Conditional Loss and Deep Euler Scheme for Time Series Generation

Carl Remlinger[1,2,3,*], **Joseph Mikael**[2,3], **Romuald Elie**[1]

[1]Université Gustave Eiffel, [2]EDF Lab, [3]FiME (Laboratoire de Finance des Marchés de l'Energie)
[*]carl.rlgr@pm.me

Abstract

We introduce three new generative models for time series that are based on Euler discretization of Stochastic Differential Equations (SDEs) and Wasserstein metrics. Two of these methods rely on the adaptation of generative adversarial networks (GANs) to time series. The third algorithm, called Conditional Euler Generator (CEGEN), minimizes a dedicated distance between the transition probability distributions over all time steps. In the context of Itô processes, we provide theoretical guarantees that minimizing this criterion implies accurate estimations of the drift and volatility parameters. Empirically, CEGEN outperforms state-of-the-art and GANs on both marginal and temporal dynamic metrics. Besides, correlation structures are accurately identified in high dimension. When few real data points are available, we verify the effectiveness of CEGEN when combined with transfer learning methods on model-based simulations. Finally, we illustrate the robustness of our methods on various real-world data sets.

Introduction

Time series Monte Carlo simulations are widely used for multiple industrial applications such as investment decisions (Kelliher and Mahoney 2000), stochastic control (Pham 2009) or weather forecasts (Mullen and Baumhefner 1994). They are notably considered in the financial sector, for market stress tests (Sorge 2004), risk management and deep hedging (Buehler et al. 2019; Fecamp, Mikael, and Warin 2020), or for measuring risk indicators such as Value at Risks (Jorion 2000) among others. Providing Monte Carlo simulations representative of the time series of interest is a difficult and mostly manual task, which requires underlying modeling assumptions about the time dependence of the variables. Hence, it is difficult to update these models when a new type of data is observed, such as negative interest rates, negative electricity prices or unusual weather conditions. This naturally calls for the development of reliable model-free data generators for time series.

Generative methods such as Variational Auto Encoders (Kingma and Welling 2013) or Generative Adversarial Networks (GAN) (Goodfellow et al. 2014) provide state-of-the-art accuracy for the generation of realistic images (Xu et al. 2018) or text (Zhang et al. 2017). The development of similar generative methods for time series is very promising (Lyu et al. 2018; Chen et al. 2018). However, due to the complex and possibly non-stationary underlying temporal structure of some time series, these generative methods, especially GANs, are unsatisfactory applied as is (Yoon, Jarrett, and Van der Schaar 2019). Efficient generation of time series requires a proper learning of time-marginals as well as a faithfully representation of the underlying temporal dynamic.

In this paper, time series are represented as a discretized Euler approximation of continuous-time Itô processes. The three proposed generators rely on deep learning approximation of the deterministic drift and volatility functions. This representation benefits from a theoretically grounded temporal dynamic and provides a meaningful structure that avoids complex neural network architectures. Moreover, the considered Euler generators allow tractable, at least controllable, outputs, which can be difficult with deep embedding such as Yoon, Jarrett, and Van der Schaar (2019). This feature is a key component in industrial applications, especially for decision-making-process.

By combining deep Euler representation with Wasserstein distance (Villani 2008), we introduce the Euler Wasserstein GAN (EWGAN), inspired by (Arjovsky, Chintala, and Bottou 2017). Our second GAN-based-model, called Euler Dual Discriminator (EDGAN) is an adaptation of the DVDGAN presented in Clark, Donahue, and Simonyan (2019). A spatial discriminator focuses on the accuracy of time-marginal distributions, while a temporal one focuses on the full sequence. Nevertheless, all these GAN approaches still have difficulties to capture a proper temporal dynamic.We remedy to this problem by introducing the Conditional Euler Generator (CEGEN) which optimizes a distance between the transition probability distributions at each time step. On the (large) class of Itô processes, we prove that minimizing this metric provides an accurate estimation of both the drift and volatility parameters.

A numerical study compares the three approaches with state-of-the-art GANs on synthetic and real data sets. We first verify that our methods can learn to replicate Monte Carlo simulations of classical stochastic processes. Synthetic models give access to more reliable metrics (including theoretical), and allow to make connections between model-based and model-free approaches. EWGAN and EDGAN show a similar accuracy as state-of-the-art GANs

but capture more efficiently the time dynamics in dimension up to 20. The best performing model, CEGEN, recovers the underlying correlation (or independence) structure of time series particularly well, even in high dimensions. On real data, CEGEN outperforms every other GAN-based methods on five quantitative metrics. Moreover, we highlight the robustness of CEGEN, when combined with a transfer learning procedure when too few data are available. By properly mixing model-based simulations with sparse real data during training, the generator can take advantage of the prior from synthetic samples to improve its accuracy.

Main Contributions:

- A theoretically grounded time series generator combining an Euler discretization of Itô processes with a dedicated loss on conditional distributions (CEGEN) is proposed. The conditional distance ensures that the generator learns the distribution around each data point and the temporal dependence.
- Relying on a similar Itô process restriction, we also introduce two Euler GANs alternatives inspired by (Arjovsky, Chintala, and Bottou 2017) and (Clark, Donahue, and Simonyan 2019).
- A thorough numerical study on synthetic and various real world data sets demonstrate the robustness of our generators. CEGEN outperforms the other considered methods on five distinct metrics. A transfer learning application when sparse data is available is provided. Euler GANs exhibit close performance to the state-of-the-art GANs on marginal metrics, but capture more accurately the correlation and temporal structure on Itô processes.

Related Works

The bootstrap method (Efron 1982) is one of the first purely data-driven attempt to generate data. Samples are simply taken randomly with replacement. The scope of this technique is limited as it does not generate additional information. On the opposite, model-free approaches such as GAN allow to learn empirical distribution from data and thus to generate new samples. However, initial GAN proposals focus on the generation of non-temporally ordered outputs. GAN's architecture improvement for the time series case is an intensive area of research. For instance, WaveGAN (Donahue, McAuley, and Puckette 2018) uses the causal architecture of WaveNet (Oord et al. 2016) for unsupervised synthesis of raw-waveform audio. Alternatively, several works consider recurrent neural networks to generate data sequentially and keep memory of the previous states (Mogren 2016; Esteban, Hyland, and Rätsch 2017).

Time Series GAN (TSGAN, Yoon, Jarrett, and Van der Schaar (2019)) introduces a state-of-the-art method for time series generation which stands out by its specific learning process. At each time step, an embedding network projects the sequences onto a latent space on which a GAN operates. TSGAN manages to get accurate distributions and correlation on classical processes, we use it as a baseline in this paper. This method makes theoretical analysis of the generator outputs difficult due to its specific embedding. As the usage of model-free methods grows rapidly, their application to sensitive fields (e.g. finance) must be considered cautiously and requires theoretical and empirical guarantees on the behavior of these generators. For this purpose, an active line of research looks towards reliable embedding of time series, such as signature (Fermanian 2019; Buehler et al. 2019) or Fourier representation (Steinerberger 2018).

Most recent applications on video generation focus on specific GAN architectures to capture the spatial-temporal dynamics. For instance, MoCoGAN (Tulyakov et al. 2018) and DVD GAN (Clark, Donahue, and Simonyan 2019) combine two discriminators, one for the temporal dynamic and another one on each static frame. Specialized generator structures have also been designed, TGAN (Saito, Matsumoto, and Saito 2017) proposed to generate a dynamic latent space and VGAN (Vondrick, Pirsiavash, and Torralba 2016) combines two generators, one for marginals and another one for temporal dependencies. Following the idea of applying optimal transport to GANs (Arjovsky, Chintala, and Bottou 2017; Genevay, Peyré, and Cuturi 2018), COTGAN (Xu et al. 2020) uses causal optimal transport (COT) for video sequence generation. To do so, the discriminant penalizes not-causal transport plans, ensuring that the generator minimizes an adapted (regularized) Wasserstein distance for time series. This approach benefits of solid theoretical foundations but still lacks of empirical success on noisy time series.

A very recent approach (Kidger et al. 2021) proposes as well to use Stochastic Differential Equation (SDE) formulation for time series. By combing a neural SDE and a neural CDE (Controlled Differential Equation) in a GAN setup, the authors show that the classical approach to fit SDEs may be generalized. Our generators do not use any of these neural SDE and our discriminators do not aim to solve SDE.

Problem Formulation

We aspire to design a time series generator which combines accurate estimation of time-marginal distributions while properly capturing temporal dynamics. The generator we propose is designed to be simple enough to be tractable (in the sense that outputs could be controlled) and theoretically grounded. To do so, we feed our algorithms with training time series data and seek to learn an empirical probability distribution that best approximates the data one. This task can be tricky, depending on the sequences lengths, the dimension, and the shape of the data distribution.

Although the idea of a model-free approach is attractive, we restrict ourselves to the context of Itô processes. This class of processes encompasses a wide range of time series and yet allows us to develop tractable models based on theory. In addition to providing a robust theoretical framework and controls on the generation, Itô processes allow to measure the accuracy of our generators on synthetic samples via closed form expressions or Monte Carlo simulators. In comparison to common literature (Wiese et al. 2020; Buehler et al. 2020), we do not assume the time series to be stationary and allow ourselves to consider not-stationary sequences.

Itô Process A time series observed on a time grid $\mathcal{T} = \{0 = t_0 < t_1 < ... < t_N = T\}$. For the sake of simplicity, we assume a regular time grid with mesh size Δt. We are given i.i.d. samples of a random vector $X = (X_{t_i})_{t_i \in \mathcal{T}}$ on $\mathbb{R}^{d\times(N+1)}$, $N \in \mathbb{N}^*$, starting from X_0 valued in $\mathbb{R}^d$. The discrete time samples are supposed to be drawn from a continuous time underlying process having the following Itô dynamics:

$$dX_t \quad = \quad b_X(t, X_t)dt + \sigma_X(t, X_t)dW_t, \qquad (1)$$

where $b_X : \mathbb{R}\times\mathbb{R}^d \to \mathbb{R}^d$ is the drift, $\sigma_X : \mathbb{R}\times\mathbb{R}^d \to \mathbb{M}_{d\times d}$ the volatility and W is a d-dimensional Brownian motion on some probability space $(\Omega, \mathcal{F}, \mathbb{P})$ equipped with a filtration $(\mathcal{F}_t)_{t\in[0,T]}$ representing the information available at time t. The parameters b_X and σ_X are supposed to satisfy the usual Lipschitz conditions (Ikeda and Watanabe 2014) ensuring existence and uniqueness of the solution of Eq.(1).

Deep Euler Representation Samples of X can be viewed as samples drawn from the Euler discretization scheme of (1) given by

$$X_{t_i+\Delta t} \quad = \quad X_{t_i} + b_X(t_i, X_{t_i})\Delta t + \sigma_X(t_i, X_{t_i})\Delta W_{t_i},$$

where $(\Delta W_{t_i})_{t_i\in\mathcal{T}}$ is a collection of i.i.d. $\mathcal{N}(0, \Delta t I_d)$ random variables. Relying on this scheme, we introduce the deep Euler representation. Starting at $t_0 = 0$, from $Y_0^\theta = X_0$ we generate time series $Y^\theta = (Y_{t_i}^\theta)_{t_i\in\mathcal{T}}$ by the following scheme:

$$Y_{t_i+\Delta t}^\theta = Y_{t_i}^\theta + b_Y^\theta(t_i, Y_{t_i}^\theta)\Delta t + \sigma_Y^\theta(t_i, Y_{t_i}^\theta)Z_{t_i}, \qquad (2)$$

where Z_{t_i} are $\mathcal{N}(0, \Delta t I_d)$ i.i.d. random variables. b_Y^θ and σ_Y^θ are θ-parametrized functions approximated by a neural network. Our objective is to learn b_Y^θ and σ_Y^θ, so that the distributions of the processes Y^θ and X are close.

Evaluation During the learning phase, neither b_X nor σ_X are given as inputs to any of the proposed generator. However, this formulation allows to compare *a posteriori* b_X and σ_X, when they are known, to the estimated b_Y^θ and σ_Y^θ. This provides a reliable metric on the generation accuracy.
Moreover, this setup provides a convenient way to control the drift b_Y^θ and volatility σ_Y^θ functions. This task is delicate with deep embedding proposals for instance.

Euler Generators

Euler Generators proposed in this paper rely on two main elements: a network generating the drift and volatility of an Itô process and a distance between distributions to be minimized. The Itô structure facilitates the time series construction, while the distance focuses on the probability law accuracy of the generated sequences. Both GAN-based and Conditional loss methods described hereafter share this design.

Euler Generative Adversarial Networks

Among various generative models, GAN (Goodfellow et al. 2014) stands out by its specific optimization process. The adversarial training defined by a zero-sum game between a discriminator and a generator allows to implicitly learn data distributions while preventing over-fitting. The Wasserstein GAN (Arjovsky, Chintala, and Bottou 2017) seems to get rid of stability problems encountered in training (mainly mode collapse) by adapting to the geometry of the underlying space. We propose two adaptations of GANs to time series that are based on the deep Euler representation presented in Eq.(2) and on the (differentiable) Wasserstein-1 ($\mathcal{W}_1$) distance. The Rubinstein-Kantorovich duality allows to rewrite $\mathcal{W}_1$ between two random variables Z_1 and Z_2 as follow:

$$\begin{aligned}\mathcal{W}_1(\mathcal{L}(Z_1), \mathcal{L}(Z_2)) \quad &= \quad \sup_{||f||_L\leq 1} \mathbb{E}_{Z_1\sim\mathcal{L}(Z_1)}\left[f(Z_1)\right] \\ &- \quad \mathbb{E}_{Z_2\sim\mathcal{L}(Z_2)}\left[f(Z_2)\right],\end{aligned}$$

where $||f||_L$ denotes the smallest Lipschitz constant of the real-valued function f.

Euler Wasserstein GAN (EWGAN) This model considers a Wasserstein GAN, where the generator relies on the Deep Euler representation (2) and minimizes according to parameter θ the $\mathcal{W}_1$ distance between the distribution of $X = (X_{t_i})_{t_i\in\mathcal{T}}$ and the generated one from $Y^\theta = (Y_{t_i}^\theta)_{t_i\in\mathcal{T}}$ (Eq.(3)). The discriminator d_φ parametrized by φ tries to find the optimal 1-Lipschitz function allowing to compute $\mathcal{W}_1(\mathcal{L}(X), \mathcal{L}(Y^\theta))$ using the Rubinstein-Kantorovich duality. The Lipschitz property of d_φ is guaranteed using the gradient penalty trick mentioned in Gulrajani et al. (2017).

$$\begin{aligned}\inf_\theta \mathcal{W}_1(\mathcal{L}(X), \mathcal{L}(Y^\theta)) \quad &= \quad \inf_\theta \sup_\varphi \mathbb{E}_{X\sim\mathcal{L}(X)}\left[d_\varphi(X)\right] \\ &- \quad \mathbb{E}_{Y^\theta\sim\mathcal{L}(Y^\theta)}\left[d_\varphi(Y^\theta)\right]. \qquad (3)\end{aligned}$$

Pseudocode of EWGAN is given in Alg.2 in Appendix C.

Euler Dual Discriminator (EDGAN) Our second GAN-based model is an adaptation of the Dual Video Discriminator GAN (Clark, Donahue, and Simonyan 2019). DVD GAN uses attention networks and two discriminators in order to generate high fidelity videos. While the spatial discriminator focuses on time marginals and critics images in high resolution, the temporal one considers the full sequence of frames in low resolution. We adapt these ideas to our context by considering in EDGAN a similar dual discriminator architecture while the generator creates samples using the Deep Euler representation of Eq.(2). The temporal discriminator is similar as the one of EWGAN and focuses on $\mathcal{W}_1(\mathcal{L}(X), \mathcal{L}(Y^\theta))$. At the same time, the marginal discriminator focuses on the computation of the $\mathcal{W}_1$ distance between marginal distributions $\mathcal{W}_1(\mathcal{L}(X_t), \mathcal{L}(Y_t^\theta))$, for each $t_i \in \mathcal{T}$. Pseudocode is given in Alg.3 in Appendix C.

Conditional Loss Method

As already mentioned, it is highly challenging for generators of temporally ordered data to capture the temporal dynamic (Yoon, Jarrett, and Van der Schaar 2019). In order to remedy to this weakness, we introduce below an innovative loss function based on conditional distributions.

A Conditional Loss on Distributions The difficulty arising when trying to design a loss function for a data-driven time series generator comes from the need to get the correct balance between the marginal distribution fitness and

Algorithm 1: CEGEN

Input: $\mathcal{D}$ samples of X, m batch size, K Nb of subdivisions, γ learning rate
Initialize: θ (randomly picked)
while Not converged **do**
  for $i = 0 \dots N-1$ **do**
    Sample m observations $x_{t_{i+1}}$ from of $X_{t_{i+1}}$
    Sample $z \sim \mathcal{N}(0, I_d \Delta t)$
    $y_{t_{i+1}} \leftarrow y_{t_i} + g_\theta^b(t_i, y_{t_i})\Delta t + g_\theta^\sigma(t_i, y_{t_i})z$
    $I_J \leftarrow J$ subdivisions of Supp$(X_{t_i}) \cup$ Supp(Y_{t_i})
    for $j = 1 \dots J$ **do**
      $\ell_{t_{i+1},j} \leftarrow \mathcal{W}_2^2(\mathcal{L}(x_{t_{i+1}}|x_{t_i} \in I_j), \mathcal{L}(y_{t_i+1}|y_{t_i} \in I_j))$
    end for
  end for
  $\theta = \theta - \gamma \nabla_\theta \sum_{i=0}^{N-1} \sum_{j=1}^{J} \ell_{t_{i+1},j}$
end while
Output: $y = (y_{t_i})_{i \in \{0,\dots,N\}}$

the good representation of the temporal structure. On the one hand, we cannot only focus on marginals because having $\mathcal{L}(X_{t_i}) \sim \mathcal{L}(Y_{t_i}^\theta)$ for all $t_i \in \mathcal{T}$ does not imply that $b_X = b_{Y^\theta}$ nor that $\sigma_X = \sigma_{Y^\theta}$ (see the counterexample in Appendix A). On the other hand, instead of working on marginals, one can wonder if considering time series realization as a vector defined on $\mathbb{R}^{d\times(N+1)}$ provides better results. Unfortunately, and as mentioned in Yoon, Jarrett, and Van der Schaar (2019), learning the joint distribution $\mathcal{L}(X_{t_0}, \dots, X_{t_N})$ may not be sufficient to guarantee that the network captures the temporal dynamics, even with memory-based networks. An empirical example of unsatisfactory generations based on joint laws is illustrated in Figure 4 in Appendix, the generated trajectories are smooth. In the case of time series, one should simply refrain from applying a loss based only on marginal or joint distributions. To provide a reliable solution to this issue, we propose to focus on the transition probabilities at each time step by conditioning on the previous state. Moreover, by doing so, we are able to produce theoretical bounds on Itô coefficient estimation accuracy.

CEGEN Algorithm Contrarily to the previous GAN-based methods, CEGEN does not require a discriminator network. The idea consists in considering a loss function that compares the conditional distributions $\mathcal{L}(Y_{t_{i+1}}^\theta \,|\, Y_{t_i}^\theta)$ with $\mathcal{L}(X_{t_{i+1}} \,|\, X_{t_i})$ for each time step $t_i \in \mathcal{T}$. The latter conditional distributions are Gaussian when considering Euler-discretized Itô processes. We consider the following metric:

$$\begin{aligned} \mathcal{W}_2^2(\mathcal{L}(X), \mathcal{L}(Y)) &= \|\mathbb{E}[X] - \mathbb{E}[Y]\|_2^2 \\ &+ \mathcal{B}^2(Var(X), Var(Y)), \quad (4) \end{aligned}$$

where $\mathcal{B}$ is the Bures metrics (Bhatia, Jain, and Lim 2019; Malago, Montrucchio, and Pistone 2018) defined by $\mathcal{B}^2(A, B) = Tr(A) + Tr(B) - 2Tr(A^{\frac{1}{2}} B A^{\frac{1}{2}})^{\frac{1}{2}}$, for positive definite matrices A and B. If X and Y are gaussian, W_2 is the definition of the Wasserstein-2 distance (Gelbrich 1990). This metric (4) captures meaningful geometric features between distributions, and $\mathcal{W}_2$ transportation plan is very sensitive to the outliers thus increases the distribution estimation accuracy. The Bures formulation allows us to consider exactly the Wasserstein-2 distance, instead of regularized ones (Genevay, Peyré, and Cuturi 2018; Cuturi 2013). The benefit is double, its value as well as its gradients admit closed forms, and can handle degenerate measures (Muzellec and Cuturi 2018). Moreover, the Bures metric allows us to provide theoretically guarantees that minimizing the conditional loss implies accurate estimation for the drift and volatility coefficients. Indeed, whenever the distributions of the form $\mathcal{L}(X_{t_{i+1}} \,|\, X_{t_i} = z)$ and $\mathcal{L}(Y_{t_{i+1}}^\theta \,|\, Y_{t_i}^\theta = z)$ for $z \in \mathbb{R}^d$ coincide in $\mathcal{W}_2$, the process parameters coincide as well (see Prop. A.1 in Appendix). This is encouraging but in general conditioning from the very same point is complicated. Proposition 1 extends this property when the previous states belong to a small ball around z. To compute the loss, we create at each time t_i a partition $(I_j)_{1\le j\le J}$ of the union of supports of X_{t_i} and $Y_{t_i}^\theta$. For a given batch of samples, $\mathcal{L}(X_{t_{i+1}} \,|\, X_{t_i} \in I_j)$ is approximated by extracting the elements $X_{t_{i+1}}$ such that $X_{t_i} \in I_j$. $\mathcal{L}(Y_{t_{i+1}}^\theta \,|\, Y_{t_i}^\theta \in I_j)$ is approximated in the same way. The $\mathcal{W}_2^2$ distance between conditional distributions are then summed up over all subdivisions and over all time steps:

$$l(X, Y^\theta) = \sum_{i=0}^{N-1} \sum_{j=1}^{J} \mathcal{W}_2^2(\mathcal{L}(X_{t_{i+1}}|X_{t_i} \in I_j), \mathcal{L}(Y_{t_{i+1}}^\theta|Y_{t_i}^\theta \in I_j)).$$

By computing $\mathcal{W}_2^2$ between the elements such that the previous states are close to each other, i.e. belonging to the same ensemble, the generator is able to learn the distribution around data points and ensures the temporal dependence. The pseudocode of CEGEN is given in Alg.1 and details are provided in Appendix D. Bures metrics is computed using the Newton-Schulz method (Muzellec and Cuturi 2018), which is a differentiable way to get covariance matrice square roots.

Theoretical Guarantee In order to theoretically ground the choice of the loss function, we need to quantify how reducing the $\mathcal{W}_2$ distance implies proximity between drift and volatility parameters. The following result is allowed by the specific Bures-Wasserstein formulation (Eq.4) implemented in CEGEN.

Proposition 1. *Let $t_i \in \mathcal{T}$. Assume that $\sigma_X^2(t_i, .)$, $\sigma_Y^2(t_i, .)$ are strictly positive and, together with $b_X(t_i, .)$ and $b_Y(t_i, .)$, are K-Lipschitz in their second coordinate. Let $(I_j)_{1\le j\le J}$ be a regular partition covering Supp$(X_{t_i}) \cup$ Supp(Y_{t_i}) with mesh size Δx. Let $\varepsilon > 0$.*
If $\mathcal{W}_2^2\left(\mathcal{L}(X_{t_{i+1}}|X_{t_i} \in I_j), \mathcal{L}(Y_{t_{i+1}}|Y_{t_i} \in I_j)\right) \le \varepsilon$ for any j, then, for z in I_j

$$\|b_X(t_i, z) - b_Y(t_i, z)\|_2 \le \frac{\sqrt{\varepsilon} + \Delta x}{\Delta t} + 2K\Delta x.$$

Furthermore,

$$\|\sigma_X(t_i, z) - \sigma_Y(t_i, z)\|_2 \le \begin{cases} \sqrt{\frac{\varepsilon}{\Delta t}} + 2K\Delta x & \text{if } d = 1 \\ \sqrt{\frac{2\alpha^2\varepsilon}{\Delta t}} + 2K\Delta x & \text{if } d > 1 \end{cases},$$

where $Tr\big(\sigma_X^2(t_i, z)\big) = Tr\big(\sigma_Y^2(t_i, z)\big) = \alpha$.

As described in Appendix A, this result is proved using useful inequalities between Hellinger and Bures distances.

The α coefficient comes from the need of using density matrices, in practice one can easily normalize covariance matrices by their traces. Proposition 1 implies that by conditioning over sufficiently small intervals, a low $\mathcal{W}_2$ loss between transition distributions guarantees a good process estimation.

Numerical Study

EWGAN, EDGAN and CEGEN are compared numerically to the state-of-the-art TSGAN and COTGAN. Two additional benchmarks are provided, but in Appendix due to poor results: RCGAN (Esteban, Hyland, and Rätsch 2017) a conditional GAN based on recurrent neural networks and GMMN (Li, Swersky, and Zemel 2015) an unconditional MMD with Gaussian kernel. Networks are composed of 3-layers of 4 times the data dimension neurons each. Euler generator networks are feed-forward, while benchmarks benefit of recurrent networks (LSTMs). Hyper-parameters are described in Appendix D.

Two kinds of data sets are used: synthetic and real time series. In single dimension, we use Black-Scholes (BS) model ($dX_t = rX_t dt + \sigma X_t dW_t$) and Ornstein-Uhlenbeck (OU) model ($dX_t = \theta(\mu - X_t)dt + \sigma dW_t$). For these two stochastic models, our empirical references are drawn from Monte Carlo simulations. The latter are performed on a regular time grid of 30 dates, the maturity is 0.25 (1 simulation per day for 3 months) and $X_0 = 0.2$. BS model (resp. OU) has coefficients of $r = 0.8$, $\sigma = 0.3$ (resp. $\sigma = 0.1, \mu = 0.6$, $\theta = 7$). In higher dimensions, we proceed with the same methodology but with multivariate correlated BS time series (d = 4, 10, 20). The real data sets include various nature of time series and are detailed in Appendix F.

Evaluation Metrics

We consider five metrics to evaluate the accuracy of the generators. For all metrics the lower, the better.

(1) Marginal Metrics. These metrics quantify the quality at each time step of the marginal distributions induced by the generated samples compared to the reference ones. This includes **Fréchet Distance (FD)** (Fréchet 1957; Dowson and Landau 1982) as well as classical statistics (mean, 95% and 5% percentiles denoted respectively Avg, q95, q05). We compute the mean squared error (MSE) over time of these statistics. This helps measuring whether a generator manages to get an accurate overall envelope of the processes.

(2) Temporal Dynamics. This metric aims at quantifying how the generator is able to capture the underlying time structure of the signal. The difference between the quadratic variations of both reference and generated time series is computed. The **quadratic variation (QVar)** of an Itô process X is given by $[X]_t = \int_0^t \sigma_X^2(s, X_s)ds$. Thus, the temporal metric ensures that the diffusion σ_X is well estimated. In the discrete case, $[X]_t$ is obtained with $\sum_i |X_{t_{i+1}} - X_{t_i}|^2$.

(3) Correlation Structure. The metric denoted **Corr** is the term-by-term MSE between empirical correlation from reference samples on one side and from generated samples on the other side. It evaluates the ability of a generator to capture the multi-dimensional structure of the signal.

	CEGEN	EWGAN	EDGAN
Black-Scholes			
$\hat{r}$ (0.8)	**0.739**	0.581	0.996
$\hat{\sigma}$ (0.3)	0.324	**0.314**	0.379
Ornstein-Uhlenbeck			
θ (7.0)	**7.05**	4.36	4.68
$\hat{\mu}$ (0.6)	**0.60**	0.75	0.72
$\hat{\sigma}$ (0.1)	**0.11**	0.16	0.02

Table 1: Exp.A. Model parameter estimations.

(4) Process Parameters. A by-product output of Euler generators are the estimated drift $b_Y^\theta(.)$ and volatility $\sigma_Y^\theta(.)$ functions. When using synthetic data, we can compare them with the true process parameters. In the BS case, the coefficients are estimated by the empirical average of $(b_Y^\theta(t, Y_t^\theta)/Y_t^\theta)_{t\in\mathcal{T}}$ and $(\sigma_Y^\theta(t, Y_t^\theta)/Y_t^\theta)_{t\in\mathcal{T}}$. For OU, σ is estimated in a similar manner, while θ and μ are estimated by regressing b_Y^θ on (t, Y_t^θ). These statistics cannot be computed in the same way with TSGAN due to its specific deep embedding, nor COTGAN.

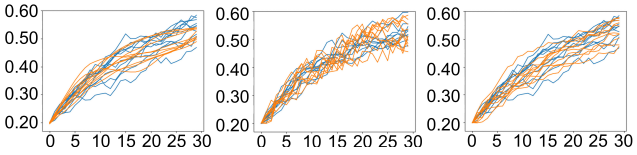
(5) Discriminative and Predictive Scores. We use two distinct scores, as proposed in Yoon, Jarrett, and Van der Schaar (2019). First, we train a classification model (a 2-layer LSTM) to distinguish real sequences from the generated ones. The accuracy of the classifier provides the discriminative score. Second, the predictive score is obtained by training a sequence-prediction model (a 2-layer LSTM) on generated time series to predict the next time step value over each input sequence. The performance is measured then by evaluating the trained model on the original data, in term of MSE.

One-Dimensional Simulated Process (Exp. A)

We start comparing the generators on unidimensional synthetic data. Figure 1 illustrates how crucial is the balance between the estimation of the marginal distributions and the temporal structure. On the one hand, the trend and marginals of generations seem close to the empirical reference. On the other hand, GANs struggle to capture the temporal dynamics between two time steps. COTGAN tends to smooth the time series while TSGAN outputs too noisy samples. However, CEGEN manages to capture the overall envelope and is able to fit the dynamics of time series. These results are confirmed quantitatively by the QVar metric in Table 5 (in Appendix B). Table 1 reports the reference drift and volatility coefficients with those obtained by the three Euler generators. CEGEN provides an accurate estimation and to a lesser extent of EWGAN. Despite benefiting of a dedicated temporal discriminator, EDGAN returns the poorest estimation. Euler structure alone does not manage to recover the right parameter values. Regarding the overall dynamics and the marginals, CEGEN seems a reliable generator for time series.

Scaling the Dimension (Exp. B)

The question we address now is how CEGEN scales to higher dimensions. Table 2 reports the discrepancies between reference empirical correlation and generated time series corre-



(a) COTGAN (b) TSGAN (c) CEGEN

Figure 1: Exp.A. Ornstein-Uhlenbeck samples (in blue) with COTGAN, TSGAN and CEGEN generations (in orange).

Dim	CEGEN	EWGAN	EDGAN	TSGAN	COTGAN
4	**.007**	.015	.053	.177	.031
10	**.011**	.055	.022	.259	.035
20	**.006**	.034	.014	.481	.019

Table 2: Exp.B. MSE between reference and generator empirical correlation matrices on Black-Scholes.

lation for dimensions $d = 4, 10, 20$ on BS simulations. Up to dimension 20, CEGEN obtains a significant improvement compared to every GANs. Figure 2 illustrates how well CEGEN outperforms the other generators with respect to the FD and QVar in higher dimensions. This is confirmed by statistics on drift and volatility of Table 7 in Appendix B, where one can also find an illustration of the process envelopes of CEGEN for the $d = 20$ case (Figure 5). These good global performances encourage us to focus on the conditional generator in the following transfer learning section.

Transfer Learning for Small Data Sets (Exp. C)

Deep generators may need more data than available to be trained effectively. As is done in transfer learning (Torrey and Shavlik 2010), we propose to start the training with a reasonably wrong probabilistic model and finish up with a few real data samples. This situation is tested on synthetic data to allows us to track the drift and volatility learning during the training phase. Both target and misspecified samples come from Monte Carlo simulations, but target sequences include only 60 trajectories of 30 dates (5 years of monthly measures). The first phase of transfer learning includes new simulated samples from the misspecified model each iteration, while the second phase loops randomly with only the 60 samples of the target model. The reference (resp. misspecified) parameters are $\sigma^* = 0.15$, $\mu^* = 0.6$, $\theta^* = 2.0$ (resp.

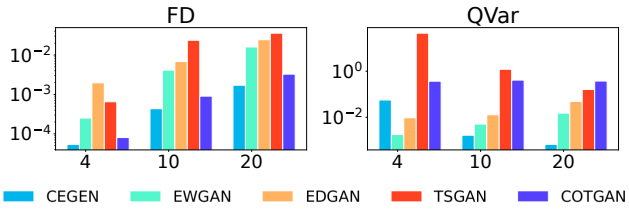


Figure 2: Exp.B. Left: Average of Fréchet Distance between distributions at each time step. Right: Difference between quadratic variations. Ordinate axis is Log scale. Both scores are provided for $d = 4, 10, 20$.

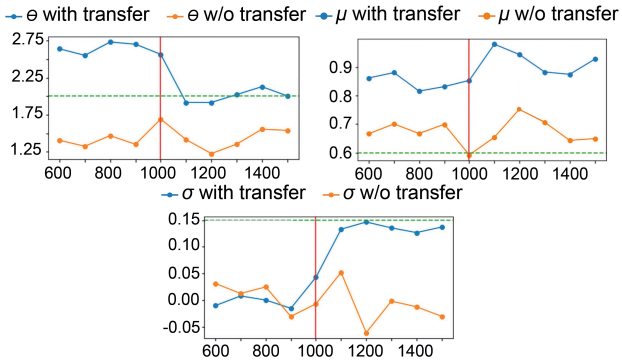


Figure 3: Exp.C. Evolution of parameter estimations during training when a transfer occurs at iteration 1000 (red lines). Dashed green lines correspond to the theoretical target values. Orange lines indicate coefficient estimations of CEGEN only trained on few data and blue lines CEGEN with transfer.

$\sigma = 0.1$, $\mu = 0.8$, $\theta = 3.0$). A CEGEN with transfer is compared with a CEGEN only trained with the few available target sequences. Figure 3 provides the coefficient evolution of both generators during the training process. The transfer iteration start is represented by the red vertical line. Firstly trained with the mis-calibrated OU model, the transfer learning approach is able to retrieve the parameters when fed with few samples of the target model. The generator only trained with few real samples is unable to estimate correctly the θ and σ coefficients, but exhibits a better estimation of μ. Table 8 in Appendix confirms quantitatively the process estimation comparison. The model benefiting from the transfer takes advantage of the initial training phase and provides an overall better estimation.

This framework is a way to update an existing model with the help of incoming real data. Transfer learning tests show how model-free methods can rely on a proven simulation model without replacing it completely.

Experiments on Real-world Data (Exp. D)

Finally, we test generators on various real time series (depending on the dimension, periodicity or noise). In Table 3, we evaluate models with the help of FD, QVar and Corr. CEGEN outperforms GANs or is close in term of FD, and captures well the correlation structure of the signals. However, some QVar from TSGAN or COTGAN are lower than CEGEN despite their generated trajectories being significantly smoother than real data. To better evaluate the fidelity of the generation we need to consider other metrics. Table 4 reports discriminative and predictive scores for each model (except EWGAN, RCGAN and GMMN where scores can be found in Appendix as we focus on state-of-the-art models). Our conditional generator almost consistently produces higher-quality time series in comparison to the benchmarks. On Electric Load data, COTGAN is able to better capture seasonality of the times series, but generates too smooth trajectories. In the opposite, CEGEN proposes more faithful times series in term of noise, but struggles to fool the classifier.

Data	CEGEN FD	CEGEN QVar	CEGEN Corr	EDGAN FD	EDGAN QVar	EDGAN Corr
Spot prices (d=2)	1.38e-04	2.09e+00	**2.10e-02**	3.11e-03	2.18e+00	4.12e-02
Stocks (d=6)	**1.04e-04**	2.10e+01	2.33e-03	7.93e-03	2.43e+01	9.78e-03
Electric Load (d=12)	6.47e-03	4.30e+00	**1.27e-03**	4.62e-02	1.27e+00	1.56e-03
Jena climate (d=15)	**1.10e-03**	**7.18e+00**	**1.75e-02**	4.39e-02	7.73e+00	1.46e-01
Data	TSGAN FD	TSGAN QVar	TSGAN Corr	COTGAN FD	COTGAN QVar	COTGAN Corr
Spot prices (d=2)	2.12e-04	**9.00e-02**	4.45e-02	**1.09e-04**	8.25e-01	4.15e-02
Stocks (d=6)	3.46e-03	2.19e+01	2.76e-01	1.49e-04	**1.86e+01**	**1.62e-03**
Electric Load (d=12)	**5.12e-03**	**9.18e-01**	1.87e-03	4.10e-01	3.45e+00	6.27e-01
Jena climate (d=15)	4.07e-03	8.49e+01	1.89e-02	4.48e-03	7.90e+00	2.34e-02

Table 3: Exp.D. Accuracy evaluations for generations on real time series (the lower, the better).

Data	CEGEN Disc	CEGEN Pred	EDGAN Disc	EDGAN Pred	TSGAN Disc	TSGAN Pred	COTGAN Disc	COTGAN Pred
Spot prices (d=2)	**.014**	**.049**	.137	**.049**	.066	.055	.033	**.049**
Stocks (d=6)	**.079**	**.040**	.429	.041	.159	.041	.116	.041
Electric Load (d=12)	.433	.028	.495	.046	.407	.032	**.277**	**.022**
Jena climate (d=15)	**.140**	**.032**	.483	.035	.179	**.032**	.227	.042

Table 4: Exp.D. Discriminative and predictive scores on real time series (the lower, the better).

Conclusion

We introduced three generative methods for times series, relying on a Deep Euler scheme of Itô processes. Considering Itô structure is a compromise between an intuitive representation and a large class of processes to help generators. Two methods EWGAN and EDGAN demonstrate an accuracy close to state-of-the-art GANs. The third method CEGEN computes a distance between the conditional distributions of the time series. The generator is thus able to learn the distribution around data points and ensures the link between time-dependent states. We prove that minimizing this loss guarantees a proper estimation of the drift and volatility coefficients of the Itô process. Our experiments on synthetic and real-world data sets demonstrate that CEGEN outperforms the other generators on marginal and temporal dynamics metrics. CEGEN is able to capture correlation structures in high dimensions and is efficient when combined with transfer learning on sparse data sets. Transfer learning tests show how this type of methods can combine model-based simulations with a data-driven approach. In further work, we plan to consider more specialized neural networks architectures for time series, extend our results to more general Lévy processes which may include jumps, and consider not Gaussian noise.

Broader Impact

Generative methods for time series may be involved in industries using stochastic control and simulation methods making them of particular interest in physics, finance or for energy companies. When applied within a decision-making process, generative methods has to be used carefully as a failure during learning phase may lead to damageable consequences. Thus, the outputs of the generators should not be left free, as this could lead to erratic optimal controls. Contrarily to the existing approaches which applies GANs and embedding to generate any kind of time series, we impose an Euler structure and we restrain ourselves within the (sufficiently) large class of Itô processes. Moreover, theoretical results give an error estimate of the process parameters for a given loss level.

Acknowledgements

We would like to thank the reviewers for their useful remarks.

References

Arjovsky, M.; Chintala, S.; and Bottou, L. 2017. Wasserstein generative adversarial networks. In *International conference on machine learning*, 214–223. PMLR.

Bhatia, R.; Jain, T.; and Lim, Y. 2019. On the Bures–Wasserstein distance between positive definite matrices. *Expositiones Mathematicae*, 37(2): 165–191.

Buehler, H.; Gonon, L.; Teichmann, J.; and Wood, B. 2019. Deep hedging. *Quantitative Finance*, 19(8): 1271–1291.

Buehler, H.; Horvath, B.; Lyons, T.; Perez Arribas, I.; and Wood, B. 2020. A data-driven market simulator for small data environments. *Available at SSRN 3632431*.

Chen, Y.; Wang, Y.; Kirschen, D.; and Zhang, B. 2018. Model-free renewable scenario generation using generative adversarial networks. *IEEE Transactions on Power Systems*, 33(3): 3265–3275.

Clark, A.; Donahue, J.; and Simonyan, K. 2019. Efficient Video Generation on Complex Datasets. *CoRR*, abs/1907.06571.

Cuturi, M. 2013. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26: 2292–2300.

Donahue, C.; McAuley, J.; and Puckette, M. 2018. Adversarial audio synthesis. *arXiv preprint arXiv:1802.04208*.

Dowson, D.; and Landau, B. 1982. The Fréchet distance between multivariate normal distributions. *Journal of multivariate analysis*, 12(3): 450–455.

Efron, B. 1982. *The jackknife, the bootstrap and other resampling plans*. SIAM.

Esteban, C.; Hyland, S. L.; and Rätsch, G. 2017. Real-valued (medical) time series generation with recurrent conditional gans. *arXiv preprint arXiv:1706.02633*.

Fecamp, S.; Mikael, J.; and Warin, X. 2020. Deep learning for discrete-time hedging in incomplete markets. *Journal of Computational Finance*.

Fermanian, A. 2019. Embedding and learning with signatures. *arXiv preprint arXiv:1911.13211*.

Fréchet, M. 1957. Sur la distance de deux lois de probabilité. *Comptes Rendus Hebdomadaires des Seances de L Academie des Sciences*, 244(6): 689–692.

Gelbrich, M. 1990. On a formula for the L2 Wasserstein metric between measures on Euclidean and Hilbert spaces. *Mathematische Nachrichten*, 147(1): 185–203.

Genevay, A.; Peyré, G.; and Cuturi, M. 2018. Learning generative models with sinkhorn divergences. In *International Conference on Artificial Intelligence and Statistics*, 1608–1617.

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

Gulrajani, I.; Ahmed, F.; Arjovsky, M.; Dumoulin, V.; and Courville, A. 2017. Improved training of wasserstein gans. *arXiv preprint arXiv:1704.00028*.

Ikeda, N.; and Watanabe, S. 2014. *Stochastic differential equations and diffusion processes*. Elsevier.

Jorion, P. 2000. *Value at risk*. McGraw-Hill Professional Publishing.

Kelliher, C. F.; and Mahoney, L. S. 2000. Using Monte Carlo simulation to improve long-term investment decisions. *The Appraisal Journal*, 68(1): 44.

Kidger, P.; Foster, J.; Li, X.; Oberhauser, H.; and Lyons, T. 2021. Neural sdes as infinite-dimensional gans. *arXiv preprint arXiv:2102.03657*.

Kingma, D. P.; and Welling, M. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Li, Y.; Swersky, K.; and Zemel, R. 2015. Generative moment matching networks. In *International Conference on Machine Learning*, 1718–1727.

Lyu, X.; Hueser, M.; Hyland, S. L.; Zerveas, G.; and Raetsch, G. 2018. Improving clinical predictions through unsupervised time series representation learning. *arXiv preprint arXiv:1812.00490*.

Malago, L.; Montrucchio, L.; and Pistone, G. 2018. Wasserstein riemannian geometry of positive definite matrices. *arXiv preprint arXiv:1801.09269*.

Mogren, O. 2016. C-RNN-GAN: Continuous recurrent neural networks with adversarial training. *arXiv preprint arXiv:1611.09904*.

Mullen, S. L.; and Baumhefner, D. P. 1994. Monte Carlo simulations of explosive cyclogenesis. *Monthly weather review*, 122(7): 1548–1567.

Muzellec, B.; and Cuturi, M. 2018. Generalizing point embeddings using the wasserstein space of elliptical distributions. *arXiv preprint arXiv:1805.07594*.

Oord, A. v. d.; Dieleman, S.; Zen, H.; Simonyan, K.; Vinyals, O.; Graves, A.; Kalchbrenner, N.; Senior, A.; and Kavukcuoglu, K. 2016. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*.

Pham, H. 2009. *Continuous-time stochastic control and optimization with financial applications*, volume 61. Springer Science & Business Media.

Saito, M.; Matsumoto, E.; and Saito, S. 2017. Temporal generative adversarial nets with singular value clipping. In *Proceedings of the IEEE international conference on computer vision*, 2830–2839.

Sorge, M. M. 2004. Stress-Testing Financial Systems: An Overview of Current Methodologies. *Risk Management & Analysis in Financial Institutions eJournal*.

Steinerberger, S. 2018. Wasserstein distance, Fourier series and applications. *arXiv preprint arXiv:1803.08011*.

Torrey, L.; and Shavlik, J. 2010. Transfer learning. In *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques*, 242–264. IGI global.

Tulyakov, S.; Liu, M.-Y.; Yang, X.; and Kautz, J. 2018. Mocogan: Decomposing motion and content for video generation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1526–1535.

Villani, C. 2008. *Optimal transport: old and new*, volume 338. Springer Science & Business Media.

Vondrick, C.; Pirsiavash, H.; and Torralba, A. 2016. Generating videos with scene dynamics. *Advances in neural information processing systems*, 29: 613–621.

Wiese, M.; Knobloch, R.; Korn, R.; and Kretschmer, P. 2020. Quant gans: Deep generation of financial time series. *Quantitative Finance*, 20(9): 1419–1440.

Xu, J.; Ren, X.; Lin, J.; and Sun, X. 2018. DP-GAN: diversity-promoting generative adversarial network for generating informative and diversified text. *arXiv preprint arXiv:1802.01345*.

Xu, T.; Wenliang, L. K.; Munn, M.; and Acciaio, B. 2020. COT-GAN: Generating Sequential Data via Causal Optimal Transport. *arXiv preprint arXiv:2006.08571*.

Yoon, J.; Jarrett, D.; and Van der Schaar, M. 2019. Time-series generative adversarial networks. *Advances in Neural Information Processing Systems*, 32.

Zhang, Y.; Gan, Z.; Fan, K.; Chen, Z.; Henao, R.; Shen, D.; and Carin, L. 2017. Adversarial feature matching for text generation. In *International Conference on Machine Learning*, 4006–4015. PMLR.