Learning summary of GANS model

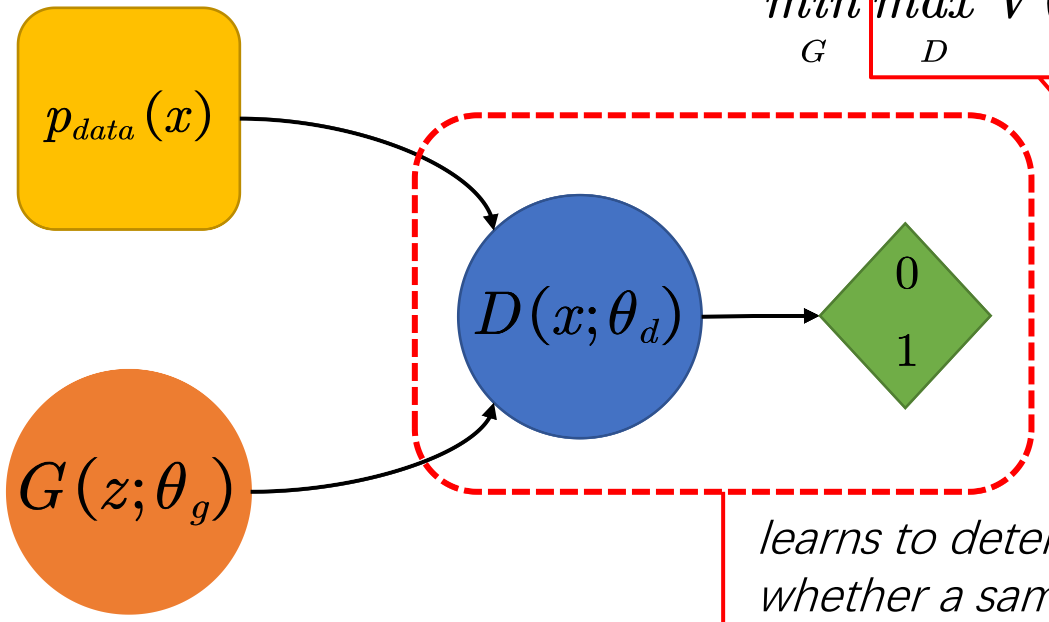
Presented by Wang, Xingrui

Outline

- Original GAN
- DCGAN
- CGAN
- WGAN
- Cycle GAN

Original GAN

$$\min_{G} \boxed{\max_{D} V(D,G)} = E_{x\sim p_{data}(x)}[logD(x)] + E_{z\sim p_z(z)}[log(1-D(G(z)))]$$



Given G, what is the optimal D^* maxmizing

$$\max_{D} V(D,G) = E_{x\sim p_{data}}[\log D(x)] + E_{x\sim p_G}\log[1-D(x)]$$

$$V = \int [P_{data}(x)\log D(x) + P_G(x)\log(1-D(x))]\, dx$$

$$D^*(x) = \frac{P_{data}(x)}{P_{data}(x)+P_G(x)}$$

$$\max_{D} V(G,D) = V(G,D^*)$$

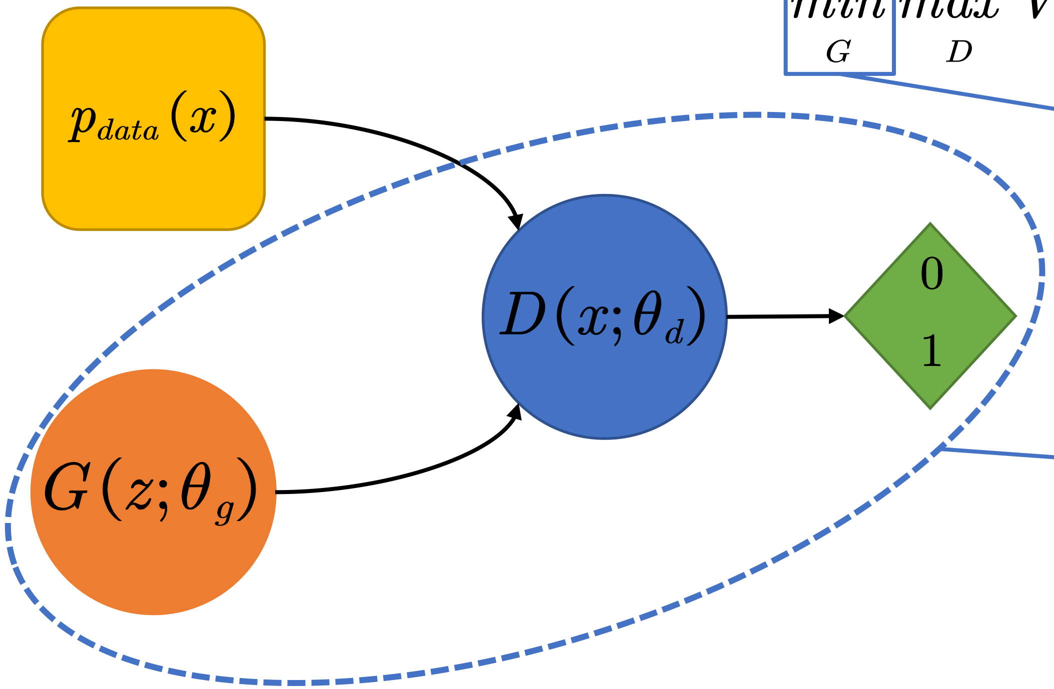
learns to determine whether a sample is from the model distribution or the data distribution

$$V(G,D^*) = -2\log 2 + 2JSD(P_{data}||P_G)$$

Goodfellow I, Pouget-Abadie J, Mirza M, et al. Generative adversarial networks[J]. Communications of the ACM, 2020, 63(11): 139-144.

Original GAN

$$\boxed{\underset{G}{min}}\ \underset{D}{max}\ V(D,G) = E_{x \sim p_{data}(x)}[logD(x)] + E_{z \sim p_z(z)}[log(1 - D(G(z)))]$$



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

$$G^* = arg\min_G L(G)$$

$$\theta_g \leftarrow \theta_g - \eta \frac{\partial L(G)}{\partial \theta_g}$$

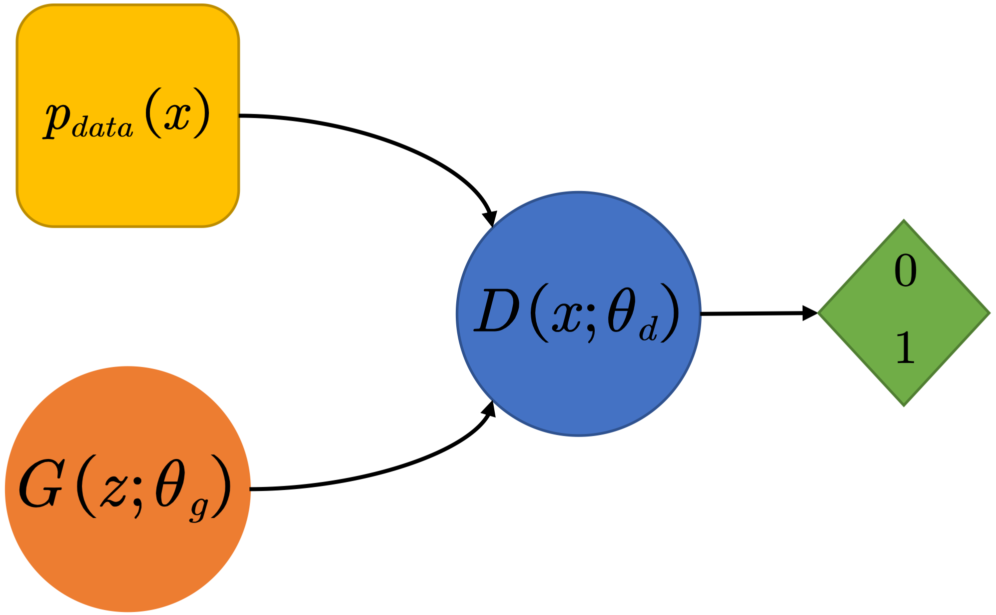
- In each training iteration:
  - ***Step1***: *Fix generator G, and update discriminator D*
  - ***Step2***: *Fix discriminator D, and update generator G*

$$V = \cancel{E_{x \sim p_{data}}[\log D(x)]} + E_{x \sim p_G} \log\,[1 - D(x)] \rightarrow MMGAN$$

$$V = \cancel{E_{x \sim p_{data}}[\log D(x)]} + E_{x \sim p_G} \log\,[-D(x)] \rightarrow NSGAN$$

Goodfellow I, Pouget-Abadie J, Mirza M, et al. Generative adversarial networks[J]. Communications of the ACM, 2020, 63(11): 139-144.

Original GAN

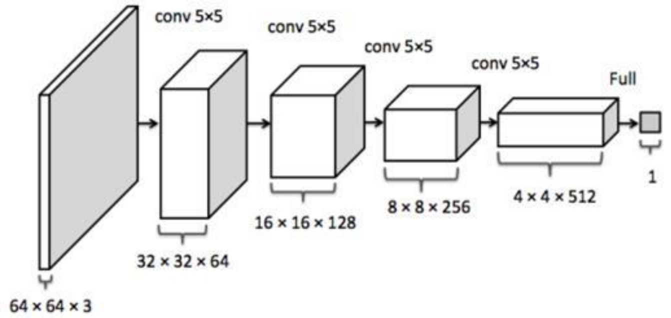


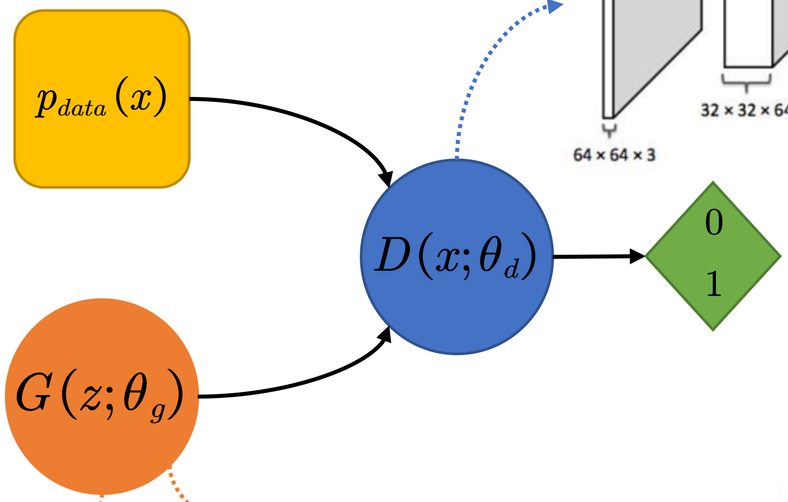
Sample noise as generator input

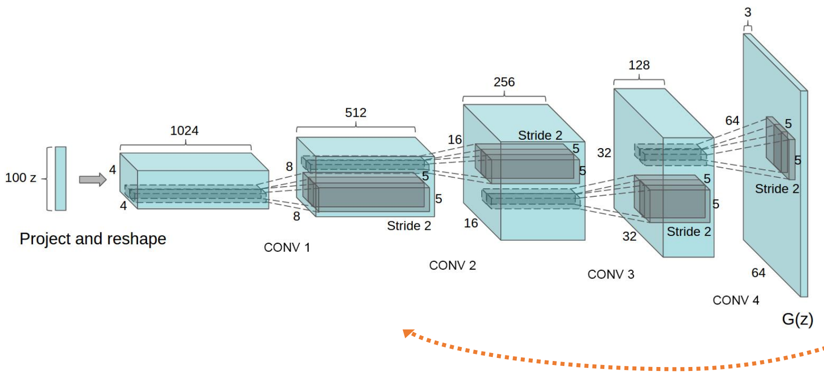
Goodfellow I, Pouget-Abadie J, Mirza M, et al. Generative adversarial networks[J]. Communications of the ACM, 2020, 63(11): 139-144.

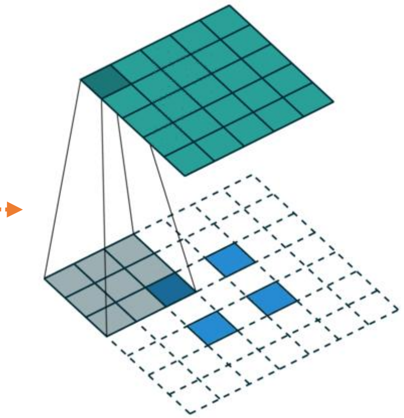
DCGAN

- Convolutions + GANs = Good for generating images
- Deconvolution layer





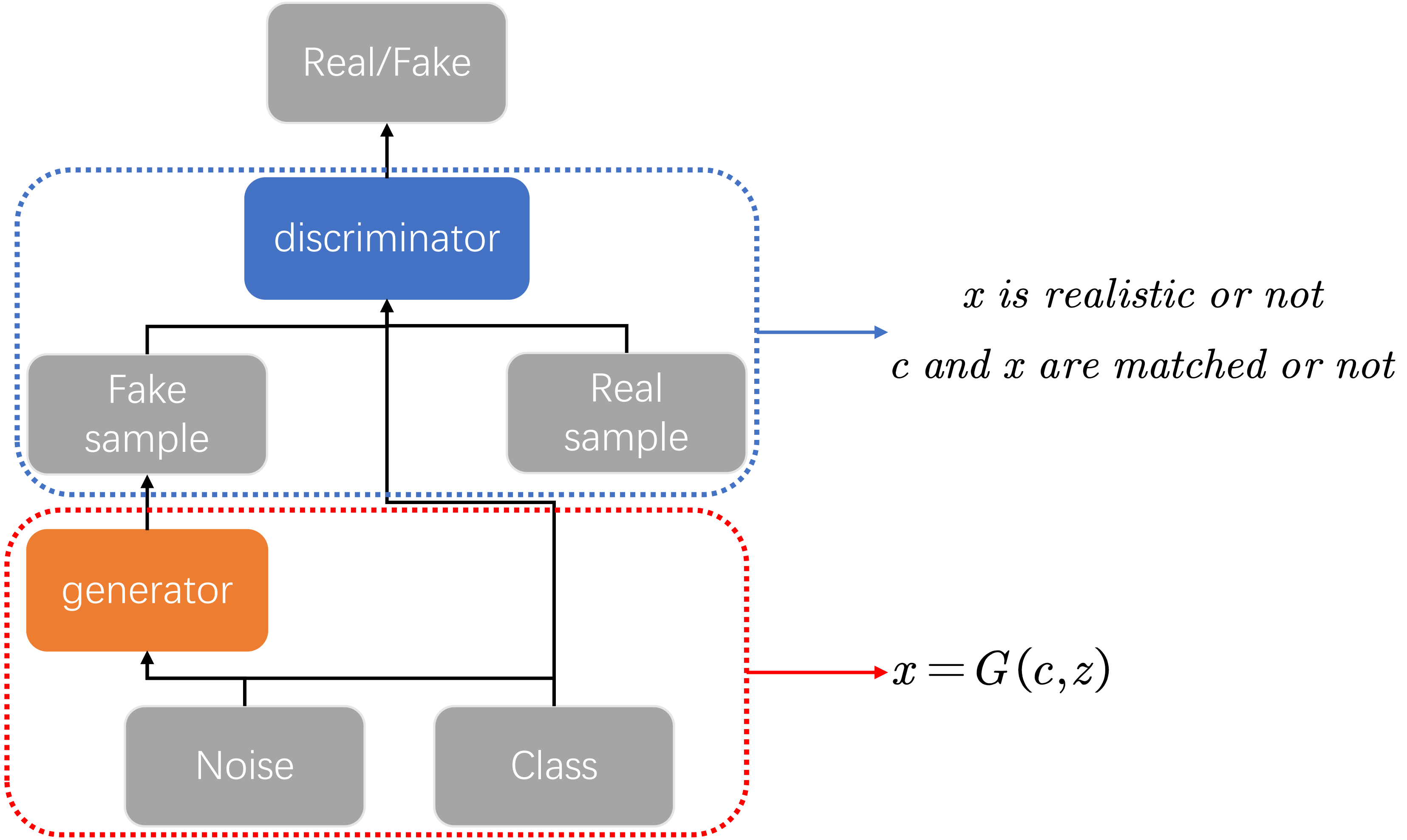




Radford A, Metz L, Chintala S. Unsupervised representation learning with deep convolutional generative adversarial networks[J]. arXiv preprint arXiv:1511.06434, 2015.

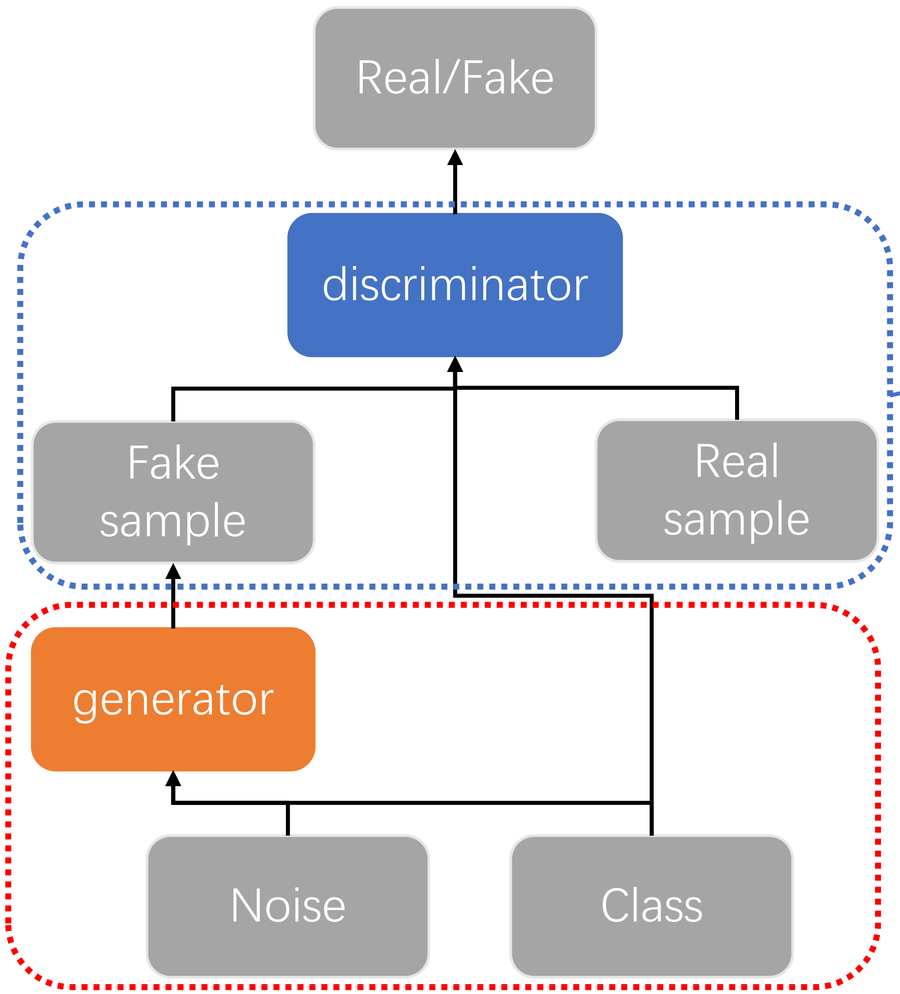
CGAN

- CGAN can directly generate specific types of outputs
- Feed the class labels to the discriminator and generator along with the image and seed, respectively



Mirza M, Osindero S. Conditional generative adversarial nets[J]. arXiv preprint arXiv:1411.1784, 2014.

CGAN



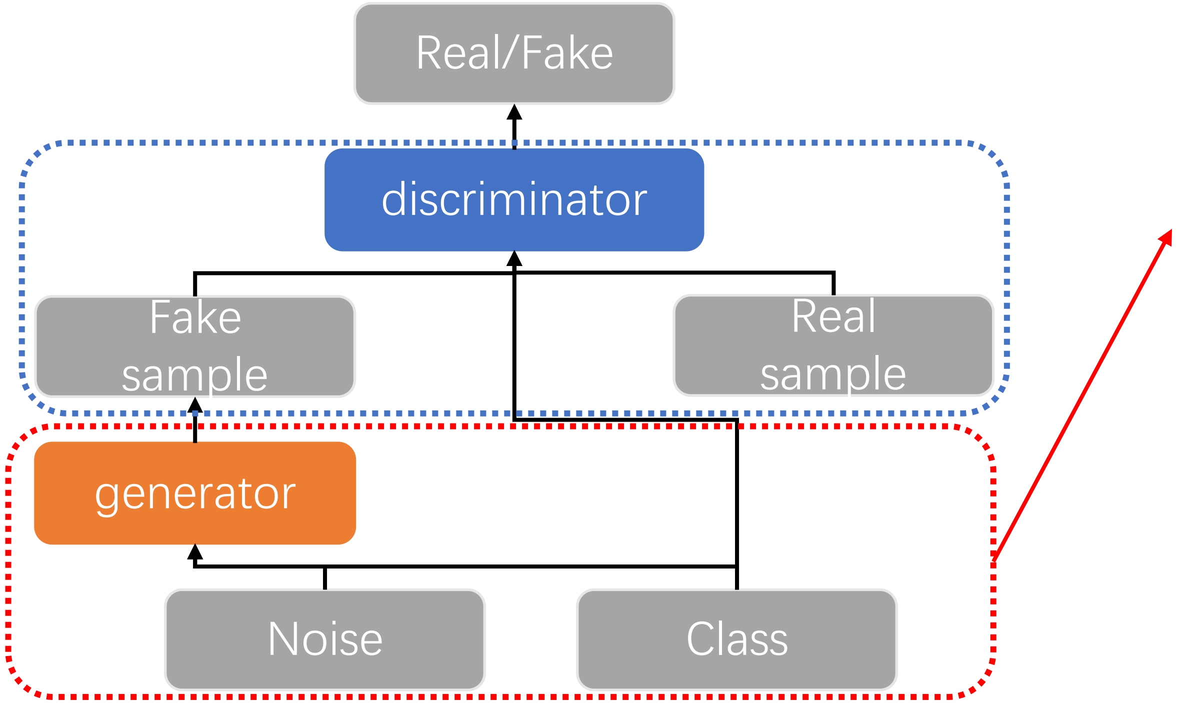
- In each training iteration:
  - Sample m positive examples $\{(c^1, x^1), (c^2, x^2), \ldots, (c^m, x^m)\}$ from database
  - Sample m noise samples $\{z^1, z^2, \ldots, z^m\}$ from a distribution
  - Sample m objects $\{\tilde{x}^1, \tilde{x}^2, \ldots, \tilde{x}^m\}$, $\tilde{x}^i = G(c^i, z^i)$
  - Sample m objects $\{\hat{x}^1, \hat{x}^2, \ldots, \hat{x}^m\}$ from database
  - Update discriminator parameter θ_d to maxmize

$$\tilde{V} = \frac{1}{m}\sum_{i=1}^{m} \underbrace{\log D(c^i, x^i)}_{\text{Correct conditions and real pictures}} + \frac{1}{m}\sum_{i=1}^{m} \log\left(1 - \underbrace{D(c^i, \tilde{x}^i)}_{\text{Correct conditions and fake pictures}}\right) + \frac{1}{m}\sum_{i=1}^{m} \log\left(1 - \underbrace{D(c^i, \hat{x}^i)}_{\text{Error conditions and real pictures}}\right)$$

$$\theta_d \leftarrow \theta_d + \eta \nabla \tilde{V}(\theta_d)$$

Mirza M, Osindero S. Conditional generative adversarial nets[J]. arXiv preprint arXiv:1411.1784, 2014.

CGAN

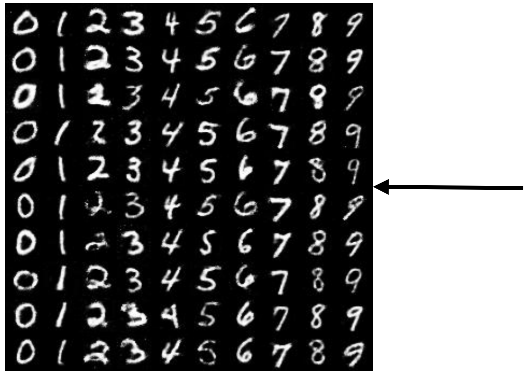


- Sample m noise samples $\{z^1, z^2, \ldots, z^m\}$ from a distribution
- Sample m conditions $\{c^1, c^2, \ldots, c^m\}$ from a database
- Update generator parameters θ_g to maxmize

$$\tilde{V} = \frac{1}{m}\sum_{i=1}^{m}\log\left(D(G(c^i, z^i))\right)$$

$$\theta_g \leftarrow \theta_g - \eta\nabla\tilde{V}(\theta_g)$$

$$\min_G \max_D V(D,G) = E_{x\sim p_{data}(x)}[logD(x|y)] + E_{z\sim p_z(z)}[log(1-D(G(z|y)))]$$



results

Mirza M, Osindero S. Conditional generative adversarial nets[J]. arXiv preprint arXiv:1411.1784, 2014.

WGAN

JS divergence is log2 if two distributions do not overlap.

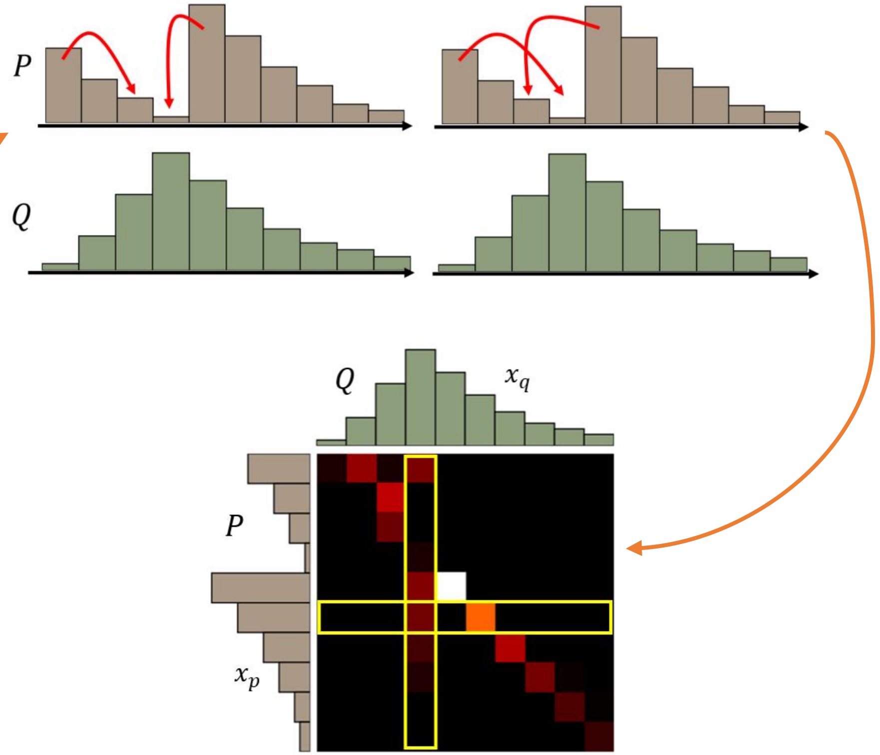
$$\forall x \in R, JSD(P||Q) = \log 2$$

Vanishing Gradient for Generator

Earth-Mover(EM) **distance**

$$W(\mathbb{P}_r, \mathbb{P}_g) = \inf_{\gamma \in \prod(\mathbb{P}_r, \mathbb{P}_g)} \mathbb{E}_{(x,y)\sim\gamma}[\|x - y\|]$$

$\gamma(x, y)$ indicates how much "mass" must be transported from x to y in order to transform the distributions $\mathbb{P}_r$ into the distribution $\mathbb{P}_g$. The EM distance then is the "cost" of the optimal transport plan.



Arjovsky M, Chintala S, Bottou L. Wasserstein generative adversarial networks[C]//International conference on machine learning. PMLR, 2017: 214-223.

WGAN

$$W(\mathbb{P}_r, \mathbb{P}_\theta) = \sup_{\|f\|_L \leqslant 1} \mathbb{E}_{x \sim \mathbb{P}_r}[f(x)] - \mathbb{E}_{x \sim \mathbb{P}_\theta}[f(x)]$$

$$|f(x_1) - f(x_2)| \leqslant K|x_1 - x_2|$$

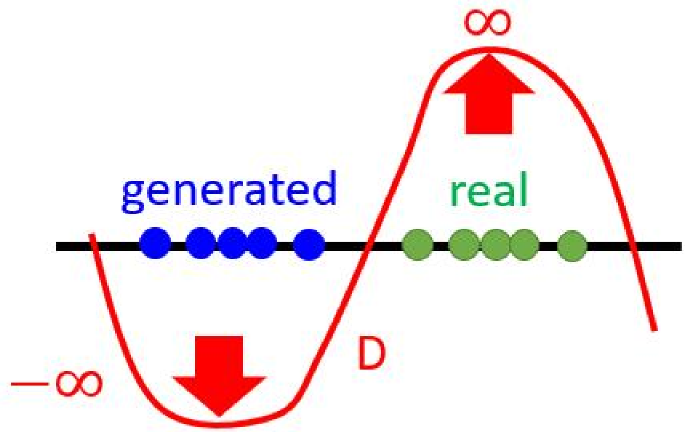
if we have a parameterized family of functions $\{f_w\}_{w \in \mathcal{W}}$ that are all K-Lipschitz for some K

$$\max_{w \in \mathcal{W}} \mathbb{E}_{x \sim \mathbb{P}_r}[f_w(x)] - \mathbb{E}_{z \sim p(z)}[f_w(g_\theta(z))]$$

$\mathbb{P}_{\mathrm{r}}$ will not keep going down, $\mathbb{P}_\theta$ will not keep going up

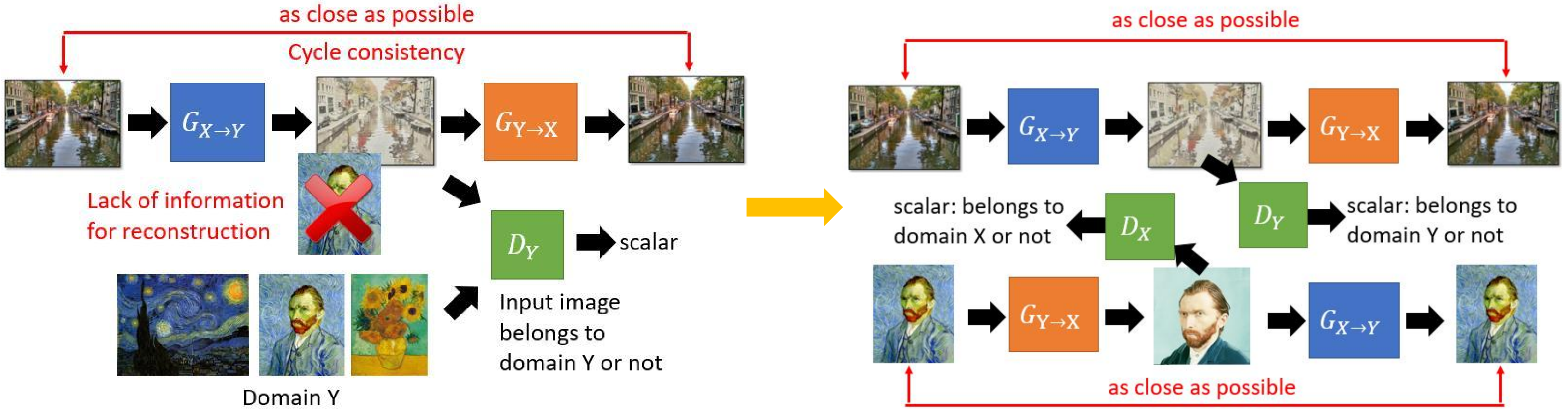
f is represented by a neural network with parameter w

Restrict all parameters w_i of the neural network f_w not to exceed a certain range $[-c, c]$



Arjovsky M, Chintala S, Bottou L. Wasserstein generative adversarial networks[C]//International conference on machine learning. PMLR, 2017: 214-223.

Cycle GAN



Make sure that the output of the generator and the input image have very similar properties

$$X\ domain \to G_1 \to Y\ domain \to G_2 \to X\ domain$$

$$Y\ domain \to G_2 \to X\ domain \to G_1 \to Y\ domain$$

Zhu J Y, Park T, Isola P, et al. Unpaired image-to-image translation using cycle-consistent adversarial networks[C]//Proceedings of the IEEE international conference on computer vision. 2017: 2223-2232.