Supplementary material

In sections A and B, we give a more thorough introduction to solving CDEs via the log-ODE method.

In section C we discuss the experimental details such as the choice of network structure, computing infrastructure and hyperparameter selection approach.

In section D we give a full breakdown of every experimental result.

A. An introduction to the log-ODE method for controlled differential equations

The log-ODE method is an effective method for approximating the controlled differential equation:

$$\begin{aligned} \mathrm{d}Y_t &= f(Y_t)\,\mathrm{d}X_t, \\ Y_0 &= \xi, \end{aligned} \tag{11}$$

where $X : [0, T] \to \mathbb{R}^d$ has finite length, $\xi \in \mathbb{R}^n$ and $f : \mathbb{R}^n \to L(\mathbb{R}^d, \mathbb{R}^n)$ is a function with certain smoothness assumptions so that the CDE (11) is well posed. Throughout these appendices, $L(U, V)$ denotes the space of linear maps between the vector spaces U and V. In rough path theory, the function f is referred to as the "vector field" of (11) and usually assumed to have Lip(γ) regularity (see definition 10.2 in Friz & Victoir (2010)). In this section, we assume one of the below conditions on the vector field:

1. f is bounded and has N bounded derivatives.
2. f is linear.

In order to define the log-ODE method, we will first consider the tensor algebra and path signature.

Definition A.1 *We say that* $T\big(\mathbb{R}^d\big) := \mathbb{R} \oplus \mathbb{R}^d \oplus (\mathbb{R}^d)^{\otimes 2} \oplus \cdots$ *is the* tensor algebra *of* $\mathbb{R}^d$ *and* $T\big(\big(\mathbb{R}^d\big)\big) := \big\{\boldsymbol{a} = \big(a_0, a_1, \cdots\big) : a_k \in \big(\mathbb{R}^d\big)^{\otimes k}\ \forall k \geq 0\big\}$ *is the set of formal series of tensors of* $\mathbb{R}^d$*. Moreover,* $T\big(\mathbb{R}^d\big)$ *and* $T\big(\big(\mathbb{R}^d\big)\big)$ *can be endowed with the operations of addition and multiplication. Given* $\boldsymbol{a} = (a_0, a_1, \cdots)$ *and* $\boldsymbol{b} = (b_0, b_1, \cdots)$*, we have*

$$\boldsymbol{a} + \boldsymbol{b} = \big(a_0 + b_0, a_1 + b_1, \cdots\big), \tag{12}$$

$$\boldsymbol{a} \otimes \boldsymbol{b} = \big(c_0, c_1, c_2, \cdots\big), \tag{13}$$

where for $n \geq 0$*, the* n*-th term* $c_n \in \big(\mathbb{R}^d\big)^{\otimes n}$ *can be written as*

$$c_n := \sum_{k=0}^{n} a_k \otimes b_{n-k}. \tag{14}$$

The use of $\otimes$ *in equation (14) denotes the usual tensor product. The use of* $\otimes$ *in equation (13) is also referred to as the "tensor product": when precisely one* a_i *and precisely one* b_i *are nonzero then it reduces to the usual tensor product; equation (13) is a generalisation.*

Definition A.2 *The* signature *of a finite length path* $X : [0, T] \to \mathbb{R}^d$ *over the interval* $[s, t]$ *is defined as the following collection of iterated (Riemann–Stieltjes) integrals:*

$$S_{s,t}\big(X\big) := \Big(1, X_{s,t}^{(1)}, X_{s,t}^{(2)}, X_{s,t}^{(3)}, \dots\Big) \in T\big(\big(\mathbb{R}^d\big)\big), \tag{15}$$

where for $n \geq 1$*,*

$$X_{s,t}^{(n)} := \int_{s<u_1<\cdots<u_n<t} \cdots \int \mathrm{d}X_{u_1} \otimes \cdots \otimes \mathrm{d}X_{u_n} \in \big(\mathbb{R}^d\big)^{\otimes n}.$$

Similarly, we can define the depth-N (or truncated) signature of the path X on $[s,t]$ as

$$S^N_{s,t}(X) := \left(1, X^{(1)}_{s,t}, X^{(2)}_{s,t}, \dots, X^{(N)}_{s,t}\right) \in T^N\left(\mathbb{R}^d\right), \tag{16}$$

where $T^N\left(\mathbb{R}^d\right) := \mathbb{R} \oplus \mathbb{R}^d \oplus (\mathbb{R}^d)^{\otimes 2} \oplus \cdots \oplus (\mathbb{R}^d)^{\otimes N}$ denotes the truncated tensor algebra.

The (truncated) signature provides a natural feature set that describes the effects a path X has on systems that can be modelled by (11). That said, defining the log-ODE method actually requires the so-called "log-signature" which efficiently encodes the same integral information as the signature. The log-signature is obtained from the path's signature by removing certain algebraic redundancies, such as

$$\int_0^t \int_0^s \mathrm{d}X^i_u \mathrm{d}X^j_s + \int_0^t \int_0^s \mathrm{d}X^j_u \mathrm{d}X^i_s = X^i_t X^j_t,$$

for $i, j \in \{1, \cdots, d\}$, which follows by the integration-by-parts formula. To this end, we will define the logarithm map on the depth-N truncated tensor algebra $T^N\left(\mathbb{R}^d\right) := \mathbb{R} \oplus \mathbb{R}^d \oplus \cdots \oplus (\mathbb{R}^d)^{\otimes N}$.

Definition A.3 (The logarithm of a formal series) *For $\boldsymbol{a} = (a_0, a_1, \cdots) \in T\left(\left(\mathbb{R}^d\right)\right)$ with $a_0 > 0$, define $\log(\boldsymbol{a})$ to be the element of $T\left(\left(\mathbb{R}^d\right)\right)$ given by the following series:*

$$\log(\boldsymbol{a}) := \log(a_0) + \sum_{n=1}^{\infty} \frac{(-1)^n}{n} \left(\mathbf{1} - \frac{\boldsymbol{a}}{a_0}\right)^{\otimes n}, \tag{17}$$

where $\mathbf{1} = (1, 0, \cdots)$ is the unit element of $T\left(\left(\mathbb{R}^d\right)\right)$ and $\log(a_0)$ is viewed as $\log(a_0)\mathbf{1}$.

Definition A.4 (The logarithm of a truncated series) *For $\boldsymbol{a} = (a_0, a_1, \cdots, a_N) \in T\left(\left(\mathbb{R}^d\right)\right)$ with $a_0 > 0$, define $\log^N(\boldsymbol{a})$ to be the element of $T^N\left(\mathbb{R}^d\right)$ defined from the logarithm map (17) as*

$$\log^N(\boldsymbol{a}) := P_N\big(\log(\widetilde{\boldsymbol{a}})\big), \tag{18}$$

where $\widetilde{\boldsymbol{a}} := (a_0, a_1, \cdots, a_N, 0, \cdots) \in T\left(\left(\mathbb{R}^d\right)\right)$ and P_N denotes the standard projection map from $T\left(\left(\mathbb{R}^d\right)\right)$ onto $T^N\left(\mathbb{R}^d\right)$.

Definition A.5 *The* log-signature *of a finite length path $X : [0,T] \to \mathbb{R}^d$ over the interval $[s,t]$ is defined as $\mathrm{LogSig}_{s,t}(X) := \log(S_{s,t}(X))$, where $S_{s,t}(X)$ denotes the path signature of X given by Definition A.2. Likewise, the depth-N (or truncated) log-signature of X is defined for each $N \geq 1$ as $\mathrm{LogSig}^N_{s,t}(X) := \log^N(S^N_{s,t}(X))$.*

In this section, we view each $\mathrm{LogSig}^N_{s,t}(X)$ as an element of $T^N\left(\mathbb{R}^d\right)$ to simplify the definition of the log-ODE method. That said, this is equivalent to the definition used in the main body of the paper, which defines the log-signature as a map from $X\colon [0,T] \to \mathbb{R}^d$ to $\mathbb{R}^{\beta(d,N)}$. This corresponds to the interpretation of a log-signature as an element of a certain free Lie algebra (see, for example, Lyons et al. (2007); Reizenstein (2017) for details). The exact form of $\beta(d,N)$ is given by

$$\beta(d,N) = \sum_{k=1}^{N} \frac{1}{k} \sum_{i|k} \mu\left(\frac{k}{i}\right) d^i$$

with μ the Möbius function. The precise order of this remains an open question.

The final ingredient we use to define the log-ODE method are the derivatives of the vector field f. It is worth noting that these derivatives also naturally appear in the Taylor expansion of (11).

Definition A.6 (Vector field derivatives) *We define $f^{\circ k} : \mathbb{R}^n \to L((\mathbb{R}^d)^{\otimes k}, \mathbb{R}^n)$ recursively by*

$$\begin{aligned} f^{\circ(0)}(y) &:= y, \\ f^{\circ(1)}(y) &:= f(y), \\ f^{\circ(k+1)}(y) &:= D\big(f^{\circ k}\big)(y) f(y), \end{aligned}$$

for $y \in \mathbb{R}^n$, where $D\big(f^{\circ k}\big)$ denotes the Fréchet derivative of $f^{\circ k}$.

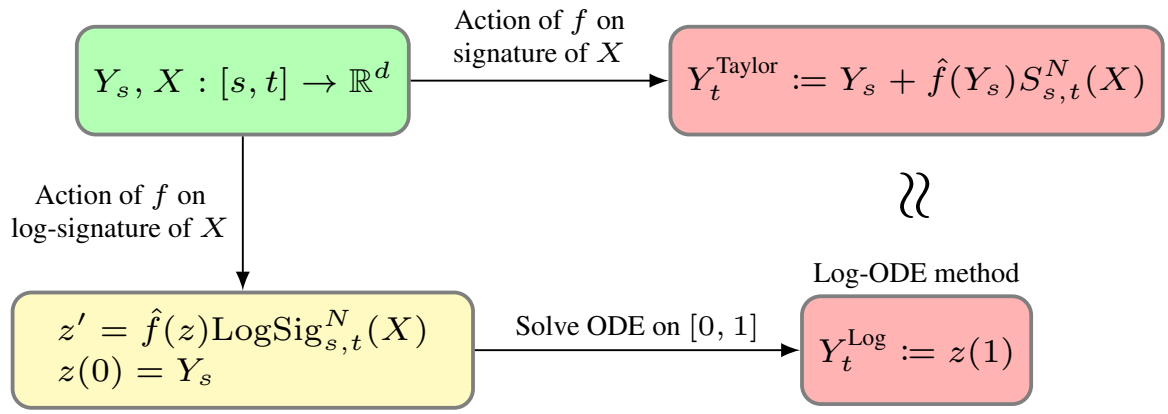

Figure 6. Illustration of the log-ODE and Taylor methods for controlled differential equations.

Using these definitions, we can describe two closely related numerical methods for the CDE (11).

Definition A.7 (The Taylor method) *Given the CDE (11), we can use the path signature of X to approximate the solution Y on an interval $[s,t]$ via its truncated Taylor expansion. That is, we use*

$$\text{Taylor}(Y_s, f, S^N_{s,t}(X)) := \sum_{k=0}^{N} f^{\circ k}(Y_s)\pi_k\big(S^N_{s,t}(X)\big), \tag{19}$$

as an approximation for Y_t where each $\pi_k : T^N(\mathbb{R}^d) \to (\mathbb{R}^d)^{\otimes k}$ is the projection map onto $\big(\mathbb{R}^d\big)^{\otimes k}$.

Definition A.8 (The Log-ODE method) *Using the Taylor method (19), we can define the function $\widehat{f} : \mathbb{R}^n \to L(T^N(\mathbb{R}^d), \mathbb{R}^n)$ by $\widehat{f}(z) := \text{Taylor}(z, f, \cdot)$. By applying $\widehat{f}$ to the truncated log-signature of the path X over an interval $[s,t]$, we can define the following ODE on $[0,1]$*

$$\begin{aligned} \frac{\mathrm{d}z}{\mathrm{d}u} &= \widehat{f}(z)\text{LogSig}^N_{s,t}(X), \\ z(0) &= Y_s. \end{aligned} \tag{20}$$

Then the log-ODE approximation of Y_t (given Y_s and $\text{LogSig}^N_{s,t}(X)$) is defined as

$$\text{LogODE}(Y_s, f, \text{LogSig}^N_{s,t}(X)) := z(1). \tag{21}$$

Remark A.9 *Our assumptions of f ensure that $z \mapsto \widehat{f}(z)\text{LogSig}^N_{s,t}(X)$ is either globally bounded and Lipschitz continuous or linear. Hence both the Taylor and log-ODE methods are well defined.*

Remark A.10 *It is well known that the log-signature of a path X lies in a certain free Lie algebra (this is detailed in section 2.2.4 of Lyons et al. (2007)). Furthermore, it is also a theorem that the Lie bracket of two vector fields is itself a vector field which doesn't depend on choices of basis. By expressing $\text{LogSig}^N_{s,t}(X)$ using a basis of the free Lie algebra, it can be shown that only the vector field f and its (iterated) Lie brackets are required to construct the log-ODE vector field $\widehat{f}(z)\text{LogSig}^N_{s,t}(X)$. In particular, this leads to our construction of the log-ODE (8) using the Lyndon basis of the free Lie algebra (see (Reizenstein, 2017) for a precise description of the Lyndon basis). We direct the reader to Lyons (2014) and Boutaib et al. (2014) for further details on this Lie theory.*

To illustrate the log-ODE method, we give two examples:

Example A.11 (The "increment-only" log-ODE method) *When $N = 1$, the ODE (20) becomes*

$$\frac{\mathrm{d}z}{\mathrm{d}u} = f(z)X_{s,t},$$
$$z(0) = Y_s.$$

Therefore we see that this "increment-only" log-ODE method is equivalent to driving the original CDE (11) by a piecewise linear approximation of the control path X. This is a classical approach for stochastic differential equations (i.e. when $X_t = (t, W_t)$ with W denoting a Brownian motion) and is an example of a Wong-Zakai approximation (see Wong & Zakai (1965) for further details).

Example A.12 (An application for SDE simulation) *Consider the following affine SDE,*

$$\mathrm{d}Y_t = a(b - y_t)\,\mathrm{d}t + \sigma\, y_t \circ \mathrm{d}W_t, \tag{22}$$
$$y(0) = y_0 \in \mathbb{R}_{\geq 0},$$

where $a, b \geq 0$ are the mean reversion parameters, $\sigma \geq 0$ is the volatility and W denotes a standard real-valued Brownian motion. The $\circ$ means that this SDE is understood in the Stratonovich sense. The SDE (22) is known in the literature as Inhomogeneous Geometric Brownian Motion (or IGBM). Using the control path $X = \{(t, W_t)\}_{t\geq 0}$ and setting $N = 3$, the log-ODE (20) becomes

$$\frac{\mathrm{d}z}{\mathrm{d}u} = a(b - z_u)h + \sigma\, z_u W_{s,t} - ab\sigma A_{s,t} + ab\sigma^2 L^{(1)}_{s,t} + a^2 b\sigma L^{(2)}_{s,t},$$
$$z(0) = Y_s.$$

where $h := t - s$ denotes the step size and the random variables $A_{s,t}, L^{(1)}_{s,t}, L^{(2)}_{s,t}$ are given by

$$A_{s,t} := \int_s^t W_{s,r}\,\mathrm{d}r - \frac{1}{2}hW_{s,t},$$
$$L^{(1)}_{s,t} := \int_s^t \int_s^r W_{s,v} \circ \mathrm{d}W_v\,\mathrm{d}r - \frac{1}{2}W_{s,t}A_{s,t} - \frac{1}{6}hW^2_{s,t},$$
$$L^{(2)}_{s,t} := \int_s^t \int_s^r W_{s,v}\,\mathrm{d}v\,\mathrm{d}r - \frac{1}{2}hA_{s,t} - \frac{1}{6}h^2 W_{s,t}.$$

In Foster et al. (2020), the depth-3 log-signature of $X = \{(t, W_t)\}_{t\geq 0}$ was approximated so that the above log-ODE method became practical and this numerical scheme exhibited state-of-the-art convergence rates. For example, the approximation error produced by 25 steps of the high order log-ODE method was similar to the error of the "increment only" log-ODE method with 1000 steps.

B. Convergence of the log-ODE method for rough differential equations

In this section, we shall present "rough path" error estimates for the log-ODE method. In addition, we will discuss the case when the vector fields governing the rough differential equation are linear. We begin by stating the main result of Boutaib et al. (2014) which quantifies the approximation error of the log-ODE method in terms of the regularity of the systems vector field f and control path X. Since this section uses a number of technical definitions from rough path theory, we recommend Lyons et al. (2007) as an introduction to the subject.

For $T > 0$, we will use the notation $\triangle_T := \{(s, t) \in [0, T]^2 : s < t\}$ to denote a rescaled 2-simplex.

Theorem B.1 (Lemma 15 in Boutaib et al. (2014)) *Consider the rough differential equation*

$$\mathrm{d}Y_t = f(Y_t)\,\mathrm{d}X_t, \tag{23}$$
$$Y_0 = \xi,$$

where we make the following assumptions:

- *X is a* geometric p-rough path *in $\mathbb{R}^d$, that is $X : \triangle_T \to T^{\lfloor p \rfloor}(\mathbb{R}^d)$ is a continuous path in the tensor algebra $T^{\lfloor p \rfloor}(\mathbb{R}^d) := \mathbb{R} \oplus \mathbb{R}^d \oplus \left(\mathbb{R}^d\right)^{\otimes 2} \oplus \cdots \oplus \left(\mathbb{R}^d\right)^{\otimes \lfloor p \rfloor}$ with increments*

$$X_{s,t} = \left(1, X_{s,t}^{(1)}, X_{s,t}^{(2)}, \cdots, X_{s,t}^{(\lfloor p \rfloor)}\right), \qquad X_{s,t}^{(k)} := \pi_k\left(X_{s,t}\right), \tag{24}$$

where $\pi_k : T^{\lfloor p \rfloor}\left(\mathbb{R}^d\right) \to \left(\mathbb{R}^d\right)^{\otimes k}$ is the projection map onto $\left(\mathbb{R}^d\right)^{\otimes k}$, such that there exists a sequence of continuous finite variation paths $x_n : [0, T] \to \mathbb{R}^d$ whose truncated signatures converge to X in the $\boldsymbol{p}$-variation metric*:*

$$d_p\left(S^{\lfloor p \rfloor}(x_n), X\right) \to 0, \tag{25}$$

as $n \to \infty$, where the p-variation between two continuous paths Z^1 and Z^2 in $T^{\lfloor p \rfloor}(\mathbb{R}^d)$ is

$$d_p\left(Z^1, Z^2\right) := \max_{1 \le k \le \lfloor p \rfloor} \sup_{\mathcal{D}} \left(\sum_{t_i \in \mathcal{D}} \left\| \pi_k\left(Z^1_{t_i, t_{i+1}}\right) - \pi_k\left(Z^2_{t_i, t_{i+1}}\right) \right\|^{\frac{p}{k}} \right)^{\frac{k}{p}}, \tag{26}$$

where the supremum is taken over all partitions $\mathcal{D}$ of $[0, T]$ and the norms $\|\cdot\|$ must satisfy (up to some constant)

$$\|a \otimes b\| \le \|a\| \|b\|,$$

for $a \in (\mathbb{R}^d)^{\otimes n}$ and $b \in (\mathbb{R}^d)^{\otimes m}$. For example, we can take $\|\cdot\|$ to be the projective or injective tensor norms (see Propositions 2.1 and 3.1 in Ryan (2002)).

- *The solution Y and its initial value ξ both take their values in $\mathbb{R}^n$.*
- *The collection of vector fields $\{f_1, \cdots, f_d\}$ on $\mathbb{R}^n$ are denoted by $f : \mathbb{R}^n \to L(\mathbb{R}^n, \mathbb{R}^d)$, where $L(\mathbb{R}^n, \mathbb{R}^d)$ is the space of linear maps from $\mathbb{R}^n$ to $\mathbb{R}^d$. We will assume that f has* Lip($\boldsymbol{\gamma}$) regularity *with $\gamma > p$. That is, f it is bounded with $\lfloor \gamma \rfloor$ bounded derivatives, the last being Hölder continuous with exponent $(\gamma - \lfloor \gamma \rfloor)$. Hence the following norm is finite:*

$$\|f\|_{\mathrm{Lip}(\gamma)} := \max_{0 \le k \le \lfloor \gamma \rfloor} \left\|D^k f\right\|_\infty \vee \left\|D^{\lfloor \gamma \rfloor} f\right\|_{(\gamma - \lfloor \gamma \rfloor)\text{-Höl}}, \tag{27}$$

where $D^k f$ is the k-th (Fréchet) derivative of f and $\|\cdot\|_{\alpha\text{-Höl}}$ is the standard α-Hölder norm with $\alpha \in (0, 1)$.

- *The RDE (23) is defined in the Lyon's sense. Therefore by the Universal Limit Theorem (see Theorem 5.3 in Lyons et al. (2007)), there exists a unique solution $Y : [0, T] \to \mathbb{R}^n$.*

We define the log-ODE for approximating the solution Y over an interval $[s, t] \subset [0, T]$ as follows:

1. *Compute the depth-$\lfloor \gamma \rfloor$ log-signature of the control path X over $[s, t]$. That is, we obtain $\mathrm{LogSig}^{\lfloor \gamma \rfloor}_{s,t}(X) := \log_{\lfloor \gamma \rfloor}\left(S^{\lfloor \gamma \rfloor}_{s,t}(X)\right) \in T^{\lfloor \gamma \rfloor}(\mathbb{R}^d)$, where $\log_{\lfloor \gamma \rfloor}(\cdot)$ is defined by projecting the standard tensor logarithm map onto $\{a \in T^{\lfloor \gamma \rfloor}(\mathbb{R}^d) : \pi_0(a) > 0\}$.*
2. *Construct the following (well-posed) ODE on the interval $[0, 1]$,*

$$\frac{\mathrm{d}z^{s,t}}{\mathrm{d}u} = F\left(z^{s,t}\right), \qquad z_0^{s,t} = Y_s, \tag{28}$$

where the vector field $F : \mathbb{R}^n \to \mathbb{R}^n$ is defined from the log-signature as

$$F(z) := \sum_{k=1}^{\lfloor \gamma \rfloor} f^{\circ k}(z) \pi_k\left(\mathrm{LogSig}^{\lfloor \gamma \rfloor}_{s,t}(X)\right). \tag{29}$$

Recall that $f^{\circ k} : \mathbb{R}^n \to L((\mathbb{R}^d)^{\otimes k}, \mathbb{R}^n)$ was defined previously in Definition A.6.

Then we can approximate Y_t using the $u = 1$ solution of (28). Moreover, there exists a universal constant $C_{p,\gamma}$ depending only on p and γ such that

$$\left\|Y_t - z_1^{s,t}\right\| \leq C_{p,\gamma}\|f\|^{\gamma}_{\mathrm{Lip}(\gamma)}\|X\|^{\gamma}_{p\text{-}var;[s,t]}, \tag{30}$$

where $\|\cdot\|_{p\text{-}var;[s,t]}$ is the p-variation norm defined for paths in $T^{\lfloor p \rfloor}(\mathbb{R}^d)$ by

$$\|X\|_{p\text{-}var;[s,t]} := \max_{1\leq k\leq \lfloor p \rfloor} \sup_{\mathcal{D}} \left(\sum_{t_i \in \mathcal{D}} \left\|X^k_{t_i,t_{i+1}}\right\|^{\frac{p}{k}} \right)^{\frac{k}{p}}, \tag{31}$$

with the supremum taken over all partitions $\mathcal{D}$ of $[s,t]$.

Remark B.2 *If the vector fields $\{f_1, \cdots, f_d\}$ are linear, then it immediately follows that F is linear.*

Although the above theorem requires some sophisticated theory, it has a simple conclusion - namely that log-ODEs can approximate controlled differential equations. That said, the estimate (30) does not directly apply when the vector fields $\{f_i\}$ are linear as they would be unbounded. Fortunately, it is well known that linear RDEs are well posed and the growth of their solutions can be estimated.

Theorem B.3 (Theorem 10.57 in Friz & Victoir (2010)) *Consider the linear RDE on $[0,T]$*

$$\begin{aligned} \mathrm{d}Y_t &= f(Y_t)\,\mathrm{d}X_t, \\ Y_0 &= \xi, \end{aligned}$$

where X is a geometric p-rough path in $\mathbb{R}^d$, $\xi \in \mathbb{R}^n$ and the vector fields $\{f_i\}_{1\leq i\leq d}$ take the form $f_i(y) = A_i y + B$ where $\{A_i\}$ and $\{B_i\}$ are $n \times n$ matrices. Let K denote an upper bound on $\max_i(\|A_i\| + \|B_i\|)$. Then a unique solution $Y : [0,T] \to \mathbb{R}^n$ exists. Moreover, it is bounded and there exists a constant C_p depending only on p such that

$$\|Y_t - Y_s\| \leq C_p\big(1 + \|\xi\|\big)K\|X\|_{p\text{-}var;[s,t]} \exp\left(C_p K^p \|X\|^p_{p\text{-}var;[s,t]}\right), \tag{32}$$

for all $0 \leq s \leq t \leq T$.

When the vector fields of the RDE (23) are linear, then the log-ODE (28) also becomes linear. Therefore, the log-ODE solution exists and is explicitly given as the exponential of the matrix F.

Theorem B.4 *Consider the same linear RDE on $[0,T]$ as in Theorem B.3,*

$$\begin{aligned} \mathrm{d}Y_t &= f(Y_t)\,\mathrm{d}X_t, \\ Y_0 &= \xi. \end{aligned}$$

Then the log-ODE vector field F given by (29) is linear and the solution of the associated ODE (28) exists and satisfies

$$\|z_u^{s,t}\| \leq \|Y_s\| \exp\left(\sum_{m=1}^{\lfloor \gamma \rfloor} K^m \left\| \pi_m\Big(\mathrm{LogSig}^{\lfloor \gamma \rfloor}_{s,t}(X)\Big) \right\| \right), \tag{33}$$

for $u \in [0,1]$ and all $0 \leq s \leq t \leq T$.

Proof B.5 *Since F is a linear vector field on $\mathbb{R}^n$, we can view it as an $n \times n$ matrix and so for $u \in [0,1]$,*

$$z_u^{s,t} = \exp(uF) z_0^{s,t},$$

where $\exp$ denotes the matrix exponential. The result now follows by the standard estimate $\|\exp(F)\| \leq \exp(\|F\|)$.

Remark B.6 *Due to the boundedness of linear RDEs (32) and log-ODEs (33), the arguments that established Theorem B.1 will hold in the linear setting as $\|f\|_{\mathrm{Lip}(\gamma)}$ would be finite when defined on the domains that the solutions Y and z lie in.*

Given the local error estimate (30) for the log-ODE method, we can now consider the approximation error that is exhibited by a log-ODE numerical solution to the RDE (23). Thankfully, the analysis required to derive such global error estimates was developed by Greg Gyurkó in his PhD thesis. Thus the following result is a straightforward application of Theorem 3.2.1 from Gyurkó (2008).

Theorem B.7 *Let X, f and Y satisfy the assumptions given by Theorem B.1 and suppose that $\{0 = t_0 < t_1 < \cdots < t_N = T\}$ is a partition of $[0, T]$ with $\max_k \|X\|_{p\text{-}var;[t_k,t_{k+1}]}$ sufficiently small. We can construct a numerical solution $\{Y_k^{\log}\}_{0\leq k\leq N}$ of (23) by setting $Y_0^{\log} := Y_0$ and for each $k \in \{0, 1, \cdots, N-1\}$, defining $Y_{k+1}^{\log}$ to be the solution at $u = 1$ of the following ODE:*

$$\begin{aligned} \frac{\mathrm{d}z^{t_k,t_{k+1}}}{\mathrm{d}u} &:= F\big(z^{t_k,t_{k+1}}\big), \\ z_0^{t_k,t_{k+1}} &:= Y_k^{\log}, \end{aligned} \tag{34}$$

where the vector field F is constructed from the log-signature of X over the interval $[t_k, t_{k+1}]$ according to (29). Then there exists a constant C depending only on p, γ and $\|f\|_{\mathrm{Lip}(\gamma)}$ such that

$$\big\|Y_{t_k} - Y_k^{\log}\big\| \leq C \sum_{i=0}^{k-1} \|X\|^{\gamma}_{p\text{-}var;[t_i,t_{i+1}]}, \tag{35}$$

for $0 \leq k \leq N$.

Remark B.8 *The above error estimate also holds when the vector field f is linear (by Remark B.6)).*

Since $\lfloor\gamma\rfloor$ is the truncation depth of the log-signatures used to construct each log-ODE vector field, we see that high convergence rates can be achieved through using more terms in each log-signature. It is also unsurprising that the error estimate (35) increases with the "roughness" of the control path. So just as in our experiments, we see that the performance of the log-ODE method can be improved by choosing an appropriate step size and depth of log-signature.

C. Experimental details

Code The code to reproduce the experiments is available at [redacted; see supplementary material]

Data splits Each dataset was split into a training, validation, and testing dataset with relative sizes 70%/15%/15%.

Normalisation The training splits of each dataset were normalised to zero mean and unit variance. The statistics from the training set were then used to normalise the validation and testing datasets.

Architecture We give a graphical description of the architecture used for updating the Neural CDE hidden state in figure 7. The input is first run through a multilayer perceptron with n layers of size h, with with n, h being hyperparameters. ReLU nonlinearities are used at each layer except the final one, where we instead use a tanh nonlinearity. The goal of this is to help prevent term blow-up over the long sequences.

Note that this is a small inconsistency between this work and the original model proposed in Kidger et al. (2020). Here, we applied the tanh function as the final hidden layer nonlinearity, whilst in the original paper the tanh nonlinearity is applied after the final linear map. Both methods are used to constrain the rate of change of the hidden state; we do not know of a reason to prefer one over the other.

Note that the final linear layer in the multilayer perceptron is reshaped to produce a matrix-valued output, of shape $v \times p$. (As $\widehat{f_\theta}$ is matrix-valued.) A matrix-vector multiplication with the log-signature then produces the vector field for the ODE solver.

ODE Solver All problems used the 'rk4' solver as implemented by `torchdiffeq` (Chen, 2018) version 0.0.1.

Computing infrastructure All EigenWorms experiments were run on a computer equipped with three GeForce RTX 2080 Ti's. All BIDMC experiments were run on a computed with two GeForce RTX 2080 Ti's and two Quadro GP100's.

Optimiser All experiments used the Adam optimiser. The learning rate was initialised at 0.032 divided by batch size. The batch size used was 1024 for EigenWorms and 512 for the BIDMC problems. If the validation loss failed to decrease after 15 epochs the learning rate was reduced by a factor of 10. If the validation loss did not decrease after 60 epochs, training was terminated and the model was rolled back to the point at which it achieved the lowest loss on the validation set.

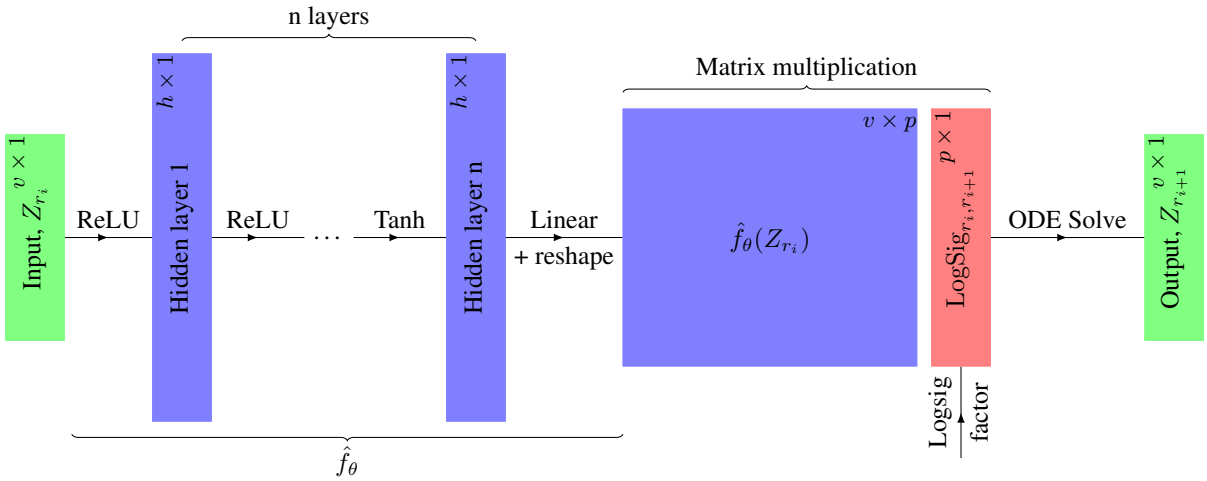

Figure 7. Overview of the hidden state update network structure. We give the dimensions at each layer in the top right hand corner of each box.

Hyperparameter selection Hyperparameters were selected to optimise the score of the NCDE_1 model on the validation set. For each dataset the search was performed with a step size that meant the total number of hidden state updates was equal to 500, as this represented a good balance between length and speed that allowed us to complete the search in a reasonable time-frame. In particular, this was short enough that we could train using the non-adjoint training method which helped to speed this section up. The hyperparameters that were considered were:

- Hidden dimension: [16, 32, 64] - The dimension of the hidden state Z_t.
- Number of layers: [2, 3, 4] - The number of hidden state layers.
- Hidden hidden multiplier: [1, 2, 3] - Multiplication factor for the hidden hidden state, this being the 'Hidden layer k' in figure 7. The dimension of each of these 'hidden hidden' layers with be this value multiplied by 'Hidden dimension'.

We ran each of these 27 total combinations for every dataset and the parameters that corresponded were used as the parameters when training over the full depth and step grid. The full results from the hyperparameter search are listed in tables (3, 5) with bolded values to show which values were eventually selected.

D. Experimental Results

Here we include the full breakdown of all experimental results. Tables 7 and 8 include all results from the EigenWorms and BIDMC datasets respectively.

Validation accuracy	Hidden dim	Num layers	Hidden hidden multiplier	Total params
33.3	16	2	3	5509
43.6	16	2	2	5509
56.4	16	2	1	4453
64.1	16	3	2	8869
38.5	16	3	3	8869
51.3	16	3	1	6517
82.1	16	4	2	12741
35.9	16	4	3	12741
53.8	16	4	1	8581
35.9	32	2	3	21253
74.4	32	2	2	21253
43.6	32	2	1	17093
53.8	32	3	3	34629
87.2	**32**	**3**	**2**	**34629**
64.1	32	3	1	25317
35.9	32	4	3	50053
71.8	32	4	1	33541
79.5	32	4	2	50053
41.0	64	2	3	83461
64.1	64	2	2	83461
48.7	64	3	3	136837
59.0	64	3	2	136837
51.3	64	2	1	66949
56.4	64	4	2	198405
64.1	64	4	3	198405
64.1	64	3	1	99781
51.3	64	4	1	132613

Table 3. Hyperparamter selection results for the EigenWorms dataset. The blue values denote the selected hyperparameters.

Validation accuracy	Hidden dim	Total params
61.5	32	11299
53.8	64	24611
64.1	**128**	**57379**
59.0	192	98339
61.5	256	147491
59.0	320	204835
64.1	388	274739

Table 4. Hyperparamter selection results for the ODE-RNN model on the EigenWorms dataset

Validation loss			Hidden dim	Num layers	Hidden hidden multiplier	Total params
RR	HR	SpO2				
1.72	6.10	2.07	16	2	1	2209
1.57	5.58	1.97	16	2	2	3265
1.55	6.10	1.33	16	2	3	3265
1.80	5.16	2.05	16	3	1	3249
1.61	5.22	1.62	16	3	2	5601
1.56	3.34	1.18	16	3	3	5601
1.57	3.86	1.97	16	4	1	4289
1.45	3.54	1.25	16	4	2	8449
1.54	3.93	1.09	16	4	3	8449
1.56	6.81	1.87	32	2	1	8513
1.42	3.11	1.43	32	2	2	12673
1.54	3.60	1.11	32	2	3	12673
1.54	3.52	1.57	32	3	1	12641
1.39	2.96	1.03	32	3	2	21953
1.47	2.95	1.05	32	3	3	21953
1.55	3.00	2.00	32	4	1	16769
1.38	3.20	1.07	32	4	2	33281
1.43	2.58	1.01	32	4	3	33281
1.51	3.21	1.10	64	2	1	33409
1.43	**2.22**	1.00	**64**	**2**	**2**	**49921**
1.51	3.34	0.94	64	2	3	49921
1.55	3.24	2.09	64	3	1	49857
1.32	2.53	0.88	64	3	2	86913
1.25	2.57	**0.73**	**64**	**3**	**3**	**86913**
1.43	5.78	1.43	64	4	1	66305
1.28	2.26	0.93	64	4	2	132097
1.32	2.46	1.15	64	4	3	132097

Table 5. Hyperparameter selection results for each problem of the BIDMC dataset. The bold values denote the selected hyperparameters for each vitals sign problem. Note that RR and SpO2 had the same parameters selected, hence why only two lines are given in bold.

Validation loss			Hidden dim	Total params
RR	HR	SpO2		
3.00	**12.82**	**3.37**	32	3871
3.00	12.82	3.37	64	9759
2.82	12.82	3.37	128	27679
2.49	12.82	3.37	192	53791
2.52	12.82	3.37	256	88095
2.50	12.82	3.37	320	130591
2.83	12.82	3.37	388	184719

Table 6. Hyperparameter selection results for the folded ODE-RNN model on the BIDMC problem. Bold values indicate selected hyperparamter values. The ODE-RNN model failed to train effectively for the HR and SpO2 problems which is why the validation losses are the same (to 2dp).

Model	Step	Test Accuracy	Time (Hrs)	Memory (Mb)
ODE-RNN (folded)	1	Memory Error	Memory Error	Memory Error
	2	36.8 ± 1.5	1.6	7170.1
	4	35.0 ± 1.5	0.8	3629.3
	6	36.8 ± 1.5	0.5	2448.6
	8	36.8 ± 1.5	0.4	1858.8
	16	32.5 ± 3.0	0.2	973.5
	32	32.5 ± 1.5	0.1	532.2
	64	41.0 ± 4.4	0.1	311.2
	128	47.9 ± 5.3	0.0	200.8
	256	46.2 ± 0.0	0.0	147.0
	512	47.9 ± 10.4	0.0	124.5
	1024	44.4 ± 7.4	0.0	122.4
	2048	48.7 ± 6.8	0.0	137.2
NCDE	1	62.4 ± 12.1	22.0	176.5
	2	69.2 ± 4.4	14.6	90.6
	4	66.7 ± 11.8	5.5	46.6
	6	65.8 ± 12.9	2.6	31.5
	8	64.1 ± 13.3	3.1	24.3
	16	64.1 ± 16.8	1.5	13.4
	32	64.1 ± 14.3	0.5	8.0
	64	56.4 ± 6.8	0.4	5.2
	128	48.7 ± 2.6	0.1	3.9
	256	42.7 ± 3.0	0.1	3.2
	512	44.4 ± 5.3	0.0	2.9
	1024	41.9 ± 14.6	0.0	2.7
	2048	38.5 ± 5.1	0.0	2.6
$NRDE_2$	2	**76.1 ± 13.2**	9.8	354.3
	4	**83.8 ± 3.0**	2.4	180.0
	6	**76.9 ± 6.8**	2.0	82.2
	8	**77.8 ± 5.9**	2.1	94.2
	16	**78.6 ± 3.9**	1.3	50.2
	32	67.5 ± 12.1	0.7	28.1
	64	73.5 ± 7.8	0.4	17.2
	128	**76.1 ± 5.9**	0.2	7.8
	256	**72.6 ± 12.1**	0.1	8.9
	512	**69.2 ± 11.8**	0.0	7.6
	1024	**65.0 ± 7.4**	0.0	6.9
	2048	**67.5 ± 3.9**	0.0	6.5
$NRDE_3$	2	66.7 ± 4.4	7.4	1766.2
	4	76.9 ± 9.2	2.8	856.8
	6	70.9 ± 1.5	1.4	606.1
	8	70.1 ± 6.5	1.3	460.7
	16	73.5 ± 3.0	1.4	243.7
	32	**75.2 ± 3.0**	0.6	134.7
	64	**74.4 ± 11.8**	0.3	81.0
	128	68.4 ± 8.2	0.1	53.3
	256	60.7 ± 8.2	0.1	40.2
	512	62.4 ± 10.4	0.0	33.1
	1024	59.8 ± 3.9	0.0	29.6
	2048	61.5 ± 4.4	0.0	27.7

Table 7. Mean and standard deviation of test set accuracy (in %) over three repeats, as well as memory usage and training time, on the EigenWorms dataset for depths 1–3 and a small selection of step sizes. The bold values denote that the model was the top performer for that step size.

Depth	Step	L^2			Time (H)			Memory (Mb)
		RR	HR	SpO_2	RR	HR	SpO_2	
ODE-RNN (folded)	1	Error	13.06 ± 0.0	Error	Error	10.4	Error	3654.0
	2	Error	13.06 ± 0.0	Error	Error	5.5	Error	1840.4
	4	2.76 ± 0.14	13.06 ± 0.0	3.3 ± 0.0	3.0	2.7	2.1	1809.0
	8	2.47 ± 0.35	13.06 ± 0.0	3.3 ± 0.0	1.5	1.2	0.9	917.2
	16	2.21 ± 0.75	13.06 ± 0.0	3.3 ± 0.0	2.2	0.7	0.4	471.9
	32	1.82 ± 0.64	13.06 ± 0.0	3.3 ± 0.0	0.7	0.3	0.2	249.4
	64	1.6 ± 0.22	13.06 ± 0.0	3.3 ± 0.0	0.5	0.1	0.1	137.0
	128	1.62 ± 0.07	13.06 ± 0.0	3.3 ± 0.0	0.2	0.1	0.1	81.9
	256	1.57 ± 0.04	7.04 ± 1.04	$\mathbf{1.43 \pm 0.11}$	0.1	0.1	0.1	53.8
	512	1.66 ± 0.06	6.75 ± 0.9	1.98 ± 0.31	0.0	0.1	0.1	40.4
	1024	$\mathbf{1.69 \pm 0.02}$	8.4 ± 0.28	2.05 ± 0.14	0.0	0.0	0.0	36.2
	2048	1.75 ± 0.03	9.2 ± 0.27	$\mathbf{2.24 \pm 0.11}$	0.0	0.0	0.0	39.6
NCDE	1	2.79 ± 0.04	9.82 ± 0.34	2.83 ± 0.27	23.8	22.1	28.1	56.5
	2	2.87 ± 0.03	11.69 ± 0.38	$\mathbf{3.36 \pm 0.2}$	19.3	9.6	8.8	32.6
	4	$\mathbf{2.92 \pm 0.08}$	11.15 ± 0.49	3.69 ± 0.06	5.3	5.7	3.2	20.2
	8	2.8 ± 0.06	10.72 ± 0.24	3.43 ± 0.17	3.0	2.6	4.8	14.3
	16	2.22 ± 0.07	7.98 ± 0.61	2.9 ± 0.11	1.7	1.4	1.8	11.8
	32	2.53 ± 0.23	12.23 ± 0.43	2.68 ± 0.12	1.9	0.9	2.2	9.8
	64	2.63 ± 0.11	12.02 ± 0.09	2.88 ± 0.06	0.2	0.3	0.4	9.1
	128	2.64 ± 0.18	11.98 ± 0.37	2.86 ± 0.04	0.2	0.2	0.3	8.7
	256	2.53 ± 0.04	12.29 ± 0.1	3.08 ± 0.1	0.1	0.1	0.1	8.3
	512	2.53 ± 0.03	12.22 ± 0.11	2.98 ± 0.04	0.1	0.0	0.1	8.4
	1024	2.67 ± 0.12	11.55 ± 0.03	2.91 ± 0.12	0.1	0.1	0.1	8.4
	2048	2.48 ± 0.03	12.03 ± 0.2	3.25 ± 0.01	0.0	0.1	0.0	8.2
$NRDE_2$	2	2.91 ± 0.1	11.11 ± 0.23	3.89 ± 0.44	12.7	9.3	8.2	58.3
	4	$\mathbf{2.92 \pm 0.04}$	11.14 ± 0.2	4.23 ± 0.57	18.1	5.0	3.4	34.0
	8	2.63 ± 0.12	8.63 ± 0.24	2.88 ± 0.15	2.1	3.4	3.3	21.8
	16	1.8 ± 0.07	5.73 ± 0.45	1.98 ± 0.21	2.2	1.4	2.5	16.0
	32	1.9 ± 0.02	7.9 ± 1.0	1.69 ± 0.2	1.2	1.1	2.0	13.1
	64	1.89 ± 0.04	5.54 ± 0.45	2.04 ± 0.07	0.3	0.3	1.7	11.6
	128	1.86 ± 0.03	6.77 ± 0.42	1.95 ± 0.18	0.3	0.4	0.7	10.9
	256	1.86 ± 0.09	5.64 ± 0.19	2.1 ± 0.19	0.1	0.1	0.5	10.5
	512	1.81 ± 0.02	5.05 ± 0.23	2.17 ± 0.18	0.1	0.2	0.4	10.3
	1024	1.93 ± 0.11	6.0 ± 0.19	2.41 ± 0.07	0.1	0.1	0.2	10.2
	2048	$\mathbf{2.03 \pm 0.03}$	$\mathbf{7.7 \pm 1.46}$	2.55 ± 0.03	0.1	0.1	0.1	10.2
$NRDE_3$	2	$\mathbf{2.82 \pm 0.08}$	$\mathbf{11.01 \pm 0.28}$	4.1 ± 0.72	8.8	9.4	6.9	125.2
	4	2.97 ± 0.23	$\mathbf{10.13 \pm 0.62}$	$\mathbf{3.56 \pm 0.44}$	3.2	4.1	2.6	71.6
	8	$\mathbf{2.42 \pm 0.19}$	$\mathbf{7.67 \pm 0.4}$	$\mathbf{2.55 \pm 0.13}$	2.9	3.2	3.1	43.3
	16	$\mathbf{1.74 \pm 0.05}$	$\mathbf{4.11 \pm 0.61}$	$\mathbf{1.4 \pm 0.06}$	1.4	1.4	6.5	29.1
	32	$\mathbf{1.67 \pm 0.01}$	$\mathbf{4.5 \pm 0.7}$	$\mathbf{1.61 \pm 0.05}$	1.3	1.8	7.3	20.5
	64	$\mathbf{1.53 \pm 0.08}$	$\mathbf{3.05 \pm 0.36}$	$\mathbf{1.48 \pm 0.14}$	0.4	1.9	3.3	17.9
	128	$\mathbf{1.51 \pm 0.08}$	$\mathbf{2.97 \pm 0.45}^*$	$\mathbf{1.37 \pm 0.22}$	0.5	1.7	1.7	17.3
	256	$\mathbf{1.51 \pm 0.06}$	$\mathbf{3.4 \pm 0.74}$	1.47 ± 0.07	0.3	0.7	0.6	16.6
	512	$\mathbf{1.49 \pm 0.08}^*$	$\mathbf{3.46 \pm 0.13}$	$\mathbf{1.29 \pm 0.15}^*$	0.3	0.4	0.4	15.4
	1024	1.83 ± 0.33	$\mathbf{5.58 \pm 2.5}$	$\mathbf{1.72 \pm 0.31}$	0.2	0.1	0.1	15.7
	2048	2.31 ± 0.27	9.77 ± 1.53	2.45 ± 0.18	0.1	0.1	0.1	15.6

Table 8. Mean and standard deviation of the L^2 losses on the test set for each of the vitals signs prediction tasks (RR, HR, SpO_2) on the BIDMC dataset, across three repeats. Only mean times are shown for space. The memory usage is given as the mean over all three of the tasks as it was approximately the same for any task for a given depth and step. Error denotes that the model could not be run within GPU memory. The bold values denote the algorithm with the lowest test set loss for a fixed step size for each task.