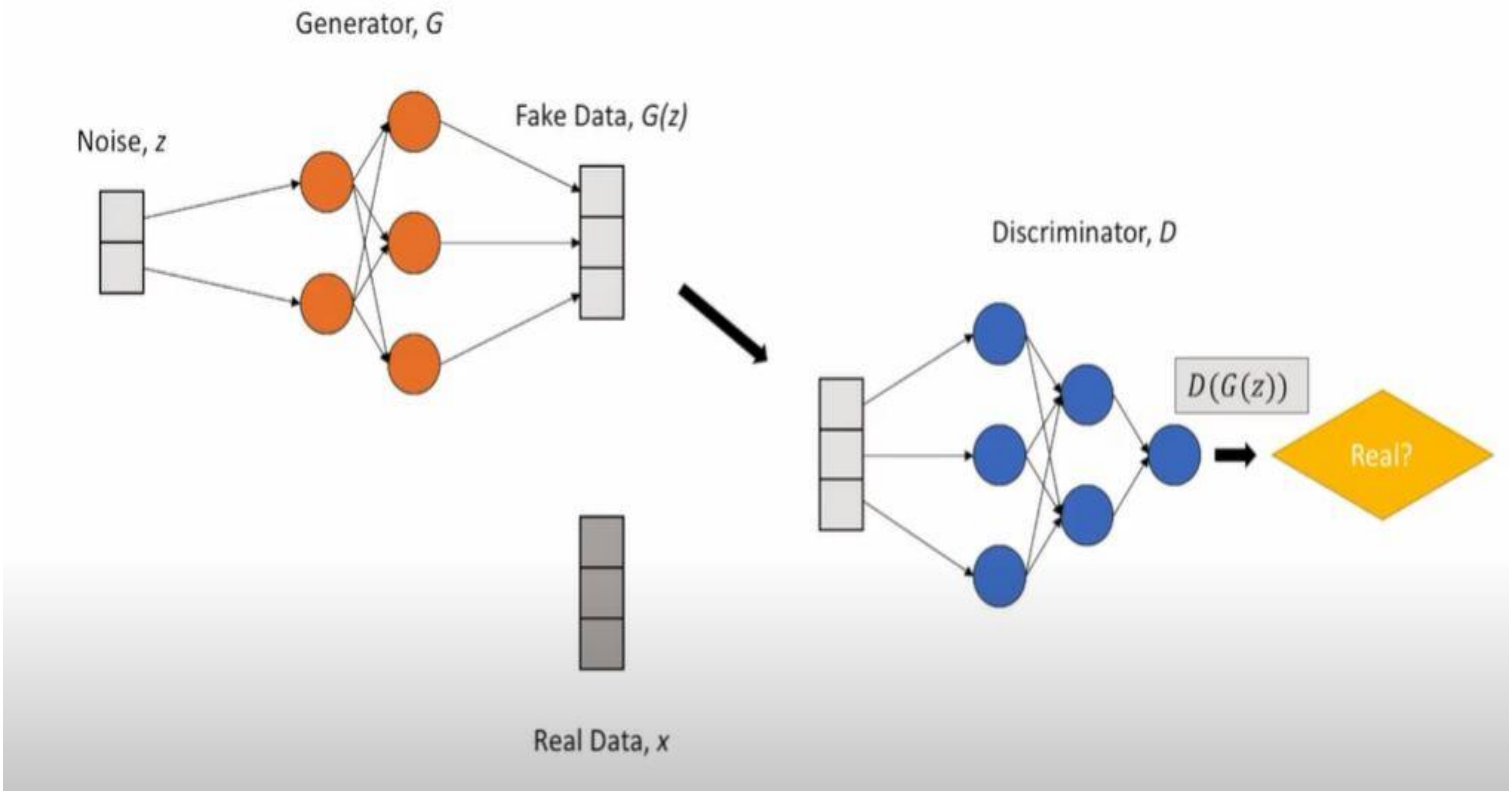
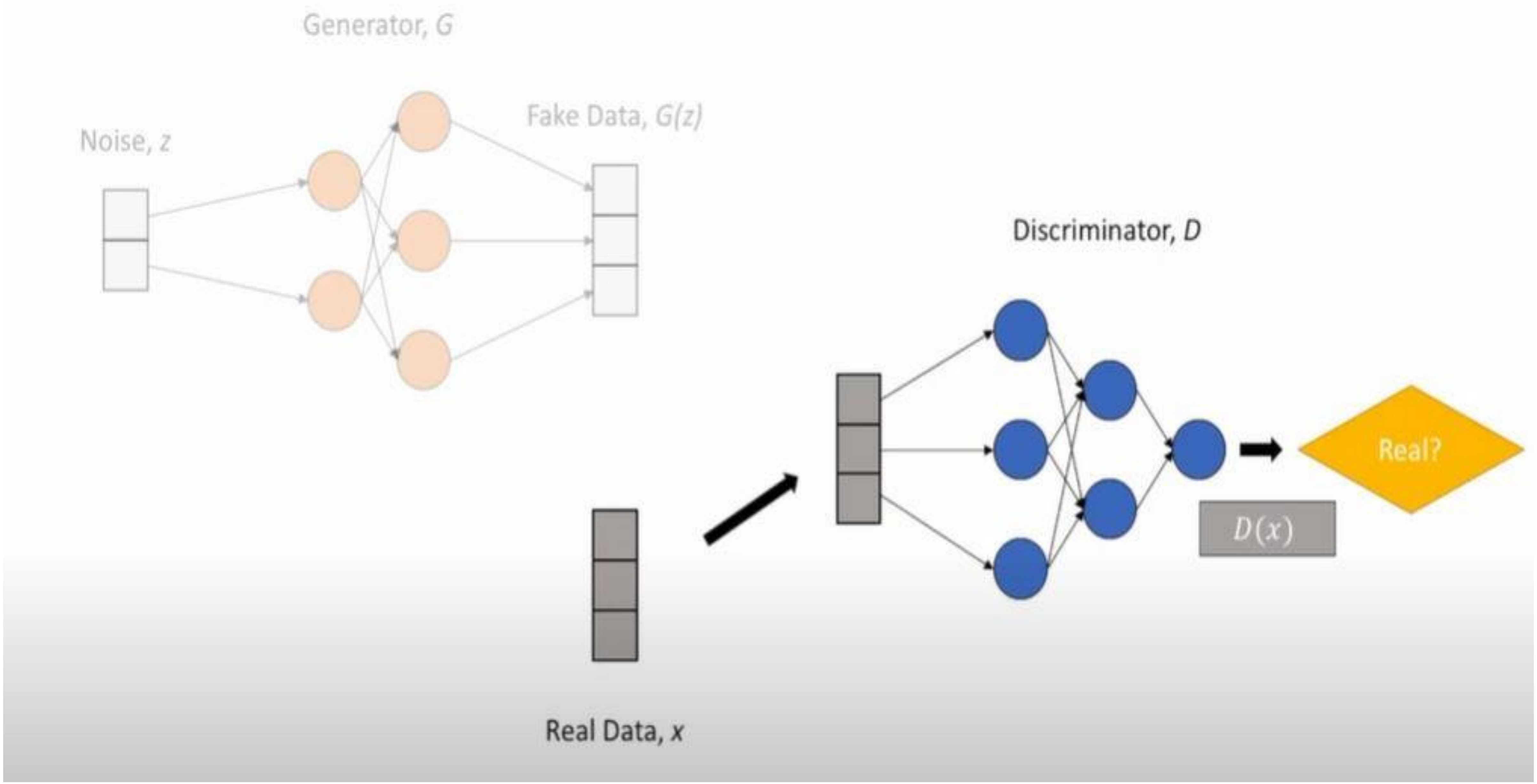
Generative adversarial Networks

A Deep Dive

By: Ahmed Ibrahim


Generator, G
Noise, z
Fake Data, G(z)
Discriminator, D
D(G(z))
Real?
Real Data, x

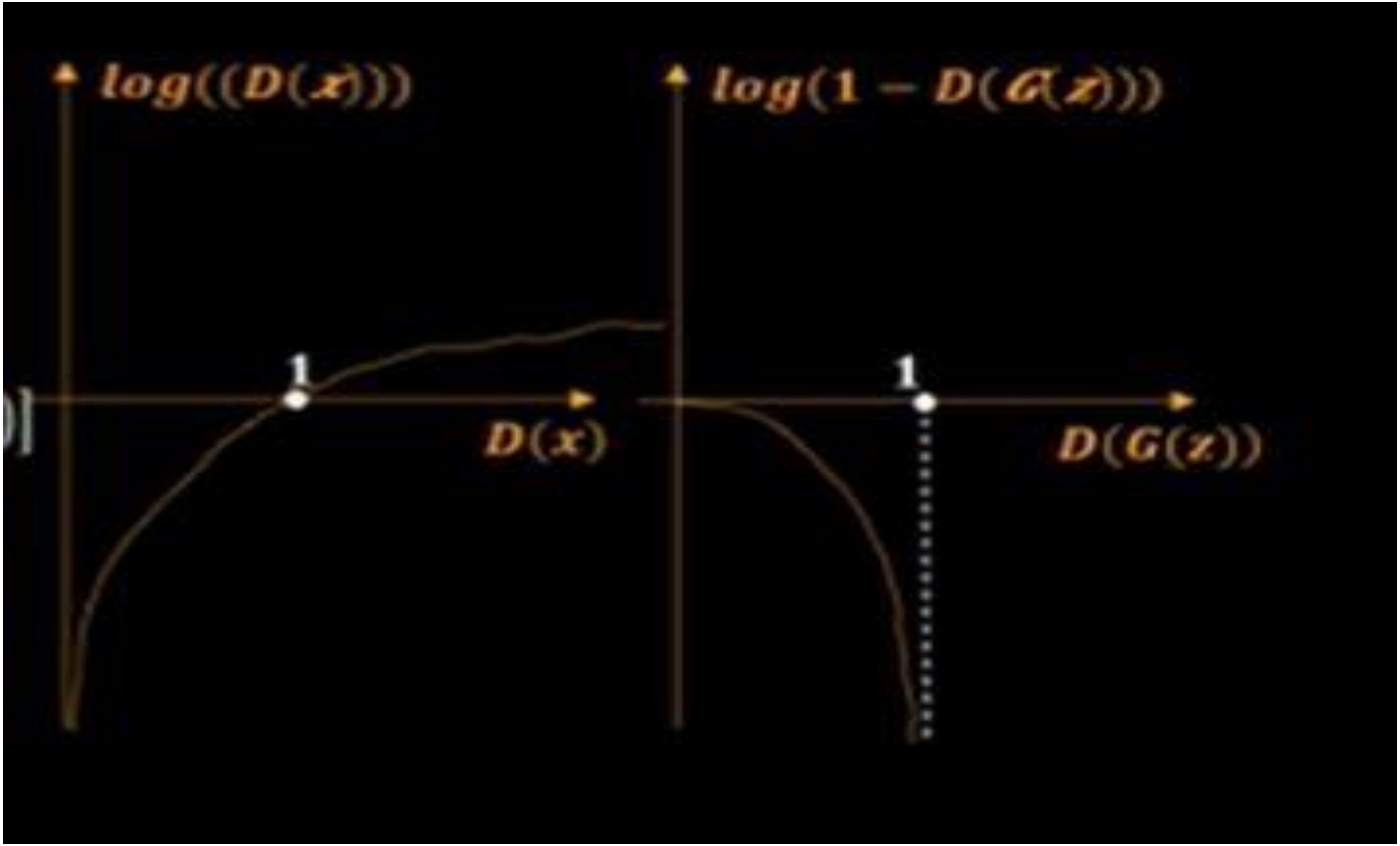


Generator, G
Noise, z
Fake Data, G(z)
Discriminator, D
Real?
D(x)
Real Data, x

Unsupervised $\xrightarrow{\text{Adversarial Framework}}$ Supervised

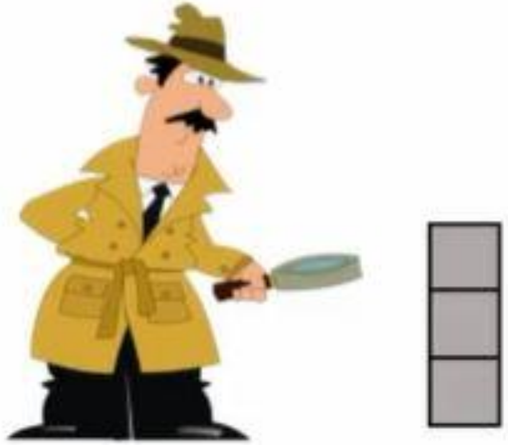
$$V(D,G) = \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z} \sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$$

- Let's say we are training a GAN to generate handwritten digits (0-9) that look like they were drawn from the MNIST dataset of handwritten digits. The generator network takes in a noise vector of size 100 as input and outputs an image of size 28x28, while the discriminator network takes in an image of size 28x28 and outputs a probability value between 0 and 1 indicating whether the image is real or fake.
- During training, the generator and discriminator networks are trained iteratively in an adversarial process. The generator tries to generate realistic images that can fool the discriminator into thinking they are real, while the discriminator tries to correctly classify real and fake images.
- Let's say we have a batch of 32 images (16 real and 16 fake) and we compute the adversarial loss function for that batch. For simplicity, let's assume that the discriminator network outputs a single probability value for each image.
- First, we compute the loss for the real images:
- For each real image, we input it into the discriminator network and compute the probability value that the image is real. Let's say the discriminator outputs a value **of 0.9 for each real image**.
- We then compute the loss for the real images using the term **-log(D(x)).** In this case, the loss for each real image **is -log(0.9) = 0.105.**
- Next, **we compute the loss for the fake images**:
- For each fake image, we input it into the discriminator network and compute the probability value that the image is real. Let's say the discriminator outputs a value of 0.2 for each fake image.
- We then compute the loss for the fake images using the term **-log(1 - D(G(z))).** In this case, the loss for each fake image is **-log(1 - 0.2) = -log(0.8) = 0.223.**
- Finally, we compute the total loss for the batch by taking the average of the losses for the real and fake images:
- Total loss = (sum of losses for real images + sum of losses for fake images) / number of images in batch
- **Total loss = (16 * 0.105 + 16 * 0.223) / 32 = 0.164**
- The goal during training is to minimize this adversarial loss function, which will result in the generator network learning to generate images that are increasingly realistic and the discriminator network becoming better at distinguishing real images from fake ones.



log((D(x)))
1
D(x)
log(1 - D(G(z)))
1
D(G(z))

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

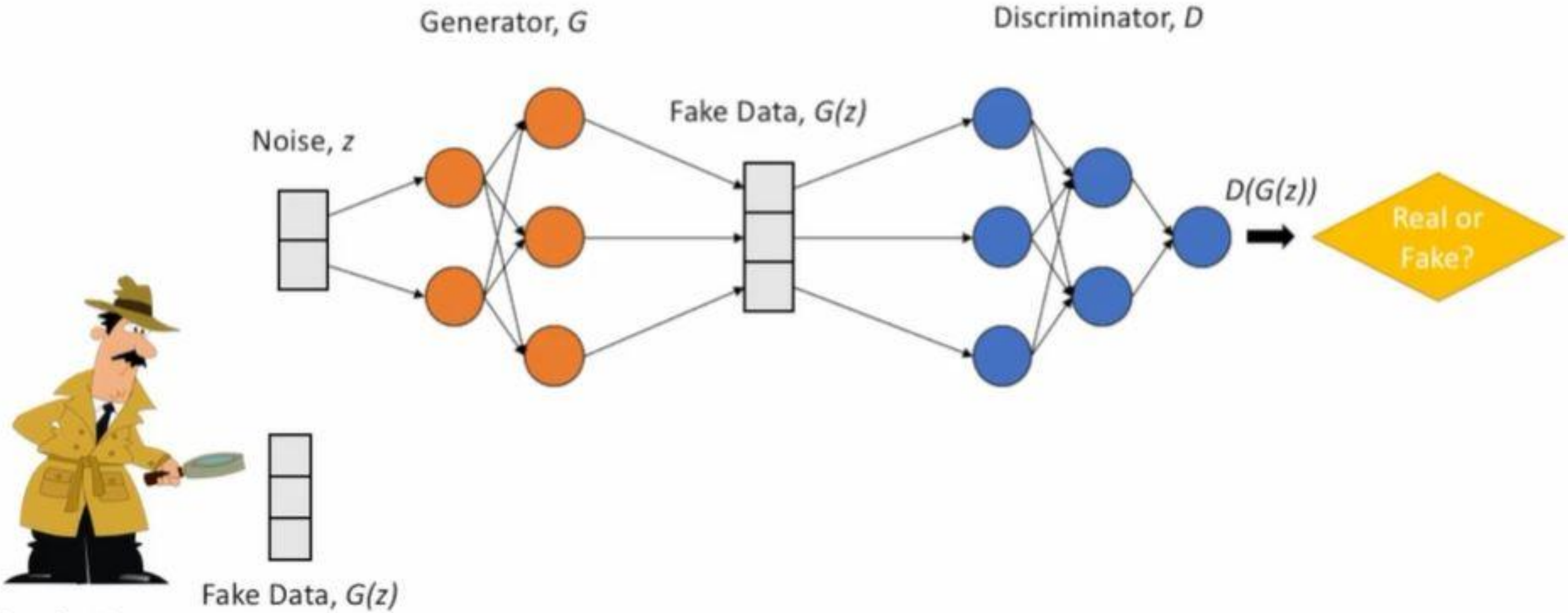


Discriminator Real Data, *x*

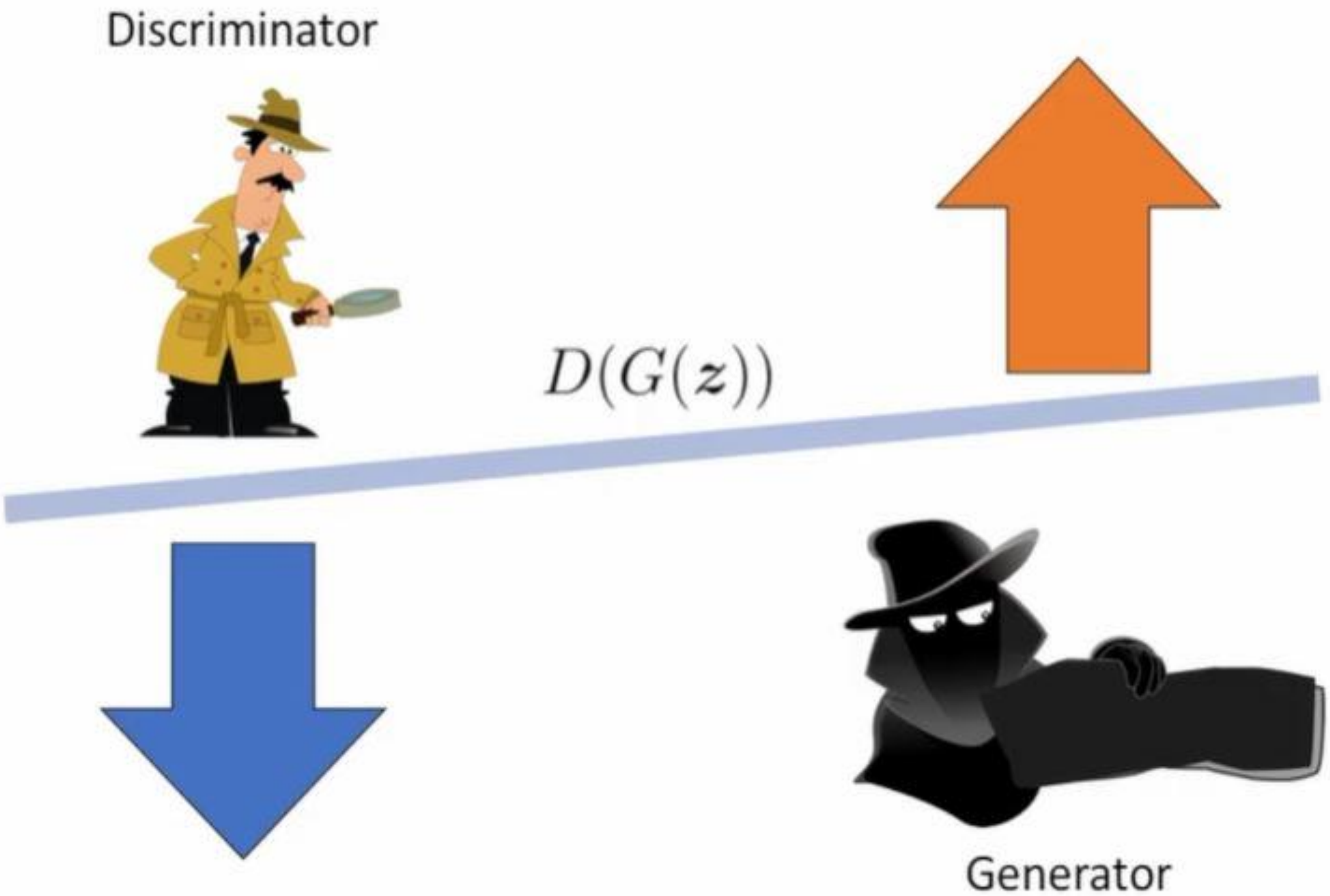
$D(x)$

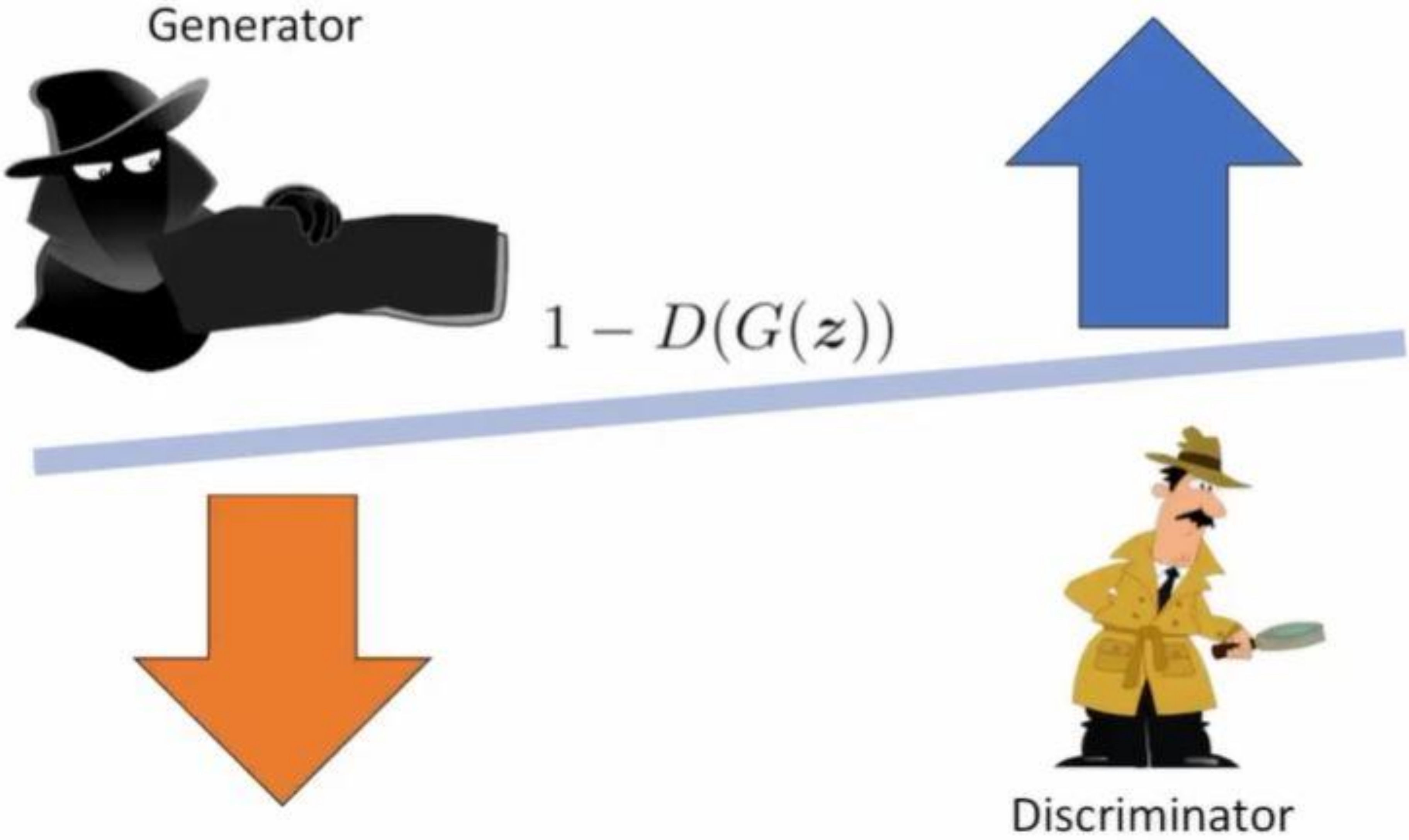
Discriminator

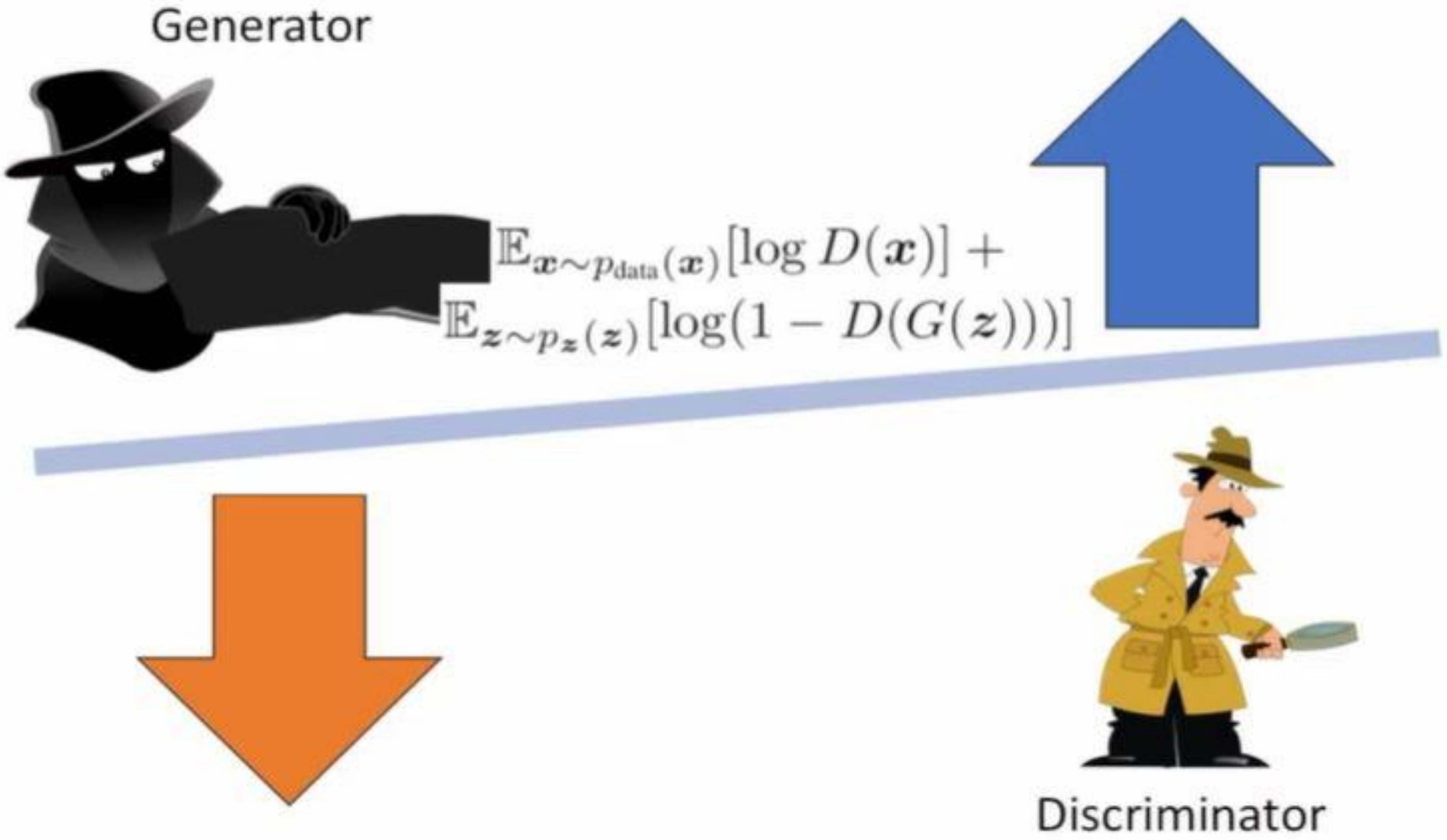
$$V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \boxed{\mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]}$$



Discriminator


Discriminator
$D(G(z))$
Generator


Generator
$1 - D(G(z))$
Discriminator



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

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

- **for** each training iteration **do**
  - **for** k steps **do**
    - Sample m noise samples $\{z_1, \dots, z_m\}$ and transform with Generator
    - Sample m real samples $\{x_1, \dots, x_m\}$ from real data
    - Update the Discriminator by ascending the 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**

- **for** each training iteration **do**
    - **for** k steps **do**
        - Sample m noise samples $\{z_1, \ldots, z_m\}$ and transform with Generator
        - Sample m real samples $\{x_1, \ldots, x_m\}$ from real data
        - Update the Discriminator by ascending the 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 m noise samples $\{z_1, \ldots, z_m\}$ and transform with Generator
    - Update the Generator by descending the gradient:

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

- **for** each training iteration **do**
  - **for** k steps **do**
    - Sample m noise samples $\{z_1, \dots, z_m\}$ and transform with Generator
    - Sample m real samples $\{x_1, \dots, x_m\}$ from real data
    - Update the Discriminator by ascending the 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 m noise samples $\{z_1, \dots, z_m\}$ and transform with Generator
  - Update the Generator by descending the gradient:

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

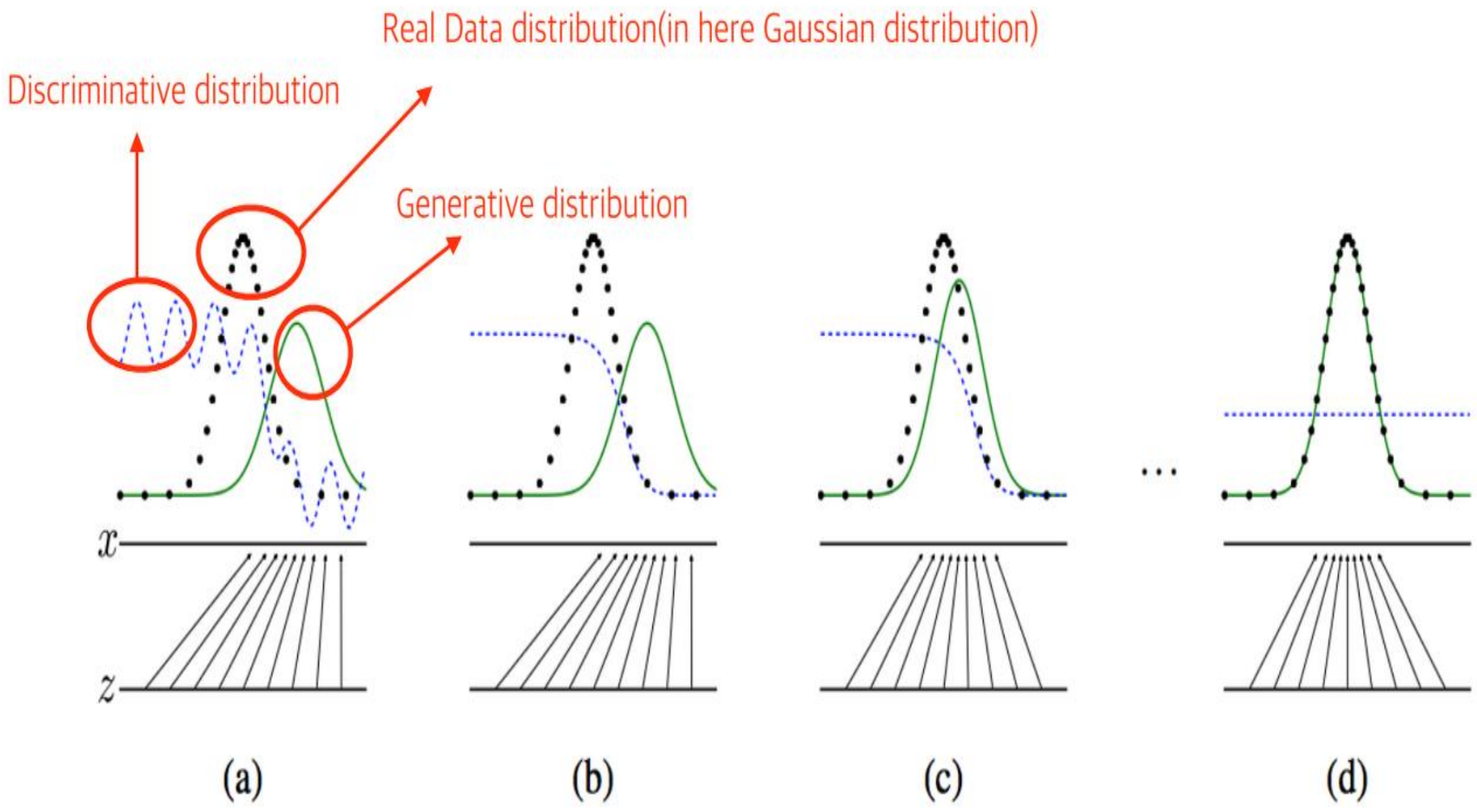
- **for** each training iteration **do**
  - **for** k steps **do**
    - Sample m noise samples $\{z_1, \dots, z_m\}$ and transform with Generator
    - Sample m real samples $\{x_1, \dots, x_m\}$ from real data
    - Update the Discriminator by ascending the gradient:

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

  - end **for**
  - Sample m noise samples $\{z_1, \dots, z_m\}$ and transform with Generator
  - Update the Generator by descending the gradient:

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

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


Real Data distribution(in here Gaussian distribution)
Discriminative distribution
Generative distribution
x
z
...
(a)
(b)
(c)
(d)

$$V(D^*, G) = -\log(4)$$

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

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

$$V(G,D) = \int_{\boldsymbol{x}} p_{\text{data}}(\boldsymbol{x}) \log(D(\boldsymbol{x}))dx + \int_{z} p_{\boldsymbol{z}}(\boldsymbol{z}) \log(1 - D(g(\boldsymbol{z})))dz$$

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

$$V(G,D) = \int_{x} p_{\text{data}}(\boldsymbol{x}) \log(D(\boldsymbol{x}))dx + \int_{z} p_{\boldsymbol{z}}(\boldsymbol{z}) \log(1 - D(g(\boldsymbol{z})))dz$$

$$= \int_{\boldsymbol{x}} p_{\text{data}}(\boldsymbol{x}) \log(D(\boldsymbol{x})) + p_g(\boldsymbol{x}) \log(1 - D(\boldsymbol{x}))dx$$

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

$$V(G,D) = \int_x p_{\text{data}}(\boldsymbol{x}) \log(D(\boldsymbol{x}))dx + \int_z p_{\boldsymbol{z}}(\boldsymbol{z}) \log(1 - D(g(\boldsymbol{z})))dz$$

$$= \int_x \underline{p_{\text{data}}(\boldsymbol{x}) \log(D(\boldsymbol{x})) + p_g(\boldsymbol{x}) \log(1 - D(\boldsymbol{x}))}dx$$

$$a \log(y) + b \log(1-y)$$

$$\frac{\partial}{\partial y} \; a\log(y) + b\log(1-y) = 0$$

$$\frac{a}{y} + \frac{b}{1-y}(-1) = 0 \quad \Rightarrow \quad \frac{a}{y} = \frac{b}{1-y}$$

$$a - ay = by \quad \Rightarrow \quad a = ay + by = (a+b)y$$

$$\Rightarrow \; y = \frac{a}{a+b}$$

$$\therefore \; D(x)^* = \frac{p_{data}(x)}{p_{data}(x) + p_g(x)}$$

$$D(x)^{*} = \frac{p_{data}(x)}{p_{data}(x)+p_{g}(x)}$$

for $p_{data}(x) = p_{g}(x)$

$$D(x)^{*} = \frac{p_{data}(x)}{2p_{data}(x)} = \frac{1}{2}$$

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

$$= \mathbb{E}_{x\sim p_{data}(x)}\left[\log\left(\frac{1}{2}\right)\right] + \mathbb{E}_{z\sim p_z(z)}\left[\log\left(\frac{1}{2}\right)\right]$$

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

$$= \mathbb{E}_{x \sim p_{data}(x)}\left[\log\left(\frac{1}{2}\right)\right] + \mathbb{E}_{z \sim p_z(z)}\left[\log\left(\frac{1}{2}\right)\right]$$

$$= -\log(2) - \log(2)$$

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

$$= \mathbb{E}_{x\sim p_{data}(x)}\left[\log\left(\frac{1}{2}\right)\right] + \mathbb{E}_{z\sim p_z(z)}\left[\log\left(\frac{1}{2}\right)\right]$$

$$= -\log(2) - \log(2)$$

$$= -\log(4)$$

$$D(x)^* = \frac{p_{data}(x)}{p_{data}(x)+p_g(x)}$$

for $p_{data}(x) = p_g(x)$

$$D(x)^* = \frac{1}{2}$$

$$V(D^*, G) = -\log(4)$$

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

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

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

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

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

$$\begin{aligned} D_{KL}(P||Q) &= \mathbb{E}_{x\sim P}\left[\log \frac{P(x)}{Q(x)}\right] \\ &= \mathbb{E}_{x\sim P}\left[\log \frac{2P(x)}{2Q(x)}\right] \\ &= \mathbb{E}_{x\sim P}\left[\log \frac{P(x)}{Q(x)/_2}\right] - \log(2) \end{aligned}$$

$$V(D,G) = -\log(4) + KL\left(p_{\text{data}} \left\| \frac{p_{\text{data}} + p_g}{2}\right.\right) + KL\left(p_g \left\| \frac{p_{\text{data}} + p_g}{2}\right.\right)$$

$$JSD(P||Q) = \frac{1}{2}D_{KL}\left(P||\frac{P+Q}{2}\right) + \frac{1}{2}D_{KL}\left(Q||\frac{P+Q}{2}\right)$$

$$\boxed{V(D,G) = -\log(4) + 2 \cdot JSD\left(p_{\text{data}} \| p_g\right)}$$

$$\min_{G} V(D,G) = -\log(4) + 2 \cdot JSD\left(p_{\text{data}} \| p_g\right)$$

$$\min_G V(D,G) = -\log(4) + 2 \cdot \cancelto{0}{JSD\left(p_{\text{data}} \| p_g\right)}$$

$$\Rightarrow p_{data}(x) = p_g(x)$$

$$\min_G V(D,G) = -\log(4)$$