CS 8501 Advanced Topics in Machine Learning

Lecture 12: Generative Adversarial Networks

Yangfeng Ji

Information and Language Processing Lab

Department of Computer Science

University of Virginia

https://yangfengji.net/

Generative Adversarial Networks

Classifier as Discriminator

Consider a classifier $D_\phi(x)$ that takes input x and predicts whether it is an example from the real data

$$0 \leq D_\phi(x) \leq 1$$

- $D_\phi(x) = p(\text{real data}|x)$ is the probability of x from real data
- ϕ is the classifier parameter

Data Sources

There are two sources of data

- $x \sim p^*(x)$, where $p^*(x)$ is the empirical distribution of the training set
  - A good discriminator $D_\phi(x)$ should be as large as possible
- $x \sim q_\theta(x)$, implicitly defined by a generator $x = G_\theta(z)$, where $z \sim p(z)$ is a pre-defined latent variable
  - A good discriminator $D_\phi(x)$ on $x \sim q_\theta(x)$ should be as small as possible

Goal:

- Teach the generator G to generate high-quality examples to fool D

GAN Game

GAN can be considered as a **two-player minimax game** with the following value function

$$V(\phi, \theta) = \frac{1}{2} E_{p^*(x)}[\log D_\phi(x)] + \frac{1}{2} E_{z \sim p(z)}[\log(1 - D_\phi(G_\theta(z)))]$$

Training GAN is equivalent to

$$\min_\theta \max_\phi V(\phi, \theta)$$

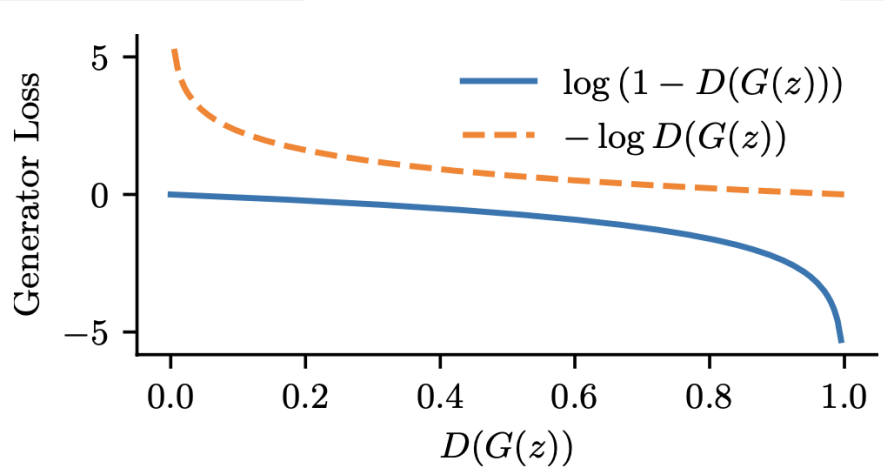
- ϕ is the parameter of D
- θ is the parameter of G

Max-max Game?

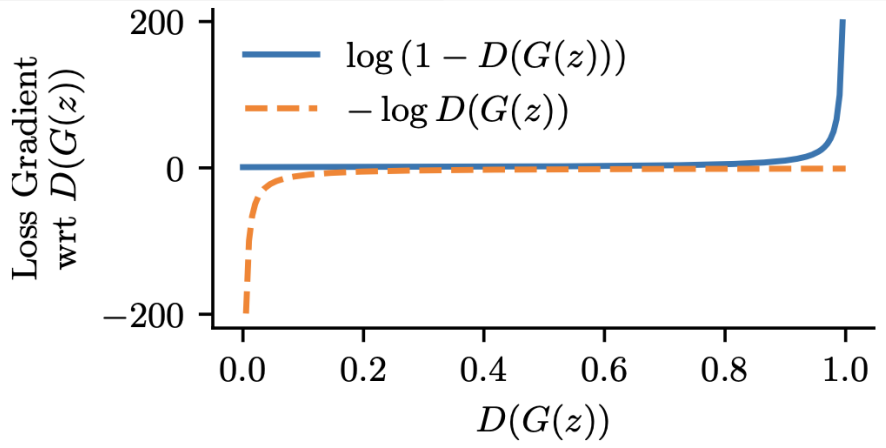
What about

$$\max_{\theta}\max_{\phi} V'(\phi,\theta) = \frac{1}{2}E_{p^*(x)}[\log D_\phi(x)] + \frac{1}{2}E_{z\sim p(z)}[\log(D_\phi(G_\theta(z)))]$$

The gradient of G wrt the loss function



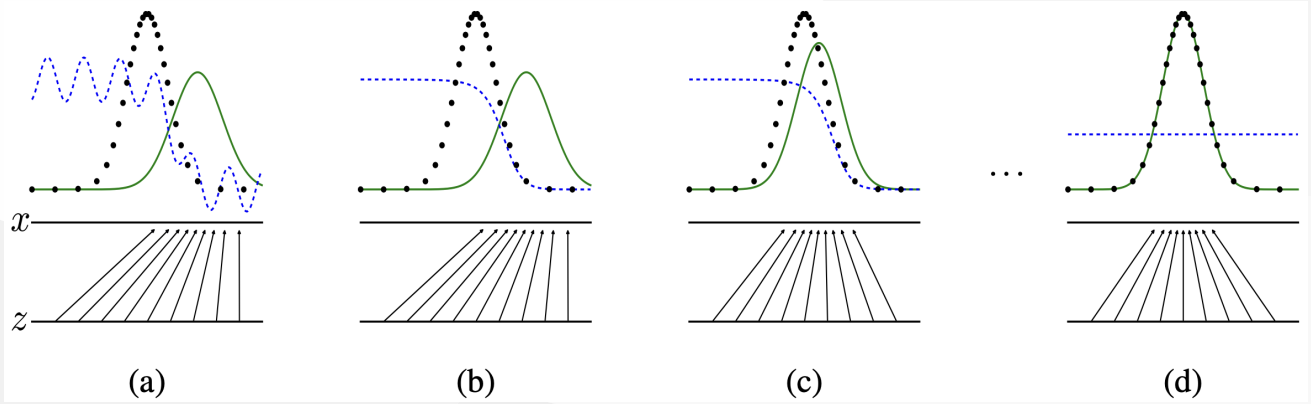
(a) Generator loss as a function of discriminator score.



(b) The gradients of the generator loss with respect to the discriminator score.

Training GAN

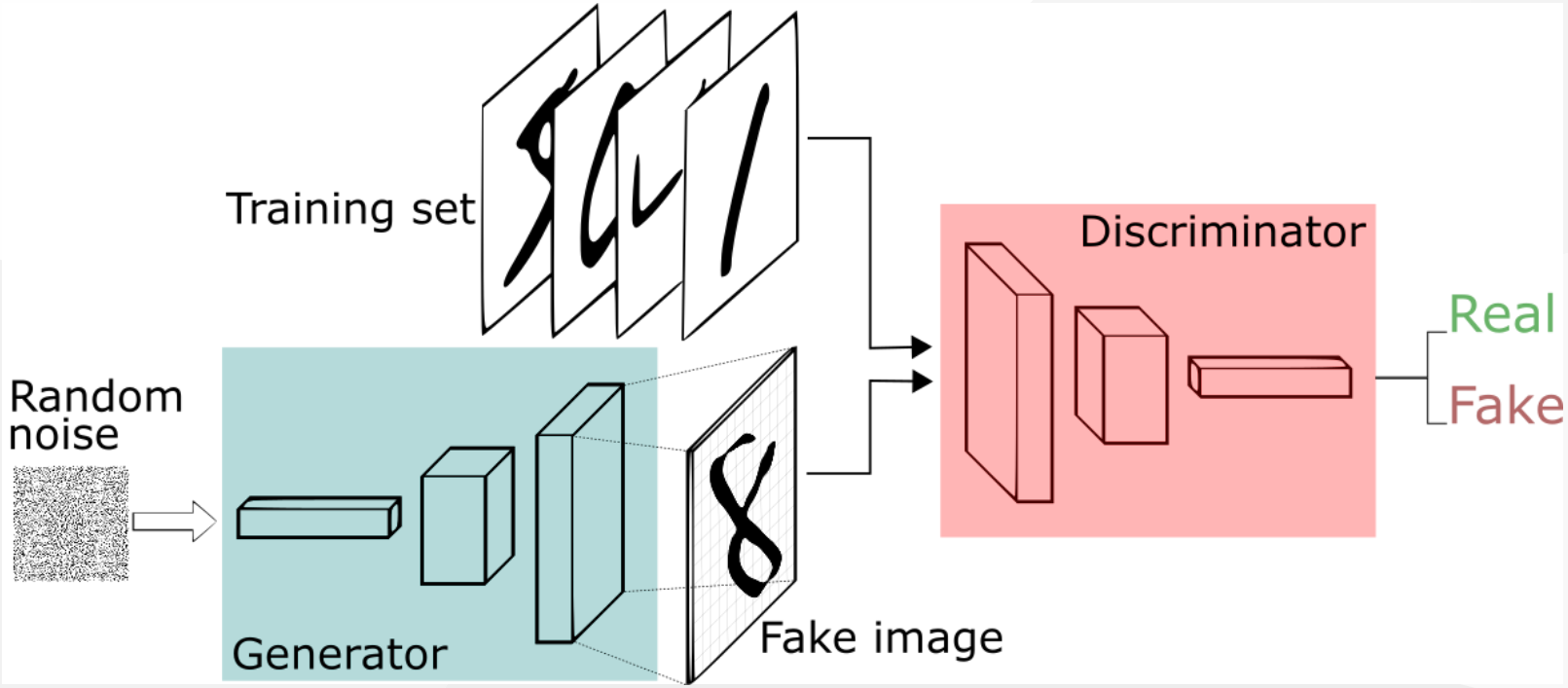
The high-level idea of training GAN can be explained by the following illustration



- black dots: training examples
- green curve: generator distribution
- blue curve: discriminator distribution

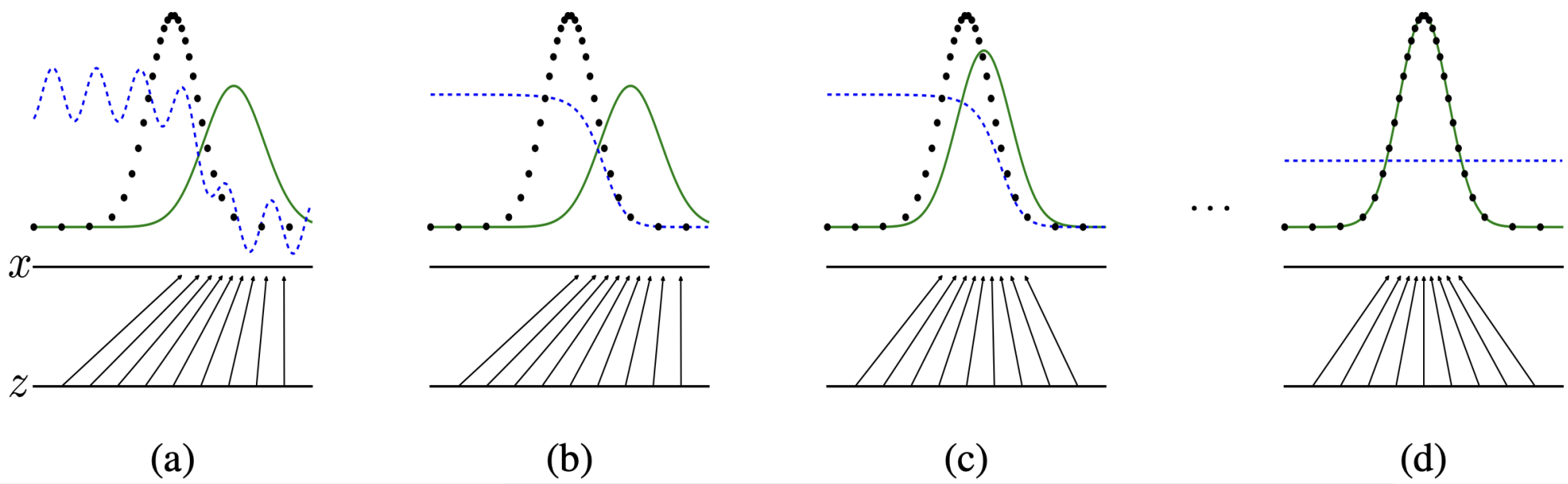
Training GAN (II)

Illustration of GAN architecture



Dilemma of Training GANs

- Need a good discriminator D to get started
- Need a perfect generator G to stop



Algorithm

Algorithm 1 Minibatch stochastic gradient descent training of generative adversarial nets. The number of steps to apply to the discriminator, k, is a hyperparameter. We used $k = 1$, the least expensive option, in our experiments.

for number of training iterations **do**

  for k steps **do**

  - Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
  - Sample minibatch of m examples $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(m)}\}$ from data generating distribution $p_{\text{data}}(\boldsymbol{x})$.
  - Update the discriminator by ascending its stochastic gradient:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(\boldsymbol{x}^{(i)}\right) + \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right) \right].$$

  end for

  - Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
  - Update the generator by descending its stochastic gradient:

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right).$$

end for

The gradient-based updates can use any standard gradient-based learning rule. We used momentum in our experiments.

[Goodfellow et al., 2014]

Challenge of Training GANs

- Mode collapse: the generator converges to a distribution that does not cover the whole data distribution
- Mode hopping: the generator hops between generating different modes of the data distribution

GAN Lab with a multimodal distribution

Convergence of Training GANs

[Murphy 2023] "if **the discriminator is optimal**, the generator is minimising a distributional divergence or distance between the data and model distribution, and thus under **infinite capacity** and **perfect optimization** can learn the data distribution."

- The discriminator is optimal?
  - Absolutely not, we training the discriminator based on the current generator's performance
- Infinite capacity?
  - We have no idea how to quantify NNs' capacity
- Perfect optimization?
  - No, we use mini-batch gradient-based optimization

Conditional GANs

AC-GANs

Consider an additional conditional variable $y \in \mathcal{Y}$, which is a discrete random variable indicating the class of x.

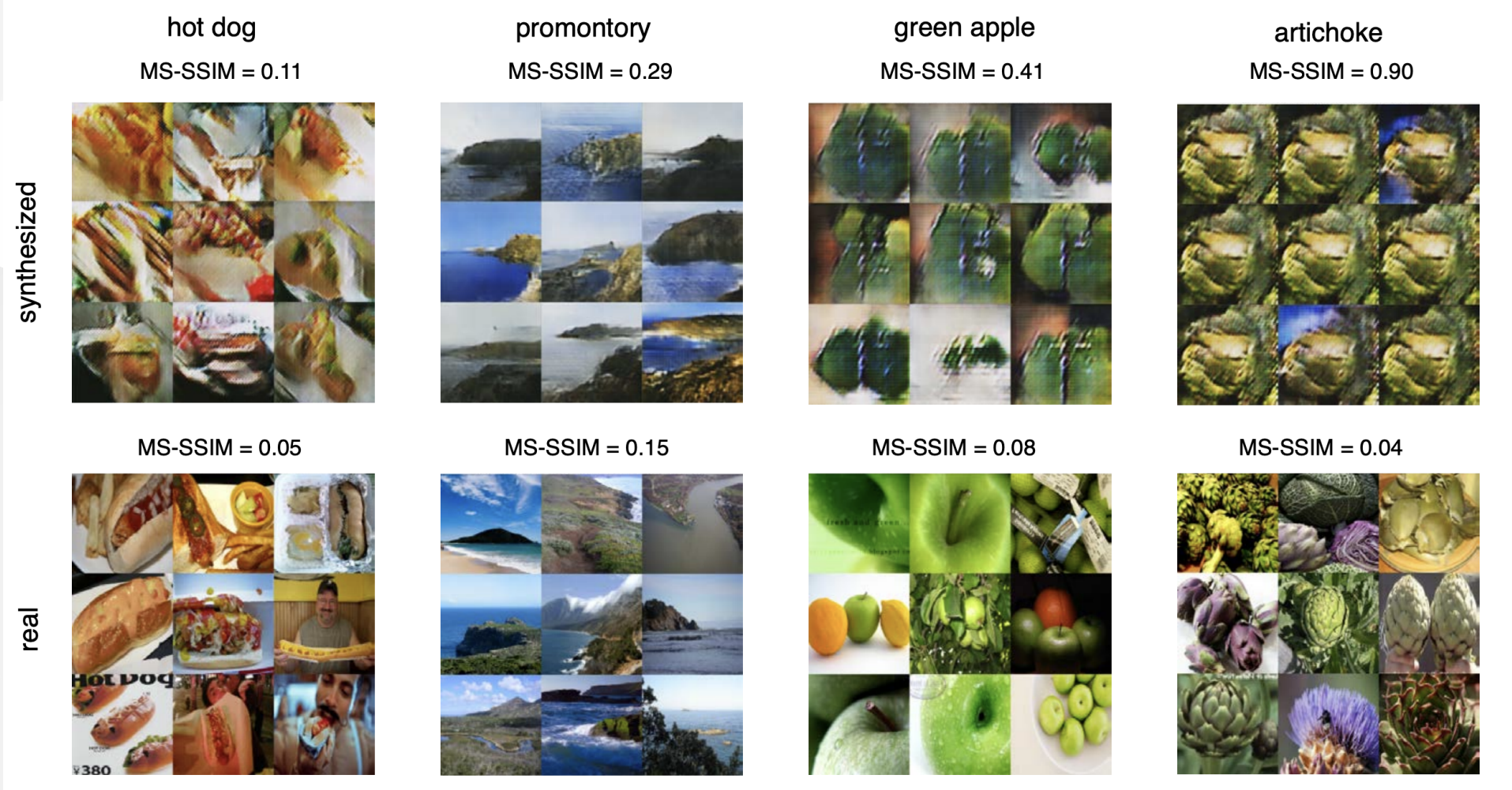
- Modified GAN objective

$$\min_{\theta} \max_{\phi} E_{x \sim p^*(x)}[\log D_\phi(x|y)] + E_{z \sim p(z|y)}[\log(1 - D_\phi(G_\theta(z|y))]$$

- Auxiliary classifier

$$\max_{\theta} \max_{\psi} E_{x \sim p^*(x)}[\log p_\psi(y|x)] + E_{z \sim p(z|y)}[\log p_\psi(y|G_\theta(z|y))]$$

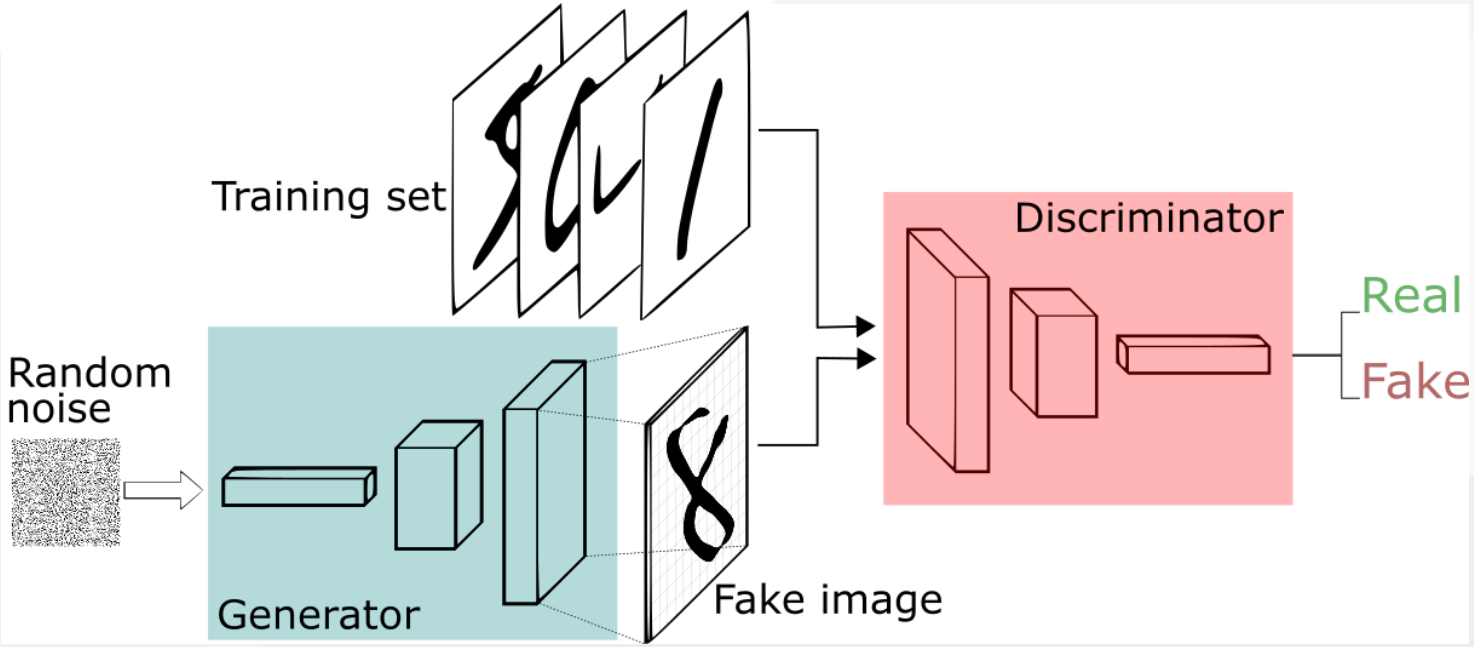
AC-GANs: Examples



Inference with GANs

Inference on Given x

- In VAE, inference network $q(z|x)$ provides an estimate of the latent distribution on z for any given x
- But original GAN does not offer this option



Adversarially Learned Inference

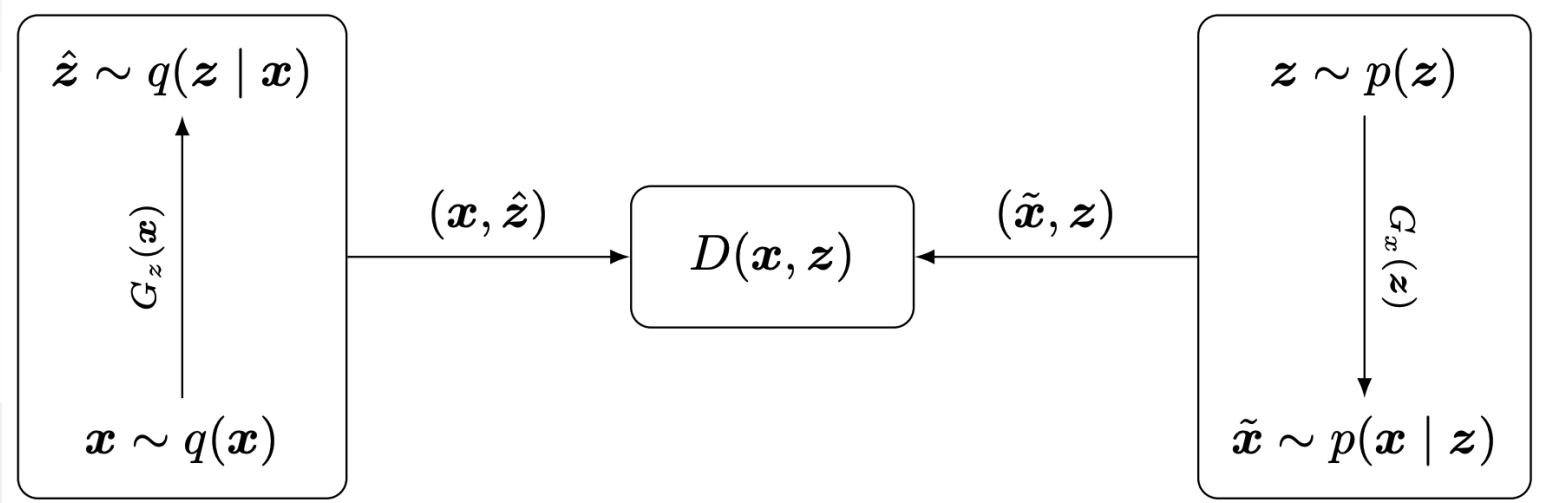


Figure 1: The adversarially learned inference (ALI) game.

Discriminator:

- Defined over x and z
- Differentiates the pair $(x, G_z(x))$ associated with real example x and the pair $(G_x(z), z)$ associated with the generated example $G_x(z)$

Adversarially Learned Inference: Objective

The value function of ALI is similar to the original GAN, except D is defined on the joint distribution of (x, z)

$$\min_G \max_D E_{q(x)}[\log D(x, G_z(x))] + E_{p(z)}[\log(1 - D(G_x(z), z))]$$

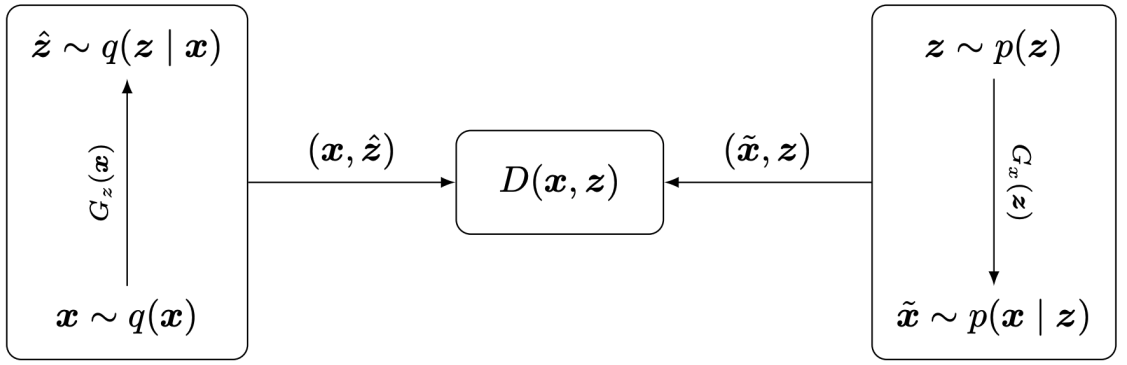


Figure 1: The adversarially learned inference (ALI) game.

$G_z(x)$ is the learned inference function, similar to $q(z|x)$ in VAE

Adversarially Learned Inference: Examples

It shares the merits from both VAE and GAN

Thank You!