# REPRODUCING PA-GAN

**Daniel Fojo*, Paula Gómez*, Janna Escur* & Miquel Tubau***

ETSETB TelecomBCN
Universitat Politecnica de Catalunya (UPC)
Barcelona, Catalonia/Spain
daniel.fojo@estudiant.upc.edu,
{paula.maria.gomez,janna.escur,miquel.tubau}@alu-etsetb.upc.edu

**Xavier Giró-i-Nieto, Noé Casas**

Intelligent Data Science and Artificial Intelligence Center (IDEAI)
Universitat Politecnica de Catalunya (UPC)
Barcelona, Catalonia/Spain
xavier.giro@upc.edu, contact@noecasas.com

## ABSTRACT

One of the challenges in machine learning research is to ensure that published results are reliable and reproducible. This work aims to reproduce the results of the paper PA-GAN: Improving GAN Training by Progressive Augmentation (Zhang & Khoreva, 2019) submitted to the 2019 International Conference on Learning Representations (ICLR).

PA-GAN proposes a novel method to improve the performance of a Generative Adversarial Network (GAN) by hindering the task of the Discriminator to avoid its rapid convergence. In this report, we reformulate the contributions of the original paper and review it based on a fresh new implementation. Besides, we also demonstrate the reliability of their experiments by showing either the achieved results or the small issues that were found, which later were corrected by contacting the authors. We are happy to say that we could get Progressive Augmentation to increase a GAN's performance. However, we could not achieve the same exact value for the metric reported in the paper.

We also provide the first public implementation of Progressive Augmentation here https://github.com/telecombcn-dl/2018-dlai-team1.

## 1 INTRODUCTION

Reproducing the results of a research is often a challenge in machine learning. This report addresses this challenge for PA-GAN (Zhang & Khoreva, 2019), a paper submitted to the 2019 International Conference on Learning Representation. Our analysis explores whether the details of the original paper are enough to obtain their results, and also validates their conclusions. This report has been developed as a response to the ICLR 2019 Reproducibility Challenge, which aims at raising awareness of the problem and fosters the conversation between researchers.

For this challenge we chose to reproduce PA-GAN (Zhang & Khoreva, 2019), a paper submitted to ICLR 2019 which we explain in the following section 2. In section 3, we explain in detail our implementation of Progressive Augmentation. After that, in section 4, the results achieved are shown.

*Equal contribution

# 2 PA-GAN

GANs are a type of neural network in which two models are trained simultaneously: a generative model (generator) that transforms a sample form one probability distribution into another, and a discriminative model (discriminator) that estimates the probability that a sample comes from the training data rather than from the generator. The training procedure for the generator is to maximize the probability of the discriminator making a mistake. This adversarial process allows models to generate realistic predictions even when the data has very complicated distributions. (Goodfellow et al., 2014)

Our reproducibility analysis reviews PA-GAN (Zhang & Khoreva, 2019), a technique to improve Generative Adversarial Networks (GANs) training process and achieve a better overall performance. The paper states that, despite recent progress, GANs still suffer from training instability thus requiring careful consideration of the architecture design and hyper-parameter tuning. It is known that the reason for this fragile training behaviour is, partially, due to the discriminator learning faster than the generator, and breaking this way the balance between the two networks. As a consequence, its loss rapidly converges to zero, thus providing no reliable backpropagation signal to the generator (Zhang & Khoreva, 2019). In order to overcome this limitation, PAGAN increases stability during training. This is achieved by augmenting the difficulty of the task of the discriminator. REEXPLAIN OR REMOVE THIS SENTENCE: In particular, PA-GAN structurally augments both fake and real training samples and, then, it minimizes the divergence between the distributions defined in the augmented sample space. The minimized divergence is computed by using the adversary process introduced in Goodfellow et al. (2014).

## 2.1 SINGLE AUGMENTATION LEVEL

To understand the progressive nature of PA, we will first introduce a single augmentation level to the samples, and in 2.2 we will see how this can be extended to multiple augmentation level to each sample

In a GAN, the task of the discriminator consists on classifying real data samples ($x_r$) into the TRUE class and, generated samples ($x_g$), produced by the generator, into the FAKE class. When introducing Progressive Augmentation (PA) to these samples, they are augmented by a bit $s \in \{0, 1\}$ and a new classification can be done. The new $(x_r, s = 0)$ and $(x_g, s = 1)$ will correspond to an updated definition of the TRUE class while $(x_r, s = 1)$ and $(x_g, s = 0)$ will belong to the FAKE one. Thus, as it can be seen in Figure 1, real samples will not belong just to the TRUE class and, analogously, generated samples will not belong just to the FAKE one.

To determine the truth value of a sample, we will consider the real and synthetic samples to convey one bit of information, $x_r$ encoding a 0 and $x_g$ encoding a 1. Thus, the checksum (which corresponds to a simple XOR) of a pair $(x, s)$ determines the respective class, i.e. checksum zero for TRUE and one for FAKE. This new checksum, poses a more difficult task for the discriminator, since it has to classify both TRUE and FAKE samples while also computing the checksum operation. In (Zhang & Khoreva, 2019) it is stated that this prevents early maxing-out of the discriminator without compromising the task of the generator.

Table 1: Example of samples' class with a single augmentation level

| **Sample** | **s** | **Class** |
|---|---|---|
| $x_r$ | 0 | TRUE |
| $x_r$ | 1 | FAKE |
| $x_g$ | 0 | FAKE |
| $x_g$ | 1 | TRUE |

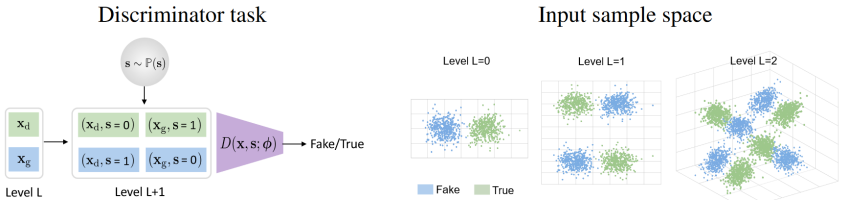



Figure 1: Visualization of progressive augmentation. Level $L = 0$ corresponds to a classical GAN, With each extra augmentation level ($L \rightarrow L + 1$) the dimensional of the discriminator input space is increased and the discrimination task gradually becomes harder. This strategy prevents the discriminator from easily finding a decision boundary between two classes and thus leads to meaningful gradients for the generator updates. Figure taken from (Zhang & Khoreva, 2019)

### 2.2 Progressive multi-level augmentation

Single level augmentation is extended by changing $s$ to be an arbitrarily long sequence of bits $\boldsymbol{s}$. Then, the augmented discriminator takes the bit sequence $\boldsymbol{s}$ as well as the sample $x$. Following the procedure from the single level case, the same checksum mechanism remains. Namely, the discriminator has to retrieve one bit of information carried by $x$, and perform the checksum operation with the arbitrarily long random sequence $\boldsymbol{s}$. The original paper claims that the difficulty of this task grows as the length of $\boldsymbol{s}$ grows. This arises the idea of increasing the length of $\boldsymbol{s}$ every time that the discriminator becomes too powerful. Moreover, it is the consistency of the checksum mechanism across different augmentation levels that allows progressive augmentation.

The same discriminator can be trained from a lower augmentation level and gradually take more bits into consideration (see Figure 2).

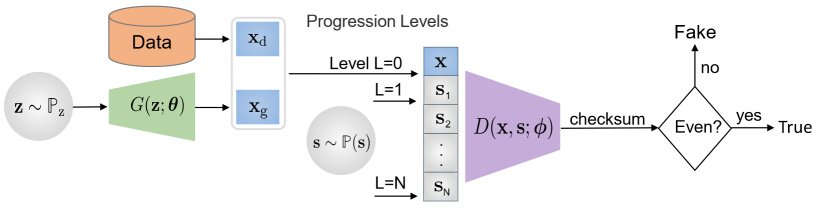



Figure 2: Overview of PA-GAN, and how the discriminator's task difficulty grows with the length of $\boldsymbol{s}$. Figure taken from (Zhang & Khoreva, 2019)

## 3 Implementation

For our implementation, we used a baseline a Spectral Normalized-DCGAN (Miyato et al., 2018), as it is used in the original paper. Then, in order to add Progressive Augmentation to the network, the generator can remain unchanged while the discriminator has to accommodate the increase of the samples with $\boldsymbol{s}$. To this end, we modify the first layer of the discriminator every time a level is increased. The only modification to this layer are the number of input channels. The kernel size, the padding and the stride of this layer, are all maintained.

The bit sequence $\boldsymbol{s}$ is pre-processed into a form compatible with $x$. For the case of images as an input, each bit of the sequence is translated to an image channel filled with either all 1s or all 0s.

To evaluate whether the discriminator is outperforming the generator, (Zhang & Khoreva, 2019) uses the Kernel Inception Distance (KID) (Bińkowski et al., 2018). This distance measures how good is a generated sample distribution with respect to the real one. If we observe that the KID value does not decrease in training, it is a sign that the network is not improving. We also implemented this distance in our code.

However, we struggled to understand the scheduling of the progressive augmentation from the PA-GAN text. particular, we did not follow the original instructions: "If the current KID score is

less than 5% of the average of the two previous ones attained at the same augmentation level, the augmentation is levelled up, i.e. $L$ is increased by one". Instead, we increased $L$ whenever the difference between the KID score and the average of the two last KID scores attained in the same augmentation level was smaller than the 5% of that average. After contacting the authors about this via OpenReview, they agreed that this was what they did, but they reported it incorrectly in the paper.

When the new level is attained, the probability distribution for the new bit of $\boldsymbol{s}$ is not uniform. We use a Bernoulli distribution with

$$p = \min\{0.5(t - t_{st})/t_r, 05\} \tag{1}$$

being the probability of 0. Here $t$ is the current iteration, $t_{st}$ is the last iteration in which $\boldsymbol{s}$ was augmented, and $t_r$ is a hyperparameter set to $10^5$ in the PA-GAN paper. The use of a non-uniform probability for 1s and 0s is justified in the paper saying that it helps with learning stability.

The PA-GAN paper also mentions that the learning rate of the generator could be decreased when a new level is attained. However, the text does not give details about how or where to implement this.Thus, we decided not to implement it but to contact the authors. Once they replied, they confirmed that it was not used by them neither.

Last but not least, in order to evaluate the performance of the GAN we used the Fréchet Inception Distance (FID) (Heusel et al., 2017) as they do in the original paper. We used the FID implementation from `https://github.com/mseitzer/pytorch-fid`.

Our PyTorch implementation can be found at `https://github.com/telecombcn-dl/2018-dlai-team1`.

## 4 EXPERIMENTS

We reproduced the experiments using the same datasets as in (Zhang & Khoreva, 2019), which are MNIST, FashionMNIST, CIFAR10 and CelebA.

We used the same hyperparameters as in the original paper: ADAM optimizer with learning rate of $10^{-3}$ for MNIST and $2 \cdot 10^{-4}$ for the rest. The rest of hyperparameters of ADAM were $\beta_1 = 0.5$, $\beta_2 = 0.999$ and $\varepsilon = 10^{-8}$.

As for the progression scheduling described in Section 3, $t_r$ (from equation 1) was fixed to $5 \cdot 10^4$, and the KID was evaluated every $10^5$ steps with $10^5$ fake samples and $10^5$ real samples. One update of the generator was done for every discriminator update. We tried progressive augmentation starting both at level $L = 0$ and $L = 2$. We used the Fréchet Inception Distance (FID) REFERENCE to evaluate the performance of the GAN. A lower FID means that the generated images and their variance are more "real-like".

In Figures 3 and 4, corresponding to MNIST and FashionMNIST we can observe how applying Progressive Augmentation with he right choice for initial augmentation level ($L$) does improve stability. (without PA, the FID value starts increasing after some point). We also observe it helps achieve a lower FID value. On the other hand, in figures 5 and 6 we see that applying progressive augmentation does not seem to increase stability in our experiments, though it helps achieve a lower total FID value anyway.

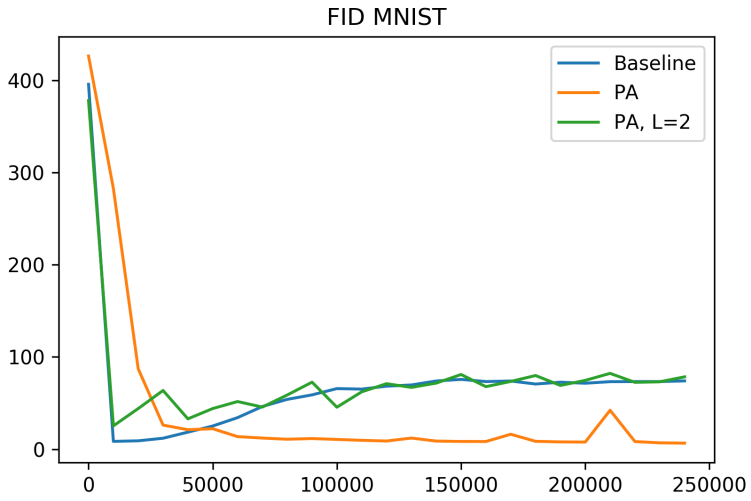



Figure 3: FID value during training with MNIST.

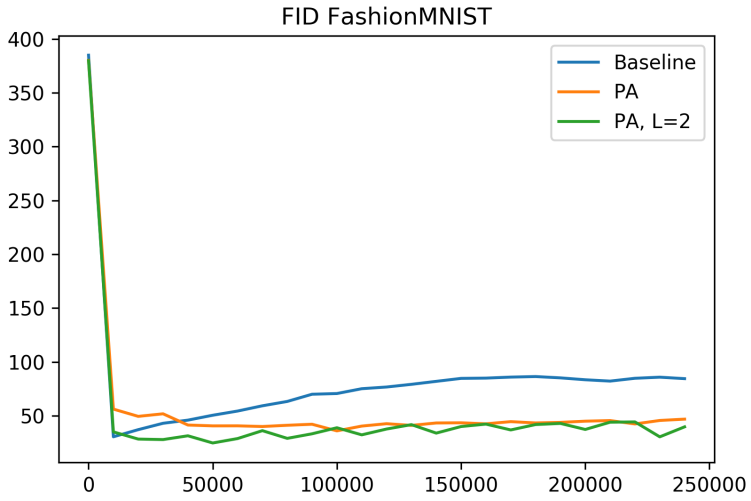
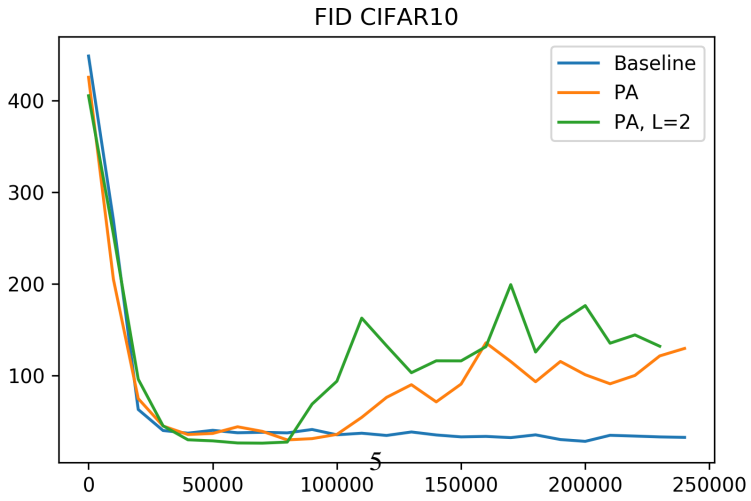



Figure 4: FID value during training with FashionMNIST.



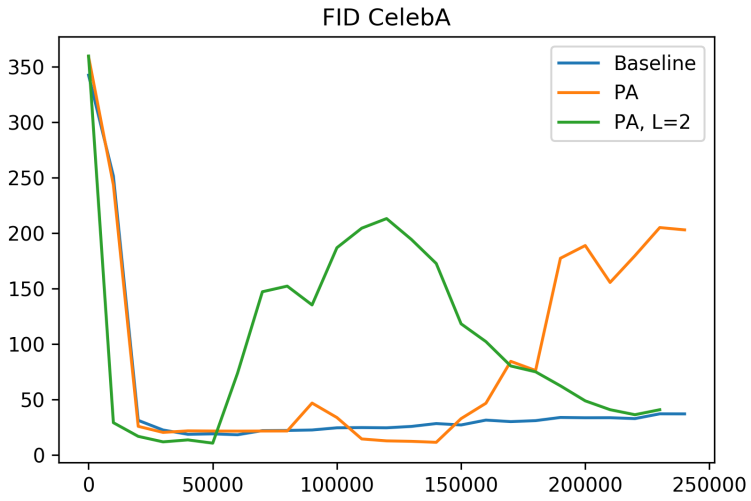
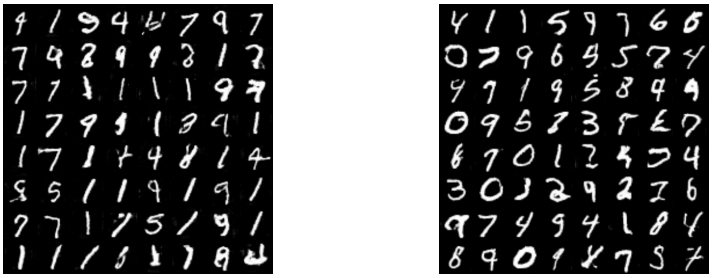



Figure 6: FID value during training with CelebA.

In Figures 7 and 8 We show the comparison between the generated images with and without Progressive Augmentation for MNIST where we can clearly see that PA helps avoid mode collapse, and CelebA, where the difference is more subtle.



(a) Baseline. (b) Progressive augmentation.

Figure 7: Comparison of training a SN-DCGAN on MNIST with and without Progressive Augmentation for the same number of iterations. Here we can see that PA helps avoid mode collapse.

(a) Baseline. (b) Progressive augmentation.

Figure 8: Comparison of training a SN-DCGAN on CelebA with and without Progressive Augmentation for the same number of iterations. For this dataset, it is harder to appreciate the difference between them.

Even though we could see an improvement when applying Progressive Augmentation to a GAN, we could not achieve the exact same metric reported in the paper Zhang & Khoreva (2019). They reported a FID value of 23.2 against our 26.3 for CIFAR10 (they do not provide the exact values for any of the other datasets). We suspect this might be due to a lack of resources to do as many trials as necessary, or an error in the hyperparameters.

## 5 CONCLUSIONS

This report reproduces the results of a paper submitted to the Reproducibility Challenge[1] set up in the 2019 International Conference on Learning Representations (ICLR).

We achieved to reproduce some of the original paper's results but we also found some sections of the paper that, in our opinion, needed more clarification. We contacted the authors regarding those via OpenReview and they were really helpful thus providing details in order to correct the explanation of the scheduling which we pointed out.

In order to reproduce the paper, we implemented a SN-DCGAN and trained it with 4 different datasets to use it as a baseline. Then, we also implemented the Progressive Augmentation(PA) technique which brings a novel approach to increase GAN's performance. Finally, in order to compare the performance of PA and reproduce the results, we used the KID and FID scores to do the evaluation.

We could implement the paper partly successfully, and so we can state that applying PA seems to improve a GAN's performance, as it was claimed originally. However, we were not able to achieve the exact same metrics stated in the paper. We think that it might due to the lack of trials computed (in comparison with theirs), or due to the missclarification when specifying the hyperparameters in the original paper. We also want to highlight that some incorrect descriptions were found in the paper and they are solved now, after discussing with the authors by OpenReview.

In conclusion, even we believe that the results are reliable and reproducible, we also believe that more specificity regarding the hyperparameters used would be helpful for other researchers that want to apply the novel PA technique.

### ACKNOWLEDGMENTS

We gratefully acknowledge the support of Google Cloud for the computational resources.

## REFERENCES

Mikołaj Bińkowski, Dougal J Sutherland, Michael Arbel, and Arthur Gretton. Demystifying mmd gans. *arXiv preprint arXiv:1801.01401*, 2018.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems 27*. 2014.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*, pp. 6626–6637, 2017.

Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*, 2018.

Dan Zhang and Anna Khoreva. PA-GAN: Improving GAN training by progressive augmentation, 2019.

[1] https://reproducibility-challenge.github.io/iclr_2019/