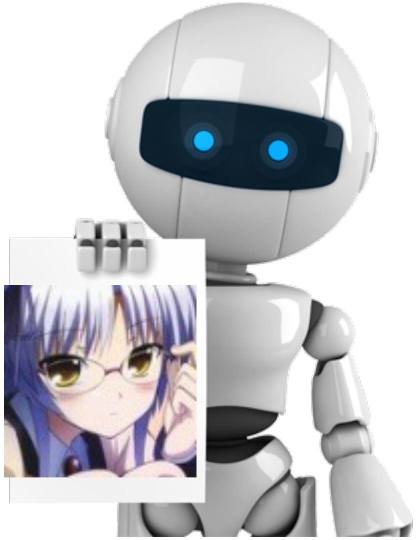
# Generative models

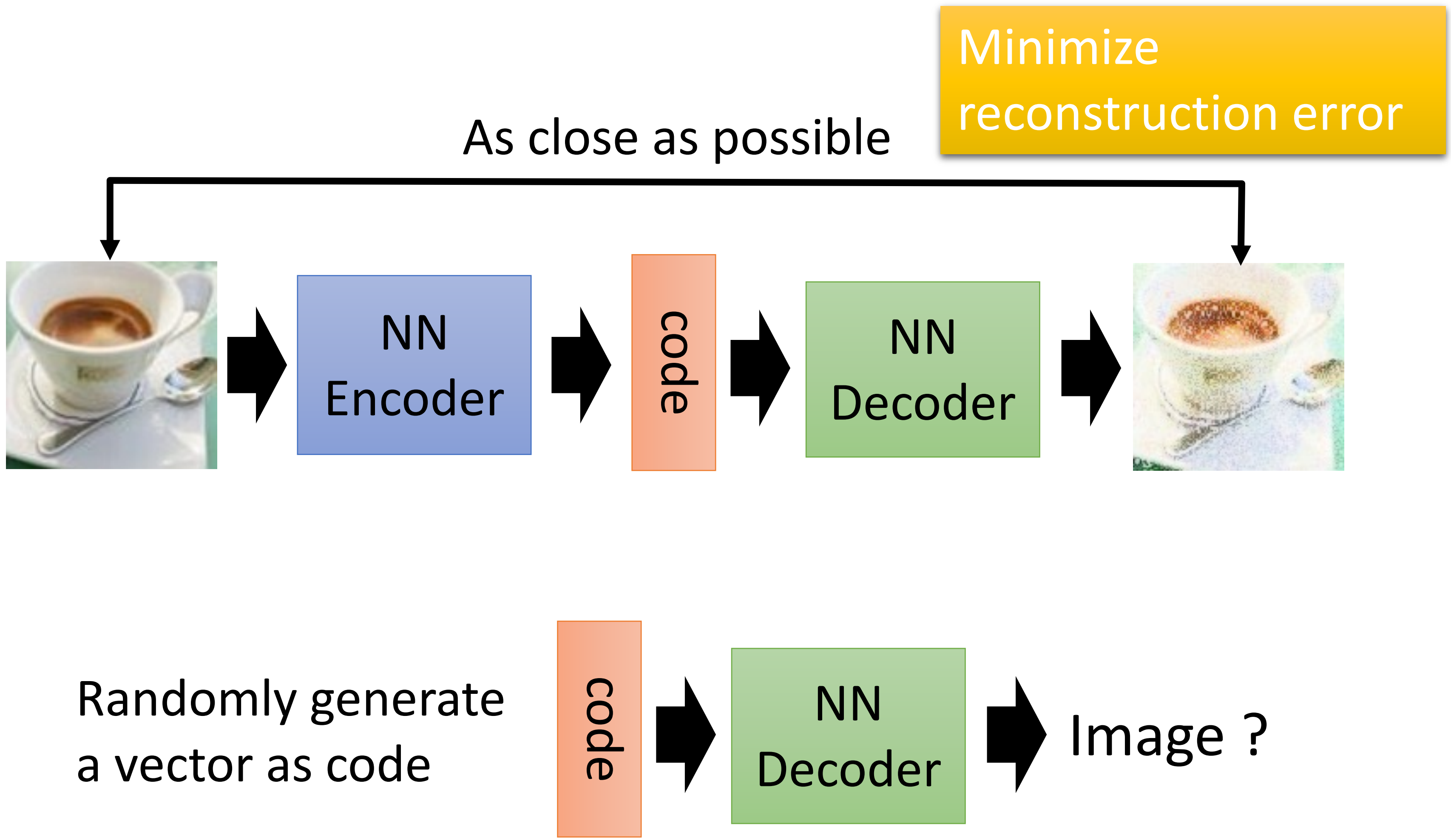
## Outline

1. **Preview: Auto-Encoders, VAE**
2. Generative models with GAN
3. GAN architectures

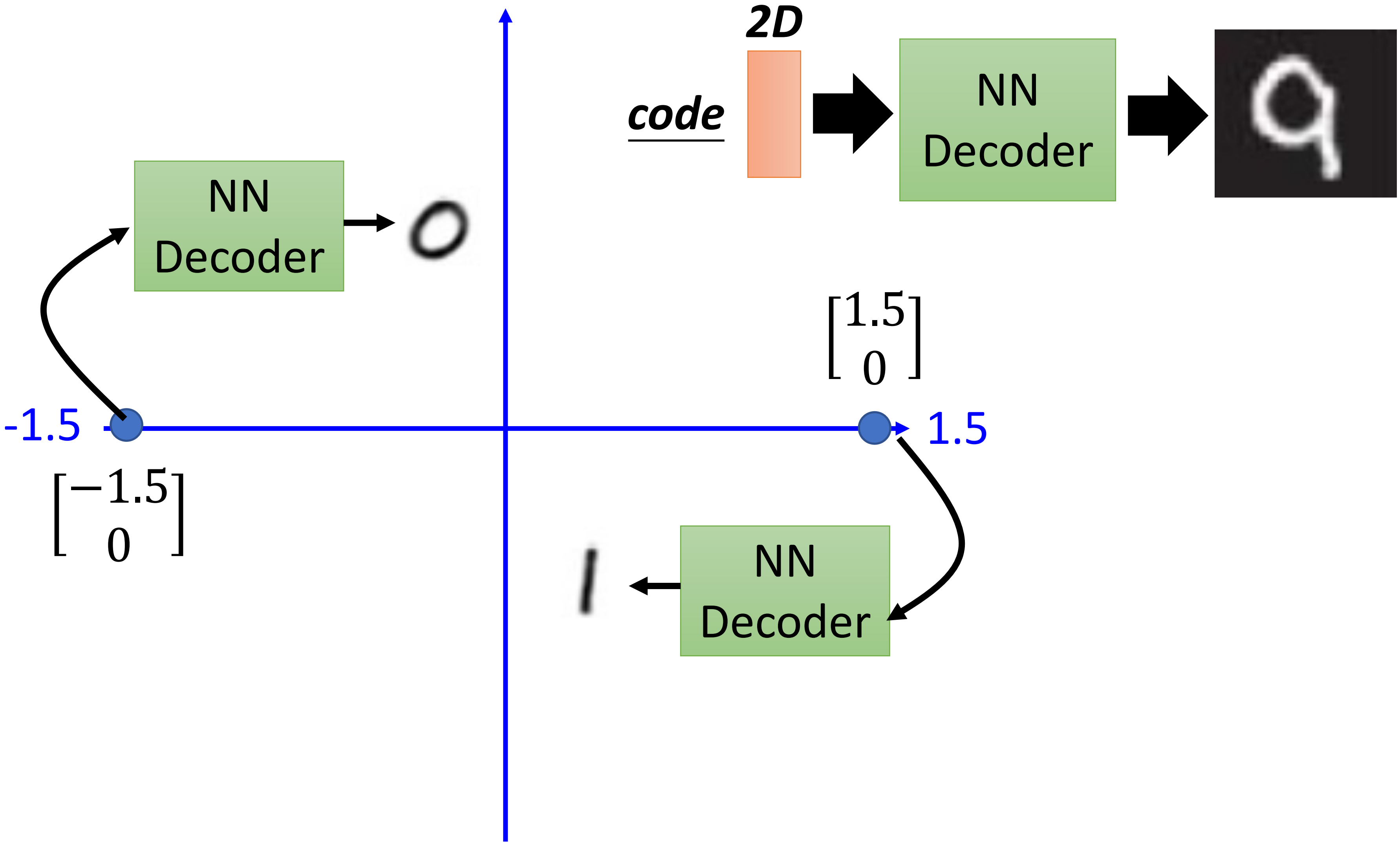


Drawing? => learning from examples

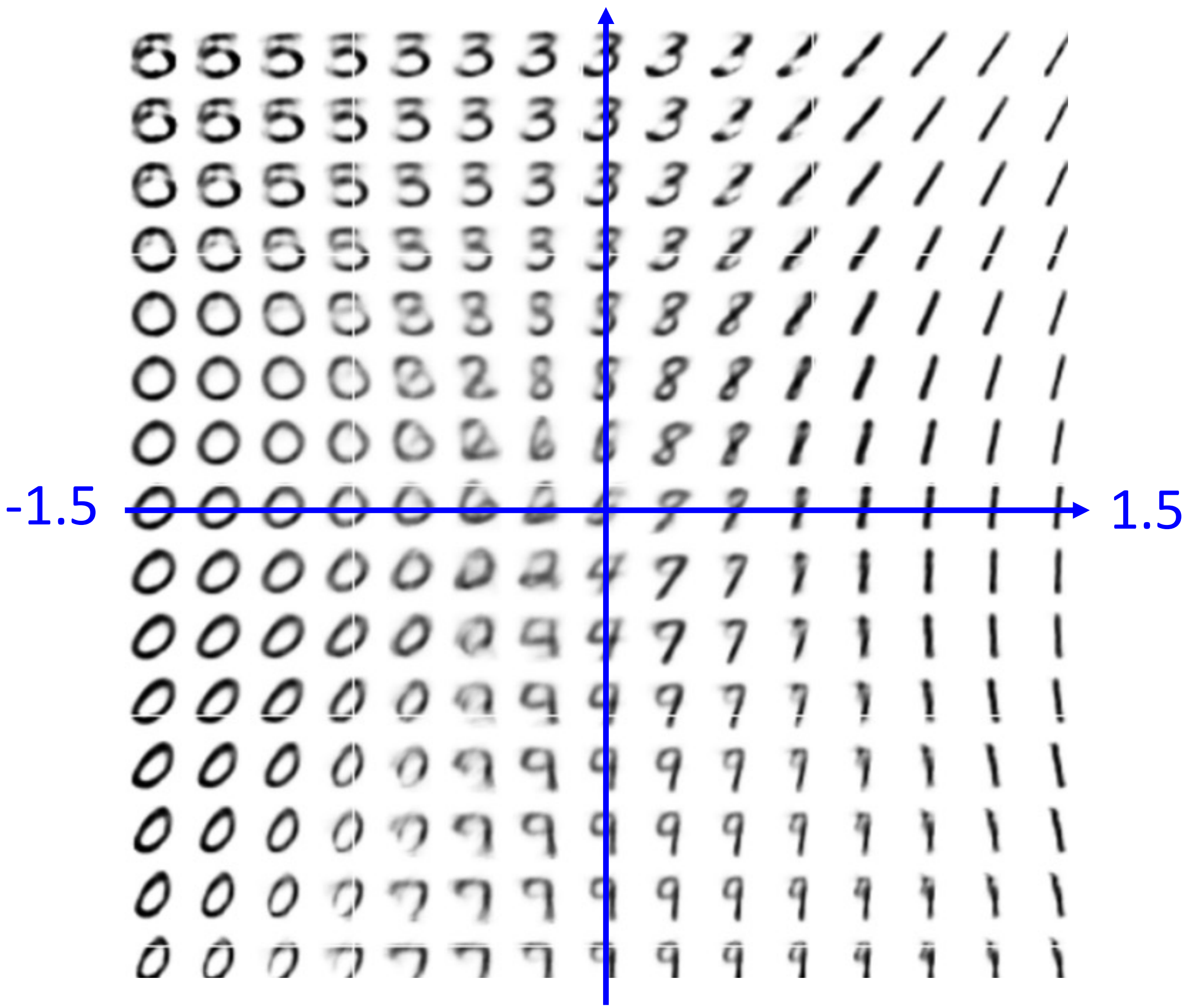
# Review: Auto-encoder

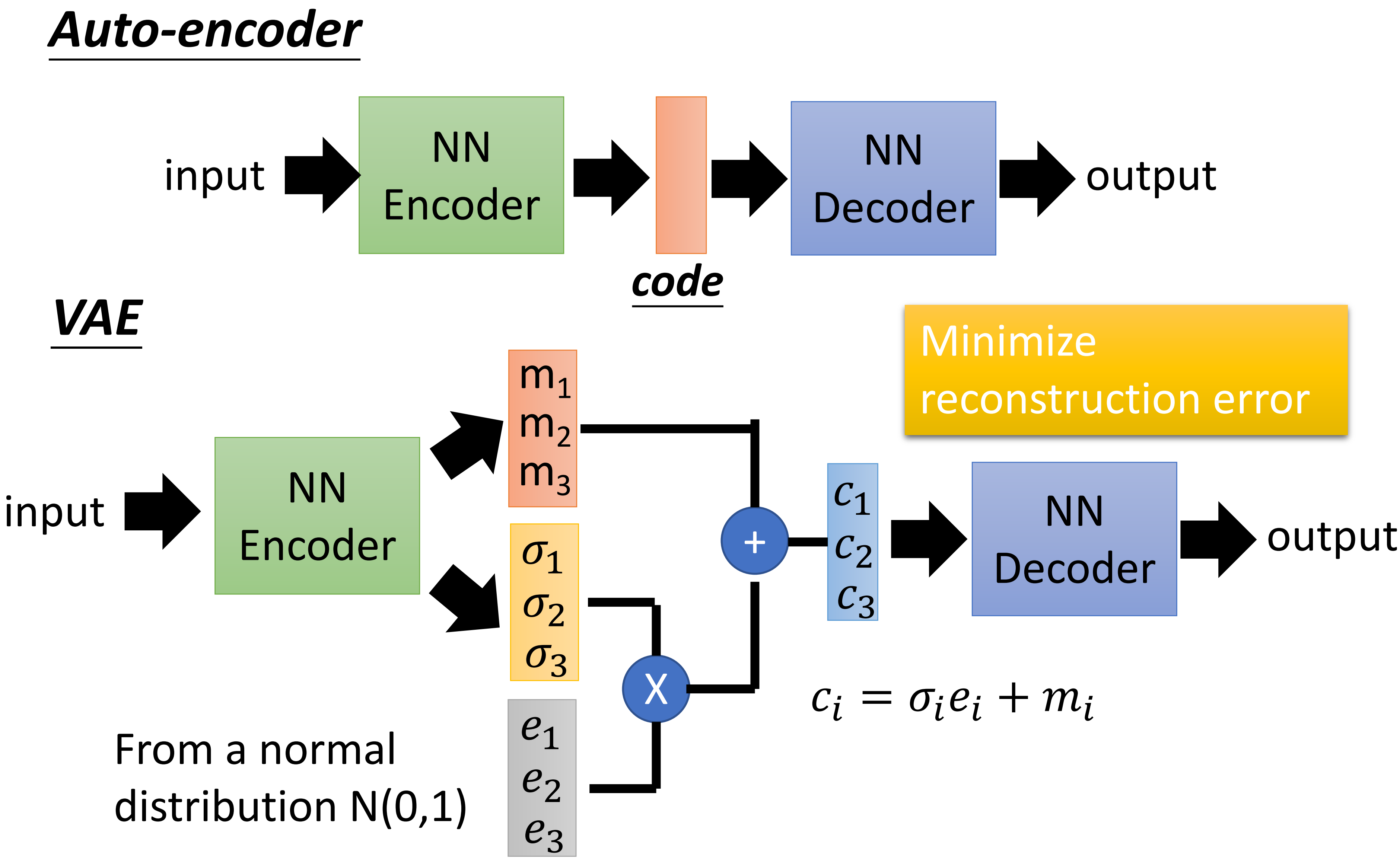


# Review: Auto-encoder



# Review: Auto-encoder

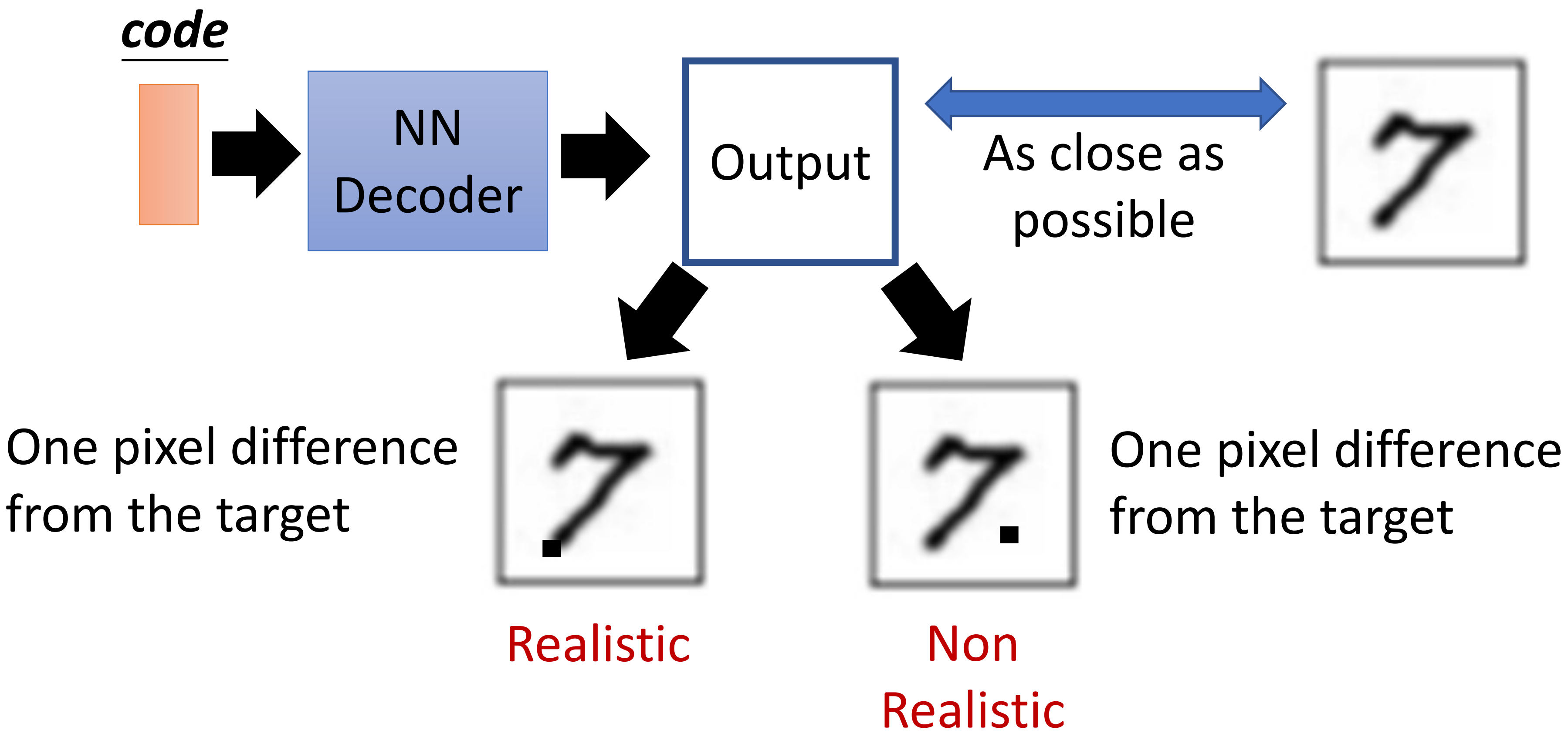





Auto-Encoding Variational Bayes, https://arxiv.org/abs/1312.6114

# Problems of AE/VAE

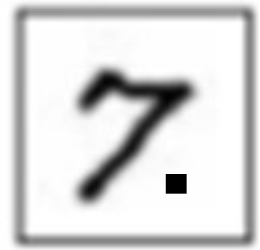
- It does not really try to simulate real images



# Problems of AE/VAE

GAN to tackle this pb:

Realistic



Non Realistic

GAN: generative adversarial networks

## Game scenario:

**Player1, Generator,** produces samples
**Player2,** – Its adversary **Discriminator**, attempts to distinguish real samples from fake generated ones (produced by Player1) !

Player1 aims at producing Realistic images to fool the Player2

# Generative models

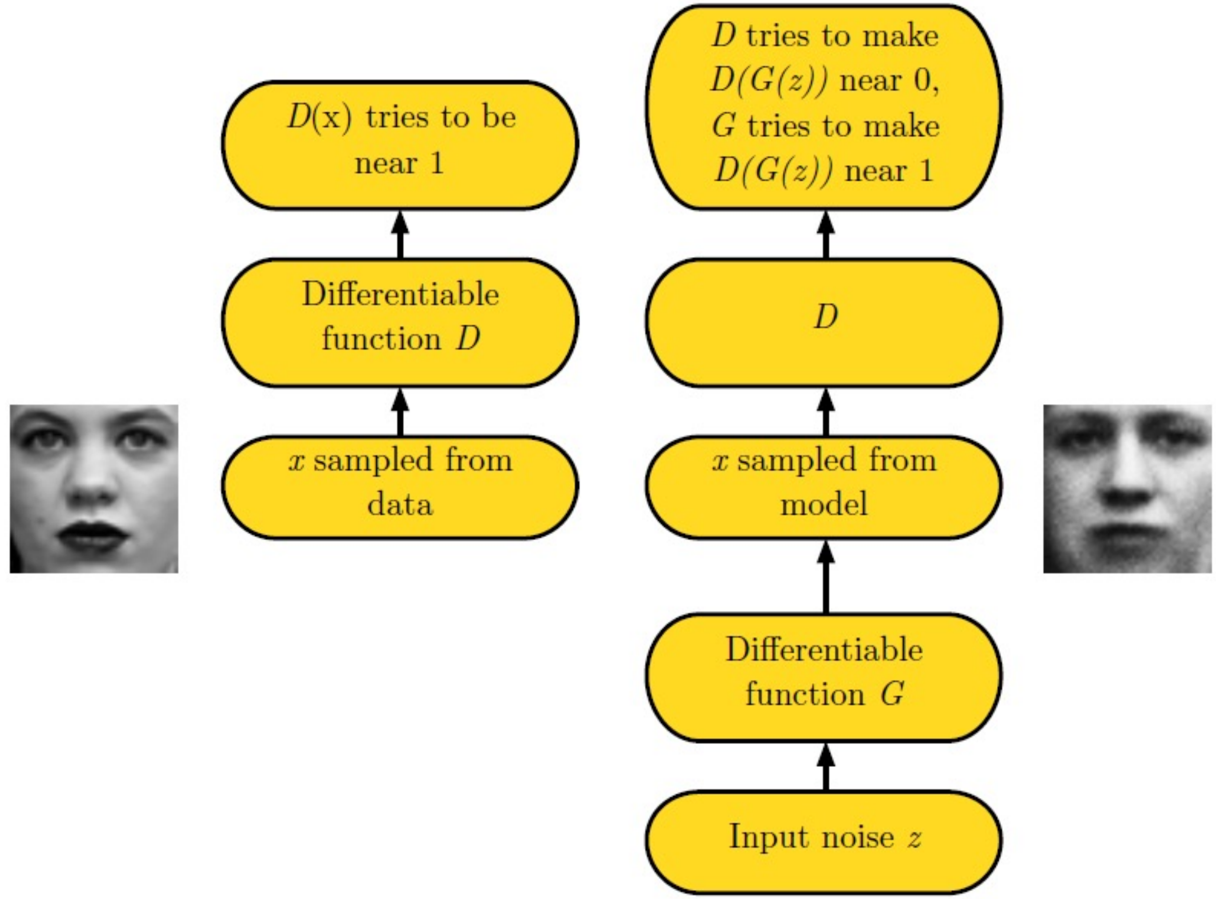
## Outline

1. Preview: Auto-Encoders, VAE
2. **Generative models with GAN**
    - GAN Algorithm

# Adversarial Nets Framework

Game scenario:

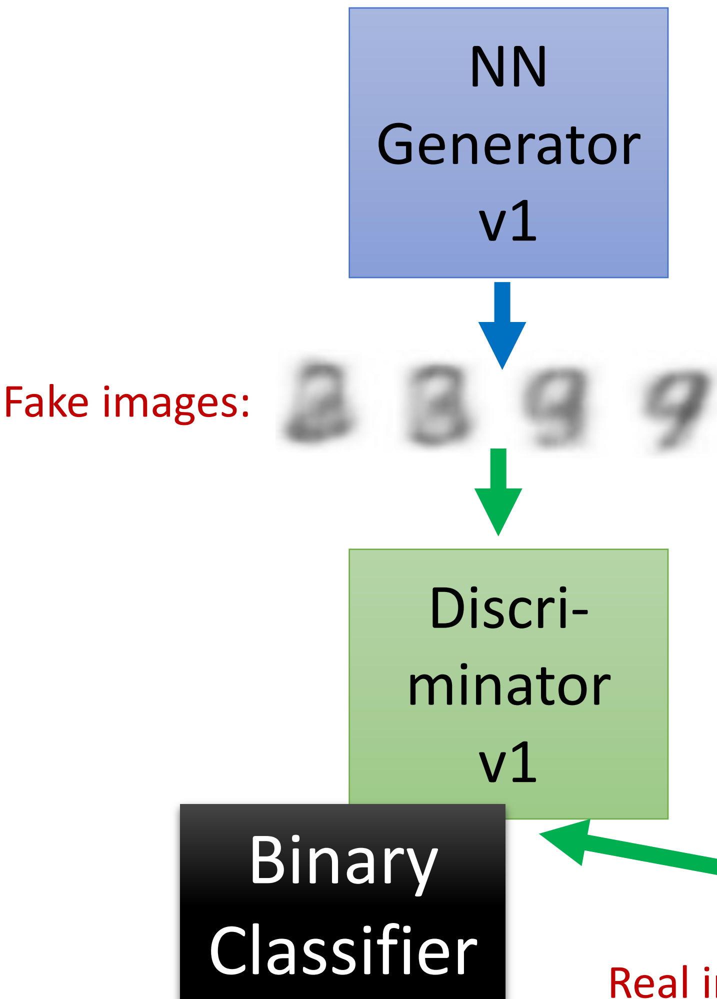
**Player1, Generator $\color{red}{G}$**
**Player2, Discriminator D**




$$V(\color{red}{G}, D) = \mathbb{E}_{x\sim P_{data}}[log D(x)] + \mathbb{E}_{\color{red}{x\sim P_G}}[log(1 - D(x))]$$

$$\color{red}{G^*} = arg \color{red}{\min_G} \max_D V(\color{red}{G}, D)$$

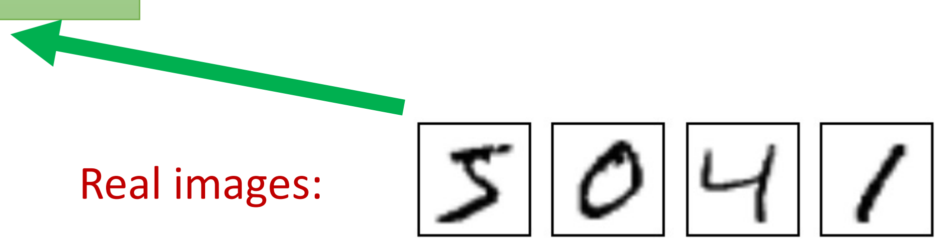
# GAN Learning – D and G updates



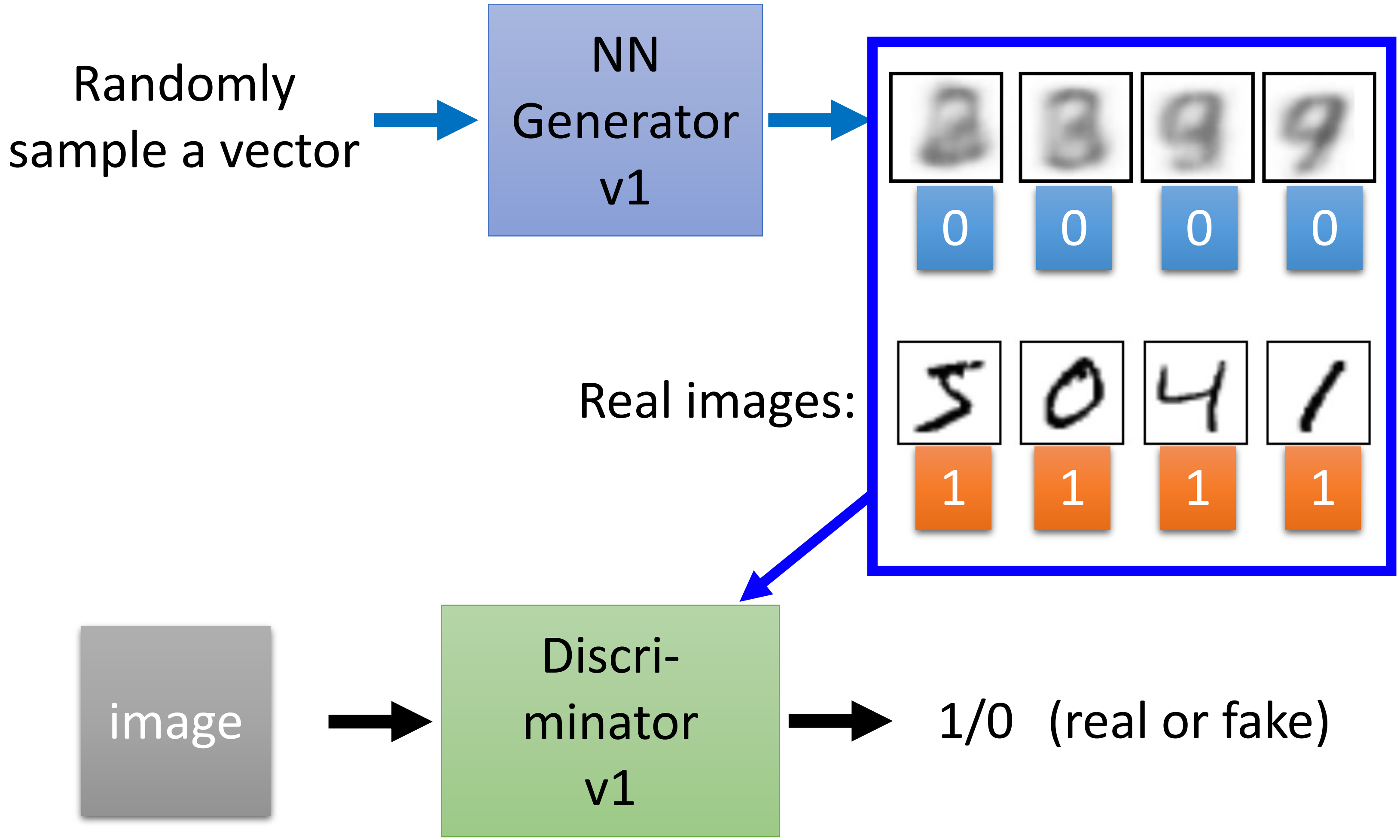

Game scenario:

**Player1, Generator G,** produces samples
**Player2,** – Its adversary **Discriminator D**, attempts to distinguish real samples from fake generated ones (produced by Player1) !

Player1 aims at producing Realistic images to fool the Player2



# GAN - Discriminator




Discriminator Optimization on a batch of images:

Using gradient descent to update the parameters in the discriminator, with a fixed generator

# GAN - Generator

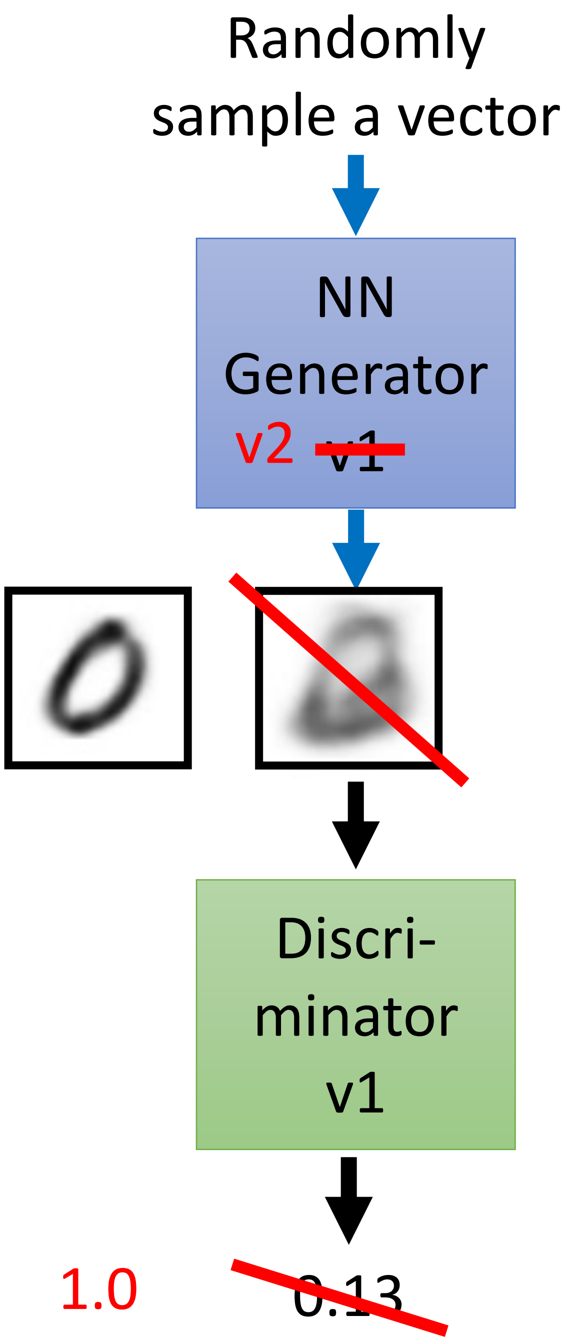
Updating the parameters of generator

The output be classified as "real" (as close to 1 as possible)

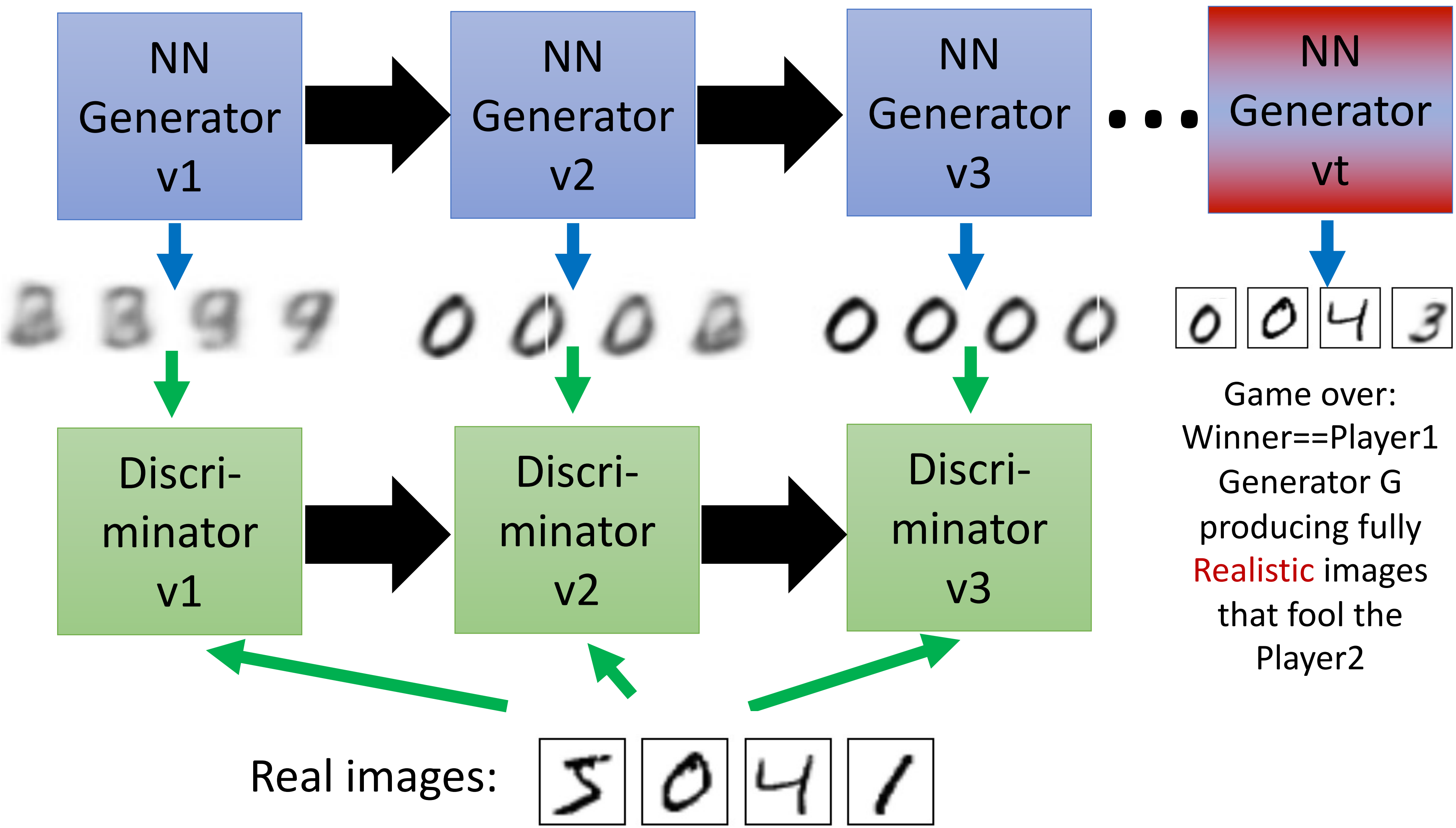
Generator + Discriminator = a network

**Optimization:**

Using gradient descent to update the parameters in the generator, but fixing the discriminator



# GAN Learning – D and G updates



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

---

**for** number of training iterations **do**
  **for** $k$ steps **do**
    • Sample minibatch of $m$ noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
    • Sample minibatch of $m$ examples $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(m)}\}$ from data generating distribution $p_{\text{data}}(\boldsymbol{x})$.
    • Update the discriminator by ascending its stochastic gradient:

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

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

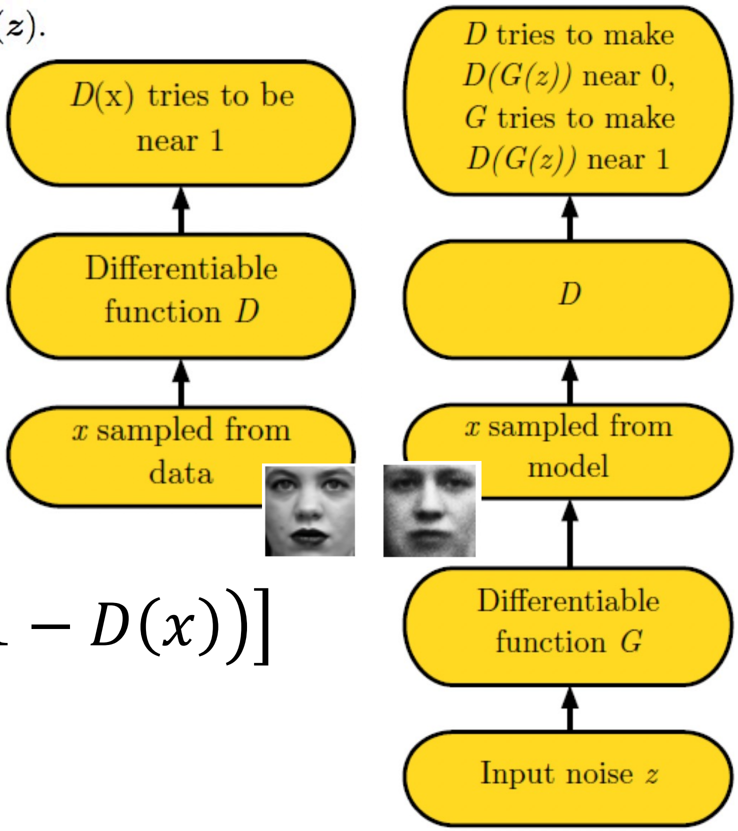
$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right).$$

**end for**

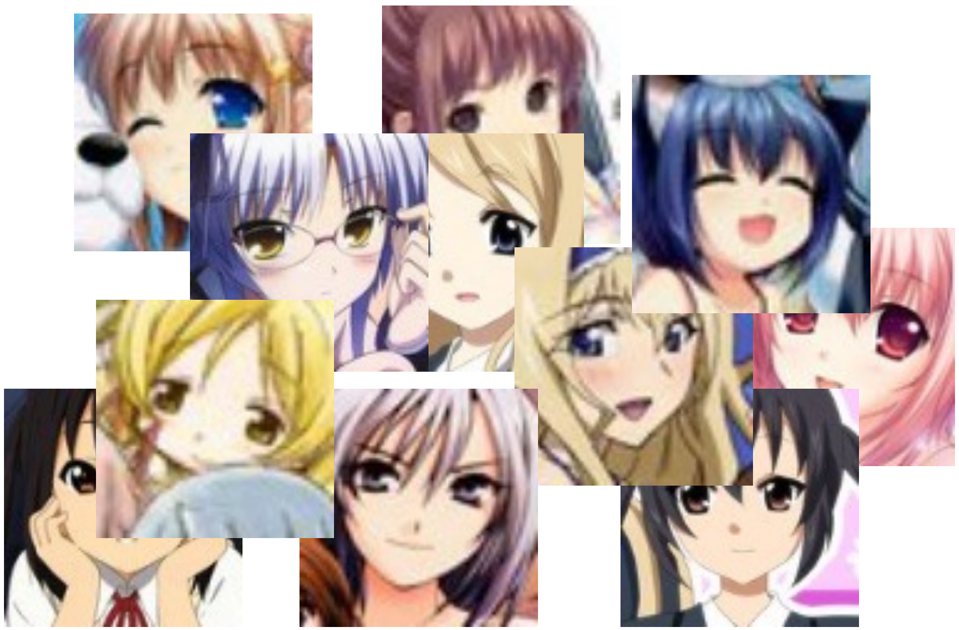
# GAN algorithm

$$V = \mathbb{E}_{x \sim P_{data}}[logD(x)] + \mathbb{E}_{x \sim P_G}[log(1 - D(x))]$$

$$G^* = arg\min_G \max_D V(G, D)$$



# One example GAN

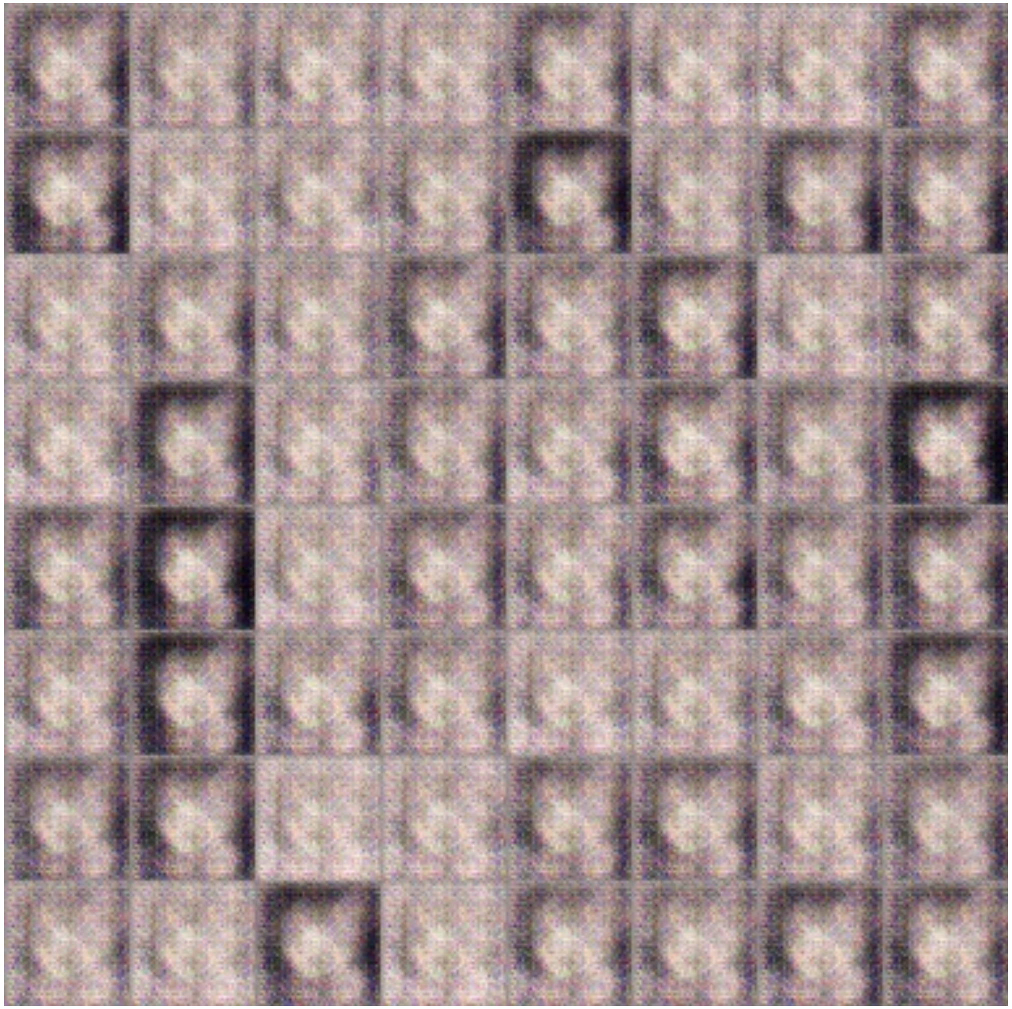


Source of images: https://zhuanlan.zhihu.com/p/24767059

DCGAN: https://github.com/carpedm20/DCGAN-tensorflow

# GAN

100 rounds



# GAN

1000 rounds

# GAN

2000 rounds

# GAN

5000 rounds

# GAN

10,000 rounds

# GAN

20,000 rounds

# GAN

50,000 rounds

# Generative models
# Outline

1. Preview: Auto-Encoders, VAE
2. Generative models with GAN
   - GAN Algorithm
   - **KL vs. Jensen Shanon Divergence**

$$V(G,D) = \mathbb{E}_{x\sim P_{data}}[logD(x)] + \mathbb{E}_{x\sim P_G}\left[log\left(1 - D(x)\right)\right]$$

$$G^* = arg\min_G \max_D V(G,D)$$

## Which measure to evaluate how $P_G(x;\theta)$ is close to $P_{data}(x)$ in Maximum Likelihood optimization?

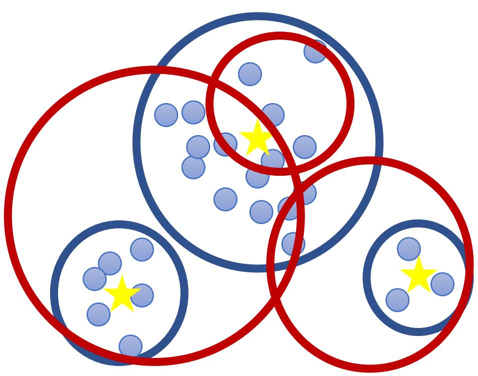
- Given a data distribution $P_{data}(x)$
- We have a distribution $P_G(x;\theta)$ parameterized by $\theta$
  - E.g. $P_G(x;\theta)$ is a Gaussian Mixture Model, $\theta$ are means and variances of the Gaussians
  - We want to find $\theta$ such that $P_G(x;\theta)$ close to $P_{data}(x)$

Sample $\{x^1, x^2, \dots, x^m\}$ from $P_{data}(x)$

We can compute $P_G(x^i;\theta)$

Likelihood of generating the samples

$$L = \prod_{i=1}^{m} P_G(x^i;\theta)$$



Find $\theta^*$ maximizing the likelihood

Which measure to evaluate how $P_G(x;\theta)$ is close to $P_{data}(x)$ in Maximum Likelihood optimization?

$$\theta^* = arg \max_\theta \prod_{i=1}^{m} P_G(x^i;\theta) = arg \max_\theta log \prod_{i=1}^{m} P_G(x^i;\theta)$$

$$= arg \max_\theta \sum_{i=1}^{m} log P_G(x^i;\theta)$$

$\{x^1, x^2, \dots, x^m\}$ from $P_{data}(x)$

$$\approx arg \max_\theta \mathbb{E}_{x\sim P_{data}}[log P_G(x;\theta)]$$

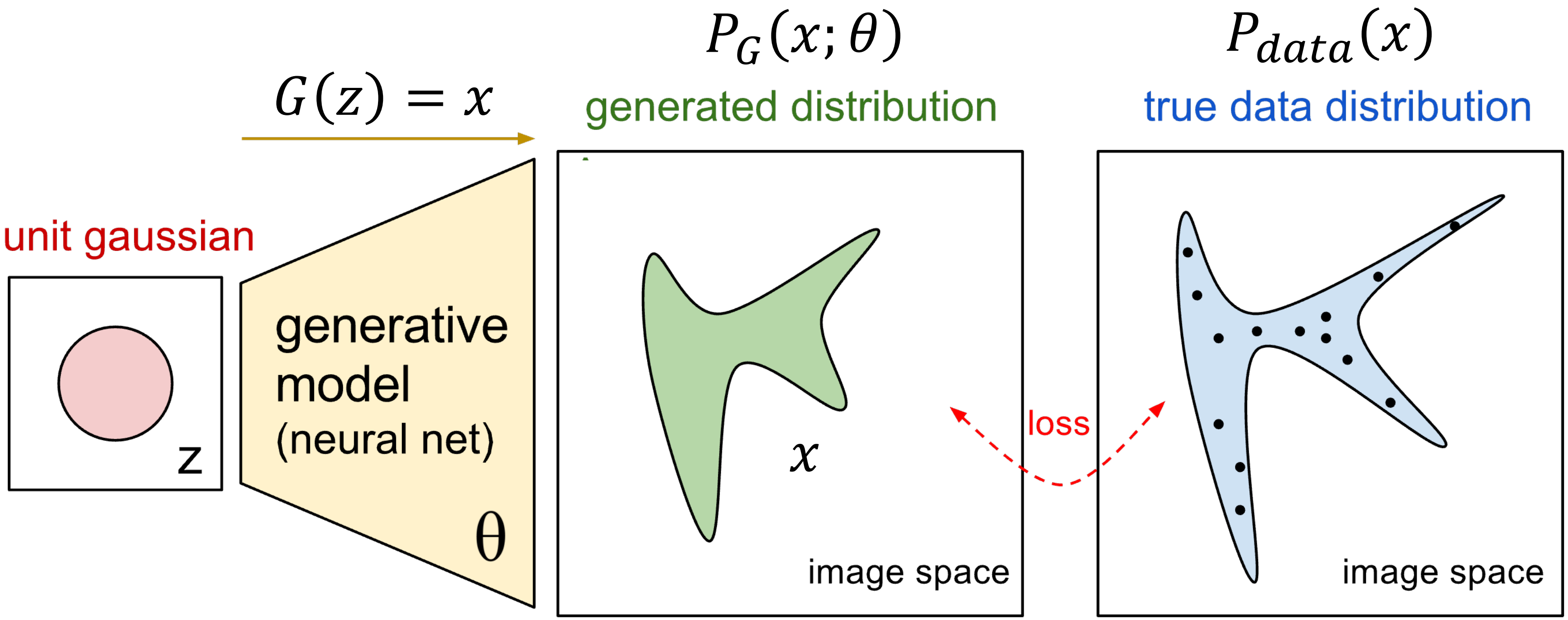
$$= arg \max_\theta \int_x P_{data}(x) log P_G(x;\theta) dx - \int_x P_{data}(x) log P_{data}(x) dx$$

$$= arg \min_\theta KL\big(P_{data}(x)||P_G(x;\theta)\big)$$

$$KL(P||Q) = \int_x P(x) log \frac{P(x)}{Q(x)} dx$$

*In Maximum Likelihood it is a KLD Kullback Leibler Divergence*

# If $P_G(x;\theta)$ is a coming with a NN




$$P_G(x;\theta) = \int_z P_{prior}(z) I_{[G(z)=x]} dz$$

It is difficult to compute the likelihood.

Credits: https://blog.openai.com/generative-models/

# Basic Idea of GAN: the 2 players G-D game

- Generator G

  Hard to learn by maximum likelihood

  - G is a function, input z, output x
  - Given a prior distribution $P_{prior}(z)$, a probability distribution $P_G(x)$ is defined by function G (and $P_{prior}$)
- Discriminator D
  - D is a function, input x, output scalar
  - Evaluate the "difference" between $P_G(x)$ and $P_{data}(x)$
- Global objective function V(G,D)

$$\theta^* = G^* = arg \min_G \max_D V(G, D)$$

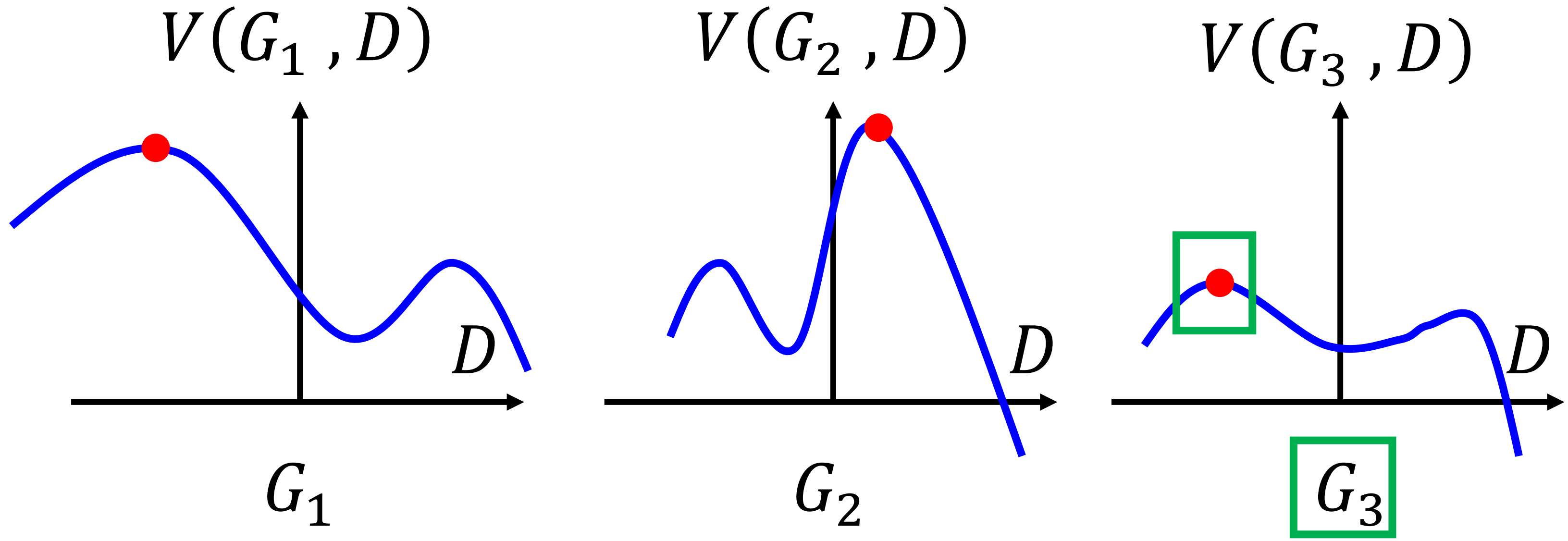
# Basic Idea

$$\boxed{G^*} = arg \min_G \underline{\max_D V(G,D)}$$

$$V = \mathbb{E}_{x \sim P_{data}}[logD(x)] + \mathbb{E}_{x \sim P_G}\left[log\left(1 - D(x)\right)\right]$$

Given a generator G, $\max_D V(G,D)$ evaluate the "difference" between $P_G$ and $P_{data}$

Pick the G defining $P_G$ most similar to $P_{data}$



$$\max_D V(G,D) \qquad G^* = arg \min_G \max_D V(G,D)$$

- Given G, what is the optimal D* maximizing

$$V = \mathbb{E}_{x\sim P_{data}}[logD(x)] + \mathbb{E}_{x\sim P_G}\left[log\left(1-D(x)\right)\right]$$

$$= \int_x P_{data}(x)logD(x)\,dx + \int_x P_G(x)log\left(1-D(x)\right)dx$$

$$= \int_x \left[P_{data}(x)logD(x) + P_G(x)log\left(1-D(x)\right)\right]dx$$

Assume that D(x) can have any value here

- Given x, the optimal D* maximizing

$$P_{data}(x)logD(x) + P_G(x)log\left(1-D(x)\right)$$

$$\max_D V(G,D) \qquad G^* = arg \min_G \max_D V(G,D)$$

- Given x, the optimal D* maximizing

$$\underbrace{P_{data}(x)}_{a} log \underbrace{D(x)}_{D} + \underbrace{P_G(x)}_{b} log\big(1 - \underbrace{D(x)}_{D}\big)$$

- Find D* maximizing: $\mathrm{f}(D) = alog(D) + blog(1-D)$

$$\frac{d\mathrm{f}(D)}{dD} = a \times \frac{1}{D} + b \times \frac{1}{1-D} \times (-1) \quad = 0$$

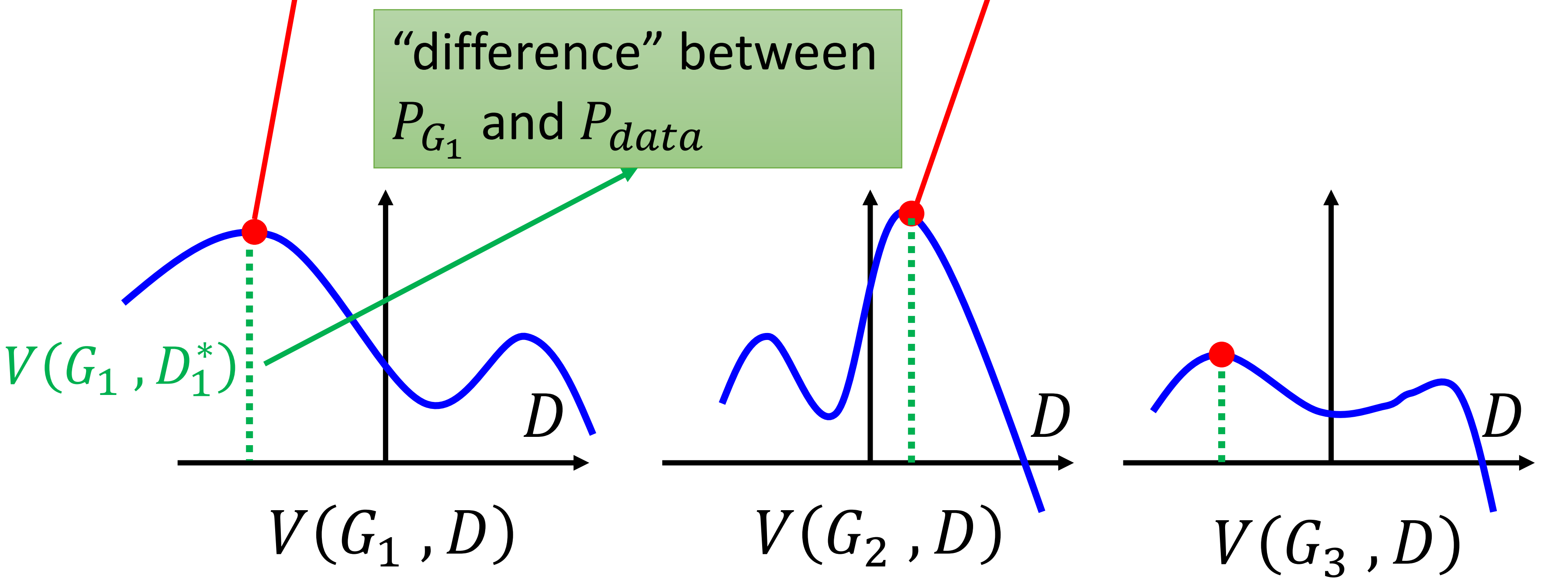
$$a \times \frac{1}{D^*} = b \times \frac{1}{1-D^*} \qquad a \times (1 - D^*) = b \times D^*$$

$$a - aD^* = bD^*$$

$$D^* = \frac{a}{a+b} \Rightarrow \qquad 0 < D^*(x) = \frac{P_{data}(x)}{P_{data}(x) + P_G(x)} < 1$$

$$\max_D V(G,D) \qquad G^* = arg \min_G \max_D V(G,D)$$

$$D_1^*(x) = \frac{P_{data}(x)}{P_{data}(x) + P_{G_1}(x)} \qquad D_2^*(x) = \frac{P_{data}(x)}{P_{data}(x) + P_{G_2}(x)}$$



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

$$\boxed{\begin{aligned} V = {} & \mathbb{E}_{x\sim P_{data}}[logD(x)] \\ & +\mathbb{E}_{x\sim P_G}[log(1-D(x))] \end{aligned}}$$

$$\max_D V(G,D) \; = V(G,D^*) \qquad D^*(x) = \frac{P_{data}(x)}{P_{data}(x)+P_G(x)}$$

$$= \mathbb{E}_{x\sim P_{data}}\left[log\frac{P_{data}(x)}{P_{data}(x)+P_G(x)}\right]$$

$$+\mathbb{E}_{x\sim P_G}\left[log\frac{P_G(x)}{P_{data}(x)+P_G(x)}\right]$$

$$= \int_x P_{data}(x)log\frac{\frac{1}{2}\;P_{data}(x)}{\frac{P_{data}(x)+P_G(x)}{2}}dx$$

$$+\int_x P_G(x)log\frac{\frac{1}{2}\;P_G(x)}{\frac{P_{data}(x)+P_G(x)}{2}}dx$$

$$=> +2log\frac{1}{2} = -2log2$$

$$\mathrm{JSD}(P||Q) = \frac{1}{2}\mathrm{KL}(P||M) + \frac{1}{2}\mathrm{KL}(Q||M)$$

$$\mathrm{M} = \frac{1}{2}(P + Q)$$

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

$$\max_D V(G,D) = V(G,D^*) \qquad D^*(x) = \frac{P_{data}(x)}{P_{data}(x) + P_G(x)}$$

$$= -2log2 + \int_x P_{data}(x) log \frac{P_{data}(x)}{\left(P_{data}(x) + P_G(x)\right)/2} dx$$

$$+ \int_x P_G(x) log \frac{P_G(x)}{\left(P_{data}(x) + P_G(x)\right)/2} dx$$

$$= -2\mathrm{log}2 + \mathrm{KL}\left(\mathrm{P}_{\mathrm{data}}(\mathrm{x})||\frac{\mathrm{P}_{\mathrm{data}}(\mathrm{x}) + \mathrm{P}_{\mathrm{G}}(\mathrm{x})}{2}\right)$$

$$+\mathrm{KL}\left(\mathrm{P}_{\mathrm{G}}(\mathrm{x})||\frac{\mathrm{P}_{\mathrm{data}}(\mathrm{x}) + \mathrm{P}_{\mathrm{G}}(\mathrm{x})}{2}\right)$$

$$= -2log2 + 2JSD\left(P_{data}(x)||P_G(x)\right)$$

Jensen-Shannon divergence

# In the end ……

$$V = \mathbb{E}_{x\sim P_{data}}[logD(x)] + \mathbb{E}_{x\sim P_G}\left[log\left(1 - D(x)\right)\right]$$

- Generator G, Discriminator D
- Looking for G* such that $G^* = arg \min_G \max_D V(G, D)$
- Given G, $\max_D V(G, D) = -2log2 + 2JSD\left(P_{data}(x) || P_G(x)\right)$

  0 < … < log 2
- What is the optimal G?

  $P_G(x) = P_{data}(x)$

  with/using the JS($P_G, P_{data}$) Divergence

  *(In Maximum Likelihood it is a KL Divergence)*