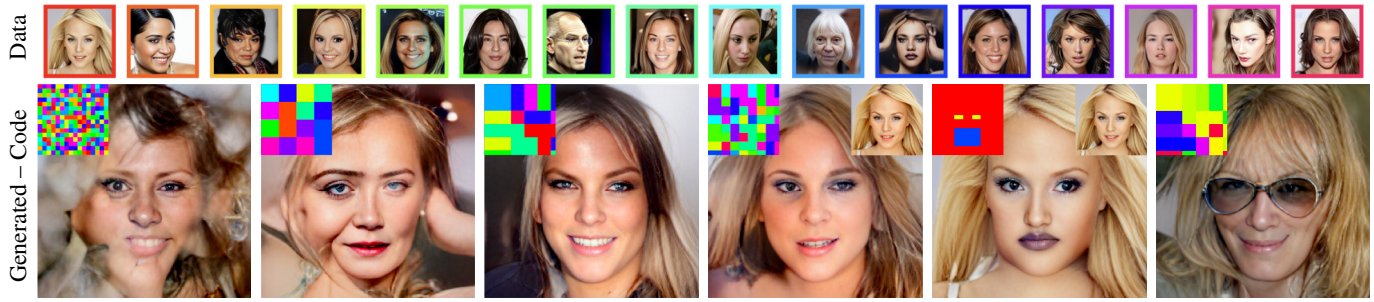
LATENTPATCH : Une approche non-paramétrique pour la génération et l'édition de visages

Benjamin SAMUTH Julien RABIN David TSCHUMPERLÉ Frédéric JURIE

Normandie Univ., UNICAEN, ENSICAEN, CNRS, GREYC, 14000 Caen, France



(a) Code aléatoire 1×1 Copie de patchs — (b) Code aléatoire 4×4 Copie de patchs — (c) LATENTPATCH 2×2 Séquence de patchs — (d) Par réference 1ère image, en rouge — (e) Édition de visage Yeux et bouche — (f) Par attributs Femme, blonde, lunettes

FIGURE 1 : Notre approche LATENTPATCH est capable de générer des images de visages comme (c) en n'utilisant que 16 images sources (première ligne), sans apprentissage additionnel. Elle permet également d'aborder facilement d'autres applications, comme la génération guidée par référence (d), l'édition de visage (e), et la génération contrainte par attributs (f) (pour ce dernier exemple, plus de 16 images sources ont été utilisées). En comparaison, les images (a), (b) ont été générées aléatoirement depuis les 16 sources. L'origine de chaque image est donné par son code couleur. Des résultats supplémentaires sont disponibles sur la page web [1].

Résumé – Ce papier introduit LATENTPATCH, une nouvelle méthode de génération d'images réalistes à l'aide de peu d'images. À la différence des approches dites de *finetuning* sur de larges modèles génératifs pré-entraînés, notre approche consiste en une affectation séquentielle des *patchs* d'exemples plongés dans un espace latent. Cette méthode est par nature géométriquement explicable et s'inspire des algorithmes non-paramétriques de synthèse de textures et de transfert de style. Elle s'assure que les caractéristiques des images générées correspondent à la distribution des images d'origine, s'affranchissant ainsi des limitations des modèles plus larges basés sur les *transformers*, les réseaux récurrents ou l'auto-attention, vis-à-vis des faibles quantités de données. Nous étendons les précédents modèles de génération utilisant une seule image, pour les faire fonctionner avec plusieurs, et démontrons que notre méthode peut générer des images réalistes, faire de l'édition et de la génération conditionnelle. Les expériences effectuées sur des bases de visages montrent que notre méthode s'avère efficace et générique.

Abstract – This paper presents LATENTPATCH, a new method for generating realistic images from a small dataset of only a few images. Unlike traditional few-shot generation methods that fine-tune pre-trained large-scale generative models, our approach is computed directly on the latent distribution by sequential feature matching, and is geometrically explainable by design. Avoiding large models based on transformers, recursive networks, or self-attention, which are not suitable for small datasets, our method is inspired by non-parametric texture synthesis and style transfer models, and ensures that generated image features are sampled from the source distribution. We extend previous single-image models to work with a few images and demonstrate that our method can generate realistic images, as well as enable conditional sampling and image editing. We conduct experiments on face datasets and show that our simplistic model is effective and versatile.

1 Introduction

Depuis la fin des années 2010, les réseaux génératifs profonds ont permis d'importants progrès dans la synthèse d'images photo-réalistes en combinant des modèles antagonistes (GAN) ou à diffusion avec un entraînement sur de grands jeux de données. À l'heure actuelle, les architectures les plus poussées de génération d'images à partir de texte, telles que la diffusion latente [2], comprennent plusieurs milliards de paramètres et nécessitent des bases de données d'un volume du même ordre à l'entraînement (*ex :* LAION-5B). Des approches récentes se sont focalisées sur l'entraînement de modèles génératifs à partir de petites bases de données, dont celles ne contenant que quelques images, voire une seule. C'est un problème difficile car pour générer des images réalistes et vraisemblables pour l'œil humain, notamment dans des domaines spécifiques (par exemple les visages), les modèles génératifs requièrent des réseaux de neurones larges et profonds (*e.g.* StyleGAN [3], VQ-VAE [4] ou VQ-GAN [5]). Par ailleurs, malgré le volume important de données apprises, il n'y a en pratique aucune garantie que ces modèles larges ne mémorisent pas une partie de leur base d'entraînement [6, 7]. Enfin, entraîner ces modèles nécessite l'utilisation de ressources de calcul très importantes (que ce soit en *RAM* ou en *jours-GPU*).

Pour prendre en compte les difficultés à entraîner des modèles larges sur de faibles jeux de données, différentes techniques ont été abordées dans la littérature. Une première approche de génération, dite *few-shot*, est basée sur la distillation de

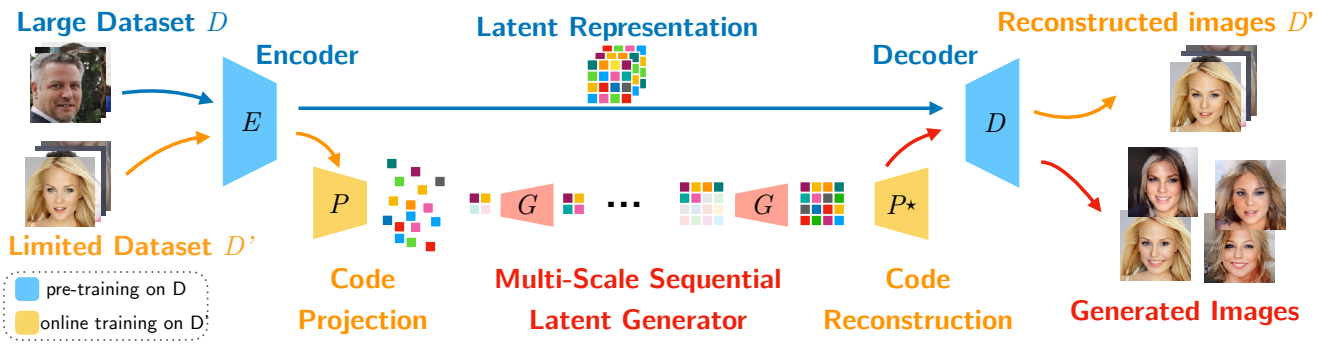


FIGURE 2 : Illustration de la méthode de génération de visages proposée, exploitant un auto-encoder pré-entraîné. Plus de détails en Section 2.

connaissances, ce qui se traduit par un affinement des poids d'un modèle large. Cette approche a été largement utilisée avec les GANs, comme FS-GAN [8, 9], mais également avec les modèles de diffusion [10].

Une autre approche consiste à utiliser des méthodes d'augmentation de données différentiables [11, 12, 13] pour entraîner des modèles *ex nihilo*. Ceci permet d'entraîner des GANs sur un jeu de données plus divers, réduisant le volume effectif à quelques milliers d'images. Cette quantité peut être encore réduite si les images de la base sont perceptuellement proches (par exemple, plusieurs images de la même personne).

La synthèse de textures et la re-génération d'images à partir d'un seul exemple constituent un cas extrême, pour lequel différentes techniques spécifiques d'apprentissage ont été proposées [14, 15]. Ces deux applications peuvent être vues comme un problème d'échantillonnage aléatoire qui peut être traité à l'aide de méthodes de comparaison de patchs comme proposé initialement dans [16]. Plus récemment encore, GPNN [17] et PSIN [18] ont démontré que des modèles génératifs larges ne sont pas toujours nécessaires pour faire de la synthèse aléatoire de qualité. Effectivement, des résultats similaires peuvent être obtenus à l'aide de variations de l'algorithme *PatchMatch* [19]. Ces techniques d'échantillonnage de patchs ont par ailleurs été étendues aux représentations latentes dans des applications plus récentes comme le transfert de style [20] et l'*inpainting* [21].

Notre méthode se positionne à l'intersection de ces trois formalismes : les modèles génératifs par copie de patchs, des entraînements à partir d'un petit jeu de données, et de l'adaptation de grands modèles pré-entraînés. Ce qui rend notre méthode efficace malgré le volume extrêmement limité de notre jeu de données est la combinaison de trois composantes clés. D'abord, nous nous basons sur un auto-encodeur pré-entraîné sur une grande collection d'images génériques. Son rôle est à la fois de fournir une représentation latente compacte des images traitées, et d'être capable de les décoder de manière la plus fidèle possible. Enfin, notre modèle génératif repose sur une synthèse par patchs qui ne requiert aucun entraînement additionnel, à la différence des modèles latents génératifs à haute-capacité (à l'image des *transformers* [5], des modèles auto-régressifs *PixelCNN* [22, 4, 23], ou encore des couches de *cross-attention* de [2]). Notre générateur non-paramétrique exploite des comparaisons multi-échelle de patchs latents afin de générer des séquences de codes latents plausibles à partir du jeu de données source.

2 Génération par patchs latents

L'approche LATENTPATCH que nous proposons permet la génération de nouvelles images depuis un jeu limité d'images sources. Elle consiste en trois étapes, comme illustré en Fig. 2. Tout d'abord, nous construisons un espace de plongement "universel" des images. Puis, nous adaptons cet espace latent pour qu'il corresponde à celui de nos images sources. Enfin, nous générons de nouvelles images dans cette représentation latente et nous la décodons. Chaque étape est détaillée ci-après.

Étape 1–Construire un espace de représentation "universel". L'objectif de cette étape est à la fois de plonger les images dans un espace générique à petite dimension et faible résolution spatiale, qui ne dépend pas spécifiquement des images sources, et de décoder fidèlement les représentations latentes correspondantes. Un réseau de neurones idéal pour réaliser cette tâche doit donc avoir suffisamment de capacité pour compresser une image naturelle sans introduire trop de distorsions, tout en veillant à éviter le phénomène de sur-apprentissage. Dans cet article, nous utilisons un auto-encodeur VQ-GAN [5], pré-entraîné sur une collection d'images génériques, notée $\mathcal{D}$. En pratique, l'encodeur produit des images quantifiée (selon un dictionnaire de 1024 codes) de résolution spatiale $M^2 = 16 \times 16$ avec $L = 256$ canaux. L'encodeur obtenu va permettre ultérieurement de simplifier l'estimation de la distribution et la comparaison des patchs des images sources.

Étape 2–Adapter la représentation des images sources. La distribution des images sources $\mathcal{D}'$ n'occupe qu'une fraction de l'espace de représentation, supposé universel. Afin d'adapter cette représentation pour rendre pertinente et rapide la comparaison des codes latents, nous projetons par P les images sources latentes dans un espace plus petit à l'aide d'une *ACP* (illustration en Fig. 2). Les vecteurs latents passent alors d'une dimension $L = 256$ vers une dimension réduite $r = 16$. Le choix de cette valeur ne révèle aucune dégradation notable quant à la qualité des reconstructions (Table 1).

Étape 3–Générer des images latentes dans l'espace de représentation. Notre méthode s'inspire à la fois de la synthèse de texture et de la génération d'images à partir d'un seul exemple. Elle vise à reproduire la distribution spatiale des patchs latents des images sources.

Plus précisément, le générateur, noté G, synthétise de manière séquentielle le vecteur latent $z(x)$, à la position $x \in \{0, \dots, M-1\}^2$, en échantillonnant de façon aléatoire les vecteurs de la distribution empirique latente, comme illustré en Fig. 2. Cette procédure simple et non-paramétrique ne requiert ainsi aucun entraînement additionnel, au contraire des

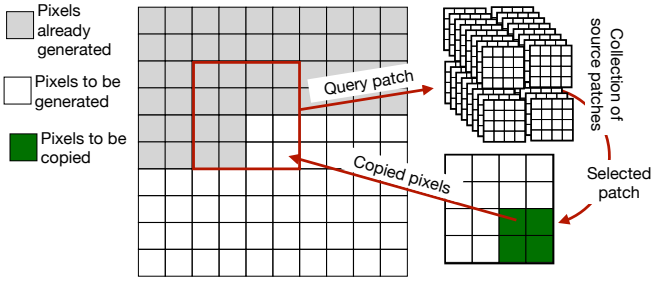


FIGURE 3 : Génération d'une image 10×10 à l'aide de patchs 4×4 et d'un pas de 2. Chaque échelle suit le même procédé. Le patch courant est masqué en fonction de ce qui a été généré précédemment.

approches usuelles de génération latentes [4, 5, 23, 2]. Comme pour ces autres approches, l'objectif reste ici de prédire, arbitrairement ligne par ligne (Fig. 3), la valeur des codes latents $z(x)$ en se basant sur l'observation des valeurs précédentes dans un voisinage restreint (patch).

Dans le but de générer des combinaisons plausibles de codes latents, le générateur échantillonne des patchs sources, de manière similaire aux méthodes de synthèses de textures par copie de patchs au plus proches voisins [16, 24]. Comme illustré en Fig. 3, l'originalité de notre approche est d'une part d'utiliser des patchs latents de taille $\omega \times \omega$ plutôt que directement dans l'espace couleur, et d'autre part de conditionner le tirage parmi la distribution des images latentes sources $\mathcal{D}'$ par la position spatiale x du patch synthétisé.

Nous adoptons une approche multi-échelle pour capturer les longues corrélations spatiales dans l'image générée. Le modèle est initialisé à la plus petite échelle ($s = 1$) en interpolant bilinéairement l'image de taille 16×16 donnée par l'encodeur E afin d'obtenir une résolution spatiale 10×10. Un patch aléatoire est placé dans le coin supérieur gauche. Les pixels des patchs n'ayant pas encore été générés dans la synthèse sont masqués à la comparaison entre le patch de synthèse et les patchs des images sources (*cf.* Fig. 3). Notons que ce masque n'est plus nécessaire aux résolutions supérieures, ou lorsqu'une image de référence est utilisé comme initialisation (*cf.* Fig. 1 (d)). Une fois achevée cette synthèse itérative, le résultat est interpolé (à nouveau de façon bilinéaire) à la résolution supérieure pour servir de référence à la synthèse à l'échelle suivante $s + 1$. Nous notons S le nombre total d'échelles utilisées par l'algorithme.

Générer des images variées avec une quantité limitée de données est un problème difficile, nécessitant de trouver un compromis entre diversité et fidélité des images générées quant à la distribution des images sources, et ce en évitant le sur-apprentissage. Une limitation à ce sujet des approches par échantillonnage de patch au plus proche voisin est de recopier de grandes régions de l'image d'exemple, ce que l'on souhaite justement éviter. Pour assurer un tel compromis, nous proposons deux variations de cette technique. La prise en compte des positions dans les caractéristiques d'une image est un aspect clé de l'entraînement des modèles génératifs (*e.g.* [25]). En supposant par simplicité les données recalées, nous piochons les patchs $p(x)$ à la même position x dans les images d'exemples afin de restreindre et accélérer la recherche des patchs plus proche voisins. Pour assurer la diversité des synthèses, nous tirons uniformément $p(x)$ parmi les $k > 1$ plus proches voisins, plutôt que le patch le plus proche ($k = 1$). Combiné avec un pas d'échantillonnage spatial w ($w < \omega$), cette approche

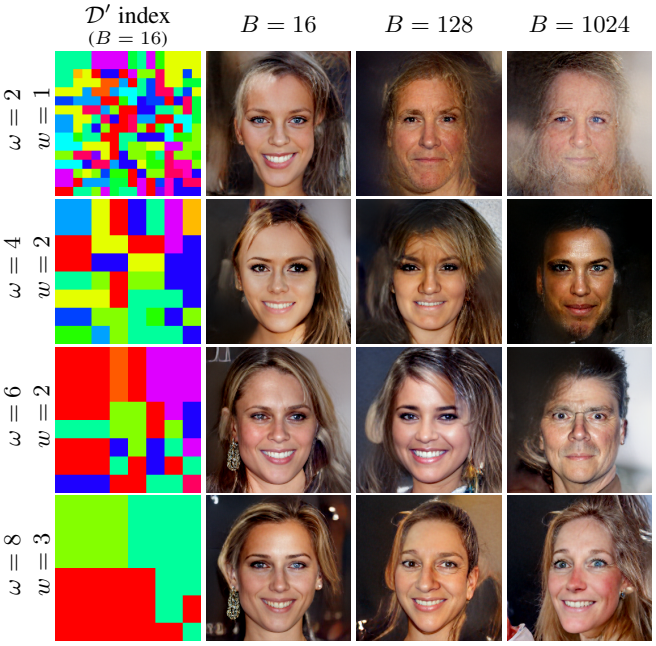


FIGURE 4 : Comparaison d'images générée avec différentes tailles de patchs (ω) et de pas (w). La cohérence des synthèses augmente avec le pas et la taille des patchs, mais diminue la diversité. La première colonne montre le code couleur des patchs, en valeur de teinte, dans le cas $B = 16$.

évite la réplication exacte d'un élément du jeu de données, les copies étant restreintes localement (voir les codes en Fig. 1). Ce principe est similaire aux paramètres de températures et de *top-k sampling* dans des modèles auto-régressifs comme [5].

3 Expériences

Contexte expérimental. Pour chaque expérience, nous utilisons un auto-encodeur basé sur VQ-GAN [5], entraîné sur la base d'images de visages FFHQ [3], faisant office de jeu de données universelles $\mathcal{D}$. L'auto-encodeur et le dictionnaire possèdent 72M paramètres. La partie générative de la méthode de [5] (800M de paramètres) n'est utilisée que pour les comparaisons. Notons que, comme dans [5], les codes z de la synthèse sont quantifiés au plus proche voisin en utilisant le dictionnaire. Dans chacune des expériences, le paramètre d'échantillonnage est fixé à $k = 3$. À l'exception des calculs de scores et des cas particuliers de la génération par référence et de l'édition, les images sont synthétisées de manière multi-échelle, allant des résolutions 10×10 à 16×16 sur $S = 5$ échelles.
Toutes les expériences reposent sur différents jeux d'images sources $\mathcal{D}'$ aléatoirement tirées parmi CelebA-HQ [26], à une résolution de 256×256 pixels. Afin de s'assurer que l'auto-encodeur de [5] ne souffre pas de sur-apprentissage (génération de données de $\mathcal{D}$ à partir de $\mathcal{D}'$), nous utilisons les techniques de détection de mémorisation de [6] (écart non significatif entre erreurs de reconstruction des images de $\mathcal{D}'$ et $\mathcal{D}$). À l'aide de l'*ACP*, la représentation latente originale à $L = 256$ dimensions est réduite à $r = 16$ dimensions. Nous précalculons l'*ACP* sur l'ensemble des données encodées $E(\mathcal{D}')$.

Génération aléatoire de visages. Les résultats de la figure 4 sont obtenus par notre méthode en choisissant $B = |\mathcal{D}'|$ images de CelebA-HQ, et en variant la taille des patchs ω (avec un pas fixé de $w = \lfloor \frac{1}{3}\omega + \frac{1}{2} \rfloor$). Le temps de calcul est

	Espace Latent	Méthode	FID↓	LPIPS↑
AE	VQ-GAN$_{\mathcal{D}}$	(Reconstruction)	8.9	1.03
	VQ-GAN$_{\mathcal{D}}$+ACP	(Reconstruction)	8.9	1.03
Génération	VQ-GAN$_{\mathcal{D}'}$	Transformers$_{\mathcal{D}'}$ [5]	10.2	1.03
	VQ-GAN$_{\mathcal{D}'}$+ACP	LatentPatch$_{\mathcal{D}'}$	31.6	0.80
	VQ-GAN$_{\mathcal{D}}$+ACP	LatentPatch$_{\mathcal{D}'}$	35.1	0.84
	VQ-GAN$_{\mathcal{D}}$+ACP	Aléatoire	123.0	0.85

TABLE 1 : *FID* et *LPIPS* moyens par paire, selon les données d'entraînement $\mathcal{D}$ (FFHQ) pour l'auto-encodeur seul (AE), ou selon les données sources $\mathcal{D}'$=CelebA-HQ pour différents modèles génératifs. Une valeur de FID faible indique une qualité proche de $\mathcal{D}'$. Plus la valeur de score *LPIPS* est faible, moins les échantillons générés seront variés par rapport à $\mathcal{D}'$. Notons que LatentPatch$_{\mathcal{D}'}$ génère des images à l'aide de l'espace latent appris sur $\mathcal{D}$ ou $\mathcal{D}'$, mais n'utilise que des images de $\mathcal{D}'$ comme source.

relativement faible pour un tel algorithme séquentiel. Après avoir chargé le modèle, 16 images sont générées en 1 seconde pour $B = 16$. Comme prévu, il y a un compromis entre fidélité et diversité, où la diversité est promue par l'augmentation de B et k. En effet, ceci permet au modèle de piocher des patchs perceptiblement proches depuis $\mathcal{D}'$. Décroître ω (et w) aide à générer plus de variations locales, aux dépends de la fidélité. La copie de plus grands patchs d'images existantes implique en effet qu'elles sont localement déjà cohérentes. Enfin, le paramètre S permet simplement à notre méthode de produire des images cohérentes, puisqu'un patch $\omega \times \omega$ capture une portion plus large de l'image à des échelles inférieures.

Mesure de qualité et de diversité. Nous évaluons la qualité et la diversité des images générées en calculant les scores de *FID*, ainsi qu'une différence moyenne entre les distances perceptuelles *LPIPS* [27], respectivement. Nous utilisons 10K images pour le score de *FID*, et 700 pour le score *LPIPS*. Ce dernier est normalisé en divisant par le score *LPIPS* des images sources $\mathcal{D}'$. Les scores sont affichés dans la Table 1. Les deux premières lignes correspondent à l'auto-encodeur seul, et montrent que l'*ACP* n'a pas d'impact sur la qualité de la reconstruction. Les lignes suivantes comparent notre méthodes avec [5], qui utilise un *transformer* entraîné sur des grandes collections d'images. Notons que le score de diversité n'est qu'un moyen de mesurer si les échantillons sont bien perceptuellement variés par rapport à $\mathcal{D}'$, et n'attestent aucunement de leur qualité visuelle. Entre autres, nous nous assurons que le processus de sélection et de copie de patchs ne va pas générer au global des visages trop similaires. Pour ces expériences, les paramètres sont $\omega = 6$ ($w = 2$), $S = 1$ et $k = 10$, de manière à éviter l'impact des doublons dans le jeu de donnée sans trop impacter le temps de calcul. Pour une comparaison équitable, LatentPatch utilise l'intégralité des images CelebA-HQ, tel que $B = |\mathcal{D}'| = 30k$.

Génération par patchs aléatoires. La Fig. 1 (a) & (b) montre que la copie purement aléatoire de codes $z(x)$ ou de patchs $p(x)$ ne produit pas de visages réalistes, et ce malgré l'alignement des valeurs sources par rapport à x. En d'autres termes, l'auto-encodeur n'est pas un projecteur qui générerait des images à partir de n'importe quel arrangement de codes latents. Comme prévu, la distribution implicite est ici toujours pertinente, justifiant ainsi la pertinence de notre méthode LatentPatch de génération d'images dans l'espace latent.

Remerciements. Ce travail est soutenu par l'Agence Nationale de Recherche via le projet ANR-19-CHIA-0017.

Références

[1] Benjamin Samuth. *Project Web page*. https://samuth211.users.greyc.fr/2023/NoParamGen/. 2023.

[2] Robin Rombach et al. "High-resolution image synthesis with latent diffusion models". In : *CVPR*. 2022.

[3] Tero Karras, Samuli Laine et Timo Aila. "A style-based generator architecture for generative adversarial networks". In : *CVPR*. 2019.

[4] Ali Razavi, Aaron Van den Oord et Oriol Vinyals. "Generating diverse high-fidelity images with vq-vae-2". In : *NeurIPS* 32 (2019).

[5] Patrick Esser, Robin Rombach et Bjorn Ommer. "Taming transformers for high-resolution image synthesis". In : *CVPR*. 2021.

[6] Ryan Webster et al. "Detecting overfitting of deep generative networks via latent recovery". In : *CVPR*. 2019.

[7] Ryan Webster et al. "This person (probably) exists. identity membership attacks against gan generated faces". In : *arXiv preprint arXiv :2107.06018* (2021).

[8] Esther Robb et al. "Few-shot adaptation of generative adversarial networks". In : *arXiv preprint arXiv :2010.11943* (2020).

[9] Yunqing Zhao et al. "A closer look at few-shot image generation". In : *CVPR*. 2022.

[10] Jingyuan Zhu et al. "Few-shot Image Generation with Diffusion Models". In : *arXiv preprint arXiv :2211.03264* (2022).

[11] Tero Karras et al. "Training generative adversarial networks with limited data". In : *NeurIPS* (2020).

[12] Shengyu Zhao et al. "Differentiable augmentation for data-efficient gan training". In : *NeurIPS* (2020).

[13] Yong Zhong et al. "Deep Generative Modeling on Limited Data with Regularization by Nontransferable Pre-trained Models". In : *arXiv preprint arXiv :2208.14133* (2022).

[14] Tamar Rott Shaham, Tali Dekel et Tomer Michaeli. "Singan : Learning a generative model from a single natural image". In : *ICCV*. 2019.

[15] Antoine Houdard et al. "A generative model for texture synthesis based on optimal transport between feature distributions". In : *Journal of Mathematical Imaging and Vision* (2022), p. 1-25.

[16] Alexei A Efros et Thomas K Leung. "Texture synthesis by non-parametric sampling". In : *ICCV*. 1999.

[17] Niv Granot et al. "Drop the gan : In defense of patches nearest neighbors as single image generative models". In : *CVPR*. 2022.

[18] Nicolas Cherel et al. "A Patch-Based Algorithm for Diverse and High Fidelity Single Image Generation". In : *ICIP*. 2022.

[19] Connelly Barnes et al. "The generalized patchmatch correspondence algorithm". In : *ECCV*. 2010.

[20] Benjamin Samuth, David Tschumperlé et Julien Rabin. "A Patch-Based Approach for Artistic Style Transfer via Constrained Multi-Scale Image Matching". In : *ICIP*. 2022.

[21] Nicolas Cherel et al. "Patch-Based Stochastic Attention for Image Editing". In : *arXiv preprint arXiv :2202.03163* (2022).

[22] Aaron Van den Oord et al. "Conditional image generation with pixelcnn decoders". In : *NeurIPS* (2016).

[23] Lior Ben-Moshe, Sagie Benaim et Lior Wolf. "FewGAN : Generating from the Joint Distribution of a Few Images". In : *ICIP*. 2022.

[24] Michael Ashikhmin. "Synthesizing natural textures". In : *Proceedings of the 2001 symposium on Interactive 3D graphics*. 2001, p. 217-226.

[25] Chieh Hubert Lin et al. "InfinityGAN : Towards Infinite-Pixel Image Synthesis". In : *ICLR*. 2022.

[26] Ziwei Liu et al. "Deep Learning Face Attributes in the Wild". In : *ICCV*. 2015.

[27] Richard Zhang et al. "The unreasonable effectiveness of deep features as a perceptual metric". In : *CVPR*. 2018.