

Préparée à Université Paris-Dauphine

Rapport projet Optimal Budget Rejection Sampling (OBRS) appliqué aux GANs sur MNIST

Spécialité DEEP LEARNING

Cameron Mouangue



Résumé

Ce rapport présente une étude approfondie de la méthode **Optimal Budget Rejection Sampling** (OBRS) appliquée aux réseaux adverses génératifs (GANs), avec un focus particulier sur le dataset MNIST. La structure inclut une présentation théorique, une implémentation détaillée, ainsi qu'une analyse des résultats expérimentaux.

Table des matières

1	Présentation Théorique de OBRS	2
2	Implémentation et Adaptation de OBRS sur MNIST	3
2.1	Les différents modèles testés	3
2.2	Pseudocode pour les calculs de M , c_K et $a_O(x)$	3
2.2.1	Calcul de M	3
2.2.2	Calcul de c_K via dichotomie	3
2.2.3	Calcul de $a_O(x, c_K)$	4
2.3	Échantillonnage avec OBRS	4
2.4	Problèmes rencontrés	4
2.5	Solutions techniques adoptées	4
3	Résultats Expérimentaux	4
3.1	Comparaison des modèles testés	4
3.1.1	Résultats sur la précision et le rappel	4
3.2	Analyse des effets de K	5
3.3	Évolution au cours de l'entraînement	5
3.4	Images générées et analyse visuelle	5
4	Conclusion et Perspectives	6
4.1	Résumé des résultats	6
4.2	Directions futures	6
4.3	Conclusion générale	6

Chapitre 1

Présentation Théorique de OBRS

Définitions et Concepts Fondamentaux

La distribution cible P et la distribution générée \hat{P}

Dans le cadre des modèles génératifs adverses (GANs), nous cherchons à approximer une distribution cible P , qui représente la distribution réelle des données, par une distribution générée \hat{P} produite par le générateur G . La divergence entre P et \hat{P} est utilisée comme mesure pour évaluer la qualité de l'approximation.

La f -divergence

La f -divergence est une classe générale de divergences entre deux distributions P et \hat{P} . Elle est définie comme suit :

$$D_f(P\|\hat{P}) = \int p(x) f\left(\frac{p(x)}{\hat{p}(x)}\right) dx,$$

où f est une fonction convexe telle que $f(1) = 0$. Les f -divergences incluent des métriques bien connues :

- La divergence de Kullback-Leibler (KL-divergence), pour $f(u) = u \log(u)$.
- La divergence Jensen-Shannon, utilisée dans les GANs traditionnels.
- La divergence de variation totale.

La distribution post-rejection \tilde{P}

Lorsque l'on applique une méthode de rejection sampling, on obtient une nouvelle distribution \tilde{P} , appelée distribution post-rejection. Sa densité est définie par :

$$\tilde{p}(x) = \frac{\hat{p}(x)a(x)}{Z},$$

où $a(x)$ est une fonction d'acceptation, et $Z = \mathbb{E}_{\hat{P}}[a(x)]$ est un facteur de normalisation garantissant que \tilde{P} est une distribution valide.

Optimal Budget Rejection Sampling (OBRS)

Définition et Fonctionnement

OBRS est une méthode optimisée pour le rejection sampling dans un contexte où le budget d'échantillonnage K est limité. Elle cherche à minimiser la divergence $D_f(P\|\tilde{P})$ sous une contrainte sur K , le nombre moyen d'échantillons nécessaires pour accepter un échantillon.

Fonction d'acceptation $a(x)$

La fonction $a(x)$ détermine si un échantillon x généré à partir de \hat{P} est accepté ou rejeté. OBRS propose une fonction d'acceptation optimale donnée par :

$$a_0(x) = \min\left(\frac{p(x)}{\hat{p}(x)} \cdot \frac{c_K}{M}, 1\right),$$

où :

- $p(x)$ est la densité de la distribution cible P ,
- $\hat{p}(x)$ est la densité de la distribution générée \hat{P} ,
- c_K est une constante calculée pour respecter la contrainte $\mathbb{E}_{\hat{P}}[a(x)] = 1/K$,
- $M = \sup_{x \in X} \frac{p(x)}{\hat{p}(x)}$ est la valeur maximale de $\frac{p(x)}{\hat{p}(x)}$ sur l'ensemble X .

Amélioration du Precision-Recall

- **Précision** (α) : Mesure la proportion des échantillons acceptés par OBRS qui appartiennent à P . - **Rappel** (β) : Mesure la couverture de P par les échantillons acceptés.

Impact de OBRS sur les Courbes PR

OBRS améliore la précision pour un rappel donné. Cela est théoriquement démontré par une mise à l'échelle verticale des courbes PR initiales (voir Annexe Fig. A.1).

OBRS avec Entraînement vs Post-Rejection

OBRS uniquement Post-Rejection

Lorsque OBRS est appliqué uniquement après l'entraînement du générateur, il améliore la qualité des échantillons générés sans influencer le processus d'entraînement.

OBRS intégré à l'Entraînement

En intégrant OBRS directement dans la fonction de perte du GAN, on optimise conjointement le générateur et la fonction d'acceptation. Cela permet :

- Une meilleure couverture de P par \tilde{P} .
- Une réduction des minima locaux dans le paysage de perte (voir Annexe Fig. A.2).

Formulation de la Perte et Gains Théoriques

Formulation de la Perte

La perte avec OBRS intégré est définie comme suit :

$$\mathcal{L}_{OBRS} = D_f(P\|\tilde{P}) = \mathbb{E}_{\hat{P}} \left[K a_0(x) f\left(\frac{r(x)}{K a_0(x)}\right) \right],$$

où :

- $r(x) = \frac{p(x)}{\hat{p}(x)}$ est le ratio de vraisemblance (likelihood ratio),
- K est le budget d'échantillonnage,

Gains Théoriques

Les résultats théoriques montrent que OBRS réduit significativement $D_f(P\|\tilde{P})$ et améliore :

- La précision pour un rappel donné.
- La stabilité de l'entraînement grâce à une meilleure couverture des modes de P .

Chapitre 2

Implémentation et Adaptation de OBRS sur MNIST

2.1 Les différents modèles testés

Pour comprendre l'impact de OBRS, plusieurs versions ont été implémentées, chacune intégrant OBRS à différents niveaux du processus :

1. **Version 0 (Baseline)** : Un GAN standard sans rejection sampling, utilisé comme référence.
2. **Version 1 (OBRS sur la génération uniquement)** : La méthode OBRS est appliquée uniquement après la génération des échantillons, sans impact sur l'entraînement, avec un budget fixé à $K = 2$.
3. **Version 2 (OBRS complet, $K = 2$)** : OBRS est intégré dans les phases de génération et d'entraînement, avec un budget fixé à $K = 2$.
4. **Version 3 (OBRS sur l'entraînement uniquement)** : OBRS est utilisé uniquement pour régulariser l'entraînement du générateur.
5. **Version 4 (OBRS complet, $K > 2$)** : Une version plus générale où le budget K est augmenté pour évaluer son impact.

Chaque version permet d'isoler l'effet de OBRS sur différentes étapes du pipeline et d'analyser ses contributions individuelles.

2.2 Pseudocode pour les calculs de M , c_K et $a_O(x)$

2.2.1 Calcul de M

Algorithm 1 Calcul de M

Require: G : Générateur, D : Discriminateur, n : Nombre d'itérations, d : Taille du batch, $\epsilon = 10^{-8}$: Facteur de régularisation.

Ensure: M : Estimation maximale du ratio $p(x)/\hat{p}(x)$.

```
1:  $M \leftarrow -\infty$ 
2: for  $i \in \{1, \dots, n\}$  do
3:   Générer un batch  $z$  tel que  $z_{i,j} \sim \mathcal{N}(0, 1)$  pour tout  $i \in \{1, \dots, d\}$  et  $j \in \{1, \dots, 100\}$ .
4:    $x_{\text{fake}} \leftarrow G(z)$ 
5:    $r(x) \leftarrow \frac{D(x_{\text{fake}})}{1 - D(x_{\text{fake}}) + \epsilon}$ 
6:    $M \leftarrow \max(M, \max(r))$ 
7: end for
8: return  $M$ 
```

2.2.2 Calcul de c_K via dichotomie

Algorithm 2 Calcul de c_K

Require: N : Nombre d'échantillons générés, $x_{\text{fake}} = (x_{\text{fake}1}, \dots, x_{\text{fake}N})$: Échantillons générés, P : Distribution cible.
 K : Budget, ϵ : Seuil de convergence.

Ensure: c_K : Constante calculée.

```
1:  $c_{\min} \leftarrow 1 \times 10^{-10}$ ,  $c_{\max} \leftarrow 1 \times 10^{10}$ 
2:  $c_K \leftarrow \frac{c_{\max} + c_{\min}}{2}$ 
3: Définir la perte  $L(c_K) = \sum_{i=1}^N a(x_{\text{fake}i}, c_K) - \frac{1}{K}$ 
4: while  $|L(c_K)| \geq \epsilon$  do
5:   if  $L(c_K) > \epsilon$  then
6:      $c_{\max} \leftarrow c_K$ 
7:   else if  $L(c_K) < -\epsilon$  then
8:      $c_{\min} \leftarrow c_K$ 
9:   end if
10:  Mise à jour :  $c_K \leftarrow \frac{c_{\max} + c_{\min}}{2}$ , Mise à jour :  $L(c_K)$ 
11: end while
12: return  $c_K$ 
```

Algorithm 3 Calcul de $a_O(x, c_K)$

Require: D : Discriminateur, x_{fake} : Échantillons générés, M, c_K : Constantes, $\epsilon = 10^{-8}$: Facteur de régularisation.

Ensure: $a_O(x, c_K)$: Probabilité d'acceptation.

- 1: $r(x) \leftarrow \frac{D(x_{\text{fake}})}{1-D(x_{\text{fake}})+\epsilon}$
 - 2: $a_O(x, c_K) \leftarrow \min\left(\frac{p(x)}{\hat{p}(x)} \cdot \frac{c_K}{M}, 1\right)$
 - 3: **return** $a_O(x, c_K)$
-

2.3 Échantillonnage avec OBRS

L'échantillonnage suit le processus suivant : 1. Générer des échantillons avec le générateur G . 2. Calculer $a(x)$ à l'aide des constantes c_K et M . 3. Accepter ou rejeter les échantillons en utilisant une distribution de Bernoulli basée sur $a(x)$.

Algorithm 4 Échantillonnage avec OBRS

Require: G, D : Générateur et Discriminateur, K : Budget, M, c_K : Constantes.

Ensure: samples : Échantillons acceptés.

- 1: $n_{\text{generated}} \leftarrow 0$, samples $\leftarrow \emptyset$
 - 2: **while** $n_{\text{generated}} < \text{num_samples}$ **do**
 - 3: Générer un batch z tel que $z_{i,j} \sim \mathcal{N}(0, 1)$ pour tout $i \in \{1, \dots, d\}$ et $j \in \{1, \dots, 100\}$, $x_{\text{fake}} \leftarrow G(z)$.
 - 4: Calculer $r(x)$ et $a(x)$ (voir Algorithme 3).
 - 5: Générer un masque $\text{accept_mask} \sim \text{Bernoulli}(a(x))$.
 - 6: Ajouter les échantillons acceptés à samples.
 - 7: $n_{\text{generated}} \leftarrow n_{\text{generated}} + \text{len}(x_{\text{accepted}})$.
 - 8: **end while**
 - 9: **return** samples
-

2.4 Problèmes rencontrés

1. **Complexité des calculs de M et c_K** : Pour chaque itération de la boucle train, calculer M nécessite plusieurs itérations sur des échantillons générés pour estimer le maximum de $r(x)$, ce qui est coûteux. De même, c_K doit être ajusté via une méthode de dichotomie pour garantir que $\mathbb{E}[a(x)] = \frac{1}{K}$, ce qui est également cher en temps de calcul.
2. **Instabilités dues à la fonction de perte personnalisée** : La fonction de perte pour le générateur, basée sur la régularisation par OBRS, introduit des instabilités (explosion gradient, division par 0...). Ces instabilités proviennent principalement de $r(x)$ et des gradients calculés à partir de $a(x)$.

2.5 Solutions techniques adoptées

1. **Clamp des ratios $r(x)$** : Limiter $r(x)$ à une plage raisonnable $[0.01, 10.0]$ pour éviter les valeurs extrêmes.
2. **Gradient Clipping** : Appliquer un seuil de gradient pour stabiliser l'entraînement, avec une norme maximale fixée à 1.0.
3. **Mise à jour périodique de M et c_K** : Recalculer M et c_K toutes les n -itérations au lieu de chaque itération, ce qui réduit considérablement le coût.

Chapitre 3

Résultats Expérimentaux

3.1 Comparaison des modèles testés

Pour évaluer l'efficacité de la méthode OBRS, nous avons comparé les cinq modèles décrits précédemment.

3.1.1 Résultats sur la précision et le rappel

Les courbes Precision-Recall (PR) permettent de mesurer à la fois la qualité (*précision*) et la diversité (*rappel*) des échantillons générés. Les résultats sont présentés dans la Figure 3.1.

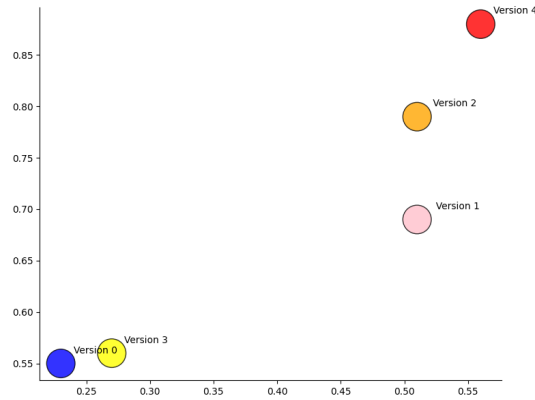


FIGURE 3.1 – Comparaison des métriques Precision-Recall pour les cinq modèles.

Analyse des résultats :

- **Baseline (V0) :** La précision est modérée (0.55), et le rappel est faible (0.23), reflétant la limitation classique des GANs à capturer la diversité des données réelles.
- **Version 1 :** Une amélioration importante de la précision (0.69) est observée, car OBRS élimine les échantillons aberrants après la génération. Le rappel augmente lui aussi (0.51) car grâce au budget K , le générateur en refusant les samples de mauvaises qualités, a plus d'opportunité à couvrir une part plus importante de P .
- **Version 2 :** Une augmentation importante de la précision (0.79) et du rappel (0.51), car il bénéficie des mêmes avantages que V1, mais en plus OBRS est intégré dans le processus d'entraînement, permettant au générateur de s'adapter à la contrainte de rejet, et de produire des samples plus proches de P . **Hypothèse :** le recall n'augmente pas par rapport à V1 car le générateur utilisé est limité.
- **Version 3 :** Une amélioration légère en précision (0.56) et notable en recall (0.27). la performance augmente car OBRS est utilisé dans l'entraînement. **Hypothèse :** la performance augmente peu par rapport à V0 car le générateur utilisé est limité.
- **Version 4 :** Avec $K = 5$, la précision (0.88) et le rappel (0.66) augmentent par rapport à $K = 2$, montrant les bénéfices d'un budget plus permissif qui peut se permettre d'accepter que les meilleurs samples, et donc donne au générateur encore plus d'opportunité à couvrir une part plus importante de P .

3.2 Analyse des effets de K

En testant $K = 5$ et $K = 10$ pour la Version 4, nous avons observé des limites sur certaines métriques :

- **FID (Frechet Inception Distance) :** Une amélioration du FID est observée pour $K = 10$ (~ 12) par rapport à $K = 5$ (~ 15), suggérant que les valeurs aberrantes sont mieux éliminées avec un budget plus strict.
- **Precision-Recall (PR) :** Les métriques PR restent stables entre $K = 5$ et $K = 10$, ce qui indique que l'amélioration du FID ne traduit pas une meilleure qualité générale, probablement en raison de la limitation intrinsèque du générateur qui ne peut plus produire des samples de meilleure qualité en dépit de la hausse de K .

3.3 Évolution au cours de l'entraînement

Pour le modèle Baseline, l'entraînement a été réalisé sur 100 epochs avec un taux d'apprentissage de 2×10^{-4} pour G et D . Toutefois pour les modèles OBRS, le discriminateur apprenait plus rapidement que le générateur, ce qui empêchait une convergence optimale. Pour résoudre ce problème, nous avons réduit le taux d'apprentissage du discriminateur à 2×10^{-5} pour les 100 premiers epochs, ce qui a été appliqué à tous les modèles OBRS train. (voir Annexe Fig. B.1)

Pour la version 4 (V4), l'entraînement a été prolongé jusqu'à 300 epochs, avec une réduction progressive du taux d'apprentissage par un facteur de 0.5 tous les 100 epochs. Cela a permis une meilleure convergence, mais les performances ont atteint un plateau après 300 epochs. Ceci supporte la théorie sur "l'aplatissement" de la fonction perte car contrairement au modèle V4, le modèle V0 atteignait déjà sa limite dès 100 epochs.

3.4 Images générées et analyse visuelle

Les Figures B.2 à B.4 (voir Annexe) montrent des exemples d'images générées pour chaque modèle. Ces visualisations renforcent les observations suivantes :

- Les modèles avec OBRS produisent des échantillons de qualité visuelle supérieure, en éliminant les artefacts présents dans la baseline.
- Avec $K > 2$, les échantillons respectent une acceptation plus stricte et sont donc de meilleures qualités.

Chapitre 4

Conclusion et Perspectives

4.1 Résumé des résultats

Ce travail a permis d'évaluer en profondeur la méthode **Optimal Budget Rejection Sampling (OBRS)** appliquée aux réseaux adverses génératifs (GANs) sur le dataset MNIST. Les principales conclusions sont les suivantes :

- Les versions intégrant OBRS (V1 à V4) montrent des améliorations notables en termes de précision, de rappel, et de diversité des modes par rapport au baseline. Ces résultats confirment l'efficacité de OBRS pour rejeter les échantillons aberrants tout en permettant une diversité élevée.
- Les modèles combinant OBRS à la fois dans l'entraînement et la génération (V2 et V4) sont les plus performants. Ces résultats supportent la théorie selon laquelle l'intégration d'OBRS dans les deux phases maximise la convergence entre la distribution générée et la distribution réelle.
- Les tests sur des budgets plus élevés ($K = 5$ et $K = 10$) montrent des **diminishing returns** : bien que le FID s'améliore légèrement pour $K = 10$, les métriques Precision-Recall (PR) restent stables. Cependant, ce léger avantage se fait au prix d'une augmentation significative du temps de génération, passant de $T \approx 105$ secondes pour $K = 5$ à $T \approx 550$ secondes pour $K = 10$.
- L'analyse des performances au cours de l'entraînement montre que OBRS permet une meilleure convergence grâce aux propriétés de la loss f -divergence, en limitant les minima locaux et en améliorant la stabilité.

Ces résultats démontrent que OBRS est une méthode efficace pour améliorer les performances des GANs, tout en mettant en évidence les limites des architectures simples pour capturer des distributions complexes.

4.2 Directions futures

Les résultats obtenus ouvrent plusieurs pistes pour des recherches futures :

1. **Application à des datasets plus complexes** : Tester OBRS sur des datasets plus complexes pourrait permettre d'évaluer son efficacité dans des contextes plus complexes, où les distributions cibles ont une plus grande variance et où les modes sont plus nombreux.
2. **échantillonnage adaptatif** : Des recherches futures pourraient également explorer des schémas d'échantillonnage adaptatif, où K varie dynamiquement en fonction des performances observées, afin d'optimiser automatiquement le compromis entre qualité et temps de calcul.

4.3 Conclusion générale

Ce projet a démontré que la méthode OBRS constitue une avancée significative pour améliorer les performances des GANs. En intégrant un rejet optimisé dans les phases d'entraînement et de génération, OBRS permet d'augmenter nettement la performance en precision-recall d'un modèle GAN classique. Les résultats obtenus confirment que OBRS est une méthode prometteuse, avec un potentiel significatif pour des applications plus complexes à l'avenir.

Bibliographie

- [1] A. Verine, M. Sreenivas Pydi, B. Negrevergne, Y. Chevalyre *Optimal Budget Rejection Sampling for Generative Models*, 2024.
- [2] S. Nowozin, B. Cseke, R. Tomioka, *f-GAN : Training Generative Neural Samplers using Variational Divergence Minimization*, Advances in Neural Information Processing Systems (NeurIPS), 2016.
- [3] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, *Generative Adversarial Networks*, Advances in Neural Information Processing Systems (NeurIPS), 2014.

Annexes

A. Améliorations Théoriques

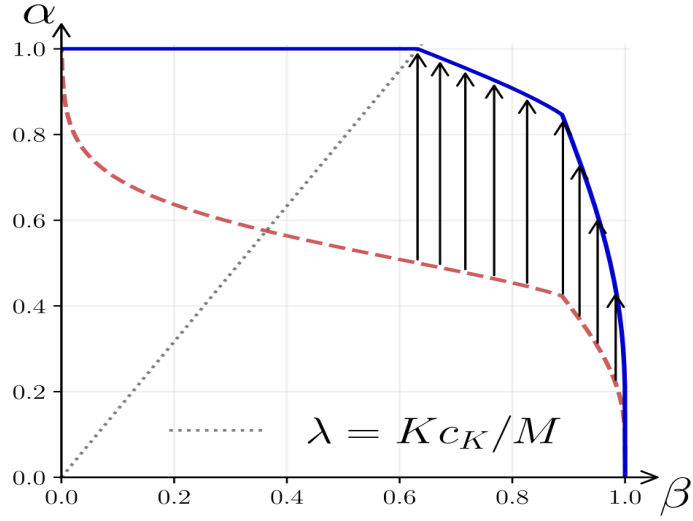


FIGURE A.1 – Amélioration théorique des courbes Precision-Recall grâce à OBRS. La mise à l'échelle verticale démontre une augmentation de la précision pour un rappel donné.

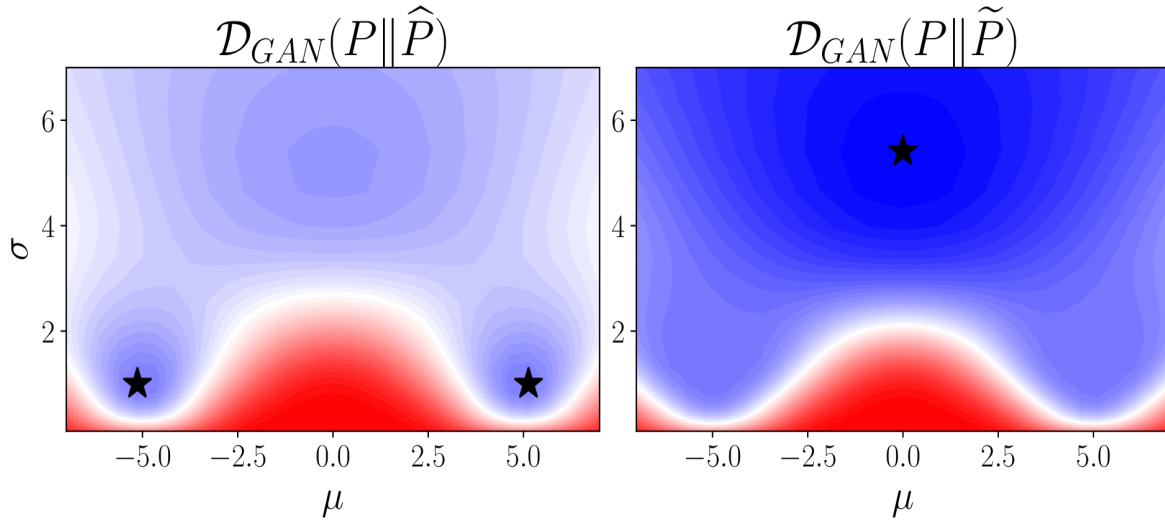


FIGURE A.2 – Aplatissement de la fonction de perte grâce à OBRS. La régularisation introduite réduit les minima locaux, améliorant la stabilité de l'entraînement.

B. Résultats Expérimentaux

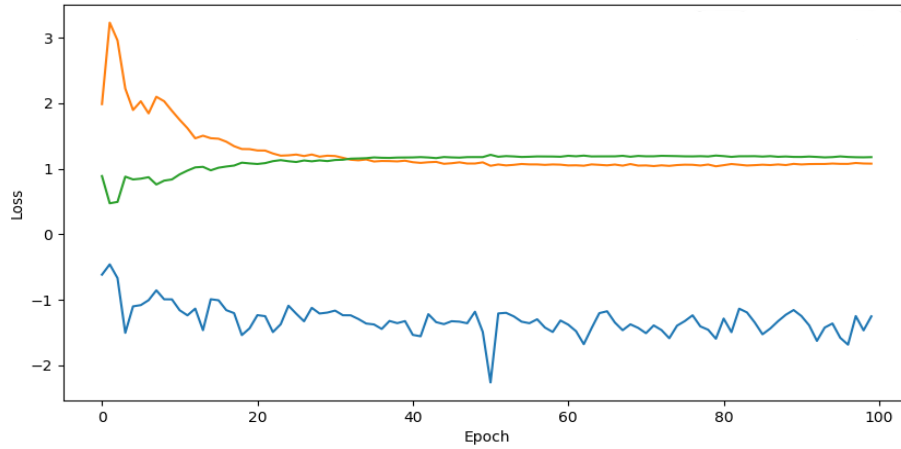


FIGURE B.1 – Évolution des pertes sur les 100 premières époques. En bleu, la perte OBRS (D_f), en vert, la perte BCE pour le discriminateur, et en orange, la perte BCE pour le générateur.

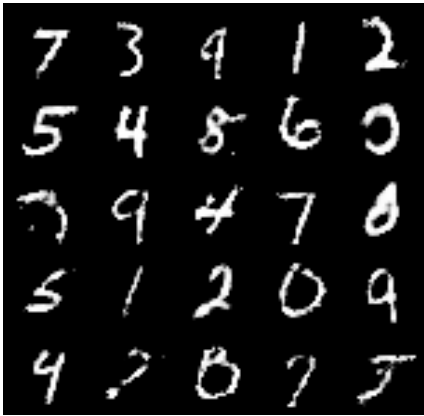


FIGURE B.2 – Exemples d'images générées par la Version 0 (Baseline). Les artefacts sont évidents.

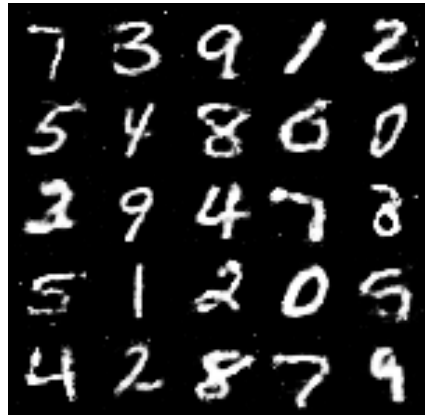


FIGURE B.3 – Exemples d'images générées par la Version 2 (OBRS complet, $K = 2$). Une meilleure qualité visuelle mais on observe des imperfections



FIGURE B.4 – Exemples d'images générées par la Version 4 (OBRS complet, $K = 5$). Les échantillons sont de très bonne qualité, suppression totale des artefacts.