Generative Data Augmentation

Multi-Agent Diverse Generative Adversarial Networks for Generative Data Augmentation

Under the supervision of:

Prof. Dr.-Ing. Johannes Maucher & Prof. Dr.-Ing. Oliver Kretzschmar

2025-06-26

Author: Nicolas Reinhart, nr063, 44100

Data Augmentation

Traditional Data Augmentation

Definition on Image Data:

Traditional Data Augmentation (TDA), is the appliance of a transformation to alter an image that preserve the respective label, either replacing of expanding training data.

Mathematical formulation:

$$\text{TDA} : (X, f) \mapsto \tilde{X}, \quad f \in \mathscr{F}$$

Generative Data Augmentation

Definition on Image Data:

Generative Data Augmentation (GDA) is the process of expanding a training dataset by generating synthetic labelled samples, typically using conditional models G_c. In case of an unconditional generative model G, auxiliary classifiers are used to label the generated images.

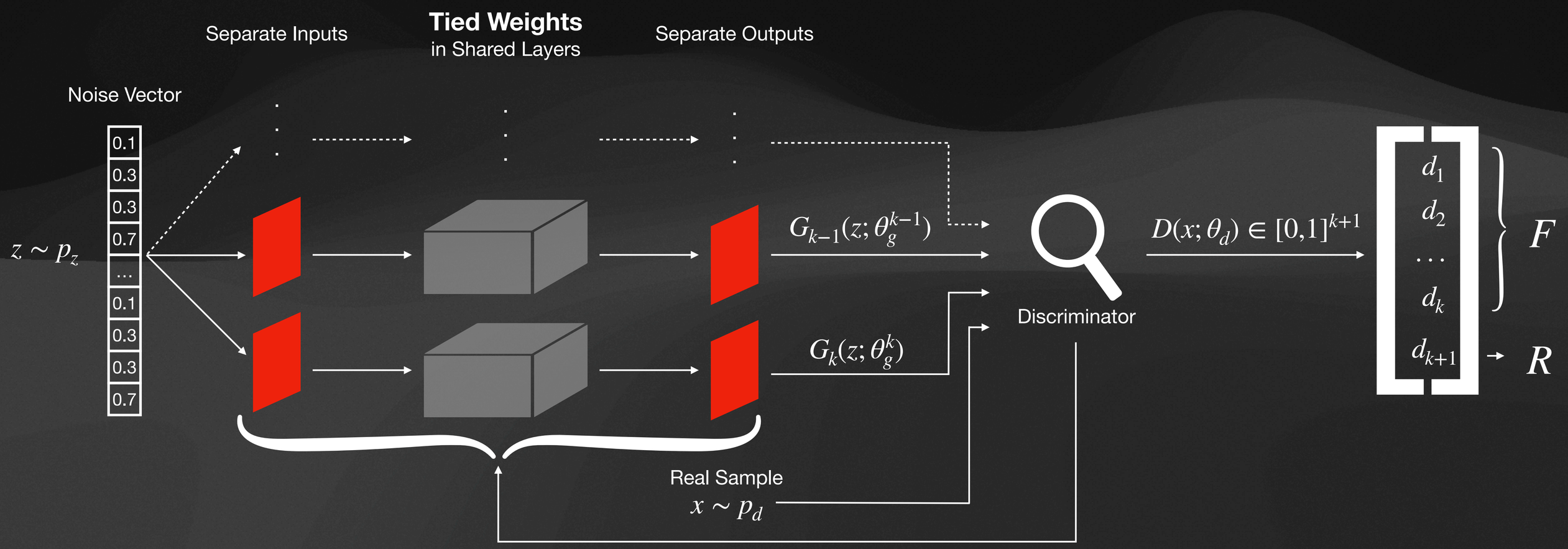
Mathematical formulation:

$$\text{Unconditional: } G : z \mapsto X_{fake}, \quad X_{fake} \sim P_G(X_{fake}) \approx P(X)$$

$$\text{Conditional: } G_c : (z, c) \mapsto X_{fake}, \quad X_{fake} \sim P_{G_c}(X_{fake} \mid c) \approx P(X \mid c)$$

Multi-Agent GAN Architectures

MADGAN


Separate Inputs
Tied Weights
in Shared Layers
Separate Outputs
Noise Vector
0.1
0.3
0.3
0.7
...
0.1
0.3
0.3
0.7
$z \sim p_z$
$G_{k-1}(z; \theta_g^{k-1})$
$G_k(z; \theta_g^k)$
Discriminator
$D(x; \theta_d) \in [0,1]^{k+1}$
d_1
d_2
...
d_k
d_{k+1}
F
R
Real Sample
$x \sim p_d$

Mathematics

MADGAN

Generalised Generator Loss:

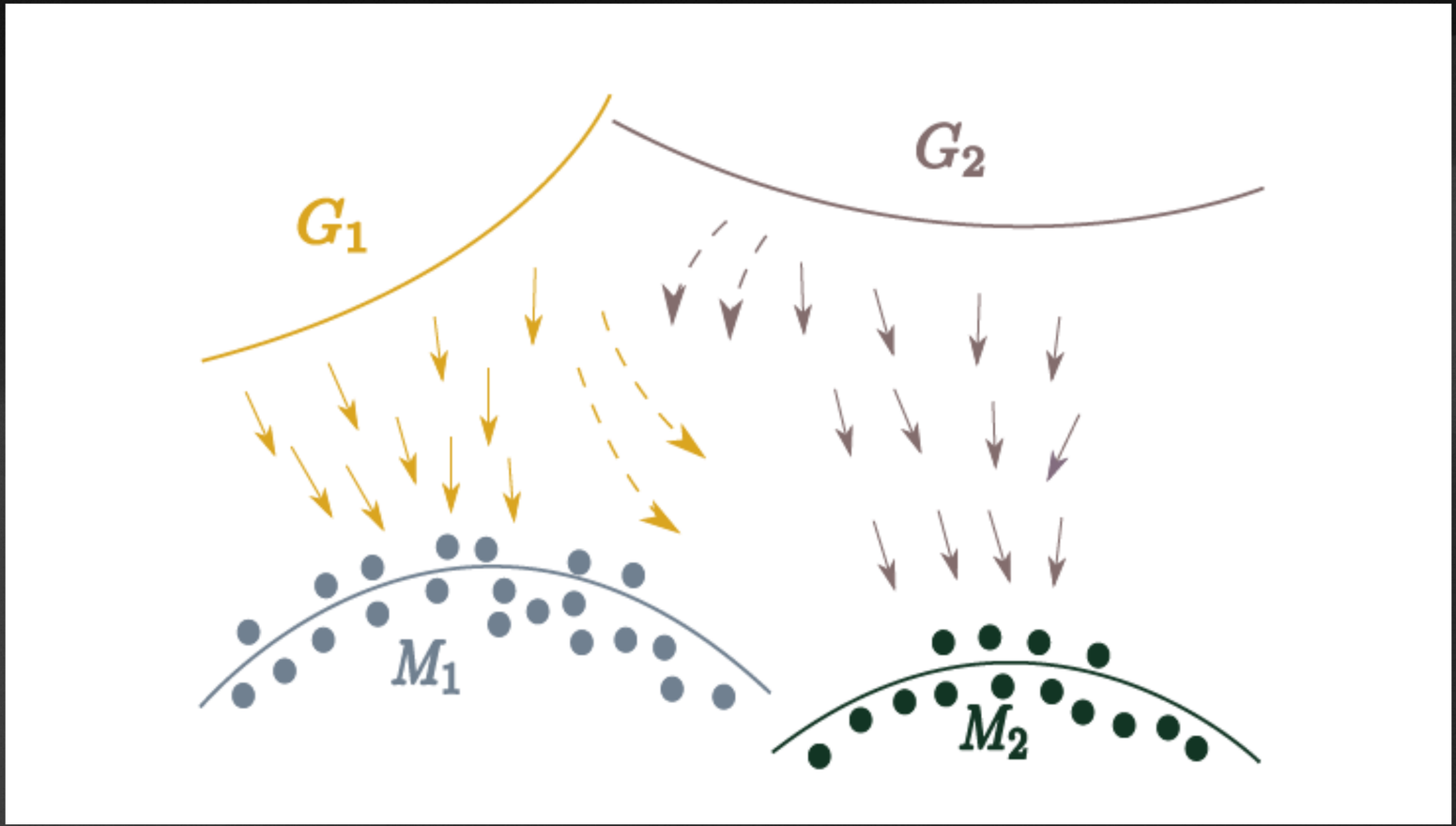
$$G_{[0\ldots k]}: \mathscr{L}_{Gen_{[0\ldots k]}} = \mathbb{E}_{x\sim P_{data}}[\log D_{k+1}(X)] + \sum_{i=1}^{k} \mathbb{E}_{x\sim p_{g_i}}[\log(1 - D_{k+1}(x))]$$

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

- $H(\ldots)$: Negative Cross-Entropy Function
- δ: Dirac Delta Distribution

Enforcing Diverse Modes

MADGAN



Source 1: Ghosh et al. 2018, *Multi-Agent Diverse Generative Adversarial Networks,* page 4

Architectural Adjustments

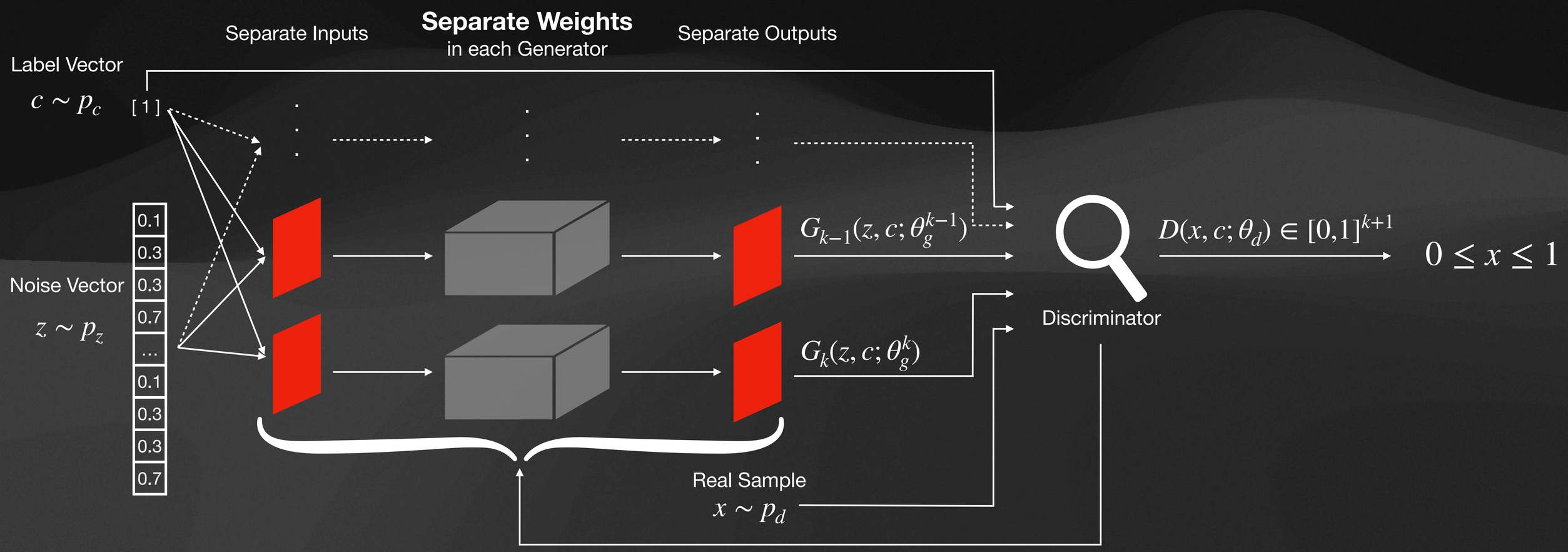
MADGAN

1. **Multiple Generators:**
   The architecture employs k generators.

2. **Parameter Sharing among Generators:**
   The k generators share all but their first and last layers.

3. **Modified Discriminator Objective:**
   D no longer only differentiates between real and fake samples. Instead it has $k+1$ outputs. $[0...k]$ identifying the generators, $k+1$ indicating a sample from P_{data}.

cMADGAN


Separate Inputs
Separate Weights
in each Generator
Separate Outputs
Label Vector
$c \sim p_c$
[1]
Noise Vector
$z \sim p_z$
0.1
0.3
0.3
0.7
...
0.1
0.3
0.3
0.7
$G_{k-1}(z, c; \theta_g^{k-1})$
$G_k(z, c; \theta_g^k)$
Real Sample
$x \sim p_d$
Discriminator
$D(x, c; \theta_d) \in [0,1]^{k+1}$
$0 \leq x \leq 1$

Mathematics

cMADGAN

D: $\mathscr{L}_D = -\mathbb{E}_{x\sim p_{data}(x|c),c\sim p(c)}[\log D(x,c)] - \frac{1}{K}\sum_{i=1}^{K}\mathbb{E}_{z\sim p(z),c\sim p(c)}\left[\log\left(1 - D\left(G_i(z,c),c\right)\right)\right]$

Composed Generator Loss G: $\mathscr{L}_G = \mathscr{L}_{adv} + \lambda_{div}\cdot\mathscr{L}_{div}$

$$\mathscr{L}_G = \underbrace{-\sum_{i=1}^{K}\mathbb{E}_{z\sim p(z),c\sim p(c)}\left[\log\left(D\left(G_i(z,c),c\right)\right)\right]}_{\mathscr{L}_{adv}} \cdot \lambda_{div}\cdot \underbrace{\frac{1}{N_p}\sum_{i=1}^{K}\sum_{j=i+1}^{K}\mathbb{E}_{z\sim p(z),c\sim p(c)}\left[\text{CosSim}\left(G_i(z,c),G_j(z,c)\right)\right]}_{\mathscr{L}_{div}}$$

- CosSim is the Cosinus Similarity, applied to a pair of generators.
- N_p: The unique combinations of generators $\frac{K\cdot(K-1)}{2}$

Architectural Adjustments

cMADGAN

1. **Multiple Generators:**
   The architecture employs k generators, **without weight-sharing**.

2. **Class Conditioning on Inputs:**
   Both models (D and G) receive a class conditioning together with their respective inputs.

3. **Explicit Diversity Enforcement:**
   Diversity is explicitly encouraged, in the loss formulation of G via the application of the Cosinus Similarity between the generator outputs.

Experimental Setup

Datasets

MNIST

Source 2: Wikipedia, *MNIST database,* image by Yann LeCun

Fashion-MNIST

Source 3: Fashion-MNIST GitHub repository by Zalando Research

Both datasets are $28 \times 28 \times 1$. They both have $60k$ training images and $10k$ test images, with 10 classes.

Experimental Workflow

Label Images

2.1

1 → 2 →

Train GAN

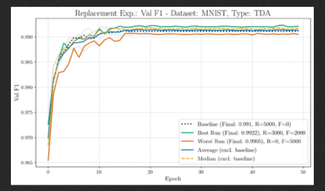
Generate Images

3 → 4

Stratified Classifiers

Evaluation

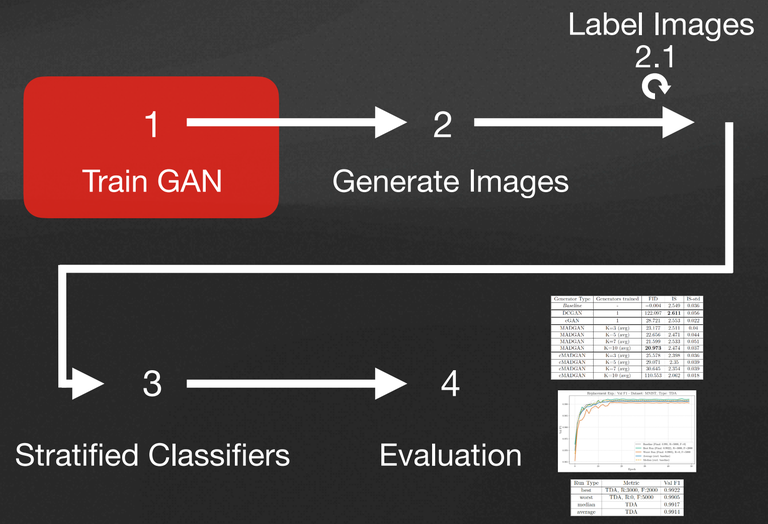
Generator Type	Generators trained	FID	IS	IS-std
Baseline	-	−0.004	2.549	0.036
DCGAN	1	122.097	**2.611**	0.056
cGAN	1	28.721	2.553	0.022
MADGAN	K=3 (avg)	23.177	2.511	0.04
MADGAN	K=5 (avg)	22.656	2.471	0.044
MADGAN	K=7 (avg)	21.599	2.533	0.051
MADGAN	K=10 (avg)	**20.973**	2.474	0.037
cMADGAN	K=3 (avg)	25.578	2.398	0.036
cMADGAN	K=5 (avg)	29.071	2.35	0.039
cMADGAN	K=7 (avg)	30.645	2.354	0.039
cMADGAN	K=10 (avg)	110.553	2.062	0.018



Run Type	Metric	Val F1
best	TDA, R:3000, F:2000	0.9922
worst	TDA, R:0, F:5000	0.9905
median	TDA	0.9917
average	TDA	0.9914

Experimental Workflow

1. **Train a GAN Architecture:** The respective GAN architectures are trained for a afore evaluated number of epochs.
   - <u>DCGAN & cGAN:</u> 1 Generator
   - <u>MADGAN & cMADGAN:</u> 3, 5, 7, 10 Generators.

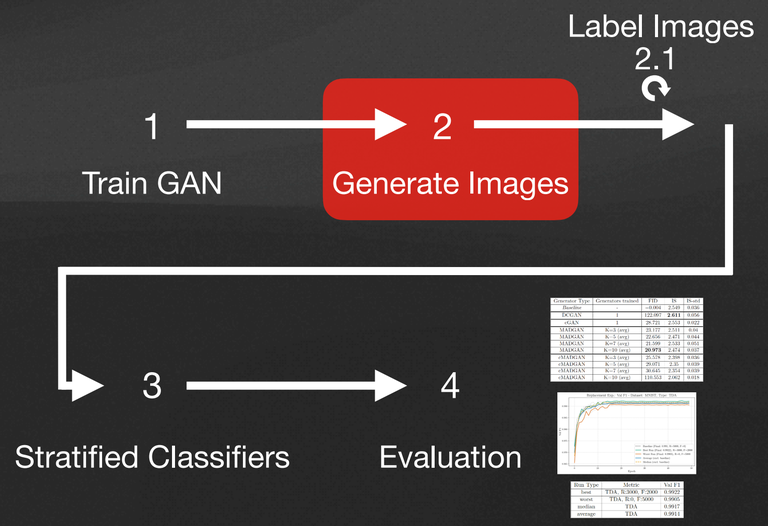


Generator Type	Generators Trained	FID	IS	Dual-I
Baseline		*0.004*	*2.149*	*0.024*
DCGAN	1	122.087	**2.611**	0.056
cGAN	1	28.721	2.552	0.022
MADGAN	K=3 (avg)	23.177	2.515	0.04
MADGAN	K=5 (avg)	22.656	2.473	0.041
MADGAN	K=7 (avg)	21.599	2.523	0.051
MADGAN	K=10 (avg)	**20.973**	2.474	0.037
cMADGAN	K=3 (avg)	25.578	2.299	0.036
cMADGAN	K=5 (avg)	20.971	2.31	0.037
cMADGAN	K=7 (avg)	30.645	2.354	0.028
cMADGAN	K=10 (avg)	110.513	2.062	0.013

Run Type	Metric	Val F1
best	TDA, K-5000, F-2000	0.9927
worst	TDA, R-5, F-5000	0.9905
median	TDA	0.9917
average	TDA	0.9911

Experimental Workflow

2. **Synthetic Sample Generation:** The trained generators are used to create at least $6k$ fake images per class.
   1. Unconditional samples from the DC- & MADGAN architectures were labeled using pre-trained classifiers.

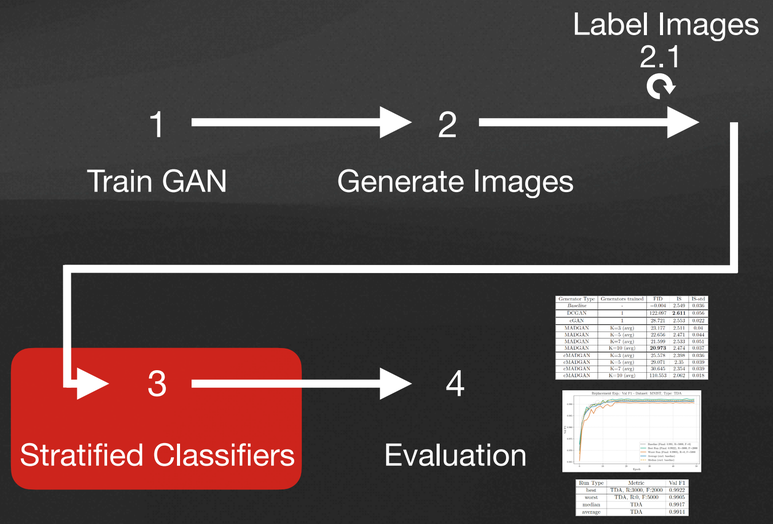


Generator Type	Generators Trained	FID	IS	Dual
Baseline	-	0.004	2.549	0.024
DCGAN	1	122.087	**2.611**	0.056
cGAN	1	28.721	2.552	0.027
MADGAN	K=3 (avg)	23.177	2.521	0.04
MADGAN	K=5 (avg)	22.656	2.471	0.041
MADGAN	K=7 (avg)	21.599	2.523	0.051
MADGAN	K=10 (avg)	**20.973**	2.474	0.025
cMADGAN	K=3 (avg)	25.578	2.209	0.036
cMADGAN	K=5 (avg)	20.971	2.31	0.022
cMADGAN	K=7 (avg)	30.645	2.354	0.028
cMADGAN	K=10 (avg)	119.503	2.062	0.013

Run Type	Metric	Val F1
best	TDA, R.5000, F.5000	0.9927
worst	TDA, R.5, F.5000	0.9905
median	TDA	0.9917
average	TDA	0.9911

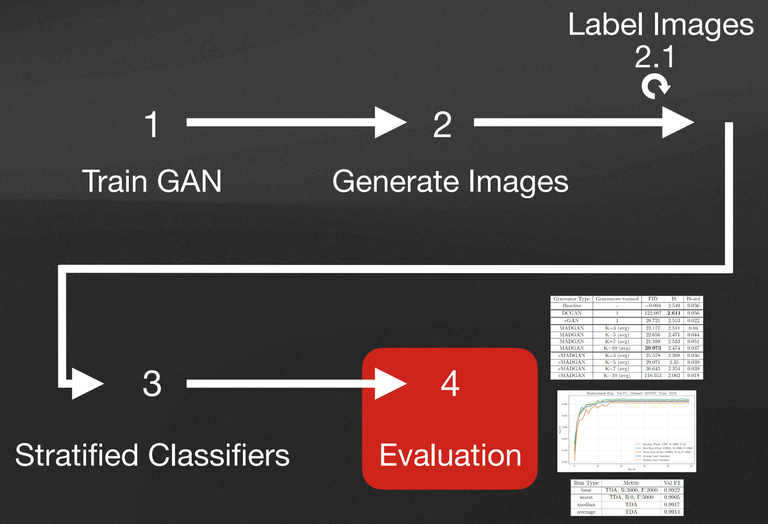
Experimental Workflow

3. **Stratified Classifier Training for each Generator:**
   - <u>Replacement Scenario:</u> Substitute real images with fake images ($5k$ real, 0 fake; $4k$ real, $1k$ fake; …; 0 real, $5k$ fake - per class)
   - <u>Expansion Scenario:</u> Expand the real images with fake images ($5k$ real, 0 fake; $5k$ real, $1k$ fake; …; $5k$ real, $5k$ fake - per class)



Experimental Workflow

4. **Evaluation of Classifiers & Image Quality:**
   - Classifiers: F1 Score
   - Image Quality: Inception Score, Fréchet Inception Distance
   - Only the best multi-generator settings are chosen



Generator Type	Generators Trained	FID	IS	Dual
Baseline	-	0.004	2.549	0.024
DCGAN	1	122.087	**2.611**	0.056
cGAN	1	28.721	2.552	0.027
MADGAN	K=3 (avg)	23.177	2.521	0.04
MADGAN	K=5 (avg)	22.656	2.471	0.041
MADGAN	K=7 (avg)	21.599	2.523	0.051
MADGAN	K=10 (avg)	**20.973**	2.474	0.037
cMADGAN	K=3 (avg)	25.578	2.209	0.036
cMADGAN	K=5 (avg)	20.971	2.31	0.037
cMADGAN	K=7 (avg)	30.645	2.354	0.028
cMADGAN	K=10 (avg)	119.503	2.062	0.013

Run Type	Metric	Val F1
best	TDA, S:3000, F:3000	0.9927
worst	TDA, R:0, F:5000	0.9905
median	TDA	0.9917
average	TDA	0.9911

Metrics

Measured Metrics in Experiments

Metrics

Classification Performance Score:

- F1 Score

Image Scores:

- Inception Score (*Inception V3*)
- Fréchet Inception Distance (*Inception V3*)

F1 Score

Metrics

F1 is the harmonic mean of Precision and Recall:

Precision: $\frac{TP}{TP + FP}$

Recall: $\frac{TP}{TP + FN}$

F1 Score: $2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$

Inception Score

Metrics

The Inception Score (IS) measures quality and diversity by evaluating how confidently a pre-trained classifier predicts labels and how varied the predictions are across the dataset.

Assessed criteria:

- Image Quality: Each image should be classified with high confidence.
  - If each image is confidently classified $p(y \mid x)$ is a low-entropy conditional distribution.
- Diversity: Across a set of generated images, the distribution of predicted labels should span all expected labels.
  - If it covers a wide range of labels $p(y)$ is a high-entropy marginal label distribution.

Inception Score

Metrics

IS Formula: $$IS = \exp\left(\mathbb{E}_{x \sim p_g}\left[D_{KL}\left(p(y \mid x) \| p(y)\right)\right]\right)$$

- D_{KL}: Denotes the Kullback-Leibler Divergence
  - $p(y \mid x)$: conditional class distribution
  - $p(y)$: marginal class distribution

Interpretation: Higher is better!

Fréchet Inception Distance

Metrics

The Fréchet Inception Distance (FID) measures the similarity between real and generated images by comparing their feature distributions using a pre-trained classifier.

Assessed criteria:

- Similarity real vs. generated: Real and generated image distributions are compared in features space.

- Intra-class Diversity: Sensitive to a collapse within a mode (measured by Σ_g) resulting in reduced variance in the captured feature space.

Fréchet Inception Distance

Metrics

FID Formula: $FID = \|\mu_r - \mu_g\|_2^2 + \text{Tr}\left(\Sigma_r + \Sigma_g - 2\left(\Sigma_r \Sigma_g\right)^{1/2}\right)$

- $\| \cdot \|_2^2$: Squared L2 norm (Euclidian Distance)
- Tr($\cdot$) : Trace of a matrix
- μ_r, μ_g : Means of features for real and generated images
- Σ_r, Σ_g: Covariance matrices for real and generated images

Interpretation: Lower is better!

Research Questions & Answers

Q1: How do the FID and the IS compare between the generative Methods?

MNIST

Generator Type	Generators trained	FID	IS	IS-std
Baseline	-	−0.004	2.549	0.036
DCGAN	1	122.097	**2.611**	0.056
cGAN	1	28.721	2.553	0.022
MADGAN	K=3 (avg)	23.177	2.511	0.04
MADGAN	K=5 (avg)	22.656	2.471	0.044
MADGAN	K=7 (avg)	21.599	2.533	0.051
MADGAN	K=10 (avg)	**20.973**	2.474	0.037
cMADGAN	K=3 (avg)	25.578	2.398	0.036
cMADGAN	K=5 (avg)	29.071	2.35	0.039
cMADGAN	K=7 (avg)	30.645	2.354	0.039
cMADGAN	K=10 (avg)	110.553	2.062	0.018

Fashion-MNIST

Generator Type	Generators trained	FID	IS	IS-std
Baseline	-	−0.004	4.723	0.056
DCGAN	1	25.56	4.21	0.099
cGAN	1	123.349	3.573	0.117
MADGAN	K=3 (avg)	26.202	4.496	0.099
MADGAN	K=5 (avg)	24.218	4.497	0.098
MADGAN	K=7 (avg)	23.875	4.523	0.094
MADGAN	K=10 (avg)	**21.587**	4.534	0.097
cMADGAN	K=3 (avg)	25.555	**4.623**	0.105
cMADGAN	K=5 (avg)	160.082	3.346	0.038
cMADGAN	K=7 (avg)	154.115	2.929	0.034
cMADGAN	K=10 (avg)	159.067	3.317	0.035

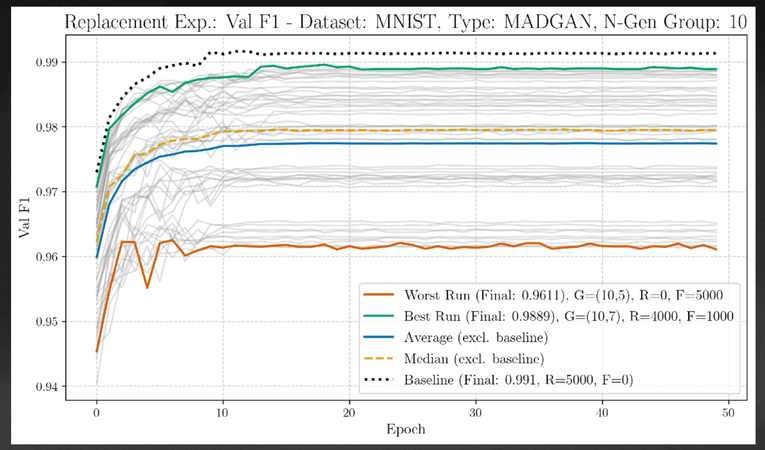
Q2.1 MNIST: Does GDA with MADGAN's enhance downstream classifier performance more effectively than TDA?

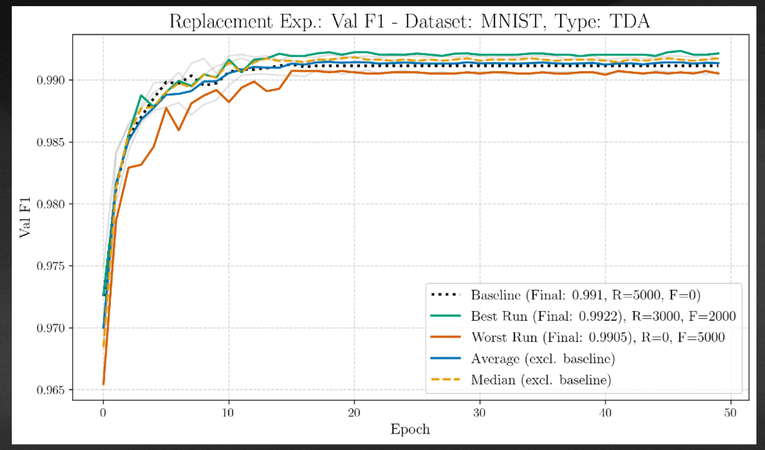
MADGAN

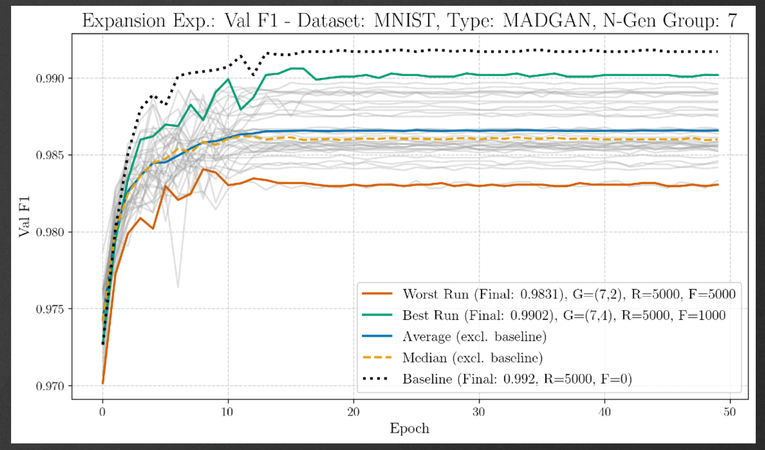
TDA

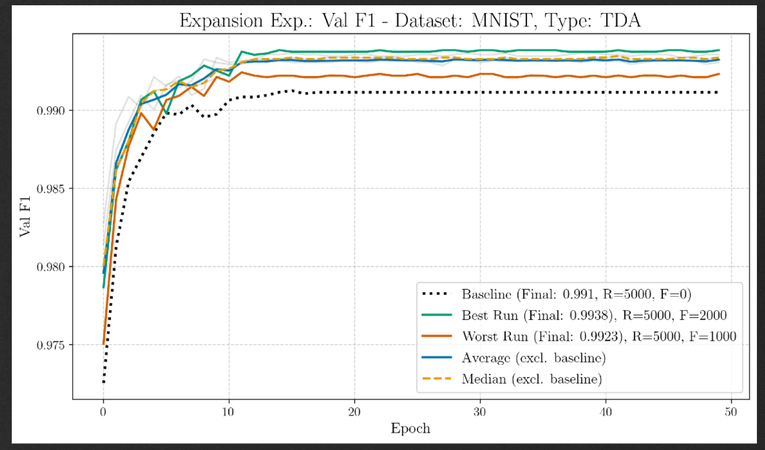
Replacement Experiment

Expansion Experiment







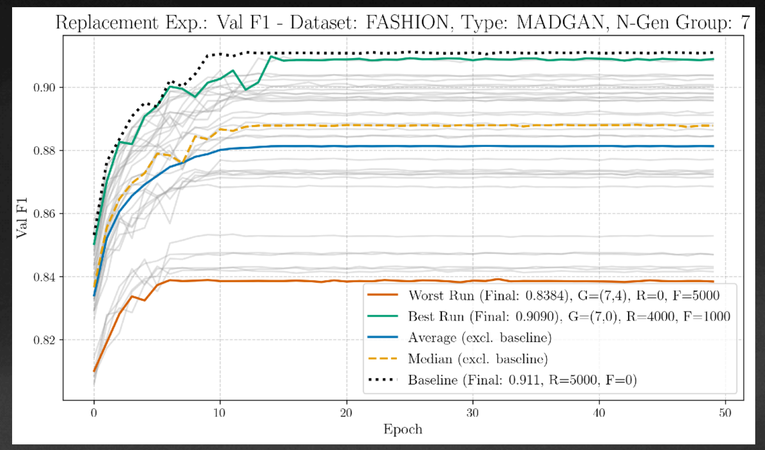


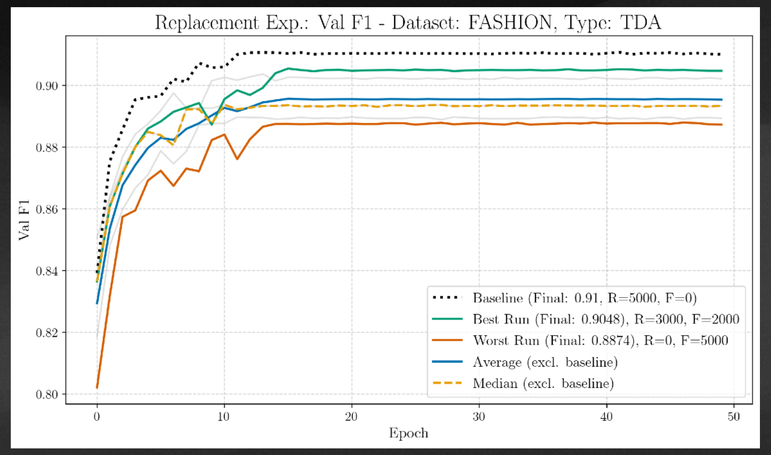
Q2.2 Fashion-MNIST: Q2.1 MNIST: Does GDA with MADGAN's enhance downstream classifier performance more effectively than TDA?

MADGAN

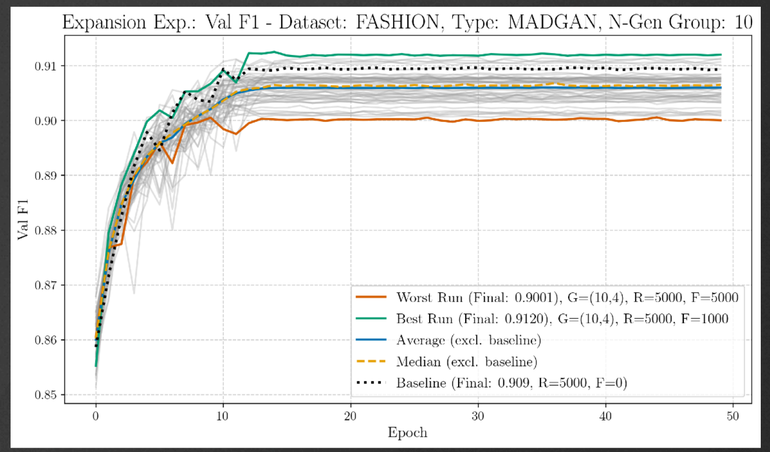
TDA

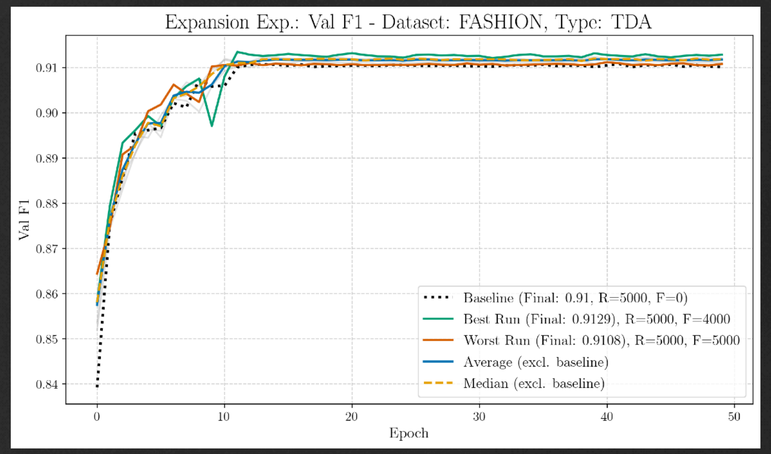
Replacement Experiment





Expansion Experiment





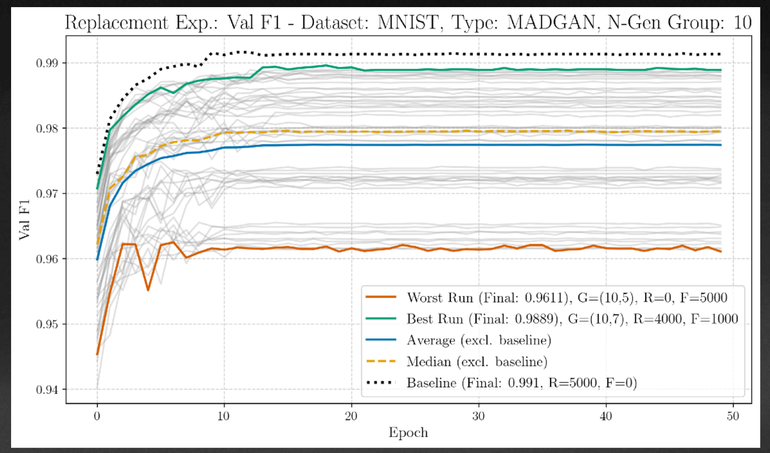
Q3.1 MNIST: How does the performance enhancement achieved with MADGAN-based GDA compare to GDA using DC- and cGAN?

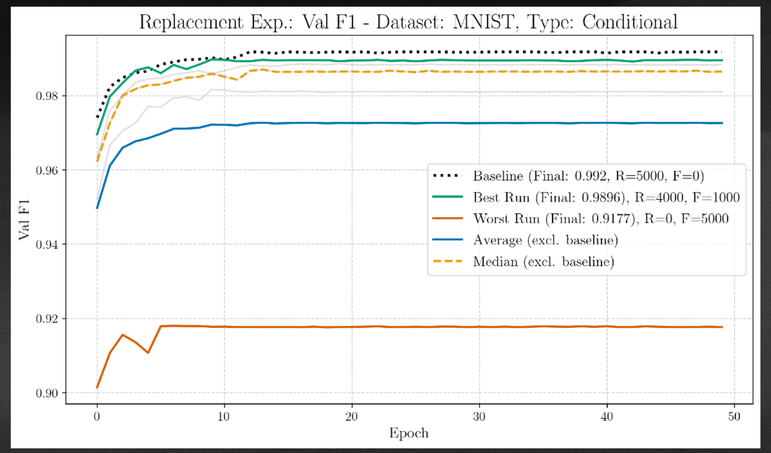
MADGAN

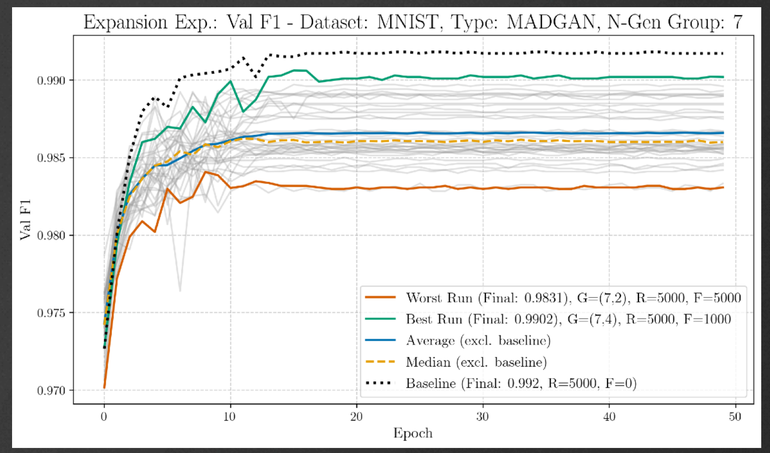
Conditional

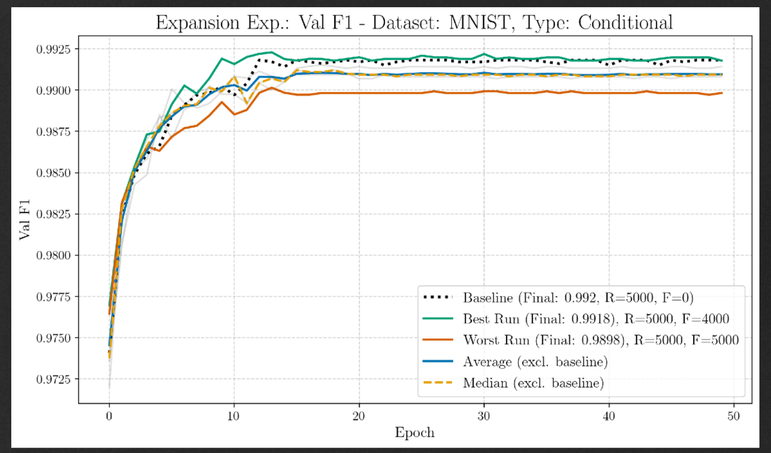
Replacement Experiment

Expansion Experiment







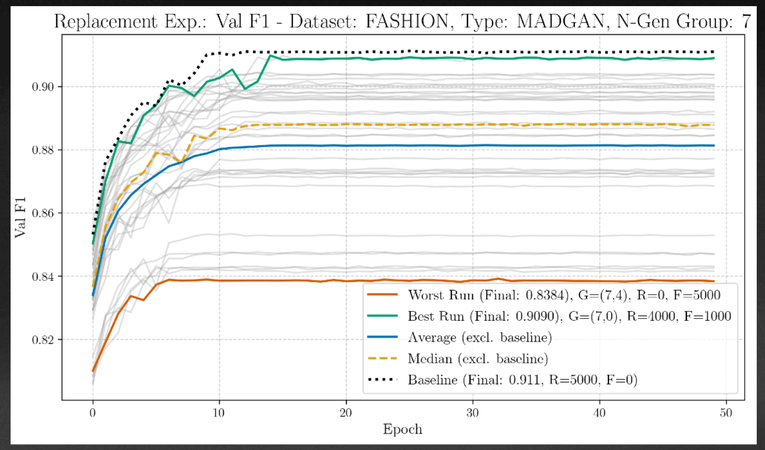


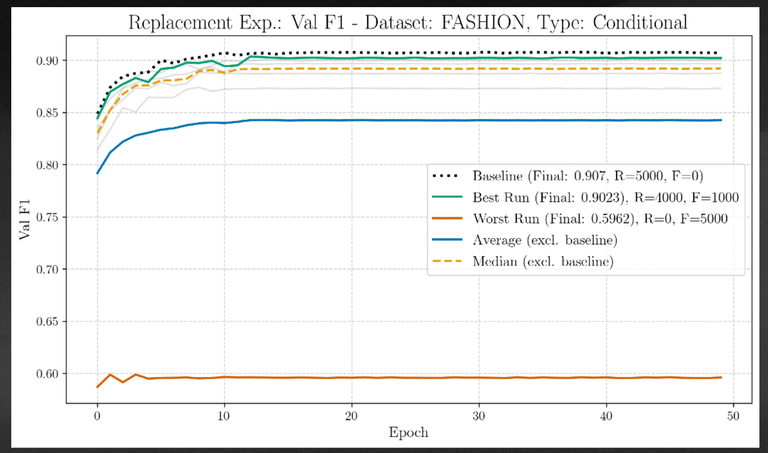
Q3.2 Fashion-MNIST: How does the performance enhancement achieved with MADGAN-based GDA compare to GDA using DC- and cGAN?

MADGAN

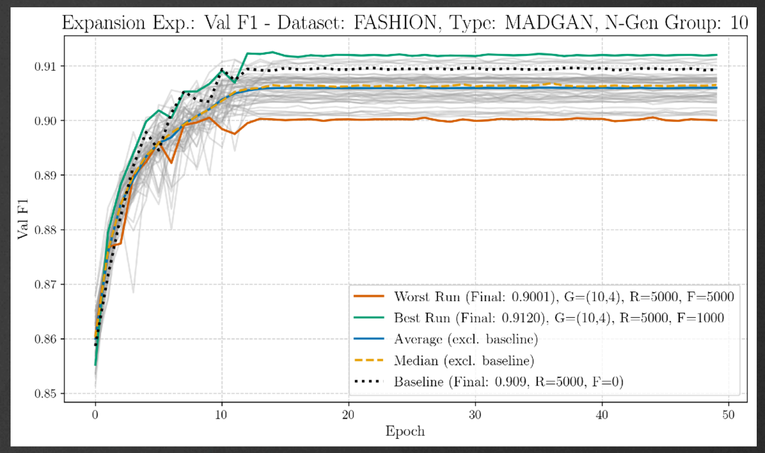
Conditional

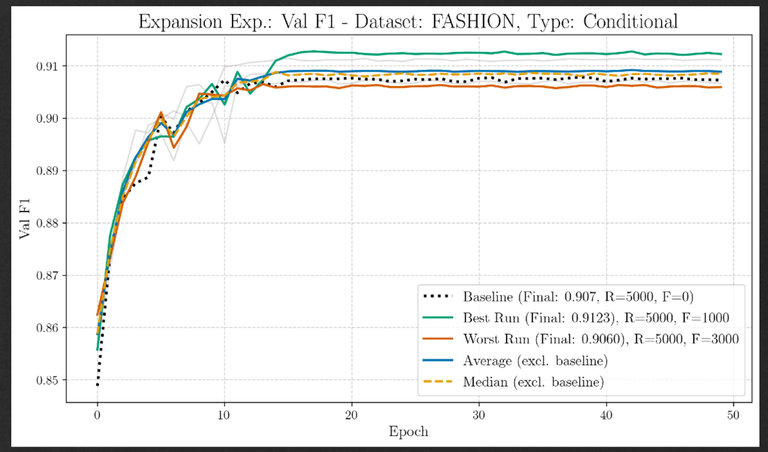
Replacement Experiment





Expansion Experiment



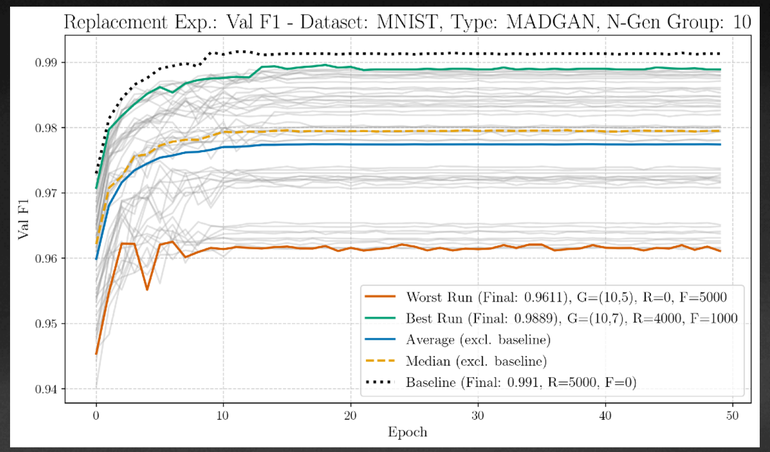


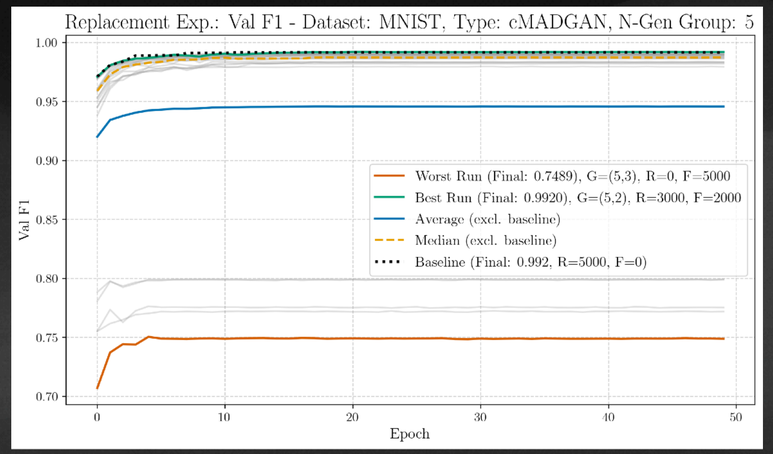
Q4.1 MNIST: How does the performance enhancement achieved with MADGAN-based GDA compare to that of cMADGAN-GDA?

MADGAN

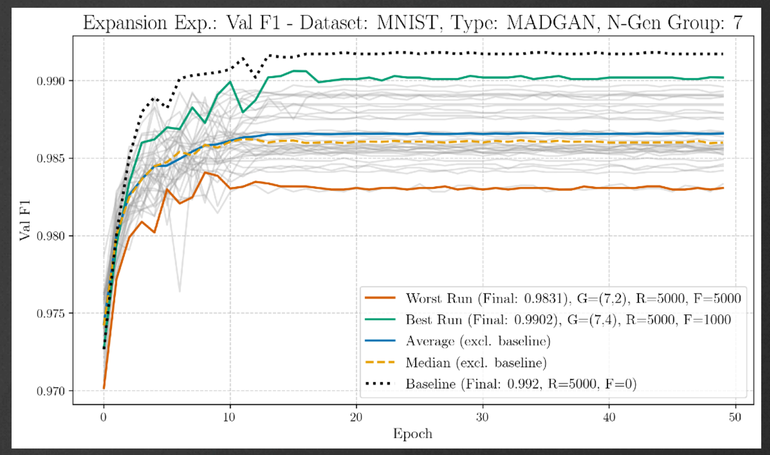
cMADGAN

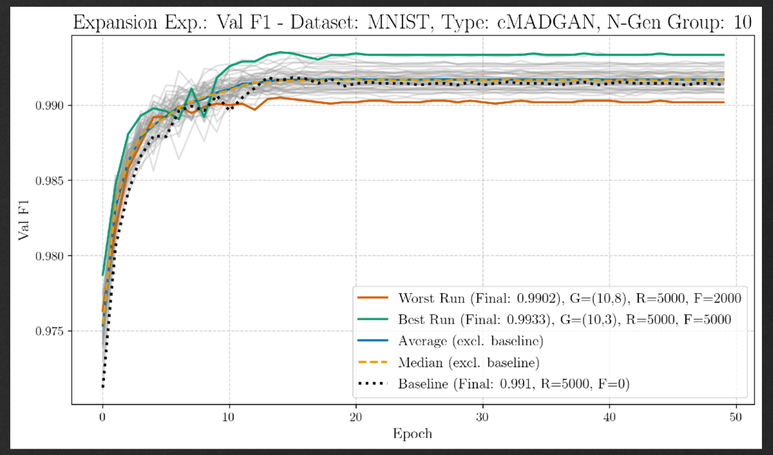
Replacement Experiment





Expansion Experiment



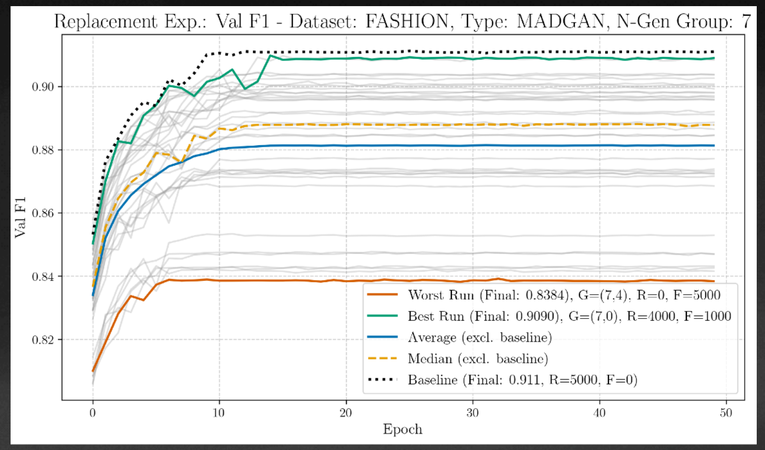


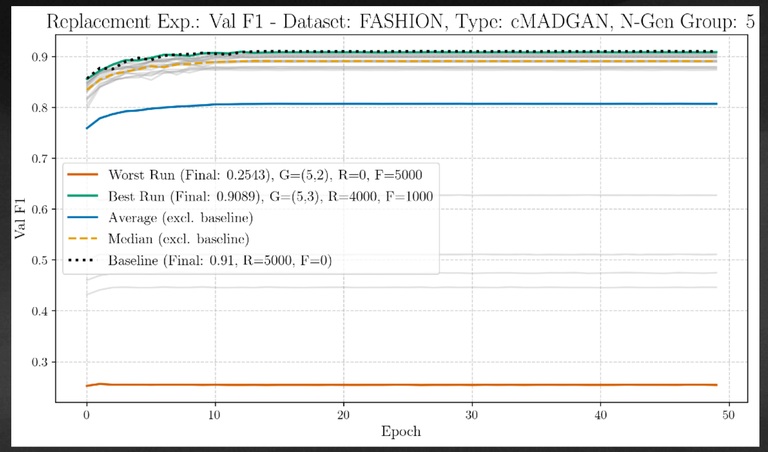
Q4.2 Fashion-MNIST: How does the performance enhancement achieved with MADGAN-based GDA compare to that of cMADGAN-GDA?

MADGAN

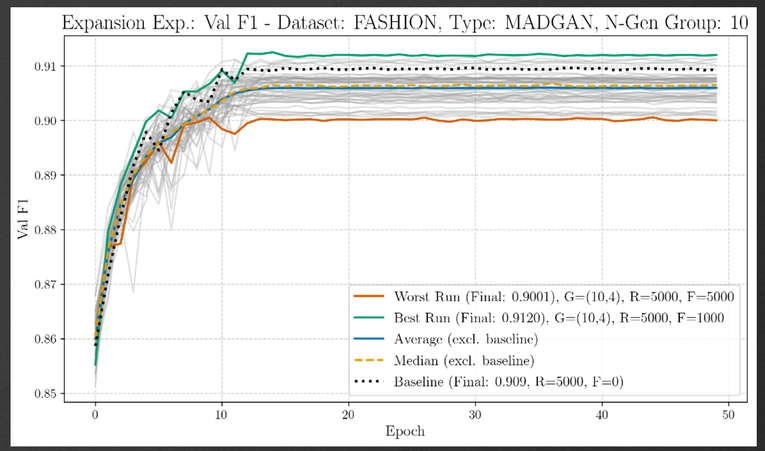
cMADGAN

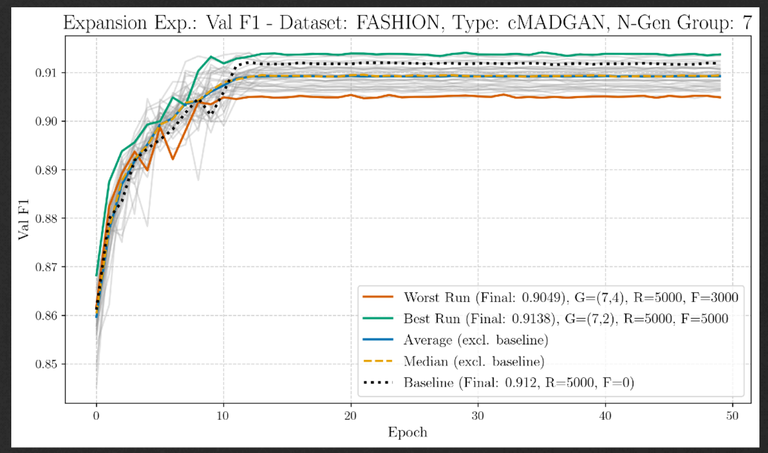
Replacement Experiment





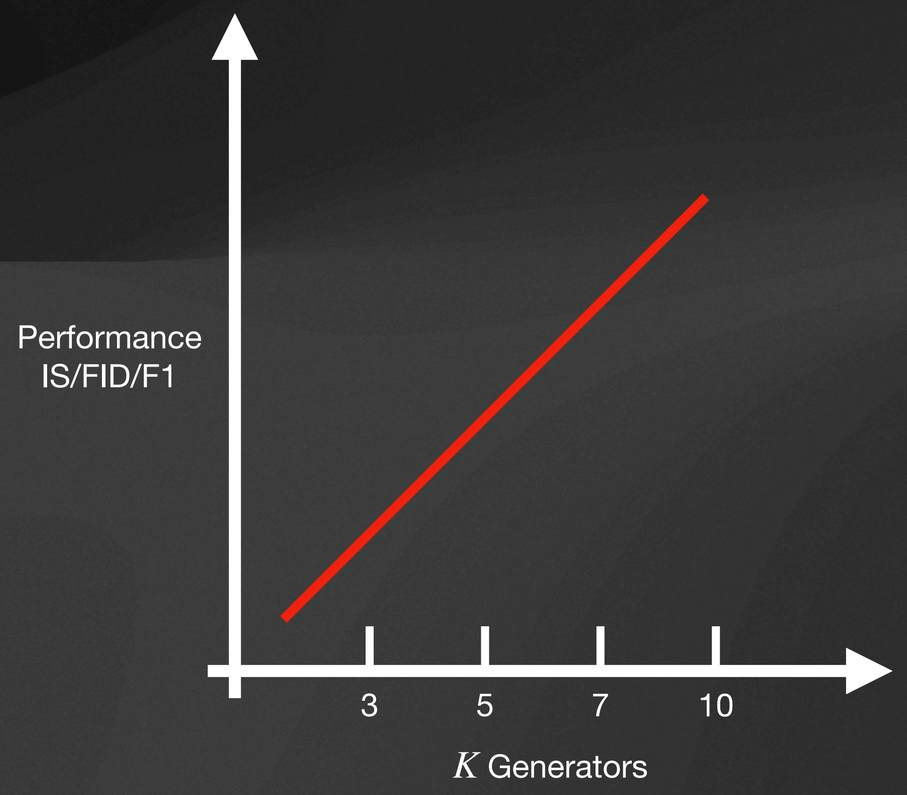
Expansion Experiment



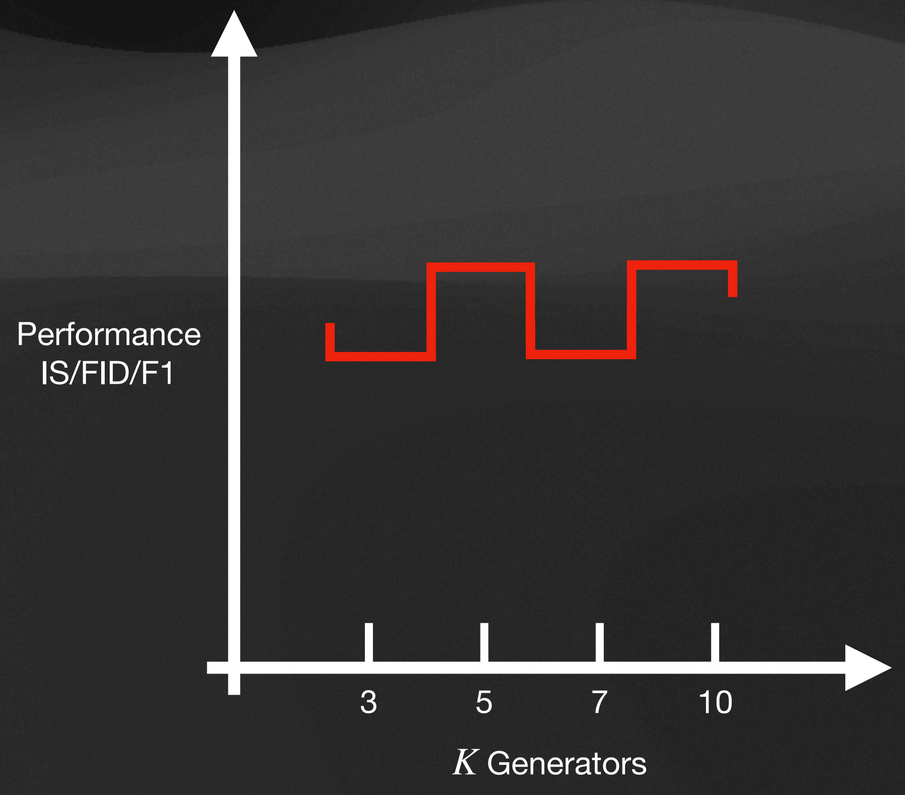


Q5: What is the impact of varying the number generators in multi-generator architectures on downstream classifiers?

MADGAN



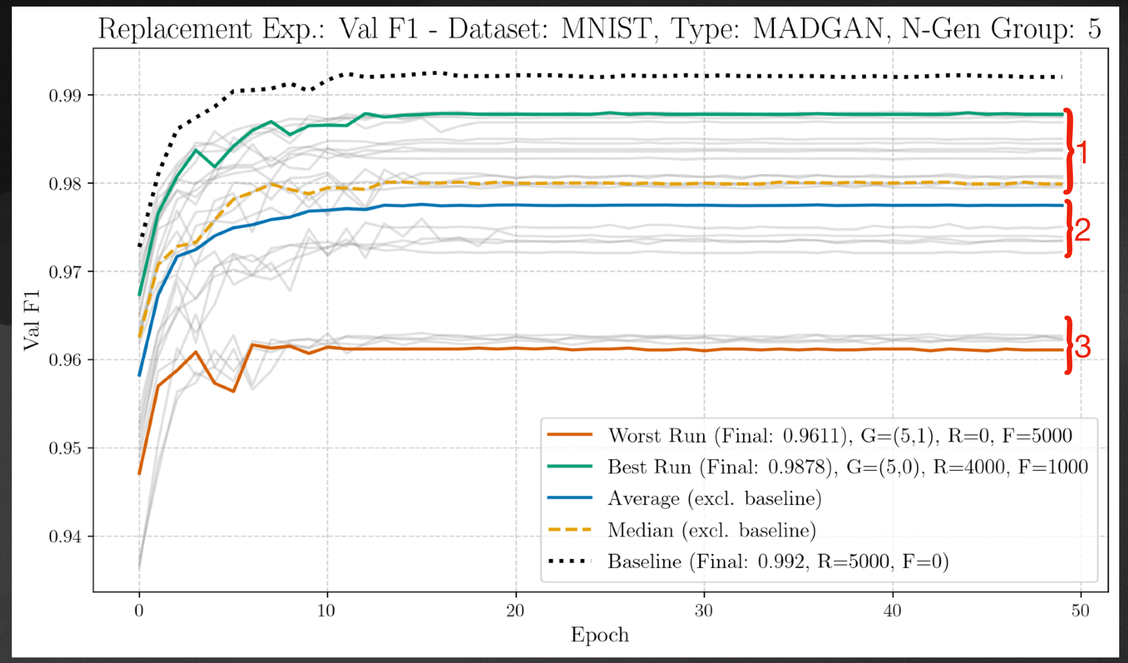
cMADGAN



Remarks

Identifying the Generators

Remarks



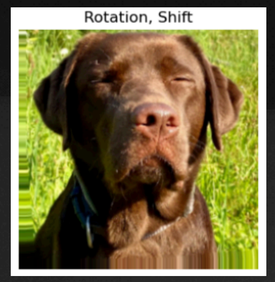
The graphs indicate three clusters of generator performance.

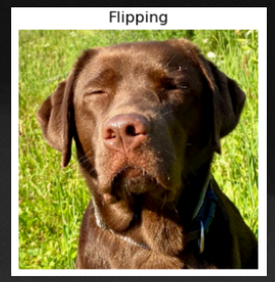
Which generators belong to which cluster?

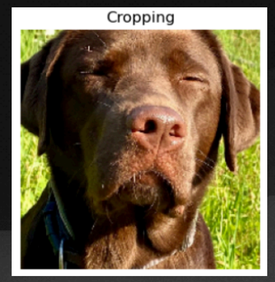
Distinctness of Generators

Remarks

G_1

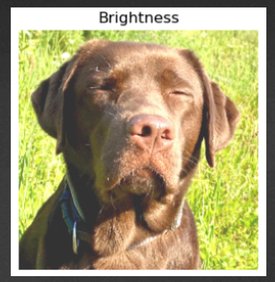


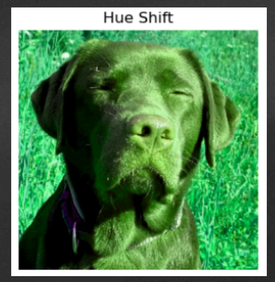


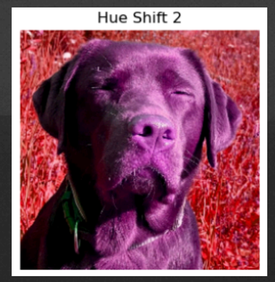


Geometric

G_2

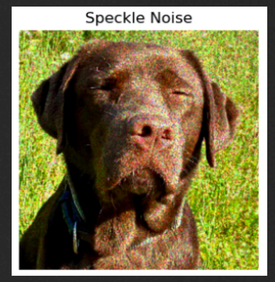






Photometric

G_3

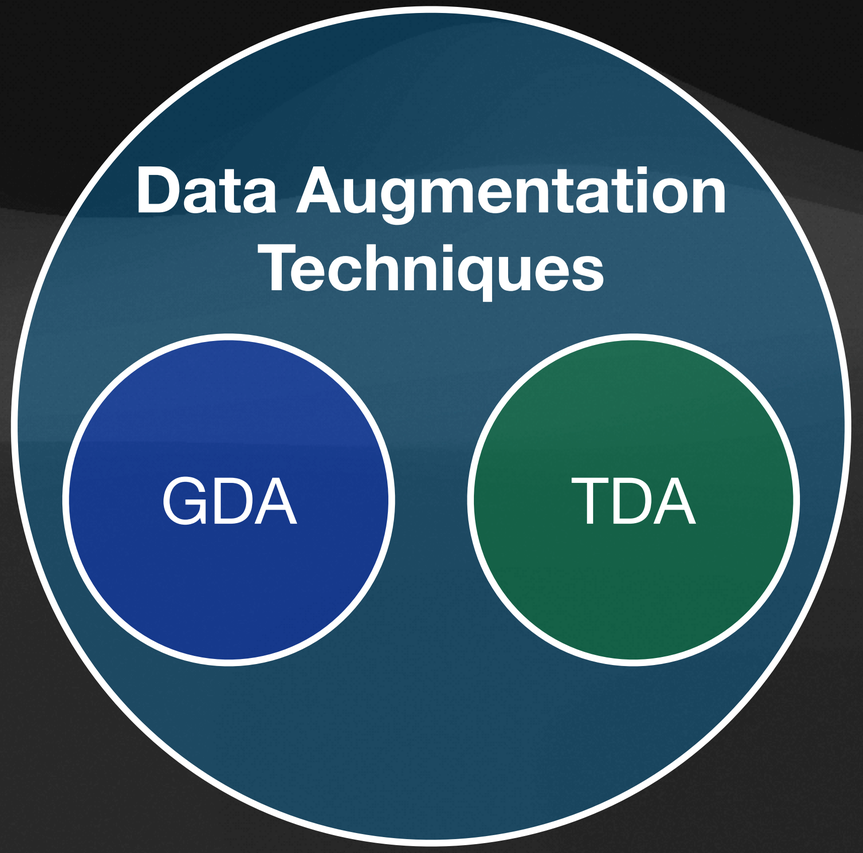


Noise

Conclusion

Conclusion

1. Generative Data Augmentation is an augmentation technique and should be treated as such.
2. The best result may occur when combining different approaches.
3. Regardless of the method augmentation techniques must be evaluated for a problem at hand.



Generative Data Augmentation

Multi-Agent Diverse Generative Adversarial Networks for Generative Data Augmentation

2025-06-26

Author: Nicolas Reinhart, nr063, 44100

Ressources

1. Ghosh et al. 2018, *Multi-Agent Diverse GANs,* figure 3, page 4
   https://arxiv.org/pdf/1704.02906

2. Wikipedia, *MNIST database*, image by Yann LeCun
   https://en.wikipedia.org/wiki/MNIST_database#/media/File:MNIST_dataset_example.png

3. Fashion-MNIST GitHub repository by Zalando Research
   https://github.com/zalandoresearch/fashion-mnist/blob/master/doc/img/fashion-mnist-sprite.png