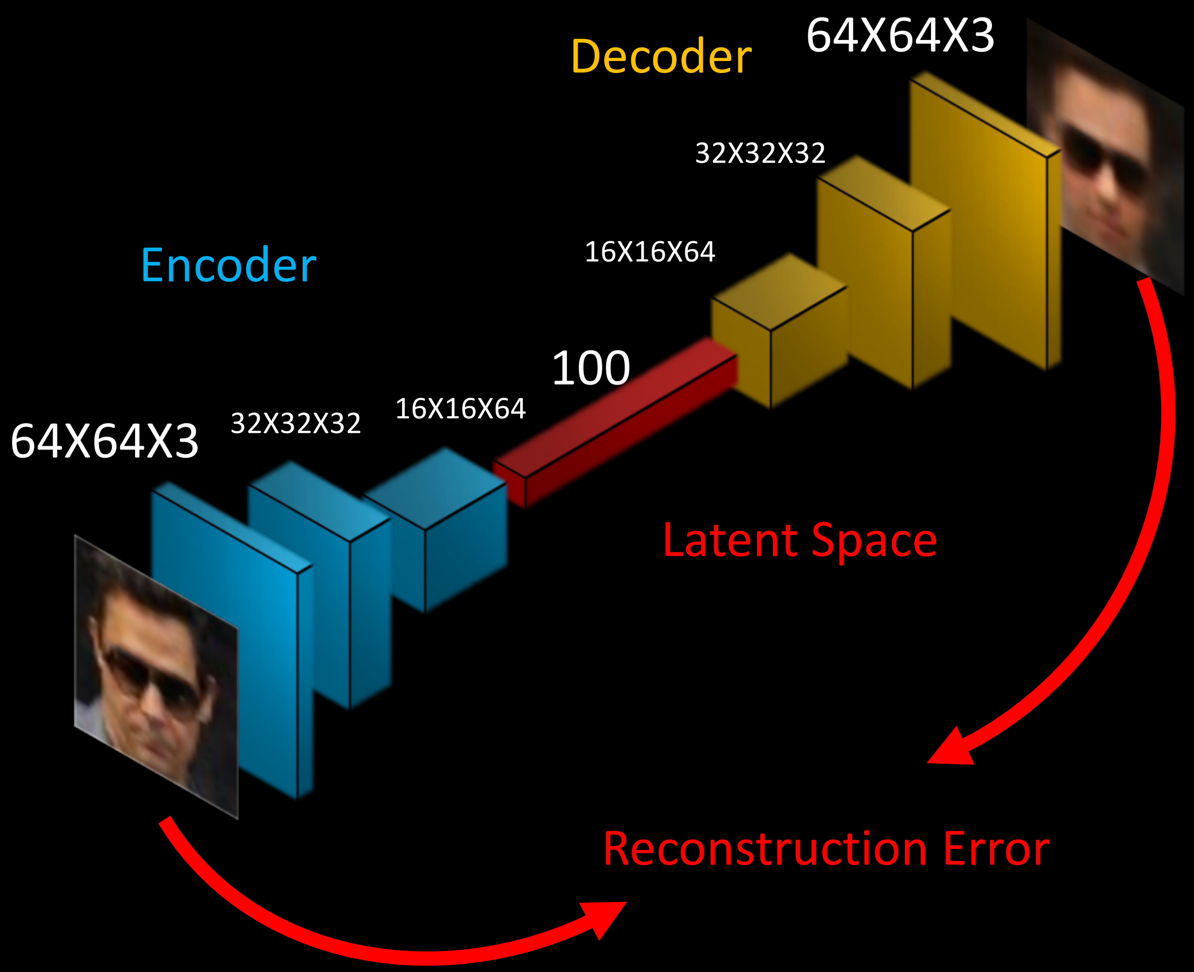
$$\text{Reconstruction Loss} = L_1(X,X') = |X-X'|$$

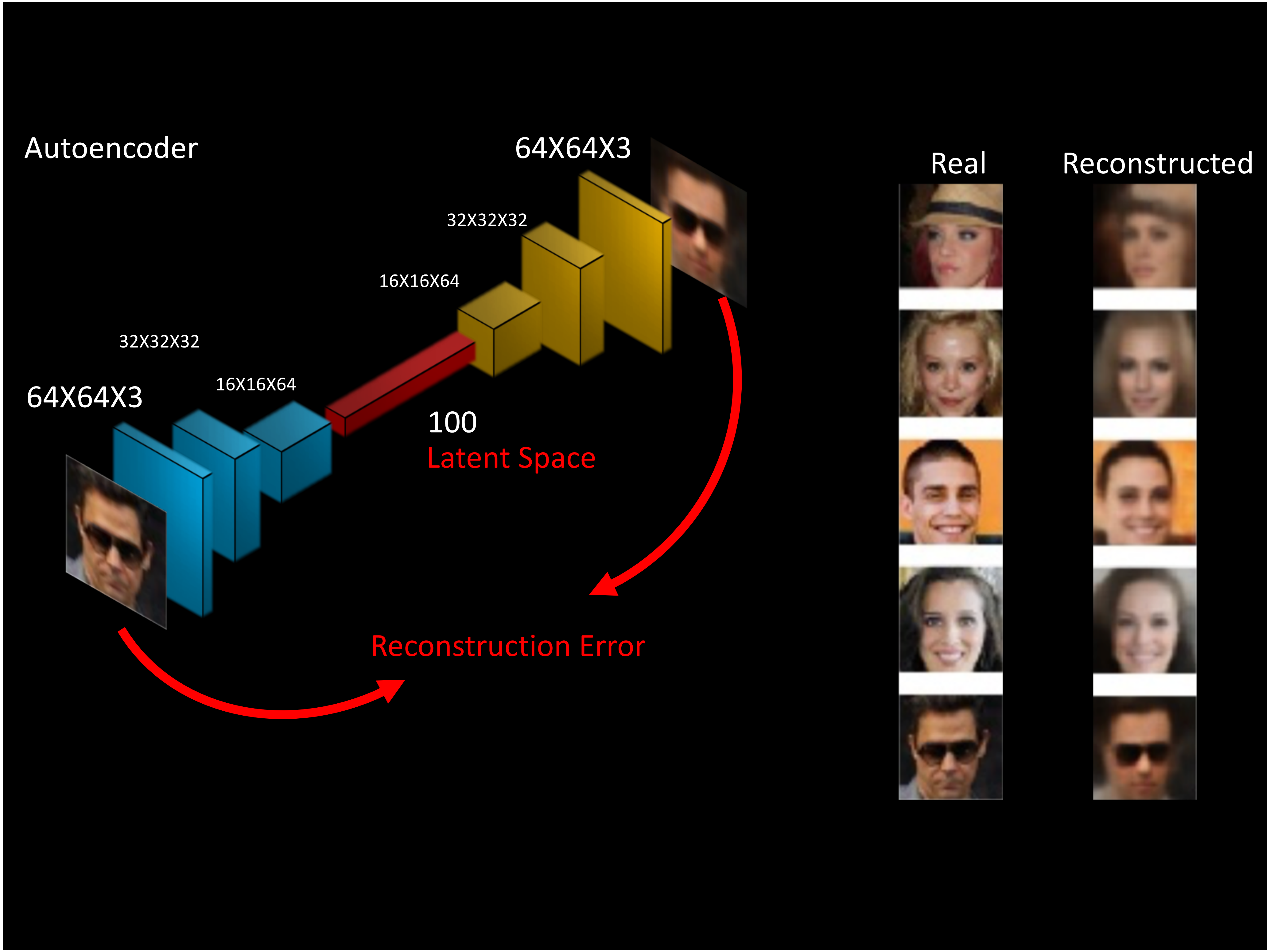
Autoencoder

```
def train(encoder, decoder, data_loader, epochs=5):

    for epoch in range(epochs):
        for x, _ in data_loader:
            # Encode and decode
            latent = encoder(x)
            output = decoder(latent)

            # Manual MSE loss
            loss = torch.mean((output - x) ** 2)
```

Autoencoder

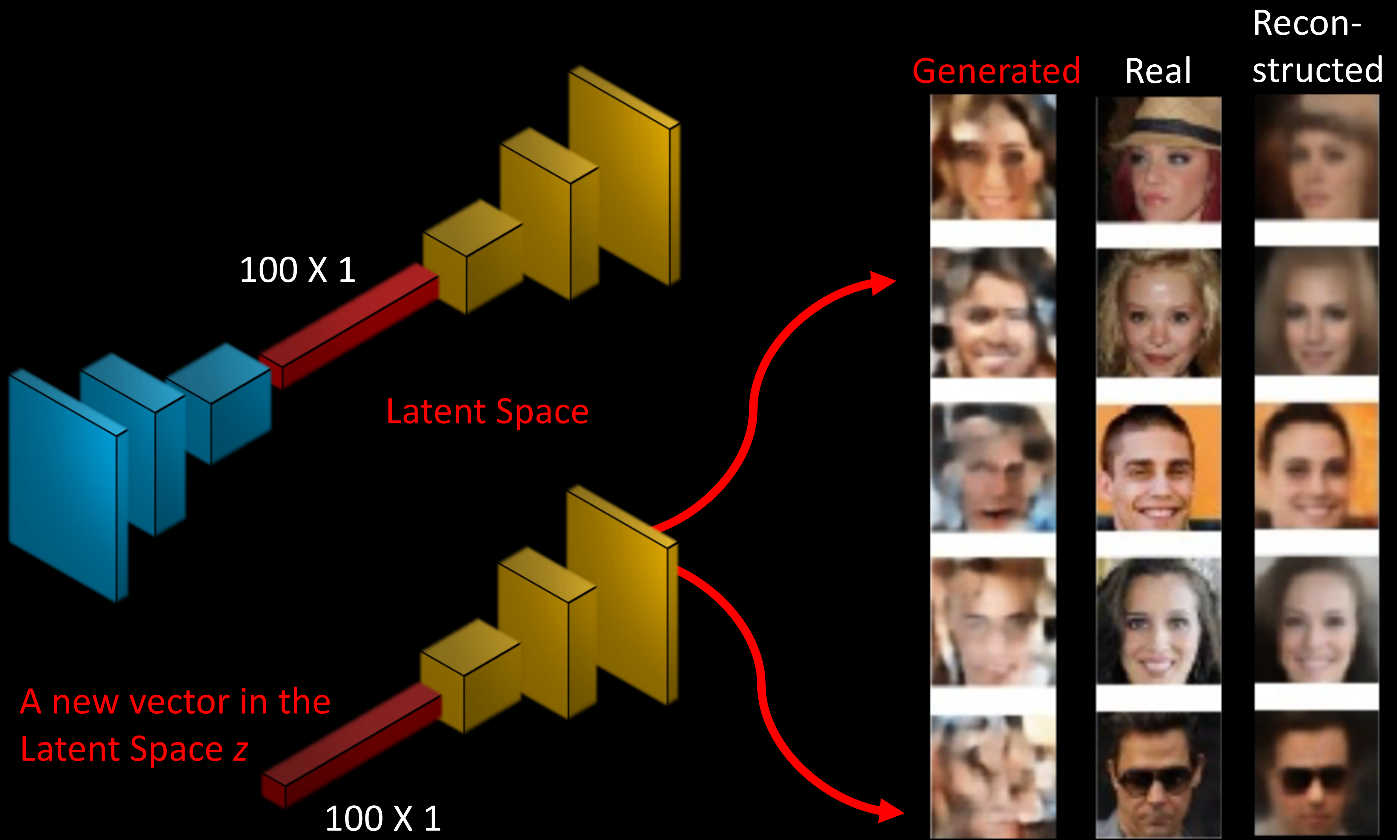
Ways to get to good representation:
(Un-) Self-supervised





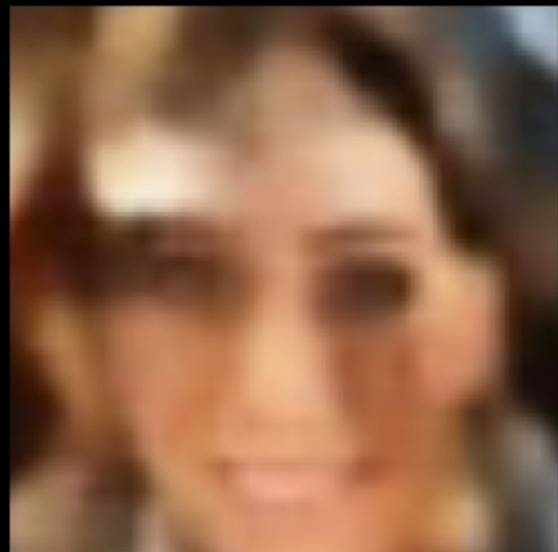
Autoencoder
64X64X3
32X32X32
16X16X64
100
Latent Space
16X16X64
32X32X32
64X64X3
Reconstruction Error
Real
Reconstructed

Autoencoder (not a generative model)

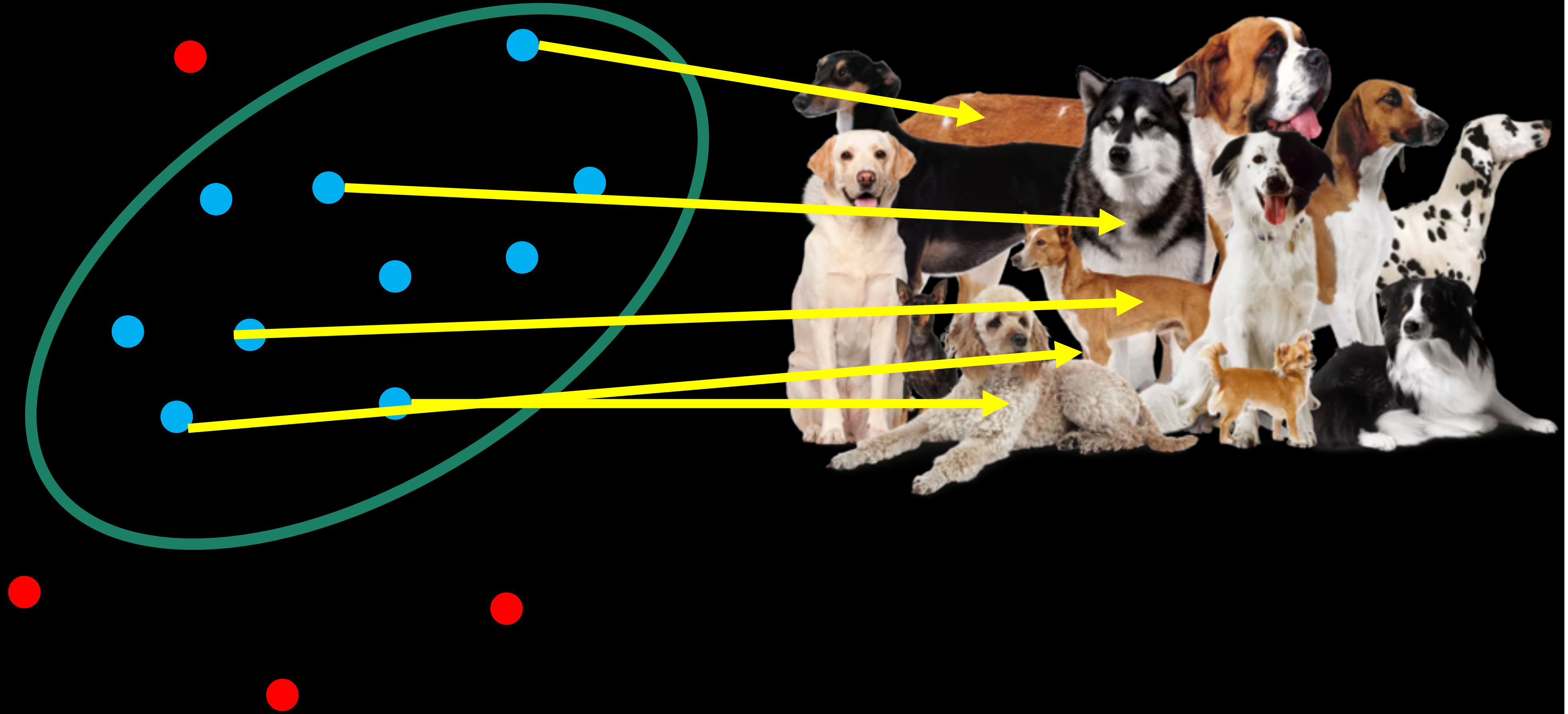


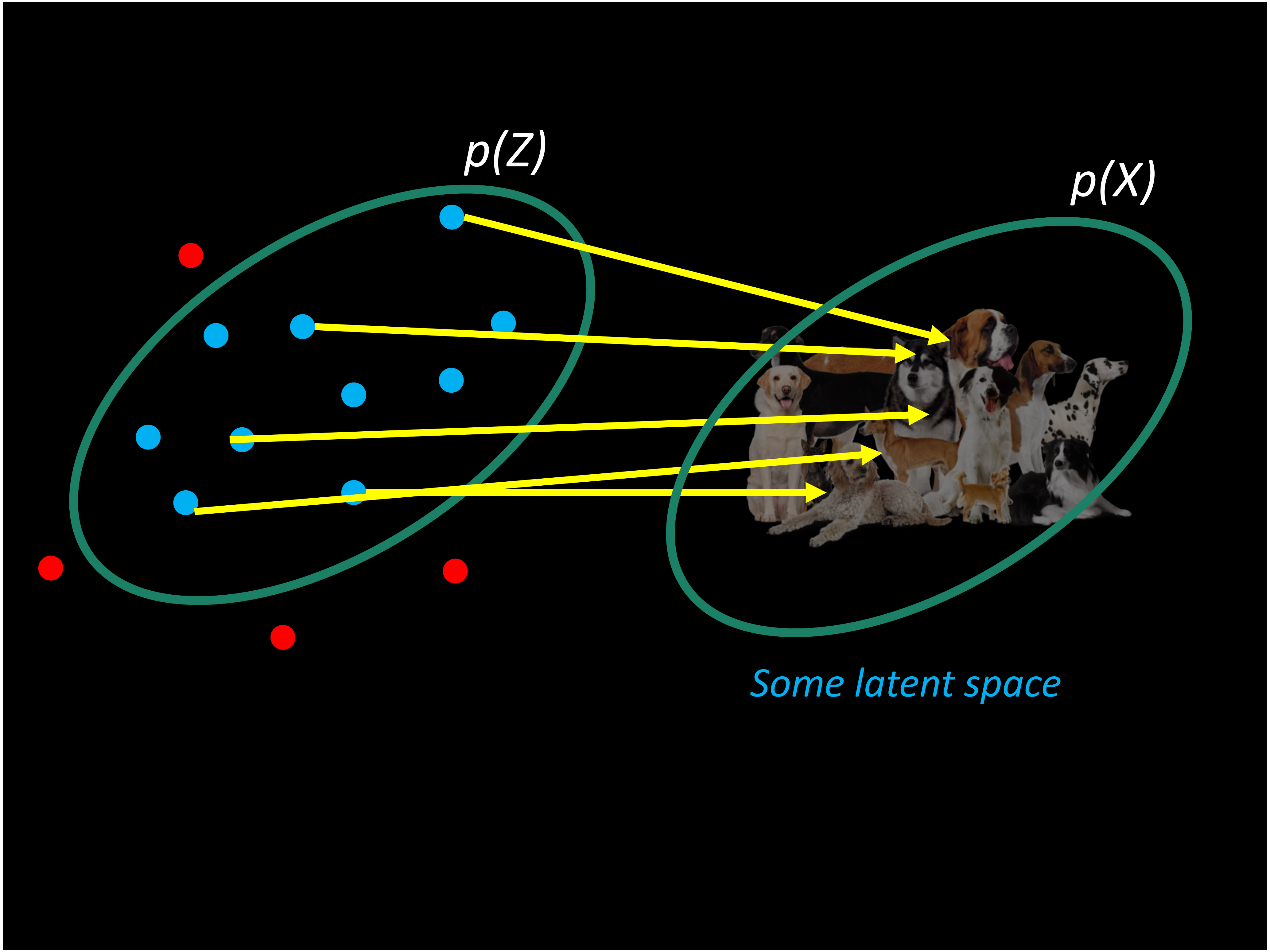
Q1: is it a person?

Q2: can you draw one?

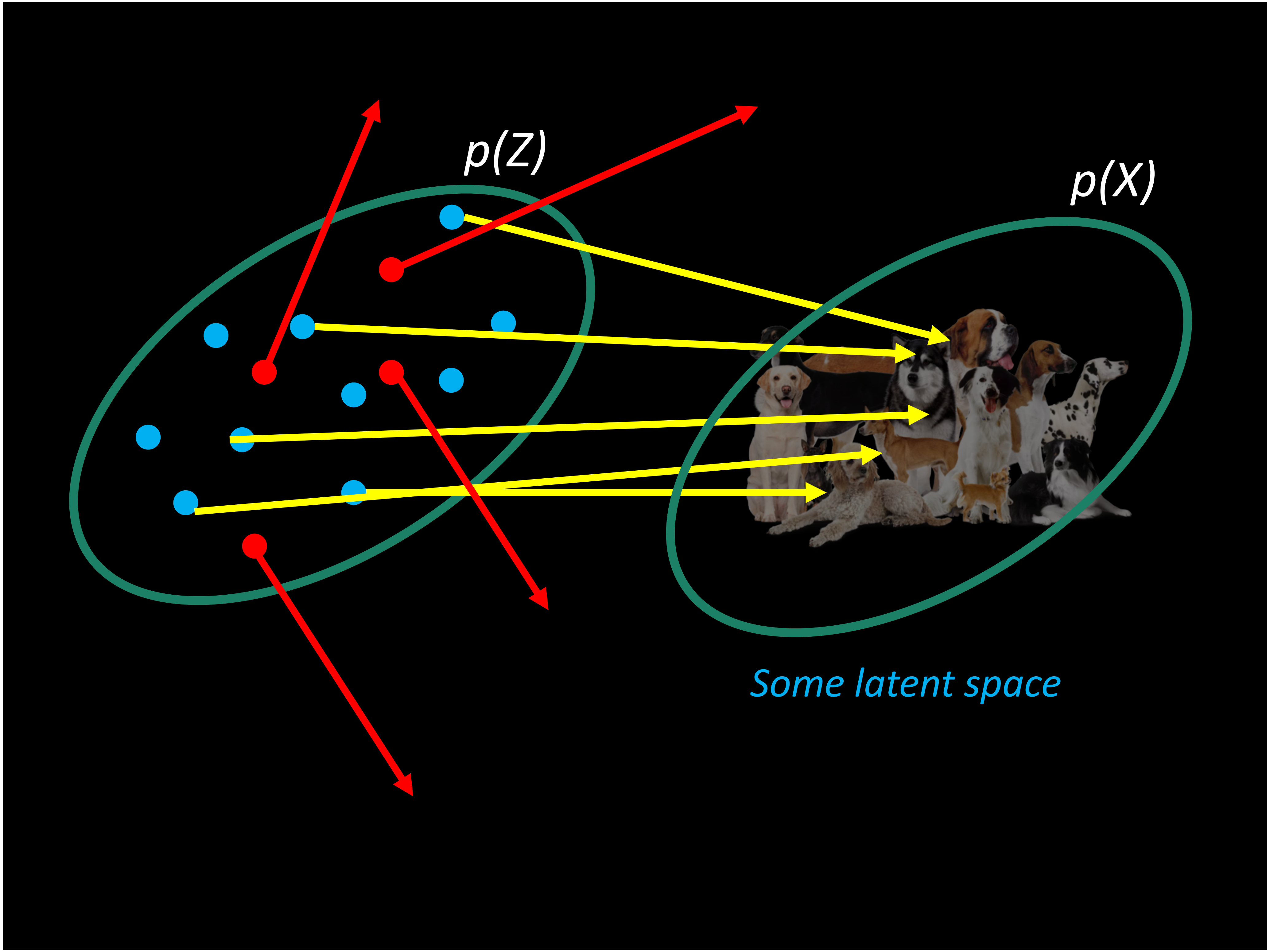
A: 

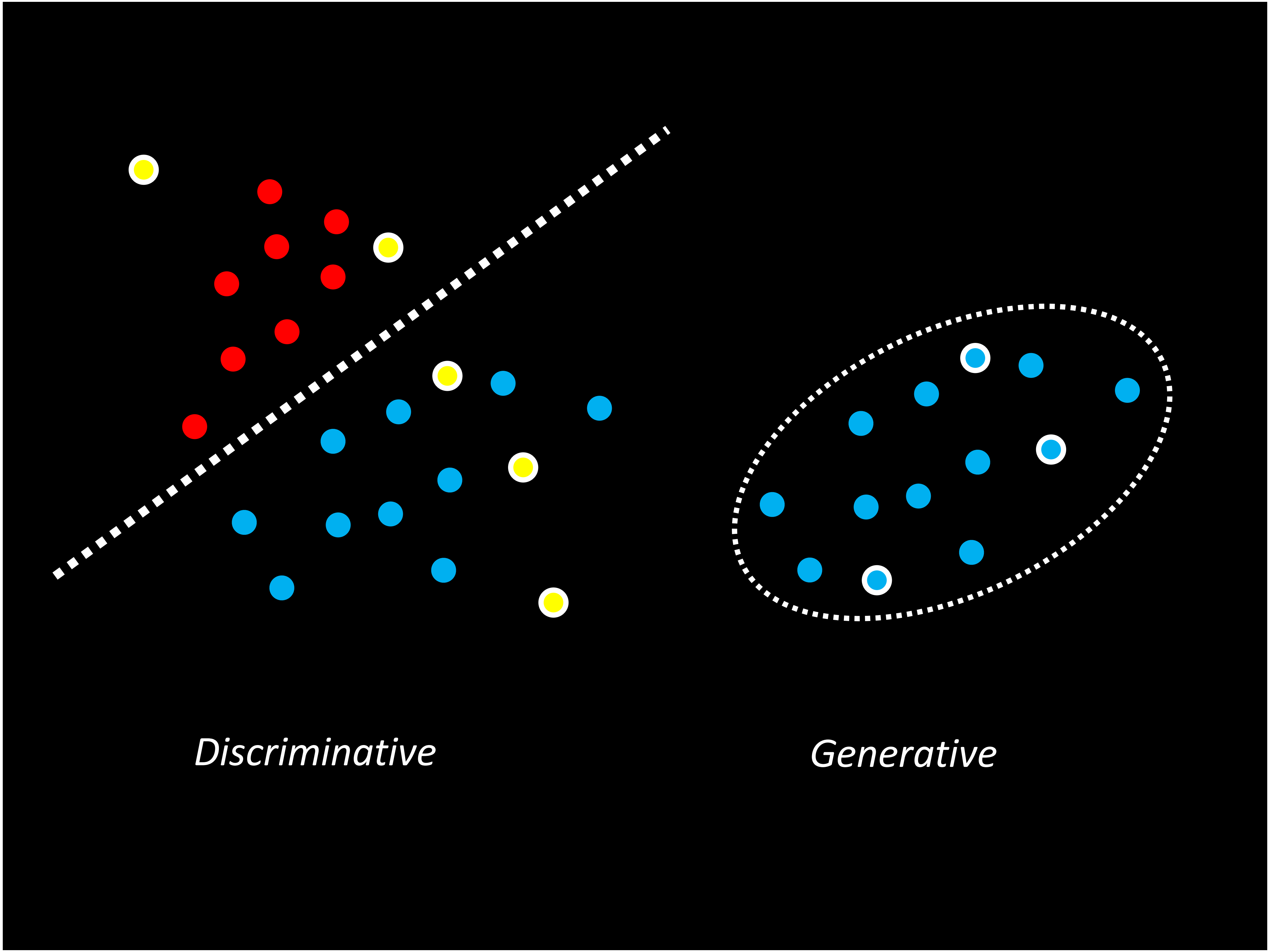
Why continuous distribution (manifold)?

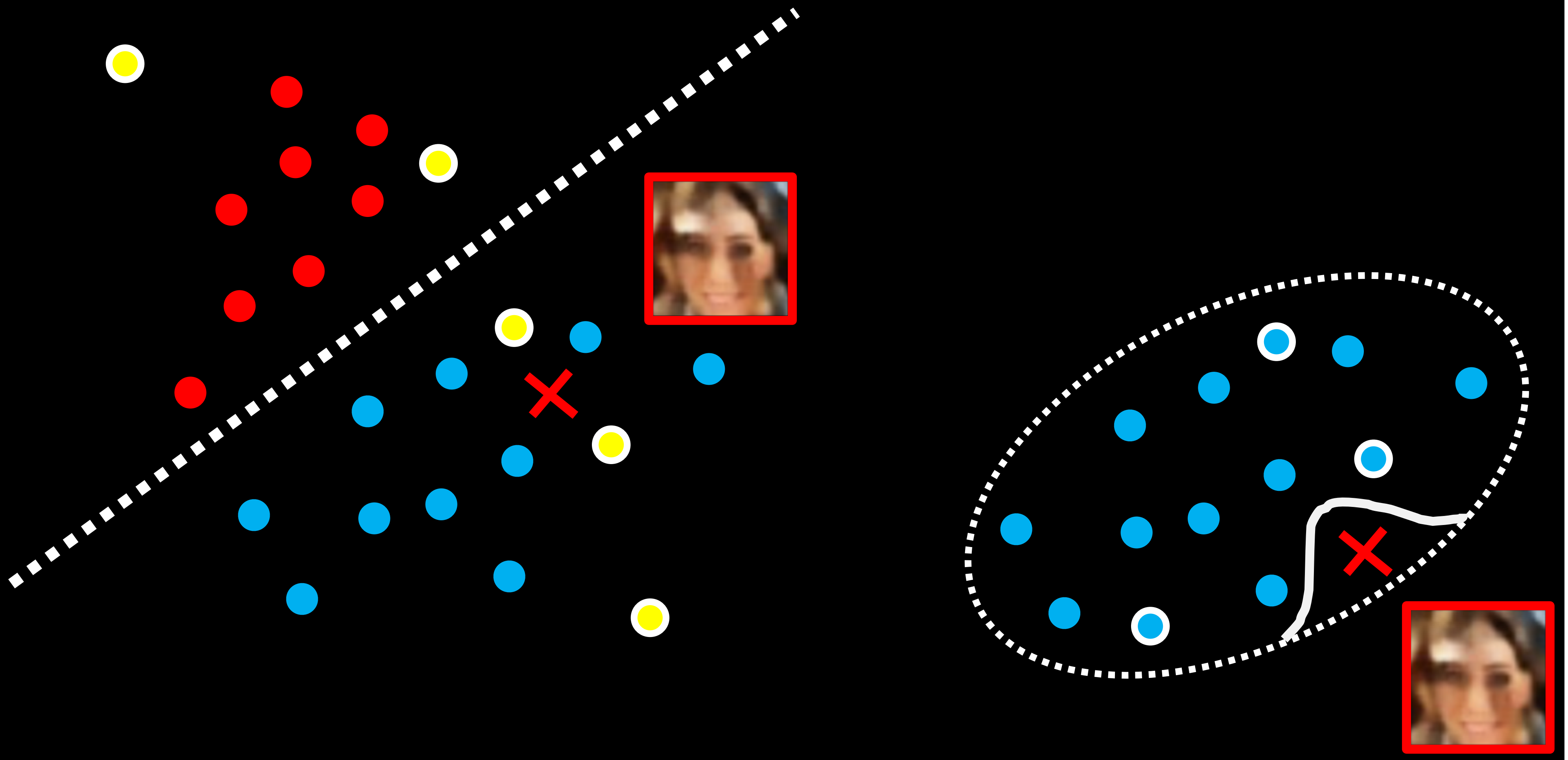




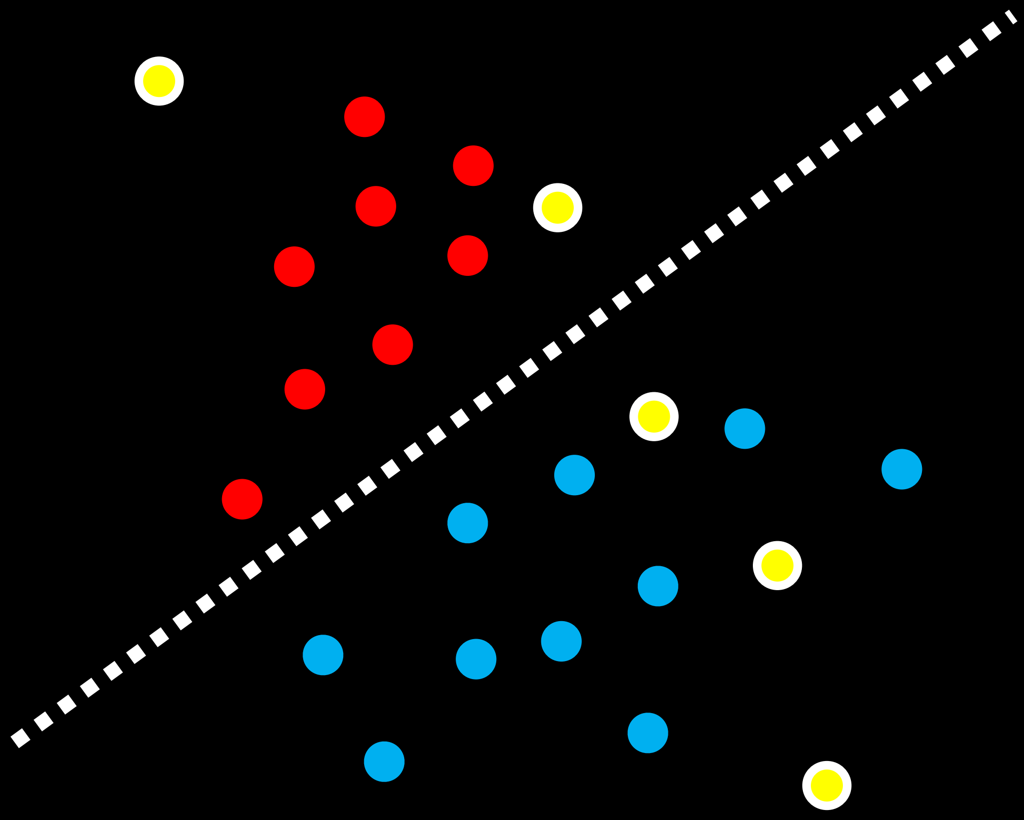
p(Z)
p(X)
Some latent space

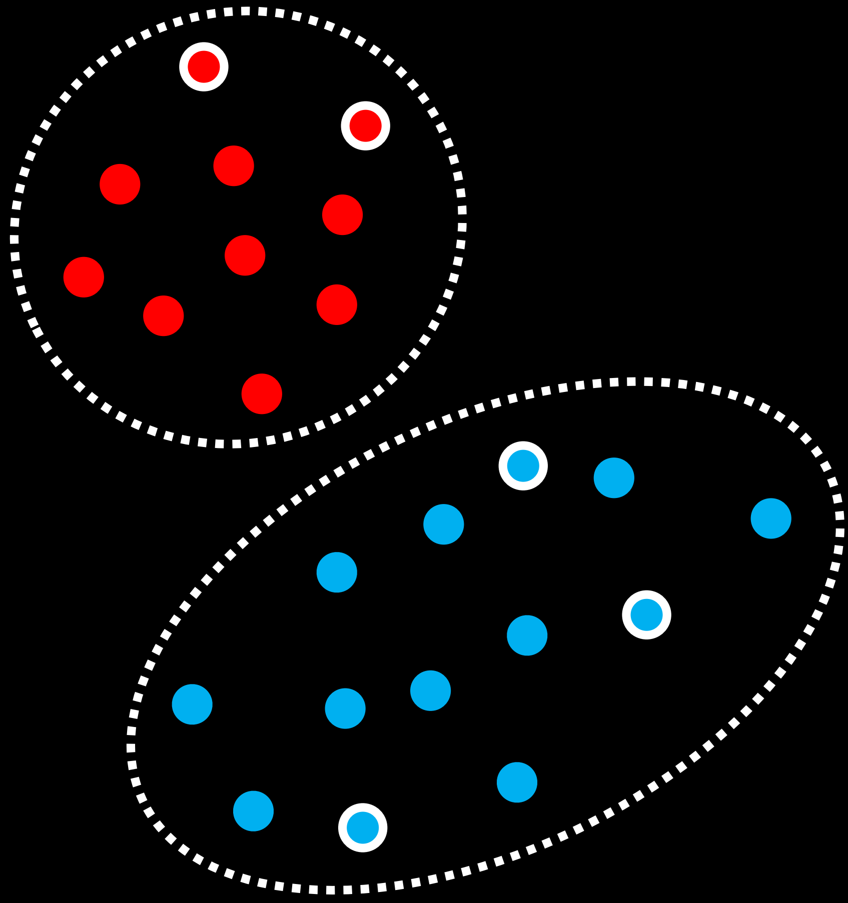

p(Z)
p(X)
Some latent space


Discriminative
Generative



Discriminative *Generative*





Discriminative classifier

$p(y=c|x)$

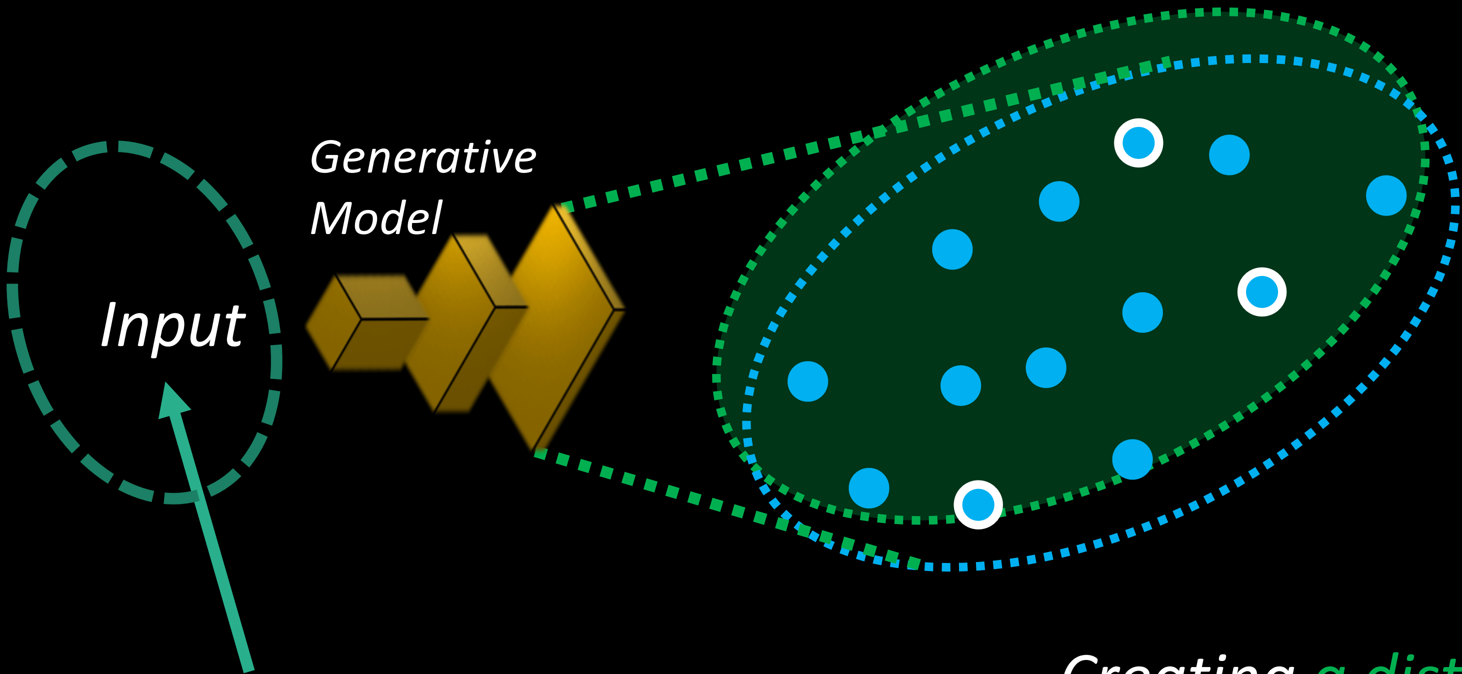
Generative classifier

$p(y=c|x) = p(y=c)\ p(x|y=c)$

Conditional Density

Goal of a Generative Model

$$p(X) \sim p'(X)$$

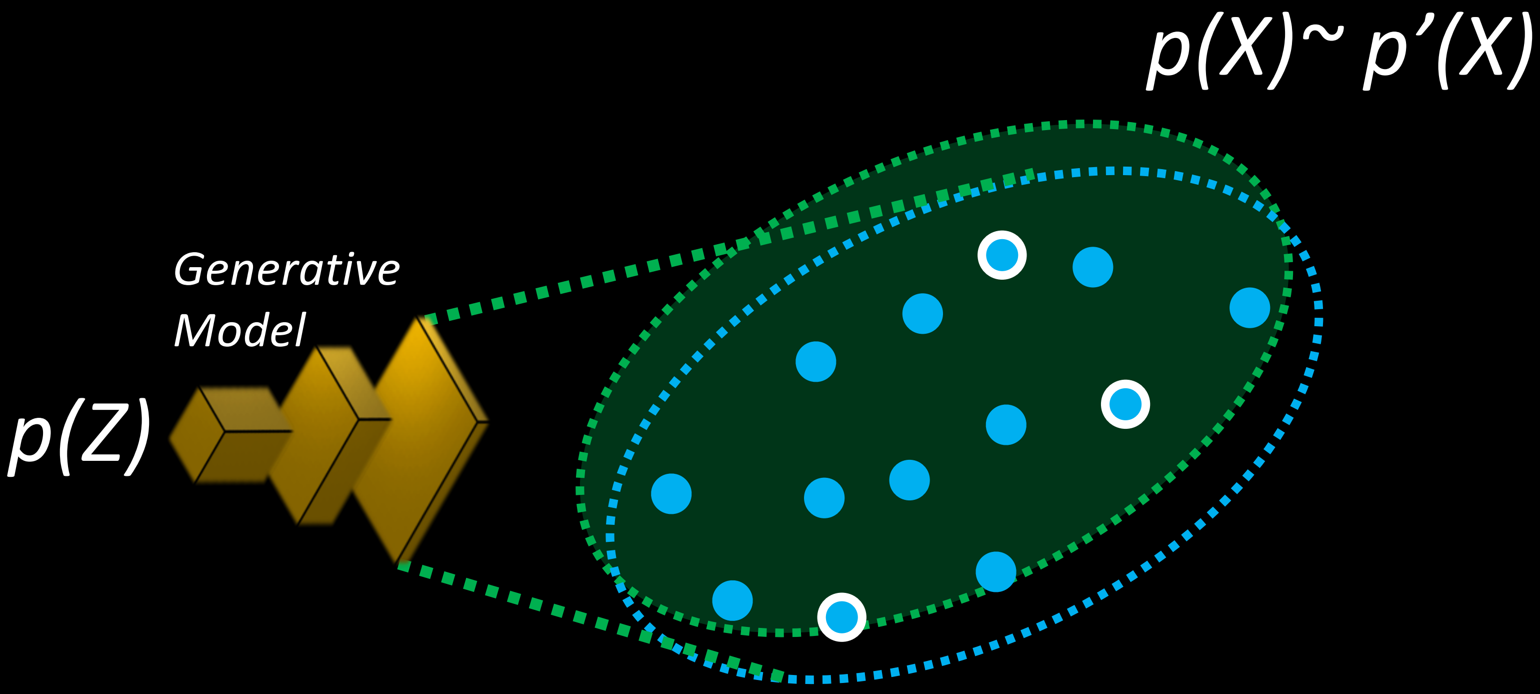


Often use normal distribution as the starting point

$$p(Z) = N(0,1)$$

Creating a distribution of data that mimic the real distribution

Goal of a Generative Model



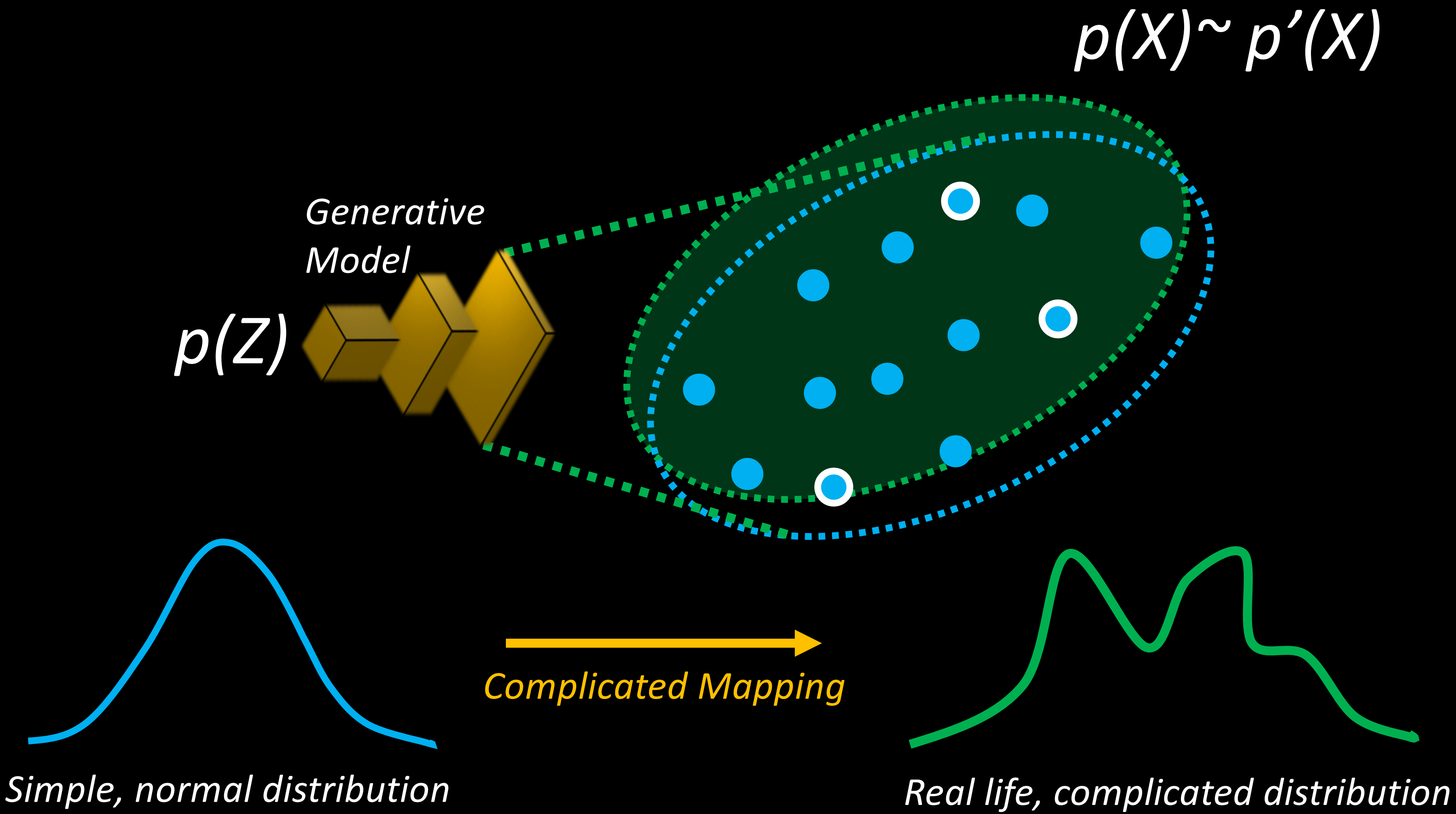
Model was optimized to get this

$$p(X) = \Sigma_z \, p(X|Z) \, p(Z)$$

We can only approximate this by observation

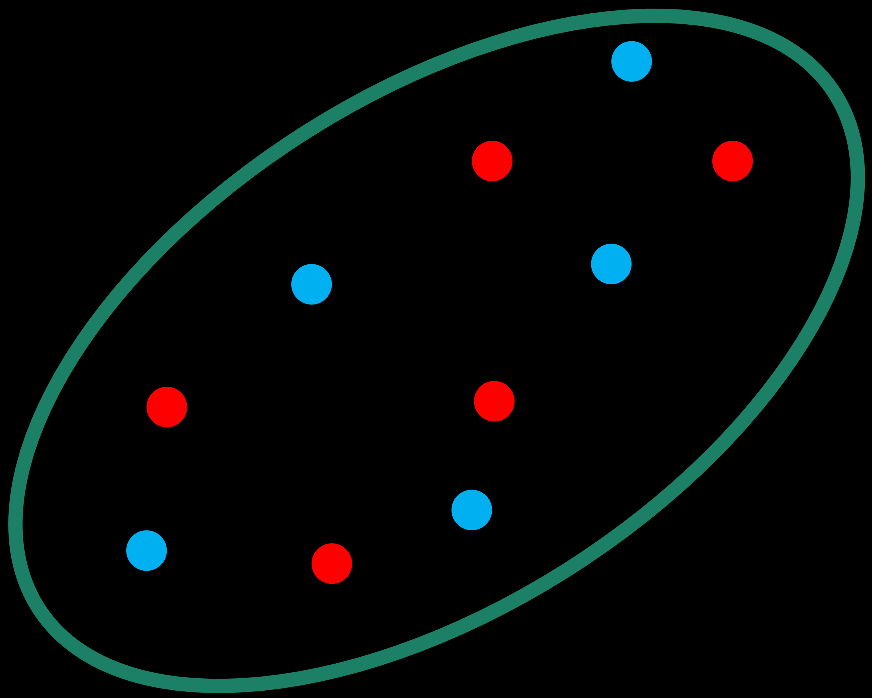
We can assign this

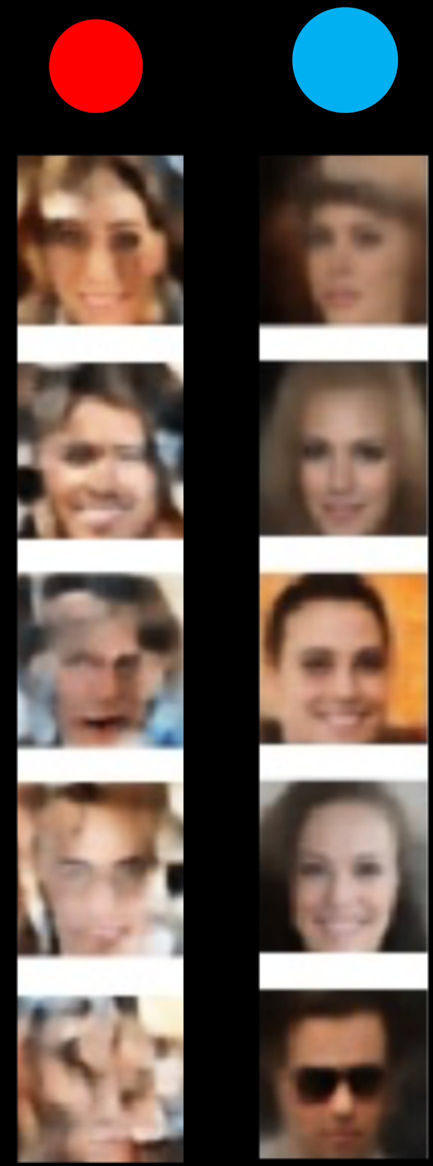
Goal of a Generative Model



$$p(X) = \Sigma_z \, p(X|Z) \, p(Z)$$

Autoencoder is not a generative model

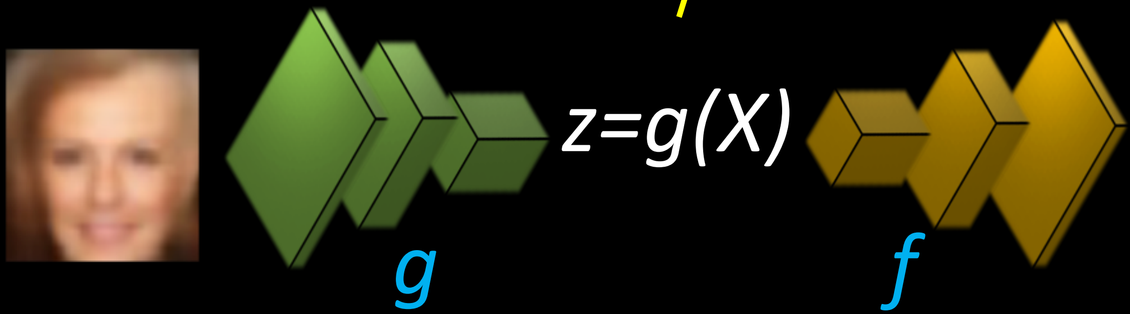




Autoencoder

No regulation for z

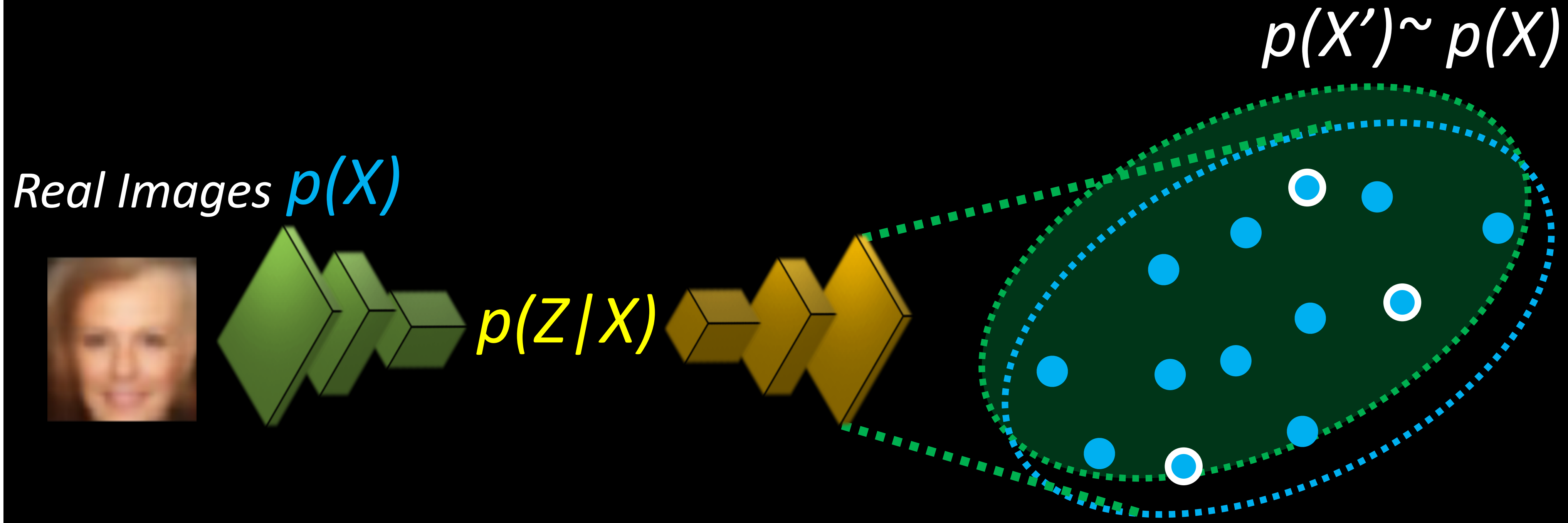
Real Images X



$X'=f(z)$
$=f(g(X))=X$

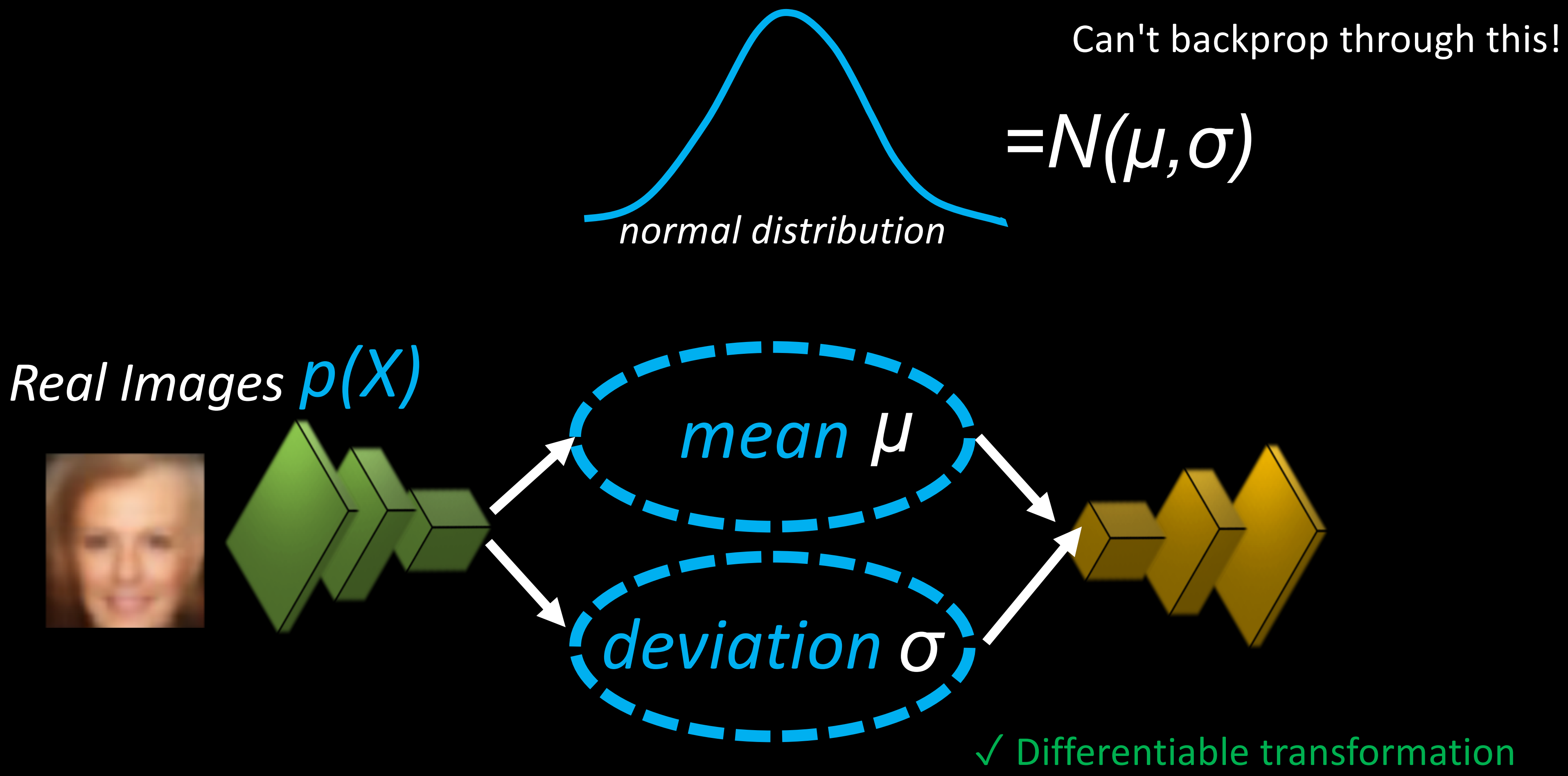
$$Loss = L_2(X,X') = |X-X'|^2$$

Variational Autoencoder (VAE)

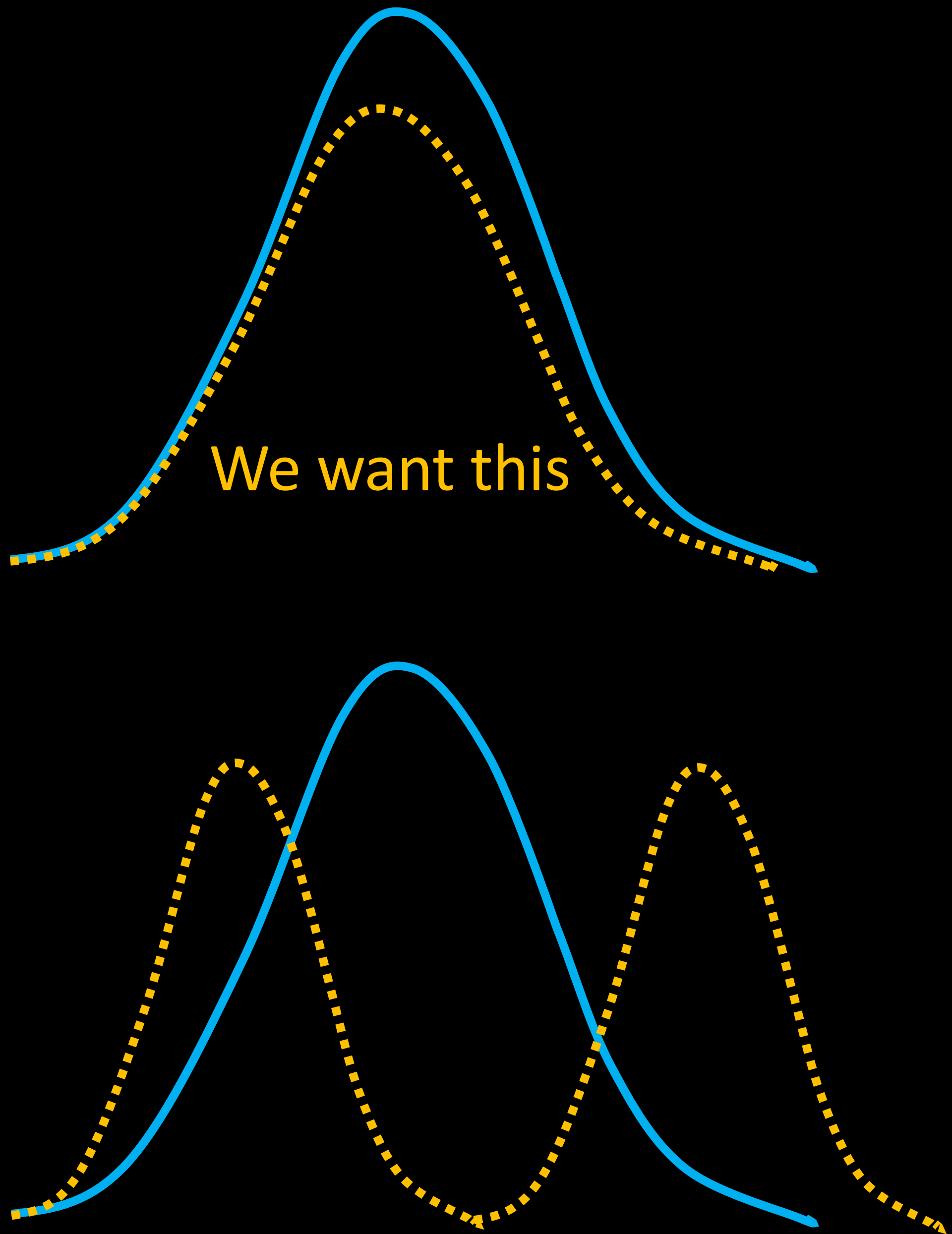


We want *p(Z|X)* to be a normal distribution where we can sampled from

Variational Autoencoder (VAE)



Optimize by KL divergence



Similar to a normal distribution
(small KL divergence)

Not similar to a normal distribution
(large KL divergence)

$$q_\phi(z|x) = \mathcal{N}(\mu, \sigma^2)$$

$$p(z) = \mathcal{N}(0, 1)$$

$$D_{KL}(q_\phi(z|x)||p(z)) = \int q_\phi(z|x) \log \frac{q_\phi(z|x)}{p(z)} dz$$

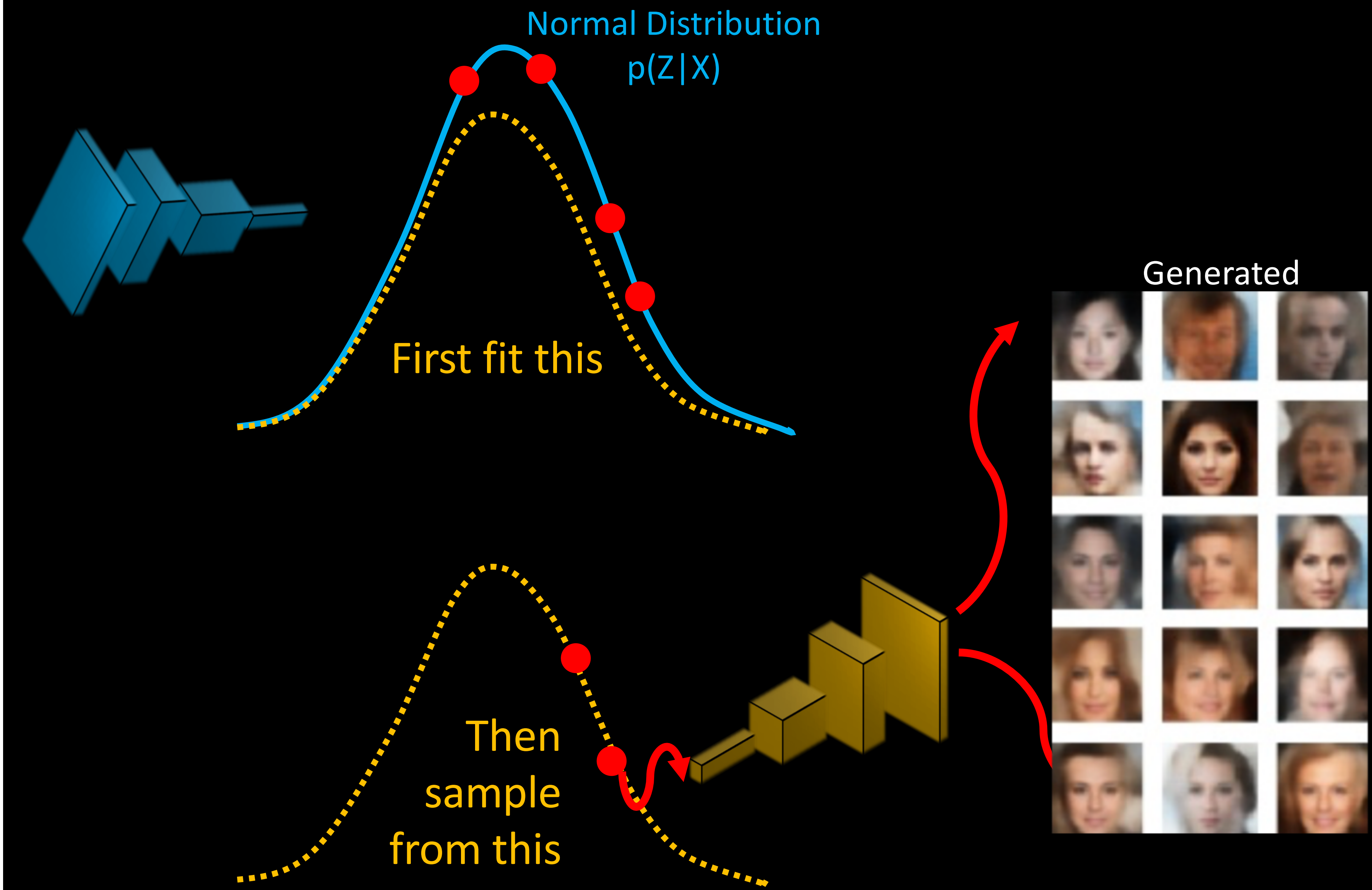
$$D_{KL} = \mathbb{E}_{z\sim q_\phi(z|x)}\left[-\frac{1}{2}\log(\sigma^2) - \frac{(z - \overset{\text{mu}}{\mu})^2}{2\sigma^2} + \frac{z^2}{2}\right]$$

$$= -\frac{1}{2}(\log(\sigma^2) + 1 - \sigma^2 - \mu^2)$$

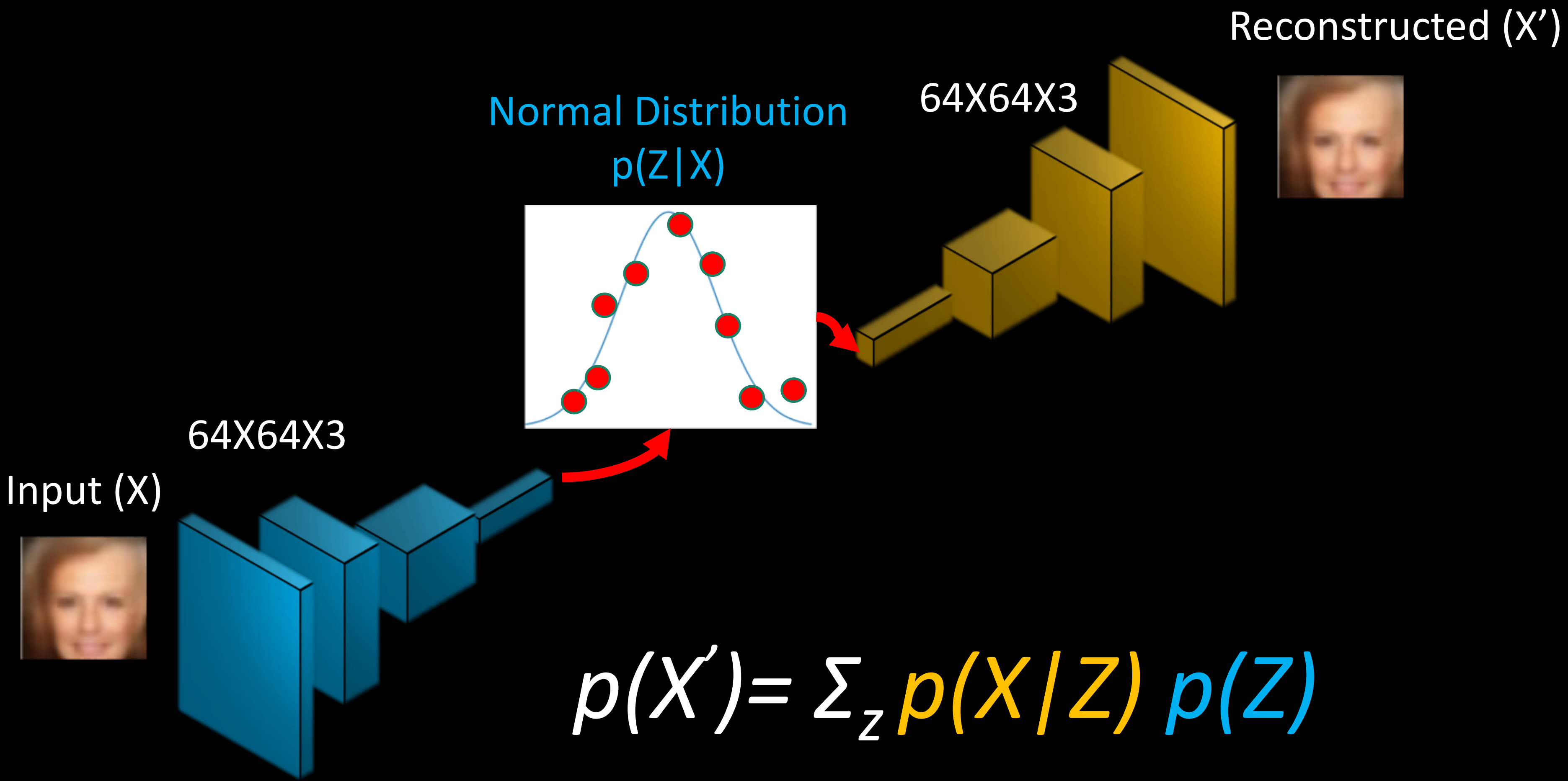
```
kl_loss = -0.5 * torch.mean
        (1 + logvar - mu.pow(2) - logvar.exp())
```

$-\frac{1}{2}\log(\sigma^2)$ μ^2 σ^2

Sample from the latent space



Variational autoencoder



$$p(X') = \Sigma_z \, p(X|Z) \; p(Z)$$

$$\sim \Sigma_z \, p(X|Z) \; p(Z|X)$$

VAE reparametrize

Input → Encoder → z (AE)

Input → Encoder→ (mμ, logvar) → Reparameterize→ z (VAE)

↑ params ↑ random ε

```
def reparameterize(mu, logvar):
    std = torch.exp(0.5 * logvar)
    eps = torch.randn_like(std)
    return mu + eps * std
```

$\sigma = e^{0.5 \log(\sigma 2)}$

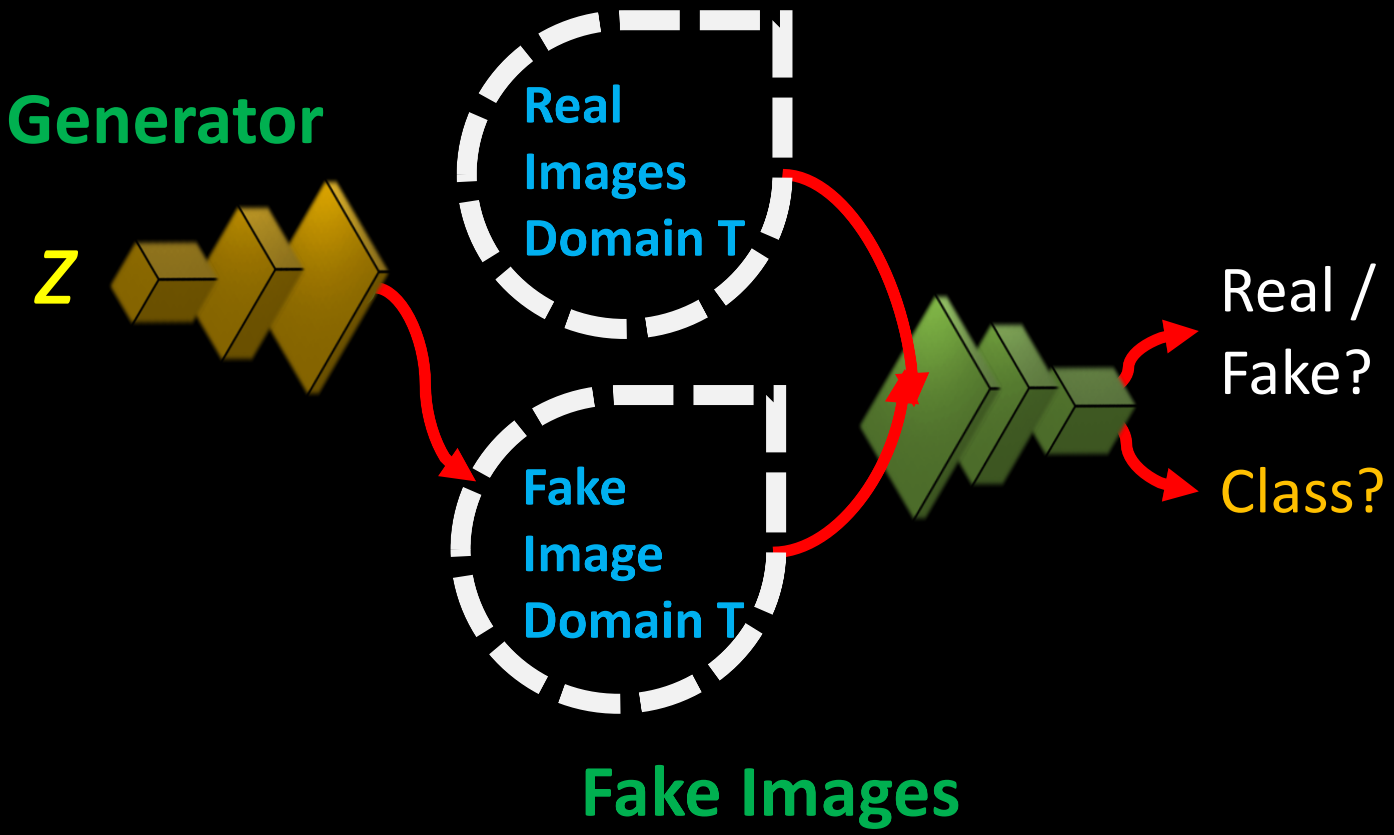
$z = \mu + \epsilon \cdot \sigma$

```
mu, logvar = encoder(x)
z = reparameterize(mu, logvar)
output = decoder(z)
```

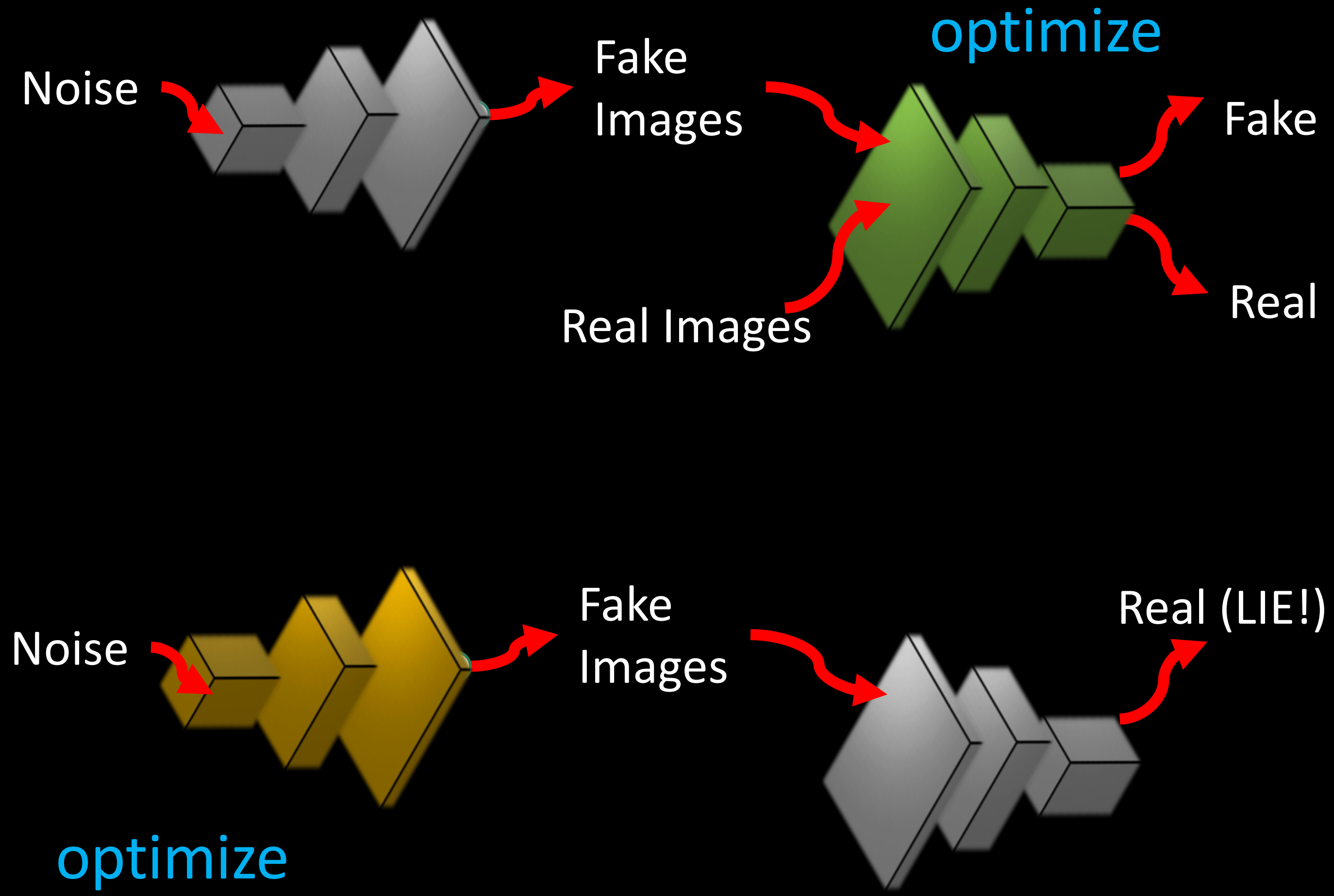
(Input → Encoder→ (mμ, logvar) → Reparameterize→ z)

```
recon_loss = torch.mean((output - x) ** 2)

# KL divergence loss
kl_loss = -0.5 * torch.mean
      (1 + logvar - mu.pow(2) - logvar.exp())

total_loss = recon_loss + kl_loss
```

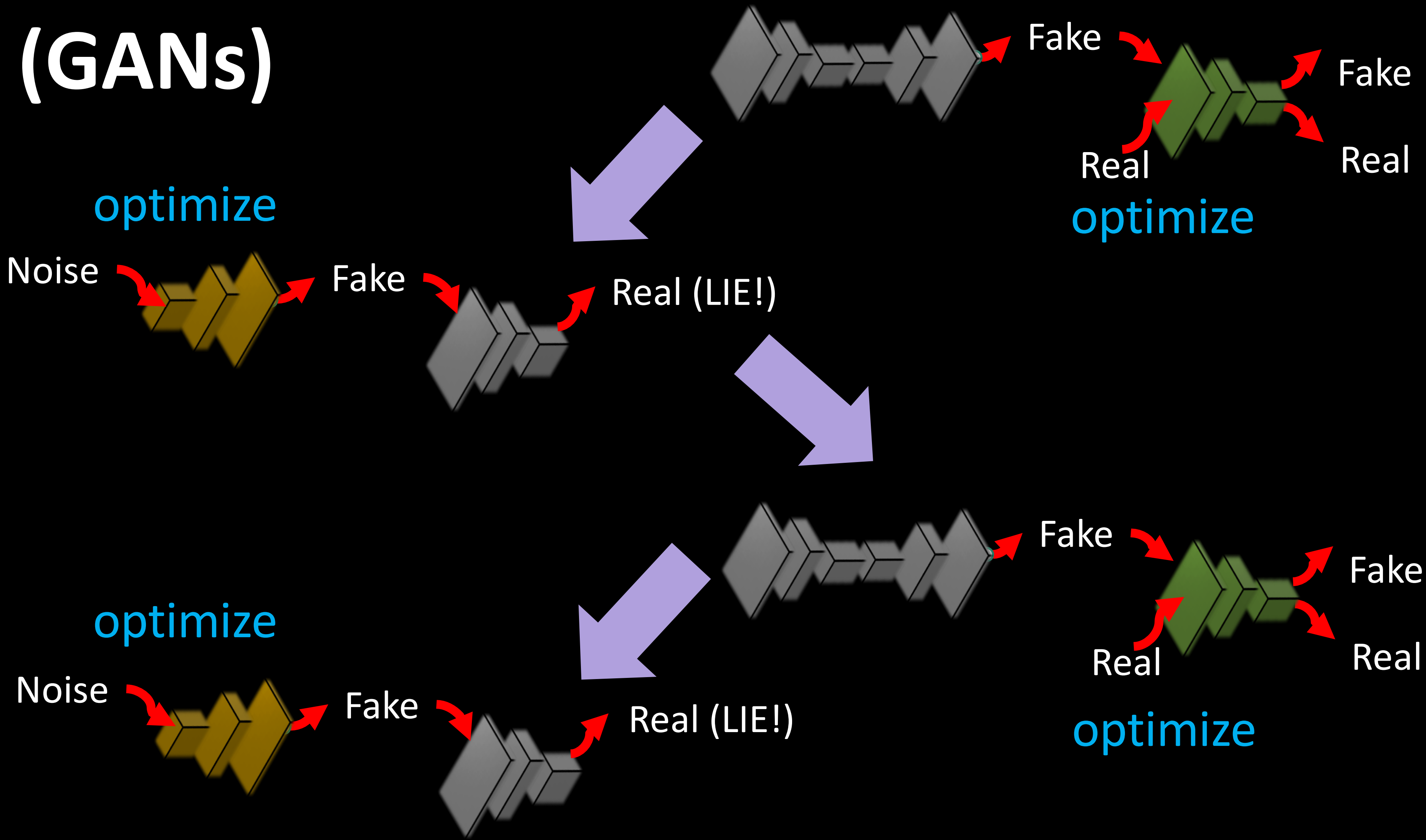
Generative Adversarial Networks (GANs)



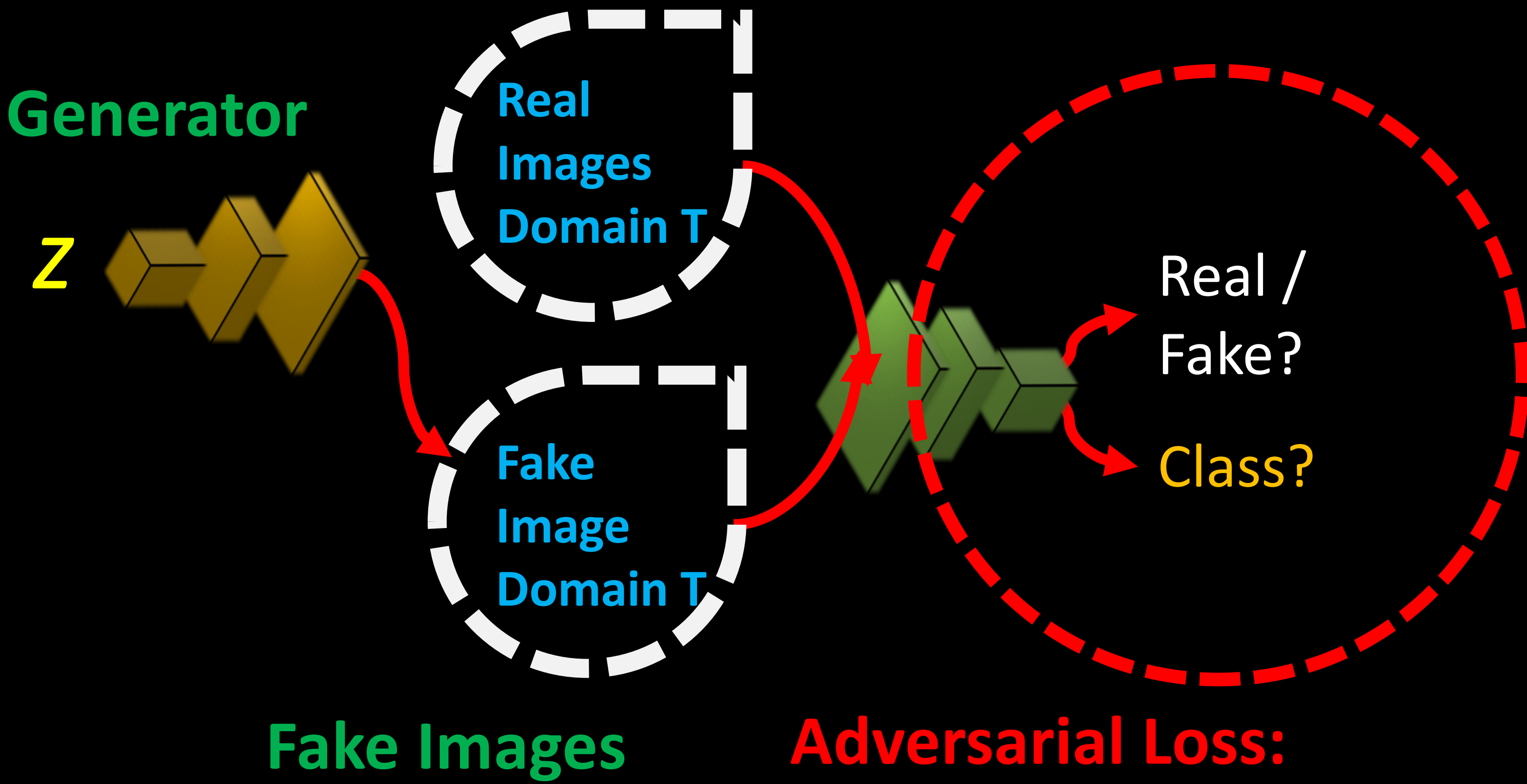
Generative Adversarial Networks (GANs)



Generative Adversarial Networks (GANs)



Generative Adversarial Networks (GANs)



Adversarial Loss:

classify (BCE) real/fake

lsgan

wgan

Vallina GAN Loss

Real　　　　　　　　　　Fake

$$Loss\_D = E[\log(D(x))] + E[\log(1-D(G(z)))]$$

$$Loss\_G = E[\log(D(G(z)))]$$

REAL

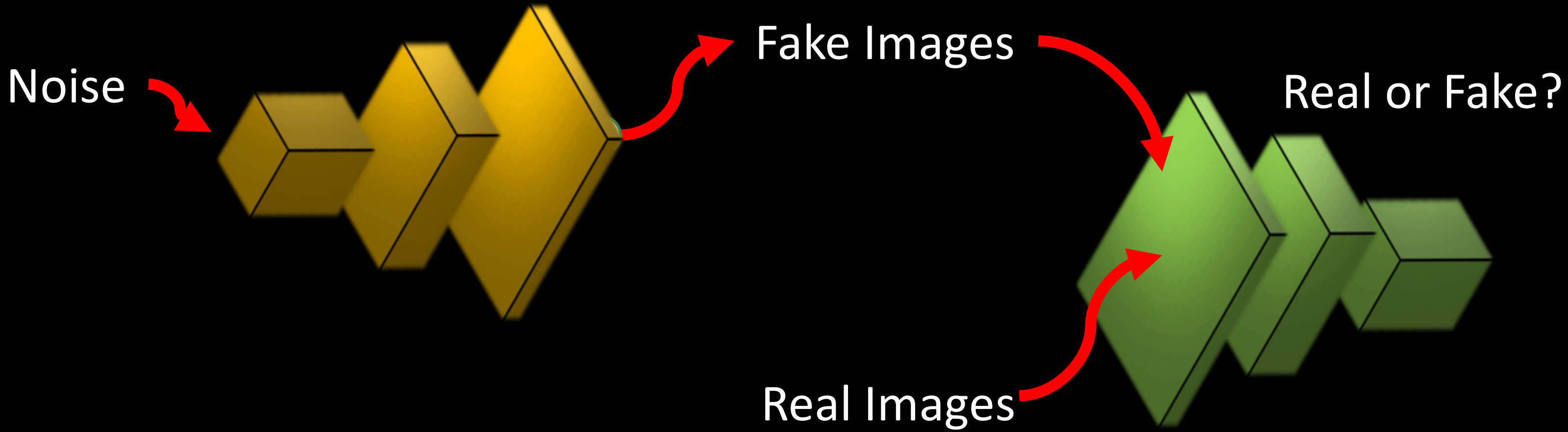
LSGAN Loss

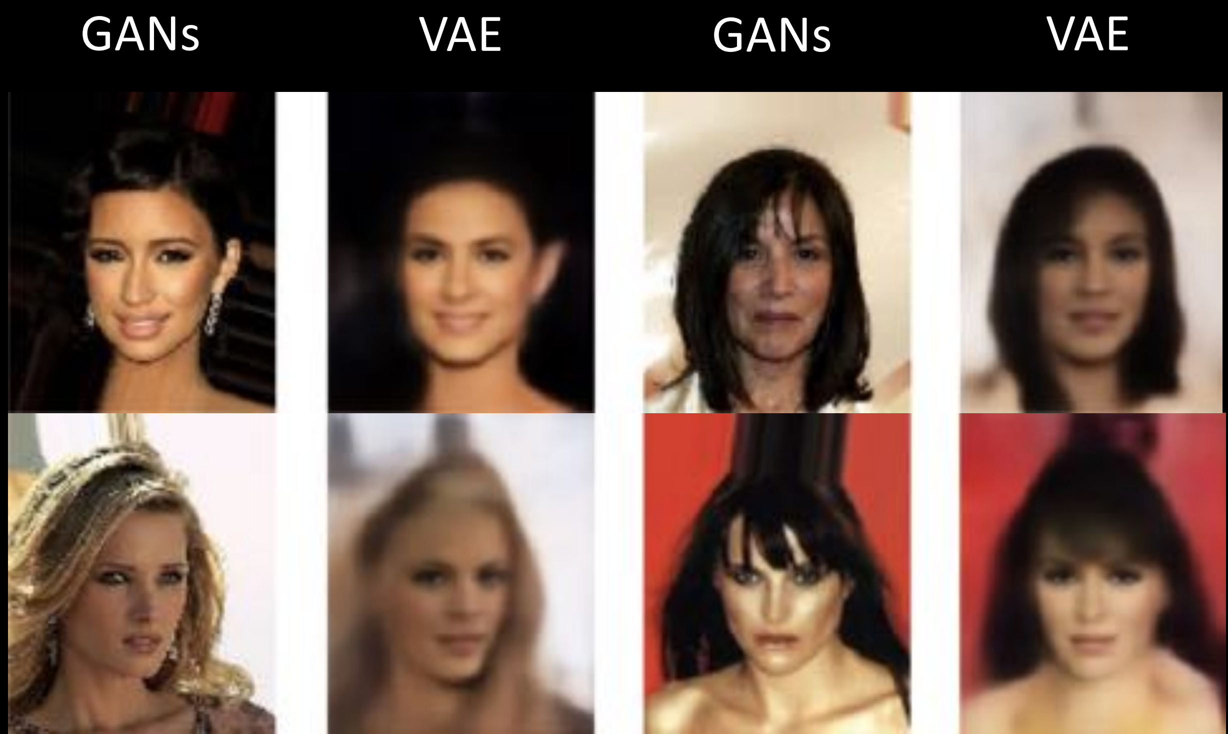
Real　　　　　　　　　　Fake

$$Loss\_D = E[(D(x) - 1)^2] + E[D(G(z))^2]$$

$$Loss\_G = E[D(G(z) - 1)^2]$$

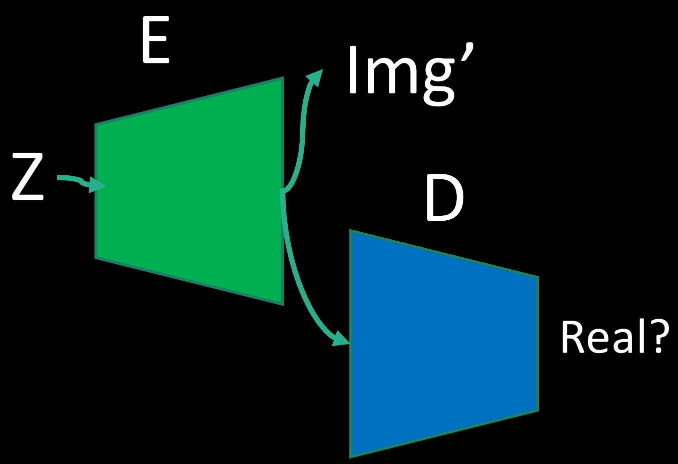
REAL


Noise
Fake Images
Real or Fake?
Real Images


GANs
VAE
GANs
VAE

Conditional GANs

GAN



Conditional GAN

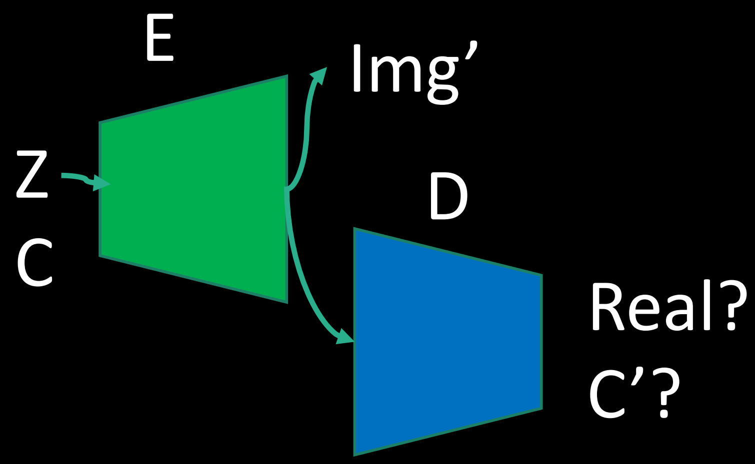


Image as condition GAN

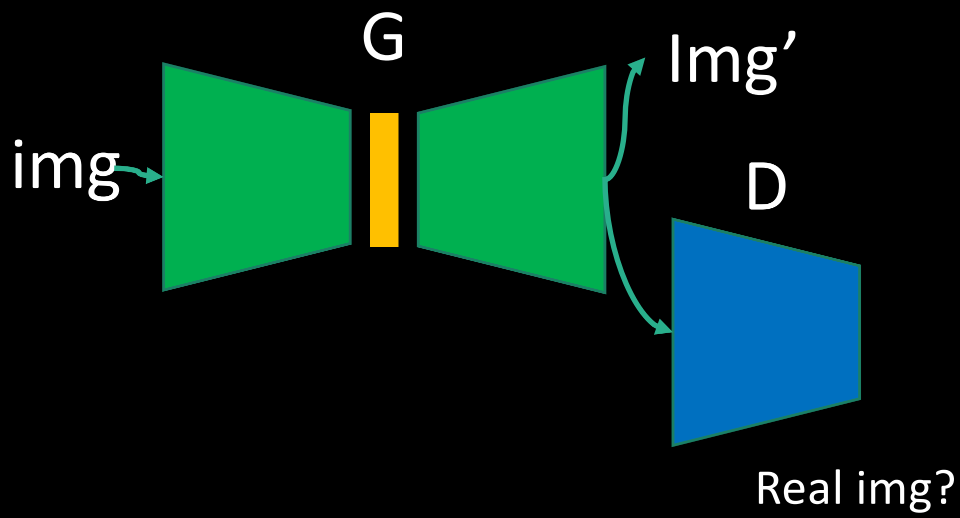
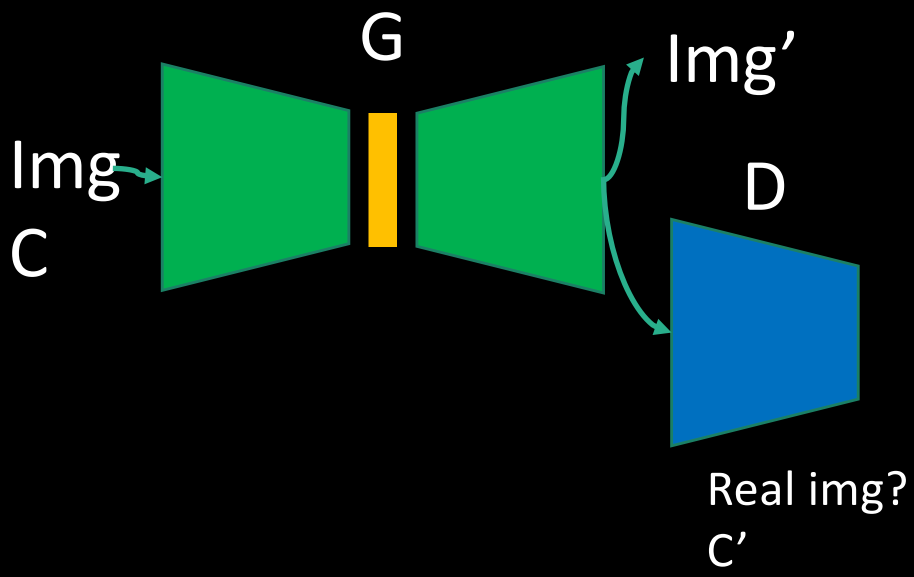


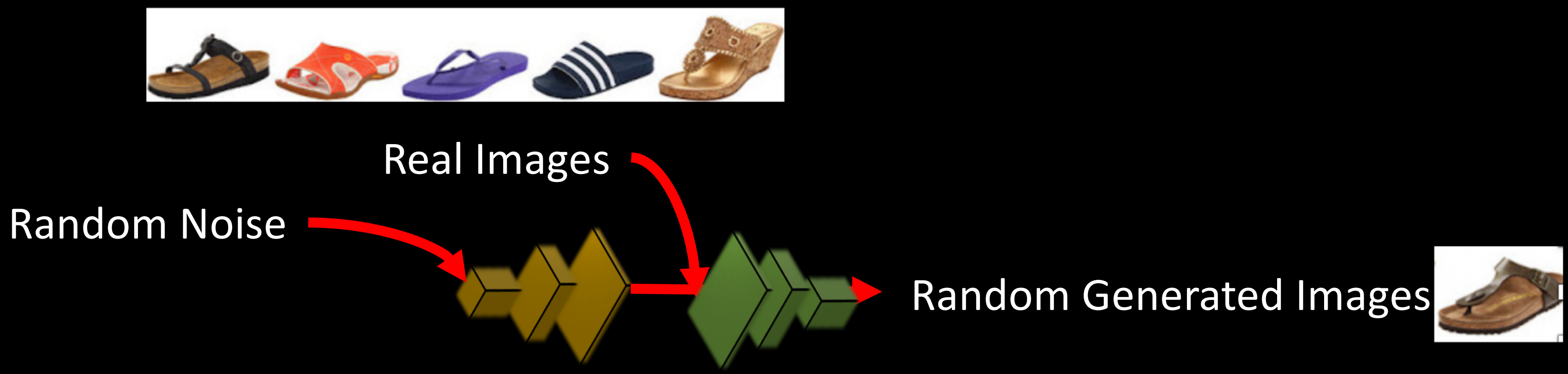
Image + condition GAN



Conditional GAN

(a) AC-GAN 10 epochs (b) AC-GAN 20 epochs (c) AC-GAN 50 epochs

GAN (no condition)



GAN (image as condition)

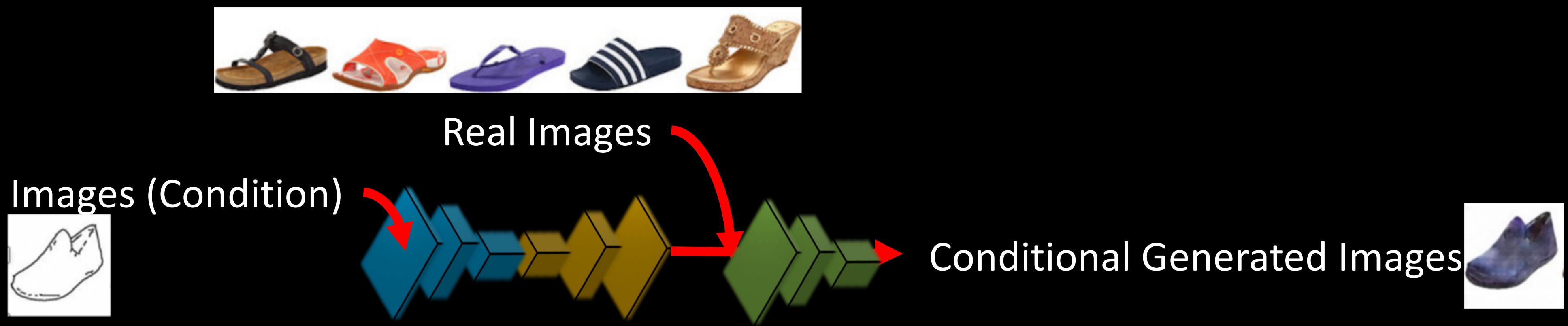


Image +
Conditional GAN

(Facial editing)

[		[
Beard		No Beard
Old	→	Old
Blond		Brown
Mouth open		Mouth close
.....]		.....]

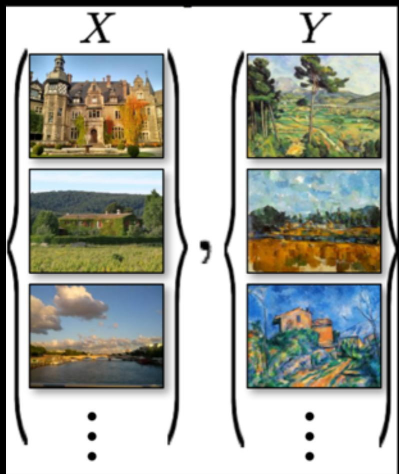
Paired (supervised)

Training:

Generating:

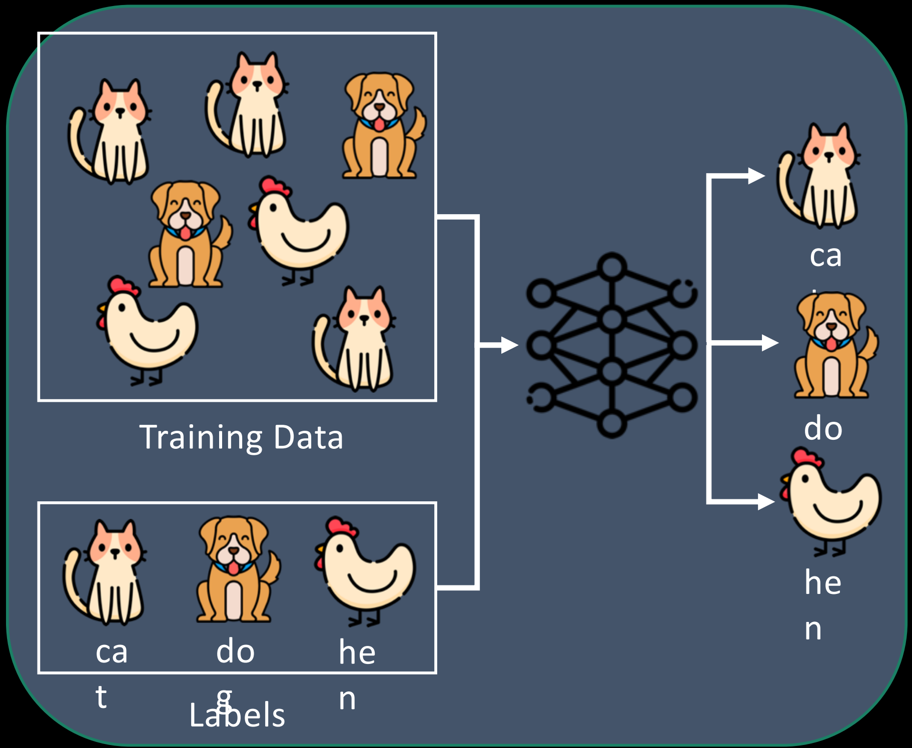
Unpaired (Unsupervised?)

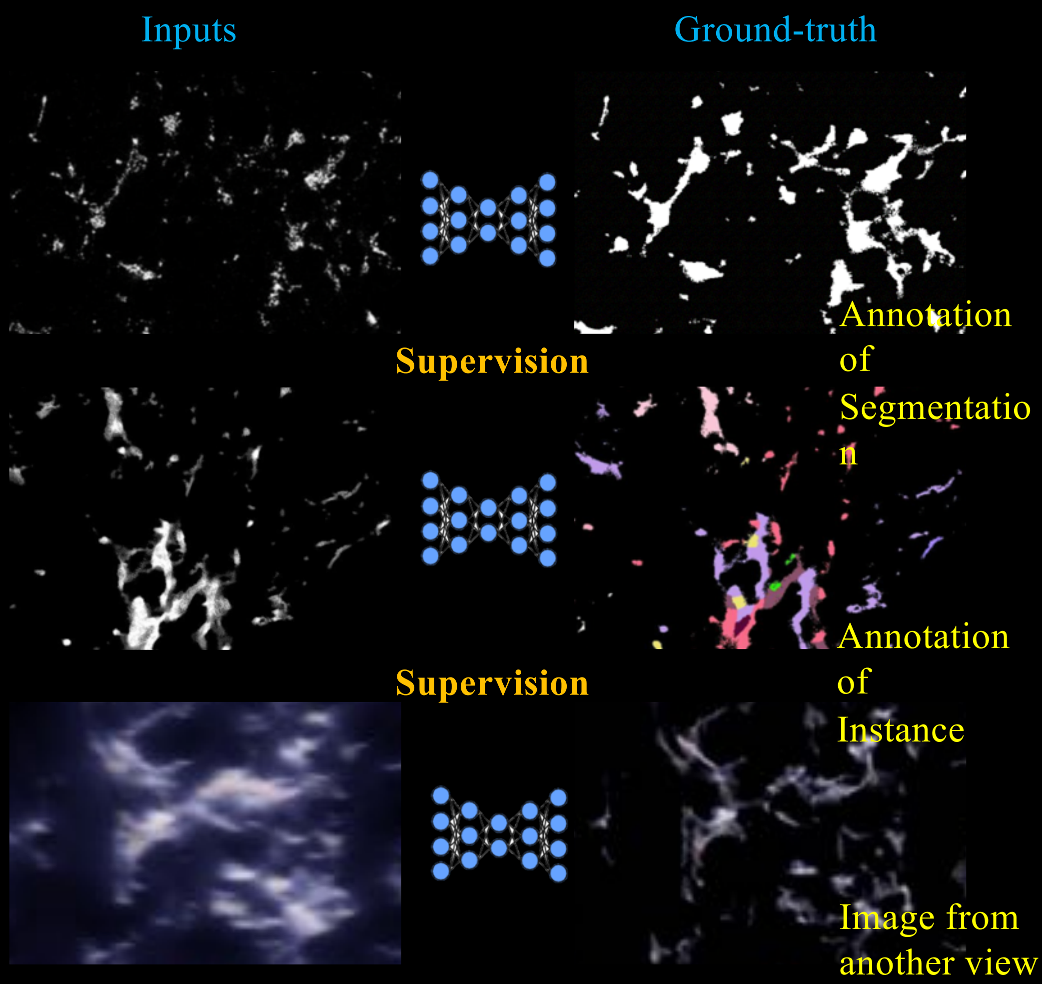
Training:



Generating:

Supervised Learning: you have the answers

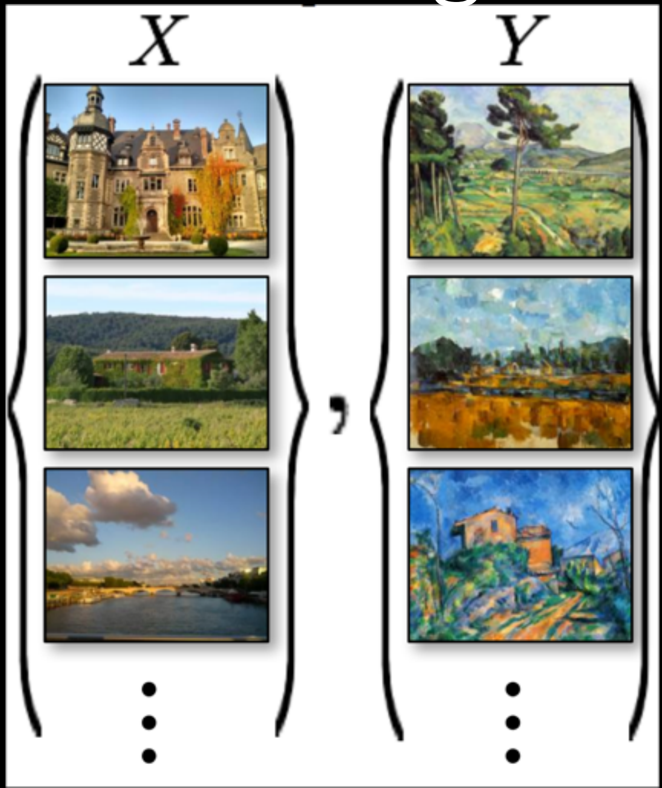


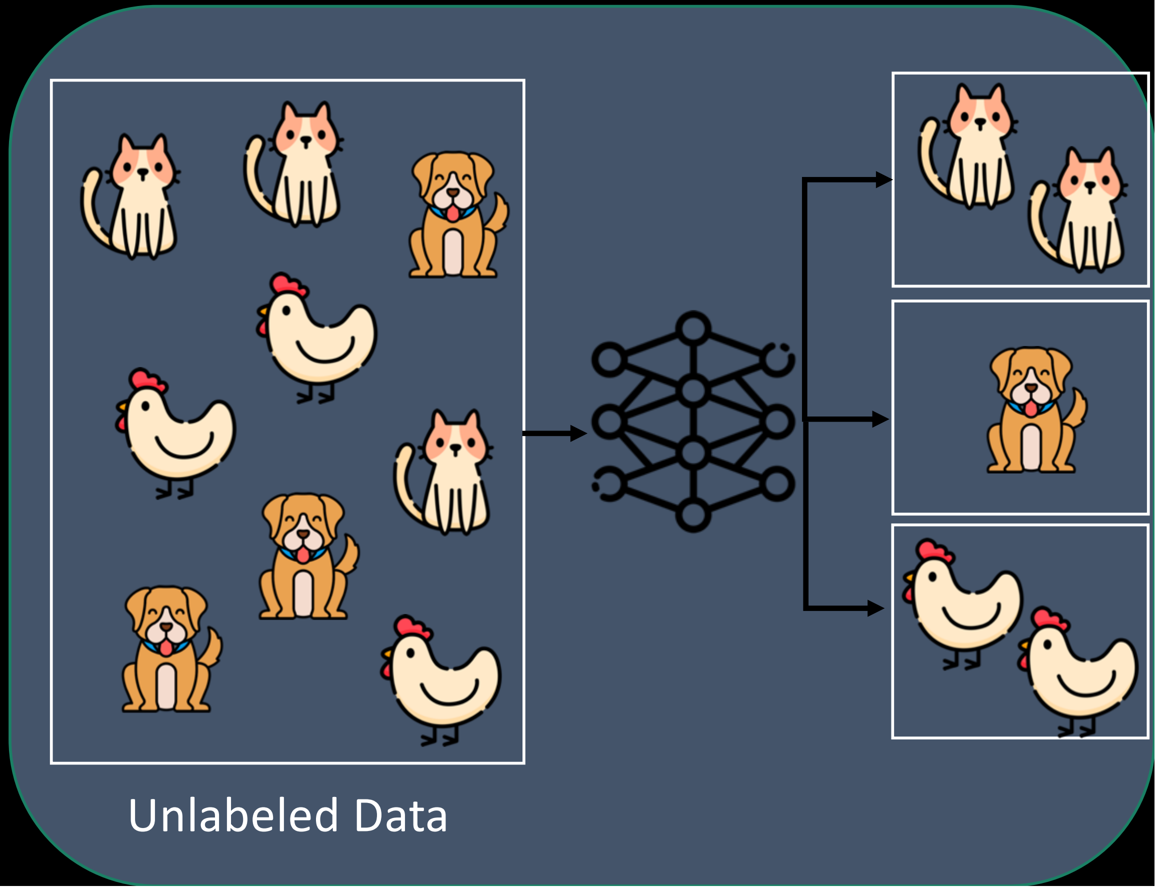


Unsupervised (unpaired) translation

Generating:

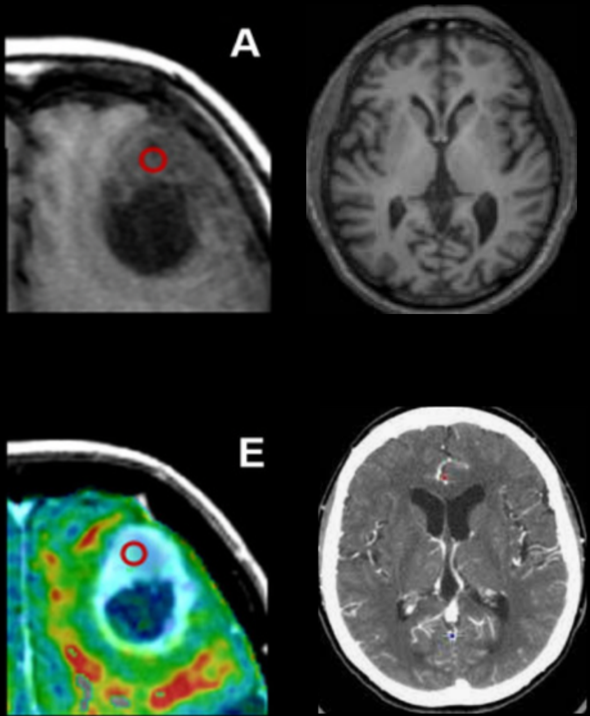
Training:



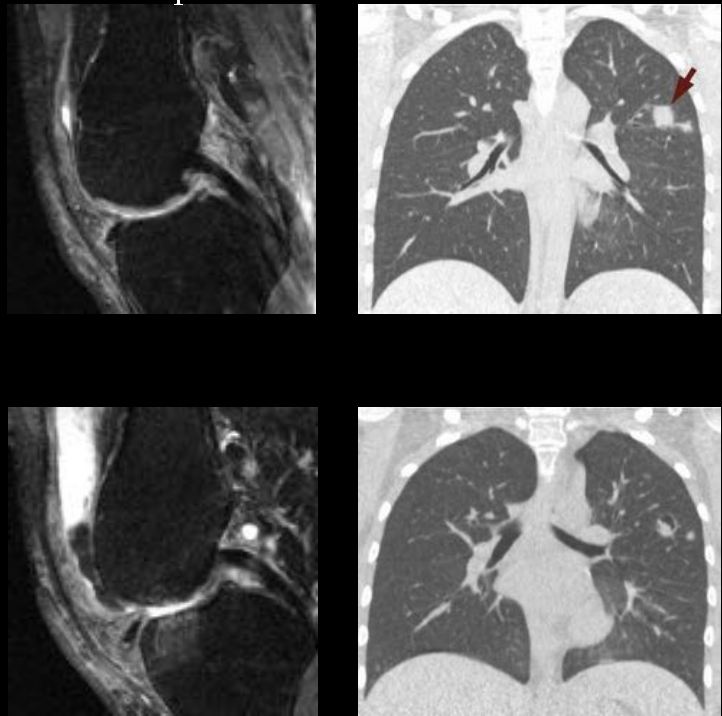


Biomedical applications : supervised

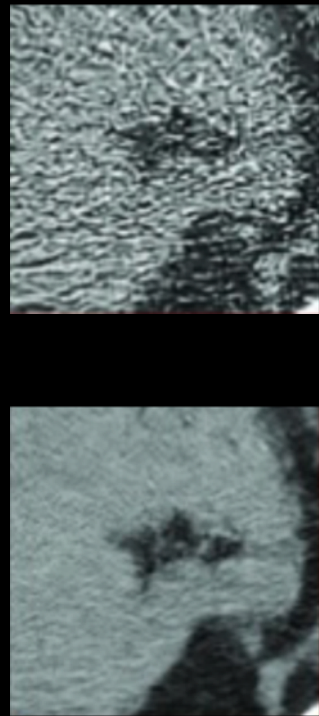
Cross-modality



Follow-ups



low tier to high tier



Supervised

Low Quality (Source)

Learned

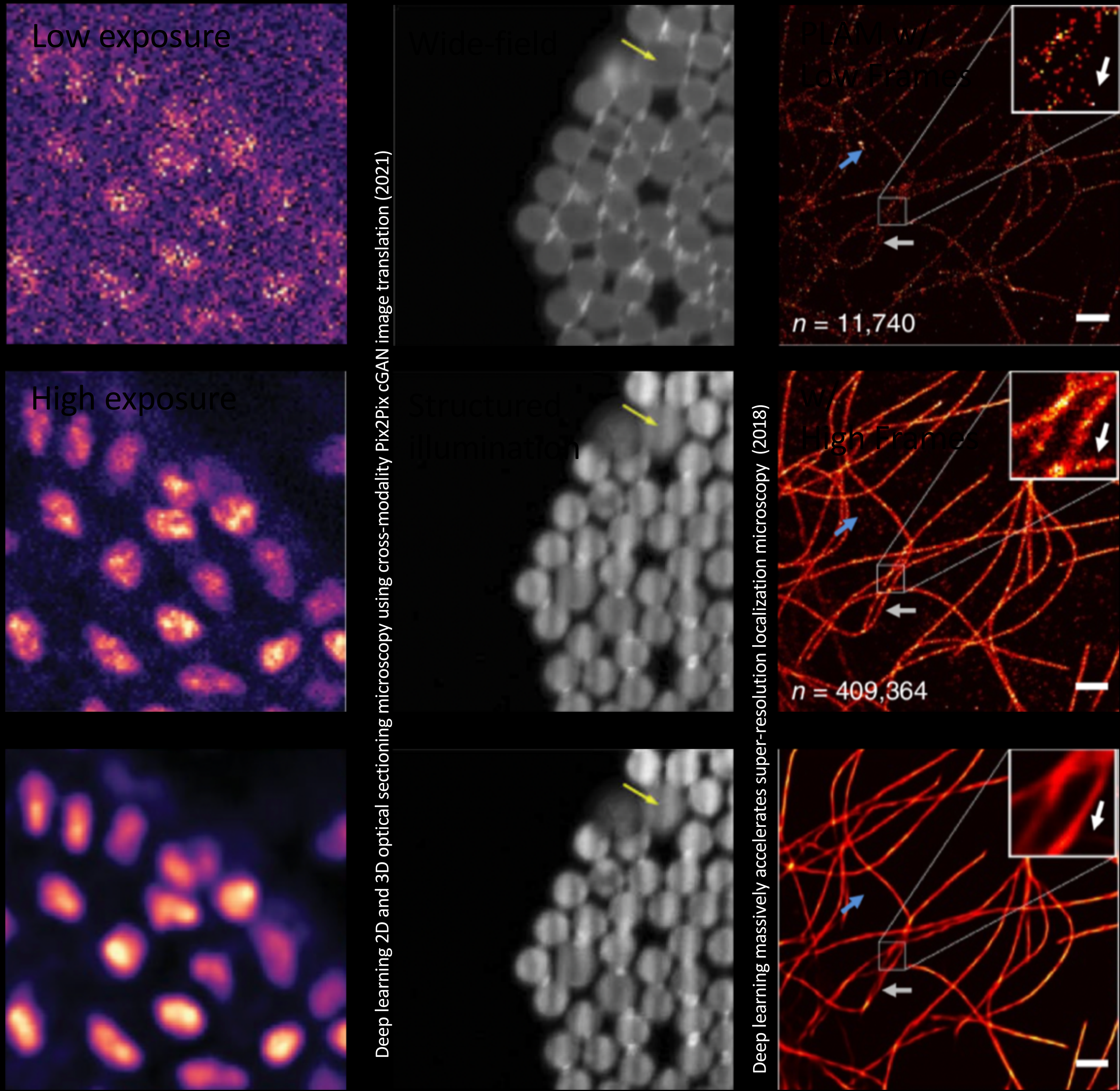
High Quality (Target)

Generated

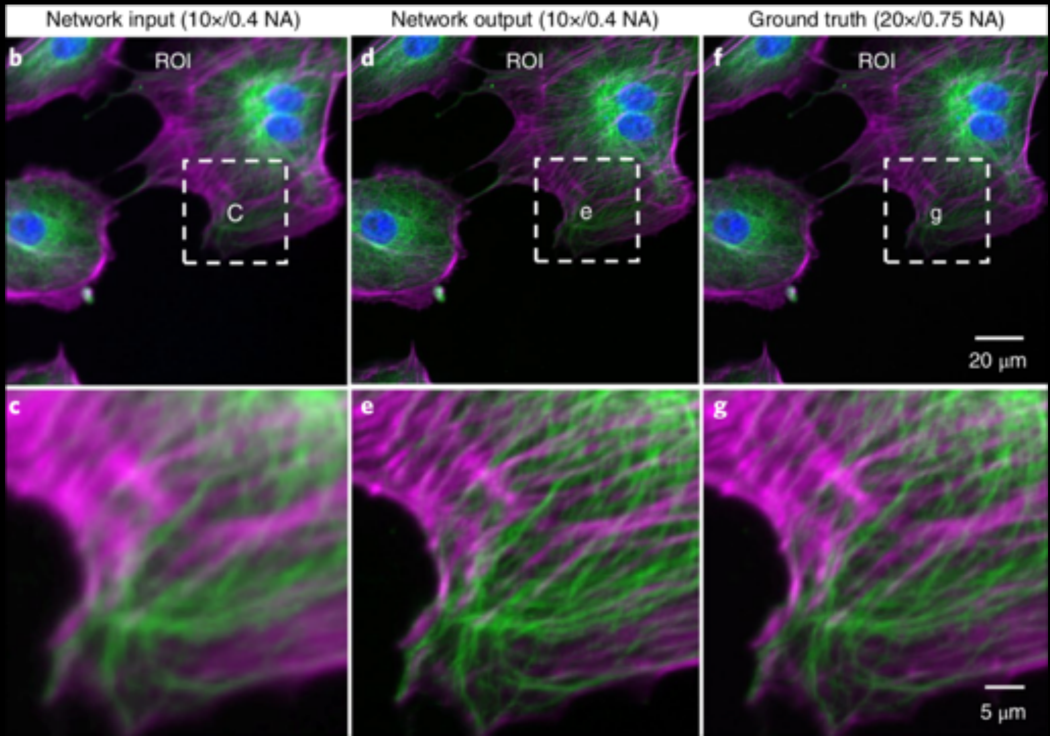
Content-aware image restoration: pushing the limits of fluorescence microscopy (2018)

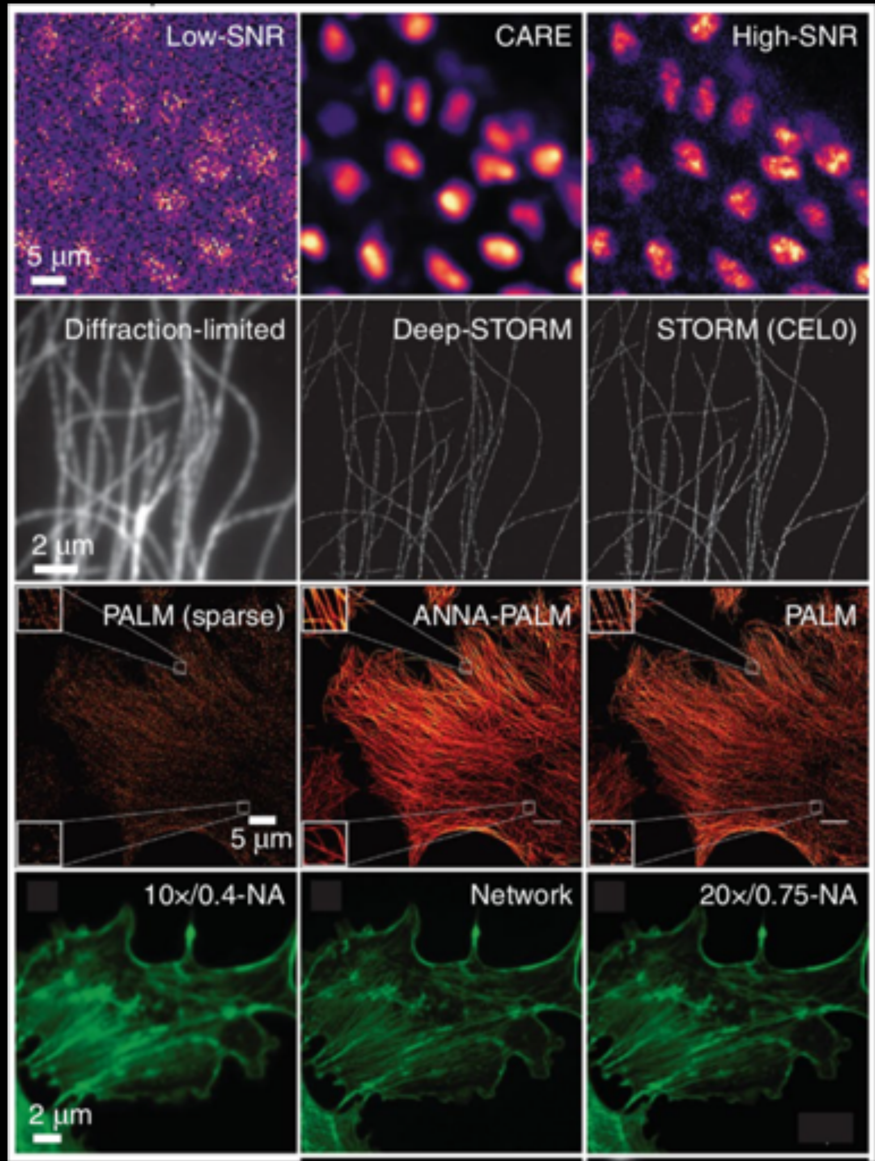
Deep learning 2D and 3D optical sectioning microscopy using cross-modality Pix2Pix cGAN image translation (2021)

Deep learning massively accelerates super-resolution localization microscopy (2018)



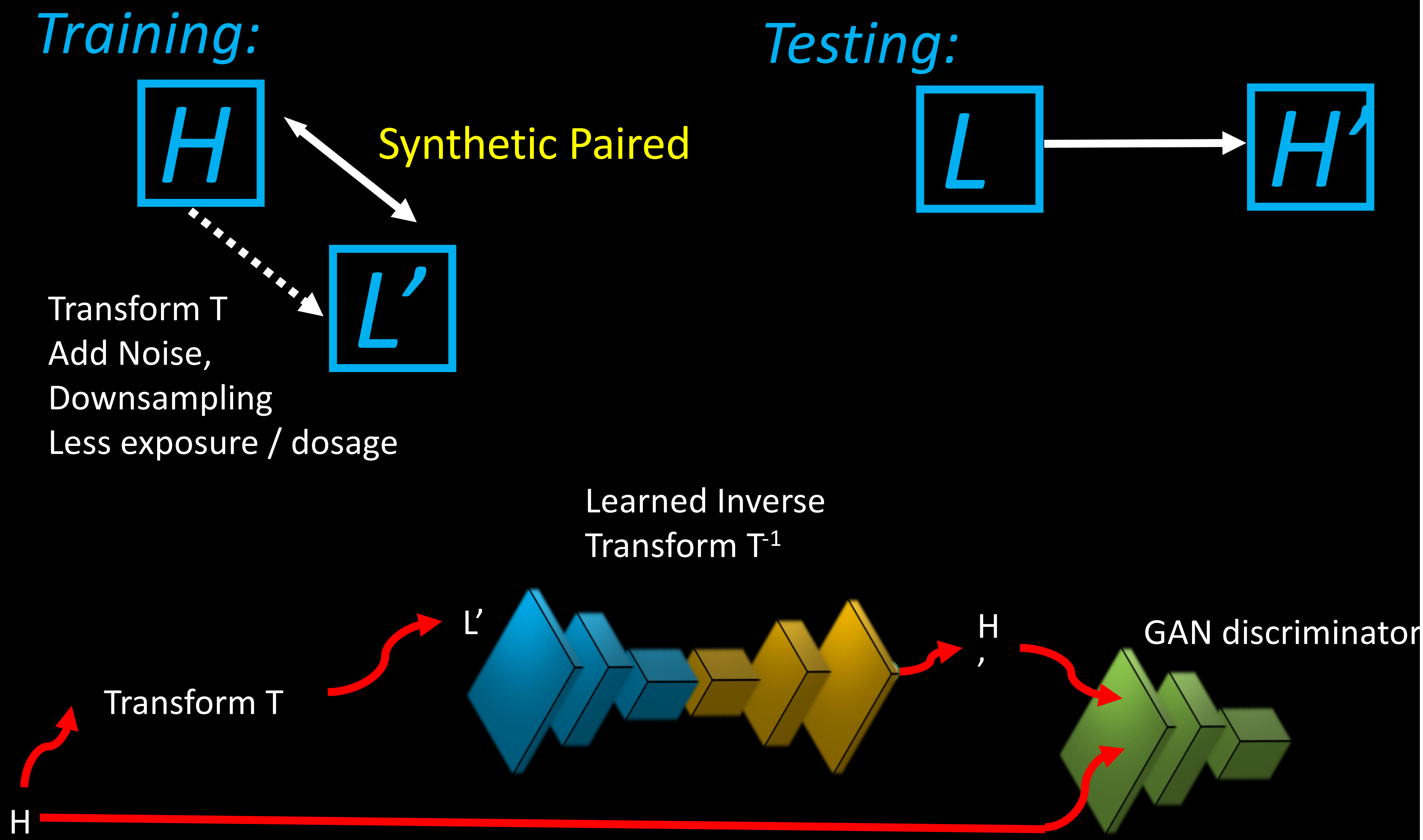
Supervised: inverse problem





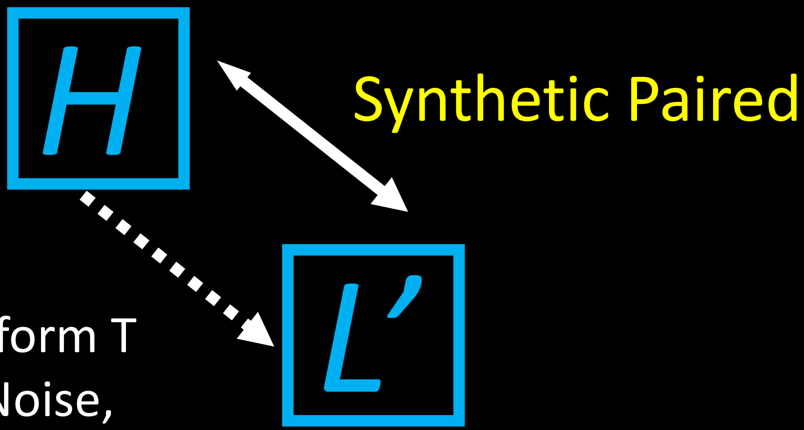
Deep learning enables cross-modality super-resolution in fluorescence microscopy (natural methods 2019)

Synthetic Supervised



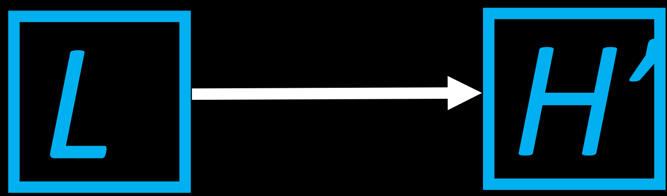
Synthetic Paired

Training:



Transform T
Add Noise,
Downsampling
Less exposure / dosage

Testing:

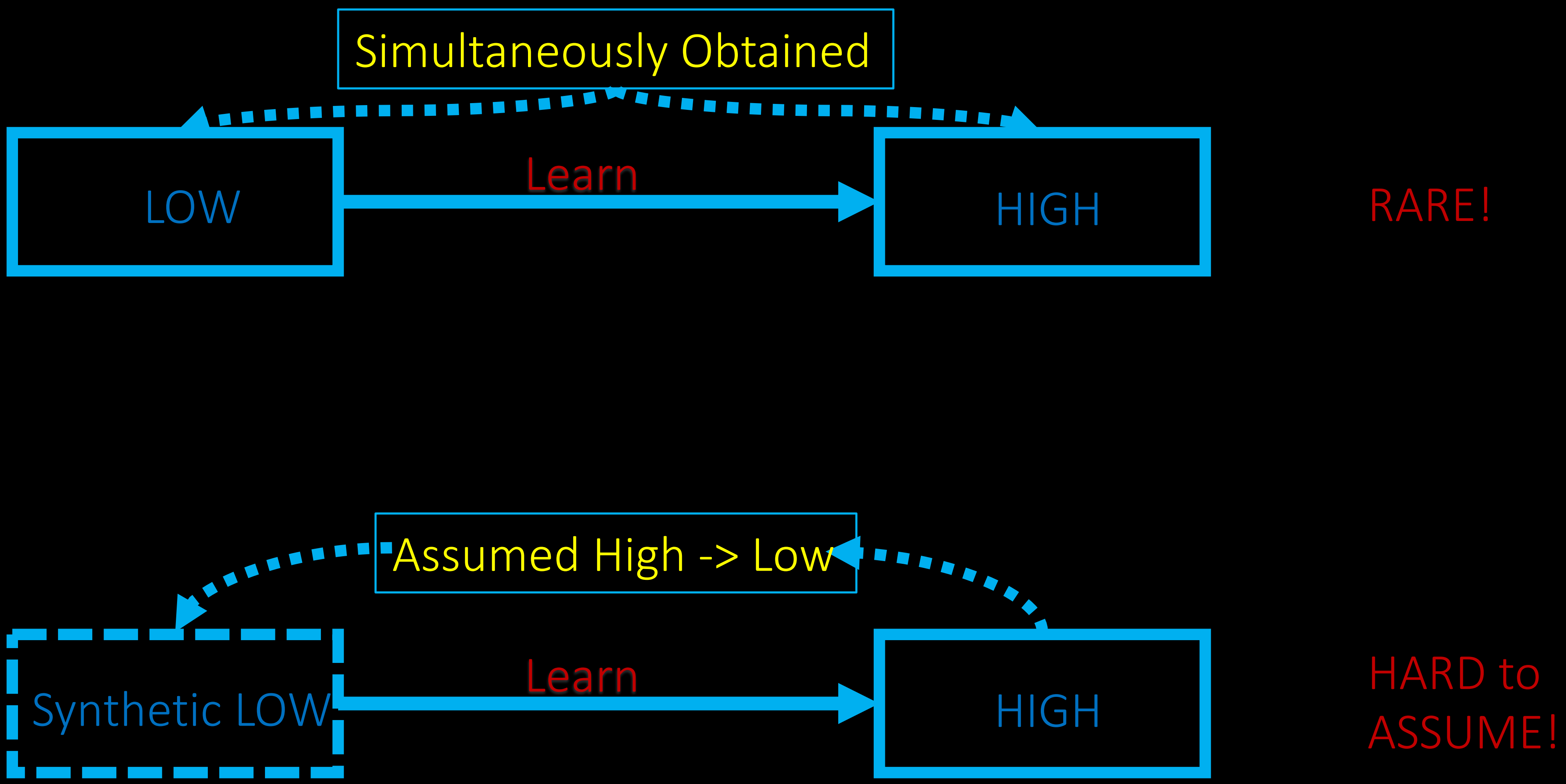


Problem with synthetic pairs:

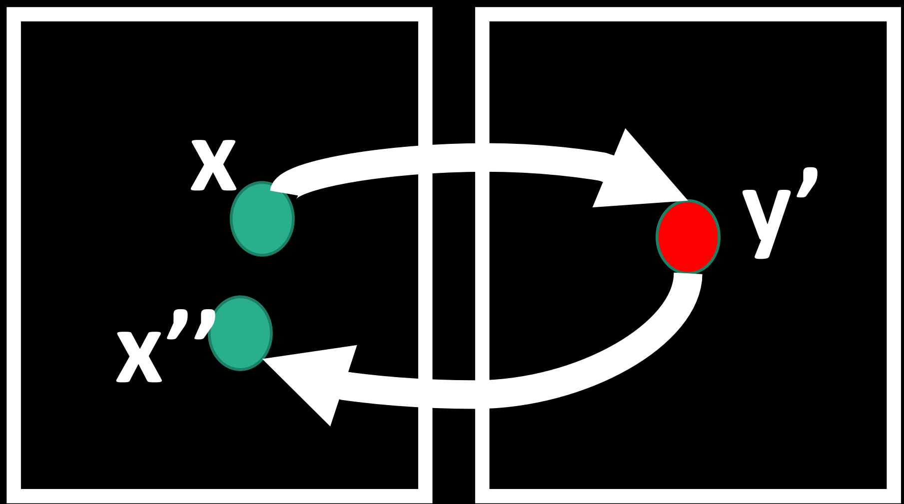
T may not be modeled accurately

T(L > H) is hard to assume

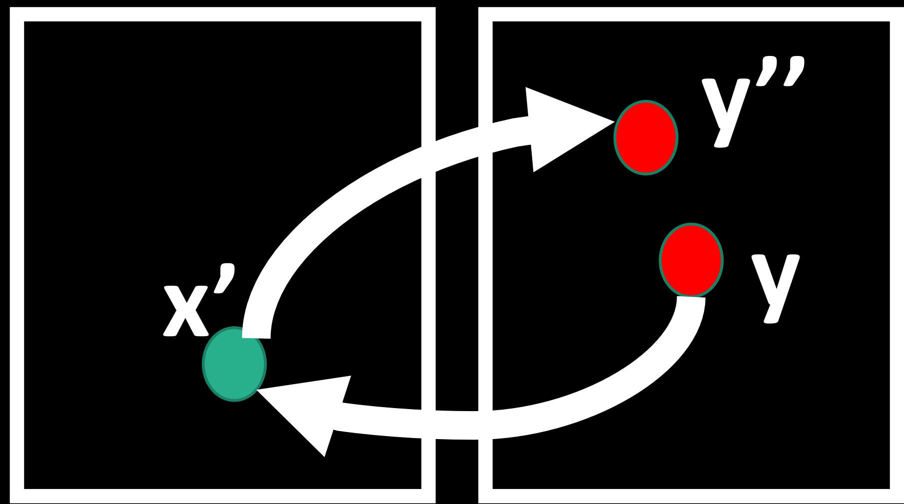
Summary of supervised (Paired) translation



Unpaired learning by "Cyclic Consistency"

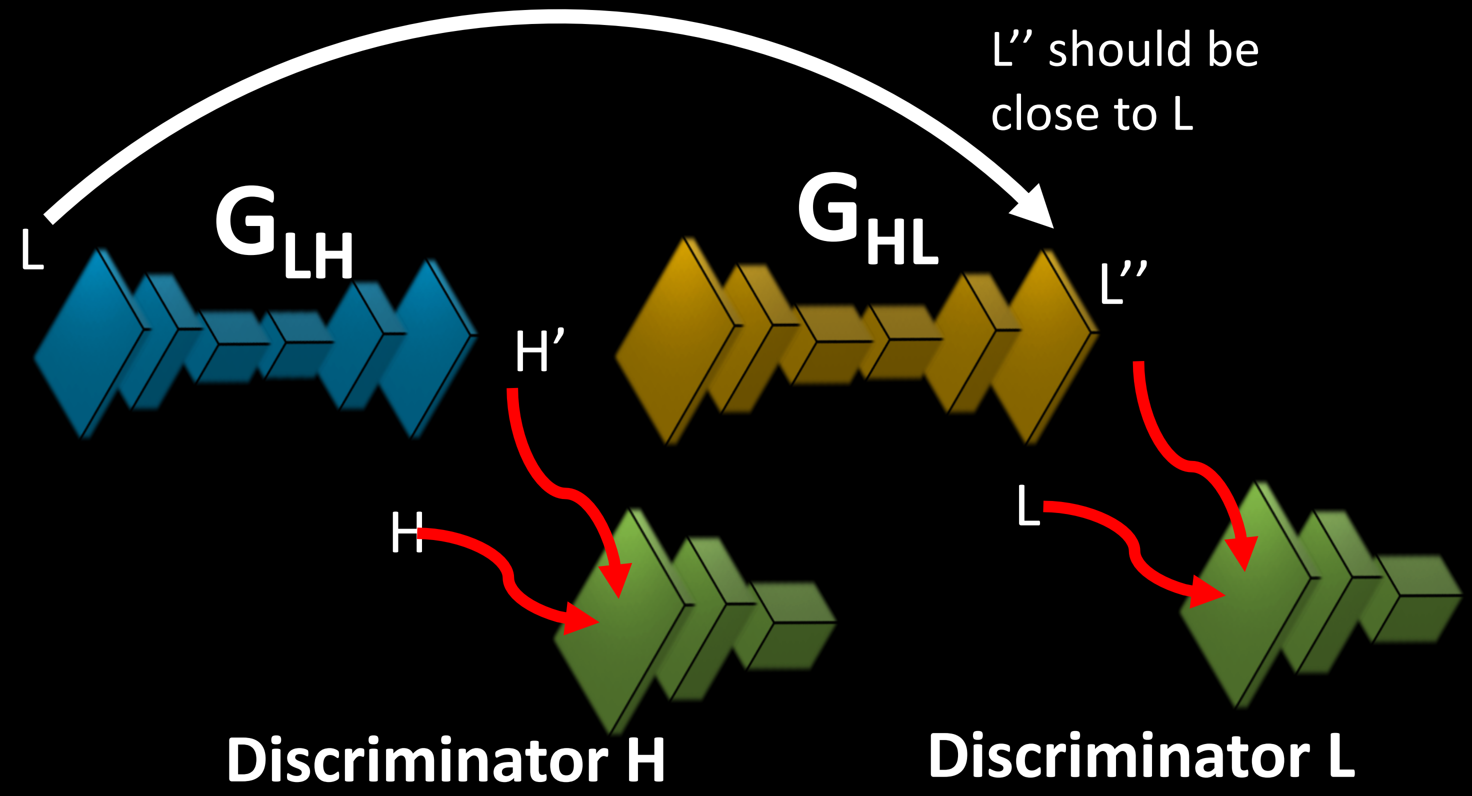


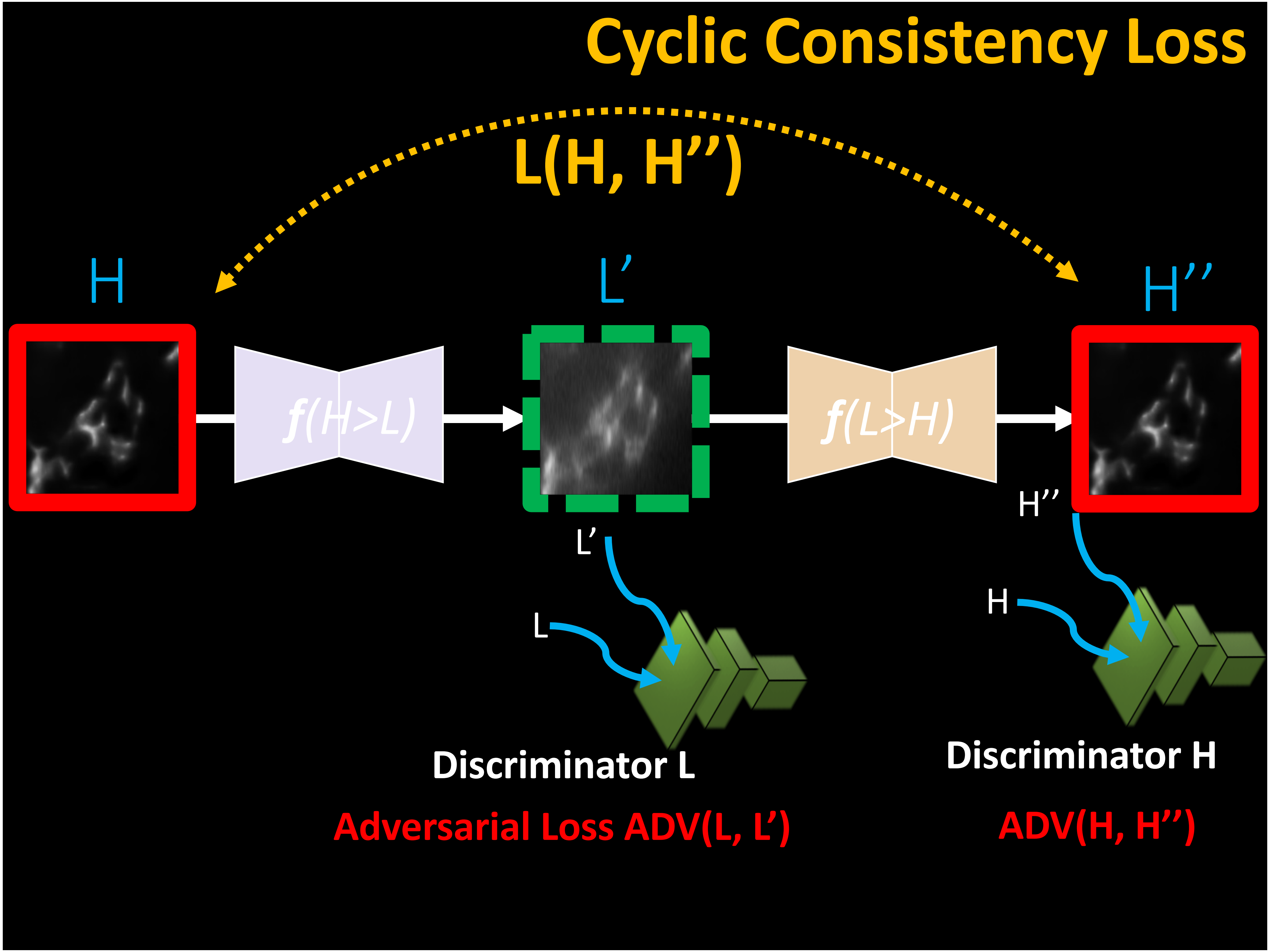
x should be close to x''



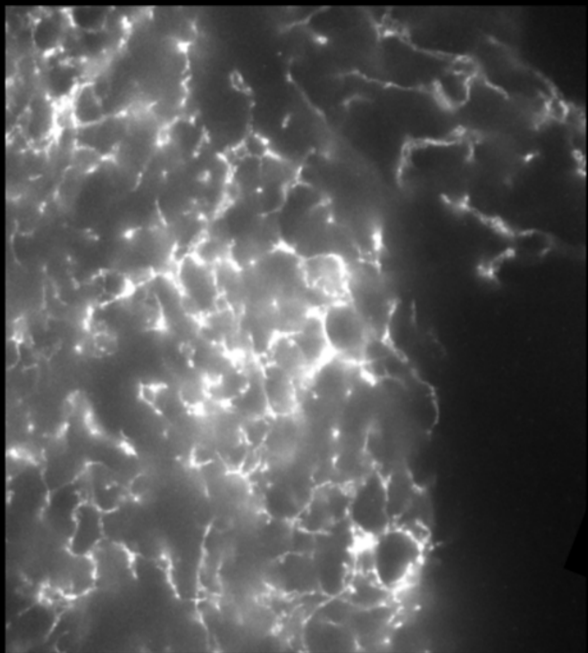
y should be close to y''

Inverse problem with unpaired data (CycleGAN, DiscoGAN, DualGAN etc.)




Cyclic Consistency Loss
L(H, H'')
H
L'
H''
f(H>L)
f(L>H)
L'
H''
L
H
Discriminator L
Discriminator H
Adversarial Loss ADV(L, L')
ADV(H, H'')

Unsupervised Learning: learn from the data



Data too costly to label

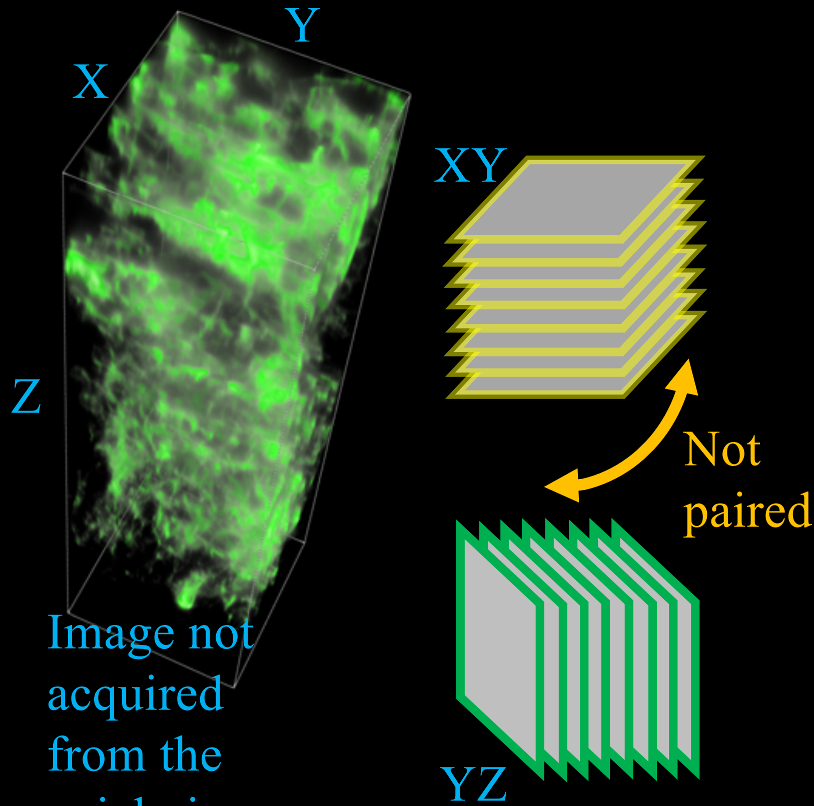
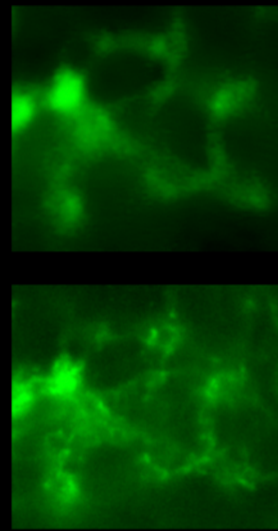
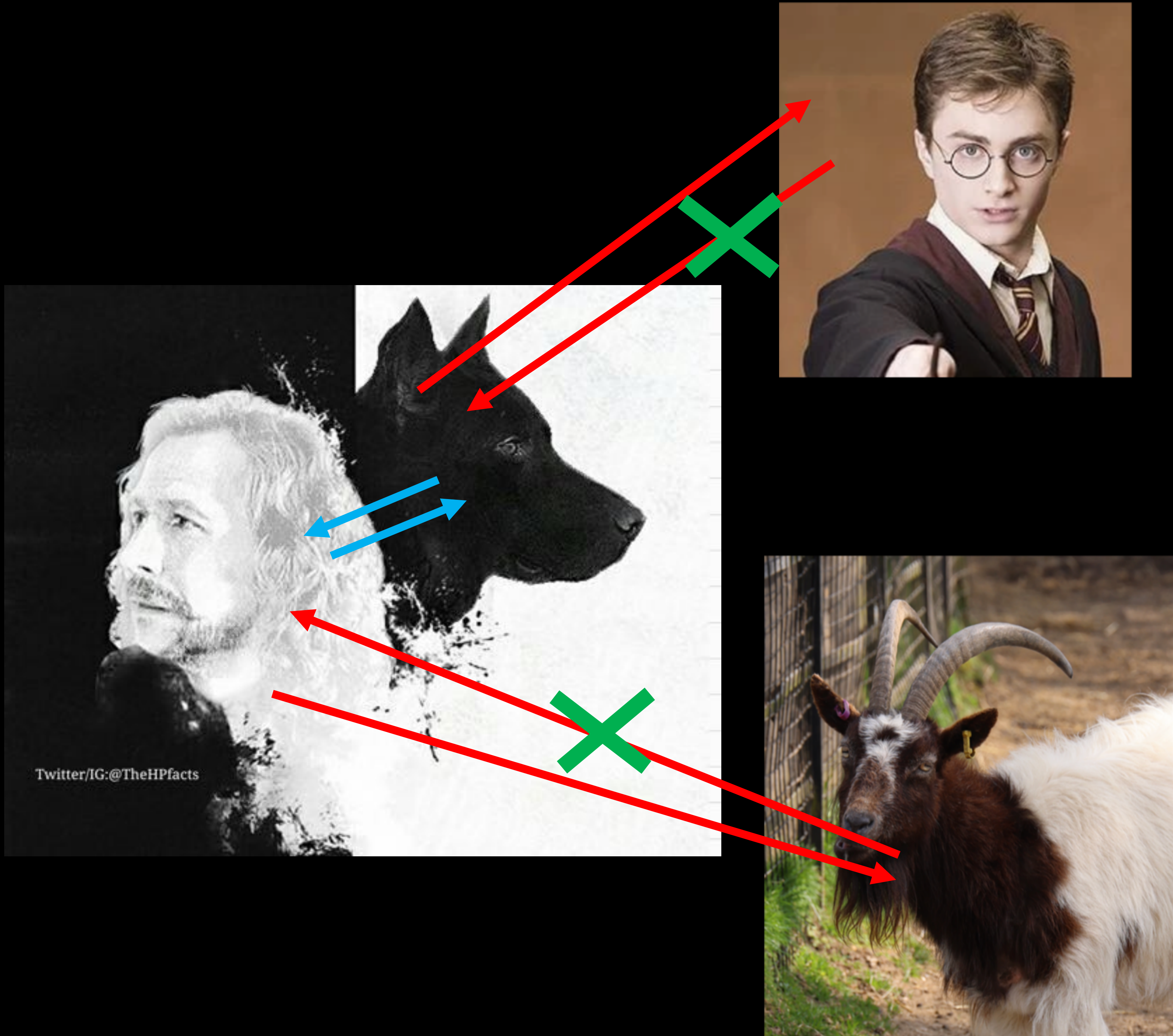
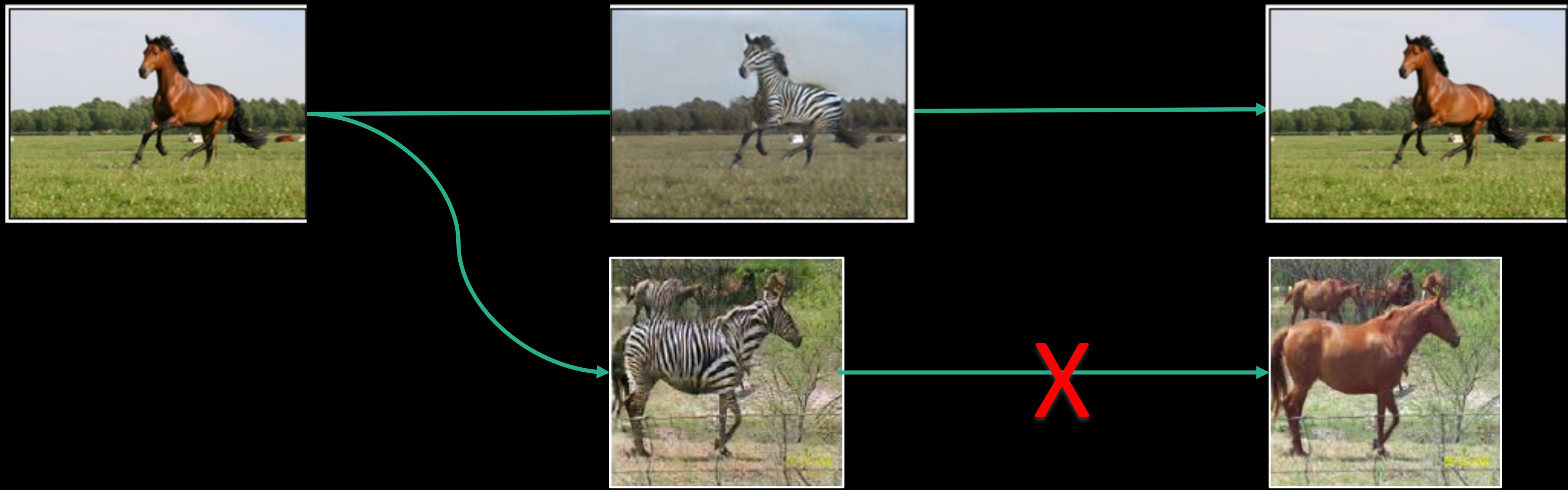


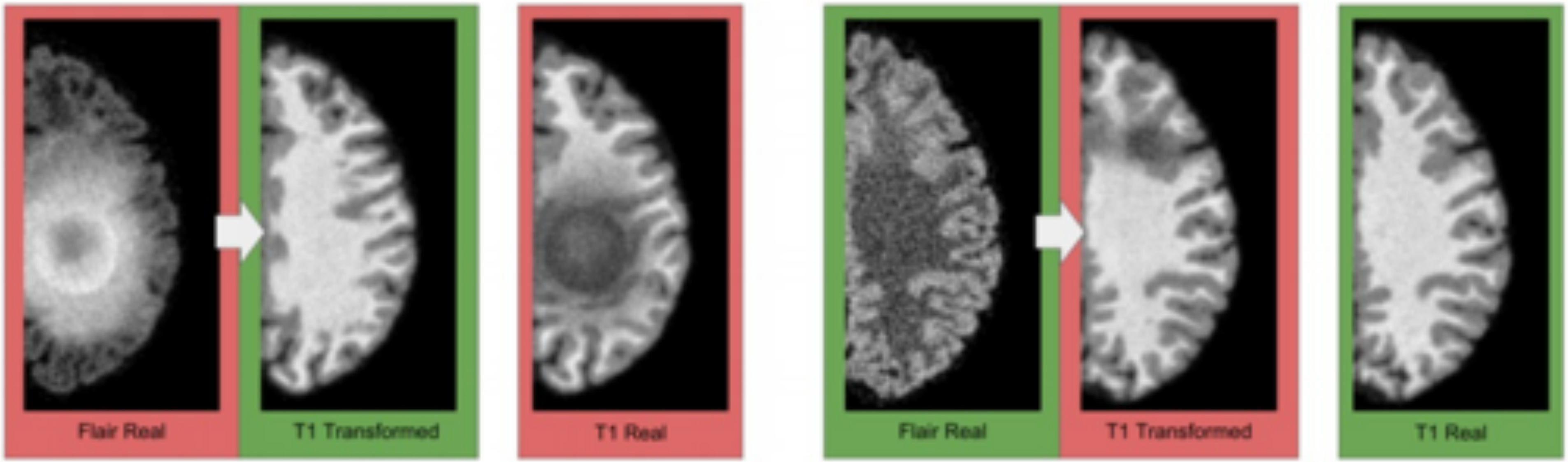
Image not acquired from the axial view



Difficult to acquire imaging with ground-truth

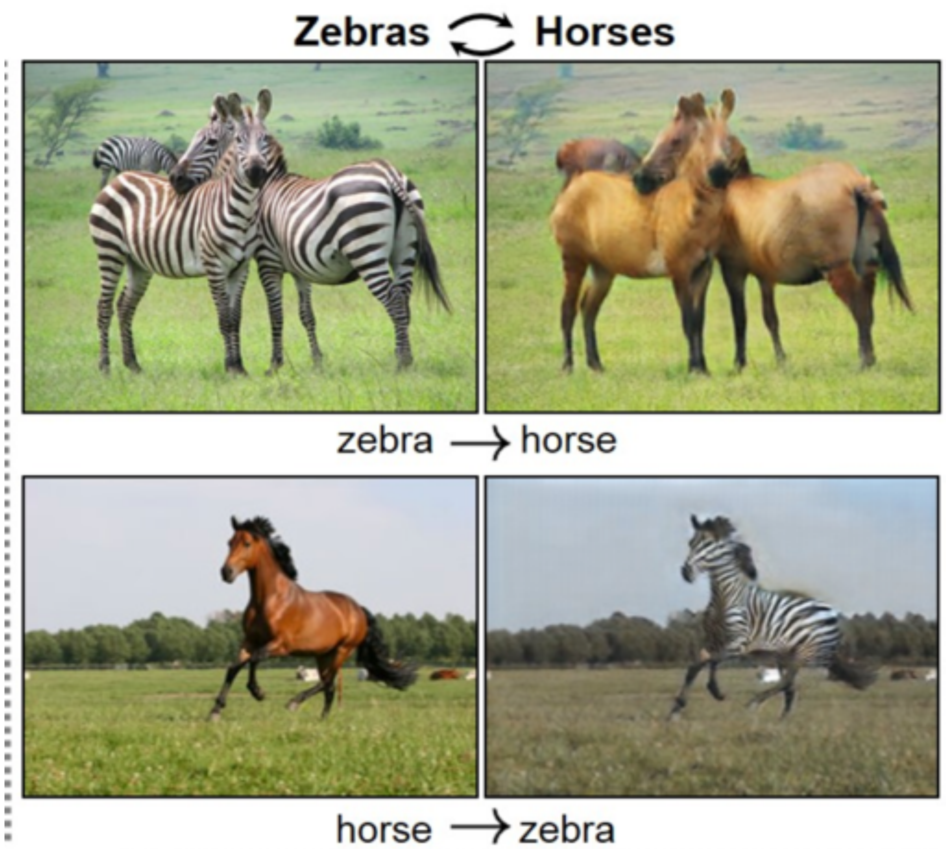

Twitter/IG:@TheHPfacts

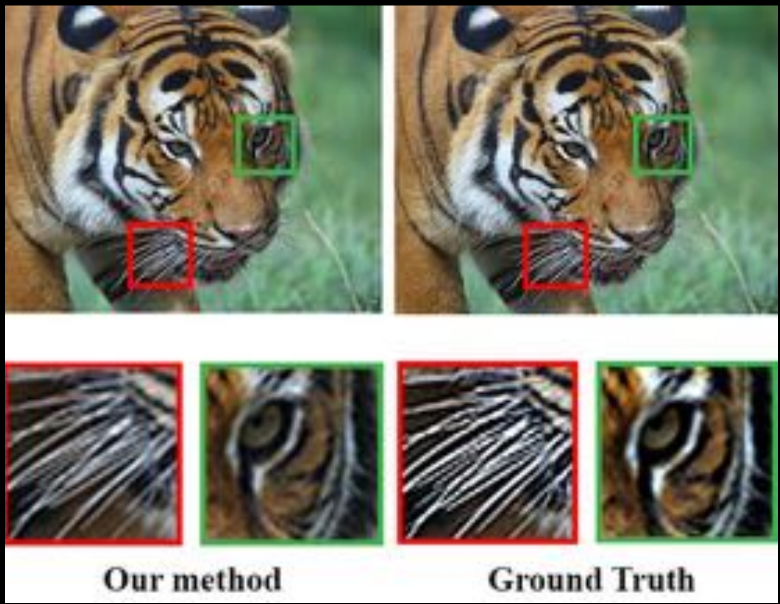


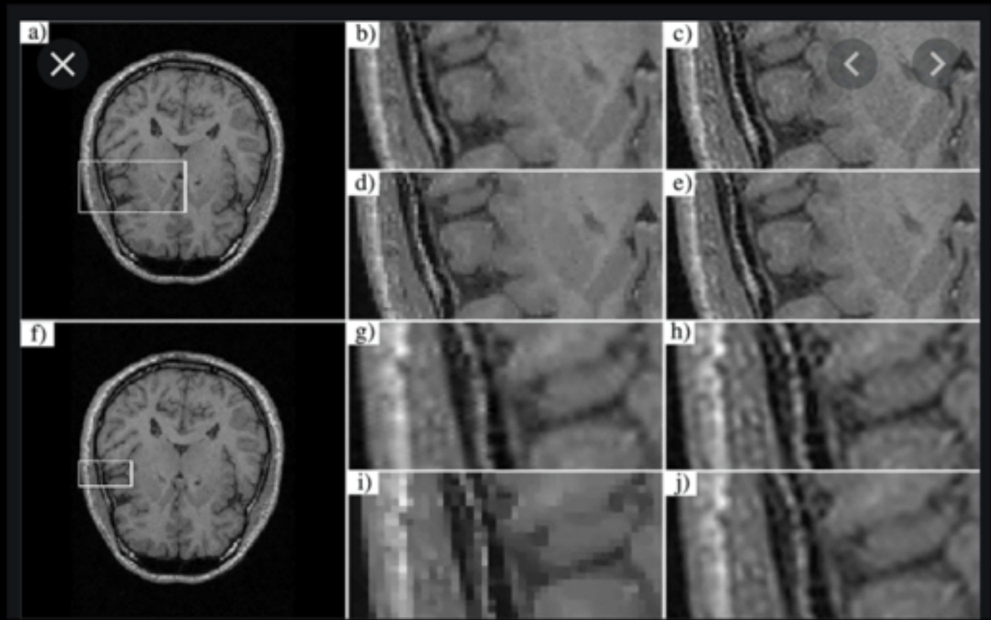


(a) A translation removing tumors

(b) A translation adding tumors


Zebras ⇄ Horses
zebra → horse
horse → zebra





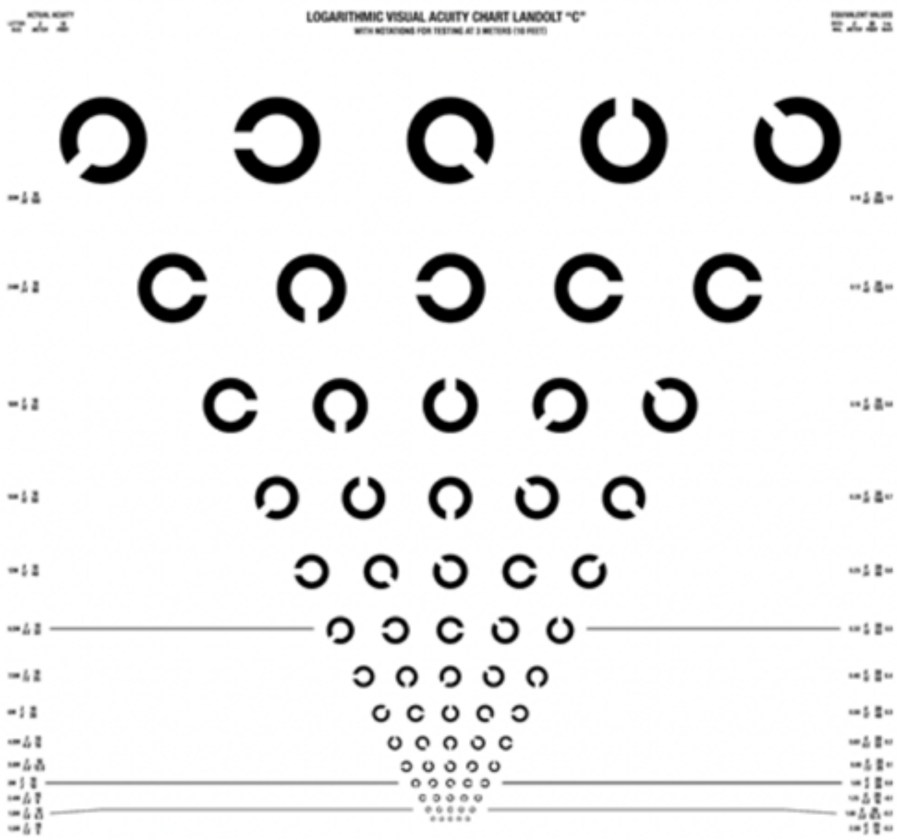
No intrinsic physical dimension

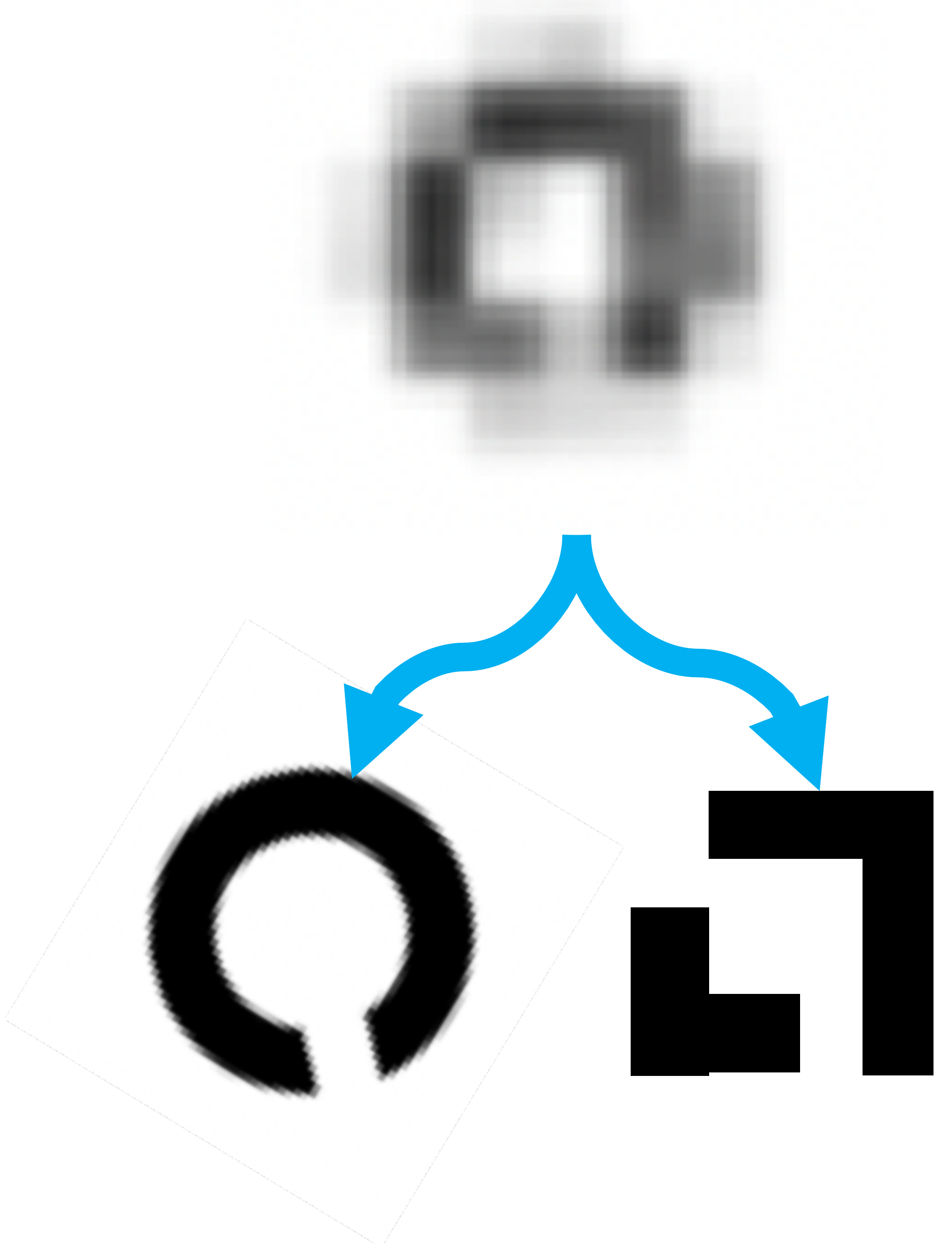
Mostly can be judged by PSNR, SSIM…etc..

Has intrinsic physical dimensions

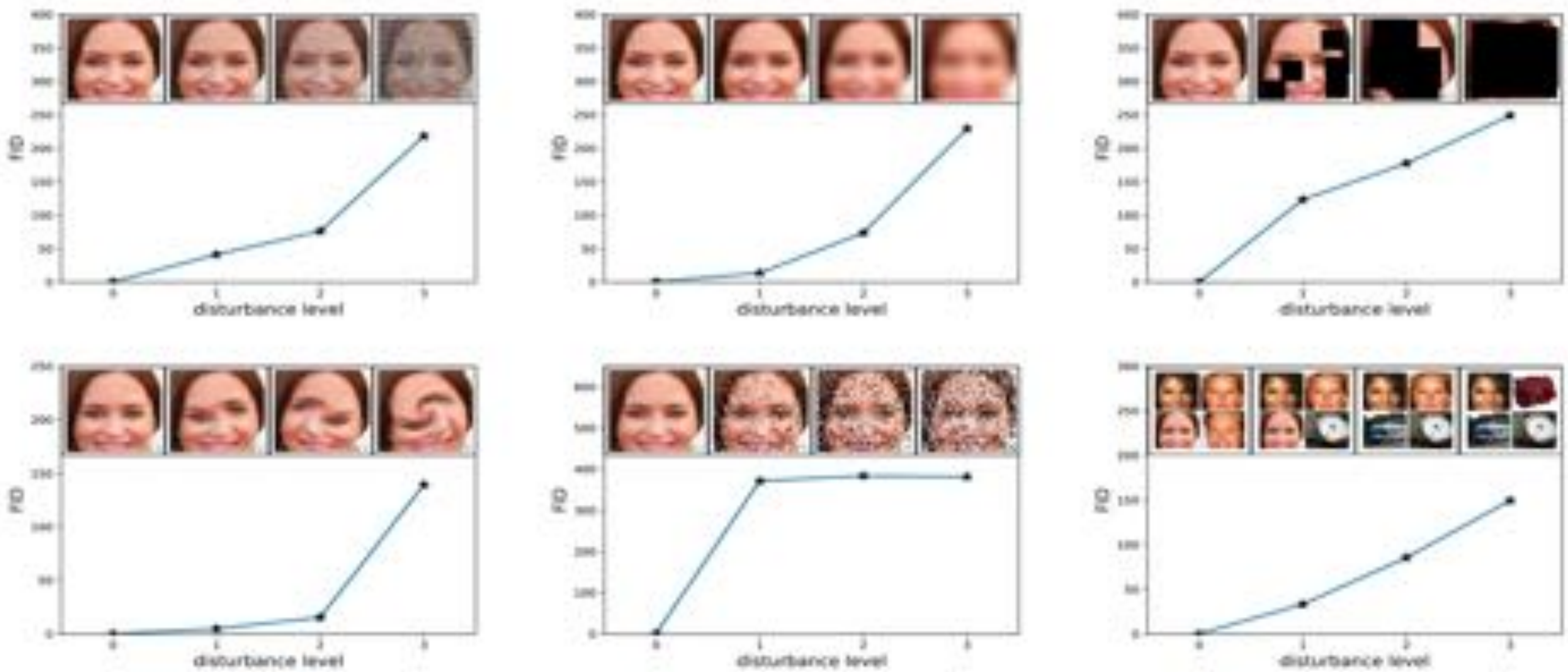
Need to be judged by some physical / clinical outcome

Are those real? "Educated Guess"

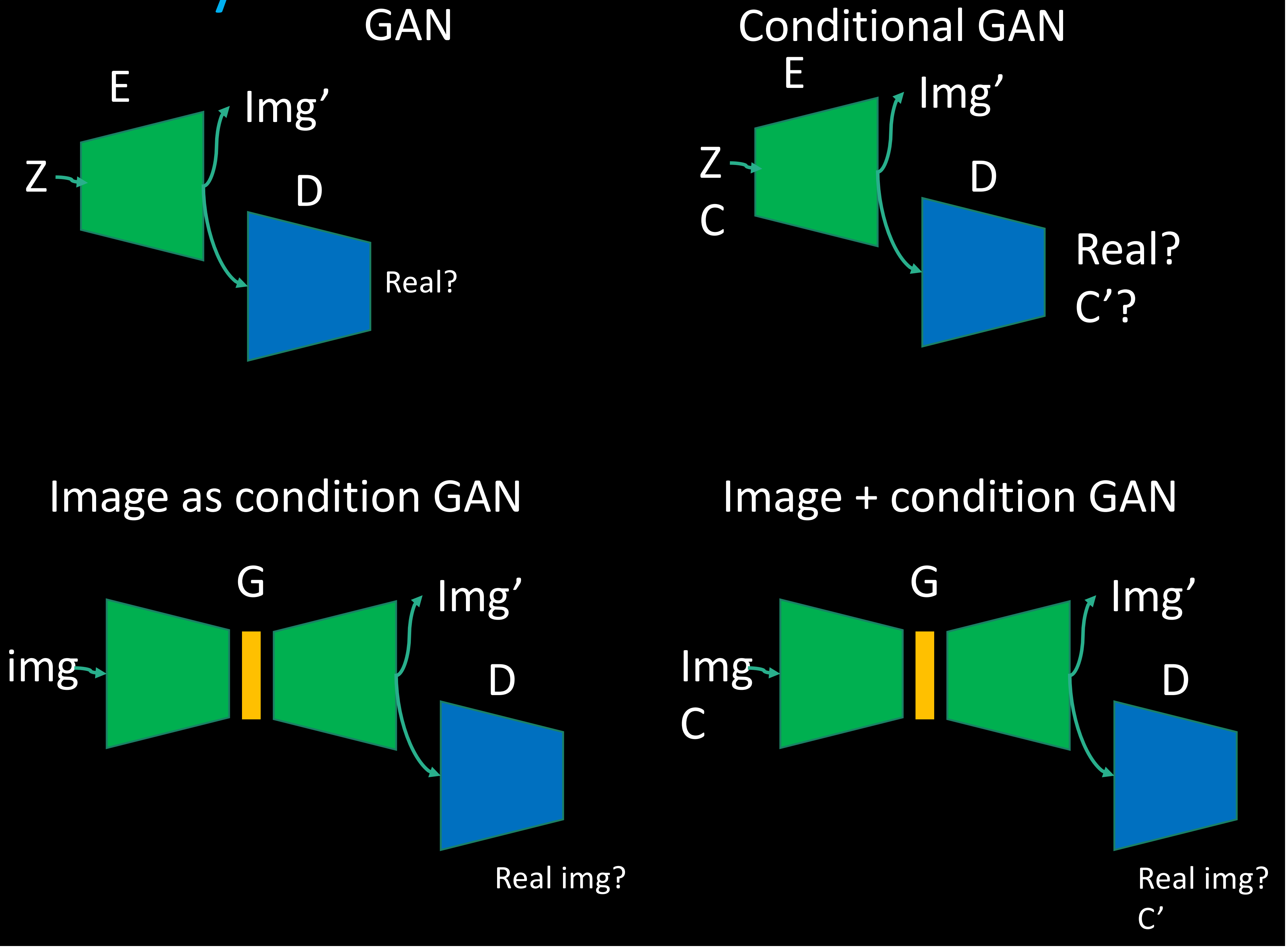




Quality Measurements: FID



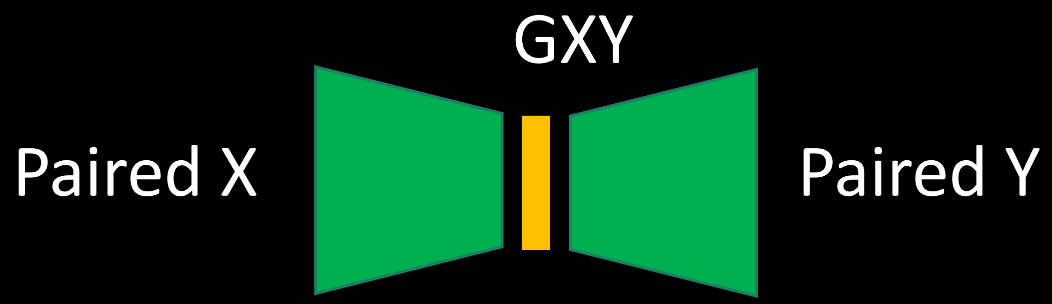
Summary



GAN
E
Img’
Z
D
Real?
Conditional GAN
E
Img’
Z
C
D
Real?
C’?
Image as condition GAN
G
Img’
img
D
Real img?
Image + condition GAN
G
Img’
Img
C
D
Real img?
C’

Summary

Paired



Unpaired (these problem is not solved and has many approaches)