GENDI

Leo S., Arthur H., Arthur G., Paul P., Pierre B., Etienne S.

Janvier 2026

GENDI

Contents

Résumé du projet

Les modèles génératifs profonds ont révolutionné le domaine de la vision par ordinateur ces dernières années. Alors que les GAN sont longtemps dominé le domaine malgré leur instabilité d'entraînement, et que les VAEs souffrent souvent d'un flou dans les générations, une nouvelle famille de modèles a émergé comme le nouvel état de l'art: les Modèles de Diffusion Probabilistes.

Ce projet, intitulé GENDI, a pour objectif de démystifier le fonctionnement de ces modèles complexes. Nous nous concentrons sur l'implémentation *from scratch* d'un DDPM appliqué au dataset MNIST. L'enjeu est double: comprendre la dérivation mathématique rigoureuse qui permet de transformer un bruit gaussien en une image structurée, et mettre en œuvre l'architecture neuronale (basée sur un U-Net) capable d'inverser ce processus de diffusion. Ce rapport détaille la théorie sous-jacente et la mise en œuvre pratique de ce modèle, avant d'analyser la qualité des chiffres manuscrits générés.

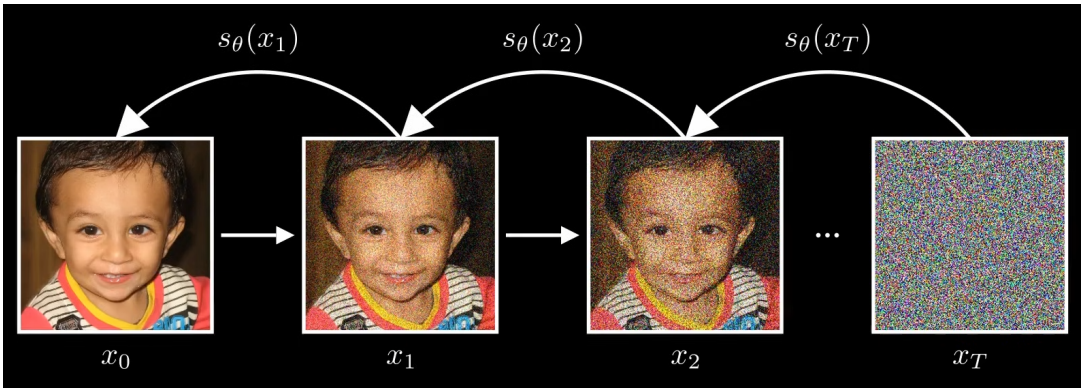
1 Introduction : diffusion (DDPM)

Notre sujet est centré autour d'un papier de recherche : *DDPM* (*Denoising Diffusion Probabilistic Models* [1]).

Le point de départ est un dataset d'images. Derrière ce dataset se cache une distribution inconnue, que l'on note $p(x)$.

Le but d'un modèle génératif est d'apprendre cette distribution afin de générer de nouvelles images qui lui ressemblent.

Les modèles de diffusion proposent une idée simple : au lieu d'essayer de modéliser directement $p(x)$, on va progressivement détruire l'image en ajoutant du bruit, jusqu'à obtenir du bruit pur.



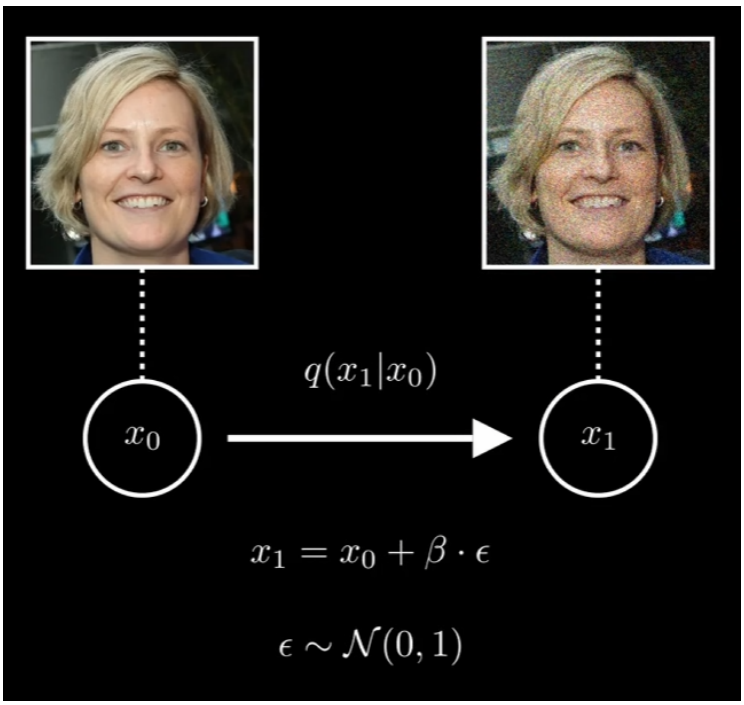
On note $x_0, x_1, \ldots, x_T$ les images obtenues au fil du processus, où T est un *timestep* (c'est-à-dire l'étape de diffusion à laquelle on se trouve). Ensuite, on voudra faire exactement l'inverse : partir du bruit et revenir vers une image réaliste. Le processus *forward* ajoute du bruit, et le *backward* l'enlève : c'est l'intuition générale.

1.1 Processus forward : une étape

Regardons une seule étape du processus forward.

On part d'une image propre x_0 et on génère une version légèrement bruitée x_1. Mathématiquement, on définit une distribution conditionnelle $q(x_1 \mid x_0)$, de la forme :

$$x_1 = x_0 + \sqrt{\beta}\,\varepsilon, \qquad \varepsilon \sim \mathcal{N}(0,1). \tag{1}$$

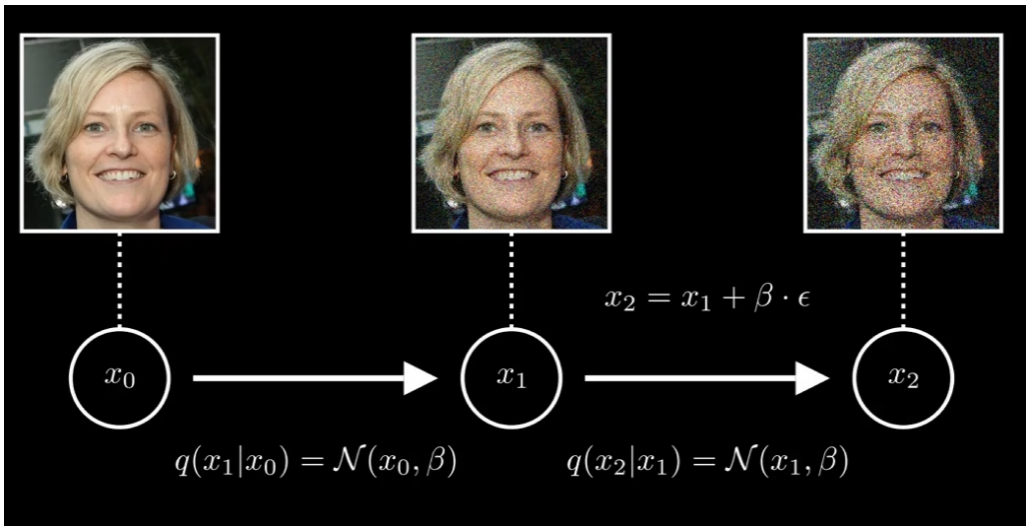


Autrement dit, x_1 est obtenu en ajoutant à x_0 un bruit gaussien contrôlé par un scalaire β. Intuitivement, on échantillonne autour de l'image d'origine : l'image reste ressemblante, mais partiellement détruite.

1.2 Processus forward : répétition

Si on répète la même opération plusieurs fois, à chaque étape on reprend l'image précédente et on ajoute encore un peu de bruit.

Par exemple, pour obtenir x_2, on part de x_1 et on ajoute à nouveau un bruit contrôlé par un scalaire.



$$x_2 = x_1 + \sqrt{\beta}\,\varepsilon_2, \qquad \varepsilon_2 \sim \mathcal{N}(0,1). \tag{2}$$

On précise que ε_1 et ε_2 sont indépendants. En remplaçant x_1 dans l'expression de x_2, on obtient :

$$x_2 = \left(x_0 + \sqrt{\beta}\,\varepsilon_1\right) + \sqrt{\beta}\,\varepsilon_2 \tag{3}$$

$$= x_0 + \sqrt{\beta}\,(\varepsilon_1 + \varepsilon_2). \tag{4}$$

Comme ε_1 et ε_2 sont indépendants et suivent chacun une loi $\mathcal{N}(0,1)$, leur somme suit une loi normale :

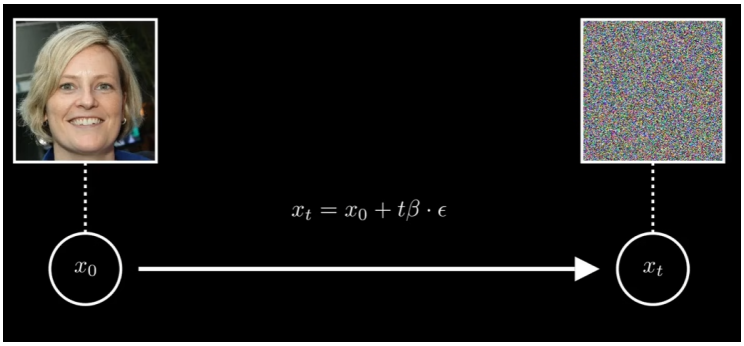
$$\varepsilon_1 + \varepsilon_2 \sim \mathcal{N}(0,2). \tag{5}$$

On peut donc écrire $\varepsilon_1 + \varepsilon_2 = \sqrt{2}\,\varepsilon$ avec $\varepsilon \sim \mathcal{N}(0,1)$.
Ainsi :

$$x_2 = x_0 + \sqrt{\beta}\,\sqrt{2}\,\varepsilon \tag{6}$$

$$= x_0 + \sqrt{2\beta}\,\varepsilon, \qquad \varepsilon \sim \mathcal{N}(0,1). \tag{7}$$

Au bout de t étapes, on peut écrire directement une distribution de x_t conditionnellement à x_0 : on obtient toujours une gaussienne centrée sur x_0, mais avec une variance qui augmente avec le nombre d'étapes.



Plus t est grand, plus l'image est proche du bruit pur. À la fin du processus forward, on a quasiment perdu toute information sur l'image originale.

1.3 Processus forward : Diffusion variance explosion

Ainsi, on peut généraliser comme :

$$x_t = \mathcal{N}(x_0, t\beta) \tag{8}$$

Un problème apparaît rapidement. Plus t est grand, plus la variance le sera et ce, sans limite. La solution retenue pour empêcher cela est de faire tendre $\mathcal{N}(x_0, t\beta)$ vers une $\mathcal{N}(0, 1)$ (loi normale centrée réduite).

De plus, cela permet de simplifier les différents calculs que ce soit pour la forward pass, la backward pass ou encore pour créer l'image de base pour la génération.

Pour permettre cela, on va introduire un nouveau coefficient $\sqrt{1-\beta}$. On obtient donc :

$$q(x_t|x_{t-1}) = \sqrt{1-\beta}x_{t-1} + \sqrt{\beta}\epsilon \tag{9}$$

Le choix de ce coefficient est arbitraire et n'a pas été expliqué pas les auteurs.

On peut donc généraliser pour n'importe quel t :

$$q(x_t|x_0) = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{(1-\bar{\alpha}_t)}\epsilon \tag{10}$$

Avec $\bar{\alpha}_t = (1-\beta)^t$.

Néanmoins, il est important de noter que cette expression n'est valable que pour un β fixe. Dans la réalité, β n'est pas fix, il est scheduler entre 0 et 1. A chaque step, β change donc. On peut donc réécrire l'équation (10) par :

$$q(x_t|x_0) = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{(1-\bar{\alpha}_t)}\epsilon \tag{11}$$

Avec $\bar{\alpha}_t = \prod_{i=1}^{t}(1-\beta_i)$.

Pour visualiser ce comportement, on choisit la distribution suivante :

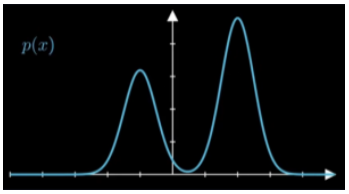
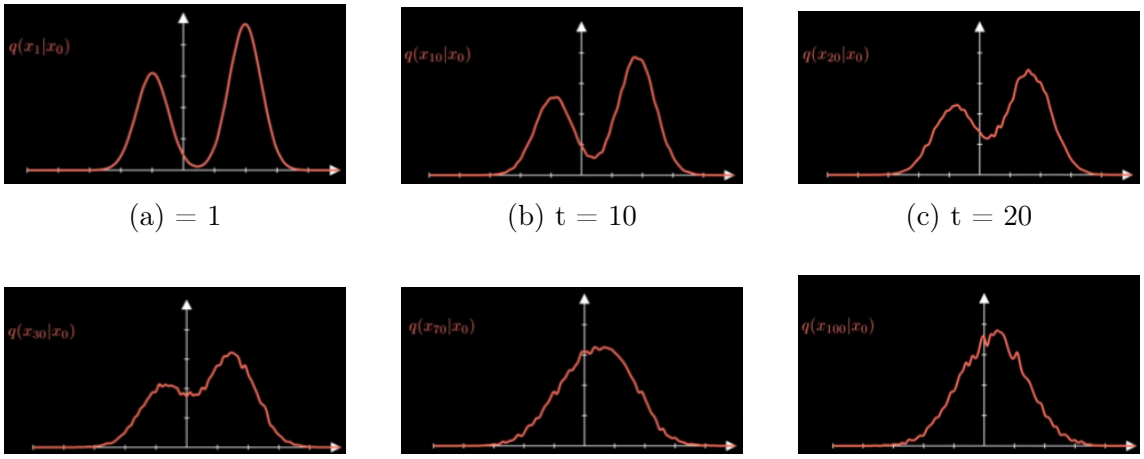


Figure 1: Distribution quelconque

En appliquant successivement la nouvelle fonction de transition, on obtient les rendus graphique suivant :



(a) = 1 (b) t = 10 (c) t = 20

(d) t = 30 (e) t = 70 (f) t = 100

Figure 2: Evolution de la distribution en fonction de t

On observe que pour t = 1, il n'y a que très peu de variation. Par contre, plus t augmente, plus la distribution se rapproche d'une distribution centrée-réduite.

1.4 Processus backward : calcul de la loss

L'objectif d'un modèle de diffusion est de maximiser la log-vraisemblance des données sous le modèle :

$$\log p_\theta(\mathbf{x}_0) \tag{12}$$

La *loss* à minimiser s'écrit alors :

$$\mathcal{L} = -\log p_\theta(\mathbf{x}_0) \tag{13}$$

1.4.1 De la log-vraisemblance à l'ELBO

On peut introduire $\mathbf{x}_{1:T}$, qui correspond au processus de diffusion :

$$\mathcal{L} = -\log \int p_\theta(\mathbf{x}_{0:T})\, d\mathbf{x}_{1:T} \tag{14}$$

$$= -\log \int q(\mathbf{x}_{1:T}|\mathbf{x}_0) \frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}\, d\mathbf{x}_{1:T} \tag{15}$$

$$= -\log \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \left[\frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \right] \tag{16}$$

Cette forme est exacte, mais elle est intractable car le calcul de l'espérance nécessite d'intégrer sur toutes les trajectoires possibles $\mathbf{x}_{1:T}$.

Pour contourner ce problème, on introduit la *Evidence Lower Bound* (ELBO) en utilisant l'inégalité de Jensen :

$$\mathcal{L} = -\log \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \left[\frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \right] \tag{17}$$

$$\leq \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \left[-\log \frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \right] \tag{18}$$

$$= \mathcal{L}_{\text{ELBO}} \tag{19}$$

1.4.2 Factorisation des distributions

On veut réécrire l'ELBO avec des distributions plus simples (pour savoir plus facilement ce que le modèle doit optimiser). Pour cela, on factorise les distributions des deux processus.

Distribution du processus backward : Par définition :

$$p_\theta(\mathbf{x}_{0:T}) = p_\theta(\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_T) \tag{20}$$

En utilisant la règle de Bayes on a :

$$p_\theta(\mathbf{x}_{0:T}) = p_\theta(\mathbf{x}_T)\, p_\theta(\mathbf{x}_{0:T-1}|\mathbf{x}_T) \tag{21}$$

$$= p_\theta(\mathbf{x}_T)\, p_\theta(\mathbf{x}_{T-1}|\mathbf{x}_T)\, p_\theta(\mathbf{x}_{0:T-2}|\mathbf{x}_{T-1}, \mathbf{x}_T) \tag{22}$$

Comme le processus est Markovien (chaque état $\mathbf{x}_{t-1}$ ne dépend que de $\mathbf{x}_t$) on a :

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_{t+1:T}) = p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) \tag{23}$$

Ainsi, en factorisant on obtient :

$$\boxed{p_\theta(\mathbf{x}_{0:T}) = p_\theta(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)} \tag{24}$$

Distribution du processus forward : De même, le processus forward $q(\mathbf{x}_{1:T}|\mathbf{x}_0)$ est une chaîne de Markov, donc :

$$\begin{aligned} q(\mathbf{x}_{1:T}|\mathbf{x}_0) &= q(\mathbf{x}_1|\mathbf{x}_0)\, q(\mathbf{x}_2, \ldots, \mathbf{x}_T|\mathbf{x}_1, \mathbf{x}_0) && (25)\\ &= q(\mathbf{x}_1|\mathbf{x}_0)\, q(\mathbf{x}_2|\mathbf{x}_1)\, q(\mathbf{x}_3, \ldots, \mathbf{x}_T|\mathbf{x}_2) && (26)\\ &\vdots && (27)\\ &= \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}) && (28) \end{aligned}$$

1.4.3 De l'ELBO à KL

On développe le logarithme :

$$\begin{aligned} \mathcal{L} &= \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}\Big[-\log \frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}\Big] && (29)\\ &= \mathbb{E}_q\Big[-\log p_\theta(\mathbf{x}_{0:T}) + \log q(\mathbf{x}_{1:T}|\mathbf{x}_0)\Big] && (30) \end{aligned}$$

On remplace par les factorisations des distributions :

$$\begin{aligned} \mathcal{L} &= \mathbb{E}_q\left[-\log p(\mathbf{x}_T) - \sum_{t=1}^{T} \log p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) + \sum_{t=1}^{T} \log q(\mathbf{x}_t|\mathbf{x}_{t-1})\right] && (31)\\ &= \mathbb{E}_q\left[-\log p(\mathbf{x}_T) - \sum_{t\geq 1} \log \frac{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_t|\mathbf{x}_{t-1})}\right] && (32) \end{aligned}$$

On isole explicitement le terme $t = 1$:

$$\mathcal{L} = \mathbb{E}_q\left[-\log p(\mathbf{x}_T) - \sum_{t>1} \log \frac{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_t|\mathbf{x}_{t-1})} - \log \frac{p_\theta(\mathbf{x}_0|\mathbf{x}_1)}{q(\mathbf{x}_1|\mathbf{x}_0)}\right] \tag{33}$$

Pour les termes $t > 1$, on utilise la formule de Bayes pour réécrire $q(\mathbf{x}_t|\mathbf{x}_{t-1})$ en

termes de $q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$:

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \frac{q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)\, q(\mathbf{x}_t|\mathbf{x}_0)}{q(\mathbf{x}_{t-1}|\mathbf{x}_0)} \tag{34}$$

En remplaçant dans l'ELBO, on a :

$$\mathcal{L} = \mathbb{E}_q \left[-\log p(\mathbf{x}_T) - \sum_{t>1} \log \frac{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}{\frac{q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)\, q(\mathbf{x}_t|\mathbf{x}_0)}{q(\mathbf{x}_{t-1}|\mathbf{x}_0)}} - \log \frac{p_\theta(\mathbf{x}_0|\mathbf{x}_1)}{q(\mathbf{x}_1|\mathbf{x}_0)} \right] \tag{35}$$

Simplification de la somme on réécrit le dénominateur :

$$\log \frac{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}{\frac{q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)\, q(\mathbf{x}_t|\mathbf{x}_0)}{q(\mathbf{x}_{t-1}|\mathbf{x}_0)}} = \log \left(\frac{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)} \cdot \frac{q(\mathbf{x}_{t-1}|\mathbf{x}_0)}{q(\mathbf{x}_t|\mathbf{x}_0)} \right) \tag{36}$$

$$= \log \frac{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)} + \log \frac{q(\mathbf{x}_{t-1}|\mathbf{x}_0)}{q(\mathbf{x}_t|\mathbf{x}_0)} \tag{37}$$

En reportant dans la somme :

$$\sum_{t>1} \log \frac{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)} + \sum_{t>1} \log \frac{q(\mathbf{x}_{t-1}|\mathbf{x}_0)}{q(\mathbf{x}_t|\mathbf{x}_0)} \tag{38}$$

On développe explicitement la seconde somme :

$$\sum_{t>1} \log \frac{q(\mathbf{x}_{t-1}|\mathbf{x}_0)}{q(\mathbf{x}_t|\mathbf{x}_0)} = \sum_{t>1} \Big[\log q(\mathbf{x}_{t-1}|\mathbf{x}_0) - \log q(\mathbf{x}_t|\mathbf{x}_0) \Big] \tag{39}$$

$$= \big(\log q(\mathbf{x}_1|\mathbf{x}_0) - \log q(\mathbf{x}_2|\mathbf{x}_0) \big) \tag{40}$$

$$+ \big(\log q(\mathbf{x}_2|\mathbf{x}_0) - \log q(\mathbf{x}_3|\mathbf{x}_0) \big) \tag{41}$$

$$+ \ldots \tag{42}$$

$$+ \big(\log q(\mathbf{x}_{T-1}|\mathbf{x}_0) - \log q(\mathbf{x}_T|\mathbf{x}_0) \big) \tag{43}$$

Tous les termes intermédiaires s'annulent, il reste :

$$\log q(\mathbf{x}_1|\mathbf{x}_0) - \log q(\mathbf{x}_T|\mathbf{x}_0) \tag{44}$$

On réinjecte ce résultat et on simplifie :

$$\mathcal{L} = \mathbb{E}_q\Bigg[-\log p(\mathbf{x}_T) - \sum_{t>1} \log \frac{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)} - \log q(\mathbf{x}_1|\mathbf{x}_0) + \log q(\mathbf{x}_T|\mathbf{x}_0) - \log \frac{p_\theta(\mathbf{x}_0|\mathbf{x}_1)}{q(\mathbf{x}_1|\mathbf{x}_0)} \Bigg] \tag{45}$$

$$= \mathbb{E}_q\Bigg[-\log p(\mathbf{x}_T) + \log q(\mathbf{x}_T|\mathbf{x}_0) - \sum_{t>1} \log \frac{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)} - \log p_\theta(\mathbf{x}_0|\mathbf{x}_1) \Bigg] \tag{46}$$

$$= \mathbb{E}_q\Bigg[-\log \frac{p(\mathbf{x}_T)}{q(\mathbf{x}_T|\mathbf{x}_0)} - \sum_{t>1} \log \frac{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)} - \log p_\theta(\mathbf{x}_0|\mathbf{x}_1) \Bigg] \tag{47}$$

Réecriture avec KL La divergence de Kullback–Leibler est une forme de distance entre deux distributions $q(z)$ et $p(z)$. Elle est définit par :

$$\mathrm{KL}(q \,\|\, p) \triangleq \mathbb{E}_{q(z)}\left[\log \frac{q(z)}{p(z)}\right] \tag{48}$$

On simplifie les négations et on applique la définition de KL :

$$\mathcal{L} = \mathbb{E}_q\Bigg[-\log \frac{p(\mathbf{x}_T)}{q(\mathbf{x}_T|\mathbf{x}_0)} - \sum_{t>1} \log \frac{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)} - \log p_\theta(\mathbf{x}_0|\mathbf{x}_1) \Bigg] \tag{49}$$

$$= \mathbb{E}_q\Bigg[\log \frac{q(\mathbf{x}_T|\mathbf{x}_0)}{p(\mathbf{x}_T)} + \sum_{t>1} \log \frac{q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)}{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)} - \log p_\theta(\mathbf{x}_0|\mathbf{x}_1) \Bigg] \tag{50}$$

$$= \mathbb{E}_q\Bigg[\mathrm{KL}\big(q(\mathbf{x}_T|\mathbf{x}_0) \,\|\, p(\mathbf{x}_T)\big) + \sum_{t>1} \mathrm{KL}\big(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) \,\|\, p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)\big) - \log p_\theta(\mathbf{x}_0|\mathbf{x}_1) \Bigg] \tag{51}$$

1.4.4 Simplification du KL

On peut simplifier cette expression en analysant chaque terme.

Le premier terme

$$\mathrm{KL}\big(q(\mathbf{x}_T|\mathbf{x}_0) \,\|\, p(\mathbf{x}_T)\big)$$

est constant et ne dépend pas des paramètres θ du modèle. Il n'influence pas l'apprentissage et peut donc être supprimé de la fonction de coût.

Le dernier terme

$$-\log p_\theta(\mathbf{x}_0|\mathbf{x}_1)$$

correspond à une perte de reconstruction entre $\mathbf{x}_0$ et sa prédiction à partir de $\mathbf{x}_1$. Ce

terme ne concerne que l'étape $t = 1$ et a un impact faible par rapport à la somme des termes KL pour $t > 1$ (les processus de diffusion peuvent avoir $t > 1000$). Comme il est négligeable, on peut le supprimer.

La fonction de coût devient :

$$\mathcal{L}_{\text{DDPM}} = \mathbb{E}_q\Big[\sum_{t>1} \text{KL}\big(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) \,\|\, p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)\big)\Big] \tag{52}$$

1.4.5 Expression des distributions

Pour calculer les termes de divergence de Kullback-Leibler, il est nécessaire d'exprimer explicitement les distributions de manière explicite.

Distribution $q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$. En appliquant la formule de Bayes, on obtient :

$$q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) = \frac{q(\mathbf{x}_t|\mathbf{x}_{t-1})\, q(\mathbf{x}_{t-1}|\mathbf{x}_0)}{q(\mathbf{x}_t|\mathbf{x}_0)} \tag{53}$$

Le terme au dénominateur $q(\mathbf{x}_t|\mathbf{x}_0)$ ne dépend pas de $\mathbf{x}_{t-1}$. Il joue uniquement le rôle d'une constante de normalisation assurant que $q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$ intègre à 1. Il est donc suffisant d'étudier le numérateur pour déterminer la forme de la distribution, quitte à normaliser ensuite.

Par définition du processus de diffusion, les deux termes du numérateur sont des lois gaussiennes :

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}\big(\sqrt{\alpha_t}\mathbf{x}_{t-1}, (1-\alpha_t)\mathbf{I}\big) \tag{54}$$

$$q(\mathbf{x}_{t-1}|\mathbf{x}_0) = \mathcal{N}\big(\sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_0, (1-\bar{\alpha}_{t-1})\mathbf{I}\big) \tag{55}$$

Le numérateur est donc le produit de deux lois normales en la variable $\mathbf{x}_{t-1}$. On rappelle que le produit de deux lois normales est proportionnel à une loi normale. Formellement, on a :

$$\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)\, \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2) \propto \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$$

avec

$$\boldsymbol{\Sigma}^{-1} = \boldsymbol{\Sigma}_1^{-1} + \boldsymbol{\Sigma}_2^{-1}, \tag{56}$$

$$\boldsymbol{\mu} = \boldsymbol{\Sigma}\big(\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\mu}_1 + \boldsymbol{\Sigma}_2^{-1}\boldsymbol{\mu}_2\big) \tag{57}$$

Après application de cette propriété, on obtient :

$$q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) = \mathcal{N}\big(\boldsymbol{\mu}_q(\mathbf{x}_t, \mathbf{x}_0), \tilde{\beta}_t \mathbf{I}\big),$$

avec

$$\boldsymbol{\mu}_q(\mathbf{x}_t, \mathbf{x}_0) = \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\,\mathbf{x}_t + \frac{\sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)}{1-\bar{\alpha}_t}\,\mathbf{x}_0, \tag{58}$$

$$\tilde{\beta}_t = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\,(1-\alpha_t). \tag{59}$$

Distribution $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$. Le processus inverse est paramétré par un réseau de neurones. On suppose que chaque transition inverse est modélisée par une loi gaussienne :

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}\big(\boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(t)\big).$$

Pour simplifier les calculs, la variance de cette loi n'est pas apprise. Elle est fixée à la variance du processus avant :

$$\boldsymbol{\Sigma}_\theta(t) = \tilde{\beta}_t \mathbf{I},$$

La seule quantité prédite par le réseau de neurones est la moyenne $\boldsymbol{\mu}_\theta(\mathbf{x}_t, t)$. Ce choix permet de simplifier l'expression de la KL, qui devient un KL entre deux lois normales de même variance. En théorie, la divergence de Kullback-Leibler entre deux lois normales peut toujours être calculée analytiquement, même lorsque les variances sont différentes.

Cependant, si la variance $\boldsymbol{\Sigma}_\theta(t)$ est apprise, l'expression du KL contient alors des termes supplémentaires (un terme de trace, un terme de déterminant et un terme quadratique) et ces termes compliquent l'optimisation.

En remplaçant les distributions par leurs formes gaussiennes, on a :

$$\mathcal{L}_{\text{DDPM}} = \mathbb{E}_q\left[\sum_{t>1} \mathrm{KL}\Big(\mathcal{N}\big(\boldsymbol{\mu}_q(\mathbf{x}_t, \mathbf{x}_0), \tilde{\beta}_t \mathbf{I}\big) \,\Big\|\, \mathcal{N}\big(\boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \tilde{\beta}_t \mathbf{I}\big)\Big)\right]$$

la divergence de Kullback-Leibler entre ces deux lois de même variance se simplifie en une norme quadratique sur les moyennes. Ainsi on obtient :

$$\mathcal{L}_{\text{DDPM}} = \mathbb{E}_q\Big[\sum_{t>1} \frac{1}{2\tilde{\beta}_t}\big\|\boldsymbol{\mu}_q(\mathbf{x}_t, \mathbf{x}_0) - \boldsymbol{\mu}_\theta(\mathbf{x}_t, t)\big\|^2\Big]$$

1.4.6 Réécriture des moyennes

Pour simplifier la fonction de coût, on commence par réécrire les moyennes en fonction du bruit. Rappelons que :

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\,\boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}).$$

On peut isoler $\mathbf{x}_0$ en fonction de $\mathbf{x}_t$ et $\boldsymbol{\epsilon}$:

$$\mathbf{x}_0 = \frac{1}{\sqrt{\bar{\alpha}_t}}\Big(\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t}\,\boldsymbol{\epsilon}\Big).$$

Réécriture de $\boldsymbol{\mu}_q(\mathbf{x}_t, \mathbf{x}_0)$ La moyenne analytique du processus inverse s'écrit :

$$\boldsymbol{\mu}_q(\mathbf{x}_t, \mathbf{x}_0) = \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathbf{x}_t + \frac{\sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)}{1-\bar{\alpha}_t}\mathbf{x}_0.$$

En remplaçant $\mathbf{x}_0$ par son expression en fonction de $\mathbf{x}_t$ et $\boldsymbol{\epsilon}$, on obtient :

$$\boldsymbol{\mu}_q(\mathbf{x}_t, \mathbf{x}_0) = \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathbf{x}_t + \frac{\sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)}{1-\bar{\alpha}_t}\frac{1}{\sqrt{\bar{\alpha}_t}}\Big(\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t}\,\boldsymbol{\epsilon}\Big),$$

En factorisant on obtient :

$$\boldsymbol{\mu}_q(\mathbf{x}_t, \mathbf{x}_0) = \mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\,\boldsymbol{\epsilon}.$$

Réécriture de $\boldsymbol{\mu}_\theta(\mathbf{x}_t, t)$ Similairement, la moyenne prédite par le réseau s'écrit :

$$\boldsymbol{\mu}_\theta(\mathbf{x}_t, t) = \mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\,\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t).$$

1.4.7 Simplification de la loss

En substituant ces expressions dans la loss, on obtient :

$$\mathcal{L}_{\text{DDPM}} = \mathbb{E}_{\mathbf{x}_0,\boldsymbol{\epsilon}}\Big[\sum_{t>1}\frac{1}{2\tilde{\beta}_t}\Big\|\Big(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\,\boldsymbol{\epsilon}\Big) - \Big(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\,\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\Big)\Big\|^2\Big],$$

Les termes $\mathbf{x}_t$ s'annulent. On peut alors factoriser ce qui donne :

$$\mathcal{L}_{\text{DDPM}} = \mathbb{E}_{\mathbf{x}_0,\boldsymbol{\epsilon}}\Big[\sum_{t>1}\frac{1}{2\tilde{\beta}_t}\left(\frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\right)^2\big\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\big\|^2\Big].$$

Le terme $\frac{1}{2\tilde{\beta}_t}\left(\frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\right)^2$ ne dépend ni des données ni des paramètres du réseau. Il peut donc être ignoré pour l'apprentissage. La loss se simplifie alors en :

$$\mathcal{L}_{\text{DDPM}} = \mathbb{E}_{\mathbf{x}_0,\boldsymbol{\epsilon}}\Big[\sum_{t>1}\big\|\boldsymbol{\epsilon}-\boldsymbol{\epsilon}_\theta(\mathbf{x}_t,t)\big\|^2\Big].$$

Cependant, cette formulation est une somme sur toutes les étapes de diffusion, ce qui devient coûteux lorsque le nombre de pas T est grand (typiquement plusieurs centaines ou milliers). Dans l'article, ils remplacent cette somme par une espérance sur t, en échantillonnant uniformément une étape de diffusion à chaque itération d'entraînement.

La fonction de coût finale utilisée pour l'apprentissage s'écrit alors :

$$\mathcal{L}_{\text{DDPM}} = \mathbb{E}_{\mathbf{x}_0,\boldsymbol{\epsilon},t}\Big[\big\|\boldsymbol{\epsilon}-\boldsymbol{\epsilon}_\theta(\mathbf{x}_t,t)\big\|^2\Big],$$

ce qui correspond à une simple erreur quadratique moyenne entre le bruit réel ajouté au processus direct et le bruit prédit par le réseau.

2 Implémentation et Analyse Expérimentale

Cette section détaille la mise en œuvre pratique du modèle, justifie nos choix architecturaux et propose une analyse approfondie des résultats empiriques.

2.1 Architecture : Le U-Net Temporel

L'élément central du DDPM est le réseau neuronal $\epsilon_\theta(x_t, t)$. Nous avons implémenté une architecture **U-Net** spécifique, distincte des modèles de segmentation classiques:

- **Injection du Temps :** Le pas de temps t est critique car il définit le niveau de bruit. Nous l'avons projeté via des *Sinusoidal Positional Embeddings* (dimension 32) injectés dans chaque bloc résiduel. Cela confère au réseau une "conscience" du niveau de dégradation : à $t = 1000$, le réseau cherche des structures globales (basses fréquences) ; à $t = 10$, il se concentre sur le nettoyage de texture (hautes fréquences).
- **Dimensionnement :** Avec environ 1,7 million de paramètres, notre modèle est relativement léger. L'utilisation de *GroupNorm* (au lieu de BatchNorm) s'est avérée cruciale pour stabiliser l'apprentissage avec des tailles de batch réduites.

2.2 Analyse de la Dynamique d'Apprentissage

L'entraînement (60 000 images, 100 époques, AdamW) révèle une dynamique intéressante, visible sur la Figure 3.

Nous identifions deux régimes d'apprentissage distincts :

1. **Régime Sémantique (Époques 0-10) :** La perte s'effondre exponentiellement. Le réseau apprend ici les caractéristiques macroscopiques : centrer le chiffre, avoir un fond noir, et tracer des traits blancs grossiers.

2. **Régime de Raffinement (Époques 10-100) :** La perte ne diminue que très lentement et oscille autour de 0.025. Cette asymptote n'est pas un échec de convergence, mais une limite théorique : le processus étant stochastique, il existe une part de bruit aléatoire que le modèle ne peut (et ne doit) pas prédire parfaitement.

2.3 Étude du Cycle de Diffusion

Pour vérifier que le modèle n'a pas simplement appris à inverser une opération mathématique, mais bien à générer, nous analysons le cycle destruction-reconstruction (Figures 4 et 5).

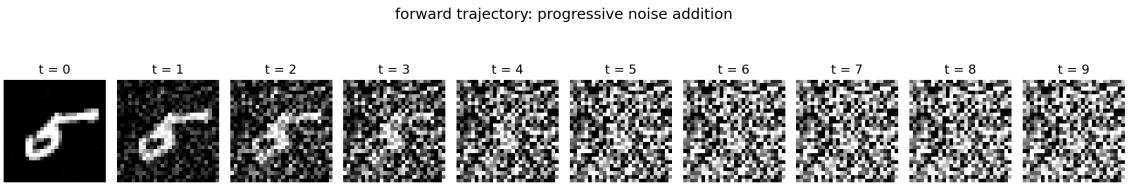


Figure 4: **Forward Process (q)** : Injection de bruit gaussien. À $t = 500$ (milieu), l'information sémantique devient ambiguë. À $t = 1000$, le signal x_0 est statistiquement perdu.

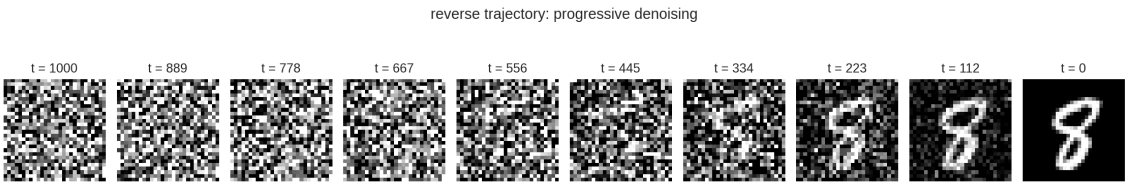
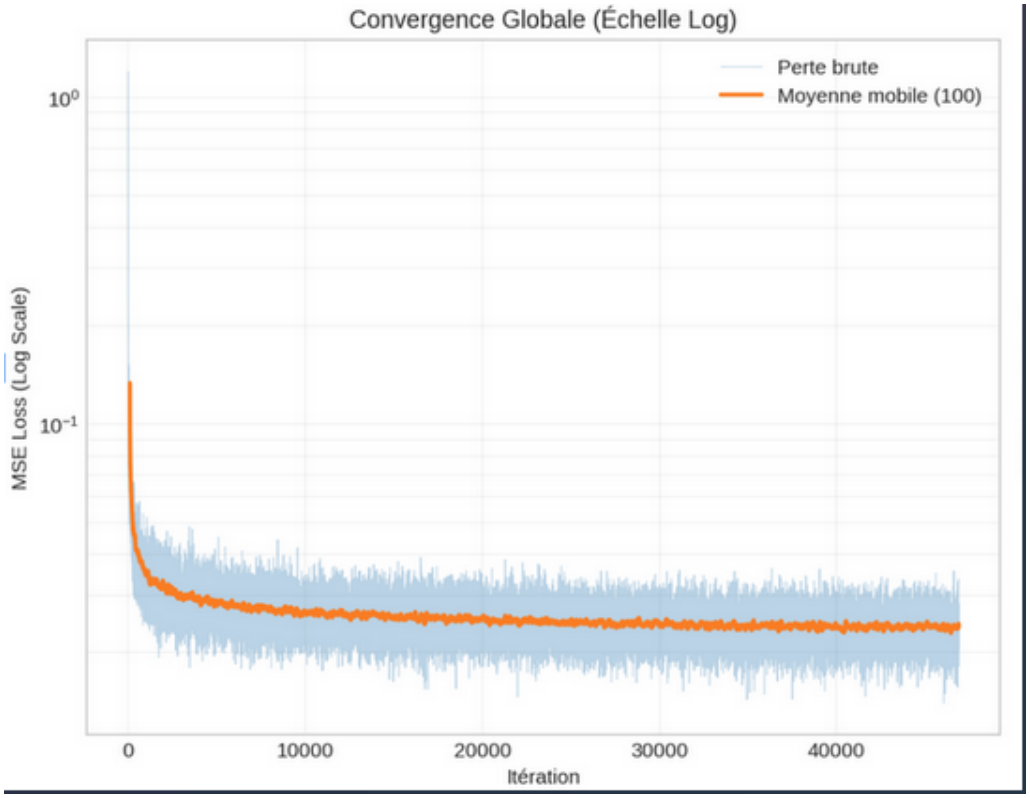


Figure 5: **Reverse Process (p_θ)** : Le modèle part d'un bruit pur $x_T \sim \mathcal{N}(0, I)$ et génère un "8".

L'observation de la trajectoire inverse (Figure 5) valide l'hypothèse de la génération hiérarchique :



(a) **Vue Logarithmique** : La chute brutale initiale correspond à l'apprentissage de la moyenne du dataset.

(b) **Vue Linéaire (Zoom)** : Stabilisation autour de la variance irréductible.

Figure 3: Analyse de la fonction de perte (MSE) au cours de l'entraînement.

- **Phase Chaotique** ($t \in [1000, 600]$) : L'image reste bruyante. Le modèle effectue des corrections minimes, cherchant à orienter le bruit vers la variété des données (le "manifold" des chiffres).
- **Phase de Structuration** ($t \in [600, 200]$) : C'est la phase critique où la classe émerge. On voit distinctement la forme du "8" apparaître.
- **Phase de Nettoyage** ($t < 200$) : La structure est fixée, le modèle retire simplement le grain résiduel pour augmenter le contraste.

Le fait de générer un "8" à partir d'un processus initié (conceptuellement) sur un "5" prouve l'indépendance du générateur : il a appris la distribution $p_{data}(x)$ et non des exemples par cœur.

2.4 Validation Quantitative et Diversité

Au-delà de l'inspection visuelle (Figure 6), nous devons quantifier la fidélité et la diversité.

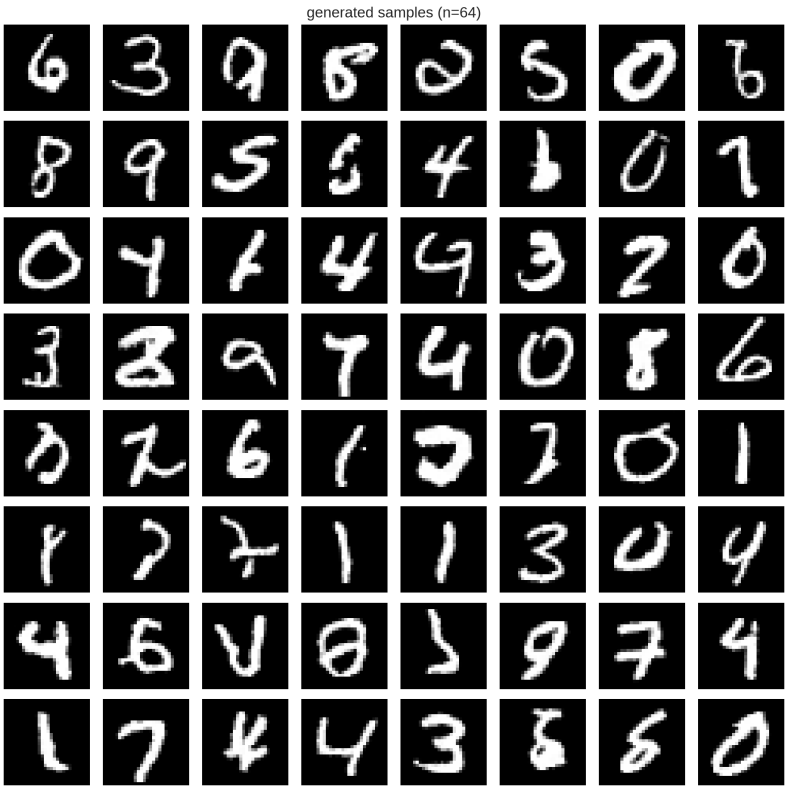


Figure 6: Grille de 64 échantillons. On note une grande variabilité dans l'épaisseur des traits et l'inclinaison.

Les métriques calculées sur 1000 échantillons sont rapportées ci-dessous :

Métrique	Valeur	Interprétation
FID Score	**22.58**	Distance distributionnelle faible (Haute qualité)
Confiance Classifieur	**93.05%**	Features discriminantes nettes (Pas de flou)

Le **FID (Fréchet Inception Distance)** de 22.58 est particulièrement encourageant pour un modèle entraîné sur une architecture modeste, indiquant que nos échantillons couvrent bien les modes de la distribution réelle. La confiance élevée du classifieur confirme que les chiffres ne sont pas des "chimères" (mélanges flous de deux chiffres), mais des entités bien définies.

Enfin, nous adressons le risque de *Mode Collapse*.

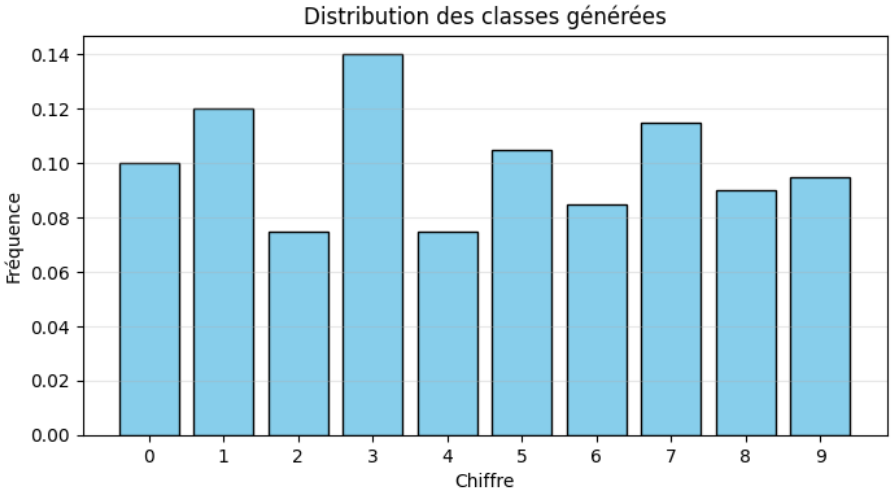


Figure 7: Histogramme des classes générées. L'uniformité relative démontre une bonne entropie de génération.

Comme l'illustre la Figure 7, le modèle génère toutes les classes de manière équilibrée (autour de 10% par classe). Contrairement à un GAN qui pourrait s'effondrer sur la génération de "1" (plus faciles à tracer), le DDPM préserve la diversité du dataset original grâce à sa formulation probabiliste qui couvre toute la distribution.

3 Conclusion et Perspectives

Ce projet a permis de valider l'efficacité des modèles de diffusion probabilistes (DDPM) pour la génération d'images, en confirmant les intuitions théoriques par une implémentation fonctionnelle sur MNIST.

Synthèse des résultats

Nous avons démontré que l'approche théorique complexe (basée sur l'inférence variationnelle) se traduit en pratique par un objectif d'entraînement simple et stable. Les analyses menées dans la section précédente confirment la robustesse de l'approche :

- **Qualité :** Un score FID de 22.58 et une confiance de classifieur élevée valident le réalisme des échantillons.
- **Diversité :** L'absence de *Mode Collapse* prouve que le modèle a capturé l'intégralité de la distribution des données.

Limites et Ouvertures

La principale limitation identifiée lors de nos expériences reste le coût computationnel de l'inférence. Le processus itératif nécessitant $T = 1000$ passages dans le réseau pour générer une seule image est significativement plus lent qu'un GAN ou un VAE (qui génèrent en une passe).

Pour aller plus loin, deux pistes d'amélioration sont prioritaires :

1. L'implémentation de **DDIM** (Denoising Diffusion Implicit Models) pour accélérer l'échantillonnage en sautant des étapes (ex: passer de 1000 à 50 pas).
2. Le passage aux **Latent Diffusion Models**, pour effectuer la diffusion dans un espace latent compressé, réduisant ainsi la dimensionnalité du problème.

References

[1] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, 2020.