EECE 571F: Deep Learning with Structures

Lecture 8 I: Generative Adversarial Networks

Renjie Liao

University of British Columbia

Winter, Term 1, 2023

Outline

- Generative Adversarial Networks (GANs)
  - Zero-sum game & min-max loss
  - Optimal discriminator
  - Global minimum
  - Architectures
  - Results & Challenges
- Variants
  - Wasserstein GANs
  - Progressive GANs
  - Cycle GANs
  - MolGANs

Outline

- Generative Adversarial Networks (GANs)
  - **Zero-sum game & min-max loss**
  - Optimal discriminator
  - Global minimum
  - Architectures
  - Results & Challenges
- Variants
  - Wasserstein GANs
  - Progressive GANs
  - Cycle GANs
  - MolGANs

Generative Adversarial Networks (GANs)

In a GAN [1], two neural networks (players) play a *zero-sum game*, i.e., one player's gain is equivalent to the other's loss.

Generative Adversarial Networks (GANs)

In a GAN [1], two neural networks (players) play a *zero-sum game*, i.e., one player's gain is equivalent to the other's loss.

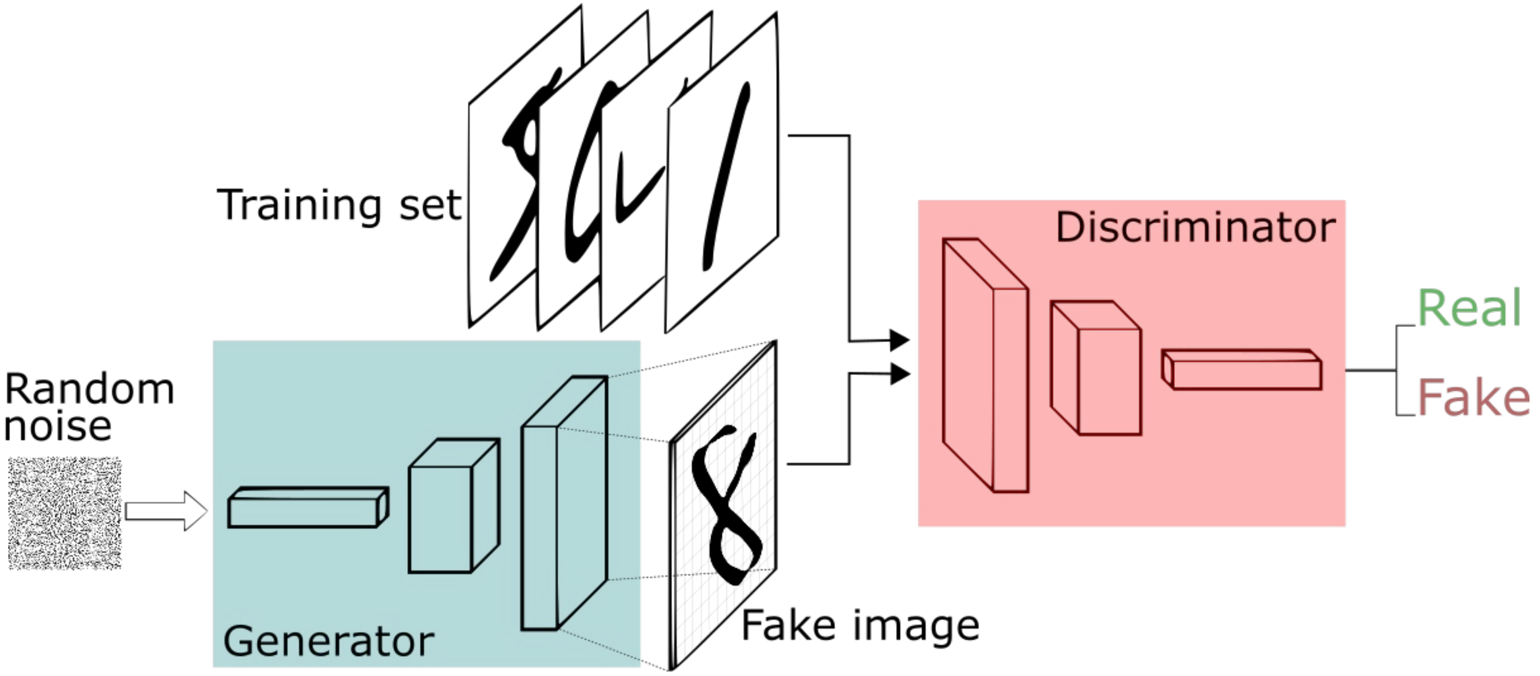
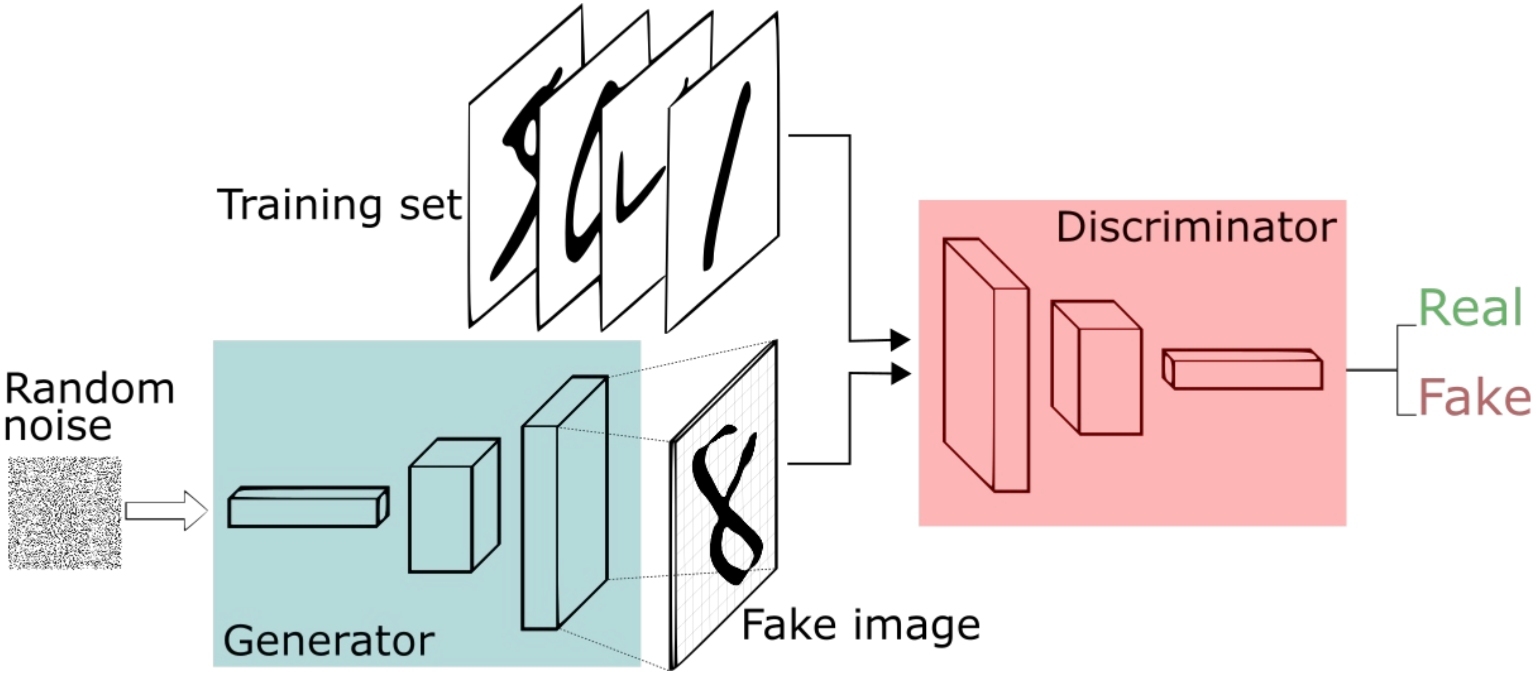


Image Credit: [2]

Generative Adversarial Networks (GANs)

In a GAN [1], two neural networks (players) play a *zero-sum game*, i.e., one player's gain is equivalent to the other's loss.



Min-max loss:

$$\min_{\theta} \max_{\phi} \quad \mathbb{E}_{X \sim p_{\text{data}}(X)}[\log D_\phi(X)] + \mathbb{E}_{Z \sim p(Z)}[\log(1 - D_\phi(G_\theta(Z)))]$$

Image Credit: [2]

Outline

- Generative Adversarial Networks (GANs)
    - Zero-sum game & min-max loss
    - **Optimal discriminator**
    - Global minimum
    - Architectures
    - Results & Challenges
- Variants
    - Wasserstein GANs
    - Progressive GANs
    - Cycle GANs
    - MolGANs

Optimal Discriminator of GANs

1. Fix generator, the optimal discriminator is $$D^*_\phi(X) = \frac{p_{\text{data}}(X)}{p_{\text{data}}(X) + p_{G_\theta}(X)}$$

Optimal Discriminator of GANs

1. Fix generator, the optimal discriminator is

$$D_\phi^*(X) = \frac{p_{\mathrm{data}}(X)}{p_{\mathrm{data}}(X) + p_{G_\theta}(X)}$$

Why?

$$\begin{aligned}
\ell(G_\theta, D_\phi) &= \mathbb{E}_{X \sim p_{\mathrm{data}}(X)}[\log D_\phi(X)] + \mathbb{E}_{Z \sim p(Z)}[\log(1 - D_\phi(G_\theta(Z))] \\
&= \mathbb{E}_{X \sim p_{\mathrm{data}}(X)}[\log D_\phi(X)] + \mathbb{E}_{X \sim p_{G_\theta}(X)}[\log(1 - D_\phi(X))] \\
&= \int p_{\mathrm{data}}(X) \log D_\phi(X) + p_{G_\theta}(X) \log(1 - D_\phi(X)) dX
\end{aligned}$$

Optimal Discriminator of GANs

1. Fix generator, the optimal discriminator is

$$D^*_\phi(X) = \frac{p_{\text{data}}(X)}{p_{\text{data}}(X) + p_{G_\theta}(X)}$$

Why?

$$\begin{aligned}\ell(G_\theta, D_\phi) &= \mathbb{E}_{X\sim p_{\text{data}}(X)}[\log D_\phi(X)] + \boxed{\mathbb{E}_{Z\sim p(Z)}[\log(1 - D_\phi(G_\theta(Z))]} \\ &= \mathbb{E}_{X\sim p_{\text{data}}(X)}[\log D_\phi(X)] + \boxed{\mathbb{E}_{X\sim p_{G_\theta}(X)}[\log(1 - D_\phi(X))]} \\ &= \int p_{\text{data}}(X)\log D_\phi(X) + p_{G_\theta}(X)\log(1 - D_\phi(X))dX\end{aligned}$$

Law Of The Unconscious Statistician (LOTUS)

Optimal Discriminator of GANs

1. Fix generator, the optimal discriminator is

$$D_\phi^*(X) = \frac{p_{\text{data}}(X)}{p_{\text{data}}(X) + p_{G_\theta}(X)}$$

Why?

$$\begin{aligned}\ell(G_\theta, D_\phi) &= \mathbb{E}_{X \sim p_{\text{data}}(X)}[\log D_\phi(X)] + \boxed{\mathbb{E}_{Z \sim p(Z)}[\log(1 - D_\phi(G_\theta(Z)))]} \\ &= \mathbb{E}_{X \sim p_{\text{data}}(X)}[\log D_\phi(X)] + \boxed{\mathbb{E}_{X \sim p_{G_\theta}(X)}[\log(1 - D_\phi(X))]} \\ &= \int p_{\text{data}}(X) \log D_\phi(X) + p_{G_\theta}(X) \log(1 - D_\phi(X)) dX\end{aligned}$$

Law Of The Unconscious Statistician (LOTUS)

Set the gradient of loss w.r.t. D to be zero, we obtain the optimal discriminator

Outline

- Generative Adversarial Networks (GANs)
    - Zero-sum game & min-max loss
    - Optimal discriminator
    - **Global minimum**
    - Architectures
    - Results & Challenges
- Variants
    - Wasserstein GANs
    - Progressive GANs
    - Cycle GANs
    - MolGANs

Global Minimum of Min-Max Loss

Suppose we found the optimal discriminator, our loss function becomes

$$
\begin{aligned}
C(G_\theta) &= \max_{D_\phi} \quad \ell(G_\theta, D_\phi) \\
&= \mathbb{E}_{X \sim p_{\text{data}}(X)} \left[\log D_\phi^*(X)\right] + \mathbb{E}_{X \sim p_{G_\theta}(X)} \left[\log(1 - D_\phi^*(X))\right] \\
&= \mathbb{E}_{X \sim p_{\text{data}}(X)} \left[\log \left(\frac{p_{\text{data}}(X)}{p_{\text{data}}(X) + p_{G_\theta}(X)}\right)\right] + \mathbb{E}_{X \sim p_{G_\theta}(X)} \left[\log \left(\frac{p_{G_\theta}(X)}{p_{\text{data}}(X) + p_{G_\theta}(X)}\right)\right]
\end{aligned}
$$

Global Minimum of Min-Max Loss

Suppose we found the optimal discriminator, our loss function becomes

$$
\begin{aligned}
C(G_\theta) &= \max_{D_\phi} \quad \ell(G_\theta, D_\phi) \\
&= \mathbb{E}_{X \sim p_{\text{data}}(X)} \left[\log D_\phi^*(X)\right] + \mathbb{E}_{X \sim p_{G_\theta}(X)} \left[\log(1 - D_\phi^*(X))\right] \\
&= \mathbb{E}_{X \sim p_{\text{data}}(X)} \left[\log \left(\frac{p_{\text{data}}(X)}{p_{\text{data}}(X) + p_{G_\theta}(X)}\right)\right] + \mathbb{E}_{X \sim p_{G_\theta}(X)} \left[\log \left(\frac{p_{G_\theta}(X)}{p_{\text{data}}(X) + p_{G_\theta}(X)}\right)\right]
\end{aligned}
$$

2. The global minimum of $C(G_\theta)$ is achieved iff. $p_{\text{data}}(X) = p_{G_\theta}(X)$

Global Minimum of Min-Max Loss

Suppose we found the optimal discriminator, our loss function becomes

$$
\begin{aligned}
C(G_\theta) &= \max_{D_\phi} \quad \ell(G_\theta, D_\phi) \\
&= \mathbb{E}_{X \sim p_{\text{data}}(X)} \left[\log D^*_\phi(X) \right] + \mathbb{E}_{X \sim p_{G_\theta}(X)} \left[\log(1 - D^*_\phi(X)) \right] \\
&= \mathbb{E}_{X \sim p_{\text{data}}(X)} \left[\log \left(\frac{p_{\text{data}}(X)}{p_{\text{data}}(X) + p_{G_\theta}(X)} \right) \right] + \mathbb{E}_{X \sim p_{G_\theta}(X)} \left[\log \left(\frac{p_{G_\theta}(X)}{p_{\text{data}}(X) + p_{G_\theta}(X)} \right) \right]
\end{aligned}
$$

2. The global minimum of $C(G_\theta)$ is achieved iff. $p_{\text{data}}(X) = p_{G_\theta}(X)$

Why?

$$
\begin{aligned}
C(G_\theta) &= \mathbb{E}_{X \sim p_{\text{data}}(X)} \left[\log \left(\frac{p_{\text{data}}(X)}{(p_{\text{data}}(X) + p_{G_\theta}(X))/2} \right) \right] \\
&\quad + \mathbb{E}_{X \sim p_{G_\theta}(X)} \left[\log \left(\frac{p_{G_\theta}(X)}{(p_{\text{data}}(X) + p_{G_\theta}(X))/2} \right) \right] + 2 \log(\frac{1}{2}) \\
&= \text{JSD}(p_{\text{data}}(X) \| p_{G_\theta}(X)) - \log(4)
\end{aligned}
$$

Global Minimum of Min-Max Loss

2. The global minimum of $C(G_\theta)$ is achieved iff. $p_{\text{data}}(X) = p_{G_\theta}(X)$

Why?
$$
\begin{aligned}
C(G_\theta) &= \mathbb{E}_{X\sim p_{\text{data}}(X)}\left[\log\left(\frac{p_{\text{data}}(X)}{(p_{\text{data}}(X)+p_{G_\theta}(X))/2}\right)\right] \\
&+ \mathbb{E}_{X\sim p_{G_\theta}(X)}\left[\log\left(\frac{p_{G_\theta}(X)}{(p_{\text{data}}(X)+p_{G_\theta}(X))/2}\right)\right] + 2\log(\frac{1}{2}) \\
&= \text{JSD}(p_{\text{data}}(X)\|p_{G_\theta}(X)) - \log(4)
\end{aligned}
$$

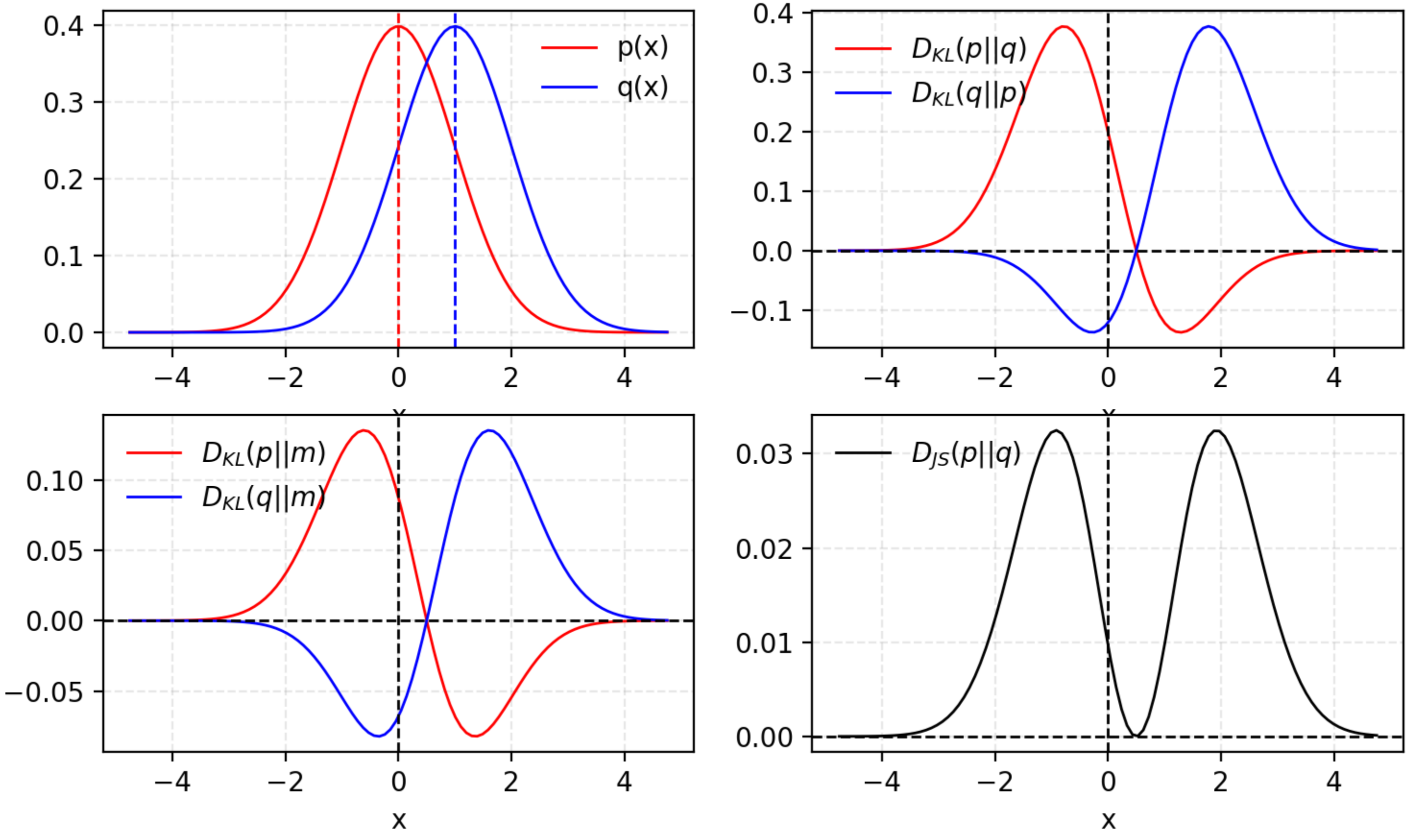
Jensen–Shannon divergence (JSD) is in $[0, \log_b 2]$ (base b) and is zero iff. P = Q

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

Global Minimum of Min-Max Loss

Jensen–Shannon divergence (JSD) is in $[0, \log_b 2]$ (base b) and is zero iff. P = Q

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



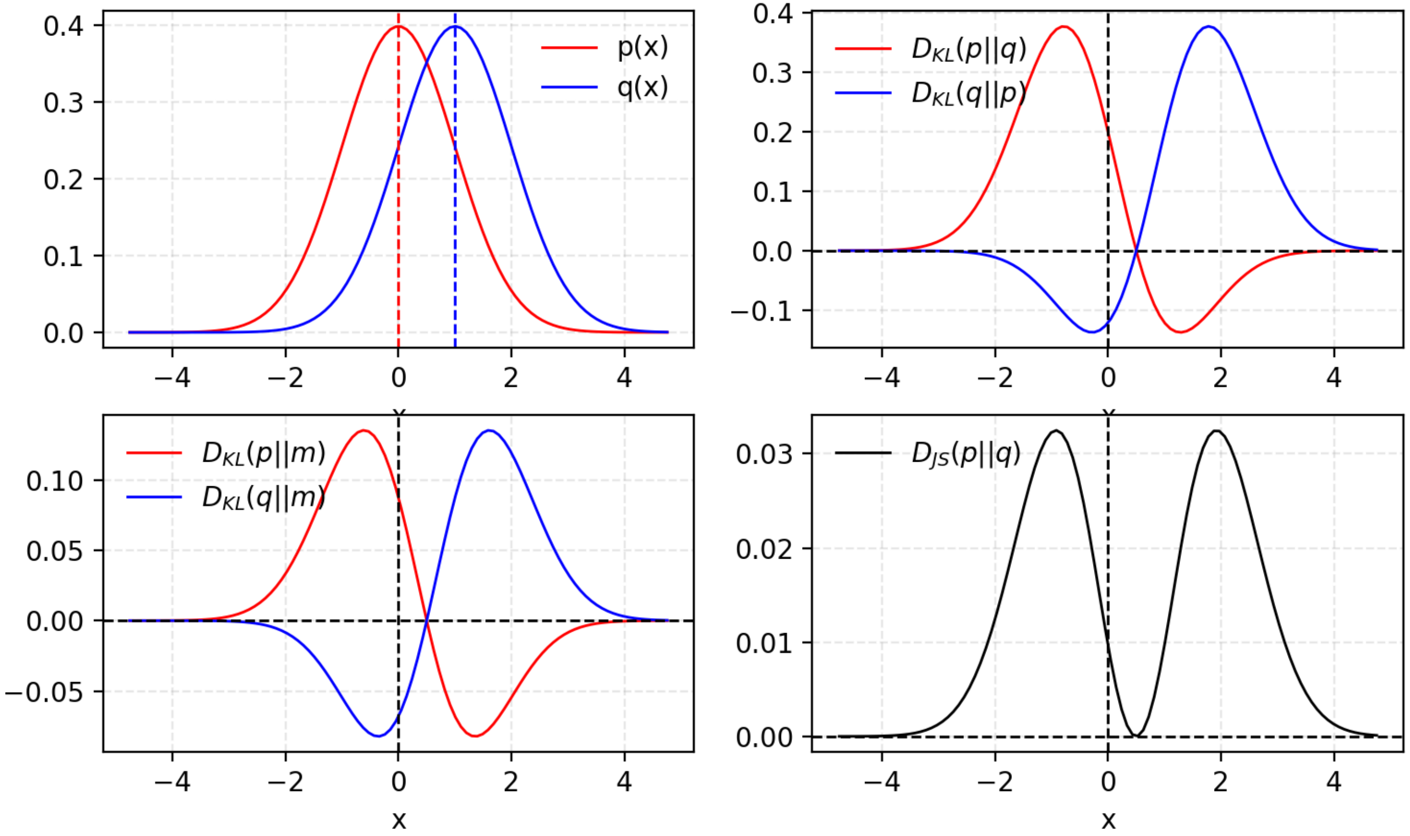
Given two Gaussian distribution, p with mean=0 and std=1 and q with mean=1 and std=1. The average of two distributions is labelled as m. KL divergence is asymmetric but JS divergence is symmetric.

Image Credit: [3]

Global Minimum of Min-Max Loss

Jensen–Shannon divergence (JSD) is in $[0, \log_b 2]$ (base b) and is zero iff. P = Q

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



Given two Gaussian distribution, p with mean=0 and std=1 and q with mean=1 and std=1. The average of two distributions is labelled as m. KL divergence is asymmetric but JS divergence is symmetric.

If high density areas of data and model (generator) distributions have less overlap, JSD is not a good objective!

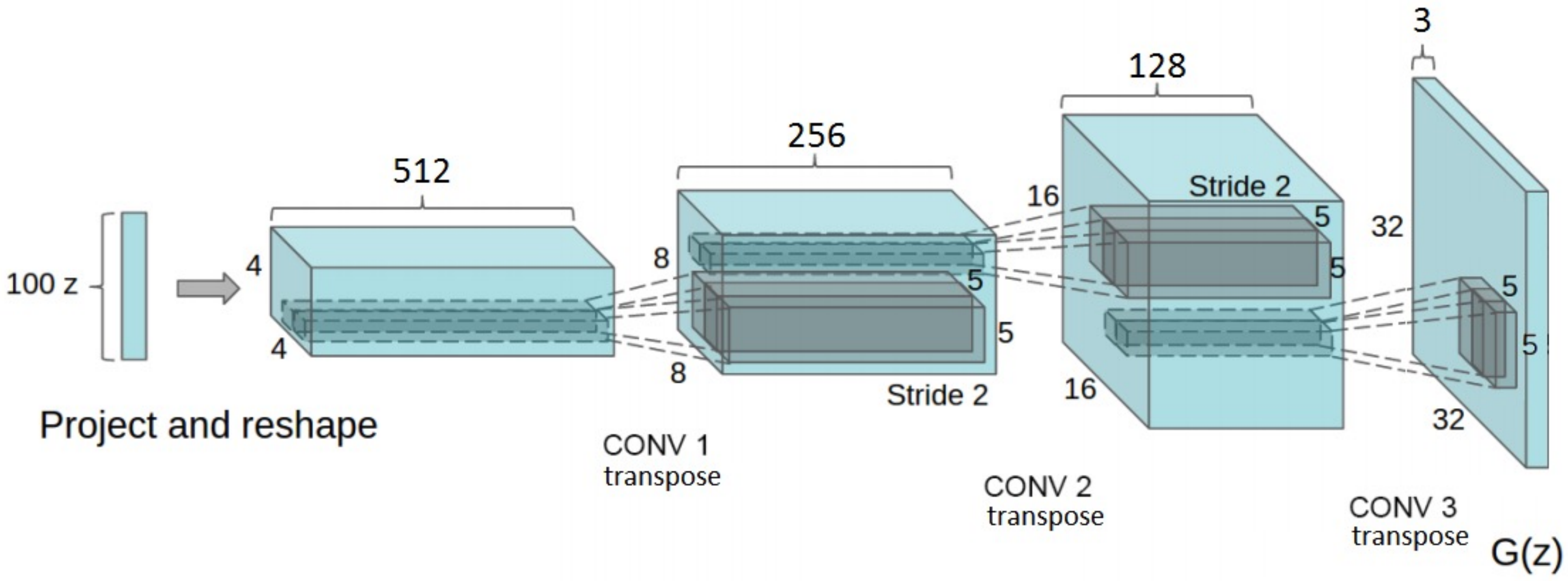
Image Credit: [3]

Outline

- Generative Adversarial Networks (GANs)
  - Zero-sum game & min-max loss
  - Optimal discriminator
  - Global minimum
  - **Architectures**
  - Results & Challenges
- Variants
  - Wasserstein GANs
  - Progressive GANs
  - Cycle GANs
  - MolGANs

Architectures

Deep Convolutional Generative Adversarial Network (DCGANs) [4] : using CNNs as both Generator and Discriminator.



Generator

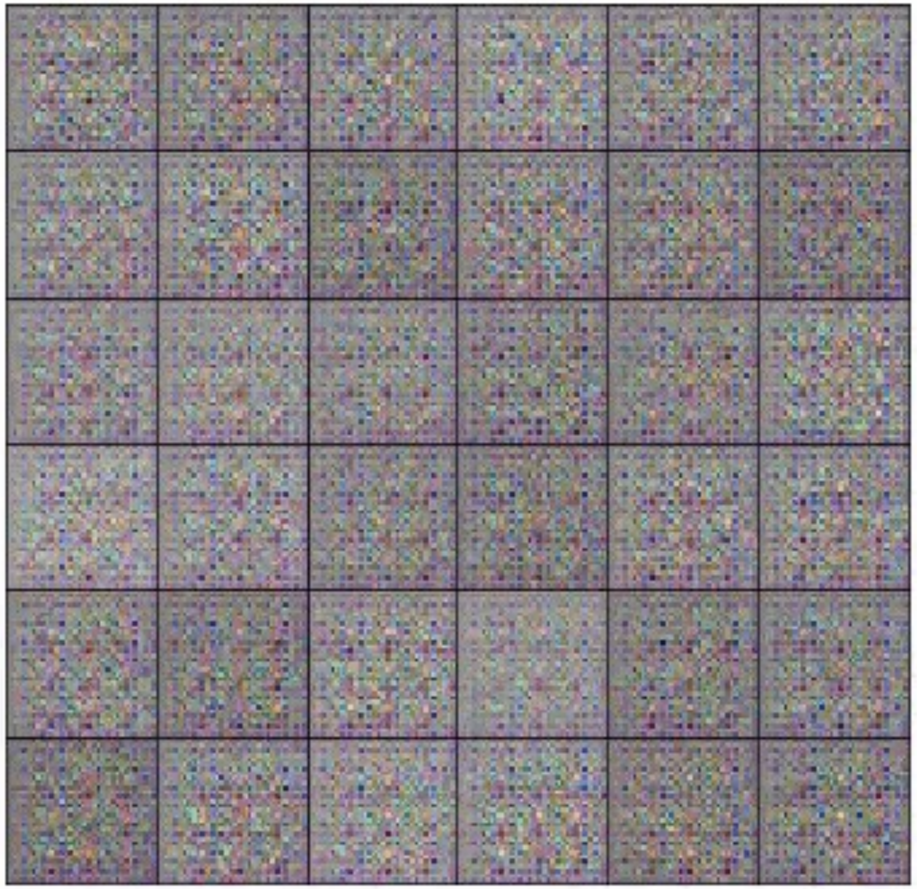
Image Credit: [2]

Outline

- Generative Adversarial Networks (GANs)
    - Zero-sum game & min-max loss
    - Optimal discriminator
    - Global minimum
    - Architectures
    - **Results & Challenges**
- Variants
    - Wasserstein GANs
    - Progressive GANs
    - Cycle GANs
    - MolGANs

Results

Samples from GANs during training on SVHNs (left) and MNIST (right)



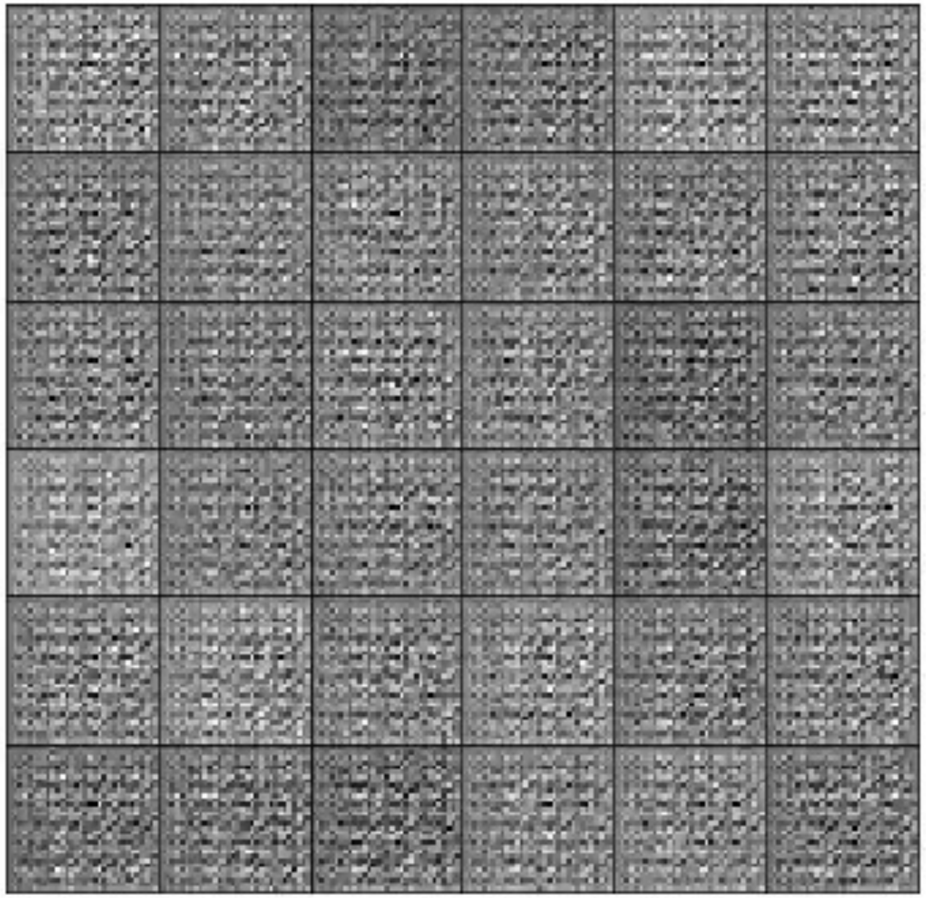
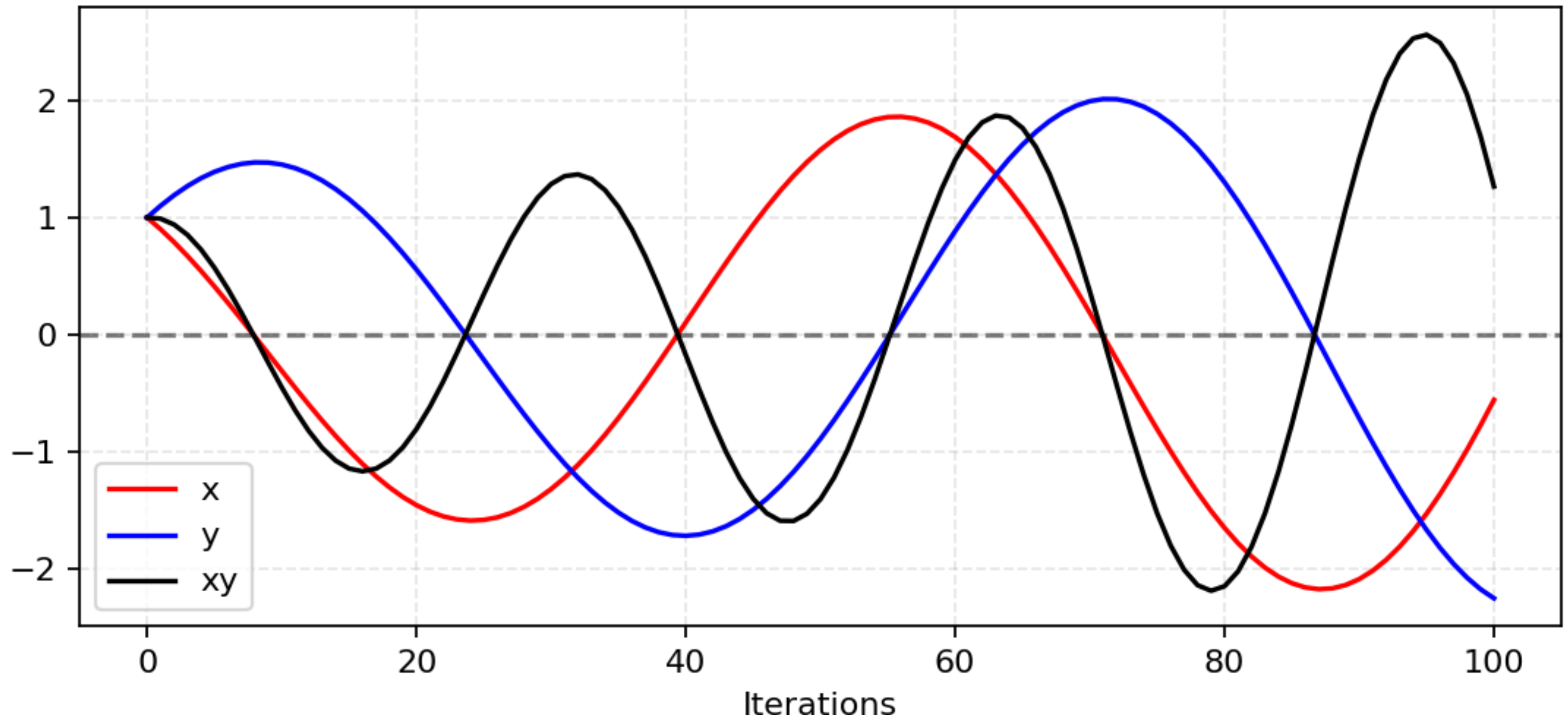


Image Credit: [2]

Challenges in Training GANs

1) ***Training instability***

Hard to reach Nash Equilibrium:

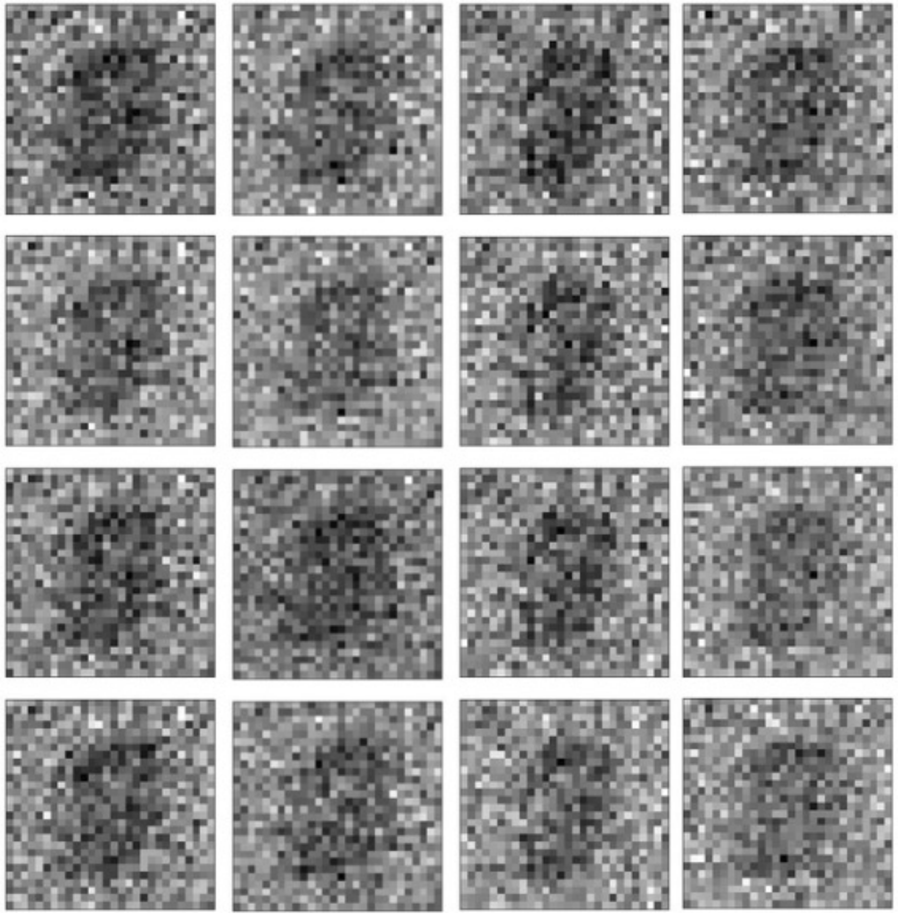


A simulation for updating x to minimize xy and updating y to minimize $-xy$. The learning rate $\eta = 0.1$. With more iterations, the oscillation grows more and more unstable.

Image & Example Credit: [3]

Challenges in Training GANs

1) ***Training instability***



Convergence Failure: e.g., caused by imbalance training of generator and discriminator

Image Credit: [5]

Challenges in Training GANs

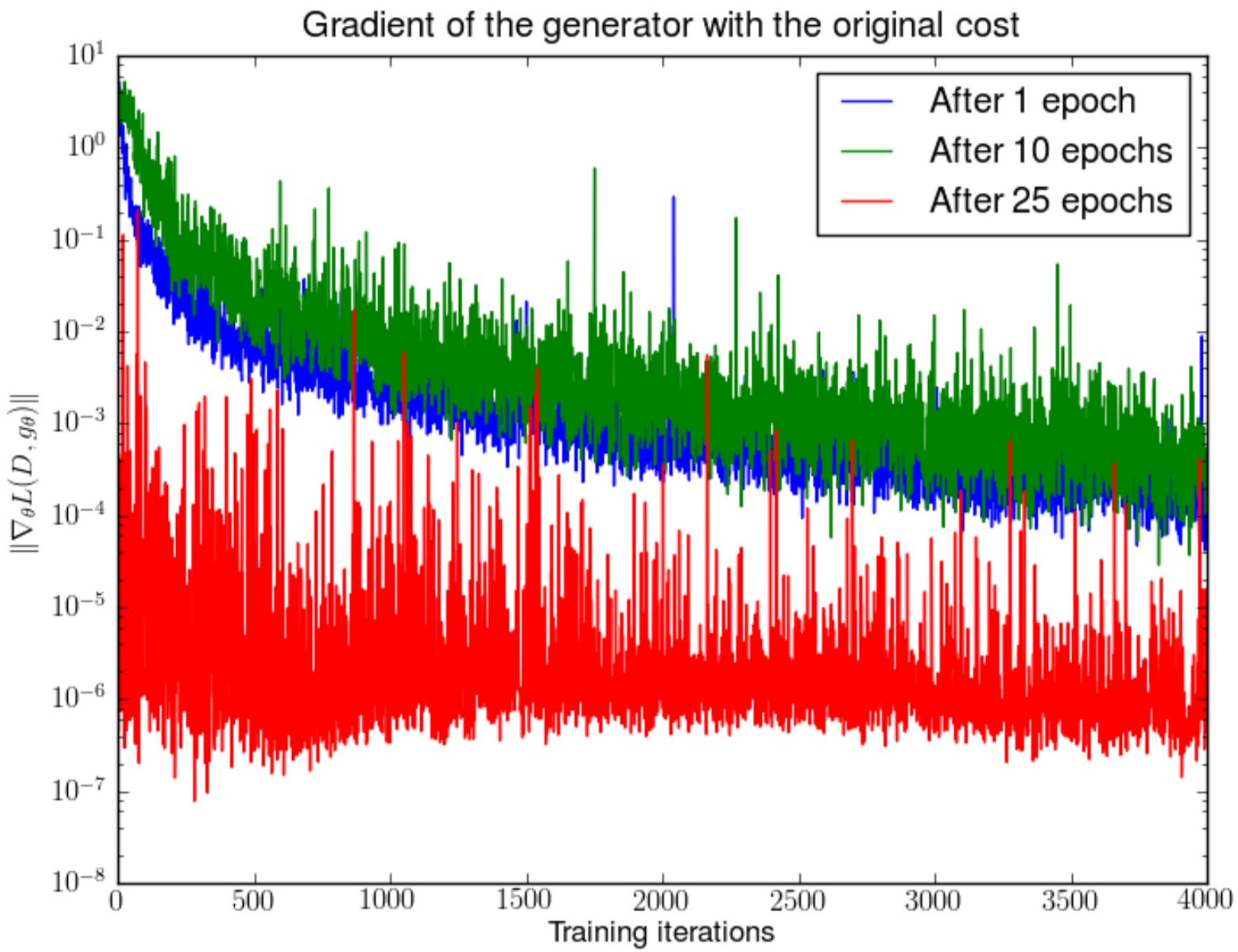
1) *Training instability, 2)* ***Mode collapse***

Mode Collapse: generating samples that are very similar or even identical

Image Credit: [5]

Challenges in Training GANs

1) *Training instability, 2) Mode collapse,* ***3) Vanishing gradient***



First, a DCGAN is trained for 1, 10 and 25 epochs. Then, with **the generator fixed**, a discriminator is trained from scratch and measure the gradients with the original cost function. We see the gradient norms **decay quickly** (in log scale), in the best case 5 orders of magnitude after 4000 discriminator iterations.

Image & Example Credit: [6]

Outline

- Generative Adversarial Networks (GANs)
  - Zero-sum game & min-max loss
  - Optimal discriminator
  - Global minimum
  - Architectures
  - Results & Challenges
- Variants
  - **Wasserstein GANs**
  - Progressive GANs
  - Cycle GANs
  - MolGANs

Earth Mover Distance (Wasserstein-1 Distance)

Suppose we have two distributions P_r and P_θ, we want to match them by “moving dirt” from P_r to P_θ.

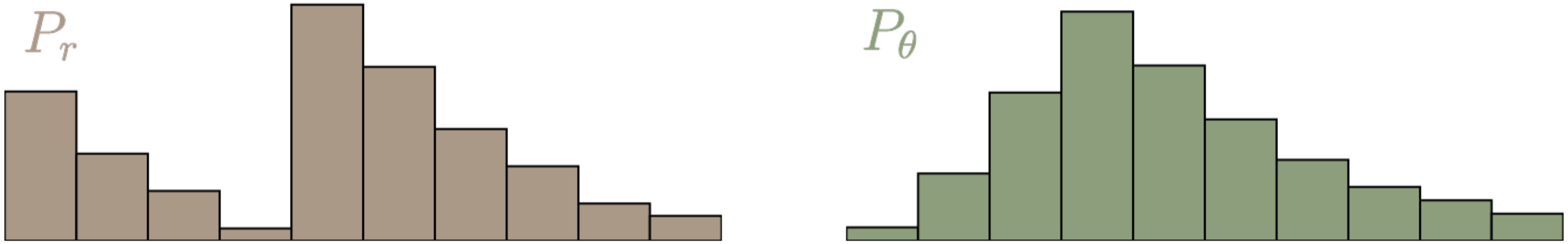
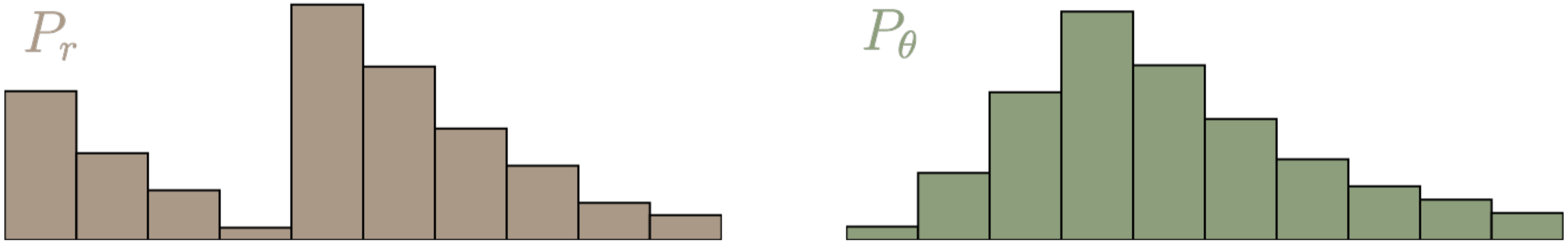


Image & Example Credit: [7]

Earth Mover Distance (Wasserstein-1 Distance)

Suppose we have two distributions P_r and P_θ, we want to match them by "moving dirt" from P_r to P_θ.



Transportation Plan: we split the "dirt" (probability) and move it to different locations to match them.

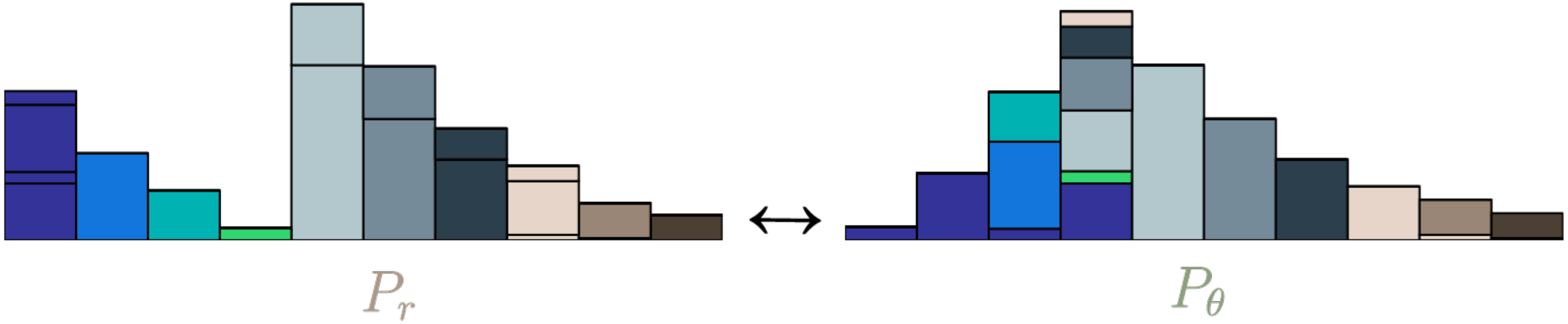


Image & Example Credit: [7]

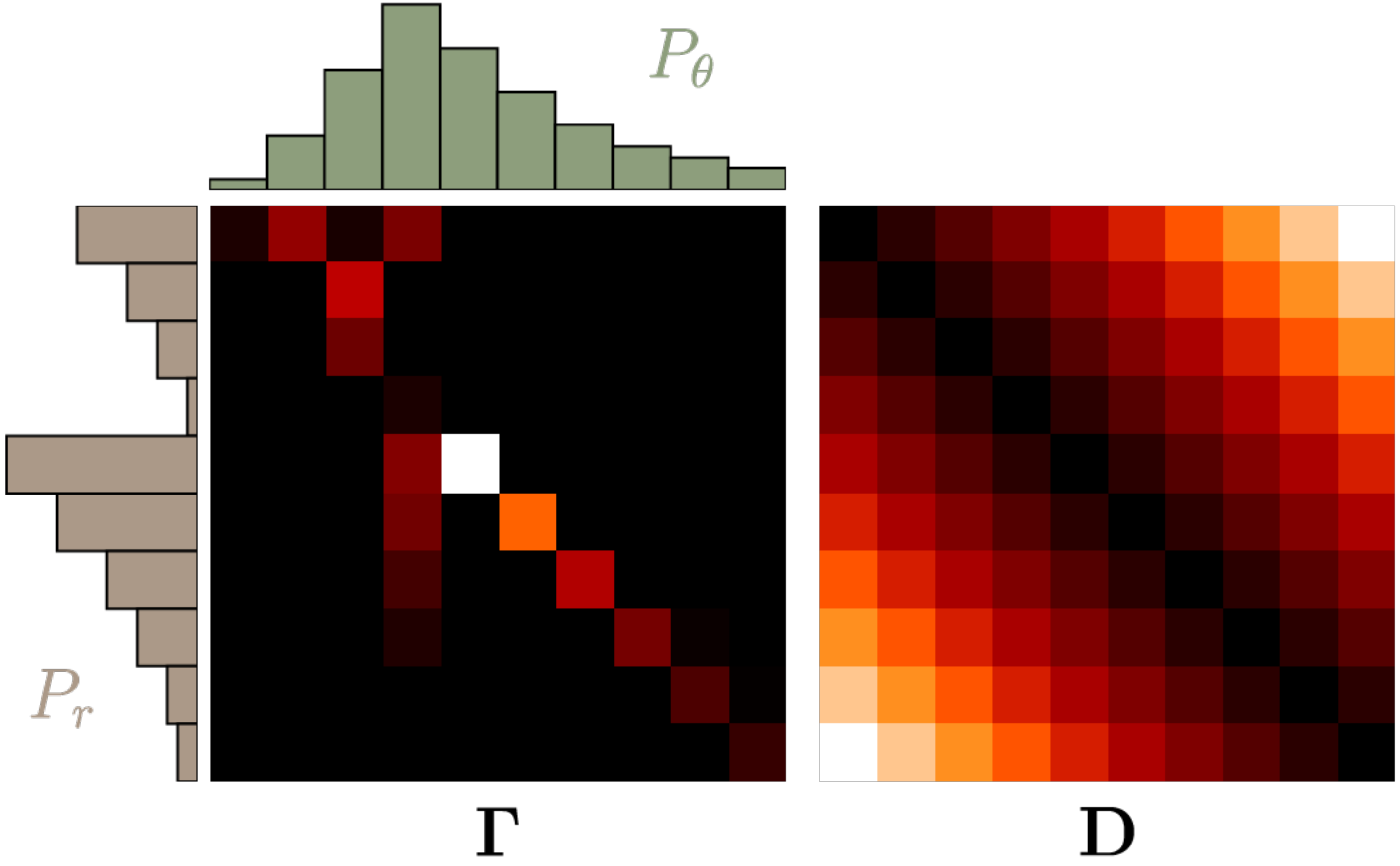
Earth Mover Distance (Wasserstein-1 Distance)

$$\mathrm{EMD}(P_r, P_\theta) = \inf_{\gamma \in \Pi} \sum_{x,y} \|x - y\| \gamma(x, y) = \inf_{\gamma \in \Pi} \mathbb{E}_{(x,y)\sim\gamma} \|x - y\| = \inf_{\gamma \in \Pi} \langle \mathbf{D}, \mathbf{\Gamma} \rangle_{\mathrm{F}}$$

$\prod$ is the set of all distributions whose marginals are P_r, P_θ respectively, called *couplings*.

Earth Mover Distance (Wasserstein-1 Distance)

$$\mathrm{EMD}(P_r, P_\theta) = \inf_{\gamma \in \Pi} \sum_{x,y} \|x - y\| \gamma(x,y) = \inf_{\gamma \in \Pi} \mathbb{E}_{(x,y)\sim\gamma} \|x - y\| = \inf_{\gamma \in \Pi} \langle \mathbf{D}, \mathbf{\Gamma} \rangle_{\mathrm{F}}$$



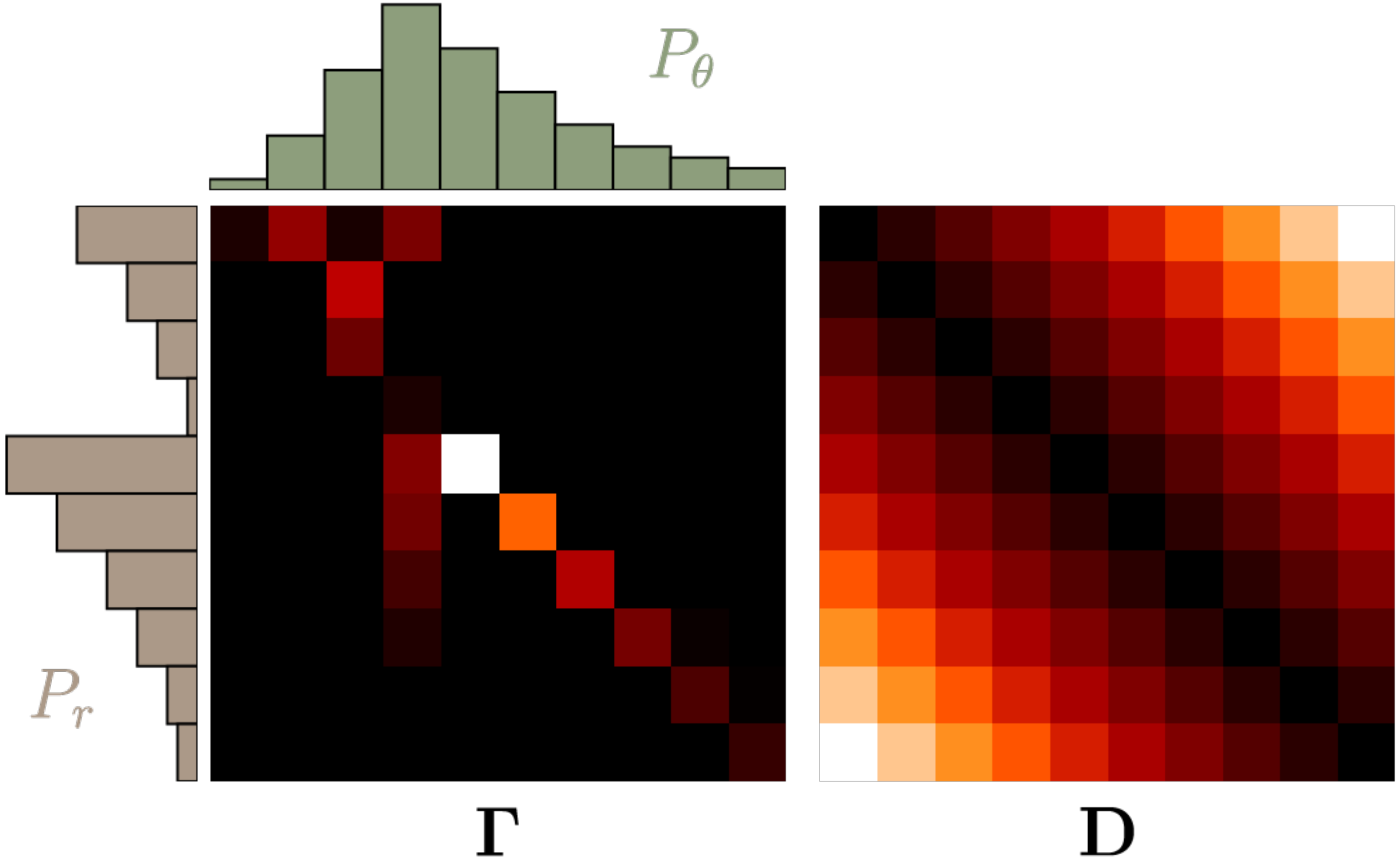
$\mathbf{\Gamma} = \gamma(x,y)$ $\mathbf{D} = \|x - y\|$

$\prod$ is the set of all distributions whose marginals are P_r, P_θ respectively, called *couplings*.

Image Credit: [7]

Earth Mover Distance (Wasserstein-1 Distance)

$$\mathrm{EMD}(P_r, P_\theta) = \inf_{\gamma\in\Pi} \sum_{x,y} \|x-y\|\gamma(x,y) = \inf_{\gamma\in\Pi} \mathbb{E}_{(x,y)\sim\gamma}\|x-y\| = \inf_{\gamma\in\Pi} \langle \mathbf{D}, \mathbf{\Gamma}\rangle_{\mathrm{F}}$$



$\mathbf{\Gamma} = \gamma(x,y)$

Transportation Plan

$\mathbf{D} = \|x-y\|$

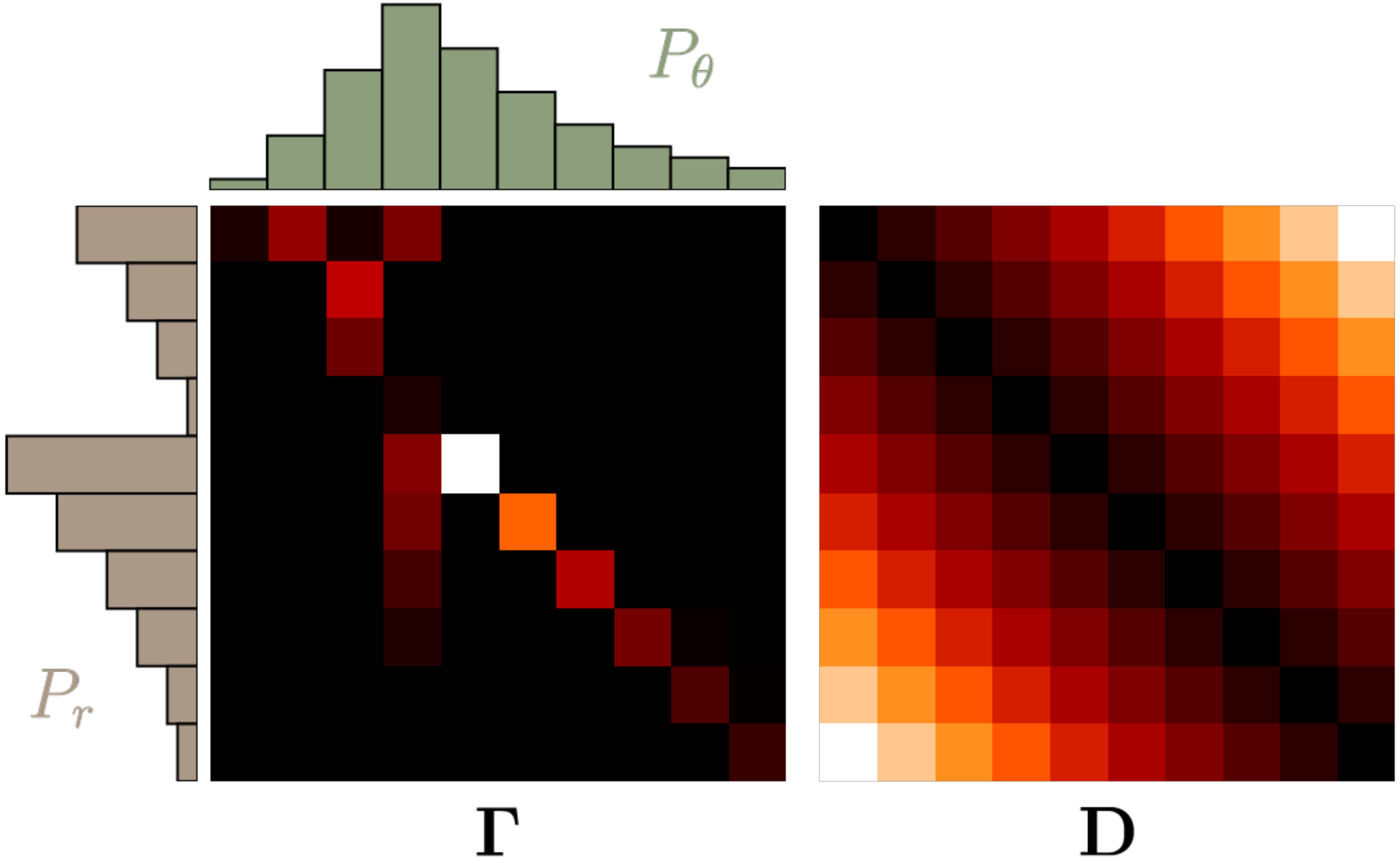
Cost

$\prod$ is the set of all distributions whose marginals are P_r, P_θ respectively, called *couplings*.

Image Credit: [7]

Earth Mover Distance (Wasserstein-1 Distance)

$$\mathrm{EMD}(P_r, P_\theta) = \inf_{\gamma \in \Pi} \sum_{x,y} \|x - y\| \gamma(x, y) = \inf_{\gamma \in \Pi} \mathbb{E}_{(x,y)\sim\gamma} \|x - y\| = \inf_{\gamma \in \Pi} \langle \mathbf{D}, \mathbf{\Gamma} \rangle_{\mathrm{F}}$$



$\mathbf{\Gamma} = \gamma(x, y)$

Transportation Plan

$\mathbf{D} = \|x - y\|$

Cost

$\prod$ is the set of all distributions whose marginals are P_r, P_θ respectively, called *couplings*.

One can generalize it to Wasserstein-p Distance:

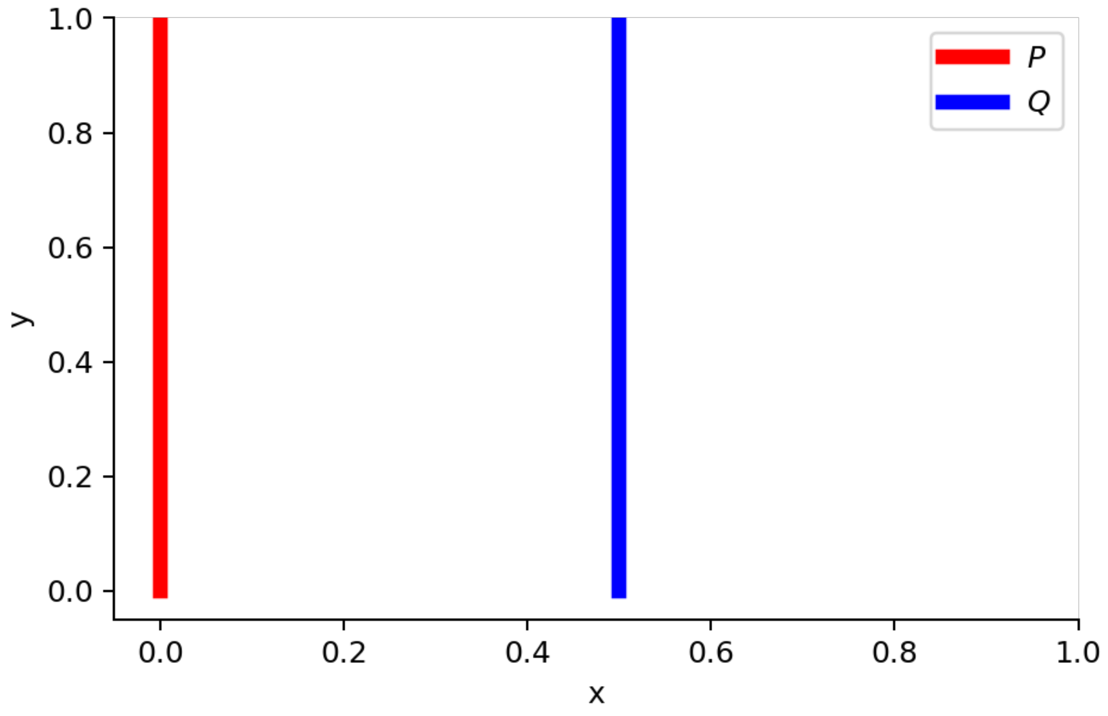
$$W_p(P_r, P_\theta) = \left(\inf_{\gamma \in \Pi} \mathbb{E}_{(x,y)\sim\gamma} d(x, y)^p \right)^{1/p}$$

Image Credit: [7]

Why Wasserstein Distance?

Consider two distributions:

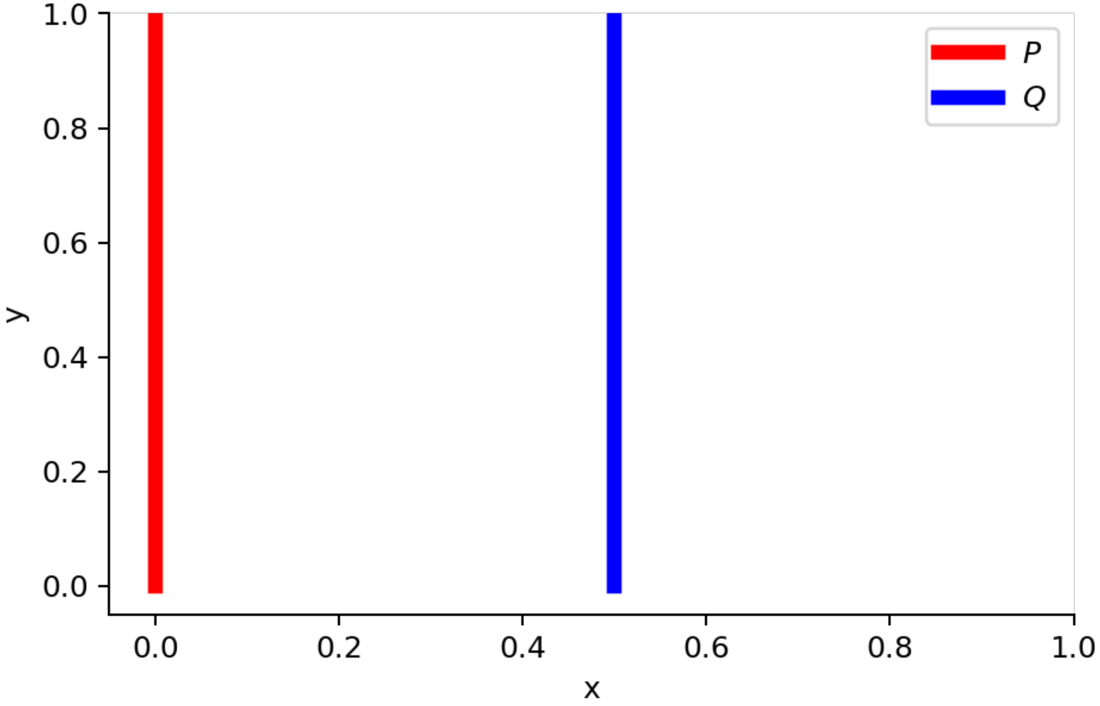
$$\forall (x, y) \in P \quad x = 0, \quad y \sim U(0, 1)$$
$$\forall (x, y) \in Q \quad x = \theta \ (0 \leq \theta \leq 1), \quad y \sim U(0, 1)$$



Example & Image Credit: [3]

Why Wasserstein Distance?

Consider two distributions:



$$\forall (x, y) \in P \quad x = 0, \quad y \sim U(0, 1)$$
$$\forall (x, y) \in Q \quad x = \theta \ (0 \leq \theta \leq 1), \quad y \sim U(0, 1)$$

If $\theta \neq 0$:

$$D_{KL}(P \| Q) = \sum_{\substack{x=0 \\ y \sim U(0,1)}} 1 \cdot \log \frac{1}{0} = +\infty$$

$$D_{KL}(Q \| P) = \sum_{\substack{x=\theta \\ y \sim U(0,1)}} 1 \cdot \log \frac{1}{0} = +\infty$$

$$D_{JS}(P, Q) = \frac{1}{2}\Big(\sum_{\substack{x=0 \\ y \sim U(0,1)}} 1 \cdot \log \frac{1}{1/2} + \sum_{\substack{x=\theta \\ y \sim U(0,1)}} 1 \cdot \log \frac{1}{1/2} \Big) = \log 2$$

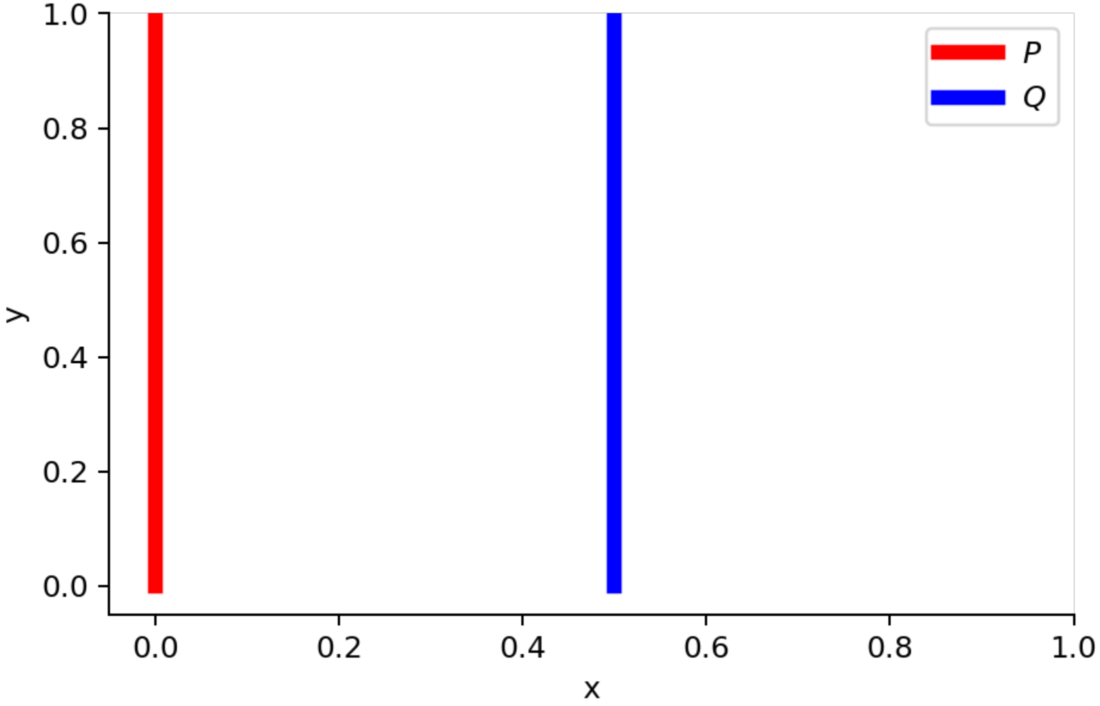
$$W(P, Q) = |\theta|$$

Example & Image Credit: [3]

Why Wasserstein Distance?

Consider two distributions:

$$\forall(x,y) \in P \quad x = 0, \quad y \sim U(0,1)$$
$$\forall(x,y) \in Q \quad x = \theta \ (0 \leq \theta \leq 1), \quad y \sim U(0,1)$$



If $\theta \neq 0$:

$$D_{KL}(P\|Q) = \sum_{\substack{x=0 \\ y\sim U(0,1)}} 1 \cdot \log \frac{1}{0} = +\infty$$

$$D_{KL}(Q\|P) = \sum_{\substack{x=\theta \\ y\sim U(0,1)}} 1 \cdot \log \frac{1}{0} = +\infty$$

$$D_{JS}(P,Q) = \frac{1}{2}\Big(\sum_{\substack{x=0 \\ y\sim U(0,1)}} 1 \cdot \log \frac{1}{1/2} + \sum_{\substack{x=\theta \\ y\sim U(0,1)}} 1 \cdot \log \frac{1}{1/2}\Big) = \log 2$$

$$W(P,Q) = |\theta|$$

Else:

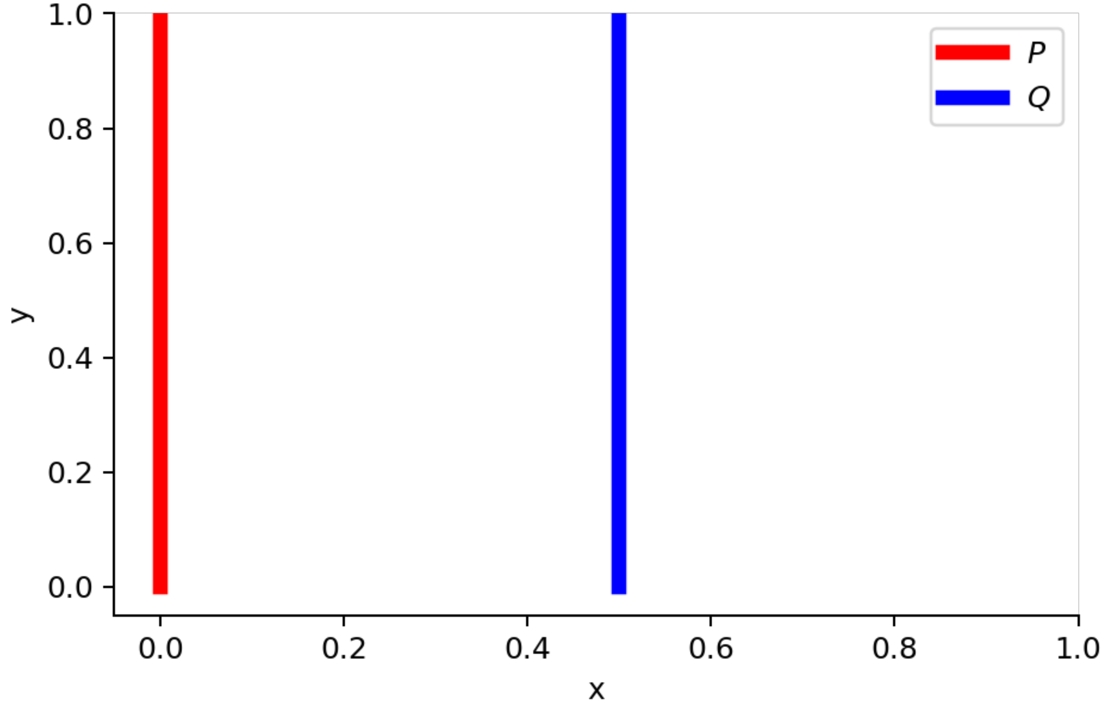
$$D_{KL}(P\|Q) = D_{KL}(Q\|P) = D_{JS}(P,Q) = 0$$
$$W(P,Q) = 0 = |\theta|$$

Example & Image Credit: [3]

Why Wasserstein Distance?

Consider two distributions:

$$\forall(x,y)\in P \quad x=0,\quad y\sim U(0,1)$$
$$\forall(x,y)\in Q \quad x=\theta\ (0\leq\theta\leq 1),\quad y\sim U(0,1)$$



If $\theta\neq 0$:

$$D_{KL}(P\|Q)=\sum_{\substack{x=0\\ y\sim U(0,1)}} 1\cdot\log\frac{1}{0}=+\infty$$

$$D_{KL}(Q\|P)=\sum_{\substack{x=\theta\\ y\sim U(0,1)}} 1\cdot\log\frac{1}{0}=+\infty$$

$$D_{JS}(P,Q)=\frac{1}{2}\Big(\sum_{\substack{x=0\\ y\sim U(0,1)}} 1\cdot\log\frac{1}{1/2}+\sum_{\substack{x=\theta\\ y\sim U(0,1)}} 1\cdot\log\frac{1}{1/2}\Big)=\log 2$$

$$W(P,Q)=|\theta|$$

Else:

$$D_{KL}(P\|Q)=D_{KL}(Q\|P)=D_{JS}(P,Q)=0$$
$$W(P,Q)=0=|\theta|$$

Wasserstein distance is smooth, which is helpful for gradient based learning!

Example & Image Credit: [3]

Wasserstein GANs

Earth Mover Distance / Wasserstein Metric:

$$\mathrm{EMD}(P_r, P_\theta) = \inf_{\gamma \in \Pi} \sum_{x,y} \|x - y\| \gamma(x, y) = \inf_{\gamma \in \Pi} \mathbb{E}_{(x,y)\sim\gamma} \|x - y\|$$

$$= \inf_{\gamma \in \Pi} \langle \mathbf{D}, \mathbf{\Gamma} \rangle_{\mathrm{F}}$$

It is typically hard to compute (need to solve linear programming for discrete distributions)!

Wasserstein GANs

Earth Mover Distance / Wasserstein Metric:

$$\mathrm{EMD}(P_r, P_\theta) = \inf_{\gamma \in \Pi} \sum_{x,y} \|x - y\| \gamma(x, y) = \inf_{\gamma \in \Pi} \mathbb{E}_{(x,y)\sim\gamma} \|x - y\|$$
$$= \inf_{\gamma \in \Pi} \langle \mathbf{D}, \mathbf{\Gamma} \rangle_{\mathrm{F}}$$

It is typically hard to compute (need to solve linear programming for discrete distributions)!

Wasserstein distance (using Kantorovich-Rubinstein duality, see, e.g., [8]):

$$\mathrm{EMD}(P_r, P_\theta) = \sup_{\|f\|_{L \leq 1}} \mathbb{E}_{x \sim P_r} f(x) - \mathbb{E}_{x \sim P_\theta} f(x).$$

Wasserstein GANs

Earth Mover Distance / Wasserstein Metric:

$$\begin{aligned}\mathrm{EMD}(P_r, P_\theta) &= \inf_{\gamma \in \Pi} \sum_{x,y} \|x - y\| \gamma(x, y) = \inf_{\gamma \in \Pi} \mathbb{E}_{(x,y)\sim\gamma} \|x - y\| \\ &= \inf_{\gamma \in \Pi} \langle \mathbf{D}, \mathbf{\Gamma} \rangle_{\mathrm{F}}\end{aligned}$$

It is typically hard to compute (need to solve linear programming for discrete distributions)!

Wasserstein distance (using Kantorovich-Rubinstein duality, see, e.g., [8]):

$$\mathrm{EMD}(P_r, P_\theta) = \sup_{\|f\|_{L \leq 1}} \mathbb{E}_{x\sim P_r} f(x) - \mathbb{E}_{x\sim P_\theta} f(x).$$

Wasserstein-GAN [8] proposes a unified objective:

Learn Discriminator via $\quad \max_{\phi} \quad \mathbb{E}_{X\sim p_{\mathrm{data}}(X)}[D_\phi(X)] - \mathbb{E}_{\epsilon\sim p(\epsilon)}[D_\phi(G_\theta(\epsilon))]$

Learn Generator via $\quad \min_{\theta} \quad \mathbb{E}_{X\sim p_{\mathrm{data}}(X)}[D_\phi(X)] - \mathbb{E}_{\epsilon\sim p(\epsilon)}[D_\phi(G_\theta(\epsilon))]$

Wasserstein GANs

Earth Mover Distance / Wasserstein Metric:

$$\mathrm{EMD}(P_r, P_\theta) = \inf_{\gamma \in \Pi} \sum_{x,y} \|x - y\| \gamma(x, y) = \inf_{\gamma \in \Pi} \mathbb{E}_{(x,y)\sim\gamma} \|x - y\|$$
$$= \inf_{\gamma \in \Pi} \langle \mathbf{D}, \mathbf{\Gamma} \rangle_{\mathrm{F}}$$

It is typically hard to compute (need to solve linear programming for discrete distributions)!

Wasserstein distance (using Kantorovich-Rubinstein duality, see, e.g., [8]):

$$\mathrm{EMD}(P_r, P_\theta) = \sup_{\|f\|_{L \leq 1}} \mathbb{E}_{x\sim P_r} f(x) - \mathbb{E}_{x\sim P_\theta} f(x).$$

Wasserstein-GAN [8] proposes a unified objective:

Learn Discriminator via $\max_{\phi} \quad \mathbb{E}_{X\sim p_{\mathrm{data}}(X)}[D_\phi(X)] - \mathbb{E}_{\epsilon\sim p(\epsilon)}[D_\phi(G_\theta(\epsilon))]$

Learn Generator via $\min_{\theta} \quad \mathbb{E}_{X\sim p_{\mathrm{data}}(X)}[D_\phi(X)] - \mathbb{E}_{\epsilon\sim p(\epsilon)}[D_\phi(G_\theta(\epsilon))]$

To enforce Lipschitz condition, one can clip weights [8], add gradient penalty (WGAN-GP) [9], and use spectral normalization [10]

Wasserstein GANs

Algorithm 1 WGAN, our proposed algorithm. All experiments in the paper used the default values $\alpha = 0.00005$, $c = 0.01$, $m = 64$, $n_{\text{critic}} = 5$.

Require: : α, the learning rate. c, the clipping parameter. m, the batch size. n_{critic}, the number of iterations of the critic per generator iteration.
Require: : w_0, initial critic parameters. θ_0, initial generator's parameters.

while θ has not converged **do**
    for $t = 0, ..., n_{\text{critic}}$ **do**
        Sample $\{x^{(i)}\}_{i=1}^m \sim \mathbb{P}_r$ a batch from the real data.
        Sample $\{z^{(i)}\}_{i=1}^m \sim p(z)$ a batch of prior samples.
        $g_w \leftarrow \nabla_w \left[\frac{1}{m}\sum_{i=1}^m f_w(x^{(i)}) - \frac{1}{m}\sum_{i=1}^m f_w(g_\theta(z^{(i)}))\right]$
        $w \leftarrow w + \alpha \cdot \text{RMSProp}(w, g_w)$
        $w \leftarrow \text{clip}(w, -c, c)$
    end for
    Sample $\{z^{(i)}\}_{i=1}^m \sim p(z)$ a batch of prior samples.
    $g_\theta \leftarrow -\nabla_\theta \frac{1}{m}\sum_{i=1}^m f_w(g_\theta(z^{(i)}))$
    $\theta \leftarrow \theta - \alpha \cdot \text{RMSProp}(\theta, g_\theta)$
end while

Image Credit: [8]

Wasserstein GANs

Image Credit: [9]

Outline

- Generative Adversarial Networks (GANs)
    - Zero-sum game & min-max loss
    - Optimal discriminator
    - Global minimum
    - Architectures
    - Results & Challenges
- Variants
    - Wasserstein GANs
    - **Progressive GANs**
    - Cycle GANs
    - MolGANs

Progressive GANs

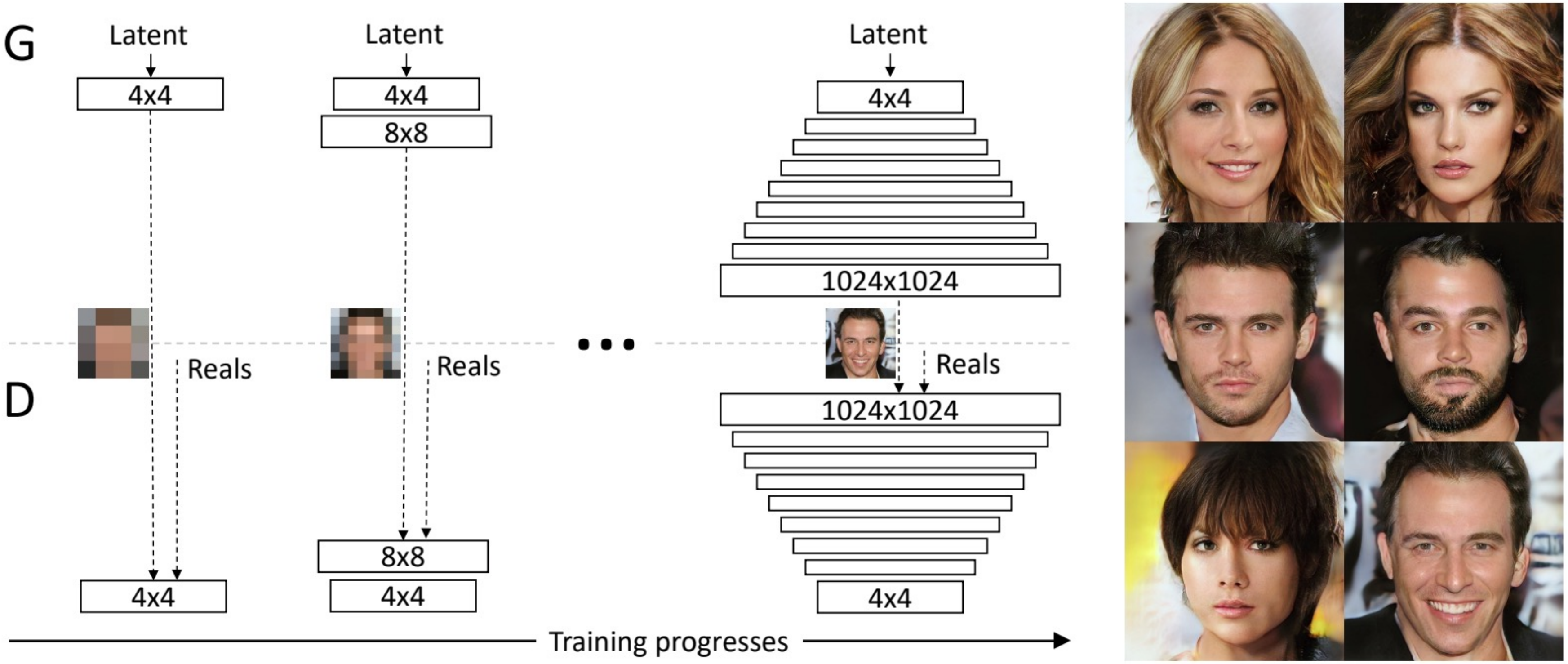


Figure 1: Our training starts with both the generator (G) and discriminator (D) having a low spatial resolution of 4×4 pixels. As the training advances, we incrementally add layers to G and D, thus increasing the spatial resolution of the generated images. All existing layers remain trainable throughout the process. Here $\boxed{N \times N}$ refers to convolutional layers operating on $N \times N$ spatial resolution. This allows stable synthesis in high resolutions and also speeds up training considerably. One the right we show six example images generated using progressive growing at 1024×1024.

Image Credit: [11]

Progressive GANs

Figure 5: 1024 × 1024 images generated using the CELEBA-HQ dataset. See Appendix F for a larger set of results, and the accompanying video for latent space interpolations.

Image Credit: [11]

Outline

- Generative Adversarial Networks (GANs)

    - Zero-sum game & min-max loss
    - Optimal discriminator
    - Global minimum
    - Architectures
    - Results & Challenges

- Variants

    - Wasserstein GANs
    - Progressive GANs
    - **Cycle GANs**
    - MolGANs

Cycle GANs

Cycle-Consistent Generative Adversarial Networks [12] learn the image-to-image translation without a training set of aligned image pairs

Image Credit: [13]

Cycle GANs

Cycle-Consistent Generative Adversarial Networks [12] learn the image-to-image translation without a training set of aligned image pairs

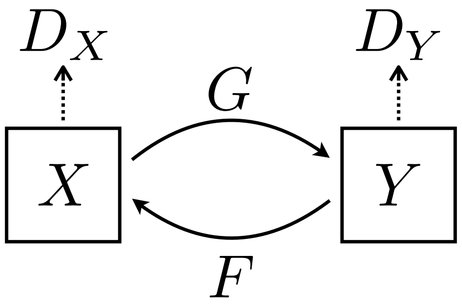
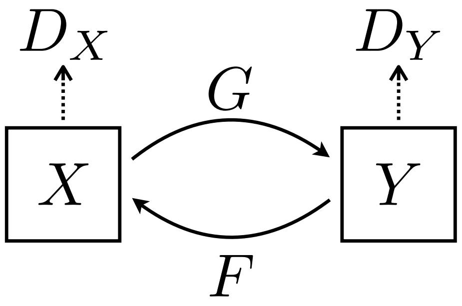


Image Credit: [12,13]

Cycle GANs

Cycle-Consistent Generative Adversarial Networks [12] learn the image-to-image translation without a training set of aligned image pairs

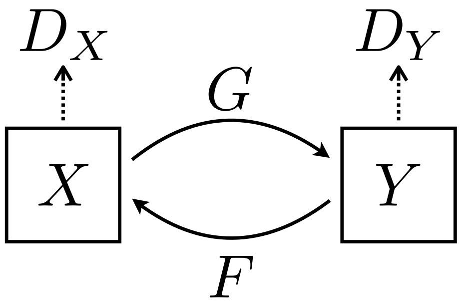


$$\mathcal{L}_{\text{GAN}}(F, D_X, X, Y) = \mathbb{E}_{x \sim p_{\text{data}}(x)}[\log D_X(x)] + \mathbb{E}_{y \sim p_{\text{data}}(y)}[\log(1 - D_X(F(y))]$$

Image Credit: [12,13]

Cycle GANs

Cycle-Consistent Generative Adversarial Networks [12] learn the image-to-image translation without a training set of aligned image pairs



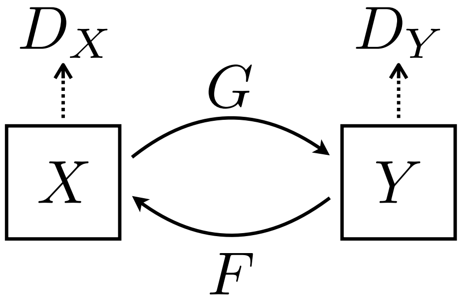
$$\mathcal{L}_{\text{GAN}}(F, D_X, X, Y) = \mathbb{E}_{x\sim p_{\text{data}}(x)}[\log D_X(x)] + \mathbb{E}_{y\sim p_{\text{data}}(y)}[\log(1 - D_X(F(y))]$$

$$\mathcal{L}_{\text{GAN}}(G, D_Y, X, Y) = \mathbb{E}_{y\sim p_{\text{data}}(y)}[\log D_Y(y)] + \mathbb{E}_{x\sim p_{\text{data}}(x)}[\log(1 - D_Y(G(x))]$$

Image Credit: [12,13]

Cycle GANs

Cycle-Consistent Generative Adversarial Networks [12] learn the image-to-image translation without a training set of aligned image pairs



$$\mathcal{L}_{\text{GAN}}(F, D_X, X, Y) = \mathbb{E}_{x\sim p_{\text{data}}(x)}[\log D_X(x)] + \mathbb{E}_{y\sim p_{\text{data}}(y)}[\log(1 - D_X(F(y))]$$

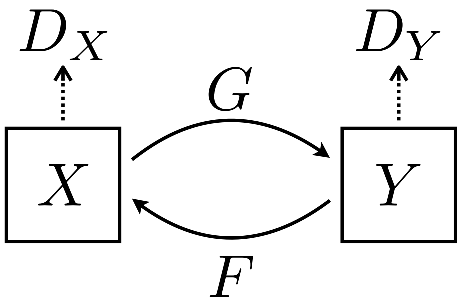
$$\mathcal{L}_{\text{GAN}}(G, D_Y, X, Y) = \mathbb{E}_{y\sim p_{\text{data}}(y)}[\log D_Y(y)] + \mathbb{E}_{x\sim p_{\text{data}}(x)}[\log(1 - D_Y(G(x))]$$

$$\mathcal{L}_{\text{cyc}}(G, F) = \mathbb{E}_{x\sim p_{\text{data}}(x)}[\|F(G(x)) - x\|_1] + \mathbb{E}_{y\sim p_{\text{data}}(y)}[\|G(F(y)) - y\|_1]$$

Image Credit: [12,13]

Cycle GANs

Cycle-Consistent Generative Adversarial Networks [12] learn the image-to-image translation without a training set of aligned image pairs



$$\mathcal{L}_{\text{GAN}}(F, D_X, X, Y) = \mathbb{E}_{x\sim p_{\text{data}}(x)}[\log D_X(x)] + \mathbb{E}_{y\sim p_{\text{data}}(y)}[\log(1 - D_X(F(y))]$$

$$\mathcal{L}_{\text{GAN}}(G, D_Y, X, Y) = \mathbb{E}_{y\sim p_{\text{data}}(y)}[\log D_Y(y)] + \mathbb{E}_{x\sim p_{\text{data}}(x)}[\log(1 - D_Y(G(x))]$$

$$\mathcal{L}_{\text{cyc}}(G, F) = \mathbb{E}_{x\sim p_{\text{data}}(x)}[\|F(G(x)) - x\|_1] + \mathbb{E}_{y\sim p_{\text{data}}(y)}[\|G(F(y)) - y\|_1]$$

$$\mathcal{L}(G, F, D_X, D_Y) = \mathcal{L}_{\text{GAN}}(G, D_Y, X, Y) + \mathcal{L}_{\text{GAN}}(F, D_X, Y, X) + \lambda\mathcal{L}_{\text{cyc}}(G, F)$$

Image Credit: [12,13]

Cycle GANs

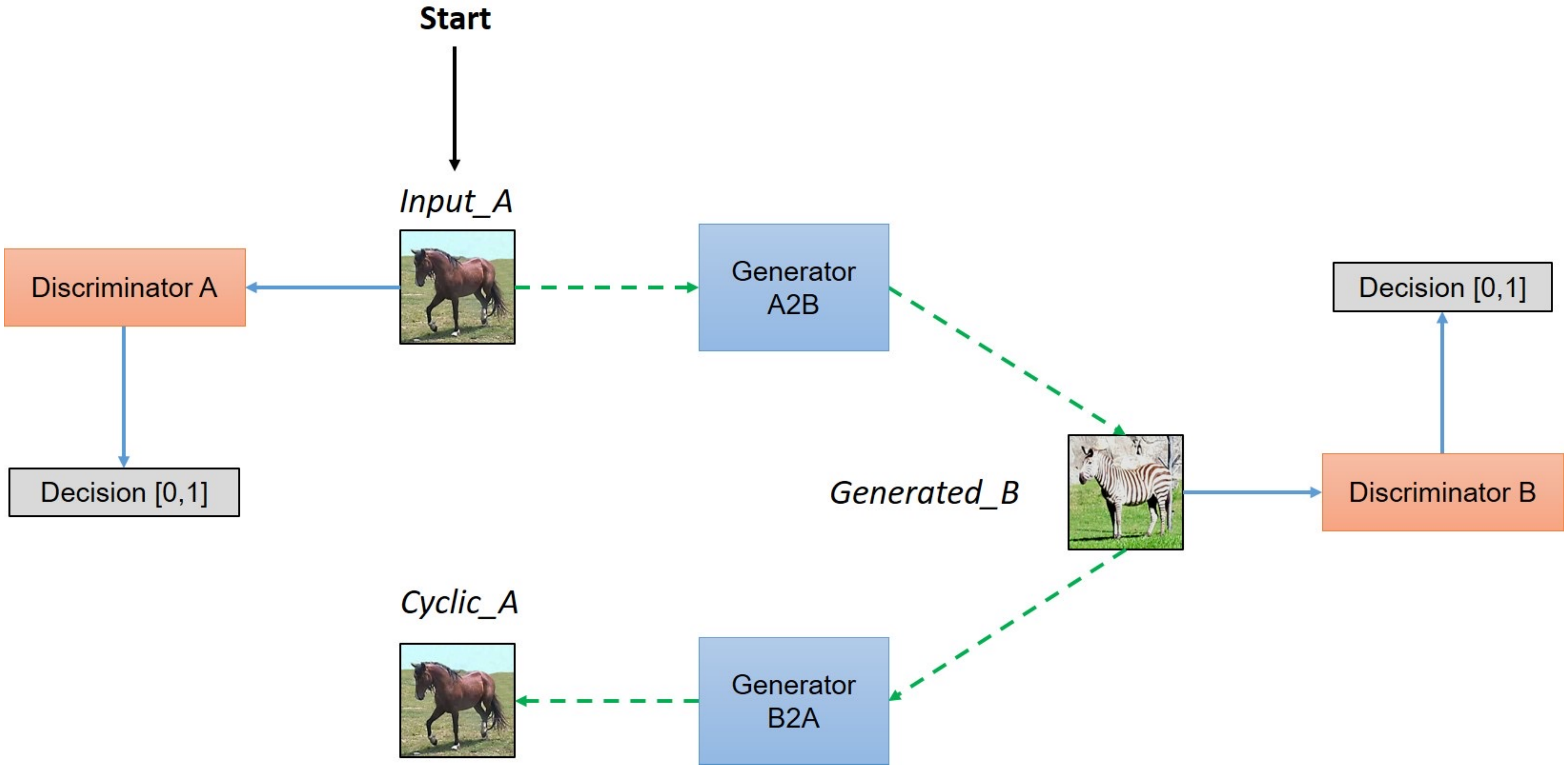


Image Credit: [14]

Cycle GANs

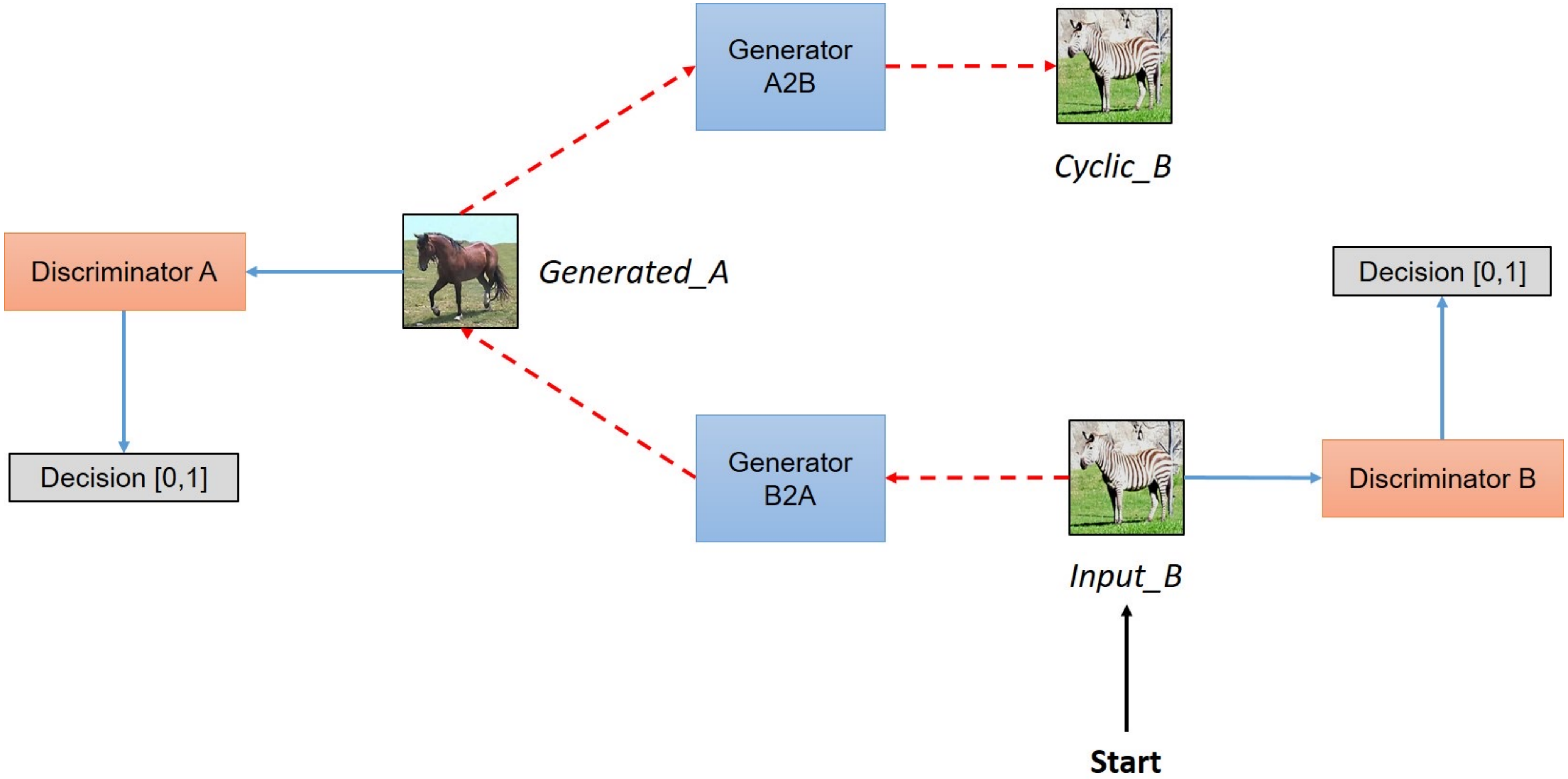


Image Credit: [14]

Cycle GANs

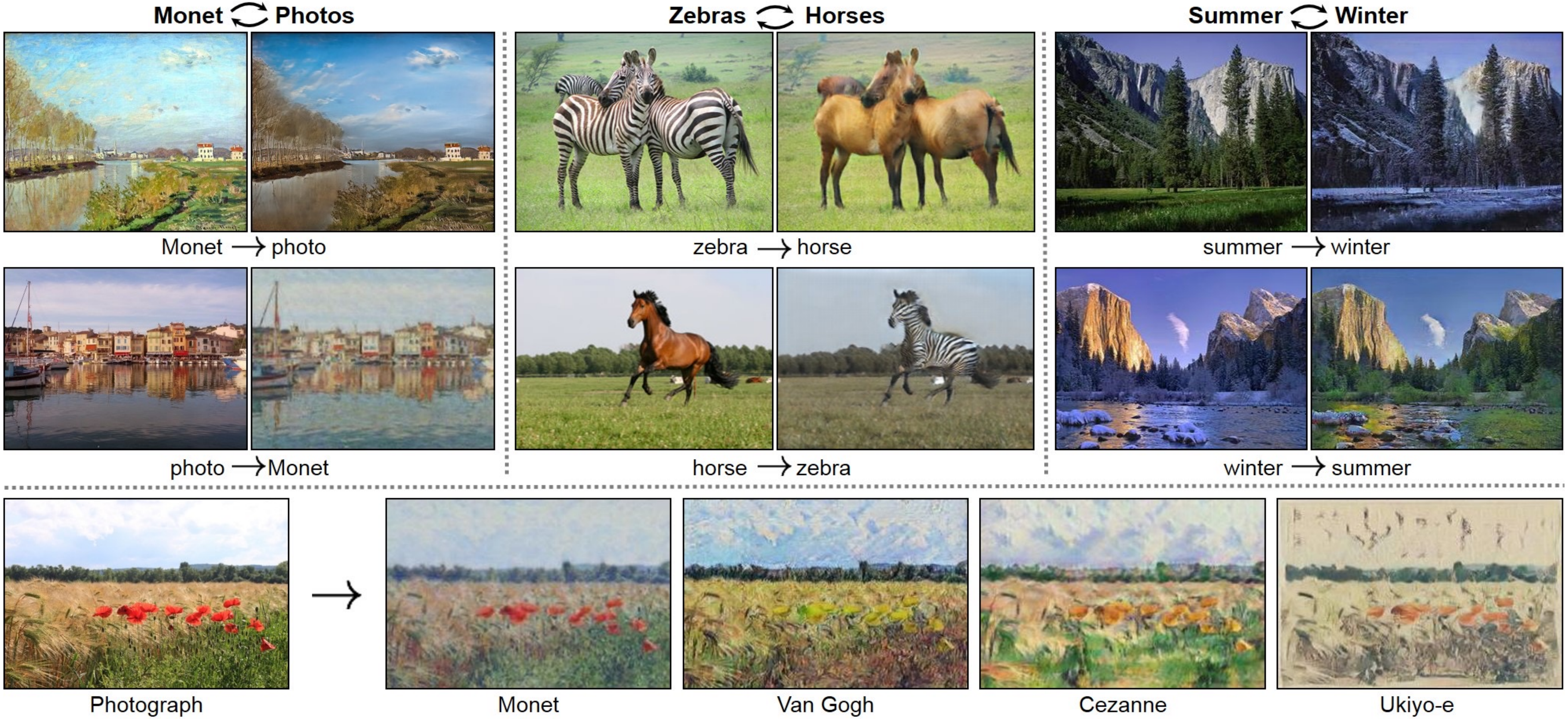


Image Credit: [13]

Outline

- Generative Adversarial Networks (GANs)
  - Zero-sum game & min-max loss
  - Optimal discriminator
  - Global minimum
  - Architectures
  - Results & Challenges
- Variants
  - Wasserstein GANs
  - Progressive GANs
  - Cycle GANs
  - **MolGANs**

MolGANs

MolGANs [15] generate molecular graphs without graph matching:

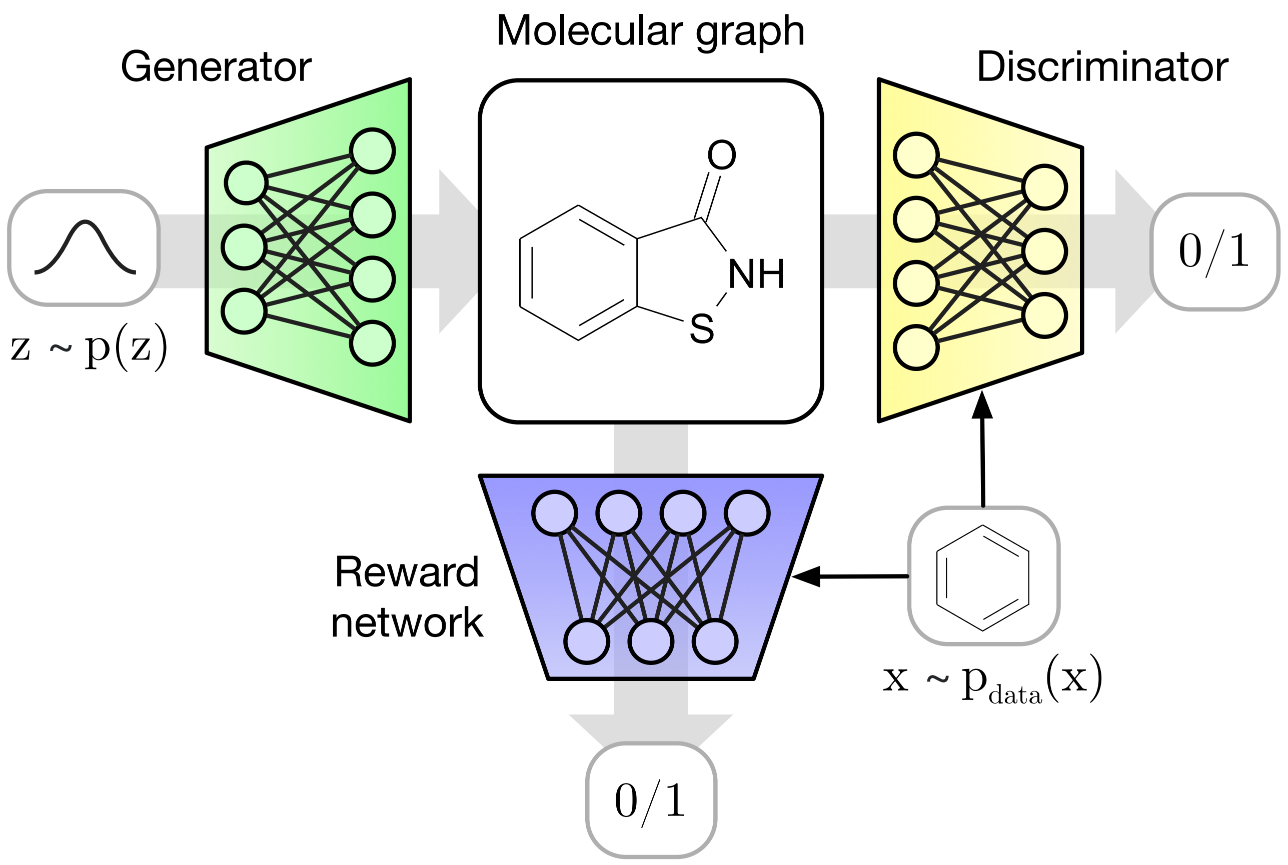


Image Credit: [15]

MolGANs

MolGANs [15] generate molecular graphs without graph matching:

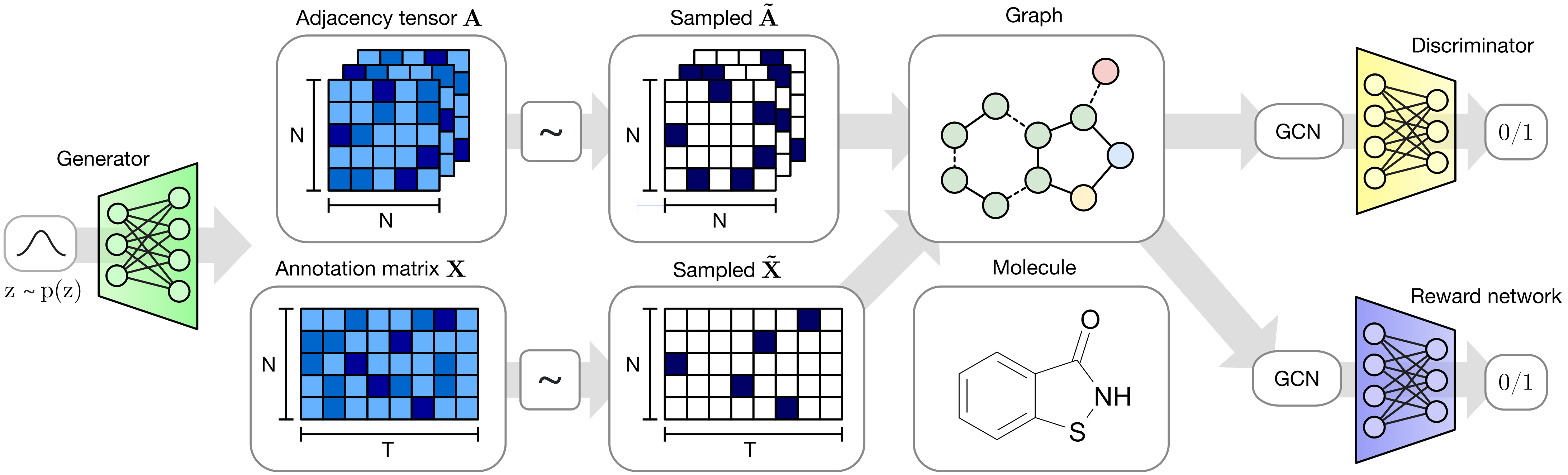


Image Credit: [15]

References

[1] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y., 2014. Generative adversarial nets. Advances in neural information processing systems, 27.

[2] https://sthalles.github.io/intro-to-gans/

[3] https://lilianweng.github.io/posts/2017-08-20-gan/

[4] Radford, A., Metz, L. and Chintala, S., 2015. Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434.

[5] https://neptune.ai/blog/gan-failure-modes

[6] Arjovsky, M. and Bottou, L., 2017. Towards principled methods for training generative adversarial networks. arXiv preprint arXiv:1701.04862.

[7] https://vincentherrmann.github.io/blog/wasserstein/

[8] Arjovsky, M., Chintala, S. and Bottou, L., 2017, July. Wasserstein generative adversarial networks. In International conference on machine learning (pp. 214-223). PMLR.

[9] Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V. and Courville, A.C., 2017. Improved training of wasserstein gans. Advances in neural information processing systems, 30.

[10] Miyato, T., Kataoka, T., Koyama, M. and Yoshida, Y., 2018. Spectral normalization for generative adversarial networks. arXiv preprint arXiv:1802.05957.

References

[11] Karras, T., Aila, T., Laine, S. and Lehtinen, J., 2017. Progressive growing of gans for improved quality, stability, and variation. arXiv preprint arXiv:1710.10196.

[12] Zhu, J.Y., Park, T., Isola, P. and Efros, A.A., 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. In Proceedings of the IEEE international conference on computer vision (pp. 2223-2232).

[13] https://junyanz.github.io/CycleGAN/

[14] https://hardikbansal.github.io/CycleGANBlog/

[15] De Cao, N. and Kipf, T., 2018. MolGAN: An implicit generative model for small molecular graphs. arXiv preprint arXiv:1805.11973.

Questions?