INFO8010: Reading assignement
Analyzing and Improving the Image Quality of StyleGAN

Pierre Navez[1] and **Antoine Debor**[2]
[1]*pierre.navez@student.uliege.be (s154062)*
[2]*antoine.debor@student.uliege.be (s173215)*

I. INTRODUCTION AND BACKGROUND

Presented for the first time in 2014 by Ian J. Goodfellow and al. [1], generative adversarial networks (GAN) have openend the door of a wide research area in image generation. StyleGANs, which have been introduced in 2019 by a research team from NVIDIA [2], a well-known company in PC gaming, allow the controllability of some style parameters in the image generation process. Their initial work is inspired by the style transfer paper [3] which presents a robust optimization process when training a network, thanks to an adaptative instance normalization layer (AdaIN). In 2020, NVDIA have published a new paper in which they present StyleGAN2, a revised version of StyleGAN, and whose main components are reviewed in the present work.

II. STYLEGAN - OVERVIEW

The particularity of StyleGANs is that the generator architecture contains multiple inputs and is composed of two networks. The first network is an MLP, which is responsible for learning a mapping $f : \mathcal{Z} \rightarrow \mathcal{W}$, between latent code in the latent space $\mathbf{z} \in \mathcal{Z}$ and an intermediate latent $\mathbf{w} \in \mathcal{W}$. The affine transforms $\mathbf{w}$ correspond to *styles* and are applied as input of each adaptative instance normalization (AdaIN) block of the synthesis network g, making the style controllable. AdaIN is responsible for aligning the mean and variance of the content features to the styles features. Additionnaly, stochastic variations in the content of the generated images are facilitated thanks to random noise input in the synthesis network g, before the normalization operation at each style block. The intensity of the noise allows more or less precision in the details.

III. STYLEGAN2

Although styleGANs have shown impressive results, when exceeding a certain resolution, the generated images were systematically containing artifacts which looked like a water drop on the image. StyleGAN2 overcomes this issue, and the overall architecture is also reviewed to enhance the general image quality.

A. Style blocks and AdaIN

AdaIN consists in a modulation step followed by a normalization step. Both are separated by a convolution and the three operations constitute a style block. Normalization is mandatory in order to preserve the controllability of the style. When doing *style-mixing* by applying different latent $\mathbf{w}$ to different style blocks, certain feature maps may be amplified by at least one order of magnitude. This effect has to be counteracted, otherwise the subsequent layers would not be efficient for processing the data correctly. The modulation step, on its side, allows to scale the incoming styles to every input feature maps of the convolution. This can be implemented by scaling the convolution weights of feature map i by a factor s_i. At the output of the convolution, the goal of instance normalization is to remove s from the statistics.

The authors identify this AdaIN operation as the origin of the blob-shaped artifacts: it normalizes the mean and the variance of every feature map separately, which causes potential loss of information learned about the magnitude of the features relative to each other past the AdaIN. As a result, the NVDIA team hypothetizes the fact that the generator produces a strong and localized spike, which results in the described artifact, in order to corrupt the signal strength information and to be free to scale the signal as it likes elsewhere. The generator architecture and the instance normalization are thus revisited.

By making the assumption that the input activations are i.i.d random variables with unit standard deviation, the expression of the standard deviation after the modulation and convolution is basically the square root of the sum of the modulated weights squared. Normalization, which aims at retrieving a unit standard deviation, can thus be achieved *before* the convolution, by dividing the scaled weights by the expected standard deviation after the convolution. This “demodulation” procedure not only reduces the entire style block to one single convolution layer, but allows to avoid the use of AdaIN and thus prevents the generated images from containing the localized water droplet artifact.

Another modification of the style block is that noise and bias are now applied outside of it, after the normalization, rather than right after the convolution and before the normalization. These modifications allow to obtain more predictable results.

B. Generator smoothness and regularization

Several metrics can evaluate the performance of a GAN's generator. In addition to FID (Fréchet Inception Distance) and Precision and Recall (of the discriminator's classification performance), perceptual path length (PPL) is used as well, after observing that the three first metrics were not sufficient to ensure good image quality. PPL quantifies the smoothness of a mapping between a latent space and a generated image. Even if a smaller PPL score seems to correlate with better image quality, encouraging a minimal PPL during the optimization phase is not acceptable because it would lead to degenerate solution with zero recall. The authors hence descibe a regularizer ensuring smoother generator mapping while bypassing this problem.

First, from a general point of view, it is shown that even if usually, during the training, the general loss function (e.g logistic loss) plus the regularization term are optimized simultaneously, optimizing the loss function and looking at the regularization term every 16 minibatches does not affect the final result and has the advantage of being less greedy in terms of computational resources and memory.

The expression of the regularizer is then formulated as follows

$$\mathbb{E}_{\mathbf{w},\mathbf{y}\sim\mathcal{N}(0,\mathbf{I})}\left(\left\|\mathbf{J}_{\mathbf{w}}^{T}\mathbf{y}\right\|_{2}-a\right)^{2} \tag{1}$$

with $\mathbf{J}_{\mathbf{w}}$ the Jacobian matrix of the mapping $g(\mathbf{w})$, $\mathbf{y}$ random images with normally distributed pixel intensities and a a constant that is set dynamically during optimization. This structure promotes $\mathbf{w}$ gradients of close to equal lengths regardless of $\mathbf{w}$ or the image-space direction, indicating that the above-mentioned mapping is well conditioned. An interesting result is that the identity $\mathbf{J}_{\mathbf{w}}^{T}\mathbf{y} = \nabla_{\mathbf{w}}(g(\mathbf{w})\cdot\mathbf{y})$ can be used and computed efficienly using backpropagation to avoid expensive computation of the Jacobian matrix.

C. Training method and modules structure

Progressive growing is a training method for GANs developped by the authors of styleGAN. The idea is to grow the generator along with the discriminator during the training by adding new layers progressively, hence increasing resolution in the image. It has the advantage of a faster training and shows efficiency in stabilizing high-resolution image synthesis.

However, progressive growing has its own characteristic artifacts, compromising shift invariance. The paper suggests that these are due to the fact that each resolution layer serves momentarily as the output layer, forcing it to generate maximal frequency details and thus leading to the final trained network having too high frequencies in the intermediate layers.

Three alternative architectures for the generator as well as for the discriminator are thus investigated in order to keep the mentioned benefits without the drawbacks. The first one is inspired by MSG-GAN which adds skip connections between the generator and the discriminator for matching resolutions. The second one is similar but the contribution of the RGB outputs are upsampled (downsampled for the discriminator) and summed up between the skip connections. Finally, the third one, inspired by LAPGAN, uses residual connections.

The second and third architectures are compared to the initial one in terms of FID and PPL. The six possible generator-discriminator pairs are evaluated and it turns out that the best results are obtained using a generator with skip connections and a discriminator with residual connections.

D. Projection of images to latent space

Inverting the synthesis network g, *i.e.* retrieving the original latent $\mathbf{w}$ from a given image, is a very important capability of styleGAN2 as it allows to determine whether a particular image is fake or not and, more precisely, to attribute a fake image to its specific source. Their method is detailed in an appendix but does not enter the scope of this summary.

The quality of their projection method is evaluated by measuring, for different datasets, the LPIPS distance between an original and re-synthetized image (after the orignal has been projected into the latent space). The results are compared to those obtained using the initial StyleGAN and are encouraging as they show that the generated images project way better into $\mathcal{W}$ when they are generated with StyleGAN2.

IV. IMPLICATIONS FOR THE FUTURE AND PERSONNAL CONCLUSIONS

More than a simple update of an existing architecture, this paper presents a state-of-the-art style transfer GAN. Although the authors already provided style controllability with StyleGAN, they review their architecture and training method to avoid unwanted by-products and achieve tremendous results. This work is a precious knowledge in the field of image generation. One can imagine such a technology to be used by the video game industry, which, for instance, might facilitate the work of graphic designers, or for the detection of fake images which might be used for detrimental purposes.

[1] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial net-

works, 2014.

[2] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. *CoRR*, abs/1812.04948, 2018.

[3] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization, 2017.