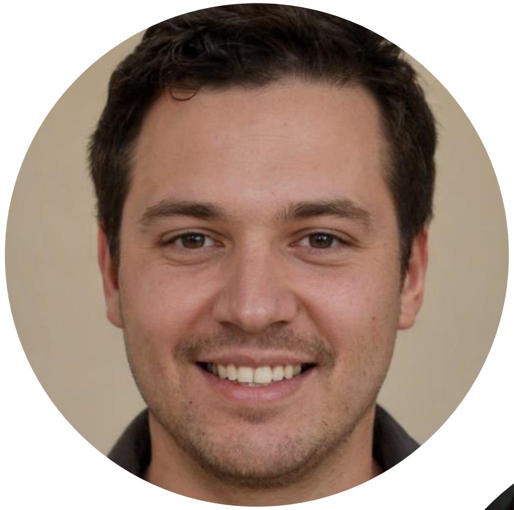
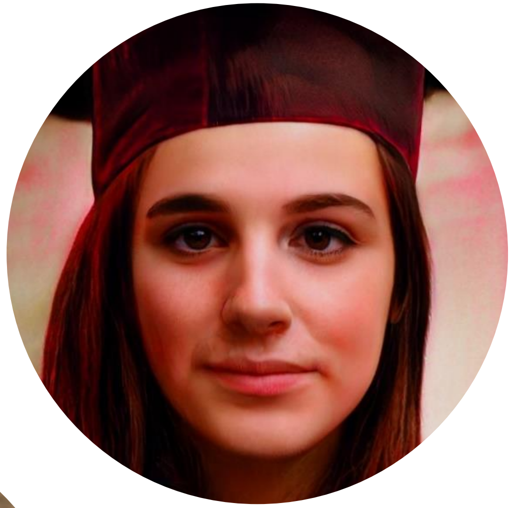
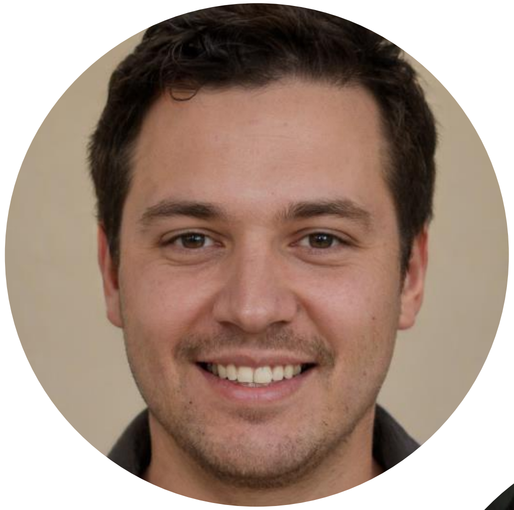
Introduction to Generative Models

Introduction to Generative Models





Introduction to Generative Models



Just kidding- They don’t exist

Agenda

Agenda

Today: Basics, Next week: Advanced

Agenda

Today: Basics, Next week: Advanced

1. Goal, motivation, Basic methods
    1. Parametric methods
    2. Autoregressive methods
    3. Latent space mapping

Agenda

Today: Basics, Next week: Advanced

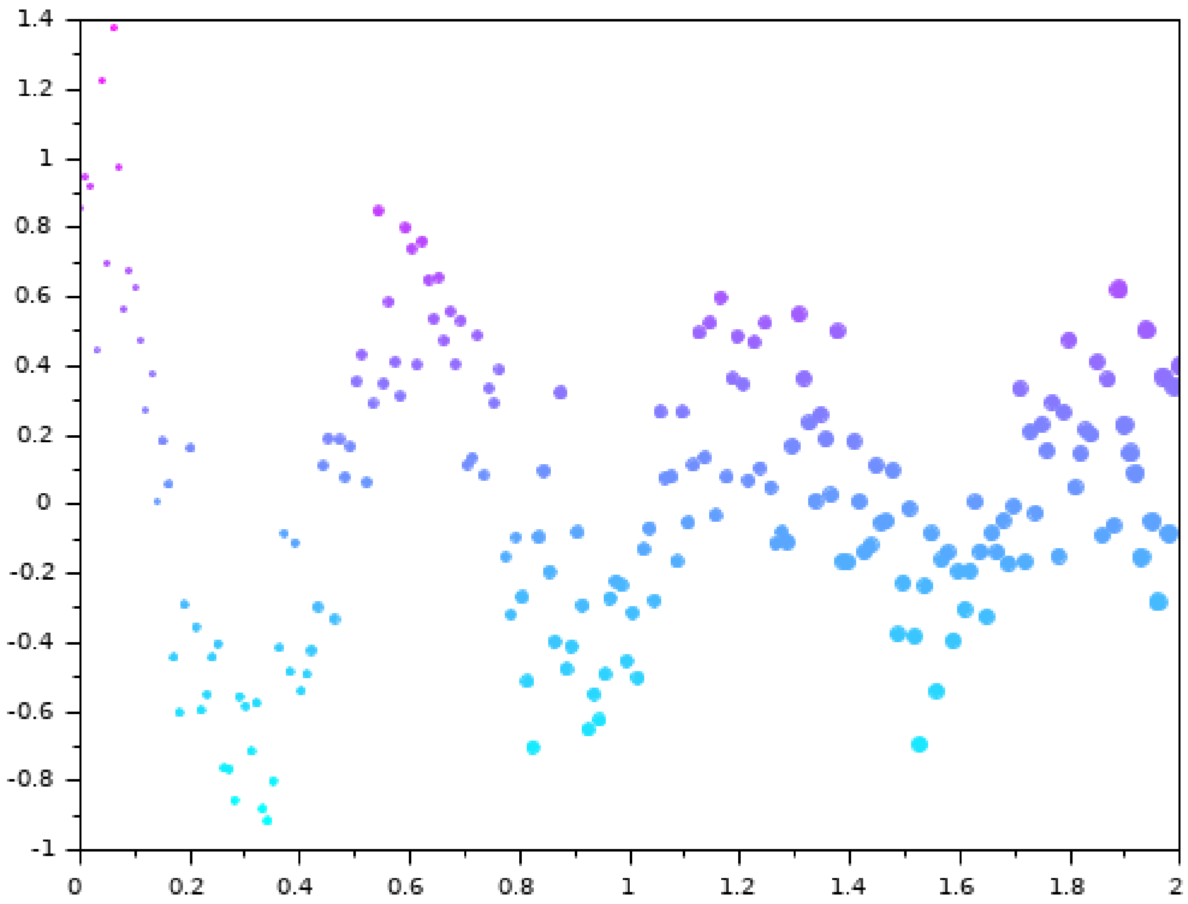
1. Goal, motivation, Basic methods
    1. Parametric methods
    2. Autoregressive methods
    3. Latent space mapping
2. Variational Auto Encoder (VAE)

Agenda

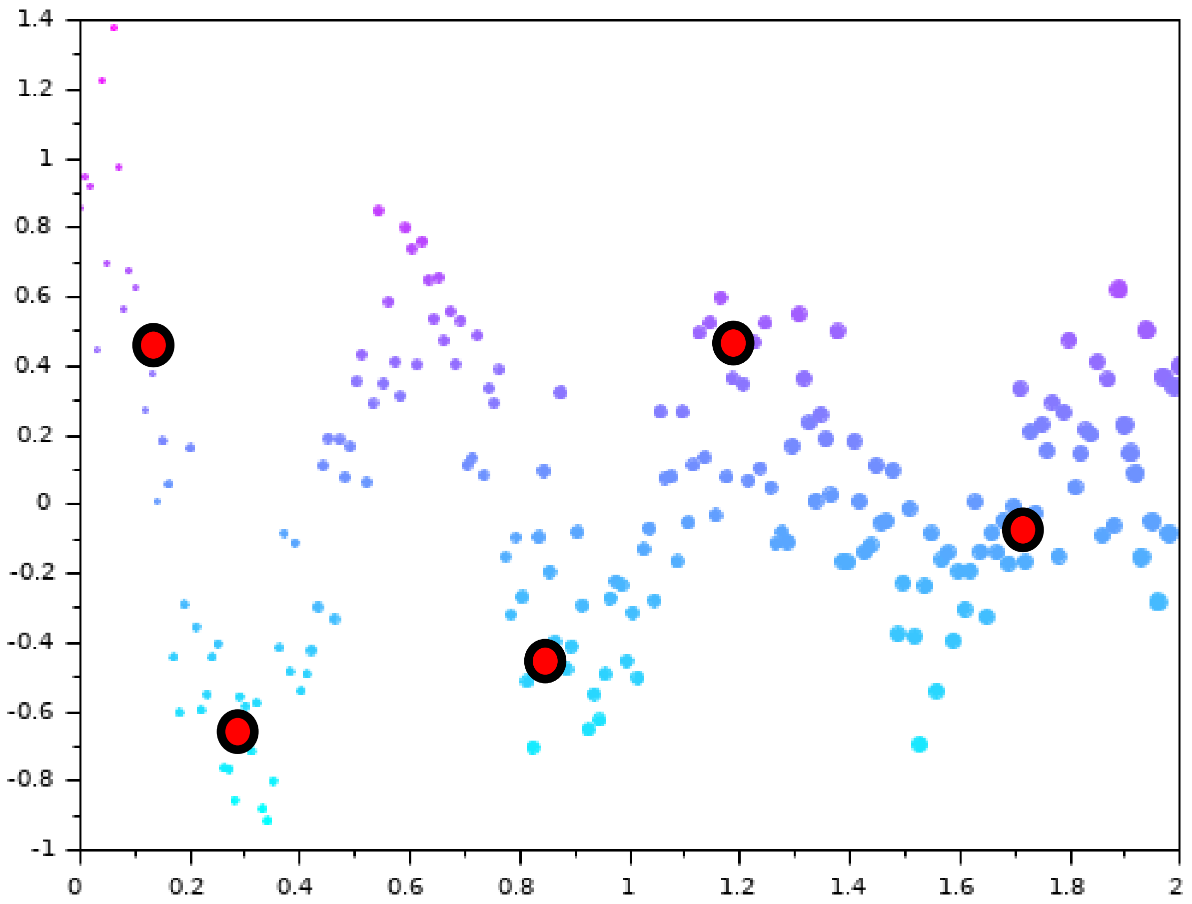
Today: Basics, Next week: Advanced

1. Goal, motivation, Basic methods
   1. Parametric methods
   2. Autoregressive methods
   3. Latent space mapping
2. Variational Auto Encoder (VAE)
3. Generative Adversarial Networks (GAN)
   1. Introduction (basic setup, intuition)
   2. Evaluation
   3. Image to image (pix2pix, ~~CycleGAN~~)
   4. Problems and how to improve GAN performance (losses, tricks etc.)
   5. StyleGAN
   6. Extras (GAN Dissection, Single Image)

Goal

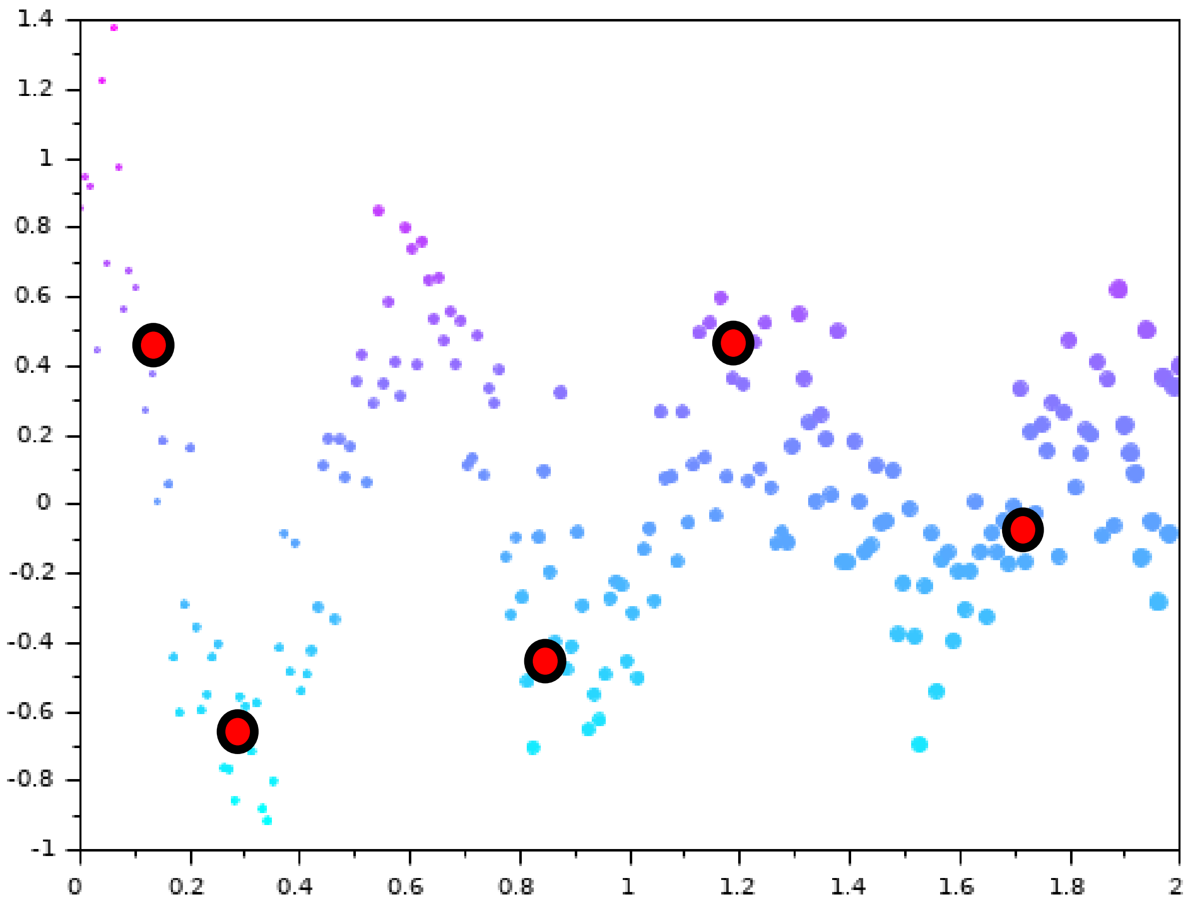


Goal



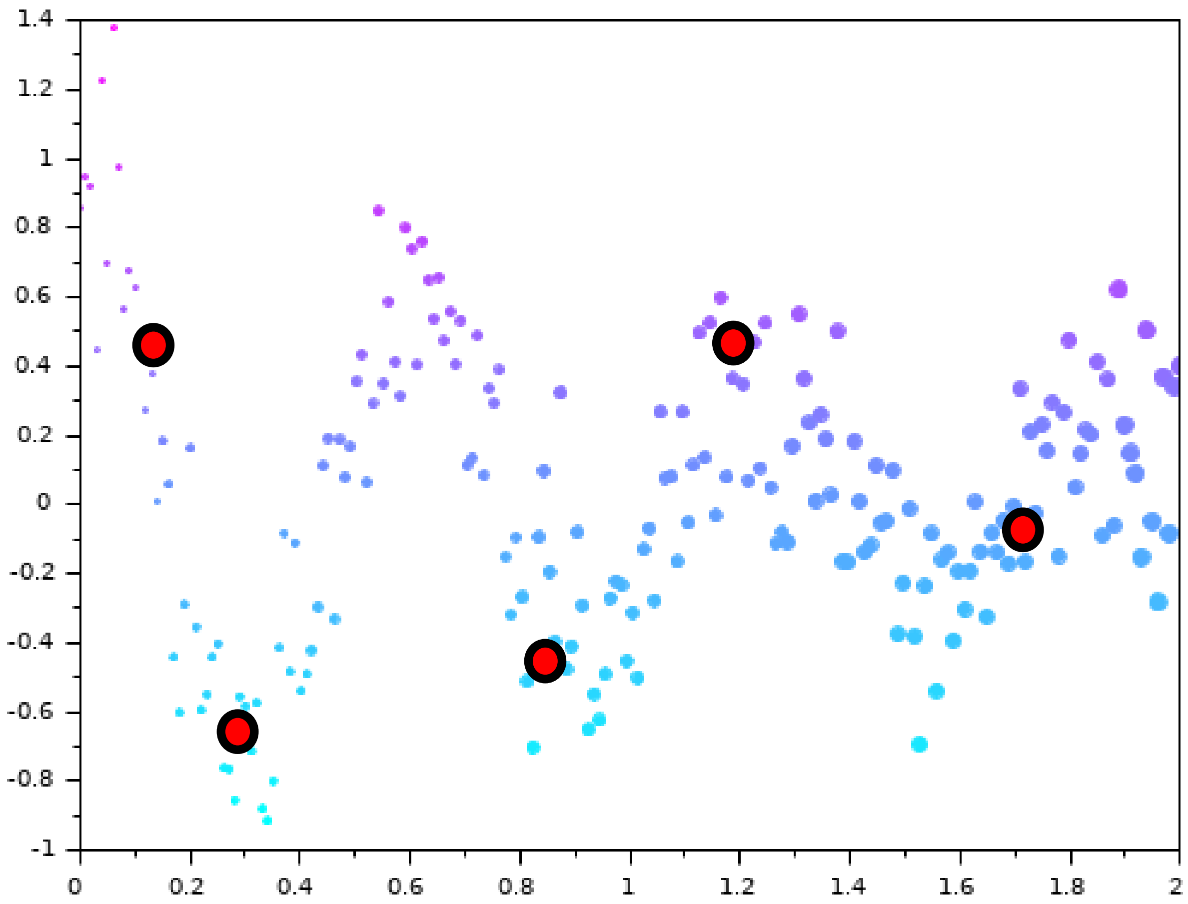
Goal

Images



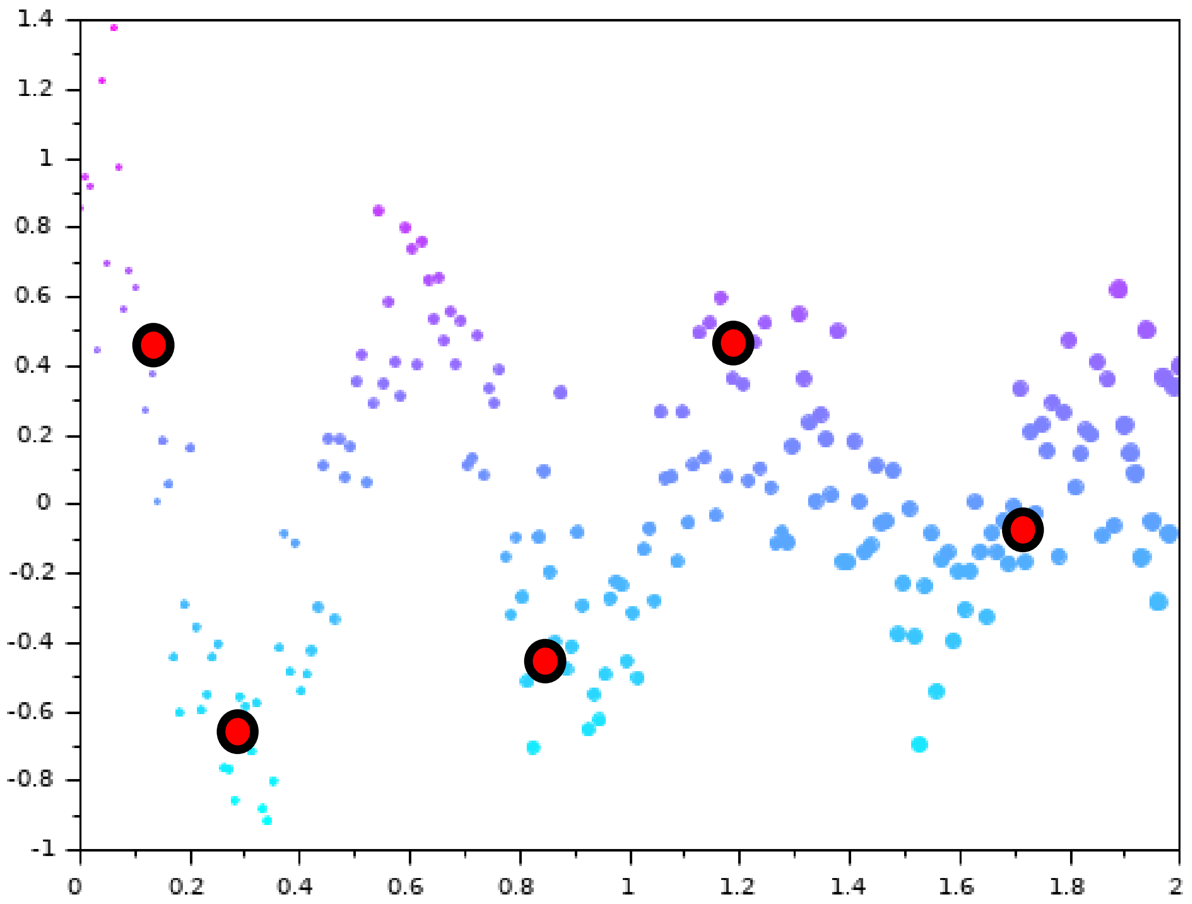
Goal

Images
Text



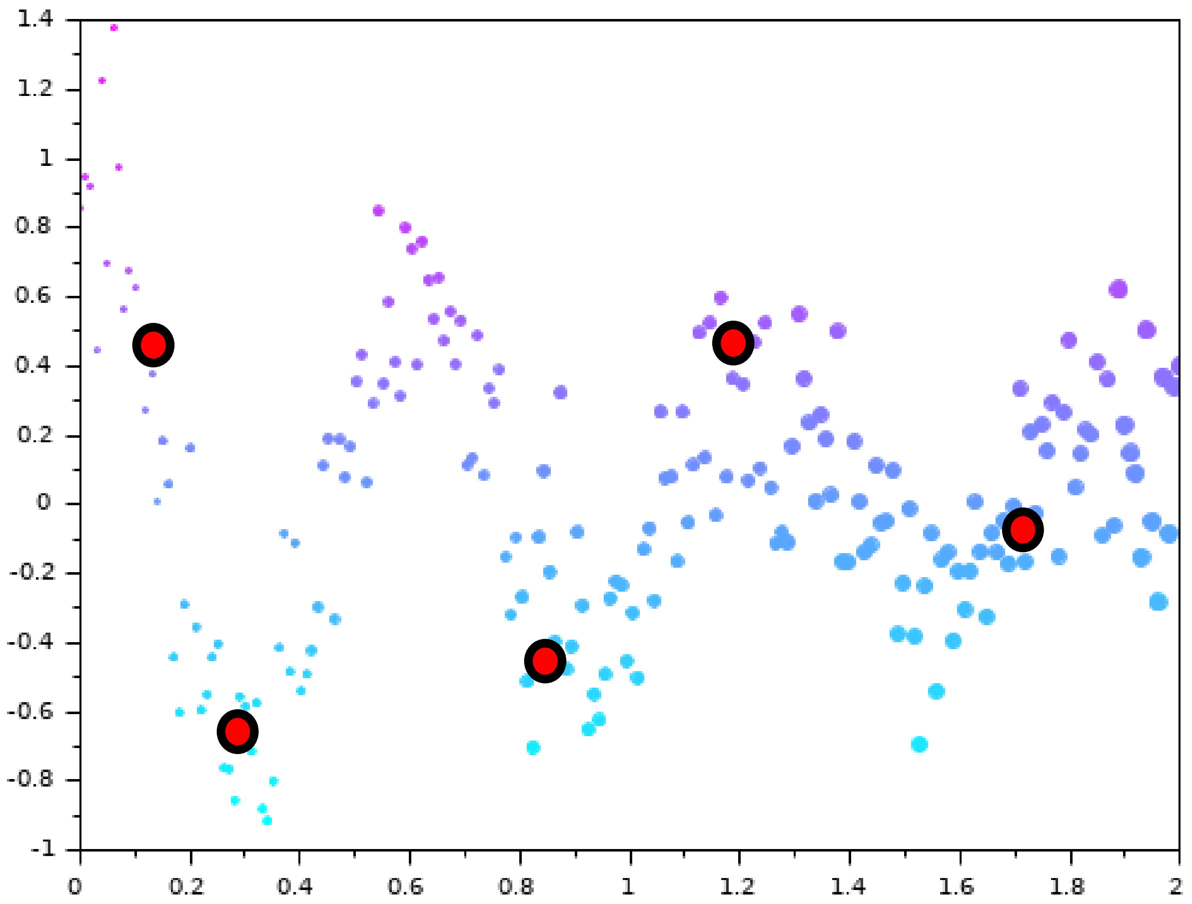
Goal

Images
Text
Audio



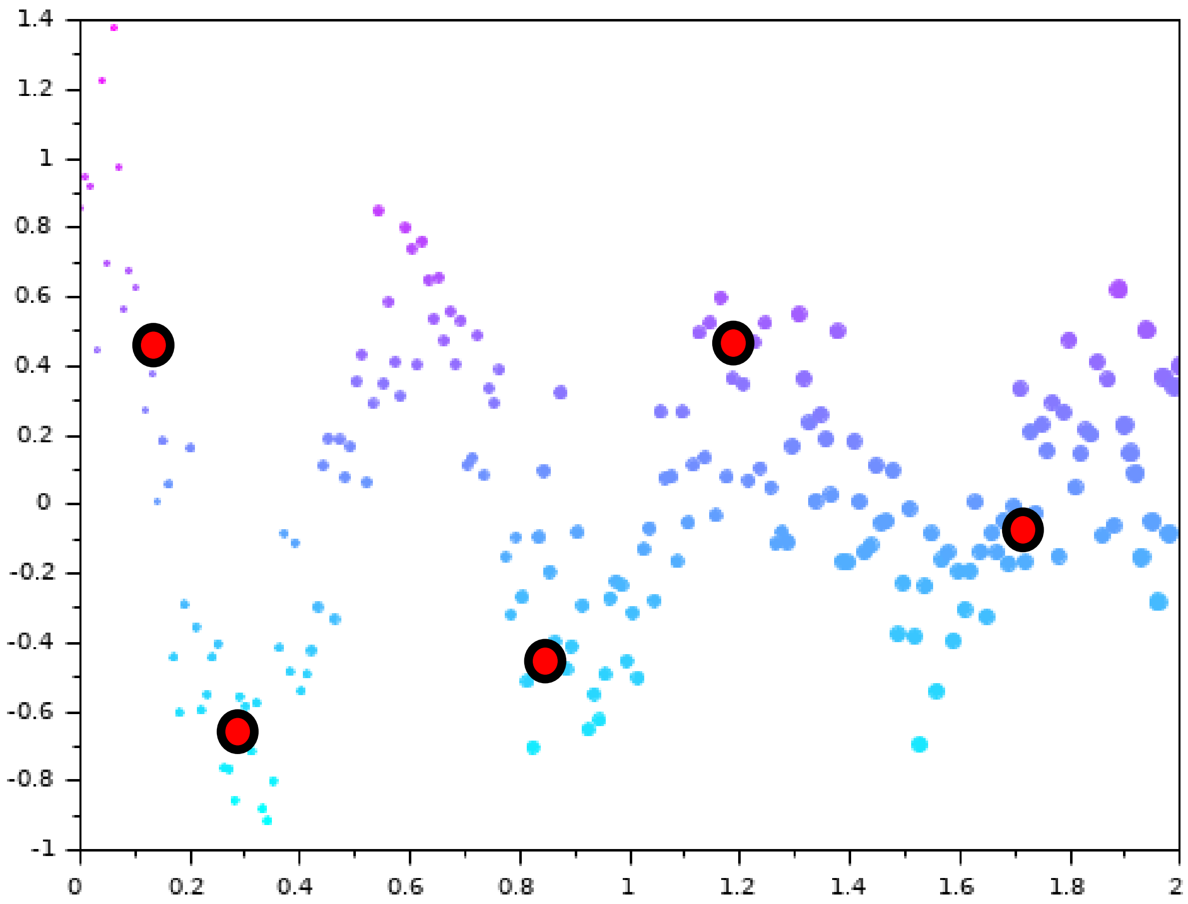
Goal

Images
Text
Audio
Video



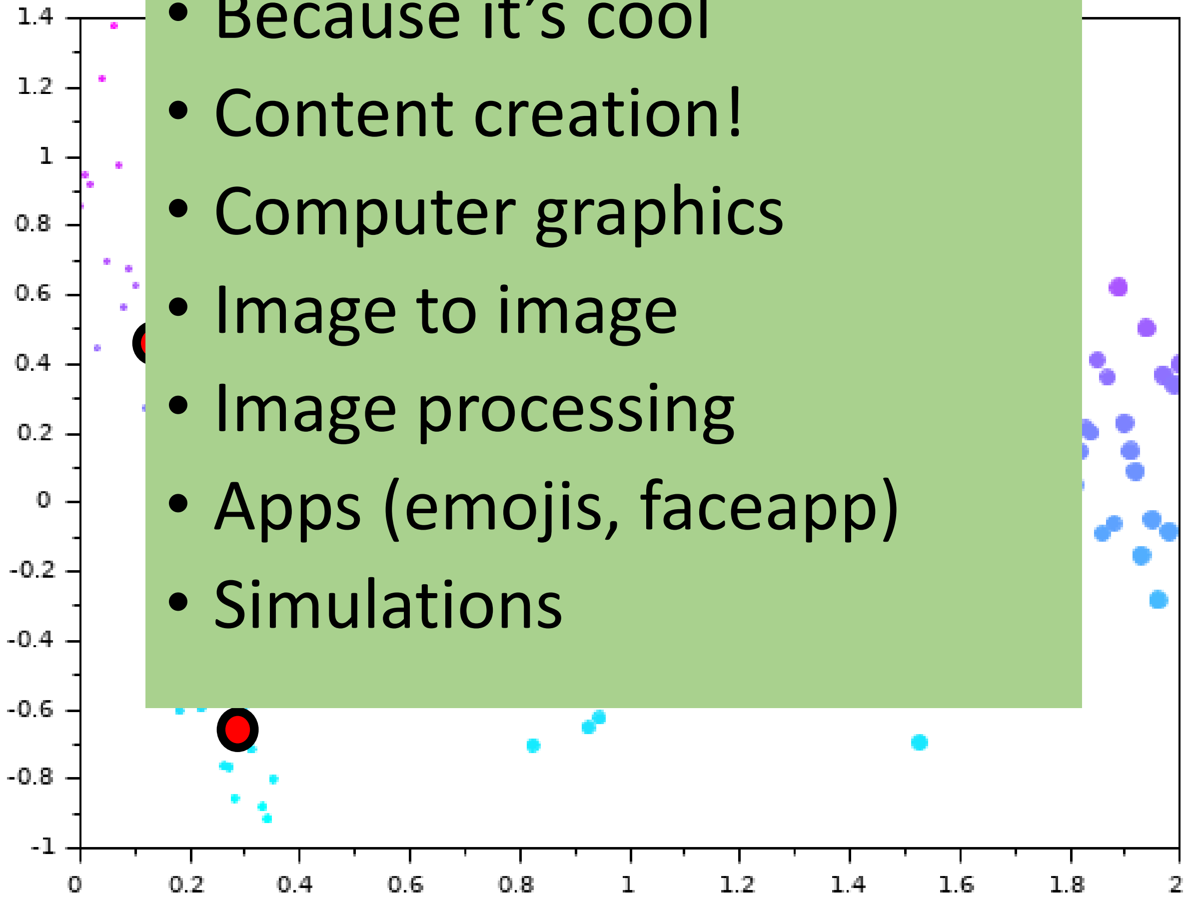
Goal

Images
Text
Audio
Video
Whatever…



Goal

Images
Text
Audio
Video
Whatever…

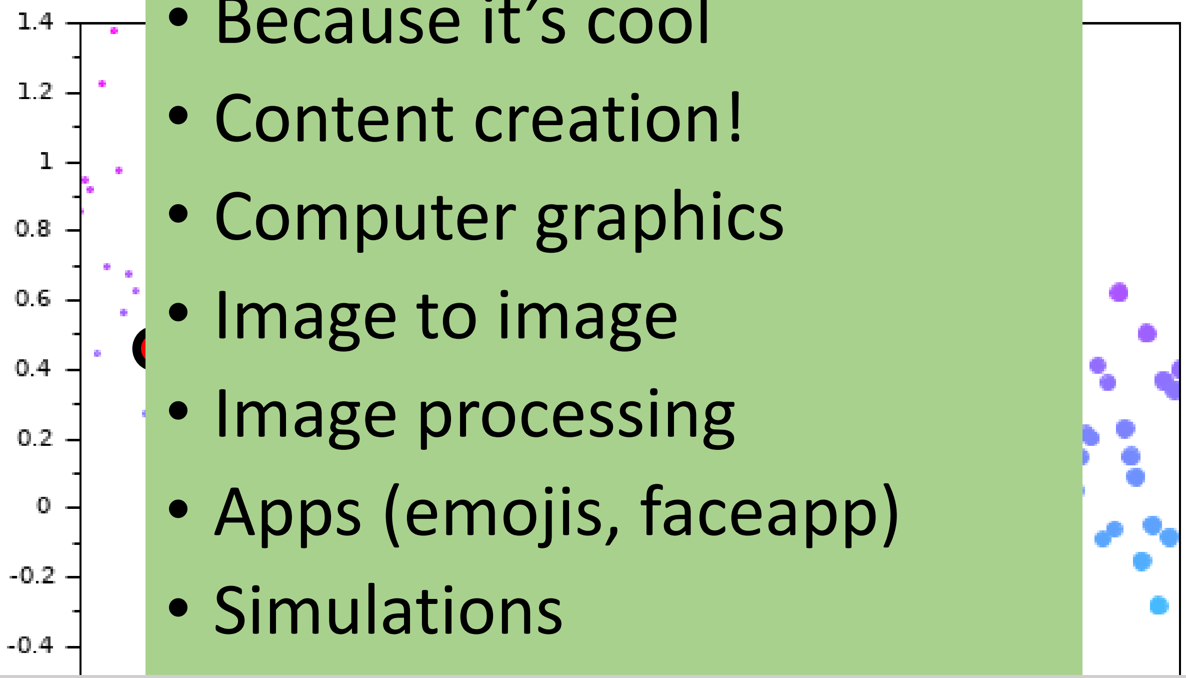


Why?

- Because it’s cool
- Content creation!
- Computer graphics
- Image to image
- Image processing
- Apps (emojis, faceapp)
- Simulations

Goal

Images
Text
Audio
Video
Whatever…



Why?

- Because it’s cool
- Content creation!
- Computer graphics
- Image to image
- Image processing
- Apps (emojis, faceapp)
- Simulations

The correct approach when there is more than one valid solution!

Prompt: A bunch of students trying to figure out what generative models are.

Generative methods:

- Parametric distribution estimation (e.g. GMM)
- Autoregressive models (e.g. RNN, Causal CNN, Diffusion)
- Latent space mapping (e.g. VAE, GAN)

Generative methods:

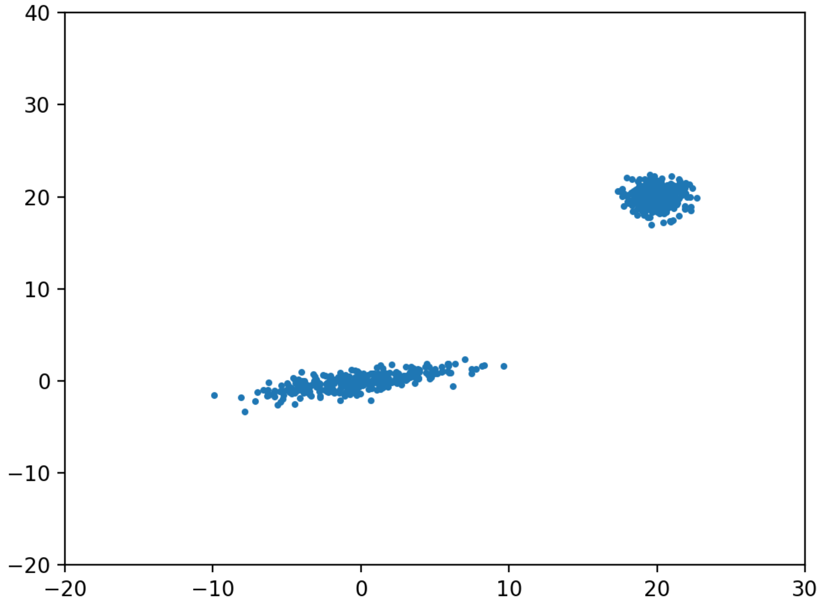
- Parametric distribution estimation (e.g. GMM)
- Autoregressive models (e.g. RNN, Causal CNN, Diffusion)
- Latent space mapping (e.g. VAE, GAN)

Parametric Distribution Estimation Example: GMM

Elements in the slide taken from NVIDIA CVPR'18 GANs tutorial

Parametric Distribution Estimation
Example: GMM

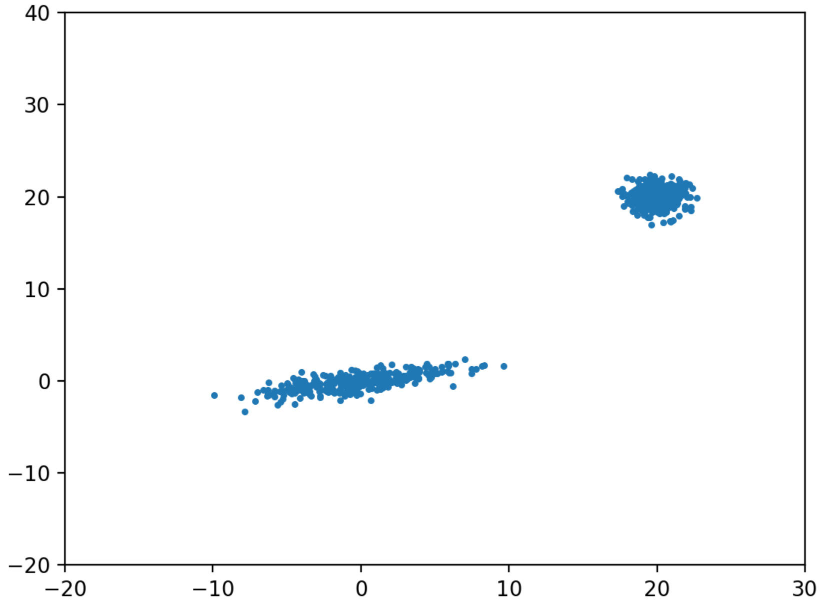
Step 1: observe a set of samples



Elements in the slide taken from NVIDIA CVPR’18 GANs tutorial

Parametric Distribution Estimation
Example: GMM

Step 1: observe a set of samples



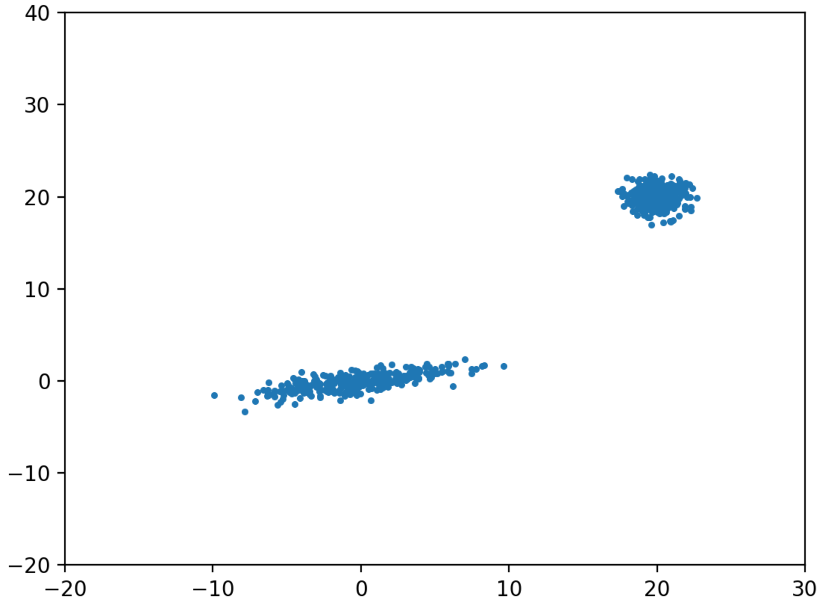
Step 2: assume a GMM model

$$p(x|\theta) = \sum_i \pi_i \mathcal{N}(x|\mu_i, \Sigma_i)$$

Elements in the slide taken from NVIDIA CVPR'18 GANs tutorial

Parametric Distribution Estimation Example: GMM

Step 1: observe a set of samples



Step 2: assume a GMM model

$$p(x|\theta) = \sum_i \pi_i \mathcal{N}(x|\mu_i, \Sigma_i)$$

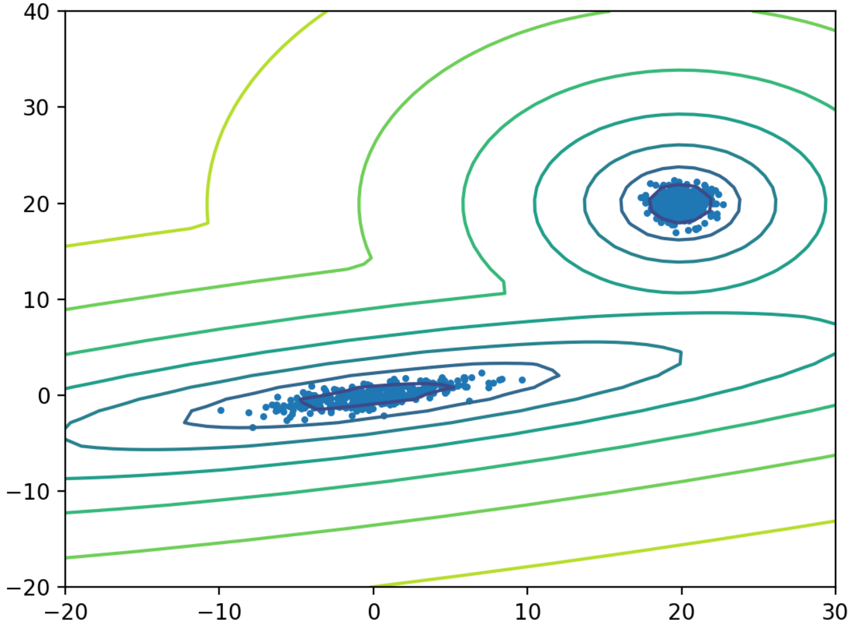
Step 3: perform maximum likelihood learning

$$\max_\theta \sum_{x^{(j)} \in \text{Dataset}} \log p(\theta|x^{(j)})$$

Elements in the slide taken from NVIDIA CVPR'18 GANs tutorial

Parametric Distribution Estimation Example: GMM

Step 1: observe a set of samples



Step 2: assume a GMM model

$$p(x|\theta) = \sum_i \pi_i \mathcal{N}(x|\mu_i, \Sigma_i)$$

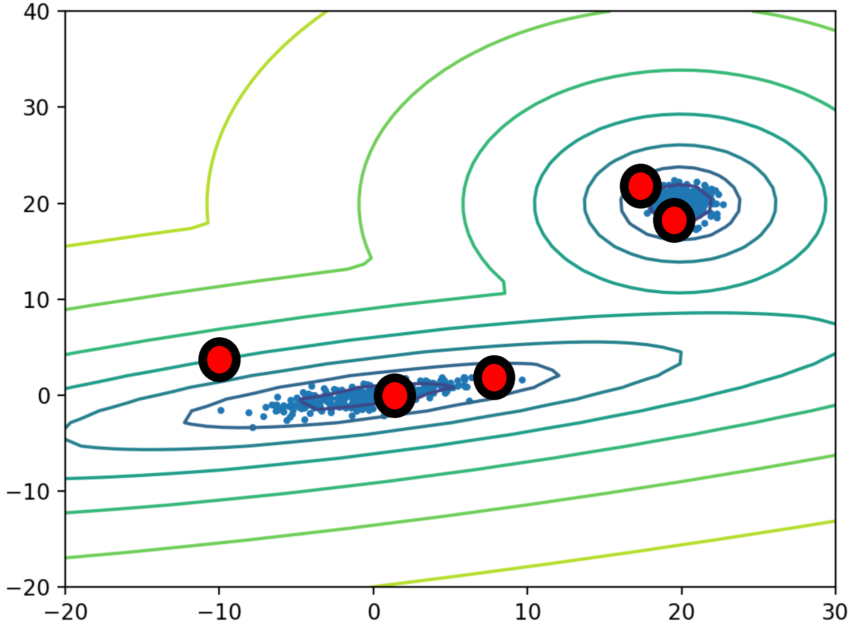
Step 3: perform maximum likelihood learning

$$\max_\theta \sum_{x^{(j)} \in \text{Dataset}} \log p(\theta|x^{(j)})$$

Elements in the slide taken from NVIDIA CVPR'18 GANs tutorial

Parametric Distribution Estimation Example: GMM

Step 1: observe a set of samples



Step 2: assume a GMM model

$$p(x|\theta) = \sum_i \pi_i \mathcal{N}(x|\mu_i, \Sigma_i)$$

Step 3: perform maximum likelihood learning

$$\max_\theta \sum_{x^{(j)} \in \text{Dataset}} \log p(\theta|x^{(j)})$$

Step 4: Sample

Elements in the slide taken from NVIDIA CVPR'18 GANs tutorial

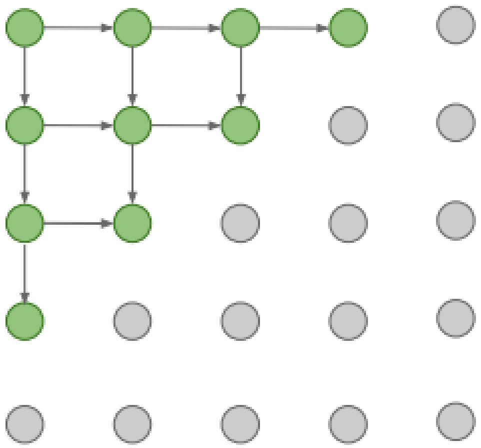
Autoregressive image generation - Basic

Autoregressive image generation - Basic

Van der Oord 2016

Autoregressive image generation - Basic

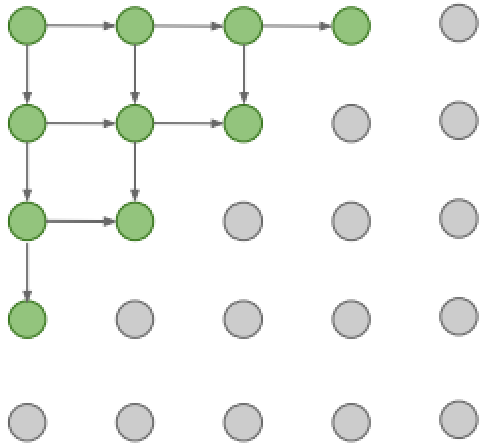
PixelRNN



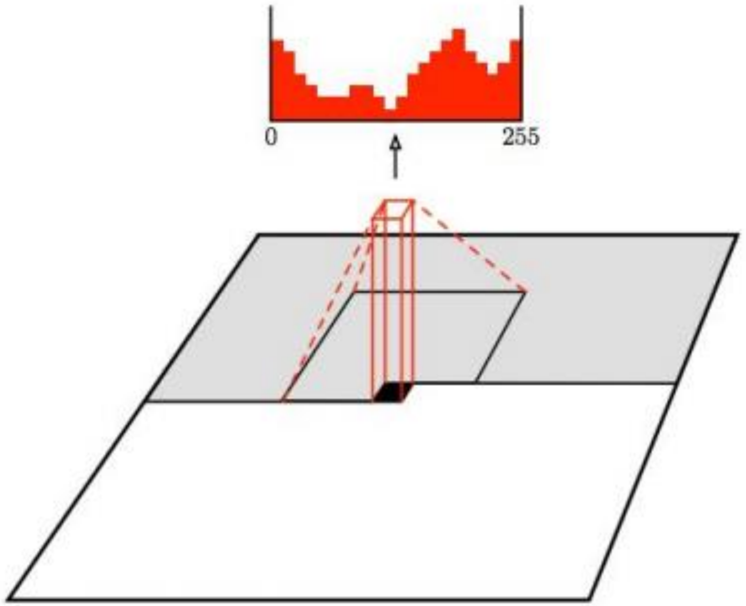
Van der Oord 2016

Autoregressive image generation - Basic

PixelRNN



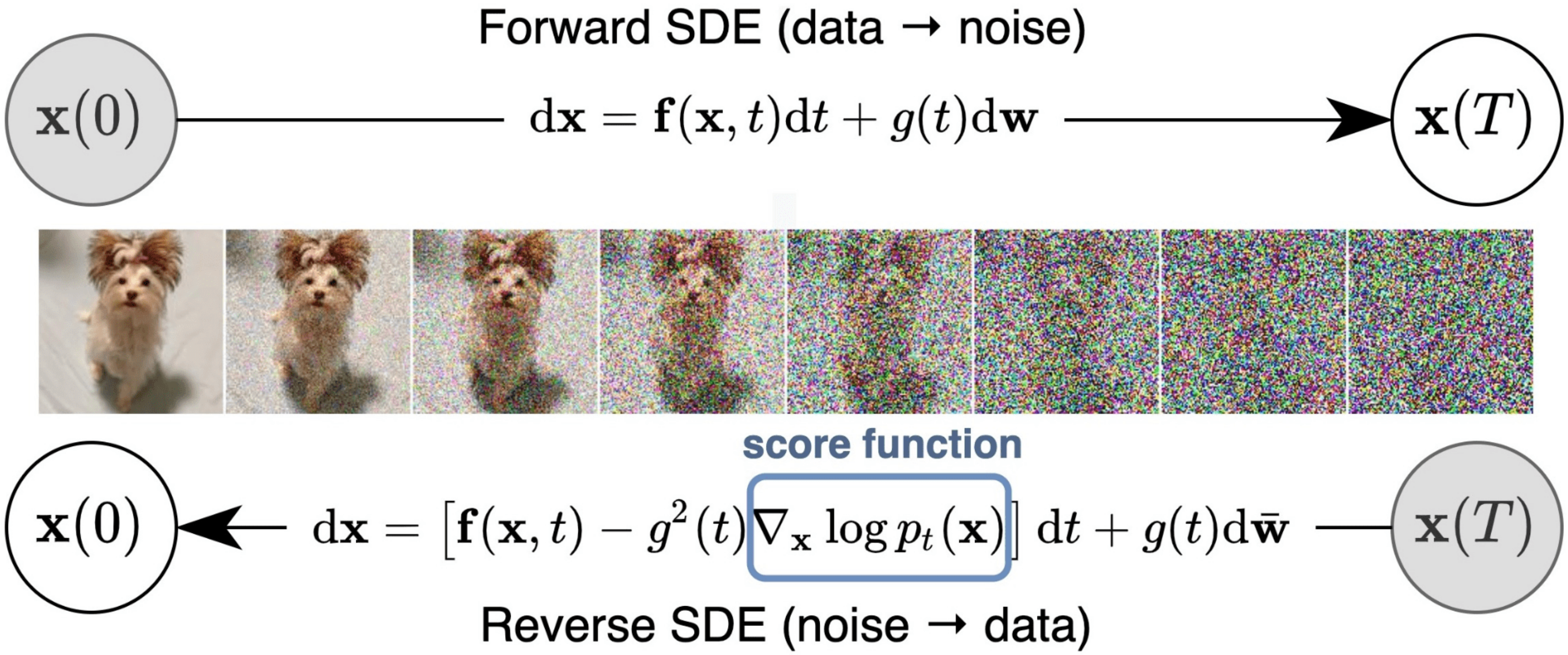
PixelCNN



Van der Oord 2016

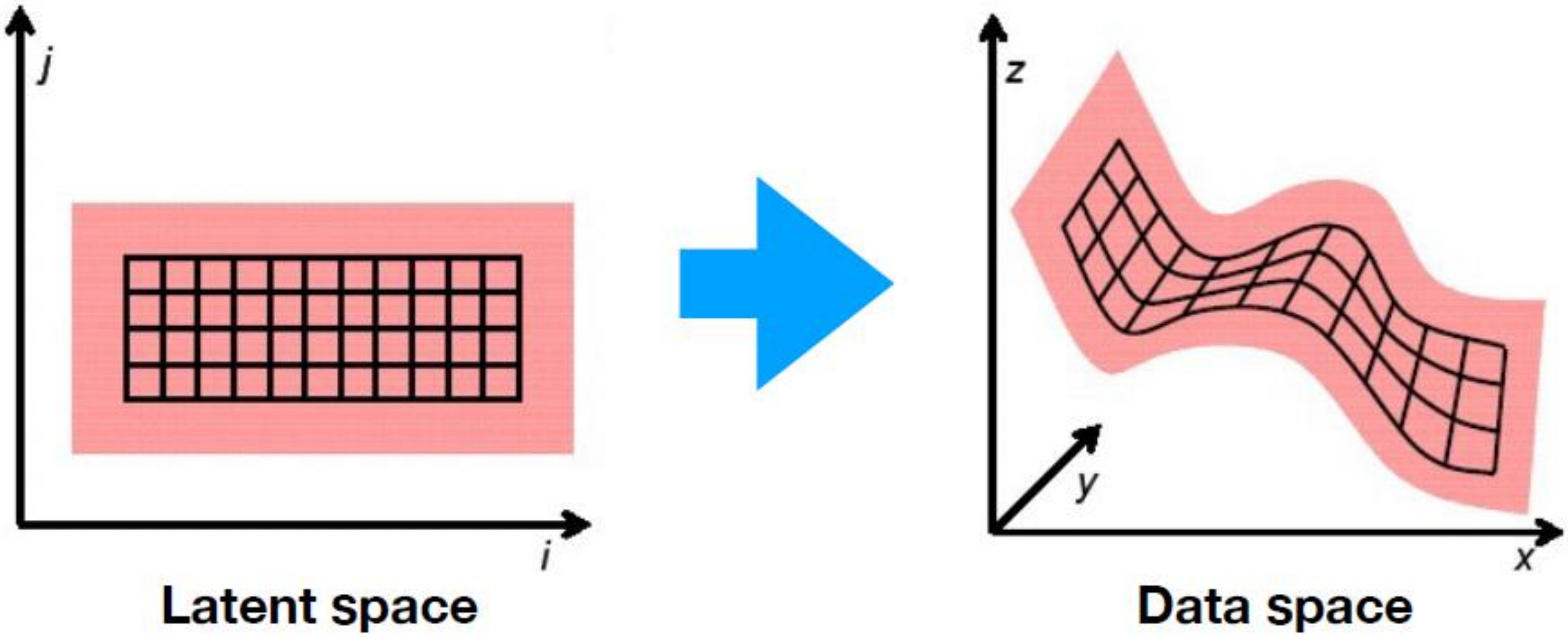
Autoregressive image generation - Recent

Diffusion models:

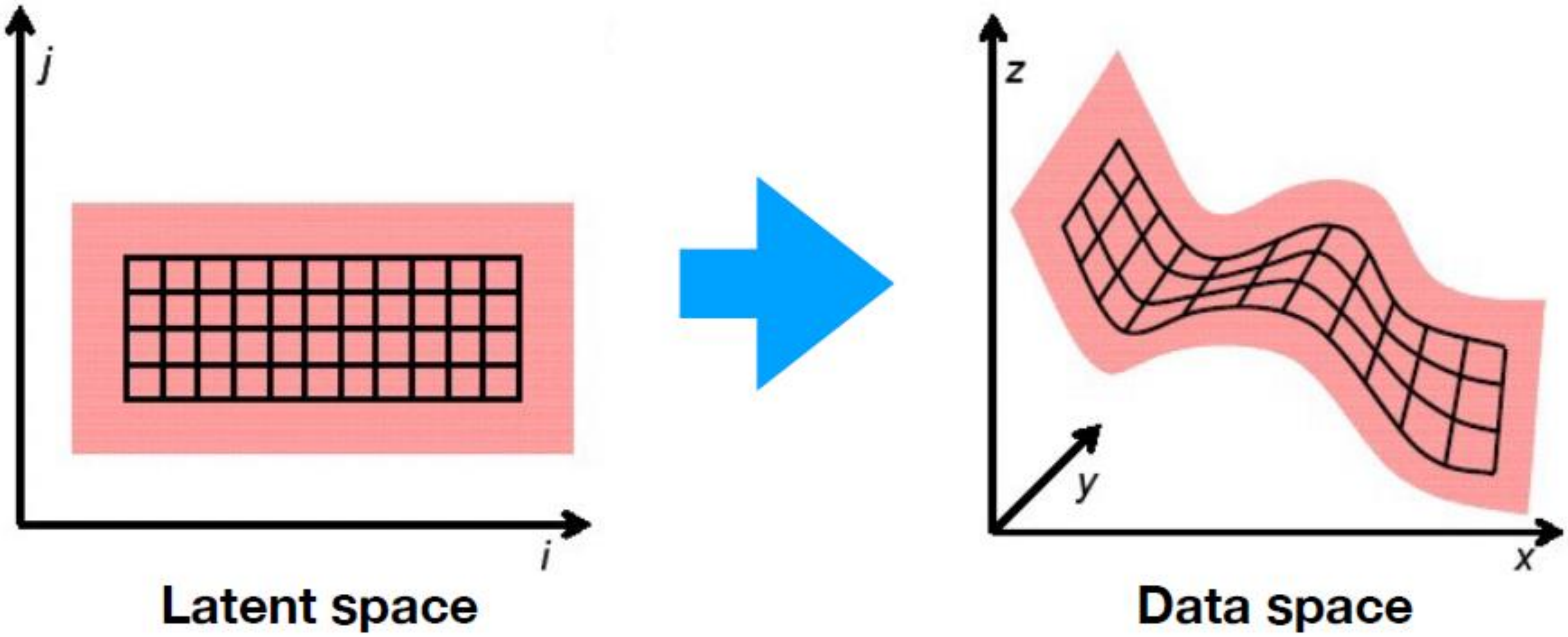


Wait for the advanced generative models class!

Latent space mapping approach

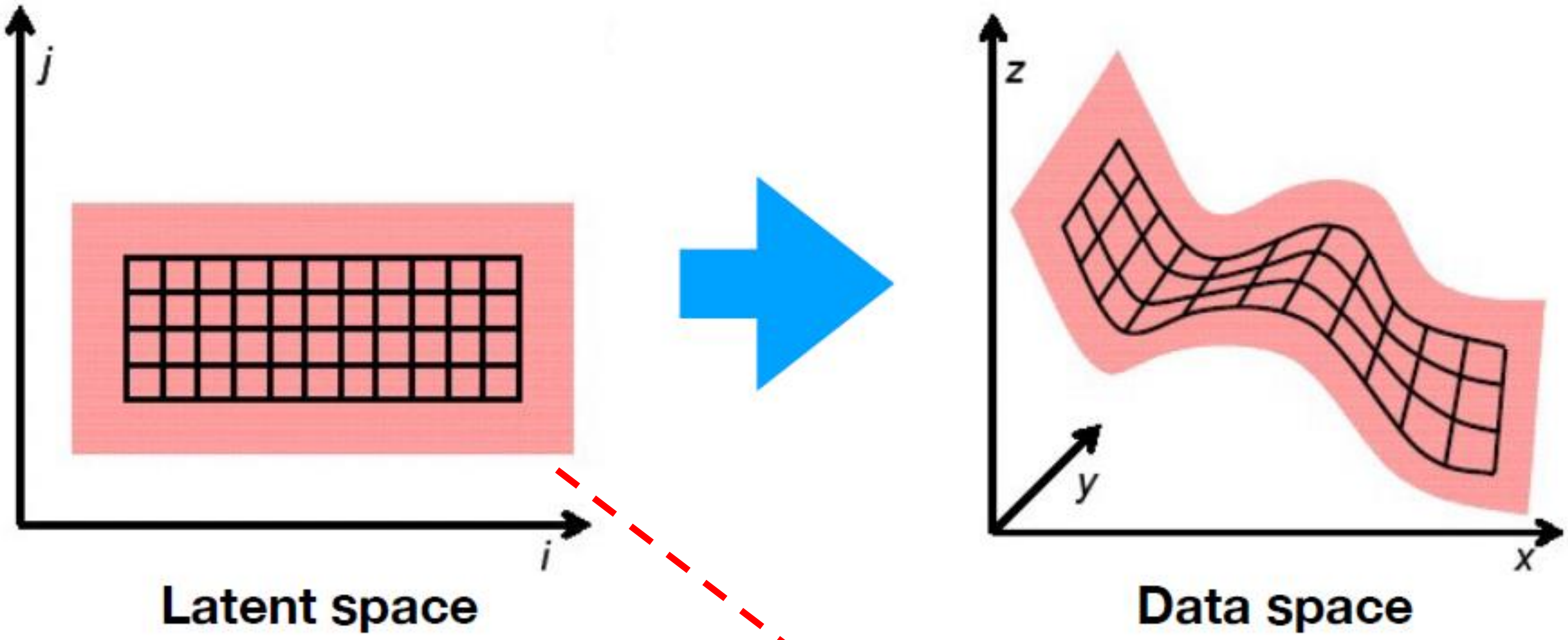


Latent space mapping approach



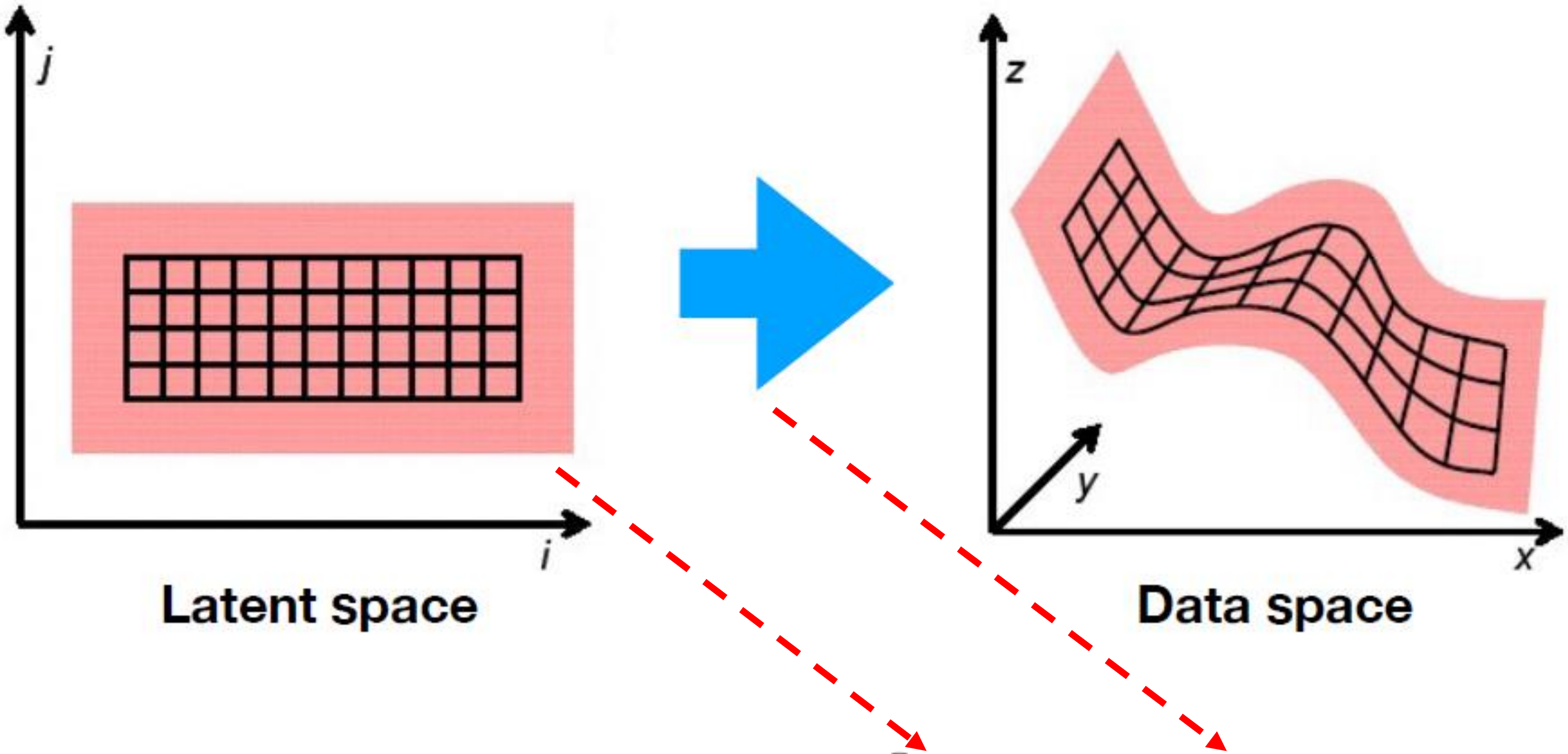
Data likelihood: $p_\theta(x) = \int p_\theta(z) p_\theta(x|z) dz$

Latent space mapping approach



Data likelihood: $p_\theta(x) = \int p_\theta(z) p_\theta(x|z) dz$

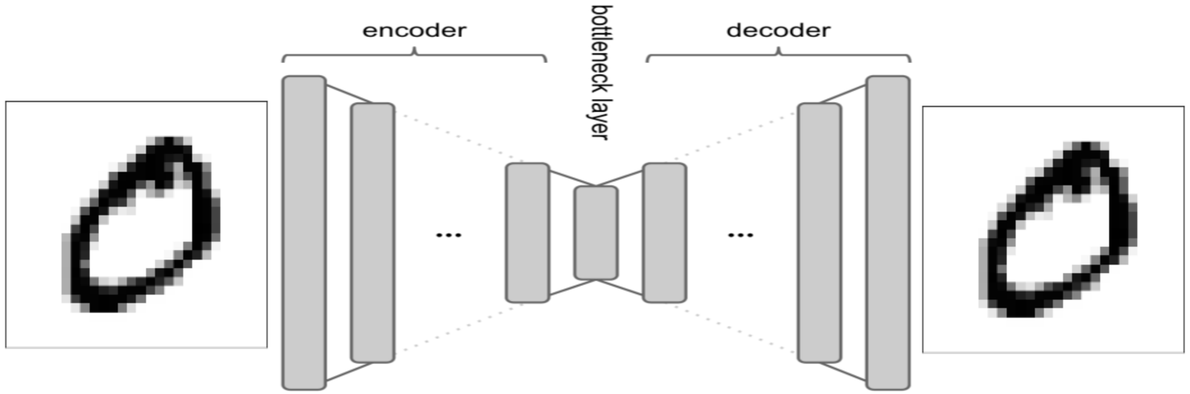
Latent space mapping approach



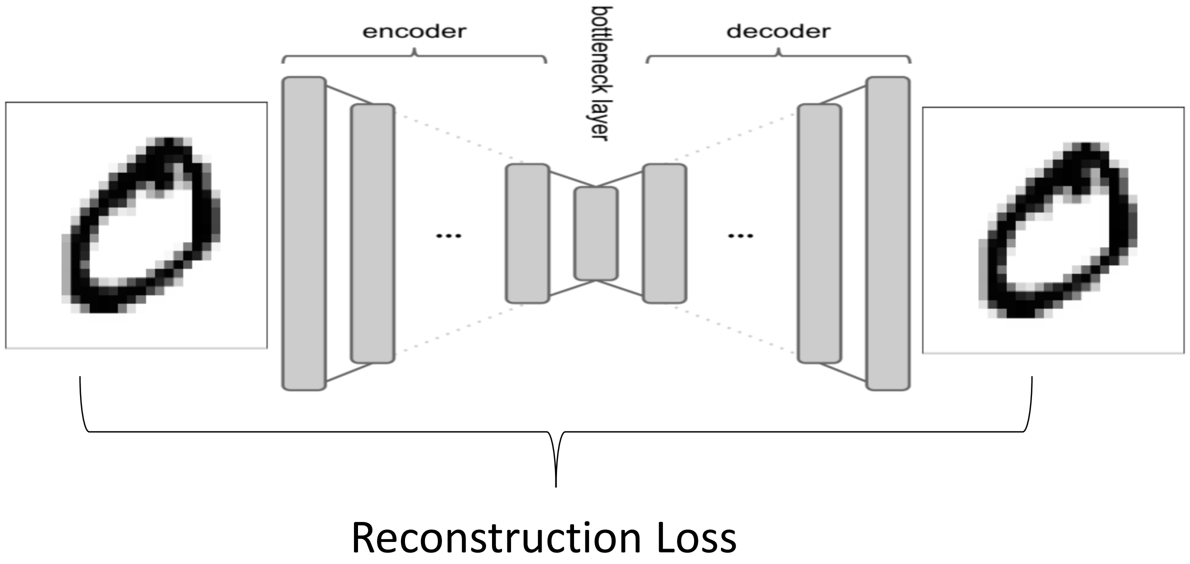
Data likelihood: $p_\theta(x) = \int p_\theta(z) p_\theta(x|z) dz$

Autoencoders

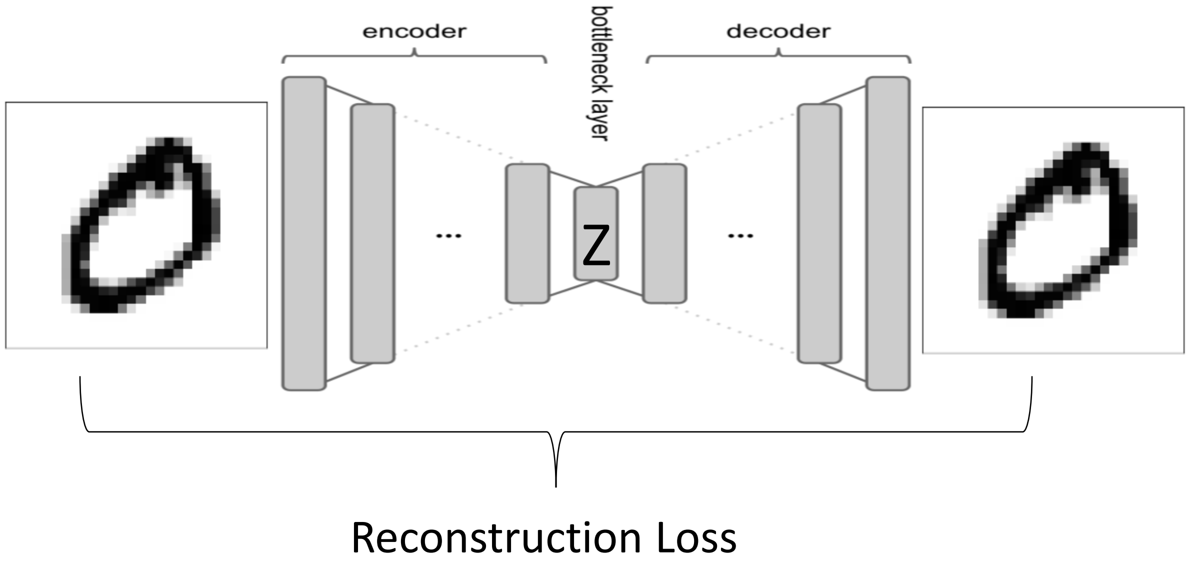
Autoencoders



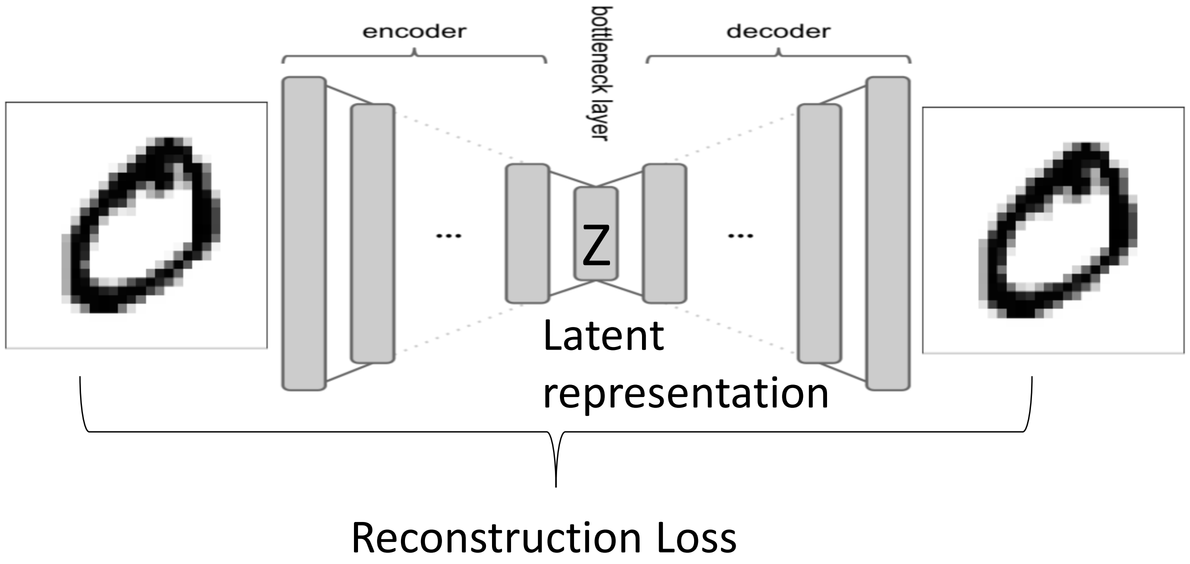
Autoencoders



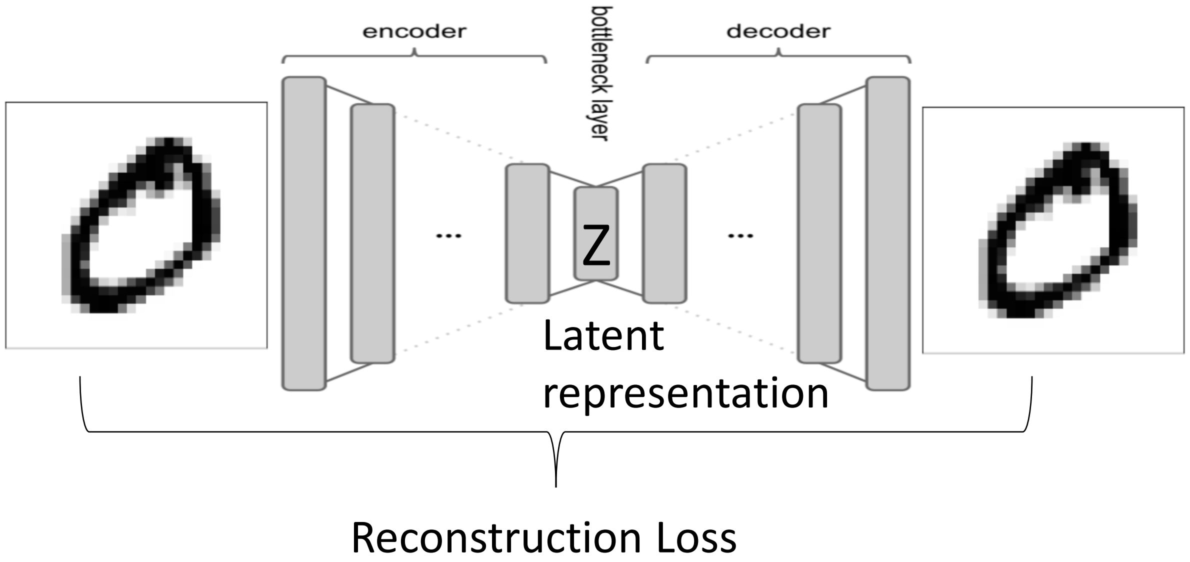
Autoencoders



Autoencoders

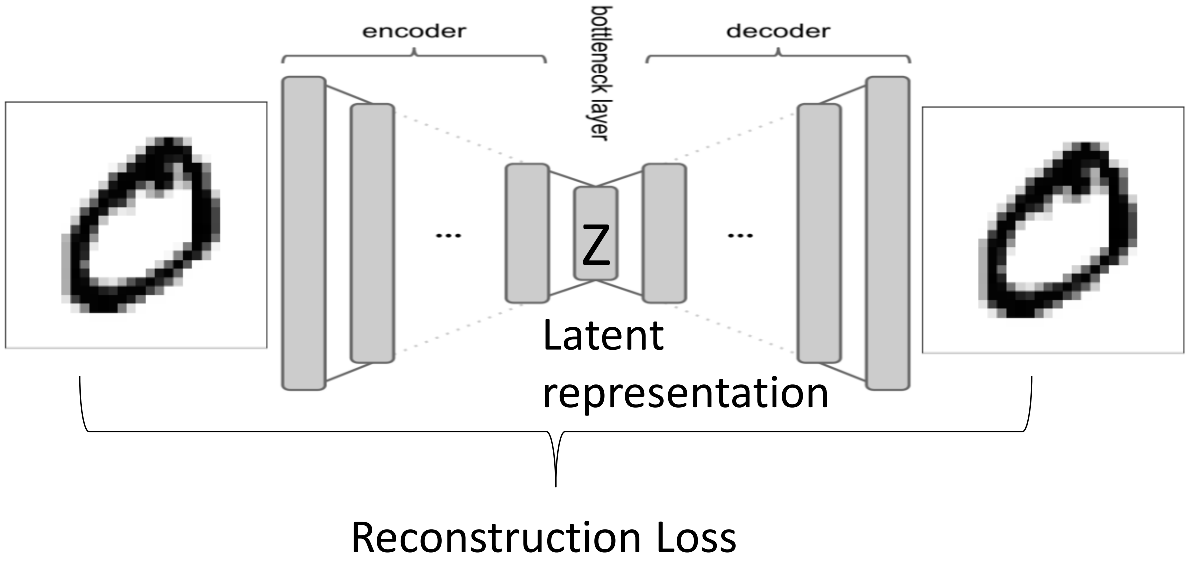


Autoencoders

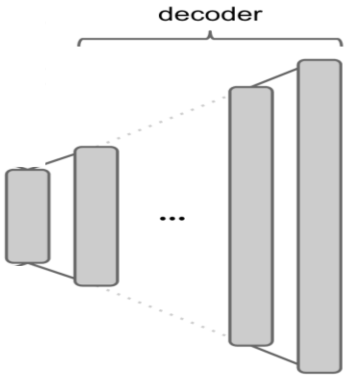


- Generative model?

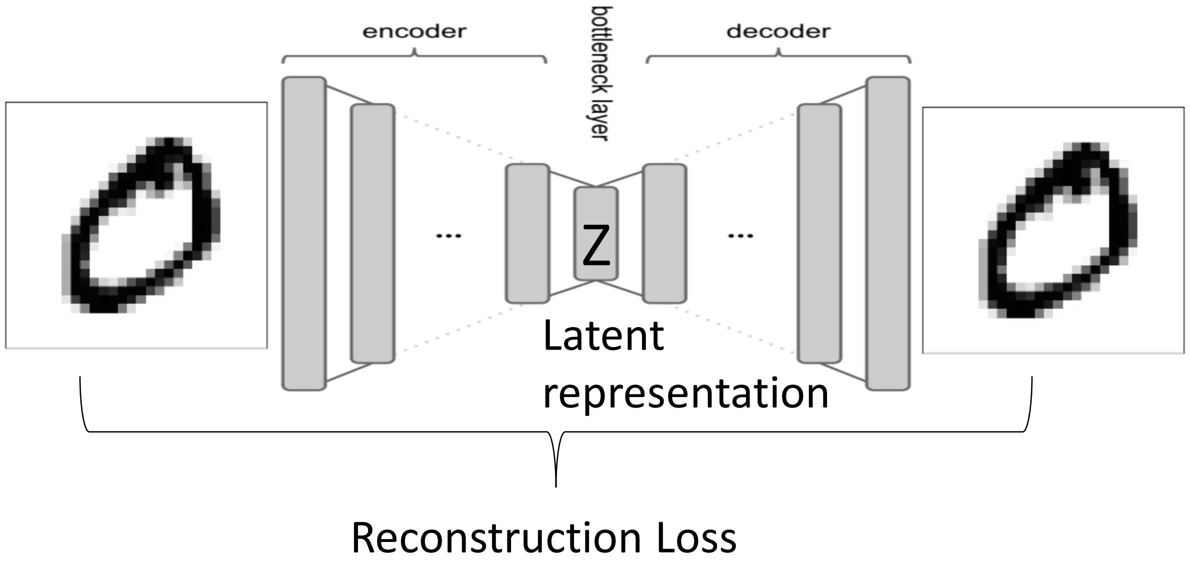
Autoencoders



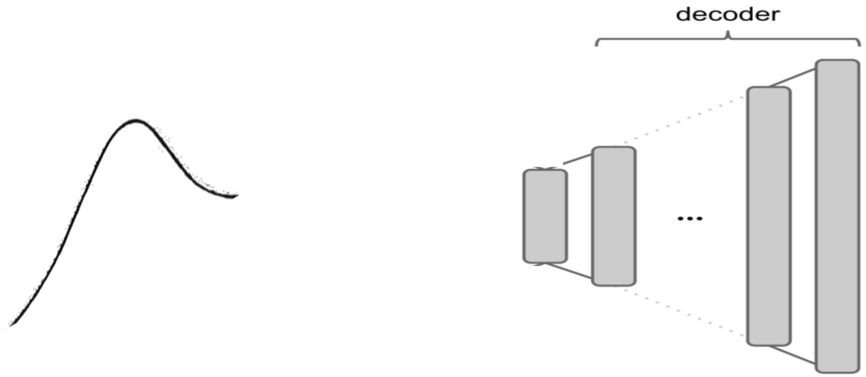
- Generative model?



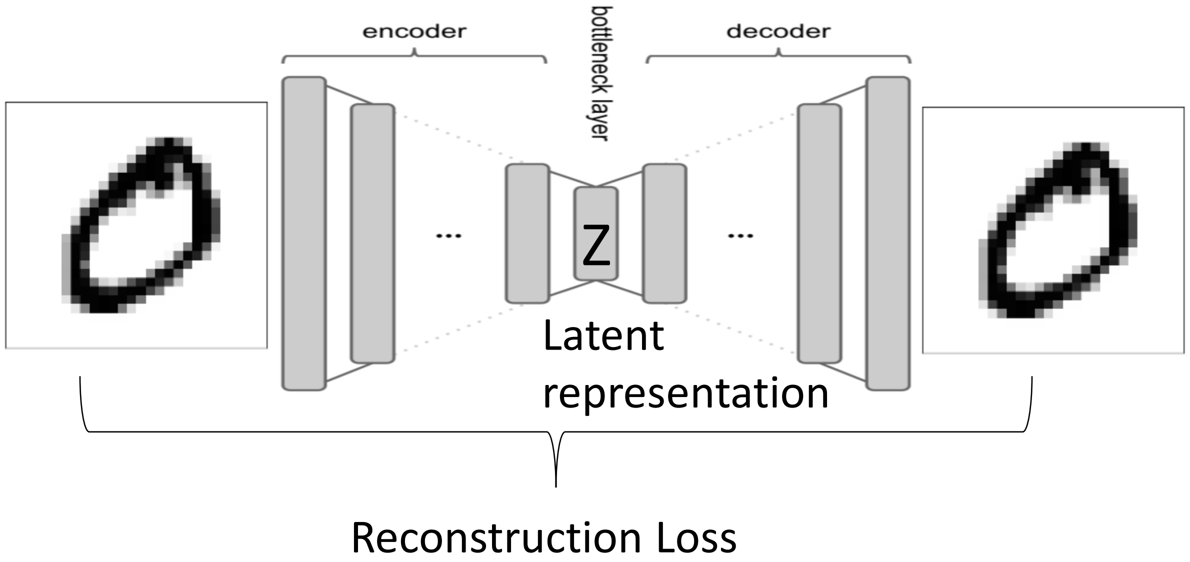
Autoencoders



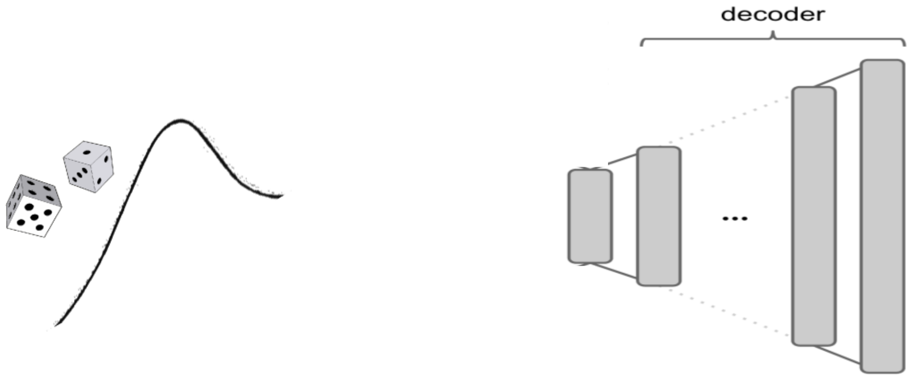
- Generative model?



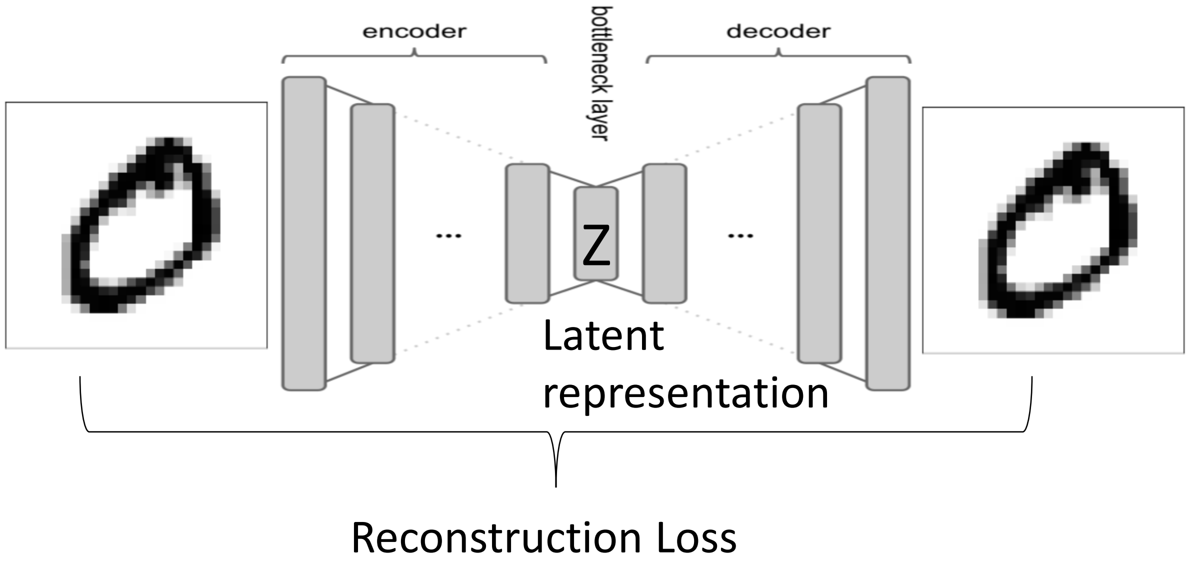
Autoencoders



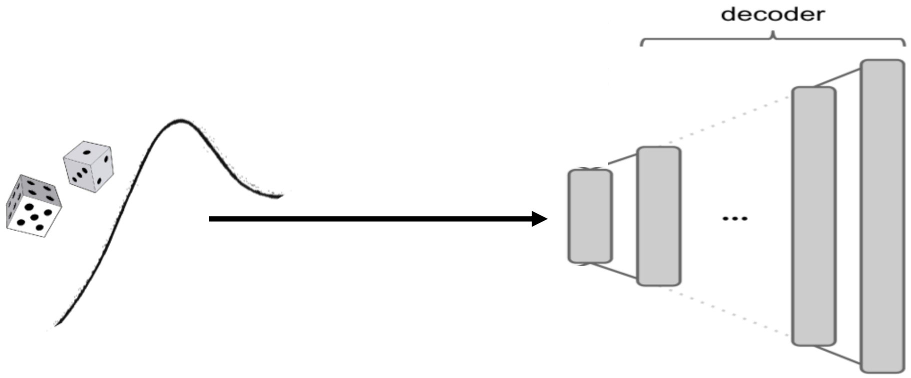
- Generative model?



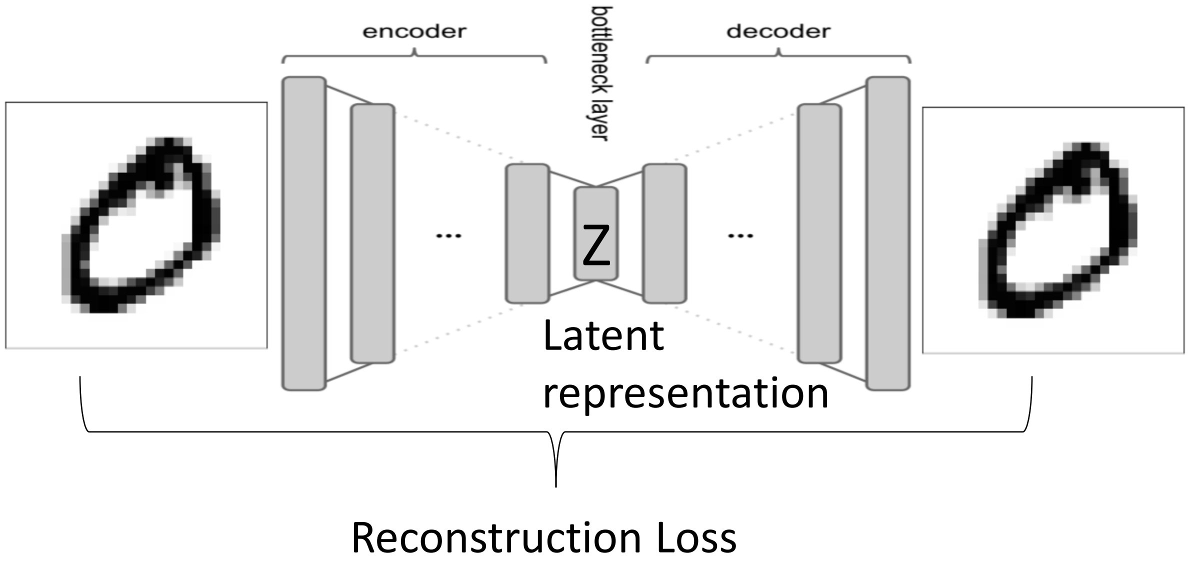
Autoencoders



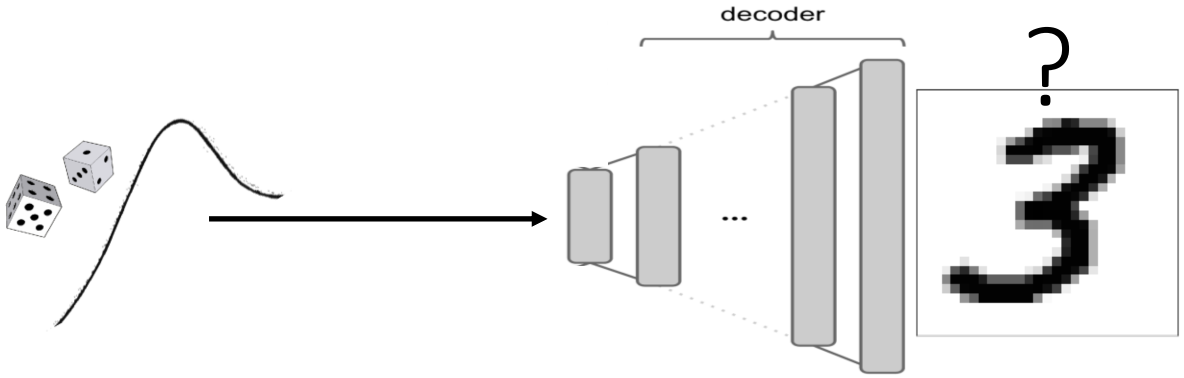
- Generative model?



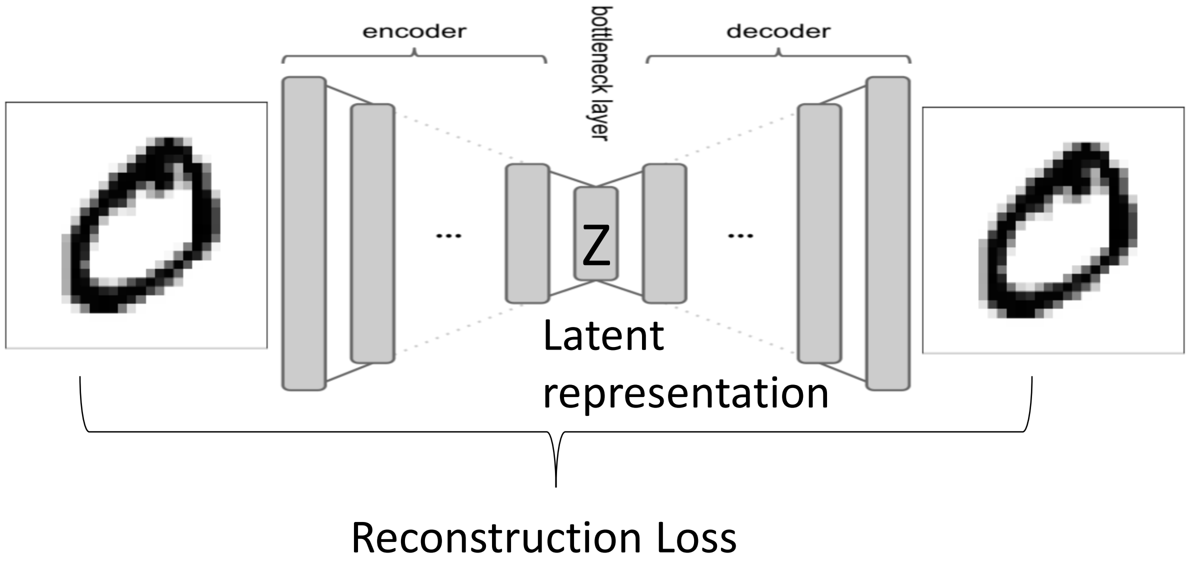
Autoencoders



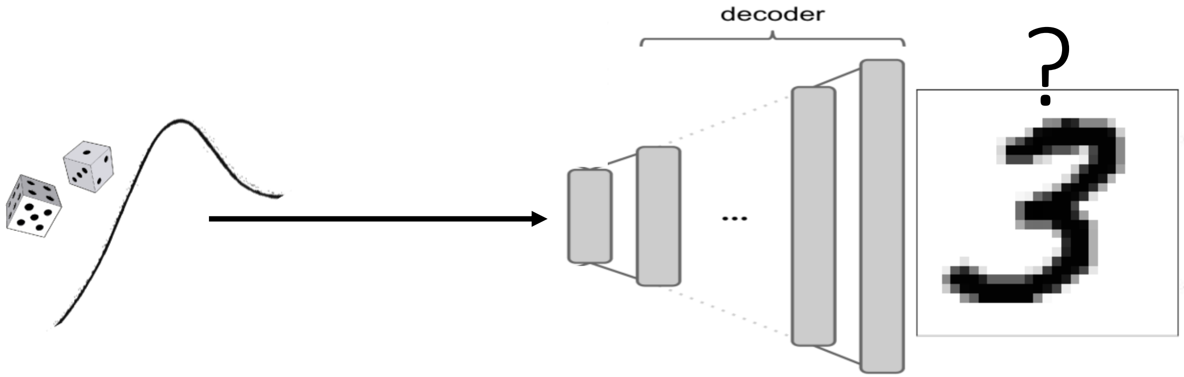
- Generative model?

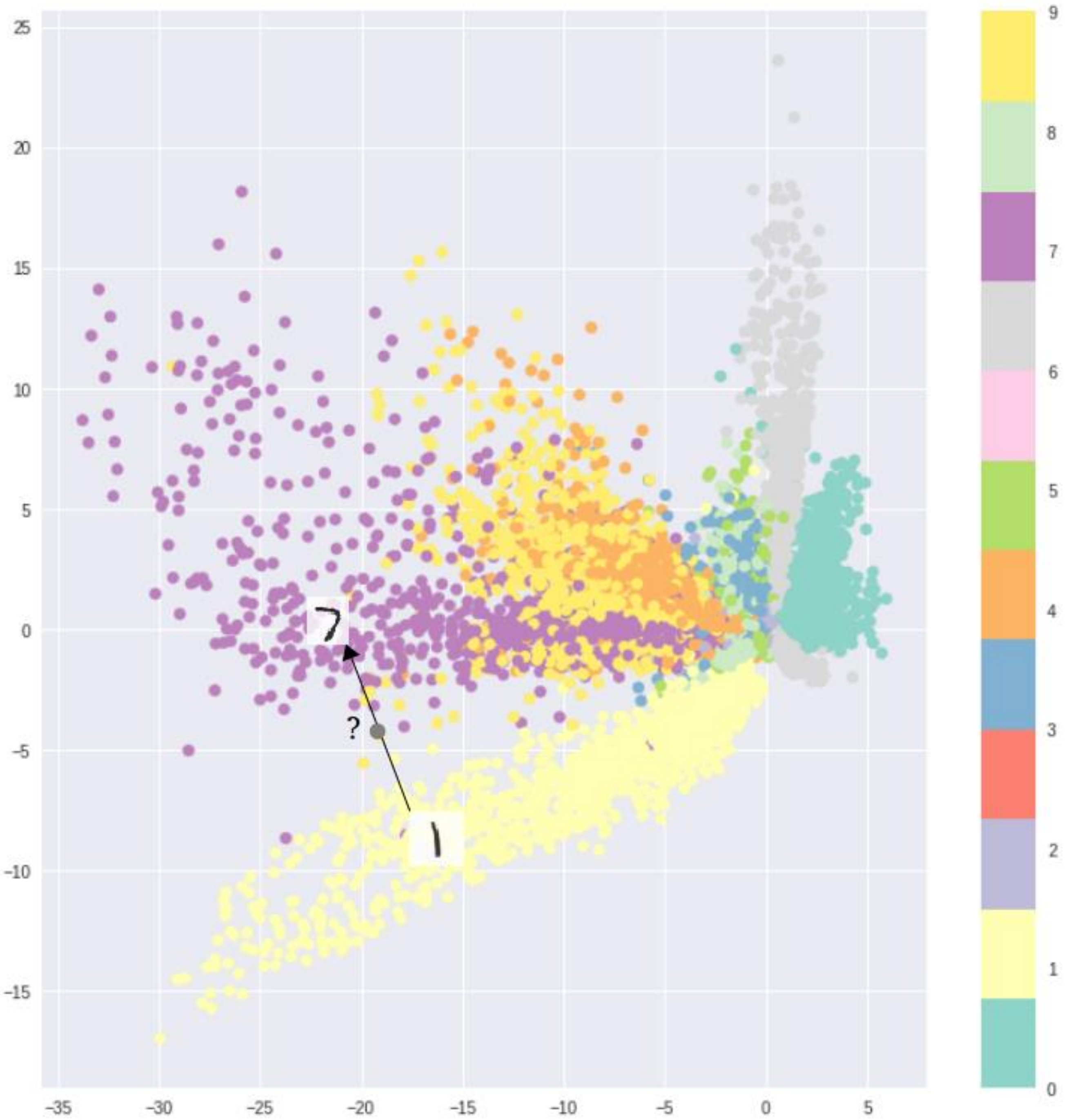


Autoencoders

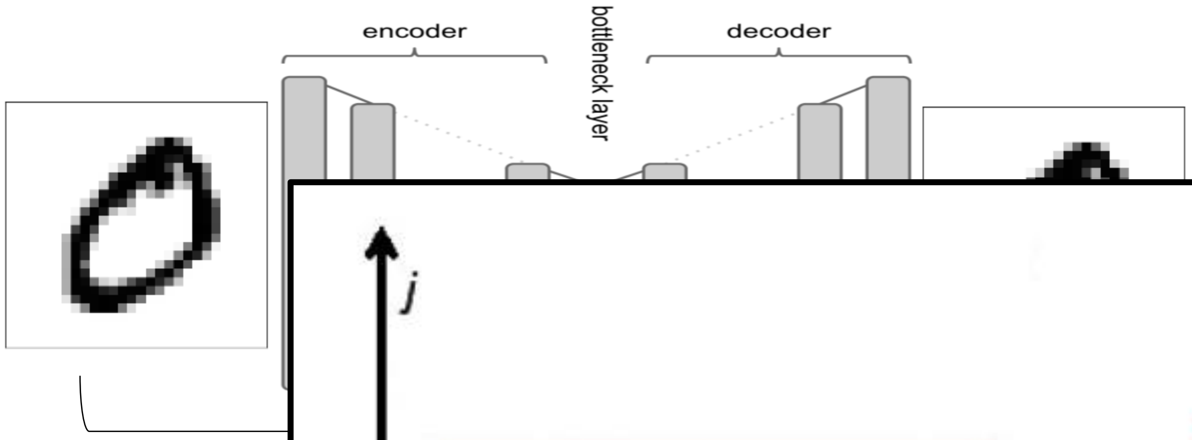


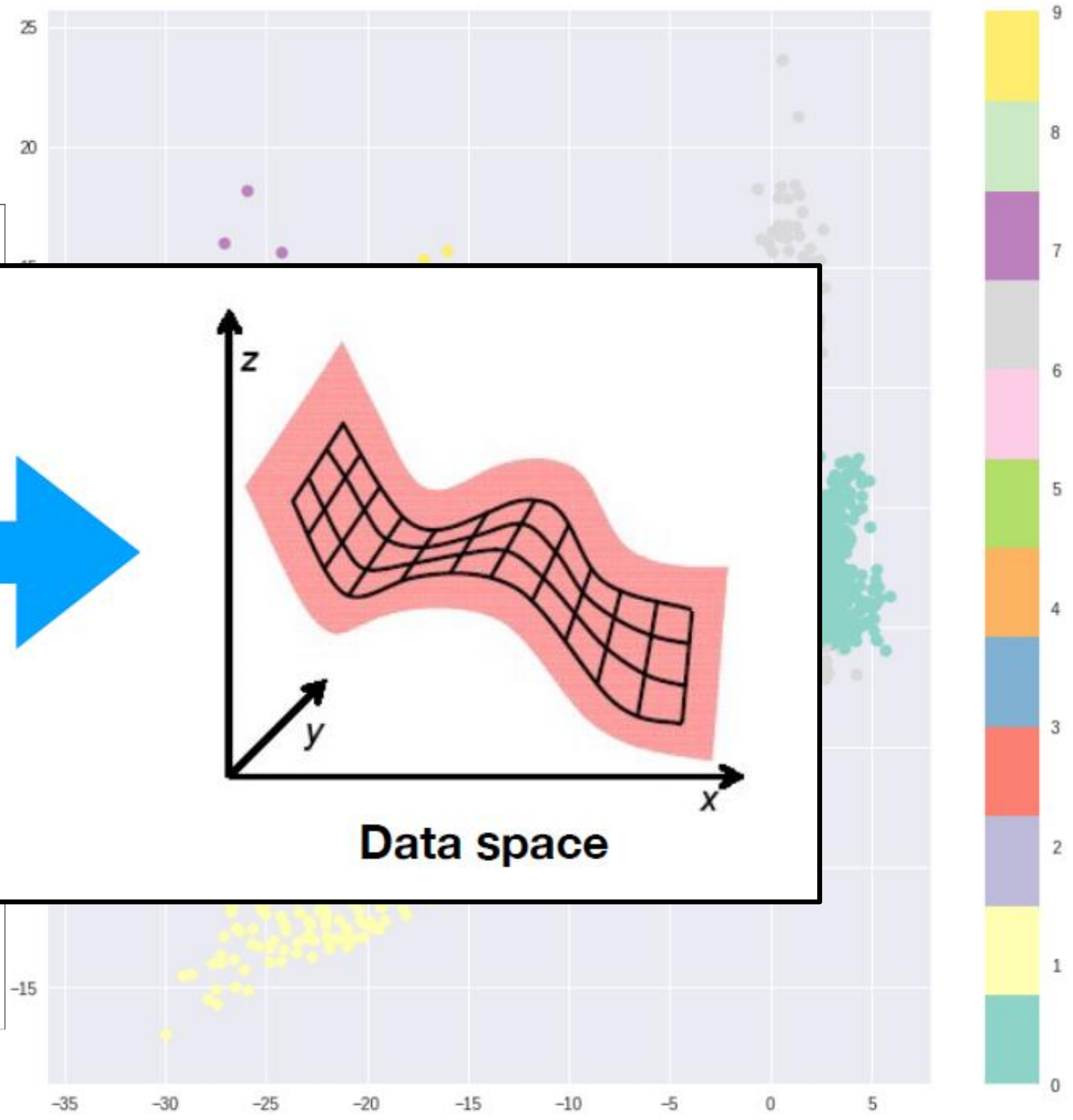
- Generative model?



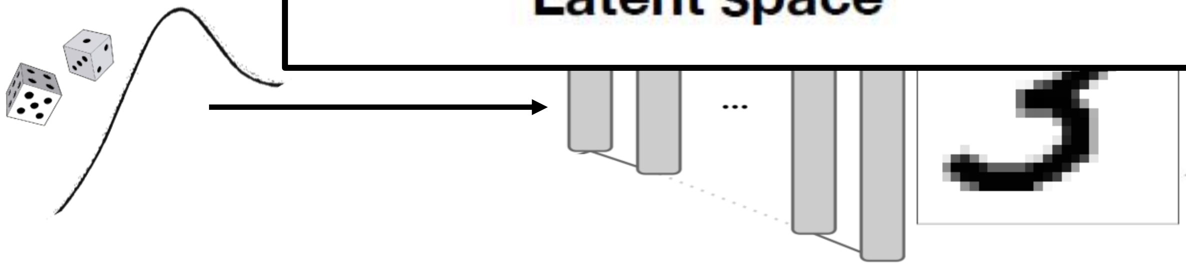


Autoencoders

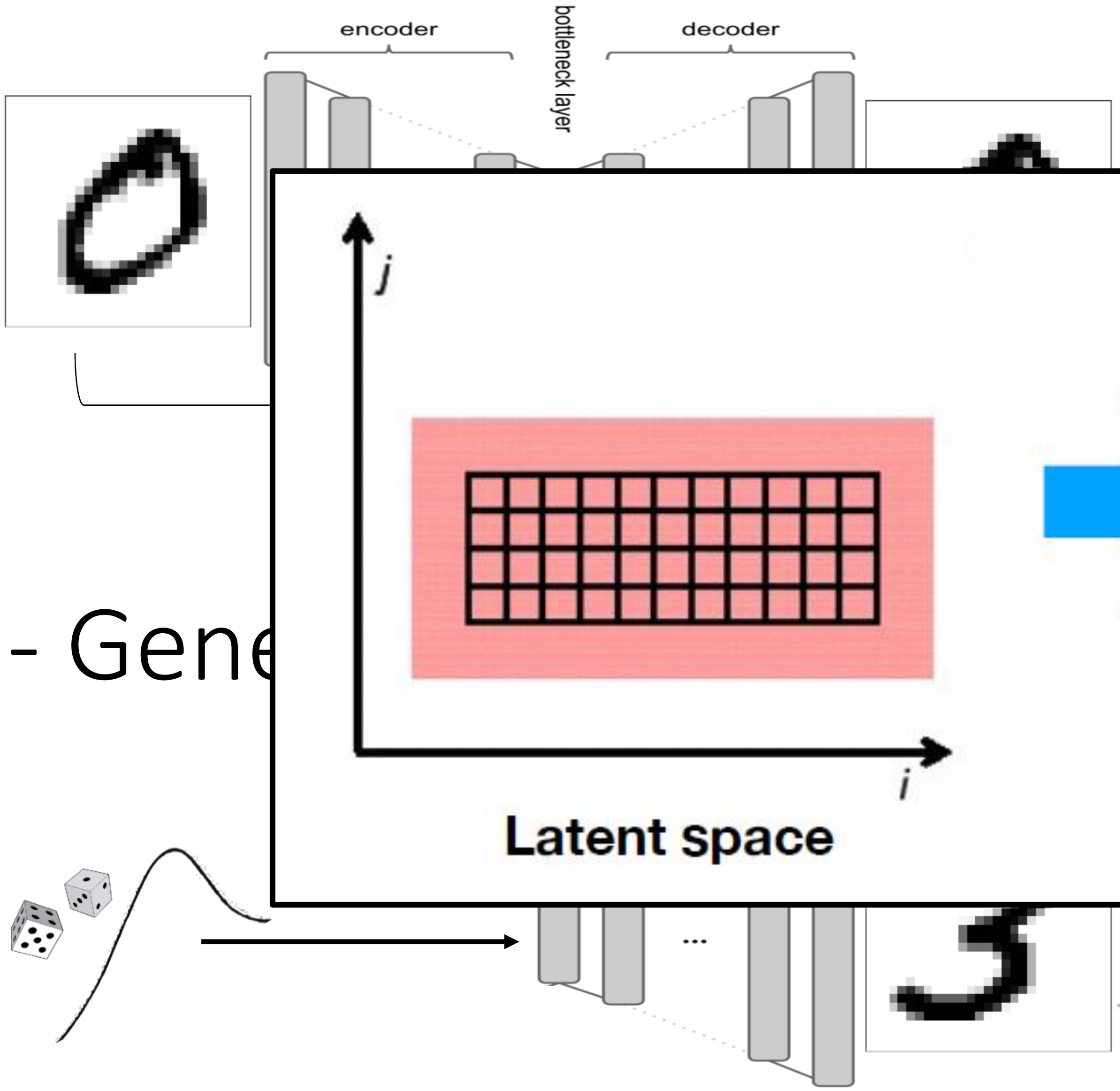


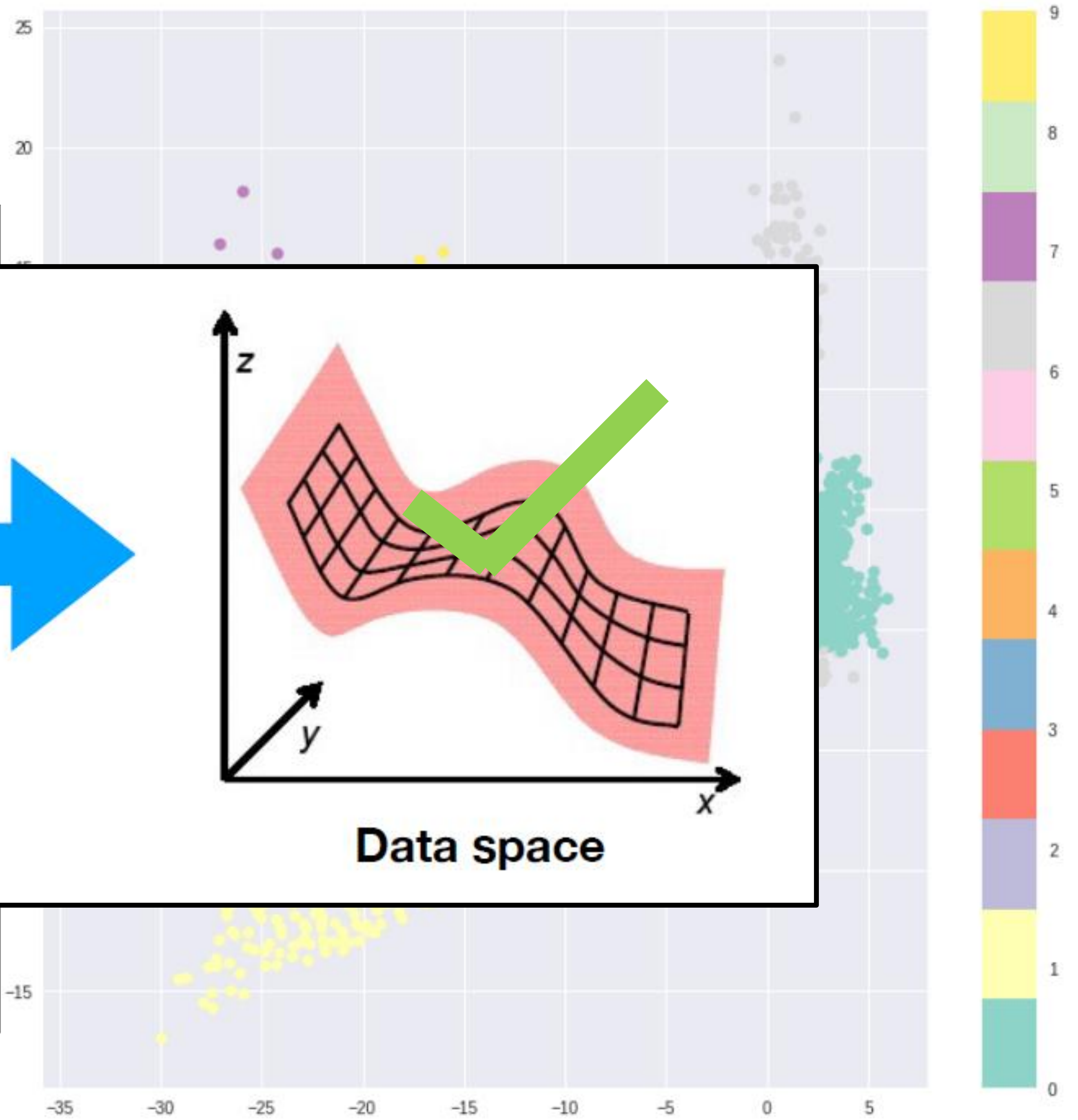


- Gene



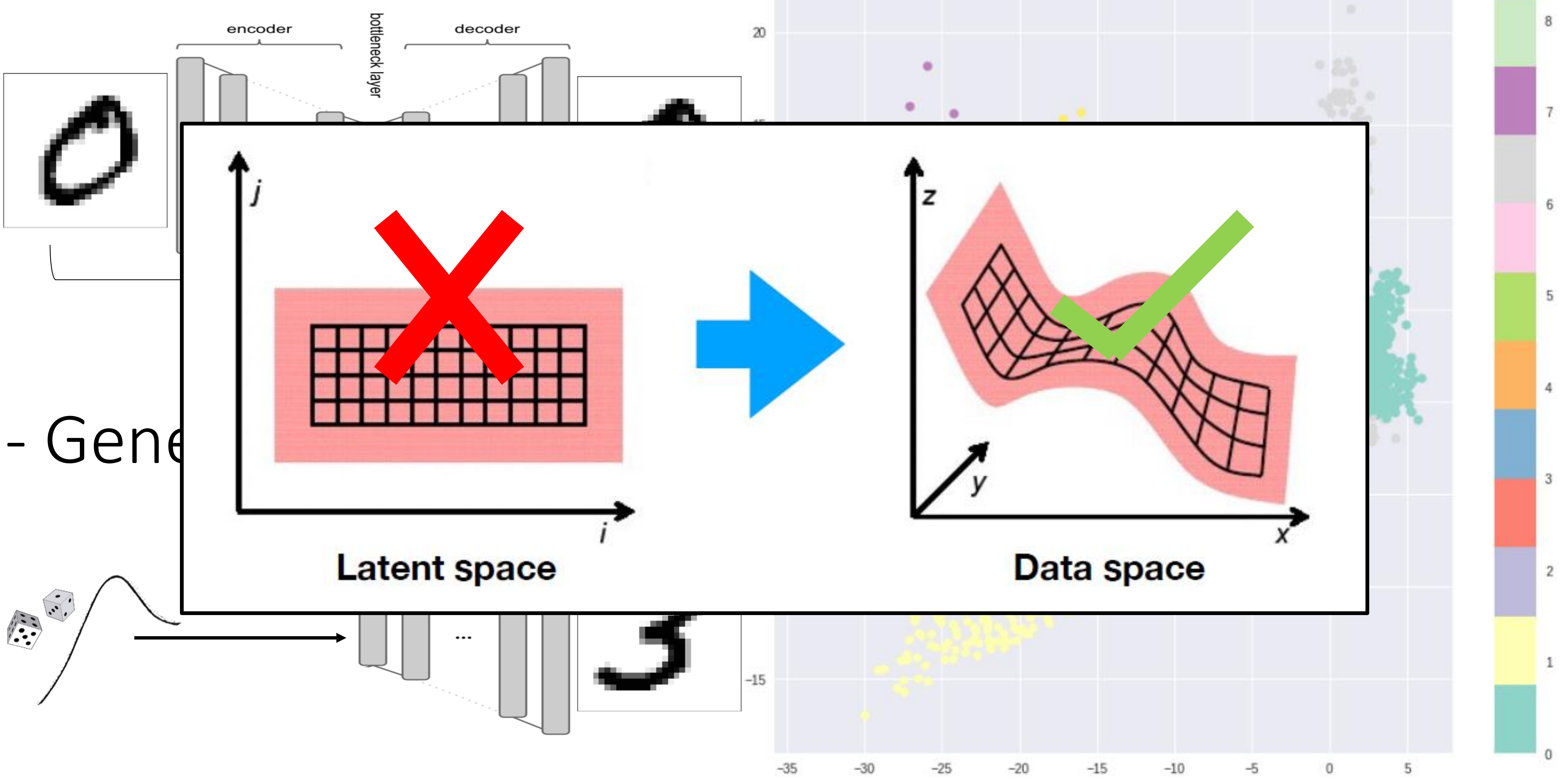
Autoencoders





- Gene

Autoencoders

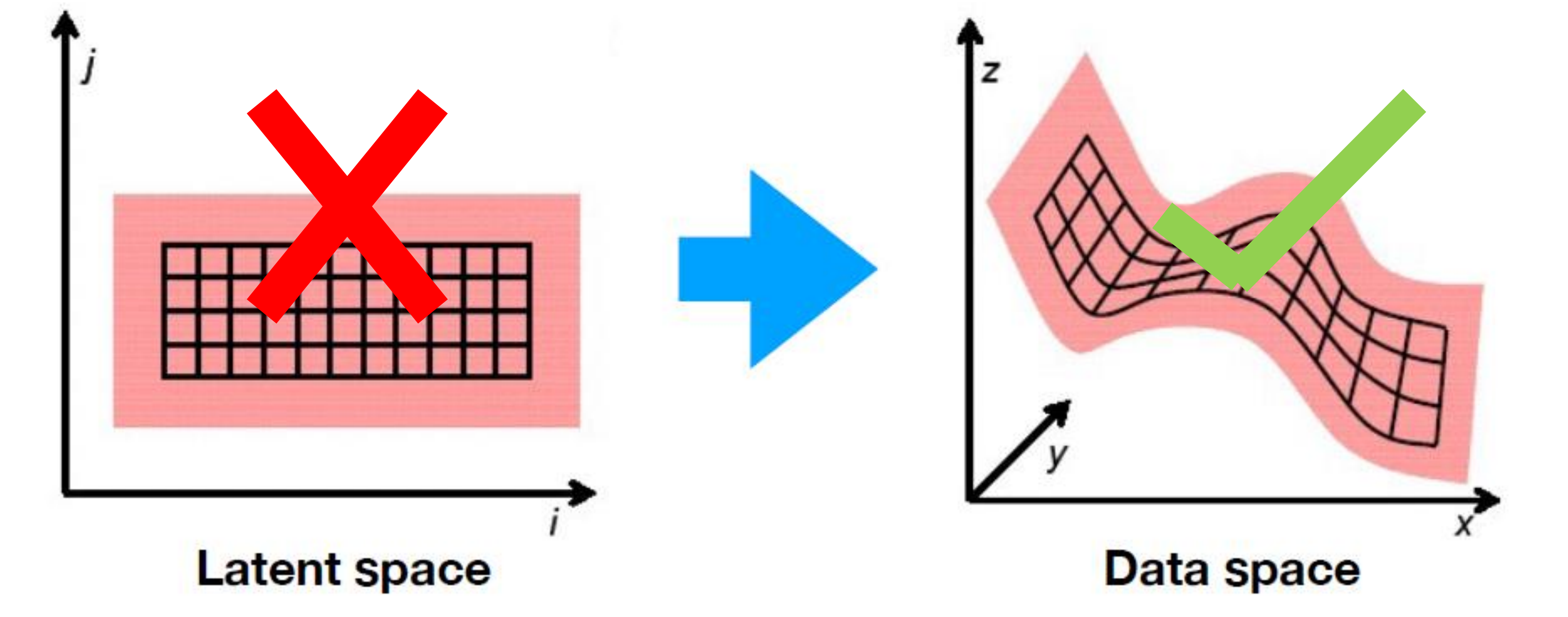


- Gene

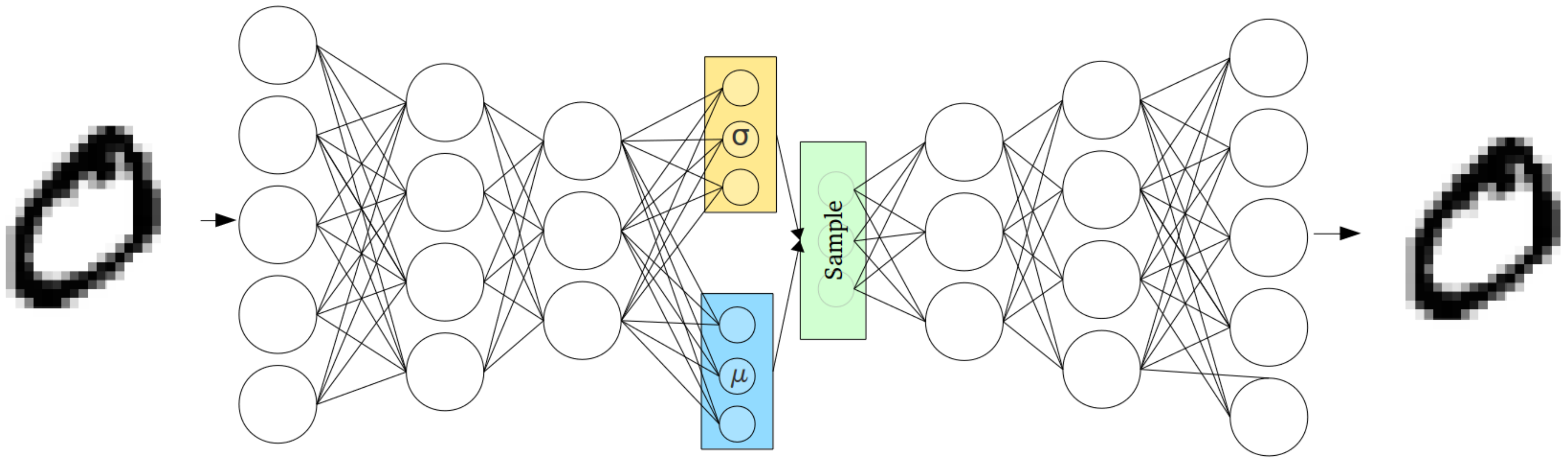
Autoencoders

- Gene

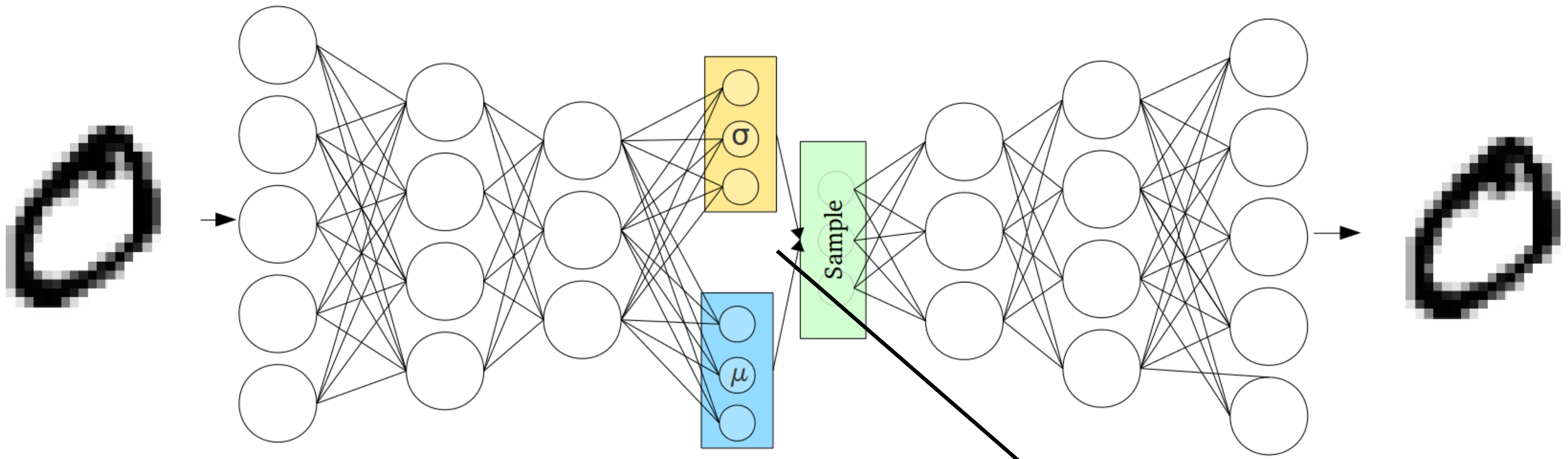
AE does not transform one *pre-determined* distribution to another!

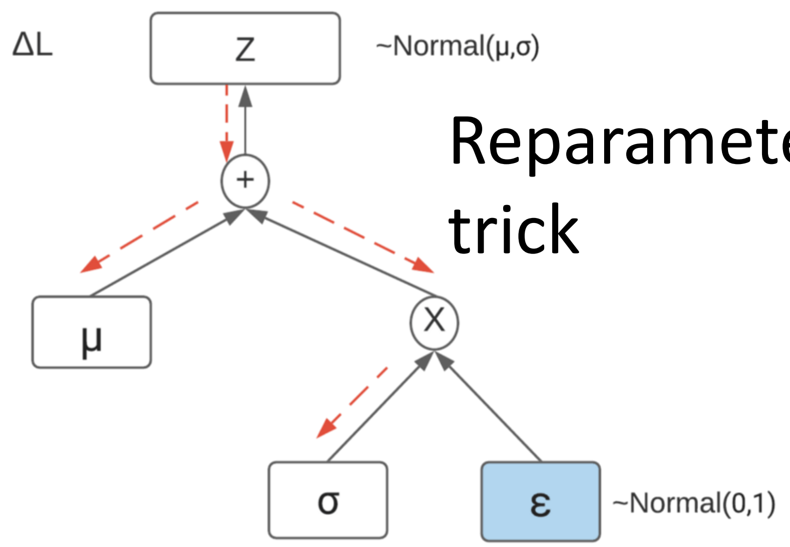


Variational Autoencoders (Kingma&Welling 2014)



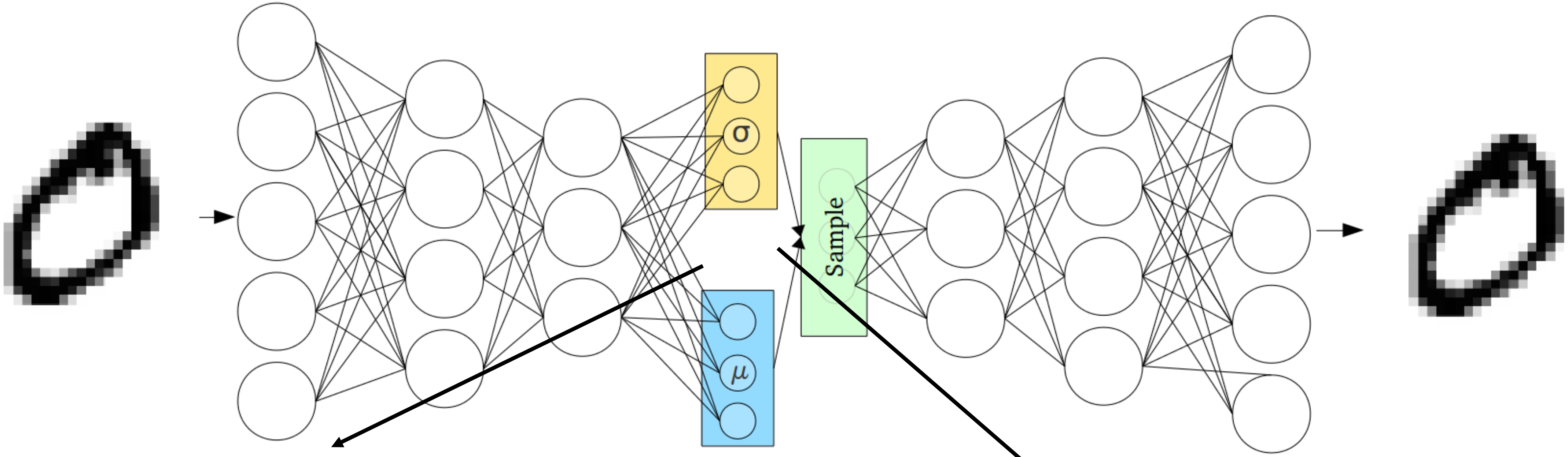
Variational Autoencoders (Kingma&Welling 2014)



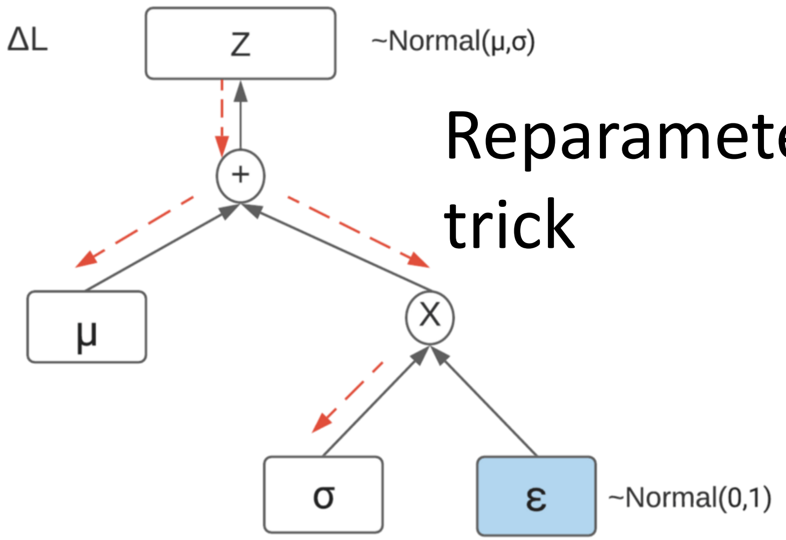


Reparameterization trick

Variational Autoencoders (Kingma&Welling 2014)

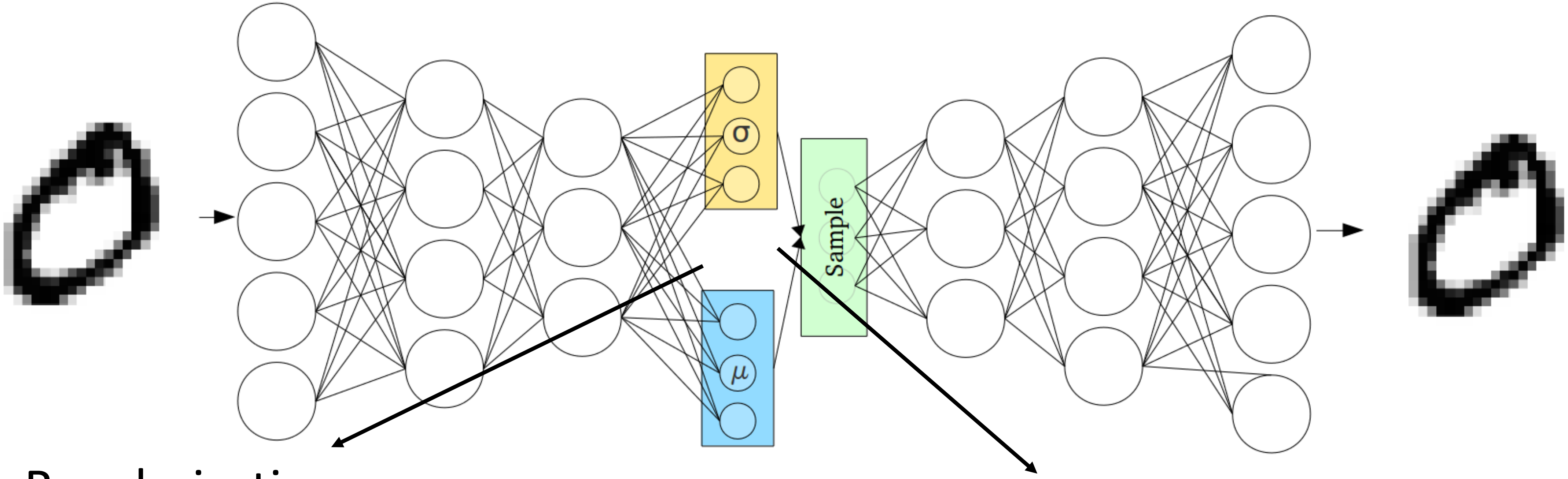


Regularization:
encourage $p(z) \sim \mathrm{N}(0,1)$



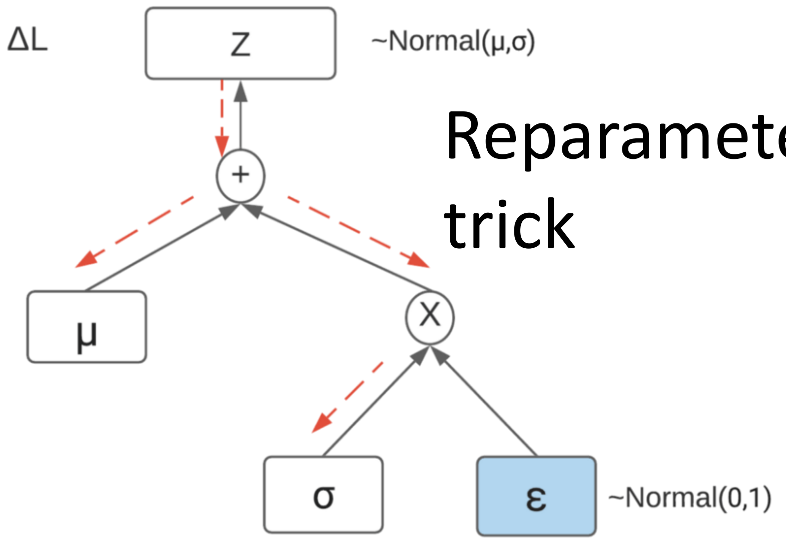
Reparameterization trick

Variational Autoencoders (Kingma&Welling 2014)



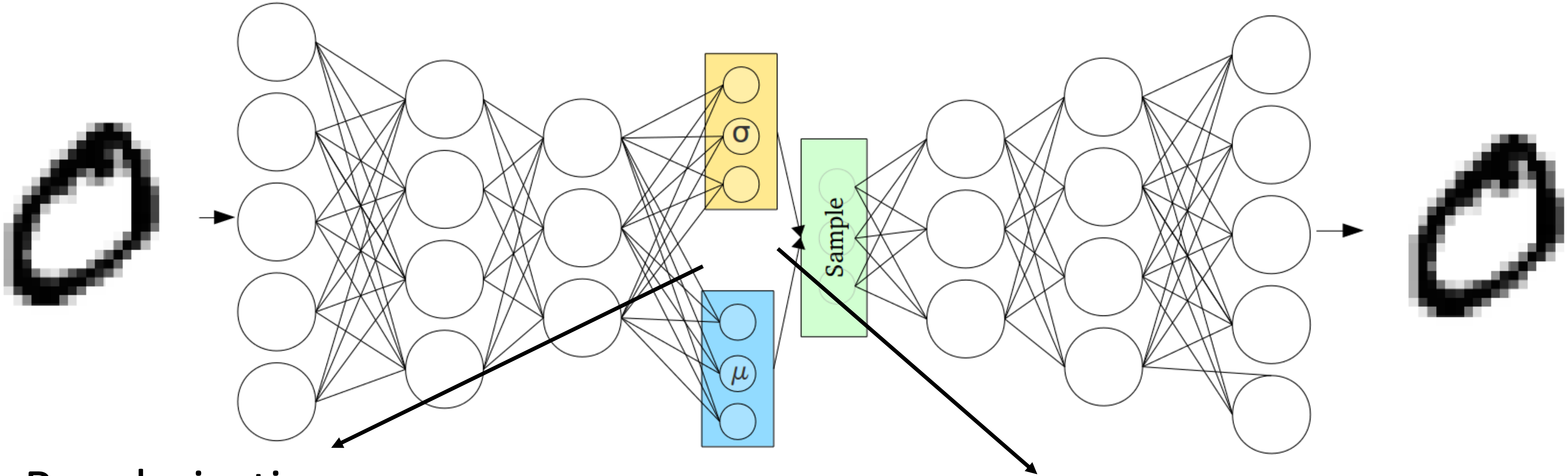
Regularization:
encourage $p(z) \sim \mathrm{N}(0,1)$

by KL divergence:



Reparameterization trick

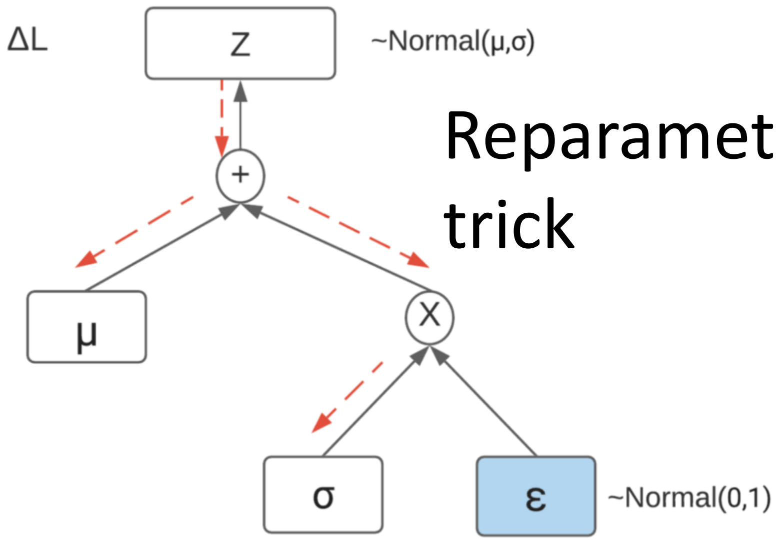
Variational Autoencoders (Kingma&Welling 2014)



Regularization:
encourage $p(z) \sim \mathrm{N}(0,1)$

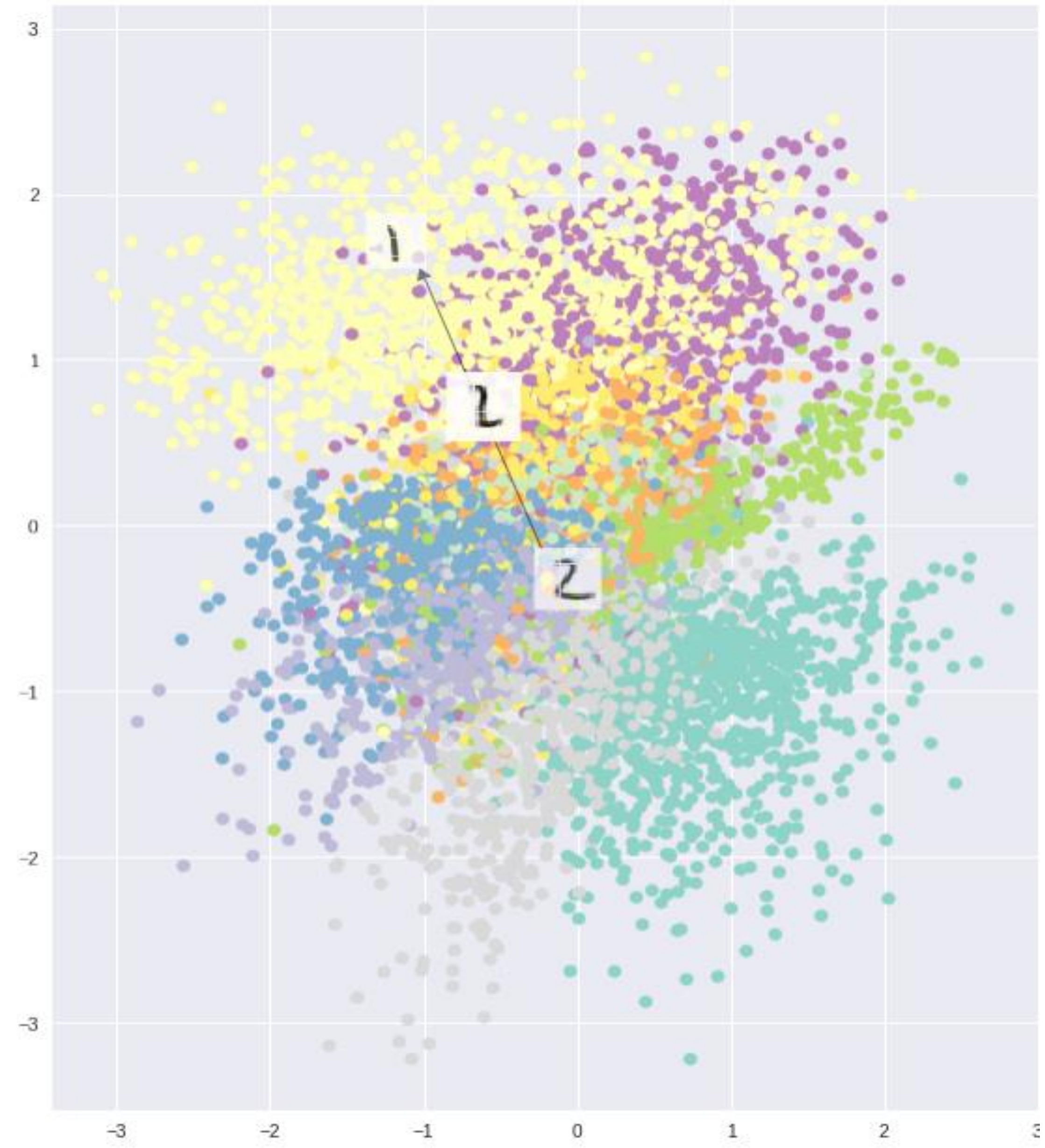
by KL divergence:

$$\sum_{i=1}^{n} \sigma_i^2 + \mu_i^2 - \log(\sigma_i) - 1$$



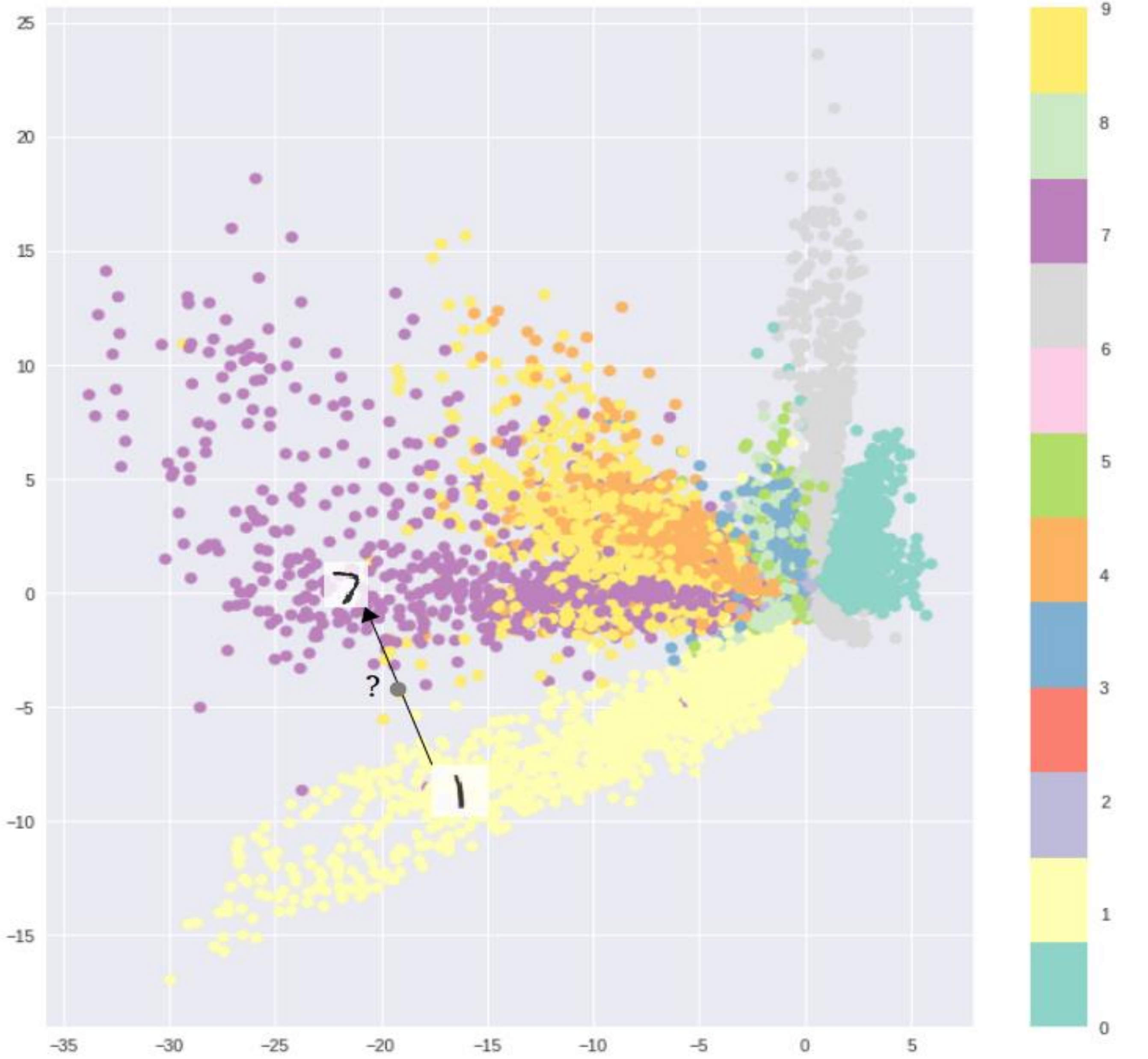
Reparameterization trick

VAE



3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3

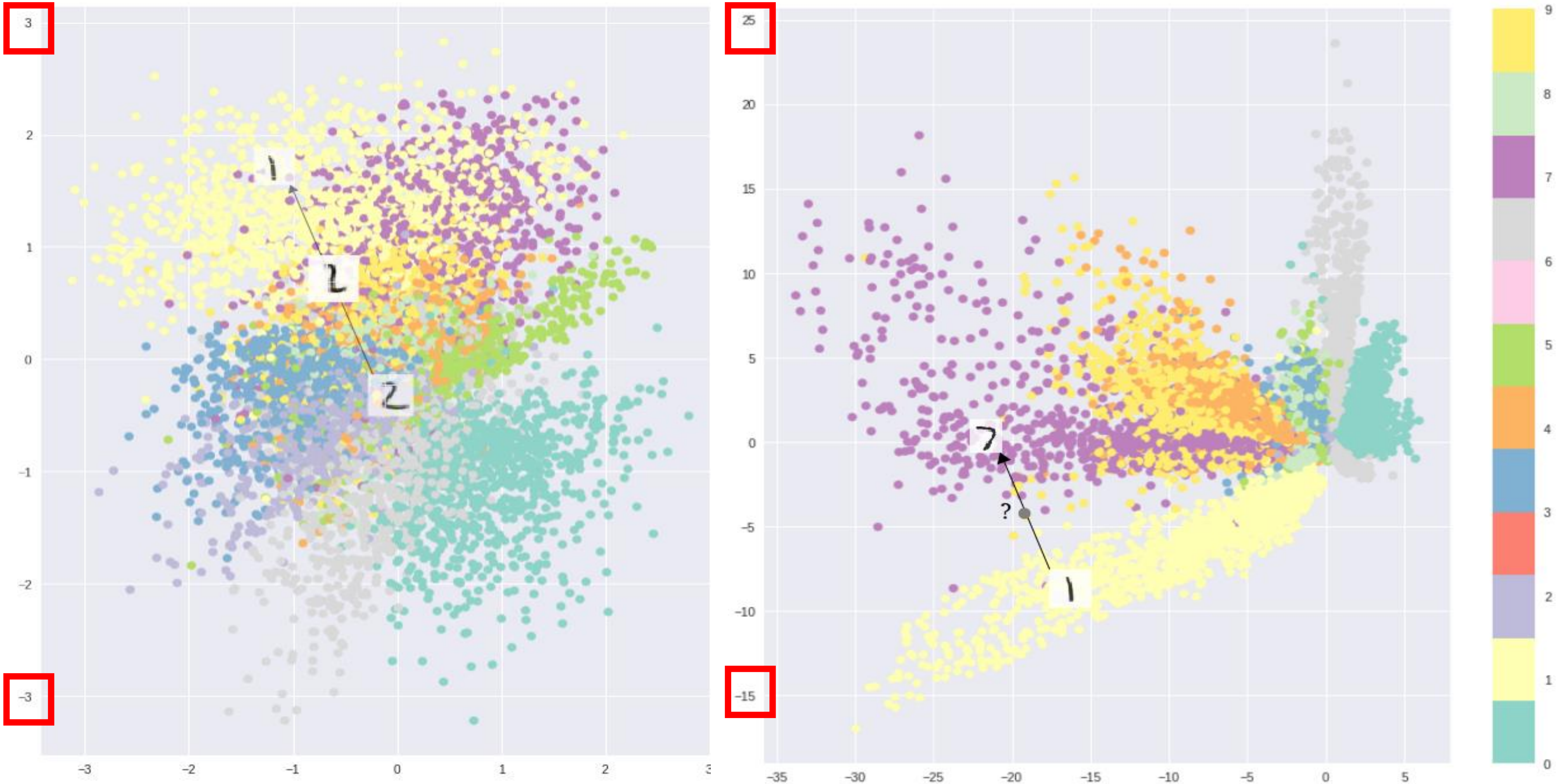
AE



25
20
15
10
5
0
−5
−10
−15
−35
−30
−25
−20
−15
−10
−5
0
5
?
9
8
7
6
5
4
3
2
1
0

VAE AE

Also check out the scale!



Probabilistic interpretation

Probabilistic interpretation

Data likelihood: $p_\theta(x)$

Probabilistic interpretation

Data likelihood: $p_\theta(x) = \int p_\theta(z)p_\theta(x|z)dz$

Probabilistic interpretation

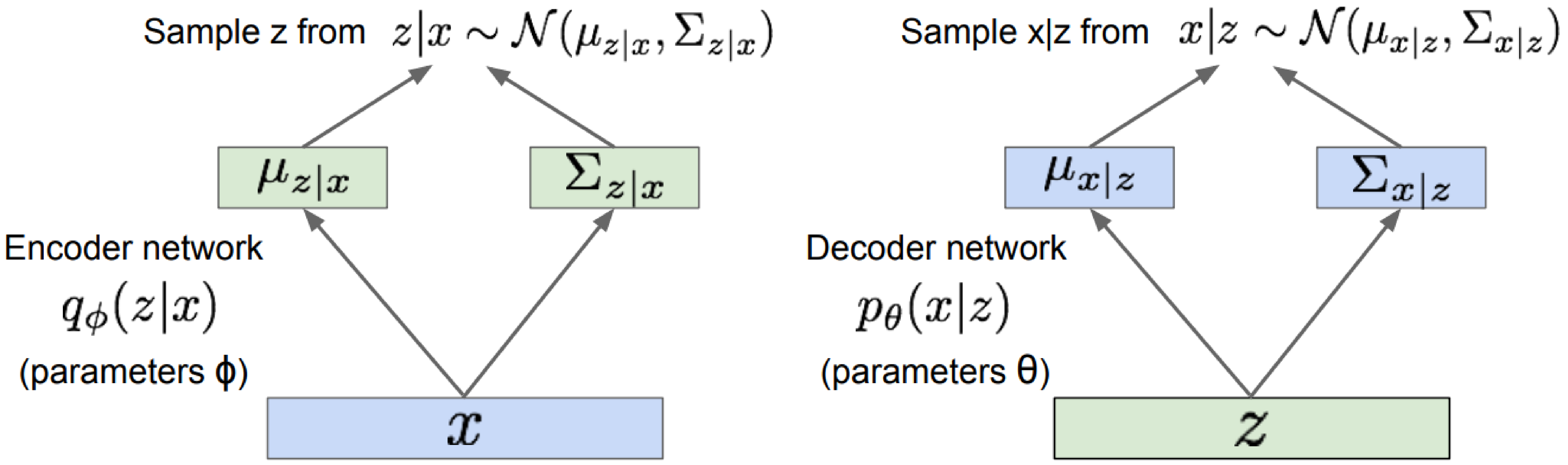
Data likelihood: $p_\theta(x) = \int p_\theta(z) p_\theta(x|z) dz$

Goal: make $\log p_\theta(x^{(i)})$ as high as possible

Probabilistic interpretation

Data likelihood: $p_\theta(x) = \int p_\theta(z) p_\theta(x|z) dz$

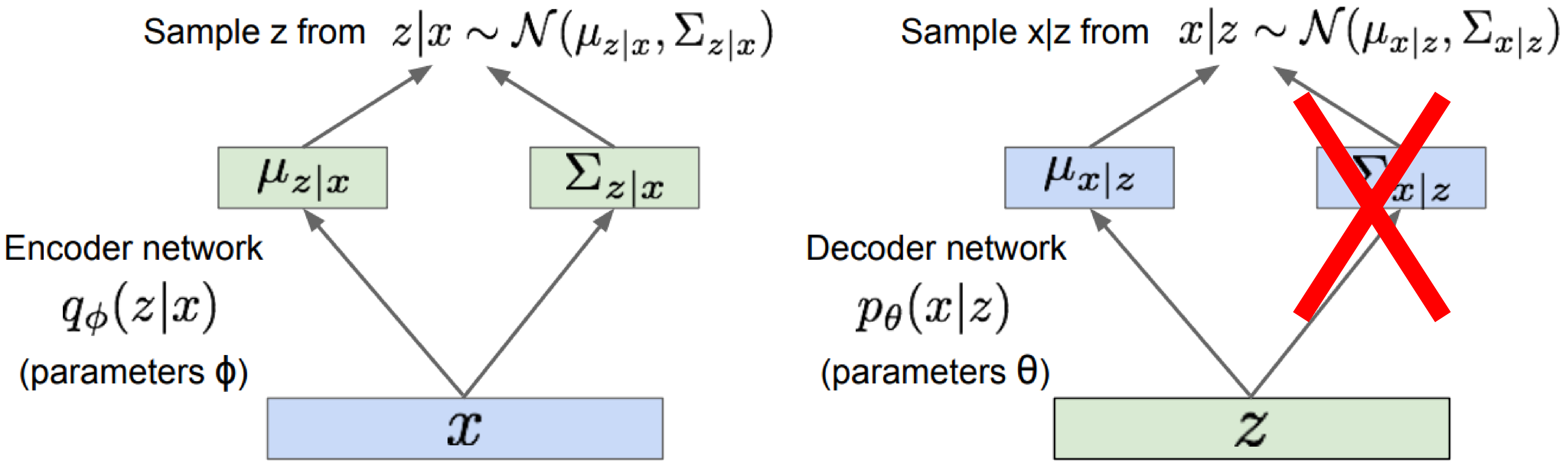
Goal: make $\log p_\theta(x^{(i)})$ as high as possible



Probabilistic interpretation

Data likelihood: $p_\theta(x) = \int p_\theta(z) p_\theta(x|z) dz$

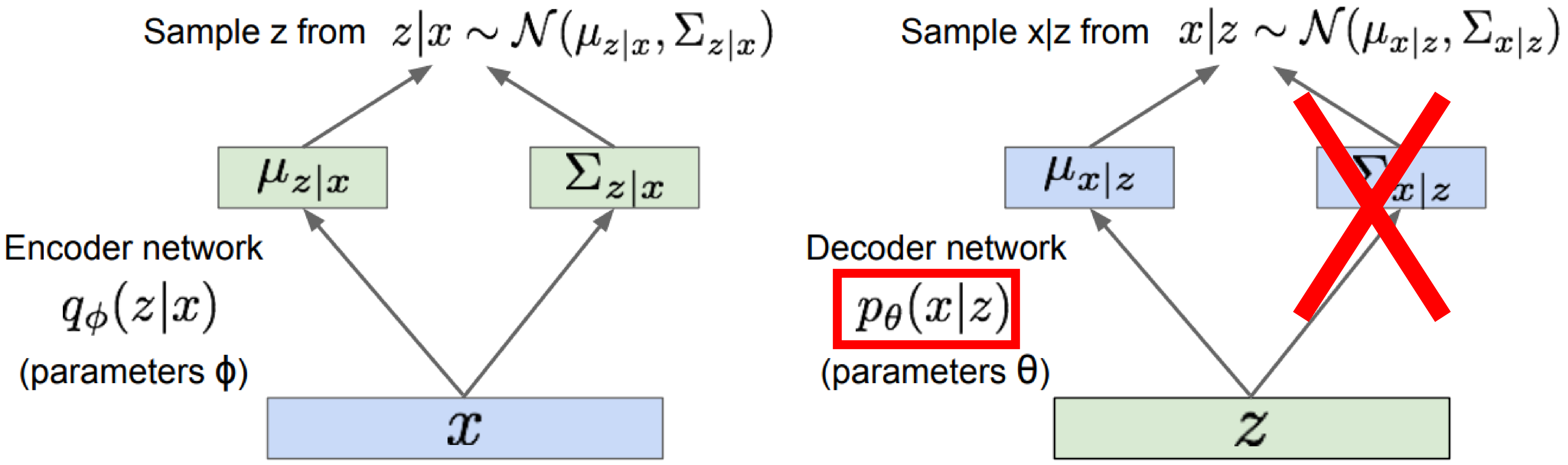
Goal: make $\log p_\theta(x^{(i)})$ as high as possible



Probabilistic interpretation

Data likelihood: $p_\theta(x) = \int p_\theta(z) \boxed{p_\theta(x|z)} dz$

Goal: make $\log p_\theta(x^{(i)})$ as high as possible

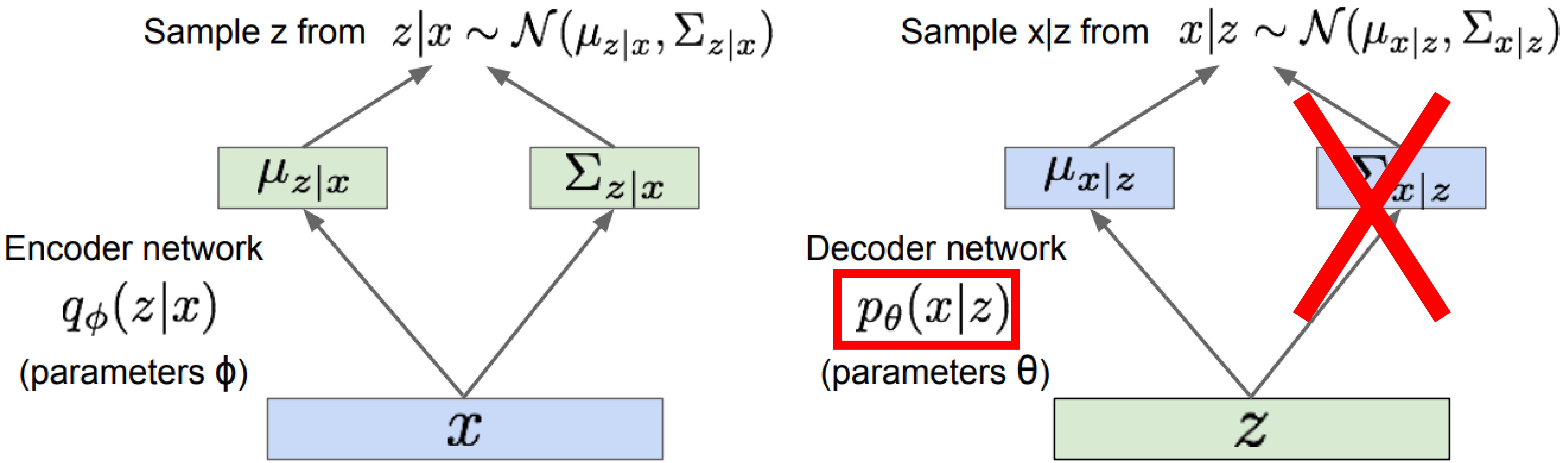


Probabilistic interpretation

N(0,1)

Data likelihood: $p_\theta(x) = \int p_\theta(z) p_\theta(x|z) dz$

Goal: make $\log p_\theta(x^{(i)})$ as high as possible



$$\log p_\theta(x^{(i)}) =$$

Slide credit: Stanford cs231n

$$\log p_\theta(x^{(i)}) = \mathbf{E}_{z\sim q_\phi(z|x^{(i)})}\left[\log p_\theta(x^{(i)})\right] \quad (p_\theta(x^{(i)}) \text{ Does not depend on } z)$$

Slide credit: Stanford cs231n

$$
\begin{aligned}
\log p_\theta(x^{(i)}) &= \mathbf{E}_{z \sim q_\phi(z|x^{(i)})} \left[\log p_\theta(x^{(i)}) \right] \qquad (p_\theta(x^{(i)}) \text{ Does not depend on } z) \\
&= \mathbf{E}_z \left[\log \frac{p_\theta(x^{(i)} \mid z) p_\theta(z)}{p_\theta(z \mid x^{(i)})} \right] \qquad (\text{Bayes' Rule})
\end{aligned}
$$

Slide credit: Stanford cs231n

$$
\begin{aligned}
\log p_\theta(x^{(i)}) &= \mathbf{E}_{z \sim q_\phi(z|x^{(i)})}\left[\log p_\theta(x^{(i)})\right] \quad (p_\theta(x^{(i)}) \text{ Does not depend on } z) \\
&= \mathbf{E}_z\left[\log \frac{p_\theta(x^{(i)} \mid z) p_\theta(z)}{p_\theta(z \mid x^{(i)})}\right] \quad \text{(Bayes' Rule)} \\
&= \mathbf{E}_z\left[\log \frac{p_\theta(x^{(i)} \mid z) p_\theta(z)}{p_\theta(z \mid x^{(i)})} \frac{q_\phi(z \mid x^{(i)})}{q_\phi(z \mid x^{(i)})}\right] \quad \text{(Multiply by constant)}
\end{aligned}
$$

$$
\begin{aligned}
\log p_\theta(x^{(i)}) &= \mathbf{E}_{z \sim q_\phi(z|x^{(i)})} \left[\log p_\theta(x^{(i)}) \right] \qquad (p_\theta(x^{(i)}) \text{ Does not depend on } z) \\
&= \mathbf{E}_z \left[\log \frac{p_\theta(x^{(i)} \mid z) p_\theta(z)}{p_\theta(z \mid x^{(i)})} \right] \qquad (\text{Bayes' Rule}) \\
&= \mathbf{E}_z \left[\log \frac{p_\theta(x^{(i)} \mid z) p_\theta(z)}{p_\theta(z \mid x^{(i)})} \frac{q_\phi(z \mid x^{(i)})}{q_\phi(z \mid x^{(i)})} \right] \qquad (\text{Multiply by constant}) \\
&= \mathbf{E}_z \left[\log p_\theta(x^{(i)} \mid z) \right] - \mathbf{E}_z \left[\log \frac{q_\phi(z \mid x^{(i)})}{p_\theta(z)} \right] + \mathbf{E}_z \left[\log \frac{q_\phi(z \mid x^{(i)})}{p_\theta(z \mid x^{(i)})} \right] \qquad (\text{Logarithms})
\end{aligned}
$$

Slide credit: Stanford cs231n

$$
\begin{aligned}
\log p_\theta(x^{(i)}) &= \mathbf{E}_{z\sim q_\phi(z|x^{(i)})}\left[\log p_\theta(x^{(i)})\right] \quad (p_\theta(x^{(i)}) \text{ Does not depend on } z) \\
&= \mathbf{E}_z\left[\log \frac{p_\theta(x^{(i)} \mid z)p_\theta(z)}{p_\theta(z \mid x^{(i)})}\right] \quad (\text{Bayes' Rule}) \\
&= \mathbf{E}_z\left[\log \frac{p_\theta(x^{(i)} \mid z)p_\theta(z)}{p_\theta(z \mid x^{(i)})}\frac{q_\phi(z \mid x^{(i)})}{q_\phi(z \mid x^{(i)})}\right] \quad (\text{Multiply by constant}) \\
&= \mathbf{E}_z\left[\log p_\theta(x^{(i)} \mid z)\right] - \mathbf{E}_z\left[\log \frac{q_\phi(z \mid x^{(i)})}{p_\theta(z)}\right] + \mathbf{E}_z\left[\log \frac{q_\phi(z \mid x^{(i)})}{p_\theta(z \mid x^{(i)})}\right] \quad (\text{Logarithms}) \\
&= \mathbf{E}_z\left[\log p_\theta(x^{(i)} \mid z)\right] - D_{KL}(q_\phi(z \mid x^{(i)}) \,||\, p_\theta(z)) + D_{KL}(q_\phi(z \mid x^{(i)}) \,||\, p_\theta(z \mid x^{(i)}))
\end{aligned}
$$

$$\log p_\theta(x^{(i)}) = \mathbf{E}_{z\sim q_\phi(z|x^{(i)})}\left[\log p_\theta(x^{(i)})\right] \qquad (p_\theta(x^{(i)}) \text{ Does not depend on } z)$$

$$= \mathbf{E}_z\left[\log \frac{p_\theta(x^{(i)} \mid z)p_\theta(z)}{p_\theta(z \mid x^{(i)})}\right] \qquad \text{(Bayes' Rule)}$$

$$= \mathbf{E}_z\left[\log \frac{p_\theta(x^{(i)} \mid z)p_\theta(z)}{p_\theta(z \mid x^{(i)})}\frac{q_\phi(z \mid x^{(i)})}{q_\phi(z \mid x^{(i)})}\right] \qquad \text{(Multiply by constant)}$$

$$= \mathbf{E}_z\left[\log p_\theta(x^{(i)} \mid z)\right] - \mathbf{E}_z\left[\log \frac{q_\phi(z \mid x^{(i)})}{p_\theta(z)}\right] + \mathbf{E}_z\left[\log \frac{q_\phi(z \mid x^{(i)})}{p_\theta(z \mid x^{(i)})}\right] \qquad \text{(Logarithms)}$$

$$= \mathbf{E}_z\left[\log p_\theta(x^{(i)} \mid z)\right] - D_{KL}(q_\phi(z \mid x^{(i)}) \,||\, p_\theta(z)) + D_{KL}(q_\phi(z \mid x^{(i)}) \,||\, p_\theta(z \mid x^{(i)}))$$

Decoder network gives $p_\theta(x|z)$, can compute estimate of this term through sampling. (Sampling differentiable through reparam. trick, see paper.)

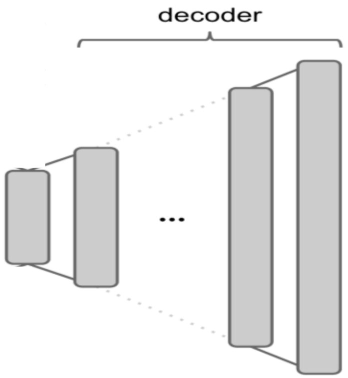
This KL term (between Gaussians for encoder and z prior) has nice closed-form solution!

$p_\theta(z|x)$ intractable (saw earlier), can't compute this KL term :(But we know KL divergence always >= 0.

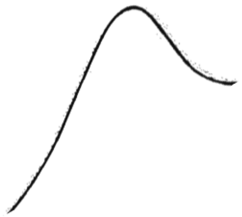
Slide credit: Stanford cs231n

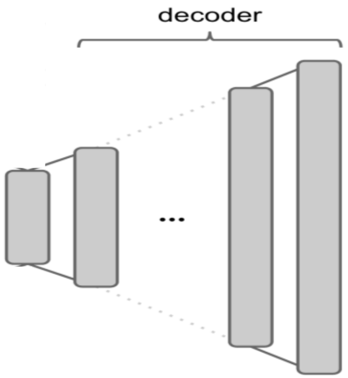
Generate data

Generate data

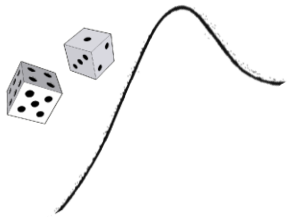


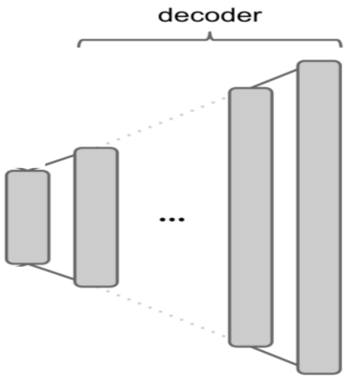
Generate data



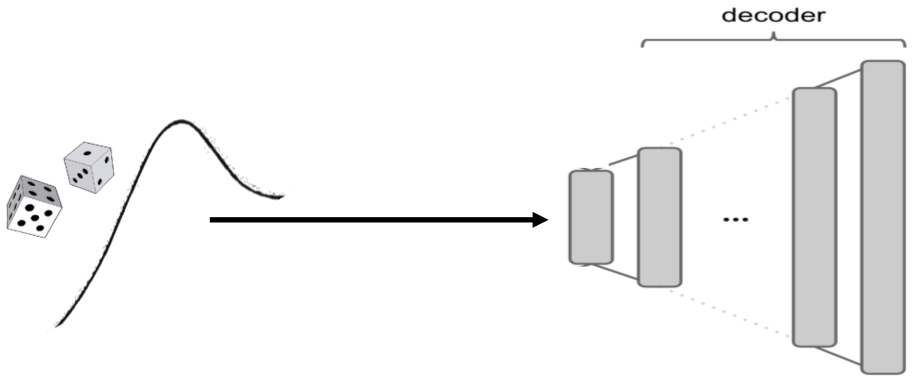


Generate data

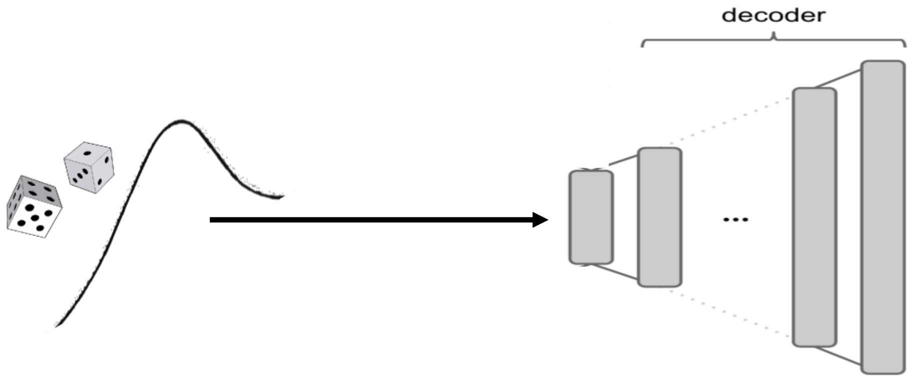


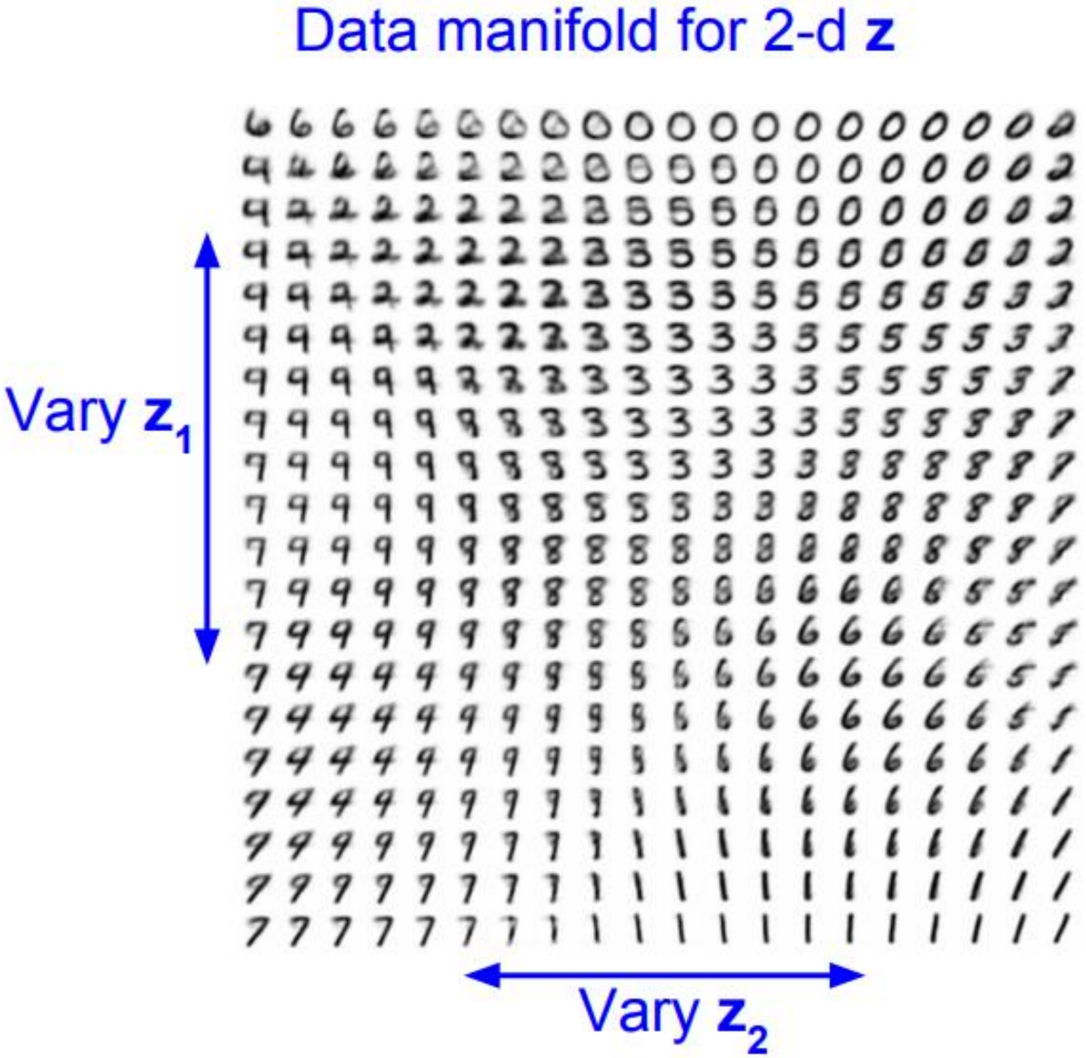


Generate data

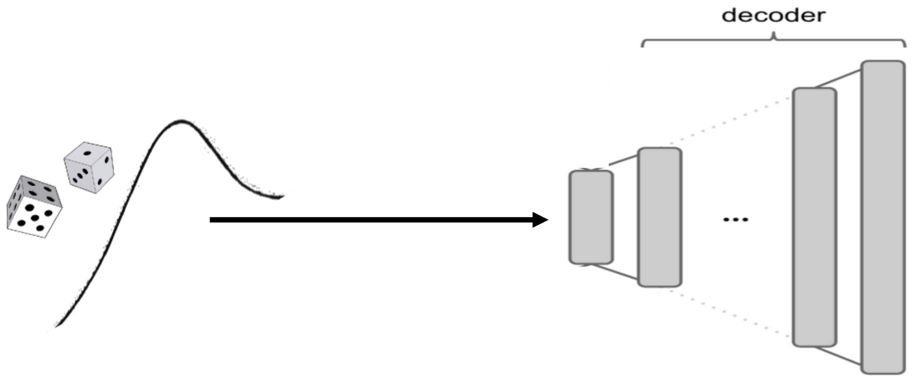


Generate data



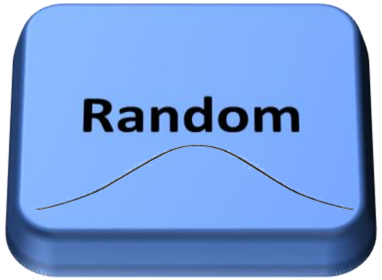


Generate data



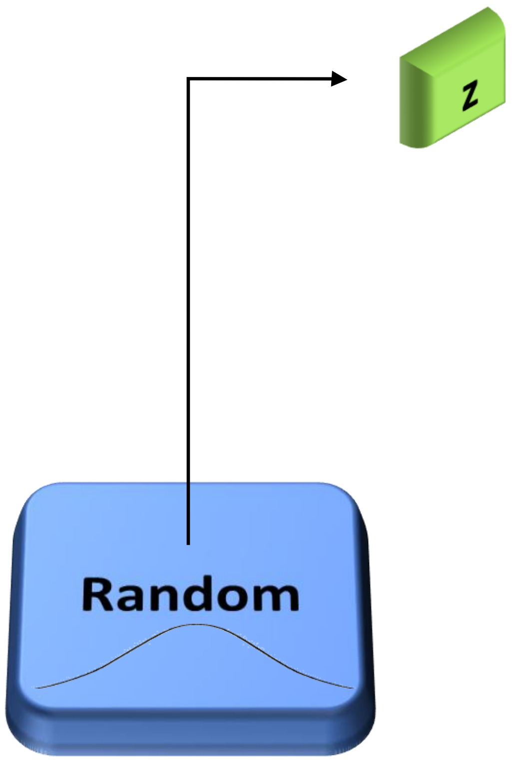
How about this idea for a generative model?

How about this idea for a generative model?



How about this idea for a generative model?

How about this idea for a generative model?



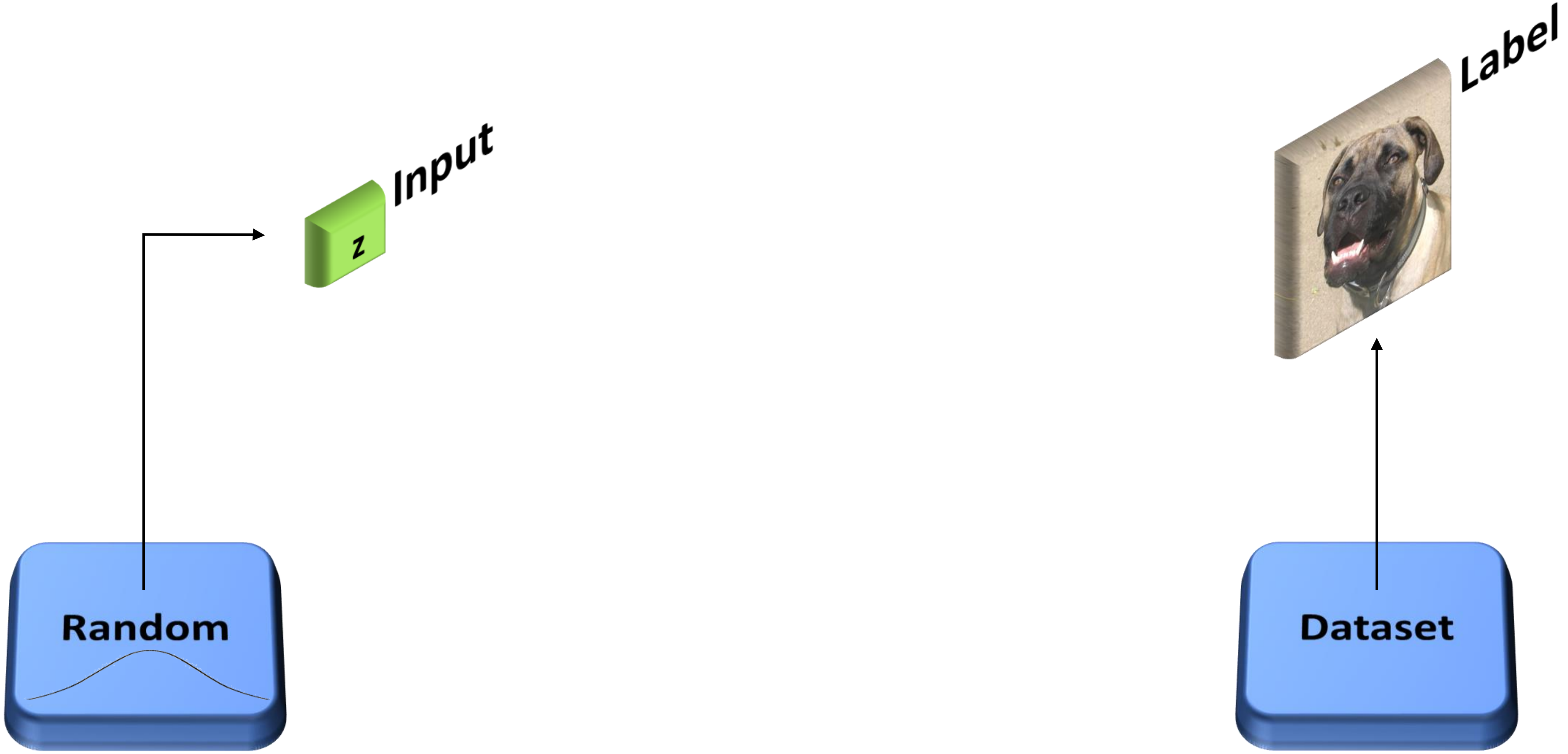
How about this idea for a generative model?

How about this idea for a generative model?

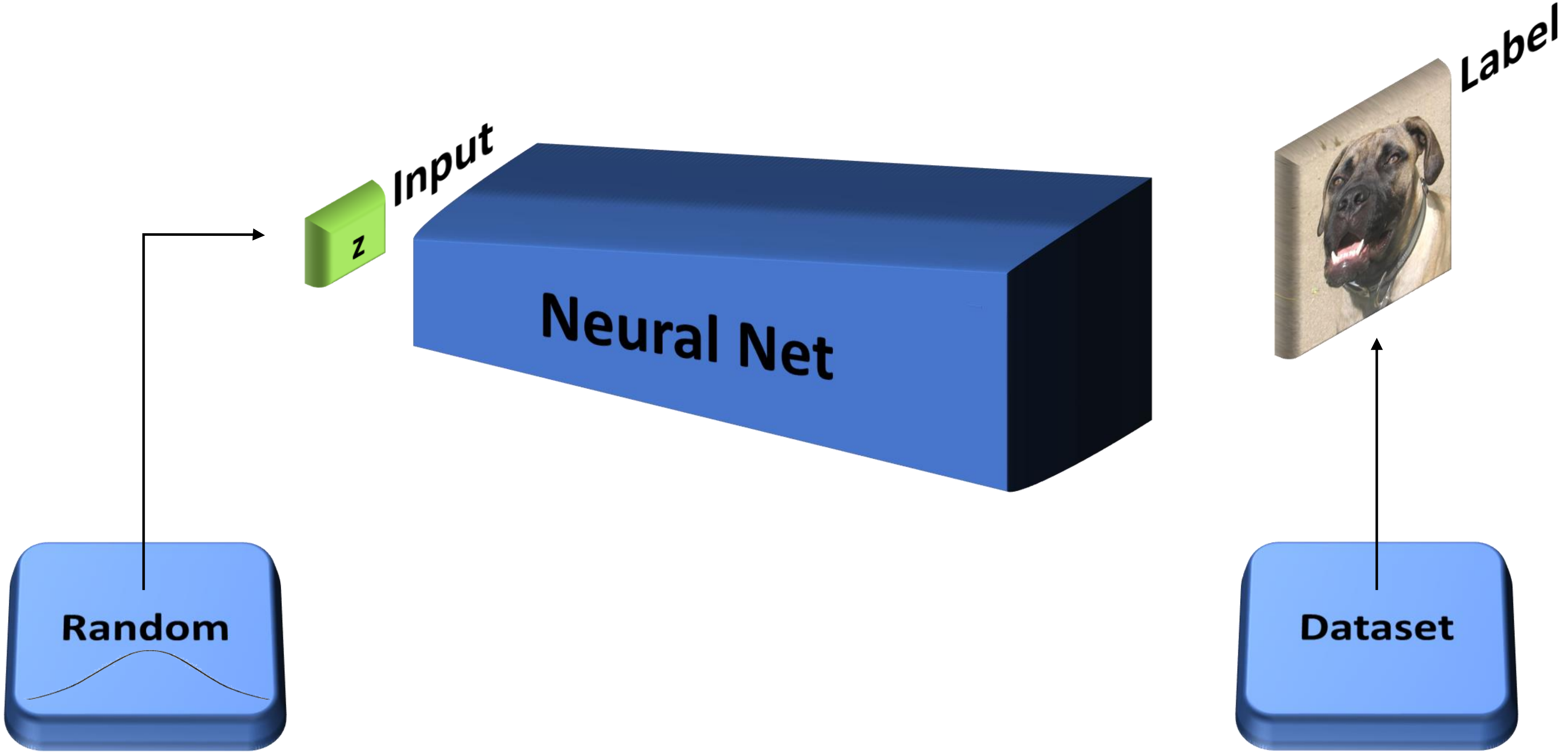
How about this idea for a generative model?

How about this idea for a generative model?

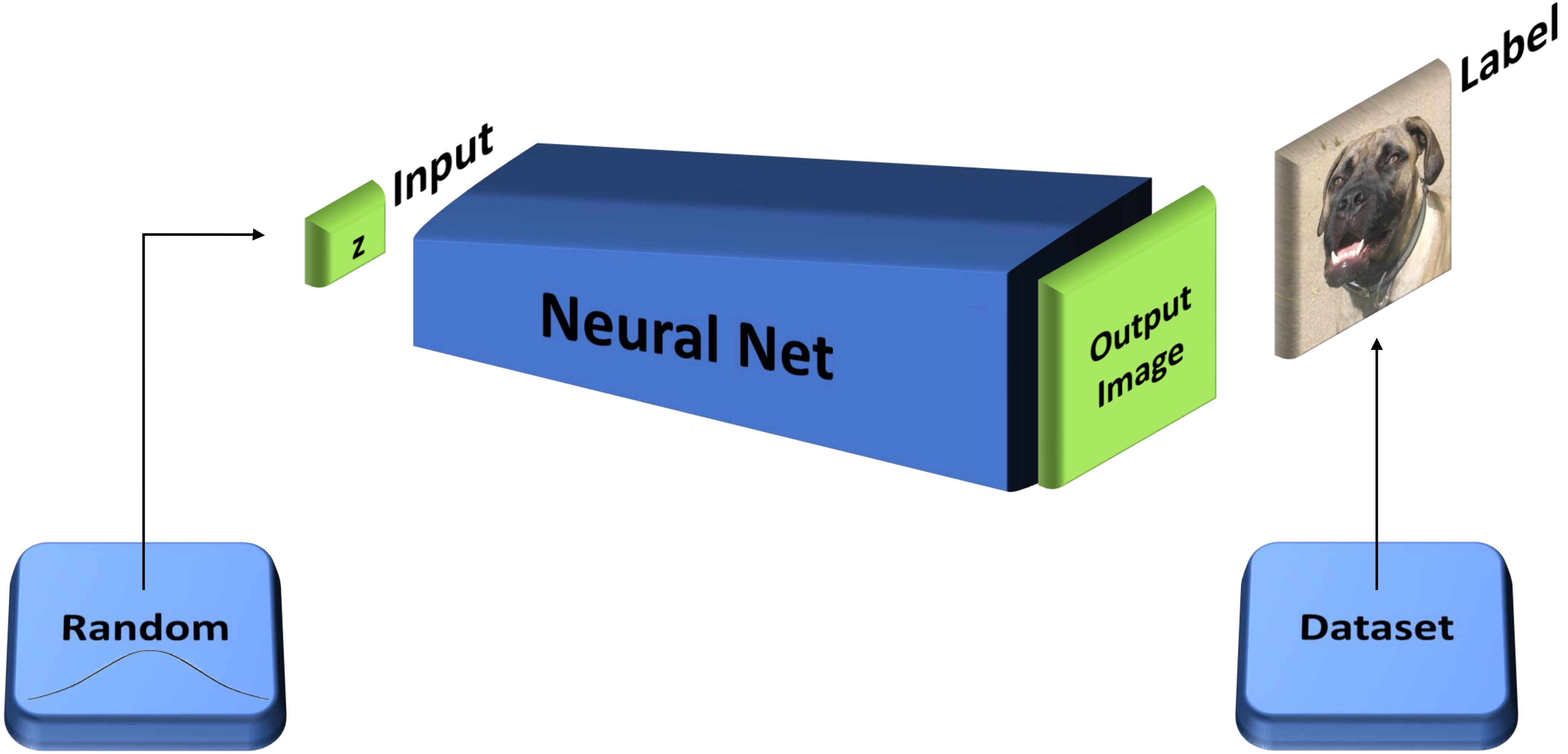
How about this idea for a generative model?



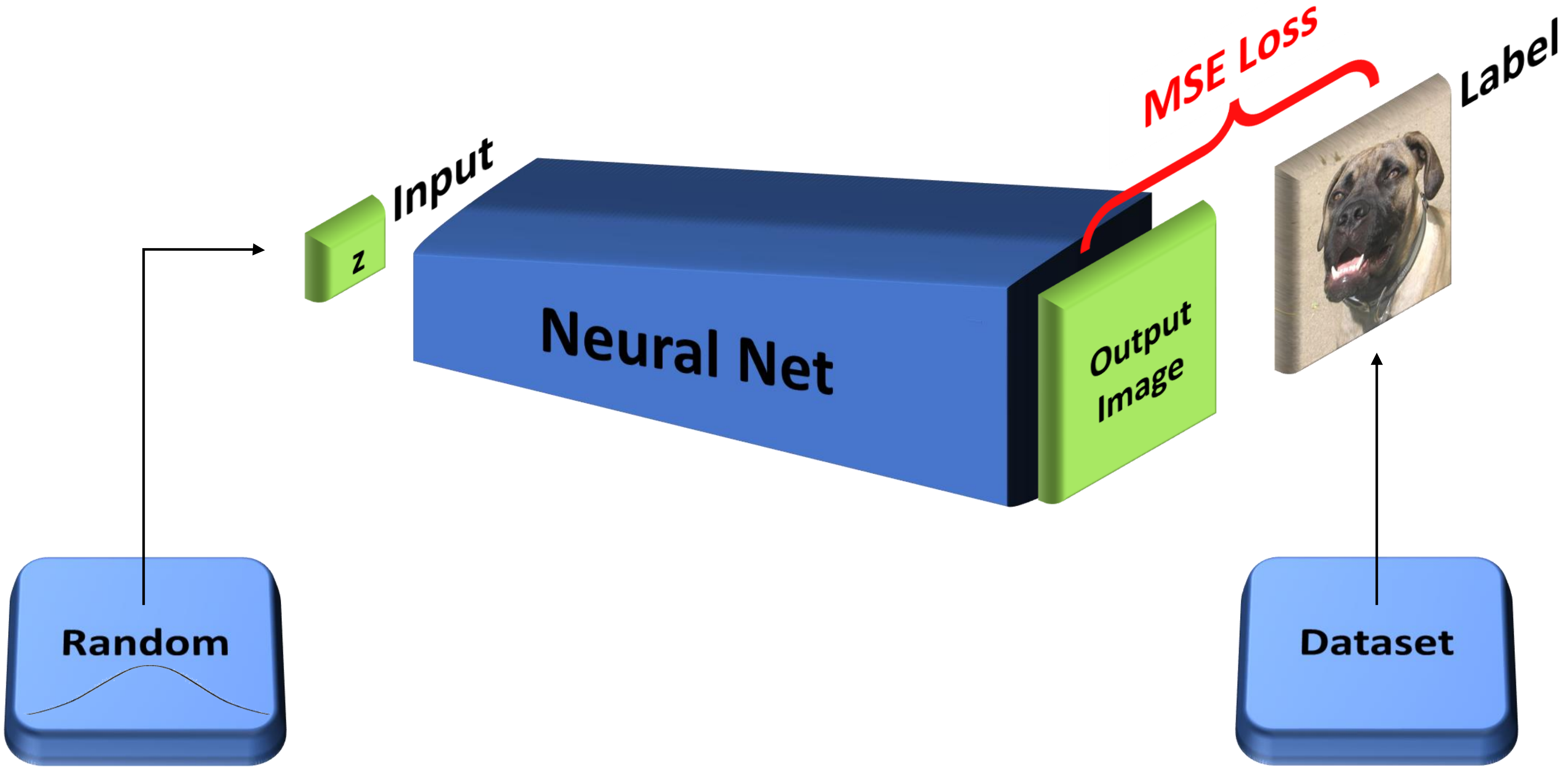
How about this idea for a generative model?



How about this idea for a generative model?

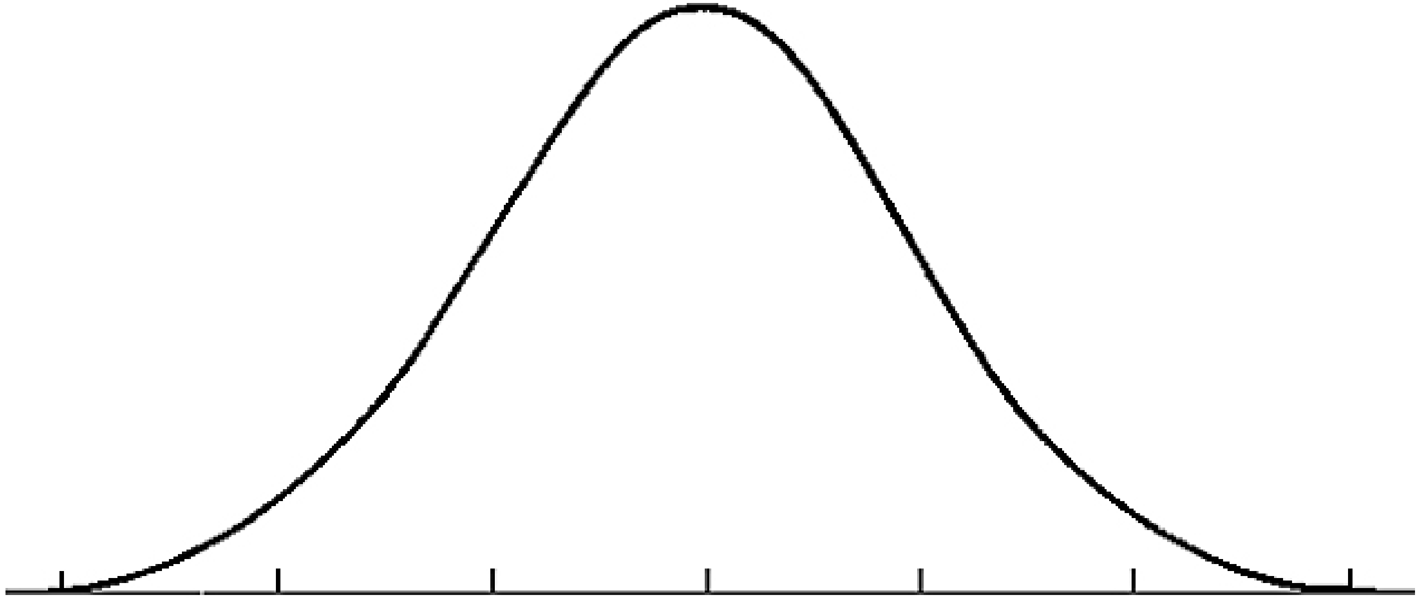


How about this idea for a generative model?

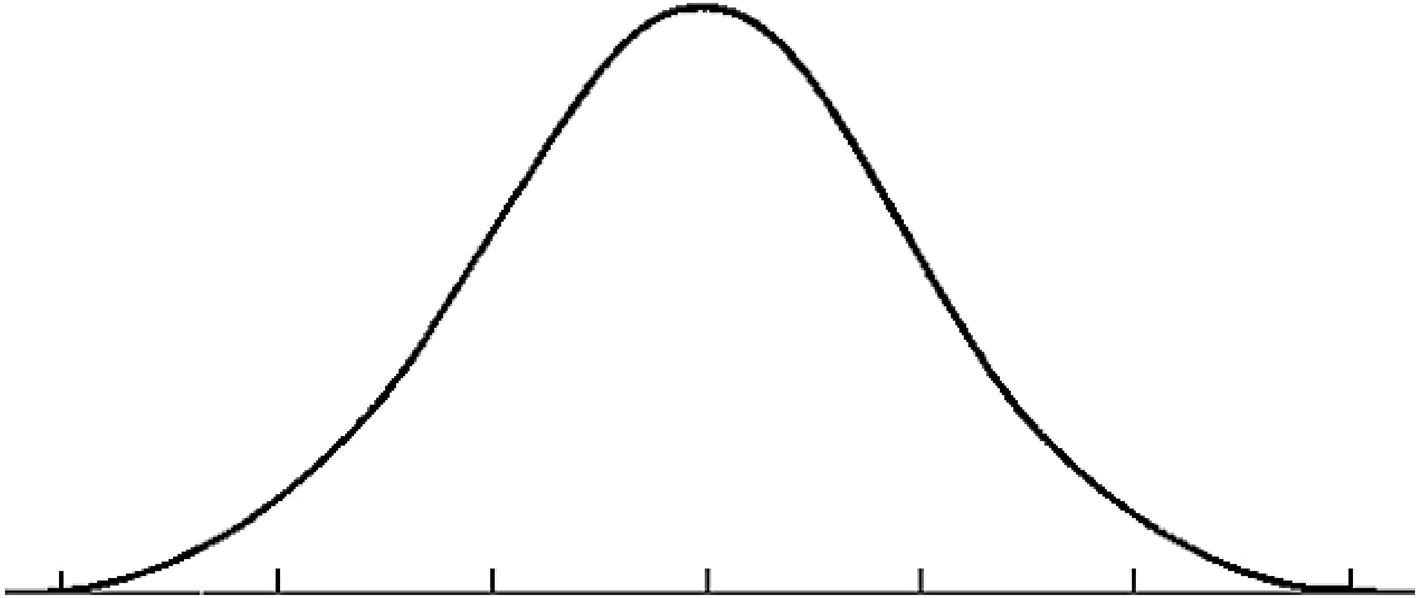


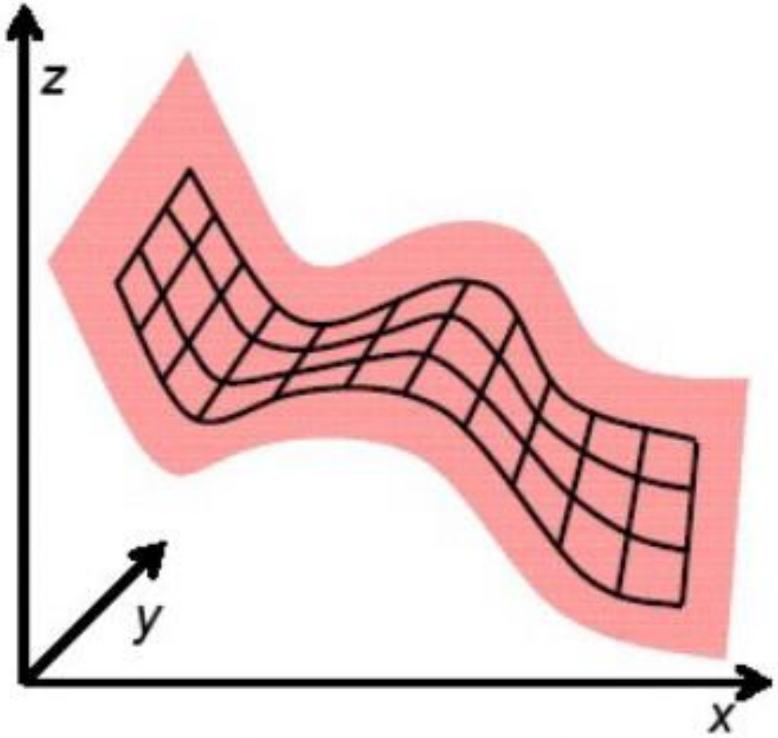
No good!

No good!

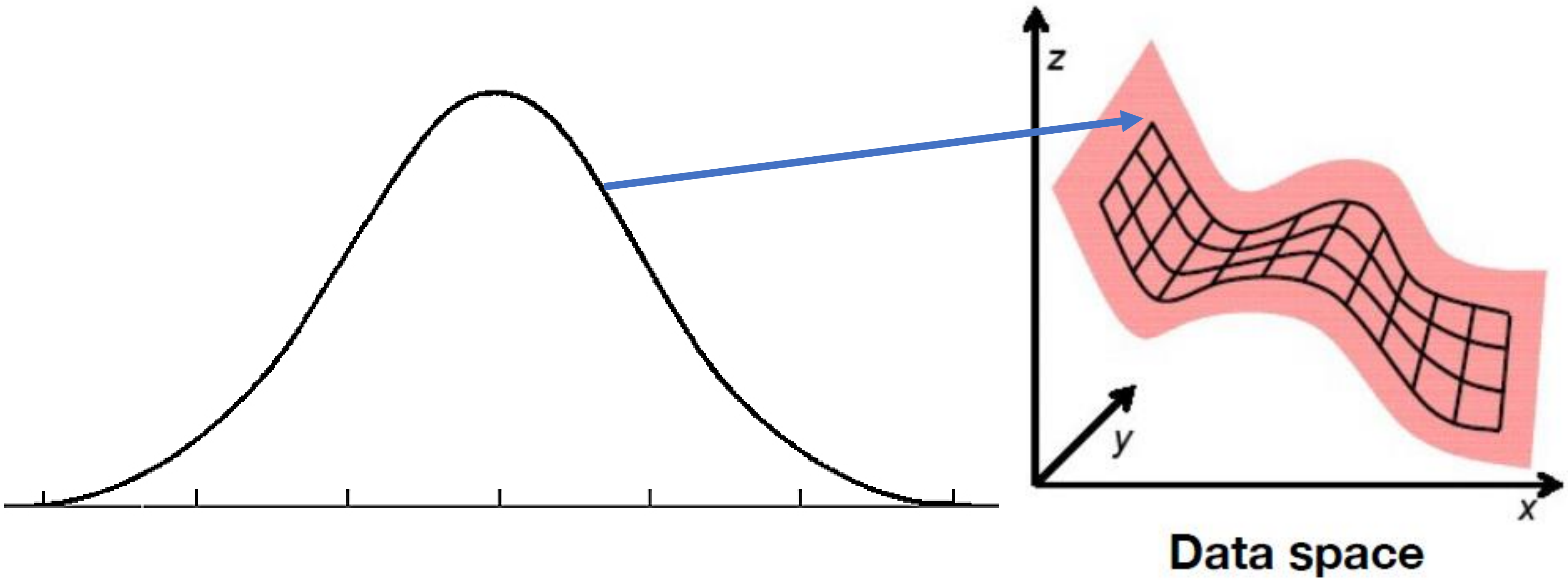


No good!

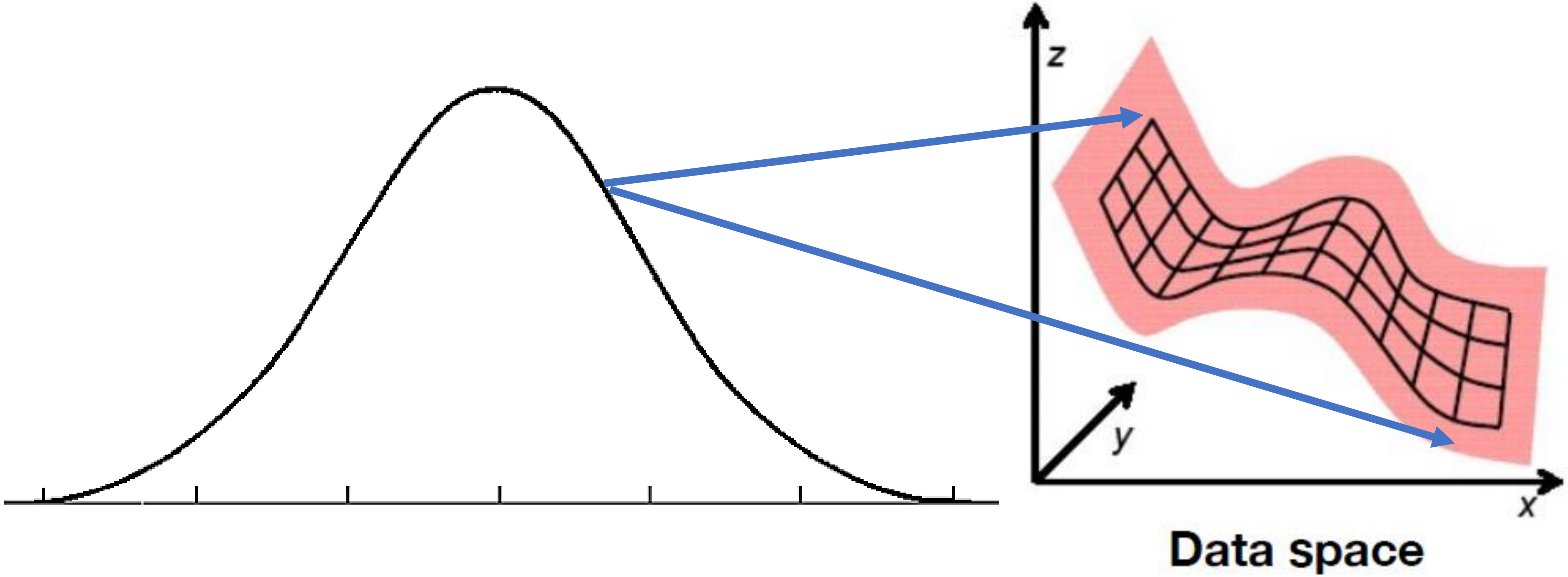




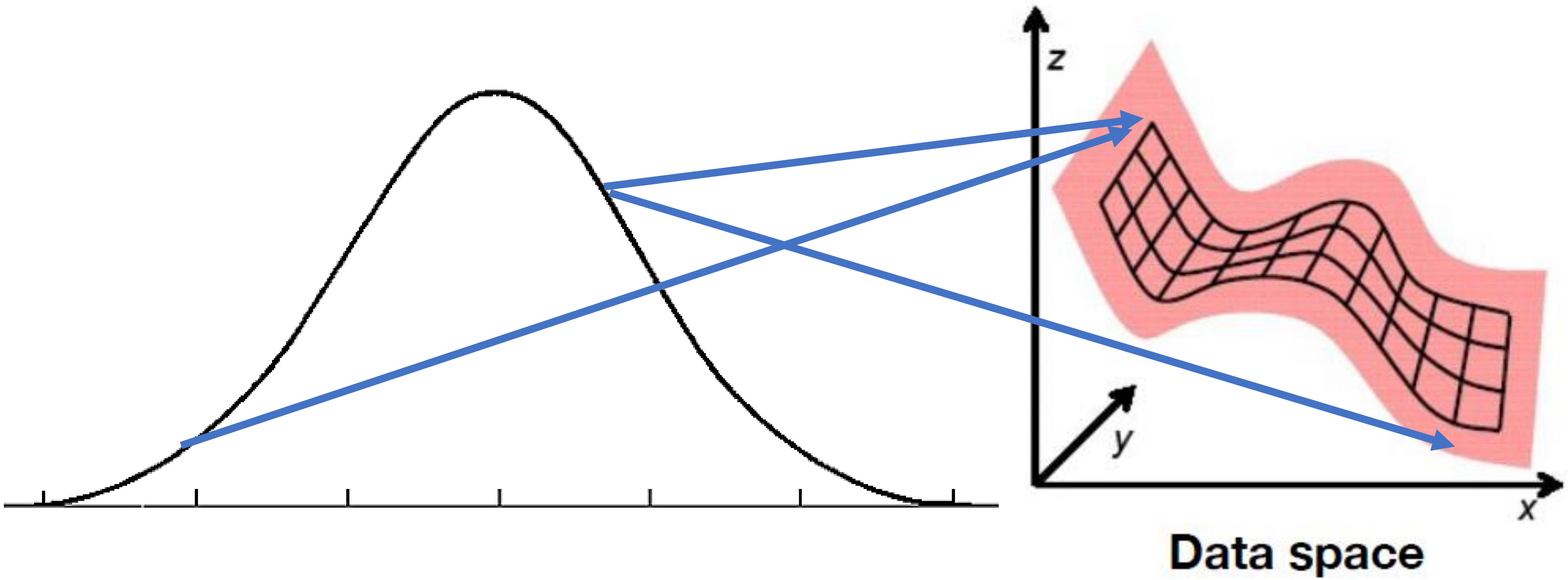
No good!



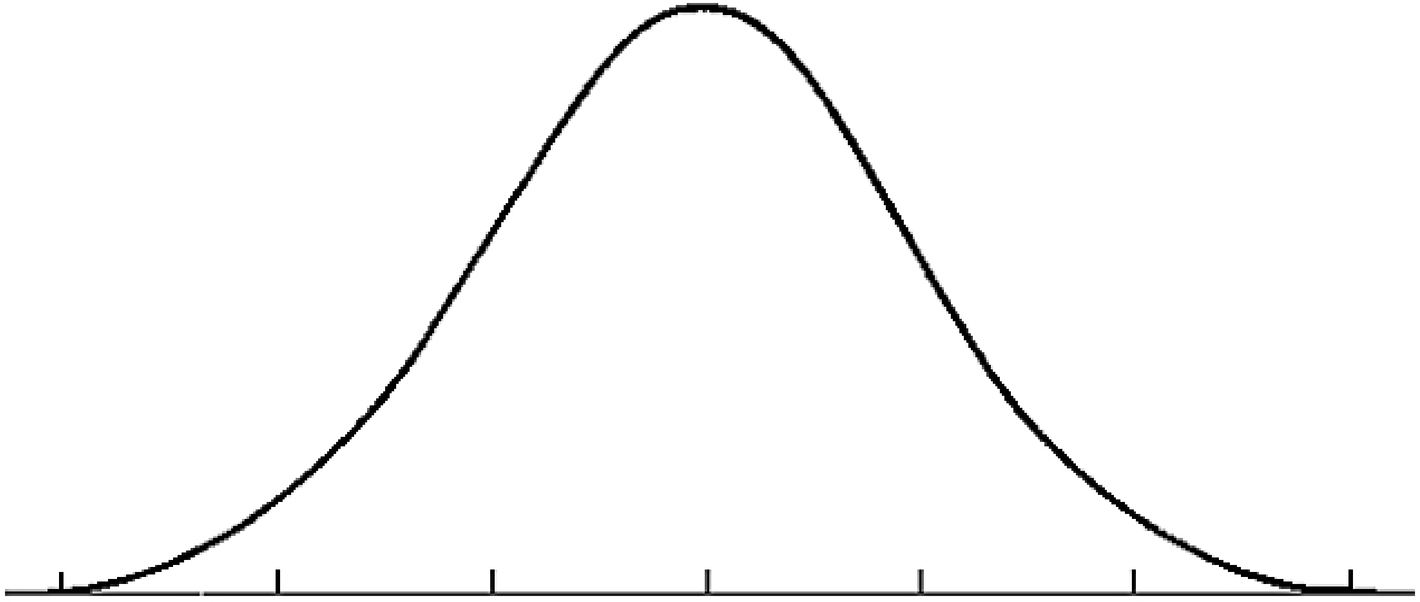
No good!

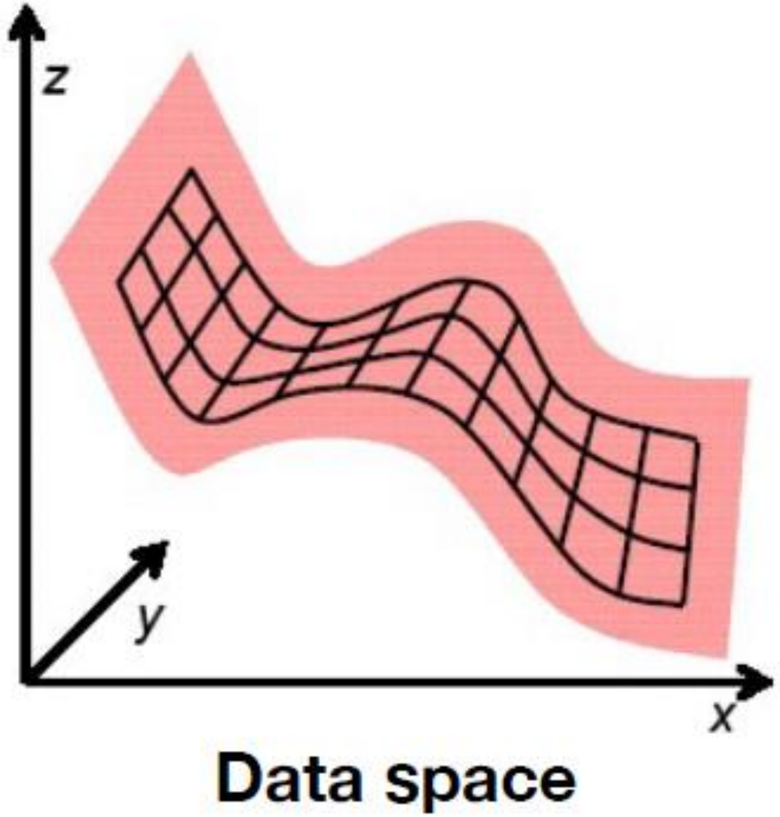


No good!



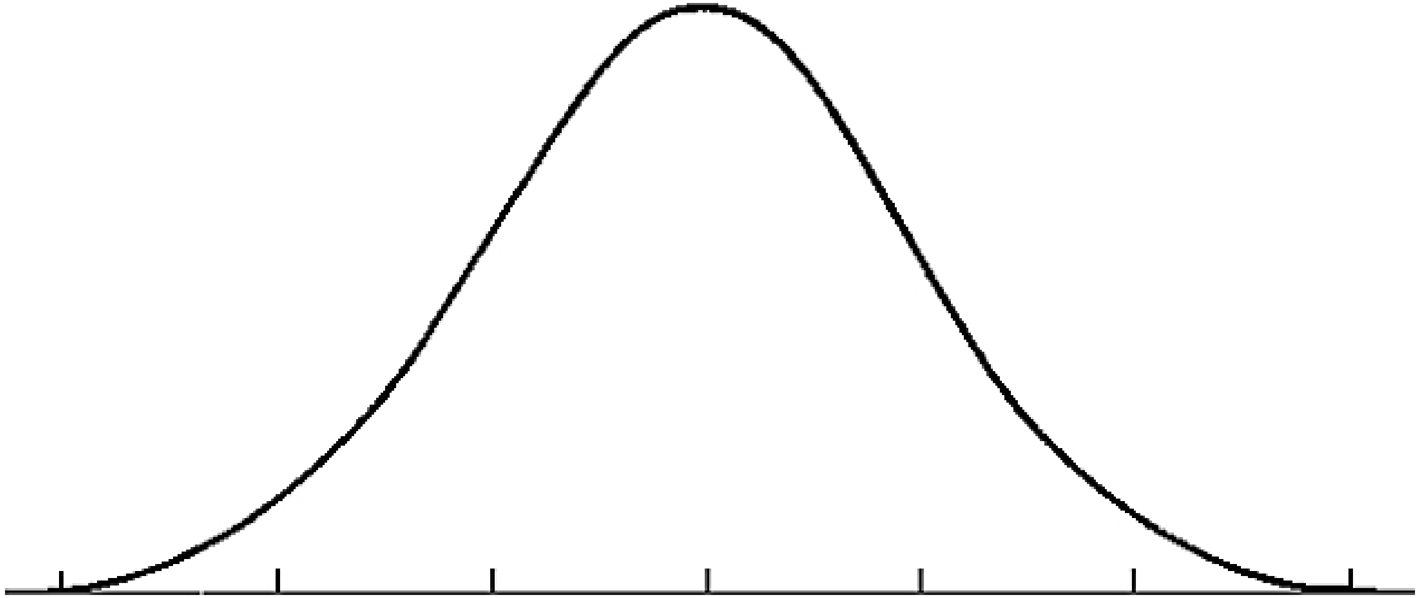
No good!

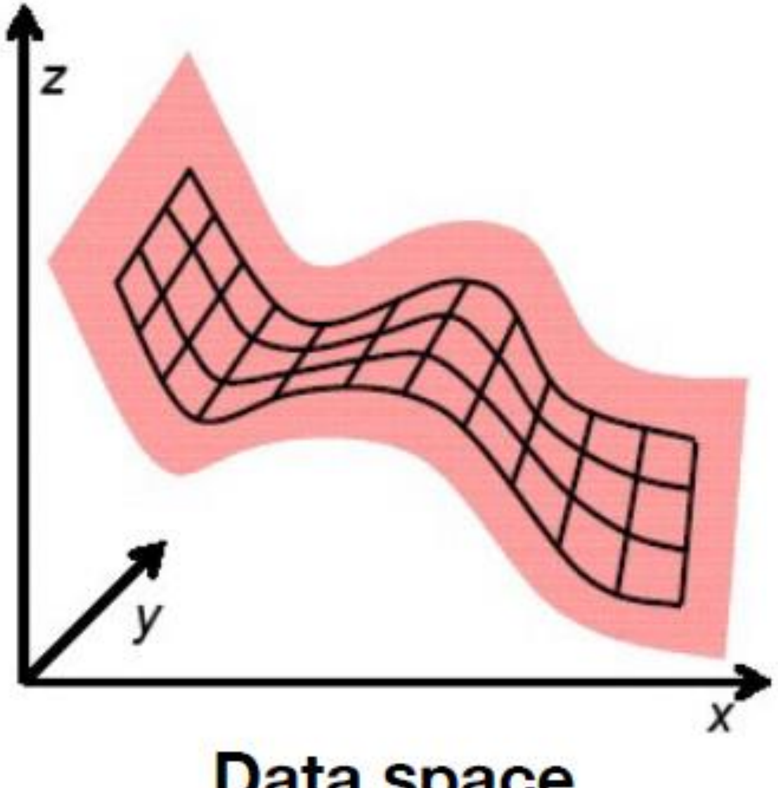




No good!

In expectation: every noise is mapped to every instance

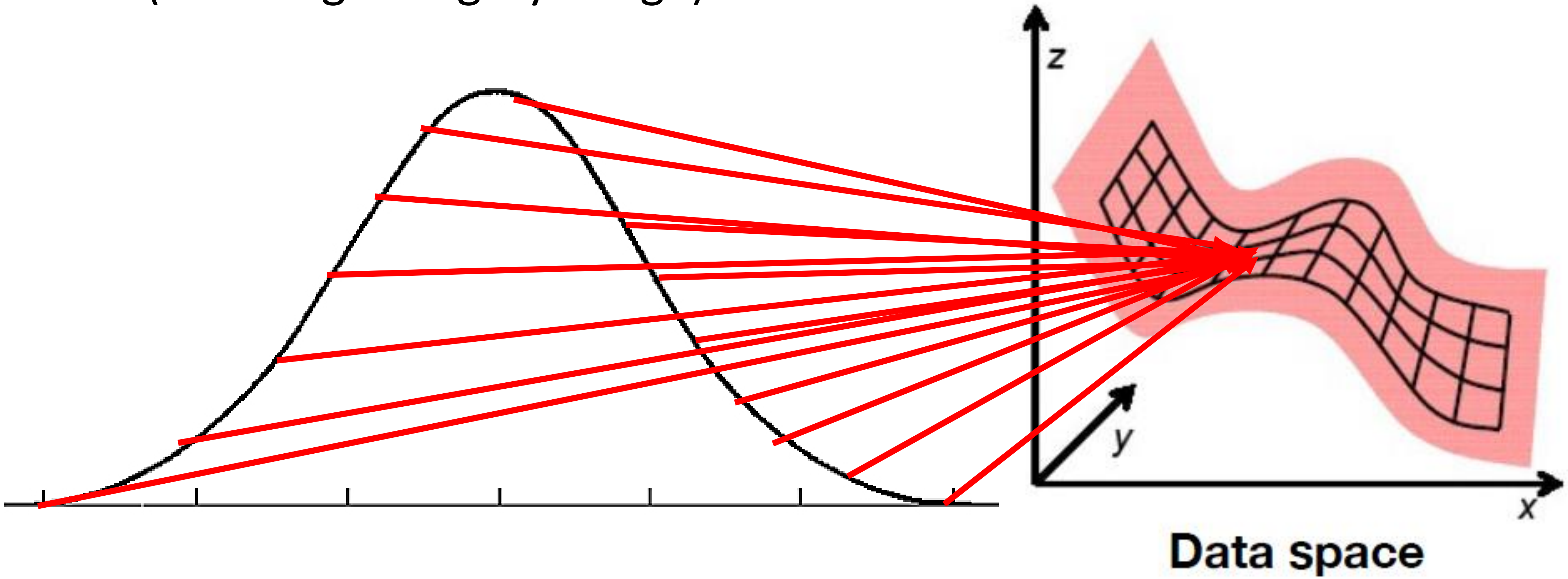




No good!

In expectation: every noise is mapped to every instance

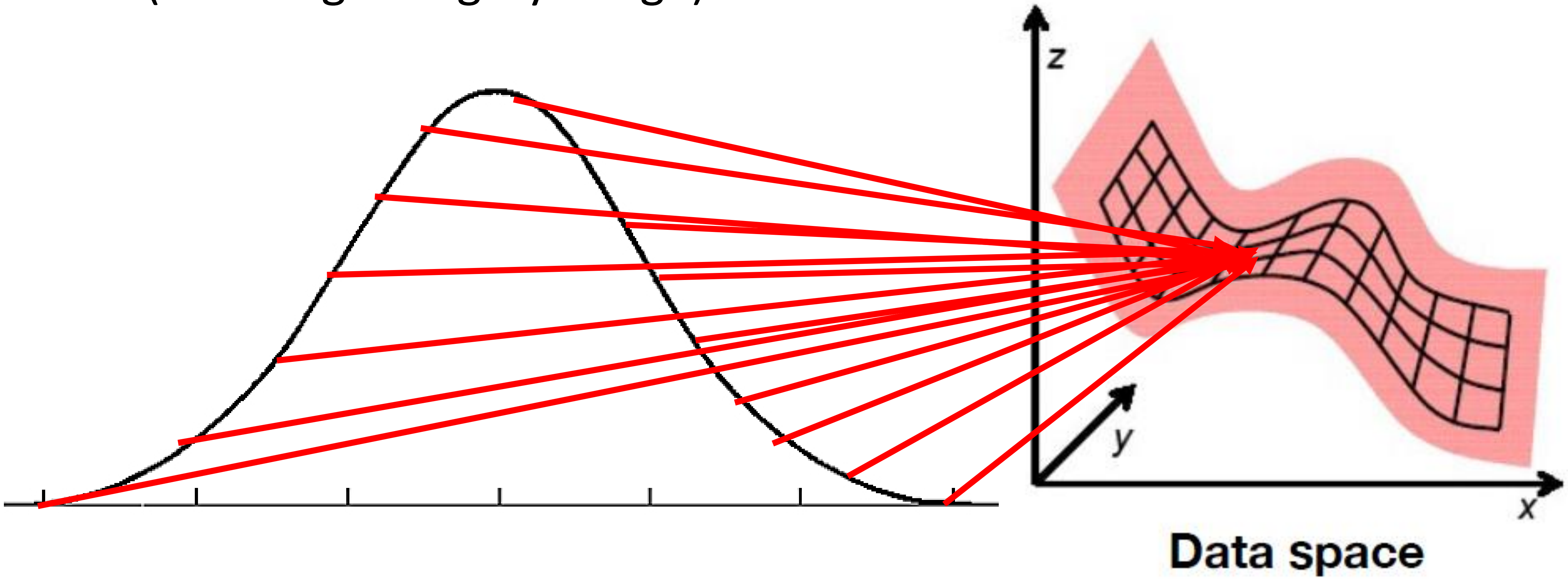
Best L2 solution: All noise is mapped to the mean
(For images: ~ grey image)



No good! **Multimodality not obtained!**

In expectation: every noise is mapped to every instance

Best L2 solution: All noise is mapped to the mean
(For images: ~ grey image)



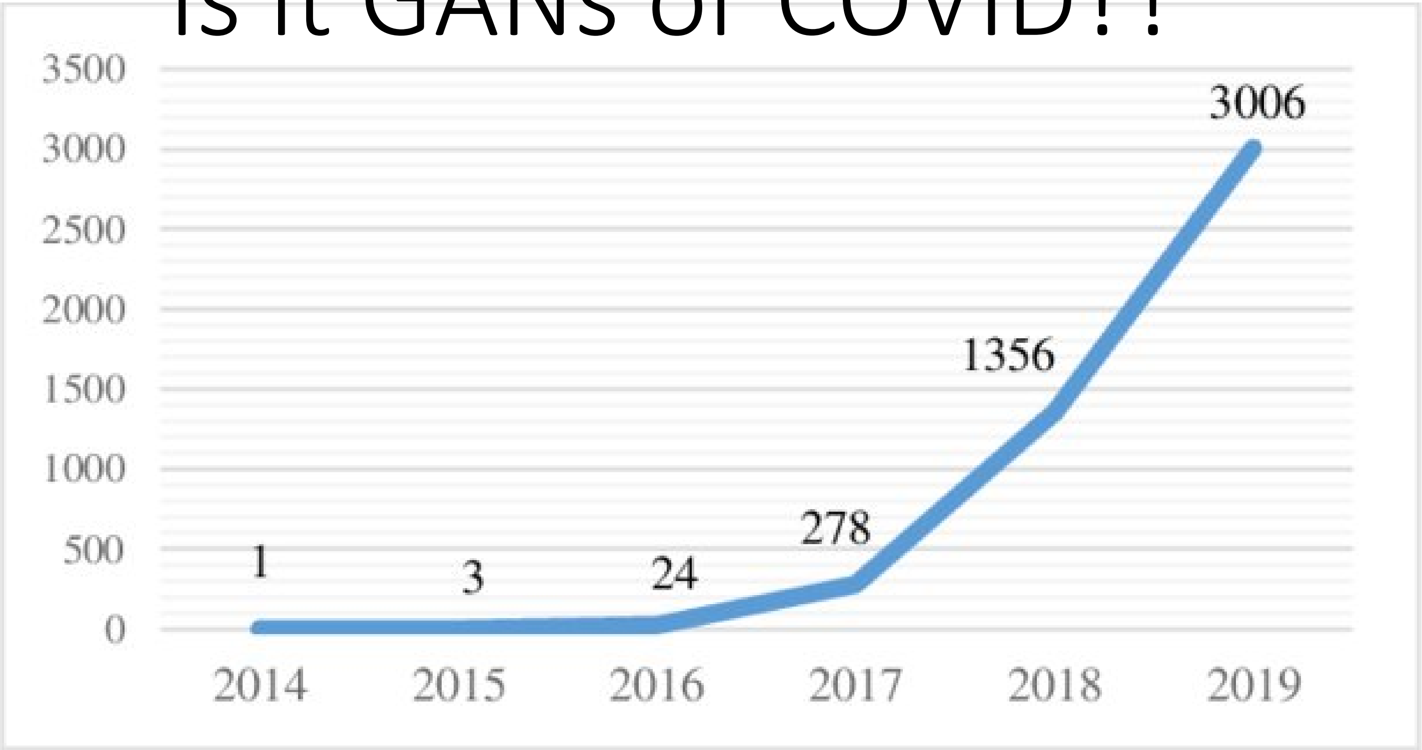
Generative Adversarial Networks

Generative Adversarial Networks

Generative Adversarial Networks

Generative Adversarial Networks

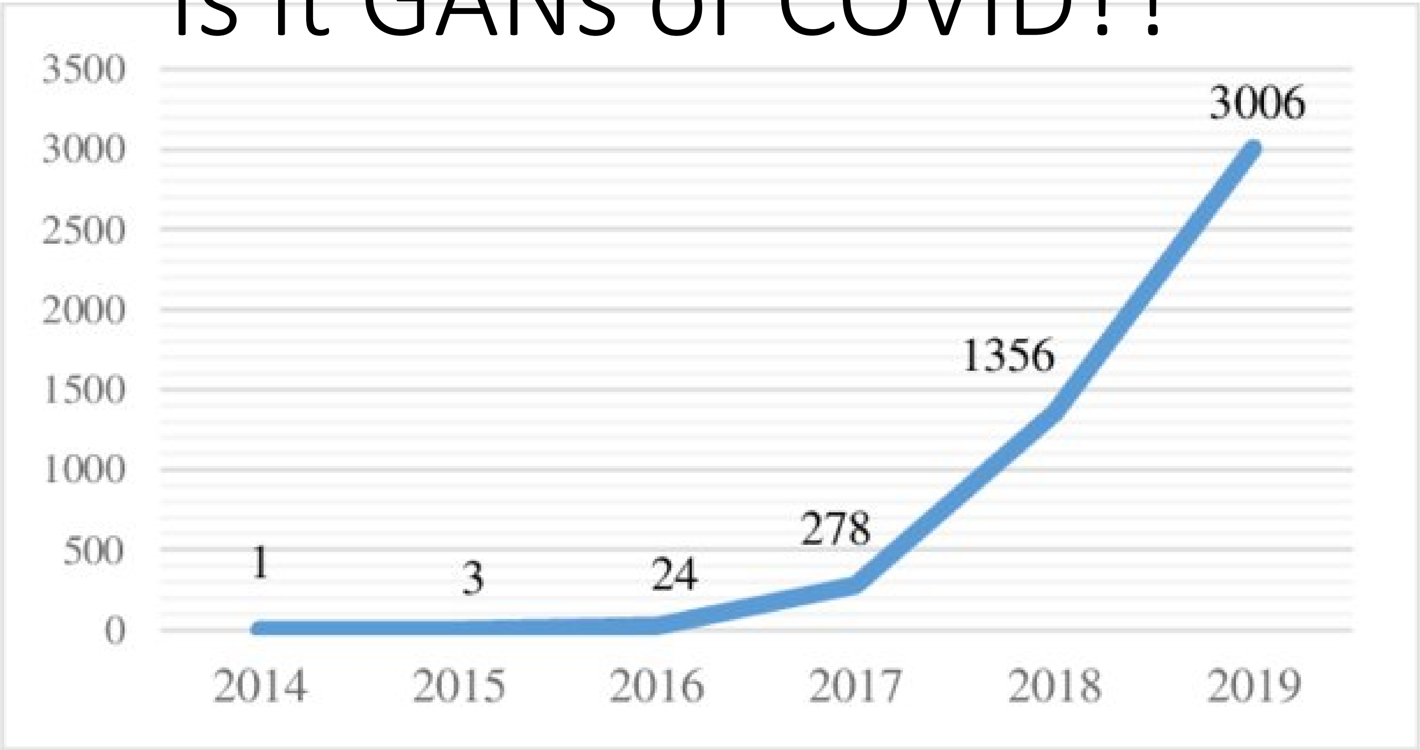
Is it GANs or COVID?!



# of GAN related papers per year
(Salehi et al.)

Generative Adversarial Networks

Is it GANs or COVID?!



# of GAN related papers per year
(Salehi et al.)

Q: What makes a good counterfeiter?

Q: What makes a good counterfeiter?

A: Can fool a good cop

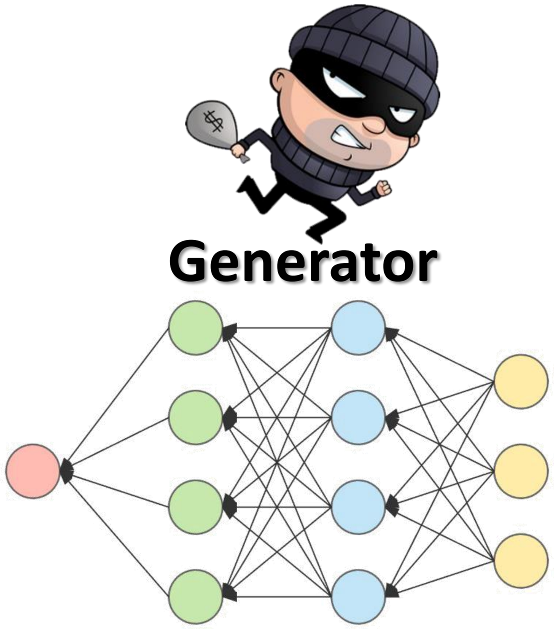
Q: What makes a good counterfeiter?

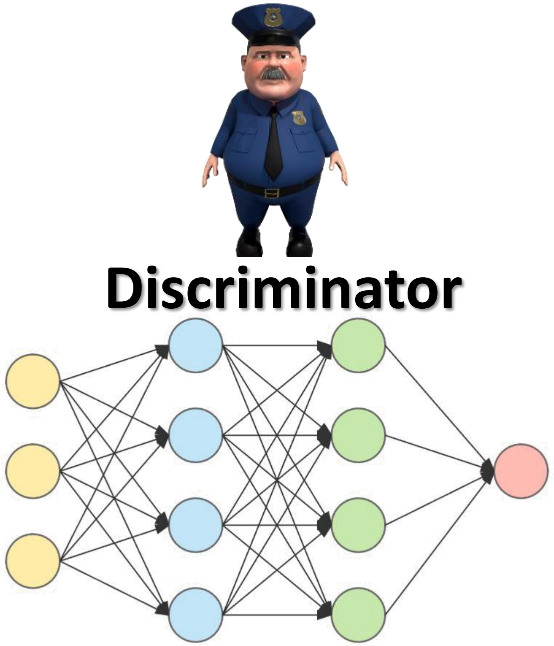
A: Can fool a good cop

Generator

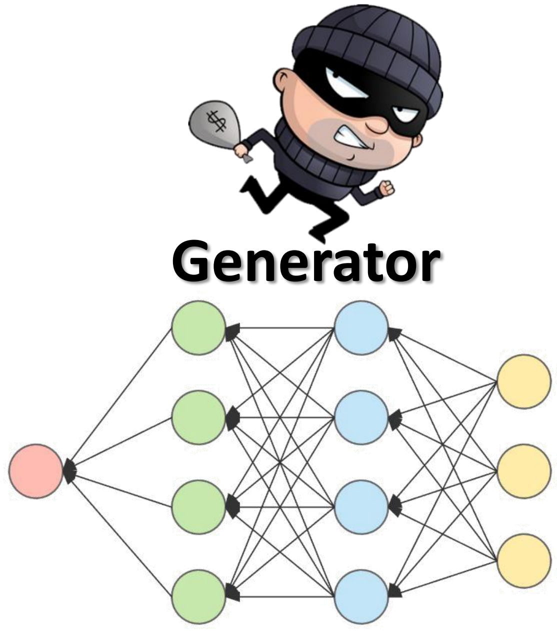
Discriminator

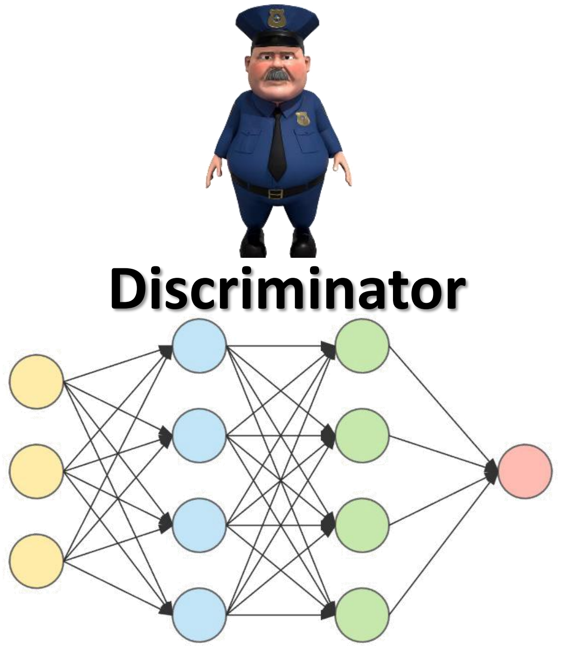
Train D



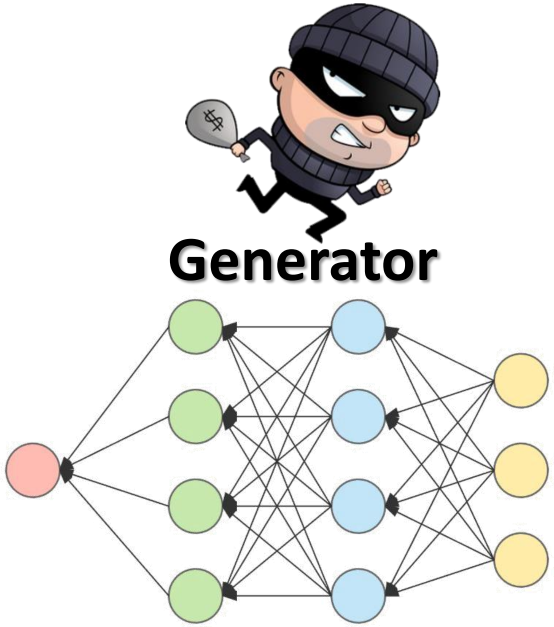


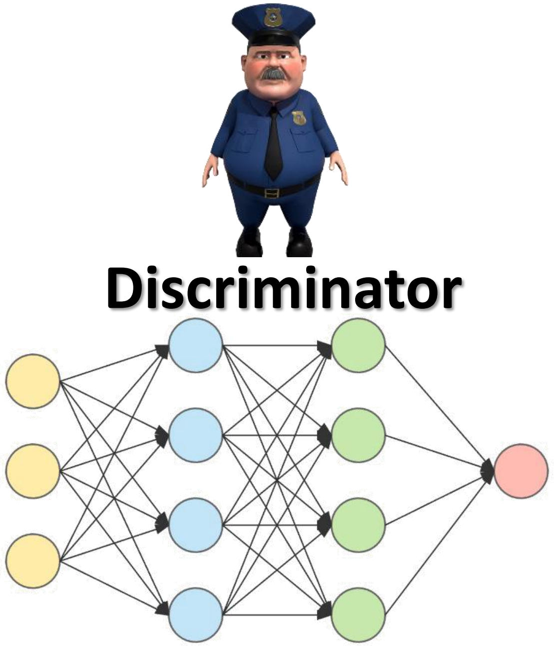
Train D



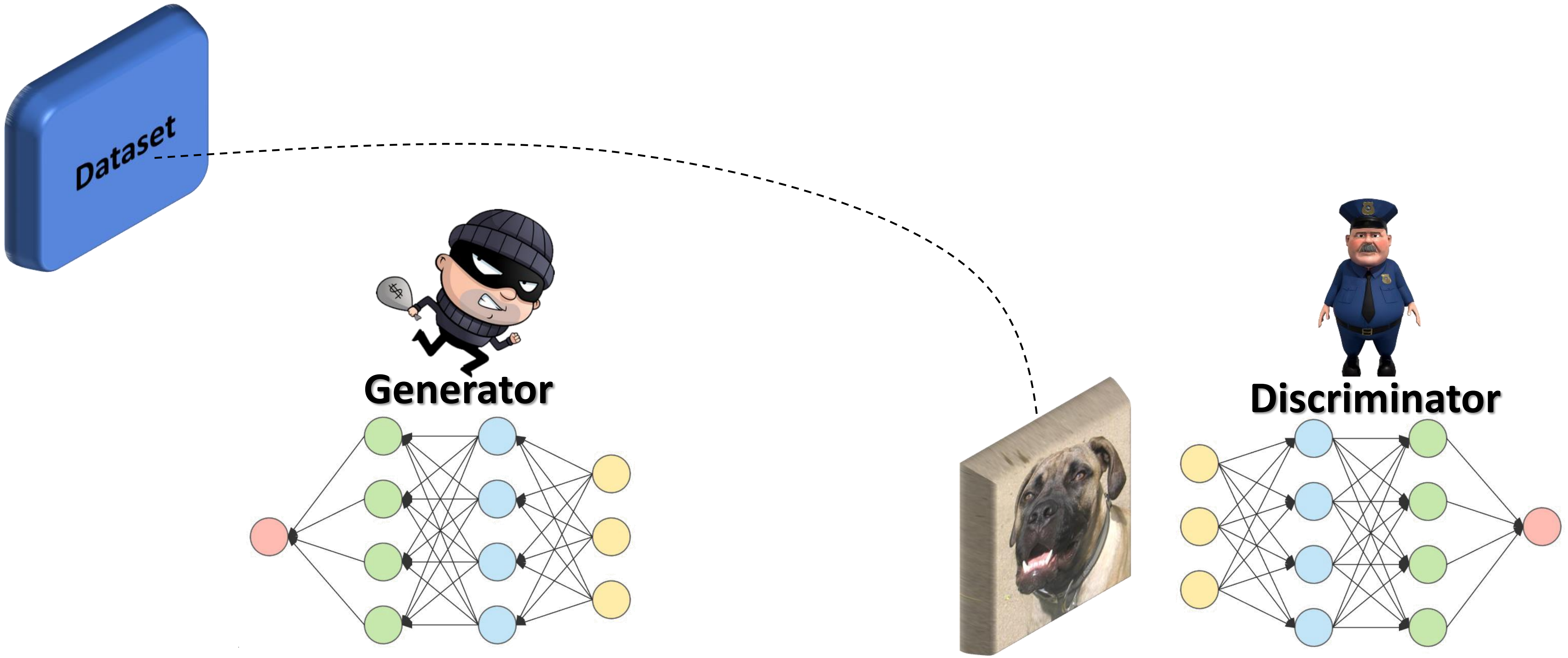


Train D

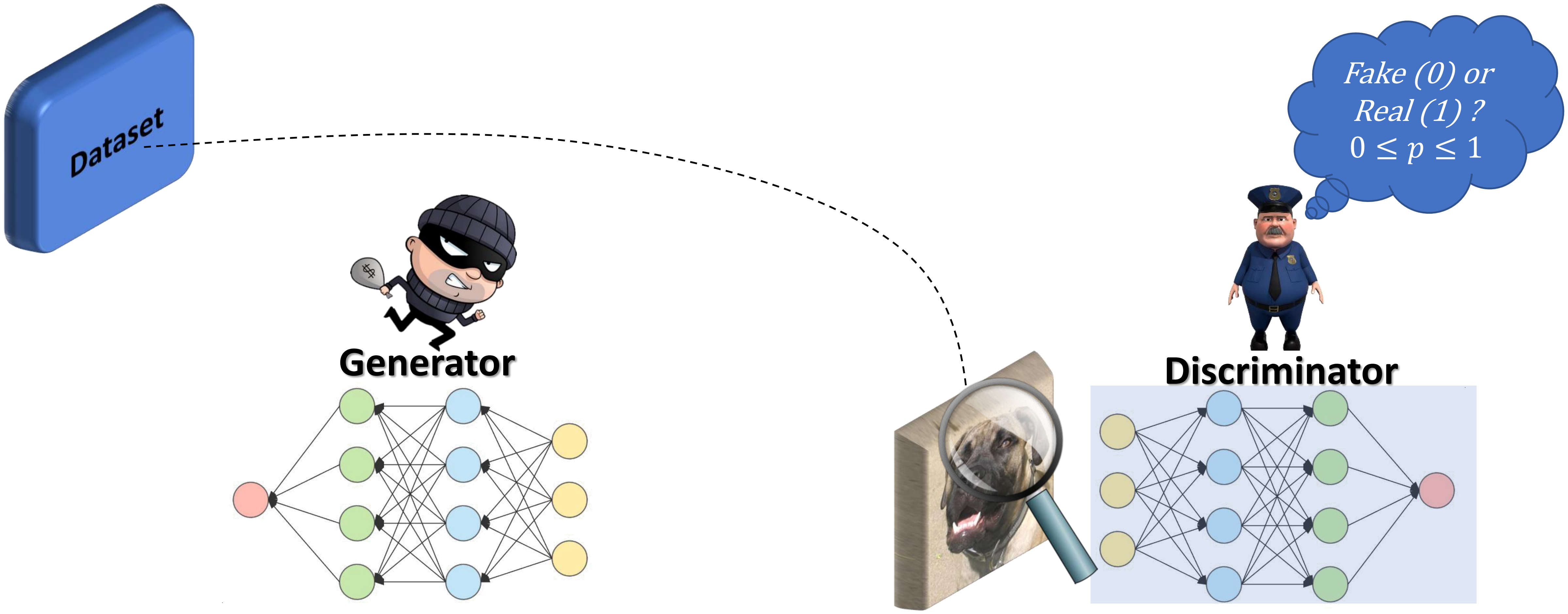




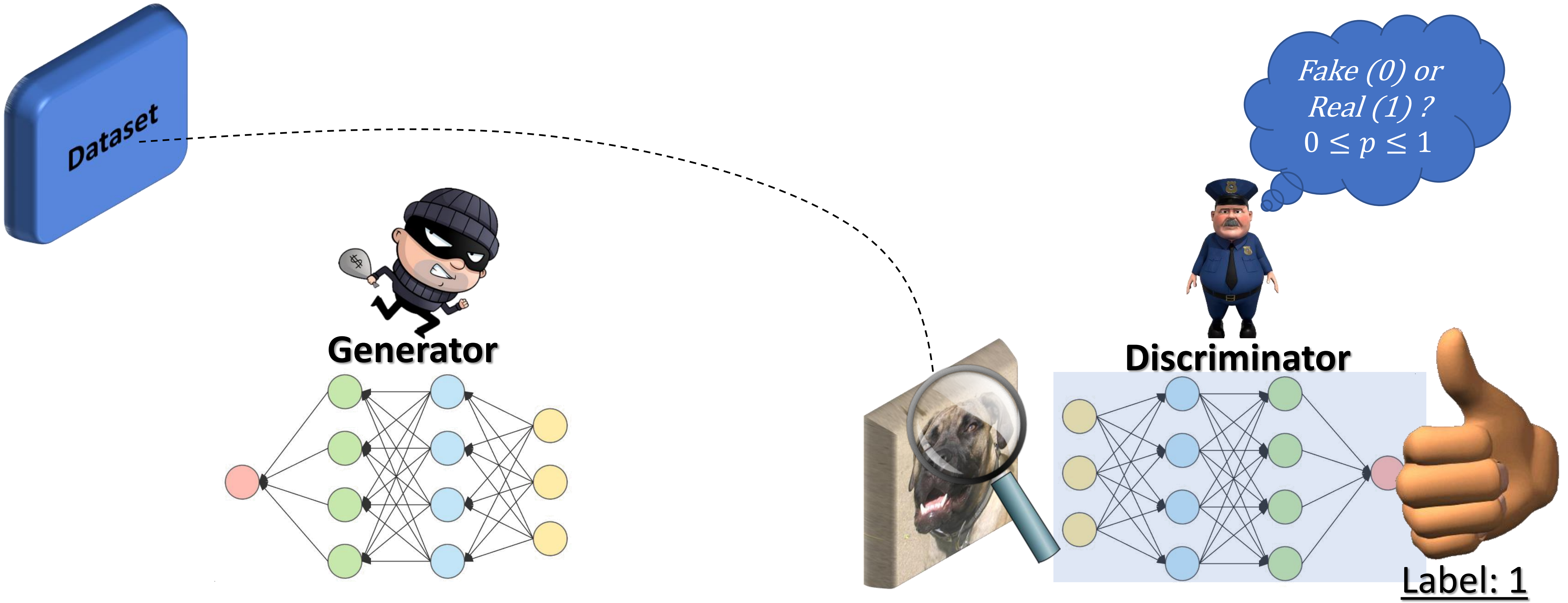
Train D



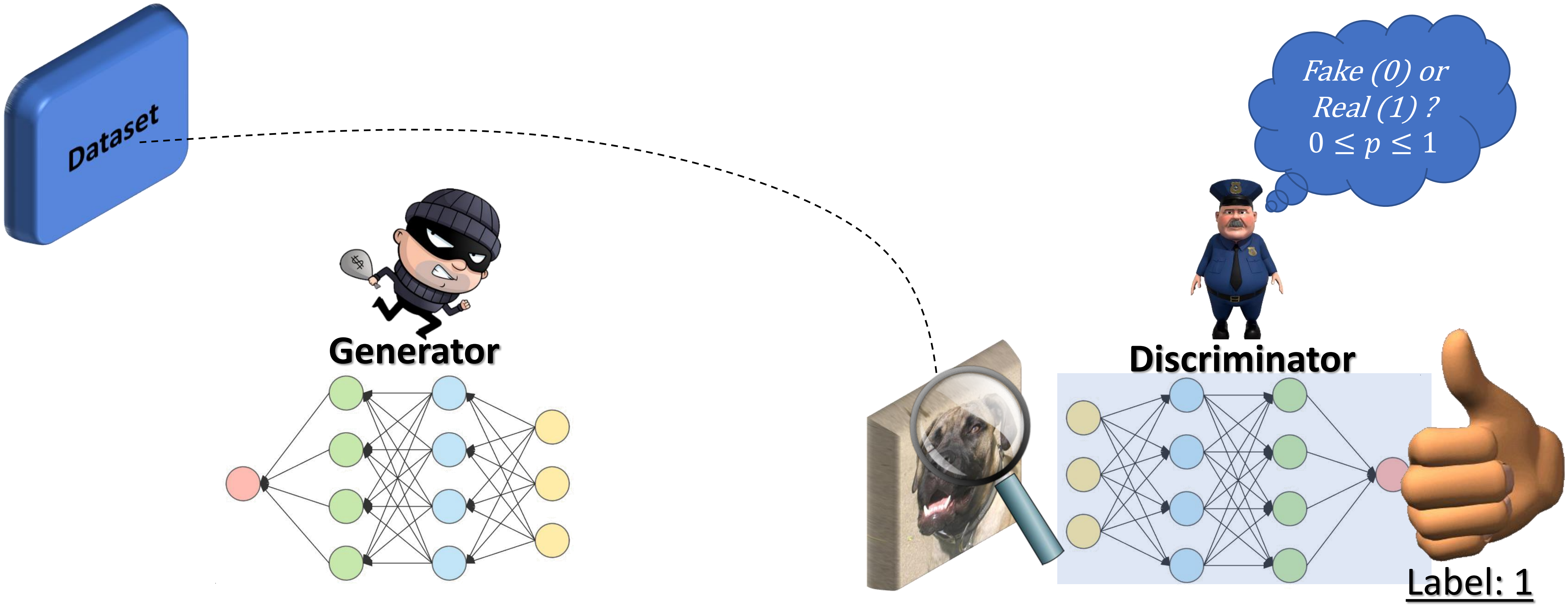
Train D


Dataset
Generator
Discriminator
Fake (0) or
Real (1) ?
$0 \leq p \leq 1$

Train D

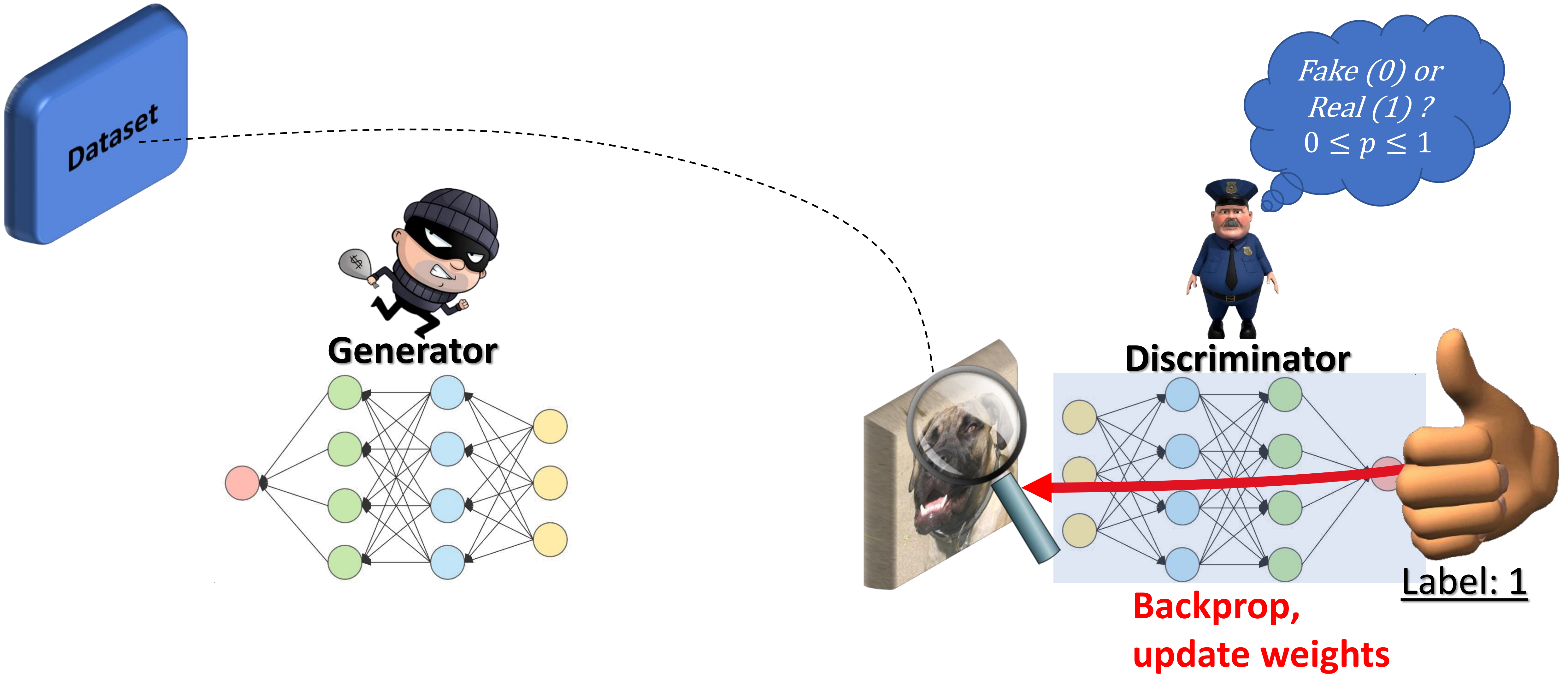

Dataset
Generator
Discriminator
Fake (0) or
Real (1) ?
$0 \leq p \leq 1$
Label: 1

Train D



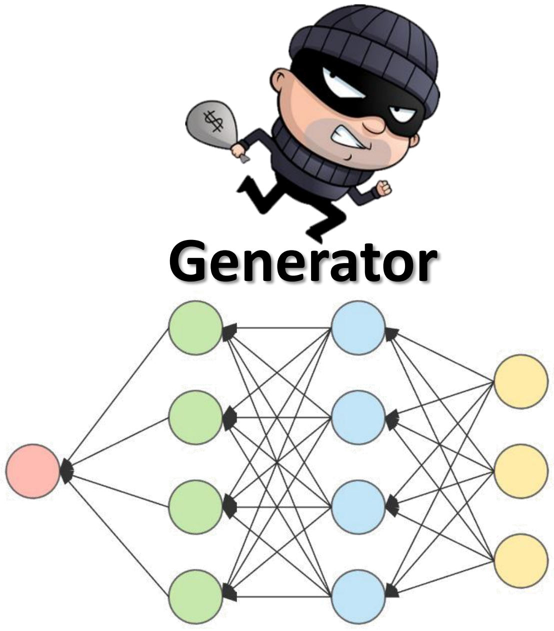
$$\max_D \{\mathbb{E}_{x \sim p_{data}} \log(D(x))\}$$

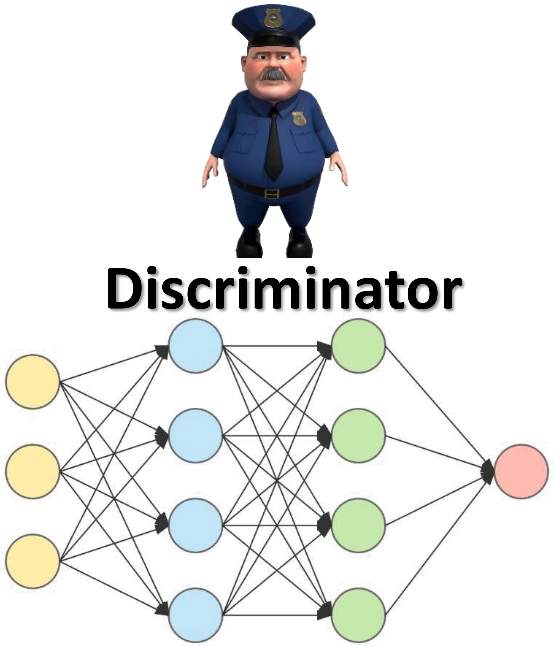
Train D



$$\max_{D}\{\mathbb{E}_{x\sim p_{data}}\log(D(x))\}$$

Train D

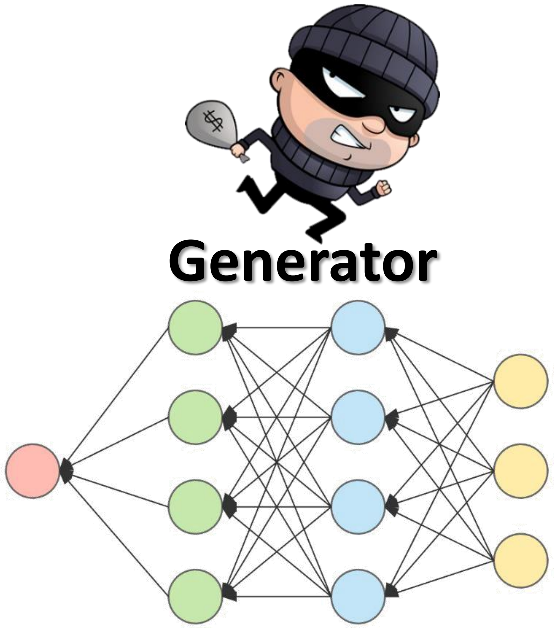


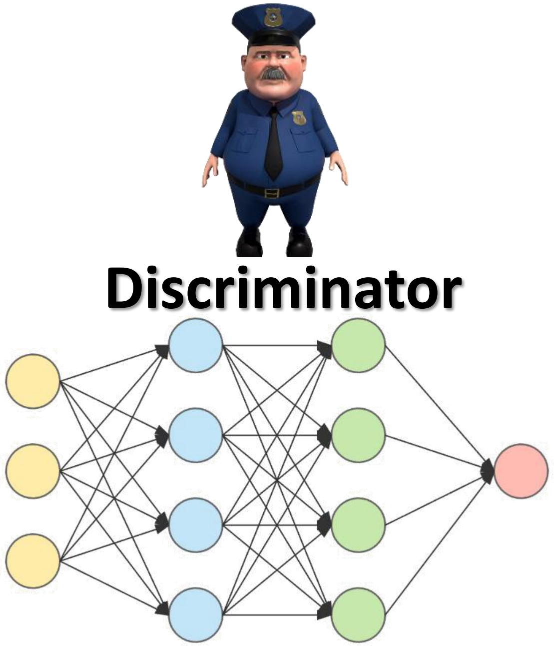


Maximize log likelihood of true examples

$$\max_D \left\{ \mathbb{E}_{x\sim p_{data}} \log(D(x)) \right\}$$

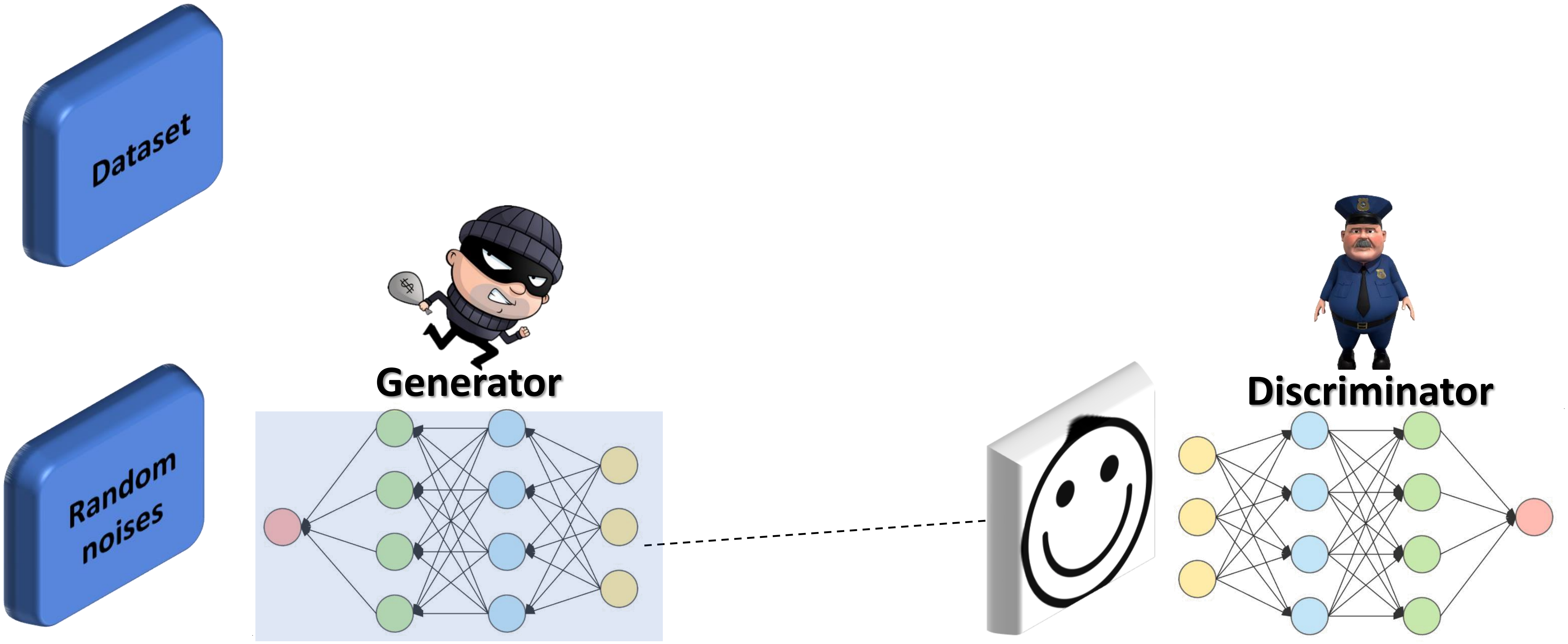
Train D





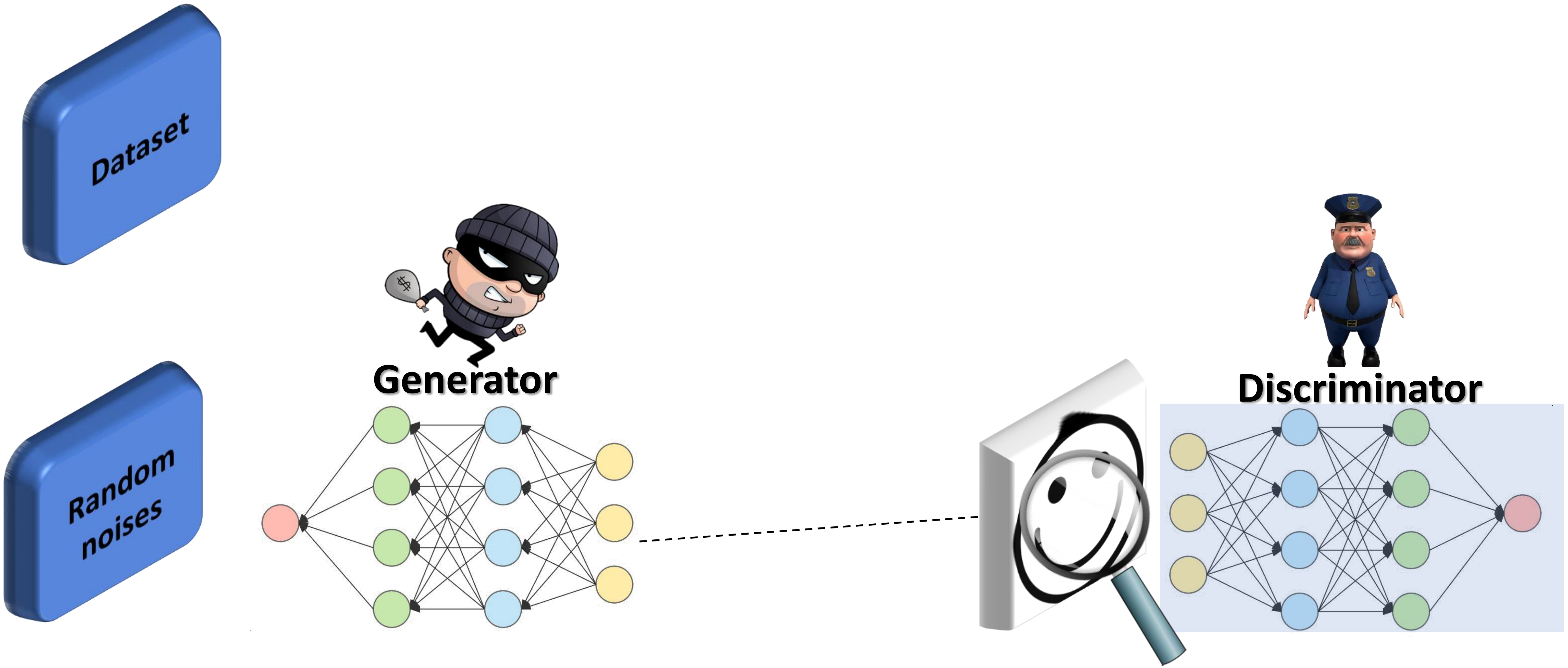
$$\max_D \{\mathbb{E}_{x \sim p_{data}} \log(D(x))\}$$

Train D



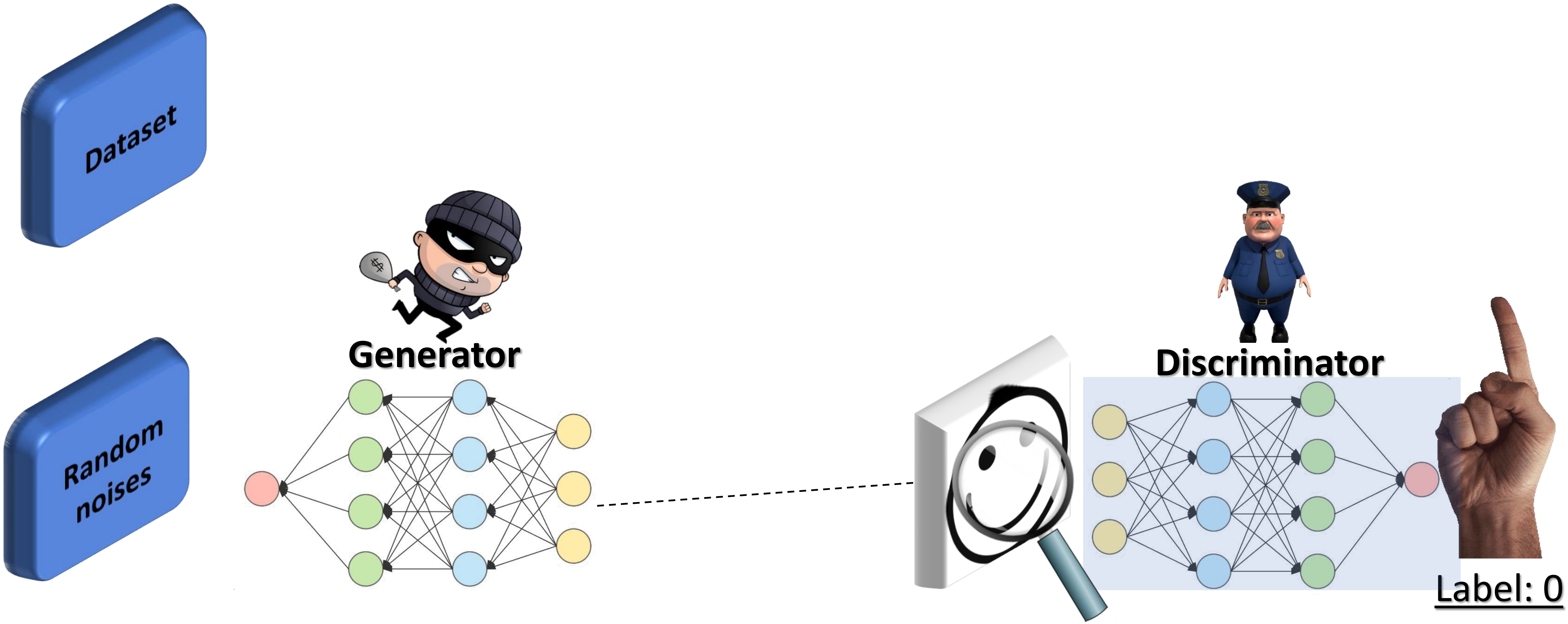
$$\max_{D}\{\mathbb{E}_{x\sim p_{data}}\log(D(x))\}$$

Train D



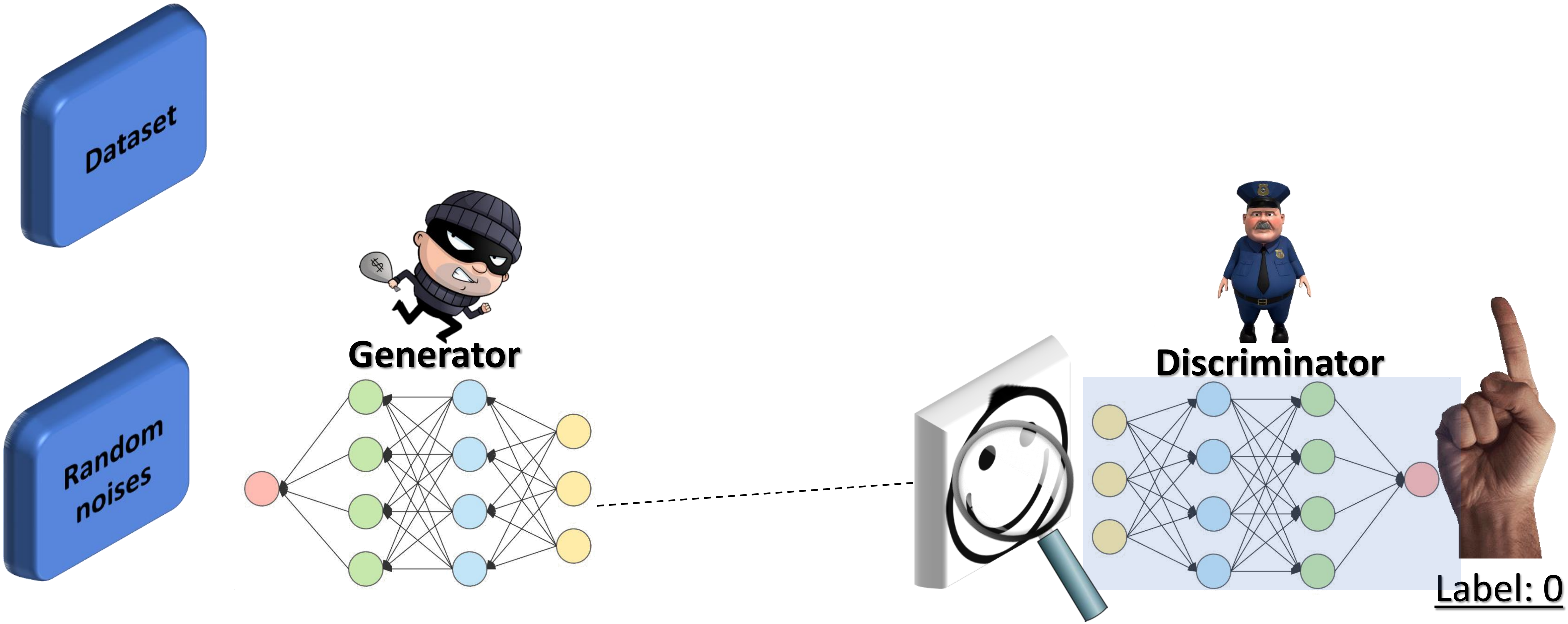
$$\max_{D}\{\mathbb{E}_{x\sim p_{data}}\log(D(x))\}$$

Train D



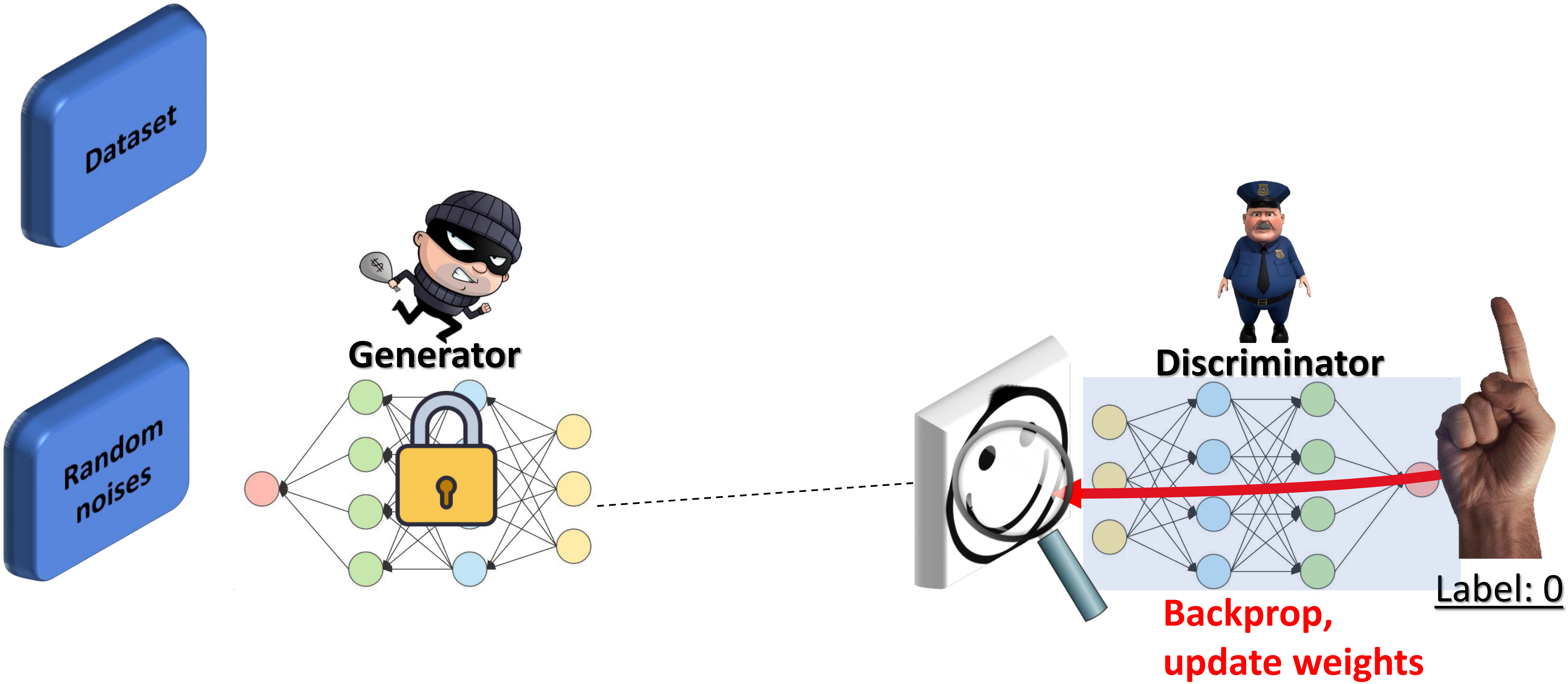
$$\max_{D}\{\mathbb{E}_{x\sim p_{data}}\log(D(x))\}$$

Train D



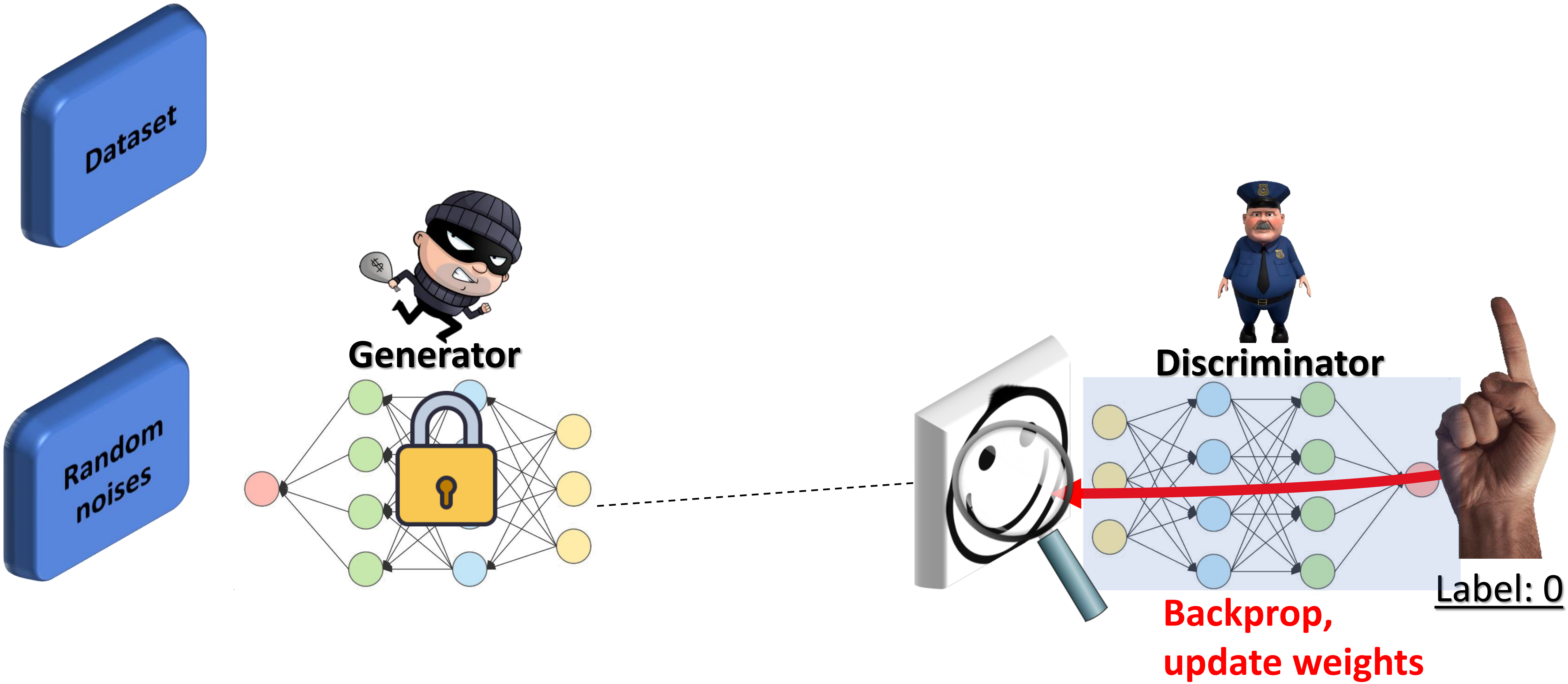
$$\max_{D}\{\mathbb{E}_{x\sim p_{data}}\log(D(x)) + \mathbb{E}_{z\sim p_z}\log(1-D(G(z)))\}$$

Train D



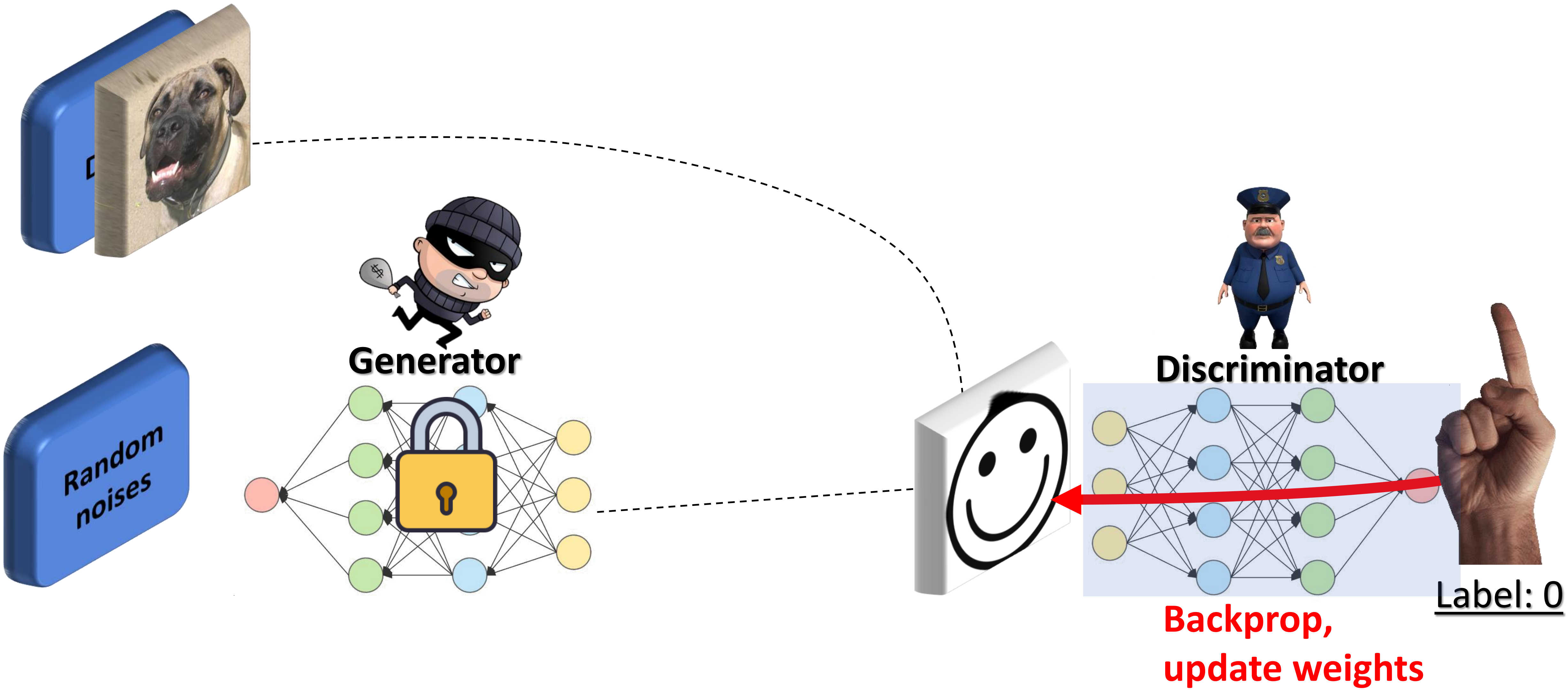
$$\max_{D}\{\mathbb{E}_{x\sim p_{data}}\log(D(x)) + \mathbb{E}_{z\sim p_z}\log(1 - D(G(z)))\}$$

Train D

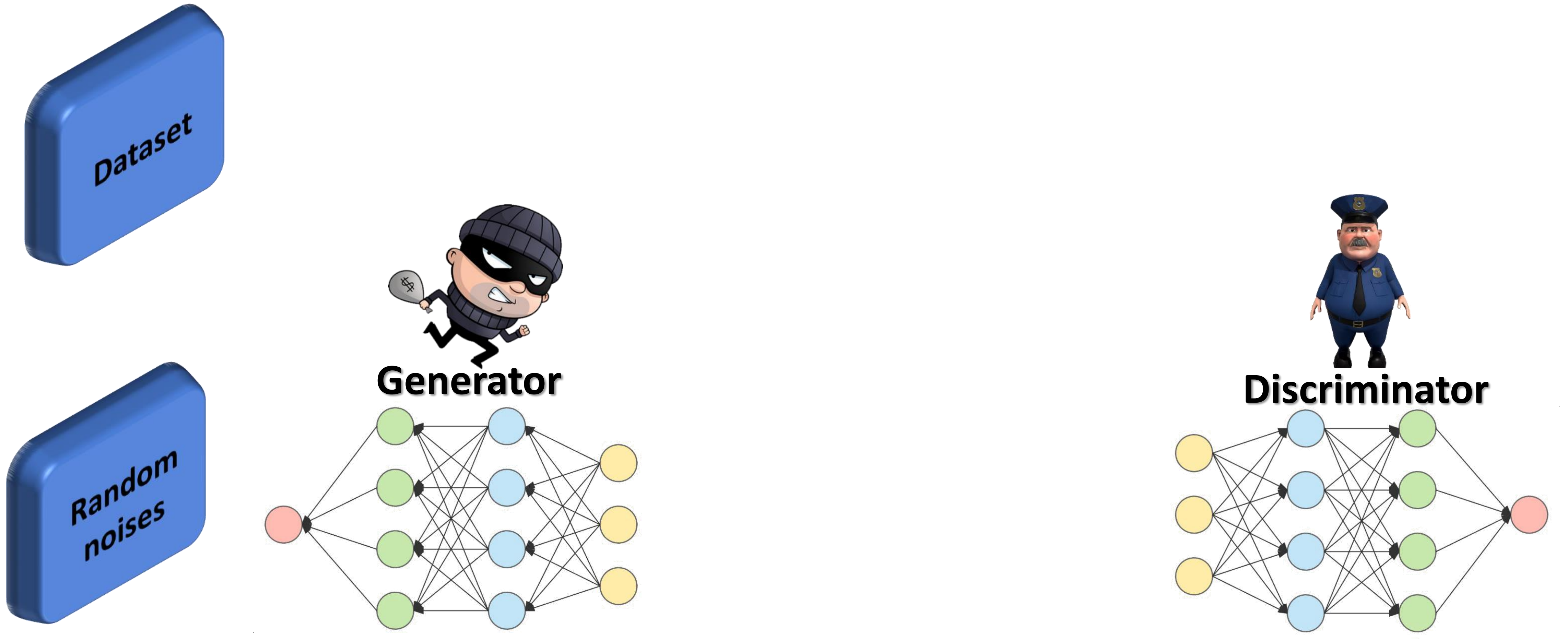


$$\max_{D}\{\mathbb{E}_{x\sim p_{data}}\log(D(x)) + \mathbb{E}_{z\sim p_z}\log(1-D(G(z)))\}$$

Train D

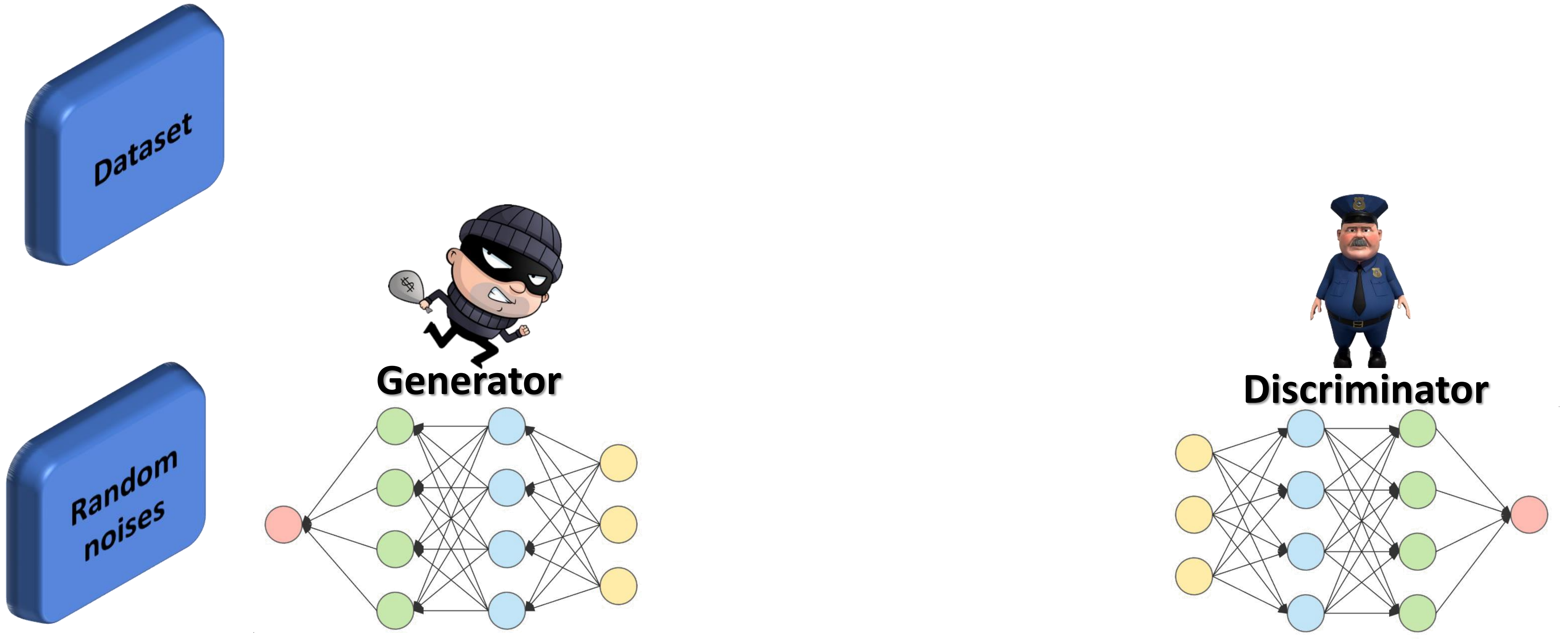


$$\max_{D}\{\mathbb{E}_{x\sim p_{data}}\log(D(x)) + \mathbb{E}_{z\sim p_z}\log(1-D(G(z)))\}$$



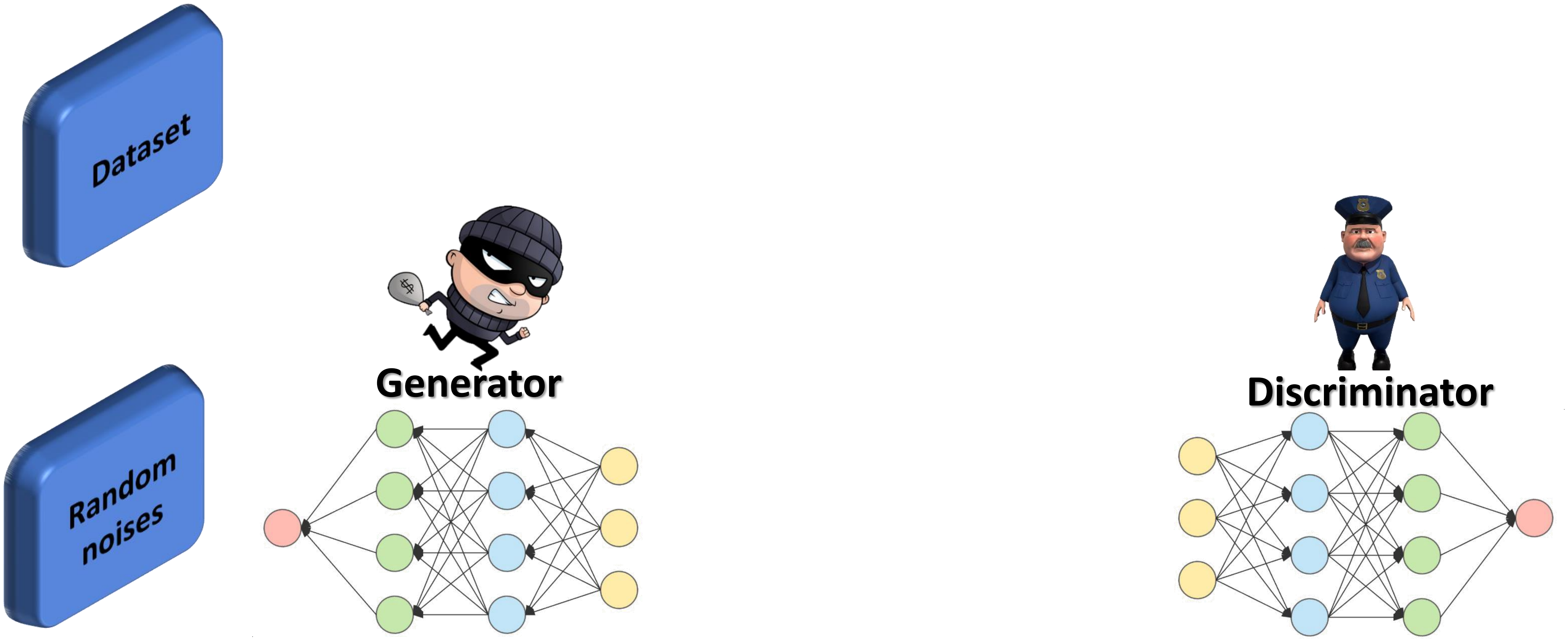
$$\max_{D}\{\mathbb{E}_{x\sim p_{data}}\log(D(x)) + \mathbb{E}_{z\sim p_z}\log(1 - D(G(z)))\}$$

Train G



$$\max_{D}\{\mathbb{E}_{x\sim p_{data}}\log(D(x)) + \mathbb{E}_{z\sim p_z}\log(1 - D(G(z)))\}$$

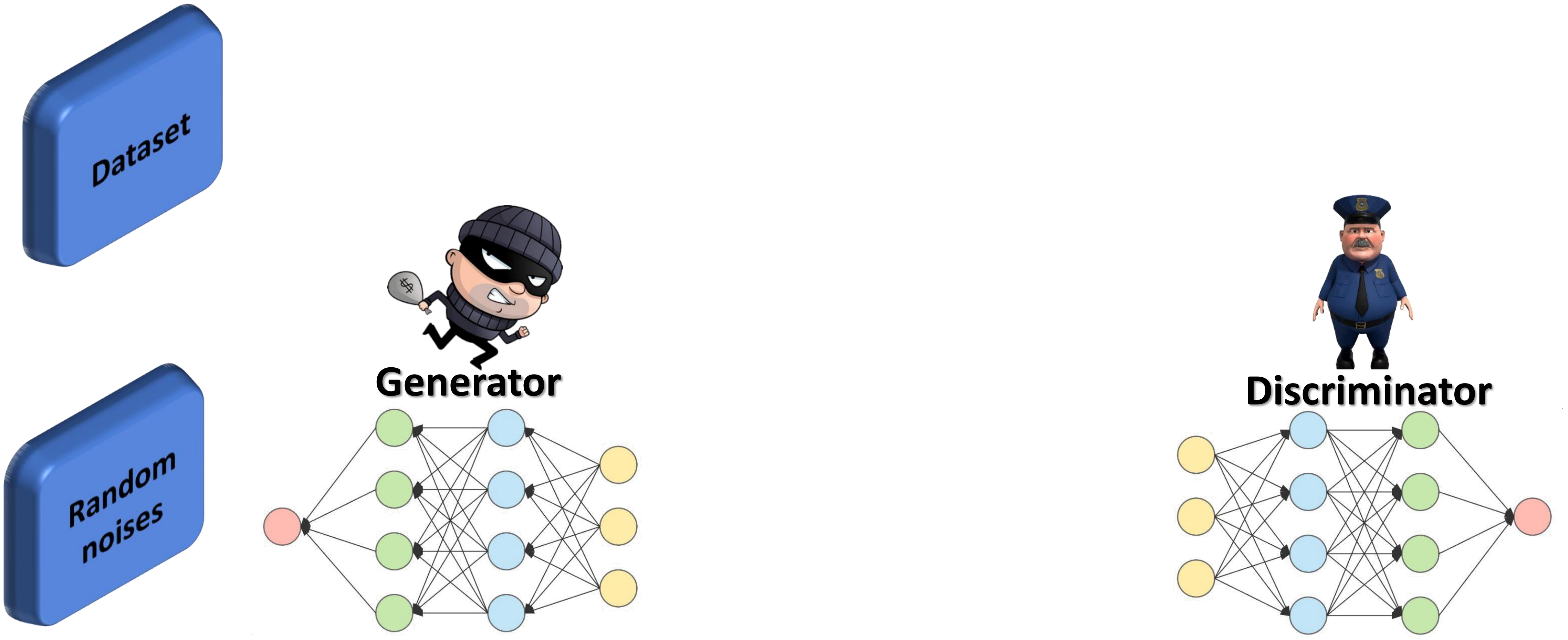
Train G



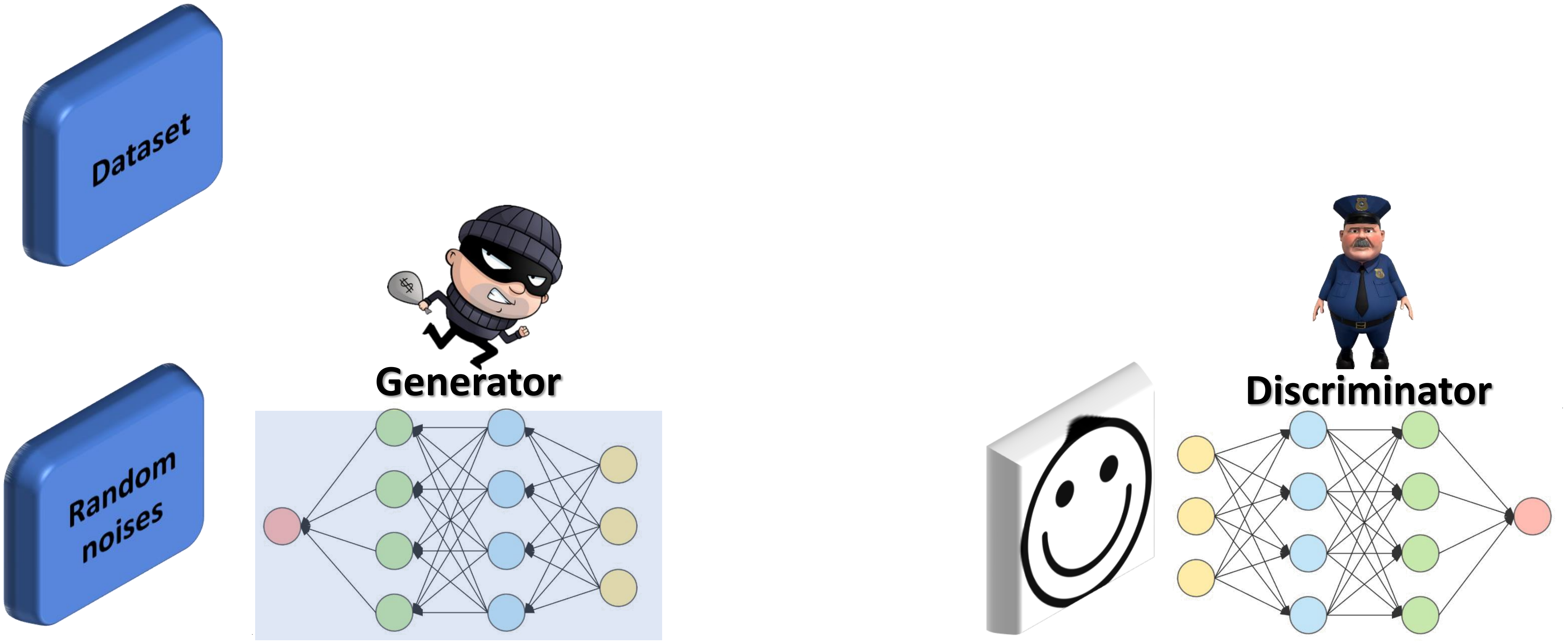
Minimax game: Make the best cop do the worst mistake!

$$\min_{G} \max_{D} \{ \mathbb{E}_{x \sim p_{data}} \log(D(x)) + \mathbb{E}_{z \sim p_z} log(1 - D(G(z))) \}$$

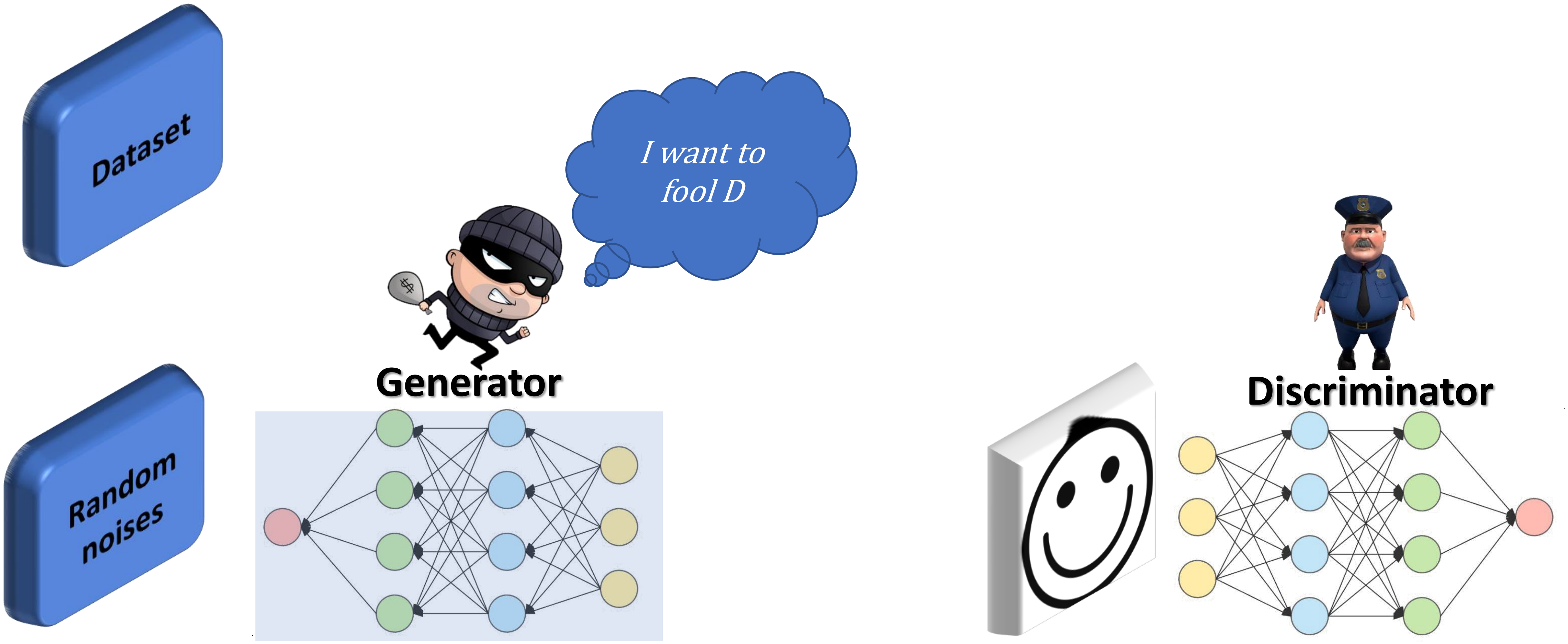
Train G



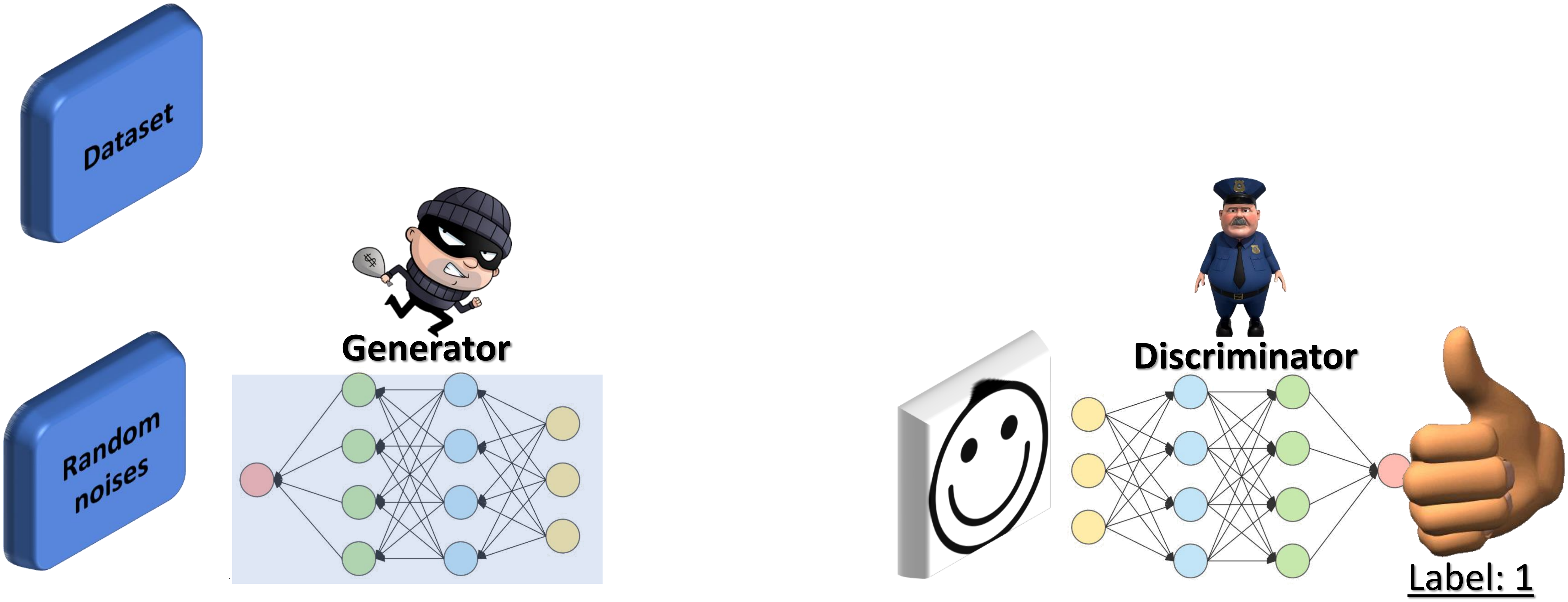
$$\min_{G} \max_{D} \{ \mathbb{E}_{x \sim p_{data}} \log(D(x)) + \mathbb{E}_{z \sim p_z} log(1 - D(G(z))) \}$$



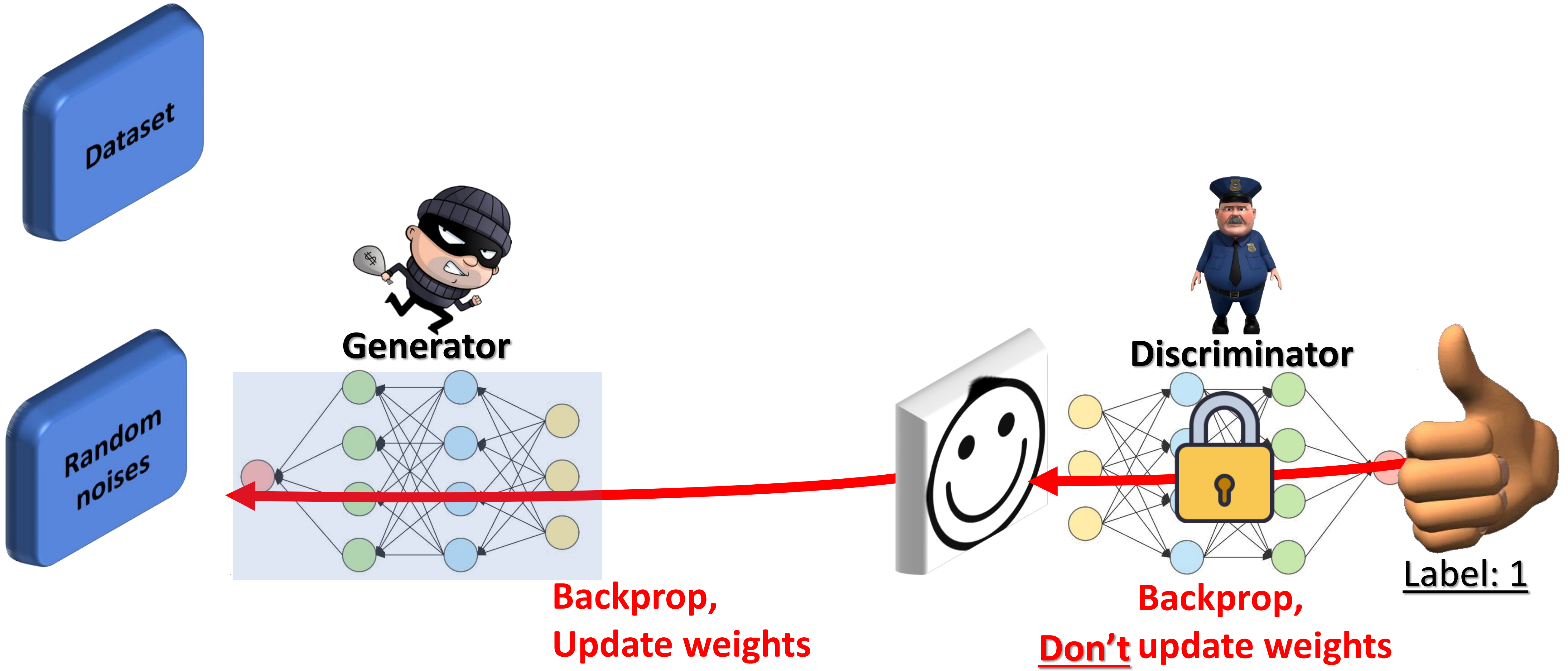
$$\min_{G} \max_{D} \{ \mathbb{E}_{x \sim p_{data}} \log(D(x)) + \mathbb{E}_{z \sim p_z} log(1 - D(G(z))) \}$$



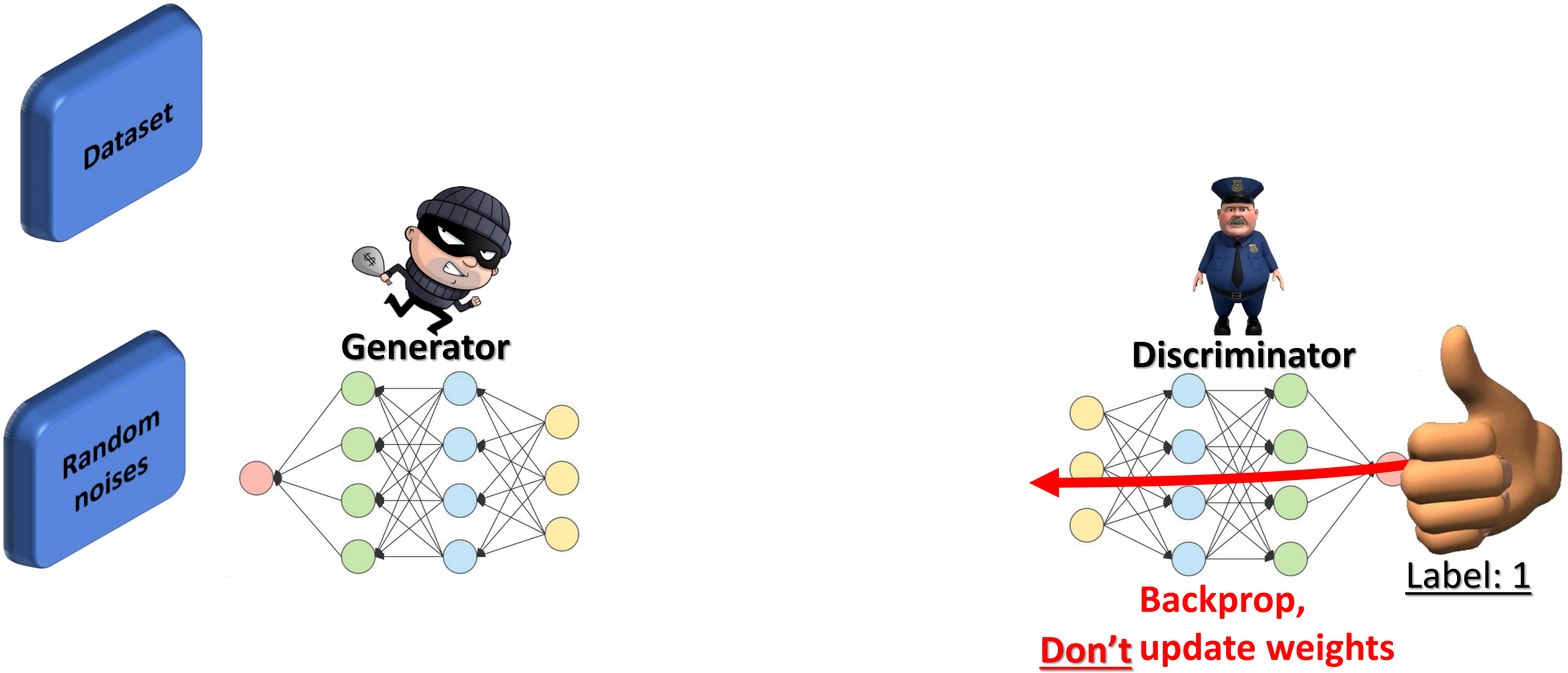
$$\min_{G} \max_{D} \{\mathbb{E}_{x \sim p_{data}} \log(D(x)) + \mathbb{E}_{z \sim p_z} log(1 - D(G(z)))\}$$



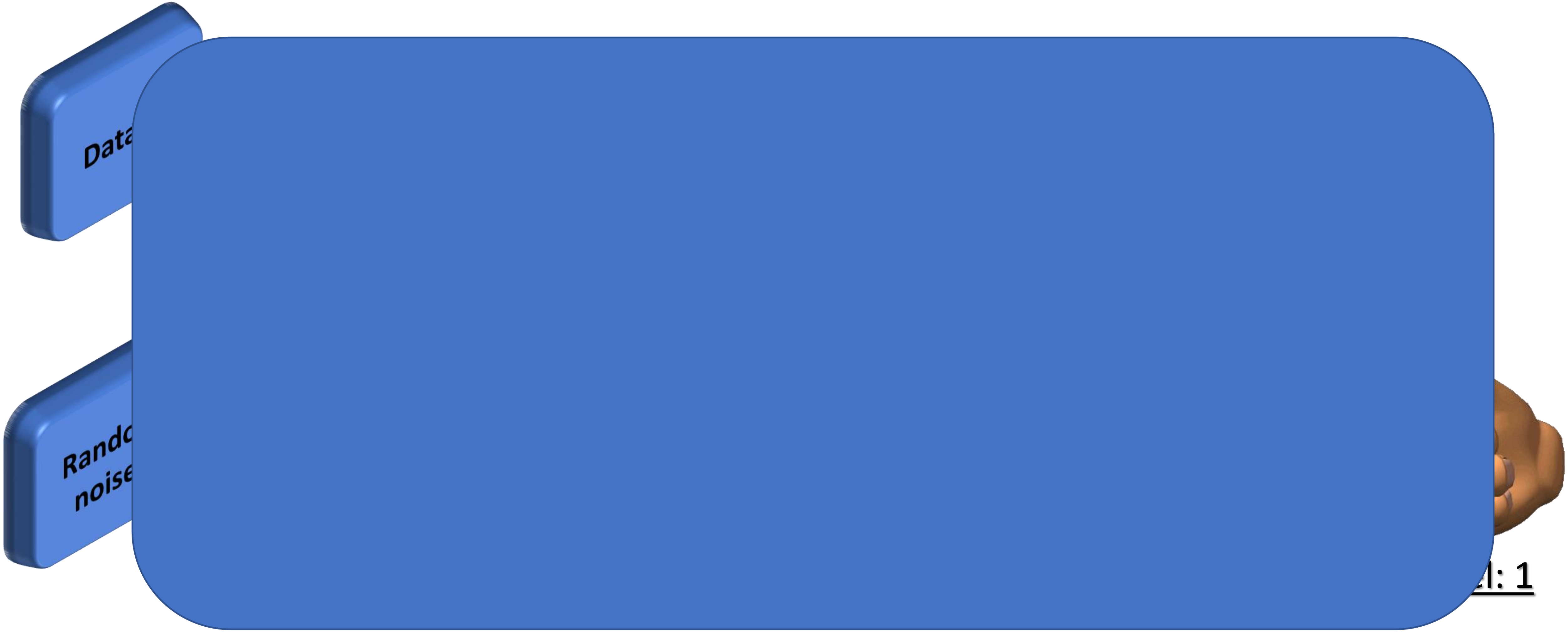
$$\min_{G} \max_{D} \left\{ \mathbb{E}_{x\sim p_{data}} \log(D(x)) + \mathbb{E}_{z\sim p_z} log(1 - D(G(z))) \right\}$$



$$\min_{G} \max_{D} \{\mathbb{E}_{x\sim p_{data}} \log(D(x)) + \mathbb{E}_{z\sim p_z} log(1 - D(G(z)))\}$$



$$\min_{G} \max_{D} \{\mathbb{E}_{x \sim p_{data}} \log(D(x)) + \mathbb{E}_{z \sim p_z} log(1 - D(G(z)))\}$$



Don't update weights

$$\min_{G} \max_{D} \left\{ \mathbb{E}_{x \sim p_{data}} \log(D(x)) + \mathbb{E}_{z \sim p_z} log(1 - D(G(z))) \right\}$$

Q: Who do you train first?

Don't **update weights**

$$\min_{G} \max_{D} \{ \mathbb{E}_{x \sim p_{data}} \log(D(x)) + \mathbb{E}_{z \sim p_z} log(1 - D(G(z))) \}$$

Q: Who do you train first?

A: Alternate training! G,D,G,D....

Don't update weights

$$\min_{G} \max_{D} \{ \mathbb{E}_{x \sim p_{data}} \log(D(x)) + \mathbb{E}_{z \sim p_z} log(1 - D(G(z))) \}$$

FAQ1: Why does it work?

FAQ1: Why does it work?

- D learns probability! G trains to sample instance with high probability!

FAQ1: Why does it work?

- D learns probability! G trains to sample instance with high probability!

- Objective does not determine mapping directly- arrangement of latent space is learned!

FAQ1: Why does it work?

- D learns probability! G trains to sample instance with high probability!
- Objective does not determine mapping directly- arrangement of latent space is learned!
- Theory: minimizes JS divergence between generated and real distributions.

FAQ2: Why alternating?

FAQ2: Why alternating?

- Gradients are meaningless when game is unbalanced.

FAQ2: Why alternating?

- Gradients are meaningless when game is unbalanced.

FAQ2: Why alternating?

- Gradients are meaningless when game is unbalanced.

- Pre-train D? Negative examples?

FAQ2: Why alternating?

- Gradients are meaningless when game is unbalanced.

- Pre-train D? Negative examples?

- Pre-train G? What loss?
  For G, D is a **learned loss function**

FAQ2: Why alternating?

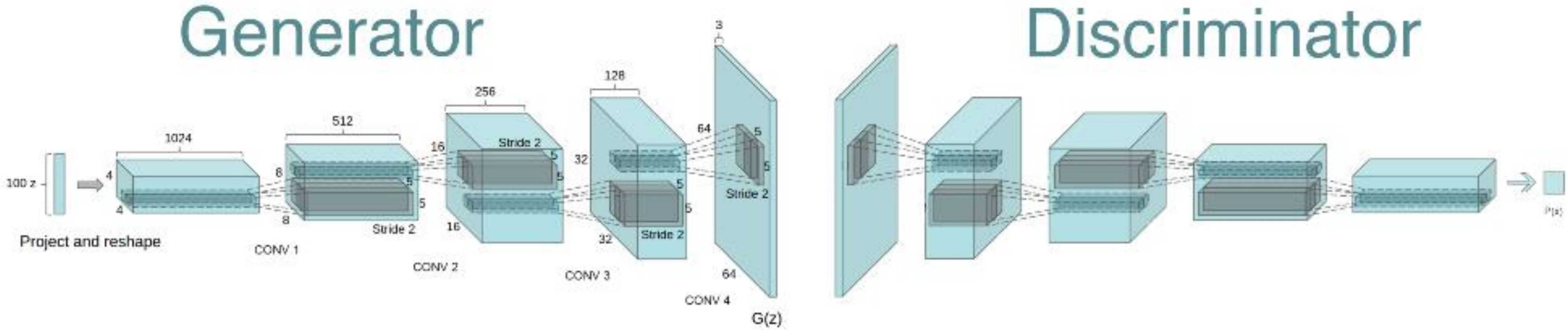
- Gradients are meaningless when game is unbalanced.

- Pre-train D? Negative examples?

- Pre-train G? What loss?
  For G, D is a **learned loss function**

GANs, Goodfellow 2014

DCGAN Radford 2015

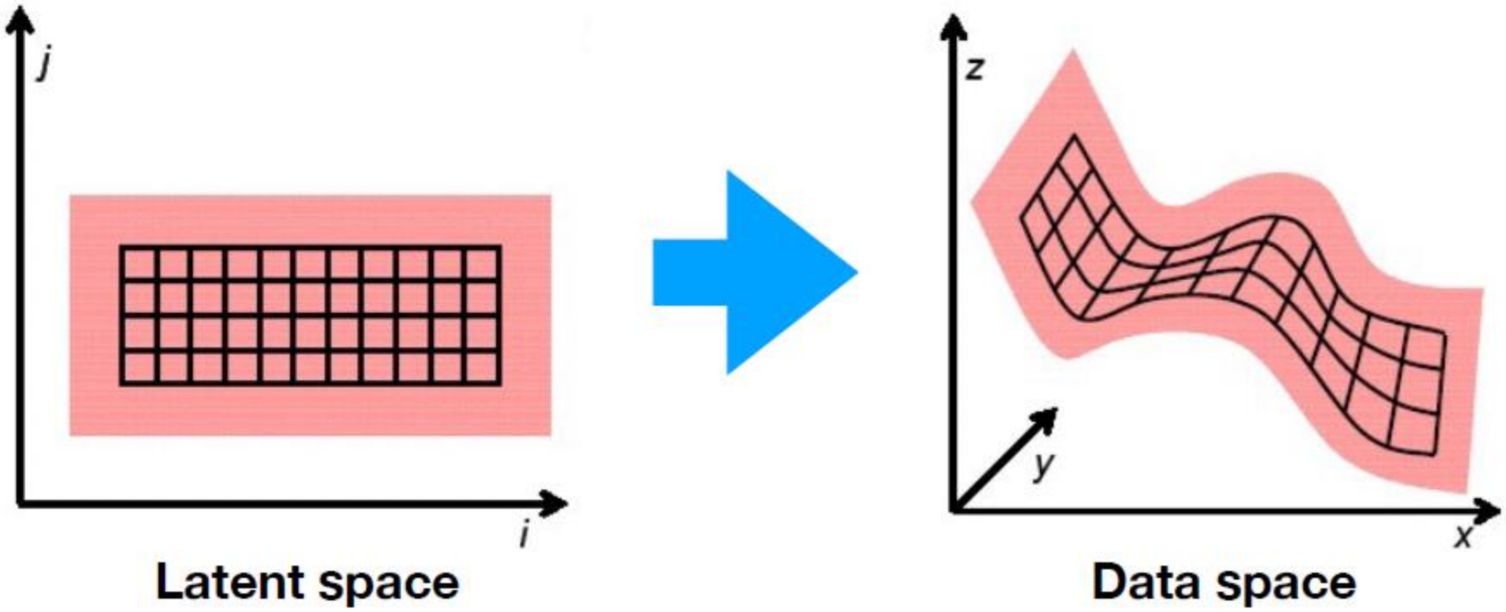


DCGAN Radford 2015

Latent space interpolation

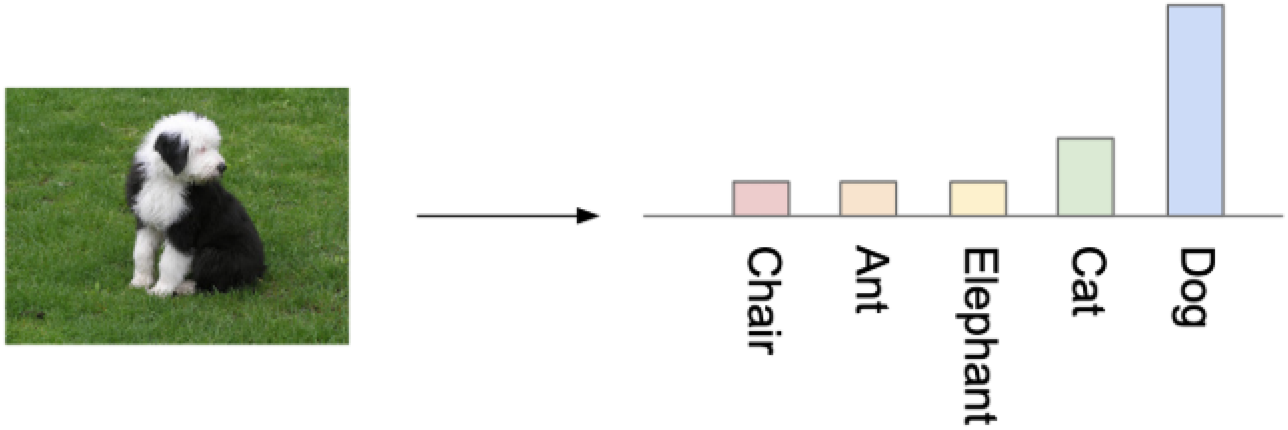
Latent space interpolation

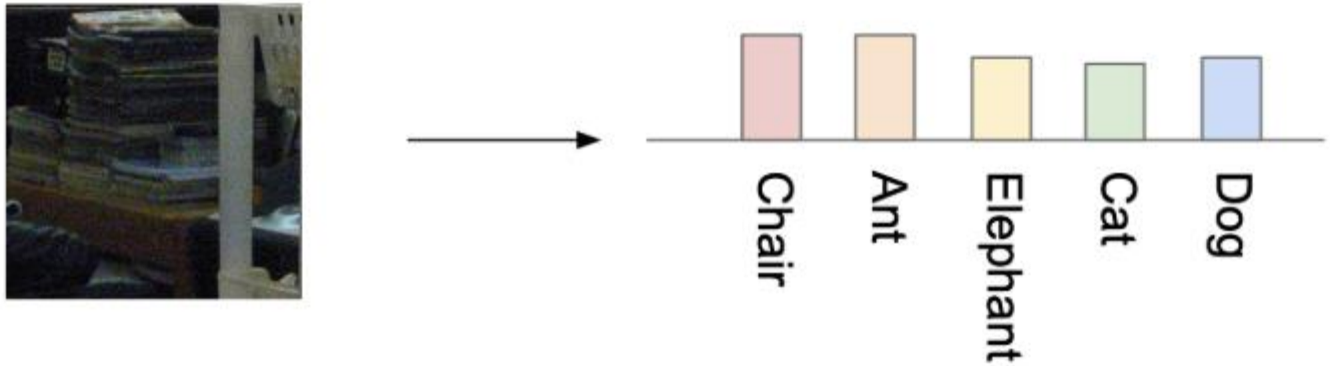
Why does it work?



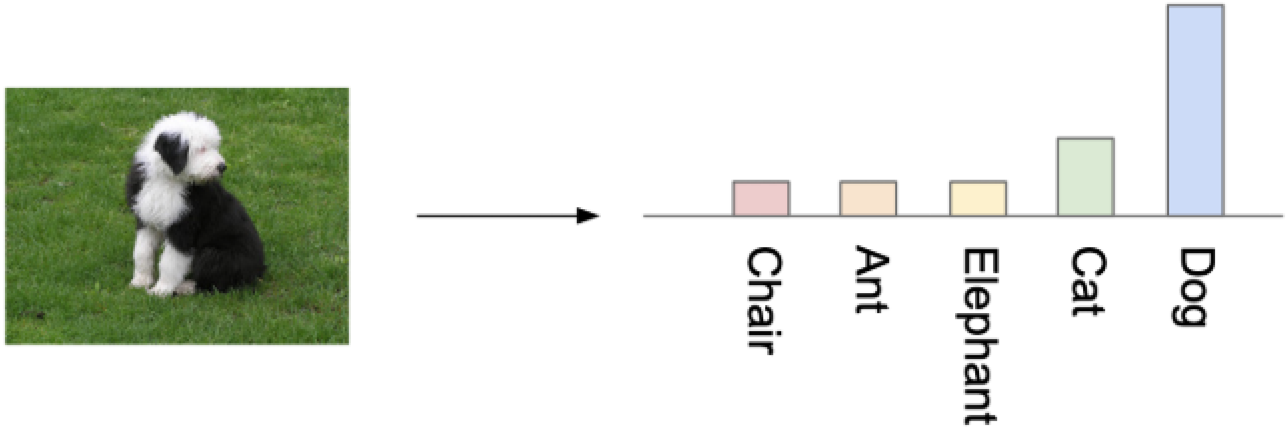
1. Every point is mapped to a valid example.
2. Network is continuous.

Evaluation metrics: Inception score

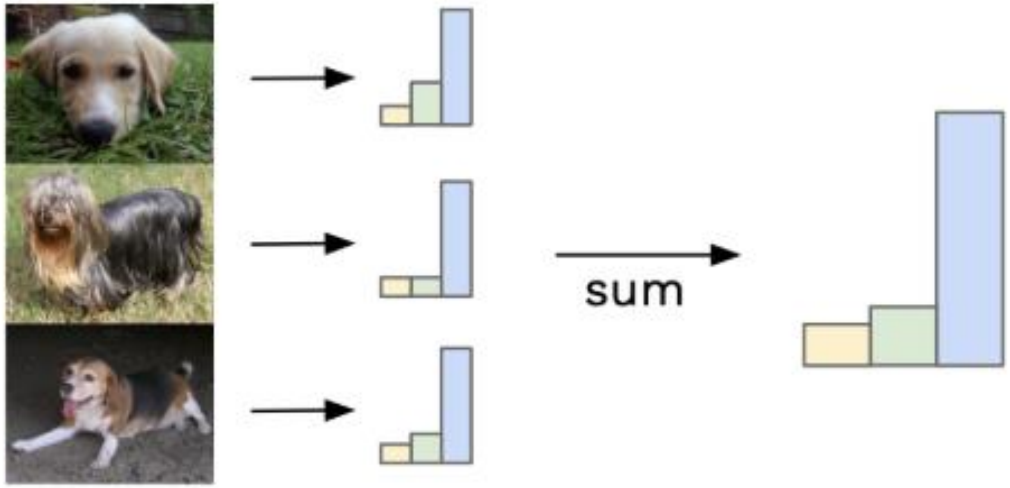




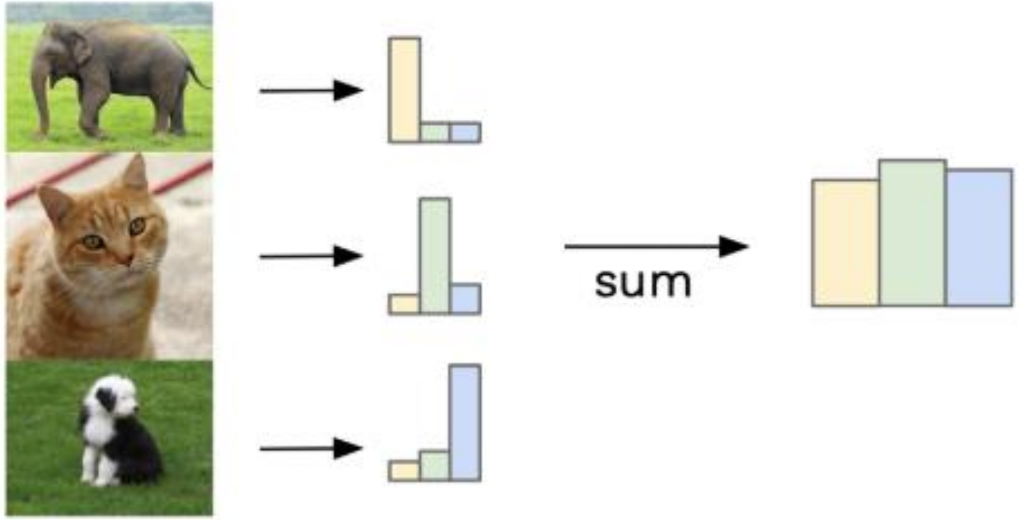
Evaluation metrics: Inception score



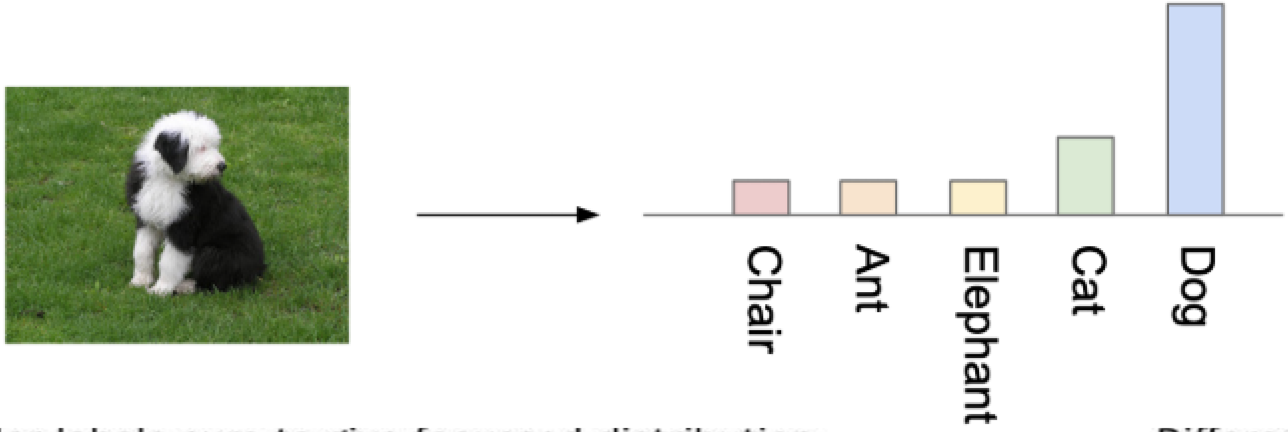
Similar labels sum to give focussed distribution



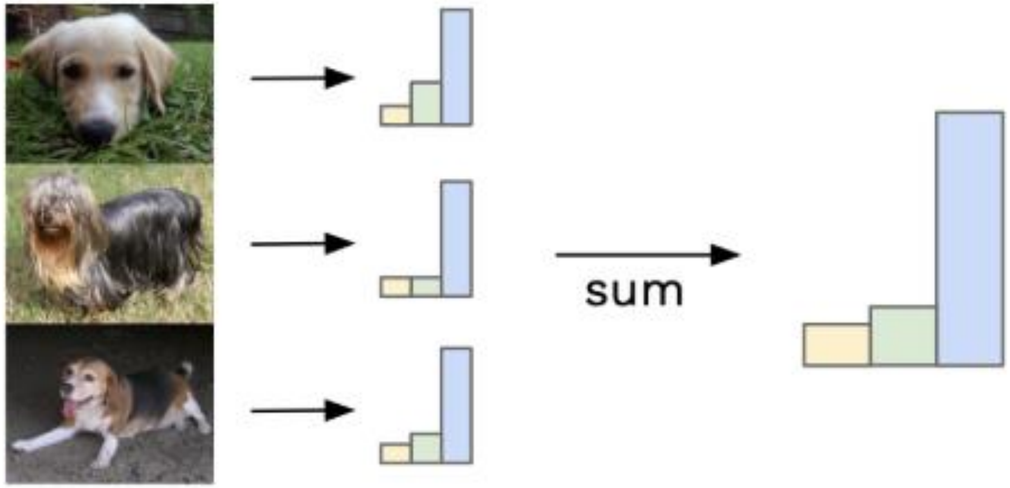
Different labels sum to give uniform distribution



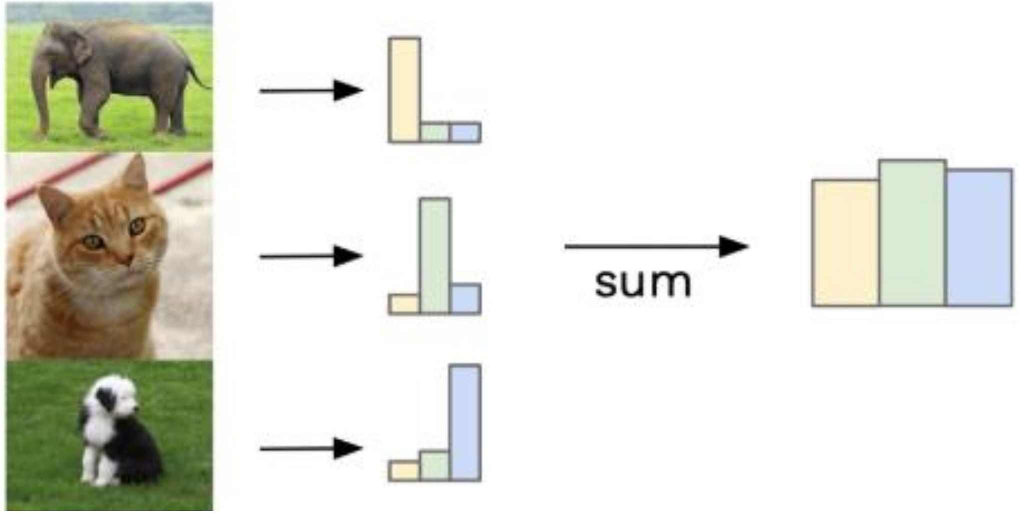
Evaluation metrics: Inception score



Similar labels sum to give focussed distribution



Different labels sum to give uniform distribution



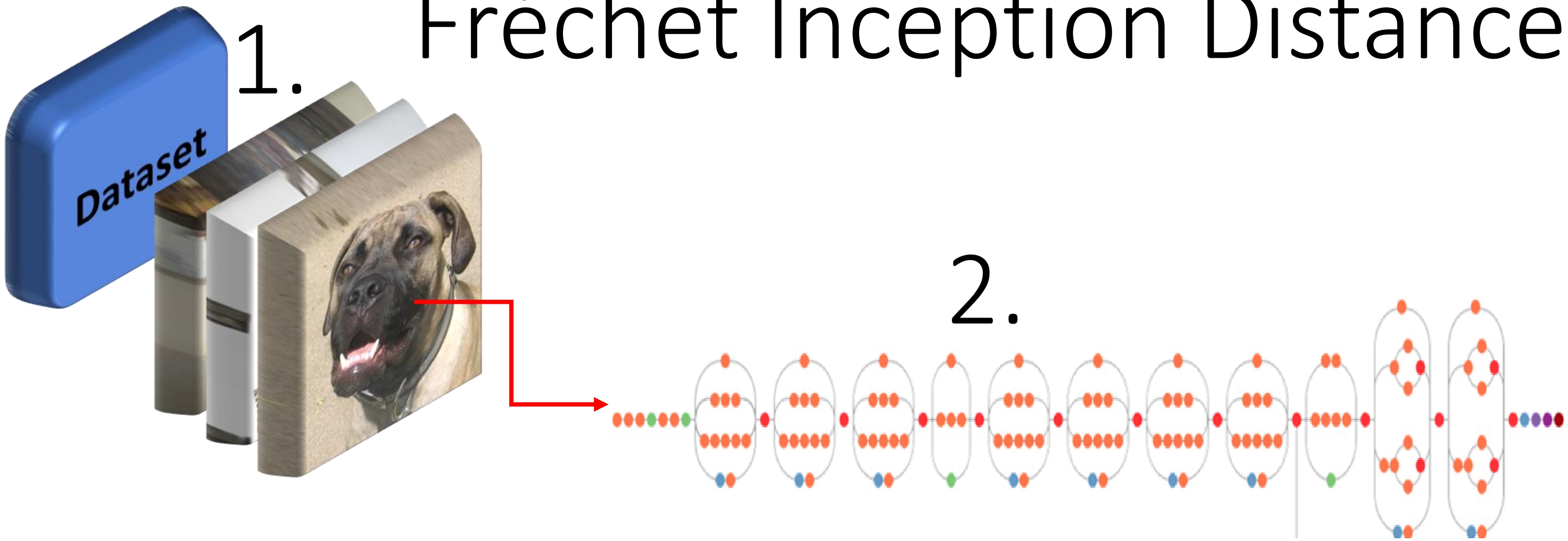
$$\mathbf{IS}(G) = \exp\left(\mathbb{E}_{\mathbf{x}\sim p_g} D_{KL}(\, p(y|\mathbf{x}) \parallel p(y)\,) \right)$$

Fréchet Inception Distance (FID)

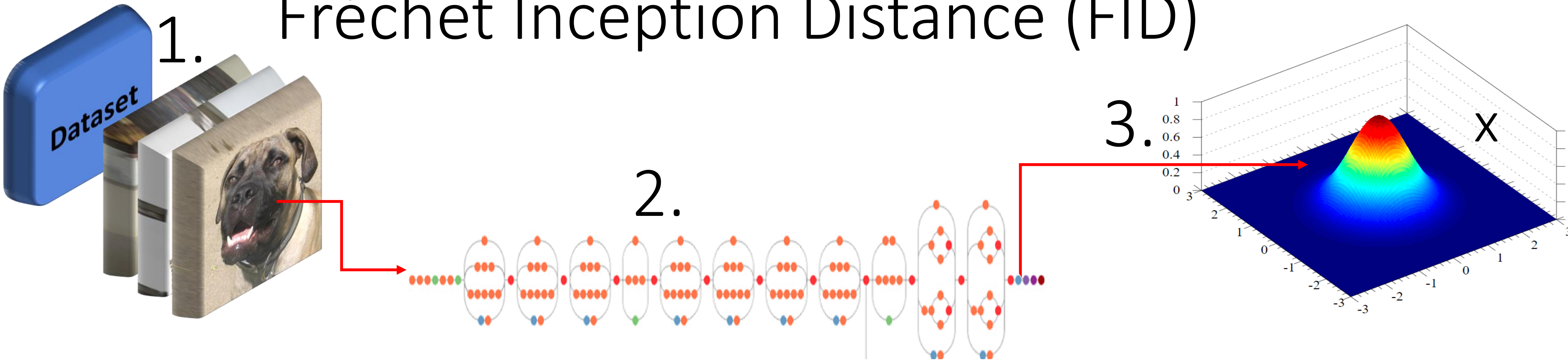
Fréchet Inception Distance (FID)

1.

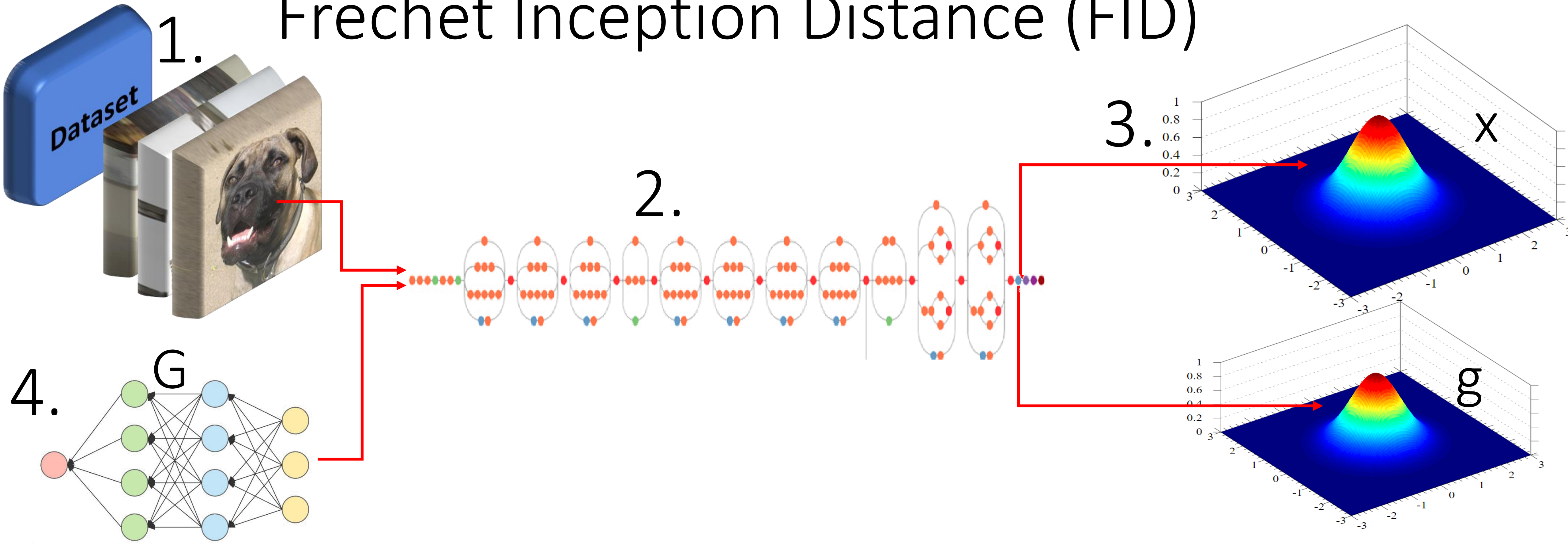
Fréchet Inception Distance (FID)



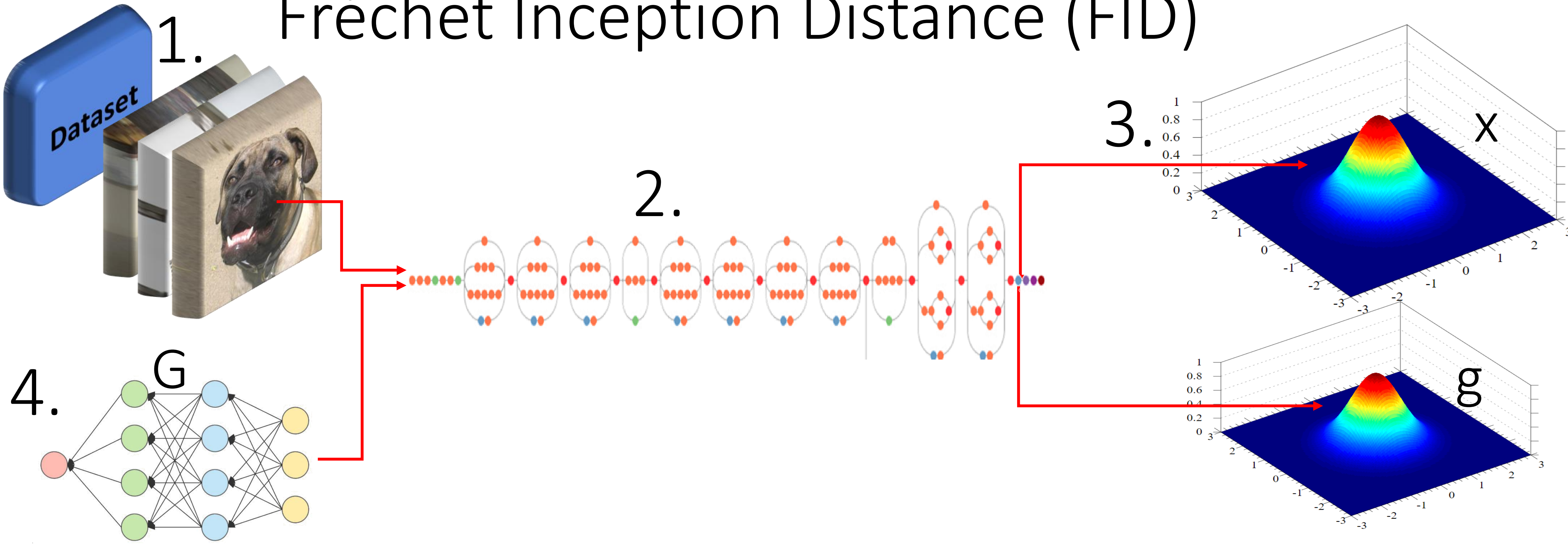
Fréchet Inception Distance (FID)



Fréchet Inception Distance (FID)

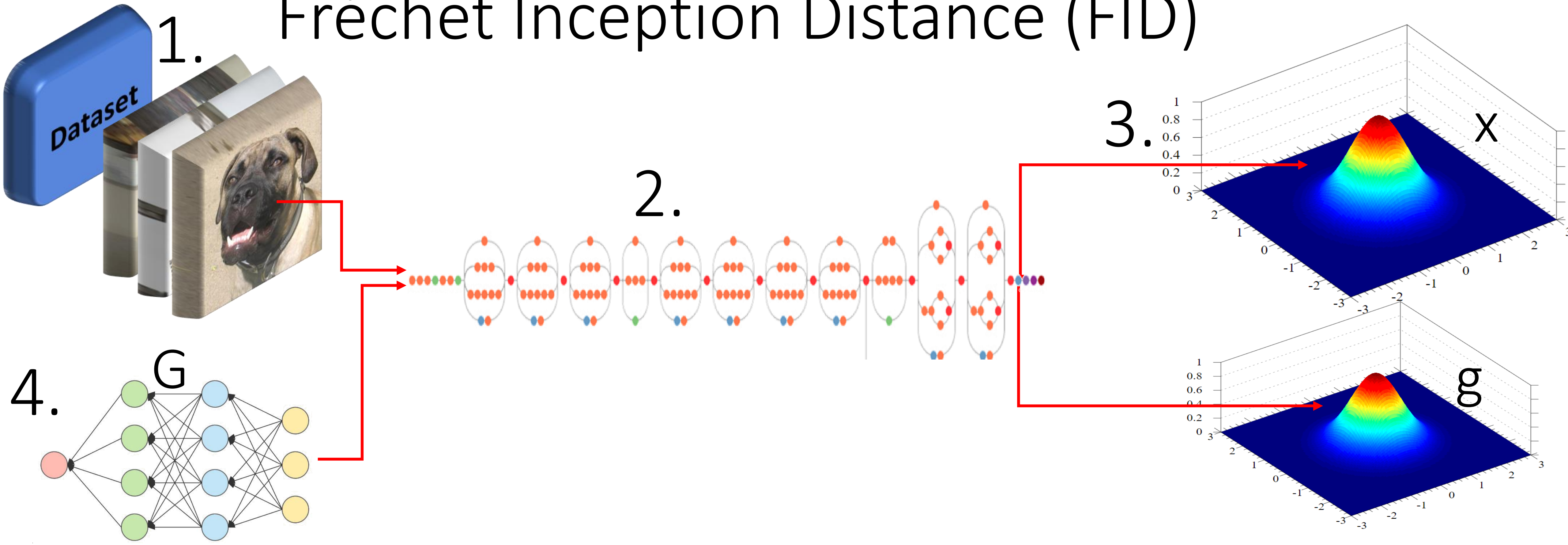


Fréchet Inception Distance (FID)

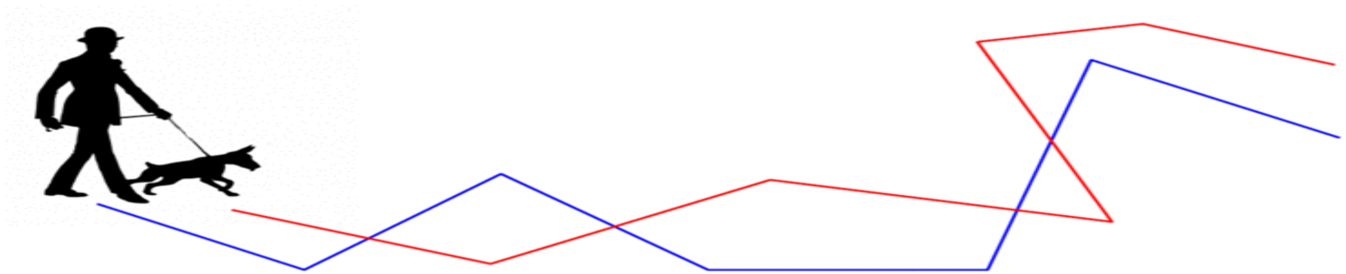


5. $\texttt{FID}(x,g) = ||\mu_x - \mu_g||_2^2 + \mathrm{Tr}(\Sigma_x + \Sigma_g - 2(\Sigma_x\Sigma_g)^{\frac{1}{2}})$

Fréchet Inception Distance (FID)



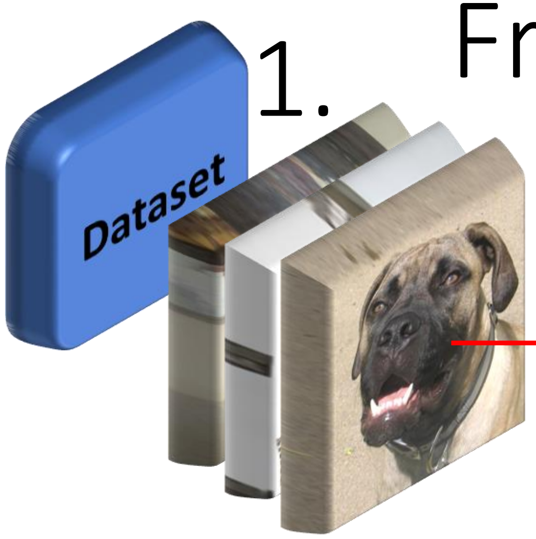
5. $$\mathtt{FID}(x,g) = ||\mu_x - \mu_g||_2^2 + \mathrm{Tr}(\Sigma_x + \Sigma_g - 2(\Sigma_x\Sigma_g)^{\frac{1}{2}})$$



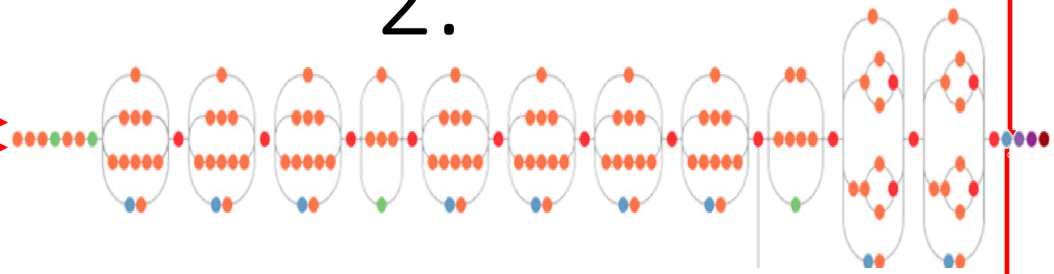
Fréchet Inception Distance (FID)

Depends on the number of samples!

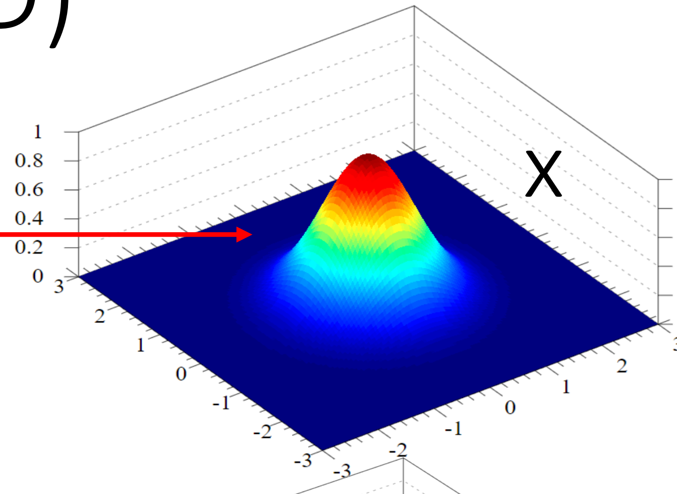
1.

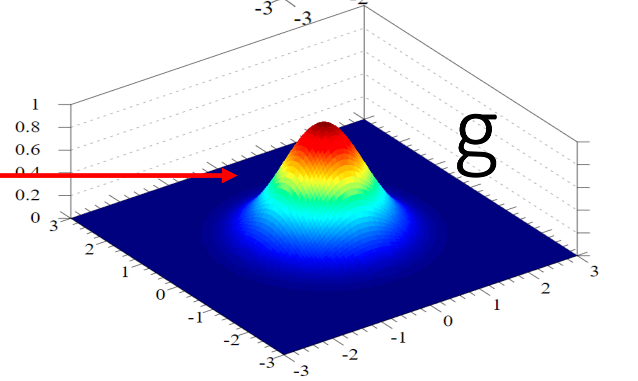


2.

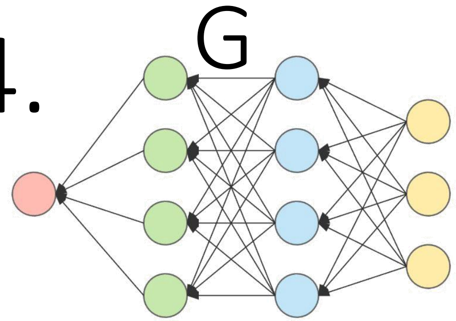


3.





4.



5. $$\texttt{FID}(x,g) = ||\mu_x - \mu_g||_2^2 + \mathrm{Tr}(\Sigma_x + \Sigma_g - 2(\Sigma_x\Sigma_g)^{\frac{1}{2}})$$

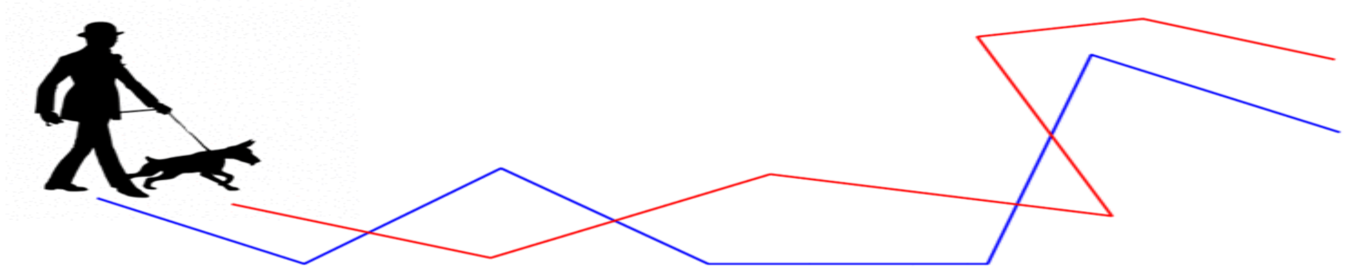
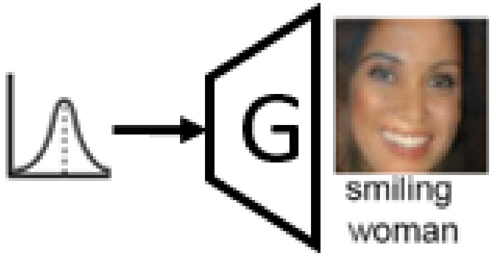
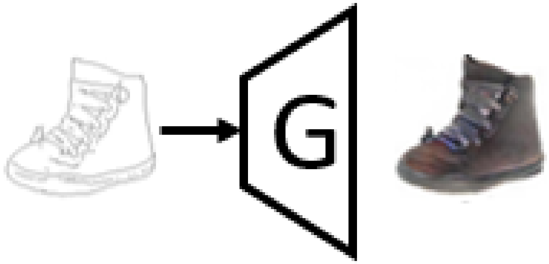


Image to Image translation

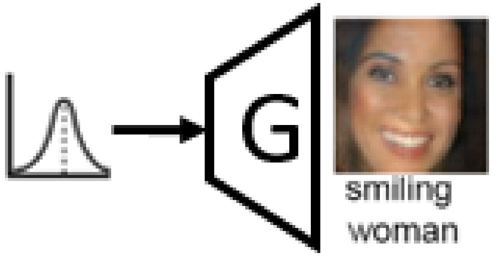


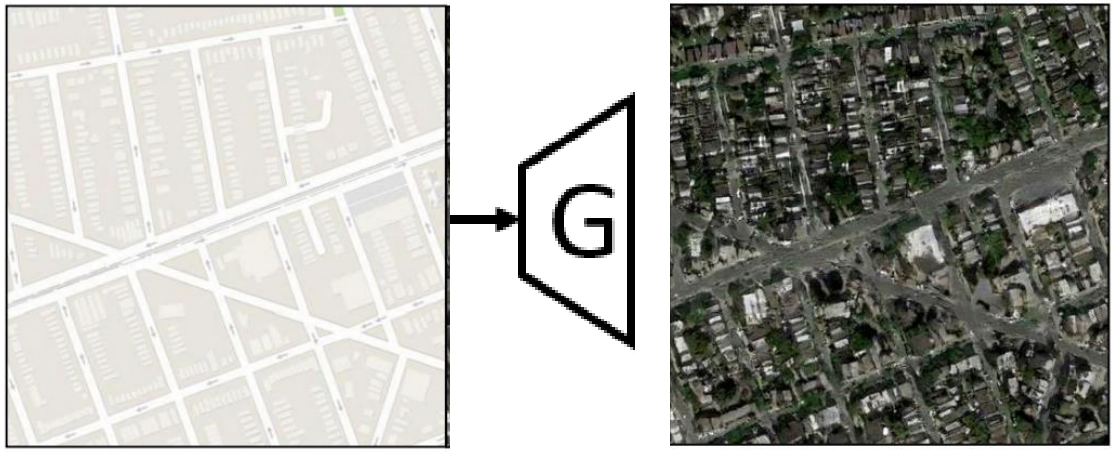


Isola et al. Nov2016

Slide by Sefi Bell-Kligler & Akhiad Bercovich

Image to Image translation

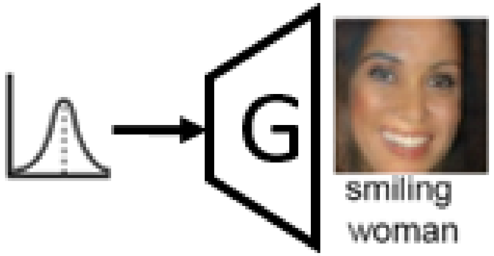


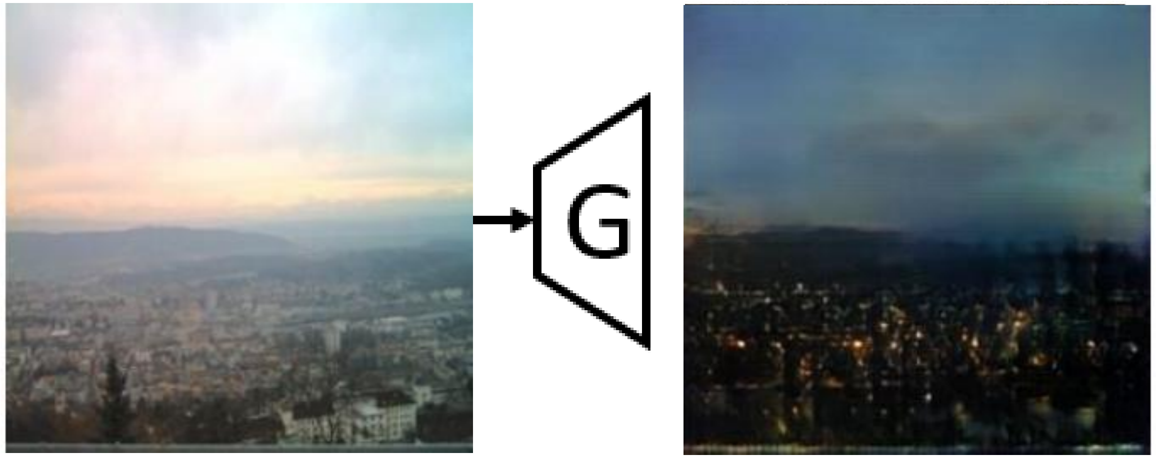


Isola et al. Nov2016

Slide by Sefi Bell-Kligler & Akhiad Bercovich

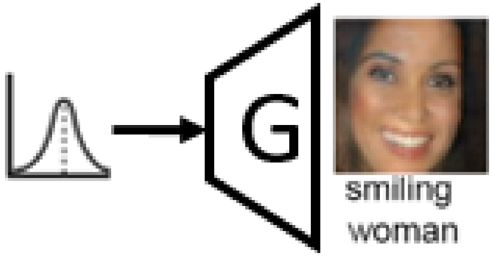
Image to Image translation

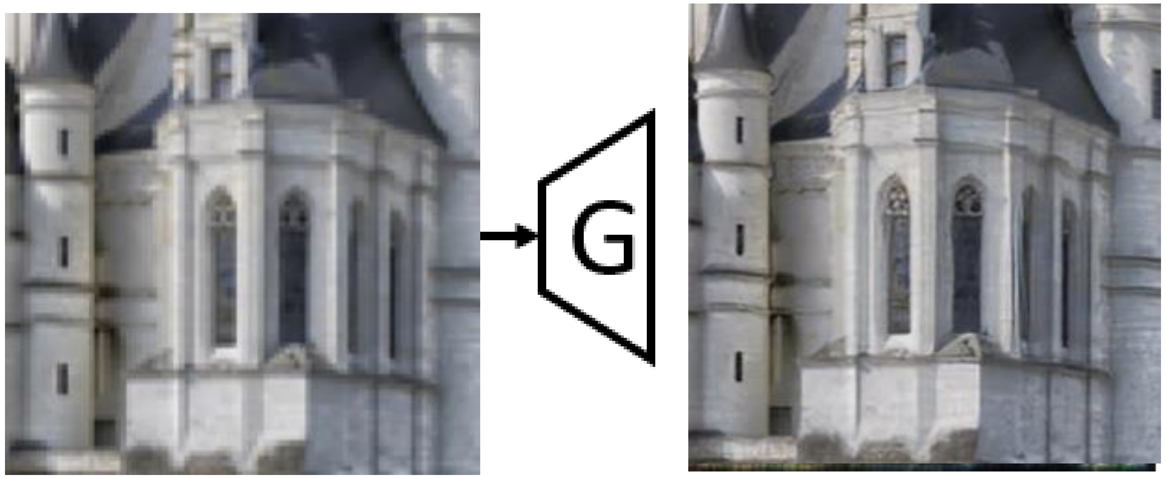




Isola et al. Nov2016

Image to Image translation

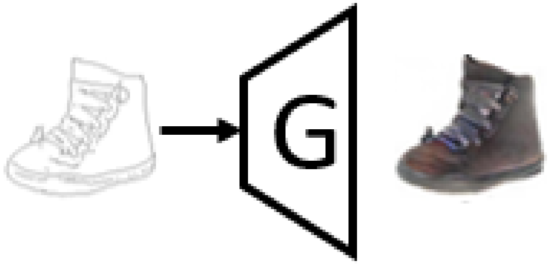




Isola et al. Nov2016

Slide by Sefi Bell-Kligler & Akhiad Bercovich

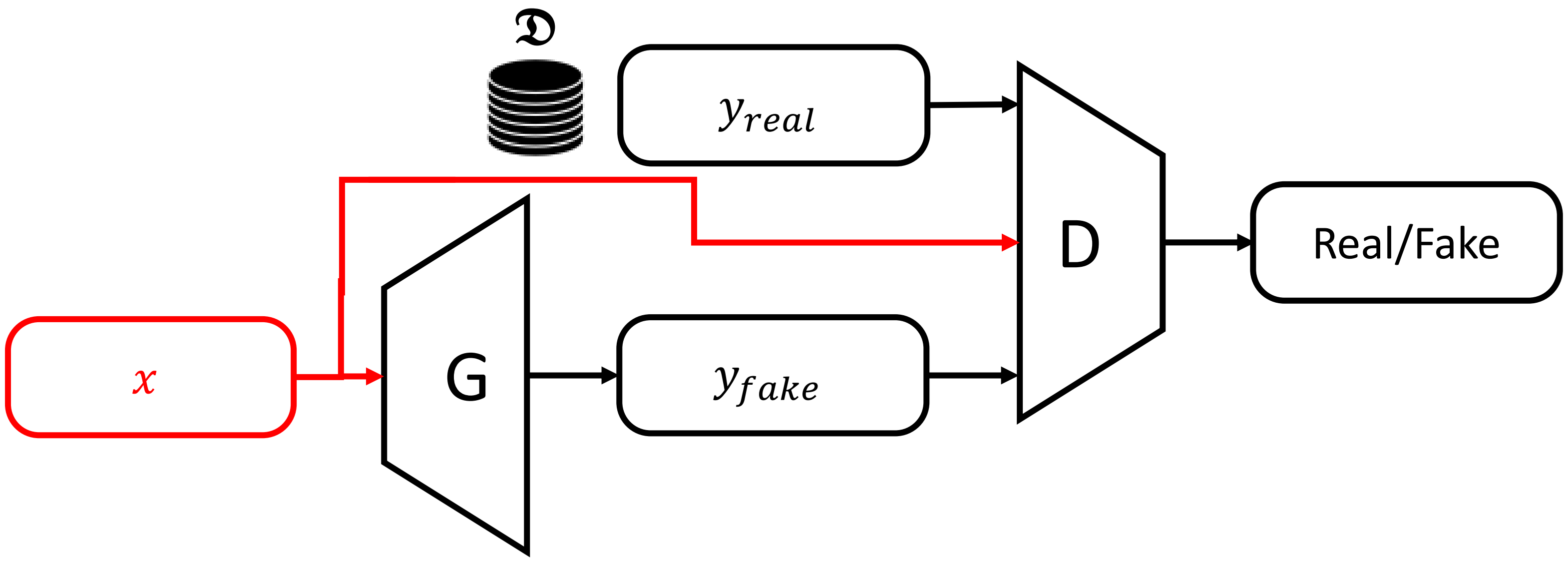
Image to Image translation



Isola et al. Nov2016

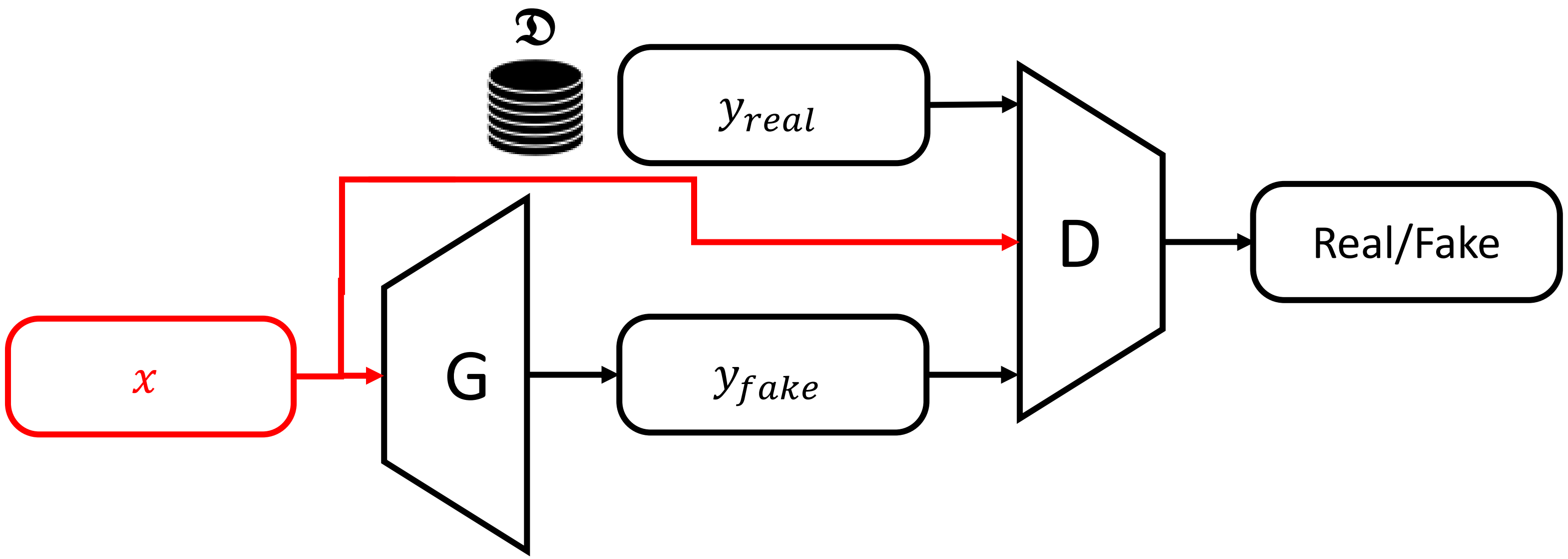
Slide by Sefi Bell-Kligler & Akhiad Bercovich

Conditional GAN



Slide by Sefi Bell-Kligler & Akhiad Bercovich

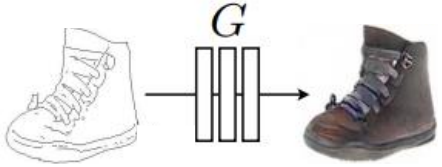
Conditional GAN



$$\mathcal{L}_{C-GAN} = \min_{G} \max_{D} \mathbb{E}[\log D(y, x)] + \mathbb{E}[\log(1 - D(G(x), x))]$$

Slide by Sefi Bell-Kligler & Akhiad Bercovich

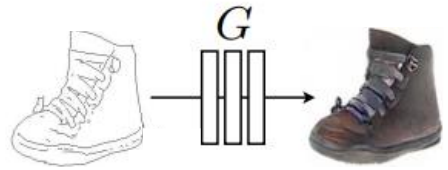
Pix2Pix

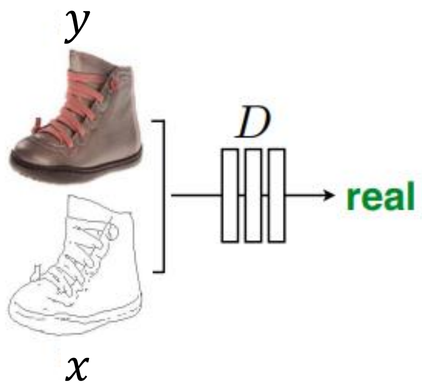


Isola et al. Nov2016

Slide by Sefi Bell-Kligler & Akhiad Bercovich

Pix2Pix

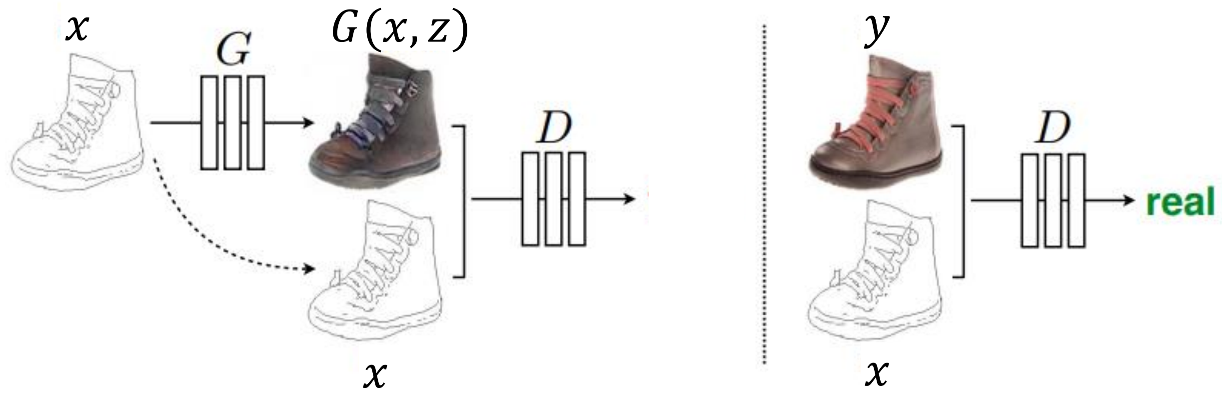




Isola et al. Nov2016

Slide by Sefi Bell-Kligler & Akhiad Bercovich

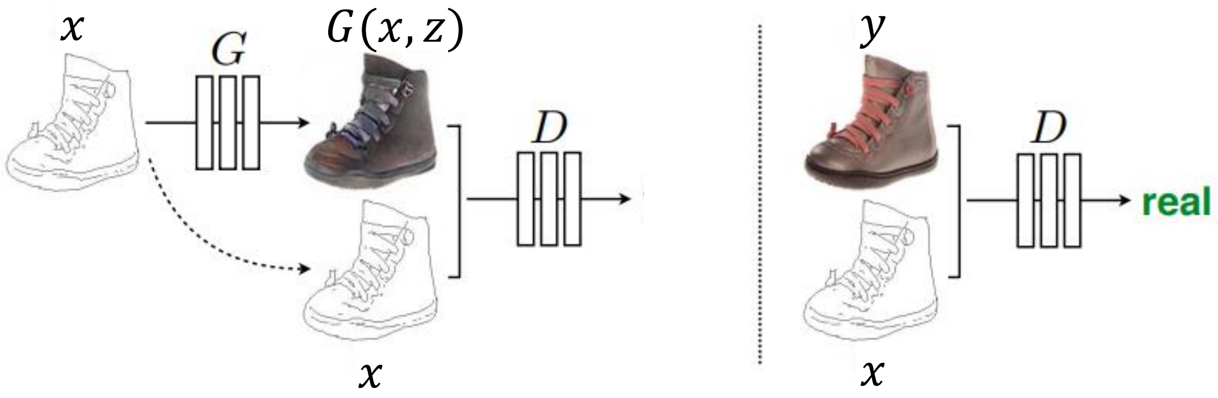
Pix2Pix



Isola et al. Nov2016

Slide by Sefi Bell-Kligler & Akhiad Bercovich

Pix2Pix

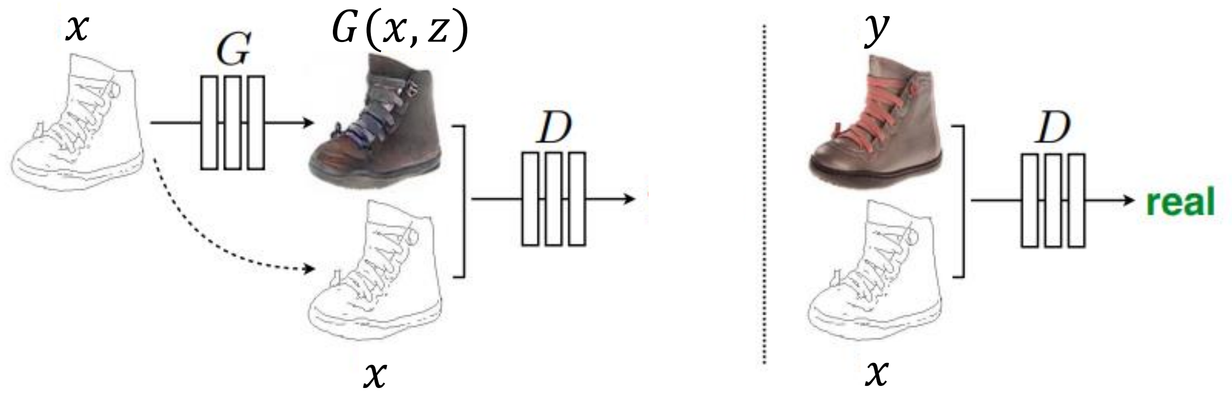


$$\mathcal{L}_{C-GAN} = \min_{G} \max_{D} \mathbb{E}[\log D(y, x)] + \mathbb{E}[\log(1 - D(G(x), x))]$$

Isola et al. Nov2016

Slide by Sefi Bell-Kligler & Akhiad Bercovich

Pix2Pix



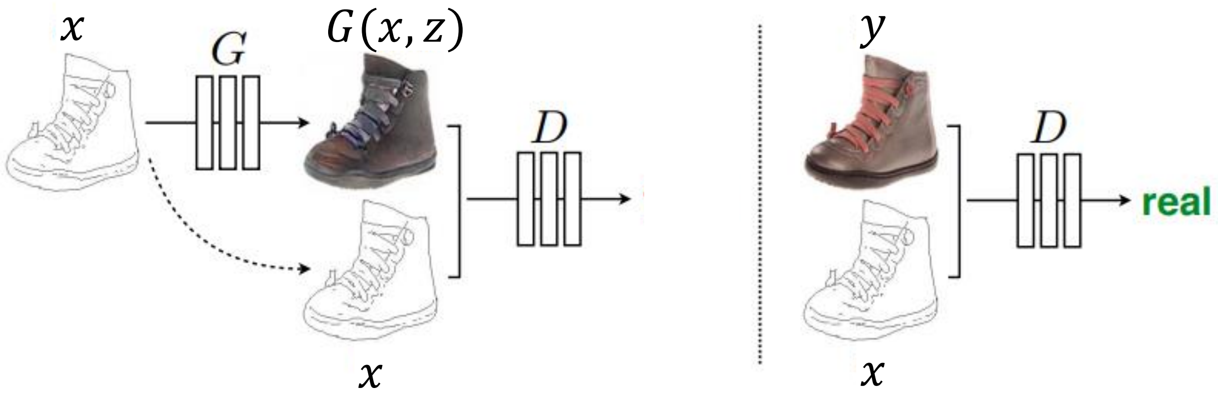
$$\mathcal{L}_{C-GAN} = \min_{G} \max_{D} \mathbb{E}[\log D(y, x)] + \mathbb{E}[\log(1 - D(G(x), x))]$$

$$\mathcal{L}_{L1} = \|y - G(x,z)\|_1$$

Slide by Sefi Bell-Kligler & Akhiad Bercovich

Isola et al. Nov2016

Pix2Pix



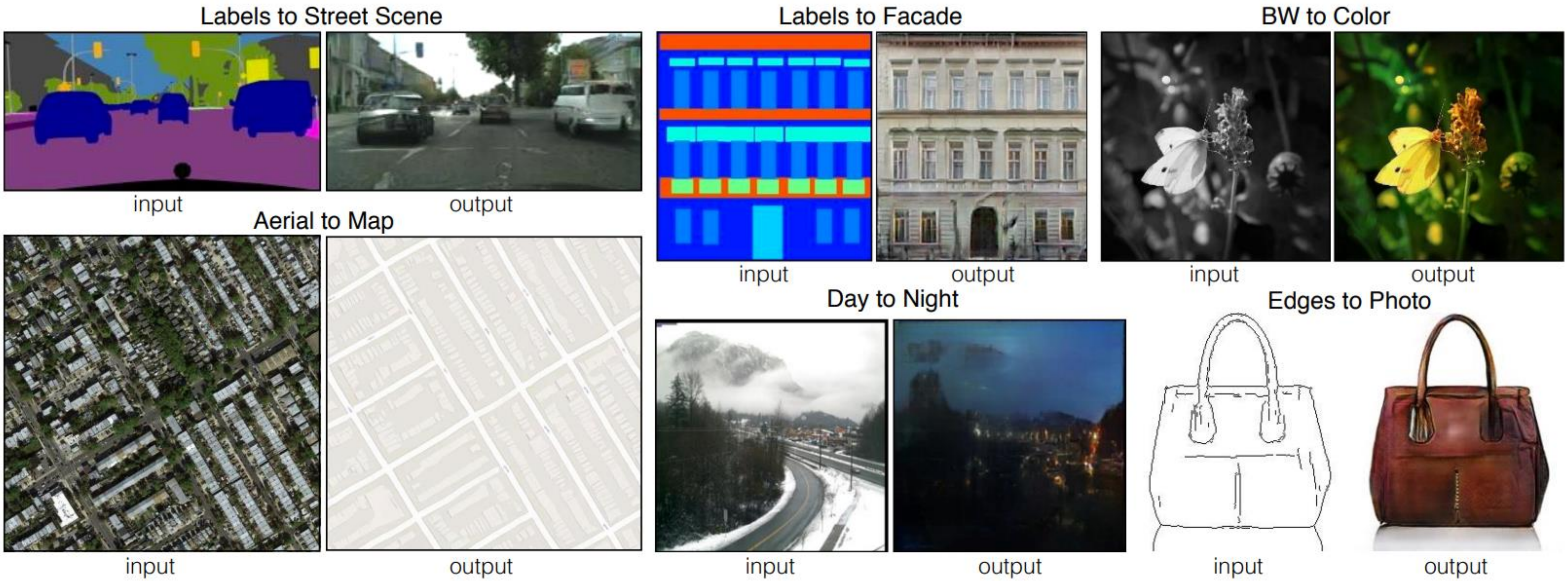
$$\mathcal{L}_{C-GAN} = \min_G \max_D \mathbb{E}[\log D(y, x)] + \mathbb{E}[\log(1 - D(G(x), x))]$$

$$\mathcal{L}_{L1} = \|y - G(x, z)\|_1$$

$$Objective = \mathcal{L}_{C-GAN} + \lambda \cdot \mathcal{L}_{L1}$$

Isola et al. Nov2016

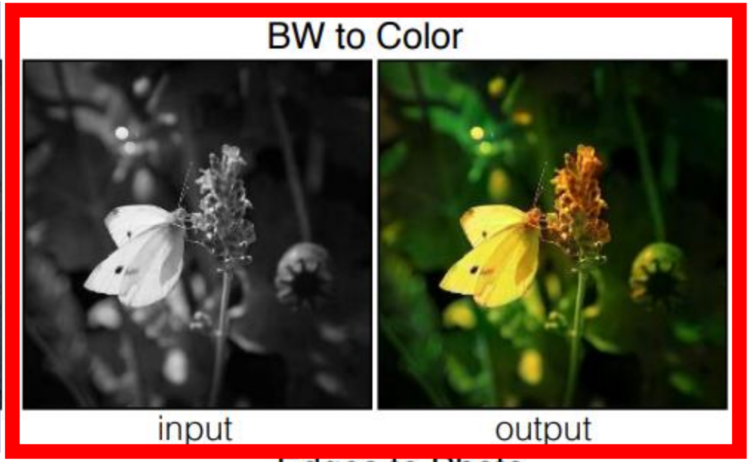
Slide by Sefi Bell-Kligler & Akhiad Bercovich

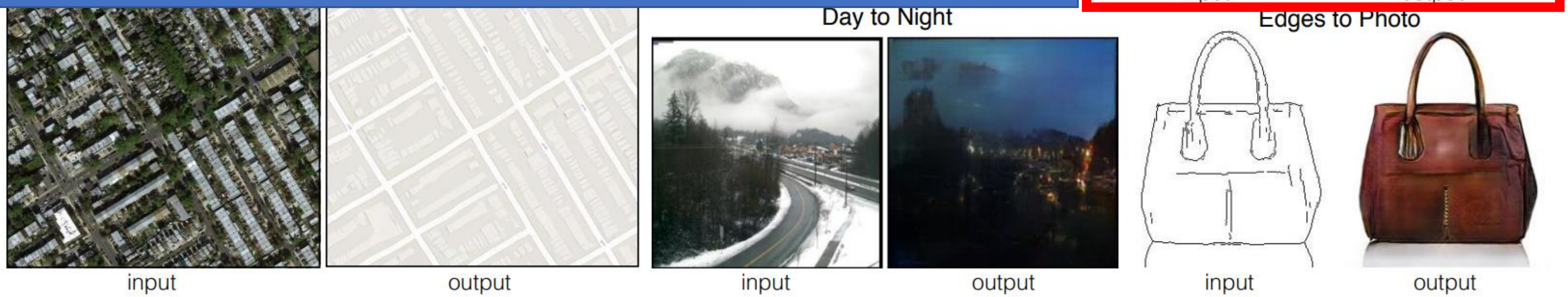


https://affinelayer.com/pixsrv/

Generative VS Discriminative

What would happen if we train regular supervised mapping?

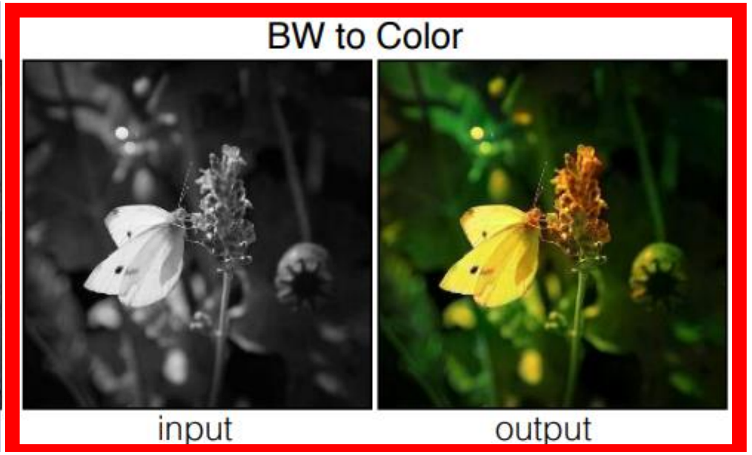


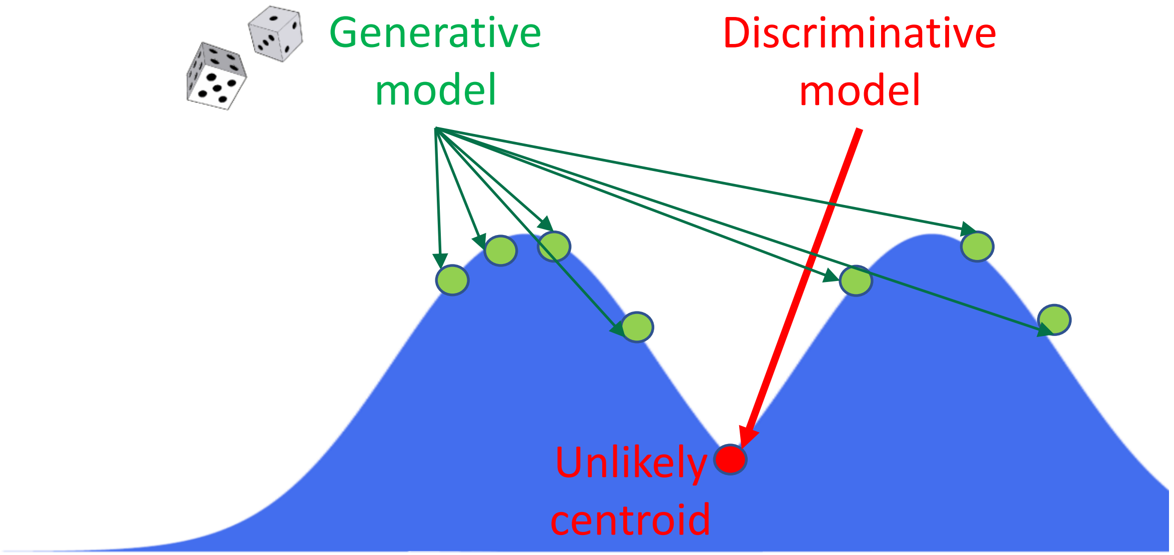


https://affinelayer.com/pixsrv/

Generative VS Discriminative

What would happen if we train regular supervised mapping?





Training GANs is hard

Training GANs is hard

Training GANs is hard

- Stability

Training GANs is hard

- Stability

Training
GANs
Be like:

Training GANs is hard

- Stability

Training GANs is hard

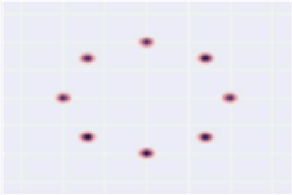
- Stability

- Mode collapse

Training GANs is hard

- Stability

- Mode collapse

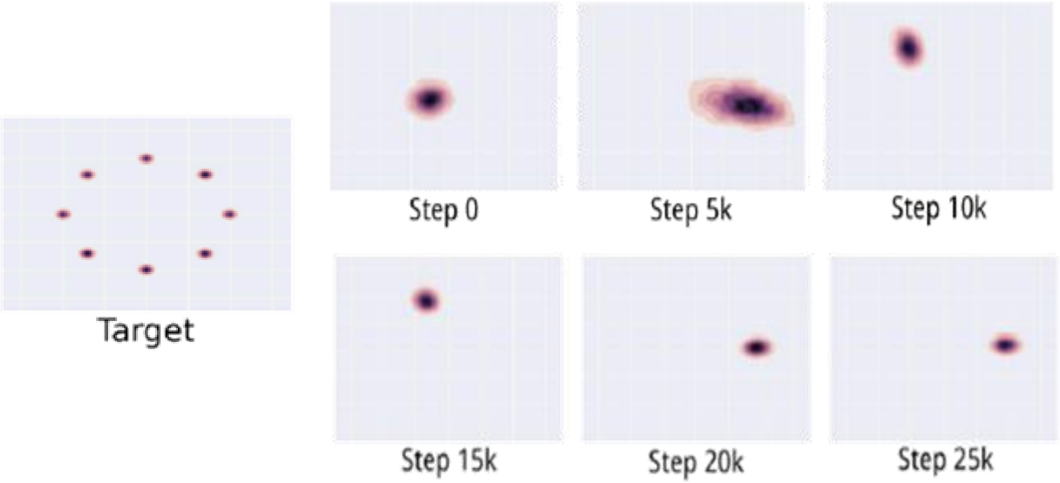


Target

Training GANs is hard

- Stability

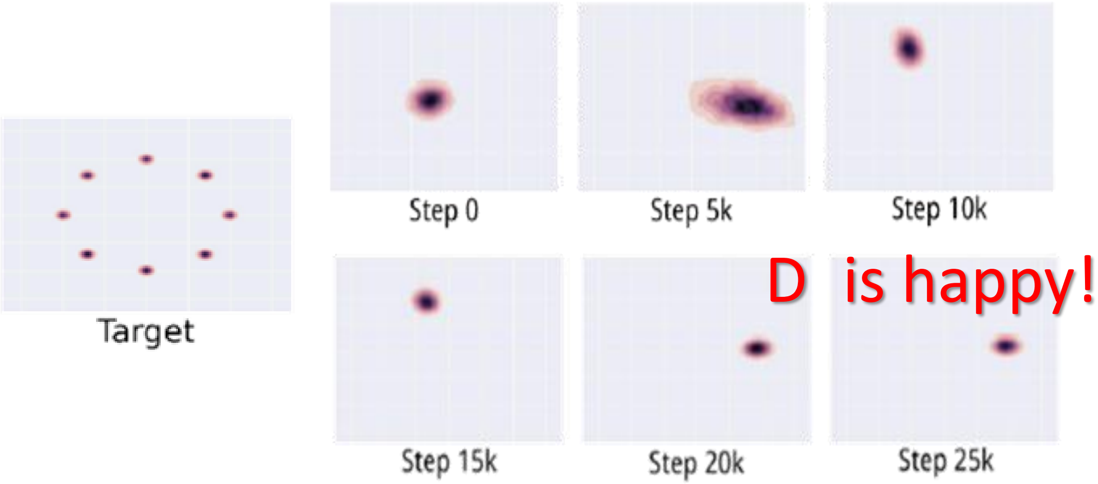
- Mode collapse



Training GANs is hard

- Stability

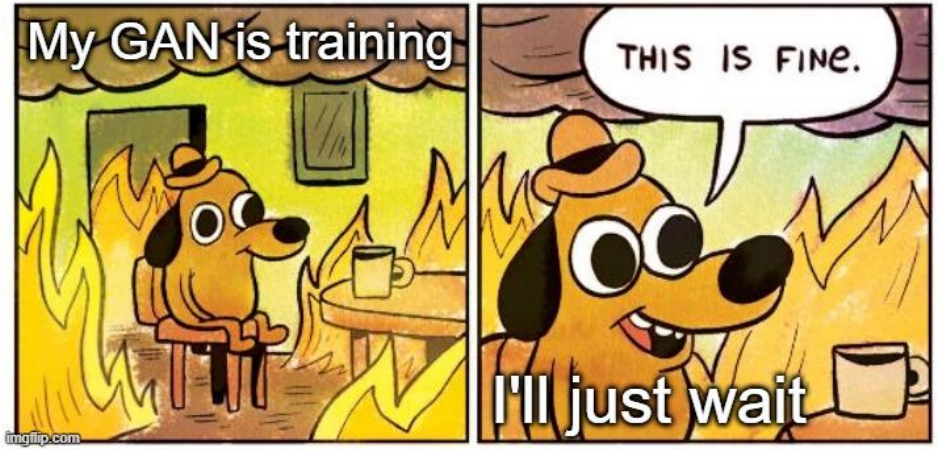
- Mode collapse



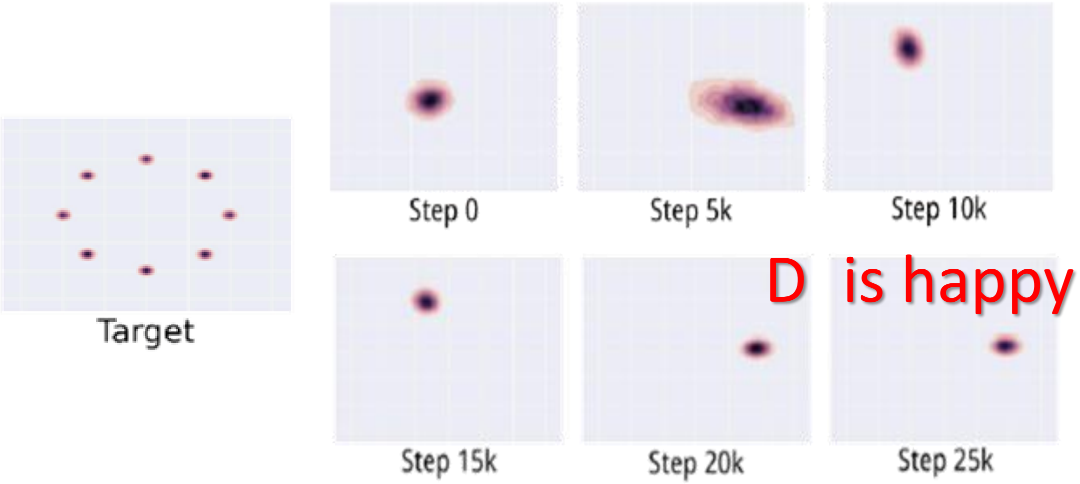
Training GANs is hard

- Stability

- GANs can over-train



- Mode collapse

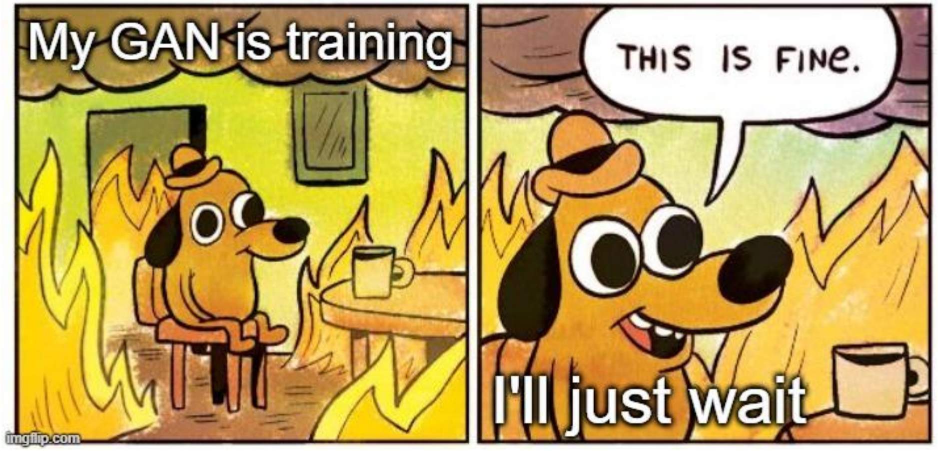


D is happy!

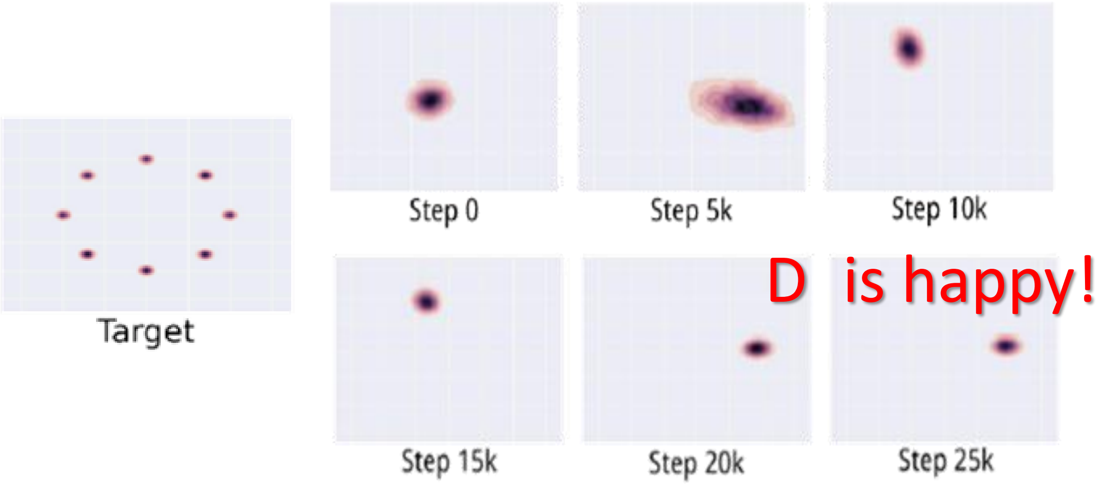
Training GANs is hard

- Stability

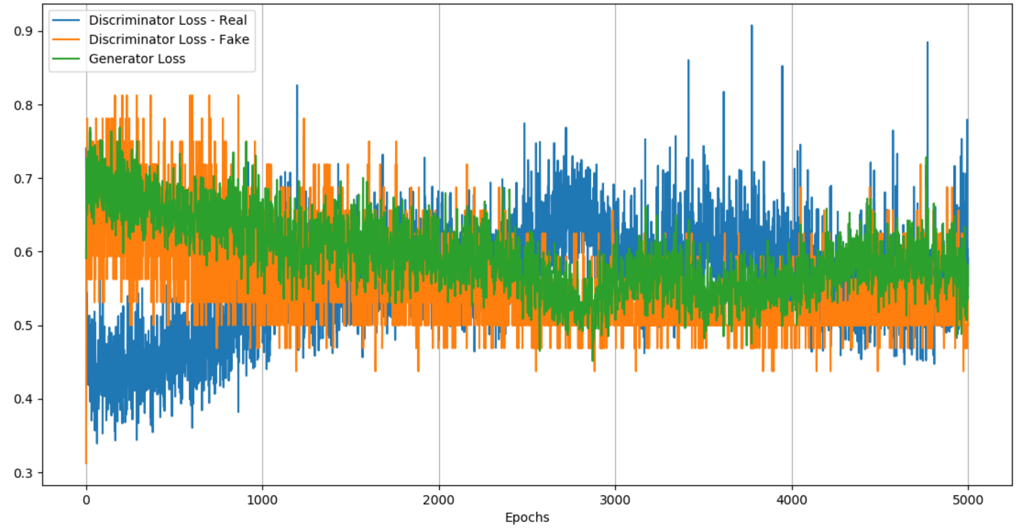
- GANs can over-train



- Mode collapse



- Losses don’t reflect quality



Types of GAN losses - Wasserstein GAN

Discriminator

GAN

$$\max_D E_{x\sim p_X}[\log D(x)] + E_{z\sim p_Z}[\log(1 - D(G(z))]$$

WGAN

$$\max_D E_{x\sim p_X}[D(x)] - E_{z\sim p_Z}[D(G(z))]$$

Types of GAN losses - Wasserstein GAN

Discriminator

GAN $$\max_D E_{x\sim p_X}[\log D(x)] + E_{z\sim p_Z}[\log(1 - D(G(z))]$$

WGAN $$\max_D E_{x\sim p_X}[D(x)] - E_{z\sim p_Z}[D(G(z))]$$

WGAN: minimize earth mover distance between p_X and $p_{G(Z)}$

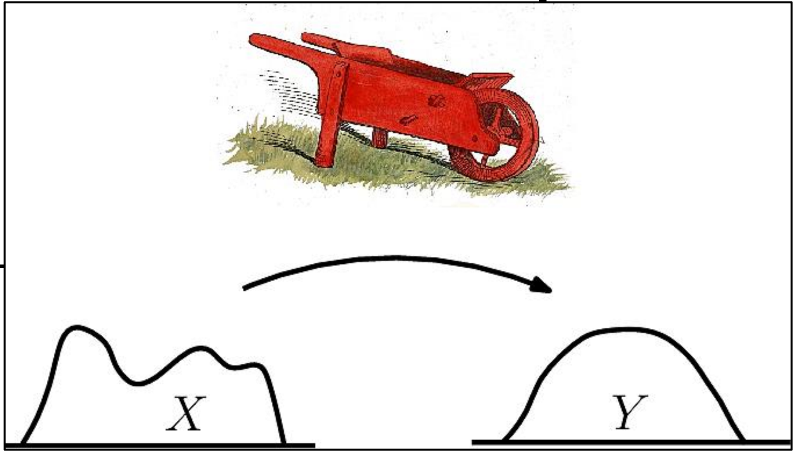
$$EM(p_X, p_{G(Z)}) = \inf_{\gamma \in \prod(p_X, p_{G(Z)})} E_{(x,y)\sim\gamma}[||x - y||]$$

Types of GAN losses - Wasserstein GAN

Discriminator

GAN $\max_D E_{x\sim p_X}[\log D(x)] + E_{z\sim p_Z}[\log(1 - D(G(z)))]$

WGAN $\max_D E_{x\sim p_X}[D(x)] - E_{z\sim p_Z}[D(G(z))]$



WGAN: minimize earth mover distance between p_X and $p_{G(Z)}$

$$EM(p_X, p_{G(Z)}) = \inf_{\gamma\in\prod(p_X, p_{G(Z)})} E_{(x,y)\sim\gamma}[||x-y||]$$

Types of GAN losses

GAN	Discriminator Loss	Generator Loss
MM GAN	$\mathcal{L}_{\text{D}}^{\text{GAN}} = -\mathbb{E}_{x\sim p_d}[\log(D(x))] - \mathbb{E}_{\hat{x}\sim p_g}[\log(1 - D(\hat{x}))]$	$\mathcal{L}_{\text{G}}^{\text{GAN}} = \mathbb{E}_{\hat{x}\sim p_g}[\log(1 - D(\hat{x}))]$
NS GAN	$\mathcal{L}_{\text{D}}^{\text{NSGAN}} = -\mathbb{E}_{x\sim p_d}[\log(D(x))] - \mathbb{E}_{\hat{x}\sim p_g}[\log(1 - D(\hat{x}))]$	$\mathcal{L}_{\text{G}}^{\text{NSGAN}} = -\mathbb{E}_{\hat{x}\sim p_g}[\log(D(\hat{x}))]$
WGAN	$\mathcal{L}_{\text{D}}^{\text{WGAN}} = -\mathbb{E}_{x\sim p_d}[D(x)] + \mathbb{E}_{\hat{x}\sim p_g}[D(\hat{x})]$	$\mathcal{L}_{\text{G}}^{\text{WGAN}} = -\mathbb{E}_{\hat{x}\sim p_g}[D(\hat{x})]$
WGAN GP	$\mathcal{L}_{\text{D}}^{\text{WGANGP}} = \mathcal{L}_{\text{D}}^{\text{WGAN}} + \lambda\mathbb{E}_{\hat{x}\sim p_g}[(\lvert\lvert\nabla D(\alpha x + (1 - \alpha\hat{x})\rvert\rvert_2 - 1)^2]$	$\mathcal{L}_{\text{G}}^{\text{WGANGP}} = -\mathbb{E}_{\hat{x}\sim p_g}[D(\hat{x})]$
LS GAN	$\mathcal{L}_{\text{D}}^{\text{LSGAN}} = -\mathbb{E}_{x\sim p_d}[(D(x) - 1)^2] + \mathbb{E}_{\hat{x}\sim p_g}[D(\hat{x})^2]$	$\mathcal{L}_{\text{G}}^{\text{LSGAN}} = -\mathbb{E}_{\hat{x}\sim p_g}[(D(\hat{x} - 1))^2]$
DRAGAN	$\mathcal{L}_{\text{D}}^{\text{DRAGAN}} = \mathcal{L}_{\text{D}}^{\text{GAN}} + \lambda\mathbb{E}_{\hat{x}\sim p_d+\mathcal{N}(0,c)}[(\lvert\lvert\nabla D(\hat{x})\rvert\rvert_2 - 1)^2]$	$\mathcal{L}_{\text{G}}^{\text{DRAGAN}} = \mathbb{E}_{\hat{x}\sim p_g}[\log(1 - D(\hat{x}))]$
BEGAN	$\mathcal{L}_{\text{D}}^{\text{BEGAN}} = \mathbb{E}_{x\sim p_d}[\lvert\lvert x - \text{AE}(x)\rvert\rvert_1] - k_t\mathbb{E}_{\hat{x}\sim p_g}[\lvert\lvert\hat{x} - \text{AE}(\hat{x})\rvert\rvert_1]$	$\mathcal{L}_{\text{G}}^{\text{BEGAN}} = \mathbb{E}_{\hat{x}\sim p_g}[\lvert\lvert\hat{x} - \text{AE}(\hat{x})\rvert\rvert_1]$

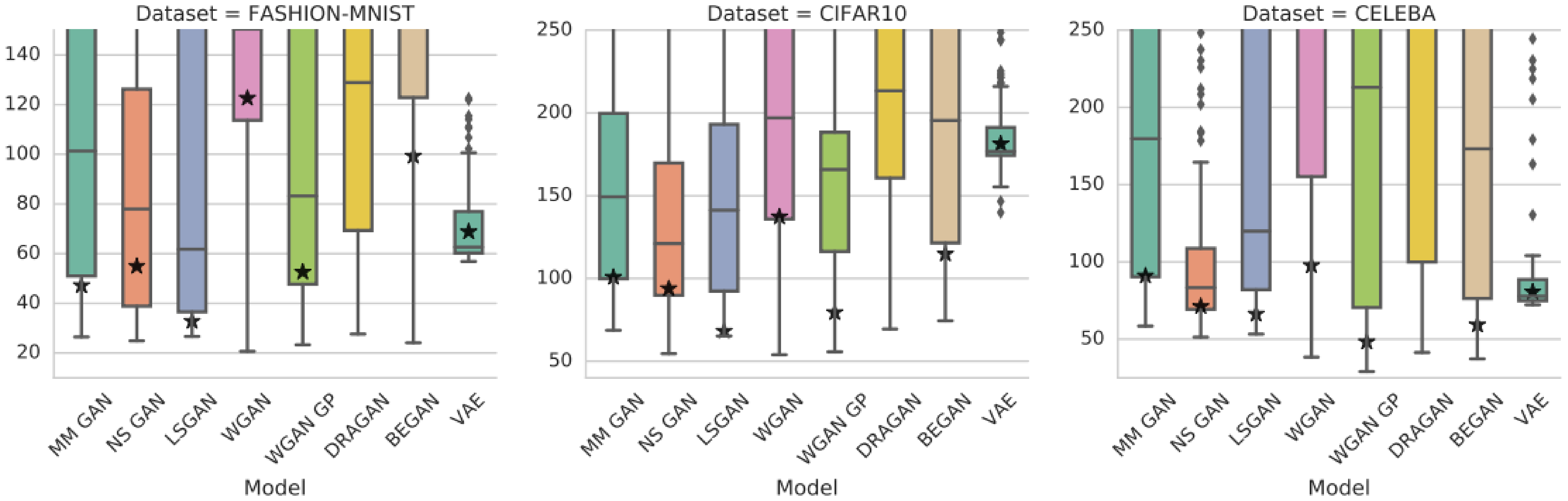
Are GANs Created Equal? Lucic&Kurach et al (Google Brain) 2018

Types of GAN losses

	Generator Loss
$_{\hat{x}\sim p_g}[\log(1-D(\hat{x}))]$	$\mathcal{L}_{\mathrm{G}}^{\mathrm{GAN}} = \mathbb{E}_{\hat{x}\sim p_g}[\log(1-D(\hat{x}))]$
$\mathbb{E}_{\hat{x}\sim p_g}[\log(1-D(\hat{x}))]$	$\mathcal{L}_{\mathrm{G}}^{\mathrm{NSGAN}} = -\mathbb{E}_{\hat{x}\sim p_g}[\log(D(\hat{x}))]$
$_g[D(\hat{x})]$	$\mathcal{L}_{\mathrm{G}}^{\mathrm{WGAN}} = -\mathbb{E}_{\hat{x}\sim p_g}[D(\hat{x})]$
$D(\alpha x + (1-\alpha\hat{x})\|_2 - 1)^2]$	$\mathcal{L}_{\mathrm{G}}^{\mathrm{WGANGP}} = -\mathbb{E}_{\hat{x}\sim p_g}[D(\hat{x})]$
$+\ \mathbb{E}_{\hat{x}\sim p_g}[D(\hat{x})^2]$	$\mathcal{L}_{\mathrm{G}}^{\mathrm{LSGAN}} = -\mathbb{E}_{\hat{x}\sim p_g}[(D(\hat{x}-1))^2]$
$[(\|\nabla D(\hat{x})\|_2 - 1)^2]$	$\mathcal{L}_{\mathrm{G}}^{\mathrm{DRAGAN}} = \mathbb{E}_{\hat{x}\sim p_g}[\log(1-D(\hat{x}))]$
$-\ k_t\mathbb{E}_{\hat{x}\sim p_g}[\|\hat{x} - \mathrm{AE}(\hat{x})\|_1]$	$\mathcal{L}_{\mathrm{G}}^{\mathrm{BEGAN}} = \mathbb{E}_{\hat{x}\sim p_g}[\|\hat{x} - \mathrm{AE}(\hat{x})\|_1]$

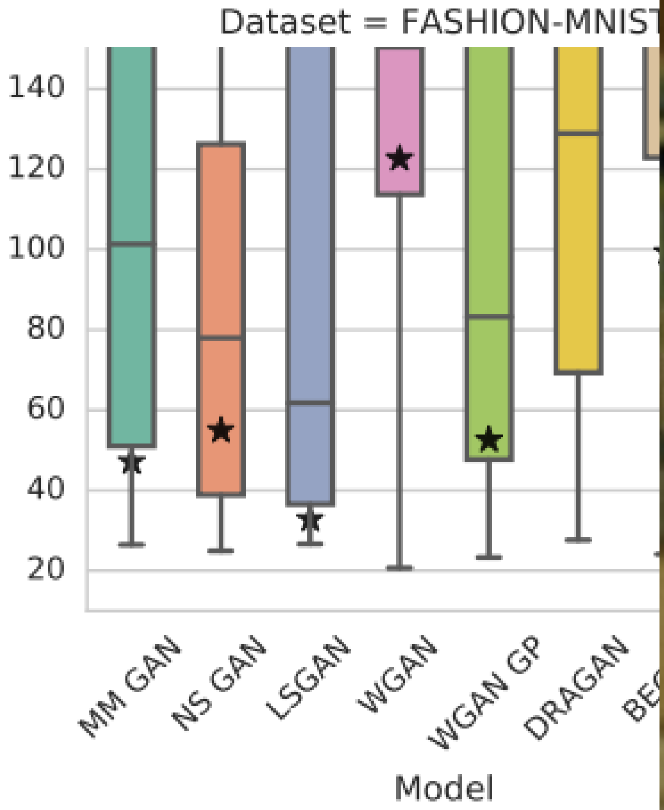
Are GANs Created Equal? Lucic&Kurach et al (Google Brain) 2018

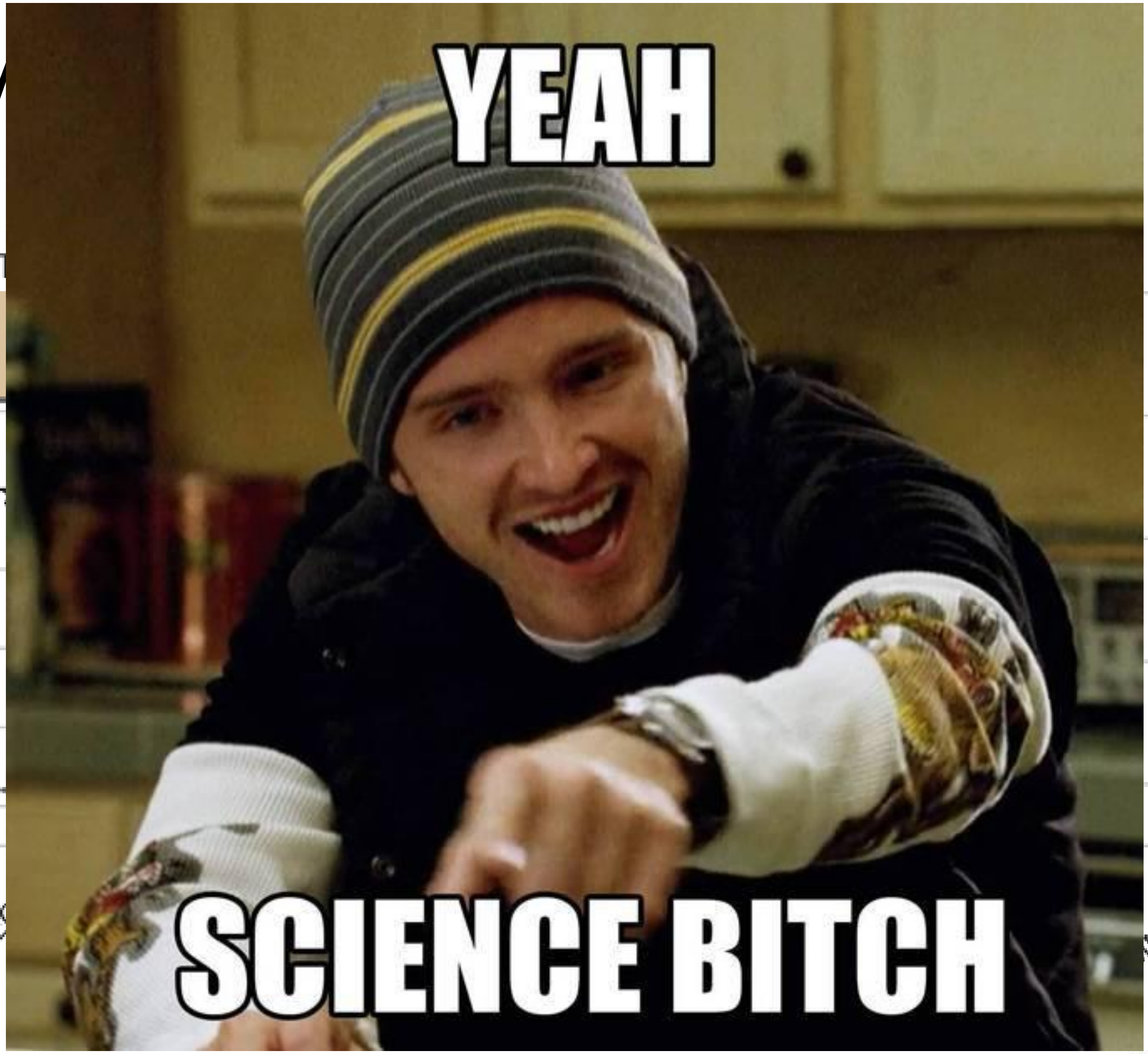
Types of GAN losses

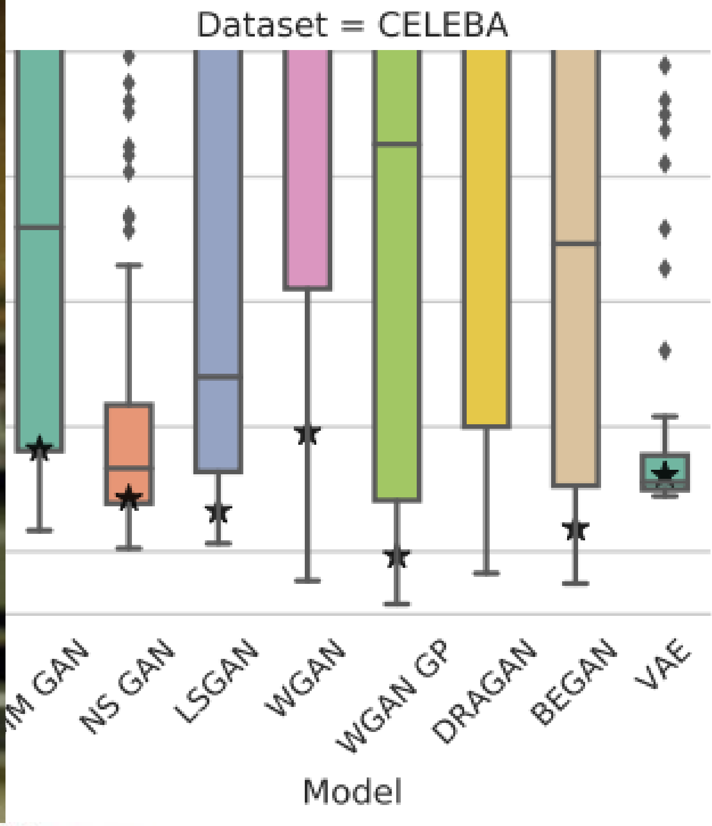


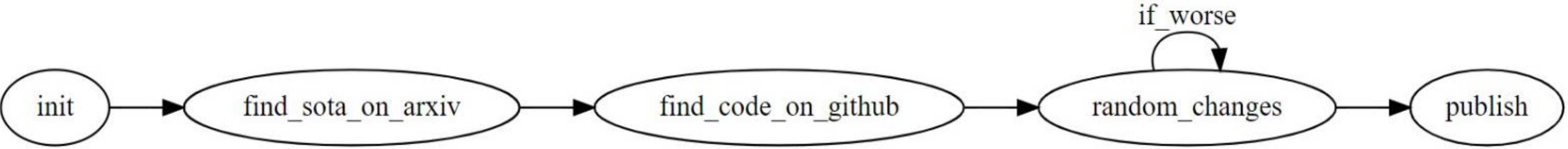
Are GANs Created Equal? Lucic&Kurach et al (Google Brain) 2018

Types of G









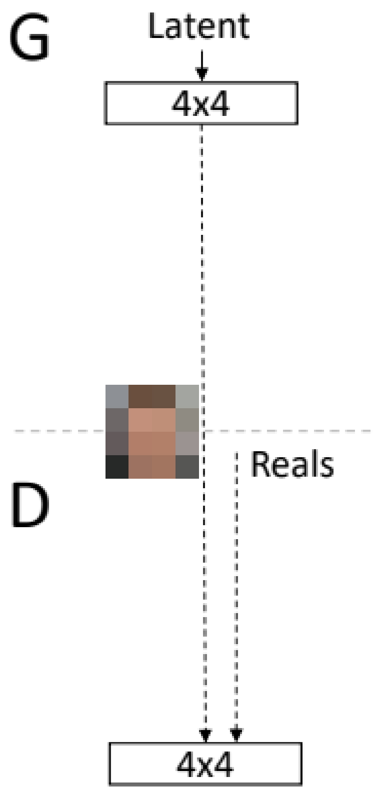
Mark Saroufim

Progressive Grow

Progressive Growing of GAN, Karras et al., Feb2018

Slide by Sefi Bell-Kligler & Akhiad Bercovich

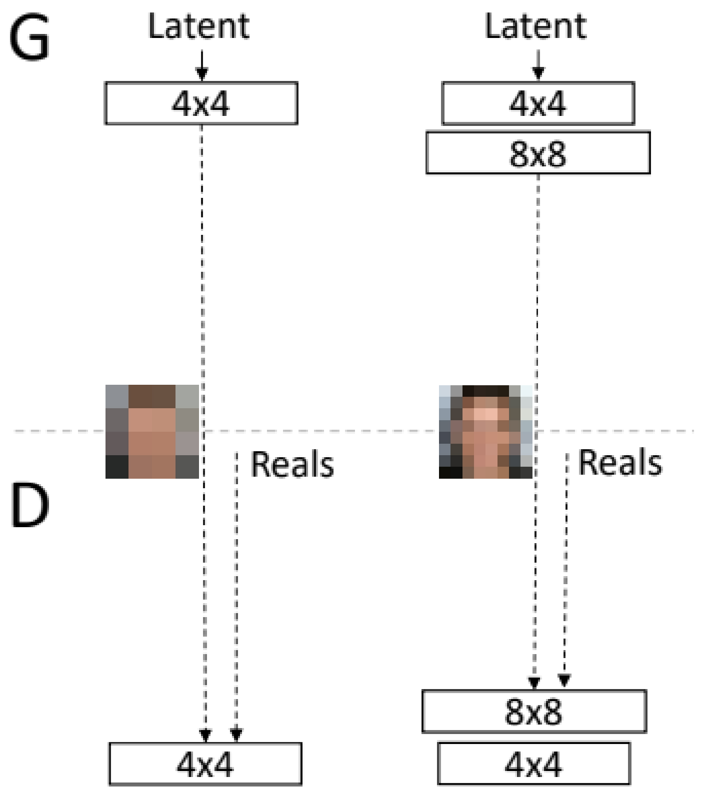
Progressive Grow



Progressive Growing of GAN, Karras et al., Feb2018

Slide by Sefi Bell-Kligler & Akhiad Bercovich

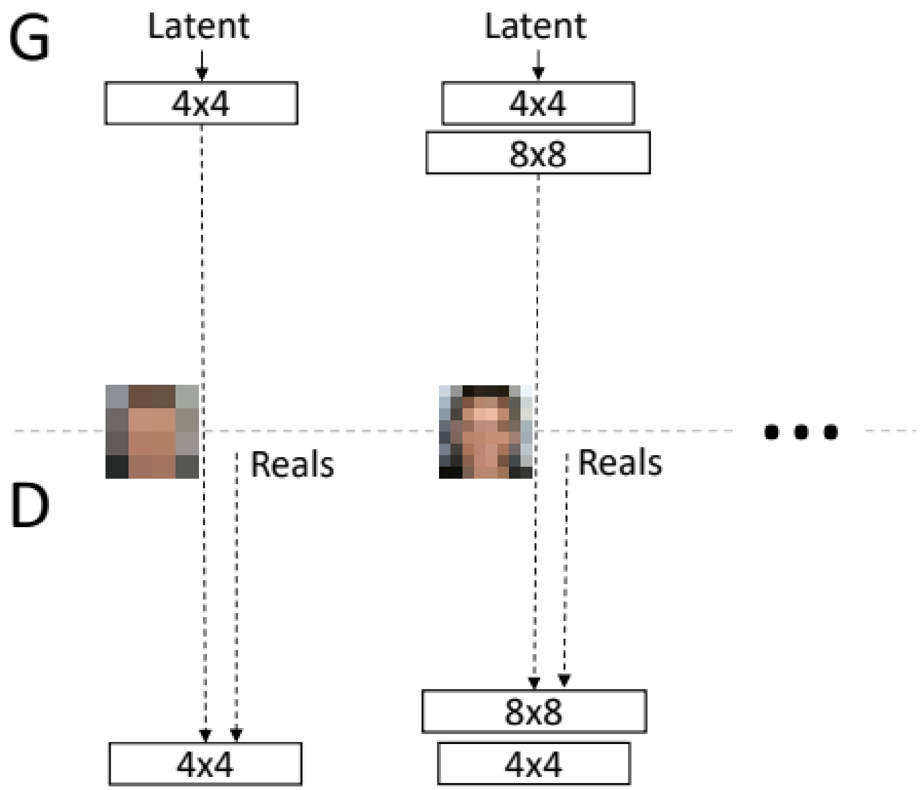
Progressive Grow



Progressive Growing of GAN, Karras et al., Feb2018

Slide by Sefi Bell-Kligler & Akhiad Bercovich

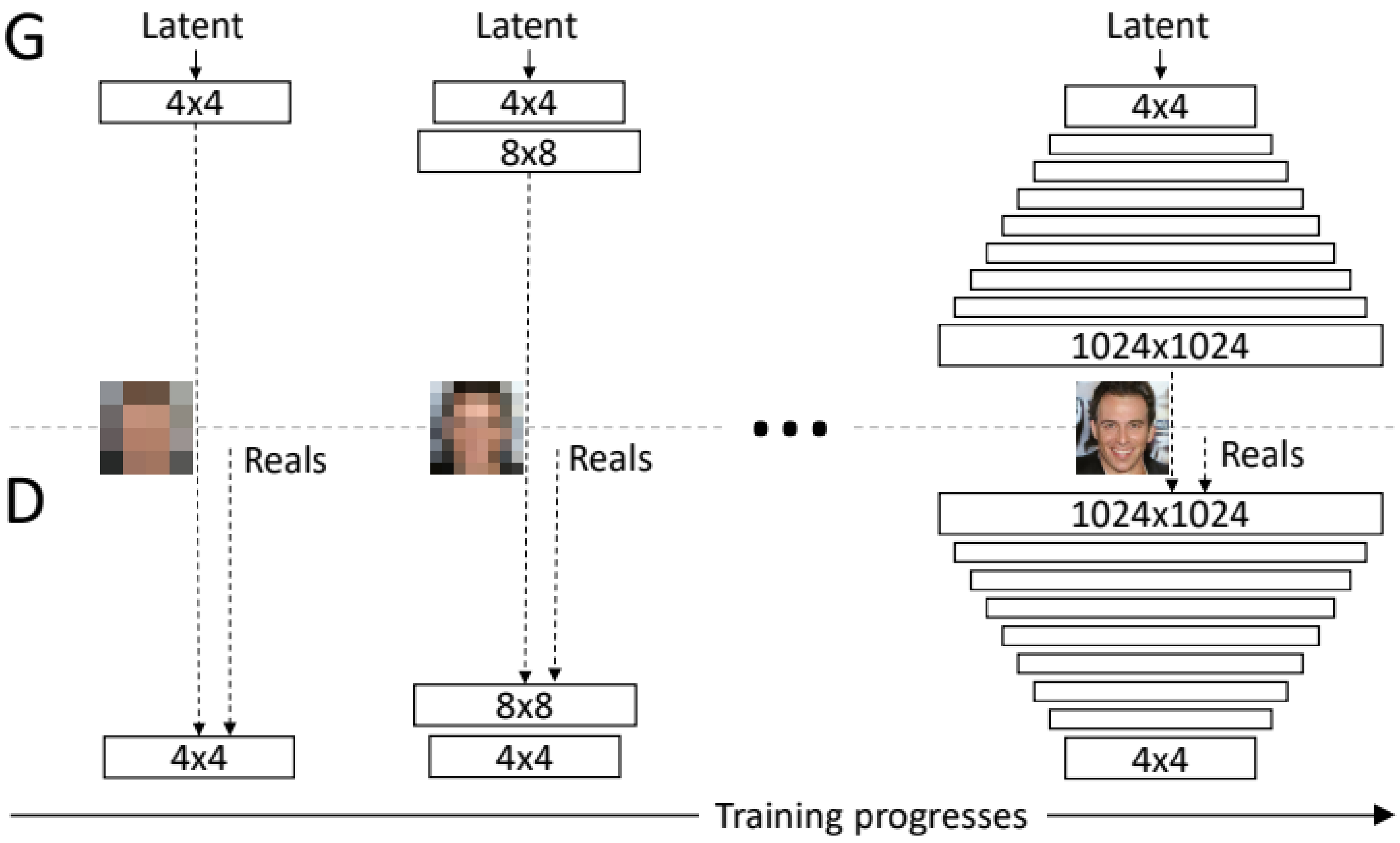
Progressive Grow



Progressive Growing of GAN, Karras et al., Feb2018

Slide by Sefi Bell-Kligler & Akhiad Bercovich

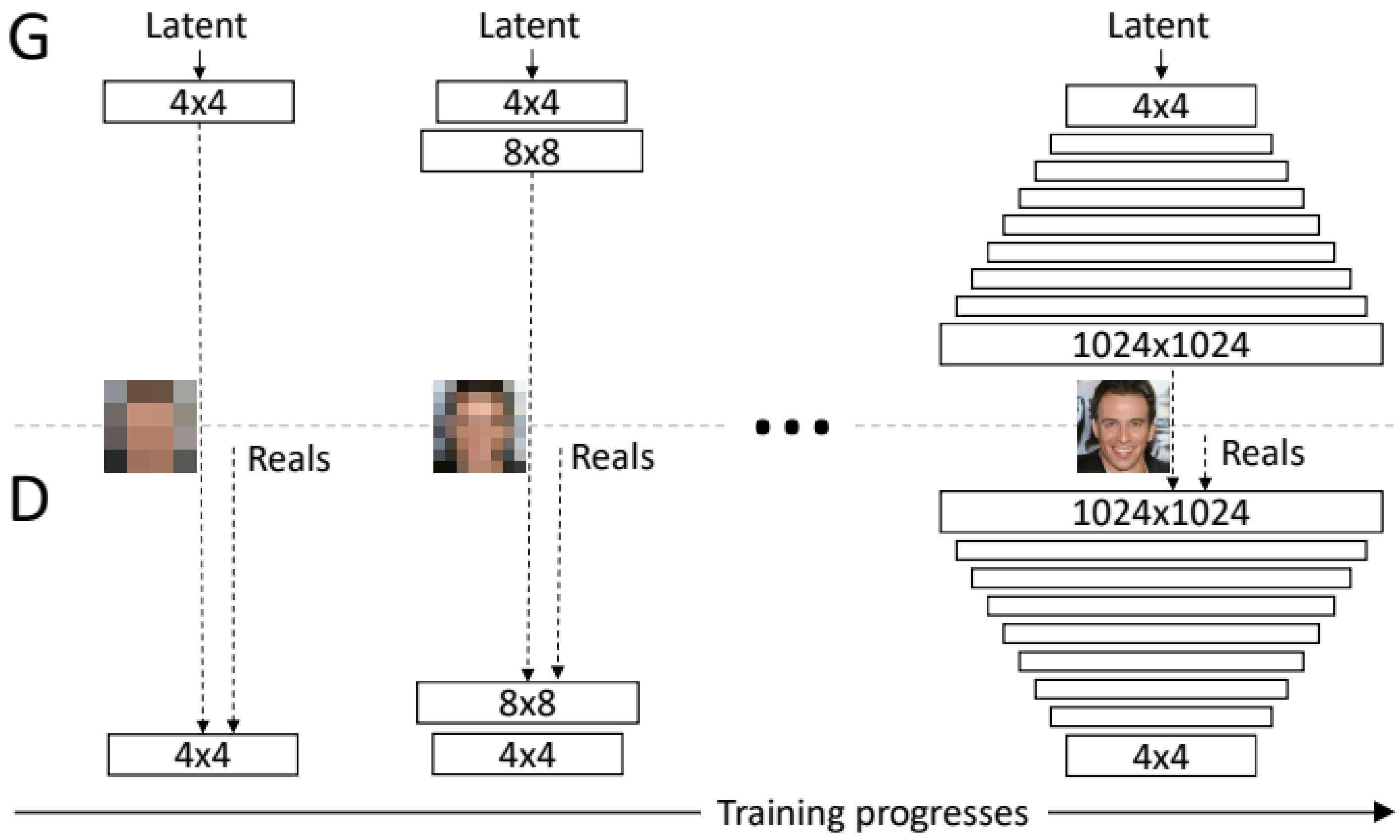
Progressive Grow



Progressive Growing of GAN, Karras et al., Feb2018

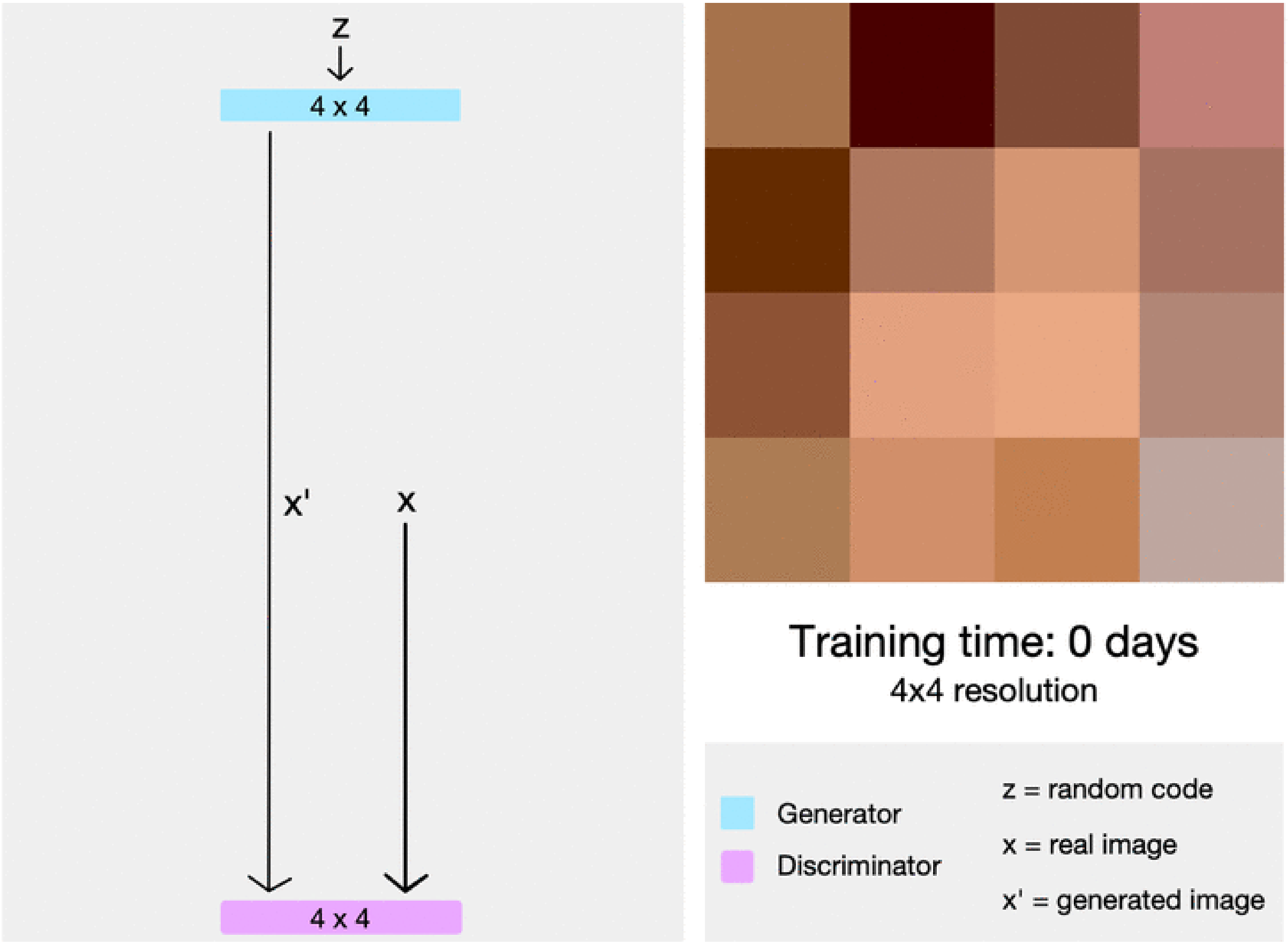
Slide by Sefi Bell-Kligler & Akhiad Bercovich

Progressive Grow



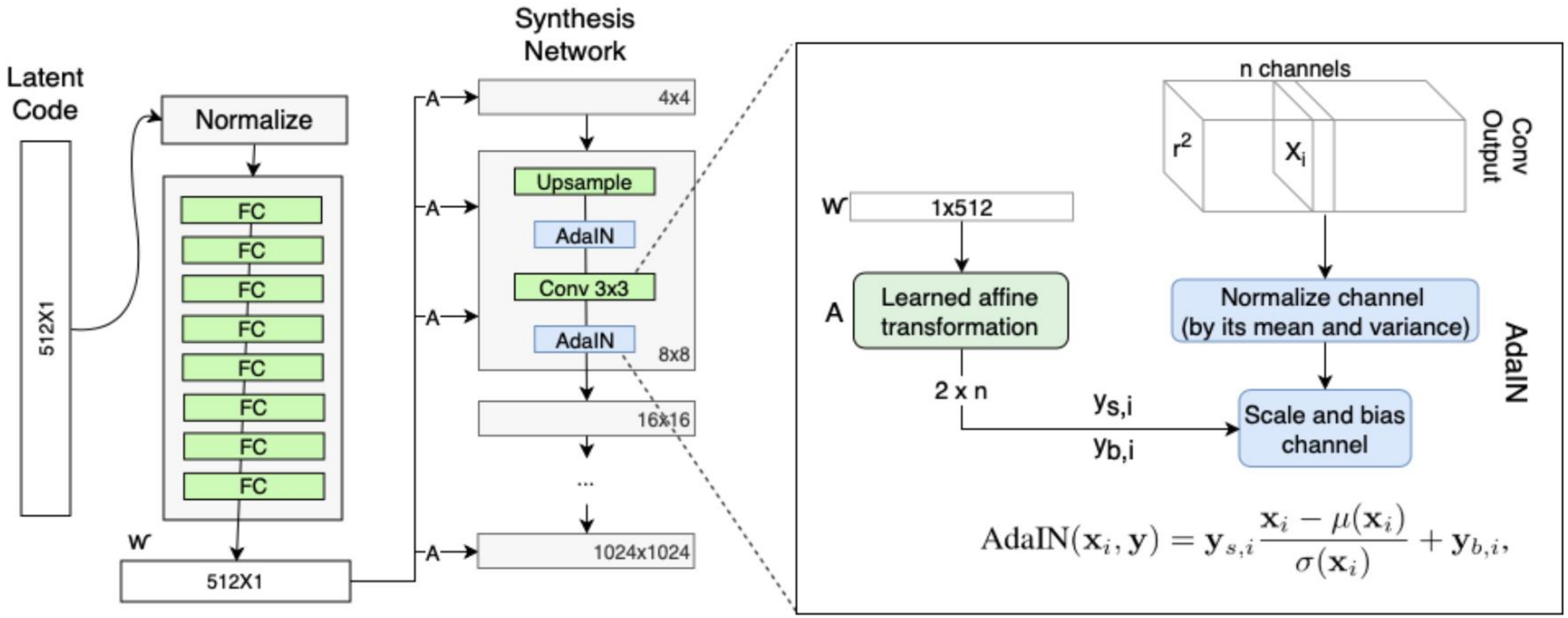
Progressive Growing of GAN, Karras et al., Feb2018

Slide by Sefi Bell-Kligler & Akhiad Bercovich



Progressive Growing of GAN, Karras et al., Feb2018

Style Modules (AdaIN)



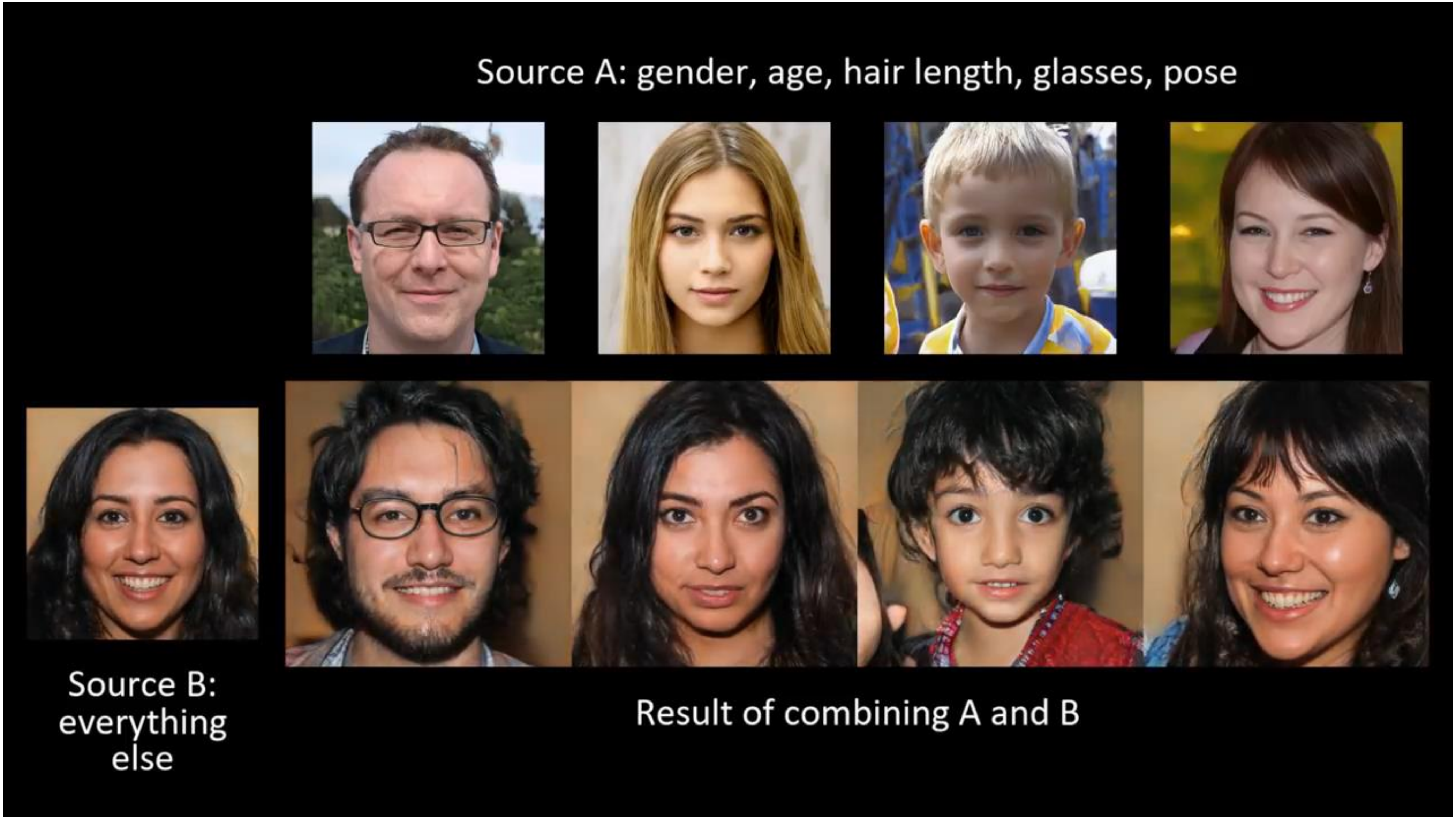
$$\mathrm{AdaIN}(\mathbf{x}_i, \mathbf{y}) = \mathbf{y}_{s,i}\frac{\mathbf{x}_i - \mu(\mathbf{x}_i)}{\sigma(\mathbf{x}_i)} + \mathbf{y}_{b,i},$$

The generator's Adaptive Instance Normalization (AdaIN)

StyleGAN, Karras et al. NVIDIA 2019

Results

Results


Source A: gender, age, hair length, glasses, pose
Source B: everything else
Result of combining A and B

Animation by Sefi Bell-Kligler & Akhiad Bercovich

https://thispersondoesnotexist.com/

https://whichfaceisreal.com/

Animation by Sefi Bell-Kligler & Akhiad Bercovich

https://thispersondoesnotexist.com/

https://whichfaceisreal.com/

Animation by Sefi Bell-Kligler & Akhiad Bercovich

https://thispersondoesnotexist.com/

https://whichfaceisreal.com/

Animation by Sefi Bell-Kligler & Akhiad Bercovich

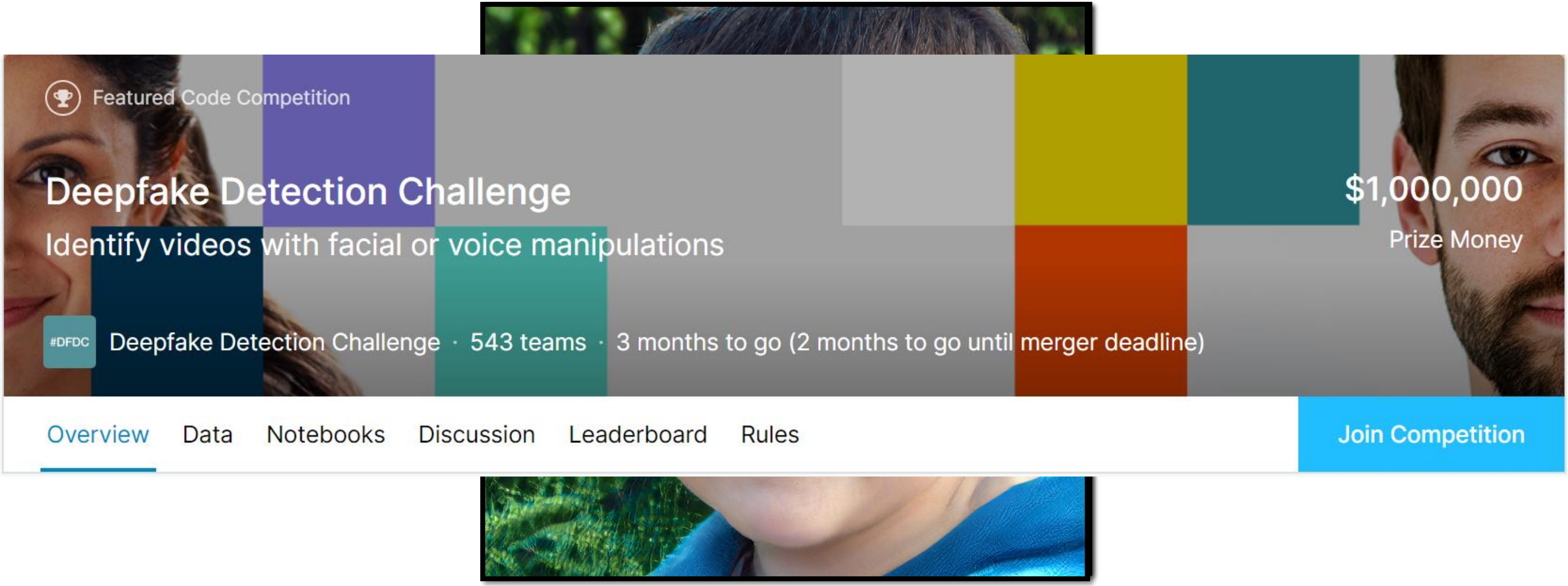
https://thispersondoesnotexist.com/

https://whichfaceisreal.com/

Animation by Sefi Bell-Kligler & Akhiad Bercovich

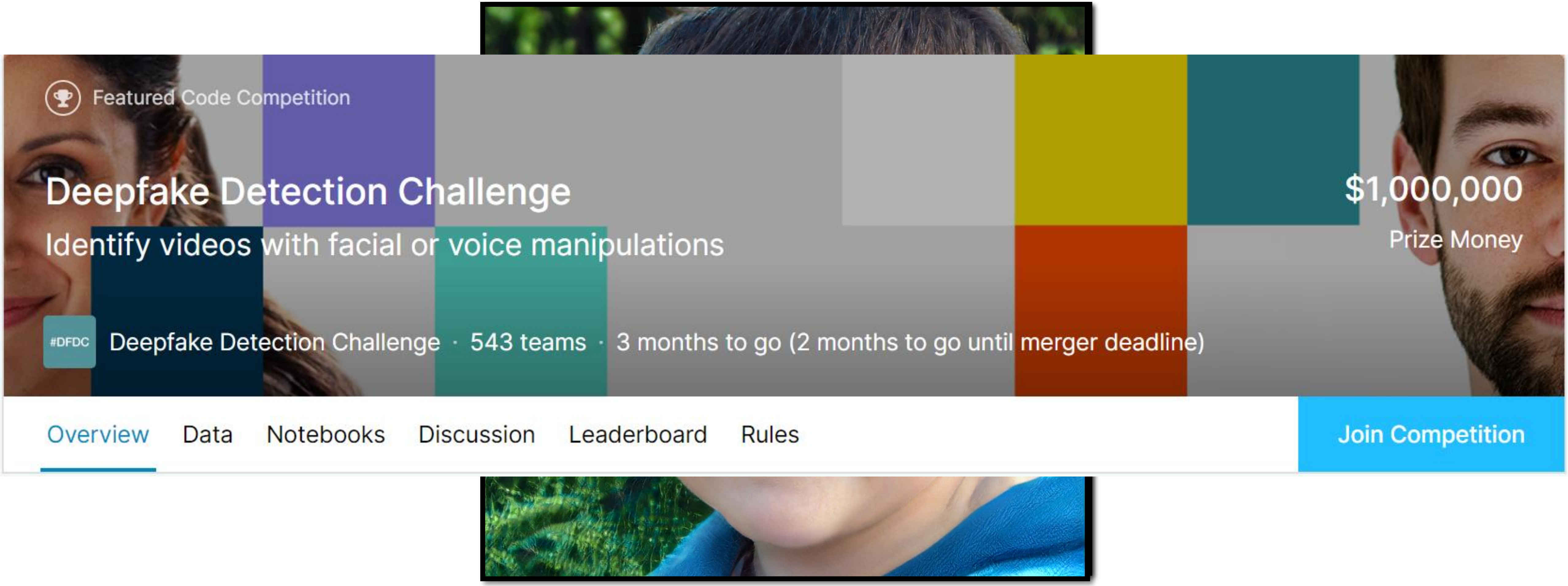
https://thispersondoesnotexist.com/

https://whichfaceisreal.com/



https://thispersondoesnotexist.com/

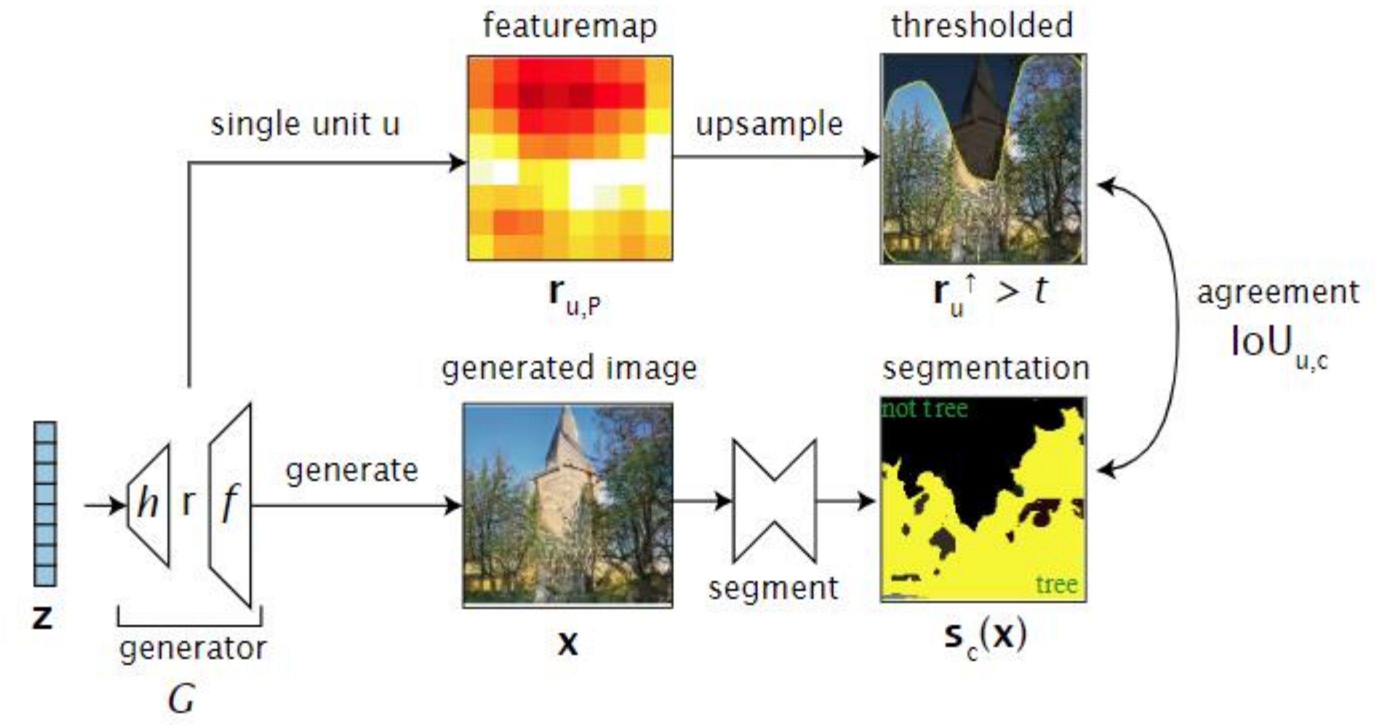
https://whichfaceisreal.com/



https://thispersondoesnotexist.com/

https://whichfaceisreal.com/ https://thisxdoesnotexist.com/

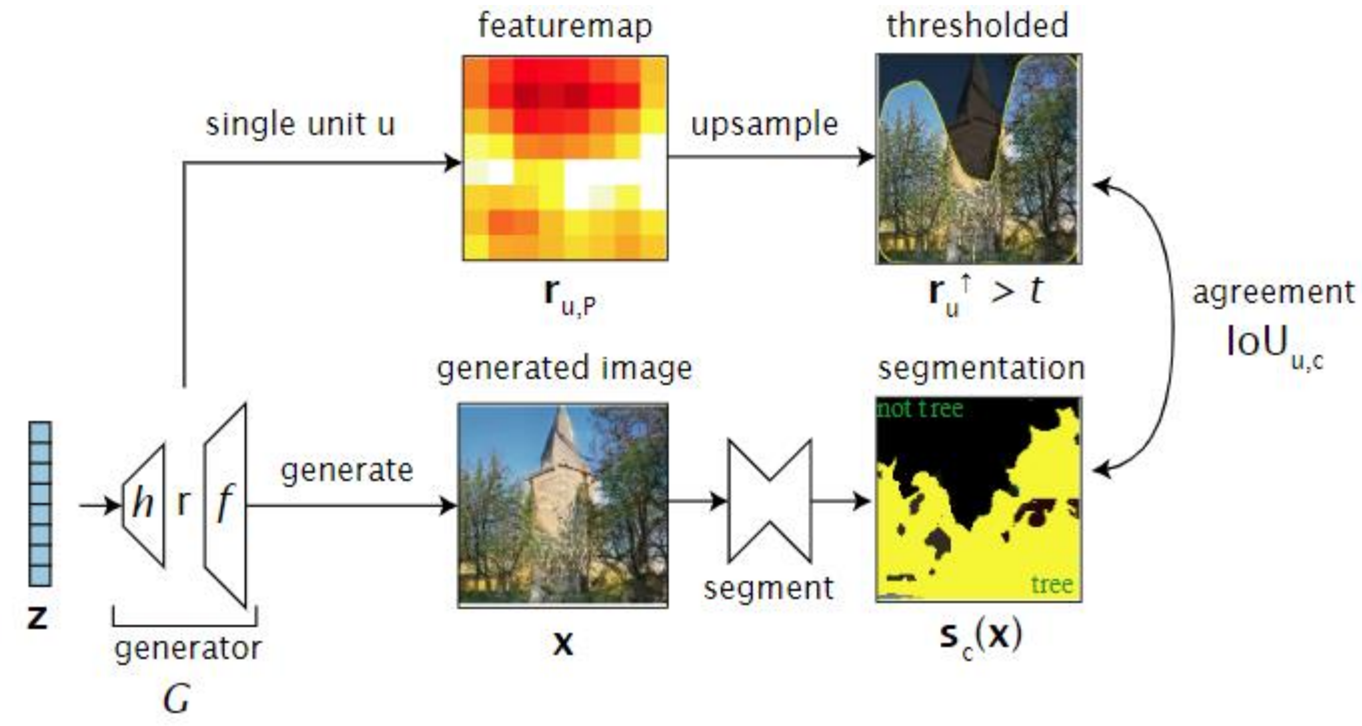
GAN Dissection

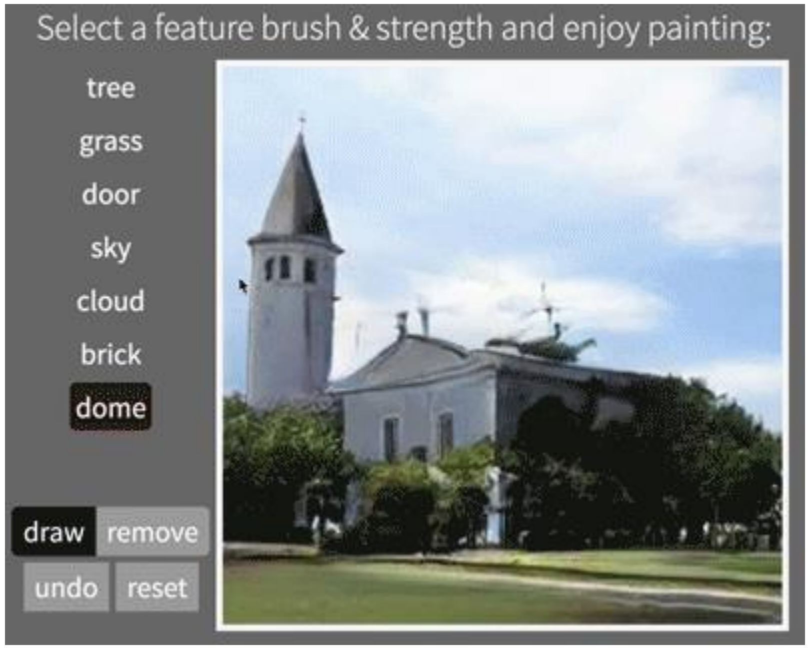


David Bau, Jun-Yan Zhu, Hendrik Strobelt, Bolei Zhou, Joshua B. Tenenbaum, William T. Freeman, Antonio Torralba

http://gandissect.res.ibm.com/ganpaint.html?project=churchoutdoor&layer=layer4

GAN Dissection





David Bau, Jun-Yan Zhu, Hendrik Strobelt, Bolei Zhou, Joshua B. Tenenbaum, William T. Freeman, Antonio Torralba

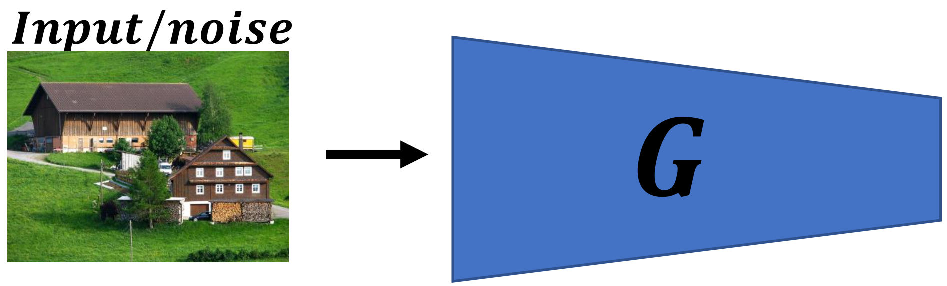
http://gandissect.res.ibm.com/ganpaint.html?project=churchoutdoor&layer=layer4

Training a GAN on a single image

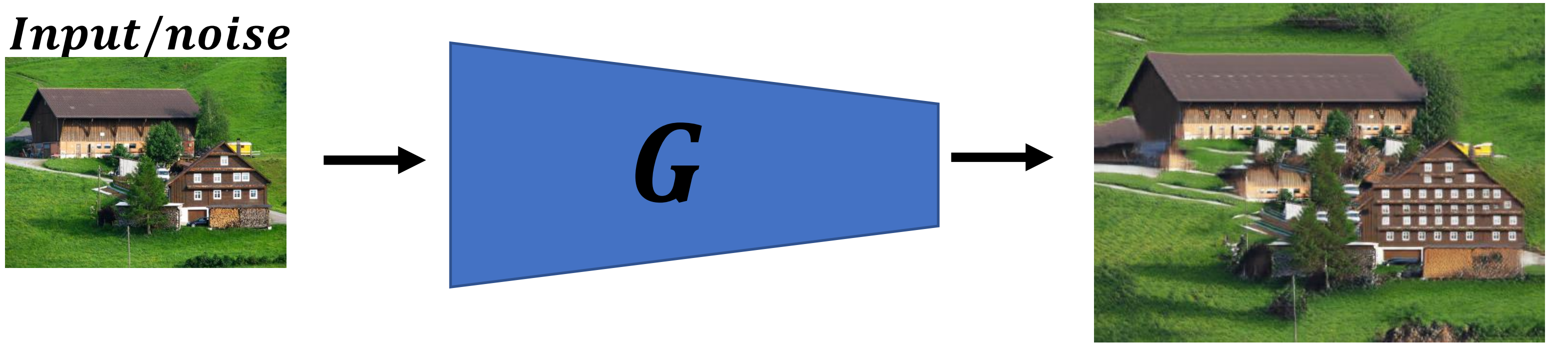
Training a GAN on a single image

Input/noise

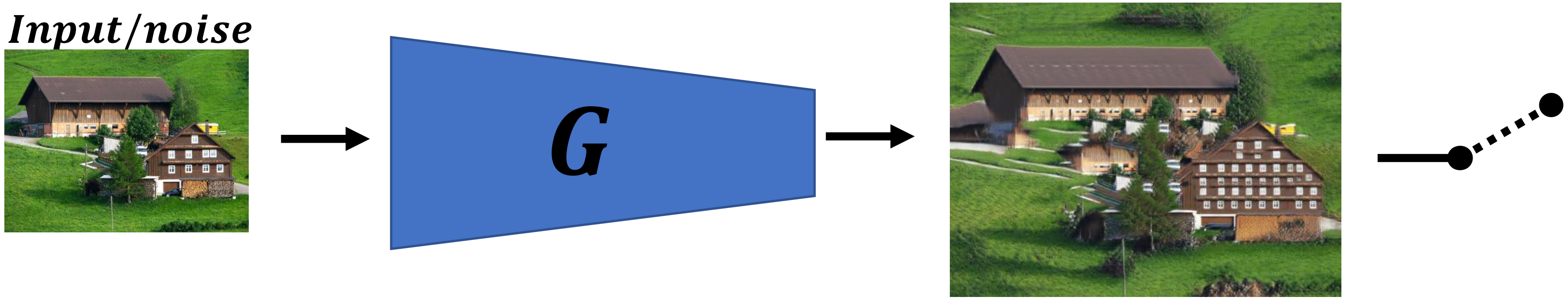
Training a GAN on a single image



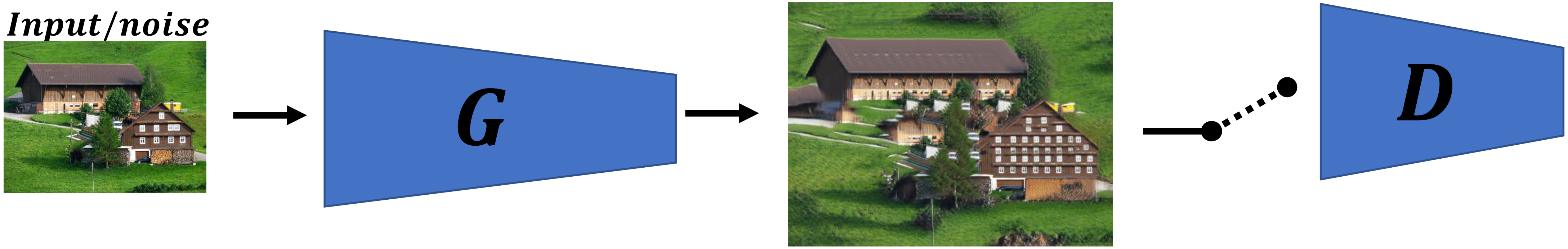
Training a GAN on a single image



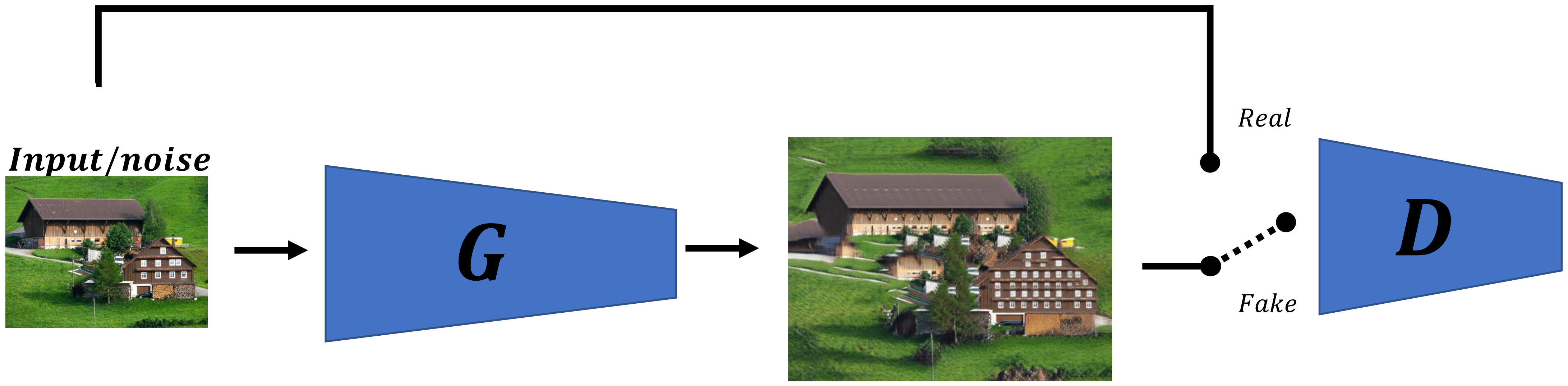
Training a GAN on a single image



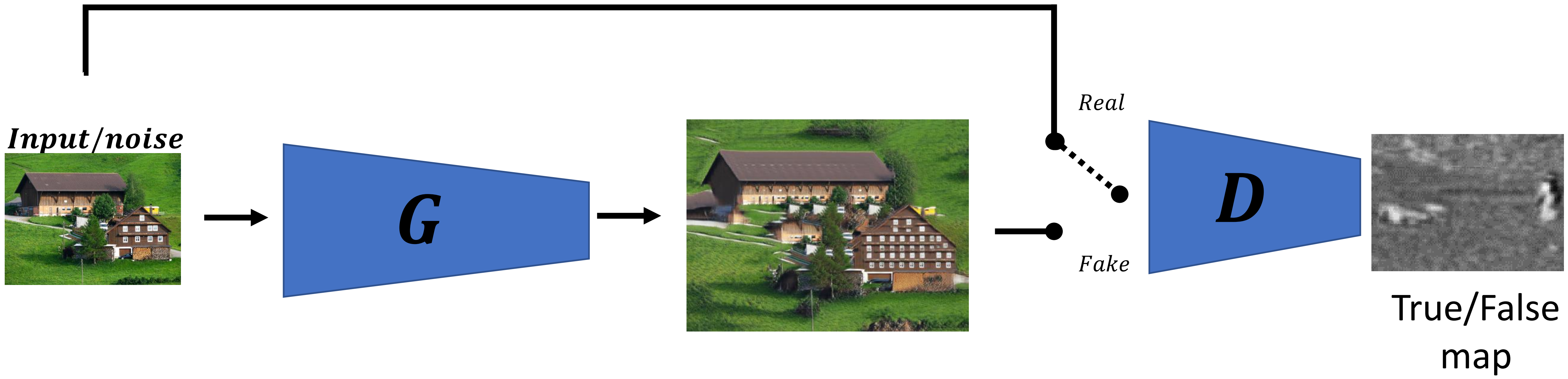
Training a GAN on a single image



Training a GAN on a single image

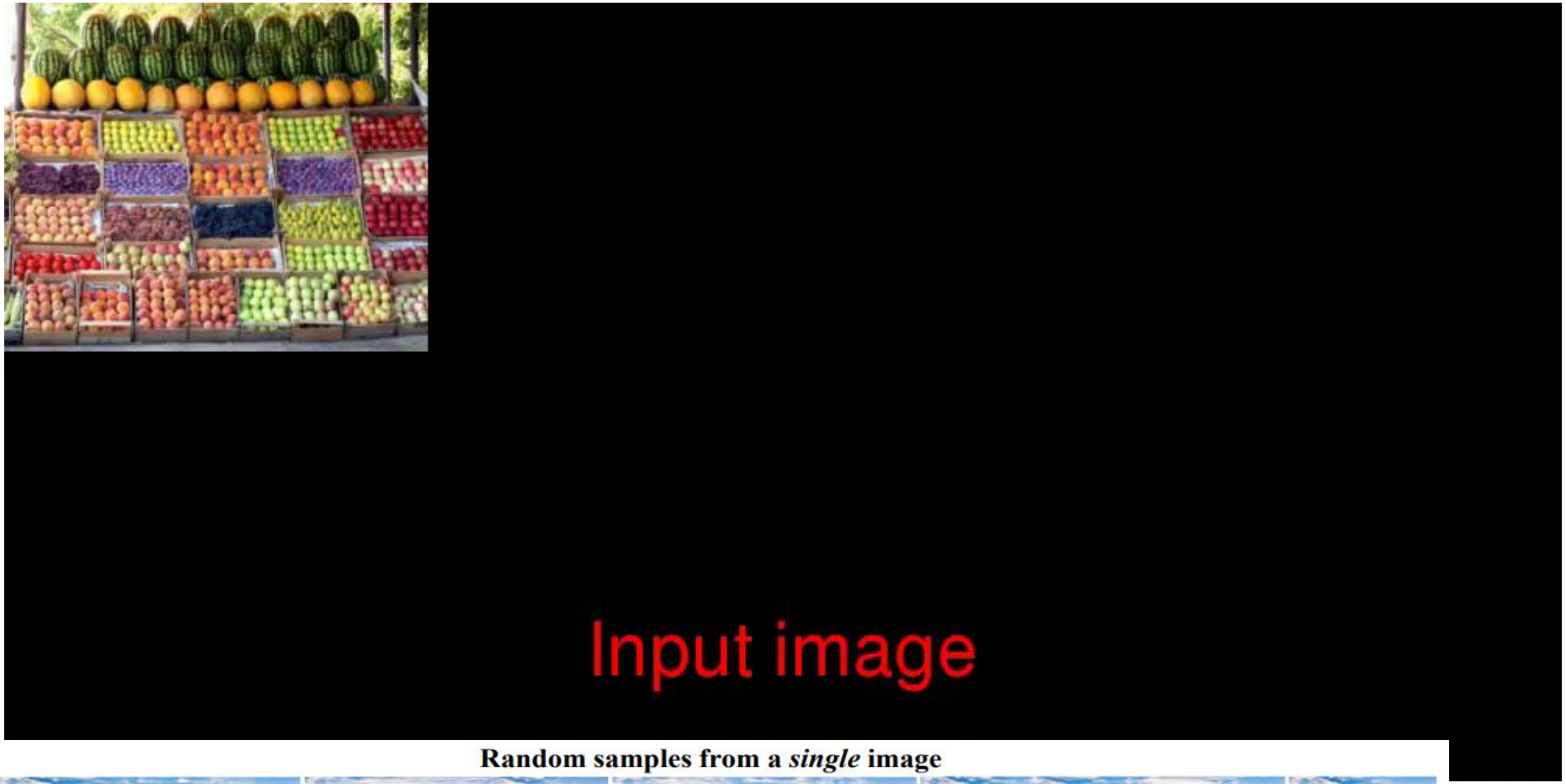


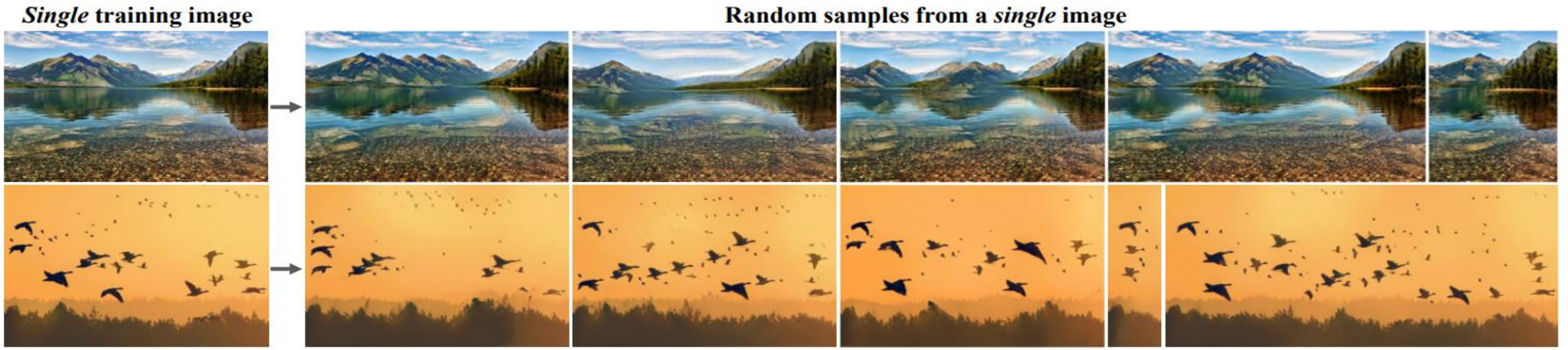
Training a GAN on a single image



Training a GAN on a single image

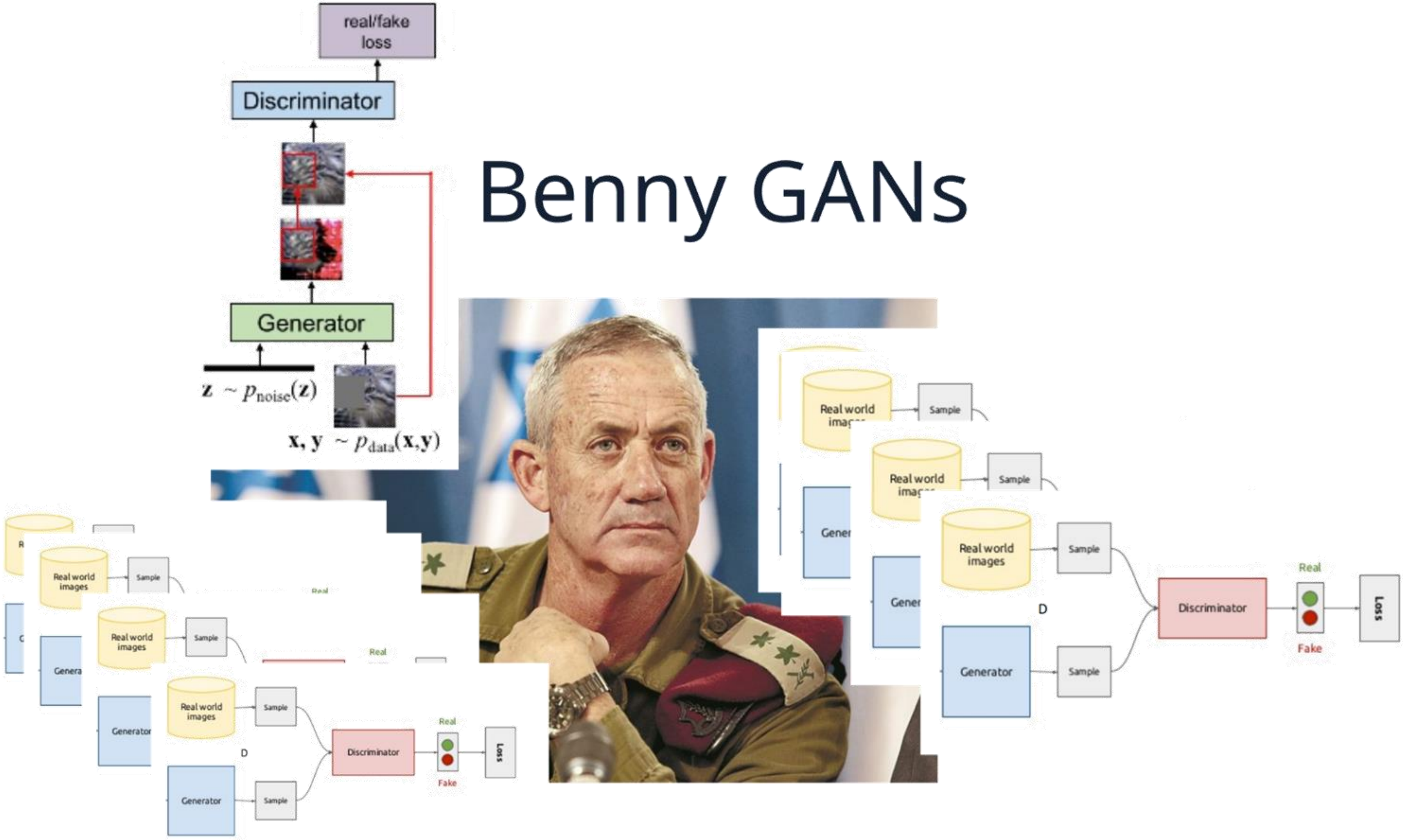
InGAN (Shocher, Bagon, Isola, Irani)





SinGAN (Rott-Shaham, Dekel, Michaeli)

Thanks



Benny GANs
real/fake loss
Discriminator
Generator
z ~ pnoise(z)
x, y ~ pdata(x,y)
Real world images
Sample
Generator
Sample
D
Discriminator
Real
Fake
Loss