SIAM J. MATH. DATA SCI.
Vol. 3, No. 3, pp. 873–899

The Signature Kernel Is the Solution of a Goursat PDE*

Cristopher Salvi†, Thomas Cass‡, James Foster†, Terry Lyons†, and Weixin Yang†

Abstract. Recently, there has been an increased interest in the development of kernel methods for learning with sequential data. The signature kernel is a learning tool with the potential to handle irregularly sampled, multivariate time series. In [F. J. Király and H. Oberhauser, *J. Mach. Learn. Res.*, 20 (2019), 31] the authors introduced a kernel trick for the truncated version of this kernel avoiding the exponential complexity that would have been involved in a direct computation. Here we show that for continuously differentiable paths, the signature kernel solves a hyperbolic PDE and recognize the connection with a well-known class of differential equations known in the literature as Goursat problems. This Goursat PDE only depends on the increments of the input sequences, does not require the explicit computation of signatures, and can be solved efficiently using state-of-the-art hyperbolic PDE numerical solvers, giving a kernel trick for the untruncated signature kernel, with the same raw complexity as the method from Király and Oberhauser, but with the advantage that the PDE numerical scheme is well suited for GPU parallelization, which effectively reduces the complexity by a full order of magnitude in the length of the input sequences. In addition, we extend the previous analysis to the space of geometric rough paths and establish, using classical results from rough path theory, that the rough version of the signature kernel solves a rough integral equation analogous to the aforementioned Goursat problem. Finally, we empirically demonstrate the effectiveness of this PDE kernel as a machine learning tool in various data science applications dealing with sequential data. We make the library `sigkernel` publicly available at https://github.com/crispitagorico/sigkernel.

Key words. kernel, path signature, Goursat PDE, sequential data, geometric rough path, rough integration

AMS subject classifications. 60L10, 60L20

DOI. 10.1137/20M1366794

1. Introduction.

Nowadays, sequential data is being produced and stored at an unprecedented rate. Examples include daily fluctuations of asset prices in the stock market, medical and biological records, readings from mobile apps, and weather measurements. An efficient learning algorithm must be able to handle data streams that are often irregularly sampled and/or partially observed and simultaneously scaling well with a high number of channels.

An important obstacle that most machine learning models have to face is the potential symmetry present in the data. In computer vision, for example, a good model should be able to recognize an image even if the latter is rotated at a certain angle. The 3D rotation group, often

*Received by the editors September 14, 2020; accepted for publication (in revised form) June 3, 2021; published electronically September 9, 2021.

https://doi.org/10.1137/20M1366794

Funding: This work was supported by DataSig under the EPSRC grant EP/S026347/1 and by the Alan Turing Institute under the EPSRC grant EP/N510129/1. The first author was supported by the EPSRC grant EP/R513295/1.

†Mathematical Institute, University of Oxford, Oxford OX2 6GG, UK, and The Alan Turing Institute, London NW1 2DB, UK (cristopher.salvi@maths.ox.ac.uk, james.foster@maths.ox.ac.uk, terry.lyons@maths.ox.ac.uk, weixin.yang@maths.ox.ac.uk).

‡Imperial College London, London SW7 2BX, UK, and The Alan Turing Institute, London NW1 2DB, UK (thomas.cass@imperial.ac.uk).

denoted by $SO(3)$, is low-dimensional (3); therefore it is relatively easy to add components to a model that build a rotation invariance. However, when dealing with sequential data one is confronted with a much bigger (infinite-dimensional) group of symmetries given by all *reparametrizations* of a path[1] (i.e., continuous and increasing surjections from the time domain of the path to itself). For example, consider the reparametrization $\phi : [0,1] \to [0,1]$ given by $\phi(t) = t^2$ and the path $\gamma : [0,1] \to \mathbb{R}^2$ defined by $\gamma_t = (\gamma_t^x, \gamma_t^y)$, where $\gamma_t^x = \cos(10t)$ and $\gamma_t^y = \sin(3t)$. As is clearly depicted in Figure 1, both channels (γ^x, γ^y) of γ are individually affected by the reparametrization ϕ, but the shape of the curve γ is left unchanged.

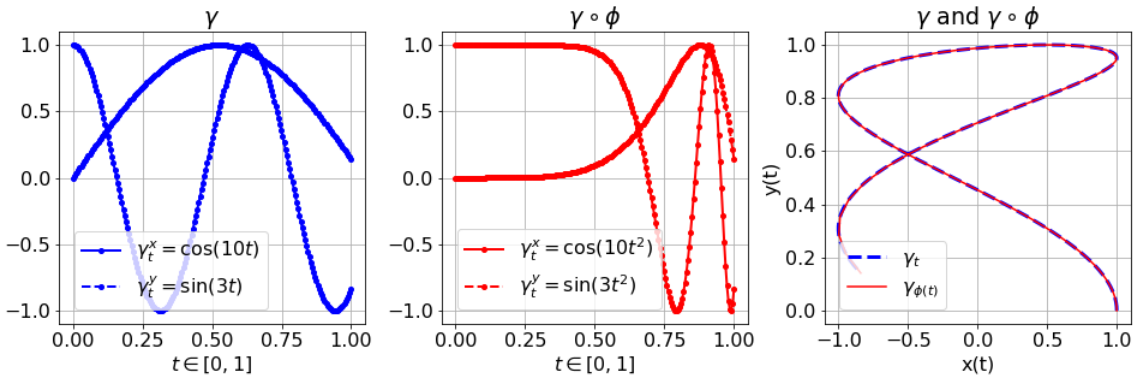

Figure 1. *On the left are the individual channels (γ^x, γ^y) of a 2D paths γ. In the middle are the channels reparametrized under $\phi : t \mapsto t^2$. On the right are the path γ and its reparametrized version $\gamma \circ \phi$. The two curves overlap, meaning that the reparametrization ϕ represents irrelevant information if one is interested in understanding the shape of γ.*

Definition 1.1. *Let V be a Banach space. The spaces of formal polynomials and formal power series over V are defined, respectively, as*

$$T(V) = \bigoplus_{k=0}^{\infty} V^{\otimes k} \quad \text{and} \quad T((V)) = \prod_{k=0}^{\infty} V^{\otimes k}, \tag{1.1}$$

where $\otimes$ denotes the (classical) tensor product of vector spaces. Both $T(V)$ and $T((V))$ can be endowed with the operations of addition $+$ and multiplication $\otimes$ defined for any two elements $A = (a_0, a_1, \dots)$ and $B = (b_0, b_1, \dots)$, respectively, as

$$A + B = (a_0 + b_0, a_1 + b_1, \dots), \tag{1.2}$$

$$A \otimes B = (c_0, c_1, c_2, \dots), \quad \text{where} \quad V^{\otimes k} \ni c_k = \sum_{i=0}^{k} a_i \otimes b_{k-i} \quad \text{for all } k \geq 0. \tag{1.3}$$

When endowed with these two operations and the natural action of $\mathbb{R}$ by $\lambda A = (\lambda a_0, \lambda a_1, \dots)$, $T((V))$ becomes a real, noncommutative unital algebra with unit $\mathbf{1} = (1, 0, 0, \dots)$ called the

[1] Or its time-augmented version.

tensor algebra. The truncated tensor algebra over V of order $N \in \mathbb{N}$ is defined as the quotient $T^N(V) = T((V))/T_N$ by the ideal

$$T_N = \{A = (a_0, a_1, \dots) \in T((V)) : a_0 = \cdots = a_N = 0\}. \tag{1.4}$$

Definition 1.2. *Let $I \subset \mathbb{R}$ be a compact interval, let V be a Banach space, and let $x : I \to V$ be a continuous path of finite p-variation (Definition SM1.4 of the supplemental material) with $p < 2$. For any $s, t \in I$ such that $s \le t$, the signature $S(x)_{[s,t]} \in T((V))$ of the path x over the subinterval $[s, t]$ is defined as the following infinite collection of iterated integrals:*

$$S(x)_{[s,t]} = \left(1, \int_{s<u_1<t} dx_{u_1}, \dots, \int\cdots\int_{s<u_1<\cdots<u_k<t} dx_{u_1} \otimes \cdots \otimes dx_{u_k}, \dots\right). \tag{1.5}$$

The *signature* (Definition 1.2) of a path x is invariant under reparametrization (i.e., $S(x) = S(x \circ \phi)$), and therefore it acts on x as a filter that systematically removes this troublesome, infinite-dimensional group of symmetries. Furthermore, it turns out that linear functionals acting on the range of the signature form an algebra (with pointwise multiplication) and separate points [32, Chapter 2]. Hence, by the *Stone–Weierstrass theorem*, for any compact set C of continuous paths of bounded variation, the set of linear functionals on signatures of paths from C is dense in the set of continuous real-valued functions on C. These two properties make the signature an ideal *feature map* for data streams [31].

For any path x of finite p-variation ($p > 1$) the terms in the signature *decay factorially* according to the following uniform estimate [31, Lemma 5.1]:

$$\left\| \int\cdots\int_{s<u_1<\cdots<u_k<t} dx_{u_1} \otimes \cdots \otimes dx_{u_k} \right\|_{V^{\otimes k}} \le \frac{\|x\|^k_{p,[s,t]}}{k!}, \tag{1.6}$$

where $\|\cdot\|_{V^{\otimes k}}$ denotes any norm on $V^{\otimes k}$ and $\|x\|_{p,[s,t]}$ denotes the p-variation of the path x restricted to the interval $[s, t]$. Therefore, the collection of iterated integrals in the signature is *graded*. This grading allows one to *truncate* the signature at a finite level $N \in \mathbb{N}$ and consider only a finite collection of integrals as features extracted from x:

$$S^N(x)_{[s,t]} = \left(1, \int_{s<u_1<t} dx_{u_1}, \dots, \int\cdots\int_{s<u_1<\cdots<u_N<t} dx_{u_1} \otimes \cdots \otimes dx_{u_N}\right) \in T^N(V). \tag{1.7}$$

However, it is clear that the (truncated) signature has an exponential growth in the number of features, limiting its successful direct usage to machine learning applications where the ambient space of the data streams is relatively low-dimensional [2, 37, 22, 51, 36, 11, 33].

Kernel methods [20] have shown to be efficient learning techniques in situations where inputs are non-Euclidean and high-dimensional (not necessarily sequential) and the number of training instances is limited [42], so that deep learning methods cannot be easily deployed. Most kernels that are used in practice can be computed efficiently without referring back to the corresponding feature map, a mechanism known as a *kernel trick*. When the data is

sequential, the design of appropriate kernel functions is a notably challenging task [15]. In [26] the authors introduce the *truncated signature kernel* as the inner product of two truncated signatures and propose an efficient algorithm to compute this kernel starting from any "static" kernel on the ambient space of the input paths.

One of our goals will be to extend the results in [26] and consider the *(untruncated) signature kernel* as an inner product of two (untruncated) signatures. For this, leveraging a fundamental property of the signature (Theorem 2.3), we prove in section 2 that if the two input paths are continuously differentiable, then the signature kernel is the solution of a linear, second order, hyperbolic *partial differential equation* (PDE). In section 3 we recognize the connection between the signature kernel PDE and a class of differential equations known in the literature as *Goursat problems* [19]. This PDE represents effectively a kernel trick for the signature kernel and can be efficiently solved by numerically leveraging any state-of-the-art hyperbolic PDE solvers; we provide ourselves a competitive finite difference explicit scheme and demonstrate the improvement in computational performance over existing approximation methods. In section 4 we extend the previous analysis to the much broader class of *geometric rough paths* and show that in this case the signature kernel satisfies an integral equation analogous to the aforementioned Goursat PDE. Finally, in section 5 we empirically demonstrate the effectiveness of the signature kernel on various data science applications dealing with sequential data.

We release the Python library `sigkernel` implementing our signature PDE kernel and various other functionalities deriving from it. All the experiments presented in this paper are reproducible following the instructions in https://github.com/crispitagorico/sigkernel.

Remark 1.3. A concise summary of rough path theory, covering the material necessary to follow the proofs in section 4, is presented in section SM1 of the supplementary material. We note that an efficient algorithm for computing the truncated signature kernel was derived in [26] and then used in [47] in the context of Gaussian processes indexed on time series. Finally, we note that the article [9] first treated the truncated signature kernel in the case of branched rough paths. Integration of two-parameter rough integrals is also discussed in [10].

2. The signature kernel is the solution of a hyperbolic PDE.

In this section we present our main result, notably that the signature kernel evaluated at two continuously differentiable paths is the solution of a hyperbolic PDE. Throughout this section, we will denote by $C^1(I,V)$ the space of continuously differentiable paths defined over an interval $I=[u,u']$ and with values on a Banach space V. We will also use the lighter notation $S(x)_t$ to denote the signature of a path x over the interval $[u,t]$ for any $t\in I$.

Definition 2.1. *Let V be a d-dimensional Banach space with canonical basis $\{e_1,\dots,e_d\}$. It is easy to verify that for any $k\geq 1$ the elements*

$$\{e_{i_1}\otimes\cdots\otimes e_{i_k} : (i_1,\dots,i_k)\in\{1,\dots,d\}^k\} \tag{2.1}$$

form a basis of $V^{\otimes k}$. Consider the inner product on $V^{\otimes k}$ defined on basis elements as

$$\langle e_{i_1}\otimes\cdots\otimes e_{i_k}, e_{j_1}\otimes\cdots\otimes e_{j_k}\rangle_{V^{\otimes k}} = \langle e_{i_1},e_{j_1}\rangle_V\dots\langle e_{i_k},e_{j_k}\rangle_V. \tag{2.2}$$

The inner product $\langle\cdot,\cdot\rangle_{V^{\otimes k}}$ can be extended by linearity to an inner product on $T(V)$ defined

for any $A = (a_0, a_1, \dots), B = (b_0, b_1, \dots)$ *in* $T(V)$ *as*

$$\langle A, B \rangle = \sum_{k=0}^{\infty} \langle a_k, b_k \rangle_{V^{\otimes k}}. \tag{2.3}$$

Remark 2.2. It is easy to verify that the space $T((V))$ has the following algebraic property, which we will refer to as the *coproduct property*. Let $m, n \in \mathbb{N}$ be two positive integers and consider any two elements $A, B \in T((V))$ and any two basis elements $e_{i_1}, e_{i_2} \in V$ seen as elements of $T((V))$, i.e., as $(0, e_{i_1}, 0, \dots)$ and $(0, e_{i_2}, 0, \dots)$, respectively. Then the following identity holds:

$$\langle A \otimes e_{i_1}, B \otimes e_{i_2} \rangle = \langle A, B \rangle \langle e_{i_1}, e_{i_2} \rangle_V. \tag{2.4}$$

As anticipated in the introduction, the signature has a fundamental characterization in terms of *controlled differential equations* (CDEs) (see subsection SM1.1 for a brief account of CDEs). In effect, the signature solves the universal differential equation stated in the next theorem, and therefore it can be equivalently defined as the *noncommutative exponential*.

Theorem 2.3 (see [32, Lemma 2.10]). *Let* $x : I \to V$ *be a continuous path of finite p-variation for* $p < 2$ *and* $A = (a_0, a_1, \dots) \in T(V)$. *Consider the vector field* $f : T(V) \to L(V, T(V))$,[2]

$$f(A)(v) = A \otimes v = (0, a_0 \otimes v, a_1 \otimes v, \dots). \tag{2.5}$$

Then the unique solution to the CDE

$$dS_t = f(S_t) dx_t, \quad S_0 = (1, 0, 0, \dots) \tag{2.6}$$

is the signature $S(x)_t$ *of the path* x. *Equation* (2.6) *can be formally rewritten as*

$$dS(x)_t = S(x)_t \otimes dx_t, \quad S(x)_0 = (1, 0, 0, \dots). \tag{2.7}$$

2.1. The signature kernel PDE.

Definition 2.4. *Let* $I = [u, u']$ *and* $J = [v, v']$ *be two compact intervals, and let* $x \in C^1(I, V)$ *and* $y \in C^1(J, V)$. *The signature kernel* $k_{x,y} : I \times J \to \mathbb{R}$ *is defined as*

$$k_{x,y}(s, t) = \langle S(x)_s, S(y)_t \rangle. \tag{2.8}$$

The following is the main result of this section; it unveils a simple relation between the signature kernel and a class of hyperbolic PDEs.

Theorem 2.5. *Let* $I = [u, u']$ *and* $J = [v, v']$ *be two compact intervals, and let* $x \in C^1(I, V)$ *and* $y \in C^1(J, V)$. *The signature kernel* $k_{x,y}$ *is a solution of the following linear, second order, hyperbolic PDE:*

$$\frac{\partial^2 k_{x,y}}{\partial s \partial t} = \langle \dot{x}_s, \dot{y}_t \rangle_V k_{x,y}, \quad k_{x,y}(u, \cdot) = k_{x,y}(\cdot, v) = 1, \tag{2.9}$$

where $\dot{x}_s = \frac{dx_p}{dp}\big|_{p=s}$, $\dot{y}_t = \frac{dx_q}{dq}\big|_{q=t}$ *are the derivatives of* x *and* y *at time* s *and* t, *respectively.*

[2] $L(V, T(V))$ denotes the space of bounded linear maps from V to $T(V)$.

Proof. Clearly, for any $t \in J$ one has

$$\begin{aligned} k_{x,y}(u,t) &= \big\langle S(x)_{[u,u]}, S(y)_{[v,t]} \big\rangle \\ &= \big\langle (1,0,\dots), S(y)_{[v,t]} \big\rangle \\ &= 1, \end{aligned}$$

and similarly $k_{x,y}(s,v) = 1$ for any $s \in I$. Recall that the signature of a path $x : I \to V$ satisfies (2.7), which is equivalent to the integral equation

$$S(x)_s = \mathbf{1} + \int_{p=u}^{s} S(x)_p \otimes dx_p,$$

where $\mathbf{1} = (1,0,0,\dots)$. Similarly for $S(y)_t$. Hence, we can compute

$$\begin{aligned} k_{x,y}(s,t) &= \langle S(x)_s, S(y)_t \rangle \\ &= \Big\langle \mathbf{1} + \int_{p=u}^{s} S(x)_p \otimes dx_p,\ \mathbf{1} + \int_{q=v}^{t} S(y)_q \otimes dy_q \Big\rangle && \text{(Theorem 2.3)} \\ &= 1 + \Big\langle \int_{p=u}^{s} S(x)_p \otimes \dot{x}_p dp,\ \int_{q=v}^{t} S(y)_q \otimes \dot{y}_q dq \Big\rangle && \text{(differentiability)} \\ &= 1 + \int_{p=u}^{s}\int_{q=v}^{t} \langle S(x)_p \otimes \dot{x}_p,\ S(y)_q \otimes \dot{y}_q \rangle dpdq && \text{(linearity)} \qquad (2.10) \\ &= 1 + \int_{p=u}^{s}\int_{q=v}^{t} \langle S(x)_p, S(y)_q \rangle \langle \dot{x}_p, \dot{y}_q \rangle_V dpdq && \text{(coproduct property (2.4))} \\ &= 1 + \int_{p=u}^{s}\int_{q=v}^{t} k_{x,y}(p,q) \langle \dot{x}_p, \dot{y}_q \rangle_V dpdq && \text{(definition of } k_{x,y}\text{)}. \end{aligned}$$

Note that the inner product and the double integral can be interchanged in (2.10) because of the factorial decay (1.6) of the terms in the signature. By the *fundamental theorem of calculus* we can differentiate first with respect to s,

$$\frac{\partial k_{x,y}(s,t)}{\partial s} = \int_{q=v}^{t} k_{x,y}(s,q) \langle \dot{x}_s, \dot{y}_q \rangle_V dq,$$

and then with respect to t to obtain the PDE (2.9),

$$\frac{\partial^2 k_{x,y}(s,t)}{\partial s \partial t} = \langle \dot{x}_s, \dot{y}_t \rangle_V k_{x,y}(s,t). \qquad \blacksquare$$

Remark 2.6. In Theorem 2.5 we have assumed the two input paths x, y to be of class C^1. However, one can lower this regularity assumption and consider two continuous paths x, y of bounded variation and obtain the integral equation

$$k_{x,y}(s,t) = 1 + \int_{p=u}^{s}\int_{q=v}^{t} \langle S(x)_p, S(y)_q \rangle \langle dx_p, dx_q \rangle_V, \qquad (2.11)$$

where $\langle dx_p, dx_q \rangle_V$ is a quantity we are going to define in section 4, when we will consider the broader class of *geometric rough paths*. We note that one can make sense of the PDE (2.9) in Theorem 2.5 for piecewise C^1 paths.

Sequential information often arrives in the form of complex data streams taking their values in nontrivial *ambient spaces*. A good learning strategy would be to first *lift* the underlying ambient space to a (possibly infinite-dimensional) *feature space* by means of a *feature map* on static data (RBF, Matern, etc.), and then consider the signature kernel of the lifted paths as the final learning tool [26]. A question that naturally arises is whether one can compute the signature PDE kernel of the lifted paths from the static kernel associated to this feature map. Király and Oberhauser [26] propose an algorithm to perform this procedure. Next, we provide an explanation of this procedure in the language of Banach spaces and PDEs.

2.2. The signature PDE kernel from static kernels on the ambient space. A kernel can be identified with a pair of embeddings of a set $\mathcal{X}$ into a Banach space E and its topological dual E^*; we denote this pair of maps by $\phi : \mathcal{X} \to E$ and $\psi : \mathcal{X} \to E^*$. A kernel induces a function $\kappa : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ through the natural pairing between a Banach space and its dual:

$$\kappa(a, b) := (\phi(a), \psi(b))_E \quad \text{for all } a, b \in \mathcal{X}. \tag{2.12}$$

Commonly, E is assumed to be a Hilbert space, in which case ψ can be taken to be the composition $e \circ \phi$, where $e : E \to E^*$ is the canonical isomorphism coming from the *Riesz representation theorem*, yielding $\kappa(a, b) = \langle \phi(a), \phi(b) \rangle_E$. It is unnecessary, however, for the general picture for E to be a Hilbert space. In the general framework, a given pair of paths $x : I \to \mathcal{X}$ and $y : J \to \mathcal{X}$, with $I = [u, u'], J = [v, v']$, can be lifted to paths on E and E^*, respectively, as follows:

$$X_s = \phi(x_s), \quad Y_t = \psi(y_t) \quad \text{for all } s \in I, t \in J \tag{2.13}$$

If we assume that X and Y are continuous and have bounded variation, then their signatures are well defined and belong to $T(E)$, which is again a Banach space with $T^N(E)^* \cong T^N(E^*)$ for any $N \geq 1$ [32]. Hence, starting with a kernel κ on $\mathcal{X}$, the signature kernel is well defined:

$$k_{X,Y}(s, t) = (S(\phi \circ x)_s, S(\psi \circ y)_t)_{T(E)} = (S(X)_s, S(Y)_t)_{T(E)}. \tag{2.14}$$

If furthermore we assume that the lifted paths X, Y are of class C^1, Theorem 2.5 applies, yielding the PDE

$$\frac{\partial^2 k_{X,Y}}{\partial s \partial t} = (\dot{X}_s, \dot{Y}_t)_E \, k_{X,Y}, \quad k_{X,Y}(u, \cdot) = k_{X,Y}(\cdot, v) = 1. \tag{2.15}$$

With a first order finite difference approximation for the derivatives $\dot{X}_s, \dot{Y}_t$, the PDE (2.15) can be entirely expressed in terms of the static kernel κ and underlying paths x, y as follows:

$$\begin{aligned} \frac{\partial^2 k_{X,Y}}{\partial s \partial t} &= ((X_s, Y_t)_E - (X_{s-1}, Y_t)_E - (X_s, Y_{t-1})_E + (X_{s-1}, Y_{t-1})_E) k_{X,Y} \\ &= (\kappa(x_s, y_t) - \kappa(x_{s-1}, x_t) - \kappa(x_s, y_{t-1}) + \kappa(x_{s-1}, y_{t-1})) k_{X,Y}. \end{aligned} \tag{2.16}$$

Remark 2.7. We note that it is possible to establish our Goursat PDE (2.9) from the results [25, Proposition 4.7, p. 16], [25, Theorem 4, Appendix A], but this requires additional arguments, as we shall explain next. Using their notation, given two differentiable paths σ, τ, the truncated (at level M) signature kernel $k^{\oplus}_{\leq M}(\sigma, \tau)$ satisfies the equation

$$k^{\oplus}_{\leq M}(\sigma,\tau)_{u,v} = 1 + \iint_{(s_1,t_1)\in(0,u)\times(0,v)} \left(1 + \cdots + \iint_{(s_M,t_M)\in(0,s_{M-1})\times(0,t_{M-1})} d\kappa_{\sigma,\tau}(s_M,t_M)\right) d\kappa_{\sigma,\tau}(s_1,t_1),$$

where $d\kappa_{\sigma,\tau}(s,t) = k(\dot{x}_s, \dot{y}_t)dsdt$ and where k is a kernel on the ambient space of the paths. The first step towards a PDE is to realize that the first integrand in the last equation is itself the truncated signature kernel, truncated at level $M-1$, yielding the expression

$$k^{\oplus}_{\leq M}(\sigma,\tau)_{u,v} = 1 + \iint_{(s_1,t_1)\in(0,u)\times(0,v)} k^{\oplus}_{\leq M-1}(\sigma,\tau)_{s_1,t_1} d\kappa_{\sigma,\tau}(s_1,t_1). \tag{2.17}$$

The untruncated signature kernel is obtained by taking the limit in (2.17) when $M \to \infty$. The factorial decay in the terms of the signature yields uniform convergence of this limiting process. Two uniformly convergent sequences of functions that are equal for all finite levels M are also equal in the limit, which implies

$$k^{\oplus}_{\leq \infty}(\sigma,\tau)_{u,v} = 1 + \iint_{(s_1,t_1)\in(0,u)\times(0,v)} k^{\oplus}_{\leq \infty}(\sigma,\tau)_{s_1,t_1} d\kappa_{\sigma,\tau}(s_1,t_1). \tag{2.18}$$

Finally one substitutes $d\kappa_{\sigma,\tau}(s,t) = k(\dot{x}_s, \dot{y}_t)dsdt$. At that point (and similarly to our argument in Theorem 2.5) one can differentiate both sides of (2.18) to get the Goursat PDE.

In the next section we recognize the link between (2.9) and a class of differential equations known in the literature as Goursat problems and propose a competitive numerical solver for our specific PDE.

3. A Goursat problem.

Equation (2.9) is an instance of a Goursat problem, which is a class of hyperbolic PDEs introduced in [19]. The PDE (2.9) is defined on the bounded domain

$$\mathcal{D} := \{(s,t) \mid u \leq s \leq u', v \leq t \leq v'\} \subset I \times J, \tag{3.1}$$

and its existence and uniqueness (for paths of class C^1) are guaranteed by setting the functions $C_1 = C_2 = C_4 = 0$ and $C_3(s,t) = \langle \dot{x}_s, \dot{y}_t \rangle_V$ in the following result.

Theorem 3.1 (see [27, Theorems 2 and 4]). *Let $\sigma : I \to \mathbb{R}$ and $\tau : J \to \mathbb{R}$ be two absolutely continuous functions whose first derivatives are square integrable and such that $\sigma(u) = \tau(v)$. Let $C_1, C_2, C_3 : \mathcal{D} \to \mathbb{R}$ be bounded and measurable over $\mathcal{D}$, and let $C_4 : \mathcal{D} \to \mathbb{R}$ be square integrable. Then there exists a unique function $z : \mathcal{D} \to \mathbb{R}$ such that $z(s,v) = \sigma(s), z(u,t) = \tau(t)$ and (almost everywhere on $\mathcal{D}$)*

$$\frac{\partial^2 z}{\partial s \partial t} = C_1(s,t)\frac{\partial z}{\partial s} + C_2(s,t)\frac{\partial z}{\partial t} + C_3(s,t)z + C_4(s,t). \tag{3.2}$$

If in addition $C_i \in C^{p-1}(\mathcal{D})$ $(i = 1,2,3,4)$ and σ and τ are C^p, then the unique solution $z : \mathcal{D} \to \mathbb{R}$ of the Goursat problem is of class C^p.

In the case of the signature PDE kernel, if the two input paths x, y are of class C^p, then their derivatives will be of class C^{p-1}, and therefore Theorem 3.1 implies that the solution $k_{x,y}$ of the PDE (2.9) will be of class C^p.

3.1. Finite difference approximation. In this section, we propose a numerical method based on an explicit finite difference scheme to approximate the solution of the Goursat PDE (2.9). To simplify the notation, we consider the case where $V = \mathbb{R}^d$. If x and y are piecewise linear, then the PDE (2.9) becomes

$$\frac{\partial^2 k_{x,y}}{\partial s \partial t} = C_3 k_{x,y} \tag{3.3}$$

on each domain $\mathcal{D}_{ij} = \{(s,t) \mid u_i \le s \le u_{i+1}, v_j \le t \le v_{j+1}\}$ where $C_3 = \langle \dot{x}_s, \dot{y}_t \rangle_V$ is constant. In integral form, the PDE (3.3) can be written as

$$k_{x,y}(s,t) = k_{x,y}(s,v) + k_{x,y}(u,t) - k_{x,y}(u,v) + C_3 \int_u^s \int_v^t k_{x,y}(r,w)\, dr\, dw \tag{3.4}$$

for $(s,t), (u,v) \in \mathcal{D}_{ij}$ with $u \le s$ and $v \le t$. By approximating the double integral in (3.4), we can derive the following numerical explicit scheme:

$$\begin{aligned} k_{x,y}(s,t) \approx\ & k_{x,y}(s,v) + k_{x,y}(u,t) - k_{x,y}(u,v) \\ & + \frac{1}{2} C_3 \big(k_{x,y}(s,v) + k_{x,y}(u,t)\big)(u-s)(t-v). \end{aligned} \tag{3.5}$$

Remark 3.2. An implicit scheme can be obtained by estimating (3.4) with all four values of $k_{x,y}$ as follows:

$$\begin{aligned} k_{x,y}(s,t) \approx\ & k_{x,y}(s,v) + k_{x,y}(u,t) - k_{x,y}(u,v) \\ & + \frac{1}{4} C_3 \big(k_{x,y}(u,v) + k_{x,y}(s,v) + k_{x,y}(u,t) + k_{x,y}(s,t)\big)(u-s)(t-v). \end{aligned} \tag{3.6}$$

As one might expect, more sophisticated approximations can be derived by applying higher order quadrature methods to the double integral in (3.4) (see [16, 50] for specific examples).

Let $\mathcal{D}_I = \{u = u_0 < u_1 < \cdots < u_{m-1} < u_m = u'\}$ be a partition of the interval I and $\mathcal{D}_J = \{v = v_0 < v_1 < \cdots < v_{n-1} < v_n = v'\}$ a partition of the interval J. Using the above, we can define *finite difference schemes* on the grid $P_0 := \mathcal{D}_I \times \mathcal{D}_J$ (and its dyadic refinements).

Definition 3.3. *For $\lambda \in \{0, 1, 2, \dots\}$, we define the grid P_λ as the dyadic refinement of P_0 such that $P_\lambda \cap ([u_i, u_{i+1}] \times [v_i, v_{i+1}]) = \{u_i + k\, 2^{-\lambda}(u_{i+1} - u_i), v_j + l\, 2^{-\lambda}(v_{j+1} - v_j)\}_{0 \le k,l \le 2^\lambda}$.*

On the grid $P_\lambda = \{(s_i, t_j)\}_{0 \le i \le 2^\lambda n, 0 \le j \le 2^\lambda m}$, we define the following explicit finite difference scheme for the PDE (2.9):

$$\begin{aligned} \hat{k}(s_{i+1}, t_{j+1}) =\ & \hat{k}(s_{i+1}, t_j) + \hat{k}(s_i, t_{j+1}) - \hat{k}(s_i, t_j) \\ & + \frac{1}{2} \langle x_{s_{i+1}} - x_{s_i}, y_{t_{j+1}} - y_{t_j} \rangle \big(\hat{k}(s_{i+1}, t_j) + \hat{k}(s_i, t_{j+1})\big), \\ \hat{k}(s_0, \cdot\,) = \hat{k}(\,\cdot\,, t_0) =\ & 1. \end{aligned} \tag{3.7}$$

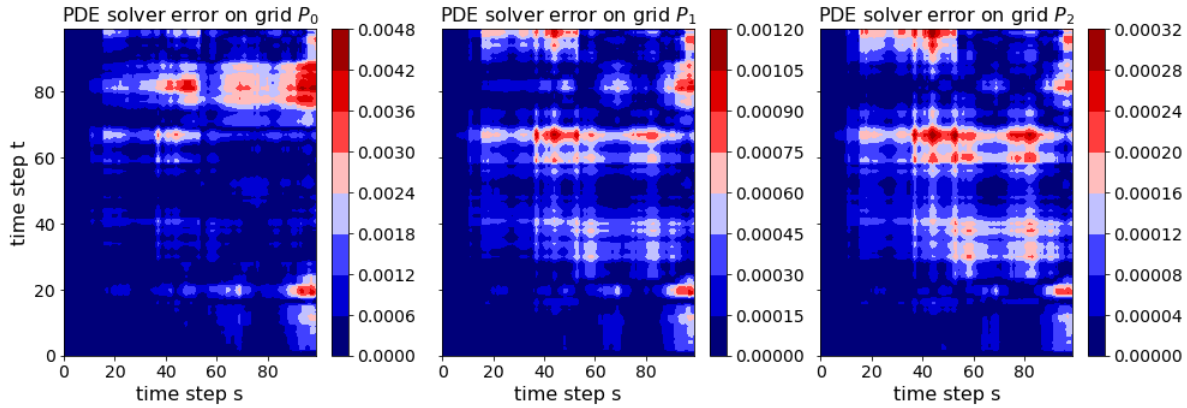

Figure 2. *Example of error distribution of $k_{x,y}(s,t)$ on the grids P_0, P_1, P_2. The discretization is roughly four times more accurate on P_1 than on P_0 (as expected by Theorem 3.5).*

Remark 3.4. If x and y are piecewise linear paths with respect to the coarsest grid P_0, then $\langle x_{s_{i+1}} - x_{s_i}, y_{t_{j+1}} - y_{t_j} \rangle = \frac{1}{2^{2\lambda}} \langle x_{u_{p+1}} - x_{u_p}, y_{v_{q+1}} - y_{v_q} \rangle$ for some $0 \le p < n$ and $0 \le q < m$.

The explicit finite difference scheme (3.7) has a time complexity of $O\big(d^2\, 2^{2\lambda}\, mn\big)$ on the grid P_λ, where d is the dimension of the input streams x, y and m, n denote their respective lengths. Theorem 3.5 (which is proved in Appendix A) ensures that by refining the discretization of the grid used to approximate the PDE, we get convergence to the true value. In practice we found that, provided the input paths are rescaled so that their maximum value across all times and all dimensions is not too large (≈ 1), coarse partitioning choices such as P_0 or P_1 are sufficient to obtain a highly accurate approximation, as shown in Figure 2.

Theorem 3.5 (global error estimate; see Appendix A). *Let $\widetilde{k}$ be a numerical solution obtained by applying one of the proposed finite difference schemes (3.7) to the Goursat problem (2.9) on P_λ, where x and y are piecewise linear with respect to the grids $\mathcal{D}_I$ and $\mathcal{D}_J$. In particular, we are assuming there exists a constant M, which is independent of λ, such that*

$$\sup_{\mathcal{D}} |\langle \dot{x}_s, \dot{y}_t \rangle| < M. \tag{3.8}$$

Then there exists a constant $K > 0$ depending on M and $k_{x,y}$, but independent of λ, such that

$$\sup_{\mathcal{D}} \left| k_{x,y}(s,t) - \widetilde{k}(s,t) \right| \le \frac{K}{2^{2\lambda}} \quad \text{for all } \lambda \ge 0 \tag{3.9}$$

3.1.1. GPU implementation of the Goursat PDE. As mentioned earlier, the time complexity for one signature PDE kernel evaluation is $\mathcal{O}(d\ell^2)$ on P_0, where d is the number of channels of the input time series and ℓ is their (maximum) length. Therefore, the complexity is quadratic in the length of the time series, which makes kernel evaluations computationally expensive for long time series. This also holds for the algorithm proposed in [26]. However, it is possible to parallelize the PDE solver by observing that instead of solving the PDE in row or column order, we can update the antidiagonals of the solution grid: each cell on an

antidiagonal can be updated in parallel as there is no data dependency between them. This breaks the quadratic complexity, which becomes linear in the length ℓ, provided the number of threads in the GPU exceeds the size of the discretization, as shown in Figure 3. This parallelization is possible thanks to the "PDE structure" of the problem, representing a considerable computational gain of our algorithm compared to the one proposed by [26]. We also note that the linear dependency on the number of channels d of the input time series allows for the evaluation of the signature PDE kernel on time series with thousands of channels. Our library `sigkernel` offers the ability to evaluate kernels on a CPU using an optimized `cython` implementation as well as on `CUDA` if GPUs are available to the user.

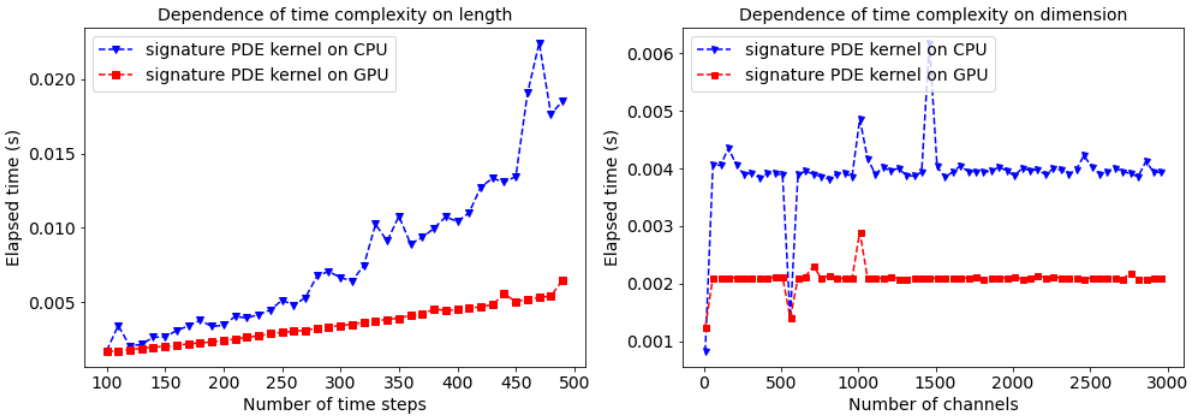

Figure 3. *Comparison of the elapsed time (s) to reach an accuracy of* 10^{-3} *from a target value obtained by solving the signature kernel PDE on a fine discretization grid* (P_5). *We simulate* $N = 5$ *(piecewise linear interpolation of) Brownian paths at each run. On the left is the dependency on the length of two paths of dimension* $d = 2$. *Note the complexity reduction from quadratic on* `CPU` *to linear on* `GPU` *(P100). On the right is the dependency on the dimension of two paths of length* $\ell = 10$.

In section 5 we will present various applications of the signature kernel to time series classification and regression problems. But first we continue our theoretical analysis and drop the smoothness assumption on the input paths x, y and extend the definition of signature kernel to far less regular classes of paths, namely, geometric rough paths. The need to investigate the rough version of the signature kernel can be motivated also from several practical viewpoints. For example, this kernel can be used to derive an (unbiased) estimator for the *maximum mean discrepancy* (MMD) distance between distributions on path-space [9, sections 7 and 8]. The MMD distance itself is useful to train models such as *neural SDEs* [48, 23, 29, 14], that is, to fit neural SDEs to time series data. Since SDE solutions are geometric p-rough paths, Theorem 4.11 provides a candidate for the limiting kernel as the mesh size of the SDE discretization tends to zero. In particular, it guarantees that the signature kernel doesn't "blow up" in the limit. Another area where the rough signature kernel could be relevant is quantitative finance, where *rough volatility models* [18, 6] try to calibrate differential equations driven by *fractional Brownian motion*, which is a rough path if the *Hurst exponent* $h \leq 2$.

4. The signature kernel for geometric rough paths. Here we extend the notion of signature kernel developed in section 2 to the broader class of geometric rough paths. To follow

the material presented in this section we assume that the reader has some level of familiarity with basic concepts of rough path theory. We provide a brief summary of this theory in section SM1 of the supplementary material. We begin by clarifying what we mean by signature of a geometric p-rough path.

4.1. The signature of a geometric rough path.

Definition 4.1. *The signature $S(X)$ of a geometric p-rough path $X \in G\Omega_p(V)$ (Definition SM1.17) controlled by a control (Definition SM1.11) ω is its unique extension to a multiplicative functional (Definition SM1.10) on $T(V)$ as given by Extension Theorem SM1.14.*

From now on, we will denote by $G\Omega_p(V)$ the space of geometric p-rough paths over V. Because all the sums in $T(V)$ are finite, $(T(V), \langle\cdot,\cdot\rangle)$ is an inner product space. Hence, denoting by $\overline{T(V)}$ the completion of $T(V)$, $(\overline{T(V)}, \langle\cdot,\cdot\rangle)$ is a Hilbert space. Let $\|\cdot\|$ be the norm on $\overline{T(V)}$ induced by the inner product $\langle\cdot,\cdot\rangle$, i.e., defined for any $A = (a_0, a_1, \dots) \in \overline{T(V)}$ as $\|A\| = \sqrt{\sum_{k\geq 0} \|a_k\|^2_{V^{\otimes k}}}$, where $\|\cdot\|_{V^{\otimes k}}$ is the norm on $V^{\otimes k}$ induced by $\langle\cdot,\cdot\rangle_{V^{\otimes k}}$ for any $k \geq 0$. In summary, we have the following chain of inclusions:

$$T(V) \hookrightarrow \overline{T(V)} \hookrightarrow T((V)). \tag{4.1}$$

Note that $\overline{T(V)} = \{x \in T((V)) : ||x|| < \infty\}$.

Lemma 4.2. *Let X be a geometric p-rough path $X \in G\Omega_p(V)$ defined over the simplex Δ_T. Then, for any $(s,t) \in \Delta_T$, one has $S(X)_{s,t} \in \overline{T(V)}$.*

Proof. To prove the statement of the lemma it suffices to find a sequence of tensors $\{X^{(n)}_{s,t} \in T^k(V)\}_{n\in\mathbb{N}}$ that converges to $S(X_{s,t})$ in the $\|\cdot\|$-topology. Setting $X^{(n)}_{s,t} = (1, X^1_{s,t}, \dots, X^n_{s,t}, 0, \dots)$ and using the bounds from Extension Theorem SM1.14, we have

$$\|S(X)_{s,t}\| = \sqrt{\sum_{k=0}^{\infty} \|X^k_{s,t}\|^2_{V^{\otimes k}}} \leq \sqrt{\sum_{k=0}^{\infty} \frac{\omega(s,t)^{2k/p}}{(\beta_p(k/p)!)^2}} \leq \sum_{k=0}^{\infty} \frac{\omega(s,t)^{k/p}}{\beta_p(k/p)!}, \tag{4.2}$$

which is clearly a convergent series because the terms decay factorially, and for all $(s,t) \in \Delta_I$

$$\|X^{(n)}_{s,t} - S(X)_{s,t}\| = \sqrt{\sum_{k\geq n+1}^{\infty} \|X^k_{s,t}\|^2_{V^{\otimes k}}} \longrightarrow 0 \quad \text{as } n \to \infty. \tag{4.3}$$

■

Next we present our second main result, that is, we extend the definition of signature kernel to the space of geometric p-rough paths (Definition SM1.17) and show that this rough version of the signature kernel solves an iterated double integral equation of two one-forms analogous to the Goursat PDE (2.9) presented in section 2.

4.2. The signature kernel for geometric rough paths.

In what follows Δ_I, Δ_J will denote the two simplices

$$\Delta_I = \{(s,t) \in [i_-, i_+]^2 : i_- \leq s \leq t \leq i_+\}, \tag{4.4}$$

$$\Delta_J = \{(s,t) \in [j_-, j_+]^2 : j_- \leq s \leq t \leq j_+\}, \tag{4.5}$$

where $i_-, i_+, j_-, j_+ \geq 0$ are positive scalars such that $i_- < i_+$ and $j_- < j_+$.

Definition 4.3. *Let $p, q \geq 1$ be two scalars. Let $X \in G\Omega_p(V)$ and $Y \in G\Omega_q(V)$ be two geometric p- and q-rough paths, respectively, and controlled by two controls ω_X and ω_Y, respectively. The (rough) signature kernel $K_{(s_1,s_2),(t_1,t_2)} : G\Omega_p(V) \times G\Omega_q(V) \to \mathbb{R}$ is defined for any $(s_1, s_2) \in \Delta_I$ and $(t_1, t_2) \in \Delta_J$ as follows:*

$$K_{(s_1,s_2),(t_1,t_2)}(X, Y) = \left\langle S(X)_{s_1,s_2}, S(Y)_{t_1,t_2} \right\rangle, \tag{4.6}$$

where the inner product is taken in $\overline{T(V)}$.

Remark 4.4. On the one hand, the signature kernel of Definition 2.4 is configured to act on two time indices and is indexed on two paths. This choice was made in order to differentiate with respect to these and obtain the PDE (2.9). On the other hand, the rough signature kernel of Definition 4.3 acts on two (rough) paths and is indexed on time indices. When dealing with highly oscillatory objects like rough paths studied in this section, one can't expect to obtain a PDE, as these paths are far from being differentiable (even locally). However, we will nonetheless be able to use a density argument to prove our main result (Theorem 4.11).

Next we show that the rough signature kernel is bounded and continuous.

Lemma 4.5. *For any $(X, Y) \in G\Omega_p(V) \times G\Omega_q(V)$ and any $(s_1, s_2) \in \Delta_I, (t_1, t_2) \in \Delta_J$,*

$$\left\langle S(X)_{s_1,s_2}, S(Y)_{t_1,t_2} \right\rangle < +\infty. \tag{4.7}$$

Furthermore, the rough signature kernel $K_{(s_1,s_2),(t_1,t_2)}$ is continuous with respect to the product p, q-variation topology.

Proof. For any $(s_1, s_2) \in \Delta_I, (t_1, t_2) \in \Delta_J$ and by definition of the inner product $\langle \cdot, \cdot \rangle$ on $\overline{T(V)}$ we immediately have

$$\begin{aligned}
\langle S(X_{s_1,s_2}), S(Y_{t_1,t_2}) \rangle &= \sum_{k=0}^{\infty} \langle X^k_{s_1,s_2}, Y^k_{t_1,t_2} \rangle_{V^{\otimes k}} \\
&\leq \sum_{k=0}^{\infty} \|X^k_{s_1,s_2}\|_{V^{\otimes k}} \|Y^k_{t_1,t_2}\|_{V^{\otimes k}} && \text{(Cauchy–Schwarz)} \\
&\leq \sum_{k=0}^{\infty} \frac{\omega_X(s_1, s_2)^{k/p} \cdot \omega_Y(t_1, t_2)^{k/q}}{\beta_p(k/p)! \cdot \beta_q(k/q)!} && \text{(Ext. Theorem)} \\
&< +\infty.
\end{aligned}$$

Consider now the function $f_{(s_1,s_2),(t_1,t_2)} : G\Omega_p(V) \times G\Omega_q(V) \to \overline{T(V)} \times \overline{T(V)}$ defined as

$$f_{(s_1,s_2),(t_1,t_2)}(X, Y) = \left(S(X)_{s_1,s_2}, S(Y)_{t_1,t_2}\right) \tag{4.8}$$

and the function $g : \overline{T(V)} \times \overline{T(V)} \to \mathbb{R}$ defined as

$$g(T_1, T_2) = \langle T_1, T_2 \rangle. \tag{4.9}$$

The map g is clearly continuous in both variables in the sense of $\|\cdot\|$. By Extension Theorem SM1.14 we know that the two maps that extend (uniquely) X and Y, respectively,

to multiplicative functionals on the full tensor algebra $\overline{T(V)}$ are continuous in the p- and q-variation topologies, respectively. Therefore $f_{(s_1,s_2),(t_1,t_2)}$ is also continuous in both of its variables. Hence, $K_{(s_1,s_2),(t_1,t_2)} = g \circ f_{(s_1,s_2),(t_1,t_2)}$ is also continuous in both variables as it is the composition of two continuous functions. ∎

In the next section we present our second main result. The core technical tool we use in the proof is the notion of *integral of a one-form along a rough path*, discussed in subsection SM1.6.

4.3. A rough integral equation. To prove our main result we ought to give meaning to the following double integral:

$$\text{“}\mathcal{I}(X,Y) = \int\int K(X,Y)\langle dX, dY\rangle\text{.”} \tag{4.10}$$

We do so by constructing a double rough integral constructed as the composition of two *one-forms* (Definition SM1.6), as we shall explain next. In what follows we let $W := V \oplus \overline{T(V)}$.

Remark 4.6. In the following construction, the spaces V, W are swapped compared to the notation used in subsection SM1.6.

For a fixed tensor $A \in \overline{T(V)}$, consider the linear one-form $\alpha_A : W \to L(W,V)$ defined as follows:[3] for any $(b,B) \in W$

$$\alpha_A(b,B) = \begin{pmatrix} \langle A,B\rangle I_V & 0 \\ 0 & 0 \end{pmatrix}, \tag{4.11}$$

where $I_V : V \to V$ is the identity on V and where the inner product is taken in $\overline{T(V)}$.

Remark 4.7. Note that a linear one-form is $Lip(\gamma)$ for all $\gamma \geq 0$ (Definition SM1.6). Hence, by Definition SM1.22 of *integral of a one-form along a rough path*, we can integrate α_A along any geometric p-rough path with $p \geq 1$.

For any $p \geq 1$ and for any fixed geometric p-rough path $Z \in G\Omega_p(V)$, consider now a second linear one-form $\beta_Z : W \to L(W,\mathbb{R})$ defined as follows: for any $(a,A) \in W$ and for any $s,t \in \Delta_I$

$$\beta_{Z_{s,t}}(a,A) = \begin{pmatrix} \left\langle \left(\int_s^t \alpha_A(Z_u)dZ_u\right)^1, I_V \right\rangle & 0 \\ 0 & 0 \end{pmatrix}, \tag{4.12}$$

where the inner product is taken in V.

Remark 4.8. Note that the rough integral $\int \alpha_A(Z)dZ$ is a p-rough path with values in $T^{\lfloor p\rfloor}(V)$ (and that, by Extension Theorem SM1.14, its values in $T(V), T((V))$ are also uniquely determined). Here, by the notation $\left(\int \alpha_A(Z_u)dZ_u\right)^1$ we mean the canonical projection of the rough path $\int \alpha_A(Z)dZ$ onto V.

Remark 4.9. We note that for any $(b,B) \in W$, the data in $b \in V$ is ignored by both one-forms α_A and β_Z when acting on (b,B). We preferred to keep this notation, as we find it more in line with the standard notation used in rough integration.

[3] Perhaps more explicitly: for any $(b,B),(b',B') \in W$, $\alpha_A(b,B)(b',B') = \langle A,B\rangle b'$.

As the one-form β_Z is $Lip(\gamma)$ for all $\gamma \geq 0$, we can integrate β_Z along any q-rough path $\widetilde{Z}$ with $q \geq 1$ and use this integral as a definition for the double integral $\mathcal{I}$ of (4.10).

Definition 4.10. *Let Δ_I, Δ_J be two simplices and $p, q \geq 1$ two scalars. Let $X \in G\Omega_p(V)$ and $Y \in G\Omega_q(V)$ be two geometric p- and q-rough paths, respectively. For any $(s_1, s_2) \in \Delta_I$ and any $(t_1, t_2) \in \Delta_J$, define the double rough integral $\mathcal{I}_{(s_1,s_2),(t_1,t_2)}(X, Y)$ as*

$$\mathcal{I}_{(s_1,s_2),(t_1,t_2)}(X, Y) = \left(\int_{u=t_1}^{t_2} \beta_{X_{s_1,s_2}}(Y_u) dY_u \right)^1. \tag{4.13}$$

Note that this definition doesn't depend on the order of integration of X and Y. Next is our second main result, an analogue of Theorem 2.5 for the case of geometric rough paths.

Theorem 4.11. *Let Δ_I, Δ_J be two simplices, let $p, q \geq 1$ be two scalars, and let $X \in G\Omega_p(V)$ and $Y \in G\Omega_q(V)$ be two geometric p- and q-rough paths, respectively. For any $(s_1, s_2) \in \Delta_I$ and any $(t_1, t_2) \in \Delta_J$ the rough signature kernel of Definition 4.3 satisfies the equation:*

$$K_{(s_1,s_2),(t_1,t_2)}(X, Y) = 1 + \mathcal{I}_{(s_1,s_2),(t_1,t_2)}(X, Y), \tag{4.14}$$

where $\mathcal{I}$ is the double rough integral of Definition 4.10.

Proof. By [32, Theorem 4.12] if $Z \in G\Omega_p(V)$ is a geometric p-rough path and $\alpha : V \to L(V, W)$ is a $Lip(\gamma)$ one-form for some $\gamma > p$, then the mapping $Z \mapsto \int \alpha(Z) dZ$ is continuous from $G\Omega_p(V)$ to $G\Omega_p(W)$ in the p-variation topology.

For any $A \in \overline{T(V)}$ and any $\widetilde{Z} \in G\Omega_p(V)$ both α_A and $\beta_{\widetilde{Z}}$ as defined in (4.11) and (4.12), respectively, are linear one-forms, and hence $Lip(\gamma)$ for any $\gamma \geq 1$. Similarly if $\widetilde{Z} \in G\Omega_q(V)$. Thus, for any $(s_1, s_2) \in \Delta_I$ and any $(t_1, t_2) \in \Delta_J$, the map $\mathcal{I}_{(s_1,s_2),(t_1,t_2)} : G\Omega_p(V) \times G\Omega_q(V) \to \mathbb{R}$ is continuous in the p, q-variation product topology.

By Lemma 4.5, the rough signature kernel $K_{(s_1,s_2),(t_1,t_2)} : G\Omega_p(E) \times G\Omega_q(E) \to \mathbb{R}$ is also continuous with respect to the p, q-variation product topology.

Following the exact same steps as in the proof of Theorem 2.5, if $X \in G\Omega_1(V)$ and $Y \in G\Omega_1(V)$ are both of bounded variation, then the following double integral equation holds:

$$K_{(s_1,s_2),(t_1,t_2)}(X, Y) = 1 + \int_{s=s_1}^{s_2} \int_{t=t_1}^{t_2} K_{(s_1,s),(t_1,t)}(X, Y) \langle dX_s, dX_t \rangle, \tag{4.15}$$

which is equivalent to the equality $K_{(s_1,s_2),(t_1,t_2)}(X, Y) = 1 + \mathcal{I}_{(s_1,s_2),(t_1,t_2)}(X, Y)$.

By Definition SM1.17 of a geometric p- (respectively, q-)rough path as the limit of 1-rough paths in the p- (respectively, q-)variation topology, the space of continuous paths of bounded variation $G\Omega_1(V)$ is dense, $G\Omega_p(V)$ (respectively, $G\Omega_q(V)$). Two continuous functions that are equal on a dense subset of a set are also equal on the whole set. The functional equation $K_{(s_1,s_2),(t_1,t_2)}(\cdot, \cdot) = 1 + \mathcal{I}_{(s_1,s_2),(t_1,t_2)}(\cdot, \cdot)$ holds on $\Omega^1 G(V) \times \Omega^1 G(V)$, which concludes the proof by the previous density argument. ∎

This is the last theoretical result of this paper. In section 5 we tackle various machine learning tasks dealing with time series data.

5. Data science applications. In this section we evaluate our signature PDE kernel on three different tasks. First, we consider the task of multivariate time series classification on UEA[4] datasets [5] with a *support vector classifier* (SVC) and compare the performance obtained by equipping the same SVC configuration with various kernel functions, including ours. Second, we run a regression task to predict future (average) bitcoin prices from previously observed prices by means of a *support vector regressor* (SVR), and similarly to the previous experiment, we compare the performance produced by a variety of kernels. Lastly, we show how the signature kernel can be easily incorporated within simple optimization procedure to represent the distribution of a large ensemble of paths as a weighted average of a small number of selected paths from the ensemble while maintaining certain statistical properties [13].

In the presence of sequential inputs, well-designed kernels must be chosen with care [40]. In the case where all the time series inputs are of the same length, standard kernels on $\mathbb{R}^d$ can be deployed by stacking each dimension of the time series into one single vector. Standard choices of kernels include the linear and Gaussian (a.k.a. RBF) kernels. When the series are not of the same length, other kernels specifically designed for time series can be used to address this issue. Other than the signature PDE kernel introduced in this paper, to our knowledge only two other kernels for sequential data have been proposed in the literature: the truncated signature kernel [26] ($\mathrm{Sig}(n)$, where n denotes the truncation level) and the *global alignment kernel* (GAK) [15]. For the classification and regression experiments we made use of the SVC and SVR estimators, respectively, from the popular Python library `tslearn` [44].

Hyperparameter selection. The hyperparameters of the SVC and SVR estimators were selected by cross-validation via a grid search on the training set. For the classification we used the train-test split as provided by UEA, while for the regression we used an 80-20 split. Both the SVC and SVR estimators depend on a kernel k and on two scalar parameters C and γ. The range of values for C was chosen to be $\{1, 10, \dots, 10^4\}$ and the one for γ to be $\{10^{-4}, \dots, 10^4\}$ for all kernel functions included in the comparison. We benchmark our Sig-PDE kernel against the linear, RBF, and GAK [15] kernels as well as the truncated signature kernel $\mathrm{Sig}(n)$ from [26]. We found that the algorithm provided to us by [7] was in general slower than directly computing the truncated signatures with `iisignature` [39]. This is because the former was implemented as pure Python code, while `iisignature` is highly optimized and uses a `C++` backend. For this reason, we ended up computing $\mathrm{Sig}(n)$ without a kernel trick for all the experiments. The truncation n is chosen from the range $\{2, \dots, 6\}$. Furthermore, we added a variety of additional hyperparameters to $\mathrm{Sig}(n)$ consisting in (1) scaling the paths by different scalar scales, (2) normalizing the truncated signatures by multiplying (or not) each level $\ell \in \{1, \dots, n\}$ by $\ell!$, and (3) equipping the SVC/SVR with a linear or RBF kernel (indexed on truncated signatures). For our Sig-PDE we used the RBF-lifted version with parameter σ taken in the range $\{10^{-3}, \dots, 10^1\}$. All the experiments are reproducible using the code in https://github.com/crispitagorico/sigkernel and following the instructions thereafter.

5.1. Multivariate time series classification. The support vector classifier (SVC) [49] is one of the simplest yet widely used supervised learning models for classification. It has been successfully used in the fields of text classification [46], image retrieval [45], mathematical

[4]Data available at https://timeseriesclassification.com.

finance [21], medicine [17], etc. We considered various UEA datasets [5] of input-output pairs $\{(x_i, y_i)\}_{i=1}^n$, where each x_i is a multivariate time series and each y_i is the corresponding class. In Table 1 we display the performance of the same SVC equipped with different kernels (including ours). As the results show, our Sig-PDE kernel is systematically among the top two classifiers across all the datasets (except for FingerMovements and UWaveGestureLibrary) and always outperforms its truncated counterpart, which often overfits during training.

Table 1
Test set classification accuracy (in %) on UEA multivariate time series datasets.

Datasets/kernels	Linear	RBF	GAK	Sig(n)	Sig-PDE
ArticularyWordRecognition	98.0	98.0	98.0	92.3	**98.3**
BasicMotions	87.5	97.5	97.5	97.5	**100.0**
Cricket	91.7	91.7	**97.2**	86.1	**97.2**
ERing	92.2	92.2	**93.7**	84.1	93.3
Libras	73.9	77.2	79.0	**81.7**	**81.7**
NATOPS	90.0	92.2	90.6	88.3	**93.3**
RacketSports	76.9	78.3	84.2	80.2	**84.9**
FingerMovements	57.0	60.0	**61.0**	51.0	58.0
Heartbeat	70.2	73.2	70.2	72.2	**73.6**
SelfRegulationSCP1	86.7	87.3	**92.4**	75.4	88.7
UWaveGestureLibrary	80.0	**87.5**	**87.5**	83.4	87.0

5.2. Predicting bitcoin prices. In the last few years, there has a remarkable rise of cryptocurrency trading where the most popular currency, bitcoin, reached its peak at almost 20,000 USD/BTC at the end of 2017, followed by a big crash in November 2018, when the price dropped to around 3000 USD/BTC. In this section, we will use daily bitcoin-to-USD prices data[5] from Gemini, which is one of the biggest cryptocurrency trading platforms in the U.S. Our goal is to use a 36-day window to predict the mean price of the next 2 days. As shown in Table 2, SVR equipped with the Sig-PDE is able to generalize better to unseen prices and produces better predictions on the test set in terms of mean absolute percentage error compared to all other benchmarks. We also note that the truncated signature kernel doesn't seem to generalize well to unseen observation for this regression example. In Figure 4 we plot the predictions obtained with SVR-Sig-PDE on the train and test sets.

5.3. Moments-matching reduction algorithm for the support of a discrete measure on paths. As described in [8], *herding* refers to any procedure to approximate integrals of functions in a *reproducing kernel Hilbert space* (RKHS). In particular, such a procedure can be useful in estimating *kernel mean embeddings* (KMEs), as we shall explain next. Consider a set $\mathcal{X}$ and a feature map Φ from $\mathcal{X}$ to an RKHS H with k being the associated positive definite kernel. All elements of H may be identified with real functions f on $\mathcal{X}$ defined by $f(x) = \langle f, \Phi(x)\rangle$ for $x \in \mathcal{X}$. Following [43] for a fixed probability measure μ on $\mathcal{X}$, we seek to approximate the KME $\mathbb{E}_\mu\Phi := \int_{\mathcal{X}} \Phi(x)d\mu(x)$, which belongs to the convex hull of $\{\Phi(x)\}_{x\in\mathcal{X}}$ [4]. To approximate $\mathbb{E}_\mu\Phi$, we consider n points $x_1, \ldots, x_n \in \mathcal{X}$ combined linearly with positive

[5]Data is from https://www.cryptodatadownload.com/.

Table 2

Test set mean absolute percentage error (MAPE) (in %) to predict the average bitcoin price over the next 2 *days given prices over the previous* 36 *days.*

Kernel	RBF	GAK	Sig(n)	Sig-PDE
MAPE	4.094	4.458	13.420	**3.253**

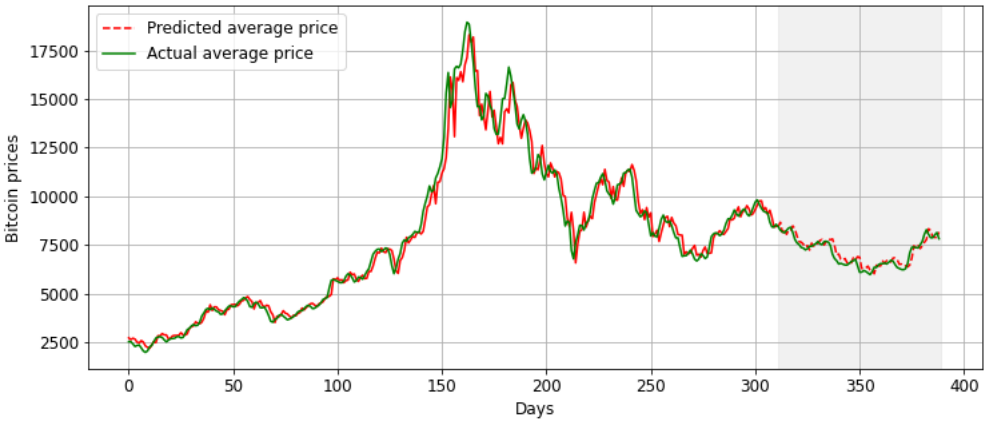

Figure 4. *SVR-Sig-PDE predictions of average bitcoin prices over the next* 2 *days from bitcoin prices over the previous* 36 *days. In white are the predicted prices in the training set; in gray are the predicted prices in the test set. Trading days range from* 2017-06-01 *to* 2018-08-01.

weights $w_1, \dots, w_n$ that sum to 1. We then consider the discrete measure $\nu = \sum_{i=1}^{n} w_i \delta_{x_i}$ and, as shown in [4], we have that

$$\sup_{f \in H, ||f|| \leq 1} |\langle \mathbb{E}_\nu \Phi, f \rangle - \langle \mathbb{E}_\mu \Phi, f \rangle| = ||\mathbb{E}_\nu \Phi - \mathbb{E}_\mu \Phi||_H, \tag{5.1}$$

which means that controlling $\mathbb{E}_{\hat{\mu}} \Phi - \mathbb{E}_\mu \Phi$ is enough to control the error in computing the expectation for all $f \in H$ with norm bounded by 1. We are interested in the setting where $\mathcal{X}$ is a set of paths of bounded variation taking values on a d-dimensional space E (or in practice a set of multivariate time series, for example). The signature being a natural feature map for sequential data, we set $\Phi = S$, k to be the untruncated signature kernel and $H = \overline{T(E)}$. Following [30, 13], we consider the problem of reducing the size of the support in $\mathcal{X}$ of a discrete measure μ while preserving some of its statistical properties. Suppose $\#supp(\mu) = N$, where N is large, and $\mu = \sum_{i=1}^{N} \alpha_i \delta_{x_i}, x_i \in \mathcal{X}$. We call a measure ν on $\mathcal{X}$ a *reduced measure* with respect to μ if

$$(1)\ supp(\nu) \subset supp(\mu) \quad \text{and} \quad (2)\ \mathbb{E}_\nu S \approx \mathbb{E}_\mu S \quad \text{(in some suitable norm)}.$$

We fix the size of the support of the reduced measure ν to be $\#supp(\nu) = n$, so that $n \ll N$. Because of condition (1) we have that ν is of the form $\mu = \sum_{i=1}^{N} \beta_i \delta_{x_i}$, where all but n of the

weights β_i's are equal to 0. Therefore the vector of weights $\beta = (\beta_1, \dots, \beta_N) \in \mathbb{R}^N$ is sparse. We are interested in the optimization problem

$$\begin{aligned}\min_{\beta\in\mathbb{R}^N} \|\mathbb{E}_\nu S - \mathbb{E}_\mu S\|^2_{T(E)} &= \min_{\beta\in\mathbb{R}^N} \left\|\sum_{i=1}^N (\alpha_i - \beta_i) S(x_i)\right\|^2_{T(E)} \\ &= \min_{\beta\in\mathbb{R}^N} \left\langle \sum_{i=1}^N (\alpha_i - \beta_i) S(x_i), \sum_{j=1}^N (\alpha_j - \beta_j) S(x_i) \right\rangle_{T(E)} \\ &= \min_{\beta\in\mathbb{R}^N} \underbrace{\sum_{i,j=1}^N (\alpha_i - \beta_i)(\alpha_j - \beta_j) k(x_i, x_j)}_{:=L(\beta)},\end{aligned}$$

where k is the signature kernel. This minimization will not yield a sparse vector β. To induce sparsity we use an l_1 penalization on the weights β as in lasso, which amounts to the following Lagrangian minimization:

$$\min_{\beta\in\mathbb{R}^N} L(\beta) + \lambda ||\beta||_1, \tag{5.2}$$

where λ is a penalty parameter determined by the size n of the support of ν. Equation (5.2) minimizes a function $f : \mathbb{R}^N \to \mathbb{R}$ that can be decomposed as $f = L + h$, where L is differentiable and $h = \lambda|| \cdot ||_1$ is convex but nondifferentiable, so a gradient descent algorithm can't be directly applied. *Subgradient descent methods* are classical algorithms that address this issue but have poor convergence rates [3]. A better choice of algorithms for this particular problem are called *proximal gradient methods* [41]. Define the soft-thresholding operator $A_\gamma : \mathbb{R}^N \to \mathbb{R}^N$ as follows:

$$A_\gamma(\beta)_i = \begin{cases} \beta_i - \gamma & \text{if } \beta_i > \gamma, \\ 0 & \text{if } |\beta_i| \le \gamma, \\ \beta_i + \gamma & \text{if } \beta_i < -\gamma. \end{cases} \tag{5.3}$$

Then it can be shown [41] that β^* is a minimizer of the optimization (5.2) if and only if β^* solves the following fixed point problem:

$$\beta^* = A_\gamma(\beta^* - \gamma \nabla_\beta L(\beta^*)). \tag{5.4}$$

The fixed point problem (5.4) can be solved iteratively fixing $\beta^0 \in \mathbb{R}^N$ and, for $k \ge 1$,

$$\beta^{k+1} = A_\gamma(\beta^k - \gamma \nabla_\beta L(\beta^k)). \tag{5.5}$$

Proximal gradient descent methods converge with rate $O(1/\epsilon)$, which is an order of magnitude better that the $O(1/\epsilon^2)$ convergence rate of subgradient methods [41].

We apply the above proximal gradient descent algorithm to an example of a set of 30 sample paths of fractional Brownian motion (fBM) with Hurst exponent drawn uniformly at random from $\{0.2, 0.5, 0.8\}$ (note that fBM(0.5) corresponds to Brownian motion). The

goal is to compute a reduced measure with smaller support size. We choose a value of the penalization constant λ in (5.2) so that the new support is of size $= 6$. The selected paths with corresponding weights are displayed in Figure 5. This selection is clearly well balanced across the samples (2 paths per exponent), so more likely to well represent the initial ensemble.

Remark 5.1. We note that our signature PDE kernel has been successfully deployed to perform *distribution regression on sequential data* in [28] and malware detection in [11].

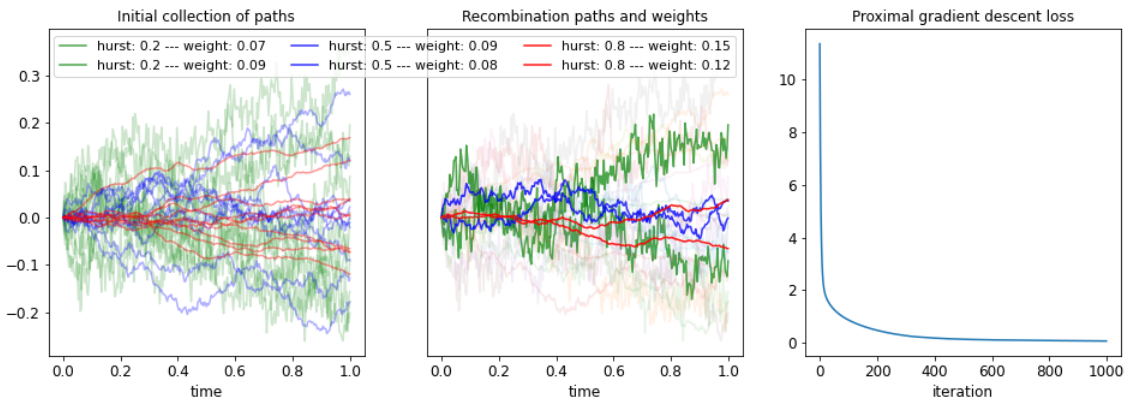

Figure 5. *On the left are* 30 *fBM sample paths. In the middle are the results obtained by solving the optimization* (5.2)*. On the right is the loss as a function of the proximal gradient descent iteration.*

6. Conclusion. In this paper we introduce the signature PDE kernel and show that when paths are continuously differentiable the latter solves a hyperbolic PDE. We recognize the connection with a well-known class of differential equations known in the literature as Goursat problems. Our Goursat PDE can be solved numerically using state-of-the-art hyperbolic PDE solvers; we propose an efficient finite difference scheme to do so that has linear time complexity when implemented on GPU and analyze its convergence properties. We extend the previous analysis to the case of geometric rough paths and establish a rough integral equation analogous to the aforementioned Goursat problem. Finally we demonstrate the effectiveness of our kernel in various data science applications dealing with sequential data.

Appendix A. Error analysis for the numerical scheme. In this section, we show that the finite difference scheme (3.7) achieves a second order convergence rate for the Goursat problem (2.9). Our analysis is based on an explicit representation of the PDE solution.

Theorem A.1 (Example 17.4 from [38]). *Consider the following specific case of the general Goursat problem* (3.2) *on the domain* $\mathcal{D} = \{(s,t) \mid u \le s \le u', v \le t \le v'\}$*:*

$$\frac{\partial^2 k}{\partial s \partial t} = C_3 k, \tag{A.1}$$

where C_3 is constant and the boundary data $k(s,v) = \sigma(s)$, $k(u,t) = \tau(t)$ is differentiable. Then the solution k can be expressed as

$$k(s,t) = k(u,v)\,R(s-u,t-v) + \int_u^s \sigma'(r)\,R(s-r,t-v)\,dr + \int_v^t \tau'(r)\,R(s-u,t-r)\,dr \tag{A.2}$$

for $s,t \in \mathcal{D}$, where the Riemann function R is defined as $R(a,b) := J_0\big(2i\sqrt{C_3\,ab}\big)$ for $a,b \geq 0$, with J_0 denoting the zero order Bessel function of the first kind.

Remark A.2. To simplify notation, we shall use the nth order modified Bessel function $I_n(z) := i^{-n}J_n(iz)$. It directly follows from the series expansion of $J_n(2z)$ [1, section 4.5] that

$$I_n(2z) = \left(\sum_{k=0}^{\infty} \frac{z^{2k}}{k!(n+k)!}\right) z^n. \tag{A.3}$$

From the identities (4.6.1), (4.6.2), (4.6.5), and (4.6.6) in [1], we can compute derivatives of I_0 as

$$I_0'(z) = I_1(z), \tag{A.4}$$

$$I_0''(z) = I_2(z) + z^{-1}I_1(z). \tag{A.5}$$

Using Theorem A.1 and the above identities, we will perform a local error analysis for the explicit scheme (3.7).

Theorem A.3 (local error estimates for the explicit scheme). *Consider the Goursat problem (A.1) on the domain $\mathcal{D} = \{(s,t) \mid u \leq s \leq u', v \leq t \leq v'\}$:*

$$\frac{\partial^2 k}{\partial s \partial t} = C_3 k,$$

where C_3 is constant and the boundary data $u(s,v) = \sigma(s)$, $u(u,t) = \tau(t)$ is differentiable and of bounded variation. We define the local approximation error of the explicit scheme (3.7) as

$$E(s,t) := k(s,t) - \Big(k(s,v) + k(u,t) - k(u,v) + \frac{1}{2}\big(k(s,v) + k(u,t)\big)\,C_3(s-u)(t-v)\Big).$$

Then

$$\begin{aligned} |E(s,t)| \leq{}& \frac{1}{2}|C_3|\big(\|\sigma\|_{1,[u,s]} + \|\tau\|_{1,[v,t]}\big)(s-u)(t-v) \sup_{z\in[0,C_3(s-u)(t-v)]} \left|\frac{I_1(2\sqrt{z})}{\sqrt{r}}\right| \\ &+ \frac{1}{2}|C_3||k(s,v) + k(u,t)|(s-u)(t-v) \sup_{z\in[0,C_3(s-u)(t-v)]} \left|\frac{I_1(2\sqrt{z})}{\sqrt{z}} - 1\right|. \end{aligned} \tag{A.6}$$

In addition, if σ, τ are twice differentiable and their derivatives have bounded variation, then

(A.7)

$$
\begin{aligned}
|E(s,t)| \le{}& \frac{1}{2}|C_3|\big(\|\sigma'\|_{1,[u,s]}(s-u) + \|\tau'\|_{1,[v,t]}(t-v)\big)(s-u)(t-v) \sup_{z\in[0,C_3(s-u)(t-v)]}\left|\frac{I_1(2\sqrt{z}\,)}{\sqrt{z}}\right| \\
&+ \frac{1}{12}|C_3|^2\big(|\sigma'(u)|(s-u) + |\tau'(v)|(t-v)\big)(s-u)^2(t-v)^2 \sup_{z\in[0,C_3(s-u)(t-v)]}\left|\frac{I_2(2\sqrt{z}\,)}{z}\right| \\
&+ \frac{1}{2}|C_3||k(s,v)+k(u,t)|(s-u)(t-v) \sup_{z\in[0,C_3(s-u)(t-v)]}\left|\frac{I_1(2\sqrt{z}\,)}{\sqrt{r}} - 1\right|.
\end{aligned}
$$

Remark A.4. From (A.3), it is clear that $\frac{I_1(2\sqrt{z}\,)}{\sqrt{z}} \sim 1$, $\frac{I_2(2\sqrt{z}\,)}{z} \sim \frac{1}{2}$, and $\frac{I_1(2\sqrt{z}\,)}{\sqrt{z}} - 1 \sim \frac{1}{2}z$.

Proof. To begin, we decompose the approximation error as $E(s,t) = E_1 + E_2$, where

$$
\begin{aligned}
E_1 &:= k(s,t) - \Big(k(s,v) + k(u,t) - k(u,v) + \frac{1}{2}\big(k(s,v)+k(u,t)\big)\big(R(s-u,t-v)-1\big)\Big), \\
E_2 &:= \frac{1}{2}\big(k(s,v)+k(u,t)\big)\big(C_3(s-u)(t-v) - \big(R(s-u,t-v)-1\big)\big).
\end{aligned}
$$

Since $R(s-u,0) = R(0,t-v) = 1$, $\sigma(s) = \tau(v) = k(u,v)$, $\sigma(s) = k(s,v)$, and $\tau(t) = k(u,t)$, it follows that

$$
\begin{aligned}
&k(s,v) + k(u,t) - k(u,v) + \frac{1}{2}\big(k(s,v)+k(u,t)\big)\big(R(s-u,t-v)-1\big) \\
&\quad = k(u,v)\,R(s-u,t-v) + \frac{1}{2}\big(k(s,v) - 2k(u,v) + k(u,t)\big)\big(R(s-u,t-v)+1\big) \\
&\quad = k(u,v)\,R(s-u,t-v) + \frac{1}{2}\big((\sigma(s)-\sigma(u)) + (\tau(t)-\tau(v))\big)\big(R(s-u,t-v)+1\big) \\
&\quad = k(u,v)\,R(s-u,t-v) + \frac{1}{2}\Big(\int_u^s \sigma'(r)\,dr + \int_v^t \tau'(r)\,dr\Big)\big(R(s-u,t-v)+1\big) \\
&\quad = k(u,v)\,R(s-u,t-v) + \frac{1}{2}\int_u^s \sigma'(r)\big(R(0,t-v) + R(s-u,t-v)\big)\,dr \\
&\qquad\qquad + \frac{1}{2}\int_v^t \tau'(r)\big(R(s-u,0) + R(s-u,t-v)\big)\,dr.
\end{aligned}
$$

Hence by (A.2), we can write $E_1 = E_3 + E_4$, where the error terms E_3 and E_4 are given by

$$
\begin{aligned}
E_3 &:= \int_u^s \sigma'(r)\Big(R(s-r,t-v) - \frac{1}{2}\big(R(0,t-v) + R(s-u,t-v)\big)\Big)\,dr, \\
E_4 &:= \int_v^t \tau'(r)\Big(R(s-u,t-r) - \frac{1}{2}\big(R(s-u,0) + R(s-u,t-v)\big)\Big)\,dr.
\end{aligned}
$$

The integrand of E_3 can be estimated as

$$\begin{aligned}
(\text{A.8}) \qquad & \left|R(s-r,t-v)-\frac{1}{2}\big(R(0,t-v)+R(s-u,t-v)\big)\right| \\
&= \left|\frac{1}{2}\big(R(s-r,t-v)-R(0,t-v)\big)-\frac{1}{2}\big(R(s-u,t-v)-R(s-r,t-v)\big)\right| \\
&\le \frac{1}{2}\left|\int_0^{s-r}\frac{\partial R}{\partial w}(w,t-v)\,dw\right|+\frac{1}{2}\left|\int_{s-r}^{s-u}\frac{\partial R}{\partial w}(w,t-v)\,dw\right| \\
&\le \frac{1}{2}(s-r)\sup_{w\in[0,s-r]}\left|\frac{\partial R}{\partial w}(w,t-v)\right|+\frac{1}{2}(r-u)\sup_{w\in[s-r,s-u]}\left|\frac{\partial R}{\partial w}(w,t-v)\right| \\
&\le \frac{1}{2}(s-u)\sup_{w\in[0,s-u]}\left|\frac{\partial R}{\partial w}(w,t-v)\right|.
\end{aligned}$$

By applying the formulae (A.4) and (A.5) to the function R, we can compute its derivatives,

$$(\text{A.9}) \qquad \frac{\partial R}{\partial w}(w,t-v)=\frac{C_3(t-v)I_1\big(2\sqrt{C_3(w(t-v))}\,\big)}{\sqrt{C_3 w(t-v)}},$$

$$(\text{A.10}) \qquad \frac{\partial^2 R}{\partial^2 w}(w,t-v)=\frac{C_3(t-v)I_2\big(2\sqrt{C_3(w(t-v))}\,\big)}{w}.$$

Therefore, it now follows from (A.8) and (A.9) that

$$\begin{aligned}
|E_3| &\le \int_u^s \big|\sigma'(r)\big|\left|R(s-r,t-v)-\frac{1}{2}\big(R(0,t-v)+R(s-u,t-v)\big)\right|dr \\
&\le \frac{1}{2}\int_u^s \big|\sigma'(r)\big|\,dr\,(s-u)\sup_{w\in[0,s-u]}\left|\frac{\partial R}{\partial w}(w,t-v)\right| \\
&\le \frac{1}{2}|C_3|\|\sigma\|_{1,[u,s]}(s-u)(t-v)\sup_{z\in[0,C_3(s-u)(t-v)]}\left|\frac{I_1(2\sqrt{z}\,)}{\sqrt{z}}\right|,
\end{aligned}$$

and we can obtain a similar estimate for E_4 (where $\|\tau\|_{1,[v,t]}$ would appear instead of $\|\sigma\|_{1,[u,s]}$). From the estimates for E_3 and E_4, we have

$$|E_1| \le \frac{1}{2}|C_3|\big(\|\sigma\|_{1,[u,s]}+\|\tau\|_{1,[v,t]}\big)(s-u)(t-v)\sup_{z\in[0,C_3(s-u)(t-v)]}\left|\frac{I_1(2\sqrt{z}\,)}{\sqrt{z}}\right|.$$

Estimating E_2 is straightforward as

$$\begin{aligned}
&\big|R(s-u,t-v)-\big(1+C_3(s-u)(t-v)\big)\big| \\
&\quad= \big|\big(I_0\big(2\sqrt{C_3\,(s-u)(t-v)}\,\big)-I_0(0)\big)-C_3(s-u)(t-v)\big| \\
&\quad= \left|\int_0^{C_3(s-u)(t-v)}\left(\frac{I_1(2\sqrt{z}\,)}{\sqrt{z}}-1\right)dz\right| \\
&\quad\le |C_3|(s-u)(t-v)\sup_{z\in[0,C_3(s-u)(t-v)]}\left|\frac{I_1(2\sqrt{z}\,)}{\sqrt{z}}-1\right|.
\end{aligned}$$

Using the above estimates for E_1 and E_2, we obtain (A.6) as required. For the remainder of this proof we will assume that σ, τ are twice differentiable and σ', τ' have bounded variation. In this case, we can apply the fundamental theorem of calculus to the integrand of E_3 so that

$$\begin{aligned} E_3 &= \int_u^s \sigma'(r)\left(R(s-r,t-v) - \frac{1}{2}\big(R(0,t-v)+R(s-u,t-v)\big)\right) dr \\ &= \int_u^s \left(\sigma'(u) + \int_u^r \sigma''(w)\,dw\right)\left(R(s-r,t-v) - \frac{1}{2}\big(R(0,t-v)+R(s-u,t-v)\big)\right) dr. \end{aligned}$$

Note that by the well-known error estimate for the trapezium rule, we have

$$\begin{aligned} &\left|\int_u^s R(s-r,t-v)\,dr - \frac{1}{2}(s-u)\big(R(0,t-v)+R(s-u,t-v)\big)\right| \\ &\qquad \le \frac{1}{12}(s-u)^3 \sup_{w\in[0,s-u]}\left|\frac{\partial^2 R}{\partial w^2}(w,t-v)\right|. \end{aligned}$$

Recall that this derivative was given by (A.10). It now follows from the above and (A.8) that

$$\begin{aligned} |E_3| &\le \big|\sigma'(u)\big|\left|\int_u^s R(s-r,t-v)\,dr - \frac{1}{2}(s-u)\big(R(0,t-v)+R(s-u,t-v)\big)\right| \\ &\qquad + \int_u^s\int_u^r \big|\sigma''(w)\big|\,dw\left|R(s-r,t-v) - \frac{1}{2}\big(R(0,t-v)+R(s-u,t-v)\big)\right| dr \\ &\le \frac{1}{12}|C_3|^2|\sigma'(u)|(s-u)^3(t-v)^2 \sup_{z\in[0,C_3(s-u)(t-v)]}\left|\frac{I_2(2\sqrt{z}\,)}{z}\right| \\ &\qquad + \frac{1}{2}|C_3|\|\sigma'\|_{1,[u,s]}(s-u)^2(t-v) \sup_{z\in[0,C_3(s-u)(t-v)]}\left|\frac{I_1(2\sqrt{z}\,)}{\sqrt{z}}\right|. \end{aligned}$$

Applying the same argument to E_4 leads to the second estimate (A.7) as required. ∎

From the estimate (A.7) we see that the proposed finite difference scheme achieves a local error that is $O(h^4)$ when the domain $\mathcal{D}$ has a small height and width of h (and provided the boundary data is smooth enough). Since discretizing a PDE on an $n\times n$ grid involves $(n-1)^2$ steps, we expect the proposed scheme to have second order convergence.

Theorem A.5 (global error estimate). *Let $\widetilde{k}$ be a numerical solution obtained by applying the proposed finite difference scheme (Definition 3.3) to the Goursat PDE (2.9) on the grid P_λ where x and y are piecewise linear with respect to the grids $\mathcal{D}_I$ and $\mathcal{D}_J$. In particular, we are assuming there exists a constant M, which is independent of λ, such that*

$$\sup_{\mathcal{D}} |\langle \dot{x}_s, \dot{y}_t\rangle| < M.$$

Then there exists a constant $K>0$ depending on M and $k_{x,y}$, but independent of λ, such that

$$\sup_{\mathcal{D}} \big|k_{x,y}(s,t) - \widetilde{k}(s,t)\big| \le \frac{K}{2^{2\lambda}} \tag{A.11}$$

for all $\lambda \ge 0$.

Proof. Using the solution $k_{x,y}$, we define another approximation k' on P_λ as

$$k'(s_{i+1},t_{j+1}) := k_{x,y}(s_{i+1},t_j) + k_{x,y}(s_i,t_{j+1}) - k_{x,y}(s_i,t_j) + \frac{1}{2}\langle x_{s_{i+1}} - x_{s_i}, y_{t_{j+1}} - y_{t_j}\rangle\big(k_{x,y}(s_{i+1},t_j) + k_{x,y}(s_i,t_{j+1})\big).$$

It follows from Theorem 3.1 that the PDE solution and boundary data are smooth on each small rectangle in P_λ. So by (A.7) there exists $K_1 > 0$, depending on M and $k_{x,y}$, such that

$$\big|k_{x,y}(s_{i+1},t_{j+1}) - k'(s_{i+1},t_{j+1})\big| \le K_1(s_{i+1}-s_i)(t_{j+1}-t_j)\big((s_{i+1}-s_i)^2 + (s_{i+1}-s_i)(t_{j+1}-t_j) + (t_{j+1}-t_j)^2\big).$$

Taking the difference between $\tilde{k}(s_{i+1},t_{j+1})$ and $\hat{k}(s_{i+1},t_{j+1})$ gives

$$\begin{aligned}&\big|k'(s_{i+1},t_{j+1}) - \hat{k}(s_{i+1},t_{j+1})\big| \\ &\quad\le \big|k_{x,y}(s_i,t_j) - \hat{k}(s_i,t_j)\big| + \frac{1}{2}\Big(1 + \big|\langle x_{s_{i+1}} - x_{s_i}, y_{t_{j+1}} - y_{t_j}\rangle\big|\Big)\big|k_{x,y}(s_{i+1},t_j) - \hat{k}(s_{i+1},t_j)\big| \\ &\qquad + \frac{1}{2}\Big(1 + \big|\langle x_{s_{i+1}} - x_{s_i}, y_{t_{j+1}} - y_{t_j}\rangle\big|\Big)\big|k_{x,y}(s_i,t_{j+1}) - \hat{k}(s_i,t_{j+1})\big|.\end{aligned}$$

Hence by the triangle inequality, we obtain a recurrence relation for the approximation errors,

$$\begin{aligned}&\big|k_{x,y}(s_{i+1},t_{j+1}) - \hat{k}(s_{i+1},t_{j+1})\big| \\ &\quad\le \big|k_{x,y}(s_i,t_j) - \hat{k}(s_i,t_j)\big| \\ &\qquad + \frac{1}{2}\Big(1 + M(s_{i+1}-s_i)(t_{j+1}-t_j)\Big)\big|k_{x,y}(s_{i+1},t_j) - \hat{k}(s_{i+1},t_j)\big| \\ &\qquad + \frac{1}{2}\Big(1 + M(s_{i+1}-s_i)(t_{j+1}-t_j)\Big)\big|k_{x,y}(s_i,t_{j+1}) - \hat{k}(s_i,t_{j+1})\big| \\ &\qquad + K_1(s_{i+1}-s_i)(t_{j+1}-t_j)\big((s_{i+1}-s_i)^2 + (s_{i+1}-s_i)(t_{j+1}-t_j) + (t_{j+1}-t_j)^2\big).\end{aligned}$$

Since each $(s_{i+1}-s_i)$ and $(t_{j+1}-t_j)$ is proportional to $2^{-\lambda}$, the result for the explicit scheme follows by iteratively applying the above recurrence relation. ∎

Acknowledgments. We thank Maud Lemercier for her help with the implementation of `sigkernel`, Franz Király and Harald Oberhauser for their helpful comments, and the referees for pointing out an error in the experiments in the previous version of the paper.

REFERENCES

[1] G. E. Andrews, R. Askey, and R. Roy, *Special Functions*, Encyclopedia Math. Appl. 71, Cambridge University Press, Cambridge, UK, 2001.

[2] I. P. Arribas, G. M. Goodwin, J. R. Geddes, T. Lyons, and K. E. Saunders, *A signature-based machine learning model for distinguishing bipolar disorder and borderline personality disorder*, Transl. Psychiatry, 8 (2018), pp. 1–7.

[3] F. Bach, R. Jenatton, J. Mairal, and G. Obozinski, *Optimization with sparsity-inducing penalties*, Found. Trends Mach. Learn., 4 (2012), pp. 1–106.

[4] F. R. Bach, S. Lacoste-Julien, and G. Obozinski, *On the equivalence between herding and conditional gradient algorithms*, in Proceedings of the 29th International Conference on Machine Learning (ICML), Omnipress, 2012, pp. 1355–1362.
[5] A. Bagnall, J. Lines, A. Bostrom, J. Large, and E. Keogh, *The great time series classification bake off: A review and experimental evaluation of recent algorithmic advances*, Data Min. Knowl. Discov., 31 (2017), pp. 606–660.
[6] C. Bayer, P. Friz, and J. Gatheral, *Pricing under rough volatility*, Quant. Finance, 16 (2016), pp. 887–904.
[7] C. T. C. Salvi, *private communication*, 2020.
[8] Y. Chen, M. Welling, and A. Smola, *Super-samples from kernel herding*, in Proceedings of the 26th Conference on Uncertainty in Artificial Intelligence, AUAI Press, 2010, pp. 109–116.
[9] I. Chevyrev and H. Oberhauser, *Signature Moments to Characterize Laws of Stochastic Processes*, preprint, https://arxiv.org/abs/1810.10971, 2018.
[10] K. Chouk and M. Gubinelli, *Rough Sheets*, preprint, https://arxiv.org/abs/1406.7748, 2014.
[11] T. Cochrane, P. Foster, V. Chhabra, M. Lemercier, C. Salvi, and T. Lyons, *SK-Tree: A Systematic Malware Detection Algorithm on Streaming Trees via the Signature Kernel*, preprint, https://arxiv.org/abs/2102.07904, 2021.
[12] *CoRoPa: Computational Rough Paths*, software library, 2010, http://coropa.sourceforge.net/.
[13] F. Cosentino, H. Oberhauser, and A. Abate, *A Randomized Algorithm to Reduce the Support of Discrete Measures*, preprint, https://arxiv.org/abs/2006.01757, 2020.
[14] C. Cuchiero, W. Khosrawi, and J. Teichmann, *A generative adversarial network approach to calibration of local stochastic volatility models*, Risks, 8 (2020), 101.
[15] M. Cuturi, *Fast global alignment kernels*, in Proceedings of the 28th International Conference on Machine Learning (ICML-11), Omnipress, 2011, pp. 929–936.
[16] J. T. Day, *A Runge-Kutta method for the numerical solution of the Goursat problem in hyperbolic partial differential equations*, Comput. J., 9 (1966), pp. 81–83.
[17] T. S. Furey, N. Cristianini, N. Duffy, D. W. Bednarski, M. Schummer, and D. Haussler, *Support vector machine classification and validation of cancer tissue samples using microarray expression data*, Bioinform., 16 (2000), pp. 906–914.
[18] J. Gatheral, T. Jaisson, and M. Rosenbaum, *Volatility is rough*, Quant. Finance, 18 (2018), pp. 933–949.
[19] E. Goursat, *A Course in Mathematical Analysis: Part 2. Differential Equations*, Vol. 2, Dover, 1916.
[20] T. Hofmann, B. Schölkopf, and A. J. Smola, *Kernel methods in machine learning*, Ann. Statist., 36 (2008), pp. 1171–1220.
[21] W. Huang, Y. Nakamori, and S.-Y. Wang, *Forecasting stock market movement direction with support vector machine*, Comput. Oper. Res., 32 (2005), pp. 2513–2522.
[22] P. Kidger, P. Bonnier, I. Perez Arribas, C. Salvi, and T. Lyons, *Deep signature transforms*, in Advances in Neural Information Processing Systems, Vol. 32, 2019, pp. 3099–3109.
[23] P. Kidger, J. Foster, X. Li, H. Oberhauser, and T. Lyons, *Neural SDEs as Infinite-Dimensional GANs*, preprint, https://arxiv.org/abs/2102.03657, 2021.
[24] P. Kidger and T. Lyons, *Signatory: Differentiable Computations of the Signature and Logsignature Transforms, on Both CPU and GPU*, preprint, https://arxiv.org/abs/2001.00706, 2020; see also https://github.com/patrick-kidger/signatory.
[25] F. J. Király and H. Oberhauser, *Kernels for Sequentially Ordered Data*, preprint, https://arxiv.org/abs/1601.08169, 2016.
[26] F. J. Király and H. Oberhauser, *Kernels for sequentially ordered data*, J. Mach. Learn. Res., 20 (2019), 31.
[27] M. Lees, *The Goursat problem*, J. Soc. Indust. Appl. Math., 8 (1960), pp. 518–530, https://doi.org/10.1137/0108036.
[28] M. Lemercier, C. Salvi, T. Damoulas, E. V. Bonilla, and T. Lyons, *Distribution Regression for Continuous-Time Processes via the Expected Signature*, preprint, https://arxiv.org/abs/2006.05805, 2020.
[29] X. Li, T.-K. L. Wong, R. T. Chen, and D. K. Duvenaud, *Scalable gradients and variational inference for stochastic differential equations*, in Symposium on Advances in Approximate Bayesian Inference,

PMLR, 2020, pp. 1–28.
[30] C. Litterer and T. Lyons, *High order recombination and an application to cubature on Wiener space*, Ann. Appl. Probab., 22 (2012), pp. 1301–1327.
[31] T. Lyons, *Rough paths, signatures and the modelling of functions on streams*, in International Congress of Mathematicians, Seoul, 2014.
[32] T. Lyons, M. Caruana, and T. Lévy, *Differential equations driven by rough paths*, Ecole d'été de Probabilités de Saint-Flour XXXIV, Lectures from the 34th Summer School on Probability Theory held in Saint-Flour, Springer, 2004, pp. 1–93.
[33] T. Lyons, S. Nejad, and I. Perez Arribas, *Numerical method for model-free pricing of exotic derivatives in discrete time using rough path signatures*, Appl. Math. Finance, 26 (2019), pp. 583–597.
[34] T. Lyons and N. Victoir, *An extension theorem to rough paths*, Ann. Inst. H. Poincaré Anal. Non Linéaire, 24 (2007), pp. 835–847.
[35] T. J. Lyons, *Differential equations driven by rough signals*, Rev. Mat. Iberoamericana, 14 (1998), pp. 215–310.
[36] P. Moore, T. Lyons, and J. Gallacher for the Alzheimer's Disease Neuroimaging Initiative, *Using path signatures to predict a diagnosis of Alzheimer's disease*, PloS ONE, 14 (2019), e0222212.
[37] J. H. Morrill, A. Kormilitzin, A. J. Nevado-Holgado, S. Swaminathan, S. D. Howison, and T. J. Lyons, *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 (2020), pp. e976–e981.
[38] A. D. Polyanin and V. E. Nazaikinskii, *Handbook of Linear Partial Differential Equations for Engineers and Scientists*, 2nd ed., Chapman and Hall/CRC, 2015.
[39] J. Reizenstein and B. Graham, *The iisignature Library: Efficient Calculation of Iterated-Integral Signatures and Log Signatures*, preprint, https://arxiv.org/abs/1802.08252, 2018.
[40] N. I. Sapankevych and R. Sankar, *Time series prediction using support vector machines: A survey*, IEEE Comput. Intell. Mag., 4 (2009), pp. 24–38.
[41] M. Schmidt, N. L. Roux, and F. R. Bach, *Convergence rates of inexact proximal-gradient methods for convex optimization*, in Advances in Neural Information Processing Systems, 2011, pp. 1458–1466.
[42] J. Shawe-Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis*, Cambridge University Press, 2004.
[43] A. Smola, A. Gretton, L. Song, and B. Schölkopf, *A Hilbert space embedding for distributions*, in International Conference on Algorithmic Learning Theory, Springer, 2007, pp. 13–31.
[44] R. Tavenard, J. Faouzi, G. Vandewiele, F. Divo, G. Androz, C. Holtz, M. Payne, R. Yurchak, M. Russwurm, K. Kolar, and E. Woods, *Tslearn, a machine learning toolkit for time series data*, J. Mach. Learn. Res., 21 (2020), pp. 1–6, http://jmlr.org/papers/v21/20-091.html.
[45] S. Tong and E. Chang, *Support vector machine active learning for image retrieval*, in Proceedings of the 9th ACM International Conference on Multimedia, ACM, 2001, pp. 107–118.
[46] S. Tong and D. Koller, *Support vector machine active learning with applications to text classification*, J. Mach. Learn. Res., 2 (2001), pp. 45–66.
[47] C. Toth and H. Oberhauser, *Bayesian learning from sequential data using Gaussian processes with signature covariances*, in Proceedings of the 37th International Conference on Machine Learning (ICML), 2020, pp. 9548–9560.
[48] B. Tzen and M. Raginsky, *Neural Stochastic Differential Equations: Deep Latent Gaussian Models in the Diffusion Limit*, preprint, https://arxiv.org/abs/1905.09883, 2019.
[49] V. Vapnik, *The support vector method of function estimation*, in Nonlinear Modeling, Springer, 1998, pp. 55–85.
[50] A. M. Wazwaz, *On the numerical solution for the Goursat problem*, Appl. Math. Comput., 59 (1993), pp. 89–95.
[51] W. Yang, T. Lyons, H. Ni, C. Schmid, and L. Jin, *Developing the Path Signature Methodology and Its Application to Landmark-Based Human Action Recognition*, preprint, https://arxiv.org/abs/1707.03993, 2017.