CPEN 400D: Deep Learning

Lecture 9: Generative Adversarial Networks

Renjie Liao

University of British Columbia

Winter, Term 2, 2022

Outline

- **Motivation**
- GANs
  - Overview
  - Minimax Loss
  - Properties
  - Architectures
  - Challenges of GANs
- GANs Variants
  - Wasserstein GANs
  - Progressive GANs
  - Cycle GANs

Deep Generative Models

Deep generative models are the most exciting area in deep/machine learning, AI…

Models can even generate the reflection in the puddle!

Image Credit: Midjourney V5

Deep Generative Models

Deep generative models are the most exciting area in deep/machine learning, AI…

We have learned deep generative models like deep auto-regressive models (e.g., GPT series) and variational auto-encoders (VAEs).

They are trained to maximize the log likelihood of observed data (e.g., next tokens) or its lower bound (ELBO).

Image Credit: Midjourney V5

Deep Generative Models

Deep generative models are the most exciting area in deep/machine learning, AI…

We have learned deep generative models like deep auto-regressive models (e.g., GPT series) and variational auto-encoders (VAEs).

They are trained to maximize the log likelihood of observed data (e.g., next tokens) or its lower bound (ELBO).

Is the likelihood the only good objective in learning deep generative models?

Image Credit: Midjourney V5

Deep Generative Models

Deep generative models are the most exciting area in deep/machine learning, AI…

We have learned deep generative models like deep auto-regressive models (e.g., GPT series) and variational auto-encoders (VAEs).

They are trained to maximize the log likelihood of observed data (e.g., next tokens) or its lower bound (ELBO).

Is the likelihood the only good objective in learning deep generative models?

No, we have adversarial learning, score matching / denoising diffusion, moment matching, and so on.

Image Credit: Midjourney V5

Deep Generative Models

Deep generative models are the most exciting area in deep/machine learning, AI…

We have learned deep generative models like deep auto-regressive models (e.g., GPT series) and variational auto-encoders (VAEs).

They are trained to maximize the log likelihood of observed data (e.g., next tokens) or its lower bound (ELBO).

Is the likelihood the only good objective in learning deep generative models?

No, we have adversarial learning, score matching / denoising diffusion, moment matching, and so on.

The beauty of deep generative models is *all models are wrong, but many are useful!*

Image Credit: Midjourney V5

Outline

- Motivation
- GANs
  - **Overview**
  - Minimax Loss
  - Properties
  - Architectures
  - Challenges of GANs
- GANs Variants
  - Wasserstein GANs
  - Progressive GANs
  - Cycle GANs

Overview

Generative Adversarial Networks (GANs) [1]

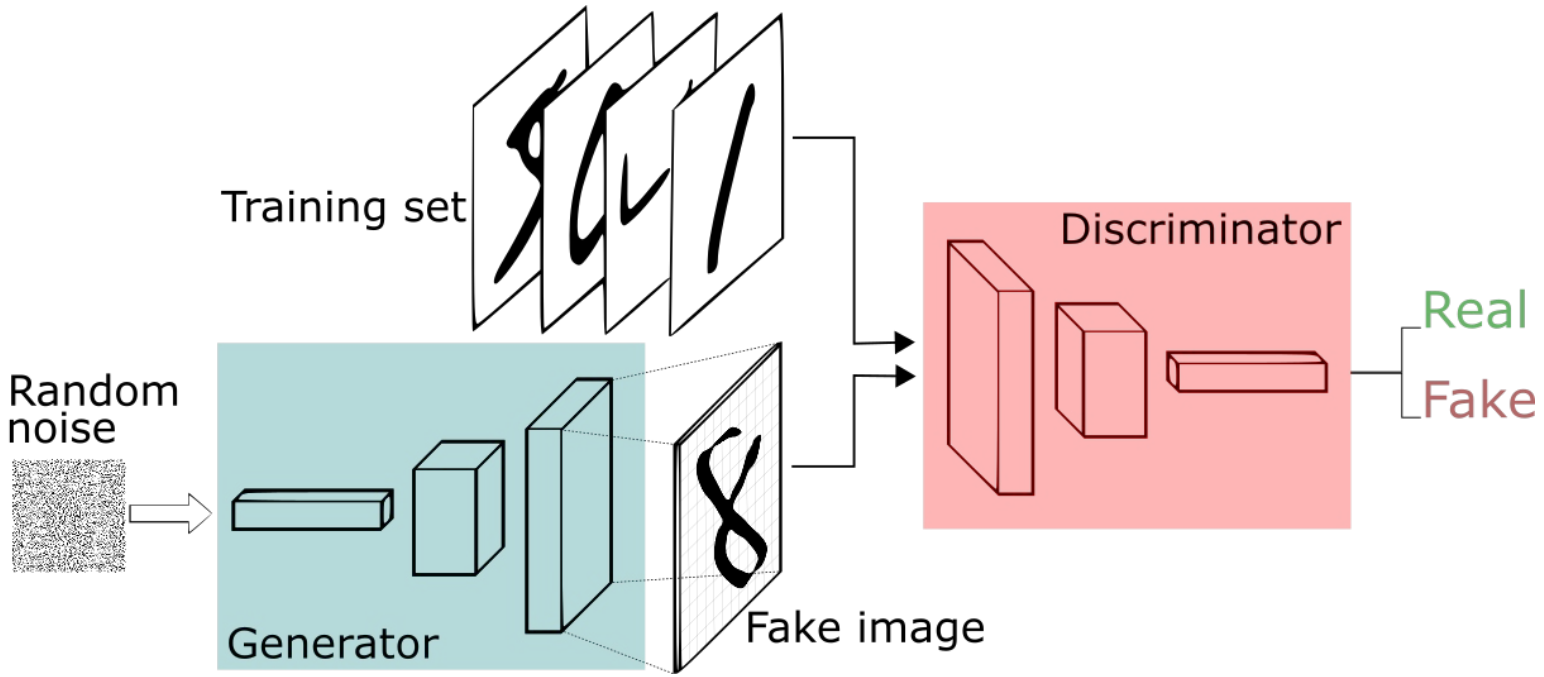


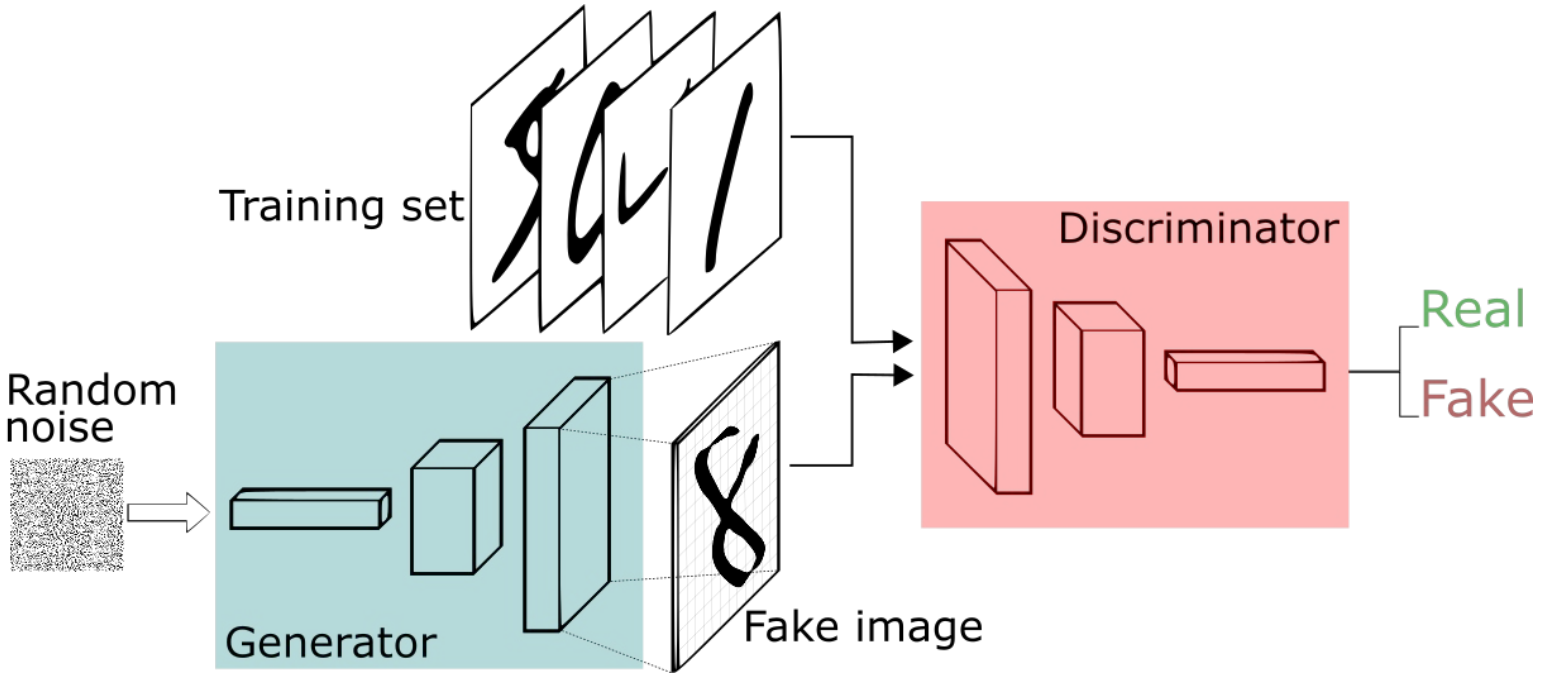
Image Credit: [2]

Outline

- Motivation
- GANs
  - Overview
  - **Minimax Loss**
  - Properties
  - Architectures
  - Challenges of GANs
- GANs Variants
  - Wasserstein GANs
  - Progressive GANs
  - Cycle GANs

Minimax Loss

GANs [1] : Two neural networks (generator and discriminator) contest with each other in the form of a zero-sum game, where one agent's gain is another agent's loss.



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

Image Credit: [2]

Outline

- Motivation
- GANs
  - Overview
  - Minimax Loss
  - **Properties**
  - Architectures
  - Challenges of GANs
- GANs Variants
  - Wasserstein GANs
  - Progressive GANs
  - Cycle GANs

Properties of GANs

Generative Adversarial Networks (GANs) [1]

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

Properties of GANs

Generative Adversarial Networks (GANs) [1]

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

Why?

Properties of GANs

Generative Adversarial Networks (GANs) [1]

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

Why?

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

Properties of GANs

Generative Adversarial Networks (GANs) [1]

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

Why?

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

Law Of The Unconscious Statistician (LOTUS)

Set the gradient of loss w.r.t. D to be zero

Properties of GANs

Generative Adversarial Networks (GANs) [1]

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

Properties of GANs

Generative Adversarial Networks (GANs) [1]

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

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

Why?

Properties of GANs

Generative Adversarial Networks (GANs) [1]

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

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

Why?

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

Properties of GANs

Generative Adversarial Networks (GANs) [1]

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

Why?

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

Jensen–Shannon divergence is non-negative and is zero iff. P = Q

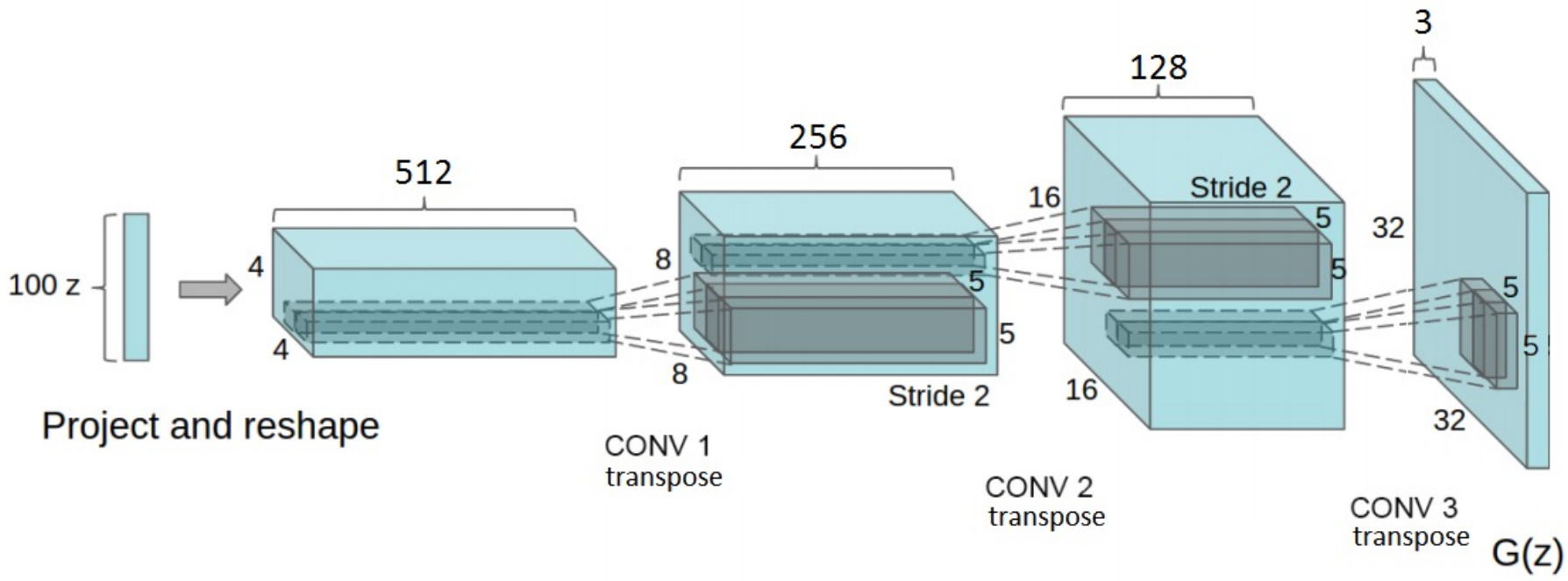
$$
\text{JSD}(P\|Q) = \frac{1}{2}\text{KL}(P\|\frac{P+Q}{2}) + \frac{1}{2}\text{KL}(Q\|\frac{P+Q}{2})
$$

Outline

- Motivation
- GANs
  - Overview
  - Minimax Loss
  - Properties
  - **Architectures**
  - Challenges of GANs
- GANs Variants
  - Wasserstein GANs
  - Progressive GANs
  - Cycle GANs

Architecture

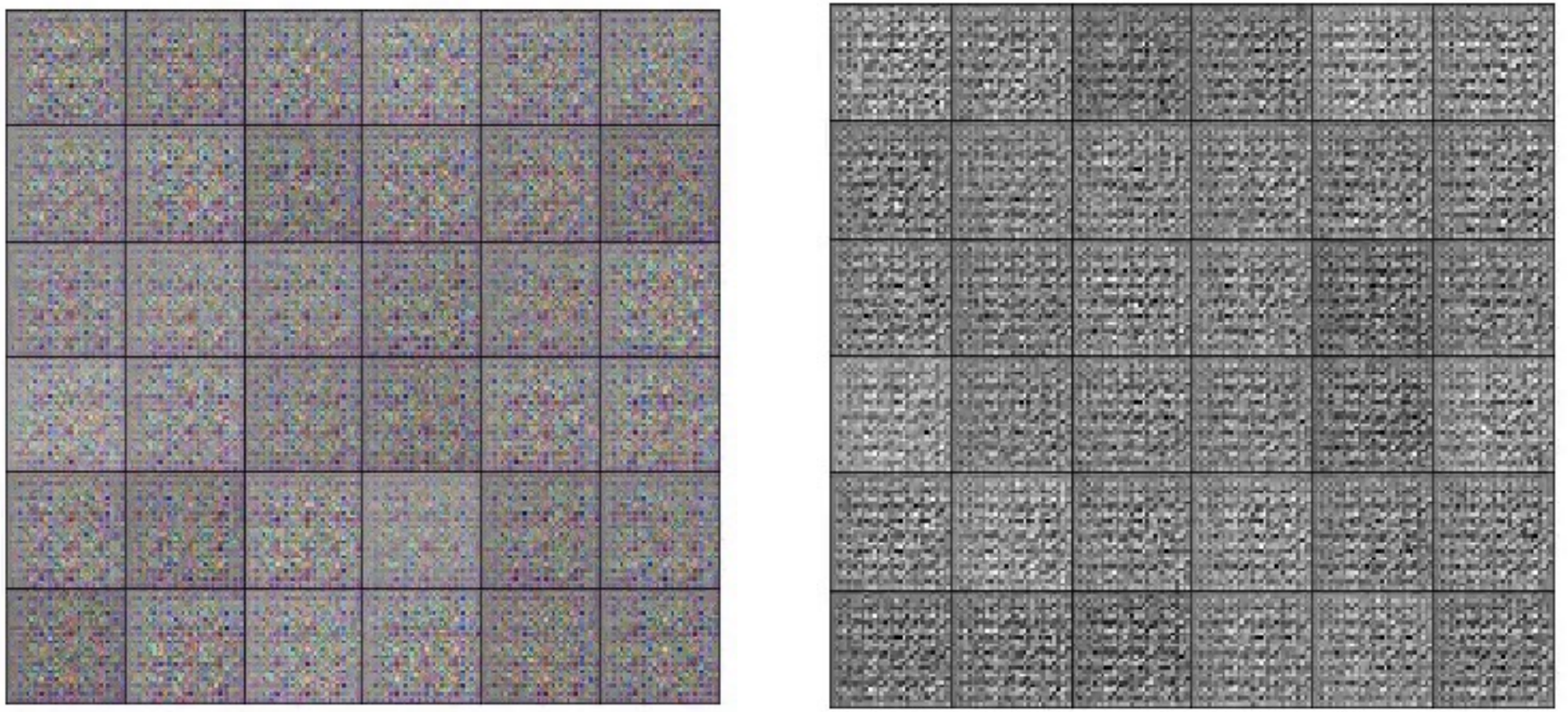
Deep Convolutional Generative Adversarial Network (DCGANs) [3] : using CNNs as both Generator and Discriminator.



Generator

Image Credit: [2]

Demo of GANs



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

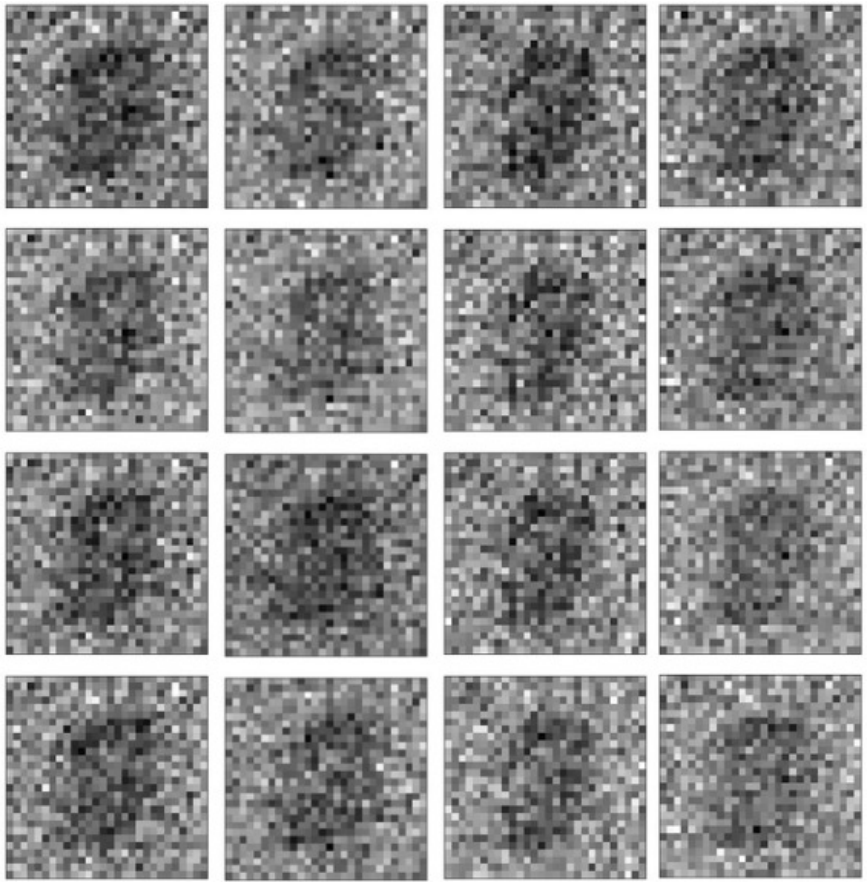
Image Credit: [2]

Outline

- Motivation
- GANs
  - Overview
  - Minimax Loss
  - Properties
  - Architectures
  - **Challenges of GANs**
- GANs Variants
  - Wasserstein GANs
  - Progressive GANs
  - Cycle GANs

Challenges in Training GANs

Two common problems in training GANs are: *training instability*

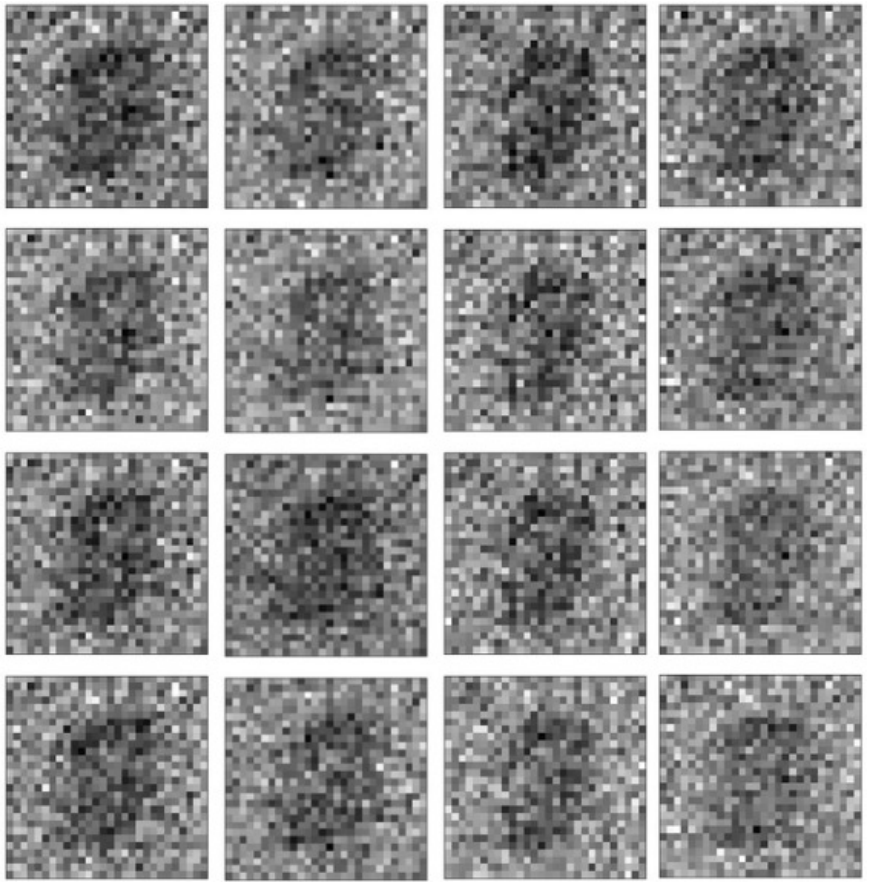


Convergence Failure: e.g., caused by imbalance training of generator and discriminator

Image Credit: [4]

Challenges in Training GANs

Two common problems in training GANs are: *training instability* and *mode collapse*



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

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

Image Credit: [4]

Outline

- Motivation
- GANs
  - Overview
  - Minimax Loss
  - Properties
  - Architectures
  - Challenges of GANs
- GANs Variants
  - **Wasserstein GANs**
  - Progressive GANs
  - Cycle GANs

Wasserstein GANs

Earth Mover Distance / Wasserstein Metric:

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

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

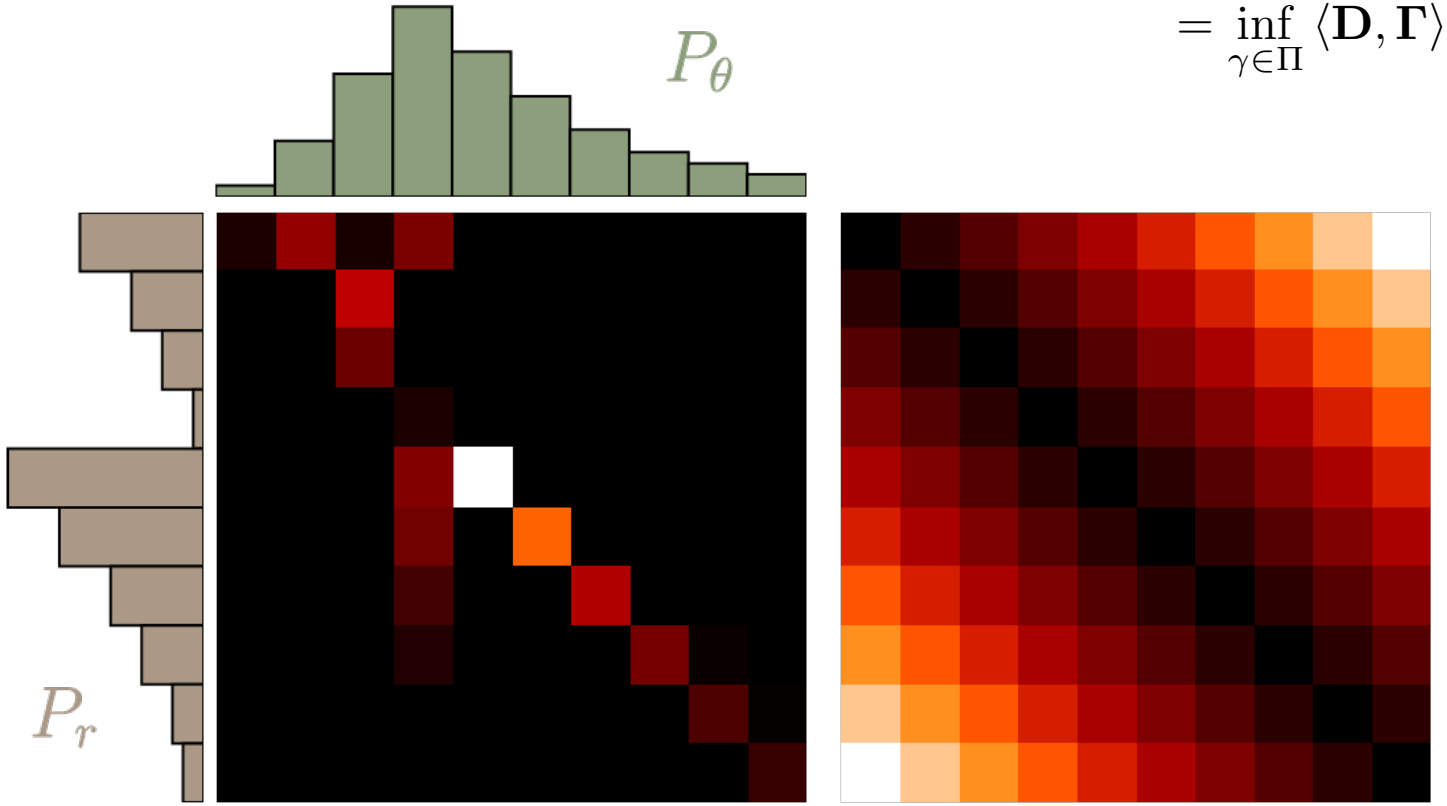
$\prod$ is the set of all distributions whose marginals are P_r, P_θ respectively.

Wasserstein GANs

Earth Mover Distance / Wasserstein Metric: $\text{EMD}(P_r, P_\theta) = \inf_{\gamma \in \Pi} \sum_{x,y} \|x - y\| \gamma(x, y) = \inf_{\gamma \in \Pi} \mathbb{E}_{(x,y) \sim \gamma} \|x - y\|$

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

Π is the set of all distributions whose marginals are P_r, P_θ respectively.



$\mathbf{\Gamma}$

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

$\mathbf{D}$

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

Image Credit: [5]

Wasserstein GANs

Earth Mover Distance / Wasserstein Metric:

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

Wasserstein distance (using Kantorovich-Rubinstein duality):

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

Wasserstein GANs

Earth Mover Distance / Wasserstein Metric: $$\begin{aligned}\text{EMD}(P_r, P_\theta) &= \inf_{\gamma \in \Pi} \sum_{x,y} \|x - y\| \gamma(x, y) = \inf_{\gamma \in \Pi} \mathbb{E}_{(x,y)\sim\gamma} \|x - y\| \\ &= \inf_{\gamma \in \Pi} \langle \mathbf{D}, \mathbf{\Gamma} \rangle_{\text{F}}\end{aligned}$$

Wasserstein distance (using Kantorovich-Rubinstein duality):

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

Wasserstein-GAN [6] proposes a unified objective:

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

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

Wasserstein GANs

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

Wasserstein distance (using Kantorovich-Rubinstein duality):

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

Wasserstein-GAN [6] 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 [6], add gradient penalty (WGAN-GP) [7], and use spectral normalization [8]

Wasserstein GANs

Image Credit: [7]

Outline

- Motivation
- GANs
  - Overview
  - Minimax Loss
  - Properties
  - Architectures
  - Challenges of GANs
- GANs Variants
  - Wasserstein GANs
  - **Progressive GANs**
  - Cycle GANs

Progressive GANs

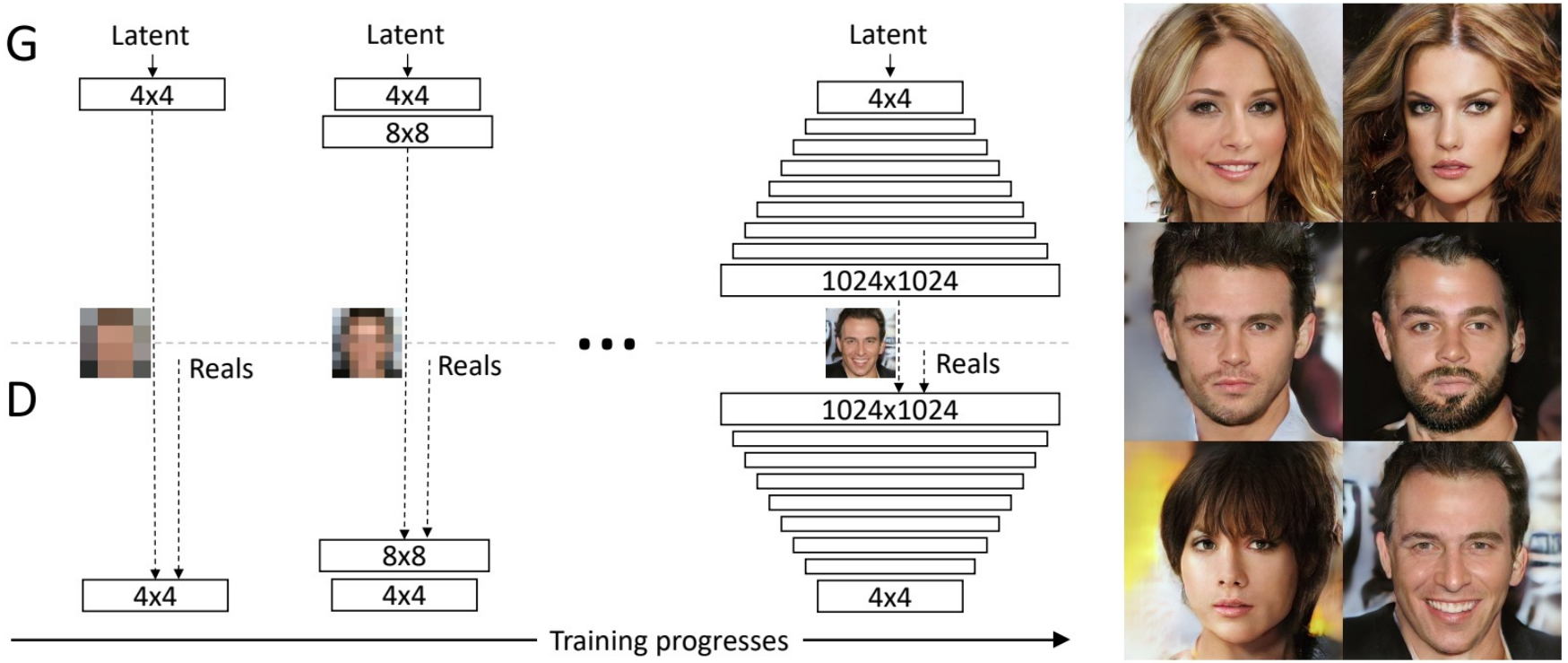


Figure 1: Our training starts with both the generator (G) and discriminator (D) having a low spatial resolution of 4×4 pixels. As the training advances, we incrementally add layers to G and D, thus increasing the spatial resolution of the generated images. All existing layers remain trainable throughout the process. Here $N \times N$ 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: [9]

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: [9]

Outline

- Motivation
- GANs
  - Overview
  - Minimax Loss
  - Properties
  - Architectures
  - Challenges of GANs
- GANs Variants
  - Wasserstein GANs
  - Progressive GANs
  - **Cycle GANs**

Cycle GANs

Cycle-Consistent Generative Adversarial Networks [10] learn the image-to-image translation without a training set of aligned image pairs

Image Credit: [12]

Cycle GANs

Cycle-Consistent Generative Adversarial Networks [10] learn the image-to-image translation without a training set of aligned image pairs

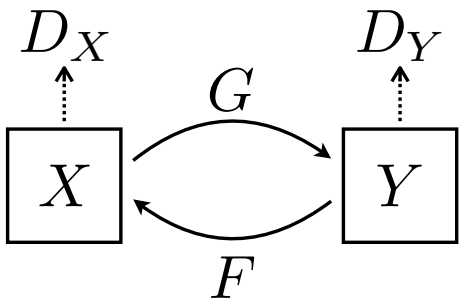
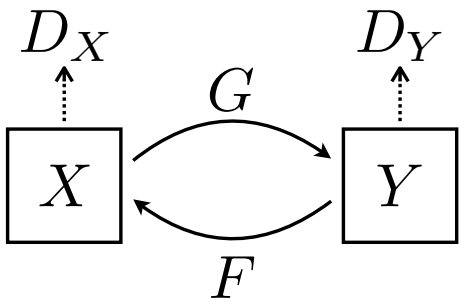


Image Credit: [10,12]

Cycle GANs

Cycle-Consistent Generative Adversarial Networks [10] learn the image-to-image translation without a training set of aligned image pairs

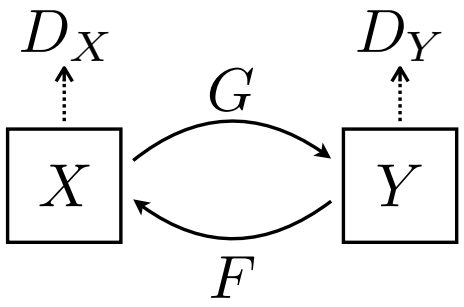


$$\mathcal{L}_{\text{GAN}}(F, D_X, X, Y) = \mathbb{E}_{x \sim p_{\text{data}}(x)}[\log D_X(x)] + \mathbb{E}_{y \sim p_{\text{data}}(y)}[\log(1 - D_X(F(y))]$$

Image Credit: [10,12]

Cycle GANs

Cycle-Consistent Generative Adversarial Networks [10] learn the image-to-image translation without a training set of aligned image pairs



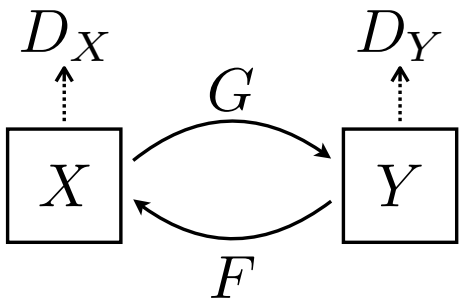
$$\mathcal{L}_{\text{GAN}}(F, D_X, X, Y) = \mathbb{E}_{x \sim p_{\text{data}}(x)}[\log D_X(x)] + \mathbb{E}_{y \sim p_{\text{data}}(y)}[\log(1 - D_X(F(y))]$$

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

Image Credit: [10,12]

Cycle GANs

Cycle-Consistent Generative Adversarial Networks [10] learn the image-to-image translation without a training set of aligned image pairs



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

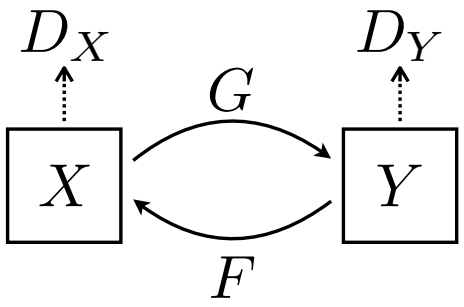
$$\mathcal{L}_{\text{GAN}}(G, D_Y, X, Y) = \mathbb{E}_{y \sim p_{\text{data}}(y)}[\log D_Y(y)] + \mathbb{E}_{x \sim p_{\text{data}}(x)}[\log(1 - D_Y(G(x))]$$

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

Image Credit: [10,12]

Cycle GANs

Cycle-Consistent Generative Adversarial Networks [10] 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: [10,12]

Cycle GANs

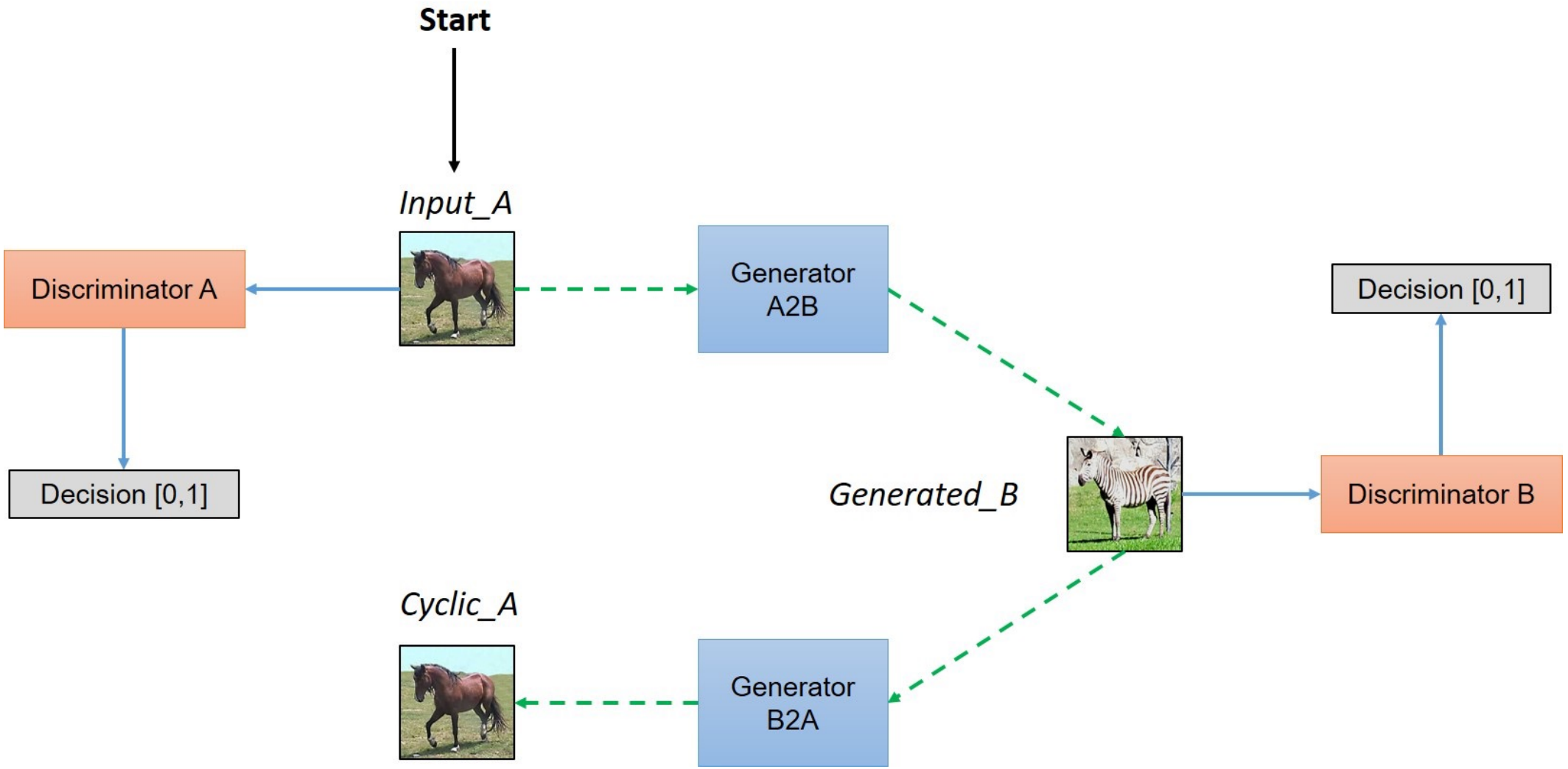


Image Credit: [9]

Cycle GANs

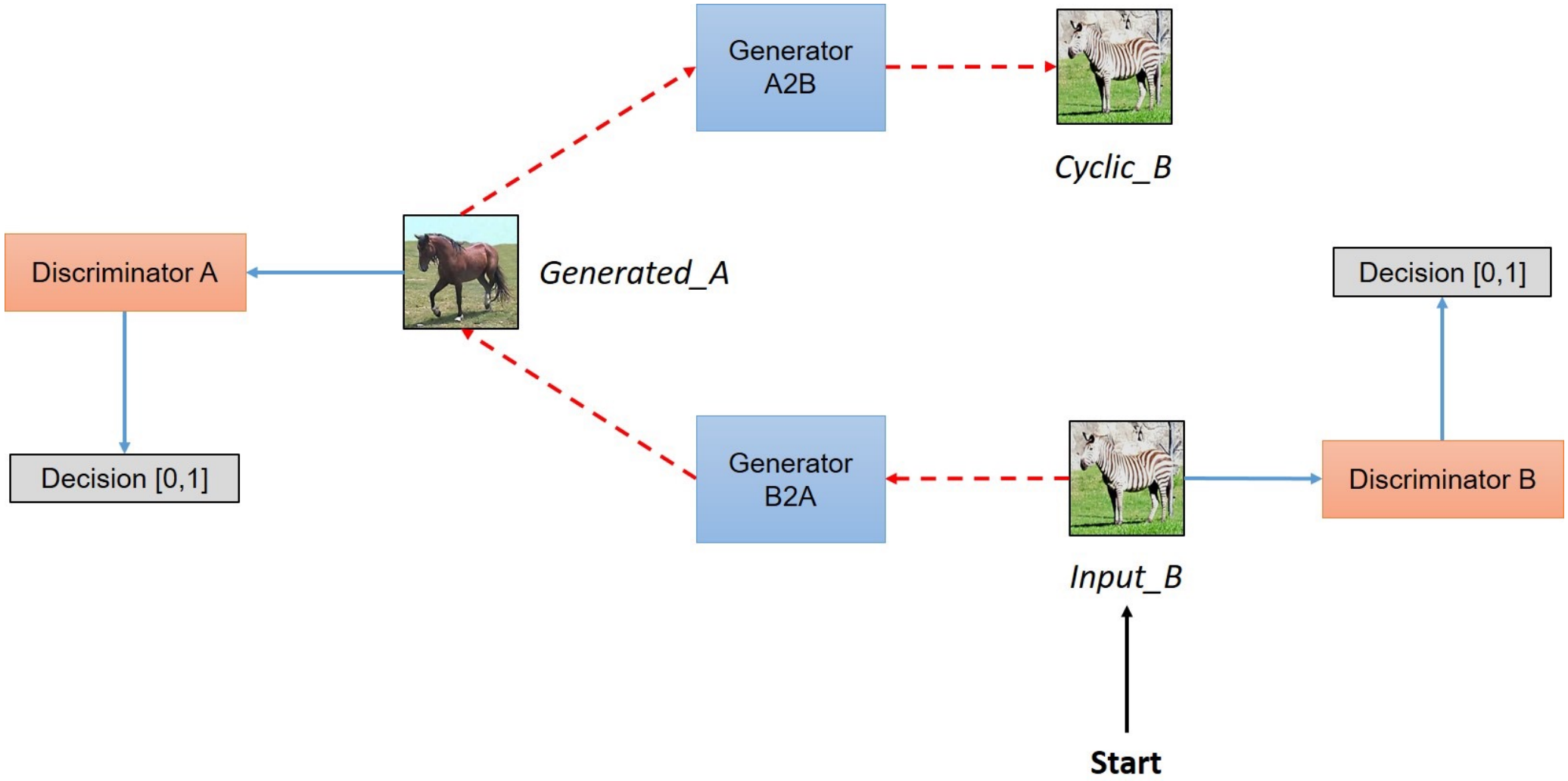


Image Credit: [9]

Cycle GANs

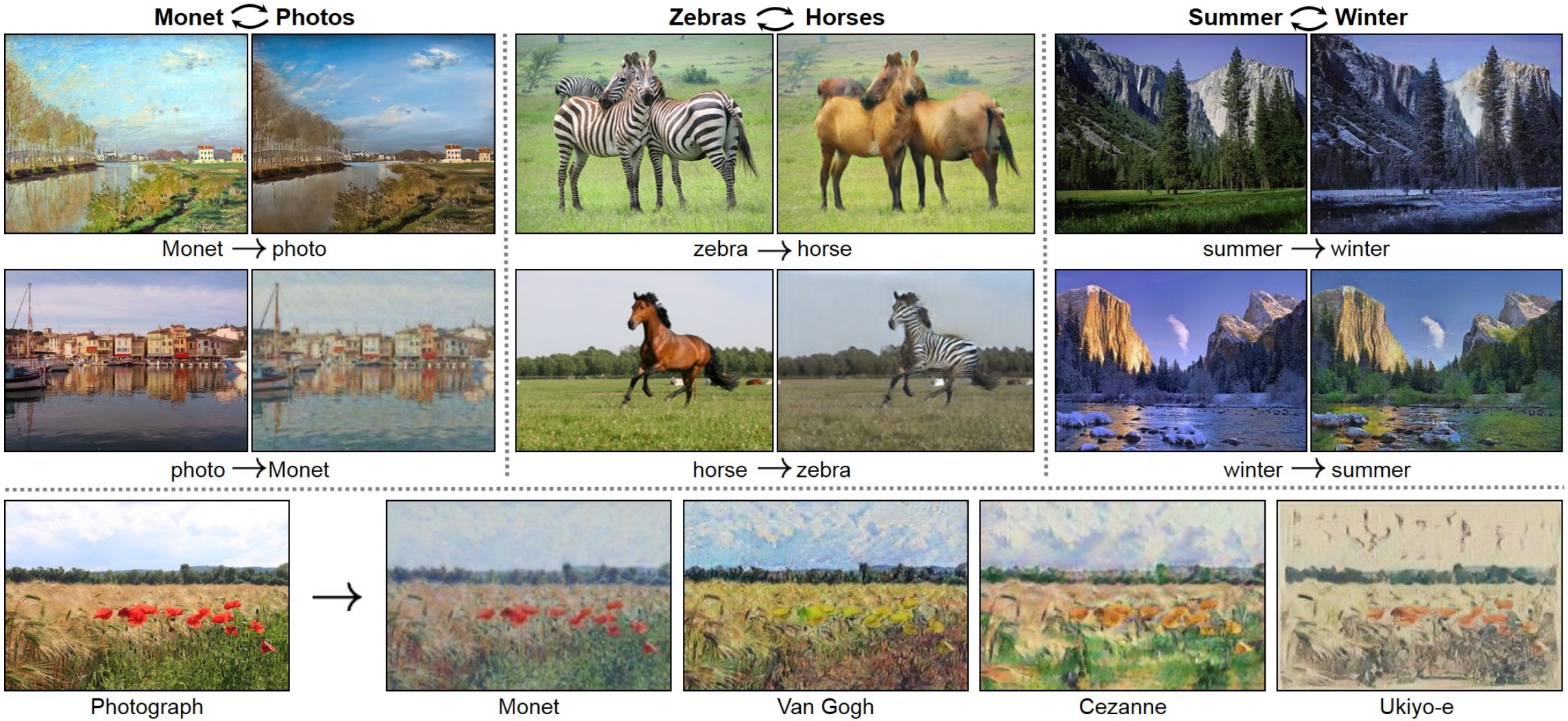


Image Credit: [12]

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

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

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

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

[7] 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.

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

[9] 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.

[10] 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).

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

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

Questions?