Generating drawdown-realistic markets using path signatures

Emiel Lemahieu[123], Kris Boudt[1], Maarten Wyns[2]

Ghent University, InvestSuite

Abstract

The simulation of markets with drawdowns that are faithful to empirical market data is a challenging task as this path dependent measure depends both on the moments of the underlying price process as well as its dynamics such as the occurence of consecutive losses. In applications, such as pricing drawdown insurance options or developing portfolio drawdown control or optimization strategies, it is paramount that one reproduces this measure correctly in synthetic scenarios. We advocate an essentially non-parametric Monte Carlo approach from machine learning, combining a variational autoencoder generative model with a path signature-based drawdown reconstruction loss function. Machine learning requires a system of differentiable equations that provides numerical simulations by iteration or learning. To overcome issues of numerical complexity and non-differentiability, we approximate drawdown as a linear function of its dynamic or path moments, known in the literature as path signatures. For (fractional) Brownian and empirical data, we obtain excellent approximations using simple linear regression. We then appreciate that drawdown as a linear combination of path moments gives a mathematically non-trivial smoothing (a non-commutative exponential alternative for Taylor series) which gives one leeway to simulate drawdown-realistic markets by including a drawdown evaluation metric in the learning objective. We conclude with a number of numerical experiments on mixed equity, bond, real-estate and commodity portfolios and obtain a host of realistic drawdown scenarios.

Keywords: Simulation, Non-Parametric Monte-Carlo, Path functions, Signatures, Drawdowns
MSC: 91B28, 91B84

1. Introduction

Market generators are generative machine learning (ML) models with the specificity of modeling financial markets such as stock price time series, order books, implied volatility surfaces, and more. They have gained popularity in recent years (Wiese et al. [1], Koshiyama et al. [2], Cont et al. [3], Bergeron et al. [4]). Arguably the main advantage of generative ML over existing simpler solutions such as standard Monte Carlo engines is being able to simulate realistic financial time series without having to prespecify the dynamics, which means handcrafting a data generating process (DGP) a priori. Realistic means low distributional divergences between the real and generated return distributions (such as difference in normalized moments or max mean discrepancy (MDD), entropy or Kullback-Leibler divergence, optimal transport measures like Wasserstein distance, and so and so forth), and distances between their autocorrelation functions and tail indices. Some striking results in the latter regard have been found in Wiese et al. [1], Cont et al. [3], Buehler et al. [5] and Vuletić et al. [6].

A drawdown is a price fall relative to the historical maximum. In this article, we argue that one can use a market generator as a non-parametric Monte Carlo engine to solve the difficult problem of generating realistic drawdowns. The latter is prohibitive from a parametric DGP perspective as, to the best of our knowledge, no literature has explicitly modeled drawdown path dependencies as part of the DGP. The literature furthermore indicates that a general link between the underlying data generating process and the drawdown is unknown and depends on the specific process. These works have commonly relied on Brownian assumptions as to use Lévy's theorem to obtain the analytical drawdown distribution. Especially in applications where this measure is crucial (such as pricing drawdown options for max loss insurance (Carr et al. [7],Atteson and Carr [8]) or controlling for drawdown in a portfolio optimization context (Chekhlov et al. [9], [10])) one would need theoretical guarantees that upon convergence of the model drawdowns are explicitly preserved.

The main contribution of this paper is overcoming the issues of non-differentiability and numerical complexity inherent to the drawdown function by approximating it with a linear regression on the path's signature terms. Signatures serve as the moment generating function on the path space and are because of their universality (Chevyrev and Oberhauser [11]) a likely candidate for universal approximation of drawdown with a linear function. We argue that by adding weighted path moments to the reconstruction loss of the market generator, one can obtain realistic drawdown scenarios, solving the understatement of adverse drawdown scenarios of traditional generative model architectures or common DGP assumptions.

The article is structured as follows. Section 2 covers the background on market generators and positions our problem setting within this literature. Section 3 discusses drawdowns, notations and its Lipschitz continuity following from its reformulation as a non-linear dynamic system. Section 4 summarizes some key elements from rough path theory and the approximation of functions on paths. We propose a linear approximation of drawdowns in the signature space. Section 5 discusses the market generator model and its implementation. Section 6 covers our numerical study with experiments on (fractional) Brownian and empirical data. Section 7 concludes.

2. Market Generators

This section briefly covers the background on market generators, related papers and their commonalities that lead to the questions posed in this paper.

The potential benefits of generative machine learning for modeling financial time series have been early recognized by Kondratyev and Schwarz [12] and Henry-Labordere [13], and Wiese et al. [1], who first applied restricted Boltzmann machines (Smolensky [14]) and GANs (Goodfellow et al. [15]) respectively to financial sequences data. Since those papers, a host[4] of use cases have been proposed that include time series forecasting (Wiese et al. [1]), trading strategy backtesting (Koshiyama et al. [2]), hands-off volatility surface modeling (Cont and Vuletić [37]), option pricing and *deep hedging* (Buehler et al. [5], [23]) and more.

Closest related to this work is the paper of Buehler et al. [5] who also use the combination of variational autoencoder and signatures, with the general aim of reproducing some of the stylized facts of financial time series (Cont [38]). However, they use signatures in the input space and then output signature values, which implies that deploying their model requires a scalable inversion of signatures back to paths, which is far from trivial. Similar to Ni et al. [17] who use signatures in the discriminator of a GAN architecture, the generator in this paper does not output signatures but only uses them in the loss evaluation step. Moreover, much work on market generators has revolved around adjusting the loss function such that desired features of the timeseries are reproduced. This is very close to our proposed approach. Cont et al. [3] evaluates the tail risk (value-at-risk and expected shortfall) of the simulations to evaluate their adequacy. Recently, Vuletić et al. [6] included profit-and-loss (P&L) similarity and Sharpe ratios in the loss function to increase the financial realism of the generated scenarios. This paper builds on top of that. Drawdown is a similar metric that in most financial contexts would be a useful feature to reproduce, because it captures both P&L and autocorrelation, but in fact is in some financial contexts the most important metric, such as portfolio drawdown control, optimization, or drawdown insurance. Think about asset managers who want to backtest their strategies which get closed down if their strategy breaches a certain threshold. However, as a path functional, rather than a function on a P&L distribution like a tail loss or Sharpe ratio, drawdown is much more difficult to evaluate. That is why we introduce the approximation trick in the next sections.

3. Drawdowns

This section introduces the concept of a drawdown and its notation used throughout the paper. We briefly touch upon how it is usually approached in the literature, the issues for data-driven simulation of drawdowns and the expression of drawdown as a non-linear dynamic system. The latter insight, combined with the concept of a path signature, will allow us to introduce the approximation in the next sections.

3.1. Introduction and notation

The drawdown is the difference at a point in time t between a price S_t and its historical maximum up to

[4] A non-exhaustive list of important contributions is included in the bibliography: Cuchiero et al. [16] , Ni et al. [17], Li et al. [18] , Storchan et al. [19] , Benedetti [20] , Xu et al. [21], Pardo and López [22], Buehler et al. [23] , Ni et al. [24], Pfenninger et al. [25], Rosolia and Osterrieder [26] , Koshiyama et al. [2] , van Rhijn et al. [27], Marti et al. [28], Coyle et al. [29], Wiese et al. [30] , Lezmi et al. [31] , Wang [32], Buehler et al. [5] , Cont et al. [3], Fung [33] , Frandsen [34], Bergeron et al. [35], Ning et al. [36] and Vuletić et al. [6]

that point. For a price path (in levels) $S : [0, T] \to \mathbb{R}$ define drawdown function Ξ as:

$$\xi_t = \Xi(S)_t = \max(\max_{k<t}(S_k) - S_t, 0) \tag{1}$$

It is clear that the drawdown is a non-trivial function of the underlying DGP and captures at least three dimensions. Firstly, the drift of S or essentially the deterministic tendency of achieving a new max. Secondly, the dispersion in S or its volatility, albeit in a negative deviation sense, which is in essence the probability of a loss vis-a-vis this monotonic drift. Thirdly, and often overlooked, the autocorrelation of losses or the probability that the nonzero drawdown will persist, i.e. the duration of the drawdown. The link between a DGP that captures these three components and its drawdown distribution is unknown in a general case and depends on the specific process. In the drawdown literature, one often assumes (geometric) Brownian motion and uses Lévy's theorem (Lévy [39]) to find the distribution of drawdown:

$$(M_t - B_t, M_t) \approx (B_t, L(B_t)) \tag{2}$$

where B_t is a Brownian motion at t, M_t is its running maximum and $L(B)_t$ is the so-called local time of B. This yields closed-form expressions of the distribution $\mathcal{P}(\xi)$ when S is a Brownian motion (e.g. see Douady et al. [40] and more concisely Rej et al. [41]). However, martingale processes like Brownian motion do not exhibit return autocorrelation and one expects a sequence of consecutive losses to result in larger drawdown scenarios, such that martingales can merely be optimistic lower bounds on worst case drawdown scenarios. This was emphasized in Rej et al. [41].

One way to cope with this issue is to simply not embed these drawdown parameters in the DGP or derive its distribution from said DGP assumptions, but to *rely on a fundamentally non-parametric simulation approach*. We show that one can rather rely on a market generator to abstract both the DGP and the link between DGP and drawdown distribution, and learn to reproduce this distribution in a Monte Carlo. This Monte Carlo could be used to devise strategies that control drawdown (such as drawdown optimization) or as a simulation engine for pricing drawdown insurance in a fully non-parametric way.

3.2. Issues with the drawdown measure for data-driven simulation

The problem with the drawdown measure, compared to simpler measures on the moments of the P&L, is that at first glance it looks unsuited for learning. The reasons are twofold.

- **Differentiability**: as we will discuss below, Ξ is non-differentiable w.r.t. a parametrized path S_θ. At first glance it looks impossible to change the DGP parameters θ by including 'feedback' on $\frac{\delta(\Xi(S_\theta))}{\delta(\theta)}$, without making simple assumptions on the specification of θ.
- **Complexity**: evaluating the maximum of a vector of n numbers takes n operations or $O(n)$. Naively evaluating the running maximum takes $n(n+1)/2$ or $O(n^2)$ operations. If one would use a smoothed approximation of the maximum (like smooth normalized exponentials [42]) to resolve non-differentiability, this would be computationally prohibitive (especially for long n paths). Moreover, accuracy of such naive exponential smoothing would rely on the scale of S and the variation around the local maxima.

The main contribution of this paper is that these issues of non-differentiability and numerical complexity can be jointly overcome by approximating the drawdown of a path by using linear regression on its signature terms. This idea will be developed in Section 4.

3.3. Drawdown as a non-linear dynamic system and Lipschitz continuity

Assume the price path $S : [0, T] \to \mathbb{R}$ is a piecewise linear continuous path of bounded variation (e.g. finite variance), such as daily stock prices or NAV values[5]. Denote by $\mathcal{V}([0, T], \mathbb{R})$ the (compact) space of all such paths. Firstly, we need to remark that while from (1) we know the max operator makes Ξ not continuously differentiable (also see Eq. 6 below), the variation in Ξ is bounded by the variation in the paths. When taking two bounded paths the distance between their maximum values is bounded by a norm on the distance between their path values:

$$|\max_{0<i<T}(S^1_i) - \max_{0<j<T}(S^2_j)| \leq \max_{i,j\in[0,T]} |S^1_i - S^2_j| = \|S^1 - S^2\|_\infty \tag{3}$$

Or in words, the distance between two maxima is capped by the maximum distance between any two values on path S^1 and S^2, which can be written as a norm. More specifically, the maximum is Lipschitz-γ continous, with distance inf-norm and $\gamma = 1$. Similar arguments can be made for ξ, i.e. apply max and a linear combination of two piecewise linear paths is Lipschitz continuous. In particular, Proposition 3.1 proofs

[5]For the definition of boundedness in the *p-variation* sense see [43].

how much the drawdown function can change when the underlying paths change in terms of a defined distance metric.

Proposition 3.1 (***Lip*-regularity of Ξ).** *The difference in drawdown between two paths is bounded by the distance between the paths. For two paths S_1, S_2 in the space of continuous (e.g. piecewise linear) paths of bounded variation $\mathcal{V}([0,T],\mathbb{R})$ and drawdown function $\Xi(S)_t = \max_{k\leq t}(S_k) - S_t$, we have the following Lipschitz regularity:*

$$|\Xi(S^1)_t - \Xi(S^2)_t| \leq C|S^1 - S^2| \tag{4}$$

Proof.

$$\begin{aligned}|\Xi(S^1)_t - \Xi(S^2)_t| &= |\max_{i\leq t}(S^1_i) - S^1_t - \max_{j\leq t}(S^2_j) + S^2_t| \\ &\leq \max_{i,j\in[0,t]} |S^1_i - S^2_j| + |S^2_t - S^1_t| \leq C|S^1 - S^2|\end{aligned} \tag{5}$$

through triangle inequality, the Lip condition holds for C at most 2 and distance metric inf norm $\|.\|_\infty$, which concludes the proof. □

In this article, we explore what this generalized continuity or regularity assumption implies for local approximation. Below, we observe the differentials of ξ and S, i.e. treat drawdown as a non-linear dynamic system, which is unnatural as non-differentiability implies $\frac{d\xi_t}{dS_t}$ is not a continuous function as one would expect for Taylor series-like expansions. Consider the following dynamic system (also see [40]):

$$d\xi_t = f(\xi_t, dS_t) = \begin{cases} -dS_t, \xi_t > 0 \\ \max(0, -dS_t), \xi_t = 0 \end{cases} \tag{6}$$

Depending on the current level of drawdown the effect of a price change is either linear or none. This is what creates the general non-differentiability of Ξ. One realizes that because of the conditionality, direct path integration does not make sense. However, the differentials $|d\xi_t| \leq |dS_t|$ are bounded. This makes local approximation still feasible. It does make sense to do an interpolation between the linear and zero effects, and because of this boundedness one would not be arbitrarily off. For instance, one could assume linear dynamics but with a stochastic component derived from the average time in drawdown, i.e. the probability of a linear effect. However, a practical solution of this stochastic equation would still be based on the full path (average time in drawdown) and thus be path dependent. We would thus rather express ξ as some bounded function of the mathematical object S (or intervals of S) rather than S_t or increments dS_t. This idea is developed in the next section.

4. Rough path theory and the approximation of functions on paths

This section briefly recapitulates some of the central ideas of rough path theory, path signatures and the approximation of functions on paths. Next, we discuss the signature approximation of drawdown where we consider the drawdown of a path as an approximate linear weighing of its path moments. This will offer the foundation for generating weighted path moments in the market generator of Section 5.

4.1. Path signatures

Rough path theory was developed by mathematician Terry J. Lyons (Lyons et al. [44], [43], [45]) and concerns solving rough differential equations. Consider the controlled differential equation:

$$dY_t = g(Y_t)dX_t \tag{7}$$

where X is a path on $[0,T] \to \mathbb{R}$, called the driving signal of the dynamic system. g is a $\mathbb{R} \to \mathbb{R}$ mapping called the physics that models the effect of dX_t on the response dY_t. A controlled differential equation (CDE) distinguishes itself from an ordinary differential equation (ODE) in the sense that the system is controlled or driven by a path (dX) rather than time (dt) or a random variable (stochastic SDEs, $d\epsilon$).

Rough path theory considers solution maps for driving signals that are much rougher than e.g. a traditional Brownian X. As shown in Appendix 7, through Picard iteration, one easily arrives at expressions such as:

$$\hat{Y}(N)_t = y_0 + \sum_{n=1}^{N} g^{\circ n}(y_0) \int\limits_{u_1<\ldots<u_n, u_1,\ldots,u_n\in[0,T]} \ldots \int dX_{u_1} \otimes \ldots \otimes dX_{u_n} \tag{8}$$

which correspond to N-step Taylor approximation $\hat{Y}(N)_t$ on the path space for the Y_t in Eq (7).

The n-fold iterated integral on the right hand side of Eq (8) is called the n-th order *signature* of the path X, or $\Phi_n(X)$. Signatures have some remarkable properties, most notably universality and uniqueness, which we will briefly explain below.

Clearly, $\hat{Y}(N)_t$ is linear in the truncated signature of X up to order N. Moreover, the error bounds of $\hat{Y}(N)_t$ to approximate Y_t display a factorial decay in terms of N:

$$|Y_t - \hat{Y}(N)_t| \leq C_\gamma \frac{|X|^{N+1}_{1,[0,t]}}{N!} \tag{9}$$

where the rate decay depends on the roughness of the signal γ [46]. Eq. (8) aptly summarizes the signature's

universality, i.e. a non-linear Y on X can be well approximated by a linear L on $\Phi(X)$.

For instance, if one replaces the path X by scalar x and thus non-commutative tensor products by simple products, the n-times repeated integral on dx is $\frac{x^n}{n!}$ which is the classical exponential series we know from a standard Taylor expansion. Signatures are thus the monomials of the path space (Lyons [43]), or put simply, the generalization of local approximation to functions of paths by offering a strictly non-commutative exponential function.

Chevyrev and Oberhauser [11] introduce signatures as the moment generating function on the path space, as they play a similar role to normalized moments but for path-valued data instead of scalar-valued data. Paths with identical signatures are identical in the distributional sense as are scalars with the same sequence of normalized moments. This is the uniqueness property.

Proof of signature universality[6] is due to Lemercier et al. [48] (also see Lyons and McLeod [45] for a more recent redefinition of the theorem (Theorem 3.4)) who prove that for a (Lipschitz) continous h, path X, there exists a vector of linear weights L such that for any small ϵ:

$$|h(X) - \langle L, \Phi(X)\rangle| \leq \epsilon \tag{10}$$

where $\langle x_1, x_2\rangle$ denotes the standard inner product between vectors x_1 and x_2. By the Stone-Weierstrass theorem (a crucial theorem in proving the universal approximation capabilities of neural networks, Cotter [49] and Hornik [50]) it is thus proven that signatures are a universal basis in the sense that they allow us to express non-linear path functions h as a linear function of signatures $\Phi(X)$ provided that they have some form of regularity (see next section). As for classical Taylor series, although a function might not be k-times differentiable, a higher order smooth approximation can be a quantifiably close approximation for bounded Lipschitz functions provided γ is small. In this paper, we apply this common insight to path functions, and drawdowns in particular.

[6]The proof is extremely concise, and that is why we summarize it here. Since $\Phi(X)$ is a tensor algebra of X, the family of all linear combinations of $\Phi(X)$ is an algebra. This can be seen from the shuffle product property, as explained in [47] and the primer on emiellemahieu.github.io. Constant functions are preserved by the constant zeroth term of $\Phi(X)$. The algebra separates the points as comes naturally from [44] Corollary 2.16. Stone-Weierstrass then yields that the linear family on $\Phi(X)$ is dense in the space of continuous functions on X.

4.2. Signature approximation of drawdown

We argued in section 3 that drawdown is a non-linear and non-differentiable function of its underlying path, which is smooth in the bounded differentials or Lipschitz sense. Its outcome is an interpolation between two smooth effects that are path dependent. In this section, we leverage the boundedness of section 3 together with the universality of signatures of section 4 to introduce a smooth local approximation by linear approximation of drawdown on path signatures.

We propose an approximation $\hat{\xi}(N)_t$ of ξ_t of the form:

$$\hat{\xi}(N)_t = \xi_0 + \sum_{n=1}^{N} L_n \underbrace{\int ... \int_{u_1<...<u_n, u_1,...,u_n \in [0,t]} dS_{u_1} \otimes ... \otimes dS_{u_n}}_{\Phi^n(S)} \tag{11}$$

where L_n is a vector of linear coefficients linking the drawdown at t with the signature terms of order n of the path S. In other words, considering (6) we do not know whether the impact of an individual increment is linear or none, we consider the nested impact of intervals on the drawdown. This works as a strictly non-commutative version of the exponential function Taylor expansion would suggest. The ordered iterated integrals represent the drift, Levy area, and higher order moments of the path distribution (see Chevyrev and Oberhauser [11]).

Hence, we argue that drawdown can be approximated as a linear function of these path moments. Moreover, leveraging factorial decay of the approximation error for Lip functions, for the full signature $N \to \infty$ we get an arbitrarily close approximation of ξ where the rate decay depends on the roughness of the underlying price process S (Levin et al. [51], Boedihardjo et al. [46]).

More compactly we propose to replace Eq. (10) by:

$$|\Xi(S) - \langle L, \Phi(S)\rangle| \leq \mu \tag{12}$$

Note that this arbitrary precision μ is only in theory, because in practice we rely on the truncated signatures of level M, and thus there will be an error κ that decays factorially in M (e.g. Eq. 9 for the intuition and Boedihardjo et al. [46] for the proofs):

$$\Xi(S) = \langle L, \Phi^M(S)\rangle + \kappa \tag{13}$$

Note that we could also do an equivalent truncation of the number of linear coefficients $len(L)$ rather than the signature order. However, from Eq. (11) we know that it is more natural to choose a set of linear coefficients that corresponds to a number of signature terms following the choice of M.

Proposition 4.1 looks into the consistency behaviour of κ with respect to a sample size of K sample paths drawn from $\mathcal{V}$ and next we highlight small sample properties that become apparent from the proof.

Proposition 4.1 (Consistency of linear $\Xi(S)$ approximation on signatures $\Phi(S)$). *Consider by $\mathcal{V}([0,T],\mathbb{R})$ the space of continuous paths of bounded variation $[0,T] \to \mathbb{R}$, $\mathcal{K} \subset \mathcal{V}$ is a compact subset comprising sample paths S_k, $k \in [0,...,K]$, and $\Xi : \mathcal{K} \to \mathbb{R}$ is the Lip continuous drawdown function. The approximation error κ of $\Xi(S)$ for any S in $\mathcal{V}$ by $\hat{\Xi}(S)$ is bounded through the regularity of Ξ (Proposition 3.1) and the distance in the signature space between S and any S_k, such that for $K \to \infty$, $\kappa \to 0$, or*

$$|\Xi(S) - \hat{\Xi}(S)| \to 0, \text{ for } K \to \infty \tag{14}$$

Proof. The universal nonlinearity of signatures (Eq. 10) is proven in Theorem 2.1 in Lemercier et al. [48] and means that for defined $\mathcal{V}$ and $\mathcal{K}$ there exists a truncation level $M \in \mathbb{N}$ and coefficients L such that for every $S_k \in \mathcal{K}$ we have that for any ϵ

$$|\Xi(S_k) - \langle L, \Phi^M(S_k)\rangle| \leq \epsilon \tag{15}$$

To prove that this approximation then generalizes to any $S \in \mathcal{V}$ we propose to decompose $\kappa = |\Xi(S) - \hat{\Xi}(S)|$ with a triangle inequality as suggested by Eq. (3.6) in Lyons and McLeod [45]:

$$|\Xi(S) - \hat{\Xi}(S)| \leq |\Xi(S) - \Xi(S_k)| + \tag{16}$$

$$|\Xi(S_k) - \hat{\Xi}(S_k)| + |\hat{\Xi}(S_k) - \hat{\Xi}(S)| \tag{17}$$

$$= \mathbf{A} + \mathbf{B} + \mathbf{C} \tag{18}$$

This inequality bounds the error by the regularity of Ξ in **A**, the ϵ of Eq. (15) in **B** and a signature distance in **C**:

- **A**: By Proposition 3.1, we find that $\mathbf{A} \leq C\|S - S_k\|_\infty$, or Lipschitz regularity assures that the distance between the true drawdowns of two paths is bounded by the maximal distance between the paths. For any $k \in [1,...,K]$ with $K \to \infty$ and the compactness of $\mathcal{V}$ there exists a k such that $\|S - S_k\|_\infty \to 0$ such that **A** could be reduced to zero.
- **B**: By Eq. (15), $\mathbf{B} \leq \epsilon$, such that this term is governed by the rate decay of Eq. (9). Eq. (15) guarantees we can pick an M high enough such that through Eq. (9), $\epsilon \leq C_\gamma \frac{|S|^{M+1}_{1,[0,T]}}{M!}$. Notice that (1) the required truncation level to shrink ϵ to a desired level is a function of the roughness of the underlying process (!) and (2) this term converges to zero as M can be set arbitrarily large for infinitely growing samples K.
- **C**: As per Eq. (3.7) in Lyons and McLeod [45] the difference between the approximated drawdown of two paths can be bounded by a linear combination of the difference in signatures:

$$\mathbf{C} \leq |\langle L, |\Phi^M(S_k) - \Phi^M(S)|\rangle| \tag{19}$$

  or the distance between the approximated drawdowns is, because of the linearity, proportional with the distance between two paths in the signature space (according to a choice of *rough path distance metric*). As with the first bullet, for any $k \in [1,...,K]$ with $K \to \infty$ and the compactness of $\mathcal{V}$ there exists a k such that $|\Phi^M(S_k) - \Phi^M(S)| \to 0$ such that **C** could be reduced to zero.

In sum, κ converges to zero for $K \to \infty$ provided that Proposition (3.1) holds (Lipschitz regularity), which concludes the proof. □

For finite K, we can still use the decomposition in the proof to do error analysis. The drawdown approximation will generalize well as far as (A) the maximum distance between available samples S_k and new samples is small, (B) the signature truncation level is chosen appropriately high for the given roughness of the process to let ϵ be small enough, (C) as per (A) also the distance between the signatures of observed and unobserved paths is reasonably small such that bullet **C** is small.

Finally, a remaining modelling choice that is important before we can introduce the model is the specific choice of L. As the approximation is essentially linear, Eq. (13) can be estimated using simple linear regression (OLS) and higher order polynomials are not required (as one would need to do with e.g. *logsignature* regression). However, since in practice the sample K is limited and the number of signature terms scales exponentially with M and t, one has to be mindful of overfitting for a limited sample size K, i.e. $len(\Phi^M) > K$. Therefore, we study the impact of regularized linear regression of $\Xi(S)$ on $\Phi^M(S)$ with a penalty for the number (absolute shrinkage or selection) and size (proportional shrinkage) of estimated coefficients (i.e. the elastic net (ElNet) regression). Hence, we conclude by specifying:

$$\hat{L} = \min_L(\|\Xi(S) - \langle L, \Phi^M(S)\rangle\|_2 + \lambda_1\|L\|_1 + \lambda_2\|L\|_2) \tag{20}$$

$$\hat{\Xi}(S) = \langle \hat{L}, \Phi^M(S)\rangle \tag{21}$$

where $\lambda_1 = \lambda_2 = 0$ corresponds to OLS, $\lambda_1 = 0$ corresponds to Ridge and $\lambda_2 = 0$ to LASSO regression [52]. In applications below, lambdas were set using 10-fold cross-validation.

5. The model

5.1. Introduction

In this section we introduce the intuition behind our model. The aim of our drawdown market generator is that upon convergence of (train and validation) reconstruction loss terms we have theoretical guarantees that the synthetic samples have preserved the drawdown distribution of the original samples, something which is generally not the case in generative machine learning models or DGPs with standard martingale assumptions.

First, we need to appreciate that crafting a DGP to obtain a certain level of drift, volatility or higher order moments is mathematically straightforward, as those are typically the equations that constitute the DGP. For other measures on the P&L, such as value-at-risk (VaR) or expected shortfall (ES) (Cont et al. [3]), this can be done by appreciating the direct link between quantiles and moments (Boudt et al. [53]). One would also expect drawdown to be a function of the static moments of the stochastic process described by the DGP (e.g. Douady et al. [40] and Rej et al. [41] illustrate this in the Markovian case) and the autocorrelation function in the non-Markovian case. In fact, one would expect drawdown to be a function of the moments of the path S, rather than the stochastic variable S_t.

As per the previous section, Chevyrev and Oberhauser [11]) define signatures as the moments of the path. In this article, we proposed to express drawdown as an approximate linear combination of these moments. Similar to the above logic, we will simulate path moments and weigh them according to their importance to simulate realistic drawdowns.

Another crucial insight that follows naturally from the definition of signatures, is that this approximation is a smooth function of the path. The signature is a non-parametric sum of path values. It is a complex sum because of its iterative or nested nature, but it is essentially just a sum. (Weighted) sums are differentiable. That is why Kidger and Lyons [54] focus on this differentiability property for efficient CPU and GPU implementations in their *signatory*[7] package, which we use in our Python code as well.

What we propose in our drawdown market generator in the next section is to simulate paths, evaluate their path moments, and not just take the divergence between their moments like a MMD (e.g. Buehler et al. [5]) but to weigh their moments according to pre-trained linear combinations that effectively measure the drawdown divergence between the input and the output samples. The next section will make this intuition specific.

5.2. The algorithm

The core of our algorithm is a variational autoencoder (VAE), which is a general generative machine learning architecture that links an encoder and decoder network to generate new samples from a noisy, in this case Gaussian, latent space. The idea of a Monte Carlo is to transform noise into samples that are indistinguishable from the real samples by scaling them, adding drift, and so and so forth. In other words, the neural network that constitutes the decoder is our non-parametric DGP. It does contain the parameters of the neural network, of course, but it is non-parametric in the sense that we do not have to specify the dynamics in a handcrafted formula before we can do Monte Carlo. We rather rely on the universal approximation theorems behind (even shallow) neural networks (Hornik [50] and Cotter [49]) to approximate a realistic DGP by simply iterating data through the network and updating the parameters θ with feedback on the drawdown distribution of the batch, assuring that the approximated DGP converges in train and validation loss to the empirical DGP.

We will not discuss the VAE architecture in depth here, but refer the interested reader to Kingma and Welling [55] for details on the encoder and decoder networks g, backpropagation, the latent Kullback-Leibler $\mathcal{L}_L$ loss and the standard $L2$ reconstruction loss $\mathcal{L}_R$. Although we should stress that the main reason for picking a rather standard VAE (over restricted Boltzmann machines, generative adversarial networks, generative moment matching networks or normalizing flow-based VAEs) is their simplicity, speed, flexibility, scalability and stability during training. Boltzmann machines are very efficient to train, but the energy-based loss function and their binary values makes them very inflexible for adjusting loss functions. Through the discriminator mechanism GANs are most flexible and a very popular choice in related literature, but notoriously expensive to train in terms of required data and speed, and associated instabilities such as *mode collapse* and *vanishing gradients* leading to subpar results (Eckerli and Osterrieder [56]).

[7] `https://pypi.org/project/signatory/`

Algorithm 1 VAE market generator given drawdown function Ξ

Input Historical price paths $S : [0, T] \to \mathbb{R}$, hyperparameters, signature truncation level M and feature weight α.

Output Trained VAE Market Generator g_θ

procedure TRAIN
  Divide historical sample into batches $\mathcal{B}$ of length τ, calculate the signatures of these paths truncated at level M, $\Phi_M^{\mathcal{B}}$, calculate the drawdowns Ξ of these paths $\Xi(S^{\mathcal{B}})_\tau = \int_0^\tau (max_{t_i<t}(S_{t_i}^{\mathcal{B}}) - S^{\mathcal{B}})dt$ denoted $\hat{\Xi}(S_b)$
  $\hat{L} \leftarrow LinearRegression(\hat{\Xi}(S^{\mathcal{B}}), \Phi_M^{\mathcal{B}})$
  Initialize the parameters θ of the VAE.
  for $i : \{1, ..., N\}$ **do**:
    Sample a batch $\mathcal{B}$ and pass it through the encoder g_θ and . decoder network g_θ^{-1}
    Calculate drawdown $\Xi(S')$ of the output sample S' using the differentiable signature approximation: $\langle \hat{L}, \Phi_M(S') \rangle$
    Define the reconstruction loss term as the weighted average . of RMSE error and drawdown loss: $\mathcal{L}_R = \mathbb{E}_{\mathcal{B}} \|S - S'\|^2 + \alpha \mathbb{E}_{\mathcal{B}} \|\langle \hat{L}, \Phi_M(S) \rangle - \langle \hat{L}, \Phi_M(S') \rangle\|^2$
    $\mathcal{L} = \mathcal{L}_L + \mathcal{L}_R$
    $\theta \leftarrow \theta - l \frac{d\mathcal{L}(\theta)}{d\theta}$

Algorithm 2 Sampling from the market generator

Input Trained VAE Market Generator g_θ.
Output N_s generated samples S'

procedure GENERATE
  for $j : \{1, ..., N_s\}$ **do**
    Sample a random Gaussian variable Z
    $S' \leftarrow g_\theta^{-1}(Z)$

The proposed algorithm[8] is provided in Algorithm 1 and 2. In short, we propose to include the divergence between the observed drawdown distribution of a batch and the synthetic drawdown distribution in the reconstruction loss function. The market generator can be interpreted as a moment matching network, adding the moments of drawdown rather than returns[9].

6. Numerical results

This section comprises the numerical results of the outlined methods. First, we discuss the accuracy of linear approximation of drawdown in the signature space on simulated and real world data. Second, we discuss the accuracy of the drawdown market generator.

[8]The implementation can be found on `https://github.com/emiellemahieu` for reproduction purposes, where one can also find the code of the simulation studies in 6 for full reproducibility. Also see `https://emiellemahieu.github.io` for updates.

[9]This distance over a particular batch is identical to the distance between the real and fitted distributions in e.g. Figure 7, which is the drawdown distribution we want to preserve in synthetic samples.

6.1. Linear drawdown approximation with signatures

This part analyses the accuracy of the approximation. Section 6.1.1 describes the simulation set up and results for fractional Brownian simulated data. Section 6.1.2 discusses the accuracy of the approximation on empirical data.

6.1.1. Bottom-up simulations

Simulation set up. We expect to find uniform decay as a function of truncation level M of estimated drawdown approximation error $\hat{\kappa}$ in-sample, where the decay constant is a function of the roughness of the underlying process γ. Out of sample we rely on the error decomposition in Proposition 4.1 and expect the error to shrink to zero if K grows very large. Since the roughness of empirical data is unknown and has to be estimated, it is useful to test the approximation in an experimental setup where γ can be specified. We thus first test the approximation (20) on simulated Brownian motion (BM) and fractional Brownian motion (fBM) paths.

Consider first the simplest case of homoskedastic BM ($dS_t = \mu dt + \sigma d\epsilon$). In this simple case, the price path S can thus be seen as the cumulative sum process of a random uncorrelated Gaussian, scaled with σ and added a deterministic drift μ. We consider piecewise linear paths of length T=20 days with values $\mu = 1\%/252$ and $\sigma = 20\%/252$.

Next, fBM implements BM where the uncorrelated Gaussian increments are replaced by fractional Gaussian increments that have a long-memory structure. The martingale property that the autocovariance between Gaussian increments has expectation zero, is replaced by a generalized autocovariance function for two increments at t and s (i.e. lag $t - s$):

$$E[B_H(t)B_H(s)] = \frac{1}{2}[|t|^{2H} + |s|^{2H} - |t - s|^{2H}] \qquad (22)$$

where H is the so-called Hurst exponent (H). Note that a $H = 0.5$ corresponds to Brownian increments, while $H > 0.5$ yields smooth, persistent, positively autocorrelated paths and $H < 0.5$ yields rough, antipersistent negatively autocorrelated paths. Intuition tells us that the smaller H, the more granularity the path has, and the worse the approximation will become.

There are hence three dimensions to this simulation study. We want to evaluate the estimated error $\hat{\kappa}$ as a function of (1) roughness H, (2) signature approximation order M and (3) simulation size K. Therefore, we:

- Vary H between 0.4 and 0.7 with step size 0.05
- Vary M between 1 and 10 (naturally with unit steps)

- Repeat the experiment for K in [1000, 5000, 10000, 20000, 50000]
- Fit regression on K samples (train) and simulate $p_{test}K$ new samples to evaluate test accuracy ($p_{test} = 0.1$).

Results and discussion. Figure 1 shows a RMSE accuracy heatmap that summarizes these three dimensions for the 350 experiments with Linear Regression[10]. These three dimensions go down in order of K, then H, then M. We thus expect the approximation to be better as we progress down to the bottom of the map. It highlights the important findings in one image:

- As we go down the approximation does become better, both in $RMSE$ and R^2 terms.
- The discrepancy between in-sample and out-of-sample accuracy is high(er) for the smaller sample, high M approximations, which is indicative of overfitting. We will delve deeper into this relationship.
- Globally, the first bullet becomes apparent, which means the approximation becomes better with sample size K.
- Locally, the red lines separate the different N so in between them only H and M vary. It is clear that accuracies improve much with increasing H.
- More fine-grained, the grey lines separate the H, so in between only M varies. It is obvious for LinReg and larger samples that higher M implies higher accuracy.
- For small sample sizes the in- and out-of-sample performances differ a lot for LinReg, which indicates overfitting while this is not the case for large N or Elnet regression.
- Although the out-of-sample (OOS) variance is more consistent, the overall performance of biased regularized regression is much worse.
- Within the grey lines ElNet regression seems to have little difference, higher signature orders seem to contribute little to nothing with regularization.
- Finally, in- and out-of-sample consistency is found for larger ($log(K) > 3$) simulations.

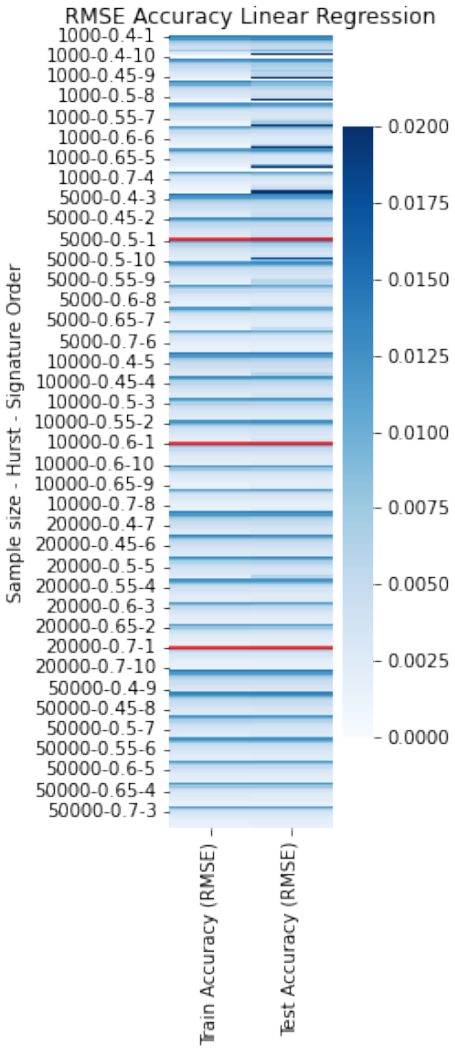


Figure 1: Accuracy Heatmap (Linear Regression)

[10]Same heatmaps for R2 and ElNet are included in Appendix 7

Figures 2, 3 and 4 and 5 take a slice of these heatmaps and investigate individual dimensions. Given our first comments, we are initially interested in the stability of linear regression and evaluate the in- and out-of-sample discrepancy as a function of sample size. Figure 2 shows the difference between in- and out-of-sample accuracy as a function of sample size $log(K)$. To be more precise, we averaged the difference for all simulation sizes over either the values of M to get a curve per value of H (blue line) or the opposite (red line). The conclusions are twofold: (1) for a sufficiently large sample ($log(K) > 3$) the overfitting problem disappears, which argues in favour of the proposition that the relationship is not spurious and relates to Proposition 4.1 in the sense that for $K \to \infty$ the three error components that explain differences in in- and out-of-sample performance shrink to zero, and (2) the in-sample/out-of-sample accuracy divergence is significantly higher when deconfounding for M implying an improperly chosen M for smaller samples might lead to overfitting.

Next, we look in Figure 3 the train and test R^2 as a function of the signature approximation order M. Now we take the average performance over K and H and find uniform increase of performance, except for a kink around $M = 8$. When deconfounding for sample size K, which you can find on the right hand side of Figure 3, we find that only small samples do not show uniform increase, which suggest that signature order $M > 8$ will overfit for $log(K) \leq 3$. This also makes sense given previous comments. When looking at the $RMSE$ in Figure 4, we have exactly the same findings. Train $RMSE$ shows nice uniform decay, except for $M > 8$ where it starts to overfit. When deconfounding for K in the same figure on the right hand side, we see this problem only occurs for small sample sizes ($log(K) \leq 3$). Finally, we check the third dimension of the simulation, the relationship between the accuracy of the approximation and the roughness H of the assumed process. We find approximate[11] uniform increase of R^2 and uniform decay in $RMSE$.

In summary, the simulation study confirms with our initial expectations (e.g. Proposition 4.1) but raises some practical warnings as well:

- For a large number K of sample paths, say $log(K) > 3$, one generally does not get into any issues and finds uniform decay in κ with both H and M, both test and training, i.e. without overfitting. In terms of Proposition 4.1, the distances between any S and the S_k become smaller with K and the estimated $\hat{L}$ generalize better. Linear regression is then the best unbiased estimator and the advised way to go.

- For small samples, there are no guarantees that in-sample fit is indicative of out-of-sample fit. This relates to Proposition 4.1 in the sense that the distance between any S and S_k (in inf-norm and signature terms) might be relatively large. We also see bad generalization for large M, because larger M implies more regression terms and therefore more room for overfitting. For small samples, say $log(K) < 3$, we expect the biased regularized El-Net to give more robust results, and in general signatures of relative low order $M < 8$ are advised as not to use too many terms in the regression.

- Finally, and importantly, (1) the accuracy that is good enough for the application at hand of course depends on the application, (2) we will deal with data sets that are considered large for these standards.

6.1.2. Empirical data

Simulation set up. Consider U=4 investible instruments: equity (S&P500), fixed income (US Treasuries), commodities (GSCI index) and real estate (FTSE NAREIT index). We collect price data $S : [0, T] \to \mathbb{R}^U$ (adjusted close prices) between Jan 2000 and May 2022, which gives us T=5840 daily observations. Clearly, these 4 different asset classes have different return, volatility and drawdown characteristics. This can clearly be seen from Figure 6.

As an investor, we are interested in the drawdown of a portfolio $\mathbf{w}$, $w_i, i \in 1, ..., U$, which allocates a weight w_i to each investible asset. In these experiments, we attach $P \in \mathbb{N}$ sets of weights to each asset and pick a $\tau < T$ such that we have $T - \tau$ overlapping sample paths or a simple buy-and-hold strategy over τ days for every $p \in [0, ..., P]$. Here we first pick $\tau = 20$, so we model monthly sample paths.

The drawdown distribution of these hypothetical portfolio paths is shown in Figure 7. This was the probability measure of crucial interest in the previous sections and our application in the next section[12]. In our market generator, we thus adapt parameterized versions of sample paths such that their drawdown distribution converges towards this empirical drawdown distribution. In

[11] Note that this is a mean over M values.

[12] And of course in above-mentioned real-world financial problems such as max loss insurance or portfolio drawdown control.

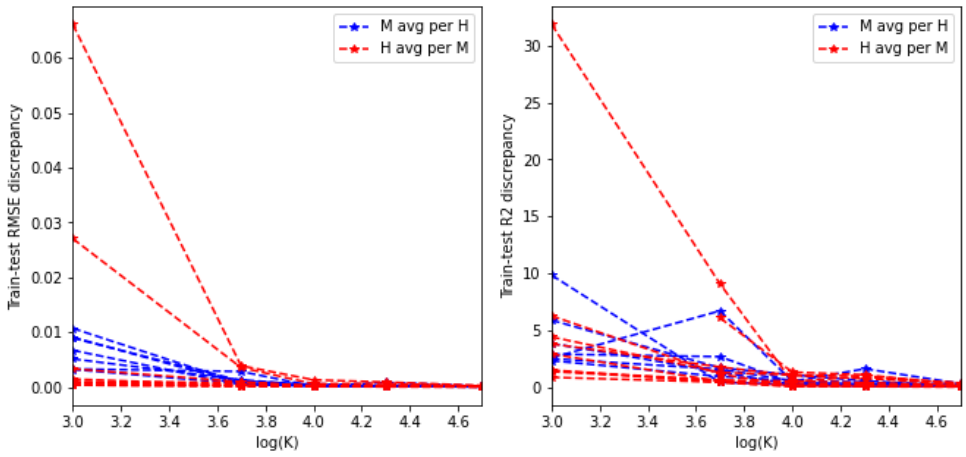


Figure 2: In- and out-of-sample accuracy difference as a function of sample size $log(K)$ (Red = averaging out over H to get a line per value of M, and blue = opposite).

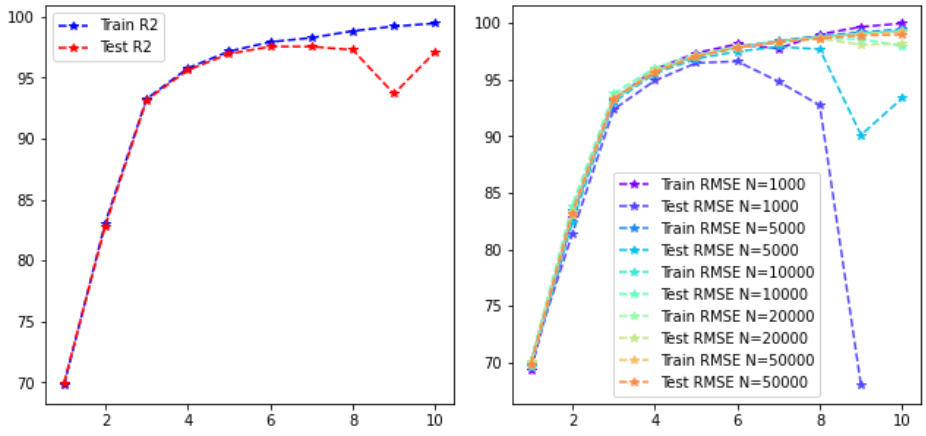


Figure 3: Train and test R^2 (y-axis) as a function of signature approximation order M. Average over N and H is shown on the left. An average per N is shown on the right.

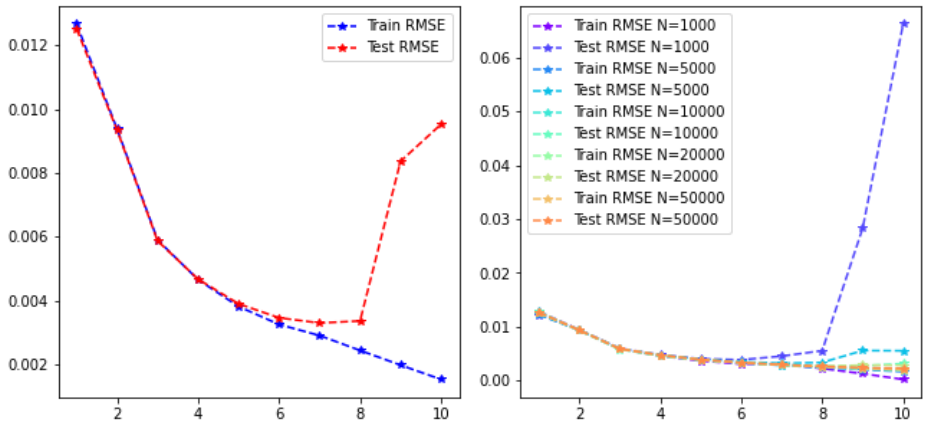


Figure 4: Train and test $RMSE$ (y-axis) as a function of signature approximation order M. Average over N and H is shown on the left. An average per N is shown on the right.

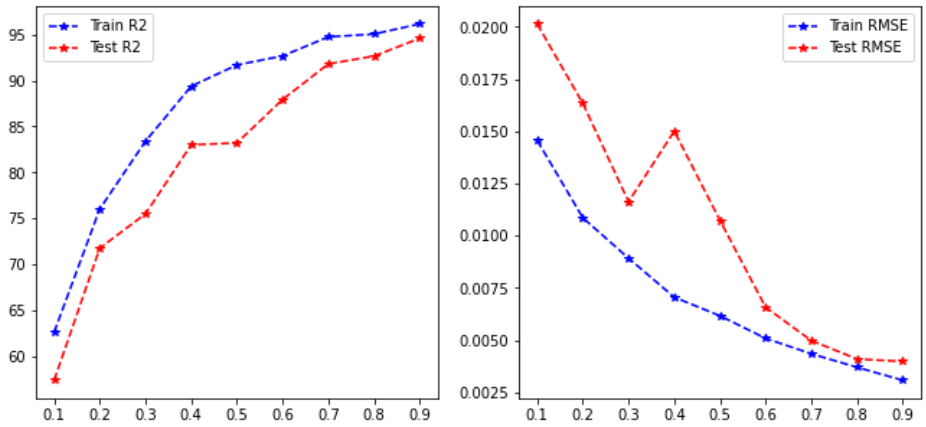


Figure 5: Train and test R^2 (left) and $RMSE$ (right) as a function of the roughness H

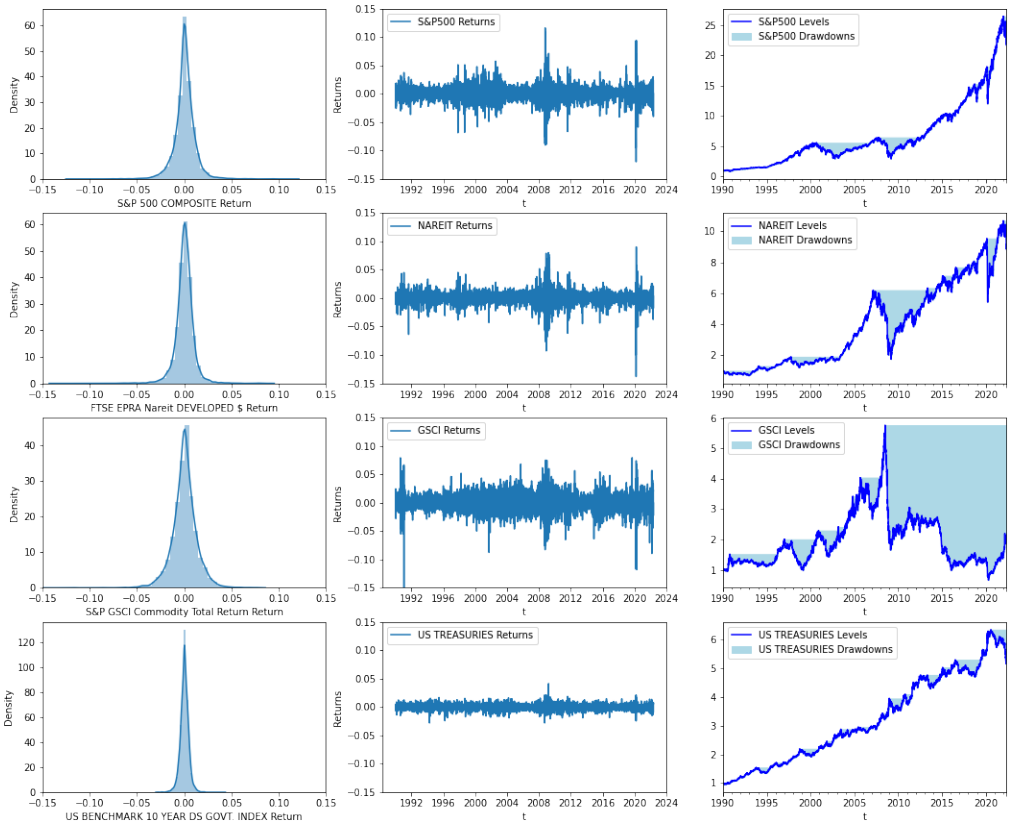


Figure 6: Data overview: return, volatility and drawdown heterogeneity among asset classes.

Figure 7, we also compare the distribution with the distribution that is implied by standard simulated Brownian motion with the same volatility σ parameter as the portfolios we have constructed. Note that one could also use the closed forms from Douady et al. [40], Rej et al. [41] and alike here. We importantly notice that the blue density is merely an optimistic lower bound on the true distribution of portfolio drawdown, and misses out on the tails. This can also be seen in Figure 8. The link between generalized stochastic processes and this distribution is far from trivial, as one can not rely on Lévy's theorem anymore (in contrast to the theoretical (f)BM case). In the next section, our methodology thus advocates a fundamentally non-parametric approach to simulate paths that closely resemble the correct drawdown distribution.

In this simulation we will:

- Generate a set of portfolio weights $\mathbf{w}_p$ (index p for portfolio) and construct $T - \tau$ paths, based on taking blocks from the dot product of w_p and S's cumulative returns, the univariate portfolio paths S_p
- Calculate the drawdown $\Xi(S_p^{\mathcal{B}})$ and signatures $\Phi^M(S_p^{\mathcal{B}})$ for each block $\mathcal{B}$ and regress the two (for increasing M in [1, ..., 10]), with defining the $(1 - p_{test})(T - \tau)$ first blocks as train sample and remaining blocks as test sample.
- Repeat $P = 100$ times and report average performance.

Results and discussion. Our conclusions are analogous to the bottom-up simulation experiments. The accuracies are shown in Tables 1 and 2. Given the large K, train en test R^2 and $RMSE$ are consistent, answering our biggest concern. From $M > 5$ we get accurate approximations of ξ both in R^2 and $RMSE$ terms, after which it further improves as expected at a much slower rate.

We also included the computation time for calculating one signature of the order. Clearly for large samples the computational bottleneck, i.e. the hardware, will influence the exact choice of M as the accuracy increases monotonically but with diminishing returns versus a quadratically increasing computing cost.

In line with expectations, ElNet performs significantly worse. From $M > 4$ the regressions are not improved. Train and Test accuracy is as consistent as LinReg, so one would be inclined to stick to ordinary least squares.

It is worthwile to note that, ceteris paribus, assuming K and M are large, the maximum $80\%R^2$ and minimum

M	Avg train R2	Time	len($\Phi^M(S)$)	Avg test R2	Avg train RMSE	Test RMSE
1	0.0100	0.00211	2	0.0110	0.0142	0.0141
2	0.1973	0.00460	6	0.2036	0.0128	0.0126
3	0.4833	0.00268	14	0.4819	0.0102	0.0102
4	0.6670	0.00283	30	0.6728	0.0082	0.0081
5	0.6991	0.00208	62	0.7065	0.0078	0.0077
6	0.7335	0.00209	126	0.7408	0.0074	0.0072
7	0.7488	0.00271	254	0.7559	0.0071	0.0070
8	0.7677	0.00201	510	0.7752	0.0069	0.0067
9	0.7843	0.00284	1022	0.7899	0.0066	0.0065
10	0.8021	0.00301	2046	0.8070	0.0063	0.0062

Table 1: Linear regression fit for empirical data

M	Avg train R2	Time	len($\Phi^M(S)$)	Avg test R2	Avg train RMSE	Test RMSE
1	0.0008	0.0021	2	-0.0020	0.0140	0.0136
2	0.0616	0.0022	6	0.0600	0.0136	0.0132
3	0.4367	0.0016	14	0.4459	0.0105	0.0101
4	0.6348	0.0015	30	0.6449	0.0085	0.0081
5	0.6611	0.0015	62	0.6725	0.0081	0.0078
6	0.6612	0.0015	126	0.6732	0.0081	0.0077
7	0.6594	0.0024	254	0.6719	0.0082	0.0078
8	0.6580	0.0018	510	0.6707	0.0082	0.0078
9	0.6568	0.0026	1022	0.6697	0.0082	0.0078
10	0.6559	0.0021	2046	0.6688	0.0082	0.0078

Table 2: Elastic net CV(10) fit for empirical data

6.5 bps $RMSE$ error corresponds to a roughness of our portfolios of about $H = 0.5$. Actually, slightly less from R^2 point of view and slightly more from $RMSE$ perspective. However, this roughly confirms our expectation that real-life portfolios are approximately Brownian, or approximately efficient (nor mean-reverting, nor trending a lot) on this timescale, although with fatter tails. The approximation seems to grasp both, as e.g. Figures .14, .15 and .16 in Appendix 2 (Section 7) suggest: the linear signature approach has an overall good fit without missing out on the higher drawdown scenarios.

6.2. Drawdown market generator: results and discussion

The main findings are summarized in Figures 9, 10, 11 and 12. Figure 9 shows the generated paths of the standard VAE model versus the drawdown market generator ξ-VAE. It is clear that the original VAE does not do a bad job at generating realistic scenarios, but it is reminiscent of Figure 7 in the sense that the paths are non-Brownian, or returns non-normal, but still too centered around the Brownian-like distribution (the densely colored areas).

This is also clear from Figure 11. To summarize our architecture rather simplisticly, the standard VAE focuses on reproducing the top distribution and tries to do a good average job over a batch ($L2$ loss) such that it misses out on the tails. The ξ-VAE fits both the top and the bottom distribution such that it includes more tail scenarios. It is again obvious from an autocorrelation point of view that a standard VAE thus shows a lack of extreme adverse scenarios (remember this important

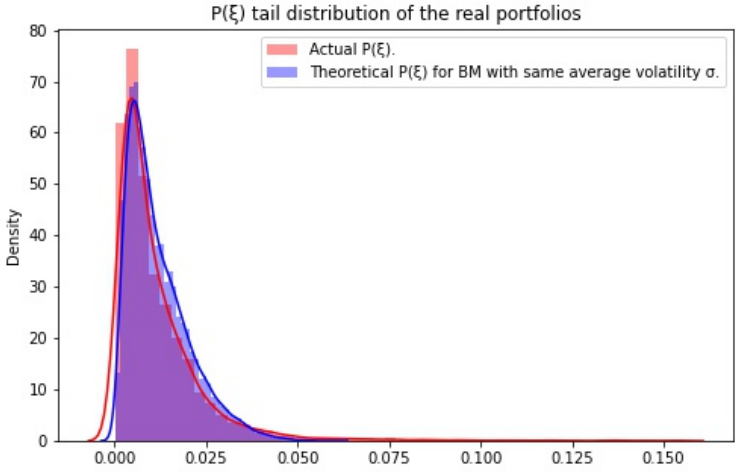


Figure 7: Drawdown distribution empirical real-world portfolios versus fitted theoretical drawdown distribution if the underlying DGP would be Brownian Motion (BM) with the same average volatility as the sample paths.

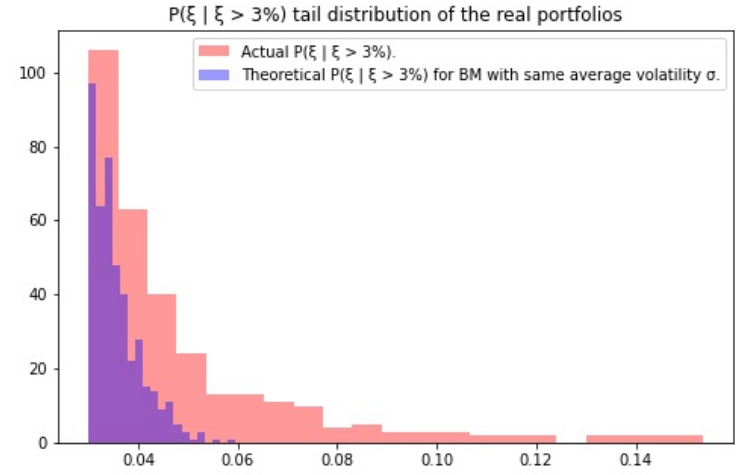


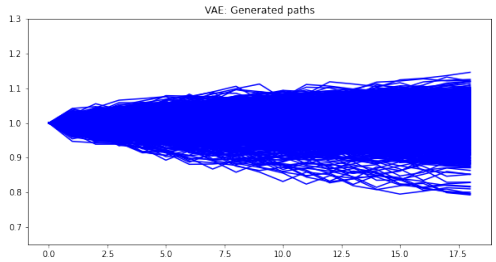
Figure 8: Zoom on the tail of the empirical versus theoretical drawdown distribution.

point in the introduction 1 and 3). This also becomes apparent in Figure 10, which plots the fake and real drawdowns as a scatter. The standard VAE does a good job in capturing most part of the moderate drawdowns, but is scattered towards the higher drawdowns and produces none in the tail, while this is resolved with the ξ-VAE. A closely related graph is Figure 12, which plots the same scatter but using ordered observations (i.e. quantiles) and a first bisector line which would fit the data if the synthetic drawdown distribution has exactly the same quantiles as the original distribution. Again we find that the VAE misses out on capturing tail drawdown scenarios, while ξ-VAE does a much better job. Importantly, this could be done by just memorizing the training samples and reproducing them exactly. However, the *bottleneck* VAE architecture is designed to summarize the training data in fewer parameters (i.e. the non-parametric DGP) as a dimension reduction, rather than trivially mirroring the training data. That is why one sees different simulated paths than identical ones to the input samples, and why by using the trained decoder as a non-parametric DGP one can create an *infinite* amount of genuinely new data with the same distribution as the training data. Moreover, both train and validation losses flat out at convergence (see appendix 7 with details on the training convergence) while purely mirroring data would imply training losses would be minimal at the cost of high validation losses.

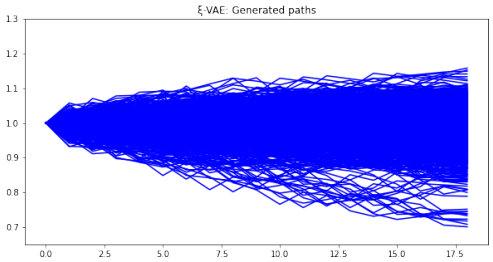
7. Conclusion

Learning functions on paths is a highly non-trivial task. Fortunately, rough path theory offers tools to approximate functions on paths that are sufficiently smooth by considering a controlled differential system and iterating the effect of intervals of the (rough) path on the (smooth) outcome function. One key path dependent risk functional in finance is a portfolio value path's drawdown. This paper takes the perspective of portfolio drawdown as a non-linear dynamic system (a controlled differential equation $d\xi$), rather than directly analyzing the solution, the exact expression of Ξ. It relied on some important insights from the theory of rough paths to pinpoint that by taking this perspective, rather than continuous differentiability, a more general regularity condition of bounded differentials is sufficient to interpolate between two types of effects of a path on its resulting drawdown and thus locally approximating this drawdown as a function of a path without having to evaluate Ξ explicitly. We prove this regularity w.r.t. drawdown and consistency of the approximation. On simulated Brownian, fractional Brownian and on real-world data, regression results exhibit a good fit for ordinary least squares linear regression, and do not favor regularized versions (LASSO, ElNet) when one has a reasonable sample size. Finally, a proposed application of the approximation is a so-called *market generator* model that evaluates the synthetic time series samples in terms of their drawdown. We argue that by doing so one gets more realistic scenarios than the standard VAE model and one is able to reproduce the drawdown distribution without trivially memorizing it.

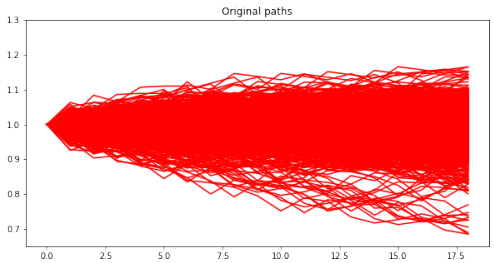
Future work will focus on extending this application and further applying it to portfolio drawdown optimization, where one can e.g. test a data hungry drawdown control strategy over a host of synthetic scenarios rather than the single historical one. One of the use cases that has not been mentioned yet is non-parametric Monte Carlo as a mathematically principled data augmentation technique. Denoising autoencoders have served as a tool to denoise historical data by, ironically, adding noise to the data and let the model learn the difference. A regression, classification or optimization model can be trained as an *ensemble* of the noisy predictions, such that by construction on average one is close to the true (denoised) regression prediction, classification or optimal decision. For financial time series a similar motivation could be made. Defining an optimal portfolio as an ensemble average (e.g. bootstrapped) of multiple historical scenarios (implicitly) assumes noise is cancelled out, or robustness is achieved, by taking an additional mean. With the dimension reduction tool proposed in this paper and the added *noise-by-construction* one makes this idea explicit, rather than removing *noise-by-assumption*. This non-parametric Monte Carlo idea could hence robustify his or her methodology further as a mathematically principled data augmentation technique. Moreover, the fundamentally non-parametric nature of our Monte Carlo engine opens possibilities to full non-parametric pricing of max loss or drawdown insurance options.



(a) Generated VAE

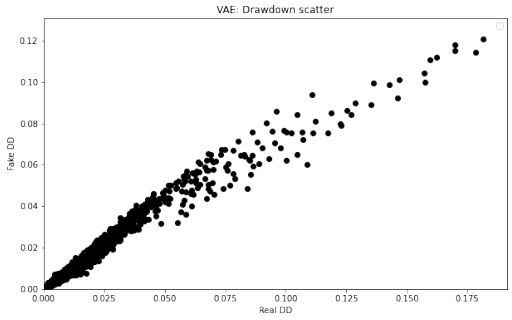


(b) Generated ξ-VAE

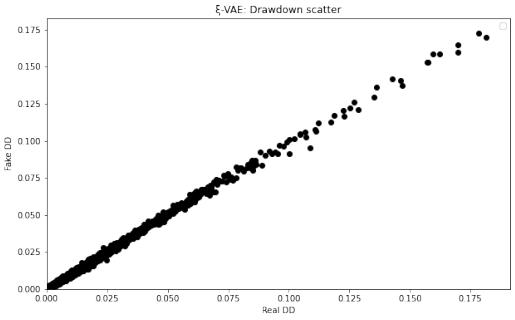


(c) Original

Figure 9: Generated paths vs original paths

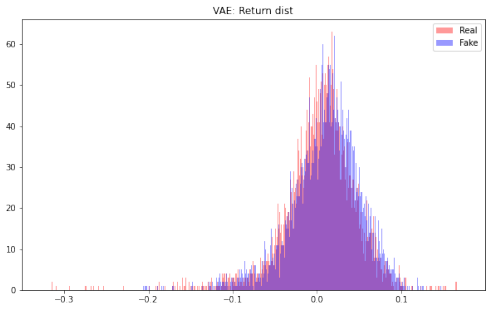


(a) VAE

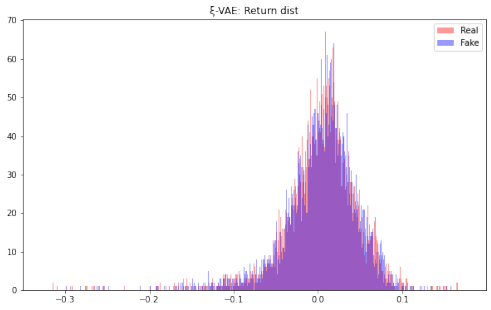


(b) ξ-VAE

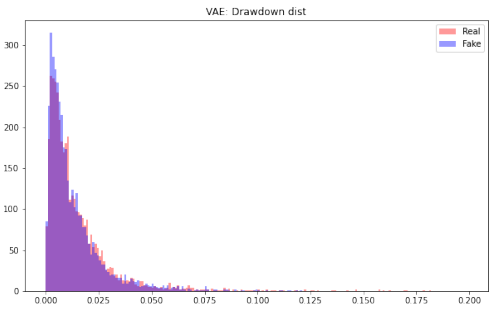
Figure 10: Synthetic versus real drawdowns scatter



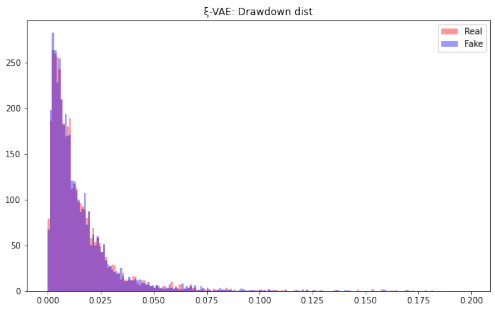
(a) Return distribution VAE



(b) Return distribution ξ-VAE

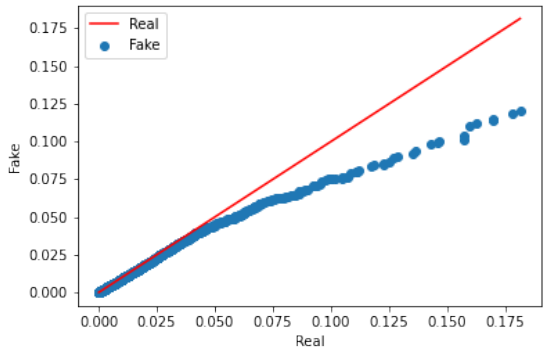


(c) Drawdown distribution VAE

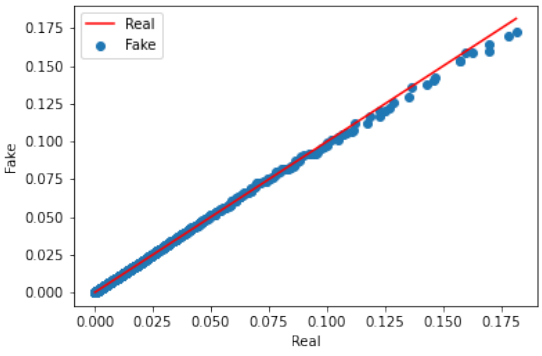


(d) Drawdown distribution ξ-VAE

Figure 11: Generated versus real drawdowns and returns distribution



(a) VAE fake versus real ξ quantiles



(b) ξ-VAE fake versus real ξ quantiles

Figure 12: Drawdown QQ plots

Appendix: Controlled differential equations and path signatures

Controlled differential equations

We are generally interested in a CDE of the form:

$$dY_t = g(Y_t)dX_t \tag{.1}$$

where X is a path on $[0,T] \to \mathbb{R}$, called the driving signal of the dynamic system. g is a $\mathbb{R} \to \mathbb{R}$ mapping called the physics that models the effect of dX_t on the response dY_t. A controlled differential equation (CDE) distinguishes itself from a ordinary differential equation in the sense that the system is controlled or driven by a path (dX) rather than time (dt) or a random variable (stochastic SDEs, $d\epsilon$).

Signatures

A series of coefficients of the path that naturally arrives from this type of equation is the series of iterated integrals of the path, or the path signature Φ. The signature of a path $X : [0,T] \to \mathbb{R}$ can be defined as the sequence of ordered coefficients:

$$\Phi(X) = (1, \Phi_1, ..., \Phi_n, ...) \tag{.2}$$

where for every integer n (order of the signature):

$$\Phi_n(X) = \int_{u_1<...<u_n, u_1,...,u_n \in [0,T]} ... \int dX_{u_1} \otimes ... \otimes dX_{u_n} \tag{.3}$$

where we define the n-fold iterated integral as all the integrals over the n ordered intervals u_i in [0,T]. The signature is the infinite collection for $n \to \infty$, although typically lower level M truncations are used.

$$\Phi^M(X) = (1, \Phi_1, ..., \Phi_M) \tag{.4}$$

Picard Iterations

The idea behind a Picard iteration is to define for:

$$dY_t = g(Y_t)dX_t \tag{.5}$$

a sequence of mapping functions $Y(n) : [0,T] \to \mathbb{R}$ recursively such that for every $t \in [0,T]$:

$$Y(0)_t \equiv y_0 \tag{.6}$$

$$Y(1)_t = y_0 + \int_0^t g(y_0)dX_s \tag{.7}$$

$$Y(n+1)_t = y_0 + \int_0^t g(Y(n)_s)dX_s \tag{.8}$$

Now by simple recursion one finds that (for a linear g):

$$Y(n)_t = y_0 + \sum_k^n g^{\otimes k}(y_0) \int_{u_1<...<u_n, u_1,...,u_n \in [0,T]} ... \int dX_{u_1} \otimes ... \otimes dX_{u_n} \tag{.9}$$

Such that a solution for Y_t would be:

$$Y_t = y_0 + \sum_k^\infty g^{\otimes k}(y_0) \int_{u_1<...<u_k, u_1,...,u_k \in [0,T]} ... \int dX_{u_1} \otimes ... \otimes dX_{u_k} \tag{.10}$$

This result shows how the signature, as an iterative representation of a path over ordered intervals, naturally arises from solving CDEs using Picard iterations, and how it is a natural generalization of Taylor series on the path space when the physics is linear.

$$g^{\circ 1} = g \tag{.11}$$

$$g^{\circ n+1} = D(g^{\circ n})g \tag{.12}$$

then it is natural to define the N-step Taylor expansion for Y_t by $\hat{Y}(N)_t$ as:

$$\hat{Y}(N)_t = y_0 + \sum_{n=1}^N g^{\circ n}(y_0) \int_{u_1<...<u_n, u_1,...,u_n \in [0,T]} ... \int dX_{u_1} \otimes ... \otimes dX_{u_n} \tag{.13}$$

Clearly, $\hat{Y}(N)_t$ is linear in the truncated signature of X up to order N[13].

Example. The simplest example of:

$$dY_t = g(Y_t)dX_t \tag{.14}$$

Is a linear physics for a linear path X:

$$dY_t = Y_t dX_t \tag{.15}$$

where:

$$g = g^\circ \tag{.16}$$

$$g^{\circ n+1} = D(g^{\circ n})g \tag{.17}$$

$$X_t = X_0 + \frac{X_T - X_0}{T}t \tag{.18}$$

and assuming:

$$y_0 = 1 \tag{.19}$$

$$X_0 = 0 \tag{.20}$$

[13] Moreover, the error bounds of $\hat{Y}(N)_t$ to approximate Y_t yield a factorial decay in terms of N, i.e. $|Y_t - \hat{Y}(N)_t| \leq C\frac{|X|^{N+1}_{1,[0,t]}}{N!}$. This result can be extended to p-geometric rough paths where g is a $Lip(K)$ where $K > p-1$ [46].

Indeed, this yields the exponential function $Y_t = \exp(X_t)$. For non-linear driving signals (where the order of the events matter), one generally gets a non-commutative version of the exponential function in Eq. (.10)! For linear time, the order of events does not matter and we generally get the increment of the path raised to the level of the iterated integral, divided by the level factorial (i.e. the area of an n-dimensional simplex).

This can be seen from:

$$Y_t = y_0 + \sum_{n=1}^{N} Y^{\circ n} \int_{u_1<...<u_n, u_1,...,u_n \in [0,T]} ... \int dX_{u_1} \otimes ... \otimes dX_{u_n} \quad (.21)$$

now it is easy to see that:

$$\begin{aligned} \Phi^n &= \int_{u_1<...<u_n, u_1,...,u_n \in [0,t]} ... \int dX_{u_1} \otimes ... \otimes dX_{u_n} \\ &= \int_{0<u_1<...<u_n<t} ... \int d(\frac{X_t}{t} u_1)...d(\frac{X_t}{t} u_n) \\ &= \prod_{j=1}^{n} (\frac{X_t}{t}) \int_{0<u_1<...<u_n<t} ... \int du_1...du_n \\ &= \frac{1}{t^N} \prod_{j=1}^{n} (X_t) \frac{t^N}{n!} = \frac{(X_t)^n}{n!} \end{aligned} \quad (.22)$$

such that:

$$Y_t = y_0 + \sum_{n=1}^{N} y_0 \frac{1}{n!} (X_t)^n = 1 + (X_t) + \frac{(X_t)^2}{2!} + \frac{(X_t)^3}{3!} + ... \quad (.23)$$

Which is the classical Taylor expansion for the exponential function, i.e. linear physics integrated over a linear path (time). Now the signature approximation is the generalization of this idea to the path space (i.e. Y is a function on a path), where the path X_t need not be a linear map of time, and the physics g need not be linear (e.g. drawdown Eq. (6)).

Appendix 2: Regression fits details

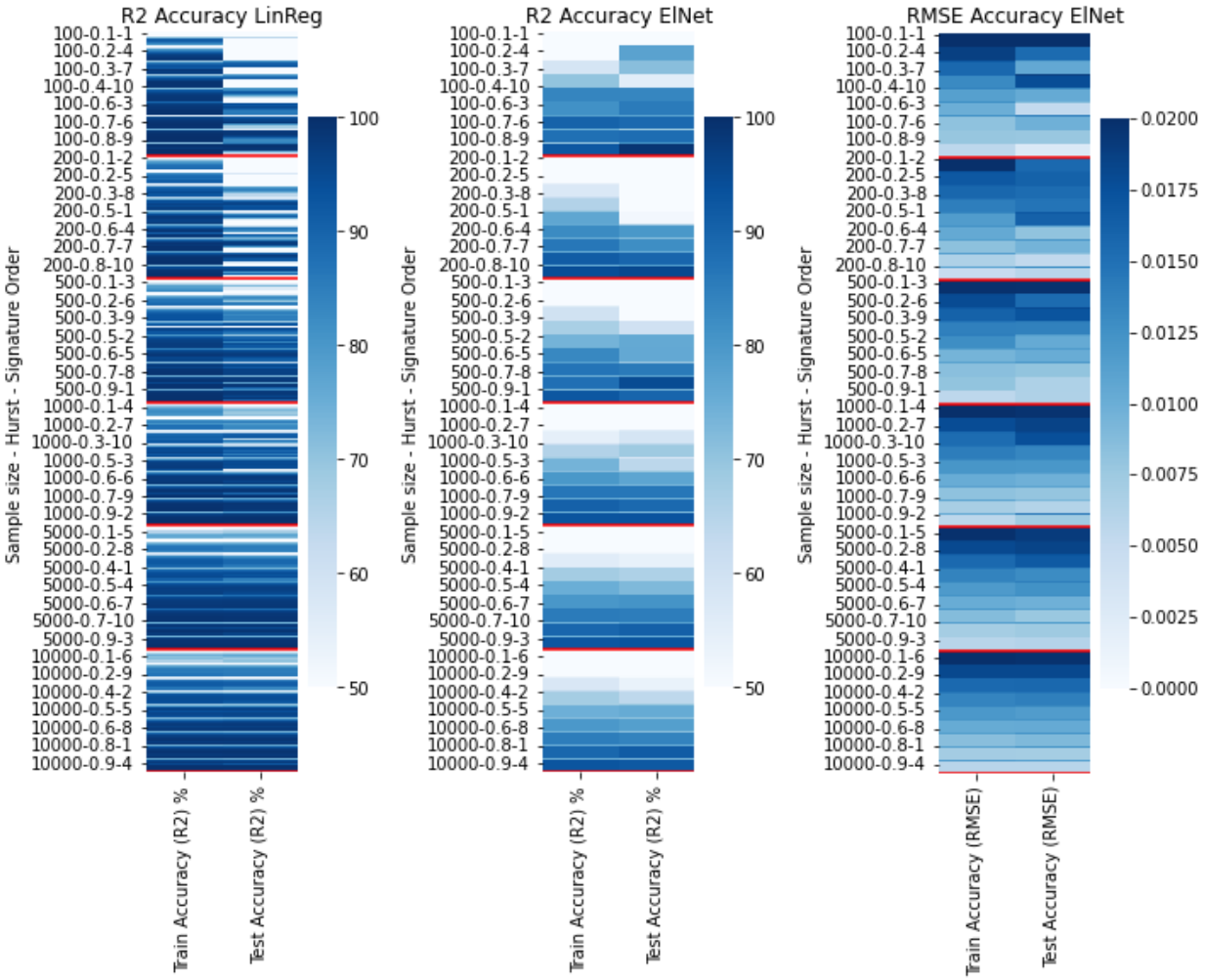


Figure .13: Extended accuracy heatmap: R2 and RMSE for ElNet and further details.

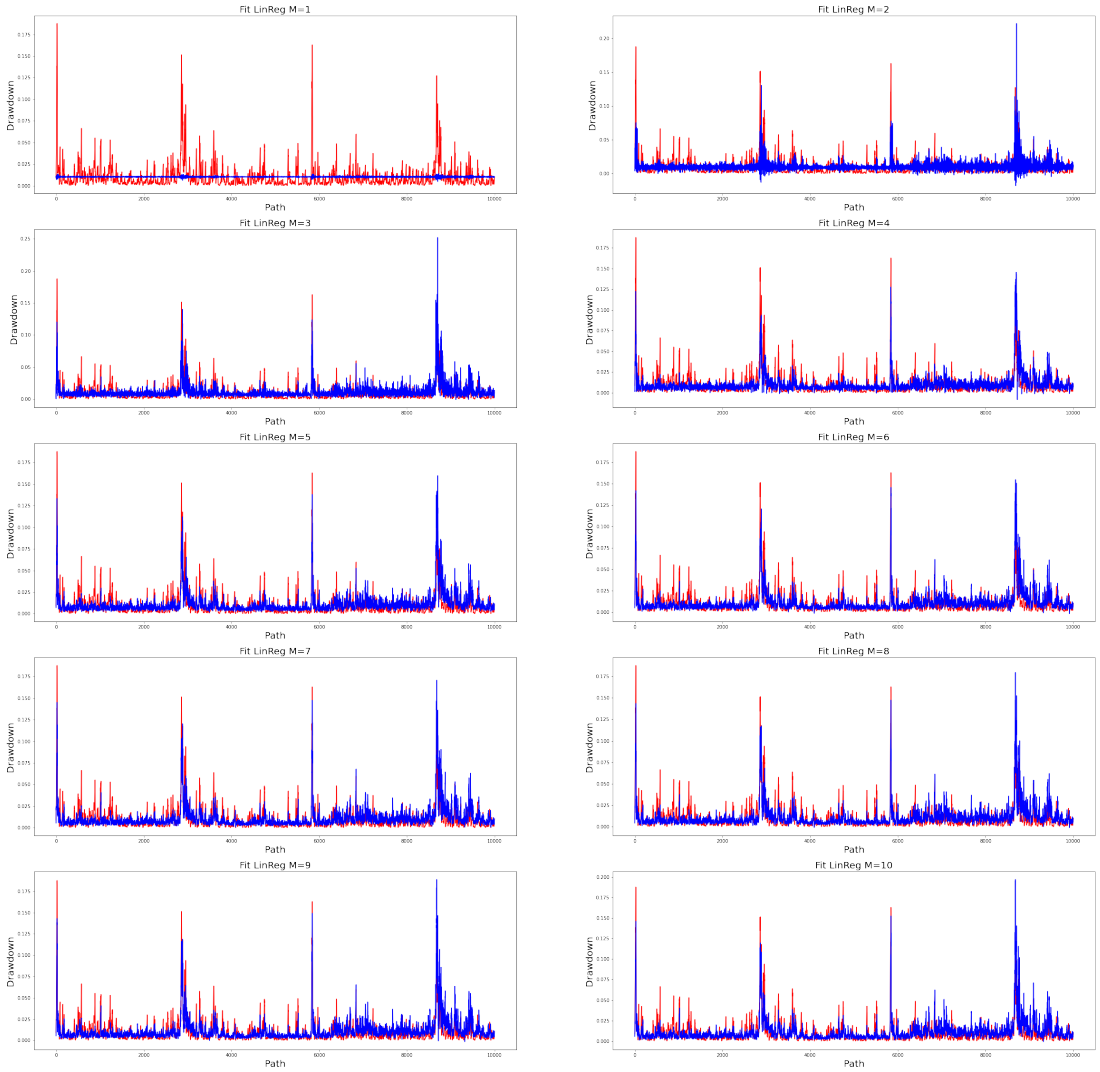


Figure .14: Linear regression fit

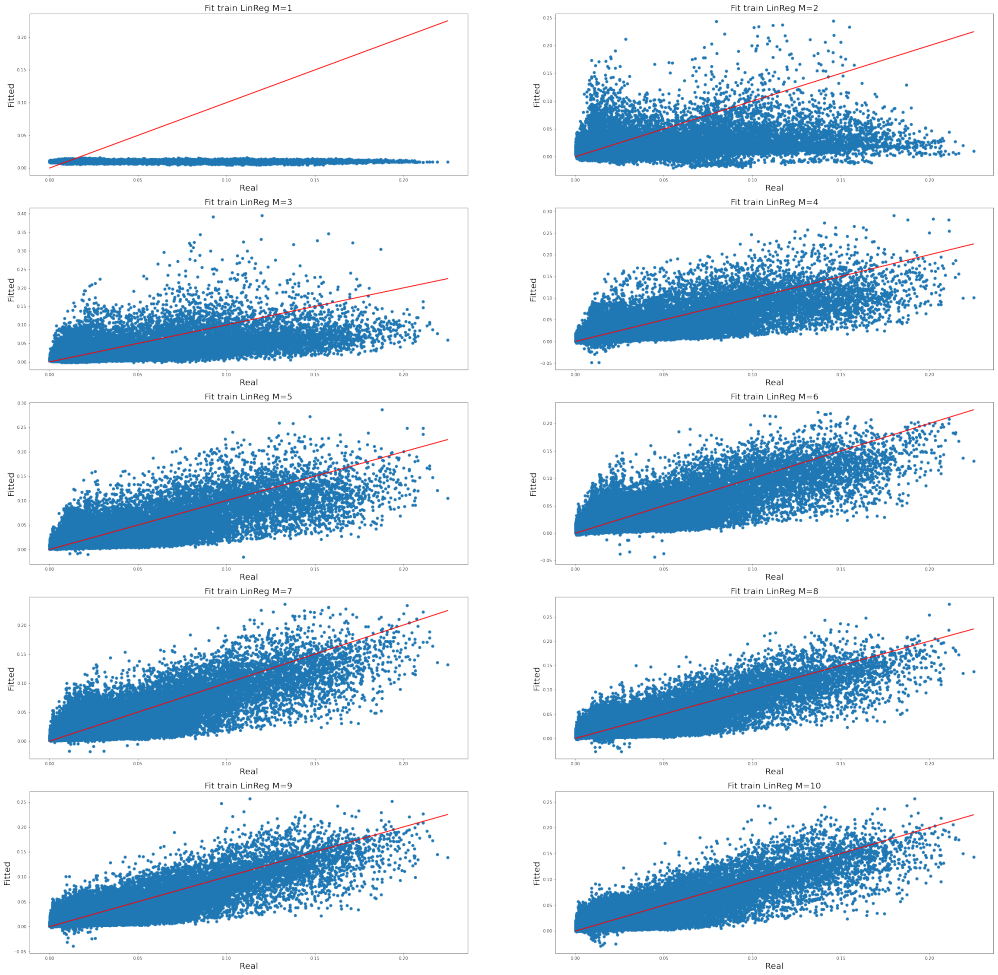


Figure .15: Linear regression train fit scatter

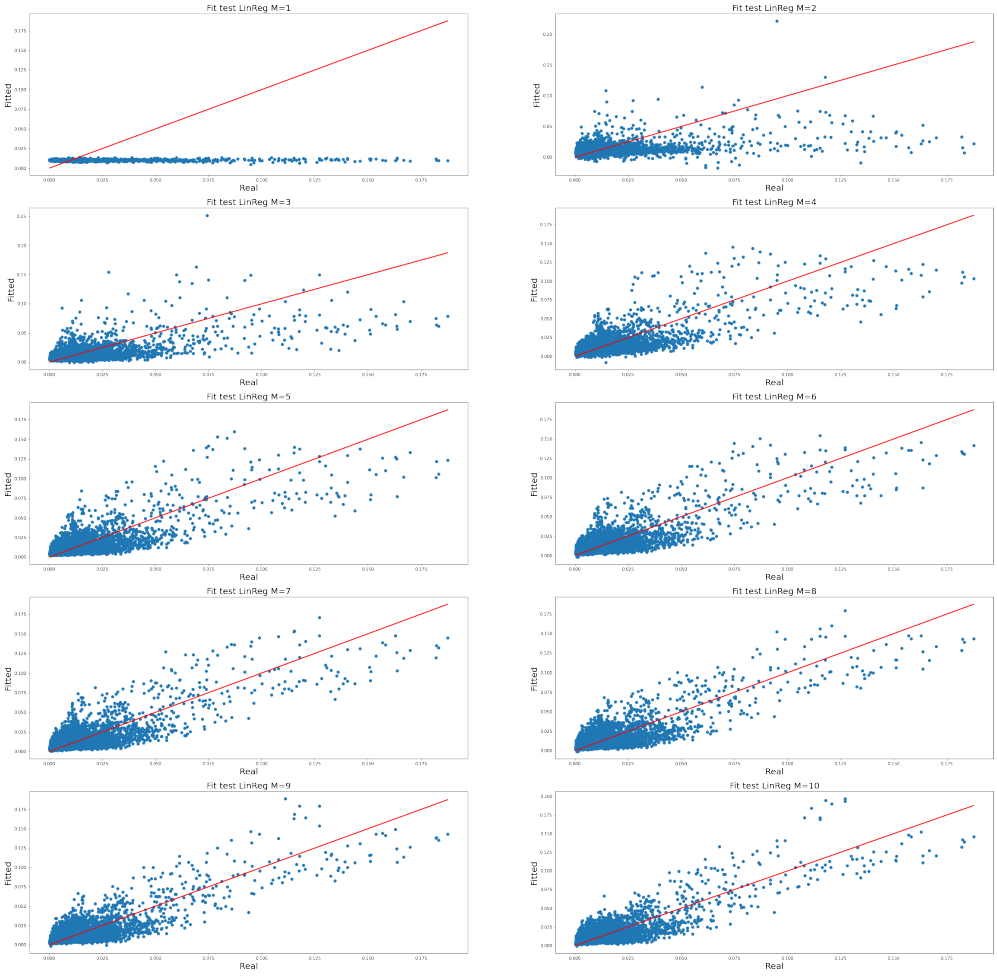


Figure .16: Linear regression test fit scatter

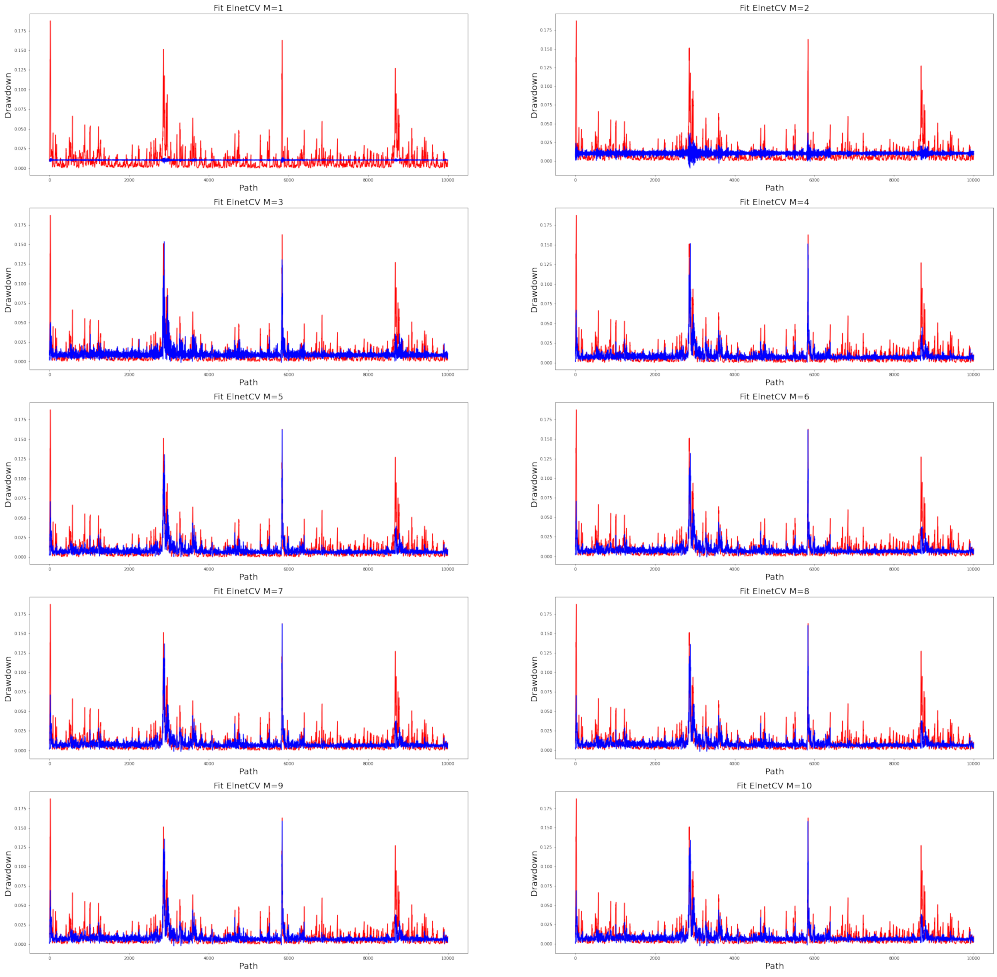


Figure .17: ElNet CV(10) fit

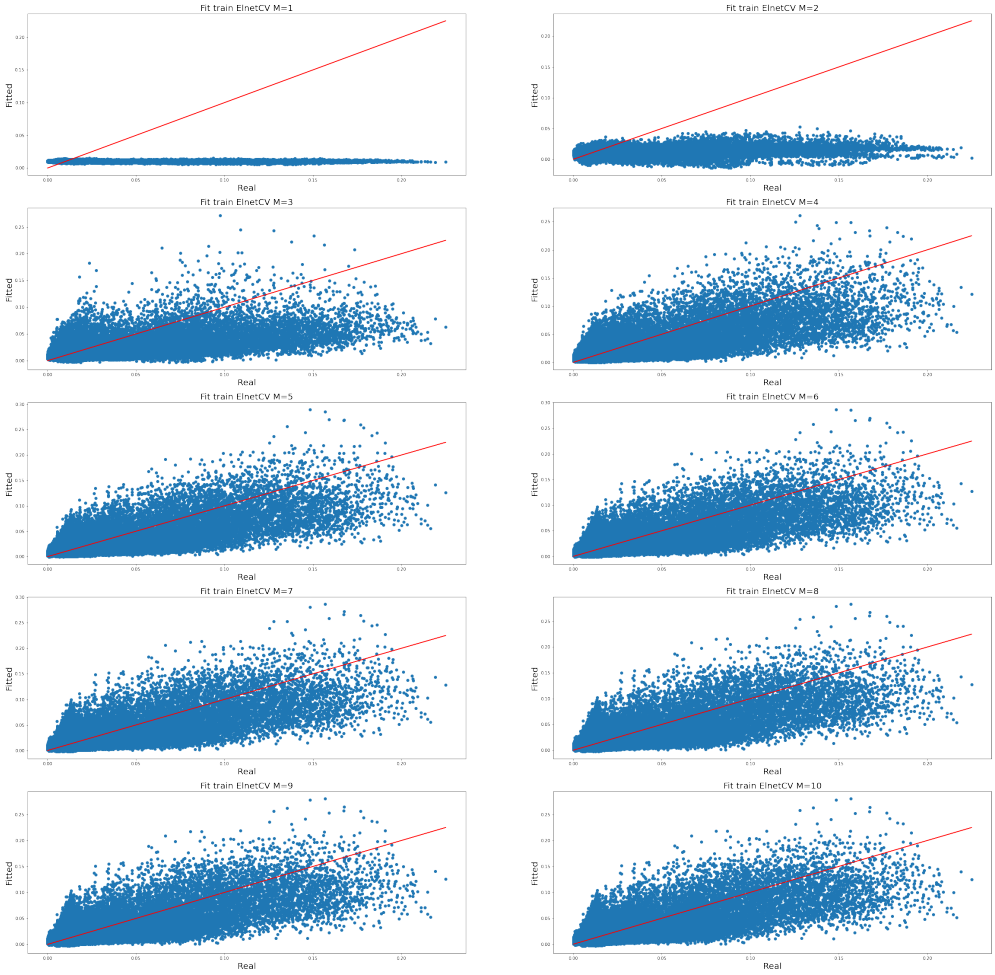


Figure .18: ElNet CV(10) train fit scatter

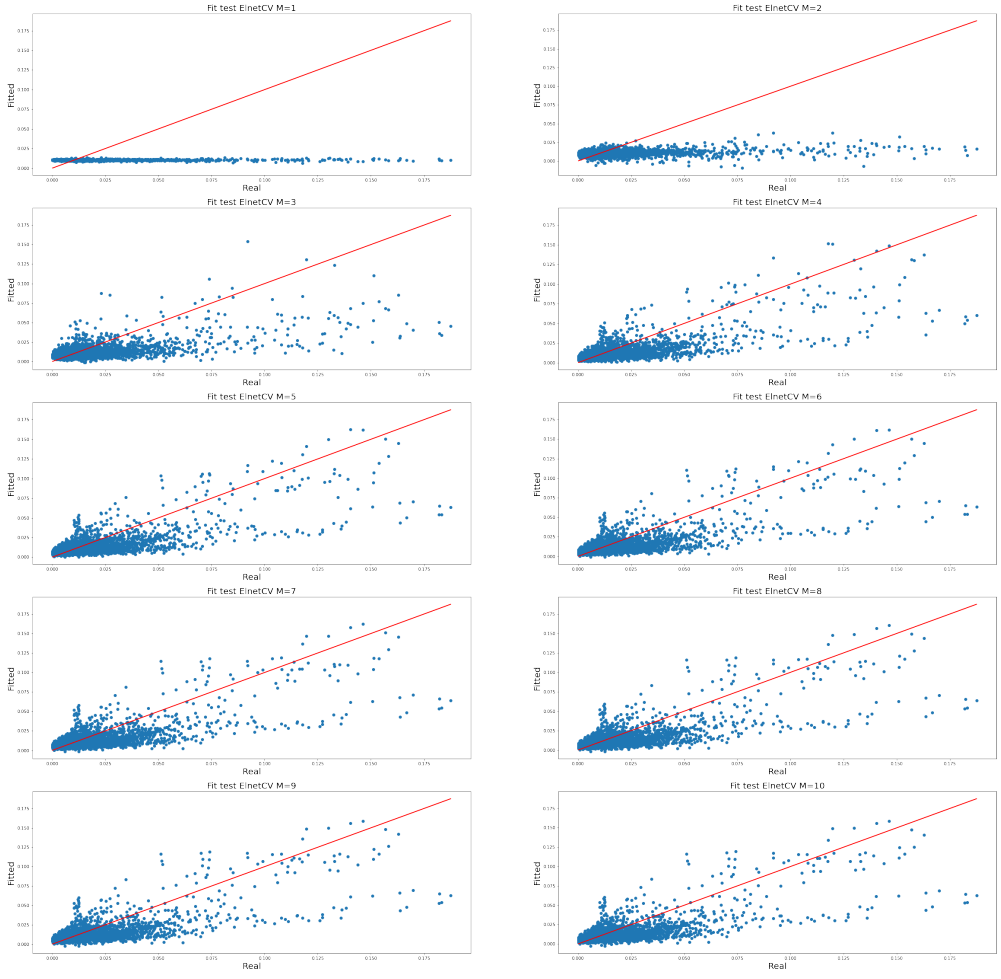


Figure .19: ElNet CV(10) test fit scatter

Appendix 3: drawdown market generator details

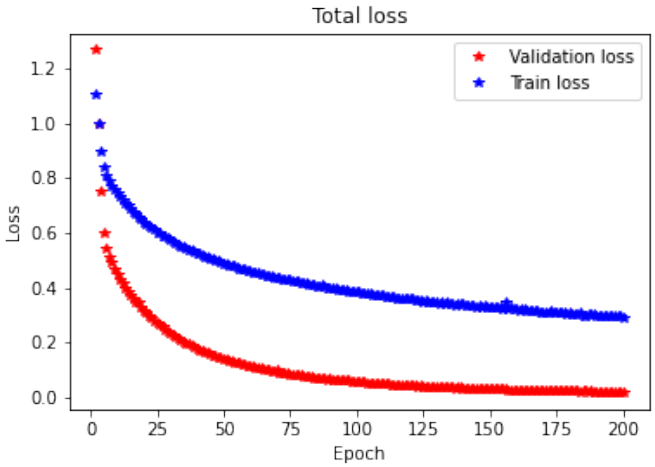


Figure .20: Drawdown market generator: total loss convergence (rescaled)

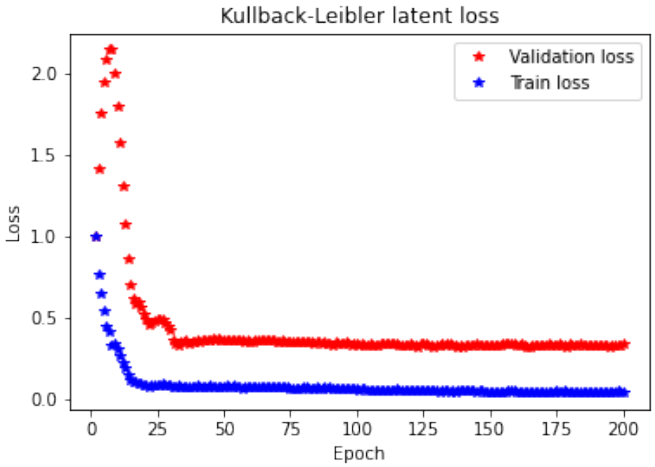


Figure .21: Drawdown market generator: latent (KL) loss convergence (rescaled)

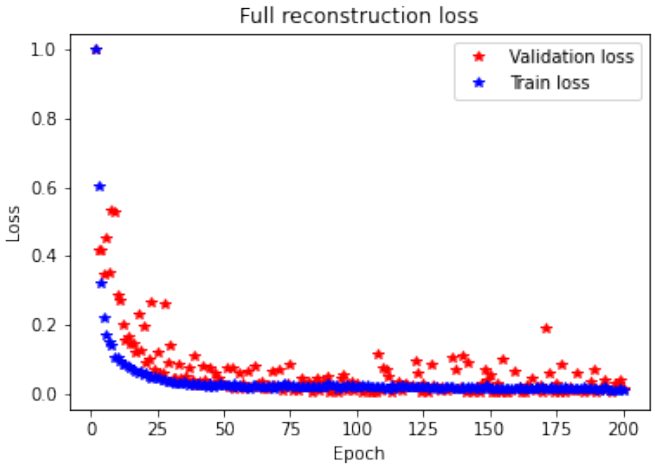


Figure .22: Drawdown market generator: total reconstruction loss convergence (rescaled)

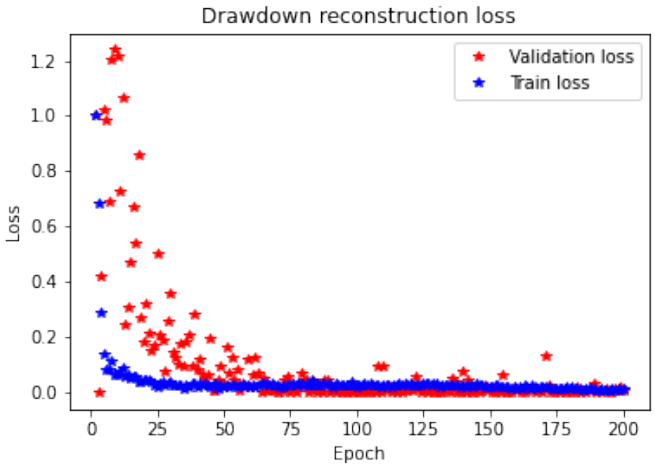


Figure .23: Drawdown market generator: drawdown reconstruction loss convergence (rescaled)

References

[1] M. Wiese, R. Knobloch, R. Korn, P. Kretschmer, Quant gans: Deep generation of financial time series, Quantitative Finance 20 (2020) 1419–1440.

[2] A. Koshiyama, N. Firoozye, P. Treleaven, Generative adversarial networks for financial trading strategies fine-tuning and combination, Quantitative Finance 21 (2021) 797–813.

[3] R. Cont, M. Cucuringu, R. Xu, C. Zhang, Tail-gan: Nonparametric scenario generation for tail risk estimation, arXiv preprint arXiv:2203.01664 (2022).

[4] M. Bergeron, N. Fung, J. Hull, Z. Poulos, A. Veneris, Variational autoencoders: A hands-off approach to volatility, The Journal of Financial Data Science 4 (2022) 125–138.

[5] H. Buehler, B. Horvath, T. Lyons, I. P. Arribas, B. Wood, A data-driven market simulator for small data environments, arXiv preprint arXiv:2006.14498 (2020).

[6] M. Vuletić, F. Prenzel, M. Cucuringu, Fin-gan: Forecasting and classifying financial time series via generative adversarial networks (2023).

[7] P. Carr, H. Zhang, O. Hadjiliadis, Maximum drawdown insurance, International Journal of Theoretical and Applied Finance 14 (2011) 1195–1230.

[8] K. Atteson, P. Carr, Carr memorial: Maximum drawdown derivatives to a hitting time, The Journal of Derivatives 30 (2022) 16–31.

[9] A. Chekhlov, S. Uryasev, M. Zabarankin, Portfolio optimization with drawdown constraints, in: Supply chain and finance, World Scientific, 2004, pp. 209–228.

[10] A. Chekhlov, S. Uryasev, M. Zabarankin, Drawdown measure in portfolio optimization, International Journal of Theoretical and Applied Finance 8 (2005) 13–58.

[11] I. Chevyrev, H. Oberhauser, Signature moments to characterize laws of stochastic processes, arXiv preprint arXiv:1810.10971 (2018).

[12] A. Kondratyev, C. Schwarz, The market generator, Available at SSRN 3384948 (2019).

[13] P. Henry-Labordere, Generative models for financial data, Available at SSRN 3408007 (2019).

[14] P. Smolensky, Information processing in dynamical systems: Foundations of harmony theory, Technical Report, Colorado Univ at Boulder Dept of Computer Science, 1986.

[15] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial networks, Communications of the ACM 63 (2020) 139–144.

[16] C. Cuchiero, W. Khosrawi, J. Teichmann, A generative adversarial network approach to calibration of local stochastic volatility models, Risks 8 (2020) 101.

[17] H. Ni, L. Szpruch, M. Wiese, S. Liao, B. Xiao, Conditional sig-wasserstein gans for time series generation, arXiv preprint arXiv:2006.05421 (2020).

[18] J. Li, X. Wang, Y. Lin, A. Sinha, M. Wellman, Generating realistic stock market order streams, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 34, pp. 727–734.

[19] V. Storchan, S. Vyetrenko, T. Balch, Mas-gan: Adversarial calibration of multi-agent market simulators. (2020).

[20] G. Benedetti, Yield curve quantization and simulation with neural networks, Available at SSRN 3577555 (2020).

[21] T. Xu, L. K. Wenliang, M. Munn, B. Acciaio, Cot-gan: Generating sequential data via causal optimal transport, arXiv preprint arXiv:2006.08571 (2020).

[22] F. D. M. Pardo, R. C. López, Mitigating overfitting on financial datasets with generative adversarial networks, The Journal of Financial Data Science 2 (2020) 76–85.

[23] H. Buehler, P. Murray, M. S. Pakkanen, B. Wood, Deep hedging: Learning to remove the drift under trading frictions with minimal equivalent near-martingale measures, arXiv preprint arXiv:2111.07844 (2021).

[24] H. Ni, L. Szpruch, M. Sabate-Vidales, B. Xiao, M. Wiese, S. Liao, Sig-wasserstein gans for time series generation, arXiv preprint arXiv:2111.01207 (2021).

[25] M. Pfenninger, S. Rikli, D. N. Bigler, Wasserstein gan: Deep generation applied on financial time series, Available at SSRN 3877960 (2021).

[26] A. Rosolia, J. Osterrieder, Analyzing deep generated financial time series for various asset classes, Available at SSRN 3898792 (2021).

[27] J. van Rhijn, C. W. Oosterlee, L. A. Grzelak, S. Liu, Monte carlo simulation of sdes using gans, arXiv preprint arXiv:2104.01437 (2021).

[28] G. Marti, V. Goubet, F. Nielsen, ccorrgan: Conditional correlation gan for learning empirical conditional distributions in the elliptope, in: International Conference on Geometric Science of Information, Springer, pp. 613–620.

[29] B. Coyle, M. Henderson, J. C. J. Le, N. Kumar, M. Paini, E. Kashefi, Quantum versus classical generative modelling in finance, Quantum Science and Technology 6 (2021) 024013.

[30] M. Wiese, B. Wood, A. Pachoud, R. Korn, H. Buehler, P. Murray, L. Bai, Multi-asset spot and option market simulation, arXiv preprint arXiv:2112.06823 (2021).

[31] E. Lezmi, J. Roche, T. Roncalli, J. Xu, Improving the robustness of trading strategy backtesting with boltzmann machines and generative adversarial networks, Available at SSRN 3645473 (2020).

[32] R. Wang, Discriminating modelling approaches for point in time economic scenario generation, arXiv preprint arXiv:2108.08818 (2021).

[33] N. J. C.-k. Fung, Variational Autoencoders for Volatility Surfaces, Ph.D. thesis, 2021.

[34] M. G. Frandsen, Greeks need not apply (2021).

[35] M. Bergeron, N. Fung, Z. Poulos, J. C. Hull, A. Veneris, Variational autoencoders: A hands-off approach to volatility, Available at SSRN 3827447 (2021).

[36] B. Ning, S. Jaimungal, X. Zhang, M. Bergeron, Arbitrage-free implied volatility surface generation with variational autoencoders, arXiv preprint arXiv:2108.04941 (2021).

[37] R. Cont, M. Vuletić, Simulation of arbitrage-free implied volatility surfaces, Available at SSRN (2022).

[38] R. Cont, Empirical properties of asset returns: stylized facts and statistical issues, Quantitative finance 1 (2001) 223.

[39] P. Lévy, Sur certains processus stochastiques homogènes, Compositio mathematica 7 (1940) 283–339.

[40] R. Douady, A. N. Shiryaev, M. Yor, On probability characteristics of" downfalls" in a standard brownian motion, Theory of Probability & Its Applications 44 (2000) 29–38.

[41] A. Rej, P. Seager, J.-P. Bouchaud, You are in a drawdown. when should you start worrying?, Wilmott 2018 (2018) 56–59.

[42] C. M. Bishop, N. M. Nasrabadi, Pattern recognition and machine learning, volume 4, Springer, 2006.

[43] T. Lyons, Rough paths, signatures and the modelling of functions on streams, arXiv preprint arXiv:1405.4537 (2014).

[44] T. J. Lyons, M. Caruana, T. Lévy, Differential equations driven by rough paths, Springer, 2007.

[45] T. Lyons, A. D. McLeod, Signature methods in machine learning, arXiv preprint arXiv:2206.14674 (2022).

[46] H. Boedihardjo, T. Lyons, D. Yang, Uniform factorial decay estimates for controlled differential equations, Electronic Communications in Probability 20 (2015) 1–11.

[47] I. Chevyrev, A. Kormilitzin, A primer on the signature method in machine learning, arXiv preprint arXiv:1603.03788 (2016).

[48] M. Lemercier, C. Salvi, T. Damoulas, E. Bonilla, T. Lyons, Distribution regression for sequential data, in: International Conference on Artificial Intelligence and Statistics, PMLR, pp. 3754–3762.

[49] N. E. Cotter, The stone-weierstrass theorem and its application to neural networks, IEEE transactions on neural networks 1 (1990) 290–295.

[50] K. Hornik, Approximation capabilities of multilayer feedforward networks, Neural networks 4 (1991) 251–257.

[51] D. Levin, T. Lyons, H. Ni, Learning from the past, predicting the statistics for the future, learning an evolving system, arXiv preprint arXiv:1309.0260 (2013).

[52] T. Hastie, R. Tibshirani, J. H. Friedman, J. H. Friedman, The elements of statistical learning: data mining, inference, and prediction, volume 2, Springer, 2009.

[53] K. Boudt, B. G. Peterson, C. Croux, Estimation and decomposition of downside risk for portfolios with non-normal returns, Journal of Risk 11 (2008) 79–103.

[54] P. Kidger, T. Lyons, Signatory: differentiable computations of the signature and logsignature transforms, on both cpu and gpu, arXiv preprint arXiv:2001.00706 (2020).

[55] D. P. Kingma, M. Welling, Stochastic gradient vb and the variational auto-encoder, in: Second International Conference on Learning Representations, ICLR, volume 19, p. 121.

[56] F. Eckerli, J. Osterrieder, Generative adversarial networks in finance: an overview, arXiv preprint arXiv:2106.06364 (2021).