UNIVERSITE D'ÉVRY PARIS SACLAY

M1 COMPUTER & NETWORK SYSTEMS
SYSTEMES AUTONOMES

RAPPORT DEEP LEARNING

DISENTANGLING DISENTANGLEMENT IN VARIATIONAL AUTOENCODERS

Présenté par

Mohammed Yacine BRAHMIA brahmia.my@gmail.com
Nabil DJELLOUDI nabildjelloudi772@gmail.com

Encadré par

Mr. Massinissa HAMIDI

Année Universitaire 2024-2025

Contents

1 INTRODUCTION

Les modèles génératifs profonds, et en particulier les autoencodeurs variationnels (VAEs), ont suscité un intérêt croissant dans la communauté scientifique en raison de leur capacité à apprendre des représentations latentes compactes et exploitables des données. Parmi les nombreuses applications des VAEs, la recherche sur le désentrelacement (disentanglement) des représentations latentes occupe une place centrale. Cette notion, souvent associée à l'interprétabilité des modèles, vise à découvrir des facteurs latents indépendants qui capturent les variations sous-jacentes des données.

Dans ce rapport, nous allons nous concentrer sur l'article "Disentangling Disentanglement in Variational Autoencoders", rédigé par Emile Mathieu et al., qui propose une généralisation du désentrelacement, appelée "décomposition".

ORGANISATION DU RAPPORT

Nous commencerons par présenter un bref contexte général sur ce qu'est un Autoencodeur, un Autoencodeur variationnel, un espace latent et quelques notions importantes à la compréhension de l'article et de notre travail.

Par la suite, nous discuterons les méthodes différentes du papier, sa contribution et notre reproduction des expériences différentes menées par les auteurs.

Enfin, nous présenterons notre adaptation et application de ces expériences dans un environnement structuré, représenté par le dataset Rotated-MNIST ainsi que les résultats obtenus et les perspectives pour le futur.

En annexe, vous trouverez un résumé des différentes difficultés techniques rencontrées lors de l'adaptation du code fourni par l'article, les solutions apportées ainsi que les ajouts spécifiques que nous avons implémentés.

2 CONTEXTE GÉNÉRAL

2.1 AUTO-ENCODEUR ET AUTO-ENCODEUR VARIATIONNEL

AUTO-ENCODEURS

Un autoencodeur est un réseau de neurones utilisé en apprentissage non supervisé pour apprendre une représentation compacte (ou encodage) des données. Il se compose de trois parties principales : l'encodeur, le décodeur et l'espace latent entre les deux.

L'ENCODEUR

Réduit les données d'entrée X à une représentation de dimension inférieure z, appelée espace latent, par exemple, pour une image de 28x28 pixels, l'encodeur pourrait réduire les 784 dimensions d'entrée à un vecteur latent de 50 ou même de 10 dimensions.

$$z = f_{\text{encodeur}}(X)$$

L'ESPACE LATENT

L'espace latent représenté par z dans la figure correspond à une représentation compressée de l'entrée x obtenue grâce à l'encodeur g_ϕ. Cet espace latent est de faible dimension par rapport à l'espace original des données d'entrée. L'objectif est de capturer les caractéristiques essentielles de l'entrée x pour permettre au décodeur f_θ de reconstruire une sortie x' aussi proche que possible de l'entrée originale, soit $x \approx x'$.

LE DÉCODEUR

Reconstruit les données d'entrée à partir de la représentation latente z

$$\hat{X} = f_{\text{décodeur}}(z)$$

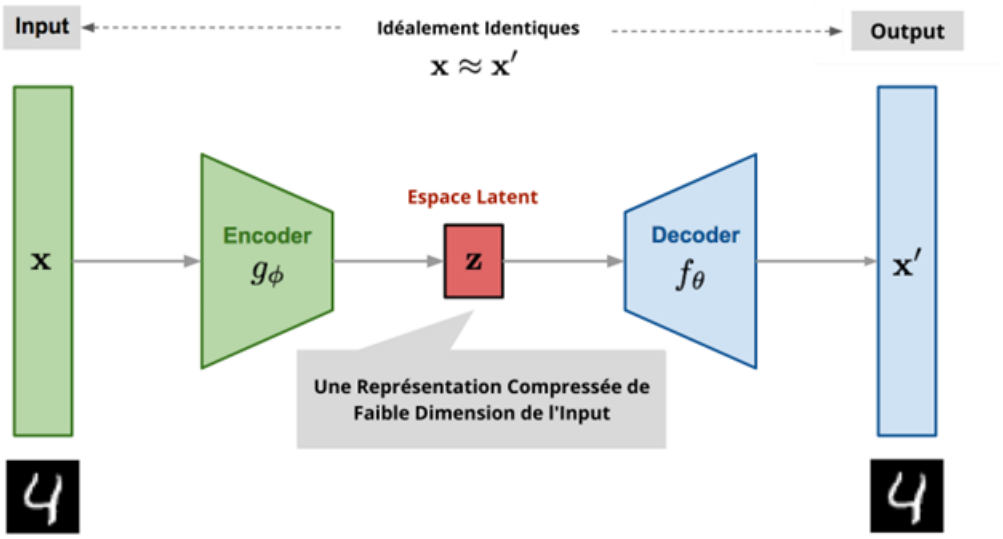

Figure 1: Architecture d'un Auto-encodeur

Hélas, avec les autoencodeurs traditionne, l'espace latent peut être discontinu ou désorganisé, et il est difficile d'échantillonner z aléatoirement pour générer de nouvelles données cohérentes. C'est ici qu'interviennent les Variational Autoencoder (VAE).

AUTO-ENCODEURS VARIATIONNELS

Les autoencodeurs variationnels (VAE) sont une extension probabiliste des autoencodeurs classiques, conçus pour apprendre des représentations compactes et structurées tout en permettant de générer de nouvelles données réalistes. Contrairement aux autoencodeurs classiques qui compriment les données dans un espace latent sans contrainte particulière, les VAE imposent une structure probabiliste à cet espace. Cela est réalisé en modélisant chaque donnée d'entrée comme une distribution gaussienne dans l'espace latent, définie par un vecteur de moyennes (μ) et un vecteur d'écarts-types (σ). Cette approche garantit que l'espace latent est continu, organisé et qu'il suit une distribution normale standard $\mathcal{N}(0, 1)$, facilitant ainsi l'échantillonnage pour la génération de nouvelles données.
Le processus d'apprentissage des VAE repose sur une combinaison d'inférences bayésiennes et d'apprentissage profond. L'encodeur transforme les données d'entrée en paramètres de la distribution latente (μ et σ), tandis que le décodeur génère les données reconstruites à partir d'échantillons tirés de cette distribution. Pour permettre un échantillonnage différentiable, les VAE utilisent une technique appelée **reparamétrisation**, où un bruit gaussien $\epsilon \sim \mathcal{N}(0, 1)$ est ajouté à la représentation latente selon :

$$z = \mu + \sigma \cdot \epsilon$$

Cela permet au modèle d'être entraîné de bout en bout avec la rétropropagation.

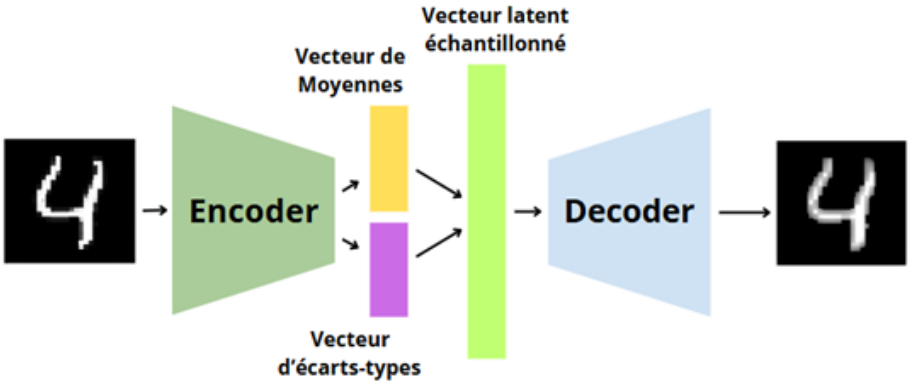

Figure 2: Architecture d'un AutoEncodeur variationnel

La fonction de perte des VAE combine deux objectifs : la **perte de reconstruction**, qui mesure à quel point les données reconstruites ressemblent aux données d'entrée, et une **perte de régularisation**, qui force la distribution latente $q(z|X)$ à être proche d'une distribution normale standard $p(z)$.
Ensemble, ces deux termes forment ce que l'on appelle l'**Evidence Lower Bound (ELBO)**, qui représente une borne inférieure de la vraisemblance des données.
Maximiser l'ELBO revient à optimiser à la fois la fidélité de la reconstruction et l'organisation de l'espace latent.
En pratique, l'ELBO est exprimé comme une somme :

$$\text{ELBO} = \mathbb{E}_{q(z|X)}[\log p(X|z)] - \text{KL}(q(z|X)\|p(z)),$$

où :

- $\mathbb{E}_{q(z|x)}[\log p(x|z)]$ est le **terme de reconstruction**, qui évalue la qualité de la reconstruction des données.
- $\text{KL}(q(z|x)\|p(z))$ est le **terme de régularisation**, qui mesure la divergence de Kullback-Leibler entre la distribution latente approximative $q(z|x)$ et la distribution prior $p(z)$.

En minimisant la perte opposée à l'ELBO, les VAE apprennent un **espace latent structuré** tout en générant des reconstructions fidèles.

2.2 Les β (BETA) VAE

Un β-VAE est une extension des autoencodeurs variationnels (VAE) qui introduit un facteur d'échelle β dans la fonction de perte pour contrôler le degré de régularisation de l'espace latent.
Les β-VAE introduisent un hyperparamètre $\beta > 1$ pour renforcer cette régularisation, augmentant ainsi la pression pour désentrelacer les dimensions latentes. Cela permet d'apprendre des représentations où chaque variable latente correspond à un facteur de variation unique (par exemple, l'orientation, la taille ou la couleur d'un objet).
Cependant, un β élevé peut détériorer la qualité des reconstructions, car le modèle privilégie la structuration de l'espace latent au détriment de la fidélité des données reconstruites. Ce mécanisme est particulièrement utile pour découvrir des facteurs sous-jacents dans des données complexes et pour les applications nécessitant un contrôle précis des générateurs.
La fonction de perte d'un β-VAE s'écrit comme suit :

$$L_\beta(x) = \mathbb{E}_{q_\phi(z|x)}\left[\log p_\theta(x|z)\right] - \beta\,\text{KL}\left(q_\phi(z|x)\|p(z)\right),$$

où :

- β est un hyperparamètre qui contrôle l'importance de la régularisation.

2.3 Le Disentangling (désentrelacement) dans les β VAE

Si chaque variable dans la représentation latente inférée est sensible à un seul facteur générateur et relativement invariant aux autres facteurs, nous dirons que cette représentation est démêlée, entrelacée ou factorisée. L'un des avantages qui vient souvent avec une représentation démêlée est une bonne interprétabilité et une généralisation facile à une variété de tâches.
Le *disentangling* dans les VAE, et particulièrement dans les β-VAE, consiste à apprendre un espace latent où chaque dimension capture un **facteur de variation distinct et interprétable** des données, de manière indépendante.
Dans un VAE classique, la régularisation via la divergence KL force la distribution latente $q(z|X)$ à se rapprocher d'une distribution normale standard, mais elle ne garantit pas l'indépendance entre les dimensions latentes.

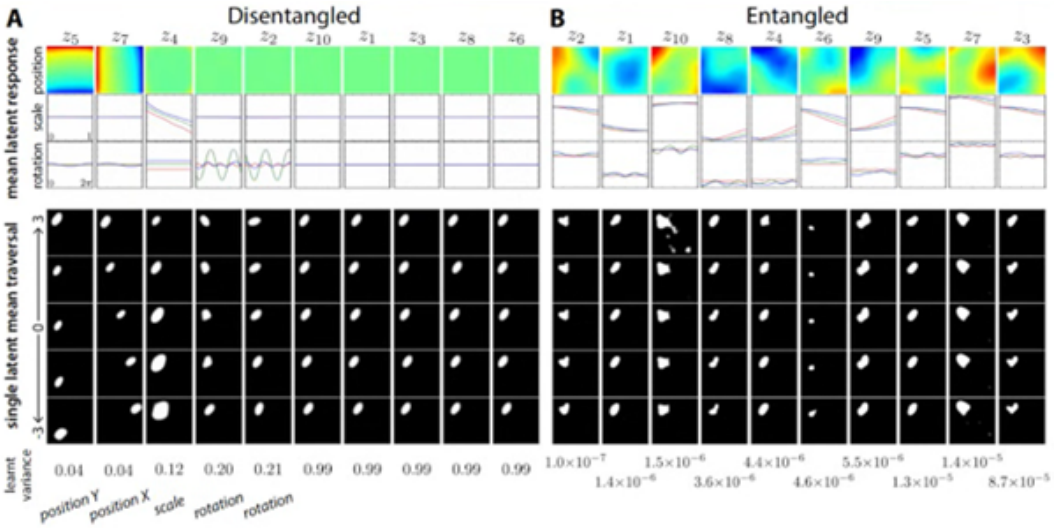

Figure 3: Comparaison des représentations latentes: : Désentrelacées vs Entrelacées (1)

En termes simples, imaginons que notre encodeur, basé sur un CNN, ait comme couche finale (bottleneck) un total de 10 nœuds, ce qui signifie que nous avons 10 dimensions ou 10 variables dans notre espace latent. Dans les VAEs classiques, sans utiliser de β, il est difficile de déterminer quel nœud est responsable de l'apprentissage d'une caractéristique spécifique (comme l'orientation, la taille ou la couleur).
Cependant, comme le montre la figure 3, en augmentant la valeur de β (en bas à gauche), nous pouvons forcer l'encodeur à n'utiliser que les variables essentielles (par exemple, 5 au lieu de 10). En variant les valeurs des variables dans l'espace latent, nous pouvons alors identifier quelles caractéristiques sont apprises par chaque nœud.
Dans cet exemple, les 5 nœuds actifs apprennent respectivement :

- la position en Y,
- la position en X,
- l'échelle (*scale*),
- et deux dimensions pour la rotation.

Les 5 nœuds restants, en revanche, ne contribuent pas de manière significative à la compression et sont peu représentatifs des données.

3 LE PAPIER

Dans cette section nous parlerons de ce que propose l'article en termes de contributions et d'expériences et nous détaillerons comment nous avons parvenues a reproduire ces expériences.

Afin de parvenir à la reproduction des résultats de l'article nous avons adopter l'environnement MyDocker IA GPU - Torch - python3.12 qui fait partie des ressources GPU fournies par Paris-Saclay gérées par la DSI CentraleSupelec, en utilisant l'outil JupyterLab.

Nous nous sommes basés sur le code de l'article fournie dans leurs Github, en apportant multiples modifications détaillés en annexe.

3.1 COMPRÉHENSION & MÉTHODES

Les auteurs abordent d'abord les limites des approches existantes pour le **désentrelacement** (*disentanglement*) dans les Variational Auto-encoders (VAEs) et, en particulier, dans les β-VAEs. Ils expliquent que le désentrelacement vise à apprendre un **espace latent** où chaque dimension capture un facteur de variation unique des données, ce qui permet d'obtenir des représentations latentes interprétables.

GÉNÉRALISATION DU DÉSENTRELACEMENT AVEC "LA DÉCOMPOSITION"

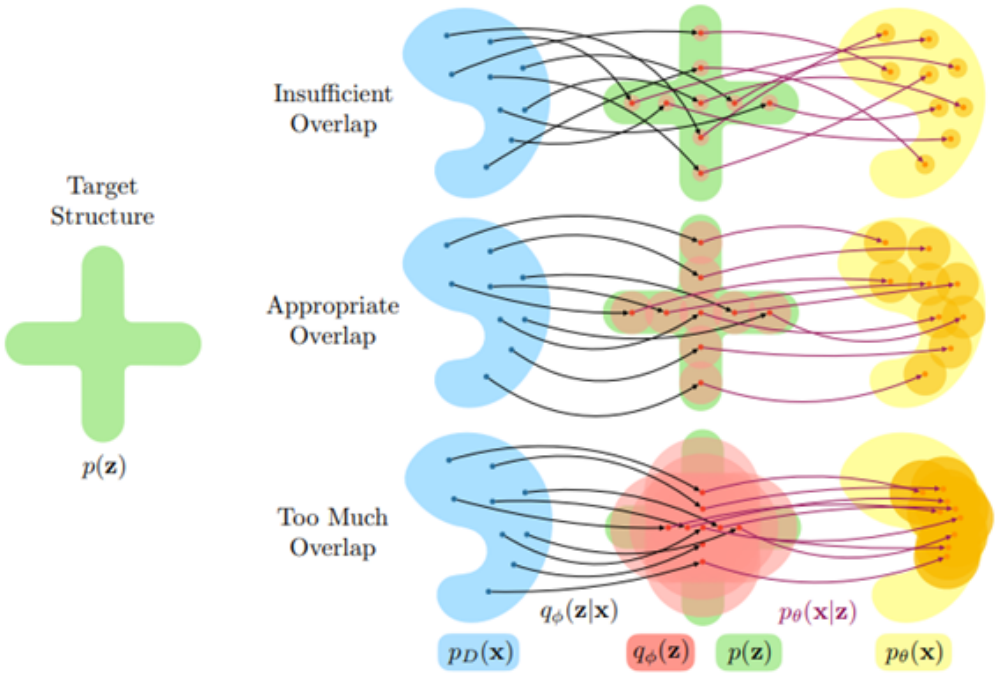

Figure 4: Illustration de la Décomposition de : Relation entre Les données d'entrée, la structure cible et les reconstructions

La figure 4 illustre l'importance d'un **chevauchement approprié** entre les encodages latents $q_\phi(z)$ et le prior $p(z)$ pour obtenir une représentation latente significative.
Lorsque le chevauchement est **insuffisant**, les encodages latents agissent comme une *"lookup table"*, où des points proches dans l'espace des données $p_D(x)$ ne sont pas proches dans l'espace latent, ce qui détruit la structure latente.
À l'inverse, un **chevauchement excessif** rend les encodages indistincts et peu informatifs, empêchant la correspondance entre $q_\phi(z)$ et $p(z)$, et dégradant ainsi la signification de l'espace latent.
Seul un **chevauchement modéré** permet aux encodages latents de correspondre à la structure cible définie par le prior (ici une croix), tout en capturant fidèlement les relations entre les données.

Contrairement au **désentrelacement traditionnel**, qui se limite à l'indépendance des dimensions latentes, la **décomposition** permet d'imposer une gamme plus large de propriétés sur les représentations latentes. Elle repose sur deux facteurs fondamentaux :

1. **Un chevauchement approprié des encodages latents**, garantissant que la représentation n'est ni trop spécifique (*lookup table*) ni trop généralisée (perte d'information).
2. **Une conformité de la distribution agrégée des encodages** $q_\phi(z)$ avec une **structure cible** représentée par un prior $p(z)$.

Cette formulation flexible englobe le **désentrelacement** comme un cas particulier, mais permet également d'imposer des structures complexes telles que la **parcimonie**, le **regroupement** (clustering), ou encore des **dépendances hiérarchiques**.

PROPOSITION D'UN NOUVEL OBJECTIF (UNE NOUVELLE FORMULE DE LOSS)

Pour les VAEs standards, cet objectif est l'Evidence Lower Bound (ELBO), qui équilibre deux termes : la qualité de reconstruction des données et une régularisation pour aligner l'encodage latent avec un prior donné.
Dans l'article, les auteurs proposent un nouvel objectif, noté $L_{\alpha,\beta}(x)$, pour prendre en compte simultanément les deux facteurs de la décomposition : Le chevauchement approprié (*a*). L'adéquation à une structure cible définie par le prior (*b*).
L'objectif proposé par les auteurs s'écrit comme suit :

$$L_{\alpha,\beta}(x) = \mathbb{E}_{q_\phi(z|x)}\left[\log p_\theta(x|z)\right] - \beta\, KL\left(q_\phi(z|x) \,\|\, p(z)\right) - \alpha\, D\left(q_\phi(z), p(z)\right),$$

où :

- $\mathbb{E}_{q_\phi(z|x)}\left[\log p_\theta(x|z)\right]$: mesure la capacité du modèle à **reconstruire les données d'origine** x à partir de la représentation latente z.
- $\beta\, KL\left(q_\phi(z|x) \,\|\, p(z)\right)$: contrôle le **chevauchement latent** (facteur *a*) en régulant la proximité entre $q_\phi(z|x)$ (le posterior estimé) et $p(z)$ (le prior choisi).
- $\alpha\, D\left(q_\phi(z), p(z)\right)$: ajoute une **régularisation supplémentaire** pour aligner la distribution agrégée des encodages $q_\phi(z)$ avec le prior $p(z)$ (facteur *b*).

Avec α et β tels que :

- β ajuste l'importance du contrôle du **chevauchement latent** (facteur *a*).
- α ajuste l'importance de l'**alignement global** de l'encodage agrégé avec le prior (facteur *b*).

PROPOSITION D'UN SYSTÈME DE PRIORS

Dans les Variational Autoencoders (VAEs), le prior représente une hypothèse sur la structure ou la distribution des représentations latentes, les auteurs démontrent que des modifications subtiles du prior peuvent améliorer significativement le désentrelacement, en particulier en remplaçant le prior gaussien isotrope classique par des priors non isotropes.
Les auteurs ont proposé 3 priors, nous nous focaliserons sur le 3eme de parcimonie (clustering) qui utilise le dataset Fashion-MNIST.

PRIOR POUR LA PARCIMONIE (SPARSITY)

Le prior pour la parcimonie proposé dans l'article a pour objectif de structurer l'espace latent de manière à n'activer qu'une fraction limitée des dimensions pour chaque échantillon. Cette approche garantit des représentations compactes et interprétables, où seules les dimensions pertinentes participent activement à l'encodage. Le prior est conçu comme un mélange de deux distributions gaussiennes, favorisant l'extinction de certaines dimensions tout en permettant à d'autres de capturer les variations importantes des données. Il est défini comme suit :

$$p(z) = \prod_d \left((1-\gamma)\mathcal{N}(z_d; 0, 1) + \gamma\mathcal{N}(z_d; 0, \sigma_0^2)\right),$$

où γ contrôle la proportion de dimensions actives, et σ_0^2, fixé à 0.05, limite la variance des dimensions inactives, les forçant à rester proches de zéro.

DATASET UTILISÉ : FASHION-MNIST

Pour tester l'efficacité de ce prior, les auteurs utilisent le dataset Fashion-MNIST, un benchmark populaire pour l'évaluation des modèles d'apprentissage automatique. Ce dataset contient 60,000 images d'entraînement et 10,000 images de test, chaque image représentant un article vestimentaire en niveaux de gris (28x28 pixels). Il inclut 10 classes (t-shirts, pantalons, robes, chaussures, sacs, etc.), offrant une complexité légèrement supérieure à celle de MNIST classique. L'objectif est de structurer l'espace latent avec le prior sparse tout en préservant une reconstruction fidèle des images.

Figure 5: Echantillon du Dataset Fashion-MNIST

Les auteurs évaluent la parcimonie à l'aide de la métrique de Hoyer :

$$\text{Hoyer}(\mathbf{y}) = \frac{\sqrt{d} - \frac{\|\mathbf{y}\|_1}{\|\mathbf{y}\|_2}}{\sqrt{d} - 1}, \quad \mathbf{y} \in \mathbb{R}^d.$$

Cette métrique varie entre 0 (représentation dense) et 1 (représentation entièrement sparse). Afin d'éviter des biais dus à des échelles différentes dans les dimensions latentes, les encodages sont normalisés avant calcul. Les résultats montrent que le prior sparse ($\gamma = 0.8$) améliore significativement la parcimonie des représentations par rapport à un prior gaussien standard ($\gamma = 0$).

REPRODUCTION DES EXPÉRIENCES SUR LE PRIOR DE PARCIMONIE

Nous avons commencé par forker le dépôt GitHub de l'article et installé les bibliothèques nécessaires à partir du fichier `requirements.txt`.

Après avoir téléchargé et configuré toutes les dépendances, nous avons lancé notre modèle d'entraînement en utilisant le dataset *Fashion-MNIST*.

Afin de reproduire fidèlement les résultats obtenus par les auteurs, nous avons scrupuleusement suivi les *"experimental settings"* fournis dans l'article. Cela inclut l'utilisation des mêmes paramètres, tels que :

- le nombre d'epochs,
- la taille du batch (*batch size*),
- la taille de l'espace latent,
- ainsi que les valeurs des différentes variables : α, β et γ spécifiées pour chaque expérience.

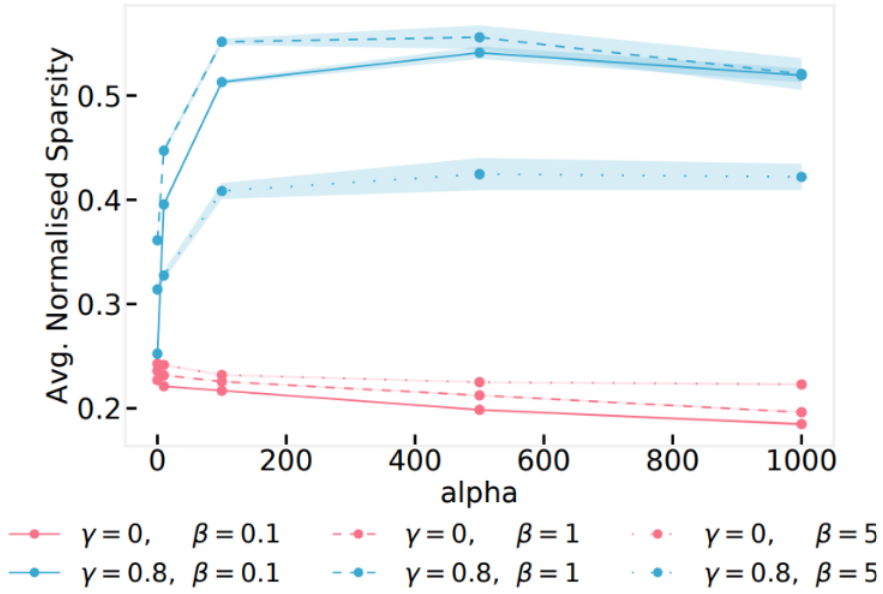

Figure 6: Impact du paramètre α, β et γ sur la parcimonie

La figure illustre l'impact du prior pour la parcimonie sur la représentation latente en fonction de la régularisation via le paramètre α, en utilisant différents réglages pour γ (proportion de dimensions actives) et β (contrôle du chevauchement).
La figure montre que l'augmentation de α renforce la parcimonie dans les représentations latentes, mesurée à l'aide d'une métrique de Hoyer normalisée. Lorsque $\gamma = 0.8$ (prior sparse), la parcimonie est nettement plus élevée qu'avec $\gamma = 0$ (prior gaussien classique). Cette tendance confirme que le prior sparse favorise une activation limitée des dimensions latentes, réduisant ainsi le nombre de dimensions actives. Cependant, après une certaine valeur de α, la parcimonie se stabilise, indiquant un point où les représentations atteignent leur niveau optimal de sparsité.

L'effet des paramètres α et β sur le prior pour la parcimonie est crucial pour équilibrer la structure des représentations latentes et la qualité des reconstructions. Une augmentation de α renforce la correspondance entre $q_\phi(z)$, la distribution agrégée des encodages latents, et $p(z)$, le prior, ce qui améliore la parcimonie. Cependant, des valeurs trop élevées de α peuvent légèrement affecter la qualité des reconstructions.

En parallèle, β, qui régule le chevauchement des encodages latents, doit être ajusté avec soin : des valeurs trop faibles ou trop élevées peuvent nuire à la parcimonie en déséquilibrant la reconstruction et la régularisation.

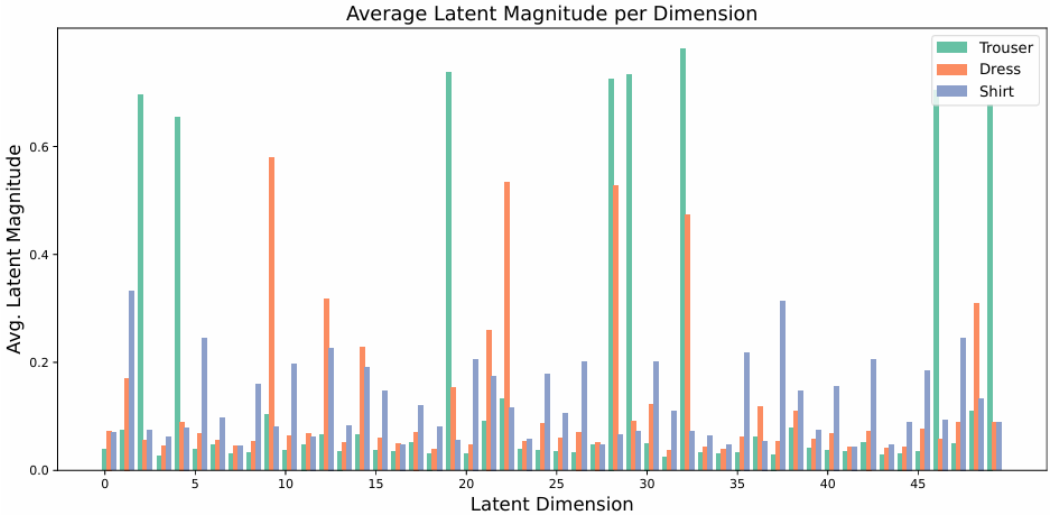

Figure 7: Visualisation du désentrelacement dans l'espace latent pour 3 classes

Cette visualisation est le résultat d'un réseau entraîné avec $\alpha = 1000$, $\beta = 1$ et $\gamma = 0.8$ et montre la *magnitude moyenne des encodages latents* pour trois classes de *Fashion-MNIST* (pantalons, robes et chemises), démontrant que différentes dimensions latentes sont activées pour des caractéristiques spécifiques.
Par exemple :

- La séparation des jambes des pantalons (dimension $2, 4, 19, 28, 29, 32, 46, 49$).
- La largeur du col des robes (dimension $9, 22, 28, 32$).
- La forme des chemises (parcimonie basse, dimension $1, 5, 37$).

Nous observons clairement le caractère sparse de cette visualisation. En effet, certaines dimensions latentes (ou nœuds) restent pratiquement inactives et ne contribuent pas de manière significative à l'apprentissage des trois classes étudiées (pantalons, robes et chemises), tandis que d'autres dimensions sont fortement sollicitées.
Ces caractéristiques sont capturées dans des dimensions latentes distinctes, démontrant la capacité du modèle à apprendre des représentations spécifiques pour chaque facteur de variation présent dans les données.

Après avoir réussi à reproduire les résultats de l'article avec le dataset initiale, nous procédons à les adapter à un environnement structuré, représenté par le dataset Rotated-MNIST.

4 NOTRE TRAVAIL : REPRODUIRE L'ARTICLE DANS UN ENVIRONNEMENT (SETTING) STRUCTURÉ

Également pour adapter notre nouveau dataset au code de l'article, nous avons utiliser le même environnement MyDocker IA GPU - Torch - python3.12, aussi avec l'outil JupyterLab .

4.1 ROTATED-MNIST

Le dataset *Rotated-MNIST* est une variante du célèbre dataset *MNIST*, conçu pour évaluer la capacité des modèles à gérer des variations géométriques, notamment les rotations. Ce dataset est constitué de $60\,000$ images en niveaux de gris, où chaque image représente un chiffre (de 0 à 9) avec une résolution de 28×28 pixels, auquel est appliqué un angle de rotation aléatoire, souvent compris entre $0°$ et $360°$.
Ce dataset représente un **setting structuré**, ce qui signifie que les variations dans les données suivent une organisation claire et bien définie. Cela permet d'évaluer des propriétés spécifiques des modèles. Ce setting structuré garantit que les variations sont introduites de manière systématique et contrôlée, en combinant deux facteurs de variation distincts : l'identité discrète du chiffre (de 0 à 9) et l'angle de rotation continu. Cette structuration permet aux chercheurs de tester et d'analyser précisément la robustesse et les capacités des modèles dans des contextes où il est crucial de capturer ces deux dimensions de manière séparée.
Le Rotated MNIST est particulièrement utilisé dans des tâches telles que la classification de chiffres indépendamment de leur orientation, la reconstruction d'images, et l'apprentissage désentrelacé. Son setting structuré offre une opportunité unique pour les modèles d'apprendre à séparer les facteurs de variation génératifs, comme l'identité du chiffre et l'angle de rotation, démontrant ainsi leur capacité à produire des représentations latentes interprétables et robustes.

Figure 8: Echantillons du dataset Rotated-MNIST

Nous avons la possibilité d'utiliser le dataset Rotated-mnist de deux facons : Task-Specific (Spécifique à un angle de rotation) et Flattened (Aplati en entrée).

4.1.1 Task-Specific (Spécifique à un angle de rotation)

Dans cette approche, chaque angle de rotation est traité comme une tâche distincte, et un modèle ou sous-module spécifique est dédié à chaque groupe d'images correspondant à un angle particulier. Les données sont segmentées par angle, ce qui permet au modèle d'apprendre des représentations spécialisées pour chaque rotation

4.1.2 Flattened (Aplati en entrée)

Dans cette approche, toutes les rotations possibles des chiffres du dataset *Rotated-MNIST* (par exemple, entre $0°$ et $360°$) sont combinées dans un même ensemble de données et traitées simultanément par un unique modèle. L'idée est d'aplatir les variations liées aux rotations pour entraîner un encodeur capable de gérer ces transformations de manière générique.

Dans ce travail nous essayons principalement de capturer l'effet que peut avoir un environnement structuré (sur les degrés de rotation) tel que Rotated-MNIST sur l'espace latent, c'est pour cela nous nous focaliserons sur la méthode Flattened.
Pour toutes les expériences, nous utilisons une configuration *aplatie* avec 5 variations de $45°$, en commençant par $0°$.

4.2 EXPÉRIENCES

Nous avons suivi les mêmes étapes que lors de notre reproduction de l'article. Nous avons commencé par forker le dépôt GitHub de l'article, puis nous avons utilisé la classe `rotated_mnist.py` fournie, en l'ajoutant comme option dans le paramètre `MODEL_NAME`.
Tout d'abord, nous avons tenté de produire une représentation similaire à celle de la *figure 6*, illustrant l'impact des paramètres α, β et γ sur la parcimonie, mais cette fois en utilisant notre propre dataset. Pour cela, nous avons appliqué les mêmes configurations que celles spécifiées dans l'article :

- `--epochs 80`
- `--batch-size 500`
- `--beta 1.0`
- `--alpha 1000`
- `--gamma 0.8`

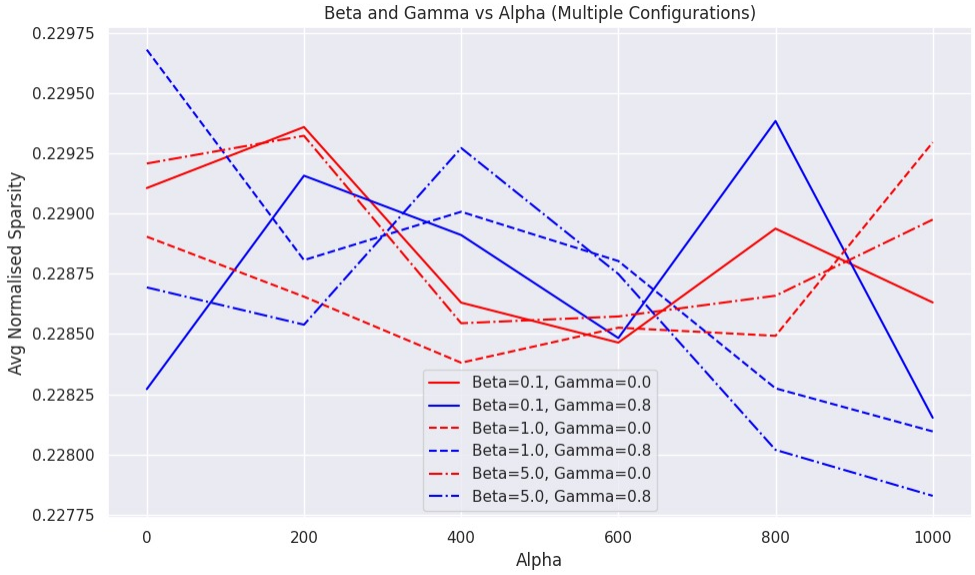

Figure 9: Impact du paramètre α, β et γ sur la parcimonie

Par la suite, nous avons tenté de visualiser le *désentrelacement* dans l'espace latent pour les trois premières classes (similaire a la figure 7), à savoir les chiffres *zéro*, *un* et *deux*. Cette visualisation a été réalisée en utilisant différentes configurations, parmi lesquelles : avec :

- `--epochs 80`
- `--batch-size 500`
- `--Dimension latente 50`

- `--beta 1.0`
- `--alpha 1000`
- `--gamma 0.8`

qui est la meme que leur configuration, nous obtenons :

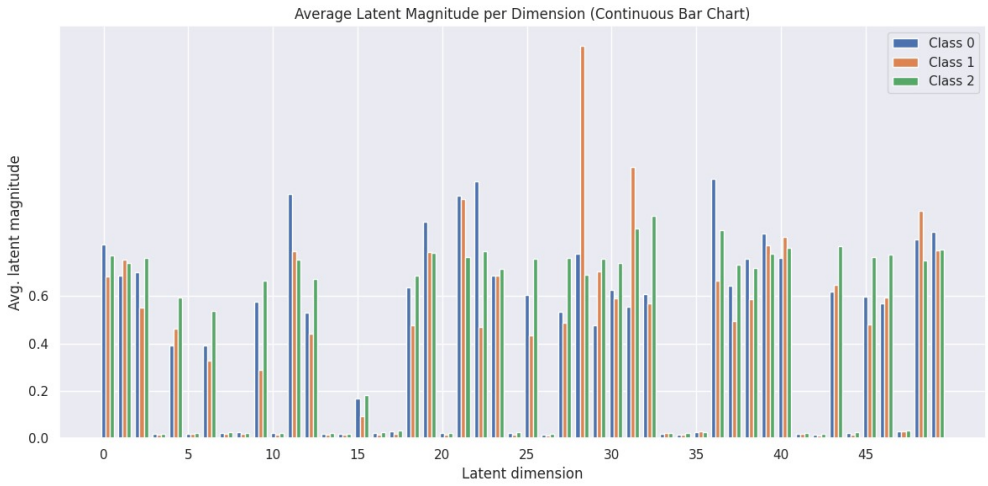

Figure 10: Visualisation du désentrelacement sur 3 classes

Aussi avec :

- `--epochs 80`
- `--batch-size 500`
- `--Dimension latente 50`
- `--beta 128`
- `--alpha 0`
- `--gamma 0.8`

Nous obtenons :

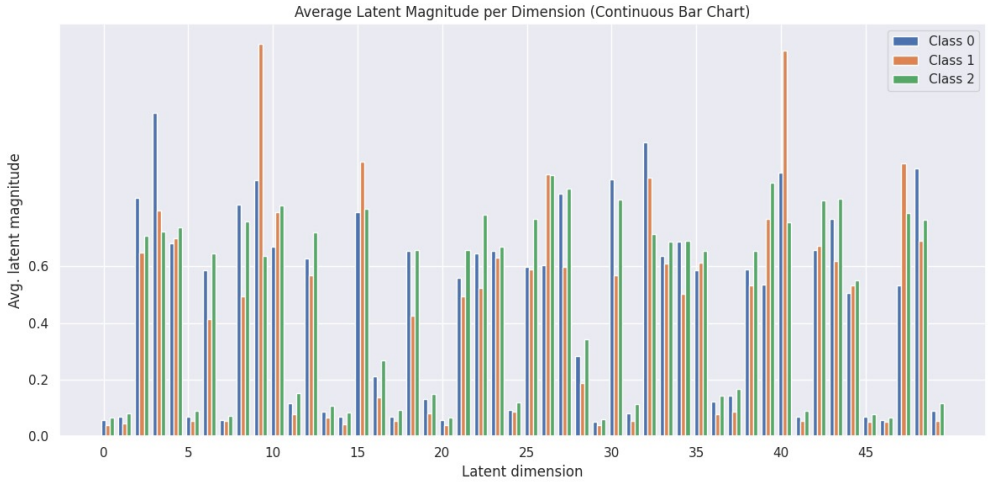

Figure 11: Visualisation du désentrelacement sur 3 classes

D'après nos expériences, nous remarquons que l'augmentation des valeurs de α et β permet d'obtenir une certaine *parcimonie*, bien que celle-ci reste relativement faible. Toutefois, il est intéressant de constater que l'augmentation de α procure une parcimonie plus marquée que celle obtenue en augmentant uniquement β.

Cependant, nos visualisations ne sont pas suffisamment concluantes pour affirmer que les facteurs de rotation sont correctement capturés par certaines dimensions latentes et pas d'autres, ce qui témoigne d'une parcimonie faible. Afin de vérifier cette hypothèse, nous avons décidé de comparer nos résultats avec un *VAE standard* afin de déterminer si une différence visible en termes de parcimonie peut être observée.
Après réflexion, nous avons réalisé qu'il n'était pas nécessaire d'implémenter un tout nouveau VAE traditionnel. En effet, nous pouvons simplement utiliser la configuration suivante :

- $\beta = 1$,
- $\alpha = 0$,
- $\gamma = 0$.

Cette configuration revient à utiliser la formule de perte traditionnelle sans l'impact des variables supplémentaires introduites dans l'article. Nous avons alors entraîné le modèle avec les paramètres suivants :

- `80 epochs`,
- `batch size = 500`.

Voici la représentation obtenue :

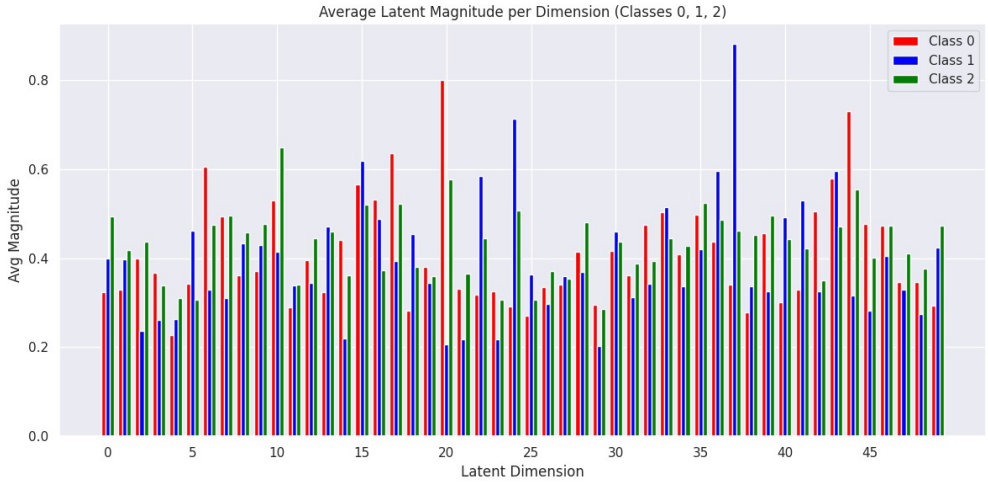

Figure 12: Visualisation du désentrelacement sur 3 classese

Ici, nous observons clairement la différence : l'absence de parcimonie sur cette figure s'aligne avec nos prédictions initiales ainsi qu'avec celles des auteurs de l'article. L'augmentation des paramètres α et β permet effectivement d'introduire de la parcimonie et d'apporter du sens à l'espace latent.
Ensuite nous avons procedé a visualiser les espaces latent pour voir si on peut extraire du sens par exemple des regroupement ou des clusters de representations de données sur l'espace latent :
Donc en utilisant la configuration suivante:

- `--epochs 80`
- `--batch-size 500`
- `--Dimension latente 50`
- `--beta 1`
- `--alpha 0`

nous avons obtenu :

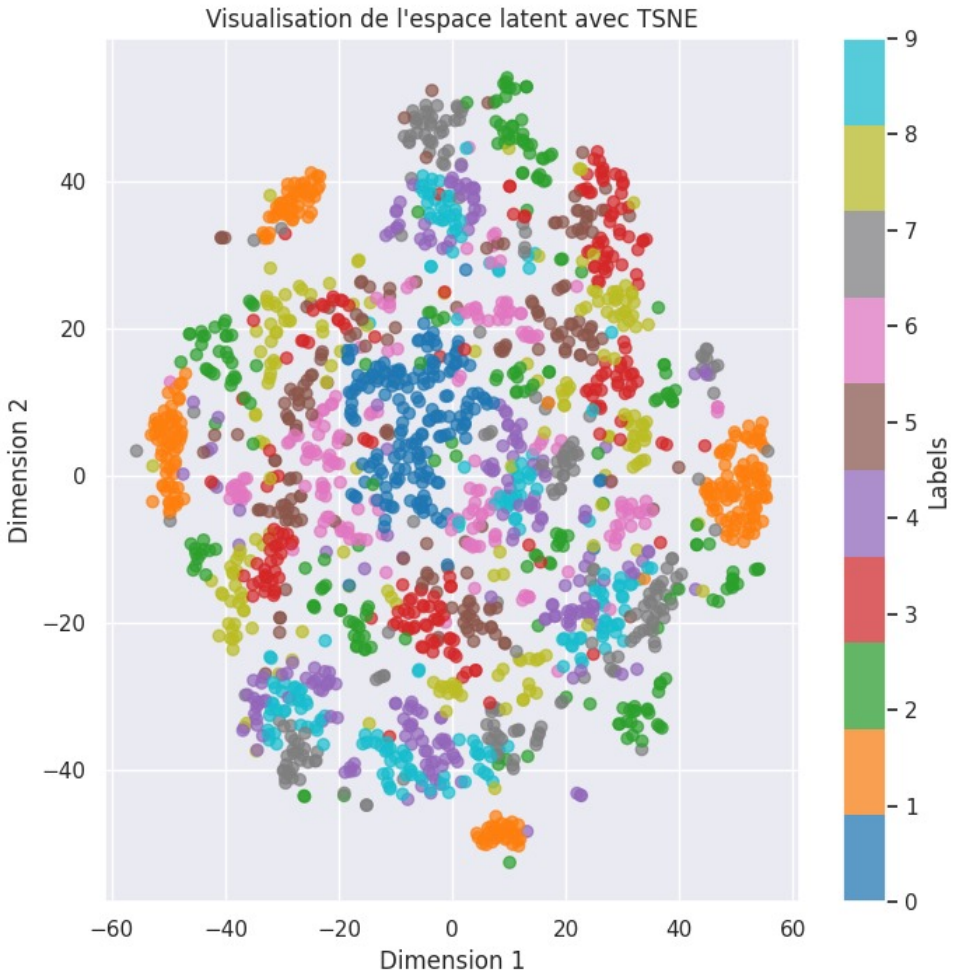

Figure 13: Visualisation de l'espace latent avec TSNE

Nous utilisons le TSNE afin de projeter les données de notre espace latent à haute dimension (50 dimensions) vers une representation basse dimension (2 D) tout en préservant les relations de proximité entre les points.
Et avec cette configuration:

- `--epochs 80`
- `--batch-size 500`
- `--Dimension latente 50`
- `--beta 10`
- `--alpha 100`

nous avons obtenu :

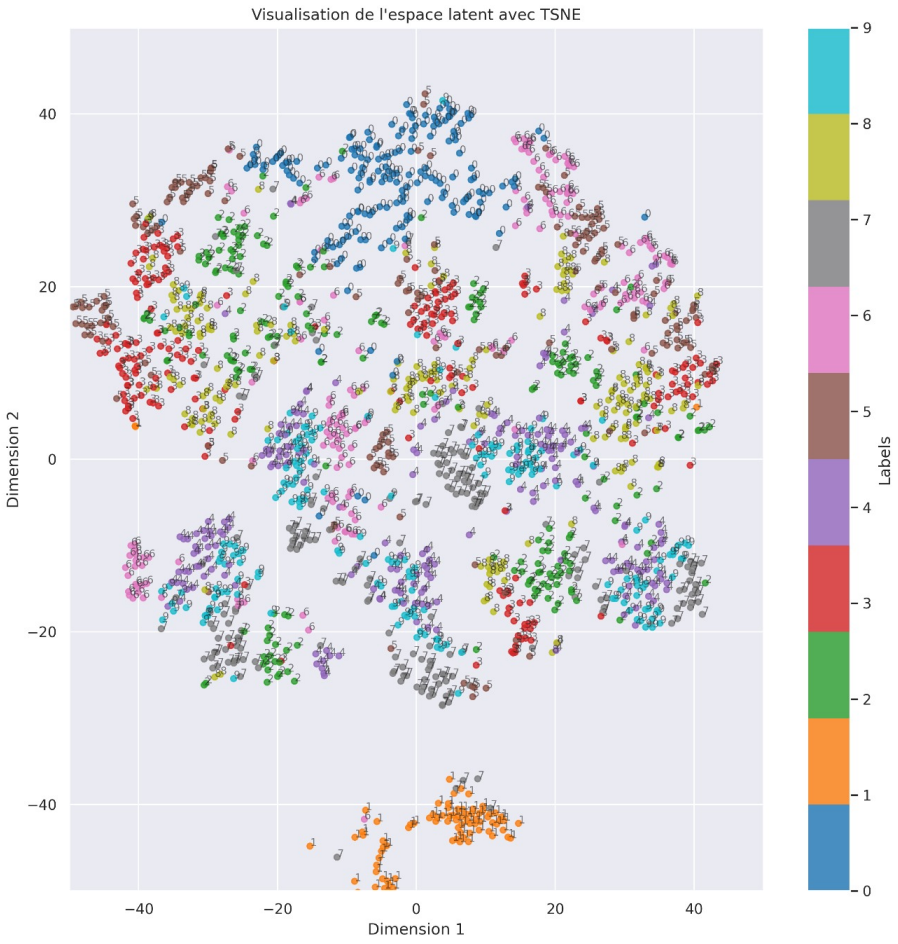

Figure 14: Visualisation de l'espace latent avec TSNE

Nous observons un *regroupement* nettement plus cohérent lorsque les valeurs de α et β sont élevées. Cela apporte également du *sens* à l'espace latent. Par exemple, le groupe correspondant au chiffre *1* (en orange) se trouve systématiquement proche du groupe représentant le chiffre *7* (en gris). Cette proximité est logique, car dans la manière d'écrire ces deux chiffres, leurs formes présentent une certaine similarité.

5 PERSPECTIVES POUR LE FUTUR

- **Optimisation des hyperparamètres** : Affiner les valeurs de α, β, et γ pour des résultats plus robustes.
- **Amélioration de la parcimonie** : Tester d'autres types de priors structurés ou régularisations pour mieux capturer les rotations.
- **Exploitation de la stratégie task-specific de Rotated-MNIST** : Appliquer cette stratégie au sein d'un *federated learning*, qui sera le sujet de notre projet de master.

6 CONCLUSION

Au final, nous avons réussi à comprendre le contexte général entourant l'article *"Disentangling Disentanglement in Variational Autoencoders"* ainsi que ses méthodes spécifiques. Nous avons reproduit avec succès la majorité des résultats et expériences présentés par les auteurs, en suivant fidèlement leurs configurations expérimentales. Une contribution majeure de notre travail a été

l'adaptation du code proposé dans l'article à un environnement structuré : le dataset *Rotated-MNIST*, conçu pour introduire des variations de rotation systématiques. Cette adaptation nous a permis d'explorer et de visualiser les aspects essentiels de nos expériences et de notre espace latent, en particulier en analysant l'impact des paramètres α, β et γ sur la parcimonie et l'organisation des dimensions latentes. Cependant, nous avons rencontré certaines difficultés dans l'interprétation des

visualisations obtenues. Bien que nous ayons observé des différences notables dans la structuration de l'espace latent, il reste complexe de déterminer avec certitude si l'ajout de paramètres de rotation a eu un impact significatif et cohérent sur les dimensions latentes apprises. Ces limites ouvrent la voie à des investigations futures plus approfondies. En somme, notre travail nous a permis de

démontrer que l'adaptation des méthodes de l'article à un environnement structuré peut offrir des perspectives intéressantes pour mieux comprendre l'organisation des facteurs de variation. Malgré les défis rencontrés, ces résultats représentent une étape prometteuse pour des études futures, notamment dans des contextes plus avancés comme le *federated learning* appliqué à *Rotated-MNIST*.

7 Annexe

Problèmes Rencontrés

Lors de la reproduction du Papier

- **Problème** : Avertissement lié à `torch.load` (pickle non sécurisé).
  Solution : Utilisation de `weights_only=True` dans le futur pour éviter les risques de sécurité.
- **Problème** : Manque de reproductibilité dans les résultats.
  Solution : Fixer les graines avec `torch.manual_seed()`, `random.seed()`, et `np.random.seed()` et activer `torch.backends.cudnn.deterministic = True`.
- **Problème** : Erreur d'attribut lors de l'accès aux labels (`int object has no attribute 'view'`).
  Solution : Remplacer la ligne avec `L = 1` car les labels étaient déjà sous forme d'entiers.
- **Problème** : Affichage des classes individuellement dans `plot_latent_magnitude`.
  Solution : Combiner les barres des 3 classes (*Trouser, Dress, Shirt*) sur un seul graphique.
- **Problème** : Ticks des axes non alignés avec les besoins (dimensions latentes et valeurs spécifiques).
  Solution : Modifier les `xticks` pour afficher 0, 5, 10, ..., 45 et les `yticks` pour 0.0, 0.2, 0.4, 0.6.
- **Problème** : Configuration d'un VAE classique.
  Solution : Utiliser `--beta 1.0 --alpha 0.0 --gamma 0.0` pour désactiver les régularisations supplémentaires.

Lors de l'adaptation sur le nouveau dataset

- **Problème** : Lignes plates et superposées dans le graphe `beta_gamma_vs_alpha`.
  Cause : Les valeurs de sparsité ne variaient pas correctement.
  Solution : Vérifier l'implémentation de `compute_sparsity` pour s'assurer qu'il fonctionne dynamiquement avec les valeurs de `beta` et `gamma`.
- **Problème** : `NaN` dans les données latentes (`zs_mean`).
  Solution : Vérifier les valeurs avec des assertions pour détecter et corriger des problèmes numériques.
- **Problème** : Affichage des ticks dans les plots continus.
  Solution : Utiliser `set_xticks` et `set_yticks` pour personnaliser les positions des ticks.
- **Problème** : Perte trop élevée pendant l'entraînement initial.
  Solution : Ajuster l'apprentissage en réduisant `beta` et en ajustant le *learning rate*.
- **Debugging** : Ajout de logs (`print`) pour suivre les étapes critiques (ex. : dimensions latentes, valeurs de sparsité).

Modifications Apportés au Code

Voici le lien vers notre GitHub où nous effectuons une comparaison complète, incluant les ajouts et les suppressions apportés au code initial clien : Lien vers GitHub.

References

[1] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess et al. "Learning Basic Visual Concepts with a Constrained Variational Framework.," In: *International Conference on Learning Representations*