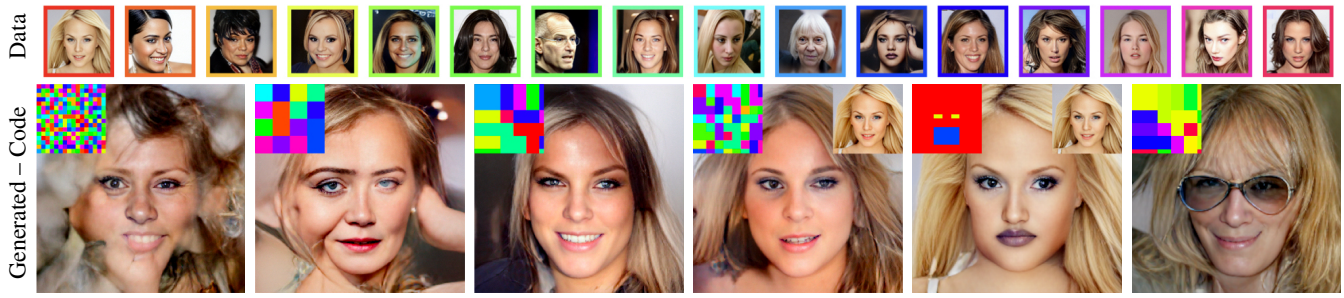
LATENTPATCH: A NON-PARAMETRIC APPROACH FOR FACE GENERATION AND EDITING

Benjamin Samuth *Julien Rabin* *David Tschumperlé* *Frédéric Jurie*

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

{Benjamin.Samuth, Julien.Rabin, David.Tschumperle, Frederic.Jurie}@unicaen.fr



(a) Random Code 1×1 Patch copy | (b) Random Code 4×4 Patch copy | (c) LATENTPATCH 2×2 Patch sequence | (d) Using reference First image, in red | (e) Face edition Eyes and mouth | (f) Selected attributes Woman, Blond, Glasses

Fig. 1: The proposed patch-based approach, coined "LATENTPATCH", can generate images like (c) using only the 16 source images shown in the first row, without any learning. It also enables easy implementation of variants such as (d) reference-based generation, (e) editing, and (f) attribute-constrained generation (on alternative data not shown here). Images (a) and (b) were generated using random patches from the source images, with the origin of each patch indicated by its color. Additional results can be found on the project page [1].

ABSTRACT

This paper presents LatentPatch, a new method for generating realistic images from a small dataset of only a few images. We use a lightweight model with only a few thousand parameters. 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 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.

Index Terms— Face Generation; Generative model; Auto-encoder; Latent representation; Image edition.

1. INTRODUCTION

Deep generative networks have made significant progress in photo-realistic image synthesis by using adversarial or diffusion models and training on large-scale datasets. Currently, the most advanced text-to-image generation architectures, such as latent diffusion [2], are comprised of billions of trainable parameters, requiring datasets of similar magnitude for training (*e.g.* LAION-5B). Recent efforts have focused on training generative models from small datasets, including those composed of just a few or even a single image. However, to produce realistic samples that are indistinguishable from true examples by human evaluation, even within a more specific image domain, such as human faces [3] or indoor scenes, generative models require deep and wide neural networks (*e.g.* StyleGAN [3], VQ-VAE [4] or VQ-GAN [5]). In some cases, building large datasets is not possible, such as in medical imaging where data is scarce, and data augmentation techniques may not be applicable [6]. Moreover, even with a large dataset, there is no guarantee that a large capacity model will not memorize some of the training samples [7, 8, 5]. This issue has recently raised concerns about confidentiality, privacy, and copyright [9, 10]. Additionally, training large models requires a significant amount of computational resources (in terms of *RAM* and *GPU-days* [2]), which seems excessive for some applications.

Given the difficulty of training models on very small image sets, different techniques have been employed. One approach, known as *few-shot generation*, is based on knowledge distillation, which involves fine-tuning a large model to a small dataset. This approach has been widely used for GANs, as evidenced by FreezeD [11], TGAN [12], MineGAN [13], FS-

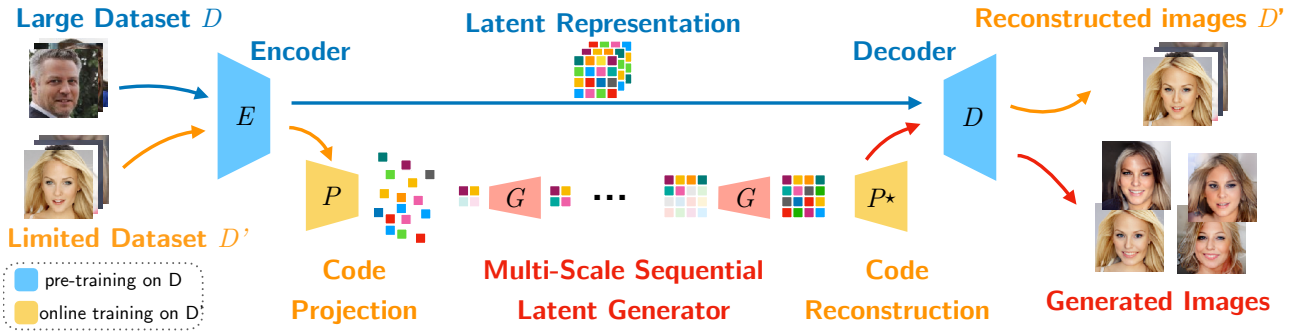


Fig. 2: Illustration of the proposed face generation framework based on a pre-trained auto-encoder. See the text for more details.

GAN [14], and [15]. It has also been successfully applied to diffusion models, including text-to-image models, such as in the case of [16].

An alternative approach is to use differentiable data augmentation methods [6, 17] to train the models from scratch. This allowed for the training of GANs on a more diverse dataset, which can be reduced to just a few thousand face images. Even fewer images may be used if they are perceptually similar, such as multiple images of the same person. As far as we know, FewGAN [18] is the only autoregressive generative model that has been trained on a very small dataset of landscapes.

One extreme example is the case of single image generation, as proposed by SinGAN [19]. Texture synthesis is a related proxy problem, for which various models have been proposed [19, 20, 21, 22]. These last two applications can be viewed as a type of image reshuffling, which can be achieved through patch-based sampling techniques, such as those described in [23]. Recently, GPNN [24] and PSIN [25] have demonstrated that generative models are not always necessary to synthesize high-quality random samples. This can be achieved using a variant of the Patch-Match algorithm [26] with GPU-based acceleration. Patch sampling has also been extended to latent representations in recent applications to image stylization [27] and inpainting [28].

Our method is positioned at the intersection of the three types of aforementioned approaches: generative methods based on patches, from-scratch training, and adaptation of pre-trained models. What makes our method effective even in the presence of an extremely limited number of images is that we mix three key ingredients. Firstly, we rely on an auto-encoder that is pre-trained on a large image dataset. We use this auto-encoder as a tool to produce a universal latent compact representation (the encoder) capable of decoding such representation into images (the decoder). Secondly, we compress the manifold described by the source image set simply through principal component analysis (PCA), in order to improve the computation time and the memory usage of our method. Finally, our generative model is a simple patch-based model that does not require any training, in contrast with high-capacity latent generative models (such as transformers in [5], auto-regressive pixelCNN [29] in [4, 18], or cross-attention layers in [2]). Our model resorts to multi-scale latent patch combinations to generate plausible latent code sequences from the source dataset, which is a non-parametric approach of the generative part, as opposed to finetuning or adapting larger models as done by other methods like [11, 12, 13].

2. A LATENT PATCH GENERATIVE MODEL

Our proposed approach, which we refer to as LATENTPATCH, enables the generation of novel images from a limited set of source images, and it consists of three steps, illustrated in Fig. 2. Firstly, we construct a "universal" latent space. Secondly, we adapt this latent space to the source images. Finally, we generate new images in latent space and then decode them. We describe each of these steps in detail below.

Step 1. Construction of a "universal" representation space. The objective of this step is to project the images into a generic, low-dimensional space with low spatial resolution, which does not depend on the source images. This makes it easier to capture the patch distributions of the source images. In our work, we use an off-the-shelf auto-encoder from VQ-GAN [5], which has already been trained on a large generic dataset $\mathcal{D}$. An ideal network for this task should have sufficient capacity to compress any natural image at a low distortion rate, without significant overfitting between the reconstruction of test and train images. In practice, the encoder produces quantized images of size M^2 =16×16 with $L = 256$ channels.

Step 2. Adapting the representation to the source images. The manifold of source images $\mathcal{D}'$ only covers a small part of the representation space, which is meant to be universal. In order to significantly speed up our method, particularly the nearest neighbor patch search, we use a standard dimension reduction technique. In practice, we resort to a PCA whose parameters are precomputed over $\mathcal{D}'$ (corresponding to P and $P^\star$ in Fig. 2). In this case, the dimensions are reduced from $L = 256$ to $r = 16$. This reduction has no noticeable impact regarding the quality of the reconstructions (*cf.* Table 1), while making the patch search almost 16 times faster.

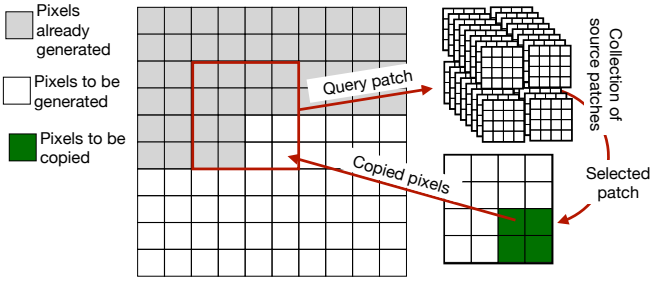


Fig. 3: Generation a 10×10 image using patches of size 4×4 and 2×2 strides, following the same process at each scale. The query patches are masked to exclude the not-yet-generated pixels.

Step 3. Generating images in the specialized representation space. As previously mentioned, our method is inspired by both texture synthesis and single image generative models, and focuses on the patch distributions of the source images. Unlike most existing approaches, we consider the joint latent patch distribution of the collection of source images.

More precisely, the generator, denoted as G, sequentially synthesizes latent codes $z(x)$, at location $x \in \{0, ..M-1\}^2$, by sampling random codes from the empirical latent distribution, as illustrated in Fig. 2. This non-parametric procedure does not require any training, contrary to other techniques based on transformers [5], auto-regressive synthesis [4, 18] or cross-attention [2]. Likewise, it aims at predicting the value of a latent code $z(x)$ based on the observation of previously predicted values, in a (arbitrarily) raster-scan order.

In order to generate plausible combinations of latent codes, the proposed generator is based on patch sampling, similarly to nearest-neighbor patch-based texture synthesis [23, 30]. As shown in Fig. 3, we consider here the latent $\omega \times \omega$ patch distribution of the source images $\mathcal{D}'$, rather than RGB patches. Let a $\omega \times \omega$ patch at location x be defined as a $\omega \times \omega \times L$ tensor defined as $p(x)[i,j] = [z(x-(i,j))]_{(i,j)\in\{0,\omega-1\}^2}$. To improve the quality of the synthesis and increase the likelihood of the generated sequence, we make use of a stride w when sampling query patches $p(x)$, as advocated in [31] for instance. This means that only a small portion (of size $w \times w$) of the retrieved patch is actually copied, as the query is shifted by w (from left to right).

We adopt a multi-scale approach to impose long-range spatial correlations in the generated images. The generative model is initialized at the coarsest scale ($s = 1$) by interpolating the 16×16 image given by the encoder E to obtain a 10×10 spatial resolution. A random patch is then placed in the top-left corner. Query patches are masked at the coarsest scale to discard pixels that have not yet been generated (see Fig. 3). It is important to note that this mask is not necessary at larger scales or when using a reference image for initialization (as shown in Fig. 1 (d)), as all the pixels are available. Once the synthesized image is upsampled to the next scale, it serves as a reference for generating the next level of detail, and this process continues until the desired 16×16 resolution is achieved at the finest resolution. The number of scale S is a parameter of the algorithm. Upsampling is performed using interpolation from the previous scale (see Fig. 3).

We still have to specify how the patches of the source images are chosen for a given patch query. Generating diverse images with limited data is a significant challenge, as it involves balancing the diversity of generated images with fidelity to the distribution of source images without overfitting. Positional embedding of the query features is a key aspect of training large generative models (see *e.g.* [32, 5]). In this work, we sample patches $p(x)$ from the same location x in example images to restrict the set of nearest-neighbor patches and speed-up the search. To ensure diversity, we uniformly sample the retrieved patch $p(x)$ from the $k > 1$ nearest-neighbors, rather than copying the closest match. This approach, combined with a small stride w, prevents exact replication of the training data, as sampling neighboring patches from the same example image can be avoided by setting $k > 1$. This acts similarly as the temperature and top-k sampling parameters in likelihood-based models such as [5].

3. EXPERIMENTS

Experimental Settings. For all experiments, we use an auto-encoder based on VQ-GAN [5], trained on the FFHQ [3] human face dataset, which contains a large number of high-quality images. FFHQ serves as the universal dataset $\mathcal{D}'$ for our experiments. The auto-encoder and quantizer have 72M parameters. The generative part of the method of [5], is only used for comparisons. Note that, as in [5], synthesized codes z are encoded using the decompressed codebook before feeding the decoder. In all experiments, the sampling parameter is fixed to $k = 3$, starting from 10×10 images, up to 16×16 over $S = 5$ scales, except for referenced generation and edition which only require a single scale.

We conduct experiments using various sets of source images $\mathcal{D}'$, the images being randomly sampled from CelebA-HQ [33], with a resolution of 256 pixels. To verify that the auto-encoder of [5] is not overfitting, we use memorization detection techniques from [7] and compare reconstruction errors of the source samples in $\mathcal{D}'$ with those from $\mathcal{D}$. Using the PCA, the original M^2 =16×16 latent representation based on a $L = 256$ dimensional codebook of 1024 atoms is compressed to $r = 16$ dimensions. We precompute the PCA over the latent features from the encoded source dataset $E(\mathcal{D}')$.

Baseline. Fig. 1 (a & b) shows that randomly copying codes $z(x)$ or $p(x)$ patches, even when sampling the same location x than in example images, does not provide realistic faces. This shows that randomly sampling latent codes or patches alone is not sufficient and motivates the use of more sophisticated samplers such as the one proposed.

Random Face Generation. The results shown in Fig. 4 are obtained with our method by randomly sampling $B = |\mathcal{D}'|$

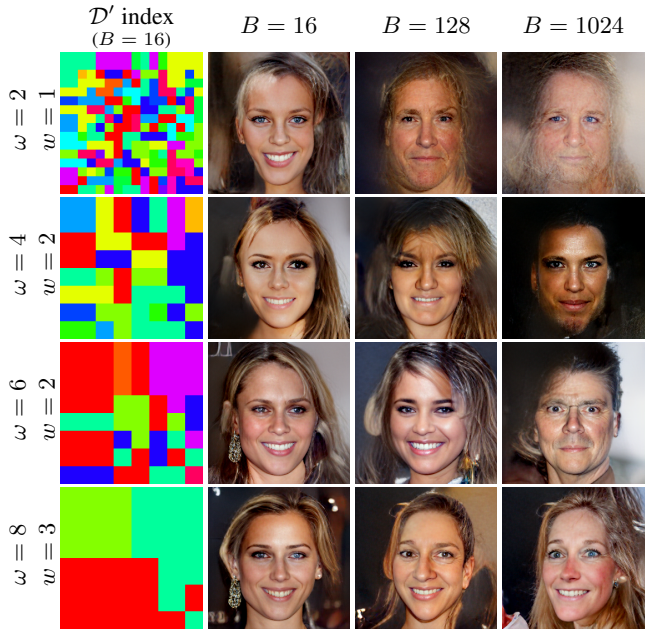


Fig. 4: Comparison of generated images with different patch sizes (ω), strides (w) and data size (B). Coherence of the generated images improves with increasing patch size and stride, but larger example regions also result in reduced diversity. The first column shows the patch index, encoded in normalized hue values, for $B = 16$.

images from CelebA-HQ, and generating images with various patch sizes ω (with a fixed stride of $w = \lfloor \frac{1}{3}\omega + \frac{1}{2} \rfloor$). Computation time is fairly low for a sequential algorithm: after loading the model, a batch of 16 images is generated in 1 second for $B = 16$. As expected, there is a trade-off between fidelity and diversity, where diversity is promoted by increasing B and k. Indeed, doing so will allow the generative algorithm to pick more perceptually fitting patches from $\mathcal{D}'$. Decreasing both w and ω will help generating more local variations, at the price of fidelity. Copying bigger patches taken from existing images implies that these are already locally coherent. Note that the multi-scale scheme, which ensures coherent image generation, relies on latent image interpolation. This may create noticeable artifacts (first row of Fig. 4), due to the smoothing in the latent space.

Quality and diversity assessment. We evaluated the quality of the generated images by computing the FID score on 10k images. We introduce a new normalized score for diversity. It computes the ratio between the average perceptual distances between images, from respectively generated and source data. In practice, we compute the average LPIPS [34] distances for 700 image pairs. The FID and diversity scores are displayed in Table 1. The first two rows correspond to the auto-encoder alone, and show that the PCA has no significant impact on the quality of the reconstructed images. The next rows compare our method with [5], which uses a transformer trained on large datasets, and we found that our LATENTPATCH model achieves competitive results without the training of an entire generator, and with only a few source images. Note that our diversity metric does not assess the visual quality of the generations, but only how close it is relative to CelebA-HQ. In these experiments, $w = 6$, $\omega = 2$, $S = 1$ and $k = 10$. For a fair comparison with [5], LATENTPATCH is able to pick patches from the entire dataset, thus $B = |\mathrm{C}| = 30\mathrm{k}$.

	Latent space	Method	FID↓	Diversity↑
AE	VQ-GAN$_\mathrm{F}$	(Reconstruction)	8.9	1.03
	VQ-GAN$_\mathrm{F}$	PCA	8.9	1.03
Generation	VQ-GAN$_\mathrm{C}$	Transformers$_\mathrm{C}$ [5]	10.2	1.03
	VQ-GAN$_\mathrm{C}$	LATENTPATCH$_\mathrm{C}$	31.6	0.80
	VQ-GAN$_\mathrm{F}$	LATENTPATCH$_\mathrm{C}$	35.1	0.84
	VQ-GAN$_\mathrm{F}$	Random sampling	123.0	0.85

Table 1: FID and diversity scores, both relative to $\mathcal{D}'$ = CelebA-HQ, for the auto-encoder alone and as well as for various generative models. For the FID, lower is better. For our diversity score, higher is better. Note that LATENTPATCH$_\mathrm{C}$ generates images using the latent space of the auto-encoder, either trained on F (FFHQ) or C (CelebA-HQ), but still use images from C as sources.

Conditional image generation. Our method allows for conditional generation of images, by conditioning on an input image which constitutes the first scale of the generation process. This means that our approach can regenerate faces that are similar to a reference image, using patches from the source images, as shown in Fig. 1 (d). Furthermore, by simply choosing random images with desired attributes (*e.g.* from CelebA) in $\mathcal{D}'$, the generated images can exhibit the desired features, as illustrated in Fig. 1 (f). A perceptually homogeneous $\mathcal{D}'$ helps the generative model produce coherent outputs without requiring as many examples. However, not all attributes imply proximity between the images of the dataset (*e.g.* "hats"). In such cases, it is preferable to directly edit the image in the latent space of VQ-GAN, as in Fig. 1 (e), which is simple to do by copying the spatial vectors of the desired attribute over a reference image. The auto-encoder is powerful enough to blend the images together convincingly.

4. CONCLUSION AND FUTURE WORK

This work proposes LATENTPATCH, a simple non-parametric model for generating near photo-realistic face images from tiny datasets, using a coarse-to-fine patch sampling approach. The model has several advantages, including not requiring the training of a generator and being versatile enough for related tasks like image editing and conditional generation. Future work will explore the possibility of generation from non-registered images, and the use of generic and lightweight auto-encoder.

Acknowledgment. This work is partially supported by the project ANR-19-CHIA-0017.

5. REFERENCES

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

[2] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer, "High-resolution image synthesis with latent diffusion models," in *CVPR*, 2022.

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

[4] Ali Razavi, Aaron Van den Oord, and Oriol Vinyals, "Generating diverse high-fidelity images with VQ-VAE-2," *NeurIPS*, vol. 32, 2019.

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

[6] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila, "Training generative adversarial networks with limited data," *NeurIPS*, 2020.

[7] Ryan Webster, Julien Rabin, Loic Simon, and Frédéric Jurie, "Detecting overfitting of deep generative networks via latent recovery," in *CVPR*, 2019.

[8] Ryan Webster, Julien Rabin, Loic Simon, and Frederic Jurie, "This person (probably) exists. identity membership attacks against GAN generated faces," *arXiv preprint arXiv:2107.06018*, 2021.

[9] G. Somepalli, V. Singla, M. Goldblum, J. Geiping, and T. Goldstein, "Diffusion art or digital forgery? investigating data replication in diffusion models," *arXiv preprint arXiv:2212.03860*, 2022.

[10] N. Carlini, J. Hayes, M. Nasr, M. Jagielski, V. Sehwag, F. Tramèr, B. Balle, D. Ippolito, and E. Wallace, "Extracting training data from diffusion models," *arXiv preprint arXiv:2301.13188*, 2023.

[11] Sangwoo Mo, Minsu Cho, and Jinwoo Shin, "Freeze the discriminator: a simple baseline for fine-tuning GANs," *arXiv preprint arXiv:2002.10964*, 2020.

[12] Yaxing Wang, Chenshen Wu, Luis Herranz, Joost Van de Weijer, Abel Gonzalez-Garcia, and Bogdan Raducanu, "Transferring GANs: generating images from limited data," in *ECCV*, 2018.

[13] Yaxing Wang, Abel Gonzalez-Garcia, David Berga, Luis Herranz, Fahad Shahbaz Khan, and Joost van de Weijer, "MineGAN: effective knowledge transfer from GANs to target domains with few images," in *CVPR*, 2020.

[14] Esther Robb, Wen-Sheng Chu, Abhishek Kumar, and Jia-Bin Huang, "Few-shot adaptation of generative adversarial networks," *arXiv preprint arXiv:2010.11943*, 2020.

[15] Yunqing Zhao, Henghui Ding, Houjing Huang, and Ngai-Man Cheung, "A closer look at few-shot image generation," in *CVPR*, 2022.

[16] Jingyuan Zhu, Huimin Ma, Jiansheng Chen, and Jian Yuan, "Few-shot image generation with diffusion models," *arXiv preprint arXiv:2211.03264*, 2022.

[17] Shengyu Zhao, Zhijian Liu, Ji Lin, Jun-Yan Zhu, and Song Han, "Differentiable augmentation for data-efficient GAN training," *NeurIPS*, 2020.

[18] Lior Ben-Moshe, Sagie Benaim, and Lior Wolf, "FewGAN: Generating from the joint distribution of a few images," in *ICIP*, 2022.

[19] Tamar Rott Shaham, Tali Dekel, and Tomer Michaeli, "SinGAN: Learning a generative model from a single natural image," in *ICCV*, 2019.

[20] Dmitry Ulyanov, Vadim Lebedev, Andrea Vedaldi, and Victor Lempitsky, "Texture networks: feed-forward synthesis of textures and stylized images," in *Proceedings of the 33rd International Conference on Machine Learning*, 2016, pp. 1349–1357.

[21] Arthur Leclaire and Julien Rabin, "A stochastic multi-layer algorithm for semi-discrete optimal transport with applications to texture synthesis and style transfer," *Journal of Mathematical Imaging and Vision*, vol. 63, pp. 282–308, 2021.

[22] Antoine Houdard, Arthur Leclaire, Nicolas Papadakis, and Julien Rabin, "A generative model for texture synthesis based on optimal transport between feature distributions," *Journal of Mathematical Imaging and Vision*, pp. 1–25, 2022.

[23] Alexei A Efros and Thomas K Leung, "Texture synthesis by non-parametric sampling," in *ICCV*, 1999.

[24] Niv Granot, Ben Feinstein, Assaf Shocher, Shai Bagon, and Michal Irani, "Drop the GAN: In defense of patches nearest neighbors as single image generative models," in *CVPR*, 2022.

[25] Nicolas Cherel, Andrés Almansa, Yann Gousseau, and Alasdair Newson, "A patch-based algorithm for diverse and high fidelity single image generation," in *ICIP*, 2022.

[26] Connelly Barnes, Eli Shechtman, Dan B Goldman, and Adam Finkelstein, "The generalized patchmatch correspondence algorithm," in *ECCV*, 2010.

[27] Benjamin Samuth, David Tschumperlé, and Julien Rabin, "A patch-based approach for artistic style transfer via constrained multi-scale image matching," in *ICIP*, 2022.

[28] Nicolas Cherel, Andrés Almansa, Yann Gousseau, and Alasdair Newson, "Patch-based stochastic attention for image editing," *arXiv preprint arXiv:2202.03163*, 2022.

[29] Aaron Van den Oord, Nal Kalchbrenner, Lasse Espeholt, Oriol Vinyals, Alex Graves, et al., "Conditional image generation with PixelCNN decoders," *NeurIPS*, 2016.

[30] Michael Ashikhmin, "Synthesizing natural textures," in *Proceedings of the 2001 symposium on Interactive 3D graphics*, 2001, pp. 217–226.

[31] Vivek Kwatra, Irfan Essa, Aaron Bobick, and Nipun Kwatra, "Texture optimization for example-based synthesis," in *SIGGRAPH*. 2005.

[32] Chieh Hubert Lin, Yen-Chi Cheng, Hsin-Ying Lee, Sergey Tulyakov, and Ming-Hsuan Yang, "InfinityGAN: Towards infinite-pixel image synthesis," in *ICLR*, 2022.

[33] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang, "Deep learning face attributes in the wild," in *ICCV*, 2015.

[34] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *CVPR*, 2018.