A Neural RDE approach for continuous-time non-Markovian stochastic control problems

Melker Höglund [1] **Emilio Ferrucci** [2] **Camilo Hernández** [3] **Aitor Muguruza Gonzalez** [1 4] **Cristopher Salvi** [5 6] **Leandro Sánchez-Betancourt** [7] **Yufei Zhang** [8]

Abstract

We propose a novel framework for solving continuous-time non-Markovian stochastic control problems by means of *neural rough differential equations* (Neural RDEs) introduced in Morrill et al. (2021). Non-Markovianity naturally arises in control problems due to the time delay effects in the system coefficients or the driving noises, which leads to optimal control strategies depending explicitly on the historical trajectories of the system state. By modelling the control process as the solution of a Neural RDE driven by the state process, we show that the control-state joint dynamics are governed by an *uncontrolled, augmented* Neural RDE, allowing for fast Monte-Carlo estimation of the value function via trajectories simulation and memory-efficient back-propagation. We provide theoretical underpinnings for the proposed algorithmic framework by demonstrating that Neural RDEs serve as universal approximators for functions of random rough paths. Exhaustive numerical experiments on non-Markovian stochastic control problems are presented, which reveal that the proposed framework is time-resolution-invariant and achieves higher accuracy and better stability in irregular sampling compared to existing RNN-based approaches.

[*]Equal contribution [1]Department of Mathematics, Imperial College London. [2]Mathematical Institute, University of Oxford. [3]ORFE Department, Princeton University. [4]Kaiju Capital Management. [5]Department of Mathematics, I-X, Imperial College London. [6]The Alan Turing Institute. [7]Department of Mathematics, King's College London [8]Department of Statistics, London School of Economics.. Correspondence to: Cristopher Salvi <c.salvi@imperial.ac.uk>.

Workshop on New Frontiers in Learning, Control, and Dynamical Systems at the International Conference on Machine Learning (ICML), Honolulu, Hawaii, USA, 2023. Copyright 2023 by the author(s).

1. Introduction

The field of stochastic control is concerned with problems where an agent interacts over time with some random environment through the action of a *control*. In this setting, the agent seeks to select the control such that some objective depending on the trajectory of the system under their control and the choice of the control itself is optimised; commonly, as the system is stochastic, such an objective takes the form of an expectation of some pathwise cost or reward. The study of this class of problems has been successfully applied to many fields of modern sciences, including biology Cucker & Smale (2007), economics Kamien & Schwartz (2012), engineering Grundel et al. (2007), finance Pham (2009), and more recently, epidemics control Hubert et al. (2022).

Stochastic control is nowadays regarded as a well-established field of mathematics. Two main approaches govern the analysis: the stochastic maximum principle and the dynamic programming approach, see Yong & Zhou (1999); Pham (2009). In either case, an agent is interested in characterising a set of optimal strategies, the dynamics of the system under such strategies, and the optimal value of the corresponding reward functional. The two main sources of complexity for tackling these problems are: 1) the continuous-time nature of the underlying stochastic dynamics, and 2) the presence of memory yielding a non-negligible impact of the system's history on its future evolution.

Continuous-time non-Markovian stochastic control problems, where the evolution of the system depends on its history and not only on its current state, have received an increasing amount of attention in recent years. Non-Markovian system provides a more faithful class of models to describe real-world phenomena than their Markov counterparts, where the (infinitesimal) displacement of the state dynamics depend only on the current state. Non-Markovianity naturally arises in control problems due to the time delay effects in the system coefficients or the driving noises, which leads to the optimal control strategy being influenced by the historical trajectories of the system states.

Typical examples of continuous-time non-Markovian

stochastic control problems include rough volatility models Gatheral et al. (2018) from quantitative finance in which the non-Markovianity stems from having a fractional Brownian motion as the driving noise. Fractional Brownian motion generalises Brownian motion and involves history-dependent increments. As a result, the state dynamics driven by fractional Brownian motion exhibit non-Markovian behavior. Another example of non-Markovian problems are delayed control problems, where memory is incorporated into the system by assuming path-dependence of the vector fields governing the dynamics (see Sec. 3 for a precise statement). Optimal decision with time delay is ubiquitous in economics, for example in the study of growth models with delayed production or pension funds models, Kydland & Prescott (1982); Salvatore (2011), in marketing for models of optimal advertising with distributed lag effects Gozzi et al. (2009), and in finance for portfolio selection under the market with memory and delayed responses Øksendal et al. (2011). See also Kolmanovskiı & Shaıkhet (1996) for modelling systems with after-effect in mechanics, engineering, biology, and medicine.

As the solution to a continuous-time non-Markovian stochastic control problem is in general not known analytically, it is important to construct effective and robust numerical schemes for solving these control problems. An essential numerical challenge is to effectively capture the nonlinear dependence of the optimal control strategy on the historical trajectories of the system states.

Contributions Using the modern tool set offered by neural rough differential equations (Neural RDEs) Morrill et al. (2021) — a continuous-time analogue to recurrent neural networks (RNNs) — we propose a novel framework which, to the best of our knowledge, is the first numerical approach allowing to solve non-Markovian stochastic control problems in continuous-time. More precisely, we parameterise the control process as the solution of a Neural RDE driven by the state process, and show that the control-state joint dynamics are governed by an uncontrolled RDE with vector fields parameterised by neural networks. We demonstrate how this formulation allows for trajectories sampling, Monte-Carlo estimation of the reward functional and backpropagation. To deal with sample paths of infinite 1-variation, which is necessary in stochastic control, we also extend the universal approximation result in Kidger et al. (2020) to a probabilistic density result for Neural RDEs driven by random rough paths. The interpretation is that we are able to approximate continuous feed-back controls arbitrarily well in probability. Through various experiments, we demonstrate that the proposed framework is time-resolution-invariant and capable of learning optimal controls with higher accuracy than traditional RNN-based approaches.

The rest of the paper is organised as follows: in Sec. 2 we discuss some related work, in Sec. 3 we present our algorithmic framework and the universal approximation result of Neural RDEs, and in Sec. 4 we demonstrate the effectiveness of the algorithm through numerical experiments.

2. Related work

Over the last decade, a large volume of research has been conducted to solve Markovian stochastic control problems numerically using neural networks, either by directly parameterising the control and then sampling from the state process, such as done by Han et al. (2016), or by solving the partial differential equations (PDEs) or backward stochastic differential equations (BSDEs) associated with the problem; see Germain et al. (2021) for a recent survey about neural networks-based algorithms for stochastic control and PDEs. We also mention two examples from the growing literature. The Deep BSDE model from Han et al. (2017), where the authors propose an algorithm to solve parabolic PDEs and BSDEs in high dimension and think of the gradient of the solution as the policy function, approximated with a neural network. The Deep Galerkin model Sirignano & Spiliopoulos (2018) is a mesh-free algorithm to solve PDEs associated with the value function of control problems; the authors approximate the solution with a deep neural network which is trained to satisfy the PDE differential operator, initial condition, and boundary conditions.

Recently, signatures methods Lyons (2014); Kidger et al. (2019) have been employed for solving both Markovian and non-Markovian control problems in simplified settings Kalsi et al. (2020); Cartea et al. (2022). This approach does not rely on a model underpinning the dynamics of the unaffected processes and has shown excellent results when solving a number of algorithmic trading problems. However, this method has two main drawbacks: (i) it suffers from the curse of dimensionality — this happens when one wishes to compute signatures of a high-dimensional (more than five) process to make online decisions, and (ii) it requires that the flow of information observed by the controller is unaffected by the control and everything else the controller observes can be explicitly constructed from such information and the policy. We also point out the theoretical contribution by Diehl et al. (2017) studying control problems where the driving noise is a random rough path.

The approach of directly parameterising the control and training by sampling trajectories from the system was recently studied in the setting of delay-type non-Markovian stochastic control by Han & Hu (2021). Specifically, the control is taken to be a Long Short-Term Memory (LSTM) recurrent neural network with the discrete simulated values of the state process as input, so as to capture the path-dependence of the problem. The method is shown to out-

perform a baseline parameterisation using a fully-connected feed-forward network taking as input a segment of the history of the sample path, and demonstrated to have theoretical advantages in handling non-Markovian problems.

Neural RDEs, as popularised by Kidger et al. (2020); Morrill et al. (2021) provide an elegant way of modelling temporal dynamics by parameterising the vector fields of some classes of differential equations by neural networks. The input to such models is a multivariate time series interpolated into a continuous path X. Depending on the level of (ir)regularity of X, the corresponding equation can be solved in different ways. In (Kidger et al., 2020), X is assumed differentiable almost everywhere, and the equation becomes an ordinary differential equation (ODE) that can be evaluated numerically via a call to an ODE solver of choice. More generally, if X is of bounded variation, then the Neural RDE can be solved using classical Riemann–Stieltjes or Young integration (Young, 1905).

Of particular interest in the field of stochastic control is the setting where the driving noise is Brownian motion , and the resulting dynamical systems are typically referred to as *stochastic differential equations* (SDEs). Because sample paths from Brownian motion are not of bounded variation, the integral cannot be interpreted in the classical sense, but rather using the framework of stochastic integration (Itô, Stratonovich, etc.). The corresponding "neural" version of such models has been the object of several studies Liu et al. (2019); Li et al. (2020); Kidger et al. (2021b;a), in particular in the context of generative modelling for time series. Rough integration Lyons (1998) is arguably the most general type of integration theory accommodating driving signals X of arbitrary roughness, and in particular non-Markovian processes such as fractional Brownian motion. In this paper, we position ourselves in this general setting. In the appendix, we provide a minimal summary of the basic notions of this theory underpinning the content of this paper.

3. Problem formulation and methodology

3.1. Control problems with path-dependent coefficients

Let us introduce the non-Markovian control problems over closed-loop controls. We fix $d, d_a, d_W \in \mathbb{N}$, a real number $T > 0$ and $\mathcal{C}^d := \mathcal{C}([0,T];\mathbb{R}^d)$, the space of continuous paths from $[0,T]$ to $\mathbb{R}^d$ endowed with the sup norm. Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space supporting a d_W-dimensional Brownian motion $W = (W_t)_{t\in[0,T]}$, and $\mathbb{F}$ be the natural filtration of W augmented with the $\mathbb{P}$-null sets. Let $\mathcal{H}^2(\mathbb{R}^{d_a})$ be the space of all square integrable $\mathbb{F}$-progressively measurable processes, and for each $\alpha \in \mathcal{H}^2(\mathbb{R}^{d_a})$, consider the following controlled state dynamics: for all $t \in [0,T]$,

$$\mathrm{d}X_t = \mu(t, X_{\cdot\wedge t}, \alpha_t)\mathrm{d}t + \sigma(t, X_{\cdot\wedge t}, \alpha_t)\mathrm{d}W_t, \quad (1)$$

where $X_0 = x_0$, $X_{\cdot\wedge t} = \{X_s\}_{s\in[0,t]}$, and $(\mu,\sigma) : [0,T] \times \mathcal{C}^d \times \mathbb{R}^{d_a} \longrightarrow \mathbb{R}^d \times \mathbb{R}^{d\times d_W}$ are non-anticipative and sufficiently regular mappings so that equation (1) admits a unique solution X in $\mathcal{H}^2(\mathbb{R}^d)$.[1] We denote by $\mathcal{A}$ the set of admissible controls containing all $\alpha \in \mathcal{H}^2(\mathbb{R}^{d_a})$ that are adapted to the filtration generated by X. Such controls are often referred to as closed-loop, or feedback, controls.

The agent's goal is to minimise the following cost functional

$$J(x_0, \alpha) = \mathbb{E}\left[\int_0^T f(t, X_{\cdot\wedge t}, \alpha_t)\mathrm{d}t + g(X_{\cdot\wedge T})\right] \quad (2)$$

over all controls $\alpha \in \mathcal{A}$, where $f : [0,T] \times \mathcal{C}^d \times \mathbb{R}^{d_a} \to \mathbb{R}$ and $g : \mathcal{C}^d \to \mathbb{R}$ are given measurable functions. Note that equation (2) is a non-Markovian control problem, as the coefficients of the state dynamics and the cost functions depend on the history of the system state. Hence, the optimal control process will also depend on the entire state trajectory, instead of the current system state.

3.2. Policy parametrisation via Neural RDE

Here, we are going to parameterise the control process α in equation (1) as the solution of a Neural RDE driven by the state process X. Let $\ell_\theta : \mathbb{R}^{d_a} \to \mathbb{R}^{d_h}, h_\theta : \mathbb{R}^{d_h} \to \mathbb{R}^{d_h\times d}, A_\theta \in \mathbb{R}^{d\times d_h}$ be (Lipschitz) neural networks. Collectively, they are parameterised by θ. The dimension $d_h > 0$ is a hyperparameter describing the size of the hidden state.

We parameterise controls $\alpha^\theta \in \mathcal{A}$ as solutions to Neural RDEs driven by X,

$$Y_0 = \ell_\theta(x_0), \quad \mathrm{d}Y_t = h_\theta(Y_t)\mathrm{d}X_t, \quad \alpha_t^\theta = A_\theta Y_t. \quad (3)$$

With this choice of parameterisation, the dynamics of the joint process (X, Y) are governed by the following *uncontrolled* RDE with structured vector fields

$$\begin{aligned}\mathrm{d}\begin{pmatrix}X\\Y\end{pmatrix}_t &= \mu\left(t, X_{\cdot\wedge t}, A_\theta Y_t\right)\begin{pmatrix}1\\h_\theta(Y_t)\end{pmatrix}\mathrm{d}t \\ &\quad + \sigma\left(t, X_{\cdot\wedge t}, A_\theta Y_t\right)\begin{pmatrix}I_d & 0\\0 & h_\theta(Y_t)\end{pmatrix}\mathrm{d}\begin{pmatrix}W\\W\end{pmatrix}_t\end{aligned} \quad (4)$$

Thus, the infinite dimensional minimisation over admissible controls of the reward functional J in equation (2) can be replaced with the finite-dimensional minimisation over the parameters θ of the following objective functional

$$J(x_0, \alpha^\theta) = \mathbb{E}\left[\int_0^T f(t, X_{\cdot\wedge t}, A_\theta Y_t)\mathrm{d}t + g(X_{\cdot\wedge T})\right], \quad (5)$$

[1] This is the case if, for instance, $\mathcal{C}^d \times A \ni (x,a) \longmapsto \varphi(t,x,a)$ has linear growth and is Lipschitz continuous uniformly in t for $\varphi = \mu, \sigma$, see Protter (2005).

Here, we perform this minimisation by first solving numerically the uncontrolled Neural RDE (4) using a classical Euler-Maruyama scheme[2]; we then use the obtained sample trajectories to compute a Monte-Carlo estimate of the objective functional in (5), where the integral is approximated using classical quadrature; finally we compute gradients of the estimated objective functional with respect to model parameters θ and optimise by (stochastic) gradient descent.

Contrary to the approach taken by Han & Hu (2021) using an LSTM-parameterisation of the control, our formulation does not rely on any specific discretisation or choice of numerical method. A key feature of Neural RDEs is their robustness to irregular sampling of the data, essentially due to operating continuously in time. The sampled data enters the model only through the construction of the interpolated path, after which the RDE can be solved numerically on any desired grid using adaptive schemes that changes the step size to appropriately resolve the variations in the path. Therefore, because our scheme can be formulated completely in continuous-time and independently of whichever way one chooses to estimate $J(\alpha)$, it is naturally *time-resolution invariant*, in the sense that even if trained on a coarser resolution it can be directly evaluated on a finer resolution without retraining.

3.3. Extension to problems with non-Markovian noises

The algorithm proposed in Section 3.2 can be easily applied to other types of non-Markovian control problems. For instance, consider minimising the following cost functional:

$$J(x_0, \alpha) = \mathbb{E}\left[\int_0^T f(t, X_t, \alpha_t)\mathrm{d}t + g(X_T)\right], \quad (6)$$

subject to the following state dynamics (cf. equation (1)): for all $t \in [0, T]$,

$$\mathrm{d}X_t = \mu(t, X_t, \alpha_t)\mathrm{d}t + \sigma(t, X_t, \alpha_t)\mathrm{d}W_t^H, \quad (7)$$

where $(\mu, \sigma) : [0, T] \times \mathbb{R}^d \times \mathbb{R}^{d_\alpha} \longrightarrow \mathbb{R}^d \times \mathbb{R}^{d\times d_W}$ are given functions, and W^H is a d_W-dimensional fractional Brownian motion with Hurst index $H \in (0, 1)$ defined by

$$W_t^H = \frac{1}{\Gamma(H + \frac{1}{2})}\int_0^t (t - s)^{H-\frac{1}{2}} dW_s,$$

with $\Gamma(\cdot)$ being the Gamma function.

Note that the distribution of the noise increment $W_{t+\delta}^H - W_t^H$, $\delta > 0$, depends on the trajectory $(W_s^H)_{s\le t}$, and hence the state process X is non-Markovian, even if all coefficients of equation (6) and equation (7) depend only on the current state and control variables. As a result, the optimal feedback control of equation (7) typically depend on the entire history of the state process (see e.g., (Duncan & Pasik-Duncan, 2010)).

However, the approach in Sec. 3.2 can be naturally extended to this setting, as the algorithm directly parameterises the feedback controls and hence is invariant with respect to different driving noises. In contrast, as in the case of path-dependent coefficients, the classical methods for Markovian control problems will result in sub-optimal policies in the presence of fractional noises.

3.4. Universality of Neural RDEs

The section provides theoretical underpinnings for the proposed algorithmic framework by demonstrating that Neural RDEs serve as universal approximators for functions of random rough paths.

More precisely, we prove the density (in probability) of linear functionals on the signature of rough paths, which implies the universality of Neural RDEs (see Theorem 3.2). We refer the reader to subsection A.1 for basic backgrounds of rough path theory, which will be used in this section. Let $\alpha \in (0, 1]$ and $\boldsymbol{X} : \Omega \times [0, T] \to T^{\lfloor 1/\alpha \rfloor}(\mathbb{R}^{1+d})$ be a stochastic α-Hölder rough path with the property that the zero-th component of its trace is the time coordinate, $X_t^0 = t$, and whose higher components that involve the zero-th are defined canonically through Stieltjes integration. In practice, $\boldsymbol{X}$ can represent either the state trajectory, or the underlying driving (fractional) Brownian noise.

The following Theorem 3.1 is the probabilistic analogue of a well-known universal approximation property of the deterministic signature, see for instance Proposition 3 in Fermanian (2021a).[3] The main contribution of Theorem 3.1, compared to its deterministic counterpart (e.g. see Kidger et al. (2020)), is that it avoids imposing a compactness assumption and establishes the universality in probability. As a result, the application of Theorem 3.1 is simplified, as it eliminates the need to select a compact set in the space of α-Hölder continuous paths, which typically entails additional boundedness and smoothness conditions of the paths.

Theorem 3.1. *Let $\beta < \alpha$, $\boldsymbol{X}$ as above, and $F : C^\beta([0, T], \mathbb{R}^{1+d}) \to \mathbb{R}$ be a continuous map. Then for each $\varepsilon, \delta > 0$ there exists a truncation level N and a linear map $\ell \in T^N(\mathbb{R}^{1+d})$ such that*

$$\mathbb{P}\big[|F(\boldsymbol{X}) - \langle \ell, S^N(\boldsymbol{X})_{0T}\rangle| \ge \varepsilon\big] < \delta, \quad (8)$$

where $S^N(\boldsymbol{X})$ refers to the truncated signature of $\boldsymbol{X}$ (cf. subsection A.1).

[2]For convergence guarantees of Euler-Maruyama schemes applied to SDEs with path-dependent vector fields we refer the reader to Mao (2003). Other choices of solvers are possible.

[3]It is not to be confused with the universality property of the expected signature with respect to functions of distributions on paths, see Theorem 3.2 in Lemercier et al. (2021).

Proof. Let D_r^α be the closed disk centred at the 0 rough path, of radius $r > 0$ in $C^\alpha([0,T],\mathbb{R}^{1+d})$. Since $\boldsymbol{X}$ is a.s. α-Hölder continuous in the rough path sense

$$\begin{aligned}\lim_{r\to\infty} \mathbb{P}[\boldsymbol{X} \in D_r^\alpha] &= \mathbb{P}\Big[\bigcup_{r>0} \boldsymbol{X}^{-1}(D_r^\alpha)\Big] \\ &= \mathbb{P}\Big[\boldsymbol{X}^{-1}(\bigcup_{r>0} D_r^\alpha)\Big] \\ &= \mathbb{P}[\boldsymbol{X} \in C^\alpha([0,T],\mathbb{R}^{1+d})] \\ &= 1\end{aligned}$$

and thus given δ as in the statement there exists r s.t. $\mathbb{P}[\boldsymbol{X} \in D_r^\alpha] > 1-\delta$. By Proposition 8.17 (ii) in Friz & Victoir (2010), for any $\beta < \alpha$, D_r^α is sequentially compact in $C^\beta([0,T],\mathbb{R}^{1+d})$, and thus compact since this is a metric space. Let $\widetilde{D}_r^\alpha$ be the intersection of D_r^α with the aforementioned set of rough paths in $C^\alpha([0,T],\mathbb{R}^{1+d})$ whose zero-th coordinate is time t; this is a closed set and thus $\widetilde{D}_r^\alpha$ is still compact. Thanks to the inclusion of the time coordinate, linear functions on the signature separate points in $\widetilde{D}_r^\alpha$, and by the Stone-Weierstrass theorem applied to $F|_{\widetilde{D}_r^\alpha}$ there exist N and ℓ as in the statement s.t. $|F(\boldsymbol{X}(\omega)) - \langle \ell, S^N(\boldsymbol{X}(\omega))_{0T}\rangle| < \varepsilon$ for $\omega \in \Omega$ s.t. $\boldsymbol{X}(\omega) \in D_r^\alpha$, and the conclusion now follows. □

The choice for F that we have in mind is the solution map of an SDE, which can be equivalently viewed as an RDE driven by the enhanced (fractional) Brownian rough path. The reduction of the Hölder exponent from α to β is due to a technical step in the proof, but this adjustment does not impact the validity of applying the theorem with this specific choice of F.

Remark 3.2 (Universal approximations of Neural RDEs). It follows immediately from Theorem 3.1 and the fact that the signature of a stochastic process satisfies a linear SDE (see equation (13)) that Neural RDEs parametrised by feedforward neural networks with linear activation functions are dense in probability (in the sense of equation (8)) in the space of all continuous functions on driving rough paths. This indicates that smooth (and indeed linear) functions of solutions to Neural RDEs can well-approximate the optimal controls of a non-Markovian control problem, as the optimal strategies typically depend continuously on the driven noise. In practice, one can expect superior performance (e.g. lower dimensions involved) when using non-linear activations, even though it is not needed for the theoretical result. This is because the non-linearity is already contained in operation of solving the SDE.

Remark 3.3. We would like to thank Terry Lyons for pointing out a shortcut to the proof of Theorem 3.1, valid in the case of Brownian rough paths, which goes as follows: the Stratonovich rough path lift $X \mapsto \boldsymbol{X}$ is measurable, and thus by Lusin's theorem, for any $\delta > 0$ continuous on a compact set K of probability $1-\delta$. We may then apply the classical Stone-Weierstrass theorem for signatures of rough paths with trace valued in K to obtain the result.

4. Experiments

We present a number of numerical experiments demonstrating the capabilities of our method to compute approximate solutions of non-Markovian stochastic control problems in continuous-time. We benchmark the performance of our approach against a selection of alternative RNN-based models parameterising the feed-back control Han & Hu (2021). This choice of benchmarks is motivated mainly by the fact that this class of models, due to the connection between neural RDEs and RNNs, are the closest currently available alternatives to our method. The three alternative architectures we consider are: 1) RNN, 2) Long Short-Term Memory (LSTM), and 3) a Gated Recurrent Unit (GRU).

One key feature of the proposed model that we wish to study empirically is the time-resolution invariance discussed at the end of Sec. 3.2. Concretely, to test their robustness to changes in time-resolution, we train all models on a coarser time grid and then we evaluate them on a finer grid. Such a property is desirable both from the perspective of efficient training under computational budget constraints, and as an indication that the model is in fact learning a solution to the actual continuous-time problem. Lastly, a learned control that is heavily dependent on the grid that it was trained on may not be suitable for practical use; in such a case, this invariance property is critical.

Another performance criterion we will be using is the pathwise L^2 error between the state trajectories obtained using the NCDE control strategy and the ones obtained using the theoretical optimal control. The lower this error, the closer the trajectories sampled from the learnt state process to the trajectories of the theoretical state process.

All models are trained by sampling batches of trajectories from the state process under the parametric control, of the form given by equation (4), and then performing direct backpropagation using an Adam optimiser (Kingma & Ba, 2014) to minimise the Monte-Carlo estimate of the value of the reward functional $J(\alpha^\theta)$ in equation (5). For each experiment, the grid on which the system is simulated and the number of sample trajectories used to train and evaluate all models are kept the same. For a fair comparison, the hyperparameters of each model are adjusted such that the models all have an approximately equal number of trainable parameters. All experiments are implemented using version 1.11.0 of PyTorch and run on an NVIDIA Tesla K80 GPU. Additional experimental details can be found in the appendix.

4.1. Stochastic control problem with delay

We consider first the example of a linear-quadratic problem with delay, also used by Han & Hu (2021). Linear-quadratic control problems are widely used to address real-world challenges. An example from the mathematical finance community is the control of intraday fill ratios when volatility is stochastic; see Cartea & Sánchez-Betancourt (2021). In their paper, the control affects the state dynamics linearly and the performance criterion is composed of a square running penalty on the control and a square running penalty on one of the entries of the state process. A detailed discussion of this problem and how an explicit solution may be obtained is given by Bauer & Rieder (2005).

With notation as before, the dynamics of the state process X under a control α are given by

$$\mathrm{d}X_t = (A_1 X_t + A_2 Y_t + A_3 X_{t-\delta} + B\alpha_t)\mathrm{d}t + \sigma \mathrm{d}W_t, \tag{9}$$

for $t \in [0, T]$ with $X_t = \phi$ for $t \in [-\delta, 0]$, $\delta > 0$ a delay parameter, the distributed delay satisfying

$$Y_t := \int_{-\delta}^{0} e^{\lambda\xi} h(X_{t+\xi})\mathrm{d}\xi, \quad t \in [0, T]$$

and the goal functional that we seek to minimise

$$J(\alpha) = \mathbb{E}\left[\int_0^T \left(Z_t^\top Q Z_t + \alpha_t^\top R \alpha_t\right)\mathrm{d}t + Z_T^\top G Z_T\right],$$
$$Z_t := (X_t + e^{\lambda\delta} A_3 Y_t),\ t \in [0, T].$$

The parameters $A_1, A_2, A_3 \in \mathbb{R}^{d\times d}$, $B \in \mathbb{R}^{d\times d_\alpha}$, $\sigma \in \mathbb{R}^{d\times d_W}$, $Q, G \in \mathbb{R}^{d\times d}$,$R \in \mathbb{R}^{d\times d}$,$\lambda, \delta$ and T are all taken to be the same values as those used by Han & Hu (2021). In particular, the problem is considered in 10 dimensions in state, noise and control, Q, R, G are proportional to identity matrices, the elements of A_1, A_3, B and σ are selected randomly and A_2 is determined by a condition guaranteeing an explicit solution. We refer to Han & Hu (2021) for further details. The constant initial condition ϕ is taken to be zero. The explicit value function and optimal control are obtained in terms of the solution to an associated Riccati equation, which can be solved numerically.

The results for this experiment are shown in table 1. We see that, trained at full resolution, the LSTM, GRU and Neural RDE models all perform approximately as well. However, when the training grid is made coarser, the Neural RDE model remains relatively stable with only slight increases in error, dramatically outperforming the benchmark models whose performance rapidly deteriorates.

4.2. Stochastic control problem driven by Fractional Brownian Motion

Next, we demonstrate the application of the proposed method to a problem with non-Markovianity stemming from correlated noise increments by considering a linear-quadratic problem driven by fractional Brownian motion. The dynamics for the state X are given by

$$\mathrm{d}X_t = (AX_t + C\alpha_t)\mathrm{d}t + \sigma \mathrm{d}W_t^H, \quad t \in [0, T], \qquad X_0 = 0 \tag{10}$$

where $A \in \mathbb{R}^{d\times d}$, $C \in \mathbb{R}^{d\times d_\alpha}$, $\sigma \in \mathbb{R}^{d\times d_W}$ are parameters and W^H is a d_W-dimensional fractional Brownian motion with components with Hurst parameters $H \in (0, 1)$ (assumed the same across all d_W channels). We choose the Hurst parameter $H = 0.3$, so as to highlight the applicability of the method also in the case where solution paths are rougher than Brownian motion ($H = 0.5$).

The quadratic cost functional is as follows

$$J(\alpha) = \frac{1}{2}\mathbb{E}\left[\int_0^T \left(X_s^\top Q X_s + \alpha_s^\top R \alpha_s\right)\mathrm{d}s + X_T^\top G X_T\right], \tag{11}$$

where Q, R, G are symmetric and positive definite. We consider the problem specifically in two dimensions in both state and control and take $T = 1$, $\sigma = I$,

$$A = \frac{1}{10}\begin{pmatrix} 12 & 2 \\ 2 & 12 \end{pmatrix}, \quad C = \frac{1}{10}\begin{pmatrix} 15 & -3 \\ -3 & 15 \end{pmatrix},$$
$$Q = R = G = \frac{1}{10} I.$$

Table 1. **Linear-quadratic problem with delay**. Final estimate of the goal functional on the evaluation grid. Lower indicates smaller error. Analytical value: 2.231. Training resolution is given as a percentage of the evaluation resolution of 80 time steps.

	Training Resolution			
Model	100%	50%	25%	12.5%
RNN	2.493	5.162	8.870	7.600
LSTM	2.357	7.323	5.888	6.863
GRU	2.356	2.830	7.311	18.70
Neural RDE (ours)	2.358	2.457	2.509	2.803

Table 2. **Linear-quadratic problem driven by fractional Brownian motion**. Final estimate of the goal functional on the evaluation grid. Lower indicates smaller error. Training resolution is given as a percentage of the evaluation resolution of 40 time steps.

	Training Resolution			
Model	100%	50%	25%	12.5%
RNN	0.923	0.911	1.246	2.785
LSTM	0.873	0.961	1.779	3.422
GRU	0.891	0.925	1.236	2.791
Neural RDE (ours)	0.896	0.902	0.927	1.104

Table 2 shows the results for this experiment. We observe comparable performance between the LSTM, GRU and Neural RDE models at full training resolution, but with the Neural RDE significantly outperforming the other models when training resolution is decreased. At 12.5% of evaluation resolution, the models are trained on simulations using just five time steps; nevertheless, the Neural RDE appears to produce reasonable results with an error compared to the full resolution case more than one order of magnitude smaller than for the other models.

4.3. Portfolio optimisation problem with complete memory

We consider a portfolio optimisation problem with complete memory also studied in Han & Hu (2021). A detailed analysis of this problem including derivations of explicit solutions under exponential, power and log utilities is given in Pang & Hussain (2017). Here, the state process X_t represents the wealth of an investor and the $\alpha_t = (\alpha_t^1, \alpha_t^2)$ is a 2-dimensional control process, where α_t^1 is the amount of investment and α_t^2 is the consumption of the underlying asset, i.e. the fraction of wealth consumed at time t. The dynamics are given, for $t \in [0, T]$, by

$$\begin{aligned} \mathrm{d}X_t &= (((\mu_1 - r)\alpha_t^1 - \alpha_t^2 + r)X_t + \mu_2 Y_t)\mathrm{d}t \\ &\qquad + \sigma \alpha_t^1 X_t \mathrm{d}W_t, \\ Y_t &:= \int_{-\infty}^{0} e^{\lambda \xi} X_{t+\xi} \mathrm{d}\xi, \end{aligned} \tag{12}$$

with $X_0 = \phi(0), Y_0 = \int_{-\infty}^{0} e^{\lambda\xi}\phi(t+\xi)\mathrm{d}\xi$, for some square integrable function ϕ. The goal functional that we seem to maximise is as follows

$$J(\alpha) = \mathbb{E}\left[\int_0^T e^{-\beta t} U_1(\alpha_t^2 X_t)\mathrm{d}t + e^{-\beta T} U_2(X_T, Y_T)\right],$$

where $U_1(x) = \log(x), U_2(x, y) = \frac{1}{\beta}\log(x + \eta y), \eta = \frac{1}{2}(\sqrt{(r+\lambda^2) + 4\mu_2} - (r+\lambda))$. As in the previous example all the parameters are taken to be the same as in Han & Hu (2021).

The results for this experiment are shown in table 3, where we report the relative difference between the estimated and the theoretical goal functionals as well as relative pathwise L^2 error between the true and estimated process trajectories. We can see that the Neural RDE model slightly outperforms all alternative models on the relative difference of goal functionals and outperforms the second best model by one order of magnitude on the pathwise L^2 error.

Table 3. **Portfolio optimisation with complete memory**. Relative difference between the estimated and the theoretical goal functionals as well as relative pathwise L^2 error between the true and estimated process trajectories. Lower indicates smaller error.

	Relative errors ($\times 10^{-3}$)	
Model	Goal functional	Pathwise L^2
RNN	0.262	0.555
LSTM	1.034	2.929
GRU	0.541	1.116
Neural RDE (ours)	**0.238**	**0.043**

5. Conclusion

We proposed a framework for solving non-Markovian stochastic control problems continuous-time leveraging Neural RDEs. The main idea consists in parameterising the control process as the solution of a Neural RDE driven by the state process, so that the control-state joint dynamics are governed by an uncontrolled RDE with vector fields parameterised by neural networks. To deal with input paths of infinite 1-variation, we prove Theorem 3.1 which extends the universal approximation result in Kidger et al. (2020) to Neural RDEs driven by random rough paths. We showcased the time-resolution-invariance of our approach on various non-Markovian problems, achieving better performance than traditional RNN-based approaches. A natural next step is to enhance the algorithm's efficiency by exploring the relationship between non-Markovian control problems and path-dependent PDEs and BSDEs.

References

Bauer, H. and Rieder, U. Stochastic control problems with delay. *Mathematical Methods of Operations Research*, 62(3):411–427, 2005.

Boedihardjo, H., Geng, X., Lyons, T., and Yang, D. The signature of a rough path: Uniqueness. *Advances in Mathematics*, 293:720–737, 2016. ISSN 0001-8708. doi: https://doi.org/10.1016/j.aim.2016.02.011. URL https://www.sciencedirect.com/science/article/pii/S0001870816301104.

Cartea, Á. and Sánchez-Betancourt, L. The shadow price of latency: Improving intraday fill ratios in foreign exchange markets. *SIAM Journal on Financial Mathematics*, 12(1): 254–294, 2021.

Cartea, Á., Arribas, I. P., and Sánchez-Betancourt, L. Double-execution strategies using path signatures. *SIAM Journal on Financial Mathematics*, 13(4):1379–1417, 2022.

Chevyrev, I. and Kormilitzin, A. A primer on the signature method in machine learning, 2016. URL https://arxiv.org/abs/1603.03788.

Cucker, F. and Smale, S. Emergent behavior in flocks. *IEEE Transactions on Automatic Control*, 52(5):852–862, 2007.

Diehl, J., Friz, P. K., and Gassiat, P. Stochastic control with rough paths. *Appl. Math. Optim.*, 75(2): 285–315, 2017. ISSN 0095-4616. doi: 10.1007/s00245-016-9333-9. URL https://doi.org/10.1007/s00245-016-9333-9.

Duncan, T. E. and Pasik-Duncan, B. Stochastic linear-quadratic control for systems with a fractional brownian motion. In *49th IEEE Conference on Decision and Control (CDC)*, pp. 6163–6168. IEEE, 2010.

Fermanian, A. Embedding and learning with signatures. *Computational Statistics & Data Analysis*, 157:107148, 2021a. ISSN 0167-9473. doi: https://doi.org/10.1016/j.csda.2020.107148. URL https://www.sciencedirect.com/science/article/pii/S0167947320302395.

Fermanian, A. *Learning time-dependent data with the signature transform*. Theses, Sorbonne Université, October 2021b. URL https://tel.archives-ouvertes.fr/tel-03507274.

Flynn, C. Fractional brownian motion in python - pypi. URL https://pypi.org/project/fbm/.

Friz, P. K. and Hairer, M. *A Course on Rough Paths: With an Introduction to Regularity Structures*. Universitext. Springer International Publishing AG, Cham, 2020. ISBN 9783030415556.

Friz, P. K. and Victoir, N. B. *Multidimensional stochastic processes as rough paths*, volume 120 of *Cambridge Studies in Advanced Mathematics*. Cambridge University Press, Cambridge, 2010. ISBN 978-0-521-87607-0. doi: 10.1017/CBO9780511845079. URL https://doi.org/10.1017/CBO9780511845079. Theory and applications.

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

Germain, M., Pham, H., and Warin, X. Neural networks-based algorithms for stochastic control and PDEs in finance. *arXiv preprint arXiv:2101.08068*, 2021.

Gozzi, F., Marinelli, C., and Savin, S. On controlled linear diffusions with delay in a model of optimal advertising under uncertainty with memory effects. *Journal of Optimization Theory and Applications*, 142(2):291–321, 2009.

Grundel, D., Murphey, R., Pardalos, P., and Prokopyev, O. *Cooperative systems: Control and optimization*, volume 588. Springer Science & Business Media, 2007.

Hambly, B. and Lyons, T. Uniqueness for the signature of a path of bounded variation and the reduced path group. *Annals of mathematics*, 171(1):109–167, 2010. ISSN 0003-486X.

Han, J. and Hu, R. Recurrent neural networks for stochastic control problems with delay. *Mathematics of Control, Signals, and Systems*, 33(4):775–795, 2021.

Han, J., Jentzen, A., et al. Deep learning-based numerical methods for high-dimensional parabolic partial differential equations and backward stochastic differential equations. *Communications in mathematics and statistics*, 5 (4):349–380, 2017.

Han, J. et al. Deep learning approximation for stochastic control problems. *arXiv preprint arXiv:1611.07422*, 2016.

Hubert, E., Mastrolia, T., Possamaï, D., and Warin, X. Incentives, lockdown, and testing: from Thucydides' analysis to the COVID-19 pandemic. *Journal of Mathematical Biology*, 84(5):37, 2022.

Kalsi, J., Lyons, T., and Arribas, I. P. Optimal execution with rough path signatures. *SIAM Journal on Financial Mathematics*, 11(2):470–493, 2020.

Kamien, M. I. and Schwartz, N. L. *Dynamic optimization: the calculus of variations and optimal control in economics and management*. Courier Corporation, 2012.

Kidger, P., Bonnier, P., Perez Arribas, I., Salvi, C., and Lyons, T. Deep signature transforms. *Advances in Neural Information Processing Systems*, 32, 2019.

Kidger, P., Morrill, J., Foster, J., and Lyons, T. Neural controlled differential equations for irregular time series. *Advances in Neural Information Processing Systems*, 33: 6696–6707, 2020.

Kidger, P., Foster, J., Li, X., and Lyons, T. Efficient and accurate gradients for neural sdes. *arXiv preprint arXiv:2105.13493*, 2021a.

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

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Kolmanovskiı, V. and Shaıkhet, L. *Control of Systems with Aftereffect*, volume 157 of *Translations of Mathematical Monographs*. American Mathematical Society, 1996.

Kydland, F. and Prescott, E. Time to build and aggregate fluctuations. *Econometrica*, 50(6):1345–1370, 1982.

Le Jan, Y. and Qian, Z. Stratonovich's signatures of Brownian motion determine Brownian sample paths. *Probab. Theory Related Fields*, 157(1-2): 209–223, 2013. ISSN 0178-8051. doi: 10.1007/s00440-012-0454-z. URL https://doi.org/10.1007/s00440-012-0454-z.

Lemercier, M., Salvi, C., Damoulas, T., Bonilla, E., and Lyons, T. Distribution regression for sequential data. In Banerjee, A. and Fukumizu, K. (eds.), *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pp. 3754–3762. PMLR, 13–15 Apr 2021. URL https://proceedings.mlr.press/v130/lemercier21a.html.

Li, X., Wong, T.-K. L., Chen, R. T., and Duvenaud, D. Scalable gradients for stochastic differential equations. In *International Conference on Artificial Intelligence and Statistics*, pp. 3870–3882. PMLR, 2020.

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

Lyons, T. Rough paths, signatures and the modelling of functions on streams. *arXiv preprint arXiv:1405.4537*, 2014.

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

Mao, X. Numerical solutions of stochastic functional differential equations. *LMS journal of computation and mathematics*, 6:141–161, 2003.

Morrill, J., Salvi, C., Kidger, P., and Foster, J. Neural rough differential equations for long time series. In *International Conference on Machine Learning*, pp. 7829–7838. PMLR, 2021.

Øksendal, B., Sulem, A., and Zhang, T. Optimal control of stochastic delay equations and time-advanced backward stochastic differential equations. *Advances in Applied Probability*, 43(2):572–596, 2011.

Pang, T. and Hussain, A. A stochastic portfolio optimization model with complete memory. *Stochastic Analysis and Applications*, 35(4):742–766, 2017.

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

Protter, P. *Stochastic integration and differential equations*, volume 21 of *Stochastic modelling and applied probability*. Springer–Verlag Berlin Heidelberg, 2nd edition, 2005.

Salvatore, F. A stochastic control problem with delay arising in a pension fund model. *Finance and Stochastics*, 15(3): 421–459, 2011.

Sirignano, J. and Spiliopoulos, K. Dgm: A deep learning algorithm for solving partial differential equations. *Journal of computational physics*, 375:1339–1364, 2018.

Yong, J. and Zhou, X. Y. *Stochastic controls: Hamiltonian systems and HJB equations*, volume 43. Springer Science & Business Media, 1999.

Young, W. H. Vi. on the general theory integration. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 204(372-386):221–252, 1905.

A. Appendix

A.1. Background on rough path theory

The purpose of this appendix is to give an informal and very concise introduction to rough paths, their signatures, and their applications to machine learning.

For the abstract theory of rough paths we refer to Friz & Victoir (2010); Friz & Hairer (2020). An α-Hölder *rough path* $\boldsymbol{X}$ consists of an α-Hölder continuous path $X\colon [0,T] \to \mathbb{R}^d$ (the *trace* of $\boldsymbol{X}$) together with a collection of higher-order functions defined on the simplex $\Delta[0,T] \coloneqq \{(s,t) \in [0,T]^2 \mid 0 \le s \le t \le T\}$ which represent, in a precise algebraic and analytic sense, iterated integrals of X against itself. When X is smooth or of bounded variation, such integrals can be defined canonically in the usual Stieltjes sense, and similarly when X is $1/2 < \alpha$-Hölder continuous they can be defined via Young integration. However, when $\alpha \le 1/2$ there is no canonical way of defining them, and if X is a stochastic process, $\boldsymbol{X}$ is often defined through some notion of stochastic integration such as Itô or Stratonovich. $\boldsymbol{X}$ takes values in $T^{\lfloor 1/\alpha \rfloor}(\mathbb{R}^d)$, where $T^N(\mathbb{R}^d) \coloneqq \bigoplus_{n=0}^N (\mathbb{R}^d)^{\otimes n}$ denotes the tensor algebra over $\mathbb{R}^d$ truncated at level N and $\lfloor \cdot \rfloor$ is the floor function: this means, the rougher X is, the more terms $\boldsymbol{X}$ must contain. Once such terms are defined, the *signature* $S(\boldsymbol{X})$ of $\boldsymbol{X}$ is canonically defined through well-known notions of path integration. $S(\boldsymbol{X})$ is a map $\Delta[0,T] \to T(\!(\mathbb{R}^d)\!)$ (the algebra of formal series of tensors), and when $\alpha > 1/2$ it is canonically defined by Young integration as

$$S(\boldsymbol{X})_{st}^{(n)} \coloneqq \int_{s<u_1<\ldots<u_n<t} \mathrm{d}X_{u_1} \otimes \cdots \otimes \mathrm{d}X_{u_n}$$

where the superscript (n) denotes projection onto $(\mathbb{R}^d)^{\otimes n}$. When $\alpha \le 1/2$ the whole of $\boldsymbol{X}$, not just the trace X, is needed to define $S(\boldsymbol{X})$, and $S(\boldsymbol{X})^{(n)} = \boldsymbol{X}^{(n)}$ for $n \le \lfloor 1/\alpha \rfloor$. We will denote $C^\alpha([0,T],\mathbb{R}^d)$ the metrisable topological space of α-Hölder rough paths taking values in $\mathbb{R}^d$ with time horizon T: this is what Friz & Victoir (2010) call $C^{\alpha\text{-Höl}}([0,T], G^{\lfloor 1/\alpha \rfloor}(\mathbb{R}^d))$; in Friz & Hairer (2020) (which only treats the case of $\alpha > 1/3$, nevertheless sufficient for Brownian motion, which is α-Hölder regular for any $\alpha < 1/2$) this space is denoted $C_g^\alpha([0,T],\mathbb{R}^d)$, the superscript g standing for "geometric". Geometric rough paths are those which satisfy integration by parts relations, and are the only ones considered here; for example, Itô and Stratonovich integration both define rough paths above Brownian motion, but only the latter is geometric. This is not an issue when considering Itô SDEs, however, which can canonically be rewritten in Stratonovich form. The main example of rough path that we will consider is the Stratonovich Brownian rough path augmented with time: if W is a d-dimensional Brownian motion, we take α to be any real number in $(1/3, 1/2)$ and for $i,j = 1,\ldots d$ we let $\boldsymbol{W}_{st}^{ij} \coloneqq \int_{s<u<v<t} \circ\mathrm{d}W_u^i \circ \mathrm{d}W_v^j$, where $\circ\mathrm{d}W$ denotes Stratonovich integration. Time will take the zero-th coordinate, which means that when i or j above is 0, the integral is defined through standard Young/Stieltjes integration.

The main purpose of rough path theory is to give meaning to *rough differential equations* (RDEs) $\mathrm{d}Y = V(Y)\mathrm{d}\boldsymbol{X}$ which, in addition to having usual existence and uniqueness theorems, have the property that the solution map $\boldsymbol{X} \mapsto Y$ is continuous. This is not the case when considering SDEs: the map sending the Brownian sample path to the corresponding path of the solution, though defined on a set of full measure and measurable, is not continuous. An important RDE is the one satisfied by the signature itself on $T(\!(\mathbb{R}^d)\!)$: given a rough path $\boldsymbol{X}$ it holds that

$$\mathrm{d}S(\boldsymbol{X})_{0t} = S(\boldsymbol{X})_{0t} \otimes \mathrm{d}\boldsymbol{X}_t \tag{13}$$

The study of signatures is somewhat independent from that of rough paths, and is interesting even in the case of smooth or bounded variation paths (in which case $\boldsymbol{X} = X$). The main property of interest of the signature, established in Hambly & Lyons (2010) (and extended to the full rough path case in Boedihardjo et al. (2016)), is that, for paths of bounded variation, the series of tensors $S(X)_{0T}$ determines the path X up to *treelike equivalence*. Roughly speaking, the latter means that if two paths X, Y are such that $X \star \overleftarrow{Y}$ — with $\star$ denoting path concatenation and $\overleftarrow{}$ denoting path inversion — is a path that retraces itself and returns to the starting point, then the signature will not distinguish them: $S(X)_{0T} = S(Y)_{0T}$. We will write $X \sim Y$ for treelike equivalence (and similarly $\boldsymbol{X} \sim \boldsymbol{Y}$ in the generalised rough path sense of Boedihardjo et al. (2016)), and note that this includes (but is not limited to) the case in which Y is a reparameterisation of X. "Generic" paths $\mathbb{R}^d$ valued paths can be expected not to be tree-like (i.e. not to retrace themselves) when $d > 1$; for example, in Le Jan & Qian (2013) it was shown that Brownian rough paths in dimension 2 or greater a.s. do not contain tree-like pieces.

The result of Hambly & Lyons (2010) is a powerful statement that makes it possible to understand a path $X\colon [0,T] \to \mathbb{R}^d$ in terms of the series of tensors $S(X)_{0T}$. What's more, the signature has the property of "linearising" all functions on paths:

any non-linear function of X can be approximately expressed as a linear functional on $S(X)$. A precise version of this statement in the random rough path case is proved in 3.1 below. A fundamental ingredient for proving this type of result is the Stone-Weierstrass theorem: given a compact Hausdorff topological space K and a subalgebra A of $C(K, \mathbb{R})$ which contains a non-zero constant function and separates points (this means that for any two distinct $x, y \in K$ there exists $a \in A$ s.t. $a(x) \neq a(y)$), it holds that A is dense in $C(K, \mathbb{R})$. The prototypical application of this theorem is the proof of density of polynomials in $C([a, b], \mathbb{R})$. An important property that makes it possible to apply it to signatures is that linear functions on the signature, just like polynomials, form an algebra: if $\ell_1, \ell_2 \colon T((\mathbb{R}^d))^* = T(\mathbb{R}^d) \to \mathbb{R}$ are linear maps then

$$\langle \ell_1, S(\boldsymbol{X})_{0T} \rangle \langle \ell_1, S(\boldsymbol{X})_{0T} \rangle = \langle \ell_1 \sqcup\!\sqcup \ell_2, S(\boldsymbol{X})_{0T} \rangle$$

where $\sqcup\!\sqcup$ is the combinatorial operation of shuffling. This relation can be understood as a generalised integration by parts relation, as can be seen by taking ℓ_1 and ℓ_2 to be evaluations against elementary tensors: in this case (and X of bounded variation) the above identity reads

$$\begin{aligned} &\left(\int_{s<u_1<\ldots<u_n<t} \mathrm{d}X_{u_1}^{i_1} \cdots \mathrm{d}X_{u_m}^{i_m} \right) \left(\int_{s<v_1<\ldots<v_n<t} \mathrm{d}X_{v_1}^{j_1} \cdots \mathrm{d}X_{v_n}^{j_n} \right) \\ = &\sum_{\boldsymbol{k} \in \mathrm{Sh}(\boldsymbol{i}, \boldsymbol{j})} \int_{s<r_1<\ldots<r_{n+m}<t} \mathrm{d}X_{r_1}^{k_1} \cdots \mathrm{d}X_{r_{m+n}}^{k_{m+n}} \end{aligned}$$

where we are summing over all multiindices $\boldsymbol{k}$ obtained by shuffling the multiindices $(i_1, \ldots, i_m)$ and $(j_1, \ldots, j_m)$. For these reasons, signatures have been extensively used for in the context of machine learning for time series, see e.g. Chevyrev & Kormilitzin (2016); Fermanian (2021b).

A.2. Additional experimental details

In this final section of the appendix we present additional experimental details.

Stochastic control problem with delay (sec. 4.1) Trajectories of equation (9) are simulated using an Euler-Maruyama-type scheme on a uniform grid. All models are trained over 300 batches of 256 sample trajectories simulated on grid with 80, 40, 20 and 10 time steps. The final evaluation estimates of the goal functional are computed using 4096 sample trajectories simulated on a grid with 80 time steps. The dimension of the hidden states in the baseline models are: 400 for the RNN, 200 for the LSTM and 230 for the GRU. The latent dimension of the Neural RDE model is 200 and the vector field and initial lift are parameterised by fully connected feed-forward neural networks with two hidden layers of width 64. We take activations given by elementwise application of the SiLU function $x \mapsto \frac{x}{1+e^{-x}}$ and apply a final $\tanh$ non-linearity to the outputs to prevent unreasonably large values and initial losses.

Stochastic control problem driven by fractional Brownian motion (sec. 4.2) Trajectories are simulated using an Euler-Maruyama scheme with increments of fractional Brownian motion sampled using the Python package fbm (Flynn). We use uniform grids with 40 time steps for evaluation and 40, 20, 10 and 5 steps for training. All models are trained over 300 batches of 256 sample trajectories. The final evaluation estimates of the value of the goal functional are computed using 4096 sample trajectories. The dimension of the hidden states in the baseline models are: 250 for the RNN, 130 for the LSTM and 150 for the GRU. The latent dimension of the Neural RDE model is 200 and the vector field and initial lift are parameterised by feed-forward neural networks with two hidden layers of width 64 respectively.

Portfolio optimisation problem with complete memory (sec. 4.3) We simulate trajectories of equation (12) using an Euler-Maruyama-type scheme on a uniform grid of 200 time steps. All models are trained over 500 batches of 256 sample trajectories (similarly for evaluation). The dimension of the hidden states in the baseline models are: 600 for the RNN, 300 for the LSTM and 300 for the GRU. The latent dimension of the Neural RDE model is 350 and the vector field and initial lift are parameterised by feed-forward neural networks with two hidden layers of width 128.