

GN GAN

Présenté par Gabriel Rochon

VUE D'ENSEMBLE

- Introduction
- Problèmes
- Solutions
- Résultats
- Conclusion
- Référence

INTRODUCTION

Qu'est-ce que le GN GAN ?

- Le GN GAN est un ensemble de deux méthodes présenté par Ngoc-Trung Tran*, Tuan-Anh Bui*, Ngai-Man Cheung en 2018
- Diffère des GAN classiques (neighbors embedding et le gradient matching)
- Introduit un autoencoder dont le decodeur est le générateur
- Est créé pour résoudre certains problèmes

PROBLEMES DES GAN “CLASSIQUES”

- Dans les GAN “classiques” le générateur peut créer des échantillons similaires. (mode collapse)
- Les GAN peuvent approximer une solution sous optimale
- L’apprentissage n’est pas toujours stable

QUELLE SOLUTION THEORIQUE ?

Neighbors embedding

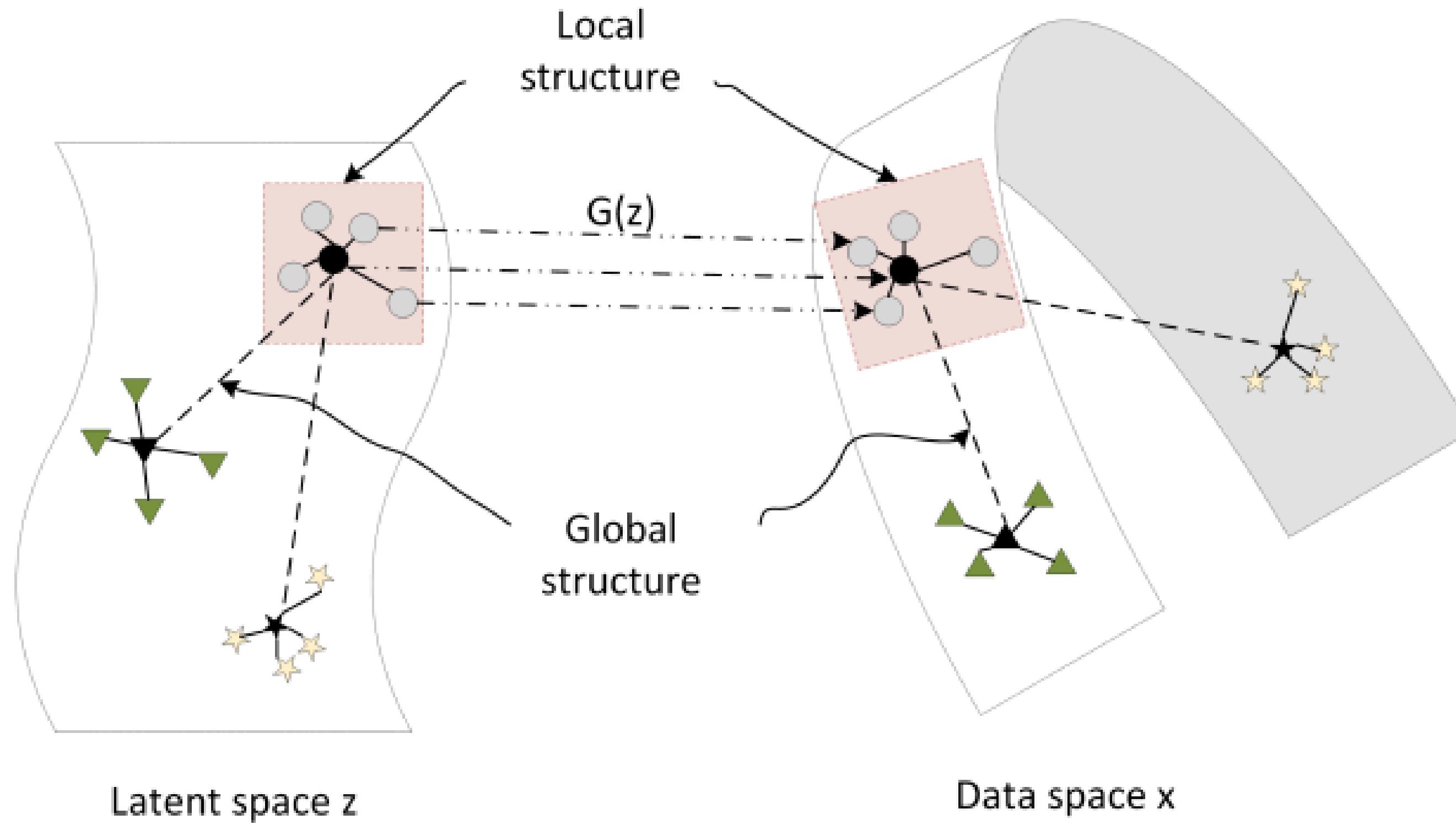
- A pour but de conserver les structures locales des échantillons latents dans l'espace des données

Gradient matching

- A pour but d'aligner la distribution des échantillons générés et les échantillons réels en utilisant le score du discriminateur

NEIGHBORS EMBEDDING

- But: éviter de générer des données identiques provenant d'échantillons différents
- But: faire en sorte que les distances relatives entre les points dans l'espace latent soient conservées dans l'espace des données générées
- Améliore la diversité et donc le recall



GRADIENT MATCHING

- Alignement en termes de valeurs des scores, mais aussi en termes de comportements locaux. En utilisant la différence des gradients
- Améliore la diversité

ALGORITHME GÉNÉRAL

Algorithm 1 Our GN-GAN model

```
1: Initialize discriminator, encoder and generator  $D, E, G$  respectively.  $N_{iter}$  is the number of iterations.  
2: repeat .  
3:    $x \leftarrow$  Random mini-batch of  $m$  data points from dataset.  
4:    $z \leftarrow$  Random  $n$  samples from noise distribution  $P_z$   
5:   // Training the auto-encoder using  $x$  and  $z$  by Eqn. 1  
6:    $E, G \leftarrow \min \mathcal{V}_{AE}(E, G)$   
7:   // Training discriminator according to Eqn. 7 on  $x, z$   
8:    $D \leftarrow \max \mathcal{V}_D(D, G)$   
9:   // Training the generator on  $x, z$  according to Eqn. 13.  
10:   $G \leftarrow \min \mathcal{V}_G(D, G)$   
11: until  $N_{iter}$   
12: return  $D, E, G$ 
```

AUTOENCODER

- Est minimisé comme un autoencoder classique mais avec le neighbours embedding

$$V_{AE}(E, G) = \|\mathbf{x} - G(E(\mathbf{x}))\|^2 + \lambda_r V_R(E, G)$$

- Augmente la diversité et la qualité
- Neighbors embedding implique que les distances relatives entre les points dans l'espace latent sont conservées dans l'espace des données générées
- But que les clusters qui soient relativement proches dans l'espace latent le soient dans celui des données

DISCRIMINATEUR

- Est maximisé comme un discriminateur classique mais avec des pénalités

$$\begin{aligned}\mathcal{V}_D(D, G) \\ = & (1 - \alpha) \mathbb{E}_x \log D(x) + \alpha V_C + \mathbb{E}_z \log(1 - D(G(z))) \\ - & \lambda_p V_P\end{aligned}$$

$$V_C = \mathbb{E}_x \log D(G(E(x)))$$

- Pénalité sur la probabilité que D attribue un score élevé ce qui a pour but que le discriminateur ne soit pas trop “fort”
- V_C va forcer à prendre pour vraies les images passées par l’autoencoder
- V_p est une interpolation entre x et $G(z)$ il a pour but de contrôler le gradient, de le maintenir stable et significatif

GENERATEUR

- But: aligner la distribution des données réelles et celles générées

$$\begin{aligned}\mathcal{V}_G(D, G) = & \|\mathbb{E}_x D(x) - \mathbb{E}_z D(G(z))\| \\ & + \lambda_m^1 \|\mathbb{E}_x (\nabla_x D(x)) - \mathbb{E}_z (\nabla_x D(G(z)))\|^2 \\ & + \lambda_m^2 \|\mathbb{E}_x (\nabla_x D(x)^T x) - \mathbb{E}_z (\nabla_x D(G(z))^T G(z))\|^2\end{aligned}$$

- Minimiser $\|\mathbb{E}x(D(x)) - \mathbb{E}z(D(G(z)))\|$ + une contrainte sur les gradients (le Gradient Matching)
- La première contrainte va faire en sorte que la direction des gradients entre les images et celles générées ne soient pas trop différentes
- La seconde va faire en sorte que l'interaction du gradient autour de x ressemble à celle autour de $g(z)$

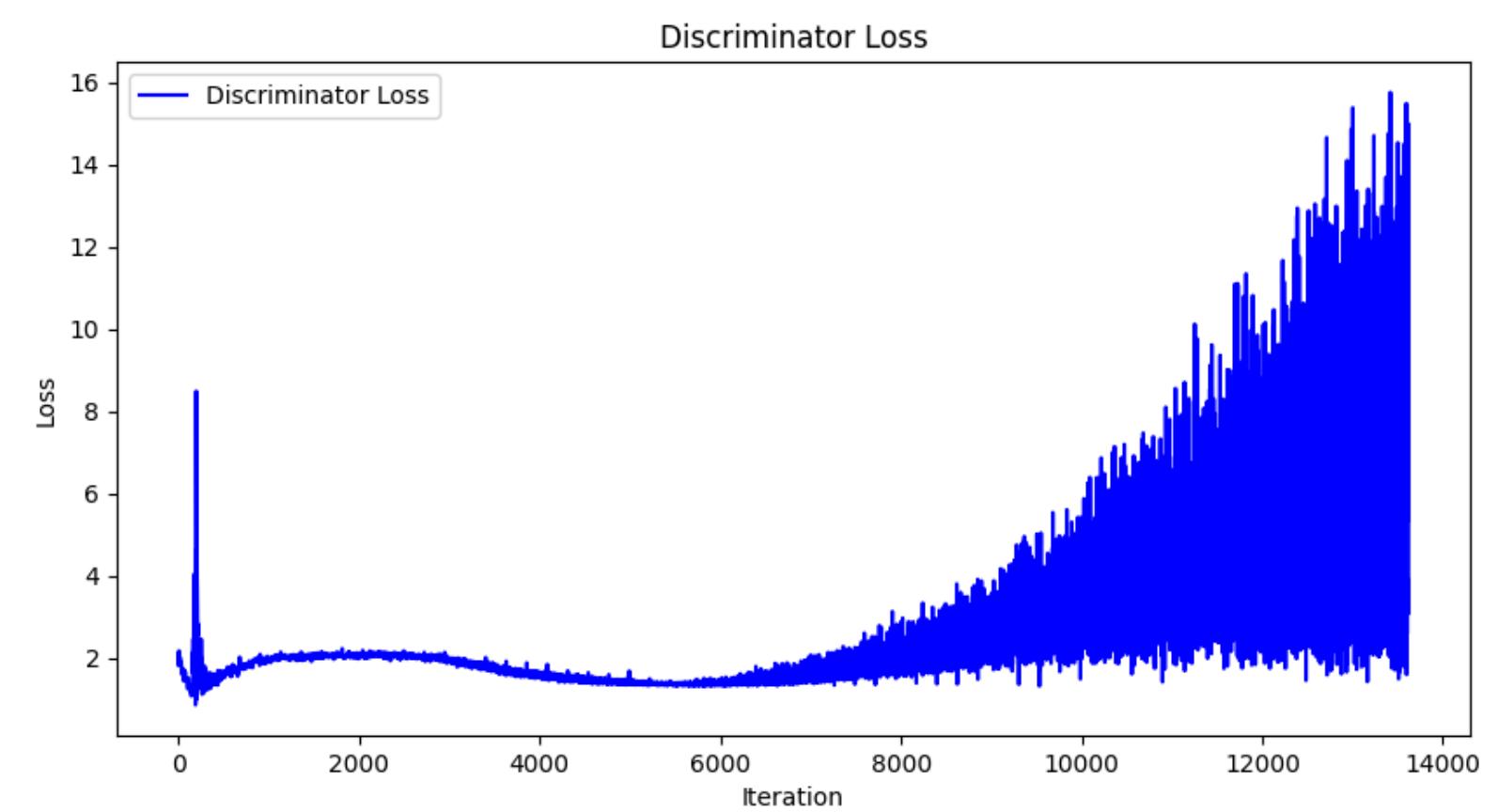
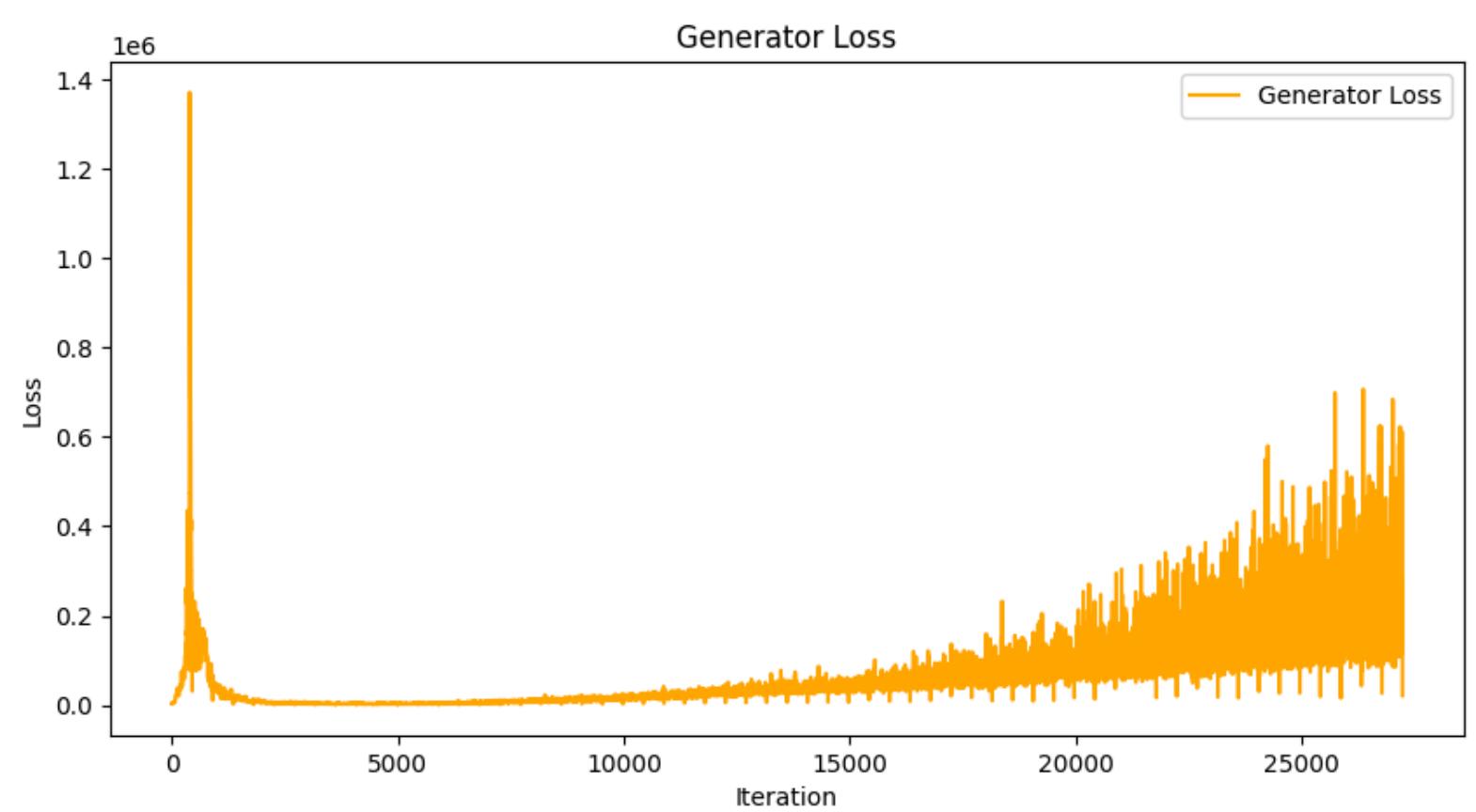
PRINCIPAUX RÉSULTATS

Method	CIFAR	STL	CIFAR (R)
GAN-GP	37.7	-	-
WGAN-GP	40.2	55.1	-
SN-GAN	25.5	43.2	$21.7 \pm .21$
Dist-GAN	22.95	36.19	-
Ours	21.70	30.80	$16.47 \pm .28$

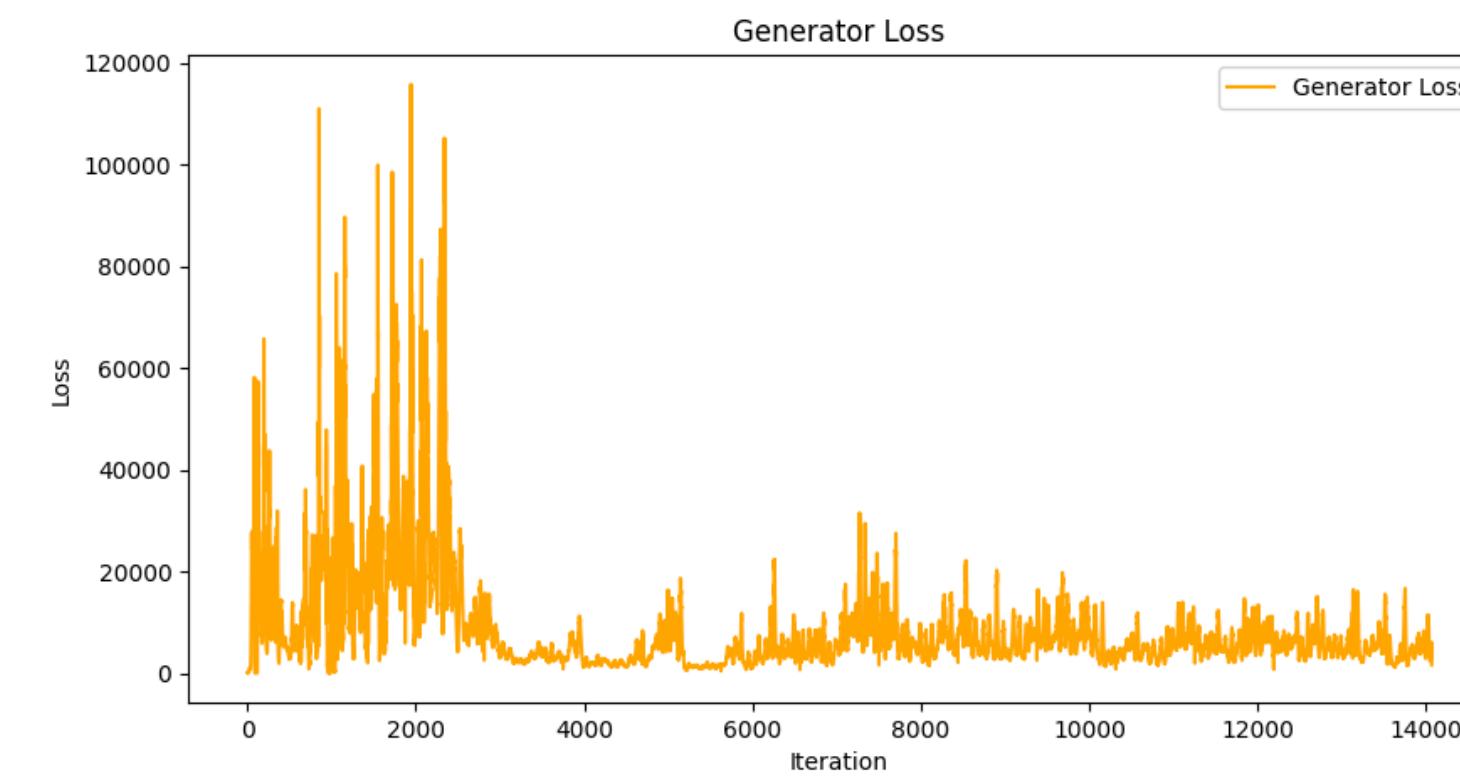
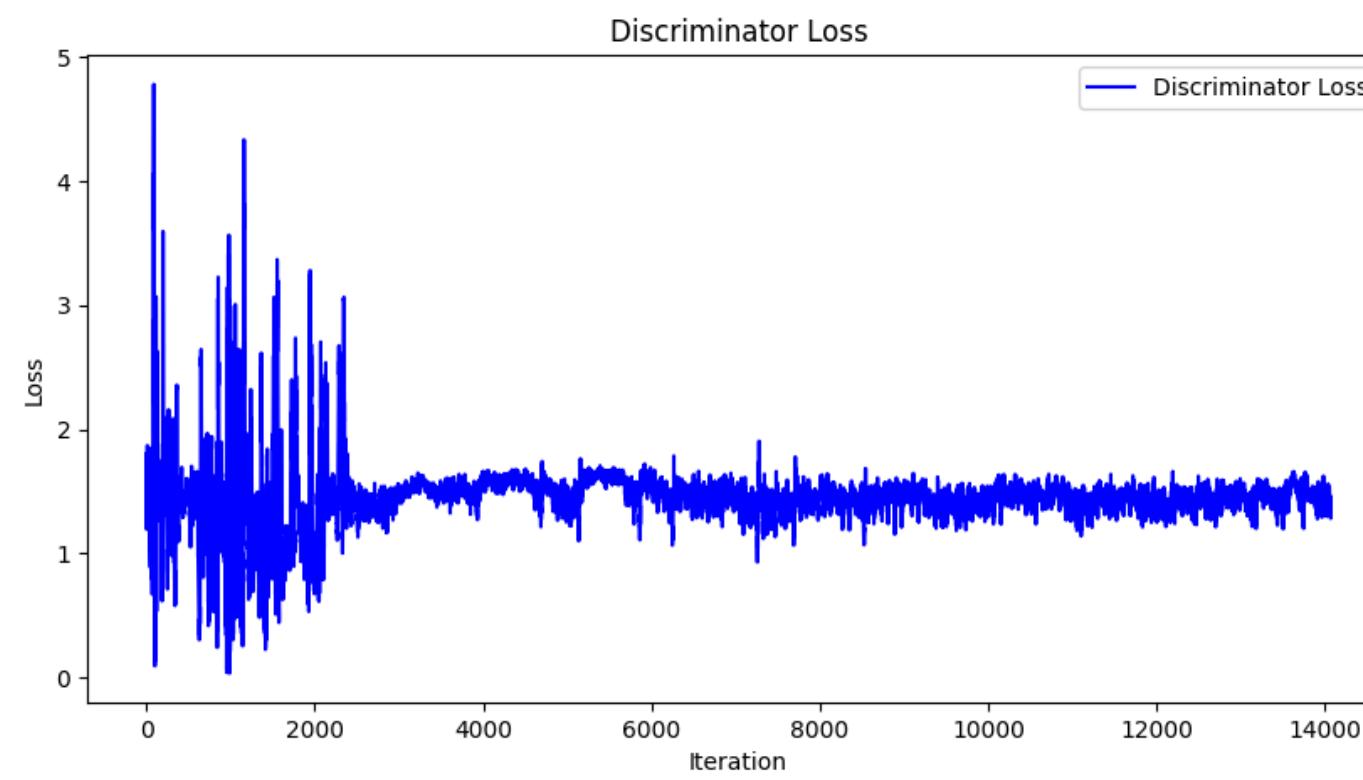
EN PRATIQUE

- Implémentation des méthodes
- Optimisation des paramètres notamment la hausse de la pénalité pour le discriminateur car il était trop “fort”, et une augmentation de la pénalité sur les gradients

AVANT L'OPTIMISATION DES PARAMETRES



APRÈS L'OPTIMISATION DES PARAMETRES



RESULT 1

Sans GN GAN

9	6	1	1	0	2	1	1	3
2	8	5	6	1	4	3	8	9
7	9	0	1	5	1	8	3	5
6	8	1	4	9	1	7	1	6
5	6	2	0	1	1	2	0	3
9	9	3	3	6	2	0	4	2
9	0	1	3	0	5	9	0	0
9	1	1	8	2	0	8	4	0
3	3	1	2	6	2	9	6	5
2	1	9	8	5	1	4	8	4

Avec GN GAN

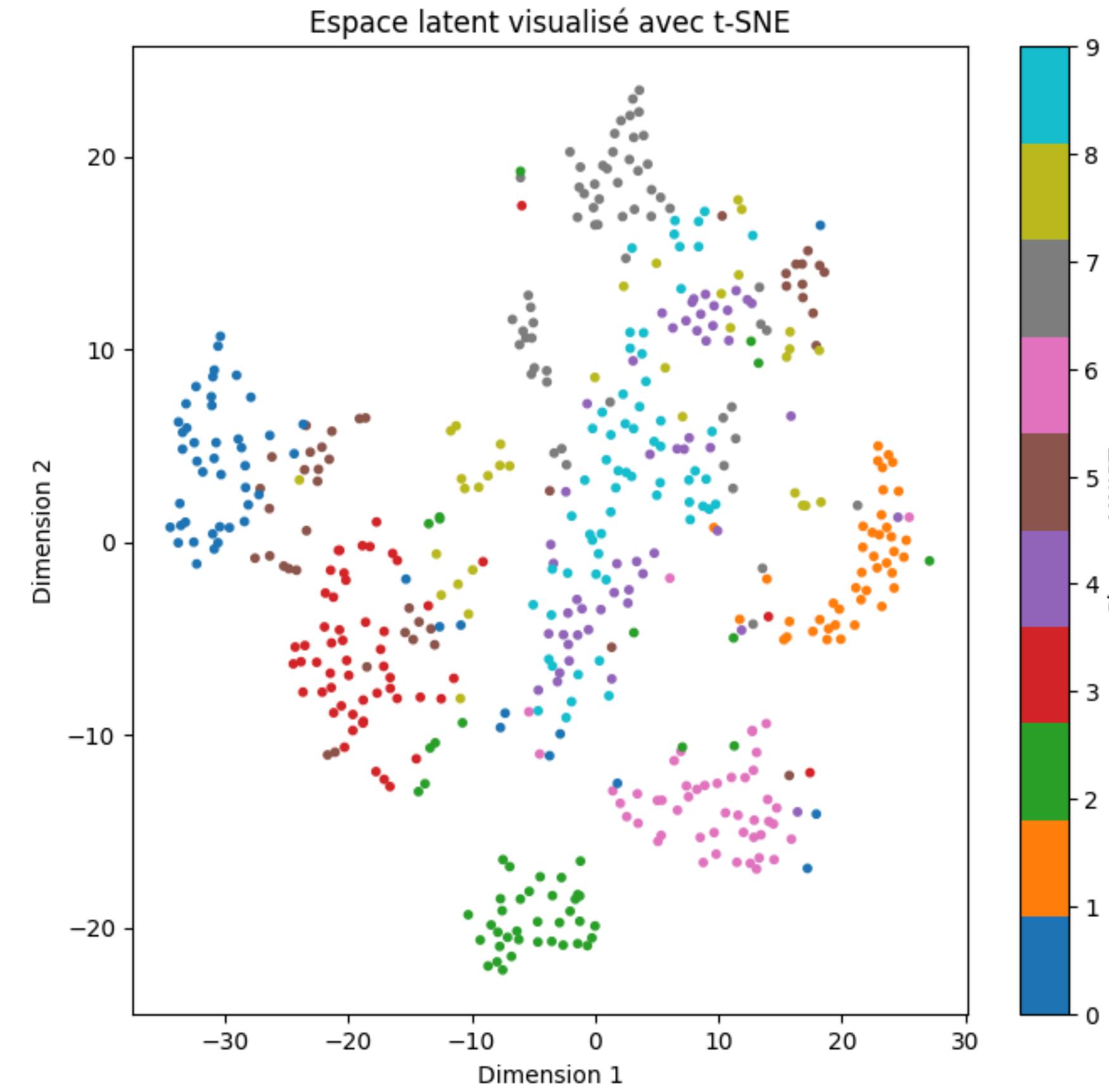
4	3	1	2	5	3	6	3	1	5
2	0	0	3	0	2	1	2	4	0
3	3	4	7	1	1	2	0	0	7
5	5	5	0	7	2	0	7	0	5
1	2	0	9	0	3	1	2	3	3
1	3	5	5	0	9	5	1	7	3
3	2	6	8	4	2	0	4	2	3
2	6	2	1	3	3	5	9	6	2
5	3	3	1	2	1	3	5	2	5
7	5	2	3	4	2	6	0	5	5

56.25

0.6

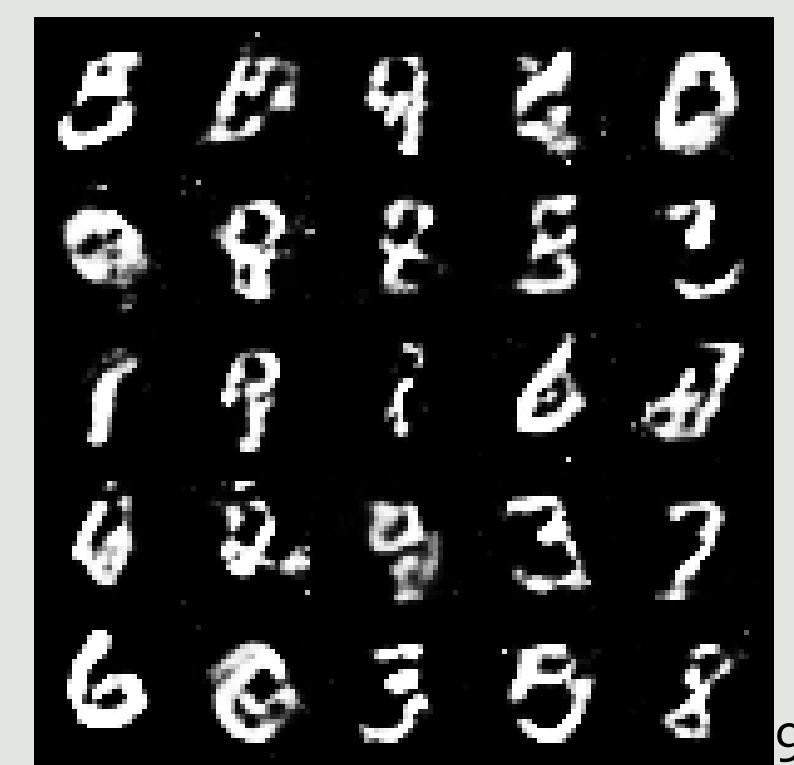
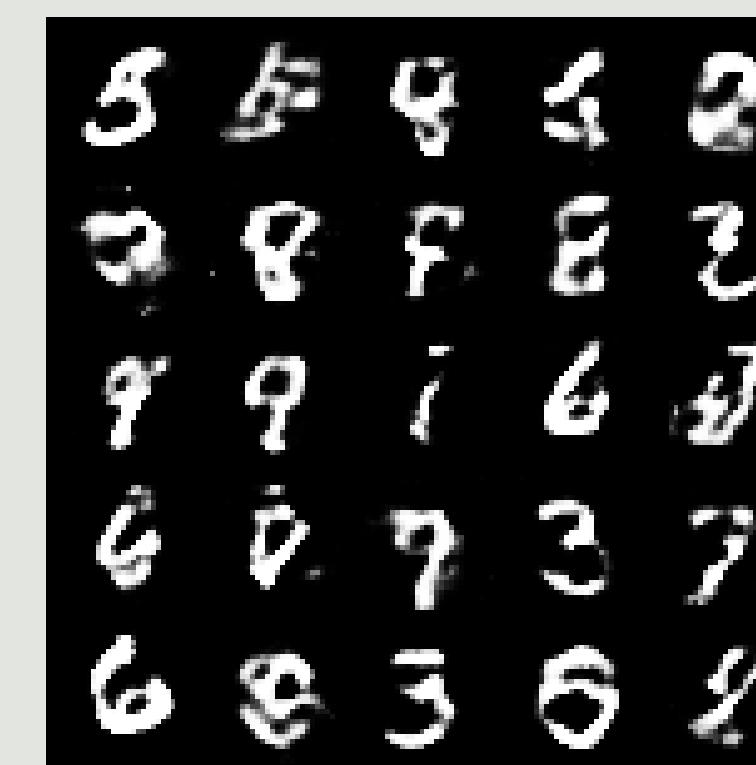
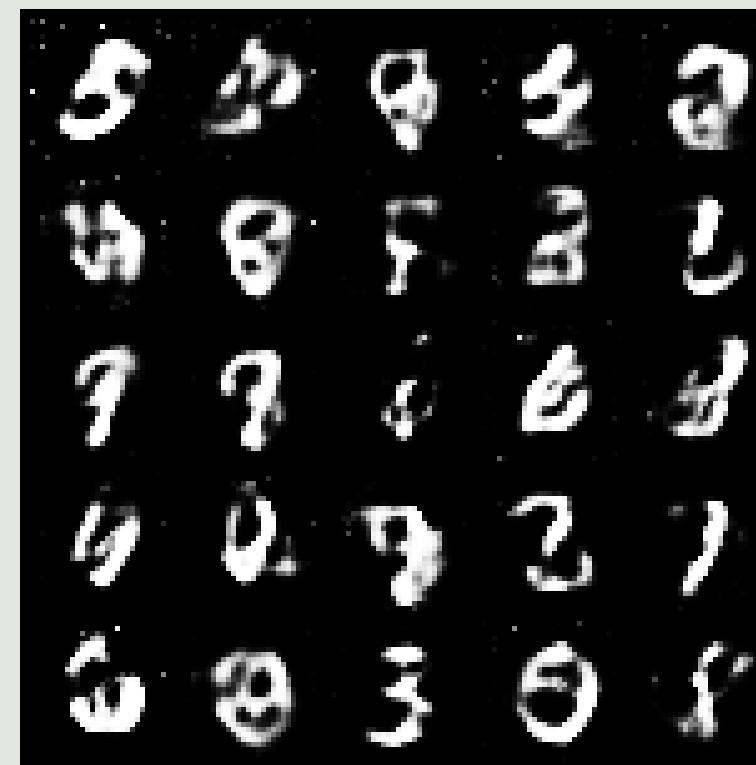
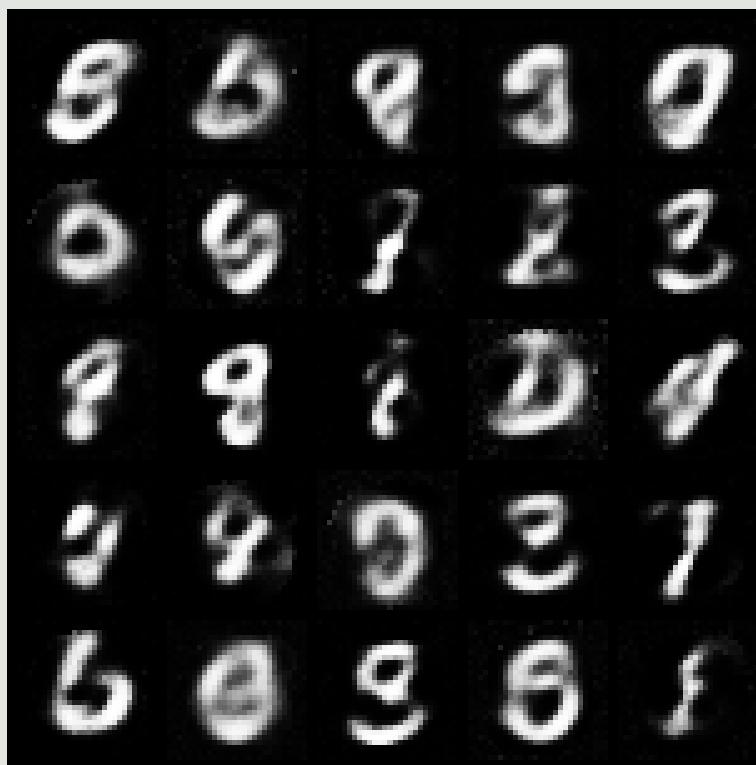
.

0.3

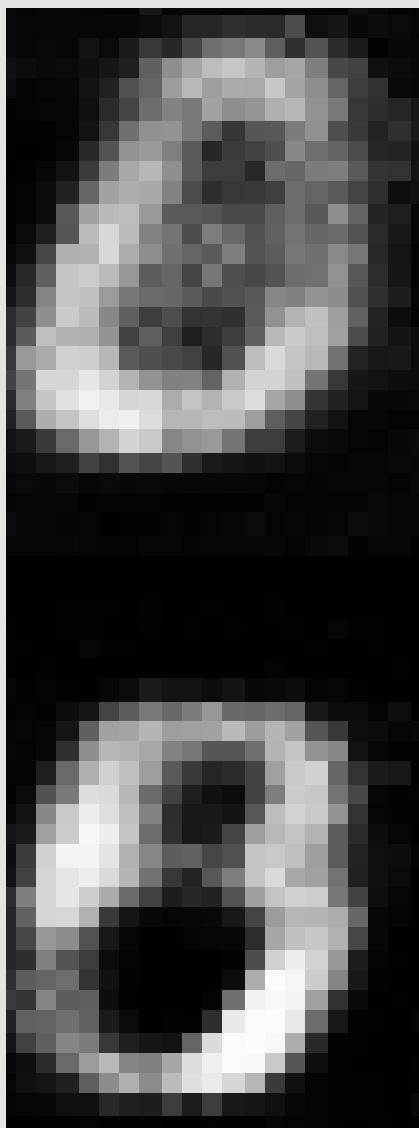


EVOLUTION DES IMAGES DURANT L'APPRENTISSAGE

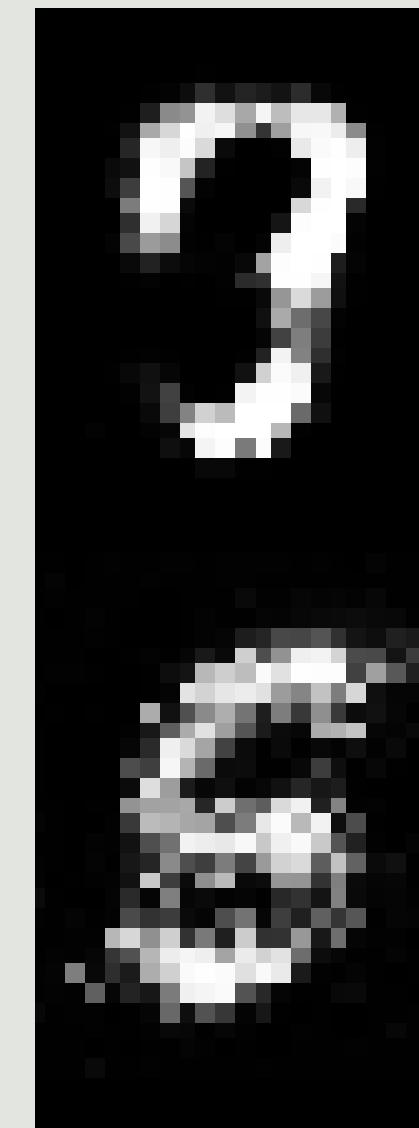
Epoch 10 Epoch 50 Epoch 80 Epoch 100



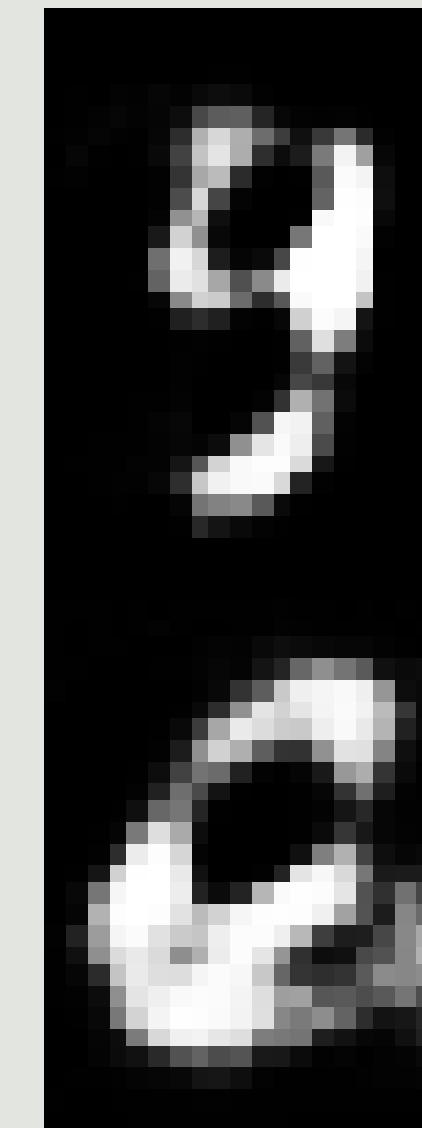
EVOLUTION DES IMAGES DURANT L'APPRENTISSAGE P2



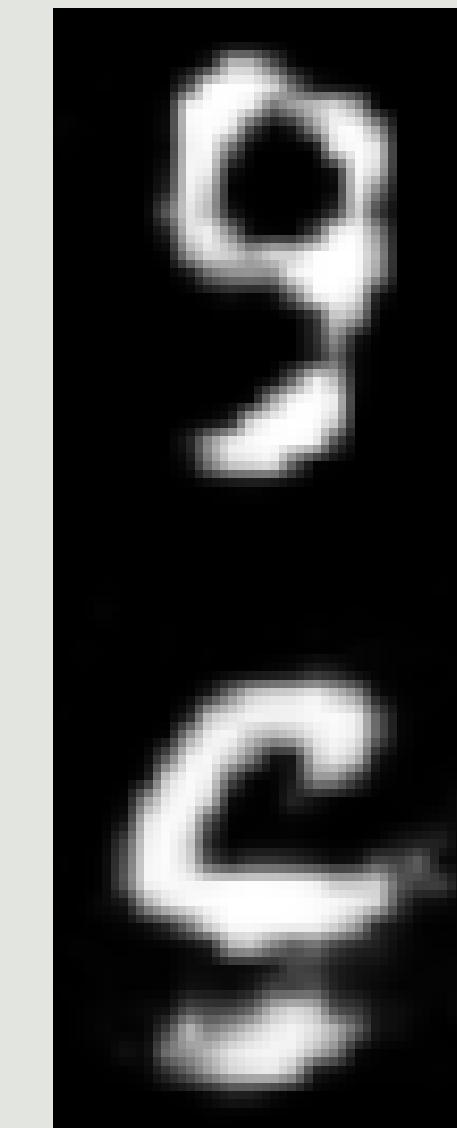
Epoch 3



Epoch 6



Epoch 9



Epoch 12

LIMITES DE LA MÉTHODE

- Apprend très vite une solution sous optimale et n'arrive pas à en sortir.
- Difficulté de stabiliser le générateur et le discriminateur

CONCLUSION

Théoriquement cette méthode est censée améliorer la convergence. En pratique sur le dataset MNIST elle n'améliore pas la fid

Hypothèse 1/2

- Optimisation des hypers paramètres (5 en plus donc gros couts computationnels)
- Cela peut être causé par mon implémentation

Hypothèse 3

- La méthode est plus performante sur des datasets de haute dimension (CIFAR STL)

REFERENCE

[https://arxiv.org/pdf/1811.01333](https://arxiv.org/pdf/1811.01333.pdf)

Improving GAN with neighbors embedding and
gradient matching Ngoc-Trung Tran * , Tuan-
Anh Bui * , Ngai-Man Cheung

MATHEMATIQUES

$$\mathcal{V}_R(E, G) = \sum_i \sum_j p_{i,j} \log \frac{p_{i,j}}{q_{i,j}}$$

$$p_{i,j} = \frac{p_{i|j} + p_{j|i}}{2n}$$

$$q_{i,j} = \frac{q_{i|j} + q_{j|i}}{2n}$$

$$p_{j|i} = \frac{(1 + \|z_j - z_i\|^2 / 2\sigma_z^2)^{-1}}{\sum_{k \neq i} (1 + \|z_k - z_i\|^2 / 2\sigma_z^2)^{-1}}$$

$$q_{j|i} = \frac{(1 + \|G(z_j) - G(z_i)\|^2 / 2\sigma_x^2)^{-1}}{\sum_{k \neq i} (1 + \|G(z_k) - G(z_i)\|^2 / 2\sigma_x^2)^{-1}}$$
 24