Deep Learning based synthetic time series: Strengths and benefits of using Wasserstein distance

Johan Macq*,†, Yannick Malevergne*, Marc Senneret†, Patrice Abry‡, Gérald Perrin† and Laurent Jaffrès†
*Université Paris 1 Panthéon-Sorbonne, PRISM Sorbonne, Paris (FR)
†Vivienne Investissement, Research & Development, Lyon (FR)
‡CNRS, ENS de Lyon, Laboratoire de Physique, Lyon (FR)

***Abstract*—The automated generation of synthetic time series with specified characteristics constitutes an everlasting and ubiquitous problem in signal processing and in a wide variety of applications. Recently, synthetic time series generation has often been reformulated as optimization problems, sometimes using deep learning or neural network architectures. In the context of financial asset replication, the aim of this paper is to compare the potential advantages and limitations of using Wasserstein p-distances vs. L^p-norms as cost functions to be minimised when replicating synthetic time series from a given set of instruments. This paper first illustrates, on a toy example, that Wasserstein p-distances suffer from a possible degeneracy of the fixed point to which the minimization converges, and proposes a solution to this problem. Second, on a realistic problem, it shows that using Wasserstein p-distances to replicate assets from a set of financial instruments leads to better performance than L^p-norms, and explains why.**

I. Introduction

Context. The automated generation of synthetic signals or images with specified statistical properties is a classical, but still important and persistent challenge in Signal/Image processing and in a wide variety of applications, very different in nature [20]. Its formulation has been significantly renewed with the advent of practical and formal tools in Optimisation theory and Artificial Intelligence [12]. However, such optimisation-based formulation in synthetic data design has raised challenging new issues, including the choice/role/impact of distances in minimisation strategies. The present paper aims to discuss the advantages of using Wasserstein p-distances to replicate synthetic time series from a set of instruments, for an application in finance where this problem is usually referred to as asset replication [7, 9, 10]. To the best of our knowledge, this is the first time that Wasserstein p-distances have been used to solve an optimal allocation problem for asset replication.

Related works. Classically, synthetic time series generation is formulated as Gaussian white noise filtering, with the key difficulties consisting in designing filters that permit to reach the targeted time series statistical properties [18]. The design of non-Gaussian time series often involves the use of non-linear filters, such as stochastic differential equations or GARCH models [16], and motivated optimization-based formulations of the filter design [1]. Alternatively, synthetic time series design has also been based on combining elements of preselected dictionaries [22]. Often, the formulation of the dictionary element combination leads to optimization problems.

In the financial sector, data synthesis also defines a classical task, either to increase and complement historical data sets, [3], or to replicate target financial time series by dynamically combining other time series consisting a set (or dictionary) of pre-selected financial instruments [13].

The broad framework of Artificial Intelligence (AI) has led to a massive revisit of these classical approaches by designing architectures that allow either the supervised design of filters (*e.g.*, variational auto-encoders [5, 15], generative adversarial networks [2, 12], diffusion networks [14]) or the supervised learning of dictionaries [23].

A common caveat in these AI & Optimization formulations lies in the choice of the criterion (or distance) to be minimized. Classical choices refer to Kullback-Leibler and/or Jensen-Shannon divergence, L^p-norms and/or Wasserstein p-distances, but the role and impact of such choices still deserve documented studies, the core question to which the present work aims to contribute.

Goals, contributions and outline. The goal of this work is to study the benefits and caveats of using Wasserstein p-distances in financial asset replication. To this end, Section II introduces the replication formalism, formulated as a causal and dynamic combination of tradable financial instruments, forming dictionary time series. As a first contribution, Section III proposes a comparative discussion of the properties of candidate optimization distances. As a second contribution, Section IV designs and studies a toy problem in financial asset replication, aiming to show that the use of the Wasserstein 1-distance suffers from a major practical caveat in the form of a potential built-in degeneracy of the fixed point to which the minimization leads. Section IV also suggests a simple way to circumvent this issue, by combining several distances. As a third contribution, Section V studies in depth, for a complex financial asset replication problem, the solutions obtained when using different minimization distances. It shows the clear benefits of involving the Wasserstein 1-distance in the replication problem compared to other distances.

II. Causal dynamic time series replication

The replication of financial assets [13] (or the hedging thereof [10]) is classically formulated as follows. Given a filtered probability space $(\Omega, \mathcal{F}, \mathbb{P})$, Y is an $\mathbb{R}$-valued $\mathcal{F}$-adapted stochastic process called the target, and X is an $\mathbb{R}^M$-valued $\mathcal{F}$-adapted stochastic process called the instruments. Both represent the prices of financial assets. Given an $\mathbb{R}^M$-

valued $\mathcal{F}$-adapted stochastic process w, called the strategy, the replication portfolio, denoted $\widehat{Y}$, is defined as a linear combination of instruments:

$$\forall t > 0,\ \widehat{Y}_t = w_{t-1} \cdot X_t\,, \tag{1}$$

which satisfies the so-called self-financing condition

$$\forall t > 0,\ \widehat{Y}_t - \widehat{Y}_{t-1} = w_{t-1} \cdot (X_t - X_{t-1})\,. \tag{2}$$

As usual, $\cdot$ denotes the inner product in $\mathbb{R}^M$.

The strategy w depends on some state variables, *i.e.* $w_t \in \mathbb{R}^M$ is determined with the information available up to time t and represents the quantities of instruments with price X to invest in.

The replication task boils down to finding an optimal strategy w^*, called the replication strategy, which ensures that at each time $\widehat{Y}$ is as close as possible to the target Y. Since there is no *a priori* guarantee of the existence of a w^* such that $\forall t > 0,\ \widehat{Y}_t = Y_t$, we need to define distance measures to assess the "closeness" between the two series. In the financial asset replication literature, the closeness is often assessed using distance measures between the (log-)returns of each series denoted $\widehat{y}$ and y respectively [21]. This ensures that past biases do not accumulate along the replicated trajectories. Furthermore, the target can be defined equivalently by the series of its returns y (given Y_0) or by the series of its price Y. For these reasons, and from now on, we will define the target by y rather than by Y.

III. Wasserstein p-distances vs. L^p-norms

Estimating the replication strategy involves minimizing a cost function that quantifies the differences between the target vector $y = (y_t)_{t\in[1,L]} \in \mathbb{R}^L$, sampled over a trajectory of length L, and the achieved estimate $\widehat{y} = (\widehat{y}_t)_{t\in[1,L]} \in \mathbb{R}^L$, with the corresponding probability distributions η and $\widehat{\eta}$ respectively. Several candidate distances between random vectors are considered: L^p-norms vs. Wasserstein p-distances.

A. L^p-distances

The L^p-distance between y and $\widehat{y}$ is defined as:

$$L^p(y, \widehat{y}) = \mathbb{E}\left[(\|y - \widehat{y}\|_p)^p\right]^{1/p} \text{ for } p \geq 1\,, \tag{3}$$

with $\|y - \widehat{y}\|_p$ the L^p-norm in $\mathbb{R}^L$.

For the replication task, the main strength of L^p-distances is that they lead to convergence in probability. In other words, if there exists a replication strategy w^* such that $L^p(y, \widehat{y}) \to 0$, then $\widehat{y} \xrightarrow{\mathbb{P}} y$. Loosely speaking, this ensures that $\widehat{y}_t$ is close to y_t for all $t \in [0, L]$ with a very high level of confidence. Furthermore, and from a practical perspective, L^p-distances benefit from low computational cost.

The present work will focus on a comparative use of $p = 1$ and $p = 2$. It is well documented that the L^1-distance is not well suited to account for rare events, or, equivalently, for distribution tails [17]. In any application for replicating financial time series, the failure to account for rare events can be a critical weakness, as such events usually result in large losses (e.g. market crash). This is partially corrected by using an L^2-distance.

B. Wasserstein p-distance

The Wasserstein p-distance between two distributions η and $\widehat{\eta}$ of random vectors in $\mathbb{R}^L$ is defined as [24]:

$$W_p(\eta, \widehat{\eta}) = \left(\inf_{\gamma\in\Gamma(\eta,\widehat{\eta})} \mathbb{E}_{(z,\widehat{z})\sim\gamma}\left[d(z, \widehat{z})^p\right]\right)^{1/p}\,, \tag{4}$$

for $p \geq 1$, where $d()$ is a distance on $\mathbb{R}^L$ and $\Gamma(\eta, \widehat{\eta})$ is the set of all couplings of η and $\widehat{\eta}$.

A major advantage of using Wasserstein p-distances is that they are everywhere continuous and almost everywhere differentiable functions [2]. This makes them well suited for training neural networks (via backpropagation).

However, an important limitation is that Wasserstein p-distances only guarantee convergence in distribution, *i.e.,* with $\widehat{y} \sim \widehat{\eta}$ and $y \sim \eta$, $W_p(\eta, \widehat{\eta}) \to 0 \implies \widehat{y} \xrightarrow{\mathcal{L}} y$, which is not sufficient to ensure that $\widehat{y}_t$ remains close to y_t with high confidence for all $t \in [1, L]$ [2]. This implies that solutions to $W_p(\eta, \widehat{\eta}) = 0$ depend on the potential symmetries of η. For example, if $\eta = \mathcal{N}(0, \Sigma)$ where Σ is a diagonal matrix, then both $\widehat{y} = y \sim \eta$ and $\widehat{y} = -y \sim \eta$ are solutions to $W_p(\eta, \widehat{\eta}) = 0$. This degeneracy can be problematic in the replication task.

Furthermore, computing the Wasserstein p-distance is numerically demanding. We only use the Wasserstein 1-distance here, as it can be efficiently estimated using its dual representation [24]: $W_1(\eta, \widehat{\eta}) = \sup_{\|f\|_L\leq 1} \mathbb{E}_{z\sim\eta}[f(z)] - \mathbb{E}_{\widehat{z}\sim\widehat{\eta}}[f(\widehat{z})]$ with $\|.\|_L$ the Lipschitz norm.

Practically, in the present work, the Wasserstein 1-distance is estimated by training a neural network named *Critic* to minimize Eq. (8) in [19]. The chosen network consists of three convolution layers, each followed by a maxpool layer, three dense layers (including the output layer). All layers (except for the output) are followed by a LeakyRelu activation ($\alpha = 0.2$). The total number of parameters of the *Critic* is $58,687$. The optimizer used is Adam with $l_r = 10^{-4}$ and $\beta = (0, 0.9)$.

IV. Replication toy problem: the caveat of using the Wasserstein 1-distance

In this first experiment, we deliberately choose a toy model and a simple target whose replication strategy is known in closed form. We show that the degeneracy in the convergence of the W_1-distance discussed above is not an isolated theoretical case, but arises very quickly and needs to be addressed. We therefore propose a simple procedure to avoid this problem.

Market definition. We consider a financial market *à la* Black and Scholes with a risky asset denoted S, whose price is an $\mathcal{F}$-adapted geometric Brownian motion [6]. For simplicity and without loss of generality, we assume that the risk-free rate is zero. For this experiment, the log-returns on the risky asset $s_t := \log S_t - \log S_{t-1}$ are iid Gaussian random variables with zero mean and volatility $\sigma > 0$. In addition, the market provides a pair of call and put options with strike price $K \in \mathbb{R}^+$ and expiration date $T > 0$.[1] Their prices at time

[1] Call (resp. put) options are financial contracts that give the right – but not the obligation – to buy (resp. sell) the underlying asset S at a specified price $K \in \mathbb{R}^+$, and at a future date $T > 0$.

$t \leq T$ are denoted C_t and P_t respectively.

Target definition. We define the target $y = (y_t)_{t\in[1,L]}$ to be replicated as the underlying asset itself, so that $y_t = s_t$, $\forall t > 0$. Since $(s_t)_{t\in[1,L]}$ is a vector of iid zero mean Gaussian random variables, the target distribution is symmetric ($y \overset{\mathcal{L}}{=} -y$) and so this simple case presents a degeneracy on the convergence of W_1 as detailed in section III.

Analytical solution. As a consequence of the call-put parity relationship, the underlying asset S can be perfectly replicated by an options portfolio:

$$\forall t \leq T,\ S_t = K + C_t - P_t\,. \tag{5}$$

This equation means that S can be perfectly replicated by holding an amount K in cash, buying a call option C and selling a put option P. Thus, the replication of the underlying asset has an analytical solution given by the replication strategy $w^* = (K, 1, -1)$ for the instruments $X_t = (1, C_t, P_t)$ (where the first instrument $X^{(0)}$ denotes the monetary unit). Eqs. (1) and (5) give $\forall t \leq T$, $\widehat{Y}_t = S_t$ and equivalently, $\forall t \leq T$, $\widehat{y}_t = y_t$.

The W_1 degeneracy in theory. To analyze the landscapes drawn by the different distances considered here as functions of the composition of the replication portfolio, we examine the set of constant strategies given by $w = (w^{(0)}, w^{(1)}, w^{(2)})$ with $(w^{(1)}, w^{(2)}) \in [-1.5, 1.5]^2$ and $w^{(0)} = S_0 - w^{(1)}C_0 - w^{(2)}P_0$ to ensure $\widehat{Y}_0 = S_0$. Note that $(w^{(1)}, w^{(2)}) = (1, -1)$ yields $w = w^*$.[2] The shapes of the mappings $(w^{(1)}, w^{(2)}) \mapsto L^1(y, \widehat{y})$, $L^2(y, \widehat{y})$ and $W_1(\eta, \widehat{\eta})$ are shown on Figure 1. They have been obtained from numerical simulations based on $N = 2^{17}$ trajectories of length $L = 110$ of the underlying price S. We set $S_0 \sim \mathcal{U}(90, 110)$, $\sigma = 0.20$ and $r = 0$. Option prices are determined using the Black-Scholes model with $K = 100$ and $T = 120 > L$ [6]. The W_1-distance is computed by training the *Critic* network over 100 backpropagations. The implementation was performed using PyTorch, on a GPU NVIDIA GeForce RTX 3090 (i9-10900K processor).

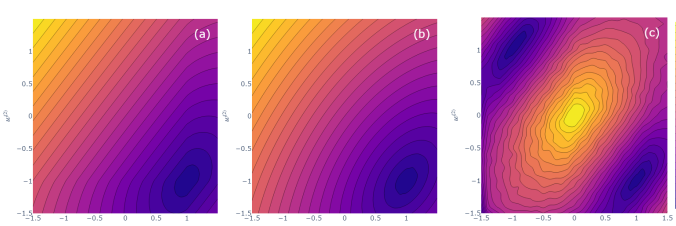

Fig. 1: L^1 (panel a), L^2 (panel b) and W_1 (panel c) normalized distances between $y \sim \eta$ and $\widehat{y} \sim \widehat{\eta}$ for $(w^{(1)}, w^{(2)}) \in [-1.5, 1.5]^2$.

Figure 1 illustrates that for constant strategies, the L^1 and L^2-distances are convex with respect to $(w^{(1)}, w^{(2)})$ and have a single global minimum at $w = w^*$ for which $L^1 = L^2 = 0 \iff \widehat{y} = y$, $\mathbb{P}$-a.s., as expected. It also illustrates that the W_1-distance is not convex and admits two symmetric minima given by $(w^{(1)}, w^{(2)}) = (1, -1)$ and $(w^{(1)}, w^{(2)}) = (-1, 1)$. These two solutions achieve $W_1 = 0$ because they both verify $\widehat{y} \overset{\mathcal{L}}{=} y$, since y is a zero mean Gaussian vector. This is a direct illustration that the degeneracy in the convergence of W_1 caused by the symmetry of the target distribution occurs even in simple cases.

The W_1 degeneracy in practice. What are the consequences of degeneracy on a minimization program if we do not have an analytical solution beforehand? To answer this question, we pretend that we do not know the optimal strategy w^*, and we train a neural network with parameters set θ under the W_1-distance to find w^*. For each time t, the network takes as input $(C_t/S_t, P_t/S_t)$ and returns as output $(w_t^{(1)}, w_t^{(2)})$.[3] Then $\widehat{Y}$ is determined according to Eq. (1). The network consists of two Dense layers of 25 units with ReLu activation and one Dense output layer of 2 units without activation, its total number of parameters is 777. For each backpropagation of the network, there are 5 backpropagations for the Critic computing W_1 [2]. The batchsize is 64 and the optimizer used is the same as for the Critic. The train and test data sets are generated as above. We performed 10 independent training of such networks by minimizing $W_1(\eta, \widehat{\eta})$ until convergence. We test the networks on the test dataset and show the results of two of them on Figure 2.

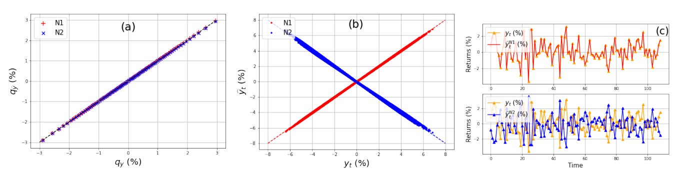

Fig. 2: (a): Quantile-quantile plot of the target returns against the returns of type N_1 (resp. N_2) network in red (resp. blue), (b): Target returns against network returns, (c): Examples of target and network returns trajectories.

All networks converged to one of the two constant strategies determined above, for which $W_1(\eta, \widehat{\eta}) = 0$. After convergence we count half of the networks with the strategy $(w^{(1)}, w^{(2)}) = (1, -1)$ – hereafter N_1 – and the other half with the strategy $(w^{(1)}, w^{(2)}) = (-1, 1)$ – hereafter N_2. We can see from the quantile-quantile plot in Figure 2a that both $\widehat{y}^{N_1}$ and $\widehat{y}^{N_2}$ have the same distribution, but Figures 2b and 2c illustrate the degeneracy described above (N_1 leads to $\widehat{y}^{N_1} = y$ while N_2 leads to $\widehat{y}^{N_2} = -y$). Whether a network yields the first or the second strategy depends on the initialization of its parameters. Since we expect the replication strategies to converge in probability, this is a critical weakness when using W_1.

Overcoming the W_1-degeneracy: Mixing L^2 and W_1. We need the network trained on W_1 to have an appropriate initialization or, at least, to steer towards the "right" path

[2] $w = (S_0 - C_0 + P_0, 1, -1)$ and from (5) $w = (K, 1, -1)$

[3] Due to the self-financing nature of the portfolio, the amount of cash $w_t^{(0)}$ is determined by $w_{t-1}^{(0)}$ and $(w_t^{(1)}, w_t^{(2)})$.

at first *i.e.* toward the lower-right corner on the map in Figure 2. To do so, we simply changed the training procedure by first training networks with the L^2-distance (over 500 backpropagations) and then training networks with the W_1-distance (until convergence). In all cases tested, the networks converged to the analytical solution $(w^{(1)}, w^{(2)}) = (1, -1)$, which solves the main weakness of W_1. Thus, this procedure brings W_1 to the same performance as L^p but does not give it any advantage. The next section explores a complex replication where the qualities of W_1 are fully leveraged.

V. Realistic replication problem: the benefit of using the Wasserstein 1-distance

In the previous section, we showed that the W_1-distance can be used for replication by introducing L^2 optimization at the beginning of the training procedure. Now, we assess the strengths of W_1 compared to L^1 and L^2 for replicating a complex target whose replication strategy does not allow an analytical solution.

Market definition. The financial market under consideration now has a risky asset whose price return s_t follows the Bates model [4]. This model is more sophisticated than the previous one and much closer to the reality of market dynamics. It introduces new difficulties with volatility clustering [8] and occasional large jumps. At $t = 0$, a set of call and put options is available for trading, with expiration date $T = 10$ and fixed discrete strikes $K = K_{\text{min}}, K_{\text{min}} + 1, ..., K_{\text{max}}$. Every T time steps the set of options expires and a new set is issued; at each issuance date the range of strike prices and the intial time to maturity remain the same.

Target definition. In this experiment, the target to be replicated is a *collar index*, *i.e.* a financial product that provides the log-return on an underlying asset (the index) whose extreme values have been clipped beyond a chosen level. More formally, the target is:

$$\forall t > 0,\ y_t = \begin{cases} \min(s_t, q) \text{ if } s_t \geq 0\,, \\ \max(s_t, -q) \text{ otherwise}\,, \end{cases} \tag{6}$$

There is no closed-form solution to this problem in the market under consideration, so we look for numerical solutions. The instruments available for $t \in [(j-1)T, jT]$, $j \in \mathbb{N}^*$, are $X_t = \Big(1, \{C_t(K, jT), P_t(K, jT)\}_{K=K_{min}}^{K_{max}}\Big)$.

Numerical solutions. The solutions are obtained by use of a neural network with parameter set θ, which takes as input, at each time $t \in [(j-1)T, jT]$, $j \in \mathbb{N}^*$, the normalized instrument prices X_t/S_t, the moneyness $\{S_t/K\}_{K=K_{min}}^{K_{max}}$ and the time to maturity $jT - t$. It outputs the vector of quantities associated with each option $(w_t^{(1)}, ...w_t^{(M-1)})$ (as in the first experiment, $w^{(0)}$ is inferred). The replication portfolio returns $\widehat{y}$ are determined according to the Eqs. (1-2). The network consists of four Dense layers of 50 units each with Relu activation and an output Dense layer of 242 units without activation. The total number of parameters is 70 992. We train the networks on W_1, L^2, and L^1 separately and refer to them as the "W_1 network" and the "L^p networks" for convenience. The training procedure is the same as the procedure of the first experiment. The networks are trained on 10^4 backpropagations, and as argued in section IV, the W_1 network is first trained on L_2 (for 10^3 backpropagations) and then trained on W_1 until the end of the training procedure. Training on W_1 is roughly six times longer than training on L^p. All networks reach convergence after 10^4 backpropagation steps, for a total training time of one hour on L^p.

For the numerical computation, we have drawn two data sets for training and testing of $N = 2^{11}$ trajectories of length $L = 110$ of the underlying price S along with the associated option prices with $K_{min} = 40$, $K_{max} = 160$ and $T = 10$. The parameters of the Bates model are set as in [11], which corresponds to the long-term dynamics of the *S&P500* index. We set the threshold level of the target strategy to $q = 2\%$. Given our parameter set, this level cuts about 10% of the returns (5% on each side of the distribution). Five networks are trained separately for each distance; all converge. Figure 3 shows the return distributions obtained by averaging over the different networks (3a) and sample trajectories (3b) for the 3 distances under consideration. Table I provides the first four moments of these distributions as well as the statistics of exceedence over the threshold q.

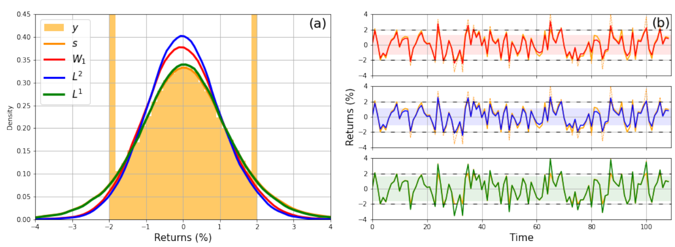

Fig. 3: (a): Kernel density estimates of the returns on the underlying asset S, the target Y and the replication strategies (W_1, L^2 and L^1), (b): Sample trajectories of the target and replication strategy returns obtained by networks trained with the distances W_1, L^2 and L^1. Coloured bands represent plus or minus one times the replication strategy volatility.

	$\overline{r}(\%)$	$\sigma(\%)$	Skew	Kurtosis	$\#\{r < -2\%\}$	$\overline{r} + 2\%$ for $r < -2\%$
s	7.27 (0.98)	21.56 (0.16)	-0.27 (0.03)	5.99 (0.25)	5.88 (0.14)	-0.84 (0.01)
y	8.13 (0.82)	17.92 (0.07)	0.00 (0.00)	2.18 (0.01)	0.00 (0.00)	0.00 (0.00)
W_1	6.58 (0.43)	17.07 (0.03)	0.12 (0.01)	3.57 (0.08)	2.49 (0.03)	-0.43 (0.01)
L^2	6.22 (0.41)	16.16 (0.03)	0.11 (0.01)	3.34 (0.04)	2.09 (0.03)	-0.40 (0.01)
L^1	7.14 (0.52)	20.93 (0.06)	-0.23 (0.01)	5.51 (0.11)	5.50 (0.06)	-0.79 (0.01)
$\overline{W_1}$	6.91 (0.41)	17.92 (0.04)	0.12 (0.01)	3.57 (0.09)	3.10 (0.03)	-0.45 (0.00)
$\overline{L^2}$	6.90 (0.41)	17.92 (0.04)	0.11 (0.01)	3.34 (0.04)	3.30 (0.04)	-0.45 (0.00)

TABLE I: Annualized return statistics and standard errors in parentheses for the risky asset S, the target Y, and the replication portfolios obtained with different networks trained on W_1, L^2, and L^1. $\overline{W_1}$ and $\overline{L^2}$ are statistics of W_1 and L^2 normalized to the target's volatility.

Figure 3 shows that the L^1 network (green line) converges to a solution that is blind to extreme events: the solution replicates the underlying asset (dark orange line) instead of the target (light orange). This is confirmed by the statistics reported in Table I, which show that the values for the target obtained with L^1 and the underlying asset s are nearly the same. In contrast, Figure 3 also shows that the L^2 network (blue line) and the W_1 network (red line) are sensitive to extreme events and, to a large extent, do indeed cut the highest returns. The proportion of returns below the negative threshold for these networks is 2.09% and 2.49% respectively (Table I). Although these values are significantly positive, they are half that of the underlying asset. However, the coloured bands on Figure 3b and the volatility statistics in Table I clearly show that the L^2 network's cut of the largest returns is mainly due to a global volatility reduction. In fact, the L^2 network converged to a solution whose returns have a much lower volatility ($\sigma \approx 16.16\%$) than the target volatility ($\sigma \approx 17.92\%$). Note that the global volatility reduction is the main effect, but not the only one, as the third and fourth order statistics (skewness and kurtosis) are closer to the target statistics for the L^2 network than for the L^1 network. Finally, the W_1 network converged to the most convincing solution, having a very similar cut of extreme events as the L^2 network, but a volatility much closer to the target volatility ($\sigma \approx 17.07$, see Table I). Moreover, if we normalize the returns of the W_1 network and the returns of the L^2 network to the target volatility, we see in Table I that the $\overline{W_1}$ network has a better performance on extreme return cuts than the $\overline{L^2}$ network. Changing the threshold to $q = 3.4\%$ (1% on each side of the distribution) and $q = 1.6\%$ (10% on each side of the distribution) does not qualitatively change these conclusions.

VI. Conclusions

We have shown the interest of using Wasserstein 1-distance to achieve the replication strategy of a complex target, outperforming the results obtained with classical distances such as L^1 and L^2. We expect this result to be confirmed and even strengthened as the difficulty of the replication task increases, especially when using more complex neural network architectures. In an effort toward reproducible research and open science, the Python codes used for this article are made publicly available[4].

References

[1] L. Aksoy, P. Flores, and J. Monteiro. "Exact and approximate algorithms for the filter design optimization problem". In: *IEEE Trans. Signal Process.* 63.1 (2014), pp. 142–154.

[2] M. Arjovsky, S. Chintala, and L. Bottou. "Wasserstein Generative Adversarial Networks". In: *Proc. 34th Int. Conf. Mach. Learn.* 2017, pp. 214–223.

[3] S. Assefa et al. "Generating synthetic data in finance: opportunities, challenges and pitfalls". In: *Proc. First ACM Int. Conf. AI Finance*. 2021, pp. 1–8.

[4] D. Bates. "Jumps and Stochastic Volatility: Exchange Rate Processes Implicit in Deutsche Mark Options". In: *Rev. Financ. Stud.* 9 (1996), pp. 69–107.

[5] T. Beroud et al. "Wassertein Gan Synthesis for Time Series with Complex Temporal Dynamics: Frugal Architectures and Arbitrary Sample-Size Generation". In: *IEEE Int. Conf. Acoust. Speech Signal Process.* 2023.

[6] F. Black and M. Scholes. "The Pricing of Options and Corporate Liabilities". In: *J. Political Econ.* 81.3 (1973), pp. 637–654.

[7] H. Buehler et al. "Deep hedging". In: *Quant. Finance* 19.8 (2019), pp. 1271–1291.

[8] R. Cont. "Empirical properties of asset returns: stylized facts and statistical issues". In: *Quant. Finance* 1.2 (2001), pp. 223–236.

[9] R. Dembo and D. Rosen. "The practice of portfolio replication. A practical overview of forward and inverse problems". In: *Ann. Oper. Res.* 85 (1999), pp. 267–284.

[10] D. Duffie. *Dynamic asset pricing theory*. Princeton University Press, 2010.

[11] B. Eraker, M. Johannes, and N. Polson. "The impact of jumps in volatility and returns". In: *J. Finance* 58.3 (2003), pp. 1269–1300.

[12] I. Goodfellow et al. "Generative adversarial networks". In: *Commun. ACM* 63.11 (2020), pp. 139–144.

[13] M. B. Haugh and A. W. Lo. "Asset allocation and derivatives". In: *Quant. Finance* 1.1 (2001), pp. 45–72.

[14] J. Ho, A. Jain, and P. Abbeel. "Denoising Diffusion Probabilistic Models". In: *Adv. Neural Inf. Process.* Vol. 33. 2020, pp. 6840–6851.

[15] D. Kingma and M. Welling. *Auto-Encoding Variational Bayes*. 2022.

[16] S. Lundbergh and T. Teräsvirta. "Evaluating GARCH models". In: *J. Econom.* 110.2 (2002), pp. 417–435.

[17] Y. Malevergne and D. Sornette. *Extreme financial risks: From dependence to risk management*. Springer, 2006.

[18] A. Papoulis and S. Unnikrishna Pillai. *Probability, random variables and stochastic processes*. McGraw-Hill, 2002.

[19] H. Petzka, A. Fischer, and D. Lukovnicov. "On the regularization of wasserstein gans". In: *Proc. Int. Conf. Learn. Represent.* 2018, pp. 1–24.

[20] T. E. Raghunathan. "Synthetic data". In: *Annu. Rev. Stat. Appl.* 8 (2021), pp. 129–140.

[21] R. Roll. "A mean/variance analysis of tracking error". In: *J. Portf. Manag.* 18.4 (1992), pp. 13–22.

[22] R. Rubinstein, A. Bruckstein, and M. Elad. "Dictionaries for sparse representation modeling". In: *Proc. IEEE* 98.6 (2010), pp. 1045–1057.

[23] I. Tošić and P. Frossard. "Dictionary learning". In: *IEEE Signal Process. Mag.* 28.2 (2011), pp. 27–38.

[24] C. Villani. *Optimal Transport: old and new*. Springer, 2009.

[4] https://github.com/johanmq/deep_learning_synthetic_series