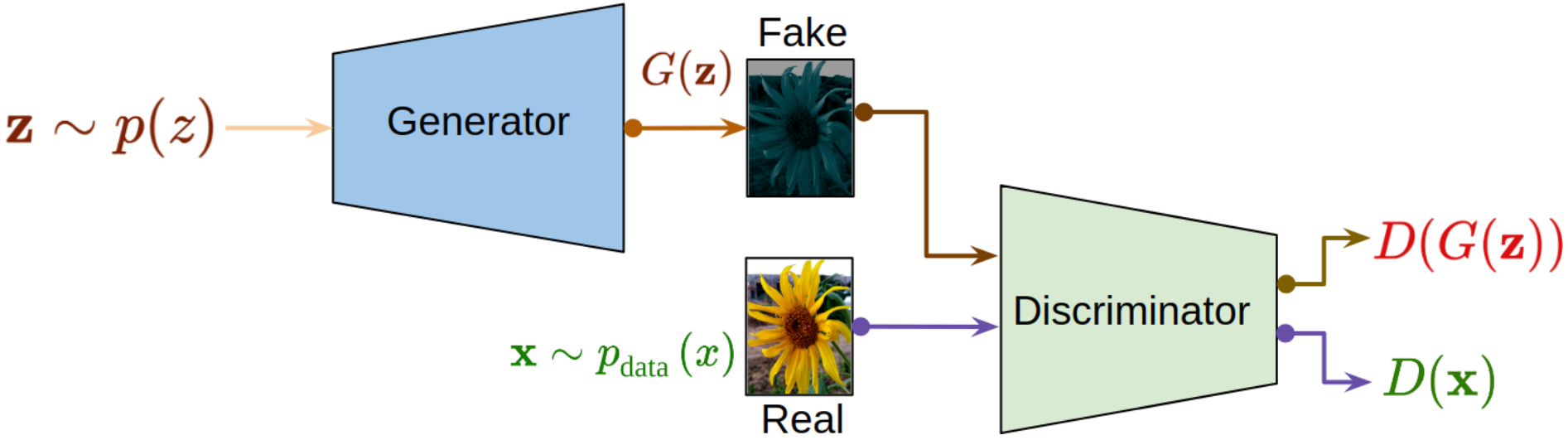
Machine Learning II: Fractal 3

Rajendra Nagar

Assistant Professor
Department of Electircal Engineering
Indian Institute of Technology Jodhpur
http://home.iitj.ac.in/~rn/

November 7, 2021

Generative Adversarial Networks: A Minimax Game



- Discriminator wants $D(\mathbf{x}) = 1$ for real samples.
- Discriminator wants $D(G(\mathbf{z})) = 0$ for fake samples.
- Generator wants $D(G(\mathbf{z})) = 1$ for fake samples.

$$\min_G \max_D \Big(\underbrace{\mathbb{E}_{\mathbf{x} \sim p_{\text{data}}} [\log D(\mathbf{x})]}_{\text{D wants } D(\mathbf{x}) = 1 \text{ for real point}} + \overbrace{\mathbb{E}_{\mathbf{z} \sim p(z)} [\log(1 - D(G(\mathbf{z})))]}^{\text{G wants } D(G(\mathbf{z})) = 1 \text{ for fake point}}_{\text{D wants } D(G(\mathbf{z})) = 0 \text{ for fake point}} \Big)$$

Generative Adversarial Networks: Optimality Analysis

Jointly train generator G and discriminator D with a minimax game.

$$\min_G \max_D \left(\mathbb{E}_{\mathbf{x}\sim p_{\text{data}}}[\log D(\mathbf{x})] + \mathbb{E}_{\mathbf{z}\sim p(z)}[\log(1 - D(G(\mathbf{z})))]\right)$$

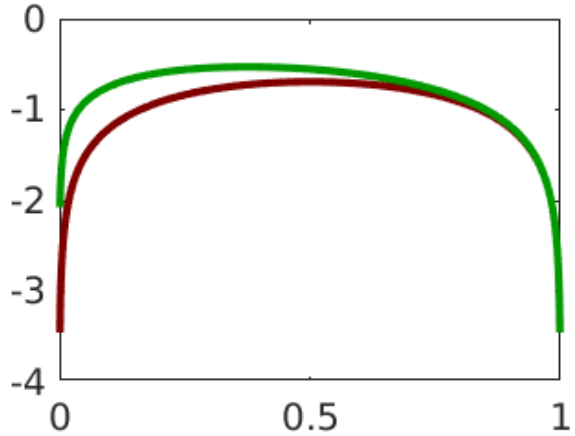
We want to verify that global minimum of this game occurs at $p_G = p_{\text{data}}$.

$$\begin{aligned}
& \min_G \max_D \left(\mathbb{E}_{\mathbf{x}\sim p_{\text{data}}}[\log D(\mathbf{x})] + \mathbb{E}_{\mathbf{z}\sim p(z)}[\log(1 - D(G(\mathbf{z})))]\right) \\
= \; & \min_G \max_D \left(\mathbb{E}_{\mathbf{x}\sim p_{\text{data}}}[\log D(\mathbf{x})] + \mathbb{E}_{\mathbf{x}\sim p_G}[\log(1 - D(\mathbf{x}))]\right) \\
= \; & \min_G \max_D \int_{\mathbf{x}} \left(p_{\text{data}}(\mathbf{x})[\log D(\mathbf{x})] + p_G(\mathbf{x})[\log(1 - D(\mathbf{x}))]\right) d\mathbf{x} \\
= \; & \min_G \int_{\mathbf{x}} \max_D \left(p_{\text{data}}(\mathbf{x})[\log D(\mathbf{x})] + p_G(\mathbf{x})[\log(1 - D(\mathbf{x}))]\right) d\mathbf{x}
\end{aligned}$$

Generative Adversarial Networks: Optimality Analysis

Consider the problem of finding the point of maximum of the function $f(t) = a\log(t) + b\log(1-t)$. Here $t, a, b \in [0, 1]$.

$$\begin{aligned} f(t) &= a\log(t) + b\log(1-t) \\ \frac{df}{dt} &= \frac{a}{t} - \frac{b}{1-t} = 0 \\ t &= \frac{a}{a+b}. \end{aligned}$$



$a = \frac{1}{2}, b = \frac{1}{2}$, $a = \frac{3}{10}, b = \frac{1}{2}$

Now, consider the main problem :

$$\max_D \left(p_{\text{data}}(\mathbf{x})[\log D(\mathbf{x})] + p_G(\mathbf{x})[\log(1 - D(\mathbf{x}))]\right)$$

The optimal discriminator $D^\star(\mathbf{x}) = \frac{p_{\text{data}}(\mathbf{x})}{p_{\text{data}}(\mathbf{x}) + p_G(\mathbf{x})}$

Generative Adversarial Networks

Let us now try to find the optimal generator given the optimal discriminator.

$$
\begin{aligned}
& \min_G \int_{\mathbf{x}} \max_D \left(p_{\text{data}}(\mathbf{x}) \log D(\mathbf{x}) + p_G(\mathbf{x}) \log(1 - D(\mathbf{x})) \right) d\mathbf{x} \\
= \; & \min_G \int_{\mathbf{x}} \left(p_{\text{data}}(\mathbf{x}) \log D^\star(\mathbf{x}) + p_G(\mathbf{x}) \log(1 - D^\star(\mathbf{x})) \right) d\mathbf{x} \\
= \; & \min_G \int_{\mathbf{x}} \left(p_{\text{data}} \log \left(\frac{p_{\text{data}}}{p_{\text{data}} + p_G} \right) + p_G \log \left(1 - \frac{p_{\text{data}}}{p_{\text{data}} + p_G} \right) \right) d\mathbf{x} \\
= \; & \min_G \int_{\mathbf{x}} \left(p_{\text{data}} \log \left(\frac{p_{\text{data}}}{p_{\text{data}} + p_G} \right) + p_G \log \left(\frac{p_G}{p_{\text{data}} + p_G} \right) \right) d\mathbf{x} \\
= \; & \min_G \int_{\mathbf{x}} \left(p_{\text{data}} \log \left(\frac{2}{2} \frac{p_{\text{data}}}{p_{\text{data}} + p_G} \right) + p_G \log \left(\frac{2}{2} \frac{p_G}{p_{\text{data}} + p_G} \right) \right) d\mathbf{x} \\
= \; & \min_G \int_{\mathbf{x}} \left(p_{\text{data}} \log \left(\frac{2p_{\text{data}}}{p_{\text{data}} + p_G} \right) + p_G \log \left(\frac{2p_G}{p_{\text{data}} + p_G} \right) \right) d\mathbf{x} - \log 4
\end{aligned}
$$

Let us now find the optimal generator given the optimal discriminator.

$$\min_G \int_{\mathbf{x}} \max_D \left(p_{\text{data}}(\mathbf{x}) \log D(\mathbf{x}) + p_G(\mathbf{x}) \log(1 - D(\mathbf{x}))\right) d\mathbf{x}$$

$$= \min_G \int_{\mathbf{x}} \left(p_{\text{data}} \log \left(\frac{p_{\text{data}}}{\frac{p_{\text{data}}+p_G}{2}} \right) + p_G \log \left(\frac{p_G}{\frac{p_{\text{data}}+p_G}{2}} \right) \right) d\mathbf{x}$$

$$= \min_G \left(\mathsf{D}_{\mathsf{KL}} \left(p_{\text{data}} \| \frac{p_{\text{data}} + p_G}{2} \right) + \mathsf{D}_{\mathsf{KL}} \left(p_G \| \frac{p_{\text{data}} + p_G}{2} \right) \right)$$

Jensen-Shannon Divergence: Between two distributions f and g is symmetric and defined as

$$\mathsf{D}_{JS}(f\|g) = \frac{1}{2}\mathsf{D}_{\mathsf{KL}} \left(f \| \frac{f+g}{2} \right) + \frac{1}{2}\mathsf{D}_{\mathsf{KL}} \left(g \| \frac{f+g}{2} \right)$$

The $\mathsf{D}_{JS}(f\|g)$ is nonnegative and zero only if both the distributions are equal that is $f = g$. Therefore, the optimal generator $p_G = p_{\text{data}}$.

Generative Adversarial Networks

Jointly train generator G and discriminator D with a minimax game.

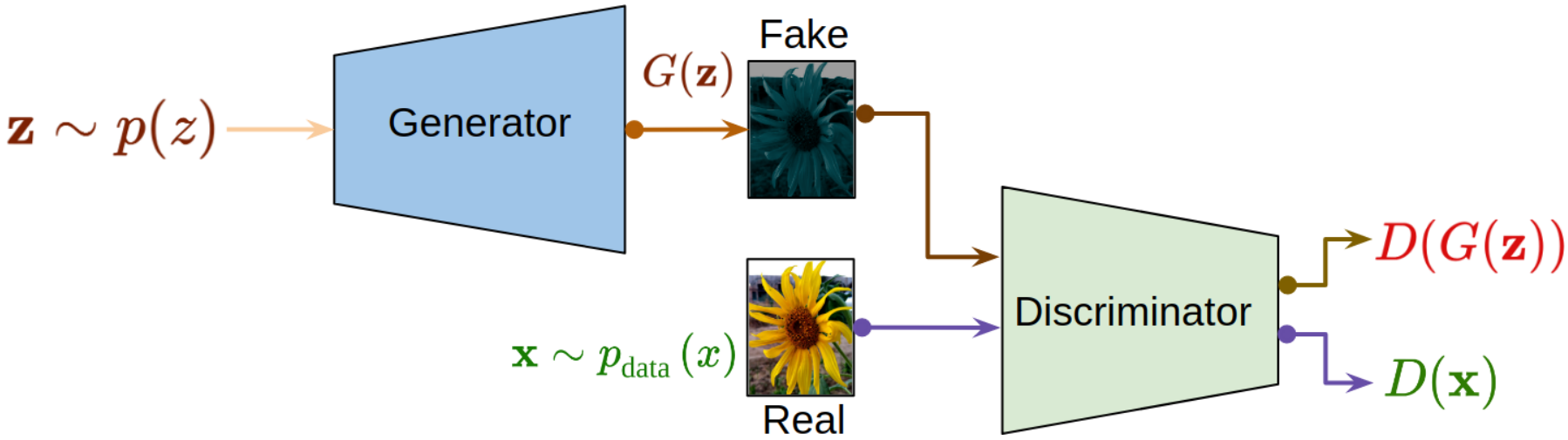
$$\min_G \max_D \left(\mathbb{E}_{\mathbf{x}\sim p_{\text{data}}}[\log D(\mathbf{x})] + \mathbb{E}_{\mathbf{z}\sim p(z)}[\log(1 - D(G(\mathbf{z})))]\right)$$

Claim: The global minimum of this game occurs at $p_G = p_{\text{data}}$.
Result: The global minimum of the minimax occurs, if

- $D^\star(\mathbf{x}) = \frac{p_{\text{data}}(\mathbf{x})}{p_{\text{data}}(\mathbf{x}) + p_G(\mathbf{x})}$
- $p_G(\mathbf{x}) = p_{\text{data}}(\mathbf{x})$

Generative Adversarial Networks

Problem Statement: Given a dataset $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ of samples drawn from $p_{\text{data}}(\mathbf{x})$, draw new samples from p_{data} with explicitly modeling it.



G wants $D(G(\mathbf{z})) = 1$ for fake point

$$\min_G \max_D \left(\mathbb{E}_{\mathbf{x} \sim p_{\text{data}}} [\log D(\mathbf{x})] + \mathbb{E}_{\mathbf{z} \sim p(z)} [\log(1 - D(G(\mathbf{z})))] \right)$$

D wants $D(\mathbf{x}) = 1$ for real point

D wants $D(G(\mathbf{z})) = 0$ for fake point

The global minimum of the minimax occurs, if $D^\star(\mathbf{x}) = \frac{p_{\text{data}}(\mathbf{x})}{p_{\text{data}}(\mathbf{x}) + p_G(\mathbf{x})}$ and $p_G = p_{\text{data}}$.

Algorithm 1 GAN Training

1: A dataset $\mathcal{S} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$.

2: Sample a batch of m training points $\{\mathbf{x}_{i_1}, \mathbf{x}_{i_2}, \ldots, \mathbf{x}_{i_m}\}$ from $\mathcal{S}$.

3: Draw a batch of m latent vectors $\{\mathbf{z}_{i_1}, \mathbf{z}_{i_2}, \ldots, \mathbf{z}_{i_m}\}$ from $p_{\mathbf{z}}$.

4: Update the parameters $\boldsymbol{\theta}_g$ of the generator $G_{\boldsymbol{\theta}_g}$ by SGD

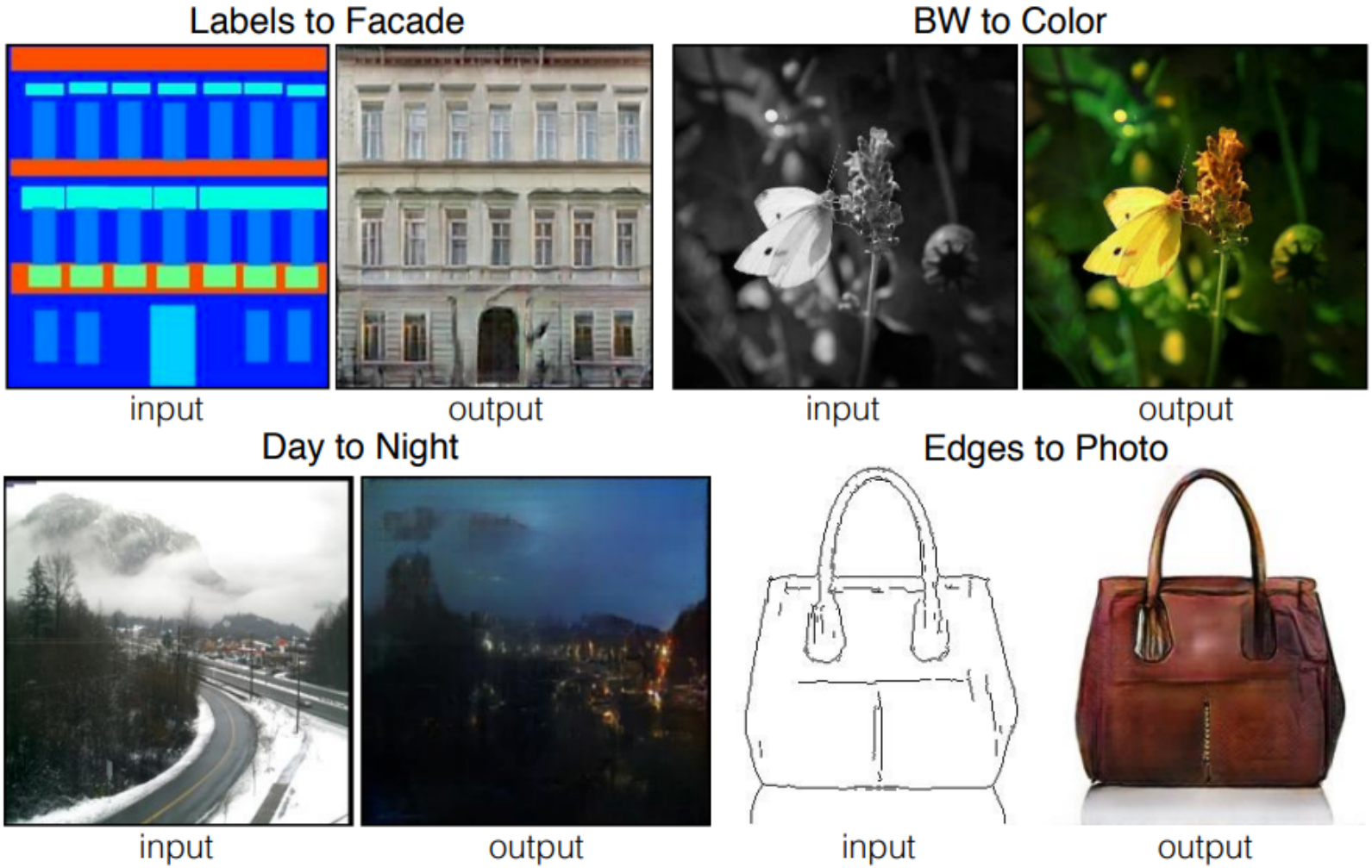
$$\begin{aligned} \nabla_{\boldsymbol{\theta}_g} V(\boldsymbol{\theta}_g, \boldsymbol{\theta}_d) &= \frac{1}{m} \nabla_{\boldsymbol{\theta}_g} \sum_{k=i_1}^{i_m} \log(1 - D_{\boldsymbol{\theta}_d}(G_{\boldsymbol{\theta}_g}(\mathbf{z}_k))) \\ \boldsymbol{\theta}_g &\leftarrow \boldsymbol{\theta}_g - \alpha_g \nabla_{\boldsymbol{\theta}_g} \end{aligned}$$

5: Update the parameters $\boldsymbol{\theta}_d$ of the discriminator $D_{\boldsymbol{\theta}_d}$ by SGD

$$\begin{aligned} \nabla_{\boldsymbol{\theta}_d} V(\boldsymbol{\theta}_g, \boldsymbol{\theta}_d) &= \frac{1}{m} \nabla_{\boldsymbol{\theta}_d} \sum_{k=i_1}^{i_m} \log(D_{\boldsymbol{\theta}_d}(\mathbf{x}_k)) + \log(1 - D_{\boldsymbol{\theta}_d}(G_{\boldsymbol{\theta}_g}(\mathbf{z}_k))) \\ \boldsymbol{\theta}_d &\leftarrow \boldsymbol{\theta}_d + \alpha_d \nabla_{\boldsymbol{\theta}_d} \end{aligned}$$

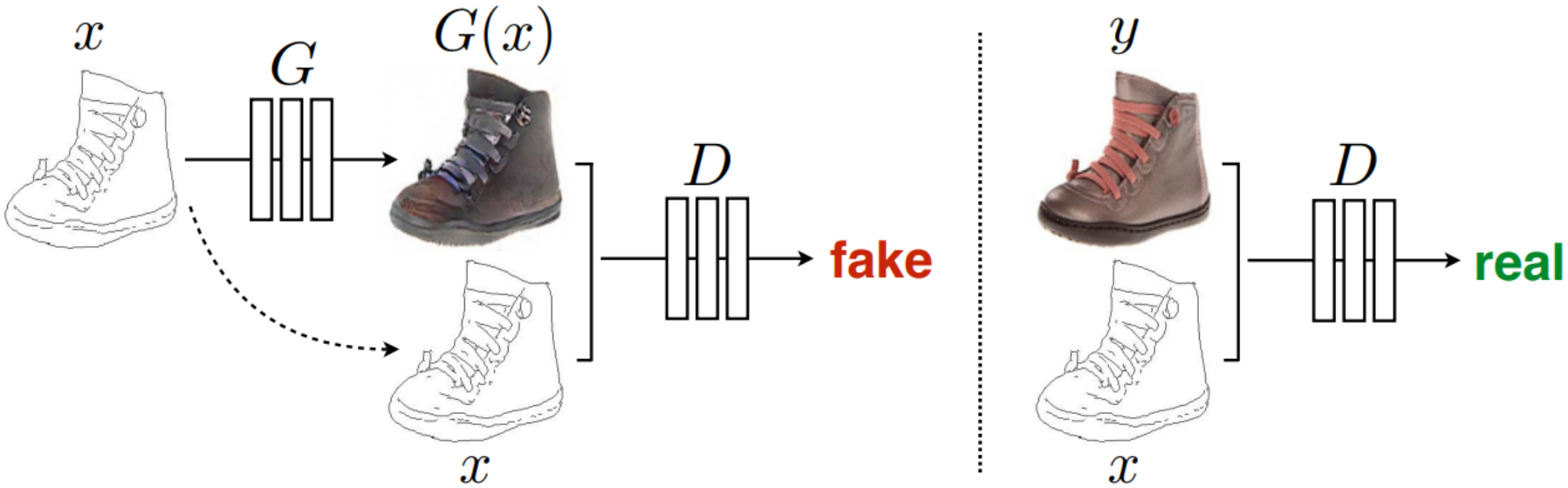
Conditional-GANs [Isola et al. 2017]

Image-to-image translation: We are given a set of paired images from two domains, X and Y. Can we translate from $X \rightarrow Y$.



Conditional-GANs [Isola et al. 2017]

Image-to-image translation: We are given paired images from two domains, X and Y. Can we translate from $X \rightarrow Y$.

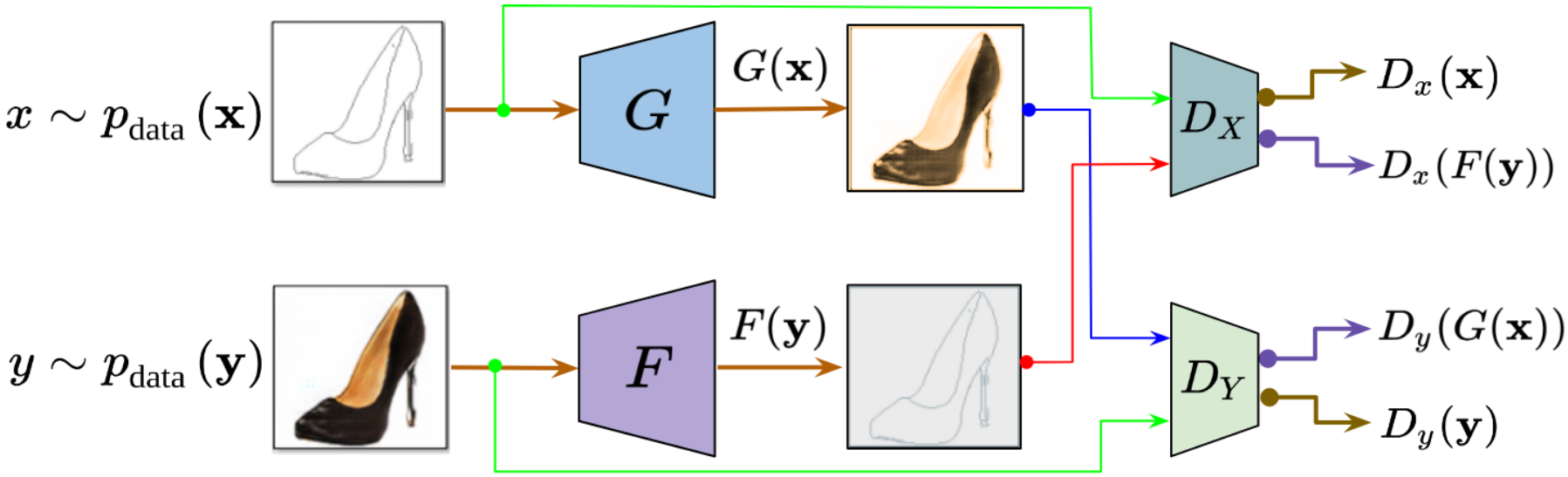


cGAN Losses

$$
\begin{aligned}
\ell_{\mathsf{GAN}}(G, D) &= \mathbb{E}_{\mathbf{x},\mathbf{y}}[\log D(\mathbf{x}, \mathbf{y})] + \mathbb{E}_{\mathbf{x}}[\log(1 - D(\mathbf{x}, G(\mathbf{x})))] \\
\ell_{\mathsf{c}}(G) &= \mathbb{E}_{\mathbf{x},\mathbf{y}}[\|\mathbf{y} - G(\mathbf{x}, \mathbf{z})\|_1] \\
\min_G \max_D & \quad \ell_{\mathsf{GAN}}(G, D) + \lambda \ell_{\mathsf{c}}(G)
\end{aligned}
$$

Cycle-GANs [Zhu et al. 2020]

Image-to-image translation: We are given images from two domains, X and Y. Can we translate from $X \leftrightarrow Y$ in an unsupervised manner?



Adversarial Losses

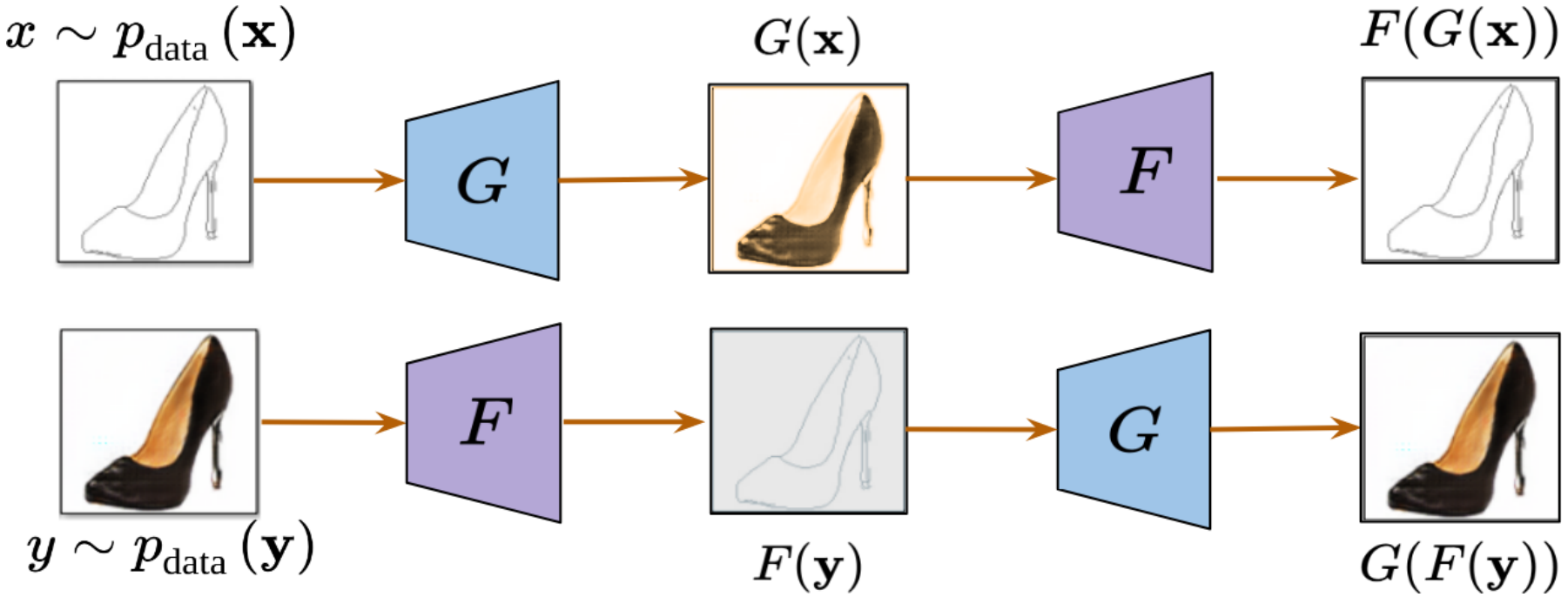
$$\min_F \max_{D_x} \quad \ell_{\text{GAN}}(F, D_x)$$

$$\ell_{\text{GAN}}(F, D_x) = \mathbb{E}_{\mathbf{x}}[\log D_x(\mathbf{x})] + \mathbb{E}_{\mathbf{y}}[\log(1 - D_x(F(\mathbf{y})))]$$

$$\min_G \max_{D_y} \quad \ell_{\text{GAN}}(G, D_y)$$

$$\ell_{\text{GAN}}(G, D_y) = \mathbb{E}_{\mathbf{y}}[\log D_y(\mathbf{y})] + \mathbb{E}_{\mathbf{x}}[\log(1 - D_y(G(\mathbf{x})))]$$

Cycle-GANs: Cycle Consistency



Cycle Consistency Losses

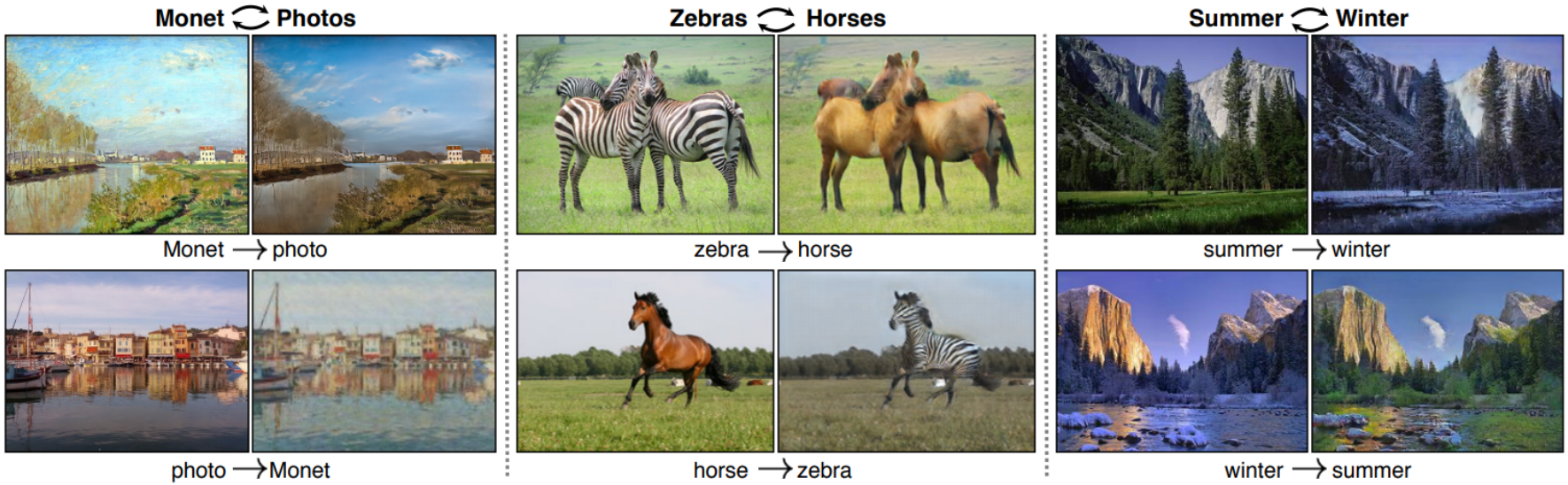
Learn the generators so that we can transform $X \leftrightarrow Y$ that is $\mathbf{x} = F(G(\mathbf{x}))$ and $\mathbf{y} = G(F(\mathbf{y}))$.

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

Cycle-GANs

Total Loss

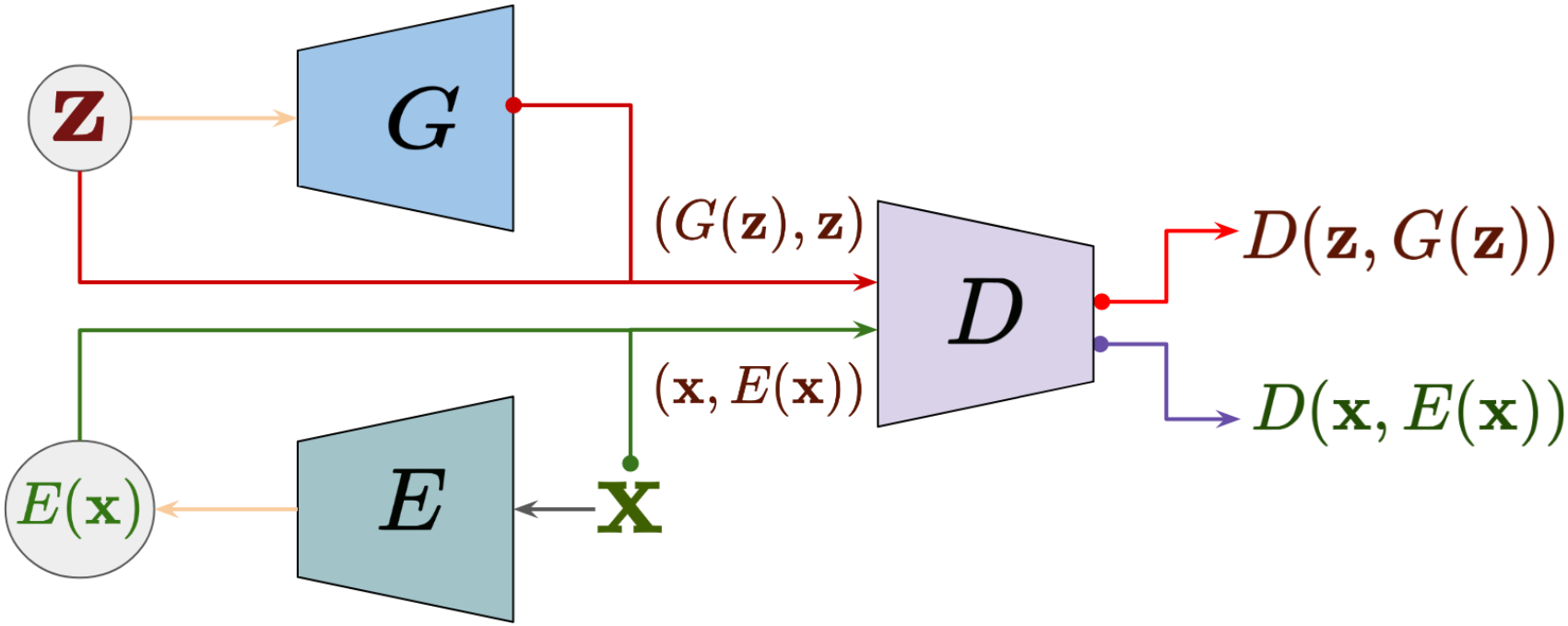
$$\ell(F, G, D_x, D_y) = \ell_{\text{GAN}}(F, D_x) + \ell_{\text{GAN}}(G, D_y) + \lambda \ell_{\text{cyc}}(F, G).$$



Understanding Latent Variables

- In variational auto-encoders, we can semantically control the latent vectors to generate new samples using the decoder.
- We can also use the encoder to find a semantic representation of an image from the learned distribution.
- These latent vectors can be used for feature representation of objects that can further be used for object classification.
- In GANs, given a latent vector $\mathbf{z}$, we can find the generated image using the the generator as $G(\mathbf{z})$.
- However, given a new point from the learned distribution, can we map it to the corresponding latent representation $\mathbf{z}$?
- In general, the generator is invertible.

Bidirectional GAN (BiGAN) [Donahue et al. 2017]



BiGAN Loss

$$\min_{G,E} \max_{D} \left(\mathbb{E}_{\mathbf{x}}[\log D(\mathbf{x}, E(\mathbf{x}))] + \mathbb{E}_{\mathbf{z}}[\log(1 - D(G(\mathbf{x}), \mathbf{z}))]\right)$$

Claim: In order to "fool" a perfect discriminator D, BiGAN encoder E and generator G must invert each other. That is, $G(E(\mathbf{x})) = \mathbf{x}$ and $E(G(\mathbf{z})) = \mathbf{z}$.

Bidirectional GAN (BiGAN) [Donahue et al. 2017]

StyleGAN2 [Karras et al. 2020]

StyleGAN2 [Karras et al. 2020]