Generative Adversarial Networks

Mark Chang

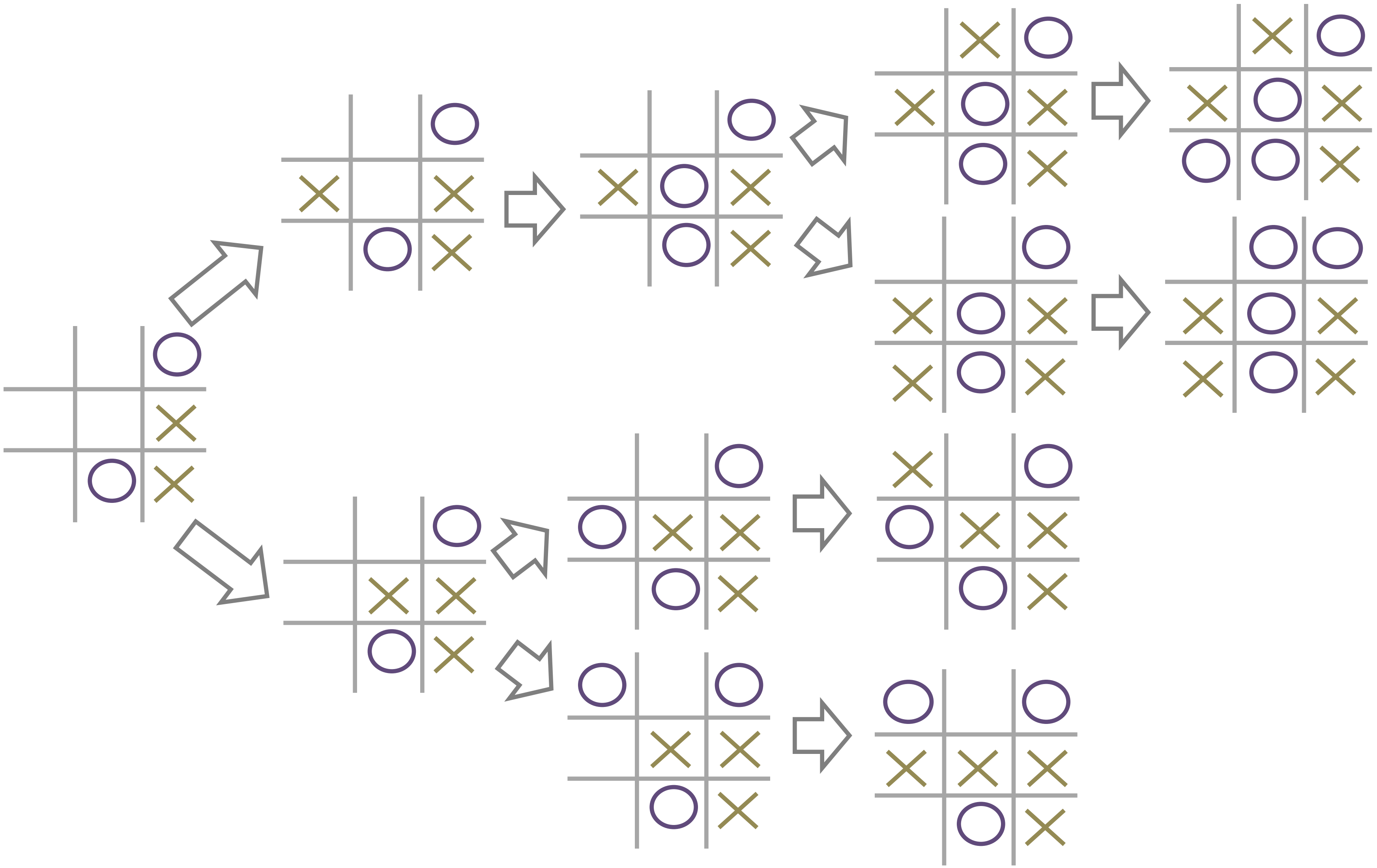
Original Paper

- Title:
  - Generative Adversarial Nets
- Authors:
  - Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio
- Organization:
  - Université de Montréal
- URL:
  - https://arxiv.org/abs/1406.2661

Outlines

- Mini-max Two-player Game
- Generative Model v.s. Discriminative Model
- Generative Adversarial Networks
- Convergence Proof
- Experiment
- Further Research

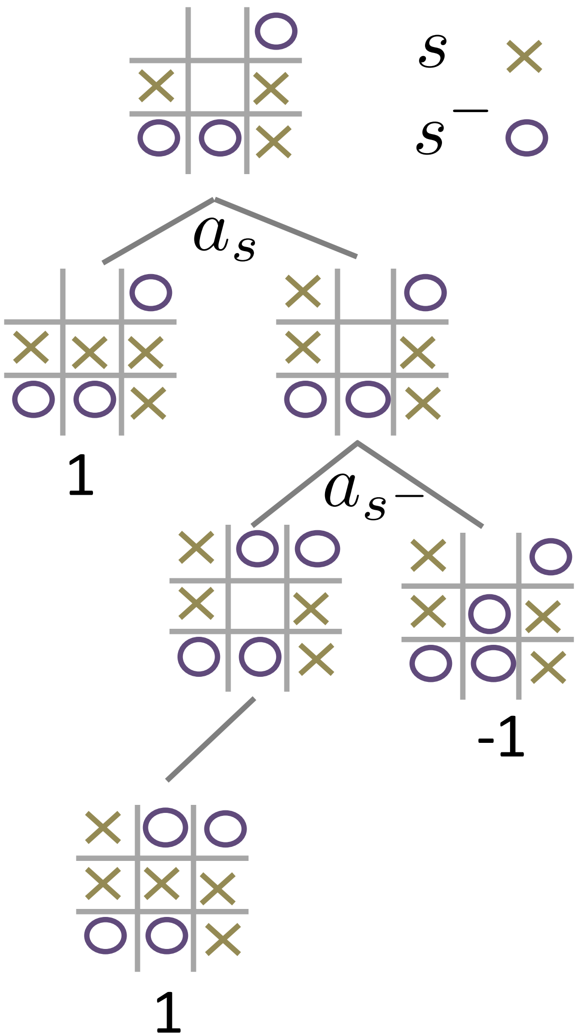
Mini-max Two-player Game



Mini-max Two-player Game

$$v_s = \max_{a_s} \min_{a_{s^-}} v_s(a_s, a_{s^-})$$

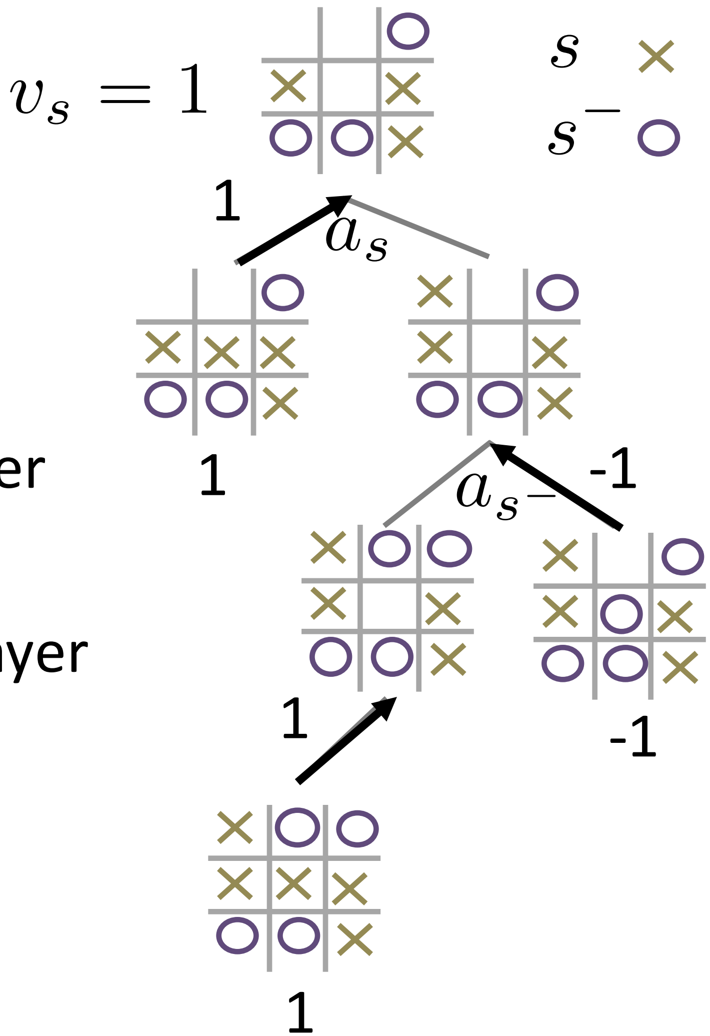
- s : the current player
- s^- : the opponent
- a_s : the action taken by current player
- a_{s^-} : the action taken by opponent
- v_s : the value function of current player



Mini-max Two-player Game

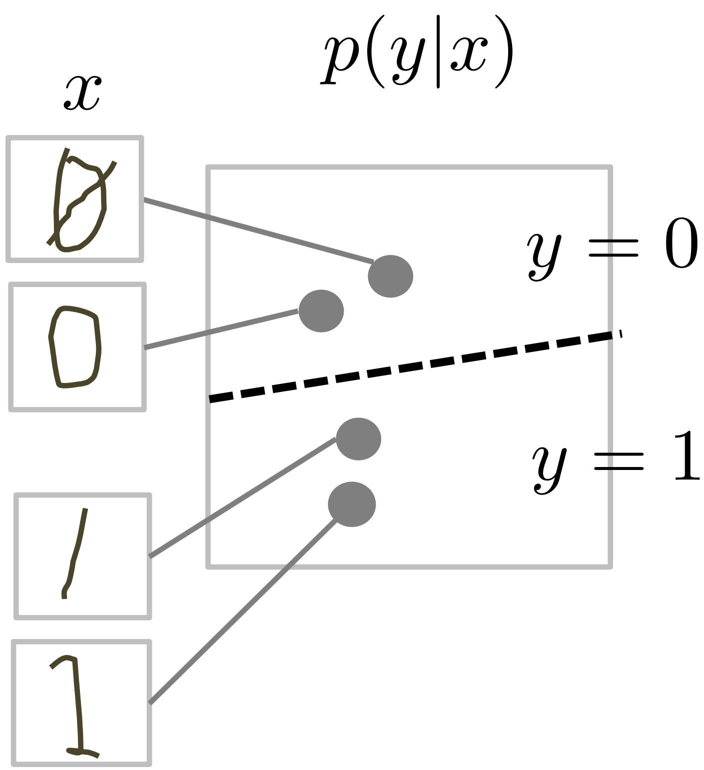
$$v_s = \max_{a_s} \min_{a_{s^-}} v_s(a_s, a_{s^-})$$

- s : the current player
- s^- : the opponent
- a_s : the action taken by current player
- a_{s^-} : the action taken by opponent
- v_s : the value function of current player

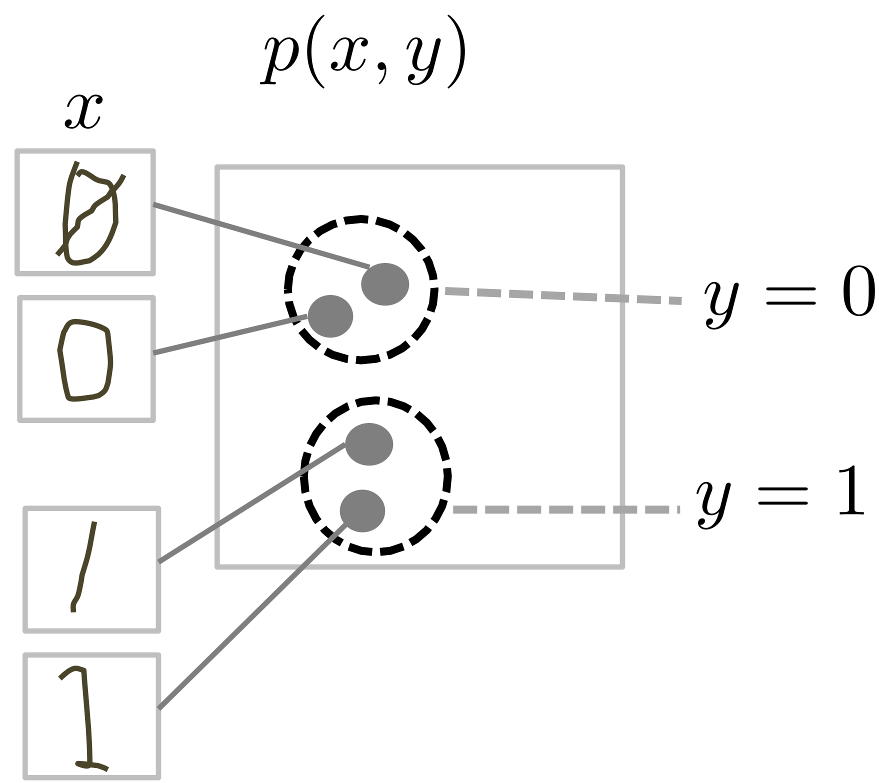


Generative Model v.s. Discriminative Model

- Discriminative Model



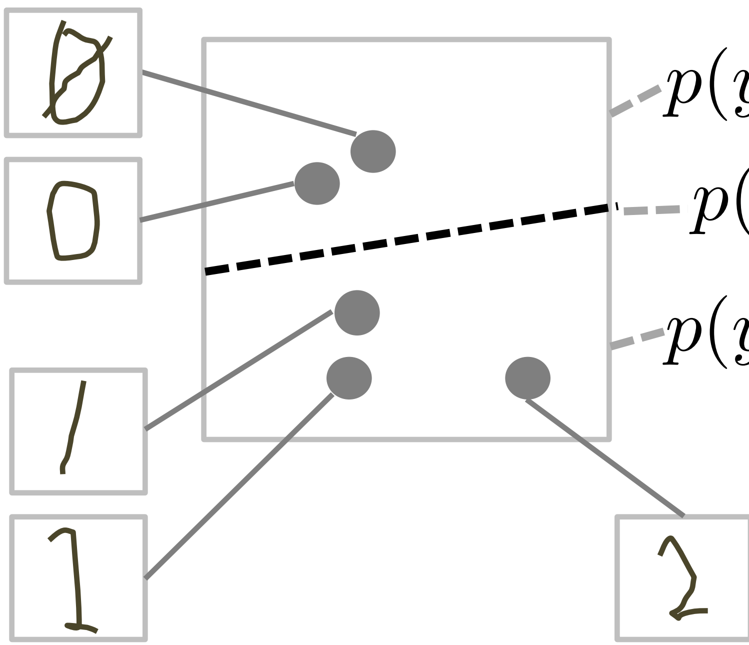
- Generative Model



Discriminative Model : p(y|x)

$$p(y=0|x) + p(y=1|x) = 1$$

x



$$p(y=0|x) > p(y=1|x) \Rightarrow y=0$$

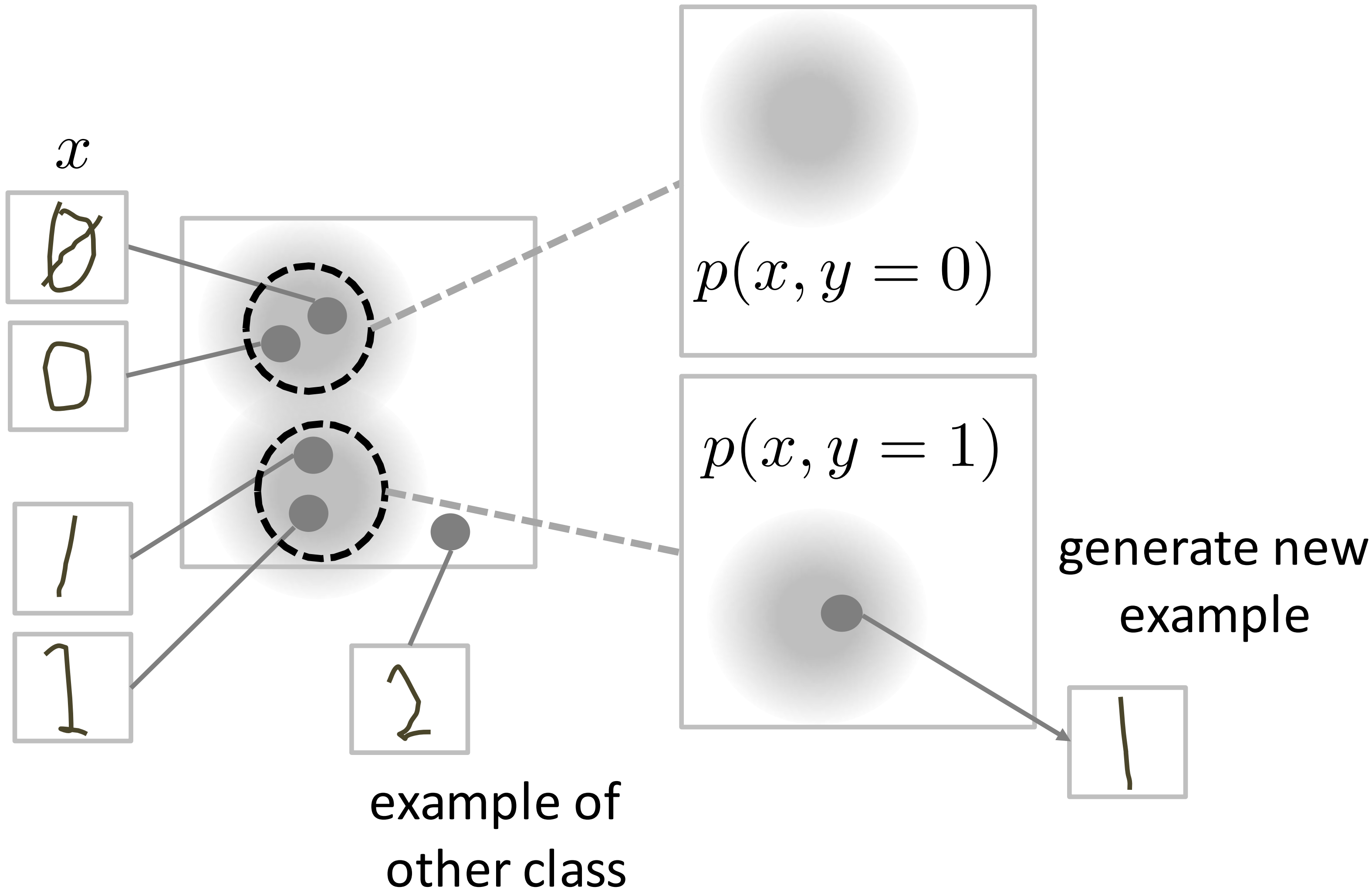
$$p(y=0|x) = p(y=1|x) = 0.5$$

$$p(y=1|x) > p(y=0|x) \Rightarrow y=1$$

$$\Rightarrow y=1$$

example of other class

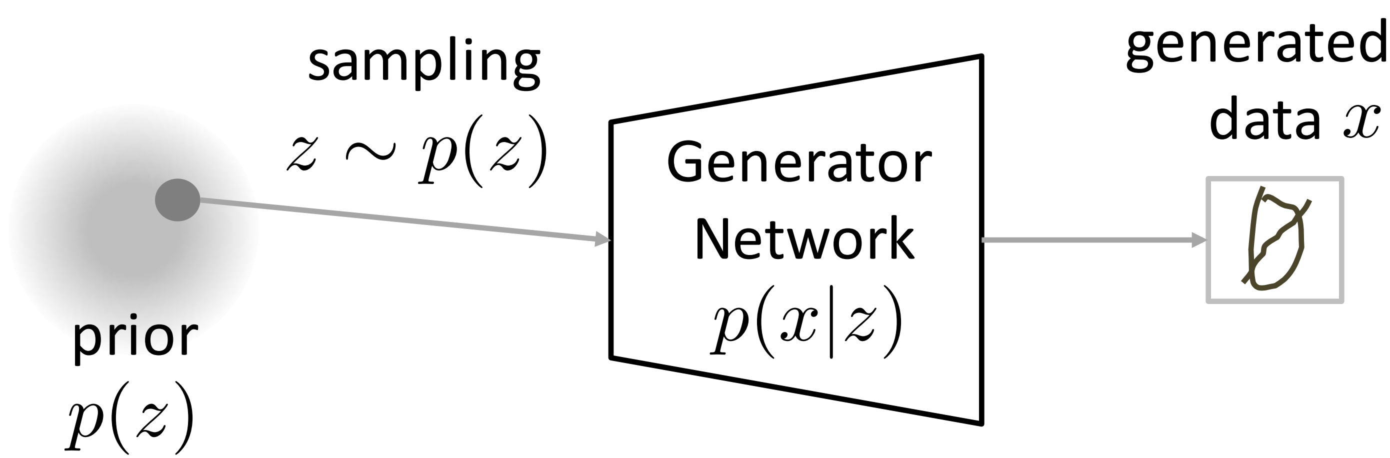
Generative Model : p(x, y)



Generative Adversarial Networks

- Generate new data by Neural Network

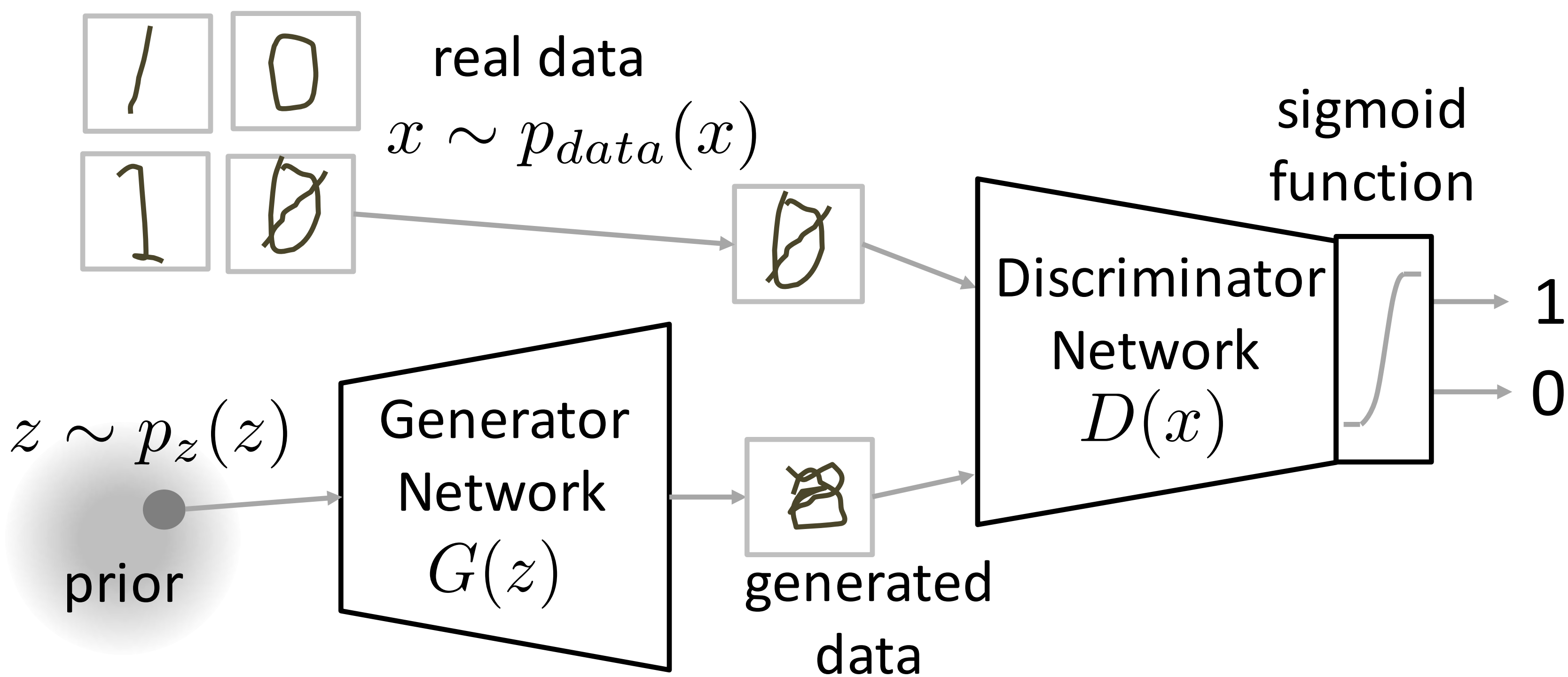
$$p(x, z) = p(z)p(x|z)$$



Generative Adversarial Networks

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

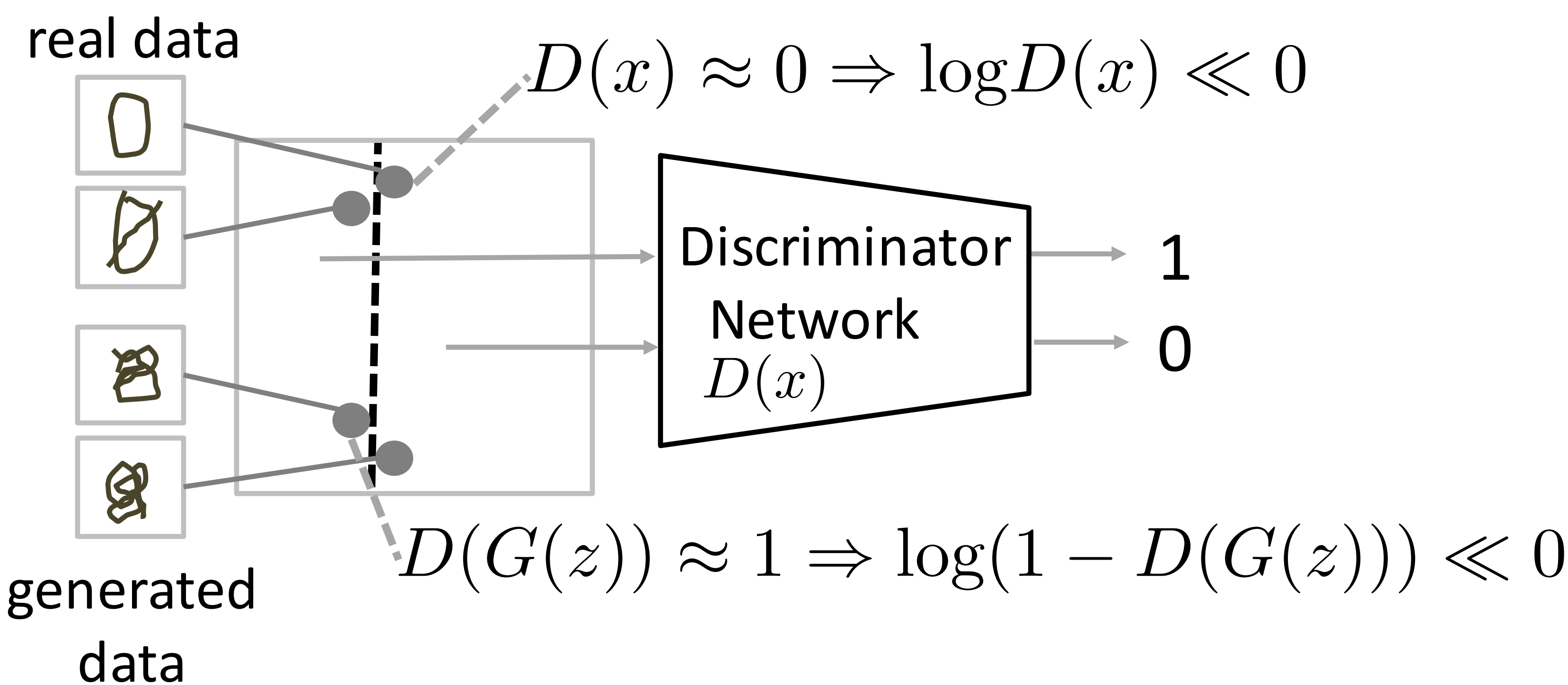
$$V(D,G) = \mathbb{E}_{x\sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z)))]$$



Training Discriminator Network

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

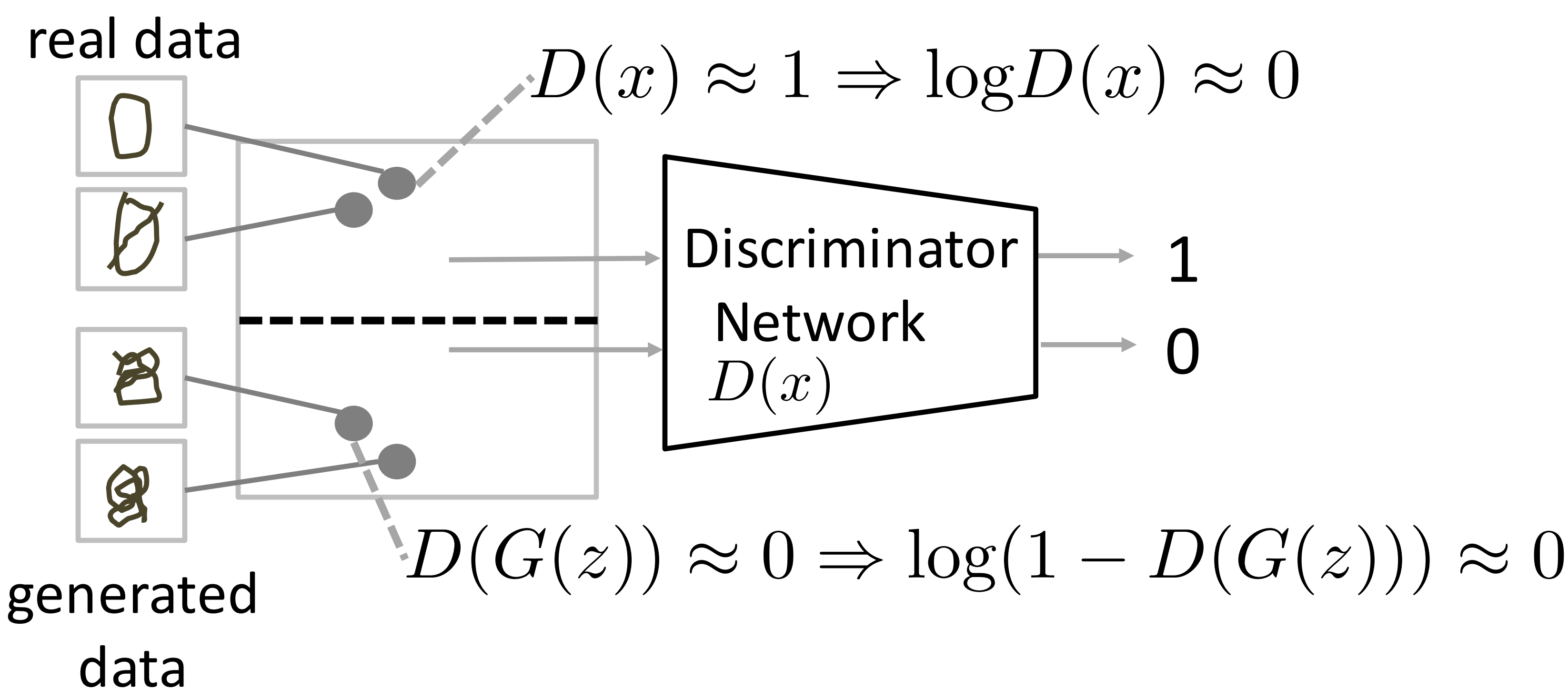
$D(x)$ should be 1 $D(G(z))$should be 0



Training Discriminator Network

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

$D(x)$ should be 1 $\quad$ $D(G(z))$should be 0



Training Generator Network

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

There is no G in this term.

$$\Rightarrow \min_G (\mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z))])$$

$D(G(z))$ should not be 0 => $D(G(z))$ should be 1

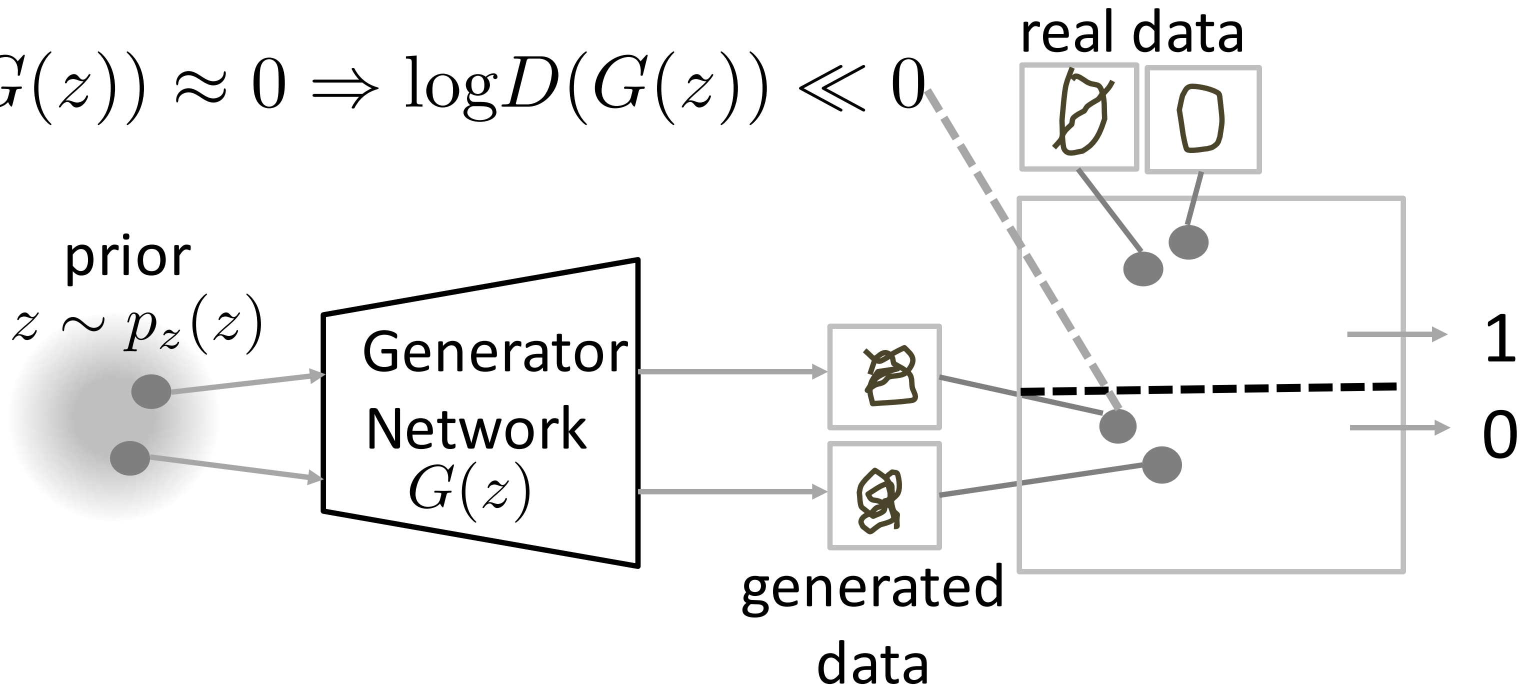
$$\Rightarrow \max_G (\mathbb{E}_{z\sim p_z(z)}[\log(D(G(z))])$$

Training Generator Network

$$\max_{G}(\mathbb{E}_{z \sim p_z(z)}[\log(D(G(z))])$$

$D(G(z))$ should be 1

$$D(G(z)) \approx 0 \Rightarrow \log D(G(z)) \ll 0$$

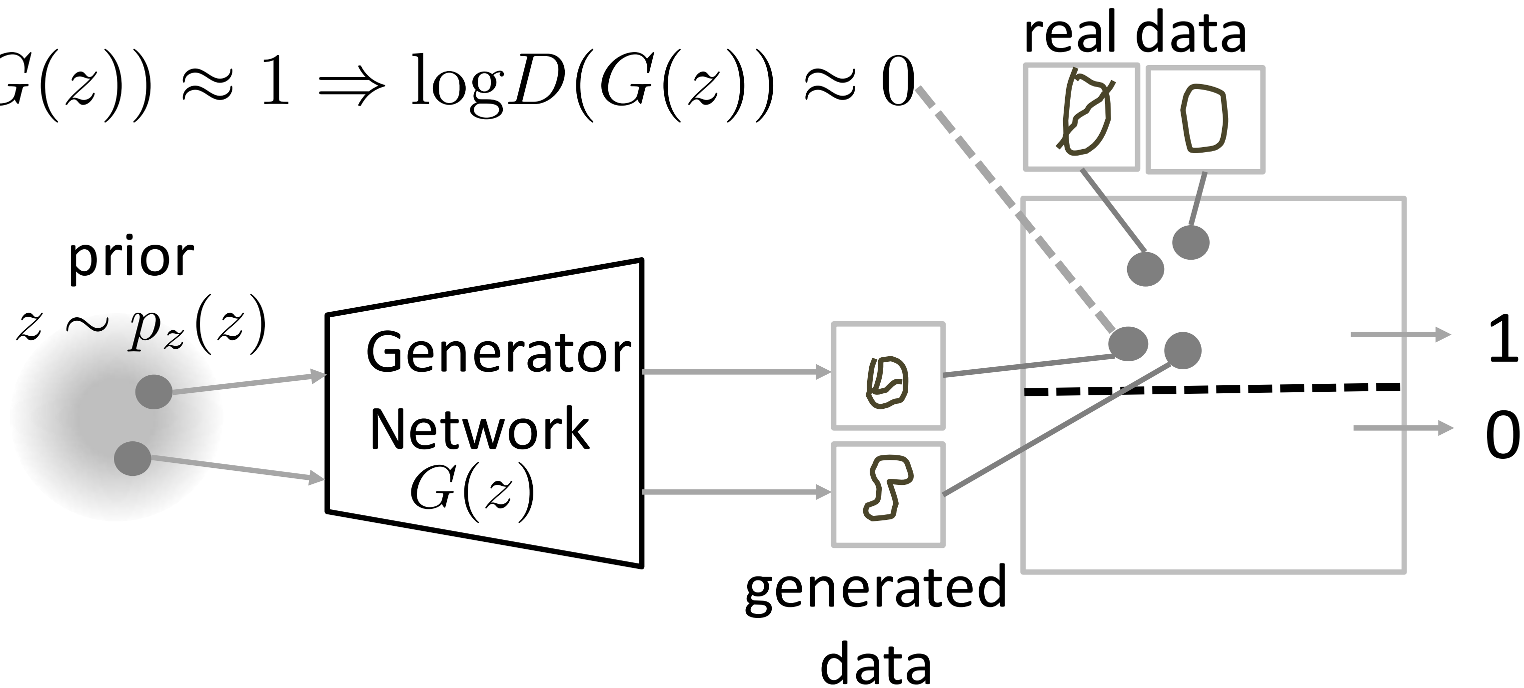


Training Generator Network

$$\max_G (\mathbb{E}_{z\sim p_z(z)}[\log(D(G(z))])$$

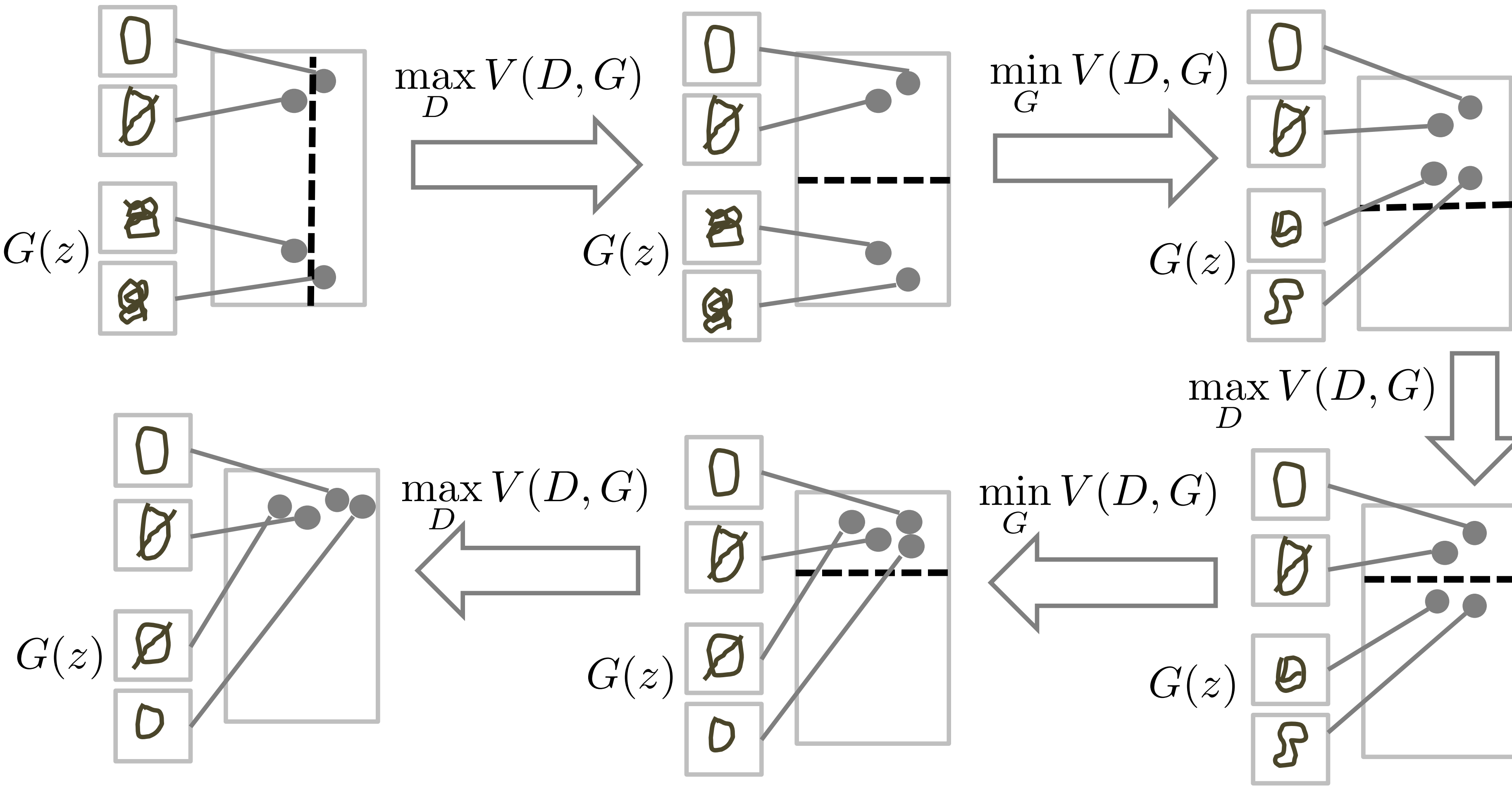
$D(G(z))$ should be 1

$$D(G(z)) \approx 1 \Rightarrow \log D(G(z)) \approx 0$$

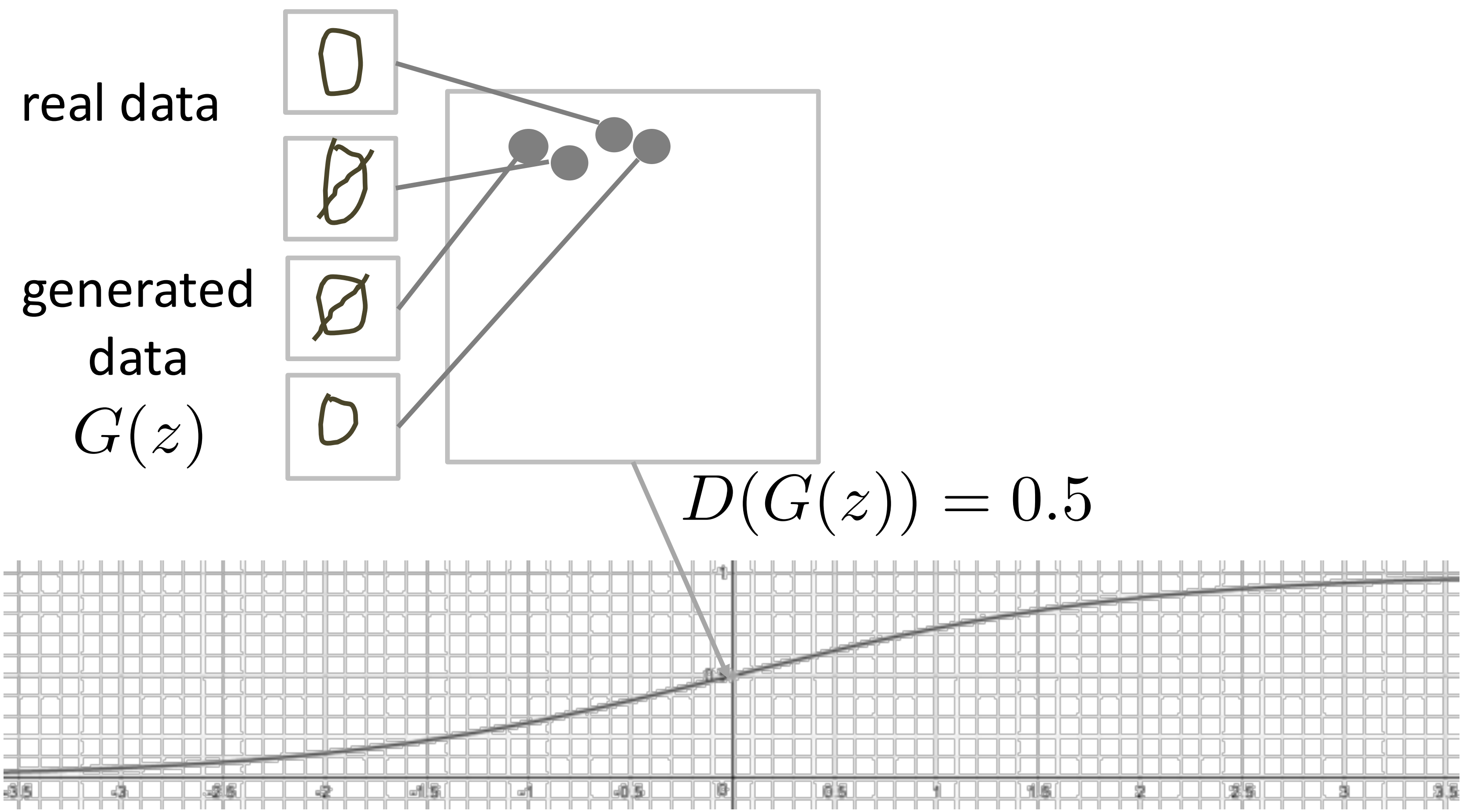


Training
Generative Adversarial Networks

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



Global Optimum

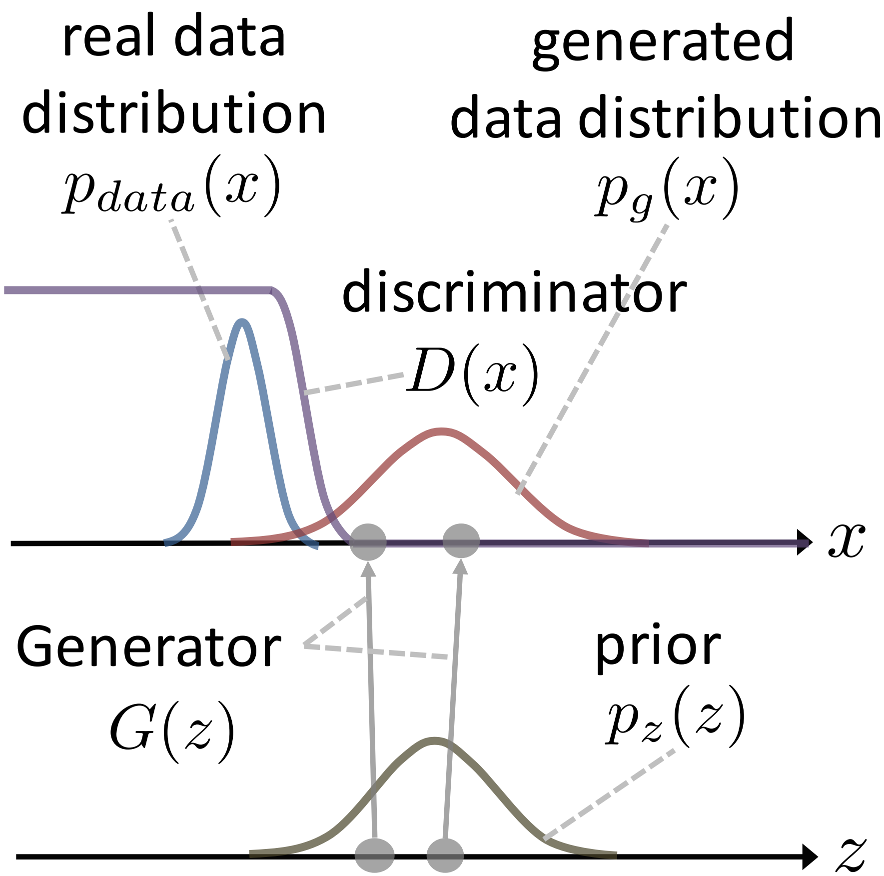


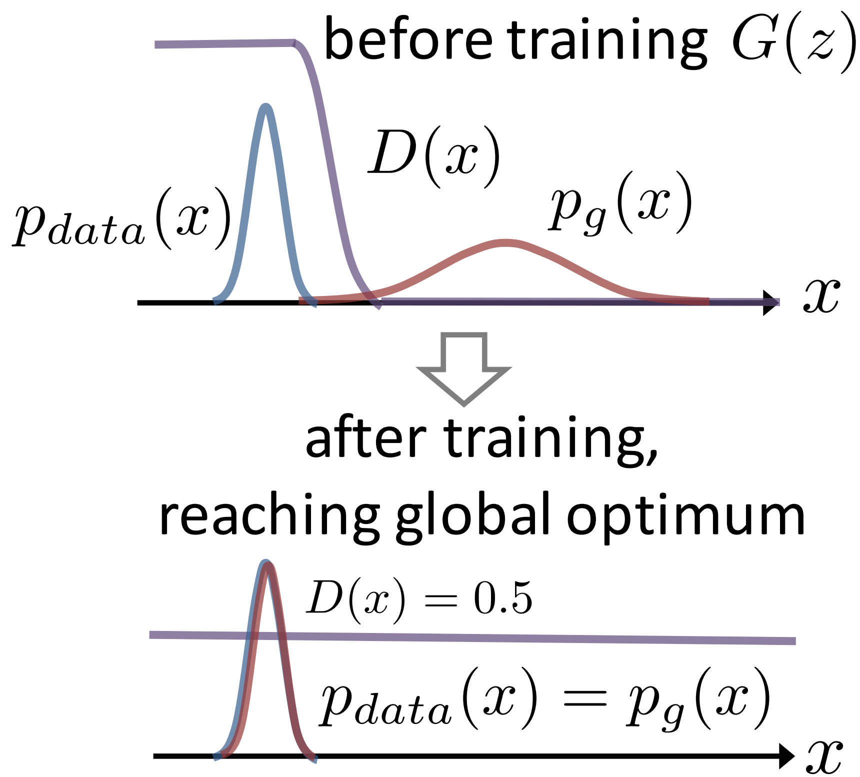
Convergence Proof

- Global Optimum Exists
- Converge to Global Optimum

Global Optimum Exists

global optimum: $p_{data} = p_g$

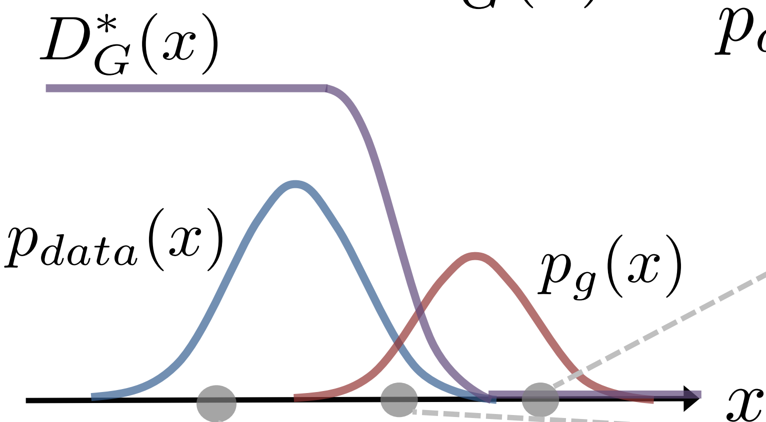




Global Optimum Exists

- For G fixed, the optimal discriminator D is:

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



$$p_{data}(x) = 0, p_g(x) = 0.5$$

$$D_G^*(x) = \frac{0}{0.5 + 0} = 0$$

$$p_{data}(x) = 0.5, p_g(x) = 0.5$$

$$D_G^*(x) = \frac{0.5}{0.5 + 0.5} = 0.5$$

$$p_{data}(x) = 0.5, p_g(x) = 0$$

$$D_G^*(x) = \frac{0.5}{0.5 + 0} = 1$$

Global Optimum Exists

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

$$= \int_x p_{data}(x) log(D(x))dx + \int_z p_z(z) log(1-D(G(z)))dz$$

$$x = G(z) \Rightarrow z = G^{-1}(x) \Rightarrow dz = (G^{-1})'(x)dx$$

$$\Rightarrow p_g(x) = p_z(G^{-1}(x))(G^{-1})'(x)$$

$$= \int_x p_{data}(x) log(D(x))dx + \int_x p_z(G^{-1}(x)) log(1-D(x))(G^{-1})'(x)dx$$

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

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

Global Optimum Exists

$$\max_D V(D,G) = \max_D \int_x p_{data}(x)log(D(x)) + p_g(x)log(1-D(x))dx$$

$$\frac{\partial}{\partial D(x)}(p_{data}(x)log(D(x)) + p_g(x)log(1-D(x))) = 0$$

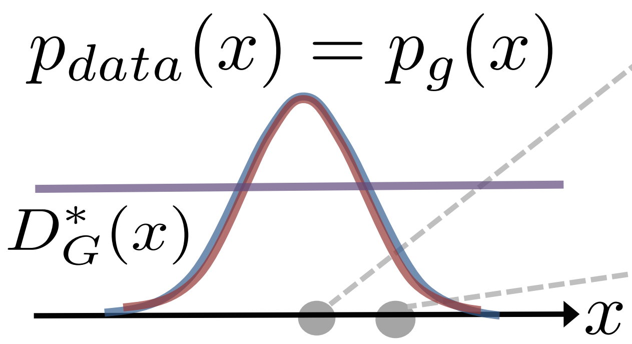
$$\Rightarrow \frac{p_{data}(x)}{D(x)} - \frac{p_g(x)}{1-D(x)} = 0$$

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

Global Optimum Exists

- Suppose the discriminator is optimal $D_G^*(x)$, the optimal generator makes: $p_{data}(x) = p_g(x)$

$$\Rightarrow D_G^*(x) = \frac{p_{data}(x)}{p_{data}(x) + p_g(x)} = \frac{1}{2}$$



$$p_{data}(x) = 0.8, p_g(x) = 0.8$$

$$D_G^*(x) = \frac{0.8}{0.8 + 0.8} = 0.5$$

$$p_{data}(x) = 0.3, p_g(x) = 0.3$$

$$D_G^*(x) = \frac{0.3}{0.3 + 0.3} = 0.5$$

Global Optimum Exists

$$C(G) = \max_D V(G, D)$$

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

$$= \int_x p_{data}(x) log(D_G^*(x)) + p_g(x) log(1 - D_G^*(x)) dx$$

$$= \int_x p_{data}(x) log(\frac{p_{data}(x)}{p_{data}(x) + p_g(x)}) + p_g(x) log(\frac{p_g(x)}{p_{data}(x) + p_g(x)}) dx$$

$$= \int_x p_{data}(x) log(\frac{p_{data}(x)}{\frac{p_{data}(x)+p_g(x)}{2}}) + p_g(x) log(\frac{p_g(x)}{\frac{p_{data}(x)+p_g(x)}{2}}) dx - log(4)$$

$$= KL[p_{data}(x) || \frac{p_{data}(x) + p_g(x)}{2}] + KL[p_g(x) || \frac{p_{data}(x) + p_g(x)}{2}] - log(4)$$

Global Optimum Exists

$$C(G) = \underbrace{KL[p_{data}(x)||\frac{p_{data}(x)+p_g(x)}{2}]}_{\geq 0} + \underbrace{KL[p_g(x)||\frac{p_{data}(x)+p_g(x)}{2}]}_{\geq 0} - log(4)$$

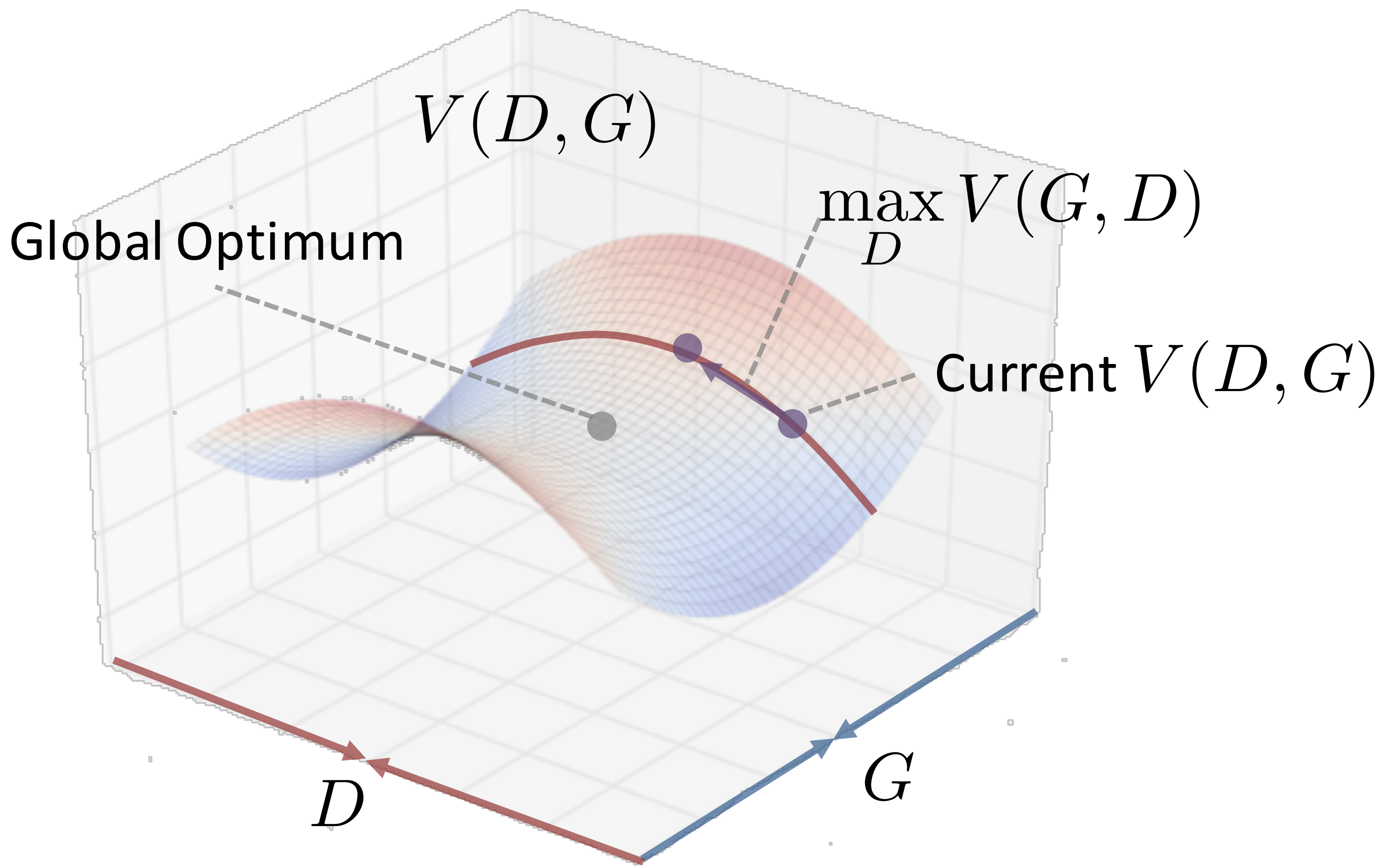
$$\min_G C(G) = 0 + 0 - log(4) = -log(4)$$

$$KL[p_{data}(x)||\frac{p_{data}(x)+p_g(x)}{2}] = 0$$

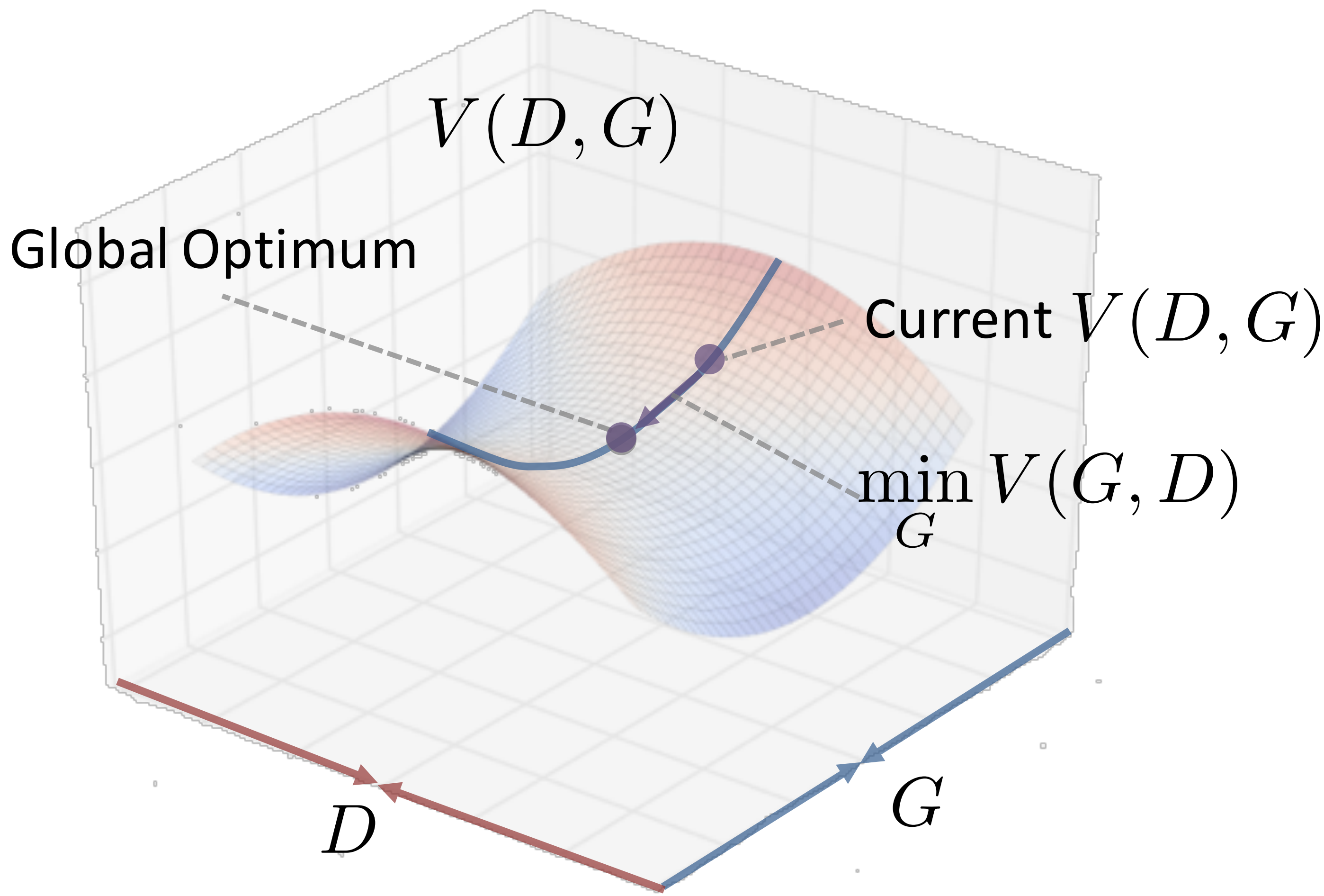
$$\text{when} \quad p_{data}(x) = \frac{p_{data}(x)+p_g(x)}{2}$$

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

Converge to Global Optimum



Converge to Global Optimum

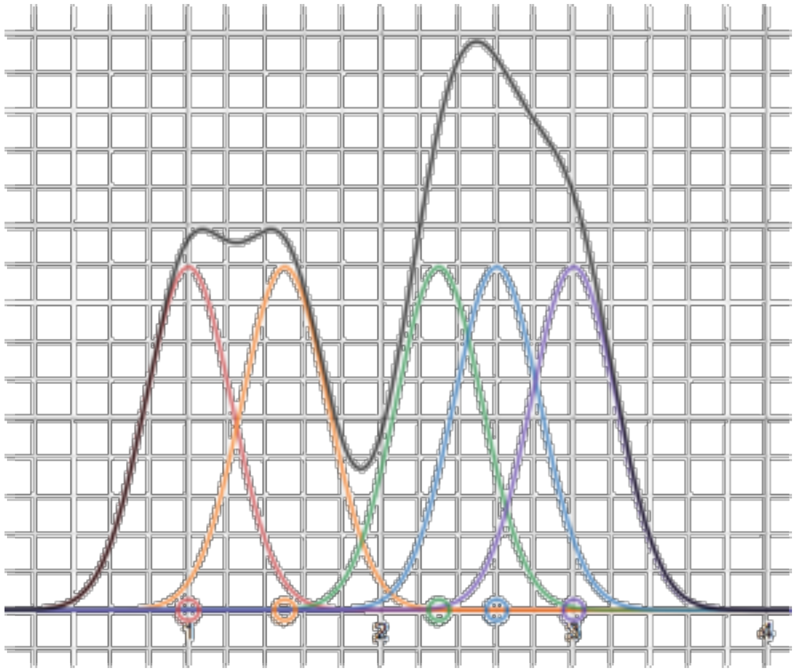


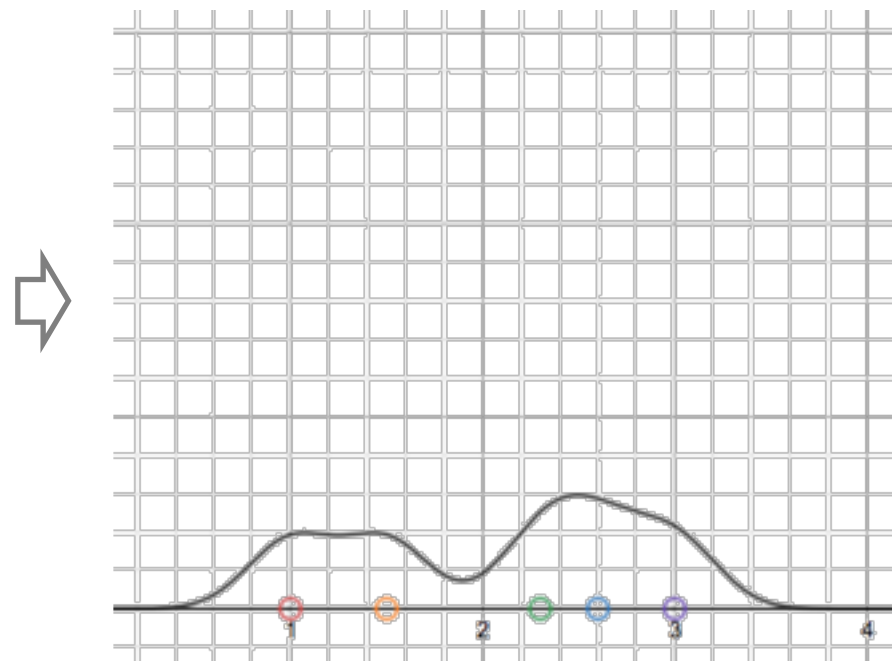
Experiment

- Quantitative Analyses
- Qualitative Analyses

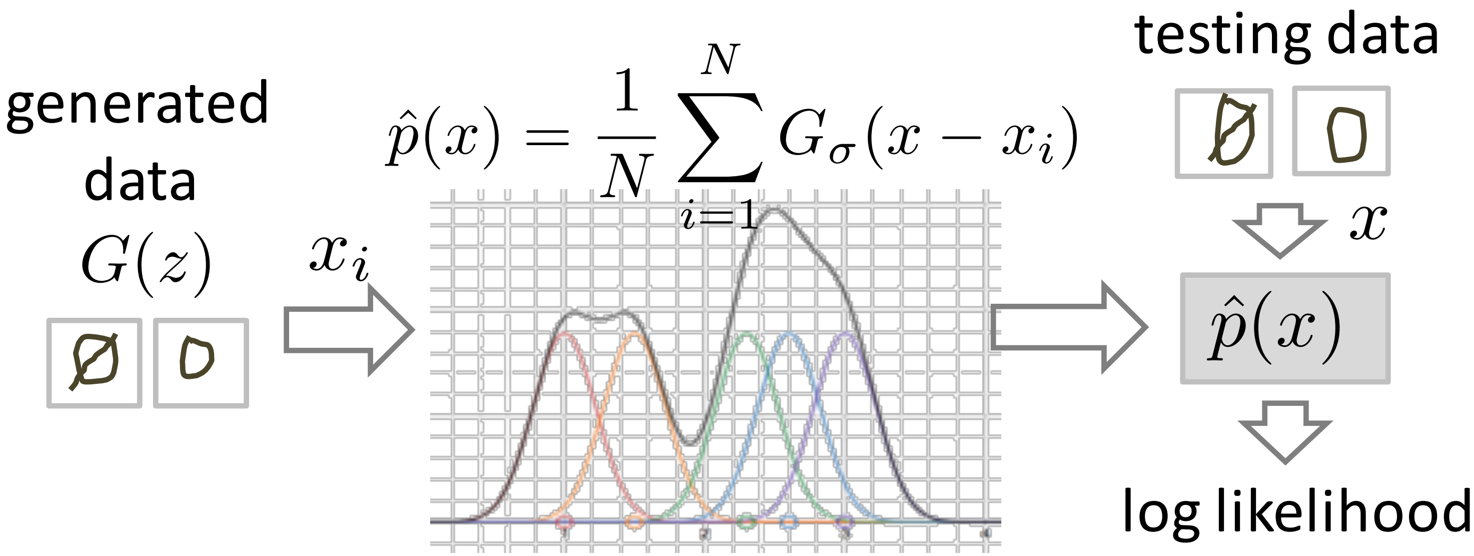
Experiment

- Quantitative Analyses
  - log-likelihood estimation
  - Parzen window: $\hat{p}(x) = \frac{1}{N}\sum_{i=1}^{N} G_{\sigma}(x - x_i)$





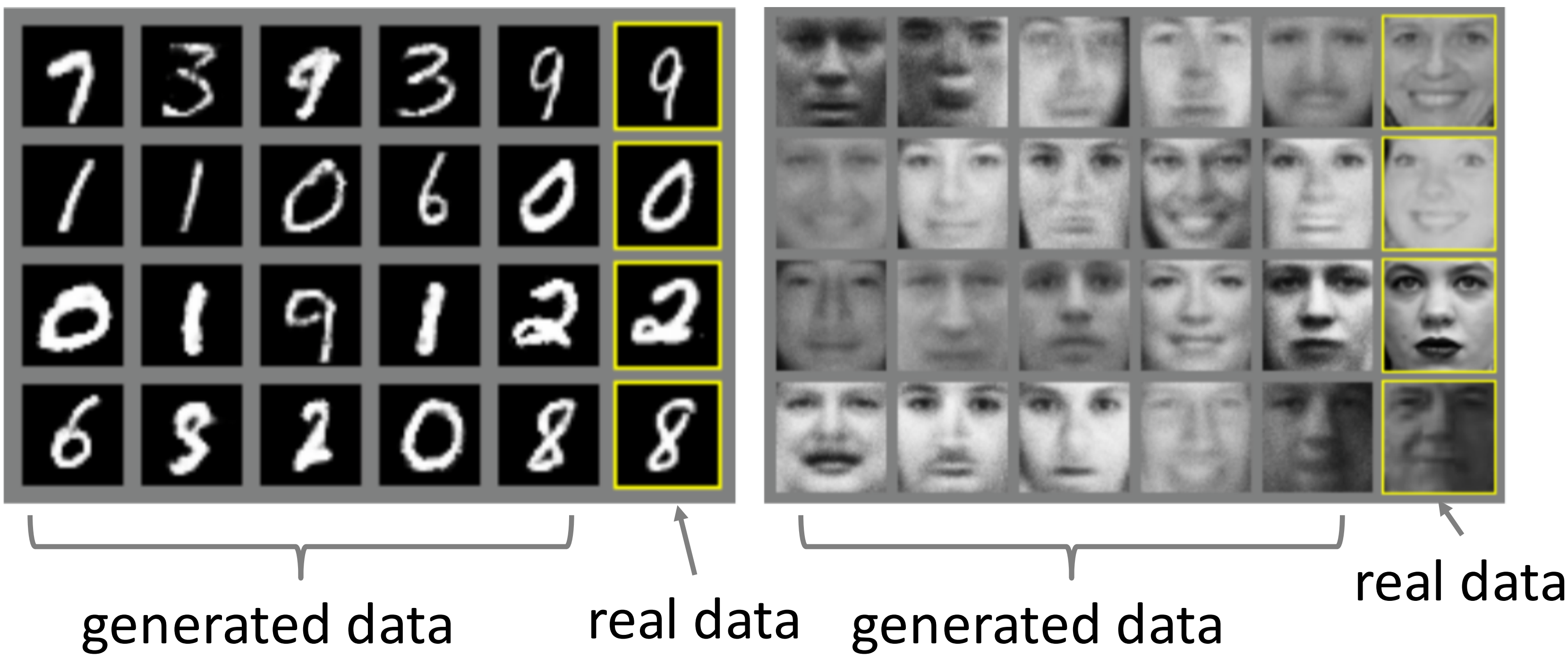
Experiment



Model	MNIST	TFD
DBN [3]	138 ± 2	1909 ± 66
Stacked CAE [3]	121 ± 1.6	$\mathbf{2110 \pm 50}$
Deep GSN [6]	214 ± 1.1	1890 ± 29
Adversarial nets	$\mathbf{225 \pm 2}$	$\mathbf{2057 \pm 26}$

Experiment

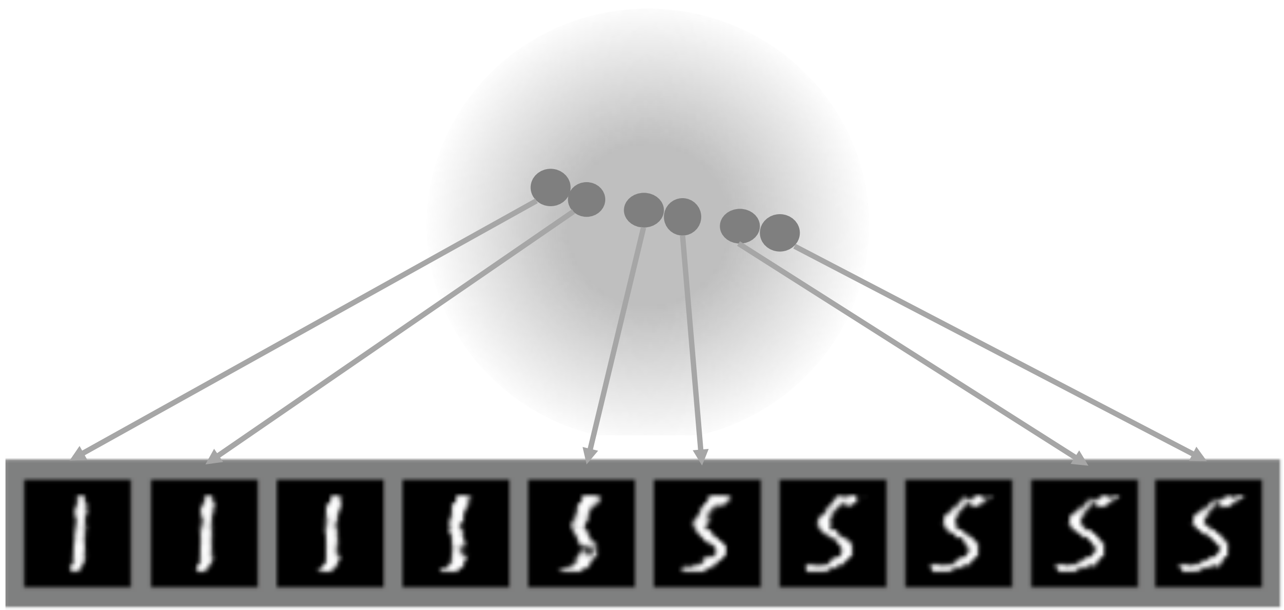
- Qualitative Analyses :
  - Visualization of samples from the model



Experiment

- Qualitative Analyses :
  - linearly interpolation in z space

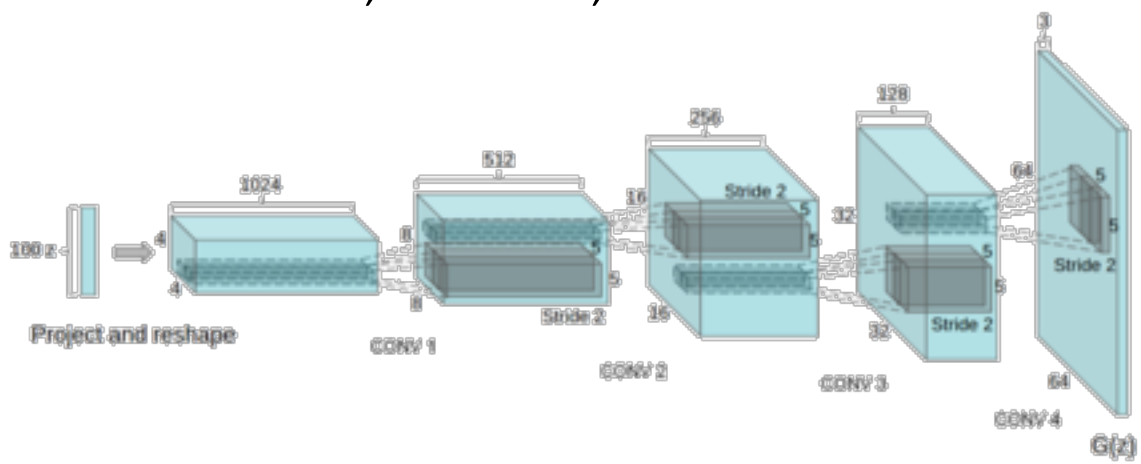
prior: $p_z(z)$



Further Research

Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks

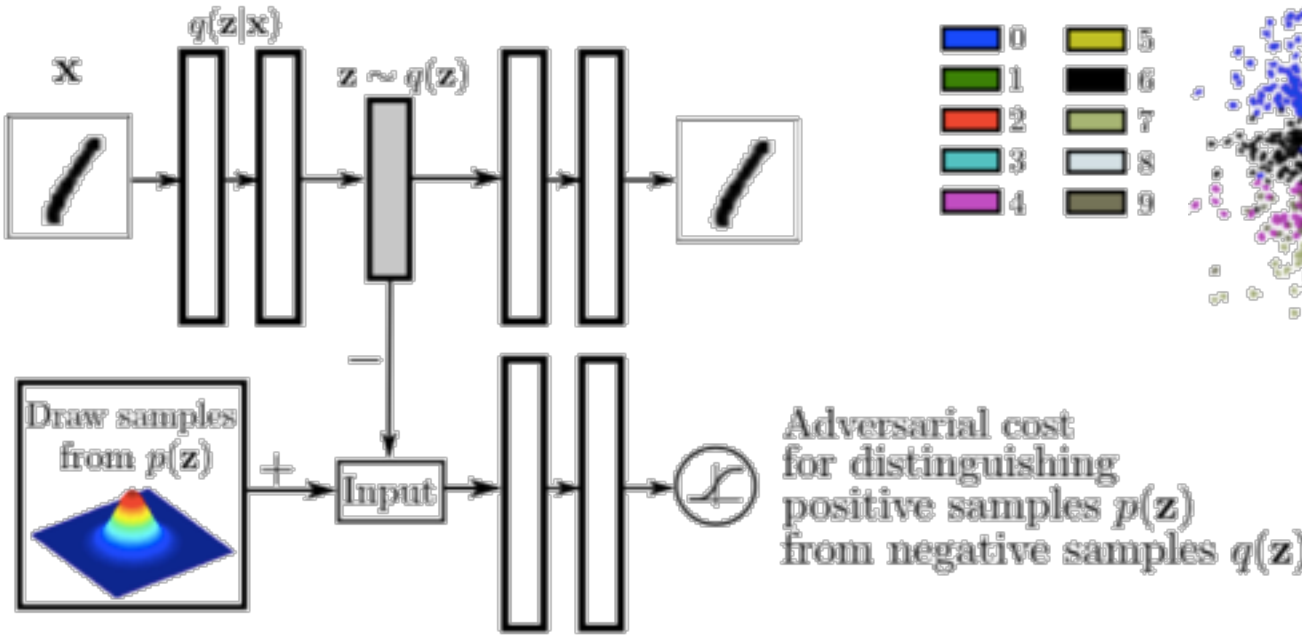
Alec Radford, Luke Metz, Soumith Chintala

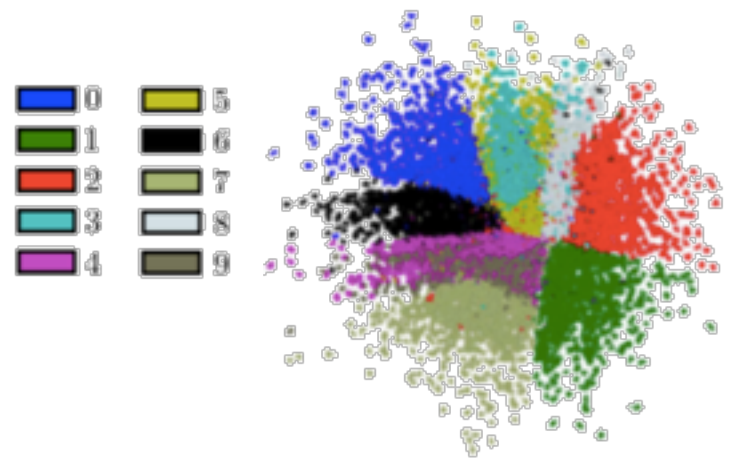


Further Research

Adversarial Autoencoders

Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, Brendan Frey



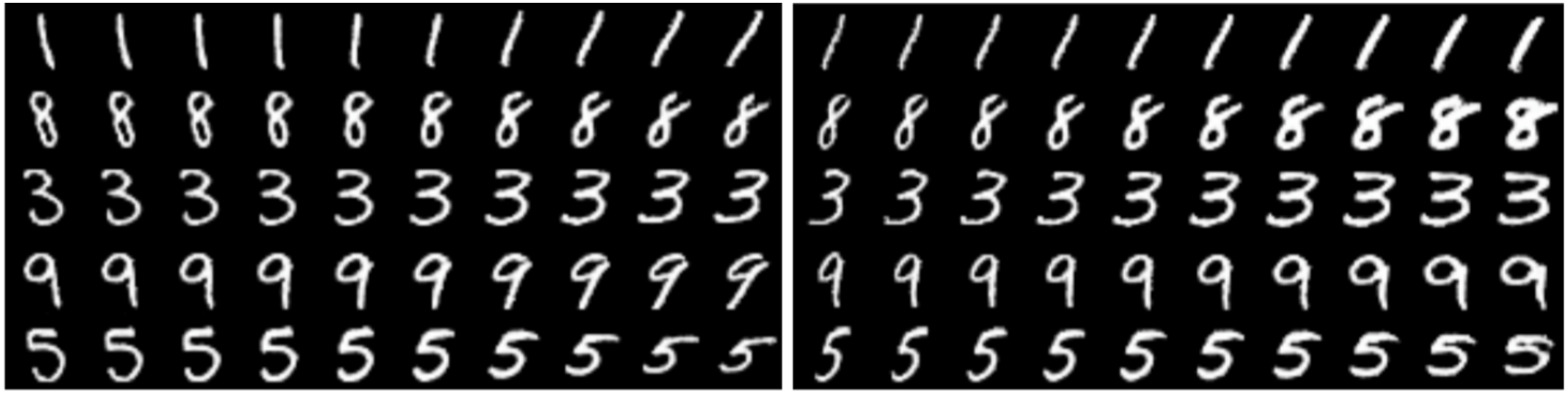


Further Research

InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets

Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, Pieter Abbeel

$$\min_{G,Q} \max_{D} V_{\text{InfoGAN}}(D, G, Q) = V(D, G) - \lambda L_I(G, Q)$$



Source Code

- Original paper (theano):
  - https://github.com/goodfeli/adversarial
- Tensorflow implementation:
  - https://github.com/ckmarkoh/GAN-tensorflow