arXiv:2406.10214v2 [cs.LG] 6 Sep 2024

Universal randomised signatures for generative time series modelling

Francesca Biagini BIAGINI@MATH.LMU.DE
Department of Mathematics
University Munich
Munich, Germany

Lukas Gonon L.GONON@IMPERIAL.AC.UK
Department of Mathematics
Imperial College London
London, UK

Niklas Walter WALTER@MATH.LMU.DE
Department of Mathematics
University Munich
Munich, Germany

Abstract

Randomised signature has been proposed as a flexible and easily implementable alternative to the well-established path signature. In this article, we employ randomised signature to introduce a generative model for financial time series data in the spirit of reservoir computing. Specifically, we propose a novel Wasserstein-type distance based on discrete-time randomised signatures. This metric on the space of probability measures captures the distance between (conditional) distributions. Its use is justified by our novel universal approximation results for randomised signatures on the space of continuous functions taking the underlying path as an input. We then use our metric as the loss function in a non-adversarial generator model for synthetic time series data based on a reservoir neural stochastic differential equation. We compare the results of our model to benchmarks from the existing literature.

Keywords: randomised path signature, reservoir computing, generative modelling

1 Introduction

In this work, we derive universal approximation results for a reservoir based on the randomised path signature. Based on this, we introduce a novel metric on the space of probability measures and investigate its application in the generation of synthetic financial time series using a non-adversarial training technique. As described in Assefa (2020); Wiese et al. (2019); Buehler et al. (2020); Koshiyama et al. (2021) the need for synthetic financial (time series) data has grown in recent years for various reasons. First of all, the lack of historical data representing certain events in the financial market might for example lead to incorrectly estimated risk measures such as value-at-risk or expected shortfall. Therefore, it is crucial to generate realistic data for market regimes which rarely or never occurred in the past. Related to that is the growing need for training data necessary to train complex machine learning models. It might be the case that there is not enough training data from historical datasets or that an institution is not able to use some of its data for training

©2024 Biagini, Gonon and Walter.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/.

purposes. Lastly, synthetic data can ease data sharing for both institutions and academia, since it is not restricted by possible regulations.

Various techniques have been studied for generating synthetic financial data. One of the pioneering papers is Kondratyev and Schwarz (2019) using restricted Boltzmann machines, firstly introduced in Smolensky (1986), to sample synthetic foreign exchange rates. In Buehler et al. (2020); Bühler et al. (2020) the authors apply (conditional) variational autoencoders on both the pure returns and signature transformed returns to generate return time series for S&P 500 data. In Wiese et al. (2021) an autoencoder and normalising flows are used to simulate option markets. This approach also incorporates a necessary no-arbitrage condition based on discrete local volatilities. In Hamdouche et al. (2023) a novel approach for time series generation is proposed using Schrödinger bridge.

In addition to these approaches, the majority of work in the field considers generative adversarial networks (GANs). Firstly proposed in Goodfellow et al. (2014), these networks are composed of a generator network generating fake data and a discriminator network trying to distinguish between real and fake data, which leads to a min-max game. One of the first papers proposing GANs to generate financial data is Takahashi et al. (2019), where the generator and discriminator are a multi-layer perceptron, convolutional networks or a combination of both. In Wiese et al. (2020) both the generator and discriminator model consists of temporal convolutional networks addressing long-range dependencies within the time series. In Fu et al. (2022) this dependency is captured by using both convolutional networks with attention and transformers for the GAN architecture. In contrast, Esteban et al. (2017) implement the generator and discriminator pair using LSTMs to generate medical data. The model proposed in Yoon et al. (2019) expands the traditional GAN framework with a trainable embedding space so that it can adhere to the underlying data dynamics during training. Another natural extension are conditional GANs, which do not only take noise as input but also some additional information the generated samples should depend on. They are used in Koshiyama et al. (2021) to produce data used to learn trading strategies and in Coletta et al. (2021) to simulate limit order book data. For applications in a non-financial context we refer to Mogren (2016); Donahue et al. (2019); Xu et al. (2020) and their corresponding references, where GAN settings are used to produce for example music samples or videos.

In recent years, several works have proposed to substitute the discriminator of the traditional GAN with a Wasserstein-type metric, which leads to more stable training procedures. This so called Wasserstein-GAN was applied in Wiese et al. (2019) to simulate option markets. Although this model can be trained by gradient descent methods, the challenge of solving a min-max problem remains. The authors in Ni et al. (2021) proposed a Wasserstein-type distance on the signature space of the input paths. In this way, the training is transformed into a supervised learning problem. This setting is extended into a conditional setting similar to the CGANs in Liao et al. (2023) and Díaz et al. (2023). Both papers considered Neural SDEs as the generator of the model. Finally, the authors in Issa et al. (2024) propose a score-based generation technique for financial data using a kernel written on the path signature. Similarly, in Lu and Sester (2024), financial data is generated using a signature kernel and a training based on the maximum mean discrepancy. We refer to Lyons (1998); Hambly and Lyons (2010) for an in-depth study of the path signature and to Cass and Salvi (2024) for an overview of its application in machine learning.

In this paper, we also replace the GAN discriminator, but instead of using the signature we employ randomised signature as introduced in Cuchiero et al. (2020, 2021) and studied in Compagnoni et al. (2023); Schäfl et al. (2023); Akyildirim et al. (2024); Cuchiero and Möller (2023). Its connection to signature kernels was examined in Cirone et al. (2023). The related concept of path development was studied in Cass and Turner (2024); Lou et al. (2022). The advantage of randomised signature is that it inherits the expressiveness and inductive bias of the signature while being finite dimensional. Therefore, no truncation is necessary as for the signature. Moreover, the computational complexity can be reduced, which makes the randomised signature more tractable for high-dimensional input paths. In particular, we study its properties as a reservoir and derive universal approximation results for continuous function on the input path space. Based on this, we introduce a Wasserstein-distance on the reservoir space induced by the randomised signature. Lastly, we show how this new metric can be used as a discriminator for the non-adversarial training of a generator with a similar structure as the randomised signature, inspired by Neural SDEs Liu et al. (2019b); Kidger (2021); Cohen et al. (2023). Here we consider generators with random feature neural networks as coefficient functions. These neural networks have a single hidden layer and the property that the parameters of this layer are randomly initialised and then fixed for the entire training procedure. Hence, only the parameters of the output layer need to be trained. Random feature neural networks were originally introduced in Huang et al. (2006) and Rahimi and Recht (2007).

To justify our approach we prove novel universal approximation results for randomised signature. Our proof builds on a universal approximation result for random feature neural networks from Hart et al. (2020). Random feature neural networks have been shown to possess universal approximation properties similar to classical neural networks, see Hart et al. (2020); Gonon (2023); Gonon et al. (2023); Neufeld and Schmocker (2023). For quantitative error bounds relating to their generalisation properties we refer to Rudi and Rosasco (2017); Carratino et al. (2018); Mei et al. (2022); Gonon (2023) and their corresponding references.

Based on universal approximation properties of random feature neural networks we are able to derive universal approximation results for linear readouts on the randomised signature. In other words, we study the expressiveness of the randomised signature as a reservoir. More general approximation results for different classes of reservoir computing systems have been studied in Sontag (1979b,a) for inputs on finite numbers of time points and in Sandberg (1991a,b); Matthews (1992) for semi-infinite inputs. Moreover, in Grigoryeva and Ortega (2018a) the authors derived results for both uniformly bounded deterministic and almost surely bounded stochastic inputs. The latter was extended to more general stochastic inputs in Gonon and Ortega (2020) and to randomly sampled parameters in Gonon et al. (2023). In Akyildirim et al. (2022) a universal approximation result for a randomised signature is derived considering real analytic activation function.

By introducing both an unconditional and a conditional generative model, we address two alternative important modelling tasks. While the unconditional model aims to learn the unknown distribution of a stochastic process, the conditional method takes given information about a path's history into account and generates possible future trajectories by learning the underlying unknown conditional distribution. We study the models' performances on both synthetic and real world financial datasets. In particular, we consider samples of a Brownian motion with possible drift, an autoregressive process and log-return

data from the S&P 500 index and the FOREX EUR/USD exchange rate obtaining more realistic results than comparable benchmark methods. We assess the quality by providing results of test metrics applied to out-of-sample data. It is worth noting that we implemented the generator in a way such that the variance of the randomly generated weights of the random feature neural networks is trainable. This led to both a higher accuracy and a speed-up of the training procedure, while at the same time much fewer parameters need to optimised than in fully trainable models. Our source code for all experiments is available online at `https://github.com/niklaswalter/Randomised-Signature-TimeSeries-Generation`.

The present paper is organised as follows. In Section 2, we introduce a randomised signature for a deterministic datastream attaining values in a compact subset of some higher dimensional space. Moreover, we derive universal approximation results for linear readouts on the randomised signature. In Section 3, we propose a new metric on the nispace of probability measures based on the approximation results in Section 2. Based on this, we introduce a novel GAN framework to generate synthetic financial time series. In Section 4, we extend these results to a conditional setting.

2 Randomised signature of a datastream

For some finite time horizon $T \in \mathbb{N}$ let $x : \{1, \dots, T\} \to \mathbb{R}^d$ be a datastream (e.g. a timeseries) of some dimension $d \in \mathbb{N}$. Consider a filtered probability space $(\Omega, \mathcal{F}, \mathbb{F}, \mathbb{P})$ with the filtration $\mathbb{F}$ satisfying the usual conditions, which we fix for the remainder of this paper. Given $N \in \mathbb{N}$, random matrices $A_1, A_2^1, \dots, A_2^d \in \mathbb{R}^{N \times N}$, random biases $\xi_1, \xi_2^1, \dots, \xi_2^d \in \mathbb{R}^N$ defined on $(\Omega, \mathcal{F}, \mathbb{P})$ and a fixed activation function $\sigma : \mathbb{R} \to \mathbb{R}$, consider the following controlled system

$$\mathrm{RS}_t(x) = \mathrm{RS}_{t-1}(x) + \sigma(A_1 \mathrm{RS}_{t-1}(x) + \xi_1) + \sum_{i=1}^{d} \sigma(A_2^i \mathrm{RS}_{t-1}(x) + \xi_2^i) x_t^i, \; t = 1, \dots, T, \quad (1)$$

with $\mathrm{RS}_0(x) = 0 \in \mathbb{R}^N$ and x^i denoting th i-th component of x. With a slight abuse of notation, in the sequel we also denote by σ the function defined on $\mathbb{R}^N$ as

$$\sigma\left(\begin{bmatrix} z_1 \\ \vdots \\ z_N \end{bmatrix}\right) := \begin{bmatrix} \sigma(z_1) \\ \vdots \\ \sigma(z_N) \end{bmatrix}, \; z \in \mathbb{R}^N.$$

Prominent examples for activation functions are the ReLU, Tanh or Sigmoid function. We later specify our assumption on σ in more detail.

Definition 1 (Randomised signature) *Let $x : \{1, \dots, T\} \to \mathbb{R}^d$ be a datastream. For any $t = 1, \dots, T$ we call the response $RS_t(x)$ to the equation* (1) *the randomised signature of x at time t. We refer to $N \in \mathbb{N}$ as the corresponding dimension of $RS = (RS_t)_{t=1,\dots,T}$.*

The coefficient functions of the system (1) are naturally linked to random feature neural networks, defined in the following.

Definition 2 (Random feature neural network) *Let $N \in \mathbb{N}$. Consider a $\mathbb{R}^{N\times N}$-valued random matrix A and a $\mathbb{R}^N$-valued random vector ξ (often the entries of A and ξ are i.i.d. random variables respectively). For a vector $w \in \mathbb{R}^N$ consider the (random) function*

$$\Psi_w^{A,\xi}(x) := w^T\sigma(Ax + \xi),\ x \in \mathbb{R}^N \tag{2}$$

for some activation function $\sigma : \mathbb{R}^N \to \mathbb{R}^N$. The function $\Psi_w^{A,\xi}$ is called random feature neural network.

The random variables A and ξ in (2) are called the random hidden weights of the random feature neural network, while w is the vector of the output weights. For the training of the network $\Psi_w^{A,\xi}$ we first consider the hidden weights as sampled and fixed. Afterwards the output vector w is trained in order to achieve a desired approximation accuracy. In particular, the training procedure of random feature neural networks corresponds to a simple linear regression problem, since only the final linear output layer, also called readout, needs to be trained. The following remark shows how equation (1) can be turned into a neural differential equation as described in the following.

Applying a trainable linear readout function $\ell \in \left(\mathbb{R}^N\right)^*$ to the randomised signature process $\mathrm{RS}(x) = (\mathrm{RS}_t(x))_{t=1,...,T}$ at some time point $t = 1, ..., T$ turns the function in the drift and controlled term into random feature neural networks and equation (1) into a discrete-time neural differential equation, where the drift and controlled component are random feature neural networks as defined in Definition 2.

Remark 3 *For an output vector $w \in \mathbb{R}^N$ we consider the linear function $\ell_w : \mathbb{R}^N \to \mathbb{R}$ defined as $\ell_w(z) := w^T z$, $z \in \mathbb{R}^N$. For the process defined in (1) and with the notation $\Delta RS_t(x) := RS_t(x) - RS_{t-1}(x)$ it follows for any $t = 1, \ldots, T$, that*

$$\ell_w(\Delta RS_t(x)) = w^T\sigma(A_1 RS_{t-1}(x) + \xi_1) + \sum_{i=1}^{d} w^T\sigma(A_2^i RS_{t-1}(x) + \xi_2^i)x_t^i,$$

which defines a discrete-time neural differential equation driven by the input stream x. For an in-depth introduction to neural differential equations we refer to Kidger (2021).

2.1 Universal approximation properties

In this section, we study the properties of the process $\mathrm{RS}(x)$ as a reservoir. In particular, we derive different universal approximation results on linear readouts on the terminal difference $\Delta\mathrm{RS}_T(x)$ approximating continuous functions on the input path x. In order to establish universality results, we consider different sampling schemes assuming certain forms for the random matrices and vectors of the system (1).

Sampling Scheme 4 *Let $N := M + Td$ for some $M \in \mathbb{N}$. We consider the following sampling scheme for the hidden weights of (1). For $i = 1, \ldots, d$ we assume*

$$A_1 = \begin{bmatrix} 0_{M\times M} & A_1^{(1)} \\ 0_{Td\times M} & A_1^{(2)} \end{bmatrix},\ \xi_1 = \begin{bmatrix} \xi_1^{(1)} \\ 0_{Td\times 1} \end{bmatrix},\ A_2^i = \begin{bmatrix} 0_{M\times M} & 0_{M\times Td} \\ 0_{Td\times M} & 0_{Td\times Td} \end{bmatrix},\ \xi_2^i = \begin{bmatrix} 0_{M\times 1} \\ \alpha_i e_i \end{bmatrix},$$

where e_i denotes the i-th canonical basis vector of $\mathbb{R}^{Td}$, $\xi_1^{(1)} \in \mathbb{R}^M$, $A_1^{(1)} \in \mathbb{R}^{M\times Td}$, $A_1^{(2)} \in \mathbb{R}^{Td\times Td}$ are random matrices and $\alpha_1, \dots, \alpha_d$ are random variables. The matrix $A_1^{(2)}$ is given by

$$A_1^{(2)} = \begin{bmatrix} 0_{d\times(T-1)d} & 0_{d\times d} \\ U & 0_{(T-1)d\times d} \end{bmatrix}, \tag{3}$$

where $U \in \mathbb{R}^{(T-1)d\times(T-1)d}$ is an upper triangular random matrix. Let μ_1 be an absolutely continuous one-dimensional distribution with full support on $\mathbb{R}$. We assume that the random entries of $A_1^{(1)}, U, \xi_1^{(1)}$ as well as the random variables $\alpha_1, \dots, \alpha_d$ are i.i.d. under μ_1.

Remark 5 *Note that $det(U) = 0$ with probability zero, because the entries of U are i.i.d. with an absolutely continuous distribution μ_1. In other words, the matrix U is almost surely invertible.*

For the remainder of the paper, we make the following assumption on the activation function σ.

Assumption 6 *The activation function σ in (1) is continuous, bounded, non-polynomial, injective and satisfies $\sigma(0) = 0$.*

Prominent activation functions satisfying the properties stated in Assumption 6 are the Tanh or a shifted Sigmoid function. Under Assumption 6, the Sampling Scheme 4 naturally induces a block structure for the process $\mathrm{RS}(x)$ defined in (1). In particular, for every $t = 1, \dots, T$ it holds

$$\begin{bmatrix} \mathrm{RS}_t^{(1)}(x) \\ \mathrm{RS}_t^{(2)}(x) \end{bmatrix} = \begin{bmatrix} \mathrm{RS}_{t-1}^{(1)}(x) \\ \mathrm{RS}_{t-1}^{(2)}(x) \end{bmatrix} + \begin{bmatrix} \sigma(A_1^{(1)}\mathrm{RS}_{t-1}^{(2)}(x) + \xi_1^{(1)}) \\ \sigma(A_1^{(2)}\mathrm{RS}_{t-1}^{(2)}(x)) \end{bmatrix} + \begin{bmatrix} 0 \\ \sum_{i=1}^d \sigma(\alpha_i) e_i x_t^i \end{bmatrix}. \tag{4}$$

The following lemma shows that under the Sampling Scheme 4 the $\mathbb{R}^{Td}$-valued random vector $\mathrm{RS}_t^{(2)}(x)$, $t = 1, \dots, T$, can almost surely be written as an injective mapping of $(x_1, \dots, x_t)$.

Lemma 7 *Following the Sampling Scheme 4, for every $t = 1, \dots, T$ there exists almost surely a unique continuous injective function $G_t : (\mathbb{R}^d)^t \to (\mathbb{R}^d)^T$ such that*

$$RS_t^{(2)}(x) = G_t(x_1, \dots, x_t). \tag{5}$$

Proof The proof consists of two parts. In the first part, we prove that for every $t = 1, \dots, T$, the function G_t satisfying equation (5) is of the following form

$$G_t(x_1, \dots, x_t) = \begin{bmatrix} g_t^{(1)}(x_1, \dots, x_t) \\ \vdots \\ g_t^{(t)}(x_1) \\ 0_{(T-t)d\times 1} \end{bmatrix} \tag{6}$$

for some unique component functions $g_t^{(k)} : (\mathbb{R}^d)^{t+1-k} \to \mathbb{R}^d$, $k = 1, \dots, t$. In the second part of the proof, we show that the functions $g_t^{(k)}(x_1, \dots, x_{t-k}, \cdot)$ are almost surely injective for fixed $x_1, \dots, x_{t-k}$, which implies the injectivity of G_t.

First part: We prove equality (5) with G_t satisfying (6) by induction over $t = 1, \dots, T$. For time $t = 1$ it follows from (4) that

$$\mathrm{RS}_1^{(2)}(x) = \begin{bmatrix} \sigma(\alpha_1)x_1^1 \\ \vdots \\ \sigma(\alpha_d)x_1^d \\ 0_{(T-1)d\times 1} \end{bmatrix} =: \begin{bmatrix} g_1^{(1)}(x_1) \\ 0_{(T-1)d\times 1} \end{bmatrix}, \tag{7}$$

as $\mathrm{RS}_0(x) = 0$, where $g_1^{(1)}$ is almost surely well-defined and linear in x_1. So representation (6) holds for $t = 1$. We now prove by induction that if (5) holds with G_t satisfying (6) for $t = 1, \dots, T-1$, then it holds for $t+1$. For this purpose, we observe that the upper triangular matrix U from (3) can be written as

$$U = \begin{bmatrix} U_1 & B_{1,1} & \cdots & \cdots & B_{1,T-2} \\ & \ddots & \ddots & & \vdots \\ 0 & & \ddots & \ddots & \vdots \\ \vdots & & & \ddots & B_{T-2,1} \\ 0 & \cdots & \cdots & 0 & U_{T-1} \end{bmatrix} \tag{8}$$

for some upper triangular random matrices $U_1, \dots, U_{T-1} \in \mathbb{R}^{d\times d}$ and some arbitrary random matrices $B_{i,j} \in \mathbb{R}^{d\times d}$, $i = 1, \dots, T-2$, $j = 1, \dots, T-1-i$, whose entries are i.i.d. under μ_1. From (4), (8) and the assumption of our induction it follows

$$\mathrm{RS}_{t+1}^{(2)}(x) = \begin{bmatrix} g_t^{(1)}(x_1, \dots, x_t) + \sum_{i=1}^d \sigma(\alpha_i)\tilde{e}_i x_{t+1}^i \\ g_t^{(2)}(x_1, \dots, x_{t-1}) + \sigma\left(U_1 g_t^{(1)}(x_1, \dots, x_t) + \sum_{k=1}^{t-1} B_{1,k} g_t^{(k+1)}(x_1, \dots, x_{t-k})\right) \\ \vdots \\ g_t^{(t)}(x_1) + \sigma\left(U_{t-1} g_t^{(t-1)}(x_1, x_2) + B_{t-1,1} g_t^{(t)}(x_1)\right) \\ \sigma\left(U_t g_t^{(t)}(x_1)\right) \\ 0_{(T-(t+1))d\times 1} \end{bmatrix},$$

where $\tilde{e}_i$ denotes the i-th canonical basis vector of $\mathbb{R}^d$. Then (6) holds for the following choices of the functions $g_{t+1}^{(k)} : (\mathbb{R}^d)^{t+2-k} \to \mathbb{R}^d$, $k = 1, \dots, t+1$, given by

$$g_{t+1}^{(1)}(x_1, \dots, x_{t+1}) := g_t^{(1)}(x_1, \dots, x_t) + \sum_{i=1}^d \sigma(\alpha_i)\tilde{e}_i x_{t+1}^i, \tag{9}$$

$$g_{t+1}^{(k)}(x_1,\dots,x_{t+2-k}) - g_t^{(k)}(x_1,\dots,x_{t+1-k}) \tag{10}$$
$$:=\sigma\left(U_{k-1}g_t^{(k-1)}(x_1,\dots,x_{t+2-k}) + \sum_{j=1}^{t+1-k} B_{k-1,j}g_t^{(j+k-1)}(x_1,\dots,x_{t+2-k-j})\right)$$

for $k = 2,\dots,t$ and

$$g_{t+1}^{(t+1)}(x_1) := \sigma\left(U_t g_t^{(t)}(x_1)\right). \tag{11}$$

Note that the construction of the component functions is unique and thus concludes the first step of the proof.

Second part: We prove that the component functions in (6) are almost surely injective as functions only of their last variable, i.e. for all $t = 1,\dots,T$ and $k = 1,\dots,t$ the function

$$g_t^{(k)}(x_1,\dots,x_{t-k},\cdot) : z \mapsto g_t^{(k)}(x_1,\dots,x_{t-k},z) \tag{12}$$

is almost surely injective. We prove this statement by induction over t. For time $t = 1$ we have only one component function defined in (7). The events $\{\alpha_i = 0\}$, $i = 1,\dots,d$, have zero probability due to the absolute continuity of the distribution μ_1. Therefore, the component function from (7) is almost surely injective, since it holds $\sigma(0) = 0$ by Assumption 6.

For the induction step $t \mapsto t+1$ we first consider $k = t+1$, i.e. the function $g_{t+1}^{(t+1)}$ is defined in (11). By assumption, the function $g_t^{(t)}$ is almost surely injective. Moreover, by Assumption 6 the activation function σ is injective and U_t is almost surely invertible by the same reasoning as in Remark 5. Therefore, it follows that $g_{t+1}^{(t+1)}$ is almost surely injective. For $k = t$ we obtain

$$g_{t+1}^{(t)}(x_1,x_2) = g_t^{(t)}(x_1) + \sigma\left(U_{t-1}g_t^{(t-1)}(x_1,x_2) + B_{t-1,1}g_t^{(t)}(x_1)\right).$$

For a fixed x_1, we can uniquely determine x_2 in $g_{t+1}^{(t)}$ again using that σ is injective and U_{t-1} is (almost surely) invertible. With fixed x_1, x_2 we can uniquely determine x_3 in $g_{t+1}^{(t-1)}$. Therefore the result follows by backward induction for all $k = 1,\dots,t+1$. This also concludes the induction on t, i.e. the function $g_t^{(k)}$ is almost surely injective in the last variable. Note that this is sufficient to guarantee that the function G_t defined in (6) is almost surely injective in all variables for all $t = 1,\dots,T$. In addition, all component functions are continuous as compositions of continuous functions, which concludes the proof. ■

Before we prove the universality of linear readouts of $\Delta\mathrm{RS}_T(x)$, we state a universal approximation result of random feature neural networks as defined in Definition 2 for continuous functions on a compact domain. The result is very similar to Theorem 2.4.5 in Hart et al. (2020), where the authors consider activation functions that are ℓ-finite for some $\ell \in \mathbb{N}_0$ to prove approximation results for functions in $\mathcal{C}^\ell$. We consider non-polynomial activation functions without integrability assumptions so that we can include the Tanh or Sigmoid function in our analysis.

Proposition 8 (Universality of random feature neural networks) *Let $d \in \mathbb{N}$, $M \in \mathbb{N}$, $K \subseteq \mathbb{R}^d$ a compact set and $f \in \mathcal{C}(K)$. If the activation function σ is continuous and non-polynomial, A is a $\mathbb{R}^{M\times d}$-valued random matrix and ξ is a $\mathbb{R}^M$-valued random vector both with i.i.d. entries with a common continuous distribution, then for any $\alpha \in (0,1)$ and $\varepsilon > 0$ there exists a choice for M such that with probability greater than α, there exists a vector $w \in \mathbb{R}^M$ such that the corresponding random feature neural network $\Psi_w^{A,\xi} : K \to \mathbb{R}$ defined by*

$$\Psi_w^{A,\xi}(x) = w^T \sigma(Ax + \xi)$$

satisfies

$$\|f - \Psi_w^{A,\xi}\|_\infty < \varepsilon,$$

where $\|\cdot\|_\infty$ denotes the supremum norm on K.

Proof The proof follows the same steps as in Hart et al. (2020) but applies Theorem 1 in Leshno et al. (1993) instead of a universality result for functions in $\mathcal{C}^k(K)$ with $k \geq 1$. ■

Based on Proposition 8 we can now state the following universality result for the terminal difference of the randomised signature $\mathrm{RS}(x)$ defined in (1).

Proposition 9 *Let $K \subseteq \mathbb{R}^d$ be a compact subset and $f \in \mathcal{C}(K^{T-1})$. The activation function σ satisfies Assumption 6. Then under the Sampling Scheme 4, for any $\alpha \in (0,1)$ and $\varepsilon > 0$ there exists a $M \in \mathbb{N}$ such that with probability greater than α, there exists $\tilde{w} \in \mathbb{R}^M$ such that*

$$\sup_{x_1,\ldots,x_{T-1}\in K} |f(x_1,\ldots,x_{T-1}) - \langle w, \Delta RS_T(x_1,\ldots,x_{T-1})\rangle| < \varepsilon$$

for the output vector $w := [\ \tilde{w}^T \quad 0_{1\times Td}\ ]^T$ and $\langle\cdot,\cdot\rangle$ denoting the standard scalar product on $\mathbb{R}^{M\times Td}$.

Proof Let $\alpha \in (0,1)$ and $\varepsilon > 0$. Consider the Sampling Scheme 4. By Lemma 7 for every $t = 1,\ldots,T$ there exists an almost surely injective function $G_t : (\mathbb{R}^d)^t \to (\mathbb{R}^d)^T$ such that $G_t(x_1,\ldots,x_t) = \mathrm{RS}_t^{(2)}(x)$ as defined in (4). So in particular, there exists a function $G_t^{-1} : \mathrm{Im}(G_t) \to (\mathbb{R}^d)^t$ such that

$$G_t^{-1}(\mathrm{RS}_t^{(2)}(x)) = G_t^{-1}(G_t(x_1,\ldots,x_t)) = (x_1,\ldots,x_t), \tag{13}$$

where $\mathrm{Im}(G_t)$ denotes the image of G_t. On the other hand, by (4) it holds that

$$\Delta\mathrm{RS}_T^{(1)}(x) = \sigma(A_1^{(1)}\mathrm{RS}_{T-1}^{(2)}(x) + \xi_1^{(1)}). \tag{14}$$

By Lemma 7 the function G_{T-1} is almost surely continuous, hence in particular the set $\bar{K} := G_{T-1}(K^{T-1})$ is compact. Moreover, the function $f \circ G_{T-1}^{-1}$ is almost surely continuous as a composition of two continuous functions. Therefore, by Proposition 8 we can choose $M \in \mathbb{N}$ such that with probability greater than α, there exists $\tilde{w} \in \mathbb{R}^M$ such that with (13)

it holds

$$\begin{aligned}&|f(x_1,\dots,x_{T-1}) - \tilde{w}^T\sigma(A_1^{(1)}\mathrm{RS}_{T-1}^{(2)}(x) + \xi_1^{(1)})|\\ =&|\left(f\circ G_{T-1}^{-1}\right)(\mathrm{RS}_{T-1}^{(2)}(x)) - \tilde{w}^T\sigma(A_1^{(1)}\mathrm{RS}_{T-1}^{(2)}(x) + \xi_1^{(1)})|\\ <&\varepsilon\end{aligned}$$

for all $x_1,\dots,x_{T-1}\in K$. So choosing $w := [\ \tilde{w}^T \quad 0_{1\times Td}\ ]^T$ yields by (4) for all $x_1,\dots,x_{T-1}\in K$

$$\begin{aligned}&|f(x_1,\dots,x_{T-1}) - \langle w, \Delta\mathrm{RS}_T(x)\rangle|\\ =&|f(x_1,\dots,x_{T-1}) - \tilde{w}^T\sigma(A_1^{(1)}\mathrm{RS}_{T-1}^{(2)}(x) + \xi_1^{(1)})|\\ <&\varepsilon,\end{aligned}$$

which concludes the proof. Note that the set $\bar{K}$ depends on $A_1^{(2)}$ and $\alpha_1,\dots,\alpha_d$. Therefore, it is worth noting that the entries of $A_1^{(2)}$ and the random variables $\alpha_1,\dots,\alpha_d$ are independent of the entries of $A_1^{(1)}$ and $\xi_1^{(1)}$ so that we could apply Proposition 8 independently of $\bar{K}$. ■

Note that in Proposition 9 the approximated function f does not take the terminal value of x as an argument. However, we can prove a universality result if the function f is of the following form

$$f(x_1,\dots,x_T) = g(x_1,\dots,x_{T-1}) + \sum_{i=1}^{d} h_i(x_1,\dots,x_{T-1})x_T^i \tag{15}$$

for continuous $g,h : (\mathbb{R}^d)^{T-1}\to\mathbb{R}$. In order to prove a approximation result for a function of the form (15), we introduce the following alternative sampling scheme.

Sampling Scheme 10 *Let $N := M + M_1 + \cdots + M_d + Td$ for some $M, M_1,\dots,M_d\in\mathbb{N}$ and denote by $\tilde{M}_j := \sum_{i=1}^{j} M_i$, $j = 1,\dots,d$. We consider the following sampling scheme for the hidden weights of* (1)*. For $i = 1,\dots,d$ let*

$$A_1 = \begin{bmatrix} 0_{M\times M} & 0_{M\times\tilde{M}_d} & A_1^{(1)}\\ 0_{\tilde{M}_d\times M} & 0_{\tilde{M}_d\times\tilde{M}_d} & 0_{\tilde{M}_d\times Td}\\ 0_{Td\times M} & 0_{Td\times\tilde{M}_d} & A_1^{(2)}\end{bmatrix},\ \xi_1 = \begin{bmatrix}\tilde{\xi}_1\\ 0_{\tilde{M}_d\times 1}\\ 0_{T\times 1}\end{bmatrix},$$

$$A_2^i = \begin{bmatrix} 0_{M+\tilde{M}_{i-1}\times M} & 0_{M+\tilde{M}_{i-1}\times\tilde{M}_d} & 0_{M+\tilde{M}_{i-1}\times Td}\\ 0_{M_i\times M} & 0_{M_i\times\tilde{M}_d} & \tilde{A}_2^i\\ 0_{Td\times M} & 0_{Td\times\tilde{M}_d} & 0_{Td\times Td}\end{bmatrix},\ \xi_2^i = \begin{bmatrix}0_{M+\tilde{M}_{i-1}\times 1}\\ \tilde{\xi}_2^i\\ \alpha_i e_i\end{bmatrix},$$

where e_i denotes the i-th canonical basis vector of $\mathbb{R}^{Td}$, $A_1^{(1)}\in\mathbb{R}^{M\times Td}$, $\tilde{\xi}_1\in\mathbb{R}^M$, $A_1^{(2)}\in\mathbb{R}^{Td\times Td}$, $\tilde{A}_2^i\in\mathbb{R}^{M_i\times Td}$, $\tilde{\xi}_2^i\in\mathbb{R}^{M_i}$ are random matrices and $\alpha_1,\dots,\alpha_d$ random variables. In

particular, the matrix $A_1^{(2)}$ *is of the form*

$$A_1^{(2)} = \begin{bmatrix} 0_{d\times(T-1)d} & 0_{d\times d} \\ U & 0_{(T-1)d\times d} \end{bmatrix}, \tag{16}$$

where $U \in \mathbb{R}^{(T-1)d\times(T-1)d}$ *is an upper triangular random matrix. Let* μ_2 *be an absolutely continuous one-dimensional distribution with full support on* $\mathbb{R}$*. We assume that the random entries of* $A_1^{(1)}, A_1^{(2)}, \tilde{\xi}_1$*,* $\tilde{A}_2^i, \tilde{\xi}_2^i$*,* U *as well as the random variables* $\alpha_1, \dots, \alpha_d$ *are i.i.d. under* μ_2*.*

As in (4) also the Sampling Scheme 10 induces a block structure for the process $\mathrm{RS}(x)$ defined in (1). In particular, for every $t = 1, \dots, T$ it holds

$$\begin{bmatrix} \mathrm{RS}_t^{(1)}(x) \\ \mathrm{RS}_t^{(2)}(x) \\ \vdots \\ \mathrm{RS}_t^{(d+1)}(x) \\ \mathrm{RS}_t^{(d+2)}(x) \end{bmatrix} = \begin{bmatrix} \mathrm{RS}_{t-1}^{(1)}(x) \\ \mathrm{RS}_{t-1}^{(2)}(x) \\ \vdots \\ \mathrm{RS}_{t-1}^{(d+1)}(x) \\ \mathrm{RS}_{t-1}^{(d+2)}(x) \end{bmatrix} + \begin{bmatrix} \sigma(A_1^{(1)}\mathrm{RS}_{t-1}^{(d+2)}(x) + \tilde{\xi}_1) \\ 0 \\ \vdots \\ 0 \\ \sigma(A_1^{(2)}\mathrm{RS}_{t-1}^{(d+2)}(x)) \end{bmatrix} + \begin{bmatrix} 0 \\ \sigma(\tilde{A}_2^1\mathrm{RS}_{t-1}^{(d+2)}(x) + \tilde{\xi}_2^1)x_t^1 \\ \vdots \\ \sigma(\tilde{A}_2^d\mathrm{RS}_{t-1}^{(d+2)}(x) + \tilde{\xi}_2^d)x_t^d \\ \sum_{i=1}^d \sigma(\alpha_i)e_i x_t^i \end{bmatrix}. \tag{17}$$

Similar to Sampling Scheme 4, we can show that the last entry $\mathrm{RS}_t^{(d+2)}$ in (17) can be written as an injective function on the values of x up to time $t = 1, \dots, T$.

Lemma 11 *Following the Sampling Scheme 10, for every* $t = 1, \dots, T$ *there exists an almost surely unique continuous injective function* $G_t : (\mathbb{R}^d)^t \to (\mathbb{R}^d)^T$ *such that*

$$RS_t^{(d+2)}(x) = G_t(x_1, \dots, x_t). \tag{18}$$

Proof This proof is exactly the same as the one of Lemma 7. ∎

Based on Lemma 11 we obtain the following universal approximation result for the terminal difference of the randomised signature for continuous functions of the form (15).

Proposition 12 *Let* $f \in \mathcal{C}(K^T)$ *be a function of the form* (15)*. The activation function* σ *satisfies Assumption 6. Then under the Sampling Scheme 10, for any* $\alpha \in (0,1)$ *and* $\varepsilon > 0$ *there exist* $M \in \mathbb{N}$ *and* $M_1, \dots, M_d \in \mathbb{N}$ *such that with probability greater than* α*, there exist* $\tilde{w} \in \mathbb{R}^M$ *and* $\tilde{w}_i \in \mathbb{R}^{M_i}$*,* $i = 1, \dots, d$*, such that*

$$\sup_{x_1,\dots,x_T\in K} |f(x_1, \dots, x_T) - \langle w, \Delta RS_T(x_1, \dots, x_T)\rangle| < \varepsilon$$

for the output vector $w = [\ \tilde{w}^T \quad \tilde{w_1}^T \quad \dots \quad \tilde{w_d}^T \quad 0_{1\times Td}\ ]^T$*.*

Proof Let $\alpha \in (0,1)$ and $\varepsilon > 0$. Consider the Sampling Scheme 10. As in the proof of Proposition 9 we have

$$G_t^{-1}(\mathrm{RS}_t^{(d+2)}(x)) = G_t^{-1}(G_t(x_1, \dots, x_t)) = (x_1, \dots, x_t) \tag{19}$$

for any $x \in K^T$. For the block components of $\mathrm{RS}_T(x)$ following Sampling scheme 10 it holds using (17)

$$\Delta\mathrm{RS}_T^{(1)}(x) = \sigma(A_1^{(1)}\mathrm{RS}_{T-1}^{(d+2)}(x) + \tilde{\xi}_1) \tag{20}$$

and for $i = 1, \dots, d$

$$\Delta\mathrm{RS}_T^{(i+1)}(x) = \sigma(\tilde{A}_2^i\mathrm{RS}_{T-1}^{(d+2)}(x) + \tilde{\xi}_2^i)x_t^i. \tag{21}$$

Since G_{T-1} from Lemma 11 is almost surely continuous, it follows that the set $\bar{K} := G_{T-1}(K^{T-1})$ is compact. Moreover, the functions $g \circ G_{T-1}^{-1}$, $h_i \circ G_{T-1}^{-1}$, $i = 1, \dots, d$, are almost surely continuous. Therefore, by Proposition 8 there exist $M, M_1, \dots, M_d \in \mathbb{N}$ such that with probability greater than α, there exist $\tilde{w} \in \mathbb{R}^M$ and $\tilde{w}_1 \in \mathbb{R}^{M_1}, \dots, \tilde{w}_d \in \mathbb{R}^{M_d}$ such that with Lemma 11 it holds

$$\begin{aligned}
&\sup_{x_1,\dots,x_{T-1}\in K} |g(x_1, \dots, x_{T-1}) - \tilde{w}^T\sigma(A_1^{(1)}\mathrm{RS}_{T-1}^{(d+2)}(x) + \tilde{\xi}_1)| \\
&= \sup_{x_1,\dots,x_{T-1}\in K} |(g \circ G_{T-1}^{-1})(\mathrm{RS}_{T-1}^{(d+2)}(x)) - \tilde{w}^T\sigma(A_1^{(1)}\mathrm{RS}_{T-1}^{(d+2)}(x) + \tilde{\xi}_1)| < \frac{\varepsilon}{2}
\end{aligned} \tag{22}$$

and for every $i = 1, \dots, d$

$$\begin{aligned}
&\sup_{x_1,\dots,x_{T-1}\in K} |h_i(x_1, \dots, x_{T-1}) - \tilde{w}_i^T\sigma(\tilde{A}_2^i\mathrm{RS}_{T-1}^{(d+2)}(x) + \tilde{\xi}_2^i)| \\
&= \sup_{x_1,\dots,x_{T-1}\in K} |(h_i \circ G_{T-1}^{-1})(\mathrm{RS}_{T-1}^{(d+2)}(x)) - \tilde{w}_i^T\sigma(\tilde{A}_2^i\mathrm{RS}_{T-1}^{(d+2)}(x) + \tilde{\xi}_2^i)| < \frac{\varepsilon}{2d\sup\{|x| \mid x \in K\}},
\end{aligned} \tag{23}$$

Therefore, by choosing $w = [\ \tilde{w}^T \quad \tilde{w}_1^T \quad \dots \quad \tilde{w}_d^T \quad 0_{1\times Td}\ ]^T$ we obtain by (20) and (21) that

$$\langle w, \Delta\mathrm{RS}_T(x)\rangle = \tilde{w}^T\sigma(A_1^{(1)}\mathrm{RS}_{T-1}^{(d+2)}(x) + \tilde{\xi}_1) + \sum_{i=1}^d \tilde{w}_i^T\sigma(\tilde{A}_2^i\mathrm{RS}_{T-1}^{(d+2)}(x) + \tilde{\xi}_2^i)x_T^i.$$

In particular, combining (22) with (23) yields

$$\begin{aligned}
&\sup_{x_1,\dots,x_T\in K} |f(x_1, \dots, x_T) - \langle w, \Delta\mathrm{RS}_T(x)\rangle| \\
\leq &\sup_{x_1,\dots,x_{T-1}\in K} |g(x_1, \dots, x_{T-1}) - \tilde{w}^T\sigma(A_1^{(1)}\mathrm{RS}_{T-1}^{(d+2)}(x) + \tilde{\xi}_1)| \\
&+ \sum_{i=1}^d \sup_{x_1,\dots,x_{T-1}\in K} |h_i(x_1, \dots, x_{T-1}) - \tilde{w}_i^T\sigma(\tilde{A}_2^i\mathrm{RS}_{T-1}^{d+2}(x) + \tilde{\xi}_2^i)| \sup\{|x| \mid x \in K\} \\
\leq &\frac{\varepsilon}{2} + d\frac{\varepsilon}{2d\sup\{|x| \mid x \in K\}} \sup\{|x| \mid x \in K\} = \varepsilon,
\end{aligned}$$

which concludes the proof. ■

3 Time series generation using the randomised signature

Consider the compact set $K \subseteq \mathbb{R}^d$ introduced in Section 2. In the following, we denote by $\mathcal{X} := K \times \cdots \times K \subseteq (\mathbb{R}^d)^T$. Let $X = (X_t)_{t=1,\dots,T}$ be a $\mathcal{X}$-valued discrete-time stochastic process defined on the probability space with unknown distribution $\mathbb{P}^{\text{real}} := \mathbb{P} \circ X^{-1}$. In this section, we want to generate trajectories of a $\mathcal{X}$-valued stochastic process $\hat{X}^\theta = (\hat{X}^\theta_t)_{t=1,\dots,T}$ with distribution $\mathbb{P}^{\text{fake}}_\theta := \mathbb{P} \circ (\hat{X}^\theta)^{-1}$ such that $d_{\text{path}}(\mathbb{P}^{\text{real}}, \mathbb{P}^{\text{fake}})$ is small, where d_{path} is a (pseudo-)metric on the space of probability measures on $\mathcal{X}$ denoted by $\mathcal{P}(\mathcal{X})$. We will introduce a specific novel choice for d_{path} based on randomised signature later this section. In addition, we define $\hat{X}^\theta$ based on randomised signature as follows.

Consider a n-dimensional $\mathbb{F}$-Brownian motion $W = (W^1_t, \dots, W^n_t)_{t\in[0,T]}$ and a random vector $V \sim \mathcal{N}_m(0, \mathrm{Id})$ both defined on the probability space $(\Omega, \mathcal{F}, \mathbb{F}, \mathbb{P})$ for some $n, m \in \mathbb{N}$ and independent of each other. For some $D \in \mathbb{N}$ we consider the D-dimensional stochastic process $R = (R_t)_{t=1,\dots,T}$ and a d-dimensional stochastic process $X^\theta = (X^\theta_t)_{t=1,\dots,T}$ defined as

$$\begin{cases} R_1 = \Psi^\theta(V), \\ R_t = R_{t-1} + \sigma(\rho^\theta_1 B_1 R_{t-1} + \rho^\theta_2 \lambda_1) + \sum_{i=1}^n \sigma(\rho^\theta_3 B^i_2 R_{t-1} + \rho^\theta_4 \lambda^i_2)\rho^\theta_5 (W^i_t - W^i_{t-1}), \\ X^\theta_t = A^\theta_t R_t + \beta^\theta_t,\ t = 1, \dots, T \end{cases} \tag{24}$$

for a neural network $\Psi^\theta : \mathbb{R}^m \to \mathbb{R}^D$, random matrices $B_1, B^1_2, \dots, B^n_2 \in \mathbb{R}^{D\times D}$, random vectors $\lambda_1, \lambda^1_2, \dots, \lambda^w_2 \in \mathbb{R}^D$, parameters $\rho^\theta_1, \dots, \rho^\theta_5 \in \mathbb{R}$, $A^\theta_1, \dots, A^\theta_T \in \mathbb{R}^{d\times D}$, $\beta^\theta_1, \dots, \beta^\theta_T \in \mathbb{R}^d$ and some activation function $\sigma : \mathbb{R} \to \mathbb{R}$, which is applied component-wise. The entries of the random matrices are i.i.d. with some absolutely continuous distribution μ_3 with full support. Moreover, we assume that the random entries are independent of the process X and of W, V. The superscript θ implies that the weights of the neural network Ψ^θ, the parameters $\rho^\theta_1, \dots, \rho^\theta_5$ and the readouts $A^\theta_1, \dots, A^\theta_N, \beta^\theta_1, \dots, \beta^\theta_N$ are trainable. One can think of the process R as a dynamical reservoir system evolving over time. Comparing to (1), we see that R is a variant of the randomised signature of $(W_t - W_{t-1})_{t=1,\dots,T}$ with randomly initialised state R_1. The value X^θ_t, $t = 1, \dots, T$, is the output of a linear readout from the state of R at time t. For some $r > 0$ we consider the following projection defined for any $x \in \mathbb{R}^d$ as

$$\pi_r(x) := \begin{cases} x, \text{ if } \|x\| \le r, \\ r\frac{x}{\|x\|}, \text{ if } \|x\| > r. \end{cases}$$

Finally, we define $\hat{X}^\theta_t := \pi_r(X^\theta_t)$ for any $t = 1, \dots, T$, where the parameter r is chosen such that $\hat{X}^\theta \in \mathcal{X}$ almost surely. From a practical point of view, the projection can be omitted in the numerical implementation if the subset $\mathcal{X}$ is considered large enough.

We can formulate the following training problem for generating synthetic trajectories of X using $\hat{X}^\theta$ as

$$\min_\theta d_{\text{path}}(\mathbb{P}^{\text{real}}, \mathbb{P}^{\text{fake}}_\theta). \tag{25}$$

Prior to the training procedure, the entries of $B_1, B^1_2, \dots, B^w_2$ and $\lambda_1, \lambda^1_2, \dots, \lambda^w_2$ are drawn from the absolutely continuous distribution μ_3 and then fixed. Hence, the only remaining source of randomness is the Brownian motion W and random vector V. In the following, we propose a novel choice for d_{path} for the problem (25).

3.1 The randomised signature Wasserstein-1 (RS-W_1) metric

In this section, we introduce a modified version of the well-known Wasserstein-1 metric first introduced in Kantorovich (1960). Recall that for two probability measures $\mu, \nu \in \mathcal{P}(\mathcal{X})$ the dual representation of the W_1 distance of Kantorovich and Rubinstein is given by

$$W_1(\mu, \nu) = \sup_{\|f\|_L \leq 1} \mathbb{E}_\mu[f(X)] - \mathbb{E}_\nu[f(X)], \tag{26}$$

where $\|\cdot\|_L$ denotes the Lipschitz norm defined for any $f : \mathcal{X} \to \mathbb{R}$ as

$$\|f\|_L := \sup_{x,y \in \mathcal{X}} \frac{|f(x) - f(y)|}{\|x - y\|_2}.$$

In the last section, we have seen that the randomised signature is a universal feature map. In other words, we can approximate continuous functions on $\mathcal{X}$ by linear readouts of the terminal difference of the randomised signature. Proposition 9 and 12 state that for a large class of functions $f \in \mathcal{C}(\mathcal{X})$ there exist with an arbitrary large probability a dimension N for the randomised signature and a vector $w \in \mathbb{R}^N$ such that $\langle w, \Delta\mathrm{RS}_T(\cdot)\rangle : \mathcal{X} \to \mathbb{R}$ approximates f up to an arbitrarily small error. In particular, this motivates that the Lipschitz-continuous function considered in (26) could be approximated as well. Therefore, we propose the following randomised signature Wasserstein-1 metric

$$\text{RS-}W_1(\mu, \nu) := \sup_{w \in \mathbb{R}^M,\, \|w\| \leq 1} \mathbb{E}_\mu[\langle w, \Delta\mathrm{RS}_T(X)\rangle] - \mathbb{E}_\nu[\langle w, \Delta\mathrm{RS}_T(X)\rangle]$$

for any $\mu, \nu \in \mathcal{P}(\mathcal{X})$. Note that the integrability of $\Delta\mathrm{RS}_T(X)$ is ensured by the fact that X and $\hat{X}^\theta$ attain only values in the compact set $\mathcal{X}$ almost surely and the activation function is bounded by Assumption 6. In particular, the problem becomes linear. Hence, using the linearity of the expectation and by choosing the Euclidean norm $\|\cdot\| := \|\cdot\|_2$ we obtain the final expression

$$\text{RS-}W_1(\mu, \nu) = \|\mathbb{E}_\mu[\Delta\mathrm{RS}_T(X)] - \mathbb{E}_\nu[\Delta\mathrm{RS}_T(X)]\|_2. \tag{27}$$

We use the (pseudo-)metric (27) for generating synthetic trajectories of the process X based on $\hat{X}^\theta$ defined in (24). In the tradition of GAN models, we call the system (24) the *generator* and the metric (27) the *discriminator*. Therefore, we consider the following generator-discriminator pair

$$\textbf{Generator: } \hat{X}^\theta \qquad \textbf{Discriminator: } \text{RS-}W_1(\mathbb{P}^{\text{real}}, \mathbb{P}^{\text{fake}}_\theta) \tag{28}$$

so that the corresponding training objective (25) can be written as

$$\min_\theta \text{RS-}W_1(\mathbb{P}^{\text{real}}, \mathbb{P}^{\text{fake}}_\theta) = \min_\theta \|\mathbb{E}_{\mathbb{P}^{\text{real}}}[\Delta\mathrm{RS}_T(X)] - \mathbb{E}_{\mathbb{P}^{\text{fake}}_\theta}[\Delta\mathrm{RS}_T(X)]\|_2. \tag{29}$$

It is worth noting that the training problem of traditional GAN models are formulated as a min-max problem. This leads to an adversarial training, which might cause problems in the optimal numerical implementation, since the problem might not converge to a solution as described in Daskalakis et al. (2017); Daskalakis and Panageas (2018). By considering the RS-W_1 metric as the discriminator, we are able to formulate the training objective as a minimisation problem of a convex function, so that the training becomes non-adversarial. A similar approach using signature was pursued in Ni et al. (2021), Liao et al. (2023), Issa et al. (2024) and Díaz et al. (2023).

3.2 Numerical results

In this section, we present out-of-sample results for the generator-discriminator pair (28). In particular, we generate timeseries data based on both synthetic and real world training data. The generator proposed in (24) is compared to the long-short-term-memory (LSTM) model introduced in Hochreiter and Schmidhuber (1997) for timeseries generation. Moreover, we compare the discriminator metric to the Sig-W_1 metric proposed in Ni et al. (2021). It is worth noting that the Sig-W_1 discriminator was tested against other benchmark models as well. We consider the path signature of time, lead-lag and visibility augmented input paths to compute the Sig-W_1 metric. We refer to Ni et al. (2021) for more details. To measure the quality of generated paths we compare the corresponding discriminator metric, as well as metrics for the covariance and autocorrelation introduced in Appendix B. We also apply a Shapiro-Wilk test for normality as defined in Shapiro and Wilk (1965) in case the generated data should be normally distributed based on its training data. For all experiments we use 80% of the available samples for the training set and 20% for the test set. Moreover, we apply the Adam algorithm Kingma and Ba (2015) for optimising (24), where each gradient step is based on a batch of randomly selected training paths. Table 9 gives on overview of the hyperparameters used. For this entire section, we set ρ_5^θ in (24) equal to one. However, for longer time horizons and to generate paths with very small variance it proved to enhance the generator's performance to include ρ_5^θ in the set of trainable parameters. It is worth noting that also the dimension of N of the randomised signature is a hyperparameter and tuned for the corresponding number of timesteps. For experiments with larger time horizons, the choice of N must be increased as well. Heuristically, Sampling Scheme 4 and 10 indicate that the relationship of N and T should be chosen linearly.

3.2.1 Brownian motion with drift

First, we generate trajectories of a discretised Brownian motion with drift $\tilde{W} = (\tilde{W}_t)_{t=1,\dots,T}$ defined as

$$\begin{cases} \tilde{W}_1 = 0 \\ \tilde{W}_t = \tilde{W}_{t-1} + \mu + \sigma Z_t, \ t = 2, \dots, N \end{cases} \tag{30}$$

for a drift parameter $\mu \in \mathbb{R}$, volatility $\sigma > 0$ and independent $Z_t \sim \mathcal{N}(0,1)$, $t = 2, \dots, T$. In our experiment we study paths of both a Brownian motion with and without drift. The out-of-sample results are summarised in Table 1.

We observe that for both drift parameters, our proposed generator-discriminator pair generates data with normal marginals due to the Shapiro-Wilk test. Further, based on out-of-sample data, both the covariance and autocorrelation metrics provide the lowest values for our model implying a more realistic dependence structure. It is also worth noting that the Sig-W_1 discriminator performs better combined with our generator than the LSTM model.

	Train Metric	Cov. Metric	ACF Metric	SW Tests Passed
$\mu = 0,\ \sigma = 1$				
RS-W_1 - LSTM	1.75×10^{-1}	2.61×10^{-1}	1.97×10^{-1}	6/9
RS-W_1 - NeuralSDE	$\mathbf{1.31 \times 10^{-1}}$	$\mathbf{1.98 \times 10^{-1}}$	$\mathbf{6.64 \times 10^{-2}}$	**9/9**
Sig-W_1 - LSTM	$\mathbf{8.15 \times 10^{-1}}$	9.14×10^{-1}	4.67×10^{-1}	5/9
Sig-W_1 - NeuralSDE	9.78×10^{-1}	8.76×10^{-1}	$4.18 \times 10{-}1$	7/9
$\mu = 1,\ \sigma = 1$				
RS-W_1 - LSTM	4.41×10^{-1}	9.32×10^{-1}	3.45×10^{-1}	**9/9**
RS-W_1 - NeuralSDE	$\mathbf{3.71 \times 10^{-1}}$	$\mathbf{7.94 \times 10^{-1}}$	$\mathbf{2.31 \times 10^{-1}}$	**9/9**
Sig-W_1 - LSTM	9.81×10^{-1}	8.55×10^{-1}	2.63×10^{-1}	8/9
Sig-W_1 - NeuralSDE	$\mathbf{8.23 \times 10^{-1}}$	8.29×10^{-1}	3.51×10^{-1}	8/9

Table 1: Out-of-sample test results for different generator-discriminator pairs generating paths of a Brownian motion.

The following plot displays 50 trajectories of the test set and generated paths of the trained generator-discriminator pair (28).

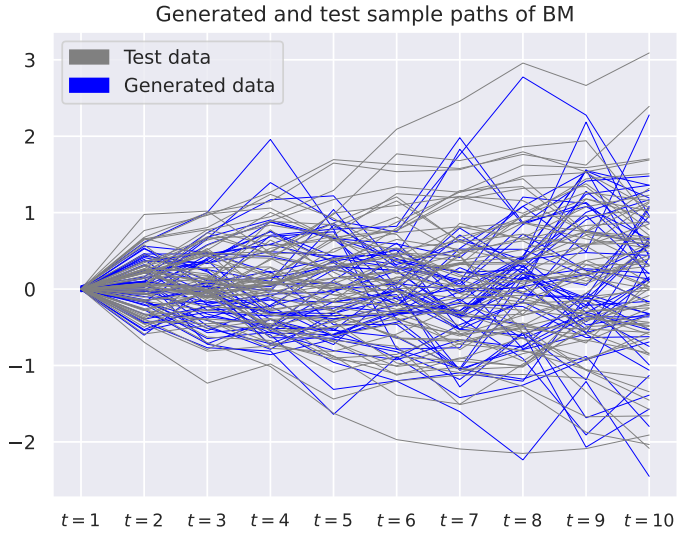

Figure 1: 50 randomly selected trajectories from both generated and real out-of-sample samples of a Brownian motion with drift $\mu = 0$ and variance $\sigma = 1$.

For the training of the generator the training set consists of 8000 and the test set of 2000 sample paths, respectively. Moreover, the numerical optimisation performed 2500 gradient steps.

3.2.2 Autoregressive process

In this section, we study the performance of the generator-discriminator pair (28) generating trajectories of an autoregressive process. In particular, recall that for some $p \in \mathbb{N}$ a process $X = (X_t)_{t\in\mathbb{R}}$ is an AR(p)-process if it is of the form

$$X_t = \sum_{i=1}^{p} \varphi_i X_{t-i} + Z_t, \tag{31}$$

where $\varphi_1, \dots, \varphi_p \in \mathbb{R}$ are constants controlling the influence of past observations of the current one and $Z_1, \dots, Z_T \sim \mathcal{N}(0, \sigma^2)$ are i.i.d for some $\sigma > 0$. The stationarity of the process X can be characterised by the roots of the autoregressive polynomial

$$\Phi(x) := 1 - \varphi_1 x - \cdots - \varphi_p x, \ x \in \mathbb{R}.$$

In fact, the process X is stationary if and only if the roots of the polynomial Φ lie outside the unit disc. For example, for $p = 1$ the process X is stationary if and only if $|\varphi_1| < 1$. The training and test set have the same size as in Section 3.2.1. Below we present out-of-sample results for an AR(1)-process for different values of φ_1.

	Train Metric	Cov. Metric	ACF Metric
$p = 1,\ \varphi_1 = 0.1$			
RS-W_1 - LSTM	8.76×10^{-2}	2.98×10^{0}	7.17×10^{-1}
RS-W_1 - NeuralSDE	$\mathbf{8.18 \times 10^{-2}}$	$\mathbf{2.78 \times 10^{0}}$	$\mathbf{2.11 \times 10^{-1}}$
Sig-W_1 - LSTM	9.58×10^{-1}	4.83×10^{0}	6.04×10^{-1}
Sig-W_1 - NeuralSDE	$\mathbf{9.34 \times 10^{-1}}$	3.61×10^{0}	3.48×10^{-1}
$p = 1,\ \varphi_1 = 0.9$			
RS-W_1 - LSTM	2.53×10^{-1}	$\mathbf{2.98 \times 10^{0}}$	3.51×10^{-1}
RS-W_1 - NeuralSDE	$\mathbf{1.32 \times 10^{-1}}$	3.37×10^{0}	$\mathbf{3.37 \times 10^{-1}}$
Sig-W_1 - LSTM	8.84×10^{-1}	7.91×10^{0}	5.30×10^{-1}
Sig-W_1 - NeuralSDE	$\mathbf{7.58 \times 10^{-1}}$	5.32×10^{0}	5.17×10^{-1}
$p = 1,\ \varphi_1 = -0.1$			
RS-W_1 - LSTM	1.23×10^{-1}	3.02×10^{0}	8.41×10^{-1}
RS-W_1 - NeuralSDE	$\mathbf{9.11 \times 10^{-2}}$	$\mathbf{1.26 \times 10^{0}}$	$\mathbf{1.24 \times 10^{-1}}$
Sig-W_1 - LSTM	9.64×10^{-1}	4.71×10^{0}	5.06×10^{-1}
Sig-W_1 - NeuralSDE	$\mathbf{9.41 \times 10^{-1}}$	3.00×10^{0}	4.37×10^{-1}
$p = 1,\ \varphi_1 = -0.9$			
RS-W_1 - LSTM	2.11×10^{-1}	7.22×10^{0}	9.87×10^{-1}
RS-W_1 - NeuralSDE	$\mathbf{1.68 \times 10^{-1}}$	$\mathbf{4.25 \times 10^{0}}$	$\mathbf{9.42 \times 10^{-1}}$
Sig-W_1 - LSTM	1.32×10^{0}	7.50×10^{0}	1.12×10^{0}
Sig-W_1 - NeuralSDE	$\mathbf{9.89 \times 10^{-1}}$	7.38×10^{0}	9.85×10^{-1}

Table 2: Out-of-sample test results for different generator-discriminator pairs generating paths of an AR(1) process with different parameter values φ_1.

Besides the covariance metric in the case $\varphi_1 = 0.9$, our proposed model yields the lowest test metrics in all cases compared to the other generator-discriminator pairs. For $\varphi_1 = 0.9$ the result is not much worse than for the RS-W_1 combination and still better than the models using the Sig-W_1 metric as the discriminator. Nevertheless, the latter performs better with our proposed generator (24) than the LSTM for all values of φ_1. We refer to Figure 3 in the Appendix for plots of kernel density estimated comparing the generated and test data for two marginals.

3.2.3 S&P 500 return data

Next, we apply the generator model to real-world financial data. In this section, we consider daily log-returns of the S&P 500 index based on its closing prices between 01/01/2005 and 31/12/2023. Note that it is often assumed that log-returns are (almost) stationary over time. Therefore, we apply a rolling window to "cut" the time series into 4316 sample paths of length 10, which is justified by the stationary behaviour of return data. The training set consists of 80% of the samples, while the rest is used for out-of-sample testing. In the following, we present the out-of-sample results on the trained generator for different dimensions N of the randomised signature as defined in (1).

	Train Metric	Cov. Metric	ACF Metric
RS-W_1 - LSTM ($N = 50$)	2.01×10^{-1}	1.03×10^{0}	6.13×10^{-1}
RS-W_1 - NeuralSDE ($N = 50$)	$\mathbf{1.84 \times 10^{-1}}$	9.84×10^{-1}	3.32×10^{-1}
RS-W_1 - LSTM ($N = 80$)	1.99×10^{-1}	9.12×10^{-1}	3.82×10^{-1}
RS-W_1 - NeuralSDE ($N = 80$)	$\mathbf{1.14 \times 10^{-1}}$	$\mathbf{8.40 \times 10^{-1}}$	$\mathbf{1.16 \times 10^{-1}}$
RS-W_1 - LSTM ($N = 100$)	2.85×10^{-1}	9.14×10^{-1}	4.98×10^{-1}
RS-W_1 - NeuralSDE ($N = 100$)	$\mathbf{1.93 \times 10^{-1}}$	9.02×10^{-1}	4.32×10^{-1}
Sig-W_1 - LSTM	4.99×10^{0}	9.22×10^{0}	8.53×10^{-1}
Sig-W_1 - NeuralSDE	$\mathbf{3.14 \times 10^{0}}$	9.98×10^{-1}	3.91×10^{-1}

Table 3: Out-of-sample test results for different generator-discriminator pairs generating paths of S&P 500 log-returns.

We see that for all generative model containing the RSig-W_1 discriminator the choice $N = 80$ yields the best result for all metrics. This observation was one reason for choosing this hyperparameter for generating paths of the Brownian motion and the AR(1)-process. In fact for all values of N, our proposed model yields lower test metric values than the models using the Sig-W_1 discriminator. Lastly, as before we see that the latter performs better paired with the generative model (24) than the LSTM. The plot in Figure 4 in the Appendix displays the kernel density estimates for generated and test data.

3.2.4 FOREX return data

Similar to Section 3.2.3, we compute the log-returns of FOREX EUR-USD on the closing rates between 01/11/2023 and 31/01/2024. The used data can be downloaded from `https://forexsb.com/historical-forex-data`. Again, based on a stationary assumption, we

collect minute-tick data and apply a rolling window the receive 5831 sample paths of length 10. The training and test set are constructed in the same manner as in Section 3.2.3. The following table summarises out-of-sample results on the trained generator for different dimensions N of the randomised signature as defined in (1).

	Train Metric	Cov. Metric	ACF Metric
RS-W_1 - LSTM ($N = 50$)	3.21×10^{-1}	2.85×10^{0}	8.63×10^{-1}
RS-W_1 - NeuralSDE ($N = 50$)	$\mathbf{2.53 \times 10^{-1}}$	4.53×10^{0}	8.15×10^{-2}
RS-W_1 - LSTM ($N = 80$)	$\mathbf{9.01 \times 10^{-2}}$	3.21×10^{0}	8.21×10^{-1}
RS-W_1 - NeuralSDE ($N = 80$)	1.01×10^{-1}	$\mathbf{2.67 \times 10^{0}}$	$\mathbf{6.19 \times 10^{-2}}$
RS-W_1 - LSTM ($N = 100$)	1.91×10^{-1}	3.42×10^{0}	$4.03 \times 10{-1}$
RS-W_1 - NeuralSDE ($N = 100$)	$\mathbf{1.76 \times 10^{-1}}$	3.39×10^{0}	2.25×10^{-1}
Sig-W_1 - LSTM	4.36×10^{0}	9.23×10^{0}	2.98×10^{-1}
Sig-W_1 - NeuralSDE	$\mathbf{2.85 \times 10^{0}}$	8.52×10^{0}	2.86×10^{-1}

Table 4: Out-of-sample test results for different generator-discriminator pairs generating paths of FOREX EUR/USD log-returns.

Similar to the results for the S&P 500 log-returns, Table 4 indicates that our model yields the best results for both test metrics if the dimension of randomised signature is chosen as $N = 80$. Moreover, the model configurations incorporating the RS-W_1 metric leads to more realistic results than the Sig-W_1 discriminator. The combination Sig-W_1 and the generator (24) has lower test metrics compared to the LSTM generator. The kernel density estimates can be found in Figure 5 in the Appendix.

4 Conditional time series generation using the randomised signature

In this section, we study the problem of generating future trajectories of a path given some information of its past. Let $K \subseteq \mathbb{R}^d$ be a compact subset. For $p, q \in \mathbb{N}$ we denote by $\mathcal{Y} := K \times \cdots \times K \subseteq (\mathbb{R}^d)^p$ and $\mathcal{Z} := K \times \cdots \times K \subseteq (\mathbb{R}^d)^q$. We consider a $\mathcal{Y} \times \mathcal{Z}$-valued adapted stochastic process $X = (X_t)_{t=1,\dots,p,\dots,p+q}$ defined on $(\Omega, \mathcal{F}, \mathbb{F}, \mathbb{P})$. Moreover, we define

$$X^{\text{past},p} := (X_1, \dots, X_p),\ X^{\text{future},q} := (X_{p+1}, \dots, X_{p+q}),$$

hence it holds $X = (X^{\text{past},p}, X^{\text{future},q})$. We denote by $\mathbb{P}^r_c(x)$ the conditional distribution of $X^{\text{future},q}$ given the information that $X^{\text{past},p} = x$ for some $x \in \mathcal{Y}$. Similar to the unconditional case in Section 3, our aim is to generate trajectories of a K^q-valued random process $\hat{X}^{q,\theta} = (\hat{X}^{q,\theta}_t)_{t=1,\dots,T}$ with conditional distribution $\mathbb{P}^{f,\theta}_c(x)$ on the information $X^{\text{past},p} = x$ such that $d(\mathbb{P}^r_c(x), \mathbb{P}^{f,\theta}_c(x))$ is small, where d is a metric on the space of probability measures on $\mathcal{Z}$ specified later.

Consider the n-dimensional $\mathbb{F}$-Brownian motion $W = (W^1_t, \dots, W^n_t)_{t\in[0,T]}$ and the random vector $V \sim \mathcal{N}_m(0, \mathrm{Id})$ defined in Section 3. For some $D \in \mathbb{N}$ and given $x \in \mathcal{Y}$, we consider the D-dimensional stochastic process $\tilde{R} = (\tilde{R}_t)_{t=1,\dots,q}$ and the d-dimensional

process $\bar{X}_t^{q,\theta} = (\bar{X}_t^{q,\theta})_{t=1,\dots,q}$ defined as follows

$$\begin{cases} \tilde{R}_1 = \tilde{\Psi}^\theta(V, \Delta\mathrm{RS}_p(x)), \\ \tilde{R}_t = \tilde{R}_{t-1} + \sigma(\tilde{\rho}_1^\theta \tilde{B}_1 \tilde{R}_{t-1} + \tilde{\rho}_2^\theta \tilde{\lambda}_1) + \sum_{i=1}^n \sigma(\tilde{\rho}_3^\theta \tilde{B}_2^i \tilde{R}_{t-1} + \tilde{\rho}_4^\theta \tilde{\lambda}_2^i)\tilde{\rho}_5^\theta (W_t^i - W_{t-1}^i), \\ \bar{X}_t^{q,\theta} = \tilde{A}_t^\theta \tilde{R}_t + \tilde{\beta}_t^\theta, \ t = 1, \dots, q, \end{cases} \tag{32}$$

for a neural network $\tilde{\Psi}^\theta : \mathbb{R}^m \times \mathbb{R}^N \to \mathbb{R}^D$, random matrices $\tilde{B}_1, \tilde{B}_2^1, \dots, \tilde{B}_2^n \in \mathbb{R}^{D\times D}$, $\tilde{\lambda}_1, \tilde{\lambda}_2^1, \dots, \tilde{\lambda}_2^n \in \mathbb{R}^D$, parameters $\tilde{\rho}_1^\theta, \dots, \tilde{\rho}_5^\theta \in \mathbb{R}$, $\tilde{A}_1^\theta, \dots, \tilde{A}_q^\theta \in \mathbb{R}^{d\times D}$, $\tilde{\beta}_1^\theta, \dots, \tilde{\beta}_q^\theta \in \mathbb{R}^d$ and some activation function $\sigma : \mathbb{R} \to \mathbb{R}$ applied component-wise. The entries of the random matrices are i.i.d. with some absolutely continuous distribution μ_4 with full support and independent of X, W and V. As before, the superscript θ denotes the trainable weights of the corresponding neural network or matrix. Finally, the process $\hat{X}^{q,\theta}$ is defined by the projection of $\bar{X}^{q,\theta}$ on a compact set. In particular, for some $r > 0$ consider the projection defined for any $x \in \mathbb{R}^d$ as

$$\pi_r(x) = \begin{cases} x, \text{ if } \|x\| < r, \\ r\frac{x}{\|x\|}, \text{ if } \|x\| \geq r. \end{cases}$$

Similar to (24), the process $\tilde{R}$ is a variant of randomised signature of $(W_t - W_{t-1})_{t=1,\dots,q}$ with initial condition chosen to incorporate the available past information x. For any $t = 1, \dots, q$ we define $\hat{X}_t^{q,\theta} := \pi_r(\bar{X}_t^{q,\theta})$, where the constant r is chosen such that $\hat{X}^{q,\theta} \in \mathcal{Z}$. The conditional version of the problem (25) can be written

$$\min_\theta d_{\mathrm{path}}\left(\mathbb{P}_c^r(x), \mathbb{P}_c^{f,\theta}(x)\right) \tag{33}$$

for an input path $x \in \mathcal{Y}$. As described before, the entries of the matrices $\tilde{B}_1, \tilde{B}_2^1, \dots, \tilde{B}_2^n$ and vectors $\tilde{\lambda}_1, \tilde{\lambda}_2^1, \dots, \tilde{\lambda}_2^n$ are once sampled from an absolutely continuous distribution and then fixed for the training procedure.

4.1 The conditional randomised signature Wasserstein-1 (C-RS-W_1) metric

In this section, we introduce a conditional version of the RS-W_1 metric introduced in Section 3. Consider the conditional distributions of $\mathbb{P}_c^r(x)$ and $\mathbb{P}_c^{f,\theta}(x)$, as defined above. Based on the construction of the unconditional metric, we define the conditional randomised signature Wasserstein-1 metric as

$$\text{C-RS-}W_1(\mathbb{P}_c^r(x), \mathbb{P}_c^{f,\theta}(x)) := \left\| \mathbb{E}_{\mathbb{P}_c^r(x)}[\Delta\mathrm{RS}_q(X^{\mathrm{future},q})] - \mathbb{E}_{\mathbb{P}_c^{f,\theta}(x)}[\Delta\mathrm{RS}_q(X^{\mathrm{future},q})] \right\|_2.$$

Note that integrability of $\Delta\mathrm{RS}_T(X^{\mathrm{future},q})$ is ensured by its definition and the fact that $X^{\mathrm{future},q}$ and $\hat{X}^{q,\theta}$ only attain values on a compact domain and that σ is bounded by Assumption 6. For an input path $x = (x_1, \dots, x_p) \in \mathcal{Y}$, we obtain the following generator-discriminator pair

$$\textbf{Generator: } \hat{X}^{q,\theta} = (\hat{X}_1^{q,\theta}, \dots, \hat{X}_q^{q,\theta}) \qquad \textbf{Discriminator: } \text{C-RS-}W_1(\mathbb{P}_c^r(x), \mathbb{P}_c^{f,\theta}(x)) \tag{34}$$

using the system (32) as the generator and the C-RS-W_1 metric as the discriminator so that the training objective (33) can be written as

$$\min_{\theta} \left\| \mathbb{E}_{\mathbb{P}_c^r(x)}[\Delta \mathrm{RS}_q(X^{\mathrm{future},q})] - \mathbb{E}_{\mathbb{P}_c^{f,\theta}(x)}[\Delta \mathrm{RS}_q(X^{\mathrm{future},q})] \right\|_2 . \tag{35}$$

We devote the following section to describe how the learning problem (35) is implemented numerically, since in the conditional setting simulating the conditional expectation under $\mathbb{P}_c^r(x)$ for some $x \in \mathcal{Y}$ is not straightforward. A similar approach using path signatures was pursued in Liao et al. (2023).

4.2 Practical implementation

As for the unconditional case, we can estimate $\mathbb{E}_{\mathbb{P}_c^{f,\theta}(x)}[\Delta \mathrm{RS}_q(X^{\mathrm{future},q})]$ by a Monte-Carlo sum. For a fixed input path x we generate samples of the future path using (32), compute their corresponding randomised signatures and take their average. However, this is not possible for the conditional expectation $\mathbb{E}_{\mathbb{P}_c^r(x)}[\Delta \mathrm{RS}_q(X^{\mathrm{future},q})]$. Note that for a fixed past we have only one future trajectory for the real data so that a Monte-Carlo approach is not applicable. In fact, in typical financial applications we observe only one long datastream/timeseries $x = (x_1, \dots, x_T) \in K^T$ for some $T \in \mathbb{N}$, which can be interpreted as the realisation of some stochastic process. Normally it holds that T is significantly bigger than p and q. Therefore, in practice we apply a rolling window to the datastream to obtain the samples

$$x_i^{\mathrm{past},p} := (x_{i-p+1}, \dots, x_i) \in \mathcal{Y}, \; x_i^{\mathrm{future},q} := (x_{i+1}, \dots, x_{i+q}) \in \mathcal{Z}$$

for $i = p, \dots, T-q$. Assuming that the observations are stationary in time, we interpret $(x_i^{\mathrm{past},p})_{i=p}^{T-q}$ as samples of $X^{\mathrm{past},p}$ and $(x_i^{\mathrm{future},q})_{i=p}^{T-q}$ of $X^{\mathrm{future},q}$ respectively. As described above, for every sample path $x_i^{\mathrm{past},p}$ there is only one corresponding future sample path $x_i^{\mathrm{future},q}$ so that Monte-Carlo techniques fail. Therefore, we use a method proposed in Levin et al. (2013), which is based on the Doob-Dynkin Lemma: we assume that there exists a measurable function $l : \mathcal{Y} \to \mathbb{R}^N$ such that

$$l(x_i^{\mathrm{past},p}) = \mathbb{E}_{\mathbb{P}_c^r(x_i^{\mathrm{past},p})} \left[\Delta \mathrm{RS}_q(X^{\mathrm{future},q}) \right] \tag{36}$$

for every $i = p, \dots, T-q$. Assuming that l is continuous, it is intuitive to approximate (36) by a linear function on $\Delta \mathrm{RS}_q(x_i^{\mathrm{past},p})$, which is motivated by the universality results presented in Section 2. In particular, approximating l reduces to a linear regression problem. In fact, we assume that for every $i = p, \dots, T-q$ it holds

$$\Delta \mathrm{RS}_q(x_i^{\mathrm{future},q}) = \alpha + \beta \Delta \mathrm{RS}_p(x_i^{\mathrm{past},p}) + \varepsilon_i \tag{37}$$

for $\alpha \in \mathbb{R}^N$, $\beta \in \mathbb{R}^{N \times N}$ and error terms $\varepsilon_i \in \mathbb{R}^N$. We denote by $\hat{\alpha} \in \mathbb{R}^N, \hat{\beta} \in \mathbb{R}^{N \times N}$ the solution to the problem

$$\min_{\alpha \in \mathbb{R}^N, \beta \in \mathbb{R}^{N \times N}} \sum_{i=p}^{T-q} \left\| \alpha + \beta \Delta \mathrm{RS}_p(x_i^{\mathrm{past},p}) - \Delta \mathrm{RS}_q(x_i^{\mathrm{future},q}) \right\|_2^2 . \tag{38}$$

Note that the regression problem (38) can be solved prior to the actual training problem (33), which improves the computation time of the problem in general. Once the solution $(\hat{\alpha}, \hat{\beta})$ is found, (35) becomes the following supervised learning problem

$$\min_{\theta} \left\| \hat{\alpha} + \hat{\beta} \Delta \mathrm{RS}_p(x) - \mathbb{E}_{\mathbb{P}_c^{f,\theta}(x)}[\Delta \mathrm{RS}_q(x)] \right\|_2 .$$

The entire implementation of the training procedure is described in Algorithm 1 which can be found in Appendix D.

4.3 Empirical results

In this section, we present out-of-sample results for the generator-discriminator pair (34). In particular, we generate future trajectories of a timeseries data conditional on a path's past for both synthetic and real-world training data. For the generator proposed in (32) we compare the result for using the C-RS-W_1 metric as the discriminator to the conditional version of Sig-W_1 metric proposed in Liao et al. (2023). As in Section 3, we state the value of the corresponding discriminator metric and the autocorrelation distance proposed in (40) based on out-of-sample test data. Note that the covariance metric cannot be estimated using (39) since only one trajectory of the real data is available so that a Monte-Carlo approach fails at this point. For an overview of the hyperparameters used during the training we refer to Table 10 in the Appendix. Similar to Section 3, we set the parameter $\tilde{\rho}_5^{\theta}$ in (32) equal to one for the following experiments. The different model configurations are trained with 2500 gradient steps or until no further improvement was achieved.

4.3.1 Brownian motion

Similar to Section 3.2.1, we train the generator model in (32) to generate sample paths of a Brownian motion conditional on some information about its past. We use (30) to generate training and test paths of length 15. For each path the first 5 datapoints are considered as the past, while the remaining 10 are supposed to be the corresponding future. The training set consists of 8000 samples and the test set of 2000. The following table summarises the result for Brownian motion paths with and without drift. Moreover, we use the Shapiro-Wilk test to study if the marginals of the generated paths follow a normal distribution with high statistical confidence.

	Train Metric	ACF Metric	SW Tests Passed
$\mu = 0,\ \sigma = 1$			
C-RS-W_1 - NeuralSDE	1.46×10^{0}	$\mathbf{3.13 \times 10^{-2}}$	**9/10**
C-Sig-W_1 - NeuralSDE	2.28×10^{0}	6.25×10^{-1}	8/10
$\mu = 1,\ \sigma = 1$			
C-RS-W_1 - NeuralSDE	1.71×10^{0}	$\mathbf{2.68 \times 10^{-1}}$	**8/10**
C-Sig-W_1 - NeuralSDE	2.42×10^{0}	3.25×10^{-1}	**8/10**

Table 5: Out-of-sample results for different generator-discriminator pairs for conditional generation of Brownian motion.

The results indicate that the model configuration incorporating the C-RS-W_1 metric yields the best results for both the autocorrelation metric and the number of normality tests that could not reject the null hypothesis of a normal distribution. Hence, it is indicated that the C-RS-W_1 leads to more realistic paths especially in terms of temporal dependence with or without drift. The following plot displays trajectories generated by our proposed model based on the past of a single trajectory. The latter is highlighted in blue together with its actual future behaviour.

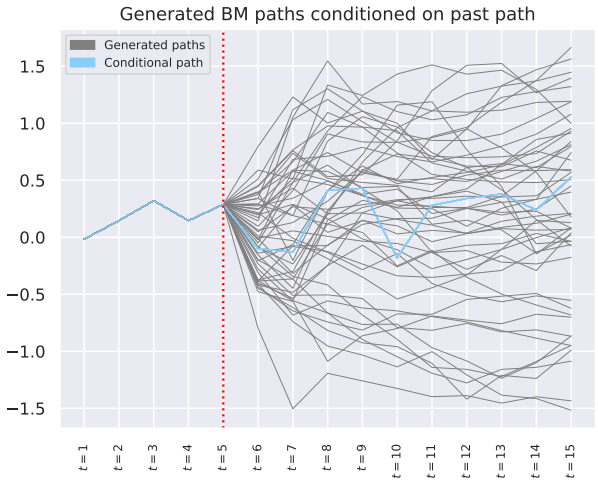

Figure 2: 50 randomly selected generated future trajectories based on a past path of a Brownian motion with drift $\mu = 0$ and variance $\sigma = 1$.

4.3.2 S&P 500 return data

We generate sample paths of future log-returns of the S&P 500 index based on past returns. In particular, given the last 5 daily log-returns, we generate possible future scenarios for the coming 10 days. We use historical data from 01/01/2005 to 31/12/2023 so that the training and test sets combined consist of 3415 samples produced by a rolling window, where 80% are for training and 20% for out-of-sample testing. The following table presents out-of-sample results for three different dimensions for the randomised signature in C-RS-W_1 and the Sig-W_1 metric.

	Train Metric	ACF Metric
C-RS-W_1 - NeuralSDE ($N = 50$)	$\mathbf{1.13 \times 10^{0}}$	$\mathbf{8.96 \times 10^{-1}}$
C-RS-W_1 - NeuralSDE ($N = 80$)	1.42×10^{0}	9.30×10^{-1}
C-RS-W_1 - NeuralSDE ($N = 100$)	1.92×10^{0}	9.37×10^{-1}
C-Sig-W_1 - NeuralSDE	9.41×10^{0}	1.14×10^{0}

Table 6: Out-of-sample results for different generator-discriminator pairs for conditional generation of S&P 500 log-returns.

It is indicated that the C-RS-W_1 leads to the lowest ACF distances for all three different dimensions N. Different to the results before, the best metric is obtained for $N = 50$. However, the temporal dependence of the generated paths seems to be very similar for all three cases.

4.3.3 FOREX return data

As for the S&P 500 index, we also generate future log-return paths for the FOREX rate of Euro and US Dollar for the time period from 01/11/2005 to 31/12/2023. The observed market data yields 5826 samples, of which 80% are used for the training and 20% for testing. The following table displays the corresponding training metric applied to the generated data and out-of-sample data as well as the ACF distance. For the C-RS-W_1 discriminator we consider three different dimensions N.

	Train Metric	ACF Metric
C-RS-W_1 - NeuralSDE ($N = 50$)	5.51×10^{-1}	9.81×10^{-1}
C-RS-W_1 - NeuralSDE ($N = 80$)	$\mathbf{4.98 \times 10^{-1}}$	$\mathbf{7.65 \times 10^{-1}}$
C-RS-W_1 - NeuralSDE ($N = 100$)	7.70×10^{-1}	1.05×10^{0}
C-Sig-W_1 - NeuralSDE	4.41×10^{0}	1.17×10^{0}

Table 7: Out-of-sample results for different generator-discriminator pairs for conditional generation of FOREX EUR/USD log-returns.

We observe that the best C-RS-W_1 metric value is obtained for the configuration with dimension $N = 80$, which also leads to the lowest ACF distance. Together with hyperparameter tuning for the experiment for Brownian motion, this observation led to the dimension choice for the results in Section 4.3.1. In all three cases C-RS-W_1 seems to yield a more realistic temporal dependence structure than Sig-W_1.

Acknowledgments and Disclosure of Funding

Niklas Walter gratefully acknowledges the financial support of the "Verein zur Förderung der Versicherungswissenschaft e.V.".

Appendix A. Overview of the training hyperparameters

The following tables present the hyperparameters used for the training procedures.

(Hyper-)Parameter	Value
Learning rate	10^{-4}
Batch size (uncond. generator)	1500
Batch size (cond. generator)	1000
Maximal number of gradient steps	2500

Table 8: Hyperparameters used for the training.

(Hyper-)Parameter	Value
Number of timesteps T	10
Dimension of RS N	80
Dimension of generator reservoir D	80
Initial noise dimension m	5
Signature truncation level L	4
Distribution $\mu_{A,\xi}$ of random weights for RSig	$\mathcal{N}(0,1)$
Distribution $\mu_{B,\lambda}$ of random weights in (24)	$\mathcal{N}(0,1)$
Activation function σ	Sigmoid
Input size of LSTM	5
Number of hidden layers of LSTM	2
Dimension of hidden layers in LSTM	64
Activation function LSTM	Tanh

Table 9: Hyperparameters used for the unconditional generative model.

(Hyper-)Parameter	Value
Number of past steps p	5
Number of future steps q	10
Dimension of RS N	80
Dimension of generator reservoir D	80
Initial noise dimension m	15
Signature truncation level L	4
Distribution $\mu_{A,\xi}$ of random weights for RSig	$\mathcal{N}(0,1)$
Distribution $\mu_{\tilde{B},\tilde{\lambda}}$ of random weights in (32)	$\mathcal{N}(0,1)$
Activation function σ	Sigmoid

Table 10: Hyperparameters used for the conditional generative model.

Appendix B. Evaluation metrics

In this section, we introduce the main test metrics measuring the performance of the different generator-discriminator pairs in Section 3. Using the previous notation, consider two d-dimensional stochastic processes $X = (X_t)_{t=1,\dots,T}$ and $X^\theta = (X_t^\theta)_{t=1,\dots,T}$.

B.1 Covariance difference

We introduce the following covariance-distance between X and X^θ as

$$\text{Cov-Dist}(X, X^\theta) := \sqrt{\sum_{j,k=1}^{d} \sum_{s,t=1}^{T} \left| \text{Cov}(X_s^j, X_t^k) - \text{Cov}(X_s^{\theta,j}, X_t^{\theta,k}) \right|^2}, \tag{39}$$

where $\text{Cov}(X_s^j, X_t^k)$ denotes the covariance between the j-th entry of X at time s and the k-th entry of X at time t. In practice, we observe samples $(x^{(i)})_{i=1}^M$ of X and $(x^{\theta,(i)})_{i=1}^M$ of X^θ respectively. for some $M \in \mathbb{N}$. Then the covariance terms in (39) are estimated using the estimators

$$\widehat{\text{Cov}}(X_s^j, X_t^k) = \frac{1}{M} \sum_{i=1}^{M} (x_s^{(i),j} - \bar{x}_s^j)(x_t^{(i),k} - \bar{x}_t^k)$$

and

$$\widehat{\text{Cov}}(X_s^{\theta,j}, X_t^{\theta,k}) = \frac{1}{M} \sum_{i=1}^{M} (x_s^{\theta,(i),j} - \bar{x}_s^{\theta,j})(x_t^{\theta,(i),k} - \bar{x}_t^{\theta,k}),$$

where $\bar{x}_s^j := \frac{1}{M} \sum_{i=1}^M x_s^{(i),j}$ denotes the sample mean.

B.2 Autocorrelation difference

The autocorrelation-distance between X and X^θ is defined as

$$\text{ACF-Dist}(X, X^\theta) := \sqrt{\sum_{j=1}^{d} \sum_{k=1}^{\lfloor T/2 \rfloor} \left| \frac{\rho_{X^j}(k)}{\rho_{X^j}(0)} - \frac{\rho_{X^{\theta,j}}(k)}{\rho_{X^{\theta,j}}(0)} \right|^2}, \tag{40}$$

where $\rho_{X^i}(k)$ denotes the autocovariance of the i-th entry of X with lag k and $\rho_{X^{\theta,i}}(k)$ of $X^{\theta,i}$ respectively. Let $(x^{(i)})_{i=1}^M$ be samples of the process X and $(x^{\theta,(i)})_{i=1}^M$ of X^θ respectively. Then the autocovariances can be estimated by

$$\hat{\rho}_{X^j}(k) = \frac{1}{M(T-k)} \sum_{i=1}^{M} \sum_{t=1}^{T-k} x_t^{(i),j} x_{t+k}^{(i),j} - \frac{1}{M^2(T-k)^2} \left(\sum_{i=1}^{M} \sum_{t=1}^{T-k} x_t^{(i),j} \right) \left(\sum_{i=1}^{M} \sum_{t=1}^{T-k} x_{t+k}^{(i),j} \right)$$

and

$$\hat{\rho}_{X^{\theta,j}}(k) = \frac{1}{M(T-k)} \sum_{i=1}^{M} \sum_{t=1}^{T-k} x_t^{\theta,(i),j} x_{t+k}^{\theta,(i),j} - \frac{1}{M^2(T-k)^2} \left(\sum_{i=1}^{M} \sum_{t=1}^{T-k} x_t^{\theta,(i),j} \right) \left(\sum_{i=1}^{M} \sum_{t=1}^{T-k} x_{t+k}^{\theta,(i),j} \right)$$

Note that for the conditional generator we do not have samples for the future real path, since there is only one trajectory. In this case the autocovariance is estimated with $M = 1$.

Appendix C. Kernel density estimate plots for the unconditional generator

The following plots display kernel density estimates (KDEs) for out-of-sample test data and generated data for paths of an AR(1)-process, log-returns of S&P 500 and FOREX EUR/USD exchange rates. The synthetic data was generated by our proposed unconditional generator model with the generator-discriminator pair (28). In particular, we considered the dimension of the randomised signature in (27) to be $N = 80$ and the dimension of the reservoir process R in (24) to be $D = 80$. The corresponding model was trained with 2500 gradient steps.

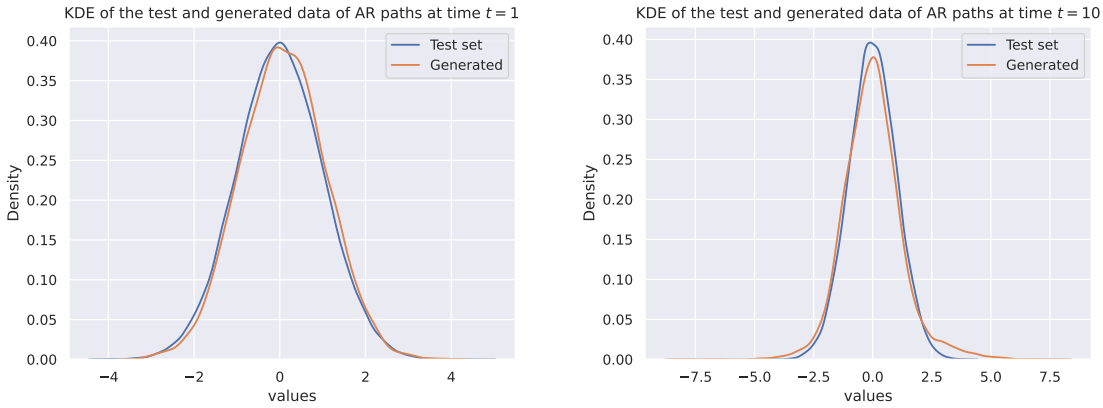

Figure 3: KDEs for generated and test data of values of a AR(1)-process with $\varphi = 0.1$ at two different timepoints.

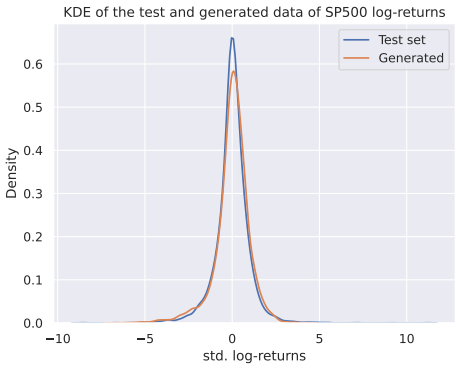

Figure 4: KDEs for generated and test data of S&P 500 log-returns.

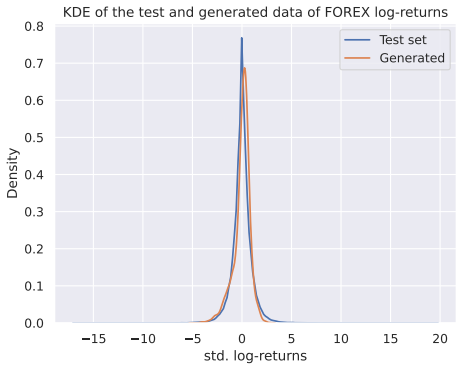

Figure 5: KDEs for generated and test data of FOREX EUR/USD log-returns.

Appendix D. Pseudocode for the conditional generator

Inspired by Liao et al. (2023), the following pseudocode describes the implementation of the conditional generator proposed in Section 4.

Algorithm 1 Conditional Generator

Input: Length of the past path p, length of the future path q, number of samples M, samples $(x_i^{\text{past},p})_{i=1}^M$ of $X_{real}^{\text{past},p}$ and $(x_i^{\text{future},q})_{i=1}^M$ of $X_{real}^{\text{future},q}$, dimension of the randomised signature N, sampled random matrices $\tilde{B}_1, \tilde{B}_2^1, \dots, \tilde{B}_2^w \in \mathbb{R}^{N\times N}, \tilde{\lambda}_1, \tilde{\lambda}_2^1, \dots, \tilde{\lambda}_2^w \in \mathbb{R}^N$, learning rate η, number of training epochs N_{epochs}, batch size B

Output: Optimal parameters for the generator $\hat{X}^{\theta,q}$

Compute $\Delta\text{RS}_p(x_i^{\text{past},p})_{i=1}^M$ and $\Delta\text{RS}_q(x_i^{\text{future},q})_{i=1}^M$.

Solve the OLS problem

$$\Delta\text{RS}_q(x_i^{\text{future},q}) = \alpha + \beta\Delta\text{RS}_p(x_i^{\text{past},p}) + \varepsilon_i$$

for every sample $i = 1, \dots, M$ to obtain the optimal $(\hat{\alpha}, \hat{\beta})$.

Initialise parameters θ of the generator $\hat{X}^\theta$.

for $i = 1, \dots, N_{epochs}$ **do**

  Initialise loss $\ell(\theta) \leftarrow 0$.

  Sample training batch $\{x_j^{\text{past},p}\}_{j=1,\dots,B}$

  for $j = 1, \dots, B$ **do**

    Generate M samples $(x_i^{\text{future},q})_{i=1,\dots,M}$ of the future path using the current configuration of generator $\hat{X}^\theta$ given $x_j^{\text{past},p}$

    Compute the MC sum

$$\frac{1}{M}\sum_{i=1}^M \Delta\text{RS}_q(x_i^{\text{future},q}).$$

    Update the loss

$$\ell(\theta) \leftarrow \ell(\theta) + \left\| \hat{\alpha} + \hat{\beta}x_j^{\text{past},p} - \frac{1}{M}\sum_{i=1}^M \Delta\text{RS}_q(x_i^{\text{future},q})] \right\|_2$$

  end for

  Update the parameters

$$\theta \leftarrow \theta - \eta\frac{\partial\ell(\theta)}{\partial\theta}.$$

end for

return θ

References

E. Akyildirim, M. Gambara, J. Teichmann, and S. Zhou. Applications of signature methods to market anomaly detection. *ArXiv*, 2201.02441, 1 2022.

E. Akyildirim, M. Gambara, J. Teichmann, and S. Zhou. Randomized signature methods in optimal portfolio selection. *SSRN Electronic Journal*, 2024.

M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein generative adversarial networks. In D. Precup and Y. W. Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pages 214–223, 06–11 Aug 2017.

S. A. Assefa. Generating synthetic data in finance: opportunities, challenges and pitfalls. *Proceedings of the First ACM International Conference on AI in Finance*, 2020.

H. Buehler, B. Horvath, T. Lyons, I. P. Arribas, and B. Wood. Generating financial markets with signatures. *Other Financial Economics eJournal*, 2020.

H. Bühler, B. Horvath, T. Lyons, I. P. Arribas, and B. Wood. A data-driven market simulator for small data environments. *ERN: Neural Networks & Related Topics (Topic)*, 2020.

L. Carratino, A. Rudi, and L. Rosasco. Learning with sgd and random features. *Advances in Neural Information Processing Systems*, 31, 2018.

T. Cass and C. Salvi. Lecture notes on rough paths and applications to machine learning. *ArXiv*, 2404.06583, 4 2024.

T. Cass and W. F. Turner. Free probability, path developments and signature kernels as universal scaling limits. *ArXiv*, 2402.12311, 2 2024.

N. M. Cirone, M. Lemercier, and C. Salvi. Neural signature kernels as infinite-width-depth-limits of controlled resnets. In *Proceedings of the 40th International Conference on Machine Learning*, 2023.

S. N. Cohen, C. Reisinger, and S. Wang. Arbitrage-free neural-sde market models. *Applied Mathematical Finance*, 30:1–46, 1 2023.

A. Coletta, M. Prata, M. Conti, E. Mercanti, N. Bartolini, A. Moulin, S. Vyetrenko, and T. Balch. Towards realistic market simulations: a generative adversarial networks approach. In *Proceedings of the Second ACM International Conference on AI in Finance*, pages 1–9, 2021.

E. M. Compagnoni, A. Scampicchio, L. Biggio, A. Orvieto, T. Hofmann, and J. Teichmann. On the effectiveness of randomized signatures as reservoir for learning rough dynamics. *2023 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 6 2023.

C. Cuchiero and J. Möller. Signature methods in stochastic portfolio theory. *ArXiv*, 2310.02322, 10 2023.

C. Cuchiero, L. Gonon, L. Grigoryeva, J.-P. Ortega, and J. Teichmann. Discrete-time signatures and randomness in reservoir computing. *IEEE Transactions on Neural Networks and Learning Systems*, 33:6321–6330, 2020.

C. Cuchiero, L. Gonon, L. Grigoryeva, J.-P. Ortega, and J. Teichmann. Expressive power of randomized signature. In *35th Conference on Neural Information Processing Systems, Sydney, Australia*, 2021.

G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems*, 2:303–314, 12 1989.

C. Daskalakis and I. Panageas. The limit points of (optimistic) gradient descent in min-max optimization. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, page 9256–9266, Red Hook, NY, USA, 2018. Curran Associates Inc.

C. Daskalakis, A. Ilyas, V. Syrgkanis, and H. Zeng. Training gans with optimism. *ArXiv*, 1711.00141, 10 2017.

C. Donahue, J. J. McAuley, and M. S. Puckette. Adversarial audio synthesis. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

P. Díaz, T. L. Bagén, and J. L. Vives. Neural stochastic differential equations for conditional time series generation using the signature-wasserstein-1 metric. *Journal of Computational Finance*, 27, 2023.

C. Esteban, S. Hyland, and G. Rätsch. Real-valued (medical) time series generation with recurrent conditional gans. *ArXiv*, 1706.02633, 8 2017.

S. Flaig and G. Junike. Scenario generation for market risk models using generative neural networks. *Risks*, 10:199, 10 2022.

W. Fu, A. Hirsa, and J. Osterrieder. Simulating financial time series using attention. *ArXiv*, 2207.00493, 7 2022.

L. Gonon. Random feature neural networks learn black-scholes type pdes without curse of dimensionality. *Journal of Machine Learning Research*, 24(189):1–51, 2023.

L. Gonon and J.-P. Ortega. Reservoir computing universality with stochastic inputs. *IEEE Transactions on Neural Networks and Learning Systems*, 31:100–112, 1 2020.

L. Gonon, L. Grigoryeva, and J.-P. Ortega. Approximation bounds for random neural networks and reservoir systems. *The Annals of Applied Probability*, 33(1):28–69, 2023.

I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. *Advances in Neural Information Processing Systems*, 27, 2014.

L. Grigoryeva and J.-P. Ortega. Universal discrete-time reservoir computers with stochastic inputs and linear readouts using non-homogeneous state-affine systems. *Journal of Machine Learning Research*, 19(24):1–40, 2018a.

L. Grigoryeva and J.-P. Ortega. Echo state networks are universal. *Neural Networks*, 108: 495–508, 2018b.

F. Guth, S. Coste, V. D. Bortoli, and S. Mallat. Wavelet score-based generative modeling. *Advances in Neural Information Processing Systems*, 35:478–491, 2022.

B. Hambly and T. Lyons. Uniqueness for the signature of a path of bounded variation and the reduced path group. *Annals of Mathematics*, 171:109–167, 2010.

M. Hamdouche, P. Henry-Labordere, and H. Pham. Generative modeling for time series via schrödinger bridge. *SSRN Electronic Journal*, 2023. ISSN 1556-5068. doi: 10.2139/ssrn.4412434.

A. Hart, J. Hook, and J. Dawes. Embedding and approximation theorems for echo state networks. *Neural Networks*, 128:234–247, 8 2020.

S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9: 1735–1780, 11 1997.

G.-B. Huang, L. Chen, and C.-K. Siew. Universal approximation using incremental constructive feedforward networks with random hidden nodes. *IEEE Transactions on Neural Networks*, 17:879–892, 7 2006.

Z. Issa, B. Horvath, M. Lemercier, and C. Salvi. Non-adversarial training of neural sdes with signature kernel scores. *Advances in Neural Information Processing Systems*, 36, 2024.

L. V. Kantorovich. Mathematical methods of organizing and planning production. *Management Science*, 6:366–422, 7 1960.

P. Kidger. On neural differential equations (phd thesis), 2021.

D. Kingma and J. Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, San Diega, CA, USA, 2015.

A. Kondratyev and C. Schwarz. The market generator. *Econometrics: Econometric & Statistical Methods - Special Topics eJournal*, 2019.

A. Koshiyama, N. Firoozye, and P. Treleaven. Generative adversarial networks for financial trading strategies fine-tuning and combination. *Quantitative Finance*, 21:797–813, 5 2021.

M. Leshno, V. Y. Lin, A. Pinkus, and S. Schocken. Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural Networks*, 6: 861–867, 1 1993.

D. Levin, T. Lyons, and H. Ni. Learning from the past, predicting the statistics for the future, learning an evolving system. *arXiv preprint arXiv:1309.0260*, 2013.

S. Liao, H. Ni, M. Sabate-Vidales, L. Szpruch, M. Wiese, and B. Xiao. Sig-wasserstein gans for conditional time series generation. *Mathematical Finance*, 11 2023.

S. Liu, A. Borovykh, L. A. Grzelak, and C. W. Oosterlee. A neural network-based framework for financial model calibration. *Journal of Mathematics in Industry*, 9:9, 12 2019a.

X. Liu, T. Xiao, S. Si, Q. Cao, S. Kumar, and C.-J. Hsieh. Neural sde: Stabilizing neural ode networks with stochastic noise. *ArXiv*, abs/1906.02355, 2019b.

H. Lou, S. Li, and H. Ni. Path development network with finite-dimensional lie group representation. *ArXiv*, 2204.00740, 4 2022.

C. I. Lu and J. Sester. Generative model for financial time series trained with MMD using a signature kernel. *ArXiv:2407.19848v2*, 2024.

T. J. Lyons. Differential equations driven by rough signals. *Revista Matemática Iberoamericana*, 14(2):215–310, 1998.

G. Marti. Corrgan: Sampling realistic financial correlation matrices using generative adversarial networks. *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8459–8463, 2019.

M. B. Matthews. On the uniform approximation of nonlinear discrete-time fading-memory systems using neural network models (phd thesis), 1992.

S. Mei, T. Misiakiewicz, and A. Montanari. Generalization error of random feature and kernel methods: Hypercontractivity and kernel matrix concentration. *Applied and Computational Harmonic Analysis*, 59:3–84, 7 2022.

O. Mogren. C-rnn-gan: Continuous recurrent neural networks with adversarial training. *ArXiv*, abs/1611.09904, 2016.

A. Neufeld and P. Schmocker. Universal approximation property of random neural networks. *ArXiv*, 2312.08410, 12 2023.

H. Ni, L. Szpruch, M. Sabate-Vidales, B. Xiao, M. Wiese, and S. Liao. Sig-wasserstein gans for time series generation. *Proceedings of the Second ACM International Conference on AI in Finance*, pages 1–8, 11 2021.

A. Rahimi and B. Recht. Random features for large-scale kernel machines. *Advances in Neural Information Processing Systems*, 20, 2007.

A. Rudi and L. Rosasco. Generalization properties of learning with random features. *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 3218–3228, 2017.

I. Sandberg. Approximation theorems for discrete-time systems. *Proceedings of the 34th Midwest Symposium on Circuits and Systems*, pages 6–7, 1991a.

I. W. Sandberg. Structure theorems for nonlinear systems. *Multidimens. Syst. Signal Process.*, 2:267–286, 1991b.

B. Schäfl, L. Gruber, J. Brandstetter, and S. Hochreiter. G-signatures: Global graph propagation with randomized signatures. *arXiv preprint arXiv:2302.08811*, 2023.

S. S. Shapiro and M. B. Wilk. An analysis of variance test for normality (complete samples). *Biometrika*, 52:591–611, 12 1965.

P. Smolensky. *Information Processing in Dynamical Systems: Foundations of Harmony Theory*, page 194–281. MIT Press, Cambridge, MA, USA, 1986.

E. Sontag. Realization theory of discrete-time nonlinear systems: Part i-the bounded case. *IEEE Transactions on Circuits and Systems*, 26:342–356, 5 1979a.

E. D. Sontag. *Polynomial Response Maps*, volume 13. Springer Verlag, 1979b.

S. Takahashi, Y. Chen, and K. Tanaka-Ishii. Modeling financial time-series with generative adversarial networks. *Physica A: Statistical Mechanics and its Applications*, 527:121261, 8 2019.

R. Tepelyan and A. Gopal. Generative machine learning for multivariate equity returns. *4th ACM International Conference on AI in Finance*, pages 159–166, 11 2023.

M. Wiese, L. Bai, B. Wood, and H. Buehler. Deep hedging: Learning to simulate equity option markets. *SSRN Electronic Journal*, 2019. ISSN 1556-5068.

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

M. Wiese, B. Wood, A. Pachoud, R. Korn, H. Buehler, M. Phillip, and L. Bai. Multi-asset spot and option market simulation. *SSRN Electronic Journal*, 2021.

T. Xu, L. Wenliang, M. Munn, and B. Acciaio. Cot-gan: Generating sequential data via causal optimal transport. *Proceedings of the 34th International Conference on Neural Information Processing Systems*, 2020.

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