

---

# DOT appliqué aux WGAN

---

**Auteur :**

Paul Arrault

Double Licence Intelligence Artificielle et Sciences des Organisations (IASO)

**Encadrant :**

Professeur Alexandre Vérine

**Plan du Rapport :**

1. [Abstract](#)
2. [Introduction](#)
3. [Wasserstein GANs](#)
4. [Discriminator Optimal Transport](#)
5. [Comparaison](#)
6. [Discussion et Perspectives](#)

# 1. Abstract

Les GANs sont très importants pour la génération d'images. Dans le cadre d'un projet au sein du parcours et du cours Deep Learning 2, nous avons cherché à améliorer un GAN classique pour obtenir les meilleurs résultats possibles. Nous testerons tout ceci sur MNIST. J'ai tout d'abord voulu appliquer le Discriminator Optimal Transport (DOT) ([1]), méthode que j'ai finalement décidé d'appuyer par un WGAN. Il s'agira d'abord de tester un WGAN et de comprendre pourquoi l'appliquer, puis d'ajouter DOT.

## 2. Introduction

### 2.1. Concepts clés

**Distance de Wasserstein** La distance de Wasserstein est une métrique qui quantifie le coût minimal pour transformer une distribution  $P$  en une autre  $Q$ . Elle prend en compte la géométrie entière des données en examinant les distributions dans leur ensemble. Une caractéristique importante pour la suite est qu'elle reste bien définie même lorsque  $P$  et  $Q$  n'ont pas de support commun. Elle est définie comme :

$$W(P, Q) = \inf_{\gamma \in \Pi(P, Q)} \int_{\mathcal{X} \times \mathcal{Y}} \|x - y\| d\gamma(x, y),$$

où  $\Pi(P, Q)$  représente l'ensemble des couplages entre  $P, Q$ . Elle peut être exprimée comme une espérance, parfois plus facilement interprétable :  $W(P, Q) = \mathbb{E}_{x \sim P} [\min_{y \sim Q} \|x - y\|]$ .

**Norme Lipschitz** La norme Lipschitz mesure le taux de variation maximal d'une fonction  $f$ . Pour un GAN, une norme Lipschitz trop élevée ou trop faible est souvent signe d'instabilité. Elle aura **une grande utilité** pour les 2 méthodes que nous utiliserons, nous verrons comment la calculer en pratique. Mathématiquement, elle est donnée par :

$$\|f\|_{\text{Lip}} = \sup_{x, y} \frac{|f(x) - f(y)|}{\|x - y\|}.$$

### 2.2. Limites des GANs classiques

Avant de discuter des limites des GANs classiques, il est important de rappeler l'optimisation de leur fonction de perte, reposant sur l'optimisation conjointe du générateur  $G$  et du discriminateur  $D$ :

$$\min_G \max_D \left( \mathbb{E}_{x \sim P_{\text{data}}} [\log D(x)] + \mathbb{E}_{z \sim P_z} [\log(1 - D(G(z)))] \right).$$

**Divergence de Jensen-Shannon (JS)** La divergence JS, utilisée pour comparer les distributions (tout comme la distance de Wasserstein), est définie comme :

$$JS(P||Q) = \frac{1}{2}KL(P||M) + \frac{1}{2}KL(Q||M), \quad \text{où } M = \frac{1}{2}(P+Q), \quad \text{et} \quad KL(P||Q) = \int P(x) \log \left( \frac{P(x)}{Q(x)} \right) dx.$$

Lorsque  $P$  et  $Q$  n'ont pas de support commun,  $JS(P||Q)$  devient inefficace car  $KL(P||M)$  diverge. Cela se traduit par un discriminateur presque parfait et donc à un vanishing gradient ou un mode collapse.

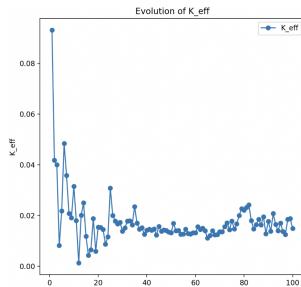
**Vanishing Gradient** Lorsque  $D(G(z)) \approx 1$ , le gradient du générateur devient très faible. Le gradient n'apporte plus assez d'informations, cela empêche le générateur de s'améliorer efficacement, ce qui se traduit par une stagnation de l'apprentissage. :

$$\nabla_G V(G, D) = \nabla_G \mathbb{E}_{z \sim P_z} [\log(1 - D(G(z)))] \approx 0.$$

**Mode Collapse** Pour contrer le discriminateur qui devient trop performant, le générateur se spécialise dans la production d'un sous-ensemble limité de la distribution cible, abandonnant la diversité. En fait, le générateur cherche à minimiser localement sa loss, mathématiquement:  $P_G(x) = \sum_{i=1}^k \delta(x - \mu_i)$ , où  $\mu_i$  représente les modes appris par le générateur.

**Instabilité de l'entraînement** L'équilibre entre le générateur  $G$  et le discriminateur  $D$  est donc difficile à atteindre. Il peut donc être intéressant d'exploiter d'autres méthodes.

**Difficulté pour calculer la norme Lipschitz** La norme Lipschitz est cruciale pour garantir des gradients stables et donc un entraînement stable. Avec le problème d'instabilité d'entraînement, elle devient rapidement inexploitable. Pourtant celle-ci est **primordiale** pour la méthode DOT. J'ai donc opté pour un modèle plus stable : le WGAN.



### 3. Wasserstein GANs

#### 3.1. Modèle

Les Wasserstein GANs ([2]) (WGANs) introduisent une nouvelle formulation de la fonction de perte pour résoudre les problèmes des GANs classiques. Leur principal objectif est d'assurer la stabilité de l'entraînement, donc de modifier la JS divergence avec une autre mesure. La distance de Wasserstein a été choisie, sa forme de base étant "assez hostile", une bonne méthode est de la modifier avec le théorème de Kantorovich-Rubinstein en :

$$W(P, Q) = \sup_{\|f\|_L \leq 1} \mathbb{E}_{x \sim P}[f(x)] - \mathbb{E}_{x \sim Q}[f(x)].$$

Donc, nous voulons approximer la fonction  $f(x)$  par  $D_\theta(x)$ . On a donc les nouvelles pertes suivante du générateur  $L_G$  et du discriminateur  $L_D$ :  $L_G = -\mathbb{E}_{x \sim Q}[D(x)]$ ,  $L_D = \mathbb{E}_{x \sim Q}[D(x)] - \mathbb{E}_{x \sim P}[D(x)]$ .

##### 3.1.1. Respecter la contrainte Lipschitz

Le discriminateur  $D$  doit alors respecter la contrainte Lipschitz inférieure à 1. Plusieurs méthodes existent, les principales étant le gradient clipping, gradient penalty et la normalisation spectrale.

J'ai d'abord opté pour la **normalisation spectrale**. Pour une transformation linéaire  $l_{W,b}$ ,  $\|l_{W,b}\|_{Lip} = \sigma(W)$ , où  $\sigma(W)$  représente la valeur singulière maximale de  $W$ . La normalisation spectrale  $SN$  est définie par :  $SN(l_{W,b}) = l_{\frac{W}{\sigma(W)}, b}$  et garantit que  $\|l_{W/\sigma(W)}\|_{Lip} = 1$ .

Dans le contexte des réseaux de neurones, cette normalisation permet de garantir que la norme Lipschitz du réseau est bornée grâce à cette propriété :  $\|f \circ g\|_{Lip} \leq \|f\|_{Lip} \cdot \|g\|_{Lip}$ .

Appliquée à un réseau de neurones avec des activations de normes inférieures à 1 et une normalisation spectrale à chaque couche, on obtient :  $\|f_{nn}\|_{Lip} \leq \prod_{l=1}^L \|l_{W_l/\sigma(W_l)}\|_{Lip} = 1$

Ainsi, cette normalisation assure que le réseau est conforme à la contrainte Lipschitz, stabilisant ainsi l'entraînement des GANs (figure 1). En revanche, cette méthode n'offre pas de contrôle sur sa valeur, ce qui peut ralentir l'entraînement avec une norme proche de 0 (figure 2).

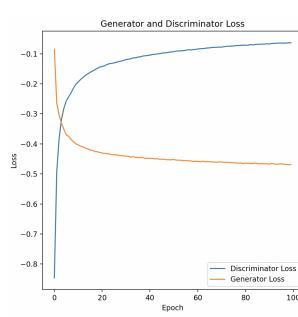


Figure 1

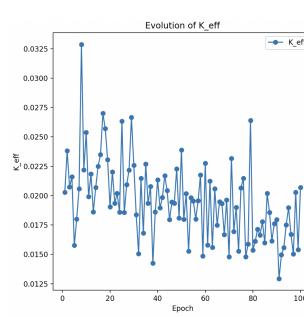


Figure 2

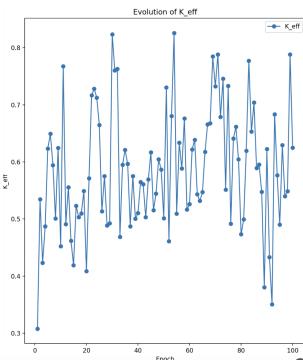


Figure 3

### 3.1.2. Le choix du Gradient Penalty

Une alternative est l'utilisation du Gradient Penalty (GP). Cette méthode ne se limite pas à contraindre, mais impose que celle-ci converge explicitement vers 1. J'ai choisi cette méthode car **avoir une norme spectrale égale à 1** sera primordiale pour le transport optimal.

La nouvelle fonction de perte pour le discriminateur devient alors :

$$L_D = \mathbb{E}_{x \sim Q}[D(x)] - \mathbb{E}_{x \sim P}[D(x)] + \lambda \cdot \mathbb{E}_{\hat{x} \sim P_{\hat{x}}} \left( (\|\nabla_{\hat{x}} D(\hat{x})\|_2 - 1)^2 \right),$$

où  $P_{\hat{x}}$  est une distribution interpolant linéairement les données réelles et générées. Plusieurs avantages :

- Plus doux que le gradient clipping, on évite les problèmes d'instabilité observés dans ce dernier.
- Elle garantit une norme proche 1 pour le discriminateur, ce qui est particulièrement utile pour DOT ! (voir section 4)

En pratique, bien que les variations autour de la norme cible soient attendues, celles-ci restent une estimation empirique du comportement global du discriminateur (figure 3).

## 3.2. Résultats

Le modèle a montré des résultats concluants. L'entraînement a été très stable, sans signe de *mode collapse* et a produit une diversité notable de chiffres. L'évolution au cours de l'entraînement est illustrée ci-dessous. Chaque chiffre évolue lentement mais converge vers une valeur sûre.

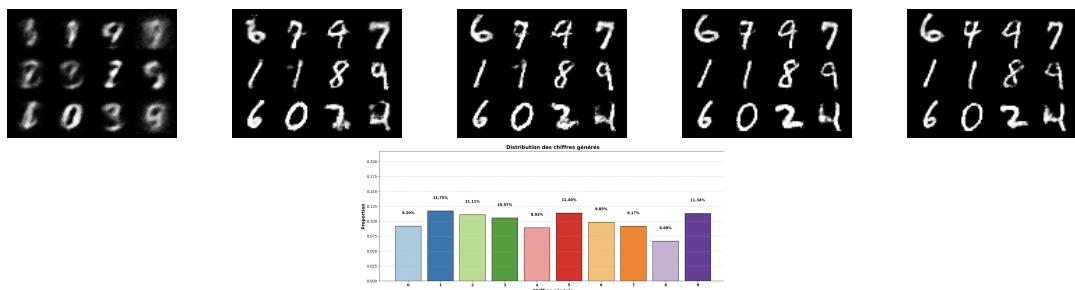


Figure 4: Évolution au cours de l'entraînement et diversité des chiffres générés.

**Paramètres d'entraînement :** Ratio D/G : 5 (40 premières époques), puis 1 ; Époques : 204 ; Learning rate :  $10^{-4}$  ; Optimiseur : Adam ( $\beta_1 = 0.0$ ,  $\beta_2 = 0.9$ ) ; Architecture : WGANGP.

## 4. Discriminator Optimal Transport

DOT repose sur le postulat suivant : dans un Auto-encodeur, l'encodeur et le décodeur sont utilisés conjointement, alors pourquoi ne pas exploiter D en post-train ? DOT vise à améliorer la qualité des images générées et à corriger les erreurs. Le transport optimale utilise la distance de Wasserstein (d'où mon choix de WGANGP).

### 4.1. Théorie

Tout repose sur les deux théorèmes suivants :

#### Théorème 1 :

*Hypothèses* :  $\pi^*$  et  $D^*$  sont des solutions optimales,  $\pi^*$  est un transport optimal déterministe décrit par  $T : X \rightarrow X$ .

#### Résultats :

$$\|D^*\|_{\text{Lip}} = 1,$$

$$T(y) = \arg \min_x \{\|x - y\|_2 - D^*(x)\},$$

$$p(x) = \int dy \delta(x - T(y))q(y)$$

#### Théorème 2 :

*Hypothèses* : Chaque fonction objective de GAN avec une pénalité sur le gradient fournit une borne inférieure de la divergence moyenne entre  $p$  et  $p_G$ .  $\tilde{D} = D/K$ , où  $K$  est le coefficient lipschitz de D.

#### Résultats :

$$V_D(G, D) \leq K (\mathbb{E}_{x \sim p} [\tilde{D}(x)] - \mathbb{E}_{y \sim p_G} [\tilde{D}(y)])$$

Les deux théorèmes soulignent le lien entre le transport optimal et le discriminateur GAN(1). Ils montrent comment maximiser la loss du discriminateur revient à minimiser la distance de Wasserstein et que cette méthode fonctionne pour n'importe quel GAN (2). On comprend pourquoi il était primordial que le coefficient Lipschitz converge vers 1, et pourquoi il est important de bien le calculer (très précisément).

**Calcul du coefficient Lipschitz** : On calcule la norme Lipschitz à partir de deux formules,  $K_{\text{eff}} = \max \left\{ \frac{|D(x) - D(y)|}{\|x - y\|_2} \mid x, y \sim p_G \right\}$  (espace image) et  $k_{\text{eff}} = \max \left\{ \frac{|D \circ G(z) - D \circ G(z')|}{\|z - z'\|_2} \mid z, z' \sim p_Z \right\}$  (espace latent); pour mes expériences, je prenais 50 000 couples, répétais 10 fois et faisais la moyenne des maximums, la moyenne pour éviter une valeur abérante.

On retiendra surtout que pour la version idéale, on a :  $T_D(y) = \arg \min_x \{\|x - y\|_2^2 - \frac{1}{K_{\text{eff}}} D(x)\}$ .

### 4.2. Target Optimal Transport

Le transport optimal dans l'espace des images, défini par  $T_D^{\text{eff}}(y) = \arg \min_x \left\{ \|x - y\|_2^2 - \frac{1}{K_{\text{eff}}} D(x) \right\}$ , permet d'optimiser la correspondance entre les échantillons générés et la distribution cible.

---

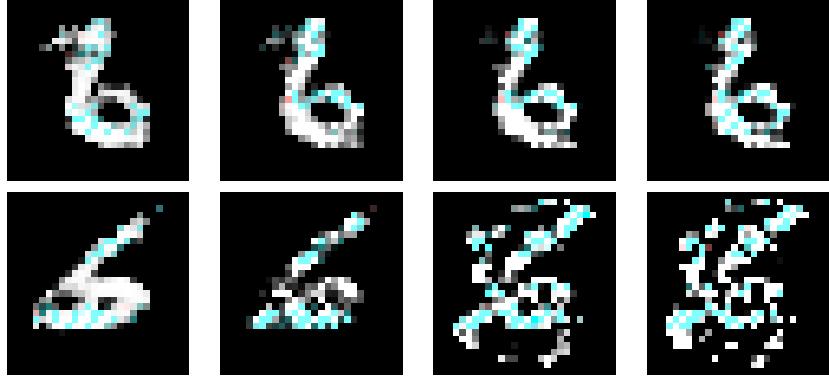
#### Algorithm 1 Transport Optimal dans l'Espace des Images (TOT)

---

**Require:** trained  $D$ , coefficient  $K_{\text{eff}}$ , échantillon  $y$ , taux d'apprentissage  $\epsilon$ , petit vecteur  $\delta$ .

- 1: Initialiser  $x \leftarrow y$ .
  - 2: **for**  $n_{\text{trial}}$  dans  $1, \dots, N_{\text{updates}}$  **do**
  - 3:   Mettre à jour  $x \leftarrow x - \epsilon \nabla_x \left( \|x - y\|_2^2 + \delta - \frac{1}{K_{\text{eff}}} D(x) \right)$ .
  - 4: **end for**
  - 5: **return**  $x$ .
- 

**Résultats** : Améliorations minimale ( 1ère ligne de la représentation qui se lit gauche à droite). Fonctionne pour des petites dimensions, donc ce n'est pas adapté pour notre cas (dimension 784). La méthode est très instable, en augmentant le nombre d'itération de 2, et en augmentant le learning rate de 0.05, on obtient un bruit illisible ( 2ème ligne)



### 4.3. Latent Space DOT

Le transport optimal dans l'espace latent, défini par  $T_{D \circ G}^{\text{eff}}(z_y) = \arg \min_z \left\{ \|z - z_y\|_2^2 - \frac{1}{k_{\text{eff}}} D \circ G(z) \right\}$ , vise à ajuster les représentations latentes pour améliorer la qualité des échantillons générés.

---

#### Algorithm 2 Transport Optimal dans l'Espace Latent (Latent Space DOT)

---

**Require:** Générateur  $G$ , Discriminateur  $D$ , coefficient  $k_{\text{eff}}$ , échantillon  $z_y$ , taux d'apprentissage  $\epsilon$ , petit vecteur  $\delta$ .

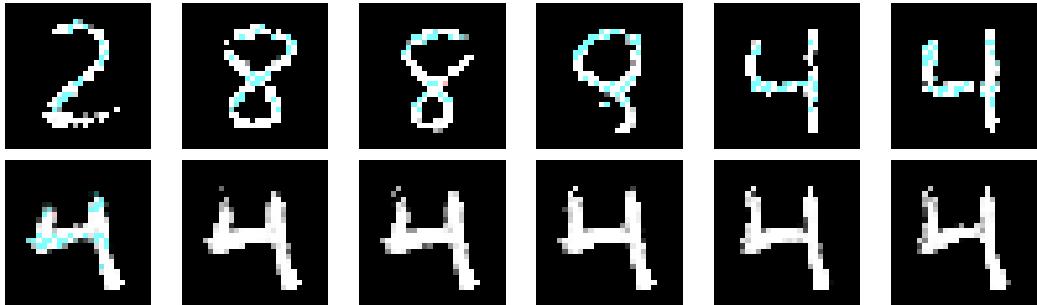
```

1: Initialiser  $z \leftarrow z_y$ .
2: for  $n_{\text{trial}}$  dans  $1, \dots, N_{\text{updates}}$  do
3:    $g \leftarrow \nabla_z \left\{ \|z - z_y\|_2^2 + \delta - \frac{1}{k_{\text{eff}}} D \circ G(z) \right\}$ .
4:   if le bruit est généré par  $\mathcal{N}(0, I_{D \times D})$  then
5:      $g \leftarrow g - (g \cdot z)z/\|D\|$ 
6:   end if
7:   Mettre à jour  $z \leftarrow z - \epsilon g$ .
8:   if le bruit est généré par  $\mathcal{U}([-1, 1])$  then
9:     Clip  $z \in [-1, 1]$ 
10:   end if
11: end for
12: return  $x = G(z)$ .

```

---

**Résultats :** On obtient une augmentation de la précision. Avec un grand learning rate on perd en diversité mais on gagne en précision (*1ère ligne*). L'objectif est alors de gagner moins de précision pour essayer de maintenir la diversité (*2ème ligne*).

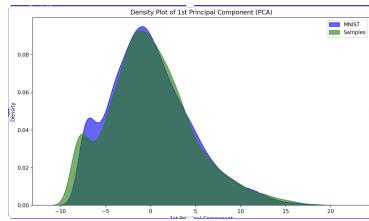


### 4.4. Comparaison

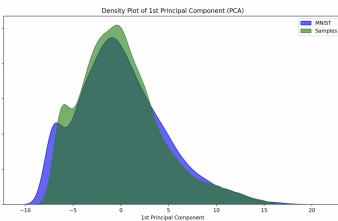
J'ai donc opté pour Latent Space DOT, l'enjeu n'était pas de faire du tuning d'hyperparamètres mais plutôt de comprendre les conséquences des modifications. Il se trouve qu'augmenter l'apprentissage menait à plus de précision ([graphique 2](#)), mais entraîne une perte de diversité. En revanche, en trouvant un juste milieu, on arrive à minimiser la perte de diversité tout en gardant la précision ([graphique 3](#)).

Mais, l'objectif est accompli, on gagne en précision par rapport au WGANGP qui était en quelque sorte notre benchmark ([graphique 1](#)). Pour mieux voir comment se passent ces déplacements dans l'espace latent et cet arbitrage entre précision et rappel, je vous propose d'aller sur mon [Canva](#).

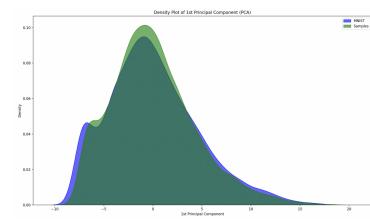
Pour rappel, la précision et le rappel se définissent comme :  $\bar{\alpha} = \hat{P}(\text{Supp}(P))$  et  $\bar{\beta} = P(\text{Supp}(\hat{P}))$ ,



Graphique 1 : WGANGP

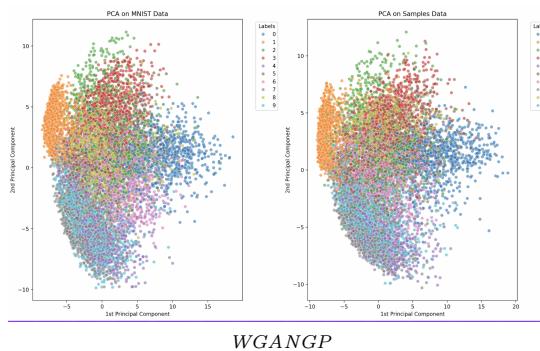


Graphique 2 : learning rate = 0.5

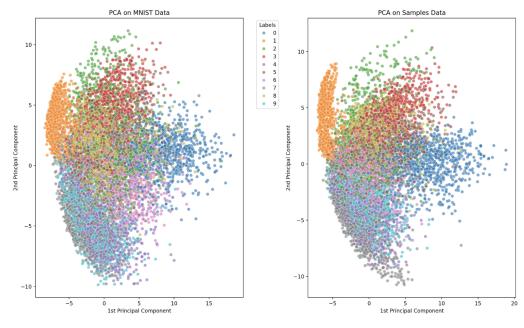


Graphique 3 : learning rate = 0.005

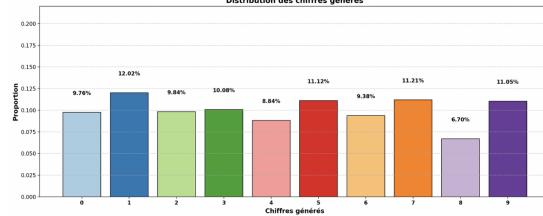
Je n'ai pas exploité le tuning, mais mon meilleures résultats s'est fait avec un learning rate de 0.05, ma norme lipschitz de 0.47, un nombre d'updates de 25, et  $\delta = 10^{-3}$ . Les schémas de visualisation d'espaces latents des données générées comparés à celles de MNIST sont présentés en dessous, avec un graphique sur la diversité des échantillons générés. Métriques : FID : 19.65, précision = 0.52, rappel = 0.32.



WGANGP



WGANGP + DOT



Graphique : Diversité des échantillons générés

## 5. Discussion et Perspectives

Les résultats obtenus sont globalement satisfaisants. Cependant, plusieurs pistes d'amélioration peuvent être explorées . Tout d'abord, avec du fine-tuning, notamment avec le learning rate, la norme Lipschitz et le nombre d'updates, afin d'améliorer à la fois la stabilité et la précision des résultats. Le calcul de la norme Lipschitz pourrait également être amélioré, ce calcul doit être réfléchi dans son execution pour assurer la mise en place de DOT. Ensuite, l'arbitrage entre précision et recall. Trouver un équilibre entre ces deux objectifs reste un défi majeur, ici encore, le fine-tuning est une piste. Malgré les résultats encourageants, certains artefacts persistent dans les images générées, témoignant d'instabilités résiduelles (répercussions des erreurs du modèle, c'est une limite de DOT, on part du principe que le discriminateur est "parfait") ou à de mauvaises convergences dans l'espace latent. Par ailleurs, une meilleure normalisation des vecteurs latents pourrait améliorer la qualité des échantillons générés en assurant une répartition plus homogène des représentations dans l'espace latent. Bien que la méthode ait été appliquée avec succès au WGANGP, il serait intéressant de l'évaluer sur d'autres variantes de GANs, comme les StyleGANs, BigGANs, FGAN afin de tester sa généralisabilité et son adaptabilité . Enfin, DOT, grâce à sa simplicité et son efficacité post-entraînement, constitue un outil rapide et intuitif pour visualiser les changements dans l'espace latent et les images générées.

## Références

- [1] DOT : Gabriel Peyré, Marco Cuturi, et al. "Computational Optimal Transport." <https://arxiv.org/abs/1803.0056>
- [2] WGAN : Martin Arjovsky, Soumith Chintala, Léon Bottou. "Wasserstein GAN." <https://arxiv.org/abs/1701.0787>