DEEP CONVOLUTIONAL GENERATIVE ADVERSARIAL NETWORK

HAONAN PENG

LINXIAO ZHANG

INTRODUCTION

- Generative adversarial network (GAN) is deep learning model to capture the training data’s distribution to generate new data from that same distribution.
- GANs are made up of two distinct models: a generator and a discriminator.
- Deep convolutional GAN is a direct extension of GANs which uses convolutional and convolutional transpose layers in discriminator and generator.

METHODOLOGY

- A zero-sum game:
  - Generator tries to minimize the probability that discriminator will predict its outputs are fake:

    $$1 - \log D(G(z))$$

  - Discriminator tries to maximize the probability it correctly classifies reals and fakes:

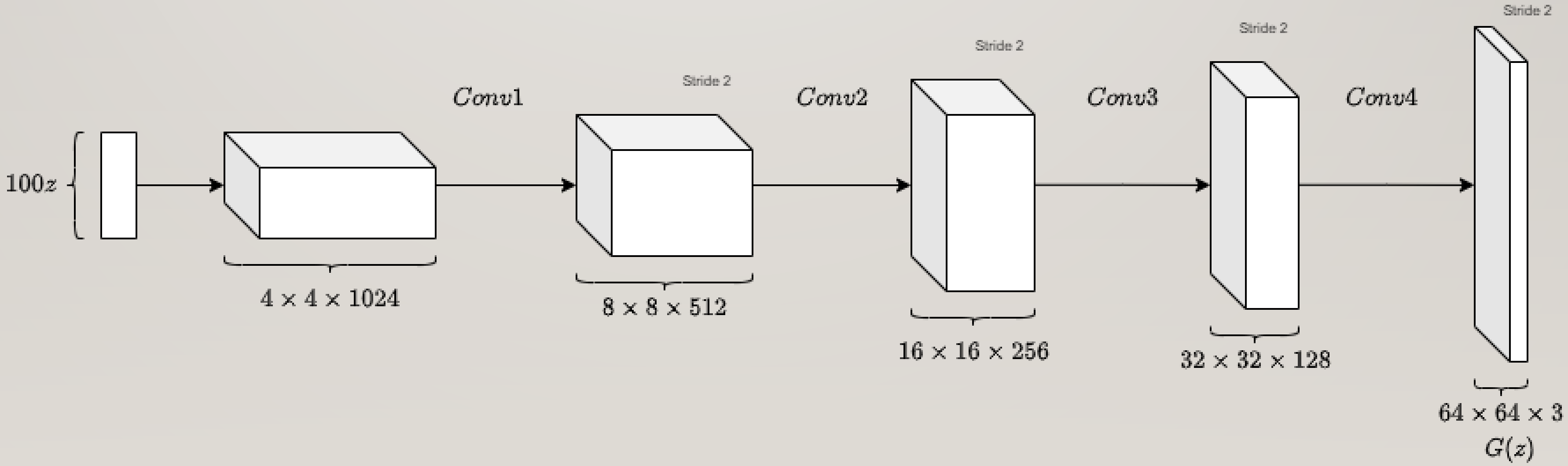
    $$\log D(x)$$

- Loss function:

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

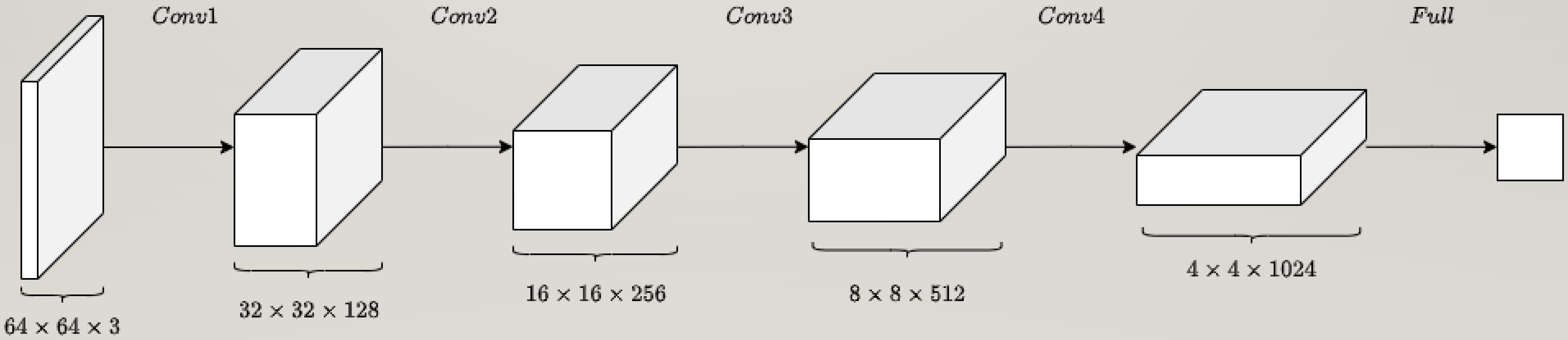
METHODOLOGY

- Generator architecture:



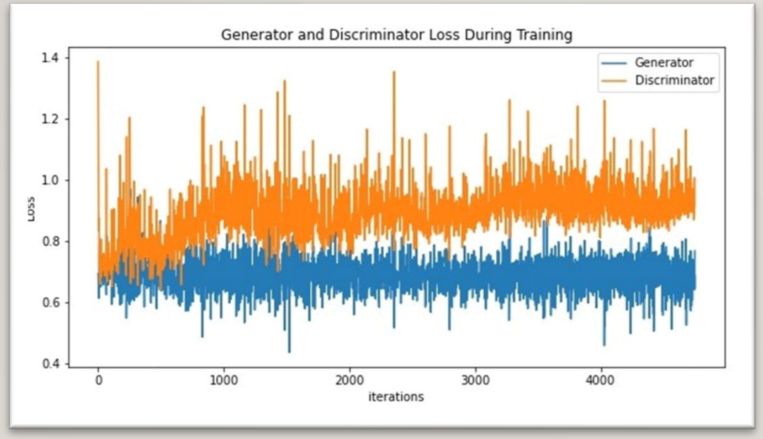
METHODOLOGY

- Discriminator architecture

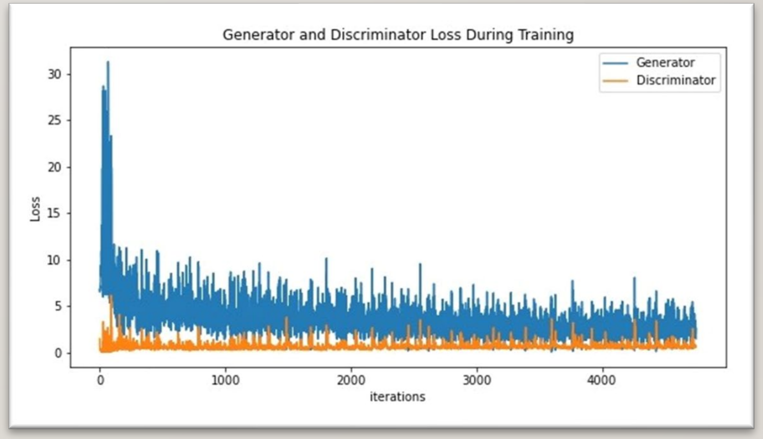


EXPERIMENT AND RESULTS

- Ablation experiment on implementation of batch normalization after convolutional and convolutional-transpose layers.



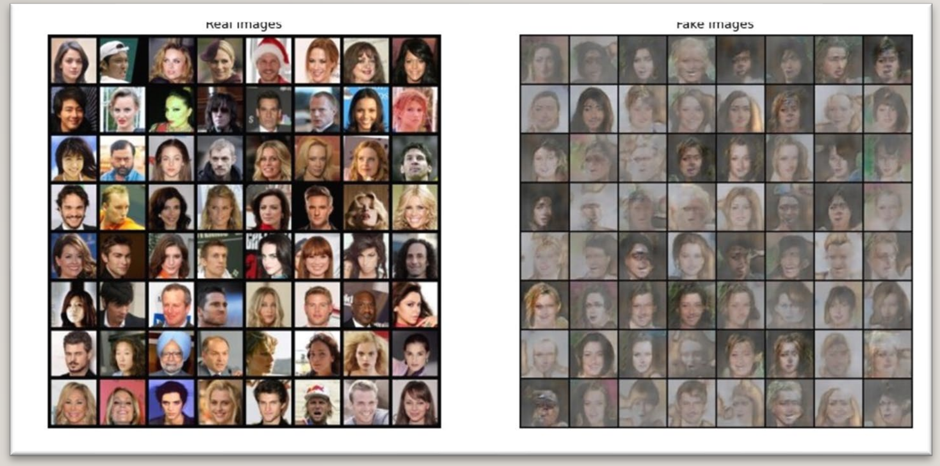
a) Training loss without batch norm



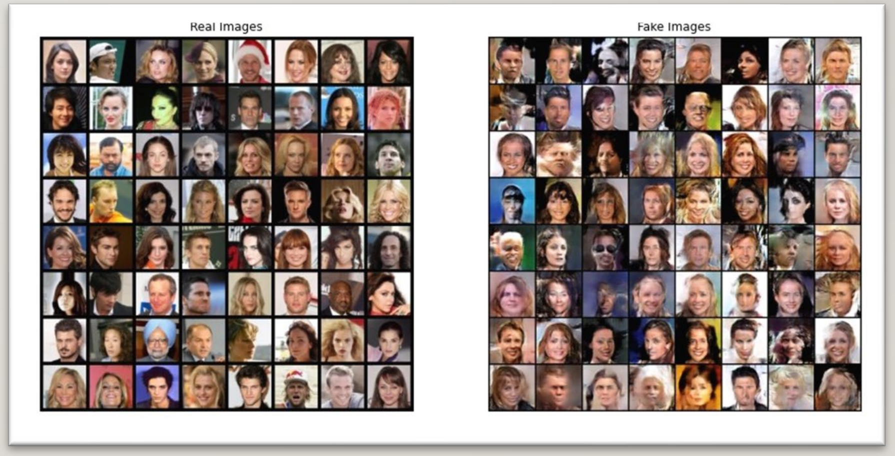
b) Training loss with batch norm

EXPERIMENT AND RESULTS

- Ablation experiment on implementation of batch normalization after convolutional-transpose layers.



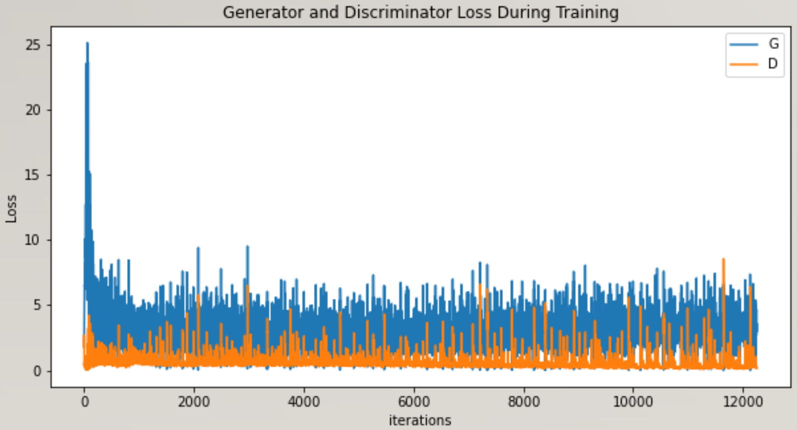
a) Generated images without batch norm



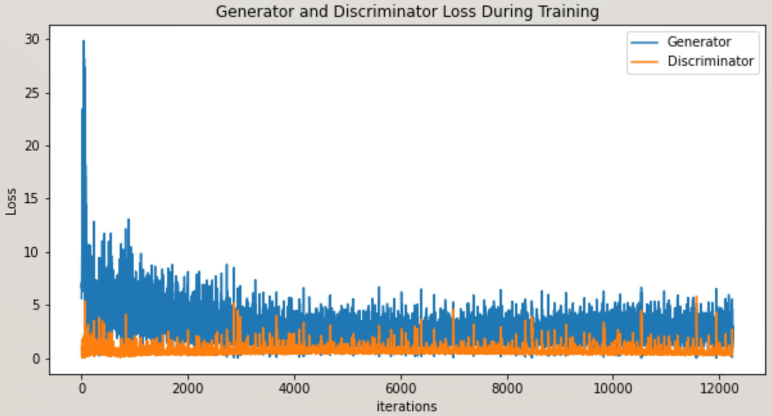
b) Generated images with batch norm

EXPERIMENT AND RESULTS

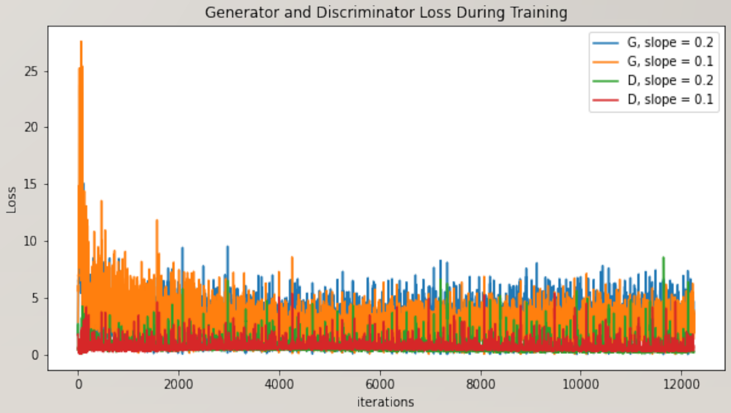
- Experiment on activation function of generator:
  - ReLU or Leaky ReLU;
  - manipulating slope of Leaky ReLU;



a) Training loss with ReLU



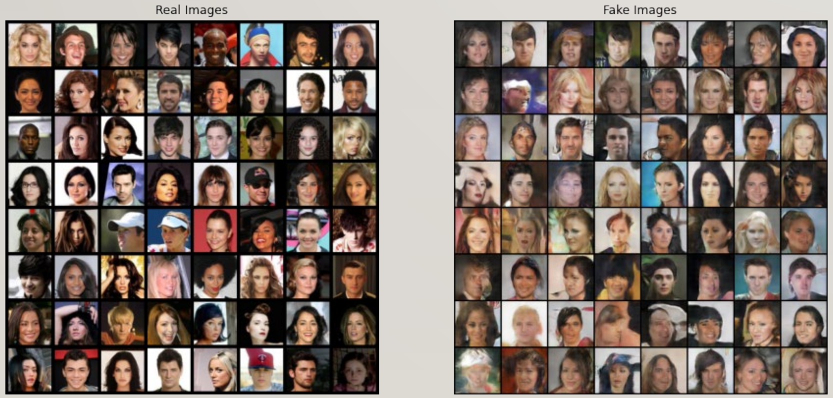
b) Training loss with leaky ReLU, slope = 0.2



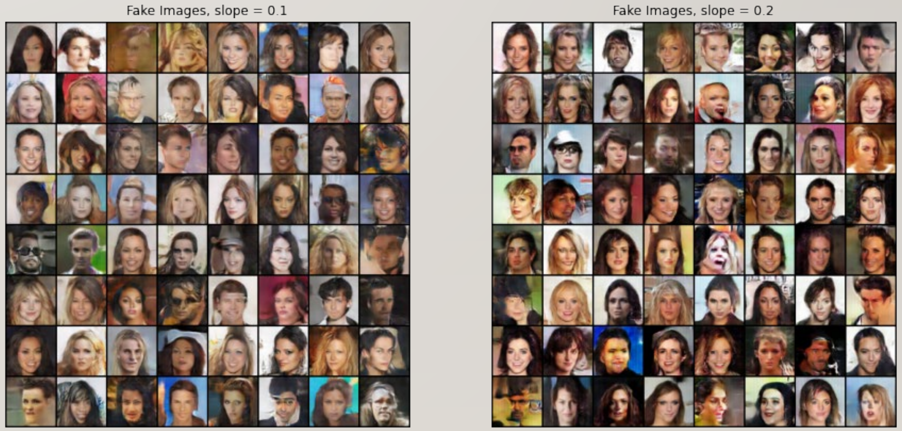
c) Training loss with leaky ReLU, slopes = 0.1 and 0.2

EXPERIMENT AND RESULTS

- Ablation experiment on activation function of generator:
  - ReLU or Leaky ReLU;
  - manipulating slope of Leaky ReLU;



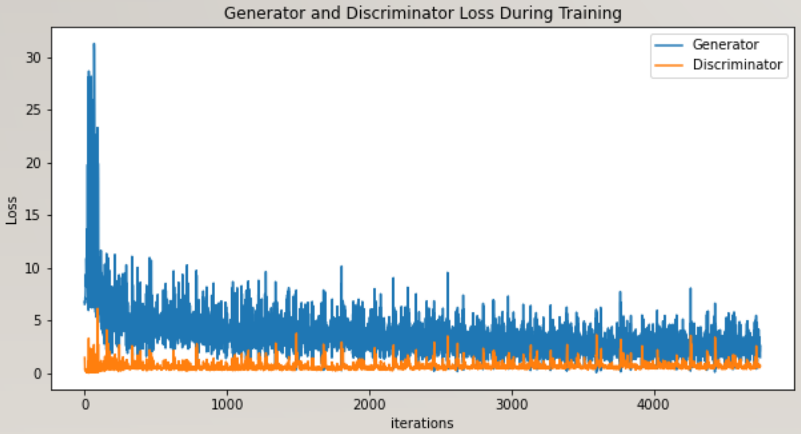
a) Generated images with ReLU



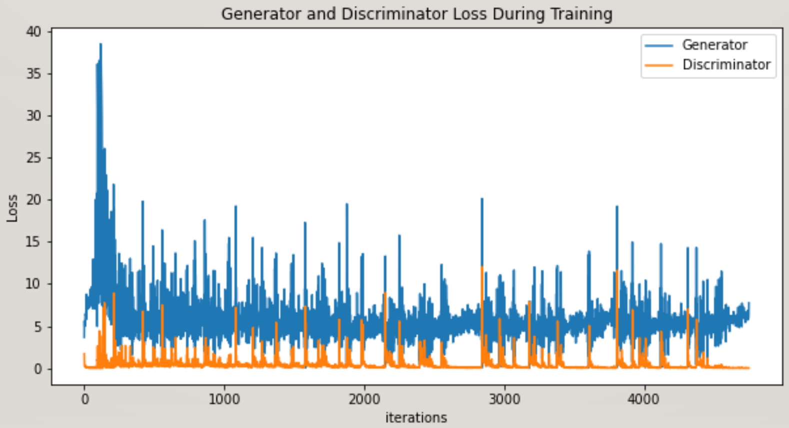
b) Generated images with leaky ReLU in different slopes

EXPERIMENT AND RESULTS

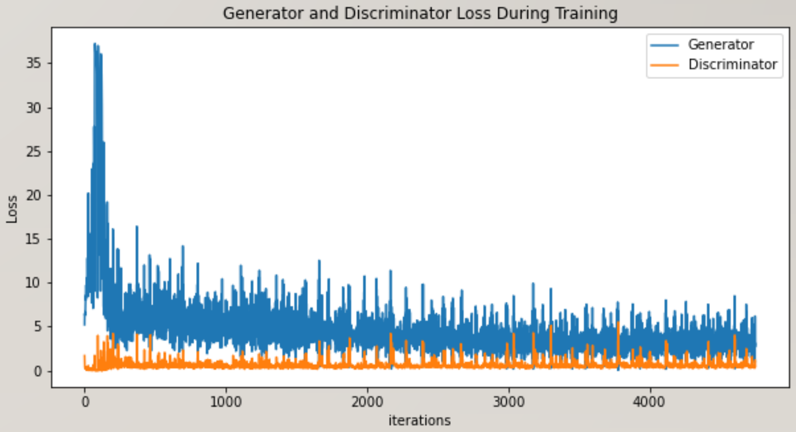
- Experiment on implementation of dropout:
  - with or without dropout;
  - different dropout rate;



a) Training loss without dropout



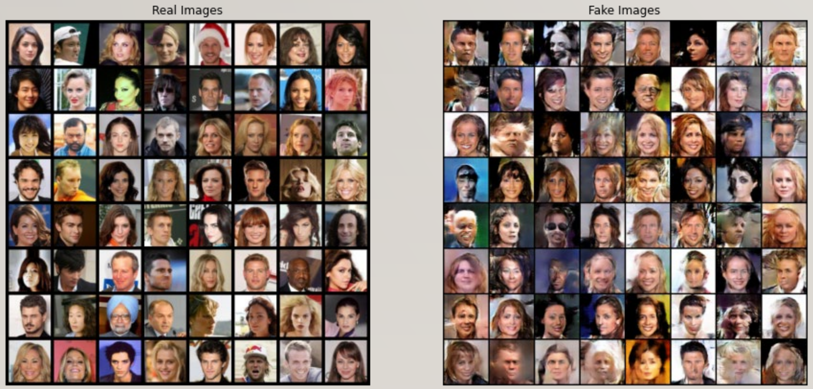
b) Training loss with dropout rate = 0.5



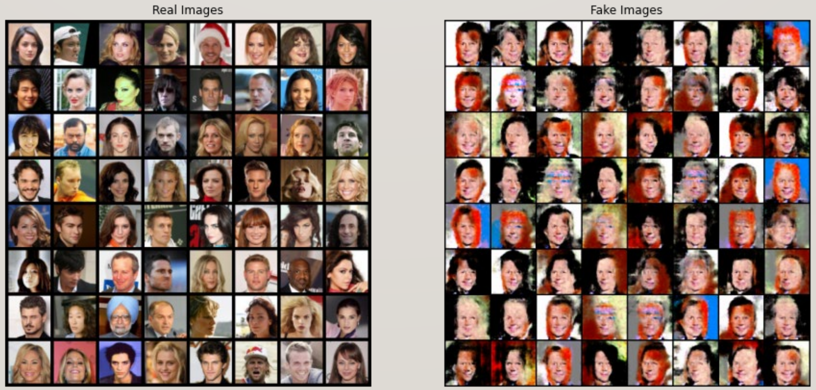
c) Training loss with dropout rate = 0.1

EXPERIMENT AND RESULTS

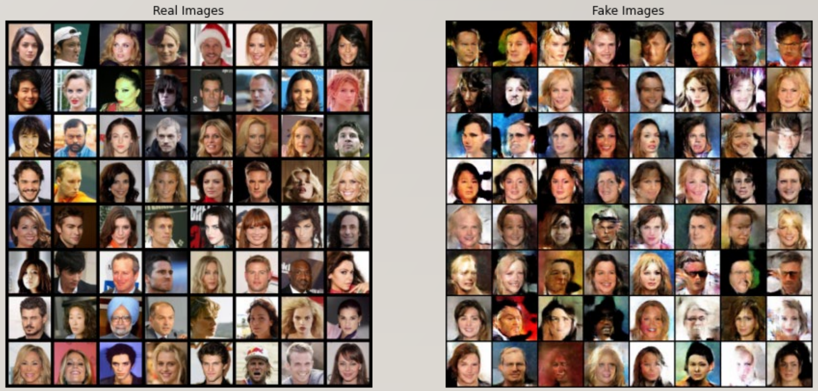
- Experiment on implementation of dropout:
  - with or without dropout;
  - different dropout rate;



a) Generated images without dropout



b) Generated images with dropout rate = 0.5



c) Generated images with dropout rate = 0.1

SUMMARY

- Future exploration: Instance-Conditioned Generative Adversarial Networks (IC-GANs):
  - Inspired from kernel density estimation
  - Non-parametric approach for modeling distribution of complex datasets

THANK YOU!