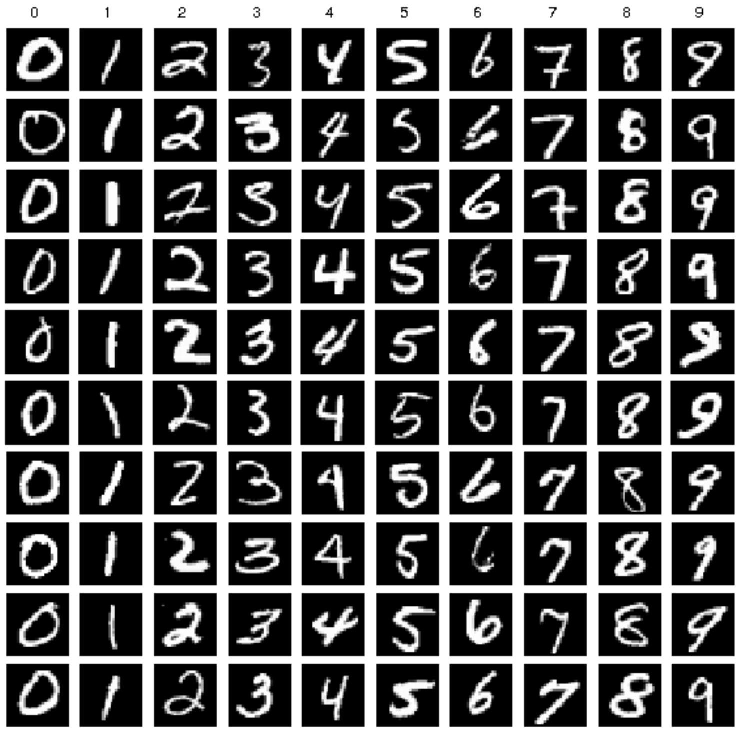
BOUSSOUF Noâm
ELAFANI Maïssa
KANLI Emre Çağan

ASSIGNMENT 2

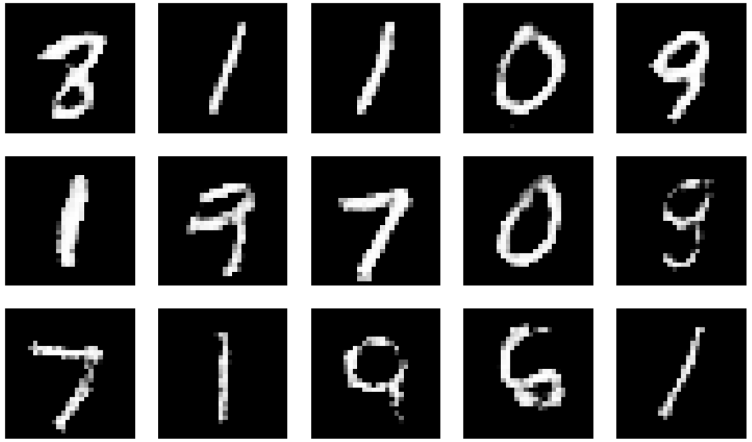
LEARNING LATENT SPACE REPRESENTATION AND APPLICATION TO IMAGE GENERATION

INTRODUCTION



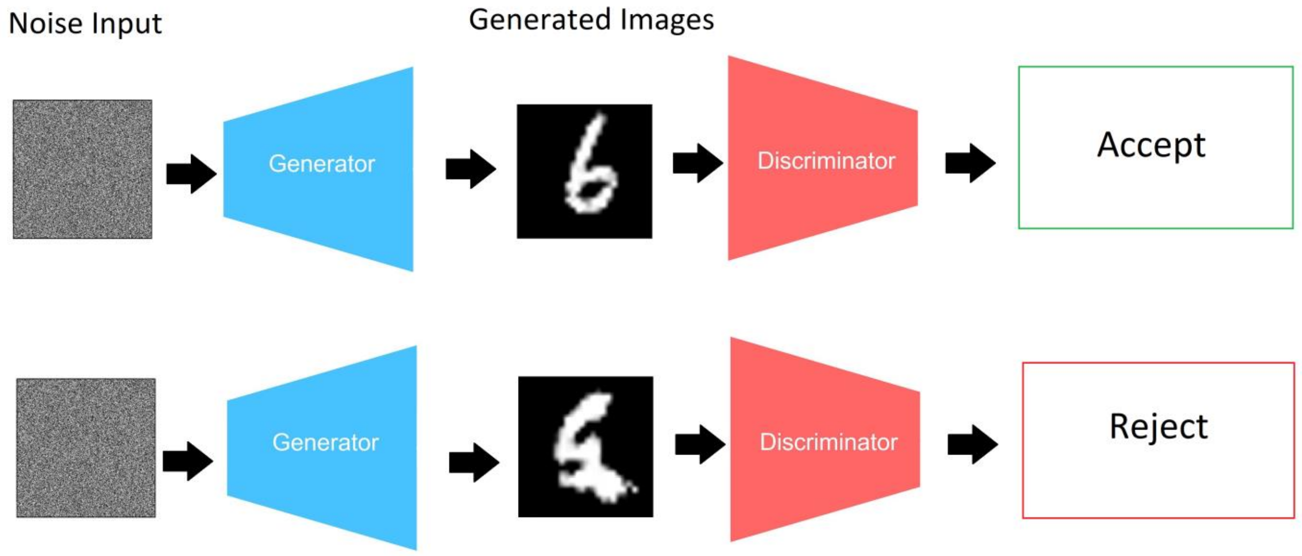
GOAL : IMPROVE THE DATA GENERATION OF A GAN ON THE MNIST DATABASE WITHOUT CHANGING THE GENERATOR'S ORIGINAL STRUCTURE

VANILLA GAN

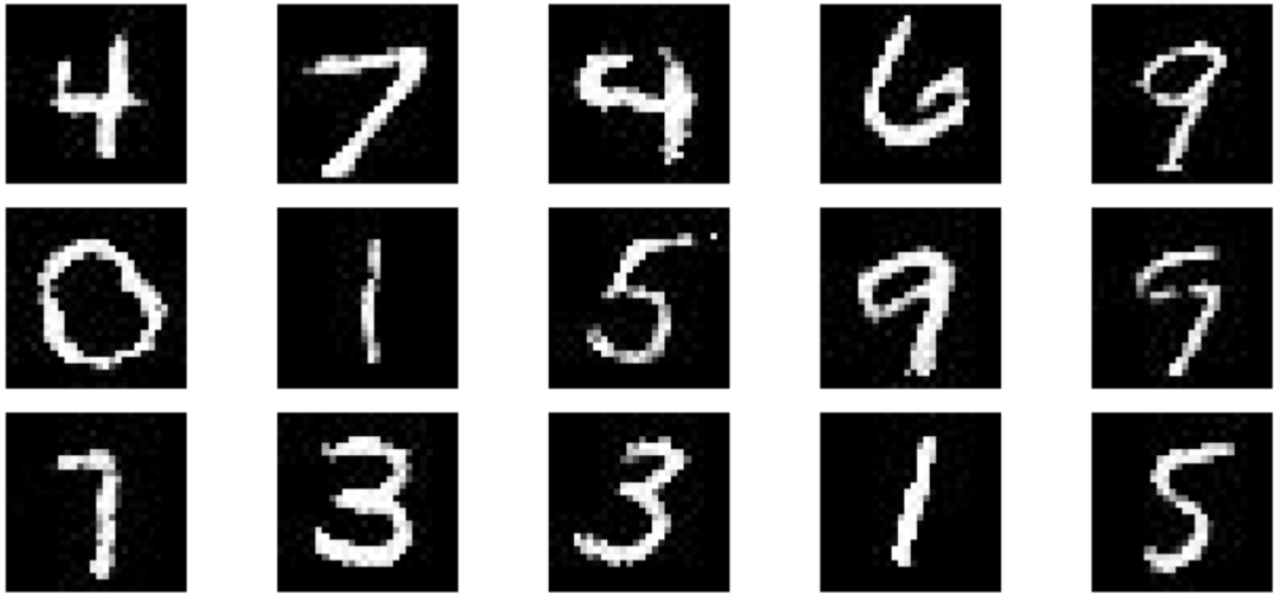


- FID: 38
- Optimization : Learning Rate, Batch Size, etc..
- Baseline for other methods

DISCRIMINATOR REJECTION SAMPLING



- Vanilla GAN FID: 38 to 22
- ~32,000 attempts
- Improves less when the model is already doing well



WASSERSTEIN GAN

$$L_{\text{discriminator}} = -\mathbb{E}_{x\sim\mathbb{P}_{\text{data}}}[D(x)] + \mathbb{E}_{z\sim\mathbb{P}_z}[D(G(z))]$$

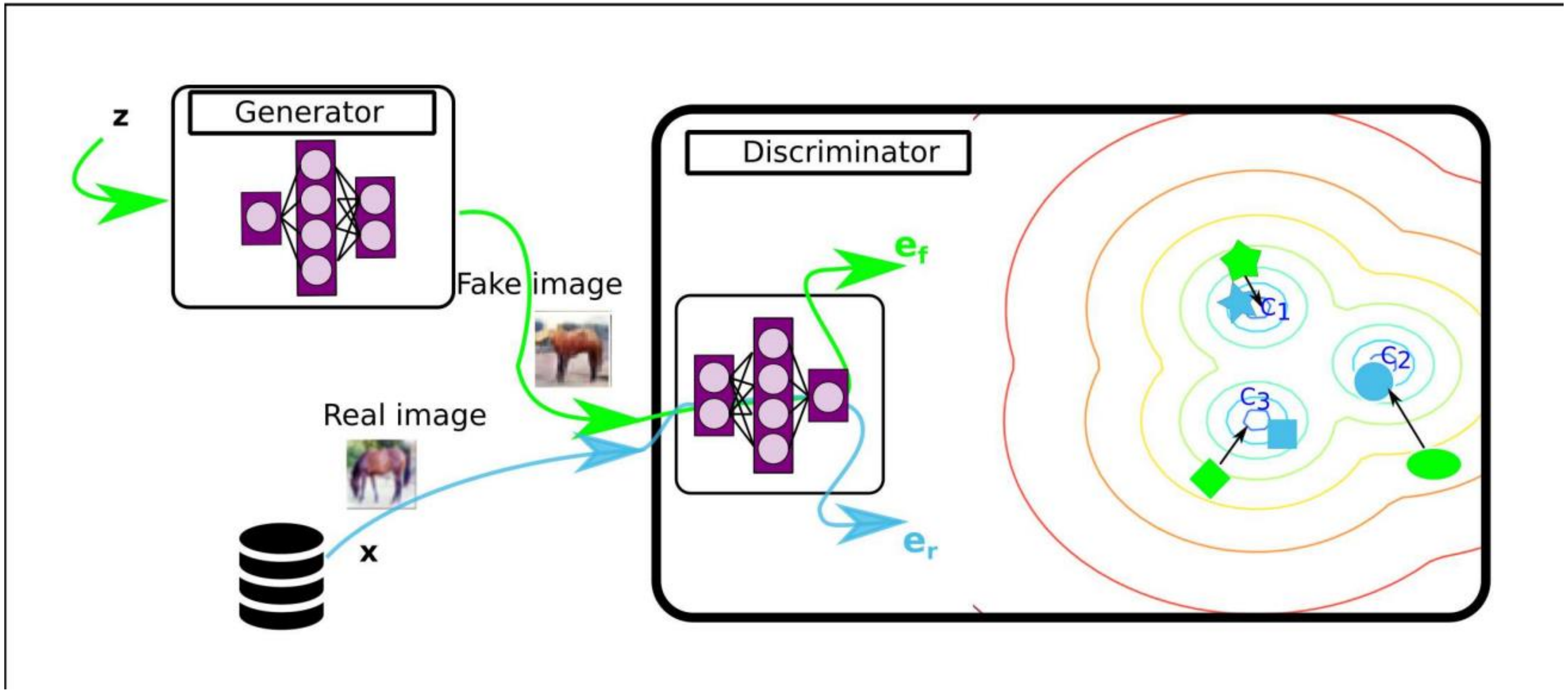
$$L_{\text{generator}} = -\mathbb{E}_{z\sim\mathbb{P}_z}[D(G(z))]$$

- Earth Mover Distance Loss Functions
- Unbounded Discriminator
- Generator Normalization or Gradient Penalty

WGAN - GP

- Discriminator function should be 1-Lipschitz
- 150 + 30 Epochs (different learning rates)
- FID Score after 150 epochs : 28
- FID Score after 150 + 30 epochs : 16
- With rejection sampling : 15

MIXTURE DENSITY GAN (MD-GAN) [4]



Idea : Structure the embedding space of the discriminator by creating clusters and guide the generator to give fake embeddings that align with these clusters.

MIXTURE DENSITY GAN (MD-GAN) [4]

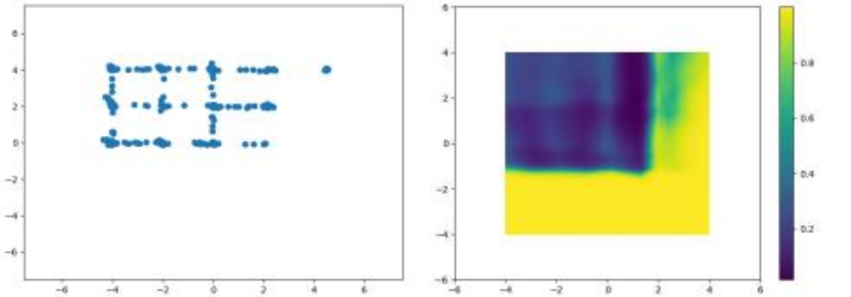
$$\min_G \max_D \mathcal{L}(G, D) =$$
$$\min_G \max_D \Big(\mathbb{E}_{\boldsymbol{x}\sim p_{data}}\big[\log(lk(D(\boldsymbol{x})))\big] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}}\big[\log(\lambda - lk(D(G(\boldsymbol{z}))))\big]\Big)$$

$$lk(e) = \sum_{i=1}^{C} \frac{1}{d+1} \cdot \Phi(e; \mu_i, \Sigma_i)$$

Each Gaussian kernel

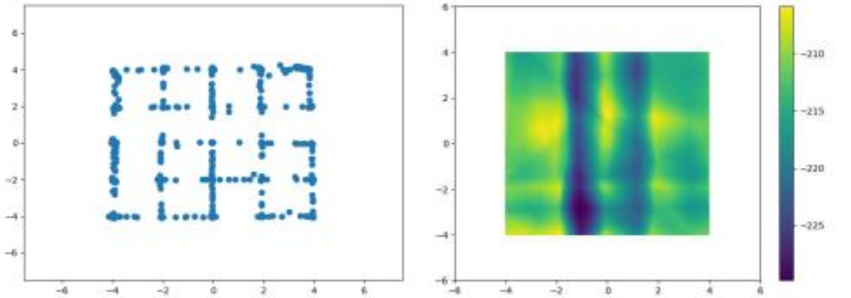
$$\Phi(e; \mu_i, \Sigma_i)$$

represents a probability density function centered around a cluster mean μ_i, with covariance Σ_i, forming one component of a Gaussian mixture.



(a) Generated samples from vanilla GAN.

(b) Probability landscape of the vanilla discriminator.



(c) Generated samples from MD-GAN.

(d) Probability landscape of the MD-GAN discriminator.

Picture taken from the original paper

MIXTURE DENSITY GAN (MD-GAN) [4]

We tried the method, it worked well on Gaussian 2D cases (see Figure 2 and 3).

But we didn't afford to find a good set of hyperparameters that was working well for MNIST with the imposed architecture.

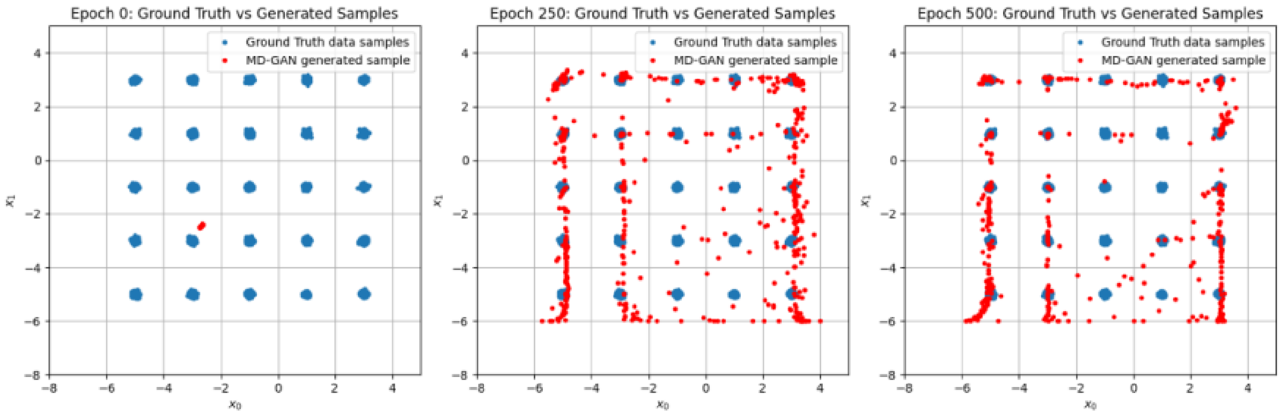


Figure 2: Results of MD-GANs for 25 gaussian components

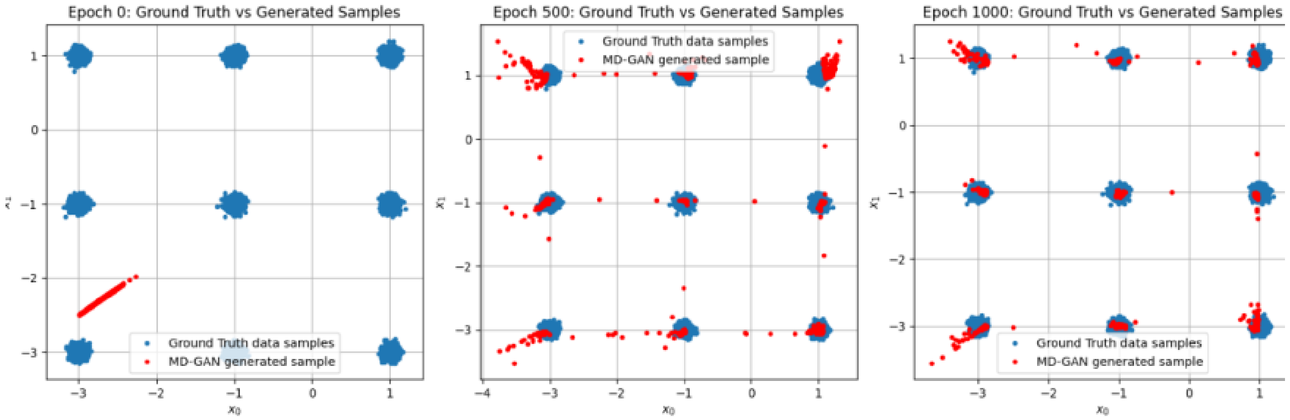
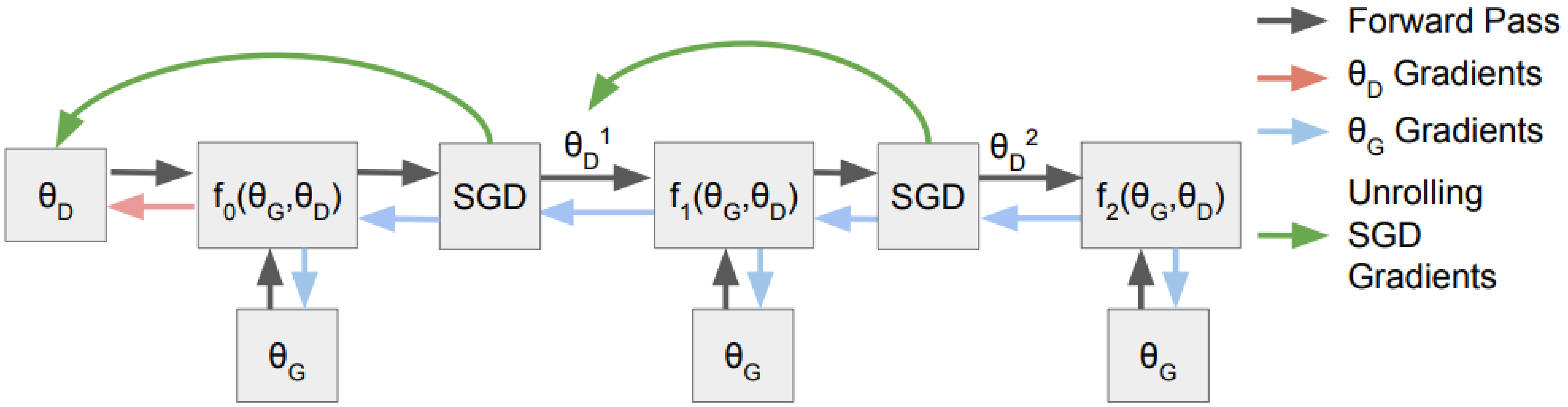


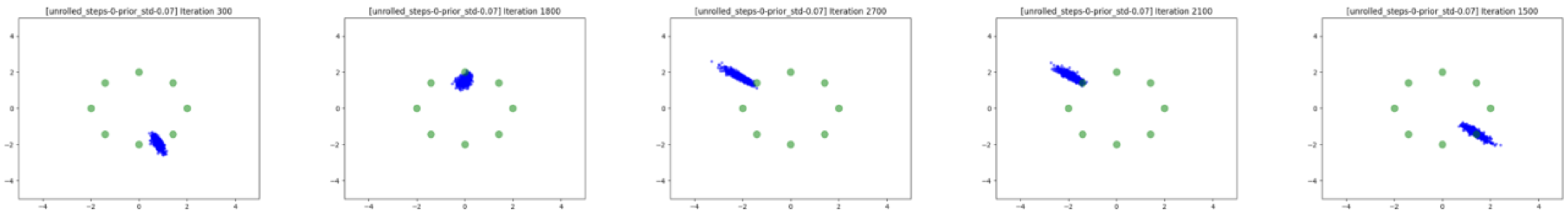
Figure 3: Results of MD-GANs for 9 gaussian components

UNROLLED GANS [5]

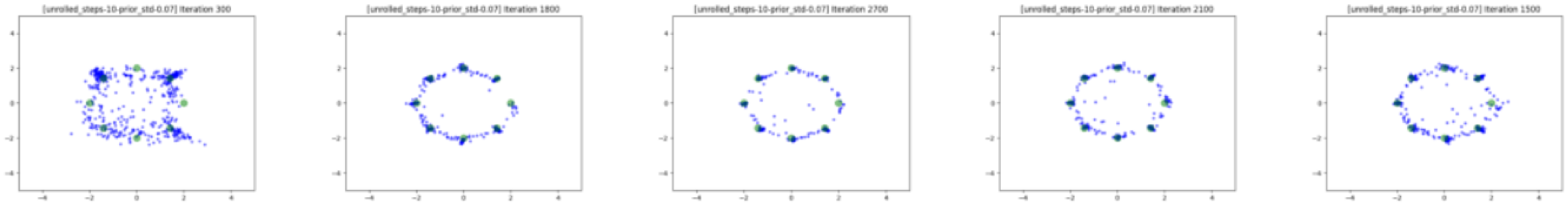
Idea : Stabilize GAN training by allowing the generator to anticipate how the discriminator will improve. Before updating the generator, we simulate multiple future steps of the discriminator's training) to see how it would respond if it were optimized further.



UNROLLED GANS [5]

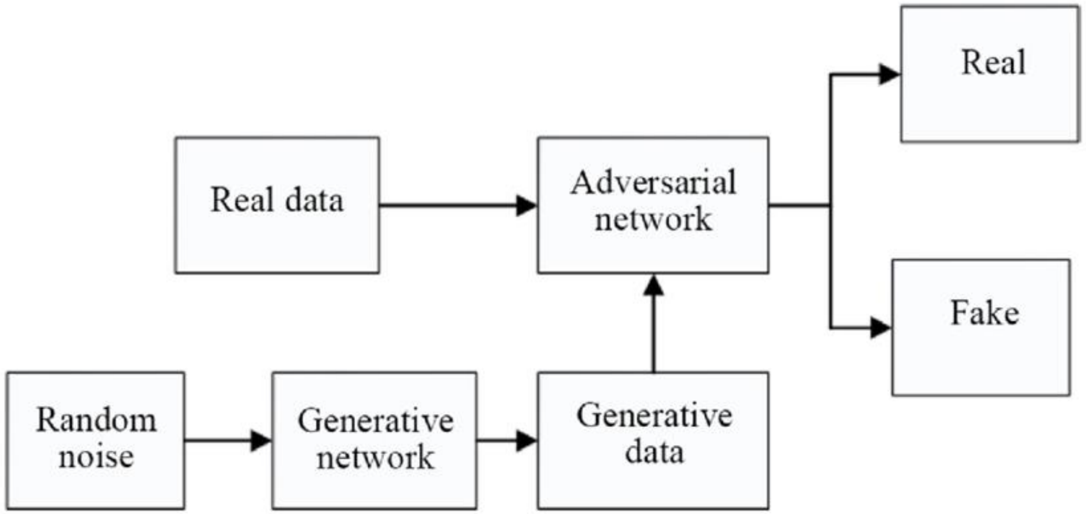


Results on 3000 iterations using Vanilla GAN architecture



Results on 3000 iterations using a Vanilla GAN architecture + Unrolling GANs

R1-REGULARIZATION



R1-REGULARIZATION

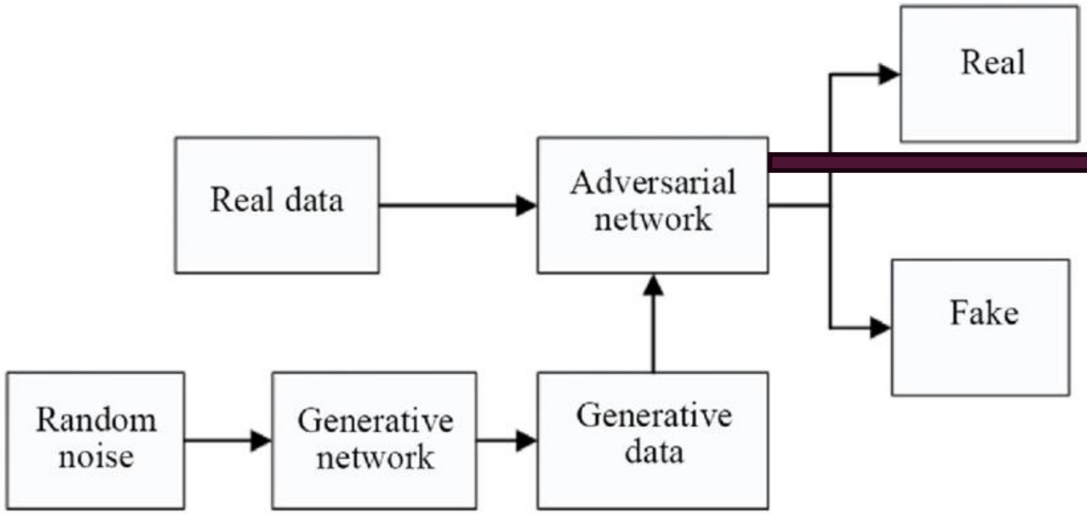


Figure : GAN Architecture

+ $$R_1(\psi) := \frac{\gamma}{2} \mathrm{E}_{p_{\mathcal{D}}(x)} \left[\|\nabla D_\psi(x)\|^2\right]$$

with ψ representing the parameters of the discriminator and γ, the regularization hyperparameter

R1-REGULARIZATION

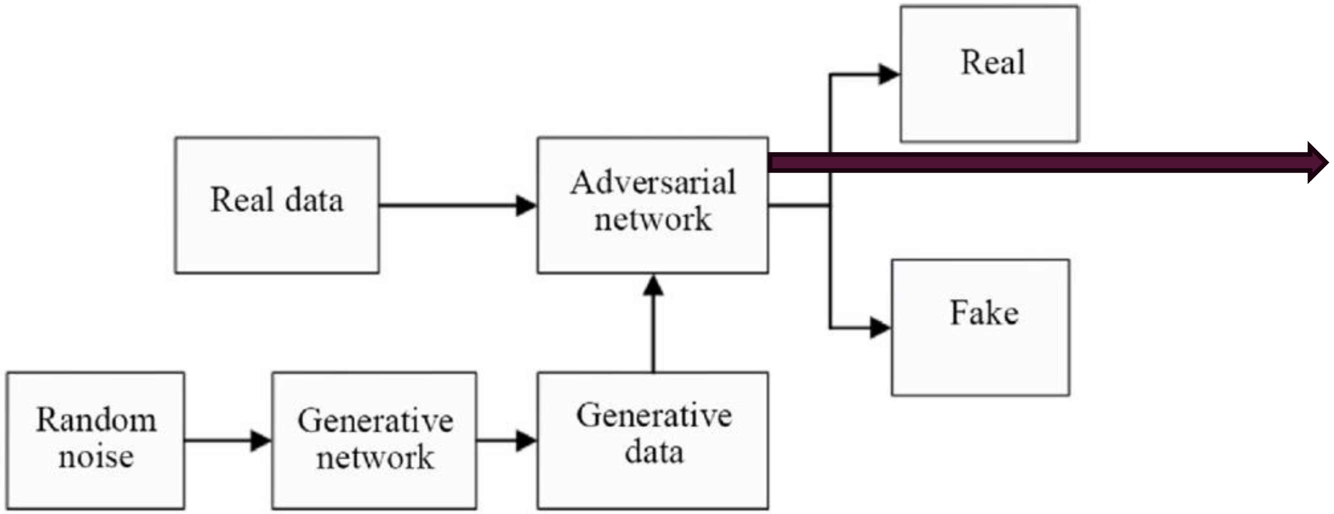


Figure : GAN Architecture

+ $$R_1(\psi) := \frac{\gamma}{2} \mathrm{E}_{p_{\mathcal{D}}(x)} \left[\|\nabla D_\psi(x)\|^2\right]$$

with ψ representing the parameters of the discriminator and γ, the regularization hyperparameter

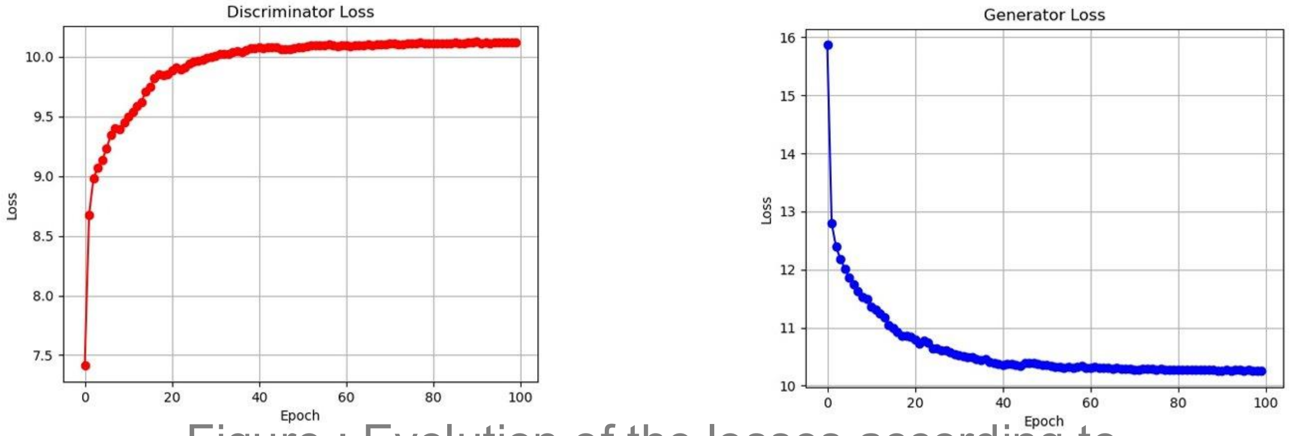


Figure : Evolution of the losses according to the number of epochs

CONCLUSION

	VGAN	R1-Reg	WGAN w\BN.	WGAN-GP
FID	38	42	27	16

VGAN w\DRS	WGAN-GP w\DRS	Unrolled	MD
22	15	**inf**	inf

Figure : Evaluation of our methods using FID (locally) with the pytorch-fid library

THANKS FOR YOUR ATTENTION

REFERENCES

[1] Rejection Sampling : Luke Metz, Ben Poole, David Pfau, and Jascha Sohl-Dickstein. Unrolled generative adversarialnetworks.arXiv preprint arXiv:1611.02163, 2016.
[2] WGAN :
Martin Arjovsky, Soumith Chintala, and L´eon Bottou. Wasserstein generative adversarialnetw orks. InInternational conference on machine learning, pages 214–223. PMLR, 2017.
[3] WGAN-GP : Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville.Improved training of wasserstein gans, 2017.
[4] MD-GAN : Hamid Eghbal-zadeh, Werner Zellinger, and Gerhard Widmer. Mixture density generative adversarial networks. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 5820–5829, 2019.
[5] Luke Metz, Ben Poole, David Pfau, and Jascha Sohl-Dickstein. Unrolled generative adversarial networks. arXiv preprint arXiv:1611.02163, 2016.