Espace latent compact et sémantique pour l'édition désenchevêtrée d'images

Gwilherm LESNÉ Yann GOUSSEAU Saïd LADJAL Alasdair NEWSON
LTCI, 19 place Marguerite Perey, CS 20031, F-91123 Palaiseau Cedex, France

Résumé – Les progrès récents dans le domaine des modèles génératifs et en particulier des réseaux antagonistes génératifs (GAN) ont permis des avancées significatives pour l'édition contrôlée d'images. Malgré leur capacité à appliquer des modifications réalistes à une image, ces méthodes peinent souvent à permettre l'édition d'attributs indépendamment les uns des autres, autrement dit à *désenchevêtrer* ces attributs. Dans cet article, nous proposons un nouvel auto-encodeur latent permettant d'améliorer cette caractéristique, potentiellement applicable à tout GAN pré-entraîné. Nous testons l'approche sur la classique architecture StyleGAN et montrons sa capacité à désenchevêtrer des attributs de visages.

Abstract – Recent advances in the field of generative models and in particular generative adversarial networks (GANs) have enabled significant advances for controlled image editing. Despite their powerful ability to apply realistic modifications to an image, these methods often lack properties like disentanglement (the capacity to edit attributes independently). In this paper, we propose a new latent auto-encoder allowing to improve this property for any pretrained GAN. We test the approach on the classical StyleGAN architecture and show its capacity to disentangle faces attributes.

1 Introduction

Les modèles de génération d'images par apprentissage profond permettent depuis quelques années de créer des images de manière photoréaliste. Parmi les méthodes les plus connues, nous pouvons notamment citer les auto-encodeurs variationnels, les flux normalisants, les modèles de diffusion ou encore les réseaux adversaires (GAN). En particulier, les GANs de type style tel que BigGAN [3] ou StyleGAN[7] se sont distingués pour leur capacité de synthèse à haute résolution. Le principe de ce dernier est de transformer une distribution connue $\mathcal{Z}$(généralement normale) en une distribution intermédiaire $\mathcal{W}$ qui sera ensuite utilisée pour générer des images. Nous nous concentrerons dans ce papier sur ce type de réseaux et plus particulièrement sur StyleGAN2 [8], qui est une version améliorée de StyleGAN. Malgré les très importants progrès permis par ces méthodes, un problème récurrent est celui de la capacité à éditer indépendemment des attributs dans un espace latent, capacité généralement désignée sous le nom de *désenchevêtrement* (*disentanglement* en anglais) des attributs.

Dans ce travail, nous proposons une méthode d'édition qui structure l'espace latent $\mathcal{W}$ avec une contrainte de désenchevêtrement tout en conservant une stabilité comparable aux méthodes état-de-l'art. Enfin, grâce à notre méthode, il est possible d'échantillonner $\mathcal{W}$ conditionnellement à certains attributs sémantiques.

Comme la plupart des méthodes d'édition d'images par manipulation de l'espace latent de StyleGAN, nous nous limiterons à des images de visages.

2 Travaux liés

Pour effectuer de l'édition d'image avec StyleGAN, une méthode courante est de manipuler son espace latent $\mathcal{W}$ qui est censé être plus désenchevêtré et sémantiquement interprétable que $\mathcal{Z}$. Cependant, Image2StyleGAN [1] a introduit un nouvel espace latent propre à StyleGAN : $\mathcal{W}+$. Il s'agit de considérer un code latent w différent pour chaque échelle à laquelle le style est introduit dans le réseau génératif de StyleGAN. De cette manière, $\mathcal{W}$ est étendu et permet une plus grande souplesse. En revanche, cette possibilité est propre aux réseaux de type "style" et ne peut donc pas être généralisée pour n'importe quel GAN.

En dehors de GANSpace[5] qui propose d'effectuer une ACP dans l'espace latent de StyleGAN puis d'analyser chaque composante principale qualitativement afin de trouver des directions d'édition pour différents attributs, la plupart des méthodes d'édition d'image via la manipulation de $\mathcal{W}$ sont semi-supervisées ou supervisées. Parmi celles-ci, InterfaceGAN [11] propose de trouver une direction linéaire à l'aide d'une SVM entraînée dans l'espace latent pour pouvoir modifier un attribut donné. Dans la prolongation de ces travaux, [13] entraîne un réseau à déterminer une direction d'édition par vecteur latent dans $\mathcal{W}+$.

D'autres méthodes performantes ont été proposées, toujours dans $\mathcal{W}+$, telles que StyleFlow[2] qui propose d'auto-encoder les vecteurs latents à l'aide d'un réseau de flux normalisant ou encore Latent2Latent[9] qui modifie les vecteurs grâce à un réseau dense. La plupart des méthodes fonctionnant dans $\mathcal{W}+$ utilisent des fonctions de coût basées à la fois dans l'espace latent et dans l'espace image, ce qui implique de forts coûts algorithmiques pour les entraînements.

3 Méthode

3.1 Architecture

Pour générer une image de visage photoréaliste via StyleGAN, un vecteur z est tiré dans un premier espace latent $\mathcal{Z}$ suivant une loi normale multidimensionnelle. Ce dernier est ensuite transformé en un second espace, $\mathcal{W}$, via un réseau de transformation M constitué d'un perceptron à 8 couches. Le nouveau

vecteur $w \in \mathcal{W}$ est alors inséré dans le réseau génératif G, à différentes résolutions, afin de produire une image y.

Notre motivation ici est de structurer l'espace latent $\mathcal{W}$ en associant à chaque dimension un attribut sémantique de l'image. De cette manière, il est possible d'échantillonner conditionnellement cet espace et d'effectuer des éditions dites désenchevêtrées.

Cette caractéristique est propre à l'édition d'image : l'utilisateur veut généralement pouvoir modifier une unique caractéristique sémantique dans ladite image. Par exemple, pour des photos de visages, ces caractéristiques peuvent correspondre à des attributs tels que l'âge, le genre, la présence de lunettes, le sourire, etc. Pour cela, on associe à chaque image y un vecteur d'attributs $a = (a_0, ..., a_K)$ déterminé à l'aide d'un réseau de classification F. Dans notre cas, nous prendrons $K = 40$ attributs. De cette manière, il est possible de créer un jeu de données (w,a) en échantillonnant $\mathcal{Z}$ puisque :

$$w = M(z) \;\; et \;\; a = F \circ G(w). \tag{1}$$

Contrairement à l'hypothèse d'InterfaceGAN[11] selon laquelle, pour modifier l'attribut d'une image via l'espace latent de StyleGAN, il suffirait de suivre une direction linéaire, nous supposons ici que la trajectoire d'édition d'une image puisse être plus générale. Ainsi, nous proposons d'entraîner un auto-encodeur dans l'espace $\mathcal{W}$ qui apprend ces trajectoires pour chaque attribut a_k. Nous appellerons par la suite, l'espace latent de cet auto-encodeur $\mathcal{C}$ et $c = (c_0, ..., c_{dim(\mathcal{C})})$ un vecteur latent de cet espace.

D'autre part, les travaux de GANSpace[5] ont permis de montrer que dans $\mathcal{W}$, toutes les dimensions n'avaient pas une influence égale vis-à-vis des images générées. En effet, en appliquant la méthode du coude sur l'Analyse en Composantes Principales (ACP) de $\mathcal{W}$, nous pouvons constater que 60 dimensions sont suffisantes pour reconstruire correctement une image (Figure 1) et conserver presque 80% de la variabilité de l'espace. C'est pourquoi on décide d'appliquer notre auto-encodeur dans $\mathcal{W}_{ACP}$, correspondant à la projection de $\mathcal{W}$ sur les 60 premières, au sens de la variabilité, dimensions de l'ACP. Ce choix a deux principaux avantages : Il permet d'éviter à notre auto-encodeur d'effectuer des trajectoires d'édition dans des dimensions de $\mathcal{W}$ de faible variance, évitant l'apparition d'artefacts. Cela permet aussi de réduire la complexité de notre auto-encodeur.

FIGURE 1 : Gauche : image générée avec un $w \in \mathcal{W}$. Droite : image générée avec w projeté sur 60 dimensions de $\mathcal{W}_{ACP}$

Afin de contrôler les valeurs de chacun des attributs et d'être en mesure d'échantillonner l'espace latent $\mathcal{W}$ conditionnellement à ces derniers, nous fixons commme objectif à notre encodeur E de faire correspondre les codes latents aux attributs :

$$a_k = c_k \; pour \; k \in [\![1; K]\!]. \tag{2}$$

on notera que cette définition suppose que notre espace latent $\mathcal{C}$ soit de dimension supérieure à K, le nombre d'attributs. Dans notre cas, nous avons $K = 40$ et $dim(\mathcal{C}) = 60$, les 20 dimensions restantes servant à encoder les informations restantes comme l'identité ou d'autres attributs.

Nous établissons donc le processus d'auto-encodage suivant : On projette notre vecteur $w \in \mathcal{W}$ dans $\mathcal{W}_{ACP}$. De cette manière, on ne conserve que l'information strictement nécessaire à notre réseau. Ensuite, nous procédons à l'encodage du vecteur w_{ACP} par un perceptron multi-couches pour obtenir un vecteur c. Ce dernier représente un attribut différent sur les K premières dimensions et laisse les dimensions restantes libres. Ce vecteur est ensuite passé à un décodeur pour obtenir $\hat{w}_{ACP}$. Enfin, nous appliquons l'opération inverse de l'ACP pour revenir dans l'espace d'origine et avoir notre vecteur latent édité $\hat{w}$. Un schéma récapitulatif de l'architecture est proposé figure 2.

3.2 Entraînement

Pour entraîner notre auto-encodeur, nous utilisons ici une base de données codes latents - attributs. Comme indiqué en partie 3.1, il est possible de constituer cette base à partir d'un réseau de classification pré-entraîné. Pour la fonction de coût, nous utilisons pendant l'entraînement une somme pondérée de trois termes :

- Une fonction de coût de reconstruction. Il s'agit là d'une simple norme 2 sur les codes latents de sorte à ce que le code latent w_{acp} soit correctement reconstruit par le décodeur si aucune édition n'est appliquée :

$$L_{recons}(w_{acp},\hat{w}_{acp}) = ||w_{acp} - \hat{w}_{acp}||_2^2. \tag{3}$$

- Une fonction de coût sur les attributs. Comme notre objectif est de pouvoir éditer les attributs d'une image. Nous souhaitons fixer les valeurs c_k comme égales à a_k :

$$L_{attr} = \sum_{k=1}^{K} (a_k - c_k)^2. \tag{4}$$

- Un terme de corrélation. La motivation derrière ce terme est de forcer le désenchevêtrement de $\mathcal{C}$, c'est-à-dire l'indépendance des dimensions de notre espace latent. Pour cela, nous essayons de fixer la matrice d'autocorrélation des codes latents $Corr$ à une matrice de référence $M_{ref} = I_K$:

$$L_{corr} = \sum_{i,j}^{K,K} |Corr(i,j) - M_{ref}(i,j)|. \tag{5}$$

  On notera que la matrice $Corr$ est calculée sur un batch. Il est donc nécessaire d'avoir une taille de batch suffisamment importante durant l'entraînement. Par ailleurs, comme notre objectif ici est d'améliorer le désenchevêtrement, on a $M_{ref} = I_K$, mais il est possible d'utiliser une autre matrice dans le cas où on souhaiterait avoir un espace latent qui prend en compte certaines corrélations.

La fonction de coût finale est donc de la forme :

$$L_{totale} = L_{recons} + \alpha L_{attr} + \beta L_{corr}. \tag{6}$$

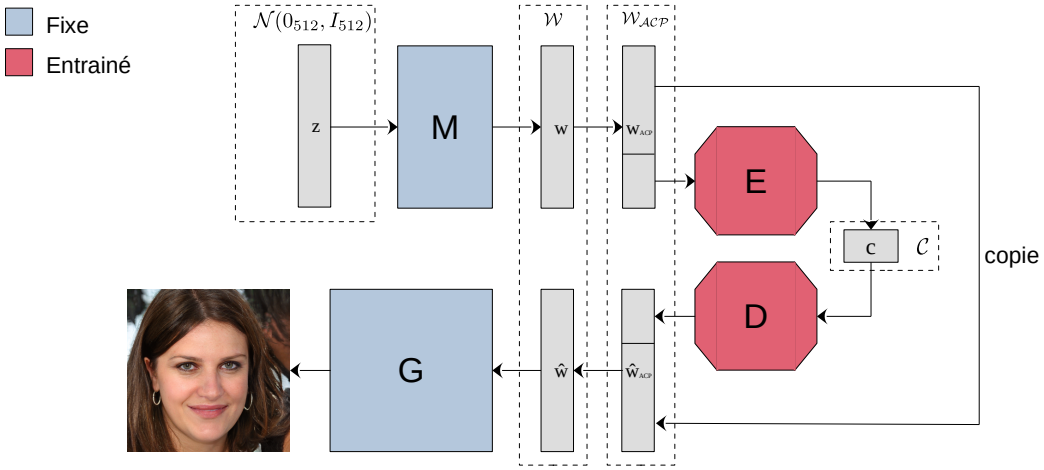

FIGURE 2 : Architecture de notre pipeline d'édition pour une image tirée depuis l'espace latent $\mathcal{Z}$

4 Expériences et métriques

4.1 Paramétrisation

Pour entraîner notre réseau, nous avons fixé expérimentalement $\alpha = 10^{-5}$ et $\beta = 10^{-5}$. L'encodeur et le décodeur possèdent la même architecture, un perceptron de 8 couches avec une taille de couche cachée de 512. L'entraînement est effectué pour 150 époques avec une taille de batch de 256. M_{ref} correspond à la matrice identité.

Notre base de données est constituée de 200 000 codes latents et les attributs correspondants sont déterminés par un réseau de type EfficientNet-B3[12] pré-entraîné sur CelebA[10]. En général, la plage de valeur des attributs d'une image est $[0,1]$, ce qui est permis par l'utilisation d'une sigmoïde en sortie du réseau de classification. Or, l'utilisation de cette fonction d'activation conduit à une sous-représentation des valeurs proches de $0,5$ dans l'intervalle. A cause de ce comportement, notre réseau risque d'apprendre à encoder une distribution bimodale pour chaque code latent et donc de ne pas être capable de produire des résultats vraisemblables pour $c_k = 0,5$. Pour éviter un tel comportement, nous décidons de "gaussianiser" les valeurs d'attribut de notre base de donnée. C'est-à-dire que pour chaque attribut, nous appliquons une égalisation d'histogramme de ses valeurs puis l'inverse de la fonction de répartition de la loi normale. De cette manière, nous pouvons toujours manipuler l'attribut souhaité dans une plage de valeurs connue à l'avance et remédier au problème.

4.2 Évaluation Qualitative

Nous proposons quelques comparaisons visuelles de notre méthode d'édition à celle d'InterfaceGAN [11] à la figure 3. On notera que sur ces exemples le désenchevêtrement est meilleur pour notre méthode.

4.3 Évaluation Quantitative

Comme notre objectif est d'améliorer le désenchevêtrement tout en maintenant une qualité d'édition comparable aux méthodes état-de-l'art, nous considérons 3 métriques quantitatives pour évaluer notre méthode : la conservation d'identité lors d'une édition, la corrélation des variations d'attributs pour quantifier le désenchevêtrement ainsi que le taux d'images correctement éditées (comme défini ci-dessous).

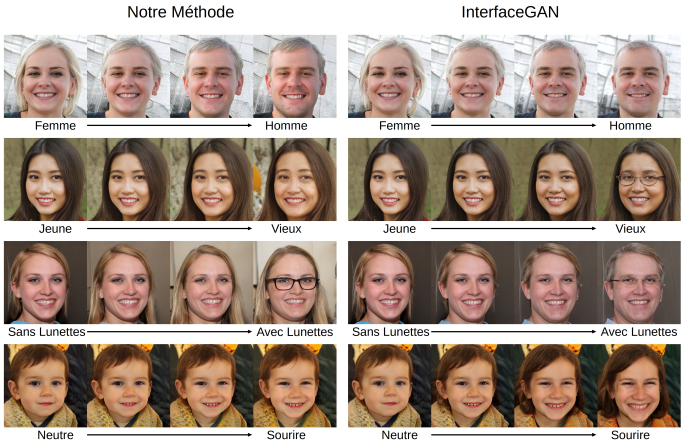

FIGURE 3 : Comparaison d'éditions avec notre méthode et InterfaceGAN.

Pour chacune de ces métriques, nous nous comparons à InterfaceGAN, puisqu'il s'agit de la seule méthode état de l'art dans $\mathcal{W}$, espace dans lequel nous travaillons. En effet, les autres méthodes d'édition via StyleGAN utilisent toutes $\mathcal{W}+$ qui se trouve être bien plus permissif mais qui ne correspond pas à l'espace sur lequel le générateur G à été entraîné.

Afin de mesurer les 3 quantités ci-dessus, nous échantillonnons $N = 256$ images dans $\mathcal{W}$ avec un attribut cible, k, négatif : $(w_1^{k-}, ..., w_N^{k-})$. Nous réalisons ensuite une édition de sorte à le rendre positif. Pour cela, nous appliquons l'édition à différentes amplitudes et mesurons la valeur de l'attribut cible. Lorsque celle-ci dépasse 0.9, nous considérons l'édition comme finie et conservons les vecteurs latents correspondant : $(w_1^{k+}, ..., w_N^{k+})$. Grâce à ces paires de vecteurs, nous pouvons alors mesurer les différences d'attributs et d'identité lors de l'édition de l'attribut k. Étant donné que nous possédons 40 attributs, dans un soucis de clarté, nous avons choisi de restreindre nos mesures quantitatives à 4 attributs cible : *lunettes*, *genre*, *sourire* et *âge* (ces attributs sont souvent utilisés dans la littérature).

Comme la propriété de désenchevêtrement suppose qu'une unique caractéristique sémantique doit être modifiée au cours d'une édition, notre objectif est de minimiser les variations d'attributs en dehors de l'attribut cible. Pour visualiser cette propriété, nous affichons une matrice (Figure 4) où chaque élément correspond à la moyenne des variations de l'attribut colonne au cours d'une édition de l'attribut ligne :

$$Mat_{k,l} = \frac{1}{N}\sum_{i=1}^{N}(F_l \circ G(w_i^{k+}) - F_l \circ G(w_i^{k-})). \quad (7)$$

où F_l est l'estimation de la valeur de l'attribut l par le réseau F.

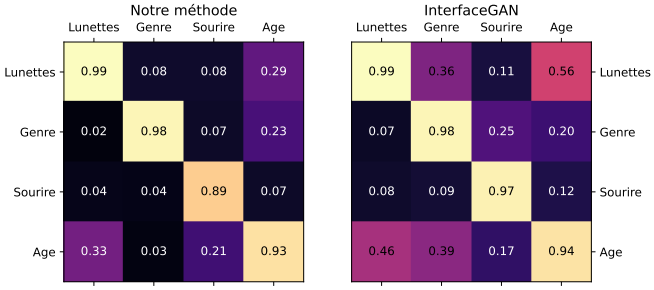

FIGURE 4 : Matrices de variation des attributs. Chaque ligne correspond à l'édition d'un attribut et indique les variations de tous les attributs. Autrement dit, le coefficient (k,l) correspond à la variation de l'attribut l lorsque l'on édite l'attribut k.

Pour quantifier la conservation d'identité au cours d'une édition, nous utilisons une librairie [4] qui consiste à encoder une image de visage dans les couches cachées d'un réseau de reconnaissance faciale H. La ressemblance, au sens de l'identité, est alors déterminée comme étant la similarité cosinus entre ces encodages :

$$D_{id}^{k} = \frac{1}{N}\sum_{i=1}^{N}\frac{< H(w_i^{k-})|H(w_i^{k+}) >}{||H(w_i^{k-})||.||H(w_i^{k+})||}. \quad (8)$$

Identité entre images d'origines et images éditées		
attribut édité	notre méthode	InterfaceGAN
Lunettes	**0.947**	0.939
Genre	0.930	**0.944**
Sourire	0.967	**0.973**
Âge	0.895	**0.906**

Proportion d'images correctement éditées		
attribut édité	notre méthode	InterfaceGAN
Lunettes	**0.999**	0.990
Genre	0.970	**0.978**
Sourire	**0.999**	**0.999**
Âge	0.950	**0.968**

5 Conclusion

Nous avons introduit une méthode permettant d'éditer des images naturelles en fonction d'attributs depuis l'espace latent de StyleGAN. Contrairement à InterfaceGAN, notre méthode permet l'édition de plusieurs attributs à la fois, et ce, de manière désenchevêtrée. De plus, la préservation d'identité et la stabilité de l'édition restent comparables à celles de l'approche classique InterfaceGAN. Enfin, notre réseau permet l'échantillonnage conditionnel d'images depuis l'espace latent $\mathcal{W}$.

Références

[1] Rameen ABDAL, Yipeng QIN et Peter WONKA : Image2stylegan : How to embed images into the stylegan latent space ? *CoRR*, abs/1904.03189, 2019.

[2] Rameen ABDAL, Peihao ZHU, Niloy J. MITRA et Peter WONKA : Styleflow : Attribute-conditioned exploration of stylegan-generated images using conditional continuous normalizing flows. *CoRR*, abs/2008.02401, 2020.

[3] Andrew BROCK, Jeff DONAHUE et Karen SIMONYAN : Large scale GAN training for high fidelity natural image synthesis. *CoRR*, abs/1809.11096, 2018.

[4] Adam GEITGEY : face_recognition python package. `https://github.com/ageitgey/face_recognition`, 2021.

[5] Erik HÄRKÖNEN, Aaron HERTZMANN, Jaakko LEHTINEN et Sylvain PARIS : Ganspace : Discovering interpretable GAN controls. *CoRR*, abs/2004.02546, 2020.

[6] Martin HEUSEL, Hubert RAMSAUER, Thomas UNTERTHINER, Bernhard NESSLER, Günter KLAMBAUER et Sepp HOCHREITER : Gans trained by a two time-scale update rule converge to a nash equilibrium. *CoRR*, abs/1706.08500, 2017.

[7] Tero KARRAS, Samuli LAINE et Timo AILA : A style-based generator architecture for generative adversarial networks. *CoRR*, abs/1812.04948, 2018.

[8] Tero KARRAS, Samuli LAINE, Miika AITTALA, Janne HELLSTEN, Jaakko LEHTINEN et Timo AILA : Analyzing and improving the image quality of stylegan. *CoRR*, abs/1912.04958, 2019.

[9] Siavash KHODADADEH, Shabnam GHADAR, Saeid MOTIIAN, Wei-An LIN, Ladislau BÖLÖNI et Ratheesh KALAROT : Latent to latent : A learned mapper for identity preserving editing of multiple face attributes in stylegan-generated images. *In 2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 3677–3685, 2022.

[10] Ziwei LIU, Ping LUO, Xiaogang WANG et Xiaoou TANG : Deep learning face attributes in the wild. *In Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

[11] Yujun SHEN, Ceyuan YANG, Xiaoou TANG et Bolei ZHOU : Interfacegan : Interpreting the disentangled face representation learned by gans. *CoRR*, abs/2005.09635, 2020.

[12] Mingxing TAN et Quoc V. LE : Efficientnet : Rethinking model scaling for convolutional neural networks. *CoRR*, abs/1905.11946, 2019.

[13] Xu YAO, Alasdair NEWSON, Yann GOUSSEAU et Pierre HELLIER : A latent transformer for disentangled and identity-preserving face editing. *CoRR*, abs/2106.11895, 2021.