Learning the Dynamics of Sparsely Observed Interacting Systems

Linus Bleistein, Adeline Fermanian, Anne-Sophie Jannot, Agathe Guilloux

Learning the Dynamics of Sparsely Observed Interacting Systems

Linus Bleistein [*1 2 3] **Adeline Fermanian** [*4 5 6 7] **Anne-Sophie Jannot** [1 8] **Agathe Guilloux** [1 2]

Abstract

We address the problem of learning the dynamics of an unknown non-parametric system linking a target and a feature time series. The feature time series is measured on a sparse and irregular grid, while we have access to only a few points of the target time series. Once learned, we can use these dynamics to predict values of the target from the previous values of the feature time series. We frame this task as learning the solution map of a controlled differential equation (CDE). By leveraging the rich theory of signatures, we are able to cast this non-linear problem as a high-dimensional linear regression. We provide an oracle bound on the prediction error which exhibits explicit dependencies on the individual-specific sampling schemes. Our theoretical results are illustrated by simulations which show that our method outperforms existing algorithms for recovering the full time series while being computationally cheap. We conclude by demonstrating its potential on real-world epidemiological data.

1. Introduction

Time series are ubiquitous in many areas such as finance, economics, robotics, agriculture, and healthcare. One is typically interested in modelling the evolution of a target quantity through time, which is known to be affected by a set of time-evolving features. For example, pollution levels in a city are driven by quantities such as temperature, pressure, traffic, or economic activity measured through time. Mathematically, one wishes to model the evolution of a quantity $y_t \in \mathbb{R}^p$, $p \geq 1$, as a function of some time evolving features $x_t \in \mathbb{R}^d$, $d \geq 1$, for $t \in [0, 1]$. In other words, the goal is to learn the dynamics that link the target to the features.

Such an interaction is typically modelled via differential equations, which are a common choice of model in natural sciences (Zwillinger, 1989). In this article, we assume that there exists a function $\mathbf{G} : \mathbb{R}^p \times \mathbb{R}^d \to \mathbb{R}^p$ such that

$$y_t = y_0 + \int_0^t \mathbf{G}(y_s, x_s) ds \tag{1}$$

or equivalently

$$dy_t = \mathbf{G}(y_t, x_t)dt, \quad y_0 \in \mathbb{R}^p.$$

The value y_t depends on the trajectory of the features time series x_s up to time t. Learning the dynamics of the system can be framed as learning the solution map of (1), i.e., a function Ψ which, given a time t, an initial point $y_0 \in \mathbb{R}^p$, and the history of the path up to time t, denoted by $x_{[0,t]} = (x_s)_{s\in[0,t]}$, outputs the value of y at time t.

If we know Ψ, we gain access to the values of y at any point in time provided we know the values of x up to this point ; this encompasses many tasks such as forecasting or interpolating between points of y. We specifically have in mind applications where we have an easy access to x but a limited one to y.

This problem is extremely common in healthcare. For example, in obstetrics, the lactic acidosis (LA) of the fetus, which is a proxy for fetal distress, is a quantity of high medical interest for predicting complications in the first hours after birth. This biomarker cannot be measured during pregnancy but only at birth because the measurement is highly invasive. Some vitals such as heart rate and fetal movement are however easy to measure during pregnancy. In this case, x is the non-invasive measurements made during pregnancy, while y is the invasive measurement of LA at birth. Predicting the value of y at any time t (both before and at birth) would allow for early diagnosis. Similarly, after surgery, patients are often monitored to detect hemorrhage. While some vitals such as heart rate of saturation are monitored in continuous time, haemoglobin—which is highly predictive of hemorrhage—is only measured by

[*]Equal contribution [1]Inria Paris, F-75015 Paris, France [2]Centre de Recherche des Cordeliers, INSERM, Université de Paris, Sorbonne Université, F-75006 Paris, France [3]LaMME, UEVE and UMR 8071, Paris Saclay University, F-91042, Evry, France [4]MINES ParisTech, PSL Research University, CBIO, F-75006 Paris, France [5]Institut Curie, PSL Research University, F-75005 Paris, France [6]INSERM, U900, F-75005 Paris, France [7]LOPF, Califrais' Machine Learning Lab, Paris, France [8]APHP, Paris, France. Correspondence to: Linus Bleistein <linus.bleistein@inria.fr>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.

blood samples taken a few times a day, which can significantly delay hemorrhage diagnostic.

Irregular data. In practice, the functions x and y are measured on discrete grids and take the form of time series. These often present a lot of heterogeneity, both within and across individuals.

(i) For every individual, the time between any two measurements can vary, and thus individuals may not be recorded on the same grid.

(ii) The number of total sampling points might vary between individuals.

(iii) Each measurement in time might be corrupted by measurement noise.

Mathematically, we consider n pairs of functions $\{(x^1, y^1), \dots, (x^n, y^n)\}$. Each x^i deterministically produces a specific y^i through the Ordinary Differential Equation (ODE) (1). We call x^i the feature path and y^i the target path. Both x^i and y^i are only observed at a finite set of times specific to every individual. We denote by

$$D^i = \left(t_1^i, \dots, t_{k_i}^i\right), \quad i = 1, \dots, n,$$

the sampling grid of x^i and by $\bar{D}^i$ the sampling grid of y^i. We stress that both the number of sampling times k_i and the sampling times $t_1^i, \dots, t_{k_i}^i$ themselves are individual specific, as described in (i) and (ii). Moreover, the observations are corrupted by additive noise, such that we observe

$$X_t^i = x_t^i + \xi_t^i$$

for all $t \in D^i$, and similarly $Y_t^i = y_t^i + \varepsilon_t^i$ for every $t \in \bar{D}^i$, where the ξ_t^i and ε_t^i are sub-gaussian i.i.d. random vectors. Each input may therefore be written as a matrix $\mathbf{X}^i = (X_t^i)_{t \in D^i} \in \mathbb{R}^{k_i \times d}$ which we call the feature time series. Similarly, the quantity of interest is a matrix $\mathbf{Y}^i = (Y_t^i)_{t \in \bar{D}^i} \in \mathbb{R}^{m_i \times p}$ (where m_i is the length of $\bar{D}^i$) and is called the target time series. The grid $\bar{D}^i$ is assumed to be a subset of D^i: in our setup y^i is hard to sample and therefore measured at only a few points (and sometimes only one) while x^i is easy to access and measured at high frequency. Our goal is to approximate the dynamics linking x and y from the irregular, heterogeneous, and fuzzy data $\mathbf{X}^i$ and $\mathbf{Y}^i$.

Such heterogeneity is difficult to handle by classical machine learning algorithms such as Long short-term memory networks (LSTM, Hochreiter & Schmidhuber, 1997) which assume that the data is regularly sampled. Some more recent approaches (Rubanova et al., 2019; De Brouwer et al., 2019; Kidger et al., 2020; Herrera et al., 2021) have adapted these models by introducing continuously evolving hidden states to account for the irregular spacing between observation times.

We build upon the approach of Neural Controlled Differential Equations (Neural CDE, Kidger et al., 2020; Morrill et al., 2021b), which have proven to be very successful for time series classification and online prediction tasks (Morrill et al., 2021a). The key idea of Neural CDE is that under some fairly mild assumptions, any general ordinary differential equation of the form (1) can be rewritten as

$$y_t = y_0 + \int_0^t \mathbf{F}(y_s) dx_s, \tag{2}$$

where $\mathbf{F} : \mathbb{R}^p \to \mathbb{R}^{p \times d}$ is a matrix-valued vector field, such that the right-hand-side of (2) is a matrix-vector product (see, e.g., Fermanian et al., 2021, Proposition 2, for a proof). The function x is often called the driver of the CDE. In a Neural CDE setting, the driver x is a continuous interpolation of the feature times series, y corresponds to a continuously-evolving state, and $\mathbf{F}$ is chosen to be a neural network. This network is then trained such that the values of (y_t) can be used as features for classification or regression tasks. While Neural CDE have been shown to outperform other architectures with limited memory usage, their training time is considerable and no statistical guarantees exist.

Model. We model the interactions between the target and the feature paths through a CDE of the form (2). This modelling choice encapsulates a broad variety of settings, since the vector field $\mathbf{F}$ can be any (regular enough) function. A priori, the solution map Ψ of this CDE is a complex function of time and the history of x up to t; however, by linearizing the model, we are able to approximate Ψ by a simple scalar product between a deterministic transformation of the history of x, called the signature of x at order $N \geq 1$ and denoted by $S_N(x_{[0,t]})$, and a time independent matrix θ_N^*. Informally, we have

$$\Psi\left(x_{[0,t]}, t\right) \approx S_N\left(x_{[0,t]}\right)^\top \theta_N^*.$$

Two striking features of this linearized model are *(i)* that θ_N^* can be learned on any time horizon $[0, t]$ since it is independent of time, and *(ii)* that once it has been learned, the model can be called at any time t.

Contributions. Our contributions are threefold. First, we frame the task of learning the interactions between two time series as learning the flow of a CDE, which can be linearized in the signature space. While the connection between CDEs and signatures is well-known, this is the first time CDEs are used as a statistical model. We then leverage this linearization to derive statistical guarantees on the prediction error with an explicit dependence on both sampling irregularities and the noise affecting measurements.

To our knowledge, this is the first bound of this type for signature-based models, allowing for better understanding of the dependencies between prediction performance and sampling roughness. Finally, the resulting algorithm, called SigLasso, is shown to be computationally cheap and competitive compared to existing baselines on a wide range of simulated data and a real-world example of hospitalization growth rate prediction during the Covid pandemic.

Related works. Signatures originated as a prominent tool in stochastic analysis (Chen, 1958; Lyons et al., 2007; Friz & Victoir, 2010) and have proven to be a powerful feature extraction method in machine learning in various domains such as healthcare (Morrill et al., 2020b; Wang et al., 2020), human action recognition (Yang et al., 2022), or financial modelling (Lyons et al., 2014; Buehler et al., 2020). Their appealing properties include a capacity to handle irregular data, to capture dependence between coordinates, and their links with the theory of CDE. We refer to Lyons & McLeod (2022) for a recent survey on their use cases. However, the statistical properties of signatures based algorithms have received little attention so far, with a few notable exceptions (Papavasiliou & Ladroue, 2011; Lemercier et al., 2021; Fermanian, 2022).

On the other hand, the interplay between dynamical systems and machine learning has received considerable attention in the recent years. A first line of work has focused on approximating the solution of ODE and Partial Differential Equations (PDE) with neural networks (Lagaris et al., 1998; Han et al., 2018; Zubov et al., 2021) and directly learning dynamical systems (Long et al., 2018; Fattahi et al., 2019). Recent approaches have been interested in combining deep learning algorithms with physical knowledge (Greydanus et al., 2019; Brunton et al., 2020; Willard et al., 2020). Finally, dynamical systems, seen as continuous versions of neural network architectures, have also been a great source of inspiration for analysing and designing machine learning algorithms in the recent years (Chen et al., 2018; Fermanian et al., 2021; Marion et al., 2022). We refer to Kidger (2022) for an extensive review.

We stress that our problem is different in nature from most problems encountered in the time series literature, since we seek to model the relationship between two sparsely observed systems with heterogeneous sampling. We do not model the evolution of the feature time series, and take it as an input, contrarily to methods such as Gaussian Process. Most models either focus on the case where one time series is observed and forecasted, or on regular sampling, or on univariate time series. Our problem bears close resemblance to frameworks encountered in sequence-to-sequence learning (Sutskever et al., 2014; Gehring et al., 2017) and functional regression (Ramsay & Dalzell, 1991; Marx & Eilers, 1999).

Overview. Section 2 introduces the CDE model for interacting systems, the mathematical context and the learning procedure. Our main theoretical result is presented in Section 3. We conclude by an empirical study on synthetic and real-world data in Section 4. All proofs are postponed to the appendix and the code to reproduce the experiments is available at https://github.com/LinusBleistein/SigLasso.

2. Model and Assumptions

A summary table of all notations introduced in the main body of this article is provided to the reader in Appendix A.

2.1. A CDE-Based Model on the Dynamics

We start by describing our assumptions on the feature and target paths, which are linked by Equation (2). To correctly define the integral of Equation (2), we need to impose some conditions on the x^i and on $\mathbf{F}$. Note that we consider that the x^i are defined on $[0,1]$ but our results extend easily to any compact time interval $[a,b]$.

Assumption 1. All paths $(x^i)_{1\leq i\leq n}$ are continuous and there exists $0<L<1$ such that, for all $i=1,\dots,n$,

$$\left\|x^i\right\|_{1\text{-var},[0,1]} = \sup_D \sum_k \left\|x^i_{t_{k+1}} - x^i_{t_k}\right\| \leq L,$$

where $\|\cdot\|$ is the Euclidean norm and the supremum is taken over all finite dissections $D=\{0=t_1<\cdots<t_k=1\}$.

We write $C_L^{1\text{-var}}([0,1],\mathbb{R}^d)$ for the set of continuous paths of total variation bounded by L. Outside the statistical context, when referring to general paths, we will drop the superscripts i and simply write x and y to alleviate notations.

We assume that the target path y is the solution of the ODE (1). This modelization choice means that the evolution of y is governed by a dynamical system whose dynamics itself are allowed to vary with the current value of the feature path. Observe that this model can be seen as a generalized form of a non-autonomous system (Lyons et al., 2007), which we recover by taking $x_t = t$. Since Equation (1) can be rewritten as a CDE, the starting point of our work is to assume that the true dynamics of the data follow such a CDE, as stated in the following assumption.

Assumption 2. There exists a smooth vector field $\mathbf{F}: \mathbb{R}^p \to \mathbb{R}^{p\times d}$ such that, for all $i=1,\dots,n$, y^i is the solution of the CDE (2) driven by x^i with initial condition $y^i_0 = y_0 \in \mathbb{R}^p$ homogeneous amongst individuals.

By "smooth" we mean that each coordinate of $\mathbf{F}$ is infinitely differentiable, that is, is $\mathcal{C}^\infty$. The vector field $\mathbf{F}$ and the initial condition y_0 are common to all individuals, which can be seen as homogeneity assumptions on our

sample. On the other hand, since every individual i has her own feature path x^i, the target paths y^i are individual specific. In other words, there exists a solution map Ψ that depends only on y_0 and $\mathbf{F}$ and is such that, for any $t \in [0,1]$, $\Psi(x^i_{[0,t]}, t) = y^i_t$.

The vector field $\mathbf{F}$ encapsulates the common physical dynamics governing the evolution of y^i, which are affected by the changes in x^i. Note that there is no parametric model on $\mathbf{F}$ (although some strong smoothness requirements will be needed) contrarily to functional or traditional time series models (Ramsay & Silverman, 2005; Morris, 2015).

2.2. Linearizing the CDE with Signatures

Before defining the Taylor expansion of the CDE (2), which will allow us to linearize the solution map Ψ, we need to introduce the notion of signature, which have emerged as a powerful tool to model time series (Levin et al., 2013; Kidger et al., 2019).

From now on, for any feature path x with values in $\mathbb{R}^d$, we denote by $x^{(j)}$ its jth coordinate, for $j = 1, \dots, d$.

Definition 2.1. Let $x \in C_L^{1\text{-var}}([0,1], \mathbb{R}^d)$. Take a word of length k from the alphabet $\{1, \dots, d\}$, that is, an element $I = (i_1, \dots, i_k) \in \{1, \dots, d\}^k$. For all $t \in [0,1]$, the signature coefficient associated to this word is the scalar

$$S^I\left(x_{[0,t]}\right) = \int_{0<u_1<\cdots<u_k<t} \cdots \int dx^{(i_1)}_{u_1} \cdots dx^{(i_k)}_{u_k}.$$

We introduce a series of notation on signature coefficients, grouping them by the length k of the words. For any $k \in \mathbb{N}$ and $t \in [0,1]$, the signature of order k of $x_{[0,t]}$ is

$$\mathbb{X}_{k,[0,t]} = \left(S^I(x_{[0,t]})\right)_{I \in \{1,\dots,d\}^k} \in \mathbb{R}^{d^k},$$

where the words $(i_1, \dots, i_k)$ are in lexicographic order. We then denote the full signature by

$$S(x_{[0,t]}) = \left(1, \mathbb{X}^1_{[0,t]}, \dots, \mathbb{X}^k_{[0,t]}, \dots\right)$$

and, for any $N \geq 1$, the signature truncated at order N by

$$S_N(x_{[0,t]}) = \left(1, \mathbb{X}^1_{[0,t]}, \dots, \mathbb{X}^N_{[0,t]}\right)^\top.$$

The size of the signature truncated at order N grows exponentially with N, since, for $d \geq 2$, it is equal to

$$s_d(N) = 1 + d + d^2 + \cdots + d^N = \frac{d^{N+1} - 1}{d - 1}.$$

When computing signatures, it is common practice to add time as a coordinate to the path (Fermanian, 2021), that is, consider the path $(t, x_t)^\top$. From now on, we assume that the first dimension of x always corresponds to time, so that $d \geq 2$.

Signatures encode geometric properties of paths and have numerous appealing properties as a feature set for time series. We refer to Chevyrev & Kormilitzin (2016); Fermanian (2021); Lyons & McLeod (2022) for more detailed introductions to signatures. In order to provide supplementary intuition, we first give three insights on signatures.

A geometric insight. Signatures are a geometric alternative to representations based on frequency such as the Fourier transform. Indeed, consider a function $f \in C^\infty([0,1],[0,1])$ and the two dimensional path $(t, f(t))$. For simplicity assume that $f(0) = 0$. Then, the first order signature coefficients are equal to the last positions t and $f(t)$. The second order signature coefficients are equal to $\int_0^t f(s)ds$, $\frac{1}{2}t^2$, $\frac{1}{2}f(t)^2$, and $f(t)t - \int_0^t f(s)ds$, and capture how the area under the curve evolves with time.

A computational insight. Consider the linear path $x_t = at + b$ for $t \in [0,1]$, where $a = (a_1, \dots, a_d)^\top \in \mathbb{R}^d$. For any word $(i_1, \dots, i_k)$ of size k the associated signature coefficient is

$$\frac{a_{i_1} \dots a_{i_k}}{k!} t^k.$$

In the linear case, signatures are therefore simply polynomials in t with path-specific coefficients. This result generalizes nicely to piecewise linear paths (via a result known as Chen's Lemma) and allows for computational efficiency when computing the signature. We refer to Kidger & Lyons (2020) for further computational details.

A functional insight. Recall that we are interested in approximating functions $f(x_t, t)$ which depend both on time and on the values of the feature path (x_t). When computing signatures, we always consider the time-augmented path (t, x_t). The coefficients will thus be divided in two parts: a first set of coefficients related to the time dimension, and a set of coefficients related to the path dimensions. The time-specific coefficients are simply

$$t, \frac{t^2}{2!}, \frac{t^3}{3!}, \dots, \frac{t^N}{N!} \tag{3}$$

and thus form a polynomial basis. Roughly speaking, these coefficients approximate the time dependant part of the function f. The path-specific coefficients, which can be thought of as polynomials of a path, approximate the part of f depending on (x_t). This highlights that the Taylor development of a CDE, which is the cornerstone of our work, is very similar in nature to the approximation of a function by its classical Taylor development (see Appendix B.4).

With these insights in hand, we are now ready to properly define the Taylor expansion of a CDE.

Definition 2.2. Let $N \geq 1$. The Taylor expansion of order

N of the solution y of Equation (2) is defined by

$$\overline{y}_{N,t} = y_0 + \sum_{k=1}^{N} \sum_{I \in \{1,\dots,d\}^k} S^I\left(x_{[0,t]}\right) \times \Phi_{\mathbf{F}}^I\left(y_0\right), \quad (4)$$

where $\Phi_{\mathbf{F}}^I(\cdot) \in \mathbb{R}^p$ is the differential product of the vector field $\mathbf{F}$ along I, whose definition is postponed to Appendix B.3.

The differential products $\Phi_{\mathbf{F}}^I(y_0)$ are essentially a combination of multiplication and summation of derivatives of the different components of $\mathbf{F}$, evaluated at y_0.

The Taylor expansion crucially allows to write the solution map as a product between a time-varying term (the signature of the feature path) and a constant-over-time term (the differential product). This is similar to a regular Taylor expansion; we discuss this analogy in Appendix B.4. Note that Equation (4) may also be written as a matrix-vector product

$$\overline{y}_{N,t}{}^{\top} = S_N(x_{[0,t]})^{\top} \theta_N^* \in \mathbb{R}^p, \quad (5)$$

where $\theta_N^* \in \mathbb{R}^{s_d(N) \times p}$ is a matrix collecting all differential products $\Phi_{\mathbf{F}}^I(y_0)$ up to order N and the offset y_0. Since θ_N^* depends neither on x nor on t, the Taylor expansion of y at order N is simply a linear function of the truncated signature.

For this expansion to become exact in the sens of pointwise convergence, we need quite strong regularity assumption on $\mathbf{F}$. This is the price to pay for the non-parametric nature of our model.

Assumption 3. The vector field $\mathbf{F}$ has fast decaying derivatives. In other words, defining

$$\Lambda_k(\mathbf{F}) = \sup_{I \in \{1,\dots,d\}^k} \left\| \Phi_{\mathbf{F}}^I\left(y_0\right) \right\|,$$

we assume that $\sum_{k=0}^{\infty} d^k \Lambda_k(\mathbf{F})/k! < \infty$.

Let y be the solution of the CDE (2), and let $(\overline{y}_{N,t})_{t\in[0,1]}$ be its N-th Taylor expansion. Under Assumptions 1, 2, and 3, we then have, for any $t \in [0,1]$,

$$\left\| y_t - \overline{y}_{N,t} \right\| \underset{N \to +\infty}{\longrightarrow} 0. \quad (6)$$

We refer the reader to Friz & Victoir (2008) and Fermanian et al. (2021, Proposition 4) for more details on this result.

While we impose relatively strong regularity assumptions on the vector field $\mathbf{F}$, the assumptions on x are mild (see Assumption 1). Therefore, our assumptions, while enforcing a fairly high amount of regularity on the dynamical structure, still accommodate most of the real world data. Also note that Fermanian et al. (2021) give conditions under which Assumption 3 is satisfied when $\mathbf{F}$ is a layer of a neural network with smooth activation functions.

2.3. The Learning Problem

We go back to the statistical learning problem. Recall that in practice, we do not observe the continuous paths $\{(x^1, y^1), \dots, (x^n, y^n)\}$ but their discretized and fuzzy counterparts $\{(\mathbf{X}^1, \mathbf{Y}^1), \dots, (\mathbf{X}^n, \mathbf{Y}^n)\}$ measured on a set of individual grids $\mathcal{D} = \{D^1, \dots, D^n\}$ and $\bar{\mathcal{D}} = \{\bar{D}^1, \dots, \bar{D}^n\}$. We allow for high data heterogeneity: two individuals can be sampled at very different frequencies and at different time-points, and therefore have different observation grids.

The meshsize of a sampling grid D, denoted by $|D|$ is defined as the largest gap between two successive sampling times, that is,

$$|D| = \max_{t_i \in D} |t_{i+1} - t_i|.$$

Its cardinality, denoted by $\#D$, is the number of sampling points in D. We make the following assumption on the sampling procedure.

Assumption 4. There exists $\eta \in [0,1]$ such that for all $i = 1, \dots, n$, one has

$$0 \in D^i, \quad \#D^i \geq 2, \quad t_{k_i}^i \geq \eta \quad \text{and} \quad \bar{D}^i \subset D^i.$$

Let us briefly comment on this assumption. We require that the measurements on all individuals start at 0. We also do not allow for arbitrarily short time series: for every individual i, the last observation time $t_{k_i}^i$ must be greater than η. Finally, we require the sampling grid of $\mathbf{Y}^i$ to be coarser than the sampling grid of $\mathbf{X}^i$. We also let $|\mathcal{D}| = \max_i |D^i|$ be the biggest gap between two successive sampling times within the whole set of individual grids and $\#\mathcal{D} = \sum_{i=1}^{n} \#D^i$ the total number of sampling points of the feature time series. We recall that the random vectors ξ_t^i, $t \in D^i$, are the noises affecting the measurements of x^i. The random vectors ε_t^i affect the measurements of y^i. We end with assumptions on the law of these measurement noises.

Assumption 5 (Noise on the feature time series)**.** The noises $(\xi_t^i)_{i\in\{1,\dots,n\}, t\in D^i}$ are i.i.d. v_ξ-subgaussian random vectors.

Assumption 6 (Noise on the target time series)**.** The noises $(\varepsilon_t^i)_{i\in\{1,\dots,n\}, t\in \bar{D}^i}$ are i.i.d. v_ε-subgaussian random vectors and independent from $(\xi_t^i)_{i\in\{1,\dots,n\}, t\in D^i}$.

The goal of the learning procedure is to learn the solution map Ψ, in order to infer the value of y at any time t, given observations of x up to time t. We have seen previously that this problem boils down to estimating the matrix θ_N^* via Equation (5). The truncation order $N \geq 1$ is an hyperparameter and will be selected by cross-validation.

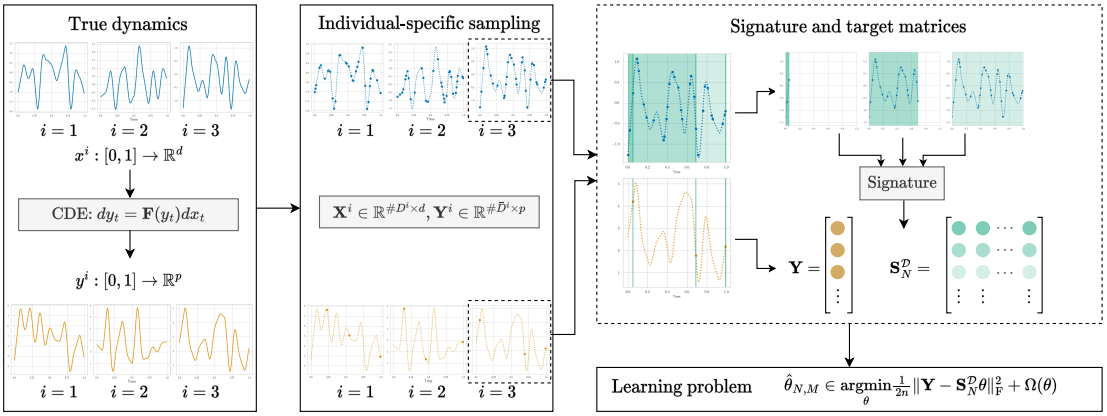

Figure 1: The workflow of our model. Starting from the left, the first panel describes our modelling hypothesis: the target and feature time series are linked through an unknown CDE. The second panel shows the observed data. The third panel shows how every observation of the target is mapped to the signature of the corresponding path, how to construct the dataset $\mathbf{S}_N^{\mathcal{D}}, \mathbf{Y}$ from this data and finally how to learn the SigLasso estimator.

Single target measurement. For the sake of simplicity, we first present the case where Y^i is measured only at the end of the observation period, as in the example of LA measurements in obstetrics. In this case, for all $i = 1, \dots, n$, we have $m_i = 1$ such that $M = n$,

$$D^i = (t_1^i, \dots, t_{k_i}^i), \quad \text{and} \quad \bar{D}^i = (t_{k_i}^i).$$

Then $\mathbf{Y}^i \in \mathbb{R}^p$, and we denote by

$$\mathbf{Y} = \left[\mathbf{Y}^1, \dots, \mathbf{Y}^n\right]^\top \in \mathbb{R}^{n \times p}$$

the matrix containing all target measurements. Since we do not have access to the feature paths x^i but only to the discrete measurements $\mathbf{X}^i$, we compute the signature of its linear interpolation normalized by its total variation, sampled up to final time $t_{k_i}^i$, and denote by $\mathbf{S}_N^{\mathcal{D}} \in \mathbb{R}^{n \times s_d(N)}$ the matrix of stacked signatures. Note that signatures of piecewise linear functions are fast to compute with packages such as `signatory` (Kidger & Lyons, 2020) or `iisignature` (Reizenstein & Graham, 2020). The complexity to compute the signature truncated at order N of the ith feature time series is of the order $\mathcal{O}(\#D^i d^N)$. Finally, we approximate $\theta_N^* \in \mathbb{R}^{s_d(N) \times p}$ by solving the optimisation problem

$$\min_{\theta \in \mathbb{R}^{s_d(N) \times p}} \frac{1}{2n} \left\|\mathbf{Y} - \mathbf{S}_N^{\mathcal{D}}\theta\right\|_{\mathrm{F}}^2 + \Omega(\theta), \tag{7}$$

where $\Omega : \mathbb{R}^{s_d(N) \times p} \to \mathbb{R}^+$ is a regularization term and $\|\cdot\|_{\mathrm{F}}$ is the Frobenius norm. We have reduced the complex problem of learning the solution map Ψ to a simple penalized linear regression in the signature space. This linear model on the signature is close to the one studied by Levin et al. (2013); Lyons et al. (2014); Fermanian (2022).

Multiple target measurements. We also cover the case when $\#\bar{D}^i > 1$, that is, the target is measured at multiple times for every individual, as in the example of hemorrhage detection. In this case, we have $\mathbf{Y}^i \in \mathbb{R}^{m_i \times p}$ and we stack the different measurement matrices $\mathbf{Y}^i$ to obtain a matrix $\mathbf{Y}$ of size $M \times p$, where $M = m_1 + \dots + m_n$. For any $i = 1, \dots, n$ and every $t \in \bar{D}^i$, we predict Y_t^i using the signature of the linear interpolation of the normalized $(X_0^i, \dots, X_t^i)$. In this manner, we will be able to predict Y_t^i at every point where X_t^i is sampled. The exact workflow of our model is described in Figure 1.

3. Theoretical Guarantees

3.1. Mathematical Setup

We consider a general multiple target measurements setting. To simplify the exposure of our results, we consider a univariate target path, i.e., $p = 1$. In this case, the true parameter θ_N^* is a vector of size $s_d(N)$ and not a matrix. The general case $p \geq 1$, which our algorithm handles as running p Lasso regressions in parallel, is considered in Appendix C, and all theoretical results are proved in this general case. In addition, to lighten the presentation of the oracle inequality, we also focus in this section on the case of ω-Lipschitz feature paths, that is, for every $i = 1, \dots, n$ and for all $s, t \in [0,1]$, $\left\|x_t^i - x_s^i\right\| \leq \omega|t - s|$. We stress that our results are valid for continuous paths of bounded variations. We let $\mathbf{y} \in \mathbb{R}^M$ be the matrix collecting all unobserved values of the target paths at measurement times such that $\mathbf{y} = \mathbb{E}(\mathbf{Y})$, where the expectation is taken over

the noises ε_t^i, and define $\widehat{\theta}_{N,M}$ as

$$\widehat{\theta}_{N,M} \in \underset{\theta \in \mathbb{R}^{s_d(N)}}{\arg\min} \frac{1}{2M} \left\| \mathbf{Y} - \mathbf{S}_N^{\mathcal{D}} \theta \right\|_2^2 + \Omega(\theta). \quad (8)$$

For $\delta \in (0,1)$, we define the set

$$A_\xi(\delta) = \left\{ \max \left\| \xi_t^i \right\| \leq \underbrace{v_\xi \sqrt{d} + v_\xi \sqrt{\frac{1}{c} \log \frac{\#\mathcal{D}}{\delta}}}_{=:C_\delta} \right\} \quad (9)$$

where the maximum is taken on all $i = 1, \dots, n$ and $t \in D^i$, and c is a universal constant. This set is of probability greater than $1 - \delta$ under Assumption 5 (see Appendix B.5). Similarly, for $k \geq 0$ and $\bar{\delta} \in (0,1)$, let

$$C_k(\bar{\delta}) = \sqrt{v_\varepsilon \log(2Nd^k/\bar{\delta})}$$

and define

$$A_\varepsilon(\bar{\delta}) = \bigcap_{k=0}^{N} \left\{ \left\| \boldsymbol{\varepsilon}^\top \mathbf{S}_{\cdot,[k]}^{\mathcal{D}} \right\|_\infty \leq \frac{M^{\frac{1}{2}} C_k(\bar{\delta})}{k!} \right\}, \quad (10)$$

where $\mathbf{S}_{\cdot,[k]}^{\mathcal{D}}$ is the sub-matrix of size $M \times d^k$ of $\mathbf{S}_N^{\mathcal{D}}$ associated to the signature coefficients of order k, and $\boldsymbol{\varepsilon} \in \mathbb{R}^M$ is a vector of i.i.d. noise terms satisfying Assumption 6 (see Appendix C.1). Under Assumptions 5 and 6, $A_\varepsilon(\bar{\delta})$ is of probability at least $1 - \bar{\delta}$, and $A_\xi(\delta) \cap A_\varepsilon(\bar{\delta})$ is of probability at least $(1-\delta)(1-\bar{\delta})$. Let

$$\Omega(\theta) = \sum_{k=0}^{N} \frac{C_k(\bar{\delta})}{k! \sqrt{M}} \left\| \theta_{[k]} \right\|_1, \quad (11)$$

where $\theta_{[k]}$ is the subvector of size d^k that collects all elements of θ associated to words of size k (see Appendix C.1). This penalization can be implemented by rescaling the feature matrix $\mathbf{S}_N^{\mathcal{D}}$ and solving a standard ℓ_1-penalized regression problem (see Appendix D.1). Our result extends to more general penalties by adapting existing techniques from Chesneau & Hebiri (2008) for the group-lasso, or Lederer et al. (2019) for the hierarchical lasso.

3.2. Main Results

The error made when learning θ_N^* by $\widehat{\theta}_{N,M}$ comes from three different sources. (i) Truncating the signature used in the regression at depth $N \geq 1$ results in a truncation bias. (ii) Discretization of the feature path and the noise affecting each measurement point induce a discretization error. In particular, there is a trade-off between sampling frequency and variance of the noise. (iii) The measurement error on $\mathbf{y}^i$ and the finite-sample setting induce a classical estimation error.

The following lemmas bound each of those errors. We first bound the variance of the estimator with arguments borrowed from Bickel et al. (2009).

Lemma 3.1. *Under Assumptions 1 and 2, on the set $A_\varepsilon(\bar{\delta}) \cap A_\xi(\delta)$, the prediction error*

$$\frac{1}{2M} \left\| \mathbf{y} - \mathbf{S}_N^{\mathcal{D}} \widehat{\theta}_{N,M} \right\|_2^2$$

is bounded above by

$$\frac{1}{2M} \left\| \mathbf{y} - \mathbf{S}_N^{\mathcal{D}} \theta_N^* \right\|_2^2 + \frac{2C_N(\bar{\delta})}{\sqrt{M}} \sum_{k=0}^{N} \frac{d^k \Lambda_k(\mathbf{F})}{k!}.$$

See Appendix C.2 for a proof. This inequality decomposes the error into a bias and a variance term. We denote the signature matrix of the unobserved paths x^i by $\mathbf{S}_N \in \mathbb{R}^{M \times s_d(N)}$.

As for the bias term, notice that one can write

$$\frac{1}{2M} \left\| \mathbf{y} - \mathbf{S}_N^{\mathcal{D}} \theta_N^* \right\|_2^2 \leq \frac{1}{M} \underbrace{\left\| \mathbf{y} - \mathbf{S}_N \theta_N^* \right\|_2^2}_{\text{Truncation bias}} + \frac{1}{M} \underbrace{\left\| \mathbf{S}_N \theta_N^* - \mathbf{S}_N^{\mathcal{D}} \theta_N^* \right\|_2^2}_{\text{Discretization error}}.$$

Bounding each of these terms corresponds to, respectively, Lemmas 3.2 and 3.3. We stress that the truncation bias is of a different nature than the discretization error since it depends on a choice of hyperparameter while the latter is inherent to the data at hand.

Lemma 3.2. *Under Assumptions 1 and 2, for any $N \geq 1$,*

$$\frac{1}{M} \left\| \mathbf{y} - \mathbf{S}_N \theta_N^* \right\|_2^2 \leq \left(\frac{d^{N+1} \Lambda_{N+1}(\mathbf{F})}{(N+1)!} \right)^2.$$

Under Assumption 3, the right-hand-side decays exponentially fast with N. This lemma is an immediate consequence of Fermanian et al. (2021) (see Appendix C.3). We now turn to the error induced by the discretization of the feature path.

Lemma 3.3. *Under Assumptions 1, 4, and 5, on the set $A_\xi(\delta)$, one has*

$$\frac{1}{M} \left\| (\mathbf{S}_N - \mathbf{S}_N^{\mathcal{D}}) \theta_N^* \right\|_2^2 \leq C_{\mathcal{D},N}(\delta) \sum_{k=0}^{N} \frac{d^k \Lambda_k(\mathbf{F})^2}{k!^2},$$

where $C_{\mathcal{D},N}(\delta)$ is equal to

$$4e^2 L^2 N!^2 \times \left(\omega |\mathcal{D}| + C_\delta + \frac{1 - L + 2\#\mathcal{D} C_\delta}{\eta} \left(\|x_0\| + L + C_\delta \right) \right)^2.$$

This lemma relies on a fine analysis of the distance between two signature layers. The dependence of $C_{\mathcal{D},N}(\delta)$ on sampling mechanisms and noise is of particular interest. First,

Table 1: Performance of SigLasso, GRU and Neural CDE in different simulation settings, averaged over 10 iterations. In every setting, $n = 50$, $\#\bar{D}^i = 5$ for all $i = 1, \dots, n$ (and therefore $M = 250$).

	L_2 error			MSE on last point		
Setting	SigLasso	GRU	Neural CDE	SigLasso	GRU	Neural CDE
Well-specified	**0.13 ± 0.07**	1.05 ± 0.42	0.61 ± 0.38	**0.73 ± 0.56**	3.32 ± 1.60	1.46 ± 1.20
Ill-specified	**0.15 ± 0.02**	0.24 ± 0.11	0.29 ± 0.15	**0.09 ± 0.05**	0.19 ± 0.09	0.22 ± 0.15
OU	**0.01 ± 0.02**	0.05 ± 0.06	0.17 ± 0.12	0.018 ± 0.025	0.014 ± 0.020	**0.013 ± 0.016**
Tumor growth	**0.16 ± 0.02**	0.66 ± 0.09	5.29 ± 1.38	**0.35 ± 0.12**	2.00 ± 0.38	8.76 ± 9.26

the term $\omega|\mathcal{D}|$ refers to the longest time between two observations amongst individuals. Not sampling an individual during a long period of time causes a loss in information, which is bounded by the Lipschitz control ω of the feature path.

The second part of $C_{\mathcal{D},N}(\delta)$ is a consequence of the noises ξ_t^i affecting the measurement points of the feature time series and does not vanish with n. The emergence of such a bias is a well studied phenomenon in errors-in-variable models, and cannot be corrected without precise knowledge of the noise's variance (Loh & Wainwright, 2011).

The last term corresponds to a bias coming from the interplay of the normalisation of the signatures by the total variation of the path and the noise, which is a standard practice (Morrill et al., 2020a). We found that this normalization performs best empirically. Note that this bias term is equal to 0 if the path has total variation exactly equal to 1 and is observed without noise.

From Lemmas 3.1 to 3.3 and the definitions of A_ξ and A_ε finally we get the following oracle inequality. The proof is given in Appendix C.6.

Theorem 3.4 (An oracle inequality for learning with signatures)**.** *Under Assumptions 1, 2, 4, 5, and 6, with probability at least $(1-\delta)(1-\bar{\delta})$, the prediction error is bounded above by*

$$\frac{1}{2M}\left\|\mathbf{y} - \mathbf{S}_N^{\mathcal{D}}\widehat{\theta}_{N,M}\right\|_2^2 \leq \left(\frac{d^{N+1}\Lambda_{N+1}(\mathbf{F})}{(N+1)!}\right)^2 \tag{12}$$

$$+ C_{\mathcal{D},N}(\delta)\sum_{k=0}^{N}\frac{d^k\Lambda_k(\mathbf{F})^2}{k!^2} \tag{13}$$

$$+ \frac{2C_N(\bar{\delta})}{\sqrt{M}}\sum_{k=0}^{N}\frac{d^k\Lambda_k(\mathbf{F})}{k!}. \tag{14}$$

All terms depend on the regularity of the vector field $\mathbf{F}$ via the constants $\Lambda_k(\mathbf{F})$: the bigger these constants, the faster the vector field $\mathbf{F}$ may vary, making the CDE harder to predict. The convergence speed in $1/\sqrt{M}$ is classical. Also note that our bound is non-asymptotic and is valid for any $M \geq 1$.

The dependence of the bound on N is highly non-trivial and requires an in-depth analysis of the regularity of $\mathbf{F}$ in order to bound $\Lambda_k(\mathbf{F})$, which is out of the scope of this paper. The asymptotic behaviour of this oracle inequality is discussed in Appendix C.7.

4. Experiments

We study the performance of SigLasso obtained by solving the optimization problem (8), where $\Omega(\theta)$ is defined by Equation (11). All details are given in Appendix D.

4.1. Simulations

We consider several settings of data generation. First, in the well-specified setting, the data is generated from a model with regular feature paths x (piecewise polynomials) and target paths y wich are solutions to the CDE

$$dy_t = \tanh(Ay_t)dx_t,$$

where A is a randomly drawn matrix. In the ill-specified setting, the target y is equal to

$$y_t = \log\|\sum_{h=1}^{10} x_{t-h}\|$$

for any $t \in [0, 1]$. In the third setting, called OU setting, the feature paths are realizations of Brownian motions and the target paths are Ornstein-Uhlenbeck processes (Borodin & Salminen, 2012) driven by the feature paths. The last setting corresponds to the tumor growth model from Simeoni et al. (2004). The feature path represents the concentration of a treatment drug, generated as the squared value of the smooth paths used in the well-specified setting, and the target path y, the weight of the tumor, is governed by a system of differential equations given in Appendix D.6.

We compare SigLasso to a GRU and Neural CDE (Kidger et al., 2020). We measure the performance of the models with two metrics on a test set: the mean squared error for predicting the last observation point of the target paths and the L_2 error for predicting the full path on a fine grid.

The results are shown in Table 1 and Figure 2. In Figure 2 we consider the well-specified setting and vary the number of sampling points of the target paths between 1 and 20. In Table 1 it is fixed to 5 but the simulation settings change. Sampling of both the target and the feature time series is

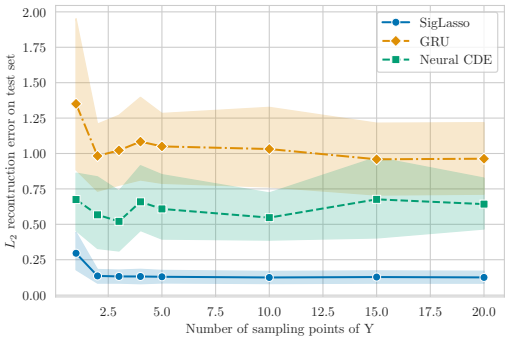

Figure 2: L_2 reconstruction error of SigLasso, GRU and Neural CDE in the well-specified setting, for varying number of target samples.

highly irregular. SigLasso outperforms Neural CDE and GRU models in generalizing from a few learning points of the target to its full trajectory in all settings. We conduct supplementary experiments with RNN and LSTM, which are also outperformed by SigLasso (see Table 2 in Appendix D.9). An additional byproduct of SigLasso's simple form is its training speed: its is approximately 10 times faster than GRU and 100 times faster than Neural CDE, including cross-validation to select N and regularization strength (see Appendix D.9).

4.2. Forecasting the Growth Rate of Hospitalizations in France During the Covid-19 Pandemic

Forecasting hospitalizations in real time during the Covid-19 pandemic is a notably difficult task. In this experiment, we train our model to learn the dynamics linking population data related to mobility, vaccination, and weather, and the hospitalization growth rate (HGR) in each of the 9 metropolitan regions of France based on the data of Paireau et al. (2022). The feature time series is regularly sampled and specific to each region and 12-dimensional. We consider prediction horizons $h = 1, \dots, 14$, meaning that we predict the hospital saturation at time t using the history of the feature time series up to $t - h$. Using our notations, we have for each region $d = 12$, $p = 1$, $n = 1$. Concretely, this means that we train one Siglasso model per region and per horizon. The target is sampled every day during the training period left of the dotted line in Figure 3, and the models are fitted to those values: on this time span, the models learns to interpolate the target time series. The model is then asked to predict the HGR on the days right of the dotted line seing only the feature time series. This means that it performs a prediction task on this time span.

Both GRU and SigLasso learn smooth and precise dynamics, which generalize well above the learning horizon and yield similar prediction performance (see Appendix D.8).

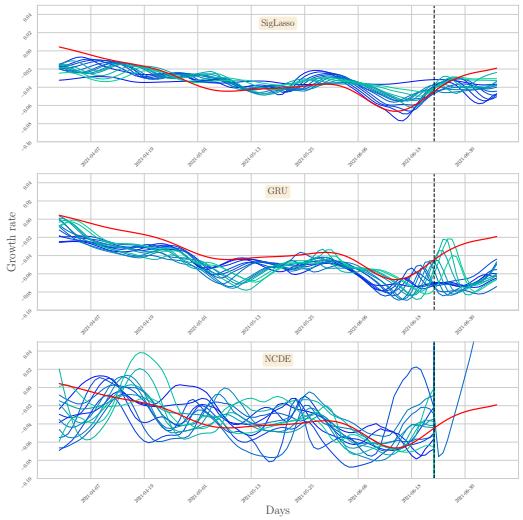

Figure 3: Interpolation (left of dotted line) and prediction (right of dotted line) of HGR in region Île de France for SigLasso, GRU and NCDE. The lighter the blue, the smaller the horizon h. Ground truth is in red. NCDE overfits and is unable to predict the HGR during the test period.

Our model is slightly outperformed by GRU for $h \geq 7$ for the prediction task, the difference in MSE being always less than 0.08. It performs similarly or better for most values of h for the interpolation task. Neural CDE and the original method proposed by Paireau et al. (2022) perform poorly. Figure 3 shows an example of reconstruction and prediction of HGR obtained with SigLasso, GRU and NCDE.

5. Conclusion

We have introduced a novel CDE-based model for interacting systems. Drawing on the theory of signatures, we derive an oracle bound that depends explicitly on the roughness of the data sampling. We illustrate the high performance of our approach on synthetic and real-world data.

The obtained theoretical guarantees rely on strong regularity assumptions on the vector field $\mathbf{F}$. The exact approximation properties of this class of vector field are a very interesting direction for future work. Considering other penalties that take into account the underlying structure of θ_N^* would also be an interesting extension of our work.

Acknowledgement. We thank anonymous ICML reviewers for their remarks, which helped us improve this paper. LB thanks Gérard Biau and Claire Boyer for supervising a previous internship which sparked his interest in signatures. LB and AF thank the Sorbonne Center for Artificial Intelligence (SCAI) and its team.

References

Bickel, P. J., Ritov, Y., and Tsybakov, A. B. Simultaneous analysis of lasso and dantzig selector. *The Annals of Statistics*, 37(4):1705–1732, 2009.

Borodin, A. N. and Salminen, P. *Handbook of Brownian Motion-Facts and Formulae*. Birkhäuser, Basel, 2012.

Brunton, S. L., Noack, B. R., and Koumoutsakos, P. Machine learning for fluid mechanics. *Annual review of fluid mechanics*, 52:477–508, 2020.

Buehler, H., Horvath, B., Lyons, T., Arribas, I. P., and Wood, B. A data-driven market simulator for small data environments. *arXiv preprint arXiv:2006.14498*, 2020.

Chen, K.-T. Integration of paths—a faithful representation of paths by non-commutative formal power series. *Transactions of the American Mathematical Society*, 89: 395–407, 1958.

Chen, R. T. Q., Rubanova, Y., Bettencourt, J., and Duvenaud, D. K. Neural ordinary differential equations. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 31, pp. 6572–6583. Curran Associates, Inc., 2018.

Chesneau, C. and Hebiri, M. Some theoretical results on the grouped variables lasso. *Mathematical Methods of Statistics*, 17(4):317–326, 2008.

Chevyrev, I. and Kormilitzin, A. A primer on the signature method in machine learning. *arXiv preprint arXiv:1603.03788*, 2016.

De Brouwer, E., Simm, J., Arany, A., and Moreau, Y. GRU-ODE-Bayes: Continuous modeling of sporadically-observed time series. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 32, pp. 7379–7390. Curran Associates, Inc., 2019.

Fattahi, S., Matni, N., and Sojoudi, S. Learning sparse dynamical systems from a single sample trajectory. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, pp. 2682–2689. IEEE, 2019.

Fermanian, A. Embedding and learning with signatures. *Computational Statistics & Data Analysis*, 157:107148, 2021.

Fermanian, A. Functional linear regression with truncated signatures. *Journal of Multivariate Analysis*, 192: 105031, 2022.

Fermanian, A., Marion, P., Vert, J.-P., and Biau, G. Framing RNN as a kernel method: A neural ODE approach. In Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 3121–3134. Curran Associates, Inc., 2021.

Friz, P. and Victoir, N. Euler estimates for rough differential equations. *Journal of Differential Equations*, 244: 388–412, 2008.

Friz, P. K. and Victoir, N. B. *Multidimensional Stochastic Processes as Rough Paths: Theory and Applications*, volume 120 of *Cambridge Studies in Advanced Mathematics*. Cambridge University Press, Cambridge, 2010.

Gehring, J., Auli, M., Grangier, D., Yarats, D., and Dauphin, Y. N. Convolutional sequence to sequence learning. In Precup, D. and Teh, Y. W. (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pp. 1243–1252. PMLR, 2017.

Greydanus, S., Dzamba, M., and Yosinski, J. Hamiltonian neural networks. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Han, J., Jentzen, A., and E, W. Solving high-dimensional partial differential equations using deep learning. *Proceedings of the National Academy of Sciences*, 115(34): 8505–8510, 2018.

Herrera, C., Krach, F., and Teichmann, J. Neural jump ordinary differential equations: Consistent continuous-time prediction and filtering. In *International Conference on Learning Representations*, 2021.

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

Kidger, P. On neural differential equations. *arXiv preprint arXiv:2202.02435*, 2022.

Kidger, P. and Lyons, T. Signatory: differentiable computations of the signature and logsignature transforms, on both cpu and gpu. In *International Conference on Learning Representations*, 2020.

Kidger, P., Bonnier, P., Perez Arribas, I., Salvi, C., and Lyons, T. Deep signature transforms. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 32, pp. 3099–3109. Curran Associates, Inc., 2019.

Kidger, P., Morrill, J., Foster, J., and Lyons, T. Neural controlled differential equations for irregular time series. In

Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 6696–6707. Curran Associates, Inc., 2020.

Lagaris, I. E., Likas, A., and Fotiadis, D. I. Artificial neural networks for solving ordinary and partial differential equations. *IEEE transactions on neural networks*, 9(5): 987–1000, 1998.

Lederer, J., Yu, L., and Gaynanova, I. Oracle inequalities for high-dimensional prediction. *Bernoulli*, 25(2):1225–1255, 2019.

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, pp. 3754–3762. PMLR, 2021.

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

Loh, P.-l. and Wainwright, M. J. High-dimensional regression with noisy and missing data: Provable guarantees with non-convexity. In Shawe-Taylor, J., Zemel, R., Bartlett, P., Pereira, F., and Weinberger, K. (eds.), *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc., 2011.

Long, Z., Lu, Y., Ma, X., and Dong, B. PDE-net: Learning PDEs from data. In Dy, J. and Krause, A. (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pp. 3208–3216. PMLR, 2018.

Lyons, T. and McLeod, A. D. Signature methods in machine learning. *arXiv preprint arXiv:2206.14674*, 2022.

Lyons, T., Caruana, M., and Lévy, T. *Differential Equations driven by Rough Paths*, volume 1908 of *Lecture Notes in Mathematics*. Springer, Berlin, 2007.

Lyons, T., Ni, H., and Oberhauser, H. A feature set for streams and an application to high-frequency financial tick data. In *Proceedings of the 2014 International Conference on Big Data Science and Computing*, pp. 5. ACM, 2014.

Marion, P., Fermanian, A., Biau, G., and Vert, J.-P. Scaling resnets in the large-depth regime. *arXiv preprint arXiv:2206.06929*, 2022.

Marx, B. D. and Eilers, P. H. Generalized linear regression on sampled signals and curves: a P-spline approach. *Technometrics*, 41:1–13, 1999.

Morrill, J., Fermanian, A., Kidger, P., and Lyons, T. A generalised signature method for multivariate time series feature extraction. *arXiv preprint arXiv:2006.00873*, 2020a.

Morrill, J., Kidger, P., Yang, L., and Lyons, T. Neural controlled differential equations for online prediction tasks. *arXiv preprint arXiv:2106.11028*, 2021a.

Morrill, J., Salvi, C., Kidger, P., and Foster, J. Neural rough differential equations for long time series. In Meila, M. and Zhang, T. (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139, pp. 7829–7838. PMLR, 2021b.

Morrill, J. H., Kormilitzin, A., Nevado-Holgado, A. J., Swaminathan, S., Howison, S. D., and Lyons, T. J. Utilization of the signature method to identify the early onset of sepsis from multivariate physiological time series in critical care monitoring. *Critical Care Medicine*, 48: e976–e981, 2020b.

Morris, J. S. Functional regression. *Annual Review of Statistics and Its Application*, 2:321–359, 2015.

Paireau, J., Andronico, A., Hozé, N., Layan, M., Crepey, P., Roumagnac, A., Lavielle, M., Boëlle, P.-Y., and Cauchemez, S. An ensemble model based on early predictors to forecast covid-19 health care demand in france. *Proceedings of the National Academy of Sciences*, 119(18):e2103302119, 2022.

Papavasiliou, A. and Ladroue, C. Parameter estimation for rough differential equations. *The Annals of Statistics*, 39 (4):2047–2073, 2011.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Ramsay, J. O. and Dalzell, C. Some tools for functional data analysis. *Journal of the Royal Statistical Society. Series B (Methodological)*, 53:539–561, 1991.

Ramsay, J. O. and Silverman, B. W. *Functional Data Analysis. 2nd Edition.* Springer, New York, 2005.

Reizenstein, J. F. and Graham, B. Algorithm 1004: The iisignature library: Efficient calculation of iterated-integral signatures and log signatures. *ACM Transactions on Mathematical Software*, 46(1):1–21, 2020.

Rubanova, Y., Chen, R. T. Q., and Duvenaud, D. K. Latent ordinary differential equations for irregularly-sampled time series. In Wallach, H., Larochelle, H., Beygelzimer,

A., d'Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 32, pp. 5320–5330. Curran Associates, Inc., 2019.

Simeoni, M., Magni, P., Cammia, C., De Nicolao, G., Croci, V., Pesenti, E., Germani, M., Poggesi, I., and Rocchetti, M. Predictive pharmacokinetic-pharmacodynamic modeling of tumor growth kinetics in xenograft models after administration of anticancer agents. *Cancer research*, 64(3):1094–1101, 2004.

Sutskever, I., Vinyals, O., and Le, Q. V. Sequence to sequence learning with neural networks. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N., and Weinberger, K. (eds.), *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014.

Vershynin, R. Introduction to the non-asymptotic analysis of random matrices. *arXiv preprint arXiv:1011.3027*, 2010.

Vershynin, R. *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge university press, 2018.

Wang, B., Wu, Y., Taylor, N., Lyons, T., Liakata, M., Nevado-Holgado, A. J., and Saunders, K. E. Learning to detect bipolar disorder and borderline personality disorder with language and speech in non-clinical interviews. *Interspeech 2020*, pp. 437–441, 2020.

Willard, J., Jia, X., Xu, S., Steinbach, M., and Kumar, V. Integrating physics-based modeling with machine learning: A survey. *arXiv preprint arXiv:2003.04919*, 2020.

Yang, W., Lyons, T., Ni, H., Schmid, C., and Jin, L. Developing the path signature methodology and its application to landmark-based human action recognition. In Yin, G. and Zariphopoulou, T. (eds.), *Stochastic Analysis, Filtering, and Stochastic Optimization: A Commemorative Volume to Honor Mark H.A. Davis's Contributions*, pp. 431–464. Springer, 2022.

Zubov, K., McCarthy, Z., Ma, Y., Calisto, F., Pagliarino, V., Azeglio, S., Bottero, L., Luján, E., Sulzer, V., Bharambe, A., et al. NeuralPDE: Automating physics-informed neural networks (PINNs) with error approximations. *arXiv preprint arXiv:2107.09443*, 2021.

Zwillinger, D. *Handbook of differential equations*. Academic Press, Inc., London, 1989.

Supplementary Material

A. Summary of used notations

The following table provides an exhaustive list of notations used in the main body of the paper. Notations are grouped by subsections.

Notation	Definition	Reference
y_t	Target path	Introduction, page 1
x_t	Feature path	—
Y_t	Target time series	Introduction, page 2
X_t	Feature time series	—
D^i	Sampling grid of the features of individual i	—
$\bar{D}^i$	Sampling grid of the target of individual i	—
ξ_t^i	Noise on the feature time series	—
ε_t^i	Noise on the target time series	—
m_i	Number of sampling points of the target for individual i	—
n	Number of samples	—
M	Total number of sampling points of the target	—
$\mathbf{X}^i$	Matrix of measurements of the feature time series for individual i	—
$\mathbf{Y}^i$	Matrix of measurements of the target time series for individual i	—
$\|x\|_{1\text{-var},[0,t]}$	Total variation of the path x on the interval $[0,t]$	Section 2, page 3, Assumption 1
$C_L^{1\text{-var}([0,1],\mathbb{R}^d)}$	Set of continuous paths of total variation bounded by L	—
$\mathbf{F}$	Unknown smooth generative vector field	Section 2, page 3, Assumption 2
$S^I(x_{[0,t]})$	Signature coefficient of the path x on $[0,t]$ associated to the word I	Section 2, page 4, Definition 2.1
$\mathbb{X}_{k,[0,t]}$	Signature of order k	—
$S(x_{[0,t]})$	Full signature at order N	—
$S_N(x_{[0,t]})$	Truncated signature at order N	—
$s_d(N)$	Size of the signature truncated at order N of a d dimensional path	—
$\bar{y}_{N,t}$	Taylor expansion of the solution of a CDE of order N evaluated in t	Section 2, page 4, Definition 2.2
$\Phi_{\mathbf{F}}^I$	Differential product of the vector field $\mathbf{F}$ along I	Appendix B.3, page 15, Definition B.3
$\theta_N^\star$	Matrix collecting all differential products up to order N	Section 2, page 5, Equation (5)
$\Lambda_k(\mathbf{F})$	Norm on the differential product of $\mathbf{F}$	Section 2, page 5, Assumption 3
η	Minimal sampling time	Section 2, page 5, Assumption 4
$\mathcal{D}$	Set of individual specific sampling grids of features	Section 2, page 5
$\bar{\mathcal{D}}$	Set of individual specific sampling grids of targets	—
$\|D\|$	Meshisze of a sampling grid D	—
$\#D$	Number of sampling points in a sampling grid	—
$\mathbf{Y}$	Matrix collecting all target measurements of the sample	—
$\mathbf{S}_N^{\mathcal{D}}$	Matrix of stacked signatures	Section 2, page 6
$\mathbf{y}$	Expectation of $\mathbf{Y}$	Section 3, page 6
$\hat{\theta}_{N,M}$	Siglasso estimator	Section 3, page 6, Equation (7)
$A_\xi(\delta)$	Set bounding the magnitude of noises (ξ_t^i)	Section 3, page 6, Equation (9)
$C_k(\bar{\delta})$	Constant	Section 3, page 6
$A_\varepsilon(\delta)$	Set bounding the magnitude of the noises (ε_t^i)	Section 3, page 7, Equation (10)
$\Omega(\theta)$	Penalty evalutated at θ	Section 3, page 7, Equation (11)

B. Mathematical details

B.1. The Riemann-Stieltjes integral

We fitst recall two key properties on the Riemann-Stieljes integral. For a general presentation of the Riemann-Stieltjes integral, we refer to Friz & Victoir (2010).

Proposition B.1. *Let $x \in C_L^{1\text{-var}}([0,1],\mathbb{R}^d)$ and $y : [0,1] \to \mathbb{R}^d$ be a continuous path. Then*

$$\left\| \int_0^t y_s dx_s \right\| \leq \|y\|_{\infty,[0,t]} \|x\|_{1\text{-var},[0,t]}$$

We refer the reader to Friz & Victoir (2010, Proposition 2.2) for a proof.

Proposition B.2 (Integration by parts)**.** *Let $x, y \in C_L^{1\text{-var}}([0,1],\mathbb{R}^d)$. Then*

$$\int_s^t y_u dx_u + \int_s^t x_u dy_u = y_t x_t - y_s x_s$$

See Friz & Victoir (2010, Proposition 2.4) for a proof.

B.2. The truncated tensor algebra

This section introduces notations and definitions on the space in which signatures are defined, namely, the tensor algebra. While for the exposition of our main results, the truncated signature of a path $x \in C_L^{1\text{-var}}([0,1],\mathbb{R}^d)$ at depth $N \geq 1$ can be assimilated to an element of $\mathbb{R}^{s_d(N)}$, it is often useful to place ourselves in the tensor algebra to obtain finer bounds or technical results.

Let $x \in C_L^{1\text{-var}}([0,1],\mathbb{R}^d)$ be a path of bounded variation. For a word $I = (i_1,\dots,i_k) \in \{1,\dots,d\}^k$ of size k, the signature coefficient $S^I(x_{[0,1]})$ can be seen as an element of the k-th tensor product of $\mathbb{R}^d$ with itself, denoted by $\left(\mathbb{R}^d\right)^{\otimes k}$. For instance, the coefficients of order $k=1$ can be written as a vector and the coefficients of order $k=2$ as a matrix, and so on, i.e.,

$$\mathbb{X}^1_{[0,1]} = \begin{bmatrix} \int_0^1 dx_s^{(1)} \\ \vdots \\ \int_0^1 dx_s^{(d)} \end{bmatrix} \quad \text{and} \quad \mathbb{X}^2_{[0,1]} = \begin{bmatrix} \int_0^1 dx_s^{(1)} dx_s^{(1)} & \dots & \int_0^1 dx_s^{(1)} dx_s^{(d)} \\ \vdots & & \vdots \\ \int_0^1 dx_s^{(d)} dx_s^{(1)} & \dots & \int_0^1 dx_s^{(d)} dx_s^{(d)} \end{bmatrix}.$$

We now define a norm on $(\mathbb{R}^d)^{\otimes k}$. Let $a \in (\mathbb{R}^d)^{\otimes k}$ and $(e_1,\dots,e_d)$ be the canonical basis of $\mathbb{R}^d$. Then $(e_{i_1} \otimes \cdots \otimes e_{i_k})_{(i_1,\dots,i_k)\in\{1,\dots,d\}^k}$ is a basis of $(\mathbb{R}^d)^{\otimes k}$. We can thus write a as $a = (a^I)_{I\in\{1,\dots,d\}^k}$. For every $k \geq 0$, the vector space $(\mathbb{R}^d)^{\otimes k}$ is naturally endowed with the norm

$$\|a\|^2_{(\mathbb{R}^d)^{\otimes k}} = \sum_{I\in\{1,\dots,d\}^k} \left(a^I\right)^2.$$

Remark that this norm satisfies for any $x \in (\mathbb{R}^d)^{\otimes k}$ and $y \in (\mathbb{R}^d)^{\otimes m}$,

$$\|x \otimes y\|_{(\mathbb{R}^d)^{\otimes(k+m)}} = \|x\|_{(\mathbb{R}^d)^{\otimes k}} \|y\|_{(\mathbb{R}^d)^{\otimes m}}. \tag{15}$$

We refer to Fermanian et al. (2021) for further deails. The signature truncated at depth $N \geq 1$ collects elements from $\mathbb{R}, \left(\mathbb{R}^d\right)^{\otimes 2}, \dots, \left(\mathbb{R}^d\right)^{\otimes N}$. It can thus be seen as an element of the truncated tensor algebra

$$T_N(\mathbb{R}^d) = \mathbb{R} \oplus \left(\mathbb{R}^d\right)^{\otimes 2} \oplus \cdots \oplus \left(\mathbb{R}^d\right)^{\otimes N}.$$

Let $a = (a_0,\dots,a_N) \in T_N(\mathbb{R}^d)$, where every $a_k \in (\mathbb{R}^d)^{\otimes k}$. We define the norm

$$\|a\|_{T_N(\mathbb{R}^d)} = \left(\sum_{k=0}^N \|a_k\|^2_{(\mathbb{R}^d)^{\otimes k}}\right)^{1/2}.$$

To clarify, if we consider the truncated signature of x at depth $N \geq 1$, which is an element of $T_N(\mathbb{R}^d)$, then

$$\left\|S_N(x_{[0,t]})\right\|_{T_N(\mathbb{R}^d)} = \Big(\sum_{k=0}^{N} \left\|\mathbb{X}^k_{[0,t]}\right\|^2_{(\mathbb{R}^d)^{\otimes k}}\Big)^{1/2} = \Big(\sum_{k=0}^{N} \sum_{I\in\{1,\dots,d\}^k} S^I(x_{[0,t]})^2\Big)^{1/2}.$$

Note that this norm is exactly equivalent to the Euclidian norm of $\mathbb{R}^{s_d(N)}$, which is the space we consider in the exposition of our main results for the sake of simplicity.

We are now ready to define the tensor product on the truncated tensor algebra. For two elements $a = (a_0, \dots, a_N)$ and $b = (b_0, \dots, b_N)$ both in $T_N(\mathbb{R}^d)$, we define

$$a \otimes b = (c_0, \dots, c_j, \dots, c_N), \qquad \text{where } c_j = \sum_{k=0}^{j} a_k \otimes b_{j-k}.$$

For any $k = 0, \dots, N$, we let $\pi_k : T_N(\mathbb{R}^d) \to (\mathbb{R}^d)^{\otimes k}$ be the canonical projection of $T_N(\mathbb{R}^d)$ onto $(\mathbb{R}^d)^{\otimes k}$. More precisely, for every $a = (a_0, \dots, a_N) \in T_N(\mathbb{R}^d)$,

$$\pi_k(a) = a_k$$

We also define the canonical projection $\Pi_k : T_N(\mathbb{R}^d) \to T_k(\mathbb{R}^d)$ defined by

$$\Pi_k(a) = (a_0, \dots, a_k)\,.$$

B.3. The differential product

We first define the differential product, in order to give a precise statement of Assumption 3.

Definition B.3. Let $F, G : \mathbb{R}^p \to \mathbb{R}^p$ be two smooth vector fields, i.e., each of their components is $\mathcal{C}^\infty$. Denote by $J(\cdot)$ the Jacobian matrix. The differential product $F \star G : \mathbb{R}^p \to \mathbb{R}^p$ is the smooth vector field defined for any $h \in \mathbb{R}^p$

$$(F \star G)(h) = \sum_{j=1}^{e} \frac{\partial G}{\partial h_j}(h) F_j(h) = J(G)(h)F(h).$$

The differential product is not associative. We therefore use the convention to evaluate it from right to left, that is,

$$F^1 \star F^2 \star F^3 = F^1 \star \left(F^2 \star F^3\right).$$

Let $F : \mathbb{R}^p \to \mathbb{R}^{p\times d}$ be a smooth vector field. We write $F^1, \dots, F^d$ the columns of $\mathbf{F}$. Every F^i, for $i = 1, \dots, d$, can thus be seen as a map from $\mathbb{R}^p$ to $\mathbb{R}^p$. Recall that $y_0 \in \mathbb{R}^p$ is the initial condition of the CDE defined in Assumption 2. Let $I = (i_1, \dots, i_k) \in \{1, \dots, d\}^k$. We now define

$$\Phi^I_{\mathbf{F}}(y_0) = \left(F^{i_1} \star \cdots \star F^{i_k}\right)(y_0) \in \mathbb{R}^p.$$

We refer to Fermanian et al. (2021) for greater details on the differential product. We now define for all $k \geq 1$

$$\Lambda_k(\mathbf{F}) = \sup_{1\leq i_1,\dots,i_k\leq d} \left\|\Phi^I_{\mathbf{F}}(y_0)\right\| \in \mathbb{R}, \tag{16}$$

and use the convention $\Lambda_0(\mathbf{F}) = \|y_0\|$. Remark that Assumption 3 implies that

$$\frac{d^{N+1}\Lambda_{N+1}(\mathbf{F})}{(N+1)!} \underset{N\to+\infty}{\longrightarrow} 0.$$

As an immediate consequence, the truncation bias

$$\left(\frac{d^{N+1}\Lambda_{N+1}(\mathbf{F})}{(N+1)!}\right)^2$$

introduced in Lemma 3.2, vanishes as N grows.

B.4. Analogy with the Taylor extension

The definition of the Taylor expansion of a CDE exposed in the previous subsection is technical. However, it can simply be thought of as a generalization of the the classical Taylor expansion of a $\mathcal{C}^\infty$ function $f : \mathbb{R} \to \mathbb{R}$. Recall that in this case, the Taylor expansion at 0 evaluated at $t \in \mathbb{R}$ of order $N \in \mathbb{N}$ writes as a power series

$$f(t) \approx f(0) + \frac{f'(0)}{1!}t + \cdots + \frac{f^{(N)}(0)}{N!}t^N. \tag{17}$$

Every element of this power series is a product of two terms: the derivatives of f encode some information about the regularity of f at the initial point 0 and do not depend on t, while the polynomial terms t^k allow this linearized form to evolve with time t. Remark that these polynomial terms do not depend on f.

Similarly, Equation (4) is also a sum of products of two terms. On the one hand, the evolving nature of the system, instead of being handled by the polynomial terms t^k, are now captured by the signature coefficients $S^I(x_{[0,t]})$. As the polynomial terms, they do not depend on $\mathbf{F}$. On the other hand, the information about the initial value of the system at time $t = 0$ and the dynamics of $\mathbf{F}$ are summarized by the differential product $\Phi^I_{\mathbf{F}}(y_0)$, which play the same role as the successive derivatives in Equation (17). To capture the multivariate nature of the paths, the Taylor expansion is summed over multi-indexes, or words, $I = (i_1, \dots, i_k) \in \{1, \dots, d\}^k$ of size k for $k \in \mathbb{N}$.

B.5. Properties of subgaussian random vectors

We start with the definition of a subgaussian random variable, see Vershynin (2010) for more details.

Definition B.4. A real-valued random variable X is said to be σ^2-subgaussian if for all $t > 0$

$$\mathbb{P}(X > t) \leq \exp(-t^2/\sigma^2),$$

or, equivalently, if for all $t \in \mathbb{R}$

$$\mathbb{E}(e^{tX}) \leq \exp(-ct^2\sigma^2),$$

where c is an universal constant. A random vector Z is subgaussian if, for any vector c of norm 1, $\langle Z, c\rangle$ is subgaussian.

The norm of a sequence of d subgaussian random variables concentrates around $\sqrt{d}$, as stated by the following lemma.

Lemma B.5. *Let $X_1, \dots, X_n$ be a sequence of i.i.d. σ^2-subgaussian random variables. Let $X = (X_1, \dots, X_d) \in \mathbb{R}^d$. There exists a universal constant c such that for all $t > 0$*

$$\mathbb{P}(\|X\|_2 \geq t + \sigma\sqrt{d}) \leq \exp(-ct^2/\sigma^2).$$

Proof. We refer to Vershynin (2018, Theorem 3.1.1) for a proof. □

We can use this lemma to bound the maximum of n sequences of d subgaussian random variables with high probability.

Lemma B.6. *Let $X_1, \dots, X_n$ be a sequence of i.i.d. σ^2-subgaussian random variables, such that for all $i = 1, \dots, n$, $X_i = (X_{i1}, \dots, X_{id})$. Then there exists a universal constant c such that for all $\delta \in (0, 1)$*

$$\mathbb{P}\Big(\max_{i=1,\dots,n} \|X_i\| \leq \sigma\sqrt{d} + \sigma\sqrt{\frac{1}{c}\log(n/\delta)}\Big) \geq 1 - \delta.$$

Proof. Using Lemma B.5 and a union bound, we have

$$\begin{aligned}
\mathbb{P}\Big(\max_{i=1,\dots,n} \|X_i\| \geq \sigma\sqrt{d} + \sigma\sqrt{\frac{1}{C}\log(n/\delta)}\Big) &= \mathbb{P}\Big(\bigcup_{i=1}^n \Big\{ \|X_i\|_2 \geq \sigma\sqrt{d} + \sigma\sqrt{\frac{1}{C}\log(n/\delta)}\Big\}\Big) \\
&\leq \sum_{i=1}^n \mathbb{P}\Big(\|X_i\|_2 \geq \sigma\sqrt{d} + \sigma\sqrt{\frac{1}{c}\log(n/\delta)}\Big) \\
&\leq \delta,
\end{aligned}$$

which yields the desired inequality. □

Notice that the universal constant is identical between both lemmas. As a consequence of this last lemma, under Assumption 5, the set

$$A_\xi(\delta) = \left\{ \max_{i=1,\dots,n, t\in D^i} \left\|\xi_t^i\right\| \leq v_\xi \sqrt{d} + v_\xi \sqrt{\frac{1}{c}\log(\#\mathcal{D}/\delta)} \right\} \tag{18}$$

where $\#\mathcal{D} = \sum_{i=1}^{n} \#D^i$ is of probability at least $1-\delta$.

We also need the following lemma.

Lemma B.7. *Let $X_1, \dots, X_n$ be a sequence of i.i.d. σ^2-subgaussian random variables. Let $Z_1, \dots, Z_n$ be random variables such that for all $i = 1, \dots, n$, $|Z_i| \leq \alpha$ almost surely. Then $\sum_{i=1}^{n} X_i Z_i$ is $n\sigma^2\alpha^2$-subgaussian.*

Proof. We use the characterization of subgaussian random variables by their characteristic function. For all $t > 0$,

$$\mathbb{E}\left[e^{t\sum_{i=1}^n X_i Z_i}\right] = \mathbb{E}\left[\prod_{i=1}^{n} \mathbb{E}\left[e^{tX_i Z_i} \mid Z_1, \dots, Z_n\right]\right] \leq \mathbb{E}\left[\prod_{i=1}^{n} \mathbb{E}\left[e^{tX_i \alpha}\right]\right] \leq \mathbb{E}\left[e^{ct^2 n\alpha^2\sigma^2}\right].$$

This finally yields that

$$\mathbb{E}\left[e^{t\sum X_i Z_i}\right] \leq \mathbb{E}\left[e^{ct^2 n\alpha^2\sigma^2}\right],$$

which concludes the proof. □

C. Proofs

C.1. Preliminary notations

Let $(E, \|\cdot\|_E)$ be a normed vector space and $x : [0,1] \to E$. The supremum norm of x is defined for all $t \in [0,1]$ as

$$\|x\|_{\infty,[0,t]} = \sup_{s\in[0,t]} \|x_s\|_E .$$

When referring to the total variation $\|x\|_{1\text{-var},[0,1]}$ of a path $x : [0,1] \to \mathbb{R}^d$ over the whole domain, depending on the mathematical context, we will sometimes drop the time subscript and simply write $\|x\|_{1\text{-var}}$.

When referring to a matrix $A = (A_{ij}) \in \mathbb{R}^{n\times p}$, we define classicaly the infinite and Frobenius norms by

$$\|A\|_\infty = \max_{\substack{i=1,\dots,n\\ j=1,\dots,p}} |A_{ij}| \quad \text{and} \quad \|A\|_F = \sqrt{\sum_{\substack{i=1,\dots,n\\ j=1,\dots,p}} |A_{ij}|^2}.$$

We now introduce some notations to take advantage of the structure of θ_N^*. The true parameter of the Taylor expansion of the model CDE, defined in Equation (5), can be written in block notation as

$$\theta_N^* = \left[\begin{array}{ccc} \theta^*_{[0],1} & \cdots & \theta^*_{[0],p} \\ \hline \theta^*_{[1],1} & \cdots & \theta^*_{[1],p} \\ \hline \theta^*_{[2],1} & \cdots & \theta^*_{[2],p} \\ \hline & \vdots & \\ \hline \theta^*_{[N],1} & \cdots & \theta^*_{[N],p} \end{array}\right] \in \mathbb{R}^{s_d(N)\times p}, \quad \text{where} \quad \theta^*_{[k],\ell} \in \mathbb{R}^{d^k\times 1}, k = 0,\dots,N,\ \ell = 1,\dots,p. \tag{19}$$

Every column of θ_N^* corresponds to a dimension of the target, while blocks of lines correspond to signatures layers. Thus for every $k = 0,\dots,N$ and $\ell = 1,\dots,p$, $\theta^*_{[k],\ell}$ is a column vector of size d^k.

Similarly, for a general $\theta \in \mathbb{R}^{s_d(N)\times p}$ and the SigLasso estimator $\widehat{\theta}_{N,M}$, we will refere to the blocks forming these matrices as respectively $\theta_{[k],\ell}$ and $\widehat{\theta}_{[k],\ell}$, for $k = 0,\dots,N$ and $\ell = 1,\dots p$.

Likewise, the signature feature matrix $\mathbf{S}_N^{\mathcal{D}} \in \mathbb{R}^{M\times s_d(N)}$ can be written in block notation as

$$\mathbf{S}_N^{\mathcal{D}} = \left[1 \;\middle|\; \mathbf{S}^{\mathcal{D}}_{\cdot,[1]} \;\middle|\; \mathbf{S}^{\mathcal{D}}_{\cdot,[2]} \;\middle|\; \cdots \;\middle|\; \mathbf{S}^{\mathcal{D}}_{\cdot,[N]}\right] = \left[\begin{array}{c|c|c|c|c} 1 & \mathbf{S}^{\mathcal{D}}_{1,[1]} & \mathbf{S}^{\mathcal{D}}_{1,[2]} & \cdots & \mathbf{S}^{\mathcal{D}}_{1,[N]} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & \mathbf{S}^{\mathcal{D}}_{n,[1]} & \mathbf{S}^{\mathcal{D}}_{n,[2]} & & \mathbf{S}^{\mathcal{D}}_{n,[N]} \end{array}\right],$$

where for any $k = 1,\dots,N$, $\mathbf{S}^{\mathcal{D}}_{\cdot,[k]} \in \mathbb{R}^{M\times d^k}$ and, for every individual $i = 1,\dots,n$, $\mathbf{S}^{\mathcal{D}}_{i,[k]} \in \mathbb{R}^{m_i\times d^k}$ (recall that m_i is the number of measurements of the target path y^i). More precisely, given her target sampling grid $\bar{D}^i = (\bar{t}^i_1,\dots,\bar{t}^i_{m_i})$, the individual-specific signature block of depth k is equal to

$$\mathbf{S}^{\mathcal{D}}_{i,[k]} = \begin{bmatrix} 1 & S^{(1)}(X^i_{[0,\bar{t}^i_1]}) & \cdots & S^{(d,\dots,d)}(X^i_{[0,\bar{t}^i_1]}) \\ \vdots & \vdots & & \vdots \\ 1 & S^{(1)}(X^i_{[0,\bar{t}^i_{m_i}]}) & \cdots & S^{(d,\dots,d)}(X^i_{[0,\bar{t}^i_{m_i}]}) \end{bmatrix},$$

where the path $t \to X^i_t$ is a linear interpolation of the observed time series $\mathbf{X}^i$. The same notations will be used for the true signature feature matrix $\mathbf{S}_N$. We use the bracket notation $[\cdot]$ both in θ_N^* and $\mathbf{S}_N^{\mathcal{D}}$ to emphasise that both the columns of the feature matrix and the lines of learned parameter correspond to words of the alphabet $\{1,\dots,d\}$.

The unobserved matrix of true values of the target writes as

$$\mathbf{y} = \begin{bmatrix} \mathbf{y}^1 \\ \vdots \\ \mathbf{y}^n \end{bmatrix} = \begin{bmatrix} \mathbf{y}_1^1 & \cdots & \mathbf{y}_p^1 \\ \vdots & & \vdots \\ \mathbf{y}_1^n & \cdots & \mathbf{y}_p^n \end{bmatrix} = \begin{bmatrix} y^1_{1,\bar{t}^1_1} & \cdots & y^1_{p,\bar{t}^1_1} \\ \vdots & & \vdots \\ y^1_{1,\bar{t}^1_{m_1}} & \cdots & y^1_{p,\bar{t}^1_{m_1}} \\ \vdots & & \vdots \\ y^n_{1,\bar{t}^n_1} & \cdots & y^n_{p,\bar{t}^n_1} \\ \vdots & & \vdots \\ y^n_{1,\bar{t}^n_{m_n}} & \cdots & y^n_{p,\bar{t}^n_{m_n}} \end{bmatrix} \in \mathbb{R}^{M \times p} \tag{20}$$

and the measurement matrix $\mathbf{Y} \in \mathbb{R}^{M \times p}$ can be written in a similar fashion as

$$\mathbf{Y} = \begin{bmatrix} \mathbf{Y}^1 \\ \vdots \\ \mathbf{Y}^n \end{bmatrix} = \begin{bmatrix} \mathbf{Y}_1^1 & \cdots & \mathbf{Y}_p^1 \\ \vdots & & \vdots \\ \mathbf{Y}_1^n & \cdots & \mathbf{Y}_p^n \end{bmatrix} = \begin{bmatrix} Y^1_{1,\bar{t}^1_1} & \cdots & Y^1_{p,\bar{t}^1_1} \\ \vdots & & \vdots \\ Y^1_{1,\bar{t}^1_{m_1}} & \cdots & Y^1_{p,\bar{t}^1_{m_1}} \\ \vdots & & \vdots \\ Y^n_{1,\bar{t}^n_1} & \cdots & Y^n_{p,\bar{t}^n_1} \\ \vdots & & \vdots \\ Y^n_{1,\bar{t}^n_{m_n}} & \cdots & Y^n_{p,\bar{t}^n_{m_n}} \end{bmatrix} = \begin{bmatrix} y^1_{1,\bar{t}^1_1} + \varepsilon^1_{1,\bar{t}^1_1} & \cdots & y^1_{p,\bar{t}^1_1} + \varepsilon^1_{p,\bar{t}^1_1} \\ \vdots & & \vdots \\ y^1_{1,\bar{t}^1_{m_1}} + \varepsilon^1_{1,\bar{t}^1_{m_1}} & \cdots & y^1_{p,\bar{t}^1_{m_1}} + \varepsilon^1_{p,\bar{t}^1_{m_1}} \\ \vdots & & \vdots \\ y^n_{1,\bar{t}^n_1} + \varepsilon^n_{1,\bar{t}^n_1} & \cdots & y^n_{p,\bar{t}^n_1} + \varepsilon^n_{p,\bar{t}^n_1} \\ \vdots & & \vdots \\ y^n_{1,\bar{t}^n_{m_n}} + \varepsilon^n_{1,\bar{t}^n_{m_n}} & \cdots & y^n_{p,\bar{t}^n_{m_n}} + \varepsilon^n_{p,\bar{t}^n_{m_n}} \end{bmatrix} \tag{21}$$

C.2. Proof of Lemma 3.1

Using the definition of $\Lambda_k(\mathbf{F})$ (see Equation (16)), we get the following proposition which allows to obtain an explicit dependence of the oracle bound on the regularity of $\mathbf{F}$.

Proposition C.1. *Let θ_N^* be defined as in Equation (5). Then*

$$\|\theta_N^*\|_{\mathrm{F}}^2 \leq \sum_{k=0}^{N} d^k \Lambda_k(\mathbf{F})^2,$$

and, for all $k = 0, \ldots, N$ and $\ell = 1, \ldots, p$,

$$\left\|\theta^*_{[k],\ell}\right\|_1 \leq d^k \Lambda_k(\mathbf{F}).$$

Proof. By definition,

$$\|\theta_N^*\|_{\mathrm{F}}^2 = \sum_{k=0}^{N} \sum_{1 \leq i_1, \ldots, i_k \leq d} \left\|F^{i_1} \star \cdots \star F^{i_k}(y_0)\right\|_2^2.$$

Since for all $(i_1, \ldots, i_k) \in \{1, \ldots, d\}^k$,

$$\left\|F^{i_1} \star \cdots \star F^{i_k}(y_0)\right\|_2^2 \leq \Lambda_k(\mathbf{F})^2,$$

we get

$$\begin{aligned} \sum_{k=0}^{N} \sum_{1 \leq i_1, \ldots, i_k \leq d} \left\|F^{i_1} \star \cdots \star F^{i_k}(y_0)\right\|_2^2 &\leq \sum_{k=0}^{N} \sum_{1 \leq i_1, \ldots, i_k \leq d} \Lambda_k(\mathbf{F})^2 \\ &\leq \sum_{k=0}^{N} d^k \Lambda_k(\mathbf{F})^2. \end{aligned}$$

We now turn to the second inequality. For $k = 0$, the inequality holds by definition. For $k = 1, \dots, N$ and $\ell = 1, \dots, p$, by definition of the ℓ_1 norm,

$$\left\|\theta^*_{[k],\cdot}\right\|_1 = \sum_{1 \le i_1, \dots, i_k \le d} \left\|\Phi^I_{\mathbf{F}}(y_0)\right\|_1,$$

This yields

$$\left\|\theta^*_{[k],\cdot}\right\|_1 \le d^k \Lambda_k(\mathbf{F})$$

and thus

$$\left\|\theta^*_{[k],\ell}\right\|_1 \le d^k \Lambda_k(\mathbf{F})$$

for $\ell = 1, \dots, p$. □

The following lemma is needed to leverage classical proof techniques to bound the prediction error of the Lasso estimator.

Lemma C.2. *Let $x \in C_L^{1\text{-var}}([0,1], \mathbb{R}^d)$. Then conditionally on $A_\xi(\delta)$, for a given signature layer $k \ge 1$, the maximum among all signature coefficients and individuals is bounded from above, that is*

$$\left\|\mathbf{S}^{\mathcal{D}}_{\cdot,[k]}\right\|_\infty \le \frac{1}{k!}.$$

Proof. It is well known (see, e.g., Fermanian, 2022, Proposition 3) that if $\mathbb{X}^k$ is the signature of a path $x \in C_L^{1\text{-var}}([0,1], \mathbb{R}^d)$, then

$$\left\|\mathbb{X}^k\right\|_{(\mathbb{R}^d)^{\otimes k}} \le \frac{\|x\|^k_{1\text{-var}}}{k!}.$$

As a consequence, for every word I of size k, one gets

$$\left|S^I(x))\right| \le \frac{\|x\|^k_{1\text{-var}}}{k!}.$$

The matrix $\mathbf{S}^{\mathcal{D}}_N$ is constructed by taking signatures of linear interpolations of the $\mathbf{X}^i$s normalized by their total variation. It therefore contains only signatures of paths of total variation bounded by 1. Taking the maximum on $I \in \{1, \dots, d\}^k$ and individuals $i = 1, \dots, n$, we get

$$\left\|\mathbf{S}^{\mathcal{D}}_{\cdot,[k]}\right\|_\infty \le \frac{1}{k!}.$$

□

This final inequality being stated, we can now go back to the proof of Lemma 3.1. We prove it in full generality for $p \ge 1$. In this proof, we make extensive use of the notations introduced in Subsection C.1 and refer the reader to it if a notation is unclear.

Proof. In all the proof, we place ourselves on the set $A_\xi(\delta)$ defined by Equation (9), which ensures that the matrix $\mathbf{S}^{\mathcal{D}}_N$, seen as a random quantity, is well defined. Recall that we have two sources of randomness: the feature noises ξ^i_t on the $\mathbf{X}^i$s and the target noises ε^i_t on the $\mathbf{Y}^i$s. The feature noises appear only in $\mathbf{S}^{\mathcal{D}}_N$ and make it a random quantity. For $\mathbf{S}^{\mathcal{D}}_N$ to be well-defined, we then need the total variation of the linear interpolation of the feature time series $\mathbf{X}^i$ to be finite. This holds on the set $A_\xi(\delta)$ since all noises are then bounded.

Recall that we have defined $\widehat{\theta}_{N,M}$ as

$$\widehat{\theta}_{N,M} \in \underset{\theta \in \mathbb{R}^{s_d(N) \times p}}{\arg\min} \frac{1}{2M} \left\|\mathbf{Y} - \mathbf{S}^{\mathcal{D}}_N \theta\right\|^2_{\mathrm{F}} + \Omega(\theta).$$

Note that

$$\frac{1}{2M} \left\|\mathbf{Y} - \mathbf{S}^{\mathcal{D}}_N \theta\right\|^2_{\mathrm{F}} + \Omega(\theta) = \sum_{\ell=1}^{p} \frac{1}{2M} \left\|\mathbf{Y}_\ell - \mathbf{S}^{\mathcal{D}}_N \theta_{[\cdot],\ell}\right\|^2_2 + \Omega(\theta_{[\cdot],\ell}),$$

where $\mathbf{Y}_\ell \in \mathbb{R}^M$ is the ℓ-th column of the target measurement matrix defined in Equation (21). The quantity $\theta_{[\cdot],\ell} \in \mathbb{R}^{s_d(N)}$ is the ℓ-th column of the parameter matrix defined in Equation (19).

By definition, for any $\theta \in \mathbb{R}^{s_d(N)}$, we have

$$\left\|\mathbf{Y}_\ell - \mathbf{S}_N^{\mathcal{D}}\widehat{\theta}_{[\cdot],\ell}\right\|_2^2 \leq \left\|\mathbf{Y}_\ell - \mathbf{S}_N^{\mathcal{D}}\theta_{[\cdot],\ell}\right\|_2^2 + \Omega(\theta_{[\cdot],\ell}) - \Omega(\widehat{\theta}_{[\cdot],\ell}).$$

Moreover, letting $\boldsymbol{\varepsilon}_\ell = (\varepsilon^1_{\ell,\bar{t}^1_1}, \dots, \varepsilon^n_{\ell,\bar{t}^n_{m_n}})^\top \in \mathbb{R}^M$ be a vector of i.i.d. noises (see Equation (21)), we have $\mathbf{Y}_\ell = \mathbf{y}_\ell + \boldsymbol{\varepsilon}_\ell$. The Pythagorean theorem then yields for any $\theta \in \mathbb{R}^{s_d(N)}$,

$$\left\|\mathbf{Y}_\ell - \mathbf{S}_N^{\mathcal{D}}\theta\right\|_2^2 = \left\|\mathbf{y}_\ell - \mathbf{S}_N^{\mathcal{D}}\theta\right\|_2^2 + \|\boldsymbol{\varepsilon}_\ell\|^2 + 2\langle \boldsymbol{\varepsilon}_\ell, y_\ell - \mathbf{S}_N^{\mathcal{D}}\theta\rangle.$$

Applying this equation to $\theta_{[\cdot],\ell}$ and $\widehat{\theta}_{[\cdot],\ell}$, we obtain

$$\frac{1}{2M}\left\|\mathbf{y}_\ell - \mathbf{S}_N^{\mathcal{D}}\widehat{\theta}_{[\cdot],\ell}\right\|_2^2 \leq \frac{1}{2M}\left\|\mathbf{y}_\ell - \mathbf{S}_N^{\mathcal{D}}\theta_{[\cdot],\ell}\right\|_2^2 + \frac{1}{M}\langle \boldsymbol{\varepsilon}_\ell, \mathbf{S}_N^{\mathcal{D}}(\widehat{\theta}_{[\cdot],\ell} - \theta_{[\cdot],\ell})\rangle + \Omega(\theta_{[\cdot],\ell}) - \Omega(\widehat{\theta}_{[\cdot],\ell}). \tag{22}$$

We now work at each layer of the signature matrix $\mathbf{S}_N^{\mathcal{D}}$. Towards that end, we rewrite

$$\mathbf{S}_N^{\mathcal{D}}\big(\widehat{\theta}_{[\cdot],\ell} - \theta_{[\cdot],\ell}\big) = \sum_{k=0}^{N} \mathbf{S}^{\mathcal{D}}_{\cdot,[k]}\big(\widehat{\theta}_{[k],\ell} - \theta_{[k],\ell}\big),$$

and bound

$$\big\langle \boldsymbol{\varepsilon}_\ell, \mathbf{S}_N^{\mathcal{D}}(\widehat{\theta}_{[\cdot],\ell} - \theta_{[\cdot],\ell})\big\rangle = \sum_{k=0}^{N}\big\langle \boldsymbol{\varepsilon}_\ell, \mathbf{S}^{\mathcal{D}}_{\cdot,[k]}(\widehat{\theta}_{[k],\ell)} - \theta_{[k],\ell})\big\rangle \leq \sum_{k=0}^{N}\|\boldsymbol{\varepsilon}_\ell^\top \mathbf{S}^{\mathcal{D}}_{\cdot,[k]}\|_\infty \|\widehat{\theta}_{[k],\ell} - \theta_{[k],\ell}\|_1$$

by $\ell_1 - \ell_\infty$ norms duality. We fix k and study the term $\|\boldsymbol{\varepsilon}_\ell^\top \mathbf{S}^{\mathcal{D}}_{\cdot,[k]}\|_\infty$. Lemma C.2 ensures that each of the words of the signature layer of depth k is bounded by $1/k!$. As a consequence, by Lemma B.7, under Assumption 6, every element of the vector $\boldsymbol{\varepsilon}_\ell^\top \mathbf{S}^{\mathcal{D}}_{\cdot,[k]}$ is $v_\varepsilon M/k!^2$-subgaussian. It follows that, for any real number $\mu > 0$,

$$\mathbb{P}\Big(\|\boldsymbol{\varepsilon}_\ell^\top \mathbf{S}^{\mathcal{D}}_{\cdot,[k]}\|_\infty > \mu\Big) \leq 2d^k \exp\Big(-\frac{(k!)^2\mu^2}{v_\varepsilon M}\Big).$$

We furthermore place ourselves on $A_\varepsilon(\bar{\delta})$ defined by

$$A_\varepsilon(\bar{\delta}) = \bigcap_{\ell=1}^{p}\bigcap_{k=0}^{N}\Big\{\|\boldsymbol{\varepsilon}_\ell^\top \mathbf{S}^{\mathcal{D}}_{\cdot,[k]}\|_\infty \leq \frac{1}{k!}\sqrt{v_\varepsilon M \log(2pNd^k/\bar{\delta})}\Big\}.$$

We have just seen that, under Assumption 6 (and still conditionally on $A_\xi(\delta)$), one has $\mathbb{P}(A_\varepsilon(\bar{\delta})) \geq 1 - \bar{\delta}$. Putting together all terms in Equation (22) and plugging the definition of Ω given in Equation (11), we obtain that, on the set $A_\varepsilon(\bar{\delta}) \cap A_\xi(\delta)$, for all $\theta \in \mathbb{R}^{s_d(N)\times p}$,

$$\begin{aligned}
\frac{1}{2M}\left\|\mathbf{y} - \mathbf{S}_N^{\mathcal{D}}\widehat{\theta}_{N,M}\right\|_{\mathrm{F}}^2 &\leq \frac{1}{2M}\left\|\mathbf{y} - \mathbf{S}_N^{\mathcal{D}}\theta\right\|_{\mathrm{F}}^2 \\
&\quad + \sum_{\ell=1}^{p}\sum_{k=0}^{N}\Big(\frac{1}{M}\|\boldsymbol{\varepsilon}_\ell^\top \mathbf{S}^{\mathcal{D}}_{\cdot,[k]}\|_\infty\|\widehat{\theta}_{[k],\ell} - \theta_{[k],\ell}\|_1 + \frac{C_k(\bar{\delta})}{k!\sqrt{M}}\big(\|\theta_{[k],\ell}\|_1 - \|\widehat{\theta}_{[k],\ell}\|_1\big)\Big) \\
&\leq \frac{1}{2M}\left\|\mathbf{y} - \mathbf{S}_N^{\mathcal{D}}\theta\right\|_{\mathrm{F}}^2 \\
&\quad + \sum_{\ell=1}^{p}\sum_{k=0}^{N}\frac{1}{k!\sqrt{M}}\sqrt{v_\varepsilon \log(2pNd^k/\bar{\delta})}\big(\|\widehat{\theta}_{[k],\ell} - \theta_{[k],\ell}\|_1 + \|\theta_{[k],\ell}\|_1 - \|\widehat{\theta}_{[k],\ell}\|_1\big).
\end{aligned}$$

Choosing $\theta = \theta_N^*$, by the triangular inequality,

$$\|\widehat{\theta}_{[k],\ell} - \theta^*_{[k],\ell}\|_1 + \|\theta^*_{[k],\ell}\|_1 - \|\widehat{\theta}_{[k],\ell}\|_1 \leq 2\|\theta^*_{[k],\ell}\|_1,$$

which finally gives us

$$\begin{aligned}\frac{1}{2M}\left\|\mathbf{y}-\mathbf{S}_N^{\mathcal{D}}\widehat{\theta}_{N,M}\right\|_{\mathrm{F}}^2 &\leq \frac{1}{2M}\left\|\mathbf{y}-\mathbf{S}_N^{\mathcal{D}}\theta_N^*\right\|_{\mathrm{F}}^2+\frac{2}{\sqrt{M}}\sqrt{v_\varepsilon\log(2pNd^N/\bar{\delta})}\sum_{\ell=1}^{p}\sum_{k=0}^{N}\frac{\|\theta^*_{[k],\ell}\|_1}{k!}\\ &\leq \frac{1}{2M}\left\|\mathbf{y}-\mathbf{S}_N^{\mathcal{D}}\theta_N^*\right\|_{\mathrm{F}}^2+\frac{2p}{\sqrt{M}}\sqrt{v_\varepsilon\log(2pNd^N/\bar{\delta})}\sum_{k=0}^{N}\frac{d^k\Lambda_k(\mathbf{F})}{k!}\\ &= \frac{1}{2M}\left\|\mathbf{y}-\mathbf{S}_N^{\mathcal{D}}\theta_N^*\right\|_{\mathrm{F}}^2+\frac{2pC_N(\bar{\delta})}{\sqrt{M}}\sum_{k=0}^{N}\frac{d^k\Lambda_k(\mathbf{F})}{k!},\end{aligned}$$

where the second inequality comes from Proposition C.1. To conclude the proof, we just need to compute the probability of the set $A_\xi(\delta)\cap A_\varepsilon(\bar{\delta})$. It is an immediate consequence of Lemma B.6 that $\mathbb{P}(A_\xi(\delta))\geq 1-\delta$, and we have seen that $\mathbb{P}(A_\varepsilon(\bar{\delta})|A_\xi(\delta))\geq 1-\bar{\delta}$, which yields that

$$\mathbb{P}(A_\xi(\delta)\cap A_\varepsilon(\bar{\delta}))\geq(1-\bar{\delta})(1-\delta).$$

□

C.3. Proof of Lemma 3.2

This proof relies on bouding the remainder of the Taylor expansion of the CDE.

Proof. For every $i=1,\dots,n$ and a given point $t_i\in\bar{D}^i$, one has, using the upper bound of the approximation error of a CDE by its Taylor expansion provided by Fermanian et al. (2021, Proposition 4)

$$\left\|y^i_{t_i}-S_N(x^i_{[0,t_i]})\theta_N^*\right\|\leq\frac{d^{N+1}\Lambda_{N+1}(\mathbf{F})}{(N+1)!}.$$

This immediately gives

$$\frac{1}{M}\left\|\mathbf{y}-\mathbf{S}_N\theta_N^*\right\|_{\mathrm{F}}^2=\frac{1}{M}\sum_{i=1}^{n}\sum_{t_i\in\bar{D}^i}\left\|y^i_{t_i}-S_N(x^i_{[0,t_i]})\theta_N^*\right\|^2\leq\frac{1}{M}\sum_{i=1}^{M}\left(\frac{d^{N+1}\Lambda_{N+1}(\mathbf{F})}{(N+1)!}\right)^2=\left(\frac{d^{N+1}\Lambda_{N+1}(\mathbf{F})}{(N+1)!}\right)^2,$$

which concludes the proof. □

C.4. A layer-wise bound on the signature

We now prove that signature layers are locally Lipschitz mappings. We start with the following proposition.

Proposition C.3. *Let $x\in C_L^{1\text{-var}}([0,1],\mathbb{R}^d)$. Then for all $t\in[0,1]$, the path $t\mapsto\mathbb{X}^k_{[0,t]}$ has 1-variation bounded by*

$$\left\|\mathbb{X}^k\right\|_{1\text{-var},[0,t]}\leq\frac{L^k}{k!}.$$

Proof. By definition of the total variation,

$$\left\|\mathbb{X}^k\right\|_{1\text{-var},[0,t]}=\sup_D\sum_{i=1}^{m}\left\|\mathbb{X}^k_{[0,t_{i+1}]}-\mathbb{X}^k_{[0,t_i]}\right\|_{(\mathbb{R}^d)^{\otimes k}}=\sup_D\sum_{i=1}^{m}\left\|\mathbb{X}^k_{[t_i,t_{i+1}]}\right\|_{(\mathbb{R}^d)^{\otimes k}},$$

since $\mathbb{X}^k_{[0,t]}=\int_0^t dx_{u_1}\otimes\cdots\otimes dx_{u_k}$, and where the supremum is taken over finite dissections $D=\{0=t_1,\dots,t_m=1\}$ of $[0,1]$. Notice that the signature layer of depth k is here written as an element of $(\mathbb{R}^d)^{\otimes k}$, which is more convenient for this proof. Then

$$\sup_D\sum_{i=1}^{m}\left\|\mathbb{X}^k_{[t_i,t_{i+1}]}\right\|_{(\mathbb{R}^d)^{\otimes k}}\leq\sup_D\sum_{i=1}^{m}\frac{\|x\|^k_{1\text{-var},[t_i,t_{i+1}]}}{k!}\leq\frac{1}{k!}\sup_D\left(\sum_{i=1}^{m}\|x\|_{1\text{-var},[t_i,t_{i+1}]}\right)^k=\frac{1}{k!}\sup_D\|x\|^k_{1\text{-var},[0,1]}\leq\frac{L^k}{k!},$$

where the second inequality follows from the multinomial theorem and the last equality comes from the fact that for all $s<u<t$, $\|x\|_{1\text{-var},[s,u]}+\|x\|_{1\text{-var},[u,t]}=\|x\|_{1\text{-var},[s,t]}$. This ends our proof. □

We now state a bound on the difference between the k-th layer of the signatures of two different paths.

Theorem C.4. *Let $x, z \in C_L^{\text{1-var}}([0,1], \mathbb{R}^d)$. Then for all $k \geq 2$, the difference in supremum norm between the paths $t \to \mathbb{X}^k_{[0,t]}$ and $t \to \mathbb{Z}^k_{[0,t]}$ is bounded by*

$$\left\|\mathbb{X}^k - \mathbb{Z}^k\right\|_{\infty,[0,t]} \leq 2L^{k-1} \sum_{j=1}^{k-1} \frac{1}{j!} \|x - z\|_{\infty,[0,t]} \leq 2eL^{k-1} \|x - z\|_{\infty,[0,t]}$$

and

$$\left\|\mathbb{X}^1_{[0,t]} - \mathbb{Z}^1_{[0,t]}\right\| \leq 2 \|x - z\|_{\infty,[0,t]}.$$

Proof. Our proof works by induction. Let $x, z \in C_L^{\text{1-var}}([0,1], \mathbb{R}^d)$, and for $t \in [0,1]$ denote by $\mathbb{X}^k_{[0,t]}$ (resp. $\mathbb{Z}^k_{[0,t]}$) the k-th layer of the signature of x (resp. z). For $k = 1$ and $t \in [0,1]$, remark that

$$\mathbb{X}^1_{[0,t]} - \mathbb{Z}^1_{[0,t]} = \int_0^t d(x_u - z_u) = x_t - z_t - (x_0 - z_0)$$

such that

$$\left\|\mathbb{X}^1_{[0,t]} - \mathbb{Z}^1_{[0,t]}\right\| \leq \|x - z\|_{\infty,[0,t]} + \|x_0 - z_0\| \leq 2 \|x - z\|_{\infty,[0,t]}.$$

Consider now $k \geq 2$. We have

$$\mathbb{X}^k_{[0,t]} - \mathbb{Z}^k_{[0,t]} = \int_0^t \mathbb{X}^{k-1}_{[0,s]} \otimes dx_s - \int_0^t \mathbb{Z}^{k-1}_{[0,s]} \otimes dz_s = \int_0^t \mathbb{X}^{k-1}_{[0,s]} \otimes d(x_s - z_s + z_s) - \int_0^t \mathbb{Z}^{k-1}_{[0,s]} \otimes dz_s,$$

and thus

$$\mathbb{X}^k_{[0,t]} - \mathbb{Z}^k_{[0,t]} = \int_0^t \mathbb{X}^{k-1}_{[0,s]} \otimes d(x_s - z_s) + \int_0^t \left(\mathbb{X}^{k-1}_{[0,s]} - \mathbb{Z}^{k-1}_{[0,s]}\right) \otimes dz_s.$$

We now bound each of these terms separately. First,

$$\left\|\int_0^t \left(\mathbb{X}^{k-1}_{[0,s]} - \mathbb{Z}^{k-1}_{[0,s]}\right) \otimes dz_s\right\|_{(\mathbb{R}^d)^{\otimes k}} \leq \left\|\mathbb{X}^{k-1} - \mathbb{Z}^{k-1}\right\|_{\infty,[0,t]} \|z\|_{\text{1-var},[0,t]} \leq \left\|\mathbb{X}^{k-1} - \mathbb{Z}^{k-1}\right\|_{\infty,[0,t]} L.$$

Moving to the first integral, integration by parts yields

$$\int_0^t \mathbb{X}^{k-1}_{[0,s]} \otimes d(x_s - z_s) = \mathbb{X}^k_{[0,t]} \otimes (x_t - z_t) - \mathbb{X}^k_{[0,0]} \otimes (x_0 - z_0) - \int_0^t (x_s - z_s) \otimes d\mathbb{X}^{k-1}_{[0,s]}.$$

We stress that Proposition (B.2) applies since the integral over the tensor product is taken coordinate-wise. Since $\mathbb{X}^{k-1}_{[0,0]} = 0$, we are left with

$$\int_0^t \mathbb{X}^{k-1}_{[0,s]} \otimes d(x_s - z_s) = \mathbb{X}^k_{[0,t]} \otimes (x_t - z_t) - \int_0^t (x_s - z_s) \otimes d\mathbb{X}^{k-1}_{[0,s]}.$$

Using Lemma C.3 and submultiplicativity of the tensor norms, this can thus be bounded by

$$\begin{aligned}\left\|\int_0^t \mathbb{X}^{k-1}_{[0,s]} \otimes d(x_s - z_s)\right\|_{(\mathbb{R}^d)^{\otimes k}} &\leq \left\|\mathbb{X}^{k-1}_{[0,t]}\right\|_{(\mathbb{R}^d)^{\otimes (k-1)}} \|x - z\|_{\infty,[0,t]} + \|x - z\|_{\infty,[0,t]} \left\|\mathbb{X}^{k-1}\right\|_{\text{1-var},[0,t]} \\ &= \frac{2L^{k-1}}{(k-1)!} \|x - z\|_{\infty,[0,t]}.\end{aligned}$$

Finally, we are left with

$$\left\|\mathbb{X}^k - \mathbb{Z}^k\right\|_{\infty,[0,t]} \leq \frac{2L^{k-1}}{(k-1)!} \|x - z\|_{\infty,[0,t]} + \left\|\mathbb{X}^{k-1} - \mathbb{Z}^{k-1}\right\|_{\infty,[0,t]} L,$$

which can be recursively bounded by

$$\left\|\mathbb{X}^k - \mathbb{Z}^k\right\|_{\infty,[0,t]} \leq 2L^{k-1} \|x - z\|_{\infty,[0,t]} \sum_{j=1}^{k-1} \frac{1}{j!} \leq 2L^{k-1} e \|x - z\|_{\infty,[0,t]}.$$

□

Note that this inequality implies that if z is chosen as the linear interpolation of a discretization of x on a grid D, and if the grid gets finer, all signature layers converge at speed $\|x - z\|_{\infty,[0,t]}$ but the multiplicative constant increases with depth (if $L \geq 1$). Figure 4 illustrates this phenomenon.

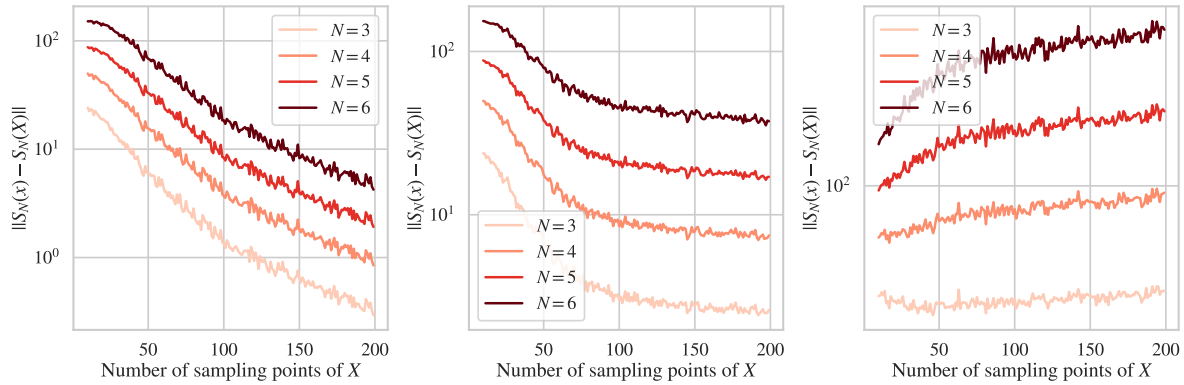

Figure 4: Difference between the signature of a continuous path x and the signature of its discretized and noisy counterpart X, without noise on the discretization points (left), with noise of variance $v_\xi = 0.08^2$ (middle) and with noise of variance $v_\xi = 0.5^2$. For every number of sampling points, we average the distance between the two signature over 50 randomly chosen discretizations of the interval $[0, 1]$. The discretized path is generated as in the well-specified setting (see Appendix D.4).

C.5. Proof of Lemma 3.3

First, recall that for a generic path $x : [0, 1] \to \mathbb{R}^d$, a modulus of continuity is a continuous function $\omega_x : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ vanishing at 0 such that for all $s, t \in [0, 1]$

$$\|x_t - x_s\| \leq \omega_x(|t - s|).$$

Also recall that by Heine's theorem, we can define such a modulus of continuity for every continuous mapping $[0, 1]$ to $\mathbb{R}^d$.

We start by giving a general lemma that bounds the difference between the signature layers of a path and its discretized version. Its proof is based on the results of the previous section.

Lemma C.5. *Let $x \in C_L^{1\text{-var}}([0, 1], \mathbb{R}^d)$ and $\omega_x : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ its modulus of continuity. Let $x^D : [0, 1] \to \mathbb{R}^d$ be the path obtained by linear interpolation of the discretization of x on a grid D corrupted by additive noise ξ. Let $\mathbb{X}^k_{[0,t]}$ and $\mathbb{X}^{k,D}_{[0,t]}$ be their respective k-th layers of signature on $[0, t]$. Then for all $k \geq 2$*

$$\left\|\mathbb{X}^k - \mathbb{X}^{k,D}\right\|_{\infty,[0,1]} \leq 2L^{k-1} \sum_{j=1}^{k-1} \frac{1}{j!} \left(\max_{0 \leq s \leq |D|} \omega_x(s) + \max_{t \in D} \|\xi_t\| \right),$$

and for $k = 1$

$$\left\|\mathbb{X}^1 - \mathbb{X}^{1,D}\right\|_{\infty,[0,1]} \leq 2 \left(\max_{0 \leq s \leq |D|} \omega_x(s) + \max_{t \in D} \|\xi_t\| \right).$$

Proof. Theorem C.4 yields for $k \geq 2$

$$\left\|\mathbb{X}^k - \mathbb{X}^{k,D}\right\|_{\infty,[0,1]} \leq 2L^{k-1} \sum_{j=1}^{k-1} \frac{1}{j!} \left\|x - x^D\right\|_{\infty,[0,t]}$$

Now, remark that

$$\left\|x - x^D\right\|_{\infty,[0,1]} \leq \|x - \tilde{x}\|_{\infty,[0,1]} + \max_{t \in D} \|\xi_t\| \tag{23}$$

from the triangular inequality, where $\tilde{x}$ is the piecewise linear path obtained by linear interpolation of $x_0, x_{t_1}, \dots, x_{t_j}$. Now, since the paths x and $\tilde{x}$ coincide on $0, t_1, \dots, t_j$, we have

$$\|x - \tilde{x}\|_{\infty,[0,1]} = \max_{i=0,\dots,j-1} \|x - \tilde{x}\|_{\infty,[t_i,t_{i+1}]} \leq \max_{i=0,\dots,j-1} \omega_x\big(|t_{i+1} - t_i|\big) = \max_{0\leq s\leq |D|} \omega_x(s).$$

This gives us

$$\big\|\mathbb{X}^k - \mathbb{X}^{k,D}\big\|_{\infty,[0,1]} \leq 2L^{k-1} \sum_{j=1}^{k-1} \frac{1}{j!} \Big(\max_{0\leq s\leq |D|} \omega_x(s) + \max_{t\in D} \|\xi_t\| \Big).$$

For the case $k = 1$, we immediately get

$$\big\|\mathbb{X}^1 - \mathbb{X}^{1,D}\big\|_{\infty,[0,1]} \leq 2\Big(\max_{0\leq s\leq |D|} \omega_x(s) + \max_{t\in D} \|\xi_t\| \Big)$$

using the same technique as above. □

This result is illustrated in Figure 4. One can notice that as predicted by our theoretical bounds, the convergence of signature of high order happens at the same rate than the convergence of signatures of lower order. However, the multiplicative constant controlling the tightness of the bound increases with N, leading to a slower convergence when N increases. Strong noise hinders the convergence of the signature of the discretized path since in this case, the noise's variance is independent of the number of sampling points : adding more sampling points means adding more noise. There are therefore two trade-offs when learning with signatures. A first trade-off is between sampling frequency and order: with paths sampled at low resolution, one should prefer lower order signatures, which trade model complexity against precise features. A second trade-off is between sampling and noise: if the feature time series are very noisy, the precision of the features increases up to a certain point, past which noise prevails.

With this result in hand, we can now prove Lemma 3.3.

Proof. We restrict ourselves to the ω-Lipschitz case.

In our setup, after linearly interpolation the time series to obtain x^D, we normalize it by its total variation $\|x^D\|_{1\text{-var},[0,1]}$, which is a standard practice when learning with signatures (Morrill et al., 2020a). This means that we compute the signature of the path

$$\frac{1}{\|x^D\|_{1\text{-var},[0,t_j]}} x^D. \tag{24}$$

Theorem C.4 gets us for $k \geq 2$

$$\big\|\mathbb{X}^k - \mathbb{X}^{k,D}\big\|_{\infty,[0,1]} \leq 2L^{k-1} \sum_{j=1}^{k-1} \frac{1}{j!} \left\| x - \frac{1}{\|x^D\|_{1\text{-var},[0,1]}} x^D \right\|_{\infty,[0,1]} \tag{25}$$

$$\leq 2L^{k-1} \sum_{j=1}^{k-1} \frac{1}{j!} \big\|x - x^D\big\|_{\infty,[0,1]} + 2L^{k-1} \sum_{j=1}^{k-1} \frac{1}{j!} \left\| x^D - \frac{1}{\|x^D\|_{1\text{-var},[0,1]}} x^D \right\|_{\infty,[0,1]}. \tag{26}$$

The first term can be bounded by using the fact that in our setting, $\omega_x(s) = \omega s$, and we thus get

$$2L^{k-1} \sum_{j=1}^{k-1} \frac{1}{j!} \big\|x - x^D\big\|_{\infty,[0,1]} \leq 2L^{k-1} \sum_{j=0}^{k-1} \frac{1}{j!} \Big(\omega|D| + \max_{t\in D} \|\xi_t\| \Big) \leq 2L^{k-1} e \Big(\omega|D| + \max_{t\in D} \|\xi_t\| \Big) \tag{27}$$

The second term can be bounded by

$$2L^{k-1}\sum_{j=1}^{k-1}\frac{1}{j!}\left\|x^D - \frac{1}{\|x^D\|_{1\text{-var},[0,1]}}x^D\right\|_{\infty,[0,t]} \leq 2L^{k-1}\sum_{j=1}^{k-1}\frac{1}{j!}\left\|\left(1-\frac{1}{\|x^D\|_{1\text{-var},[0,1]}}\right)x^D\right\|_{\infty,[0,t]} \tag{28}$$

$$\leq 2L^{k-1}\sum_{j=1}^{k-1}\frac{1}{j!}\left|1-\frac{1}{\|x^D\|_{1\text{-var},[0,1]}}\right|\left\|x^D\right\|_{\infty,[0,t]}. \tag{29}$$

In order to bound

$$\left|1-\frac{1}{\|x^D\|_{1\text{-var},[0,1]}}\right| = \left|\frac{\|x^D\|_{1\text{-var},[0,1]}-1}{\|x^D\|_{1\text{-var},[0,1]}}\right|,$$

we need both an upper and a lower bound on $\|x^D\|_{1\text{-var},[0,1]}$.

Remark that

$$\|x^D\|_{1\text{-var},[0,t_j]} = \sum_{t_u,t_{u-1}\in D}\left\|x_{t_u}+\xi_{t_u}-x_{t_{u-1}}-\xi_{t_{u-1}}\right\|. \tag{30}$$

Recall that we assume the path (x_t) to be time-augmented, and that the measurement times are not noisy. This means that

$$\sum_{t_u,t_{u-1}\in D}\left\|x_{t_u}+\xi_{t_u}-x_{t_{u-1}}-\xi_{t_{u-1}}\right\| \geq \sum_{t_u,t_{u-1}\in D}|t_u-t_{u-1}| \geq t_2-t_1 = t_2 \tag{31}$$

since $t_1=0$ and Assumption 4 guarantees that there are at least two sampling points in every grid. This gives us that

$$\frac{1}{\|x^D\|_{1\text{-var},[0,1]}} \leq \frac{1}{t_2} \leq \frac{1}{\eta}, \tag{32}$$

since we have required that the last sampling time is at least η in Assumption 4. Turning to the upper bound, we get that

$$\left|1-\|x^D\|_{1\text{-var},[0,1]}\right| \leq 1-L+\sum_{t_u,t_{u-1}\in D}\|\xi_u-\xi_{u-1}\|$$

by definition of the total variation of a piecewise linear path. Finally,

$$\sum_{t_u,t_{u-1}\in D}\|\xi_u-\xi_{u-1}\| \leq 2\#D\max_{t\in D}\|\xi_t\|. \tag{33}$$

Putting everything together gives us

$$\left|1-\frac{1}{\|x^D\|_{1\text{-var},[0,1]}}\right| \leq \frac{1-L+2\#D\max_{t\in D}\|\xi_t\|}{\eta}. \tag{34}$$

In the end, we get that

$$2L^{k-1}\sum_{j=1}^{k-1}\frac{1}{j!}\left|1-\frac{1}{\|x^D\|_{1\text{-var},[0,1]}}\right|\left\|x^D\right\|_{\infty,[0,1]} \leq 2L^{k-1}e\frac{1-L+2\#D\max_{t\in D}\|\xi_t\|}{\eta}\left\|x^D\right\|_{\infty,[0,1]}. \tag{35}$$

Now, remark that since the path x^D is piecewise linear,

$$\|x^D\|_{\infty,[0,1]} = \|x^D-x_0+x_0\|_{\infty,[0,1]} \leq \max_{t\in D}\|x_t+\xi_t-x_0\|+\|x_0\| \tag{36}$$

$$\leq \max_{t\in D}\|x_t-x_0\|+\max_{t\in D}\|\xi_t\|+\|x_0\| \tag{37}$$

$$\leq \|x_0\|+L+\max_{t\in D}\|\xi_t\| \tag{38}$$

where the inequality

$$\max_{t\in D}\|x_t - x_0\| \leq L$$

follows from the definition of the total variation.

This means that

$$2L^{k-1}\sum_{j=1}^{k-1}\frac{1}{j!}\Big|1-\frac{1}{\|x^D\|_{\text{1-var},[0,1]}}\Big|\,\|x^D\|_{\infty,[0,1]} \leq 2L^{k-1}e\frac{1-L+2\#D\max_{t\in D}\|\xi_t\|}{\eta}\Big(\|x_0\|+L+\max_{t\in D}\|\xi_t\|\Big). \tag{39}$$

We have written these inequalities for a generic random variable. Let us now consider individual observations of our dataset.

On the set $A_\xi(\delta)$, one has

$$\max_{i=1,\dots,n,t\in D^i}\|\xi_t^i\| \leq v_\xi\sqrt{d}+v_\xi\sqrt{c^{-1}\log(\delta^{-1}\#\mathcal{D})}. \tag{40}$$

To simplify notations, let us write

$$C_\delta := v_\xi\sqrt{d}+v_\xi\sqrt{c^{-1}\log(\delta^{-1}\#\mathcal{D})}. \tag{41}$$

We get

$$2L^{k-1}\sum_{j=1}^{k-1}\frac{1}{j!}\|x^i-x^{i,D}\|_{\infty,[0,1]} \leq 2L^{k-1}e\Big(\omega|\mathcal{D}|+C_\delta\Big), \tag{42}$$

where we recall that $\mathcal{D}$ is the collection of individual grids, and $|\mathcal{D}|$ is the biggest sampling gap among individuals. Similarly,

$$2L^{k-1}\sum_{j=1}^{k-1}\frac{1}{j!}\Big|1-\frac{1}{\|x^D\|_{\text{1-var},[0,1]}}\Big|\,\|x^D\|_{\infty,[0,1]} \leq 2L^{k-1}e\frac{1-L+2\#\mathcal{D}C_\delta}{\eta}\big(\|x_0\|+L+C_\delta\big) \tag{43}$$

Now moving to the feature matrices, we have

$$\begin{aligned}
\frac{1}{M}\big\|(\mathbf{S}_N-\mathbf{S}_N^{\mathcal{D}})\theta_N^*\big\|_F^2 &\leq \frac{1}{M}\sum_{i=1}^n\sum_{k=0}^N\Big\|(\mathbf{S}_{i,[k]}-\mathbf{S}_{i,[k]}^{\mathcal{D}})\theta_{[k],\cdot}^*\Big\|_F^2\\
&\leq \frac{1}{M}\sum_{i=1}^n\sum_{t\in\bar{D}^i}\sum_{k=0}^N d^k\Lambda_k(\mathbf{F})^2\big(2eL^{k-1}\big(\omega|\mathcal{D}|+C_\delta+\frac{1-L+2\#\mathcal{D}C_\delta}{\eta}\big(\|x_0\|+L+C_\delta\big)\big)\Big)^2\\
&\leq 4e^2\left(\omega|\mathcal{D}|+C_\delta+\frac{1-L+2\#\mathcal{D}C_\delta}{\eta}\big(\|x_0\|+L+C_\delta\big)\right)^2L^2\sum_{k=0}^N\frac{d^k\Lambda_k(\mathbf{F})^2}{k!^2}\times k!^2\\
&\leq 4e^2N!^2\left(\omega|\mathcal{D}|+C_\delta+\frac{1-L+2\#\mathcal{D}C_\delta}{\eta}\big(\|x_0\|+L+C_\delta\big)\right)^2L^2\sum_{k=0}^N\frac{d^k\Lambda_k(\mathbf{F})^2}{k!^2}.
\end{aligned}$$

Writing

$$C_{\mathcal{D},N}(\delta) = 4e^2L^2N!^2\left(\omega|\mathcal{D}|+C_\delta+\frac{1-L+2\#\mathcal{D}C_\delta}{\eta}\big(\|x_0\|+L+C_\delta\big)\right)^2,$$

one finally gets with probability $1-\delta$ that

$$\frac{1}{M}\big\|(\mathbf{S}_N-\mathbf{S}_N^{\mathcal{D}})\theta_N^*\big\|_F^2 \leq C_{\mathcal{D},N}(\delta)\sum_{k=0}^N\frac{d^k\Lambda_k(\mathbf{F})^2}{k!^2}.$$

□

C.6. Proof of the main Theorem

We finally combine all Lemmas to obtain the desired oracle bound.

Proof. First, we have from Lemma 3.1 that on $A_\varepsilon(\bar{\delta})$,

$$\frac{1}{2M}\left\|\mathbf{y}-\mathbf{S}_N^{\mathcal{D}}\widehat{\theta}_{N,M}\right\|_{\mathrm{F}}^2 \leq \frac{1}{2M}\left\|\mathbf{y}-\mathbf{S}_N^{\mathcal{D}}\theta_N^*\right\|_{\mathrm{F}}^2 + \frac{2pC_N(\bar{\delta})}{\sqrt{M}}\sum_{k=0}^{N}\frac{d^k\Lambda_k(\mathbf{F})}{k!}.$$

The first term of the right-hand side of this inequality is bounded by

$$\frac{1}{2M}\left\|\mathbf{y}-\mathbf{S}_N^{\mathcal{D}}\theta_N^*\right\|_{\mathrm{F}}^2 \leq \frac{1}{M}\left\|\mathbf{y}-\mathbf{S}_N\theta_N^*\right\|_{\mathrm{F}}^2 + \frac{1}{M}\left\|\mathbf{S}_N\theta_N^*-\mathbf{S}_N^{\mathcal{D}}\theta_N^*\right\|_{\mathrm{F}}^2.$$

By Lemma 3.2 and Lemma 3.3, this can in turn be bounded on $A_\varepsilon(\bar{\delta})\cap A_\xi(\delta)$ by

$$\frac{1}{2M}\left\|\mathbf{y}-\mathbf{S}_N^{\mathcal{D}}\theta_N^*\right\|_{\mathrm{F}}^2 \leq \left(\frac{d^{N+1}\Lambda_{N+1}(\mathbf{F})}{(N+1)!}\right)^2 + C_{\mathcal{D},N}(\delta)\sum_{k=0}^{N}\frac{d^k\Lambda_k(\mathbf{F})^2}{k!^2}$$

Combining all the pieces, this finally gives us, on $A_\varepsilon(\bar{\delta})\cap A_\xi(\delta)$,

$$\begin{aligned}\frac{1}{2M}\left\|\mathbf{y}-\mathbf{S}_N^{\mathcal{D}}\widehat{\theta}_{N,M}\right\|_{\mathrm{F}}^2 \leq{}& \left(\frac{d^{N+1}\Lambda_{N+1}(\mathbf{F})}{(N+1)!}\right)^2 \\ &+ C_{\mathcal{D},N}(\delta)\sum_{k=0}^{N}\frac{d^k\Lambda_k(\mathbf{F})^2}{k!^2} \\ &+ \frac{2pC_N(\bar{\delta})}{\sqrt{M}}\sum_{k=0}^{N}\frac{d^k\Lambda_k(\mathbf{F})}{k!}.\end{aligned}$$

□

C.7. Asymptotics

We briefly discuss the asymptotic behaviour of the upper bound of the oracle inequality.

Truncation depth N. A natural question is whether the bias of our estimator vanishes as $N\to\infty$. If we have perfect sampling, i.e. the limit case where $\mathcal{D}=0$ and $v_\xi=0$, our bound on the prediction error becomes on $A_\varepsilon(\bar{\delta})$

$$\begin{aligned}\frac{1}{2M}\left\|\mathbf{y}-\mathbf{S}_N^{\mathcal{D}}\widehat{\theta}_{N,M}\right\|_{\mathrm{F}}^2 \leq{}& \left(\frac{d^{N+1}\Lambda_{N+1}(\mathbf{F})}{(N+1)!}\right)^2 \\ &+ \frac{2pC_N(\bar{\delta})}{\sqrt{M}}\sum_{k=0}^{N}\frac{d^k\Lambda_k(\mathbf{F})}{k!}.\end{aligned}$$

The first term of this bound vanishes as an immediate consequence of Assumption 3, while the second term is a statistical error term that behaves like $\frac{\sqrt{\log(Nd^N)}}{\sqrt{M}}$. In order to obtain an asymptotic convergence, we thus need that $N\log(dN)=o(M)$.

In the more realistic setting where $|\mathcal{D}|>0$, the discretization bias behaves like $L^{N-1}N!|\mathcal{D}|$. It is thus sufficient to assume that $|D|=o(1/N!)$. If $v_\xi>0$, our estimator is durably biased due to the measurement noise, and this bias increases with $N\to\infty$. This is due to a "propagation of chaos" phenomenon: the difference between the unobserved feature path and the interpolated feature time series is amplified by taking the successive iterated integrals that define the signature. This advocates for using simple, low-order signature models in the presence of noise, as the gain in precision obtained when taking higher N and reducing the truncation bias will at some point be lost because of the amplified noise.

Dimension p of the target path. Our oracle bound only depends on p through the statistical error term. This term is proportional to $p\sqrt{\log p}$, which is expected in multitask regression.

Dimension d of the feature path. Our oracle bound exhibits multiple dependencies in d. First, the truncation bias grows polynomially with d. Similarly, the discretization bias also depends polynomially on d. Finally, the statistical error term is proportional to $\log d$ times a polynomial term.

D. Algorithms, experiments, and supplementary results

D.1. Implementation details

Recall that the SigLasso estimator $\widehat{\theta}_{N,M}$ is defined as

$$\widehat{\theta}_{N,M} \in \underset{\theta \in \mathbb{R}^{s_d(N)\times p}}{\arg\min} \frac{1}{2M} \left\| \mathbf{Y} - \mathbf{S}_N^{\mathcal{D}} \theta \right\|_{\mathrm{F}}^2 + \Omega(\theta),$$

where

$$\Omega(\theta) = \sum_{k=0}^{N} \frac{M^{\frac{1}{2}} C_k(\bar{\delta})}{k!} \left\| \theta_{[k],\cdot} \right\|_1,$$

and

$$C_k(\bar{\delta}) = \sqrt{v_\varepsilon \log(2pNd^k/\bar{\delta})}$$

for $\bar{\delta} \in [0,1]$. Our goal is first to rewrite the penalty $\Omega(\theta)$ as

$$\Omega(\theta) = C \sum_{k=0}^{N} \lambda_k \left\| \theta_{[k],\cdot} \right\|_1,$$

such that training will only require to scale each layer of θ and to crossvalidate the multiplicative constant C. Since for $k \geq 1$,

$$C_k(\bar{\delta}) = \sqrt{k} \times \sqrt{v_\varepsilon(\bar{\delta}/k + \log(pN)/k + \log d)} \leq \sqrt{k} \times \sqrt{v_\varepsilon(\bar{\delta} + \log(pN) + \log d)},$$

we let $\lambda_k = \frac{\sqrt{k}}{k!}$.

We now show that the minimization problem with layer-specific penalty can be written as a standard regression problem with ℓ_1 penalization by rescaling the feature matrix, that is, multiplying $\mathbf{S}_N^{\mathcal{D}}$ by a well-chosen diagonal matrix. Consider the ℓ_1-penalized problem

$$\min_{\theta \in \mathbb{R}^{s_d(N)\times p}} \frac{1}{2M} \left\| \mathbf{Y} - \mathbf{S}_N^{\mathcal{D}} \theta \right\|_{\mathrm{F}}^2 + C \sum_{k=0}^{N} \lambda_k \left\| \theta_{[k],\cdot} \right\|_1,$$

where $C > 0$ controls the strength of the penalization.

Making the change of variable

$$\tilde{\theta} = \mathrm{diag}(1, \underbrace{\lambda_1, \ldots, \lambda_1}_{d \text{ repetitions}}, \underbrace{\lambda_2, \ldots, \lambda_2}_{d^2 \text{ repetitions}}, \ldots, \underbrace{\lambda_k, \ldots, \lambda_k}_{d^N \text{ repetitions}}) \theta,$$

which is equivalent to

$$\theta = \mathrm{diag}(1, 1/\lambda_1, \ldots, 1/\lambda_1, \ldots, 1/\lambda_k, \ldots, 1/\lambda_k) \tilde{\theta},$$

and denoting by W this last weight matrix, we get the equivalent minimization problem

$$\min_{\tilde{\theta}\in\mathbb{R}^{s_d(N)\times p}} \frac{1}{2M}\left\|\mathbf{Y}-\mathbf{S}_N^{\mathcal{D}}W\tilde{\theta}\right\|_{\mathrm{F}}^2 + C\sum_{k=0}^{N}\left\|\tilde{\theta}_{[k],\cdot}\right\|_1.$$

We can thus obtain the SigLasso estimator by (i) multiplying the feature matrix $\mathbf{S}_N^{\mathcal{D}}$ by W and solving the associated ℓ_1-penalized problem (ii) multiplying the obtained solution by W.

The Learn-And-Reconstruct algorithm is the generic algorithm used in our work. It is applicable for a wide variety of tasks such as missing values inference, trajectory reconstruction, forecasting and many more. It is described in Algorithm 1.

Algorithm 1 Learn-and-Reconstruct Algorithm. The algorithm infers for every individual in the test set a reconstructed time series $\hat{Y}_t^i$.

1. Learn the dynamics
Input: train dataset of normalized paths $(\mathbf{X}^1,\mathbf{Y}^1),\dots,(\mathbf{X}^n,\mathbf{Y}^n)$ sampled on $(D^1,\bar{D}^1),\dots,(D^n,\bar{D}^n)$.
Construct the feature matrix $\mathbf{S}_N^{\mathcal{D}}$ and the target vector $\mathbf{Y}$
for $i=1$ **to** n **do**
  for t in $\bar{D}^i$ **do**
    $\mathbf{S}_N^{\mathcal{D}} \leftarrow$ Append $S_N(X^i_{[0,t]})$
    $\mathbf{Y} \leftarrow$ Append Y_t^i
  end for
end for
Compute $\hat{\theta}_{N,M}$ by solving (7) with $\mathbf{Y},\mathbf{S}_N^{\mathcal{D}}$ using coordinate descent.
2. Reconstruct trajectories
Input: test dataset $\tilde{\mathbf{X}}^1,\dots,\tilde{\mathbf{X}}^n$ sampled on $\tilde{D}^1,\dots,\tilde{D}^n$.
for $i=1$ **to** n **do**
  for t **in** $\tilde{D}^i$ **do**
    $\hat{Y}_t^i = \hat{\theta}_{N,M}S_N(\tilde{X}^i_{[0,t]})$
  end for
end for

D.2. Assessing feature importance

We two metrics used to asses the importance of the different dimensions of the feature path.

Given a truncation depth N and a dimension $i\in\{1,\dots,d\}$, we define its pure feature importance (PFI) as the sum of the norm of the coefficients (or vectors in the case the target is multivariate) of $\widehat{\theta}_{N,M}$ that are associated to signatures taken on the words $I_1=(i)$, $I_2=(i,i)$, and so forth until $I_N=(i,\dots,i)$. Mathematically,

$$PFI(i) = \frac{1}{N}\Big(\left\|\theta^{I_1}\right\|_2 + \left\|\theta^{I_2}\right\|_2 + \cdots + \left\|\theta^{I_N}\right\|_2\Big).$$

Since signatures also capture interactions between dimensions of the feature path, we also define the cross feature importance (CFI) as the sum of norms of the coefficients (or vectors) of $\widehat{\theta}_{N,M}$ that are associated to signatures coefficients of words of length $\leq N$ in which the letter i appears. Mathematically,

$$CFI(i) = \frac{1}{s_d(N)-s_{d-1}(N)}\sum_{I \text{ s.t. } i\in I}\left\|\theta^I\right\|_2.$$

For a given truncation depth N, note that there are $s_d(N)-s_{d-1}(N)=\sum_{k=0}^{N}d^k-\sum_{k=0}^{N}(d-1)^k$ terms in the last sum, which justifies our choice of normalization.

D.3. Details on model implementation and evaluation

SigLasso. The SigLasso model is implemented using the `CVLasso` class in `scikit-learn` (Pedregosa et al., 2011). This implementation optimises the objective function using coordinate descent and features automatic cross-validation of the penalty strength. We use `iisignature` (Reizenstein & Graham, 2020) to compute the signature of the feature time series. Every time series is standardized prior to this through division by its own total variation, as suggested by Morrill et al. (2020a). The depth of the signature is a hyperparameter chosen between 2 and 9 or 6 depending on the experiment. An intercept is added.

GRU. The GRU is of width 128 and systematically trained with 100 epoches using a learning rate of 0.001.

Neural CDE. We use the implementation of Neural CDE provided by `torchcde` (Kidger et al., 2020). We use the original vector field described in the documentation of this package, with the small tweak that we use a smoother non-linearity (tanh instead of ReLU). We observed that using the `rk4` solver instead of `dopri5` significantly accelerates the training time of the Neural CDE without affecting the model's performances. The learning rate is hand-tuned to either 0.001 or 0.0001 depending on the experiment. We train the model for 100 epochs and asses its convergence by using a standard stopping criteria.

Metrics. The MSE is computed in a classical fashion. To compute the integrate MSE, we compute the L^2 distance between the piecewise constant interpolations of the true y_t and the predicted $\widehat{y}_t$.

D.4. Details on the well specified model

Generation of the training data. We generate a two-dimensional feature path by interpolating for every dimension 15 points in $[0, 1]$, each of them being draw randomly for a normal distribution $\mathcal{N}(0, 1)$. The interpolation is done with Hermite cubic splines with backward differences using the package `torchcde` (Kidger et al., 2020). Time is added as a supplementary channel, which is a standard practice when learning with signatures and Neural CDEs. These paths are then downsampled by randomly drawing sampling points for the target and the feature time series specific to every individual. The target path is the solution of a CDE of the form

$$dy_t = \sigma\Big(Ay_t\Big)dx_t$$

where $\sigma : \mathbb{R}^p \to \mathbb{R}^p$ is the hyperbolic tangent, $A \in \mathbb{R}^{d\times p}$ is a matrix drawn randomly from $\mathcal{N}(0, I_{d\times p})$ and x_t is the feature path constructed as above. The solution of this CDE is computed using `torchcde`.

Generation of the test data. We generate the test data in the same way than the training data. However, this data is not downsampled as we wish to assess the generalization capacities of our model—i.e., is our model capable of approximating the dynamics and extrapolating to continuous feature paths.

D.5. Details on Ornstein-Uhlenbeck experiment

We take $(x_t)_{t\in[0,1]}$ to be a 1-dimensional Brownian motion with variance $\sigma^2 = 0.1$, and generate (y_t) as a 1-dimensional Ornstein-Uhlenbeck process driven by (x_t), that is, for all $t \in [0, 1]$

$$dy_t = \theta(\mu - y_t)dt + dx_t.$$

Simulation of (y_t) is done using a standard Euler-Maruyama simulation scheme. We let $\theta = 3$ and $\mu = 1$. The training data is then downsampled as in the well specified experiment.

D.6. Details on the tumor growth experiment

We consider the following tumor growth model taken from (Simeoni et al., 2004). Let $x \in C^{1-var}([0,1],\mathbb{R})$. The weight $y \in C([0,1],\mathbb{R}^+)$ under the concentration of a treatment drug x is governed by the differential system

$$\begin{aligned}
du_t^1 &= \left[\left(\lambda_0 u_t^1\left[1+(\frac{\lambda_0}{\lambda_1}y_t)^\psi\right]\right)^{-1/\psi} - k_2 x_t u_t^1\right]dt \\
du_t^2 &= \left[k_2 x_t u_t^1 - k_1 u_t^2\right]dt \\
du_t^3 &= \left[k_1(u_t^2 - u_t^3)\right]dt \\
du_t^4 &= \left[k_1(u_t^3 - u_t^4)\right]dt \\
y_t &= u_t^1 + u_t^2 + u_t^3 + u_t^4
\end{aligned}$$

with initial condition $(u_0^1, u_0^2, u_0^3, u_0^4, y_0) = (2,0,0,0,2)$ and parameters $(k_1, k_2, \lambda_0, \lambda_1, \psi) = (10, 0.5, 0.9, 0.7, 20)$. The concentration (x_t) is chosen to be the squared value of the paths used for the well-specified experiment. Notice that this system is non-linear w.r.t. x. Indeed, writing $dy_t = \mathbf{G}(y_t, x_t)dt$, one has, for $\alpha \in \mathbb{R}$, $G(y_t, \alpha x_t) \neq \alpha G(y_t, x_t)$. The training data is then downsampled as in the well specified experiment.

D.7. Supplementary results

Table 2: Performance of SigLasso, GRU, Neural CDE, RNN and LSTM in different simulation settings, averaged over 10 iterations. In every setting, $n = 50$, $\#\bar{D}^i = 5$ for all $i = 1, \dots, n$ (and therefore $M = 250$).

	L_2 error					MSE on last point				
Setting	SigLasso	GRU	Neural CDE	RNN	LSTM	SigLasso	GRU	Neural CDE	RNN	LSTM
Well-specified	**0.13 ± 0.07**	1.05 ± 0.42	0.61 ± 0.38	1.16 ± 0.45	0.87 ± 0.67	**0.73 ± 0.56**	3.32 ± 1.60	1.46 ± 1.20	3.56 ± 1.43	2.41 ± 1.75
Ill-specified	**0.15 ± 0.02**	0.24 ± 0.11	0.29 ± 0.15	0.18 ± 0.006	0.20 ± 0.01	**0.09 ± 0.05**	0.19 ± 0.09	0.22 ± 0.15	0.18 ± 0.05	0.10 ± 0.03
OU	**0.01 ± 0.02**	0.05 ± 0.06	0.17 ± 0.12	0.11 ± 0.09	0.46 ± 0.48	0.018 ± 0.025	0.014 ± 0.020	**0.013 ± 0.016**	0.02 ± 0.02	4.41 ± 3.77
Tumor growth	**0.16 ± 0.02**	0.66 ± 0.09	5.29 ± 1.38	0.75 ± 0.03	0.69 ± 0.04	**0.35 ± 0.12**	2.00 ± 0.38	8.76 ± 9.26	2.72 ± 0.24	2.25 ± 0.29

D.8. Details on the French Covid experiment

We illustrate the performance of our method and competitors on French Covid data from 2021-03-31 to 2021-07-07 available on Gitlab. Hospital data was obtained from the SI-VIC database, the national inpatient surveillance system.

Target path. Following Paireau et al. (2022), we chose the predict the growth rate of incident hospitalisations in each of the 9 metropolitan regions of France. The exponential growth rate was computed from raw data using a 2 days rolling window and then smoothed using local polynomial regression as in Paireau et al. (2022). Mathematically, our target time series is the $\mathbb{R}$-valued growth rate, and we fit a different model for every of the 12 regions. It is displayed for all 12 regions in Figure 6.

Feature path. As in Paireau et al. (2022), we consider a set of 12 time-dependant predictors of different types summarized in Table 3 and plotted in Figure 5. Both SIDEP ("Système d'Information de Dépistage Populationnel") and VAC-SI datasets are publicly available. The mobility data was obtained from Google. The mobility-related predictors describe travel trends for different kind of public spaces such as such as shops and leisure spaces, food stores and pharmacies, parks, public transport stations, workplaces and residential areas. The meteorological data was obtained from Météo France.

Models. SigLasso, NCDE and GRU algorithms were trained on the period from 2021-03-31 to 2021-06-23 and tested on the period from 2021-06-24 to 2021-07-07. We included a history of 10 days at each point and performed prediction for different horizons ranging from 1 to 14. In others words, at horizon h, features values from day $t-h-10$ to day $t-h$ to were used to compute the prediction at time t. All feature time series are normalized to have total variation equal to 1.

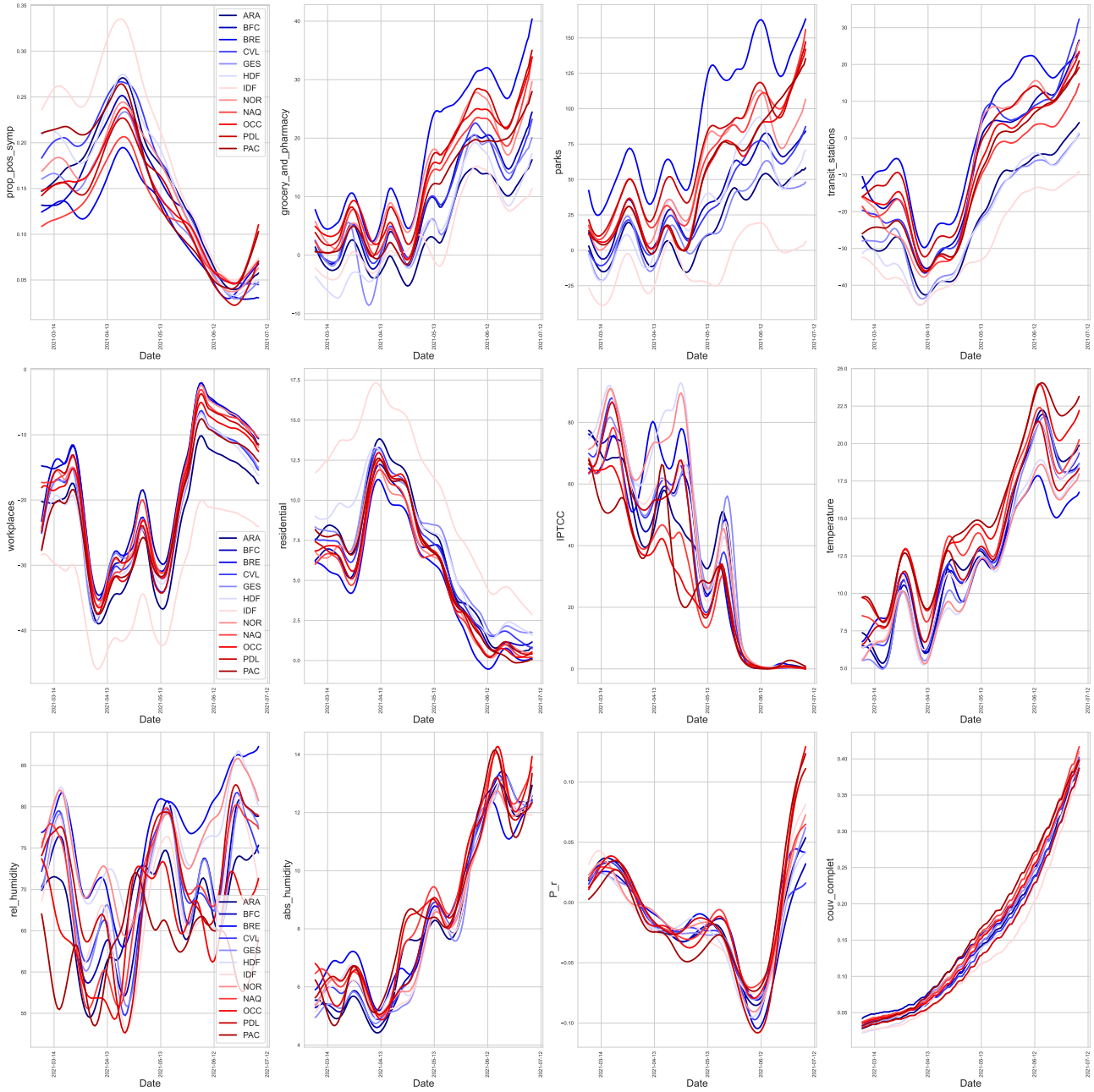

Figure 5: The 12 different feature time series used to forecast the hospitalization growth rate. Every different color corresponds to a given region of France.

Predictor	Type	Source	Description
prop_pos_symp	Epidemiological	SIDEP database	proportion of positive tests among symptomatics
P_r	Epidemiological	SIDEP database	growth rate of positive tests
couv-complet	Epidemiological	VAC-SI database	proportion of vaccinated
grocery_and_pharmacy	Mobility	Google	visits to grocery and pharmacy stores
parks	Mobility	Google	visits to parks
transit_stations	Mobility	Google	visits to transit stations
workplaces	Mobility	Google	visits to workplaces
residential	Mobility	Google	visits to residential places
IPTCC	Meteorological	Météo France	Index PREDICT of climatic transmissivity
temperature	Meteorological	Météo France	temperature
rel_humidity	Meteorological	Météo France	relative humidity
abs_humidity	Meteorological	Météo France	absolute humidity

Table 3: The set of time-dependant predictors used to predict the hospital admission growth rate

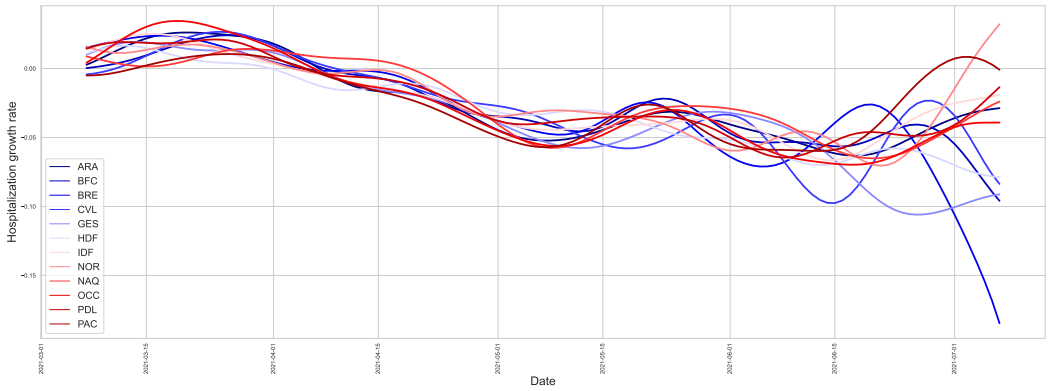

Figure 6: Hospitalization growth rate through time during the full period for the 12 different regions of France.

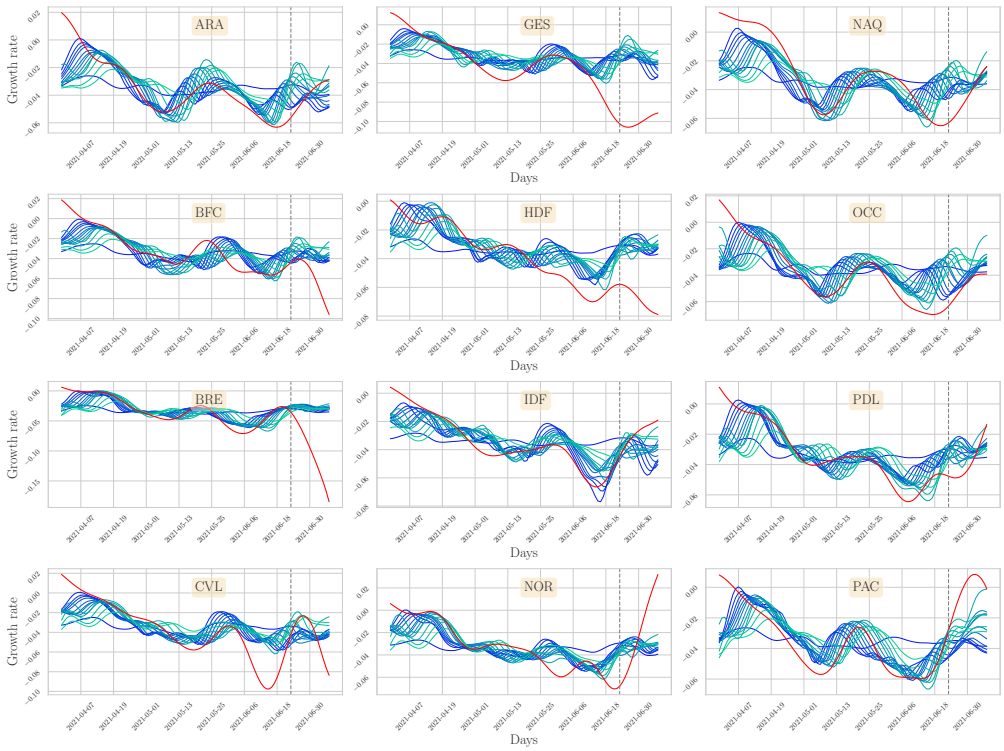

Figure 7: Interpolation (left of dotted line) and prediction (right of dotted line) of hospitalization growth rate for all 12 french regions using **SigLasso**.

Architectural details. The GRU has width 128 and is trained for 100 epochs with a learning rate of 0.0001. The NCDE is trained for 30 epochs with a learning rate of 0.001. It has 2 hidden layers of width 128, an intermediate $\mathrm{Tanh}(\cdot)$ nonlinearity and a final linear readout. This architecture is identical to the one proposed in (Kidger, 2022). Penalty strenght of the SigLasso is crossvalided using the internal implementation `LassoCV` of `scikit-learn` (Pedregosa et al., 2011).

All details, in particular the features used for each individual prediction, can be found in Paireau et al. (2022).

We refer to the supplementary information file of Paireau et al. (2022) and our code for more details.

Results. Figure 11 displays the RMSE (on all regions) of NCDE, SigLasso, GRU, and the Ensemble method for all prediction horizons $h = 1, \dots, 14$. Figures 7, 9 and 8 display the obtained interpolation for SigLasso, GRU and NCDE at different horizons (corresponding to different line colors in `winter` matplotlib palette). The lighter the blue, the smaller the time horizon: the lightest curve corresponds to a time horizon equal to $h = 1$. Truth is in red.

D.9. Additional results

We give in Table 4 some additional results on the experiments described above.

Table 4: Training time of SigLasso, GRU and Neural CDE in different simulation settings, averaged over 10 iterations. In every setting, $n = 50$, $\#\bar{D}^i = 5$ for all $i = 1, \dots, n$ (and therefore $M = 250$).

	Training time (s)		
	SigLasso	GRU	Neural CDE
Well-specified	0.37 ± 0.23	269 ± 109	1754 ± 587
OU	0.057 ± 0.005	27 ± 0.44	216 ± 2.7
Tumor growth	0.056 ± 0.007	31 ± 3.5	250 ± 14

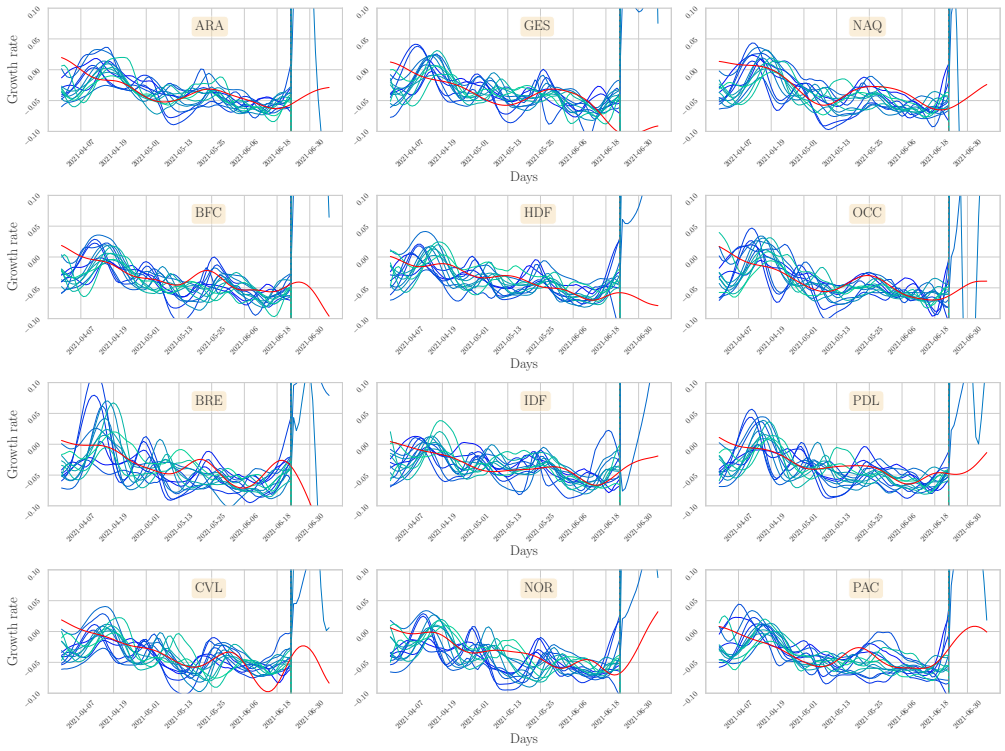

Figure 8: Interpolation (left of dotted line) and prediction (right of dotted line) of hospitalization growth rate for all 12 french regions using **NCDE**.

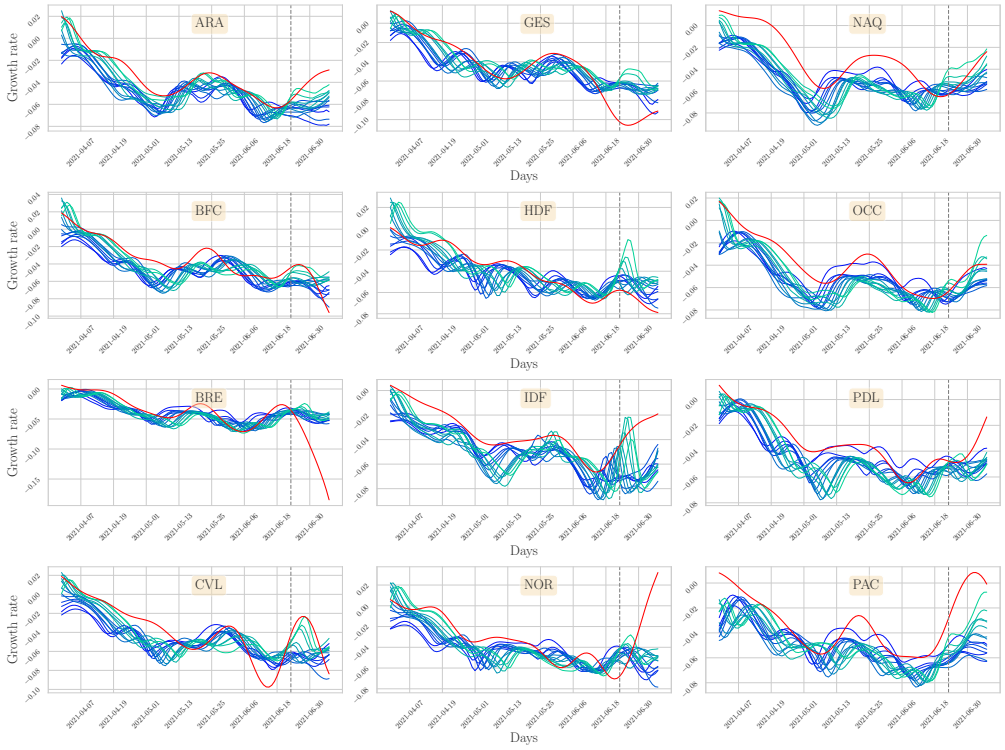

Figure 9: Interpolation (left of dotted line) and prediction (right of dotted line) of hospitalization growth rate for all 12 french regions using **GRU**.

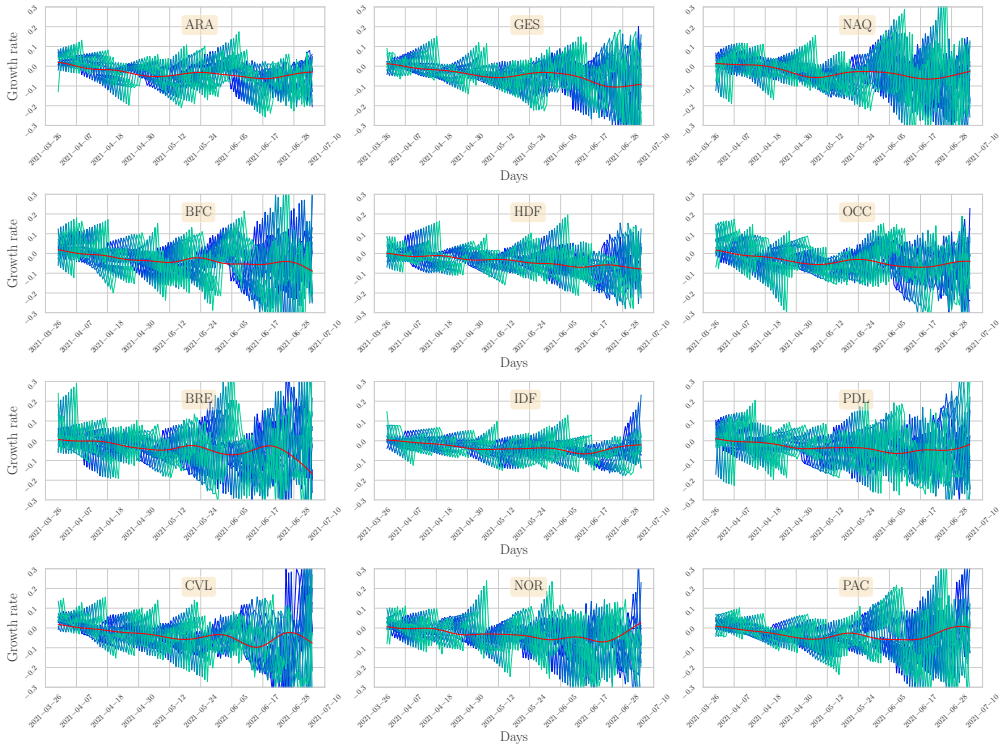

Figure 10: Interpolation (left of dotted line) and prediction (right of dotted line) of hospitalization growth rate for all 12 french regions using **ensemble methods** (Paireau et al., 2022).

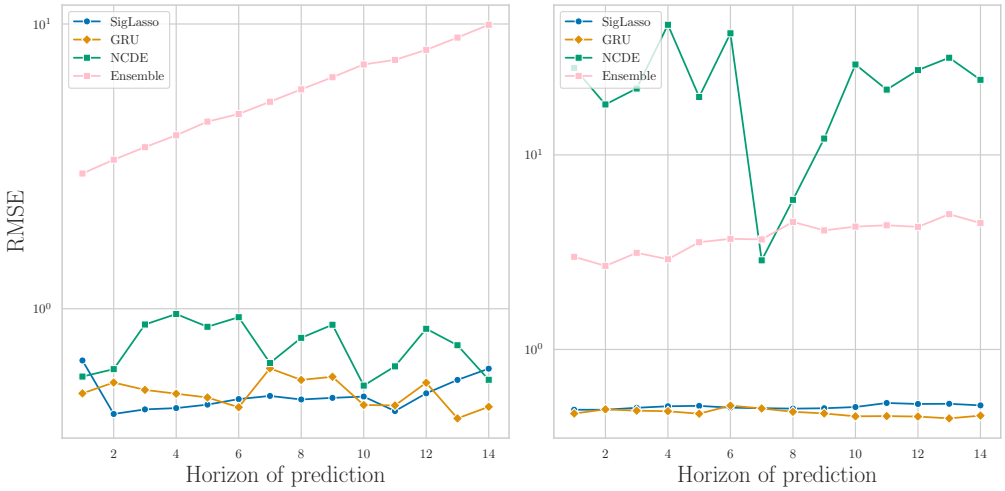

Figure 11: RMSE accross all regions on the **training period** (left) and the **testing period** (right) for the ensemble methode (Paireau et al., 2022), NCDE, GRU, and SigLasso. See Figure 12 for a zoom-in on GRU and SigLasso performances.

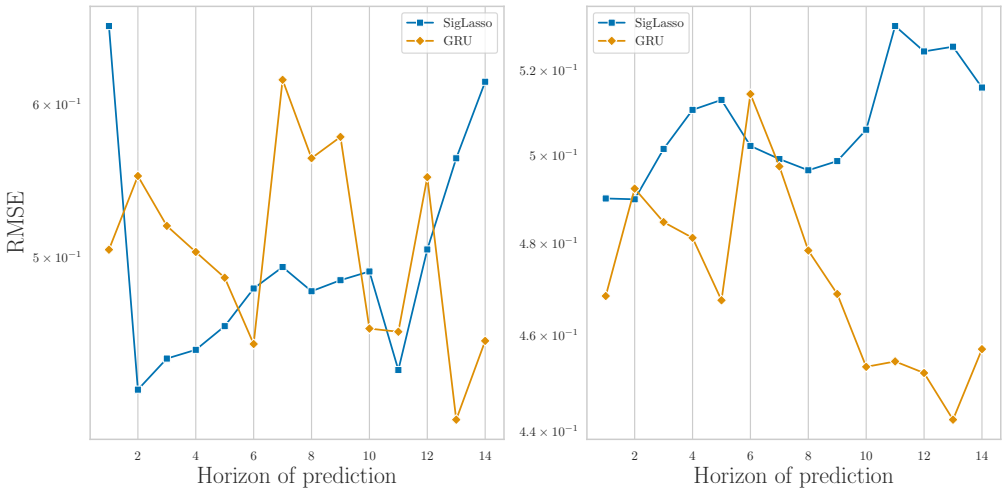

Figure 12: RMSE accross all regions on the **training period** (left) and the **testing period** (right) for GRU and SigLasso.

Moreover, we show in Figure 13 the results of the L_2 reconstruction error in the well-specified setting, when we vary the number of sampling points of the feature paths between 10 and 10^3. We see that SigLasso always outperforms GRU and Neural CDE but that the difference of performance is more important when there are only a few sampling points. In this regime SigLasso is moreover more stable.

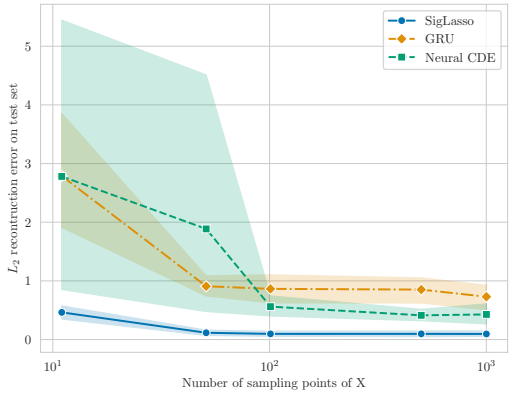

Figure 13: L_2 reconstruction error of SigLasso, GRU and Neural CDE in the well-specified setting, for varying number of feature samples.