Generating financial markets with signatures

While most generative models tend to rely on large amounts of training data, here Hans Buehler et al present a generative model that works reliably even in environments where the amount of available training data is small, irregularly paced or oscillatory. They show how a rough paths-based feature map encoded by the signature of the path outperforms returns-based market generation both numerically and from a theoretical point of view. Finally, they propose a suitable performance evaluation metric for financial time series and discuss some connections between their signature-based market generator and deep hedging

Generative modelling and market generators. The emergence of deep neural network-based financial modelling such as deep hedging, pricing, calibration and forecasting (Bayer *et al* 2019; Bühler *et al* 2019; Henry-Labordere 2019; Horvath *et al* 2021; Kondratyev & Schwarz 2019; Kondratyev *et al* 2020) has opened the door to a range of previously unseen applications. It is this development that has driven the interest of quants and researchers alike towards highly realistic market data simulators. A key factor for training these deep networks to a sufficient level of accuracy is the availability of training datasets that are both rich enough and representative. However, for many financial applications of machine learning, sufficient data is often not available (eg, when the focus is on daily returns data). This problem is amplified by the inherent non-stationarity of market data: data from even ten years ago exhibits very different statistical characteristics from data today. This gives rise to an increased interest in generating realistic, rich and robust market data.

Deep neural networks provide a powerful tool for approximating complex distributions, and this capacity, together with increases in available computational power and speed, has broadened horizons in all areas of modelling, including market simulation. The fundament of their prowess is due to the universal approximation properties of neural networks, which establish that any function or distribution with sufficient regularity can be approximated by a sufficiently large neural network. Generative models capture probability distributions by approximating these via neural networks from learned samples, from which new synthetic data samples can be drawn. The generative model is trained in such a way that its outputs resemble the underlying distribution of the given dataset with respect to some loss function, referred to as the performance evaluation metric. Generative modelling originates from more traditional applications of machine learning, and the adaptation of these techniques to a financial setting has its bespoke challenges. Identifying and addressing these challenges is one of the contributions of this work.

Definition 1 (Market generator) We refer to generative models in a financial time series context as market generators. A market generator is described by a pair $(\mathcal{A}, \mathcal{O})$ consisting of a network architecture $\mathcal{A}$ (a generative model) together with an objective function $\mathcal{O}$.

In other words, a market generator is a generative neural network (eg, a restricted Boltzmann machine (RBM), variational autoencoder (VAE) or generative adversarial network (GAN)) specified by a choice of architecture $\mathcal{A}$, together with a feature encoding of the data used in training, via the objective function $\mathcal{O}$. Since a market generator is a generative model specialised to financial time series, the feature map and objective function are designed to approximate the underlying distribution of a market from data samples (training and test data $\mathcal{D}$) given in the form of sample paths.

With that, a market generator is a parameterised family of functions $(\mathcal{A}, \mathcal{O})$ that map a sample set of time series $\mathcal{D}$ into a functional $\mathcal{T} := ((\mathcal{A}, \mathcal{O}); \mathcal{D})$. After training, the parameterised family $(\mathcal{A}, \mathcal{O})$ becomes a functional $\mathcal{T}$ that approximates a transformation of distributions $\mathcal{T}(\mu) = \nu$ from some initial source of randomness μ (say, $\mu \sim \mathcal{N}(0, 1)$ or $\mu \sim \mathcal{U}(0, 1)$ in the generative model) to the distribution ν, which approximates the empirical distribution of paths observed in the data $\mathcal{D}$. Using this functional $\mathcal{T}$, new sets of sequences of future market paths can be artificially generated (conditional on a current market state), which are indistinguishable[1] in distribution from the distribution ν of the original samples in $\mathcal{D}$.

In this article, we design an architecture $\mathcal{A}$ (conditional VAE) and objective function $\mathcal{O}$ (a feature map consisting of an L^2 error of the truncated log-signature of the time series), and suggest a solution to the question of how to evaluate the performance, ie, the 'quality' of the output, of market generators in a consistent and universal manner for financial time series (the maximum mean discrepancy (MMD) two-sample test of Chevyrev & Oberhauser (2018) on path space). This additional performance evaluation metric $\mathcal{P}$ would resemble the role of a one-step discriminator in the context of GANs.

Practical use and applications of market generators. There are several situations in which it is beneficial to rely on simulated data samples that are statistically indistinguishable from a given original dataset.

Applications in which generative simulation of financial markets can be a game changer include

- optimizing and robustifying hedging, eg, deep hedging, mean-variance hedging or portfolio optimization, and similar applications;
- computing quantities that can be computation-heavy such as risk metrics (eg, value-at-risk, conditional value-at-risk), stress testing, draw down, funding, liquidity and credit risk, such as XVA and similar applications; and
- creating reference data sets without licensing restrictions, eg, for education and academia, and for a standardized testing of model performance (see, for example, the 'SR11: Guidance on Model Risk Management' regulation in the United States), or recent approaches to crowdsourced model development.

[1] *With respect to some performance evaluation metric $\mathcal{P}$.*

Some approaches to numerical data simulation in finance: classical and new.

Classical modelling. Numerical simulation of financial time series has a long history in the related literature. Classical approaches include, for example, classical stochastic market models (eg, Heston, stochastic alpha beta rho), autoregressive models (Hamilton 1994) and variations of these. Among their advantages are their tractability and several decades' worth of experience in understanding their mathematical properties. Structural models also usually have interpretable parameters. Clearly, the advantage thereof is a more straightforward suitability to currently prevalent risk management frameworks. Disadvantages may include relative inflexibility, which can result in modelling inconsistencies.

Data-driven modern generative modelling. Approaches to generative modelling are based on the common principle of generating new synthetic data samples whose distribution resembles the distribution of some reference dataset. One of the most striking differences between modern generative modelling and classical generation of synthetic data is that explicit knowledge of the underlying data-generating distribution is no longer required. Therefore, instead of implementing (an approximation of) some known distribution or transition density, generative models often approximate the underlying distribution implicitly by drawing samples from the latter and comparing their similarity to the original dataset with respect to certain similarity metrics. This is particularly true for so-called differential generator networks, where a transformation map is learned through backpropagation from an initial source of randomness to a target distribution. The two most commonly used generative differential network-based approaches are VAEs, GANs and variations thereof.

Among the first available results on market generators were those of Kondratyev & Schwarz (2019), who applied an RBM for time series generation, controlling for the autocorrelation function and quantiles of the generated time series. Boltzmann machines were among the first generative models introduced to learn arbitrary distributions over binary vectors. Recent contributions to this stream of the literature have often used GANs as generative models (see, for example, Henry-Labordere 2019). To date, we are not aware of approaches using VAEs for this purpose.

GANs are the most popular differential generator networks, though they are typically data-hungry, and it is often difficult to guarantee their convergence and stability. Variational autoencoders maximize the likelihood of observing the given (original) data samples under the generated samples (3) and are particularly well adapted to settings where the available amount of training data is notoriously small.

Challenges of financial time series simulation: classical and new

Currently, many available neural network-based generative models originate from static applications. Therefore, several performance evaluation metrics for generative models have been developed to measure some forms of marginal distribution. Incorporating the time series aspect of the data poses additional challenges, as static performance evaluation metrics are not straightforward to generalise to the dynamic time series setting. At present, simulated financial time series data is commonly addressed by capturing specific universal features of the time series, commonly referred to as stylized facts. Below, we recall a number of stylised facts that traditional stochastic models typically reflect. Although, for classical stochastic models, these stylized facts are often formulated in terms of the (conditional) distribution of returns, this returns-based viewpoint often does not permit us to convey a sufficiently full picture for distributions of synthetic market paths that come from market simulators using generative modelling.

Problem formulation. Let us first fix some notation that will be used throughout the article. In the following, let $S(t)$ denote the price of a financial asset (stock, exchange rate or index) and let $X(t) = \ln S(t)$ denote the corresponding log price. Then, the log return (at scale t) is denoted as:

$$r(t, \Delta_t) := X(t + \Delta_t) - X(t) \tag{1}$$

where the time scale Δ_t has a scale ranging from a day to a month.[2] Note that in our experiments we transform the original calender time in the feature map that we use for training. Therefore, in the training procedure, 'time' refers not to calendar time, but to volume-adjusted business time, the natural scale for financial applications. The autocorrelation for a time lag $\tau > 0$ is denoted by:

$$\operatorname{corr}(r(t + \tau, \Delta_t) r(t, \Delta_t)) \tag{2}$$

A numerical generation with an increased emphasis on the accuracy of the data-generating process then aims to synthetically generate $M \in \mathbb{N}$ returns sequences of length k for a suitable $k \in \mathbb{N}$, conditional on the current state of the market:

$$(\widetilde{r_1}(t_1, \Delta_t), \dots, \widetilde{r_1}(t_k, \Delta_t)), \dots, (\widetilde{r_M}(t_1, \Delta_t), \dots, \widetilde{r_M}(t_k, \Delta_t)) \tag{3}$$

such that the generated set of k-sequences $(\widetilde{r_i}(t_1, \Delta_t), \dots, \widetilde{r_i}(t_k, \Delta_t))$ of returns reflects the properties of the observed k-sequences of returns:

$$(r_1(t_1, \Delta_t), \dots, r_1(t_k, \Delta_t)), \dots, (r_N(t_1, \Delta_t), \dots, r_N(t_k, \Delta_t)) \tag{4}$$

as accurately as possible, where $N \in \mathbb{N}$ denotes the number of observations in the original dataset.

Small data environments. Note that, typically, the number of generated samples M need not be identical to the number of original samples N. A small data environment would correspond to the situation where $M \gg N$, with N being relatively small from a standard deep learning perspective; for example, daily stock data or data from leading indexes (S&P 500, DAX, FTSE), where the number of data samples (the dataset covers approximately 5,000 days' worth of daily data) available for training is orders of magnitude smaller than the amount of data normally needed in most neural network applications. In such situations, the challenge is to efficiently extract the most relevant information from a small amount of available samples in a very simple generative network.

A reminder of specific stylised facts and evaluation metrics. The time series data of financial markets exhibits a set of stylised facts about financial markets that a realistic financial market model is commonly expected to reflect. These include, for example, the following.

Non-stationarity. That is, past returns do not necessarily behave like future returns; for any set of time instants $t_1, \dots, t_k$ and any time lag $\tau > 0$, the joint distribution of returns is the same as the joint distribution of the lagged returns:

$$(r(t_1, \Delta_t), \dots, r(t_k, \Delta_t)) \sim (r(t_1 + \tau, \Delta_t), \dots, r(t_k + \tau, \Delta_t)) \tag{5}$$

[2] *From a mathematical perspective, we could include shorter scales, such as a few seconds.*

Heavy tails and aggregational Gaussianity. Asset returns have (power-law-like) heavier tails than in a normal distribution as well as a distribution that is more peaked than a normal distribution.

Absence of autocorrelations of asset returns, but slow decay of autocorrelation in absolute returns. Asset returns are uncorrelated (except for very short intraday timescales) but not independent. The autocorrelation (2) function of absolute returns $|r(t, \Delta_t)|$ decays slowly as a function of the time lag τ following a power-law.

Volatility clustering and multifractal structure. Phases of high/low activity tend to be followed by phases of high/low activity (see also Gatheral *et al* 2018).

Leverage effect. Asset returns exhibit a leverage effect, ie, there is a negative correlation between the volatility of asset returns and the returns process.

■ **Performance evaluation metrics.** The stylised facts shown above are independent of the ultimate application of our market generator. To date, many performance evaluation metrics of time series data are traditionally modelled by an ad hoc selection of stylised facts. Due to the lack of an established consensus on similarity metrics for sample paths, these metrics are not only used for structural models but also for generative models.

The most commonly used evaluation scores include the following.

Distributional metrics: the difference between the historical sample cumulative distribution and the cumulative distribution function of the generated samples is measured with respect to a suitable metric $\mathcal{D}$:

$$\mathcal{D}_{b \in B} |F_{1,n}(b) - F_{2,m}(b)| \tag{6}$$

where n, m denote the number of original and generated samples, respectively, and B is (a suitable discretisation of) the sample space.

Tail behaviour scores: properties of the underlying distribution that control the tail behaviour are targeted. These are higher-order moments such as skewness and kurtosis:

$$\left.\begin{aligned} &\frac{1}{N_x}\sum_{j=1}^{N_x} |\mathrm{skew}(x_j) - \mathrm{skew}([\hat{x}_j^{(1)}, \dots, \hat{x}_j^{(m)}])| \\ &\frac{1}{N_x}\sum_{j=1}^{N_x} |\mathrm{kurt}([x_j) - \mathrm{kurt}(\hat{x}_j^{(1)}, \dots, x_j^{(m)}])| \end{aligned}\right\} \tag{7}$$

Correlation- and cross-dependence scores: to detect serial autocorrelation in the cross-correlation scores of time series, and analogously for multi-dimensional time series.

To date, many performance evaluation metrics of time series data are traditionally modelled by an ad hoc selection of such stylised facts. The above examples already suggest a wealth of possible metrics and evaluation scores to measure the quality of generated market paths, and the list does not end here, depending on the application one has in mind.

When it comes to applications of market generators (eg, optimising hedging performance such as deep hedging (Bühler *et al* 2019)), we additionally aim to ensure that our generator performs well out-of-sample with respect to the performance metric applied in the optimisation; for example, a generator is deemed better if the risk-adjusted profit-and-loss metric performs better on the test set. Thus, with deep hedging in mind as an application for the generative model, we may want to include hedging objectives in the optimisation either directly or indirectly. For the latter, we refer to Antonov *et al* (2020) for some possibilities and note that our suggested performance evaluation metric (11) links back to hedging strategies by construction.

■ **Challenges of similarity metrics for financial time series with generative models.** As seen above, traditional distributional metrics and divergences provide ample means of measuring distances between distributions. However, determining appropriate similarity metrics on the level of a stochastic process – or its data representation through a time series – is a challenge of a different nature. This section is devoted to questions related to this problem: what are the challenges in determining the appropriate similarity metrics of distributions on path space, or whether two sets of sample paths originate from the same underlying distribution? The static case (similarity of marginal distributions) is not straightforward to generalise to the dynamic one (similarity metrics on path space). Identifying a two-sample test to measure the 'goodness' of generated sample paths of financial time series is a related challenge. We list a number of the challenges that are addressed in this article.

(1) The potential non-universality (ad hoc selection) of features to control for in the generated time series. If the chosen set of optimisation objectives is bespoke to one application, the generated time series may not carry over easily to other applications.

(2) The underlying distribution of the data-generating model is often not known explicitly in generative models and only controlled implicitly through the generated data samples and their similarity to the original data. Therefore, two-sample tests are needed as performance evaluation metrics rather than distributional metrics and divergences.[3]

(3) Established distributional metrics (and two-sample tests) for marginal distributions can be generalised to (finite) multivariate marginals. However, generalising these metrics to path space is not straightforward, one of the difficulties being that the path space $C^0([0,1], \mathbb{R}^d)$ is infinite dimensional and non-locally compact (see Chevyrev & Oberhauser 2018).

(4) Non-continuous observation of the original data. Usually only discrete time observations of sample paths are available. For generative models, this renders itself more problematic than for classical models: from a hedging perspective, it may be insufficient to match marginal distributions on a discrete set of observation dates only (see Brigo 2019).[4] Also, the number n and position of observation time points might change from sample to sample, which makes learning difficult. Finally, in some applications (eg, high-frequency data), n can get very large.

(5) Non-stationarity of the target distribution. Financial time series are typically non-stationary, and underlying distributions change with market conditions. For generative models, this has two implications in particular. It means a conditional version of the market generator is needed, which allows us to produce samples that are conditioned on specific market states; however, a conditional generative model may amplify the data scarcity issue as the availability of a sufficient number of representative training samples becomes coupled with market conditions.

■ **On signatures, their advantages in encoding path-valued data and their meaning.** The challenges faced as part of the generative modelling of financial data streams – as raised at the beginning of this section – are

[3] *The latter may only be applicable after inferring from the available data sample to the underlying distribution.*

[4] *Brigo (2019) displays examples that are indistinguishable from one another on a finite set of marginals in the physical measure but lead to arbitrarily different hedging strategies and option prices.*

not limited to the choice of performance evaluation metrics; they extend to the choice of objective function and feature selection for training. An efficient parsimonious encoding of financial data streams facilitates the fast and efficient training of generative models. Furthermore, an encoding that anticipates typical properties of the data and irregularities of sampling makes training more robust with respect to data quality.

The theory of rough paths provides a theoretical framework for these aims with a feature encoding via signatures (Friz & Hairer 2014; Lyons 2014).

Definition 2 (Signature of a path) Let:

$$X : [0, T] \to \mathbb{R}^d$$

be a continuous path of bounded variation. Then, the signature of X is defined by the sequence of iterated integrals[5] given by:

$$\mathbb{X}_T^{<\infty} := (1, \mathbb{X}_t^1, \dots, \mathbb{X}_T^n, \dots) \tag{8}$$

where:

$$\mathbb{X}_T^n := \int_{0<u_1<\cdots<u_k<T} \mathrm{d}X_{u_1} \otimes \cdots \otimes \mathrm{d}X_{u_k} \in (\mathbb{R}^d)^{\otimes n} \tag{9}$$

where $\otimes$ denotes the tensor product. Similarly, given $N \in \mathbb{N}$, the truncated signature of order N is defined by:

$$\mathbb{X}_T^{\leq N} := (1, \mathbb{X}_T^1, \dots, \mathbb{X}_T^N) \tag{10}$$

Signatures provide a means of encoding financial data streams parsimoniously and efficiently, and of addressing the challenges of path-valued data.

As mentioned earlier in this section, a particular challenge in the context of synthetic generation of market paths is that the distribution in question is defined on the infinite-dimensional space of paths $C^0([0,1], \mathbb{R}^d)$, while the available generative modelling tools are finite dimensional. Thus, the infinite-dimensionality of path space is not only a challenge for the choice of suitable similarity metrics or performance evaluation metrics but also for the feature extraction of data and training. A solution to this issue is to project this infinite-dimensional space to a suitable finite-dimensional space where standard methods for generative models may be used.

However, mapping this inherently infinite-dimensional space in an optimal way to a finite-dimensional space presents a challenge, and the choice of projection is not trivial.

The most straightforward choice would be to sample the path on a fixed, discrete time grid, return by return, as in (1), for example (while accounting for the joint distribution of these), and to learn the projected probability measure using standard generative models. However, this approach would not fully capture the sequential nature of financial data and would fail to effectively capture the probability measure on the original path space. Moreover, if we project this infinite-dimensional object down to a finite-dimensional space by sampling on a discrete time grid, the projection is not a 'natural' one, as the original distribution on an intrinsically infinite-dimensional space captures much richer information about the process.

It is more effective to use the signature or log-signature to project an infinite-dimensional encoding (8) of the path space to a finite N-dimensional one, as in (10). The signature of a path is a transformation of the original continuous path into a sequence of statistics, an infinite-dimensional vector (8) of signature entries (9). These statistics fully characterise the original path up to time parameterisation and, furthermore, they already offer a faithful and parsimonious description of it in its first few entries (dimensions) of the signature vector (10). The error made by the truncation at level N decays with factorial speed as $\mathcal{O}(1/N!)$.

Moreover, the first several signature entries, ie, the first terms in the vectors (9) (respectively, (10)) have financial interpretations: the first term captures drift, ie, the increment of a price path over a period of time, and the second term indicates the volatility over a period of time (through the Lèvy area). Higher-order terms capture the finer aspects of the path, which end up fully characterising the latter. This is reminiscent of PCA, where the properties of the path are ordered from the 'most relevant properties' up to the 'finer properties'.

Higher-order terms capture finer aspects of the path that end up fully characterising the latter. An ordering, which is reminiscent of principal components from the most relevant towards finer properties of the path.

On returns-based versus signature-based data generation in a pricing and hedging context. A wealth of available theoretical results imply the signature transform is a highly efficient way of encoding the most relevant information contained in a stochastic path (Friz & Hairer 2014; Lyons 2014). We demonstrate this efficiency in our numerical results by showing that signatures-based projection guarantees a faster learning rate than training generative models in a 'traditional' returns-based manner. Learning the projection on the truncated log-signature of a set of paths converges with fewer training samples than learning the multidimensional projection of the process on a discrete time grid. Learning via signatures not only is numerically more efficient than doing so via the finite-dimensional projection on a discrete time grid, but also encodes a richer and more relevant wealth of information about the path (which also allows us to control for hedging strategies); in the latter projection, some essential information may be lost, which can have significant consequences for hedging strategies. Results in Brigo (2019) indicate that when we are sampling returns distributions of a stochastic process on a discrete time grid, even statistically indistinguishable sets of paths in the historical measure can lead to arbitrarily different option prices. If, however, the paths are sampled on the level of signatures, this ambiguity of option prices does not occur. Therefore, the signature is not only a more efficient way of learning sample paths, but also one that removes the ambiguity of the corresponding option prices and provides meaningful control over hedging performance.

Further advantages of signatures. Further advantages of working with signatures for modelling functions of data streams have been discussed and presented in Lyons (2014). These advantages include (but are not limited to) the following properties.

(1) The expected signature of a stochastic process *determines the law of the process uniquely*. With that, the expected signature plays a similar role on path space to the moment-generating function for distributions (this provides a basis for the performance evaluation metric developed in Chevyrev & Oberhauser (2018) for path space).

(2) *They permit model-free, data-driven modelling.* The framework does not impose any assumptions on the underlying stochastic dynamics. Signatures provide a flexible basis of functions for a functional on path space.

[5] *If the path X has bounded variation – which is the case for discrete data – the integrals above can be defined using Riemann-Stieltjes integrals.*

(3) *The signature transform is straightforward to implement and is an efficient encoding of data streams.* Today, there are readily available (and constantly improving) powerful Python packages and libraries[6] to transform data streams to signatures, and algorithms to transform signatures back to paths of data streams.
(4) The framework is *invariant under translation and time-parameterisation*. Therefore, in order to encode price paths in business time rather than calendar time, we apply the lead-lag transformation.
(5) Furthermore, signatures are *robust to irregular sampling* (which becomes relevant for tick data), *missing data and highly oscillatory data as well.* In particular, they provide a consistent framework for unbounded variation paths, which may arise from Donsker-type theorems in the high-frequency limit. The functions of such paths have to be treated with care, as, for example, the quadratic variation. Signatures appear naturally when describing the behaviour of functions of non-smooth paths (cf, Chevyrev & Oberhauser 2018). For more available Python packages and libraries for signatures, see Reizenstein & Graham (2018).

Distances for time series and sample paths: a computationally efficient MMD metric for laws of stochastic processes. When it comes to assessing the quality of a set of generated paths, being able to compute the distance between the laws of two stochastic processes becomes imperative. A naive metric based on the marginals of the two processes is bound to fail, as two stochastic processes can have identical marginals but very different laws (Brigo 2019). Instead, a metric that considers the entire law of the stochastic process is needed. In Chevyrev & Oberhauser (2018), a two-sample test for laws of stochastic processes is developed, based on signature kernels. We use this test to evaluate the quality of our market generator. To assess whether a generative model is able to generate paths that are realistic with respect to a sample of real paths $Y_1, \dots, Y_n$, from the generative model $n \in \mathbb{N}$, we sample paths $X_1, \dots, X_n$ and apply the two-sample test proposed in Chevyrev & Oberhauser (2018). More specifically, we compute the signature-based MMD test statistic $T(X_1, \dots, X_n; Y_1, \dots, Y_n)$:

$$\begin{aligned} T(X_1, \dots, X_n; Y_1, \dots, Y_n) \\ := \frac{1}{n(n-1)} \sum_{i,j;i \neq j} k(X_i, X_j) - \frac{2}{n^2} \sum_{i,j} k(X_i, Y_j) \\ + \frac{1}{n(n-1)} \sum_{i,j;i \neq j} k(Y_i, Y_j) \end{aligned} \tag{11}$$

where $k(\cdot, \cdot)$ is the so-called signature kernel (see Chevyrev & Oberhauser 2018, proposition 4.2). Then, given a fixed confidence level $\alpha \in (0,1)$, we compute the threshold:

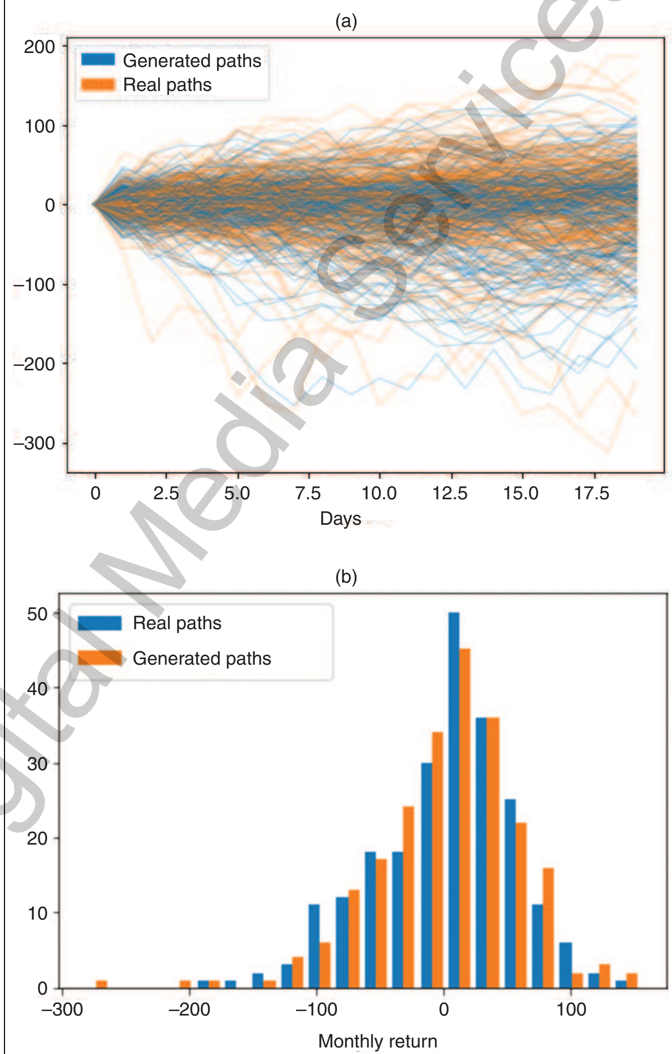
$$c_\alpha := 4\sqrt{-n^{-1} \log \alpha}$$

The generative model is said to be realistic with a confidence level α if $T_U^2 < c_\alpha$.

Numerical results

Numerical experiments with historical data of S&P. To demonstrate the accuracy of the generated log-signatures and images of the produced paths, in this section, we provide images of two-dimensional projections

1 The signature-based VAE



This is the image of generated paths inverted from log-signatures and the corresponding returns distribution at the end of the time horizon (one month)

and the resulting generated paths and corresponding returns for an optical demonstration. The rigorous numerical demonstration of the accuracy of the paths via the MMD signature two-sample test can be found in the section below. The output of the signature-based generative model is then translated into (standard) returns and the resulting marginal distributions (see the two parts of figure 1). For more details about the architecture and modelling choices we made in our numerical experiments, see the open-source Github repository https://github.com/imanolperez/market_simulator that we have prepared for the interested reader, where we demonstrate the numerical findings of this work.

Performance evaluation scores. We conduct the signature-based MMD two-sample test (11) described in the previous section for the scenarios described above. In table A, we include the confidence level at which the signature-based MMD test changes from the result 'the two samples come from the same distribution' to the result 'the two samples come from different distributions'. Clearly, if the test can be passed at a higher confidence

[6] *Such as the* `esig`, `tosig` *and* `iisignature` *libraries; see https://github.com/bottler/iisignature as well as Kidger & Lyons (2020) for more background.*

A. The highest confidence level at which synthetic samples pass the test	
	MMD signature confidence level
Weekly signature paths generated	99.998%
Monthly signature paths generated	99.987%

level, this indicates a higher level of similarity between the generated samples and the original ones.

In table A, we display the results for the weekly and monthly paths of the VAE.

We could carry on here, as in related works, to include further statistics that control for properties of stylised facts as performance evaluation metrics. However, we recall that the signature-based MMD test of Chevyrev & Oberhauser (2018) completely characterises the law of a stochastic process and therefore omit further test measures here for brevity.

Conclusions

Our experiments show that while the returns-based (classical) setup for the VAE optimisation works significantly better if more training data is provided (tested with numerically generated training data), the signature-based VAE already converges with the low number of training samples available on the historical path of the S&P. This confirms the signature-based learning already converges with the small training set provided by the S&P data. ■

Hans Buehler is global head of equities analytics, automation, and optimization at JPMorgan Chase & Co, London; Blanka Horvath is a lecturer at King's College London, an assistant professor at the Technical University of Munich and a researcher at the Alan Turing Institute; Terry Lyons is a professor at the Mathematical Institute, University of Oxford, and at the Alan Turing Institute; Imanol Perez is a quantitative researcher at Jump Trading LLC in London; and Ben Wood is global co-head of equity derivatives quantitative research at JPMorgan Chase & Co, London. The opinions expressed in this article are those of the authors, and do not necessarily reflect the views of JP Morgan. The algorithm developed in this paper is available at the Github repository provided in this article: https://github.com/imanolperez/market_simulator.
Email: hans.buehler@jpmorgan.com, blanka.horvath@kcl.ac.uk,
terry.lyons@maths.ox.ac.uk, imanol.perez@maths.ox.ac.uk,
ben.wood@jpmorgan.com.

REFERENCES

Antonov A, JF Baldeaux and R Sesodia, 2020
Quantifying model performance
Risk June, www.risk.net/7549216

Bayer C, B Horvath, A Muguruza, B Stemper and M Tomas, 2019
On deep pricing and calibration of (rough) stochastic volatility models
arXiv preprint, available at https://arxiv.org/abs/1908.08806

Brigo D, 2019
Probability-free models in option pricing: statistically indistinguishable dynamics and historical vs. implied volatility
Options: 45 Years After the Publication of the Black-Scholes-Merton Model
Conference Paper

Bühler H, L Gonon, J Teichmann and B Wood, 2019
Deep hedging
Quantitative Finance 19(8), pages 1271–1291

Chevyrev I and H Oberhauser, 2018
Signature moments to characterize laws of stochastic processes
arXiv preprint, available at https://arxiv.org/abs/1810.10971

Friz P and M Hairer, 2014
A Course on Rough Paths: With an Introduction to Regularity Structures
Springer International Publishing

Gatheral J, T Jaisson and M Rosenbaum, 2018
Volatility is rough
Quantitative Finance 18(6), pages 933–949

Hamilton JD, 1994
Time Series Analysis
Princeton University Press

Henry-Labordere P, 2019
Generative models for financial data
SSRN preprint, available at https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3408007

Horvath B, A Muguruza and M Tomas, 2021
Deep learning volatility: a deep neural network perspective on pricing and calibration in (rough) volatility models
Quantitative Finance 21(1), pages 11–27

Kidger P and T Lyons, 2020
Signatory: differentiable computations of the signature and log-signature transforms, on both CPU and GPU
arXiv preprint, available at https://arxiv.org/abs/2001.00706

Kondratyev A and C Schwarz, 2019
The market generator
Risk February, www.risk.net/7401191

Kondratyev A, C Schwarz and B Horvath, 2020
The data anonymiser
Risk September, www.risk.net/7669111

Lyons T, 2014
Rough paths, signatures and the modelling of functions on streams
Preprint, available at https://arxiv.org/abs/1405.4537

Reizenstein J and B Graham, 2018
The iisignature library: efficient calculation of iterated-integral signatures and log signatures
arXiv preprint, available at https://arxiv.org/abs/1802.08252