Kick-Off-Meeting
Master's Thesis

Generative Data Augmentation using
Multi-Agent Diverse Generative Adversarial Networks

Nicolas Reinhart - Mat. Nr. 44100 - 2024/11/27

Agenda

I. Research Topic
   I. Generative Data Augmentation (GDA)
   II. Vanilla GAN
   III. Multi-Agent Diverse GAN
   IV. Classifier with GDA & Research Questions
II. Challenges, Risks & Solutions
III. Technologies

Generative Data Augmentation

Research Topic

Generative Data Augmentation

Research Topic

Definition:

Generative Data Augmentation (GDA) is the process of expanding a training dataset by generating synthetic labelled samples, typically using conditional models. Generally, the aim is to improve classification performance of supervised learning models.

GDA is applied widely in image, text and audio domains, with a focus on generating realistic data for training.

Benefits of GDA

Research Topic

1. Improve Performance

   GDA can enhance classification accuracy, especially for scenarios with limited access to training data

2. Overfitting Mitigation

   GDA is effective in reducing overfitting, when the training set is small

3. Versatility

   GDA is applicable in multiple domains

S: 1, 2, 3

Drawbacks of GDA

Research Topic

1. Effectiveness

   GDA may not yield better results, especially when the training set already is not sufficient (size, intra/inter class variability)

2. Manual Tuning

   Number of augmentation datapoints significantly impacts the classification performance and often requires tuning

3. Negative Impact

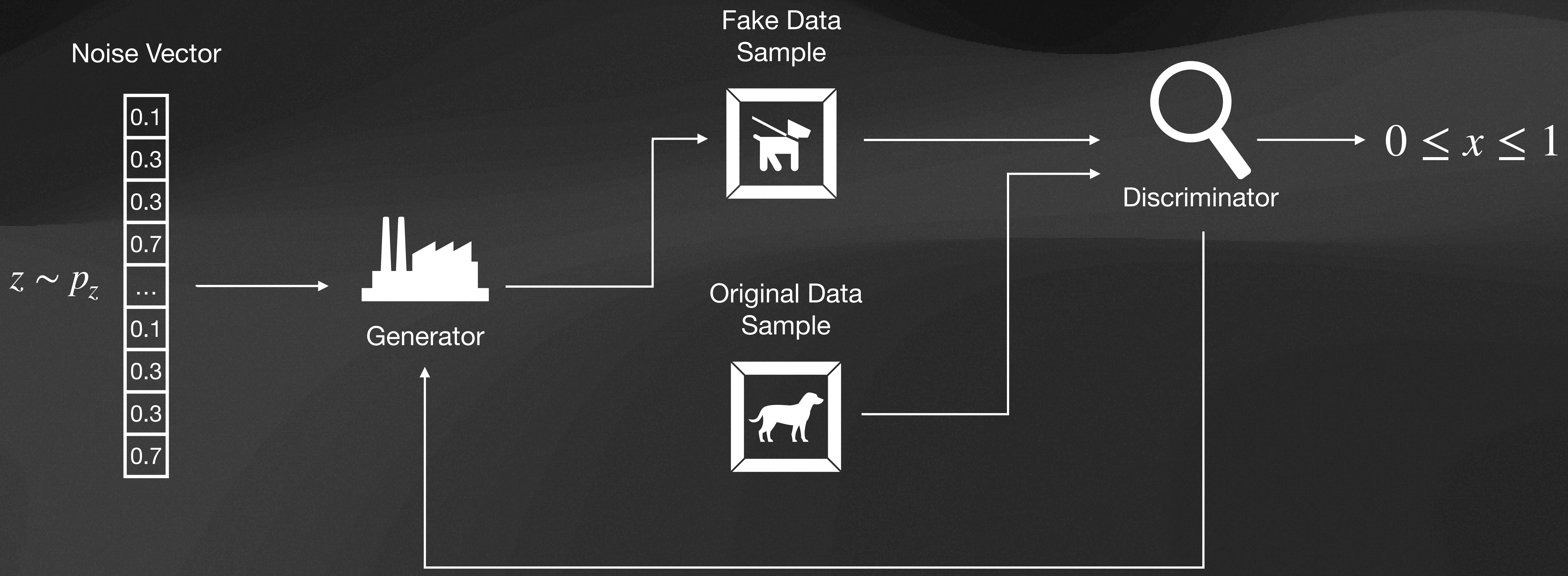
   With abundant training data, GDA may harm the generalization performance

Vanilla GAN's

Research Topic

Vanilla GAN - Training

Research Topic



Vanilla GAN - Objective Functions

Research Topic

Discriminator

maximize:

$$\mathbb{E}_{x \sim p_d} \log D(x; \theta_d) + \mathbb{E}_{z \sim p_z} \log(1 - D(G(z; \theta_g); \theta_d))$$

Generator

minimize:

$$\mathbb{E}_{z \sim p_z} \log(1 - D(G(z; \theta_g); \theta_d))$$

Theoretical equilibrium at

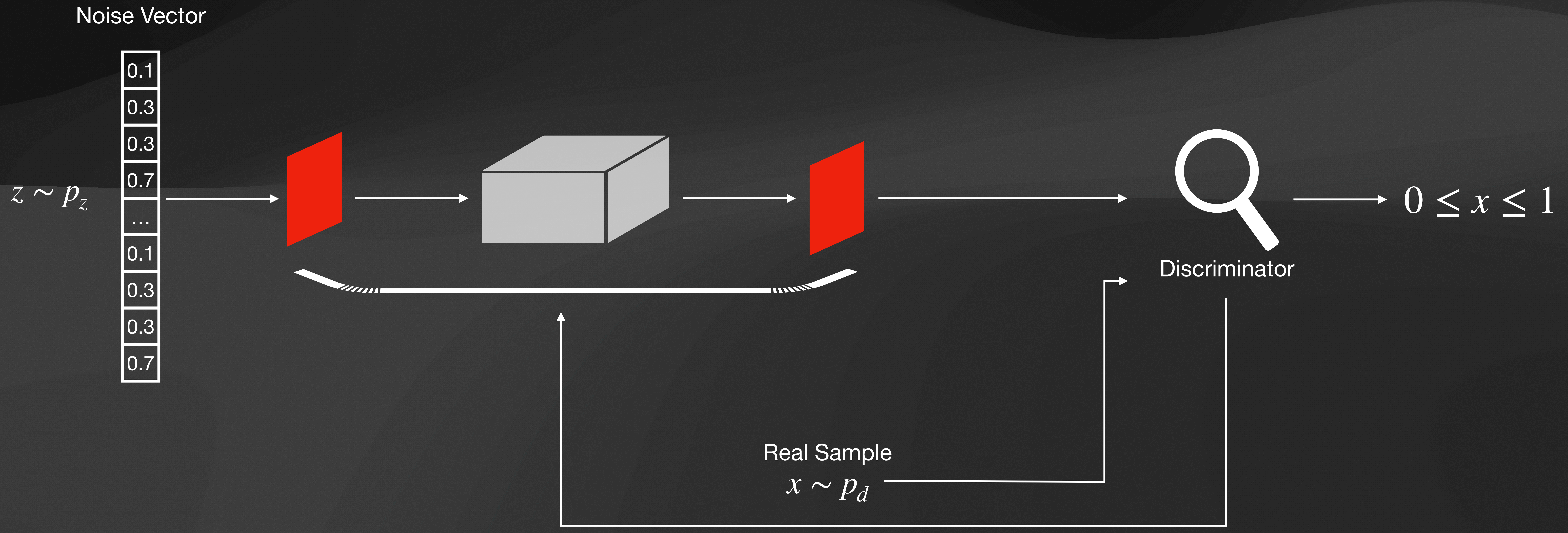
$$p_g = p_d$$

MAD GAN’s

Research Topic

Multi-Agent Diverse GAN - Training

Research Topic



S: 5

Multi-Agent Diverse GAN - Training

Research Topic

S: 5

MAD GAN - Objective Functions

Research Topic

Discriminator

maximize:

$$\max_{\theta_d} \mathbb{E}_{x \sim p} H(\delta, D(x, \theta_d))$$

with δ as Dirac delta distribution:

$$\delta \in \{0,1\}^{k+1}, j \in \{1,...,k\}$$

if sample originates from G_j:

$$\delta(j) = 1$$

if sample originates from p_d:

$$\delta(k+1) = 1$$

Generator

minimize:

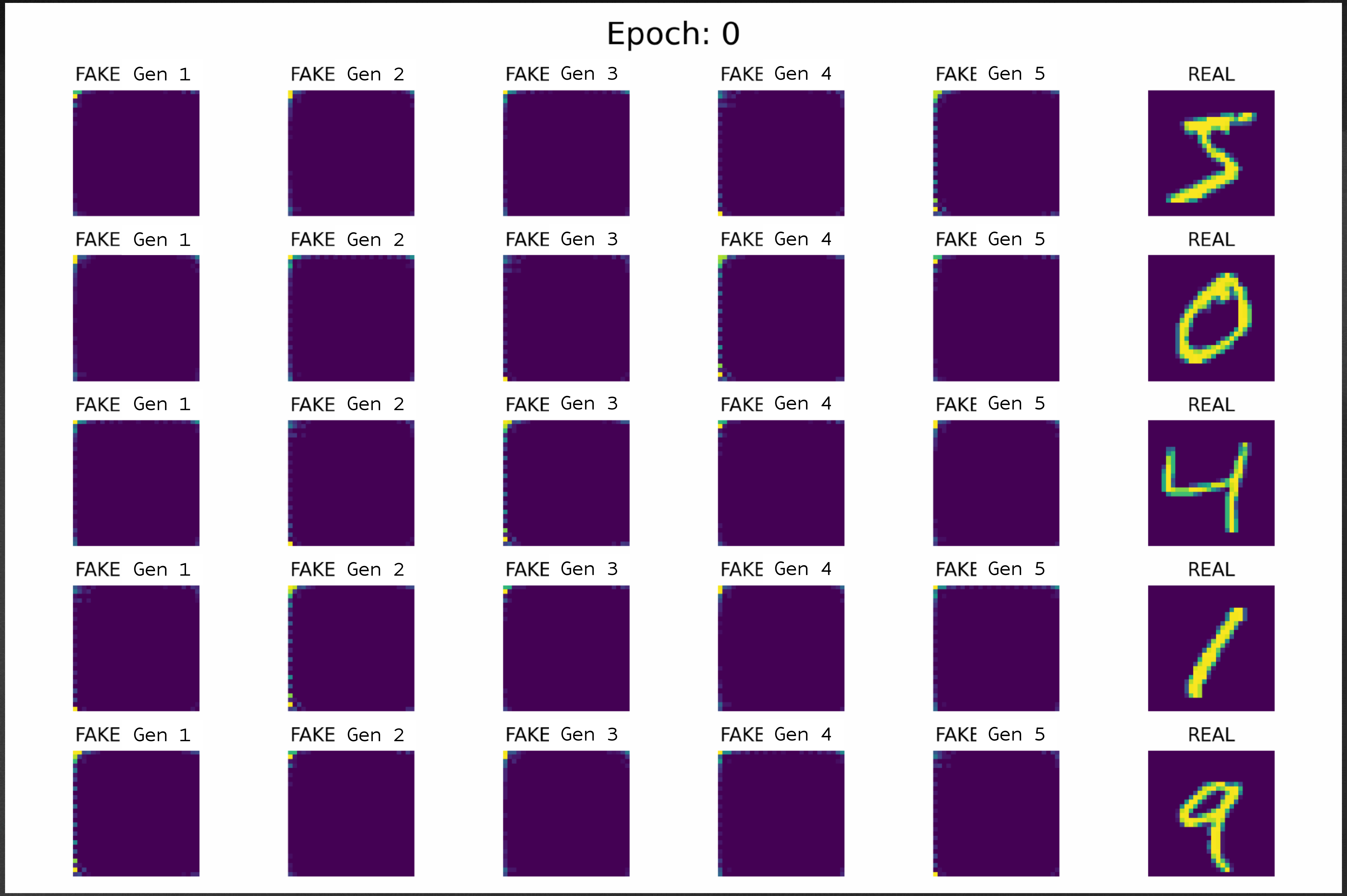
$$\mathbb{E}_{z \sim p_z} \log(1 - D(G(z; \theta_g); \theta_d))$$

Theoretical equilibrium at

$$p_g = p_d$$

Multi-Agent Diverse GAN - Prototype

Research Topic

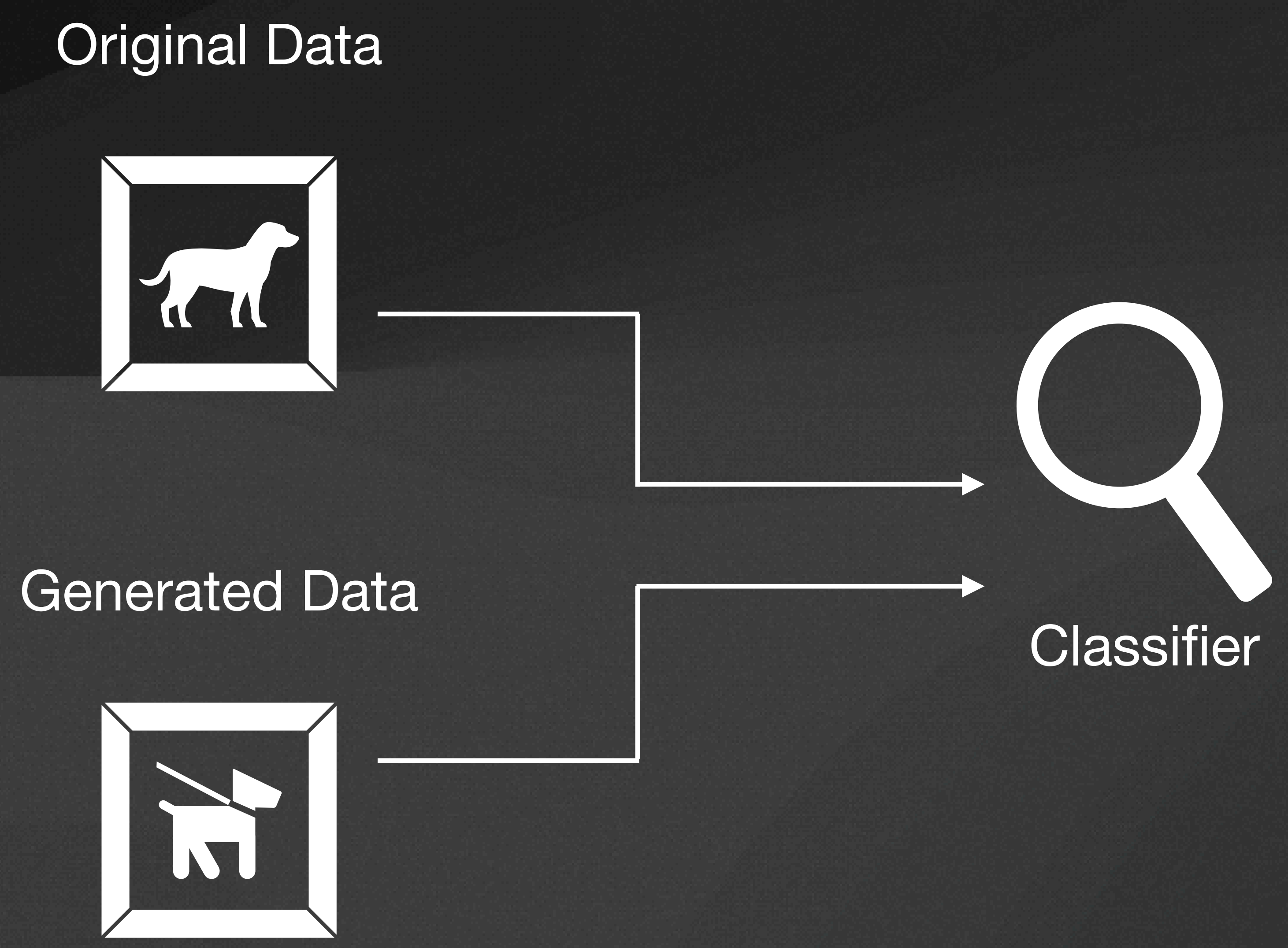


Classifier with GDA

Research Topic

Classifier with GDA & Questions

Research Topic



How does MAD-GAN GDA influence the performance of classifiers on image data?

1. Influence of different loss functions on generated images?

2. Affect of number of generators in the MAD-GAN?

3. Ratio between real and fake samples?

4. Comparison between MAD-GAN GDA to classic data augmentation & Vanilla GAN GDA

Challenges & Risks

Challenges

Challenges, Risks & Solutions

1. Conditionality Constraint

   Combining the MAD-GAN's implementation with the conditionality constraint of CGAN's

2. High Computational Costs

   Using high-definition training samples will result in long training times and number of generators scales linearly with GRAM required

3. Sensitivity to Hyperparameters

   Performance of GAN's is generally highly dependent on *optimal* hyperparameters

Risks

Challenges, Risks & Solutions

1. Conditionality Constraint

   One may not be able to combine the two objective functions correctly and effectively

2. Diminished Classifier Performance

   Generated images may introduce noise or artefacts, degrading the classifier’s accuracy and generalization capability

3. Proving Statistical Significance of Improvements

   Potential improvements in a subsequent classifier may be due to chance, misleading conclusions

Solutions

Challenges, Risks & Solutions

1. Conditionality Constraint

   A classifier can be used to sort generated images to their corresponding classes – this shall be manually checked.

2. High Computational Costs

   Stick to low-resolution datasets (MNIST, Fashion-MNIST, CIFAR-10 / -100)

3. Sensitivity to Hyperparameters

   Automated Hyperparamter Optimization, careful monitoring and early breaking unpromising runs; Curriculum Learning, …

Technologies

Technologies

Programming:

- Important Libraries: anaconda, python, tensorflow, tensorflow-probability, scikit-learn, numpy, pandas, matplot-lib, seaborn, jupyter, PIL

- Software: terminals, jupyter lab, vs code, tex shop

Did you find the straws?

Sources

1. Zheng, C., Wu, G., & Li, C. (2023). Toward understanding generative data augmentation. *arXiv preprint*, arXiv:2305.17476v1
2. Azizi, S., Kornblith, S., Saharia, C., Norouzi, M., & Fleet, D. J. (2023). Synthetic data from diffusion models improves ImageNet classification. *arXiv preprint*, arXiv:2304.08466
3. Besnier, V., Jain, H., Bursuc, A., Cord, M., & Pérez, P. (2020). This dataset does not exist: Training models from generated images. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 1-5)
4. Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative adversarial networks. In *Advances in Neural Information Processing Systems 27 (NIPS 2014)*
5. Ghosh, A., Kulharia, V., Namboodiri, V., Torr, P. H. S., & Dokania, P. K. (2018). Multi-agent diverse generative adversarial networks. *arXiv preprint*, arXiv:1704.02906v2.

Generator Model Definition

```
Model: "generator0"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_0 (InputLayer)        [(None, 256)]             0

 dense_3 (Dense)             (None, 12544)             3211264

 batch_normalization_3 (Batc  (None, 12544)            50176
 hNormalization)

 leaky_re_lu_7 (LeakyReLU)   (None, 12544)             0

 reshape_1 (Reshape)         (None, 7, 7, 256)         0

 conv2d_transpose_7 (Conv2DT  (None, 7, 7, 128)        819200
 ranspose)

 batch_normalization_4 (Batc  (None, 7, 7, 128)        512
 hNormalization)

 leaky_re_lu_8 (LeakyReLU)   (None, 7, 7, 128)         0

 conv2d_transpose_8 (Conv2DT  (None, 14, 14, 64)       204800
 ranspose)

 batch_normalization_5 (Batc  (None, 14, 14, 64)       256
 hNormalization)

 leaky_re_lu_9 (LeakyReLU)   (None, 14, 14, 64)        0

 conv2d_transpose_9 (Conv2DT  (None, 28, 28, 1)        1600
 ranspose)

=================================================================
Total params: 4,287,808
Trainable params: 4,262,336
Non-trainable params: 25,472
```

Discriminator Model Definition

```
Model: "Discriminator"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_2 (InputLayer)        [(None, 28, 28, 1)]       0

 conv2d_2 (Conv2D)           (None, 14, 14, 64)        1664

 leaky_re_lu_5 (LeakyReLU)   (None, 14, 14, 64)        0

 dropout_2 (Dropout)         (None, 14, 14, 64)        0

 conv2d_3 (Conv2D)           (None, 7, 7, 128)         204928

 leaky_re_lu_6 (LeakyReLU)   (None, 7, 7, 128)         0

 dropout_3 (Dropout)         (None, 7, 7, 128)         0

 flatten_1 (Flatten)         (None, 6272)              0

 dense_2 (Dense)             (None, 6)                 37638

=================================================================
Total params: 244,230
Trainable params: 244,230
Non-trainable params: 0
```

Model definitions for generator’s (generator 0) and the discriminator. These definitions are with respect to the prototype example, shown on the slide: “Multi-Agent Diverse GAN - Prototype ”