WASSERTEIN GAN SYNTHESIS FOR TIME SERIES WITH COMPLEX TEMPORAL DYNAMICS: FRUGAL ARCHITECTURES AND ARBITRARY SAMPLE-SIZE GENERATION

Th. Beroud[(1,4)], *P. Abry*[(2)], *Y. Malevergne*[(3)], *M. Senneret*[(4)], *G. Perrin*[(4)], *J. Macq*[(3,4)]

[(1)] École Centrale de Nantes, Nantes Université, Nantes, France
[(2)] CNRS, ENS de Lyon, Laboratoire de Physique, Lyon, France
[(3)] Université Paris 1 Panthéon-Sorbonne, PRISM Sorbonne, Paris, France
[(4)] Vivienne Investissement, Lyon, France

ABSTRACT

Generating surrogate data using Deep Neural Network (DNN) has become a classic task in image processing, while DNN time series synthesis is less often considered. The present work addresses issues related to the DNN synthesis of time series, with complex, scalefree time nonreversible temporal dynamics, using Wasserstein Generative Adversarial Network. Instead of proposing yet another overperforming architecture, it discusses, first, synthesis quality quantitative assessment and, second, architecture designs that both reduce, for the Generator, the number of trainable parameters by a factor of 10000 (compared to state-of-the-art architectures), at no expense in performance cost, and permit to generate time series of size longer than that of the training set, without retraining. This works can thus be considered a contribution towards sustainable Artificial Intelligence.

***Index Terms*—** Generative Adversarial Network, Wasserstein distance, time series, convolution layer, multifractal, scalefree, time nonreversibility.

1. INTRODUCTION

Context. In many applications, massive datasets are required, *e.g.*, for statistical learning purposes. However, such large datasets are often not available, as data collection is either costly or not feasible in many practical situations, *e.g.*, biomedical context. This has triggered researches dedicated to the synthesis of *surrogate* data, and has *classically* been addressed using theoretical models tailored to the targeted application. In the modern era, Artificial Intelligence (AI) and deep neural networks (DNN) become tools of choice for the synthesis of surrogate data, with the *hope* of reproducing the key properties of real-world data while avoiding explicit modelling [1]. The present work aims to study several issues related to the use of DNN to synthesize time series with rich and complex temporal dynamics.

Related works. Various architectures, such as Generative Adversarial Networks (GAN) [2] and Auto-Encoders [3], have been involved in image and texture synthesis [4–10], event if less attention has been paid to time series synthesis (to the noticeable exception of [11]). While many proposals of yet better performing architectures are constantly developed [12], less efforts were devoted to assessing the quality of the synthesis achieved with DNN (see *a contrario* [9, 10, 12]). Synthesis quality assessment is sometimes restricted to visual inspection, in image processing notably. In most applications, quality assessment is an issue, as real-world data are only known via key properties chosen by application experts, *e.g.*, stylized facts in financial time series [13]. Along another line, despite massive research works, progresses are still needed to understand the role and importance of each element entering DNN architectures. For instance, DNN architectures are designed to produce data of given size, and training need to be restarted from scratch when other sample sizes are targeted. The present work addresses issues in GAN synthesis for time series characterized by complex temporal dynamics.

Goals, contributions and outline. The main goals are i) to question the role of the fully-connected (FC) layer classically used as the first layer in GAN architectures, and which contains a huge proportion of the parameters to be learned; ii) to investigate to what extent an architecture trained with time series of a given size can be used to generate data with different sizes. To address these issues, two important methodological choices are made *a priori*.

First, the use of real-world data, whose properties are not precisely known and may change along time, may significantly impair or blur the comparisons of the synthesis quality between architectures. Instead, use is made here of skewed multifractal time series used as an archetypal model for data in many applications such as financial time series [14], heart rate variability [15], macroscopic infraslow brain activity [16], Internet traffic [17]... Such models have rich enough scalefree temporal dynamics to challenge DNN-synthesis: The global (or average) second-order statistics (covariance functions) and the higher-order statistics are decoupled and controlled independently, the former by the selfsimilarity parameter H and the later by the multifractality parameter λ^2. The temporal dynamics of such processes is thus characterized by transient local structures (the multifractality or intermittence) that are not *seen* by the second-order statistics. These models can be further complicated by inducing time nonreversibility in their statistics. Multifractal models are abundantly documented, both for synthesis and analysis, by means of wavelet-based multiscale representations. Therefore, the use of these controlled stochastic processes will permit to quantify precisely the quality of DNN time series synthesis, by comparing the multifractal properties of DNN-synthesized time series against the theoretical ones. Definitions, properties and theoretical analyses are recalled in Section 2.

Second, rather than trying to devise a new architecture with possibly better performance but no possibility to understand why, use is made here of the classic, already published and documented, Wasserstein GAN (wGAN) architecture, introduced in [18] and described in Section 3.

Section 4 starts by detailing training protocols and performance assessment settings (see Section 4.1). It then presents the two main achievements. First, it is shown that the parameter-intensive *fully-connected* layer classically used as input layer of most GAN-type

generators can be efficiently replaced by a parameter-frugal *convolution layer* with no loss in synthesis performance (see Section 4.2). Second, it is shown that, using a convolution, rather than FC, input layer, wGAN trained with time series of given sample size can be used to produce longer time series (see Section 4.3).

2. MULTIFRACTAL MODELS AND ANALYSES

Multifractal random walk. Use is made here of a stochastic process, proposed in [19], referred to as the *skewed Multifractal Random Walk* (sMRW), labelled $X_{H,\lambda,K_0,\alpha}(t)$, and designed to combine both the scalefree multifractal dynamics of the *Multifractal Random Walk* (MRW) introduced in [20] and non time-reversible statistics. It is defined from an integration kernel $K_{K_0,\alpha}(t) = K_0|t|^{-\alpha}$ ($K_0 \geq 0$, $\alpha > 0$), a fractional Brownian motion, $B_H(t)$, with selfsimilarity parameter H, an independent zero-mean Gaussian process $\omega_\lambda(t)$ with covariance function, $C_\omega(\tau) = \lambda^2 \ln\left(\frac{L}{||\tau||+1}\right), \tau \leq L$ and 0 otherwise, where λ and L denote the multifractality parameter and the integral scale [14]:

$$X_{H,\lambda,K_0,\alpha}(t) = \int_{\mathbb{R}} e^{\omega(s) - \int_{\mathbb{R}^+} K_{K_0,\alpha}(s-v) dB_H(v)} dB_H(s)\,. \quad (1)$$

When $K_0 \equiv 0$, sMRW reduces to the simpler MRW process, $X_{H,\lambda}(t)$, with multifractal properties but time reversible statistics. For MRW, the selfsimilarity parameter H controls the algebraic decay of the global covariance function, with positive (resp. negative) correlations among increments of $X_{H,\lambda}(t)$ when $H > 1/2$ (resp. $H < 1/2$), while λ controls the *intermittency* or *burstiness* in the increments of $X_{H,\lambda}(t)$, *i.e.*, the occurrence and intensity of local transient structures, that are not *quantified* by the covariance function but pertain to higher-order statistics.

When $K_0 \neq 0$ and $\alpha > 0$, the integration kernel $K_{K_0,\alpha}$ induces time nonreversibility in the statistics, with a well-documented consequence [21], referred to as the *leverage effect* in finance literature [22], consisting of the fact that a third-order statistics of the increments $Y_{H,\lambda,K_0,\alpha}(t) = X_{H,\lambda,K_0,\alpha}(t+1) - X_{H,\lambda,K_0,\alpha}(t)$,

$$\mathcal{L}(\tau) = \mathbb{E} Y_{H,\lambda,K_0,\alpha}(t+\tau)^2 Y_{H,\lambda,K_0,\alpha}(t) \quad (2)$$

is not invariant under time reversal : $\mathcal{L}(\tau) \neq \mathcal{L}(-\tau)$. For sMRW, the leverage function is asymmetric as it remains $\mathcal{L}(\tau) \equiv 0, \forall \tau < 0$ and decays approximately at a rate controlled by α $\forall \tau \geq 0$. For MRW, it boils down to $\mathcal{L}(\tau) \equiv 0, \forall \tau$.

Wavelet based multifractal analysis. It is well-documented that wavelet-based representations constitute reference tools to assess quantitatively scalefree temporal dynamics (see, *e.g.*, [23]). Let $\{\ldots, d_X(j,k), \ldots\}$ denote the discrete wavelet transform coefficients of X, defined as inner products between X and dilated (at scale 2^j) and translated (at location $2^j\underline{k}$) templates of a reference pattern called the mother wavelet (see, *e.g.*, [24]). For MRW, its has been shown that $\forall j$, $\mathbb{E}\left[d_X(j,k)^2\right] \equiv C 2^{j(2H+1)}$ [14] (with $\mathbb{E}$ the mathematical expectation). Therefore, the selfsimilarity parameter can be robustly and efficiently estimated by a linear regression of $\log_2 S(2^j)$ vs. $\log_2 2^j = j$, where $S(2^j) = 1/n_j \sum_{k=1}^{n_j} d_X(j,k)^2$, with n_j the number of wavelet coefficients at scale 2^j [23]:

$$\log_2 S(2^j) = \Gamma + (2H+1)j\,. \quad (3)$$

To assess multifractality, relevant multiscale quantities are the wavelet-leaders $L_X(j,k)$, obtained by replacing $d_X(j,k)$ by a local supremum of the wavelet coefficients $d_X(j',k')$ located at all finer scales $j' \leq j$ within a temporal neighbourhood $2^j(k-1) < 2^{j'}k' \leq 2^j k$ [23]. For MRW, it has been proven that the variance of the logarithm of the wavelet leaders depends on scales as :

$$C_2(j) \equiv \mathrm{Var}\, \ln L_X(j,k) = c_2^0 + c_2 \ln j\,, \quad (4)$$

with $c_2 = -\lambda^2$, permitting simple linear regression estimate.

For sMRW, the linear behaviours of $\log_2 S(2^j)$ and $C_2(j)$ still hold yet with slopes that can no longer be simply related to H and λ as they also depend on K_0 and α.

In addition to scalefree properties, the leverage function can be directly estimated by covariance sample estimators.

3. WASSERSTEIN-GAN ARCHITECTURES

Generative adversarial network (GAN). A GAN relies on the competitive training of two DNNs: a *Generator* and a *Discriminator*. It is a zero-sum game in which the Generator aims to produce data to fool the Discriminator, while the Discriminator tries to classify the data as either surrogate or real. The main weakness of the GAN architecture comes from the Discriminator which provides a value between zero (for fake data) and one (for real data). Due to the sigmoid shape of the discrimination function, its gradient cancels out for data deemed "very real" or "very fake", making the gradients improper for use in the back-propagation step of the training process and thus hindering the convergence of the iterative procedure (see, *e.g.*, [18] for details).

Wasserstein-GAN (wGAN). To address this issue, it was proposed to replace the Discriminator which tends to deliver a binary – and hence non differentiable – output, with a *Critic*, that provides a differentiable distance between the distributions of training and surrogate data. For reasons very clearly exposed in [18], the Wassertein (or Earth-Mover) distance, whose properties avoid the vanishing gradient issue during the back-propagation step, is chosen. Accounting for Kantorovich-Rubinstein duality theorem and the resulting regularity constraint (see, *e.g.*, [25] for details), the Critic C maximizes the objective function:

$$\begin{aligned} L(C) =& \mathbb{E}_{x\sim\mathbb{P}_r}\left[C(x)\right] - \mathbb{E}_{z\sim\mathbb{P}_m}\left[C(z)\right] \\ &+ \gamma \cdot \mathbb{E}_{y\sim\mathbb{P}^*}\left[\left(\max\left\{0, \|\nabla_y C(y)\|_2 - 1\right\}\right)^2\right]\,, \end{aligned} \quad (5)$$

for a given training (resp. model) distribution $\mathbb{P}_r$ (resp. $\mathbb{P}_m$), while $\mathbb{P}^*$ denotes a linear interpolation between real and fake data distributions. We refer the interested reader to [26, 27] for details.

wGAN-architectures. Use is made here of the wGAN-architecture proposed in [28]. The aim of the present work is to study the architecture of the Generator. Thus, the Critic is left unchanged: Three convolution and three FC layers, all followed by Leaky ReLU activation functions.

For the generator, it is documented that GAN architectures based on convolution layers are more stable during training [29] and it was discussed in [10] that architectures with a number of layers kept as low as can be for a given task yield better reproducibility. Therefore, a Generator with simple architecture was designed (see Fig. 1). It starts with one FC layer of the size of the data to generate, followed by a Leaky ReLU activation function (Layer0). After reshaping, it proceeds with three convolution transpose layers (Layer1, Layer2, Layer3) each implying signal length upsampling and depth reduction by a factor of two, batch normalization, and Leaky ReLU activation. Filter length are kept constant and set to 7. In Layer3, the Leaky ReLU activation is replaced with a bounding nonlinear smooth function $f_{\alpha,\beta}(x) = \beta \tanh(\alpha x)$, where α and β are two trainable parameters.

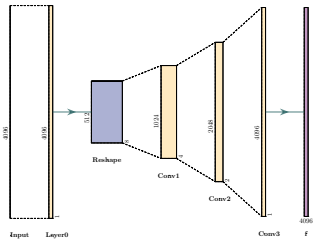

Fig. 1: **Generator Architecture.** Produced from tools at `https://github.com/HarisIqbal88/PlotNeuralNet`.

4. SYNTHESIS PERFORMANCE ASSESSMENT

4.1. Training and performance assessment settings

Training protocols. Implementation was performed on a GPU NVIDIA GeForce RTX 3090 with a i9-10900K processor, using PyTorch. Following [30], training parameters were set as follows. For optimization, RMSprop is used for both generator and discriminator with a learning rate of 1×10^{-4}. The training phase consists 10000 iterations, with 5 sub-iterations for the Critic for each iteration of the Generator. Batch normalization is applied on both generator and Critic, which consists in normalizing independently the inputs of each network layer, with batch size of 128. The regularization coefficient in Eq. (5) is set to $\gamma = 10$ as in [26].

Training Set. The training set consists of 10000 independent copies of MRW or sMRW, each with sample size 2^{12}. For MRW, (H, λ) are set to $(0.8, \sqrt{0.03})$. For sMRW, $(H, \lambda, K_0, \alpha)$ are set to $(0.5, \sqrt{0.03}, 0.1, 0.8$. It has been checked that conclusions reported below do not depend on this particular choice of parameters, only used as a representative example. sMRW and MRW are synthesized using codes, designed by ourselves, relying on circulent embedded matrix techniques classically used for Gaussian processes (see, *e.g.*, [31]).

Performance assessment. For each architecture and for each independent training, 100 independent realizations of wGAN surrogate MRW and sMRW are generated after training is completed. Functions $\log_2 S(j)$, $C_2(j)$ and $\mathcal{L}(\tau)$ are first computed for 100 independent realizations of MRW and sMRW, and averaged. The comparisons of their behaviours as functions of scales or of τ against the same functions computed from time series synthesized numerically from the definition of MRW and sMRW, quantify the quality of the wGAN synthesis. Multifractal analysis relies on least asymmetrical orthogonal Daubechies wavelets, with 3 vanishing moments (see, *e.g.*, [24]) and is implemented using toolboxes (MATLAB and PYTHON) made publicly available from the authors' webpages.

Performance reproducibility. To assess the reproducibility of the achieved performance, training is repeated 20 times, independently and from scratch, for each architecture, under the same protocol and training set. The overall synthesis quality assessment is thus quantified by averages and standard deviations across independent training sessions of functions $\log_2 S(j)$, $C_2(j)$ and $\mathcal{L}(\tau)$.

Generator architectures. In experiments reported below, the FC input layer (Layer0) is challenged with *Convolution layers*, with kernel size 512 or 32, referred to as Conv512 and Conv32 architectures, leading to drastic decreases in the number of trainable parameters for the Generator (see Table 1).

4.2. GAN Input layer: fully-connected vs. convolution layer

To compare the quality of the synthesis achieved by wGAN generator with different input layers (Layer0), Fig. 2 superimposes, for MRW (left) and sMRW (right), the average values $\pm$ one standard deviations for $\log_2 S$ (top plot) and C_2 (middle plot) as functions of the logarithm of analysis scales and estimated leverage $\mathcal{L}$ as function of the delay τ (bottom plot) obtained from true MRW/sMRW (black) and wGAN surrogate MRW/sMRW produced using a FC (blue), a (kernel512) convolution (magenta) and a (kernel32) convolution (red) Layer0. Fig. 2, the first main contribution of this work, yields a number of interesting conclusions.

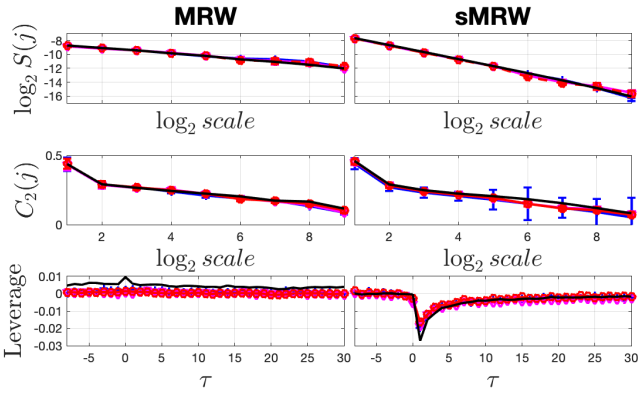

Fig. 2: **Synthesis quality assessment.** For MRW (left) and sMRW (right), $\log_2 S$ and C_2 as functions of the logarithm of analysis scales (top and middle plots) and estimated leverage $\mathcal{L}$ as function of the delay τ (bottom plot) obtained from true (s)MRW (black) and surrogate (s)MRW produced by DNN with a FC (blue), a (kernel512) convolution (magenta) and a (kernel32) convolution (red) Layer0 (averages $\pm$ one standard deviations across 20 independently repeated from scratch training).

Layer	0	1	2	3	Total
fully-connected	16781314	236	62	19	16781631
Conv512	515	236	62	19	832
Conv32	35	236	62	19	352

Table 1: Number of trainable parameters for three wGAN Generator architectures. For completeness, the Critic implies a total number of trainable parameters summing up to 1653887.

First, for each of the 3 different input layers, the match between true and surrogate MRW and sMRW of the three benchmark functions, $\log_2 S(j)$, $C_2(j)$ and $\mathcal{L}(\tau)$, is quite remarkable. Further, for most cases, the $\pm$ one standard deviation confidence intervals, obtained from repeated from scratch independent training sessions, are printed but barely visible for most cases. This indicates that achieved performance is perfectly reproducible: it is not the only training that worked well that was shown! wGAN thus produces high quality surrogates of MRW and sMRW, with impressive reproduction of complex temporal dynamics. MRW are designed to have long-memory, with a parameter H significantly above 0.5, indicating significant long term correlations, resulting in very slow algebraic decay of the second order statistics. In addition, multifractality, structured intermittent burstiness of transient local structures, conveys features of the temporal dynamics that are not coded in the 2nd order statistics but by higher order statistics. Moreover, sMRW further contains complexity as statistics are not covariant under a time change. It is thus quite impressive that wGAN architectures capture long term correlations and local transient structures together with their temporal organization as well as time non-reversibility, and this, in a very consistent, quantitative and reproducible (across training sessions) manner. It is worth recalling that wGAN had absolutely no

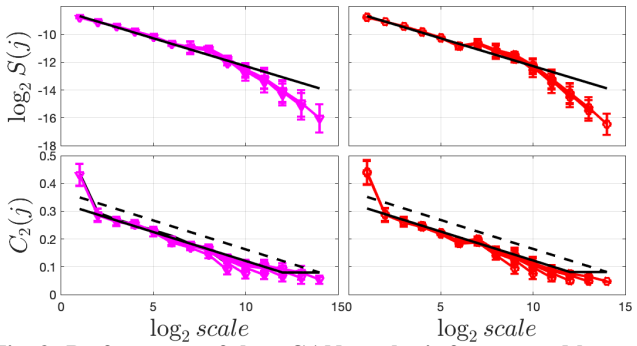

Fig. 3: **Performance of the wGAN synthesis for targeted longer MRW time series.** $\log_2 S$ (top) and C_2 (bottom) as functions of the logarithm of analysis scales, obtained as averages and standard deviations across independent training sets, for Conv32 (left, magenta) and Conv512 (right, red) input layers. The superimposed solid black lines correspond to the theoretical functions $\log_2 S$ and C_2 for a MRW of largest targeted size $N' = 2^{17}$, with integral scale kept to $L \equiv N = 2^{12}$. In addition, dashed black lines correspond to MRW of size $N' = 2^{17}$ and integral scale kept to $L \equiv N' = 2^{17}$.

a priori (expert-type) information related to the statistical properties of the data to learn and that multifractal analysis ($\log_2 S(j)$, $C_2(j)$) and leverage estimation $\mathcal{L}(\tau)$ are only used *a posteriori* for synthesis quality assessment and never enters the definition of the loss function for the training, that involves a non *a priori* informed generic Wassertein based distance. We believe that this high quality and reproducible performance in synthesis stems from the major methodological choice made here: simple architectures. The designed wGAN generators contain only 4 layers.

Second, the comparisons between the performance achieved with three different architectures designed by varying the input layer (Layer0) are essentially similar in average and standard deviations. Fig. 2 (right) shows that the FC input layer yields slightly more variability in multifractality ($C_2(j)$), notably at coarse scales. These observations indicate that replacing, in the input layer, a FC structure by a convolution one, yields equivalent or event slightly better reproducible performance. This is another impressive outcome as it yields a reduction by a factor of 10^4 in the number of parameters to train (from $\simeq 10^7$ to $\simeq 10^3$!). For the convolution input layer, two kernel sizes were tested (32 and 512) with essentially indistinguishable performance. Overall, this teaches us that comparable performance can be reached with as few as 352 parameters to train for the Generator, when blind use of classic on-the-shelves architectures involve above 10^7 parameters. Further, each short-kernel convolution layer involved in the Conv512 and Conv32 architectures cannot, by construction and *per se*, capture long-term statistical structures. The very satisfactory performance achieved with the Conv512 and Conv32 architectures therefore indicate that the long-term correlation associated with scalefree (selfsimilarity) temporal dynamics as well as long-term temporal structure in occurrences of local transient bursts (multifractality) are captured by the concatenation of layers, combining linear (convolution) and nonlinear (ReLU) filters. Remarkably, 4 layers only are needed.

Third, the case of sMRW is even more impressive as it involves time non-reversible statistics. In the proposed architectures, none of the elementary block permits to code for time nonreversible structure. Thus, again, time non reversibility is encoded via the concatenation of several linear/nonlinear layers. And again, a small number of layers is sufficient to code time nonreversibility.

4.3. Generating data longer than training size

Principle. A major side benefit of the replacement of the FC by a convolution input layer is to permit to generate time series of size N' larger than the size of the training set N, $N' \geq N$, in a simple manner: It suffices to use as input for the wGAN a white noise of the size of the targeted time series. Indeed, in major contradistinction with a FC layer, the number of parameter of an input layer based on a convolution is independent of the size of the input noise.

Synthesis quality assessment. To quantify the wGAN ability to generate longer time series, the networks trained to generate MRW of size $N = 2^{12}$ (and described in Section 4.2) are used to generate time series of size $N' = [2^{12}, 2^{13}, 2^{14}, 2^{15}, 2^{16}, 2^{17}]$. For each targeted size N', 100 time series are generated and subjected to multifractal analysis. Means and standard deviations, computed across 6 independent training for each of the different targeted sample size N', are superimposed in Fig. 3. To ease interpretation, the theoretical functions $\log_2 S$ and C_2 for a MRW of largest targeted size $N' = 2^{17}$, with integral scale kept to $L \equiv N = 2^{12}$ are superimposed as solid black lines. In addition, dashed black lines correspond to MRW of size $N' = 2^{17}$ with integral scale $L \equiv N' = 2^{17}$. Top plot in Fig. 3 shows that the linear behaviour of the scaling of $\log_2 S(j)$ is satisfactorily reproduced for wGAN surrogate MRW of size at least 2^{15}. This implies that longer wGAN surrogate MRW do reproduce selfsimilarity, with a selfsimilarity parameter that matches that of the training set.

Bottom plot in Fig. 3 displays the functions $C_2(j)$ obtained from wGAN MRW surrogate for the different targeted sample sizes N'. Interestingly, it shows that the functions C_2 for the different N' very satisfactorily reproduce the theoretical function C_2 corresponding to MRW with integral scale kept to $L = 2^{12}$. This shows that multifractality is correctly imposed to wGAN surrogate MRW of longer size, a quite remarkable fact. In addition, it is also quite impressive that the wGAN has learned the integral scale of the training data and kept it fixed in the longer size wGAN surrogate MRW: wGAN could indeed not disentangle the sample size and the integral scale as they were identical in the training set. All together, these analyses show that the trained wGAN Generators turned able to generate surrogate MRW samples which reproduce very satisfactorily selfsimilarity and multifractality for sizes up to $N' = 2^{14}$ or even $N' = 2^{15}$, that is 4 to 8 times longer than those of the training set. Similar findings (not reported here for the sake of space) are valid for sMRW.

5. CONCLUSIONS AND PERSPECTIVES

This work shows that wGAN Generators can be trained to generate surrogate time series that remarkably reproduce intricate temporal dynamics (selfsimilarity, multifractality, time nonreversibility). In addition, the limited complexity of the designed architectures (4 layers only) ensures systematic reproducibility of achieved performance across independent training sets. The key contributions of the present work are to show that i) the parameter-greedy FC input layer can be replaced by a convolution layer, reducing by a factor of 10^4 the number of parameters to be trained, with no decrease in performance; and ii) the convolution layer permits to efficiently generate wGAN surrogate time series of sizes substantially longer than that of the training set. All together, these results can also be framed into researches towards *frugal* AI, in contradistinction against the common trends of always deeper and more complex DNN architectures. This is thus a contribution to researches dedicated to limit the energy footprint of AI. Future research efforts will be devoted to wGAN Generators of multivariate time series, opening a wide set of difficult issues.

6. REFERENCES

[1] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*, MIT press, 2016.

[2] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[3] H. Huang, Z. li, R. He, Z. Sun, and T. Tan, "IntroVAE: Introspective variational autoencoders for photographic image synthesis," in *Advances in Neural Information Processing Systems*, 2018, pp. 52–63.

[4] Y. LeCun, L. Bottou, Y Bengio, P. Haffner, et al., "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[5] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[6] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[7] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations*, 2015.

[8] L. A. Gatys, A. S. Ecker, and M. Bethge, "Image style transfer using convolutional neural networks," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[9] V. Mauduit, P. Abry, R. Leonarduzzi, S.-G. Roux, and E. Quemener, "DCGAN for the synthesis of multivariate multifractal textures: How do we know it works?," in *2020 IEEE 30th International Workshop on Machine Learning for Signal Processing (MLSP)*. IEEE, 2020, pp. 1–6.

[10] P. Abry, V. Mauduit, E. Quemener, and S. Roux, "Multivariate multifractal texture DCGAN synthesis: How well does it work? how does one know?," *Journal of Signal Processing Systems*, vol. 94, no. 2, pp. 179–195, 2022.

[11] B. Zhao, H. Lu, S. Chen, J. Liu, and D. Wu, "Convolutional neural networks for time series classification," *Journal of Systems Engineering and Electronics*, vol. 28, no. 1, pp. 162–169, 2017.

[12] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, "Analyzing and improving the image quality of stylegan," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 8110–8119.

[13] R. Cont, "Empirical properties of asset returns: stylized facts and statistical issues," *Quantitative finance*, vol. 1, no. 2, pp. 223, 2001.

[14] E. Bacry, L. Duvernet, and J.-F. Muzy, "Continuous-time skewed multifractal processes as a model for financial returns," *Journal of Applied Probability*, vol. 49, no. 2, pp. 482–502, 2012.

[15] T. Nakamura, K. Kiyono, H. Wendt, P. Abry, and Y. Yamamoto, "Multiscale analysis of intensive longitudinal biomedical signals and its clinical applications," *Proc. IEEE*, vol. 104, no. 2, pp. 242–261, 2016.

[16] D. La Rocca, N. Zilber, P. Abry, V. van Wassenhove, and Ph. Ciuciu, "Self-similarity and multifractality in human brain activity: A wavelet-based analysis of scale-free brain dynamics," *Journal of neuroscience methods*, vol. 309, pp. 175–187, 2018.

[17] R. Fontunge, P. Abry, K. Fukuda, D. Veitch, K. Cho, P. Borgnat, and H. Wendt, "Scaling in internet traffic: a 14 year and 3 day longitudinal study, with multiscale analyses and random projections," *IEEE/ACM T. Networking*, vol. 25, no. 4, 2017.

[18] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *International conference on machine learning*. PMLR, 2017, pp. 214–223.

[19] B. Pochart and J.-Ph. Bouchaud, "The skewed multifractal random walk with applications to option smiles," *Quantitative finance*, vol. 2, no. 4, pp. 303, 2002.

[20] E. Bacry, J. Delour, and J.-F. Muzy, "Multifractal random walk," *Physical Review E*, vol. 64, no. 2, pp. 026103, 2001.

[21] J.-Ph. Bouchaud, A. Matacz, and M. Potters, "Leverage effect in financial markets: The retarded volatility model," *Physical review letters*, vol. 87, no. 22, pp. 228701, 2001.

[22] F. Black, "Studies of stock market volatility changes," in *1976 Proceedings of the American statistical association bisiness and economic statistics section*. 1976, American Statistical Association.

[23] H. Wendt, P. Abry, and S. Jaffard, "Bootstrap for empirical multifractal analysis," *IEEE signal processing magazine*, vol. 24, no. 4, pp. 38–48, 2007.

[24] S. Mallat, *A wavelet tour of signal processing*, Elsevier, 1999.

[25] C. Villani, *Optimal transport: old and new*, Springer, 2009.

[26] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville, "Improved training of wasserstein GANs," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017, pp. 5769–5779.

[27] H. Petzka, A. Fischer, and D. Lukovnicov, "On the regularization of wasserstein GANs," in *International Conference on Learning Representations*, 2018.

[28] F. de Meer Pardo, "Enriching financial datasets with generative adversarial networks," M.S. thesis, Delft University of Technology, The Netherlands, 2019.

[29] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.

[30] C. Hogenboom, "Generation of synthetic financial time series with GANs," M.S. thesis, Maastricht University, The Netherlands, 2020.

[31] H. Helgason, V. Pipiras, and P. Abry, "Synthesis of multivariate stationary series with prescribed marginal distributions and covariance using circulant matrix embedding," *Signal processing*, vol. 91, no. 8, pp. 1741–1758, 2011.