Chaos
An Interdisciplinary Journal of Nonlinear Science

RESEARCH ARTICLE | APRIL 15 2021

Kernel-based parameter estimation of dynamical systems with unknown observation functions

Ofir Lindenbaum ; Amir Sagiv ; Gal Mishne ; Ronen Talmon

Chaos 31, 043118 (2021)
https://doi.org/10.1063/5.0044529

AIP Publishing

Chaos ARTICLE scitation.org/journal/cha

Kernel-based parameter estimation of dynamical systems with unknown observation functions

Cite as: Chaos **31**, 043118 (2021); doi: 10.1063/5.0044529
Submitted: 17 January 2021 · Accepted: 29 March 2021 ·
Published Online: 15 April 2021

Ofir Lindenbaum,[1,a)] Amir Sagiv,[2,b)] Gal Mishne,[3,c)] and Ronen Talmon[4,d)]

AFFILIATIONS

[1]Program in Applied Mathematics, Yale University, 51 Prospect Street, New Haven, Connecticut 06511, USA
[2]Department of Applied Physics and Applied Mathematics, Columbia University, 500 West 120th Street, New York, New York 10027, USA
[3]Halicioglu Data Science Institute, UC San Diego 9500 Gilman Drive MS 0555 SDSC 215E, La Jolla, California 92093-0555, USA
[4]Faculty of Electrical Engineering, Technion—Israel Institute of Technology, Haifa 32000, Israel

[a)]ofir.lindenbaum@yale.edu
[b)]**Author to whom correspondence should be addressed:** as6011@columbia.edu
[c)]gmishne@ucsd.edu
[d)]ronen@ef.technion.ac.il

ABSTRACT

A low-dimensional dynamical system is observed in an experiment as a high-dimensional signal, for example, a video of a chaotic pendulums system. Assuming that we know the dynamical model up to some unknown parameters, can we estimate the underlying system's parameters by measuring its time-evolution only once? The key information for performing this estimation lies in the temporal inter-dependencies between the signal and the model. We propose a kernel-based score to compare these dependencies. Our score generalizes a maximum likelihood estimator for a linear model to a general nonlinear setting in an unknown feature space. We estimate the system's underlying parameters by maximizing the proposed score. We demonstrate the accuracy and efficiency of the method using two chaotic dynamical systems—the double pendulum and the Lorenz '63 model.

The purpose of many experimental designs is to measure a quantity of interest by observing a dynamical system and comparing the observations to a known model. This procedure can be difficult when the system is chaotic and can be even more challenging when the correspondence between the measurements and the model, the observation function, is unknown. For a single experiment, i.e., when the observed data is a single time series, learning the unknown observation function is not an option. Instead, we construct a kernel-based score in a way that is agnostic to the unknown observation function. We can derive a maximum likelihood (ML) estimator for identifying the observed dynamical system's parameters by maximizing that score. The intuition behind our approach is that even though the map between the coordinates/model and the observations is unknown, the dynamics of the data convey enough information on the dynamics of the model with the true parameter, thus facilitating an informed parameter estimation procedure. We propose two optimization schemes for maximizing our score. Finally, we demonstrate that our method can accurately estimate the governing parameters for two chaotic dynamical systems from complex and high-dimensional data.

I. INTRODUCTION

Consider a common situation in experimental sciences—an experiment is designed to measure a quantity of interest by observing a dynamical system and comparing the observations to a known model. But can this measurement be performed when the model is complex and chaotic? Furthermore, is this procedure possible when the correspondence between the measurement and the model, the *observation function*, is unknown?

For example, it is straightforward to estimate the gravitational free acceleration g by observing a pendulum; the angle $x(t)$ of a pendulum of length ℓ varies periodically according to the harmonic oscillator ordinary differential equation (ODE) $\ddot{x}(t) = -(g/\ell)x$. By

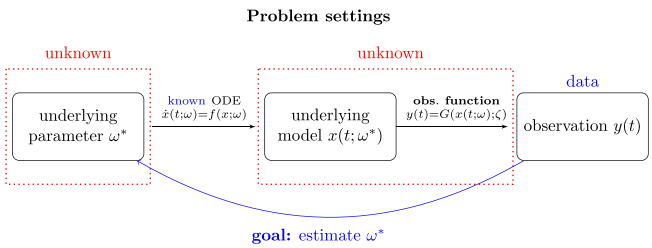

FIG. 1. The schematic settings of our problem. We are given the observations $y(t)$ and a mechanism to generate $x(t;\omega)$ for every $\omega \in \Omega$ (an ODE). What is the true underlying parameter ω^* driving $y(t)$?

solving the ODE, g may be estimated using $g = \nu^2\ell$, where the frequency ν can be directly observed from the pendulum's oscillations. But can such a measurement scheme be applied to the chaotic *double* pendulum, where no easily observable parameter like the frequency ω exists?

The double pendulum example illustrates a more general class of problems (see Fig. 1). In an experiment, an observed signal $y(t)$ is related to its governing model $x(t;\omega^*)$ by specific yet unknown parameters (or parameter vector) ω^* and an unknown observation function G, i.e.,

$$y(t) = G(x(t;\omega^*);\zeta),$$

where ζ is a noise source. The purpose of this study is to estimate the system's parameters ω^* among all possible parameters ω in a parameter space Ω, using the observation $y(t)$ and the general model $x(t;\omega)$. Critically, we note that even though the map $\omega \mapsto x(t;\omega)$ is known, the map $x \overset{G}{\mapsto} y$ is unknown to us. Therefore, we only know "half" of the forward map $\omega \mapsto y(t)$. Since the forward map is unknown, this problem does not fit into the usual notion of inverse problems.[3,61] Conversely, since we only observe a single experiment and do not have a lot of data, it is not straightforwardly amenable to standard machine learning methodology (see Sec. VI A for details).

We propose a kernel-based approach to estimate the system's parameters ω^*. We first study the case of a linear observation function G. A maximum likelihood estimation of ω^* then yields

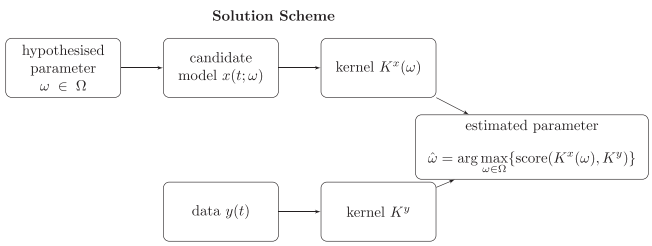

FIG. 2. The schematics of the proposed solution. For every $\omega \in \Omega$, a kernel $K^x(\omega)$ is computed. This kernel is compared to the observation kernel K^y, and the estimated $\hat{\omega}$ is chosen to maximize their similarity score. The hypothesized ω values are either predetermined (Algorithm 1) or dynamically determined using an optimization scheme (Algorithm 2).

a maximization problem for a normalized variant of the cross-covariance between the observations and the model. To carry this idea to the general nonlinear case, we "lift" both the observations and the model to an infinite-dimensional Hilbert space (feature space[36,55]). In the feature space, the two signals are again linearly dependent. By constructing kernels for the observations $y(t)$ and for the model $x(t;\omega)$, a covariance-like score in the feature space is computed (see (15)) and maximized to estimate the system's parameters. By applying our method (Algorithms 1 and 2) to two examples of chaotic dynamical systems—the double pendulum and the Lorenz system—we demonstrate empirically that maximizing the kernel-based score indeed yields an accurate estimate for ω^*.

The application of the so-called kernel trick to generalize the linear notion of covariance has been used for various statistical tasks such as kernel principle component analysis (PCA), kernel canonical-correlation analysis (CCA), and the Hilbert–Schmidt Independence criteria.[6,9,28,30–32,34,47,49,54] Kernels were also used in this context of kernel density estimators (KDEs)[63] or for extracting latent variables from multi-modal observations as in Refs. 44, 42, 53, and 72. While resembling to some nonlinear kernel statistical problems on the one hand, and to some machine learning and model discovery problems on the other hand,[4,8,12,15,18,23,25,69] we note that the problem of parameter learning under an unknown observation function is stated here, to the best of our knowledge, for the first time. Consequently, our kernel-based score does not seem to appear in the kernel methods literature (see discussion in Sec. III).

The remainder of this paper is organized as follows. Section II presents the problem in formal terms. In Sec. III, we derive our method initially for the linear case, and then to the general nonlinear case. Section IV presents the main algorithms of this paper—the search-based Algorithm 1 and the optimization-based Algorithm 2. The applications of our approach to the double pendulum and to the Lorenz system are presented in Sec. V. Finally, we discuss potential applications of the method and its relationship to previous studies on model discovery, inverse problems, and kernel methods in Sec. VI.

II. PROBLEM FORMULATION

Consider a parametric family of autonomous ordinary differential equations (ODEs),

$$\begin{cases} \dot{x}(t;\omega) = f(x;\omega), & \omega \in \Omega \subseteq \mathbb{R}^m, \\ x(0;\omega) = x_0(\omega) \in \mathbb{R}^d, \end{cases} \tag{1}$$

where $\Omega \subseteq \mathbb{R}^m$ is a convex set of possible parameters and f is sufficiently smooth such solutions are unique and exist globally. The dynamics $x(t;\omega)$ are, therefore, completely determined by a fixed vector of parameters $\omega^* \in \Omega$. Assume that ω^* is unknown and that we do not observe $x(t;\omega^*)$, but only a measurement $y(t) \in \mathbb{R}^D$ for some dimension D. This observation can be viewed as a noisy lifting of $x(t;\omega^*)$ from the latent space $\mathbb{R}^d$ to the ambient observation space $\mathbb{R}^D$ by an unknown and possibly noisy map, i.e.,

$$y(t) = G(x(t;\omega^*);\zeta), \quad G : \mathbb{R}^d \times \mathcal{Z} \to \mathbb{R}^D, \tag{2}$$

where $\mathcal{Z}$ is some manifold in which $\zeta(t)$ is a stationary random process with δ auto-correlation and the observation function $G(\cdot,0)$ guarantees identifiability of ω (see Ref. 64 and Sec. IV C for details).[74]

For example, if $x(t;\omega)$ describes the trajectory of a ballistic projectile in $\mathbb{R}^3$, its video will embed this trajectory in $\mathbb{R}^D$, where D is the number of pixels in each video frame. As a practical matter, we will further assume that $y(t)$ is measured in discrete times $\{t_j = (j-1)\Delta t\}_{j=1}^N$ for some $\Delta t > 0$. The main problem of this paper can now be formally stated:

Problem. *Given*

(1) *A single observed time series* $\{y(t_j)\}_{j=1}^N$, *defined by (2) with unknown* G *and* ω^*, *and*
(2) *A solution* $x(t;\omega)$ *to the ODE (1) for all* $t \geq 0$ *and* $\omega \in \Omega$,

find the vector of underlying parameters ω^*.

Remark 1. *Uncertainty in the observations may stem from several sources—modeling misspecification, numerical errors, measurement noise, nuisance variables in the experiment, etc. The introduction of randomness in (2) is a modeling decision aimed to capture all of these uncertainty sources.*

We start by considering the simplified linear variant of (2), in which we can derive a maximum likelihood estimator of ω^*. In the general nonlinear case, we use the kernel trick to map the phase space coordinates $x(t;\omega)$ and the observations $y(t)$ into a Hilbert space (feature space) where the linear approach can be employed again.

III. DERIVATION

A. The linear case—A maximum likelihood approach

It is instructive to first consider (2), where G is linear in x and additive with respect to a Gaussian noise term, i.e.,

$$\bar{y}(t_j) = A\bar{x}(t_j;\omega^*) + \zeta_j, \quad j = 1,\ldots,N, \tag{3}$$

where $\bar{x}(t;\omega) = x(t;\omega) - \mathbb{E}_\tau x(\tau;\omega)$ and $\bar{y}(t) = y(t) - \mathbb{E}_\tau y(\tau)$ are centered, $A \in M_{D,d}(\mathbb{R})$, and for each $1 \leq j \leq N$ the term $\zeta_j \in \mathbb{R}^D$ is drawn iid from $\mathcal{N}(0,\sigma^2 \mathrm{I})$ for some $\sigma > 0$.

Remark 2. *We center the observations* y *and model coordinates* x *since the choice of origin in either* $\mathbb{R}^D$ *and* $\mathbb{R}^d$ *is arbitrary from a modeling/physics perspective (see more on the role of such invariances in Sec. IV C). In practice, the time-averages should be replaced by their empirical estimates, e.g.,* $\mathbb{E}_\tau y(\tau) \approx N^{-1}\sum_j y(t_j)$.

To estimate ω^* from the observations $\{y(t_n)\}_{n=1}^N$, we use a maximum likelihood (ML) estimator.[1] The ML estimator is defined as

$$\hat{\omega}_{\mathrm{ml}} = \arg \max_{\substack{\omega\in\Omega \\ A\in M_{D,d}(\mathbb{R})}} \mathrm{Prob}_{\zeta_1,\ldots,\zeta_N}(\bar{y}_1,\ldots,\bar{y}_N|\omega,A).$$

Since the log function is monotonic increasing, we can replace the likelihood function by log-likelihood to exploit the independence of the normal ζ_j's to get

$$\begin{aligned}\hat{\omega}_{\mathrm{ml}} &= \arg \max_{\substack{\omega\in\Omega \\ A\in M_{D,d}(\mathbb{R})}} \log \mathrm{Prob}_{\zeta_1,\ldots,\zeta_N}(\bar{y}_1,\ldots,\bar{y}_N|\omega,A) \\ &= \arg \max_{\substack{\omega\in\Omega \\ A\in M_{D,d}(\mathbb{R})}} -\frac{\sum_{n=1}^N \|\bar{y}(t_n) - A\bar{x}(t_n,\omega)\|_2^2}{2\sigma^2} - \frac{DN}{2}\log(2\pi\sigma^2).\end{aligned} \tag{4}$$

Since D, N, σ, and the observations $y(t_j)$ are independent of ω and A, we can simplify the objective function on the right-hand-side as follows:

$$\begin{aligned}\hat{\omega}_{\mathrm{ml}} &= \arg \min_{\substack{\omega\in\Omega \\ A\in M_{D,d}(\mathbb{R})}} \sum_{n=1}^N \|\bar{y}(t_n) - A\bar{x}(t_n,\omega)\|_2^2 \\ &= \arg \min_{\substack{\omega\in\Omega \\ A\in M_{D,d}(\mathbb{R})}} \|AX(\omega)\|_F^2 - 2\langle AX(\omega), Y\rangle + \|Y\|_F^2 \\ &= \arg \min_{\substack{\omega\in\Omega \\ A\in M_{D,d}(\mathbb{R})}} \|AX(\omega)\|_F^2 - 2\langle AX(\omega), Y\rangle,\end{aligned} \tag{5}$$

where $X(\omega)$ and Y are matrices whose jth columns are $\bar{x}(t_j;\omega)$ and $\bar{y}(t_j)$, respectively, $\langle B, C\rangle = \sum_{i,j} B_{i,j}C_{i,j}$ is the Frobenius inner product on $M_{D,d}(\mathbb{R})$ and $\|B\|_F = \langle B, B\rangle^{1/2}$ is the Frobenius norm. Next, we show that one can also normalize (5) by $\|AX\|_F$. To see that, fix $\omega \in \Omega$ and $O \in M_{D,d}(\mathbb{R})$ such that $A = \lambda O$, $\|OX\|_F = 1$, and $\lambda \in \mathbb{R}$. Then, by direct differentiation in λ,

$$\|AX(\omega)\|_F^2 - 2\langle AX(\omega), Y\rangle = \lambda^2 - 2\lambda\langle OX(\omega), Y\rangle$$

is minimized when $\lambda = \langle OX, Y\rangle$. Hence,

$$\hat{\omega}_{\mathrm{ml}} = \arg \min_{\substack{\omega\in\Omega \\ \|OX\|_F=1}} \langle OX(\omega), Y\rangle^2 - 2\langle\langle OX(\omega), Y\rangle OX(\omega), Y\rangle.$$

By setting $O = A\|AX\|_F^{-1}$, we get that the maximum likelihood estimator is

$$\hat{\omega}_{ml} = \arg \max_{\substack{\omega\in\Omega \\ A\in M_{D,d}(\mathbb{R})}} \frac{\langle AX(\omega), Y\rangle_{F^2}}{\|AX(\omega)\|_F^2 \cdot \|Y\|_F^2}, \tag{6}$$

where, since Y is a constant matrix, we divide by $\|Y\|_F^2$ so that the argument on the right-hand side is always ≤ 1.

B. Maximum likelihood in the nonlinear settings—A kernel approach

In the general model (2), the observations $y(t)$ do not depend linearly on the model coordinates $x(t)$ as in (3). Rather, the two depend nonlinearly via G [see (2)]. If we knew G, the linear maximum likelihood approach (6) could be applied to y and $G(x(\cdot;\omega^*);\zeta)$ in $\mathbb{R}^D$. Even though we do not know G, the relation (2) implies that x and y are linearly dependent under nonlinear transformations $\psi : \mathbb{R}^d \to \mathbb{R}^D$ and $\phi : \mathbb{R}^D \to \mathbb{R}^D$ (feature maps), respectively, i.e.,

$$\phi(y(t)) = \psi(x(t;\omega^*);\zeta). \tag{7}$$

In what follows, we assume that the noise term ζ is again Gaussian and additive, i.e., $\phi(y(t)) = \psi(x(t;\omega^*)) + \zeta$ where $\zeta \sim \mathcal{N}(0,\sigma^2\mathrm{I})$.

In this case, one possible set of feature maps is simply $\phi(y) = y$ and $\psi(x) = G(x)$. However, we show in the numerical experiments that our algorithm can estimate ω^* even for non-Gaussian and non-additive noise sources (see Sec. V C).

Remark 3. *Seemingly, the nonlinear model (7) is more restrictive than its linear counterpart (3), absent of the freedom to choose the linear transformation A. Given the maps ϕ and ψ, however, A is "absorbed" into the definitions of ϕ and ψ.*

Using the same maximum likelihood argument of Sec. III A, the nonlinear model (7) yields the following estimator of ω^* [compare with (6)]:

$$\hat{\omega} = \arg\max_{\omega\in\Omega} \frac{\langle \bar{\Psi}(\omega), \bar{\Phi}\rangle_F^2}{\|\bar{\Psi}(\omega)\|_F^2 \cdot \|\bar{\Phi}\|_F^2}, \tag{8a}$$

where

$$\begin{aligned} \bar{\Psi}_{\cdot,j}(\omega) &:= \psi(x(t_j;\omega)) - \frac{1}{N}\sum_{i=1}^{N}\psi(x(t_i;\omega)), \\ \bar{\Phi}_{\cdot,j} &:= \phi(y(t_j)) - \frac{1}{N}\sum_{i=1}^{N}\phi(y(t_i)). \end{aligned} \tag{8b}$$

We further note that

$$\|\bar{\Phi}\|_F^2 = \mathrm{Tr}(K^y H), \quad \|\bar{\Psi}(\omega)\|_F^2 = \mathrm{Tr}(K^x(\omega)H), \quad H_{ij} := \delta_{ij} - \frac{1}{N}, \tag{9}$$

where the Gram matrix K^y is defined by

$$K^y_{ij} := k^y(y(t_i), y(t_j)), \quad k^y(y, y') := \langle \phi(y), \phi(y')\rangle \tag{10}$$

for every $1 \le i, j \le N$, and K^x and k^x are defined analogously to (10) (see the details in Appendix A).

We can, therefore, rewrite (8a) as

$$\hat{\omega} = \arg\max_{\omega\in\Omega} \frac{\langle \bar{\Psi}(\omega), \bar{\Phi}\rangle_F^2}{\|K^x(\omega)H\|_F^2 \|K^y(\omega)H\|_F^2}. \tag{11}$$

In practice, we do not know what the maps ϕ and ψ are and, consequently, cannot compute their corresponding kernels and Gram matrices (10). However, the kernels k^x and k^y are both Mercer kernels, i.e., their Gram matrices (10) are always positive semi-definite,[52] they define an inner product on some (infinite-dimensional) reproducing kernel Hilbert Space $\mathcal{H}$.[75] This observation suggests that one should apply a well known heuristic known as the "kernel trick," where rather than estimating the feature map ψ from an infinite-dimensional function space, one chooses the kernels k^x and k^y. The choice of kernels reflects our understanding and prior knowledge of the data and what makes two samples $x(t_i)$ and $x(t_j)$ similar (and similarly for y). The kernel trick here seems necessary, since we do not know the observation function G. If we were to reconstruct G, then we could have used $\psi = G$, $\phi = \mathrm{Id}$, and $\mathcal{H} = \mathbb{R}^D$ (see, e.g., Ref. 11). In our setting, however, recovering G might be nearly impossible, since we have only a single output time series $y(t)$ and do not know the parameter vector ω^*.

We revisit (11) with the kernel trick in mind. The numerator $\langle \bar{\Psi}(\omega), \bar{\Phi}\rangle_F^2$ is equal to $\mathrm{Tr}^2(C_{xy}(\omega))$, where $C_{xy}(\omega) = \bar{\Psi}(\omega)\bar{\Phi}^{\mathrm{T}}$ is the cross-covariance operator. This operator norm cannot be computed, since we only choose the kernels k^x and k^y and not the feature maps ϕ and ψ. Nonetheless, we can use the kernels k^x and k^y to estimate $\mathrm{Tr}(C_{xy}^{\mathrm{T}} C_{xy}) = \|C_{xy}(\omega)\|_F^2$ as a surrogate to $\mathrm{Tr}^2(C_{xy})$. The norm $\|C_{xy}(\omega)\|_F^2$ is known as the Hilbert–Schmidt independence criterion (HSIC), due to Gretton *et al.*[32,33] and can be estimated empirically by $\mathrm{Tr}(K^x(\omega)HK^yH)$. We, therefore, use the HSIC(ω) to define the following realizable proxy of objective (8):

$$\hat{\omega} = \arg\max_{\omega\in\Omega} \frac{\mathrm{HSIC}(\omega)}{\|\bar{\Psi}(\omega)\|_F^2 \cdot \|\bar{\Phi}\|_F^2} = \arg\max_{\omega\in\Omega} \frac{\mathrm{Tr}(K^x(\omega)HK^yH)}{\|K^x(\omega)H\|_F^2 \|K^y(\omega)H\|_F^2}. \tag{12}$$

The HSIC is an indicator for the dependence of ϕ and ψ (or x and y) as random variables. As $y(t)$ is determined by $x(t;\omega^*)$ up to a noise term, we expect that HSIC(ω^*) would express a high statistical dependency. However, HSIC is also maximized as the covariance-matrix of x (or of Ψ in the nonlinear settings) is maximized, regardless of C_{xy}. We, therefore, normalize the HSIC estimator by the norm of the standard deviation matrix $\| \left(\bar{\Psi}(\omega)\bar{\Psi}(\omega)^{\mathrm{T}}\right)^{1/2} \|_F^2$. This latter matrix is estimated in the nonlinear Hilbert space settings by $\|K^x(\omega)\|_F^2$, which leads to the right-hand side of (12). The empirical estimator of HSIC in the nominator on the right side is known to have a bias of $\mathcal{O}(1/N)$.[32]

Different perspective: The estimator ω_{ml} in (12) can also be viewed in terms of kernel density estimators (KDEs). In Ref. 63, the authors study non-rigid shape correspondence in three-dimensional objects. Given N points x_i and y_i on two respective deformations of the same shape, the goal is to find a permutation π on $1, \ldots N$ such that each y_i corresponds to $x_{\pi(i)}$ in the deformed shape. Letting $K^x(\pi)$ be the Gram matrix of the permutated x_i's, the term $\mathrm{Tr}(K^x(\pi)K^y)$ can be understood as the KDE estimator of the joint probability of the points under the permutation π. Indeed, in these settings, the maximum likelihood estimator of π among all permutations maximizes $\mathrm{Tr}(K^x(\pi)K^y)$, much as in Sec. III of this paper. Note that since the parameter estimated in Ref. 63 is a permutation, $\|K^x(\pi)\|_F^2$ is constant (independent of π) and, therefore, the normalization is not needed.

In Ref. 47, the authors use the kernel trick to solve the nonparametric CCA problem, i.e., identify nonlinear mappings ψ and ϕ that maximize the correlation between $\psi(x)$ and $\phi(y)$ [as in the numerator of (8a)]. They show that the solution can be expressed using the singular values of the joint probability density of X and Y, and use kernels to estimate this joint density in a nonparametric fashion. In the context of our problem, this solution can be used after ω^* is estimated for comparing the trajectories in a low-dimensional representation.

IV. PROPOSED METHOD

A. Exhaustive search approach

In light of the analysis of Sec. III, our first proposed method, Algorithm 1, is a straightforward numerical application of (12). Assume for simplicity that Ω is a box in $\mathbb{R}^m$, i.e., $\Omega = \prod_{i=1,\ldots,m}[a_i, b_i]$, where $b_i > a_i$ for all $i = 1, \ldots, m$. We define the Gram matrices for x and y by

$$K^y_{i,j} := \exp\left(-\frac{\|y(t_i) - y(t_j)\|_2^2}{\varepsilon_y}\right), \quad \varepsilon_y > 0. \tag{13}$$

Algorithm 1. Kernel search-based method for estimating ω^*

Given $\{y(t_n)\}_{n=1,\dots,N}$.
1. Compute the y kernel (13).
2. Choose $\Omega_{\text{search}} \subset \Omega$ to be a finite Cartesian grid in $\Omega \subseteq \mathbb{R}^n$.
3. **for** each $\omega \in \Omega_{\text{search}}$ **do**
4. Solve (1) for $x(t;\omega)$.
5. Compute the x kernel (14).
6. Compute the score $s(\omega)$ [see (15)].
7. **end for**
8. **return** $\hat{\omega} = \arg\max_{\omega\in\Omega_{\text{search}}} s(\omega)$.

$$K^x_{i,j}(\omega) := \exp\left(-\frac{\|x(t_i;\omega) - x(t_j;\omega)\|_2^2}{\varepsilon_x}\right), \quad \varepsilon_x > 0. \tag{14}$$

Then, following (12), we propose the score

$$s(\omega) = \frac{\mathrm{Tr}(K^x(\omega)HK^yH)}{\|K^x(\omega)H\|_F \cdot \|K^yH\|_F}, \tag{15}$$

where $H_{ij} = \delta_{ij} - N^{-1}$, in which Algorithm 1 maximizes on a predetermined set of grid points in $\Omega_{\text{search}} \subset \Omega$.

As noted in Sec. III B, since we do not know the feature maps ϕ and ψ, one needs to choose the kernel k^x and k^y (or their Gram matrices K^x and K^y) to use (12). There are many possible choices of kernels, which reflect different notions of affinities between samples of $x(t)$ (and of y), many of which might have worked well for estimating ω^* (see, e.g., Ref. 36). In this work, we choose the widely popular Gaussian kernel $k^x(x(t_i), x(t_j)) = \exp(-\|x(t_i) - x(t_j)\|^2/\varepsilon_x)$ (and, respectively, for y), for two main reasons: first, the Gaussian kernel is translation invariant, i.e., $k^x(x, x') = k(x - x')$, which ensures we capture only relative changes in the data, and not absolute values. Second, the exponential decay of the Gaussian kernel attenuates the effect of large distances. For the model coordinates $x(t)$, this makes intuitive sense because the governing ODE (1) is local. For the observation coordinates, attenuating large distances counteracts the spuriously large pairwise distances that tend to appear in $\ell^2(\mathbb{R}^D)$ for $D \gg 1$ (see, e.g., Refs. 19 and 35). Finally, the Gaussian kernel has expressivity properties that ensure its ability to capture polynomial-order nonlinear dependencies between different times, but it is by no means the only possible choice. For detailed analysis and alternative kernel choices (see Ref. 59, and the references therein). A key takeaway from this work is that even though other tailored kernels can be designed for better performance, the standard choice of the Gaussian kernel yields good results in our numerical experiments.

The effectiveness of Gaussian kernels strongly relies on proper tuning of the kernels' bandwidths ε_x and ε_y. These parameters directly affect the feature maps induced by the kernels. At one extreme, setting ε_x or ε_y too large would result in kernels that approach the all-ones matrices, i.e., where all samples are equally affine. At the other extreme, as $\varepsilon_x \to 0$ (respectively, ε_y), the kernel K approaches the identity matrix, i.e., all affinities between different samples are neglected. Here, we use a max–min measure suggested in Ref. 39, where the scale is set to

$$\varepsilon_y = \max_j[\min_{i,i\neq j}(||y(t_i) - y(t_j)||^2)], \quad i,j = 1,\dots,N, \tag{16}$$

and analogously for ε_x. The max–min approach guarantees that for each data point, K expresses a non-negligible affinity with at least one other point. Moreover, this scheme generates kernels K^x and K^y that are invariant to scaling in the ambient observation space $\mathbb{R}^D$. Several other methods have been proposed for tuning ε and can certainly be used (see, for example, Refs. 39, 71, 43, and 58). The third tunable parameter in Algorithm 1 is the grid size $|\Omega_{\text{search}}|$. The choice of a predetermined grid Ω_{search} affects Algorithm 1 in two ways:

(1) **Accuracy:** For any estimator $\hat{\omega}$ of ω^*, define the estimation error

$$\mathrm{Err}\ \omega_j = \frac{\|\hat{\omega}_j - \omega_j^*\|_2}{\omega_j^*}, \quad j = 1,\dots,m. \tag{17}$$

Generally in Algorithm 1, the grid does not necessarily include ω^*, i.e., $\omega^* \notin \Omega_{\text{search}}$, and so $\hat{\omega} \neq \omega^*$. Therefore, even in the best case scenario where Algorithm 1 returns the closest grid point to ω^*, the error is typically bounded from below in terms of $\Delta\omega$. This error estimation applies also to the case of a single parameter, as the average estimation error (17) over many experiments scales like $\Delta\omega$, where

$$\Delta\omega := \min\{|\omega_i - \omega_j| \text{ s.t. } \omega_i, \omega_j \in \Omega_{\text{search}},\ i \neq j\}$$

is the spacing of the grid. To see that, suppose for simplicity that $\Omega = [0, \Delta\omega]$ and that ω^* is drawn uniformly at random from Ω, i.e., the probability density function of $\omega^* = y$ is $p(y) = \Delta\omega^{-1}$. Then, by partitioning Ω to two halves, one closer to the grid point $\omega = 0$ and the other to the grid point $\omega = \Delta\omega$, we have that

$$\mathbb{E}_{\omega^*}\mathrm{dist}(\omega^*, \text{grid}) = \frac{1}{\Delta\omega}\left(\int_0^{\Delta\omega/2} \omega^*\, d\omega^* + \int_{\Delta\omega/2}^{\Delta\omega} (\Delta\omega - \omega^*)\, d\omega^*\right) = \frac{\Delta\omega}{4}.$$

A similar estimate holds for any dimension $m \geq 1$ of Ω.

(2) **Efficiency:** In the multi-dimensional case $\Omega \subseteq \mathbb{R}^m$, fine grids are computationally prohibitive since their size scales exponentially with m. Large grids are especially an issue when either (i) solving the underlying dynamical system (1) is computationally expensive, or (ii) the length of the time series N is large. In the latter case, the computation of kernel (14), which requires evaluating all pairwise distances, requires $\mathcal{O}(N^2)$ operations. The cost of the kernel computation can be reduced using methods such as k-sparse graph,[66] which enjoys a reduced complexity of $\mathcal{O}(N\log N + Nk)$, where k is the number of nearest neighbors used for building the graph.

B. Optimization approach

To overcome both of the accuracy and the efficiency problems of Algorithm 1, we would like to replace the exhaustive grid search

Algorithm 2. Kernel optimization-based method for estimating ω^*

Given $\{y(t_n)\}_{n=1,\ldots,N}$.
1. Compute the kernel for y (13).
2. Choose $\Omega_{\text{init}} \subset \Omega$ at random.
3. **for** each $\omega_j \in \Omega_{\text{init}}$ **do**
4. Solve the optimization problem (18) to find $\hat{\omega}_j$ using interior point optimization initialized at ω_j, where the kernel $K^x(\omega)$ is given by (14) and $x(t;\omega)$ are the solutions of (1).
5. **end for**
6. **return** $\hat{\omega} = \arg\max_j s(\hat{\omega}_j)$.

with an optimization scheme to solve the following problem:

$$\begin{aligned} &\text{maximizes } (\omega) = \frac{\mathrm{Tr}(\mathrm{K}^x(\omega)\mathrm{H}\mathrm{K}^y\mathrm{H})}{\|K^x H(\omega)\|_F \cdot \|K^y H\|_F}, \\ &\text{over} \omega \in \mathbb{R}^m, \\ &\text{subject to } R(\omega) \leq 0, \end{aligned} \tag{18}$$

where R can be chosen to be any function such that $R(\omega) \leq 0$ if and only if $\omega \in \Omega$.[76] Critically, optimization problem (18) is in general non-convex. This is an *inherent feature of our problem* and should not depend on the specific solution strategy. Indeed, let $\tilde{s}(\omega)$ be any convex cost function for which $\omega^* = \arg\max \tilde{s}(\omega)$. Since $\tilde{s}$ depends on ω indirectly through $f(x;\omega)$ [see (1)] and since there is no unique way to express the dependence of f on its parameters, one can find an equivalent parameterization of the ODE (1) such that the resulting $\tilde{s}(\omega)$ is no longer convex. Therefore, the standard form of our ODE of interest need not result in a convex parameterization of $\tilde{s}(\omega)$. We note that this non-convexity property is already true for the linear model (6).

To solve the constrained, nonlinear and non-convex problem (18), our proposed method, Algorithm 2, has "two layers" of optimization. At the heart of Algorithm 2, we use the interior point algorithm (IPA), a solver for nonlinear constrained optimization problems,[16,17] as the optimization scheme in Step 4 of Algorithm 2.[77] The IPA method first takes a *Newton step* by attempting to solve a linear approximation of the problem (18), then, a gradient step is performed using a trust region.[65] Since the problem is not convex, we repeatedly initialize the IPA method at random points $\Omega_{\text{init}} \subset \Omega$, and then chooses the optimal result over all iterations. In our experience, the multiple initialization mechanism improves our chances of finding the global maximum (see experimental results in Sec. V).

Since the IPA routine in Algorithm 2 dynamically samples $\omega \in \Omega$ values, the overall number of samples does not scale exponentially with the dimension, and the accuracy is not limited by the grid spacing. Algorithm 2 is, therefore, computationally cheaper than Algorithm 1. The key parameter in comparing the two is the number of times the ODE (1) is solved and K^x (14) is computed. In Algorithm 1, this is exactly the grid size $|\Omega_{\text{search}}|$. In Algorithm 2, the number of evaluations of $x(t;\omega)$ is the number of optimization initializations $|\Omega_{\text{init}}|$ multiplied by the number of iterations in each optimization process. In the examples considered in this study, $|\Omega_{\text{init}}|$ was kept orders of magnitude smaller than the $|\Omega_{\text{search}}|$ without much loss of accuracy, and the number of iterations in each optimization process is usually below 20 (see, e.g., Fig. 6). Algorithm 2, therefore, suggests an avenue to solve (18) efficiently and accurately even as the dimension m of the parameter space Ω increases.

Remark 4. *Another practical implementation aspect of Algorithms 1 and 2 is the numerical solution method of the ODE (1). Throughout this paper, we used the standard fourth-order Runge–Kutta method.*[37] *Other numerical methods for ODEs can be used (see Ref. 38 for a more thorough discussion).*

C. Degeneracies and identifiability

When is the problem of estimating ω^* unsolvable? If $x(t;\omega) = x(t;\omega^*)$ for some $\omega \neq \omega^*$, these two parameters are indistinguishable in terms of the resulting dynamics. Moreover, if $G(x(t;\omega)) = G(x(t;\omega^*))$, then the experiment/observation of the dynamical system cannot distinguish between the parameters. These are extreme cases for obstructions of identifiability and observability, topics which have been studied in both the statistics and control literature (see, e.g., Refs. 48 and 64, and the references therein). System (1) combined with the observation function (2) is said to be unidentifiable if the problem of estimating ω^* is not solvable, regardless of the method of solution. In loose terms, such G then corresponds to a flawed experiment which is not designed to estimate ω.

Even for an identifiable and observable system, the fact that the observation function G is unknown limits our use of standard inference techniques. In what follows, we wish to discuss the limitation of our method even for identifiable systems: suppose that neither x nor G are degenerate (as functions of ω and x, respectively), but that the Gaussian kernels in (13) and (14) are degenerate. To explore the effect of these degeneracies, we will consider the case where G is an $\ell^2(\mathbb{R}^d \to \mathbb{R}^D)$ isometry and noiseless, i.e., $\|G(x_1) - G(x_2)\|_2 = \|x_1 - x_2\|_2$ for any $x_1, x_2 \in \mathbb{R}^d$. This discussion highlights some of the considerations that go into designing K^x and K^y.

Lemma 1. *Let $y(t) = G(t)$ where G is an $\ell^2(\mathbb{R}^d \to \mathbb{R}^D)$ isometry. Then,*

$$s(\omega^*) = 1 = \max_{\omega \in \Omega} s(\omega).$$

The other $\omega \in \Omega$ values where $s(\omega) = 1$ are precisely those for which $x(\omega) = Tx(\omega^*)$ for some $\ell^2(\mathbb{R})$ isometry T.

See the proof in Appendix B. Intuitively, Lemma 1 implies that while $s(\omega)$ is not uniquely maximized in this case, it is "reasonably degenerate," in the sense that its only other maximizers [when $G(x) = x$] are $\omega \in \Omega$ for which the trajectory $x(t;\omega)$ is isometric to $x(t;\omega^*)$. For example, if x is a scalar, it would mean that such degeneracies are only translations and reflections.

This is a fundamental consideration in the kernel design—if $\omega, \omega^* \in \Omega$ manifest in isometric observations/trajectories, then one cannot distinguish between them using the observations. The assumption that underlies our choice of the ℓ^2 norm in kernel (14) now becomes apparent—it expresses the kind of degeneracies we wish to allow. Indeed, different choices of norms would yield different equivalence classes of parameters in Ω.

Finally, we make a crucial note regarding our choice of the Gaussian kernel. In the proof of Lemma 1 we show that, in ideal

settings, $s(\omega)$ is maximized only when $\Delta = K^x(\omega) - K^x(\omega^*) = 0$. However, since kernel (14) is *exponentially decaying* with $\|x_i - x_j\|_2^2$, the entries Δ_{ij} are practically null for most far-away indices (times) i and j. Hence, even local in time degeneracies is sufficient to cause estimation error. As noted above, the choice of ε_x and ε_y determines how "local" the respective kernels are.

V. EXPERIMENTAL RESULTS

To test Algorithms 1 and 2, we apply them to two classical chaotic dynamical systems—the double pendulum and the Lorenz system.

A. Double pendulum

The double pendulum consists of two pendulums, one attached to the end of the other. The Lagrangian of this system is given by[40]

$$L = \frac{1}{2}(m_1 + m_2)\, l_1^2 \dot{\theta}_1^2 + \frac{1}{2} m_2 l_2^2 \dot{\theta}_2^2 + m_2 l_1 l_2 \dot{\theta}_1 \dot{\theta}_2 \cos(\theta_1 - \theta_2),$$

where l_j, m_j, and $\theta_j(t)$ are the length, mass, and clock-wise angle from the negative y direction of the jth pendulum, for $j = 1, 2$. From this Lagrangian, a fourth-order system of Euler–Lagrange ODEs for $(\theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2)$ can be derived (see, e.g., Ref. 56 and Appendix C).

The parameters of the system are $\omega = (m_2, l_1, l_2)$, where we set $m_1 = 1$ since the motion of the double pendulum depends only on the ratio m_2/m_1 (see Appendix C). We describe the dynamics of the double pendulum using the Cartesian coordinates of the two bobs $x = (x_1, y_1, x_2, y_2)$ (with a slight abuse of notations). The dynamics of the pendulum are observed through a synthetic video with a frame rate of $\Delta t = 0.01$. Each frame embeds the model in a high-dimensional space $y(t_i) \in \mathbb{R}^D$, where $D = 171 \times 217 = 37\,107$ is the number of pixels in each frame (see Fig. 3).

To evaluate the proposed approach, we generate 20 instances of a double pendulum with the parameters m_2, l_1, and l_2 each drawn iid from a uniform distribution in the interval $[1, 10]$ and with initial conditions randomly drawn independently from $\mathcal{N}(0, 0.5)$, where $N = 1000$. We then generate from each instance a movie of total run time $T = N \cdot \Delta t = 10$. First, we apply Algorithm 1 to each movie and search over 20 values of each parameter (where the initial conditions are known), i.e., a search-grid of size $|\Omega_{search}| = 20^3 = 8000$ and $\Delta\omega = 0.45$. In Fig. 4, we see that the estimated parameters yield pendulum trajectories nearly indistinguishable from the true ones (Fig. 4). Overall, the median normalized parameter estimation error (17) of Algorithm 1 is 7%, 8%, and 5% for m_2, l_1, and l_2, respectively. We repeat the same experiment for the optimization-based Algorithm 2 with $|\Omega_{\text{init}}| = 1000$, where the overall median normalized errors are 3.6%, 2.3%, and 2.4% for l_1, l_2, and m_2, respectively (see box plots in Fig. 5). In this experiment, as well as in the Lorenz system (Sec. V C), we assume that the initial conditions are known. This is a realistic assumption if the experiment is controlled by the observer, or if the initial conditions can be estimated independently. If the initial conditions are not known, one option is to estimate them as additional parameters using Algorithm 1 or 2.

To provide baselines for the proposed algorithms, we use two linear variants of Algorithm 1. Both variants are based on the same grid search procedure as in Algorithm 1 but using a different score. The score for the first linear estimator (Linear 1) is defined by

$$s^{\text{lin}_1}(\omega) = \frac{\|\bar{X}\bar{Y}^T\|_F}{\|\bar{X}\|_F \|\bar{Y}\|_F}, \tag{19}$$

where $\bar{X}$ and $\bar{Y}$ are the centered versions of X and Y. For the second linear estimator, we use Gram matrices instead of Gaussian kernels for k^y and k^x, thus the score is defined by

$$s^{\text{lin}_2}(\omega) = \frac{\text{Tr}(\text{G}^x(\omega)\text{HG}^y\text{H})}{\|G^x H(\omega)\|_F \cdot \|G^y H\|_F}, \tag{20}$$

where $G^x = X^T X$ and $G^y = Y^T Y$. As evident from the box plots of Linear 1 and 2 (Fig. 5), the performance of the linear variants of Algorithm 1 are inferior compared with their kernel-based counterparts. Specifically, the overall median error of Linear 1 is higher than Algorithm 1 by a factor of 6.5. For Linear 2, this factor is 9.7.

Next, to demonstrate the efficacy of the optimization-based Algorithm 2, we record the values attained by the IPA method throughout its iterations. As can be seen in Fig. 6, most of the IPA runs converge after 10 iterations, and all of them converge in

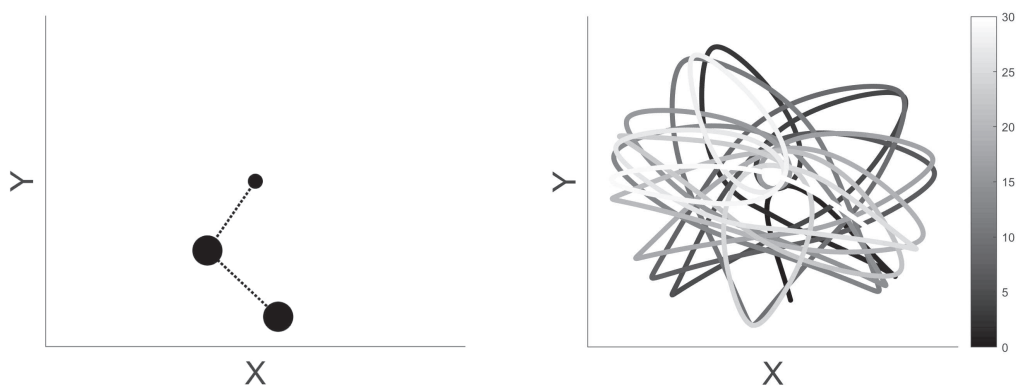

FIG. 3. Left: A frame of an artificial double pendulum video. Right: The chaotic trajectory $(x_2(t), y_2(t))$ of the bottom bob.

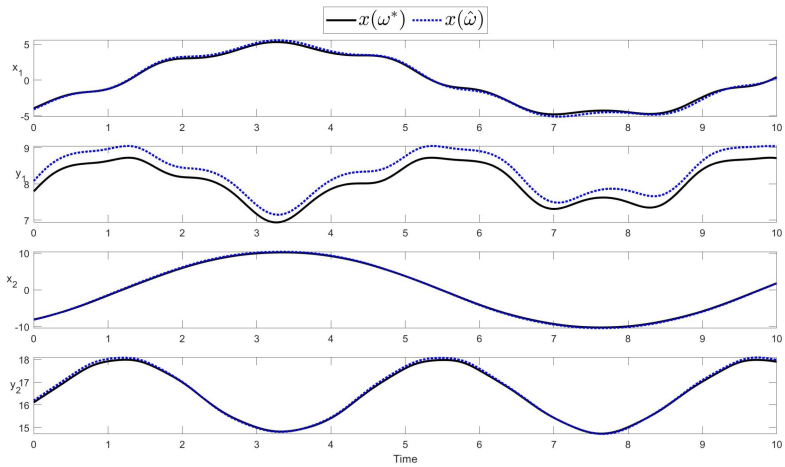

FIG. 4. Application of Algorithms 1 to estimate the double pendulum parameters $\omega = (l_1, l_2, m_2)$. The horizontal and vertical coordinates x_i and y_i of the two bobs ($i = 1, 2$) for the true parameter ω^* (black) and using $\hat{\omega}$ when estimated using Algorithm 1 (dashed blue).

less than 30 iterations (results not shown). Some runs converge to parameters with a relatively low score. Nevertheless, since we choose the IPA run of the highest score, we find that the overall error of Algorithm 2 is low.

In light of the discussion in Sec. IV C, it is worth asking whether measuring the estimation error (17) is the right way to estimate one's performance in estimating a dynamical system's parameters. Certainly, it might be that the problem is inherently ill-posed and that ω^* is non-identifiable if another parameter $\omega \in \Omega$ produce observations similar to $y(t)$. Along this reasoning, we propose another metric to measure our performance, the prediction error. This metric compares the normalized mean square error (MSE) of the predicted

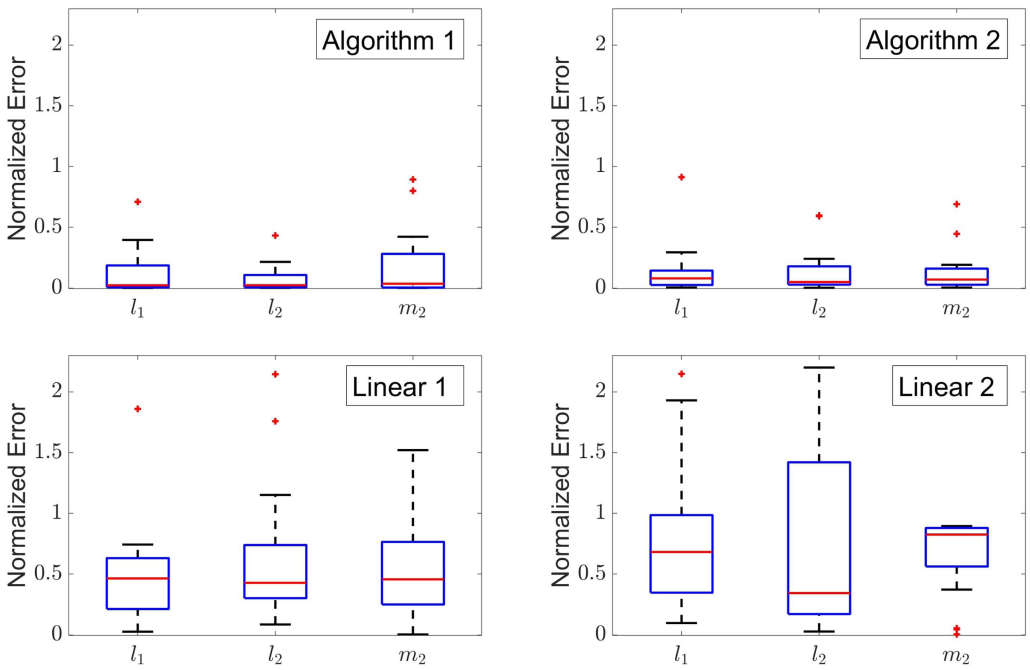

FIG. 5. Application of Algorithms 1 and 2 to estimate the double pendulum parameters $\omega = (l_1, l_2, m_2)$. Top: Box plots of normalized errors (17) of estimated parameters l_1, l_2, and m_2 for the double pendulum based on 20 test cases. Results base on Algorithm 1 (left) and Algorithm 2 (right). Bottom: Box plots of normalized errors for the linear scores 1 and 2, (19) and (20), respectively. In all box plots, red lines represent the medians, the blue box represents the 25th and 75th quantiles, the black whiskers represent ± 2.7 standard deviations, and any data point beyond this distance is considered as an outlier and marked by a red plus.

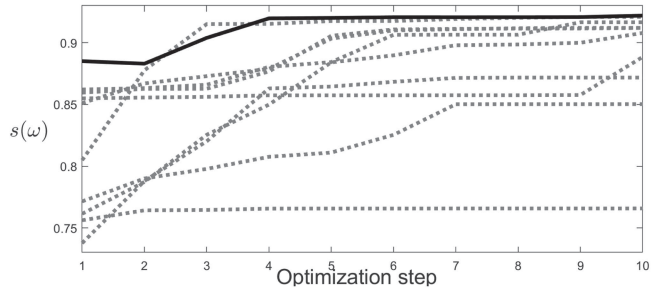

FIG. 6. Ten runs of the IPA optimization method in Algorithm 2 for the case of the double pendulum (see Sec. V A). The score (15) vs the optimization step. Here, the initial parameters for the IPA are drawn uniformly at random from $[0.9, 1.1] \cdot \omega^*$. The solid line indicates the best IPA run in terms of the score $s(\omega)$.

trajectory $x(t;\hat{\omega})$ from the true one $x(t;\omega^*)$,

$$\frac{\int_{t=0}^{t_f} \|x(\tau;\hat{\omega}) - x(\tau;\omega^*)\|_2^2 \, d\tau}{\int_{t=0}^{t_f} \|x(\tau;\hat{\omega})\|_2^2 \, d\tau}. \tag{21}$$

We evaluate the prediction error of Algorithms 1 and 2 by comparing the true pendulum trajectory to the estimated trajectory using the same 20 synthetic videos used for the box plots in Fig. 5. The results (see Fig. 7) demonstrate that the median prediction errors for Algorithm 1 and 2 are 0.27% and 0.2%, respectively. Moreover, both methods attain a prediction error smaller than 9% in all of the simulations. In this experiment, the prediction error (21) is comparable to the estimation error (17) squared, as could be expected from their respective definitions.

B. Estimation error decreases with signal length

In many experiments, the overall run time $T_f = N \cdot \Delta t$ of the experiment is a tunable parameter. How does T_f affect the estimation error (17)? In particular, we want to test the intuition according to which longer signals provide more information. For each $T_f = 1, 2, \ldots, 10$, we draw 100 samples of l_1, l_2, m_2, each with the uniform distribution on the set $\{1.5, 2, 2.5, \ldots, 7\}$. To each set of these parameters, we apply Algorithm 1 (see Sec. IV). The mean estimation error (17) decays as a function of T_f (see Fig. 8). Furthermore, it is evident that the standard deviation generally decreases with T_f, which reinforces the intuition of more information in longer signals.

We remark, however, that we do not expect the accuracy to increase with T_f for all dynamical systems, especially not for systems with attractors or limiting cycles. Consider for example $\dot{x}(t) = -x$ with $x(0) = \omega \in \mathbb{R}_+$. Since $x(t;\omega) = \omega e^{-t}$, then $x(t;\omega) \approx 0$ for $t \gg 1$ independently of ω. Since the kernels we use in Algorithms 1 and 2 weight all times equally, the longer the signal is, the more weight that is given to times $t \gg 1$, where ω^* is practically non-identifiable (see discussion in Sec. IV C).

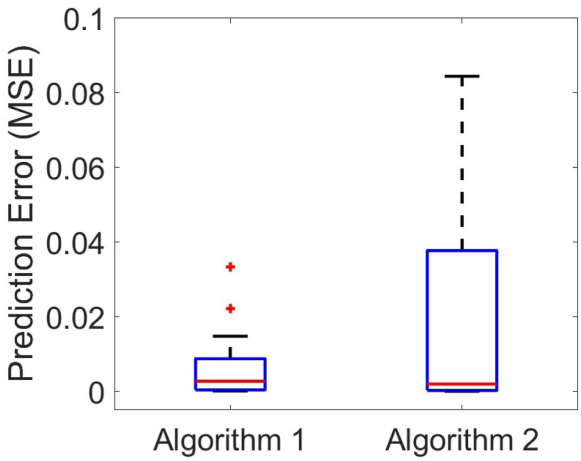

FIG. 7. Box plots of normalized prediction error (21) for the double pendulum experiment, as in Fig. 5, for Algorithms 1 and 2.

C. Lorenz'63 system

Consider the Lorenz (or Lorenz'63[46]) system of ODEs

$$\begin{aligned} \dot{x}_1(t) &= \sigma(x_2 - x_1), \\ \dot{x}_2(t) &= x_1(\rho - x_3) - x_2, \\ \dot{x}_3(t) &= x_1 x_2 - \beta x_3, \end{aligned} \tag{22}$$

where $\sigma, \rho, \beta \in \mathbb{R}_+^3$ are the model parameters. The Lorenz system is nonlinear, and for certain parameters and initial conditions, it is chaotic [see Fig. 9(a)]. To embed this system in a high-dimensional

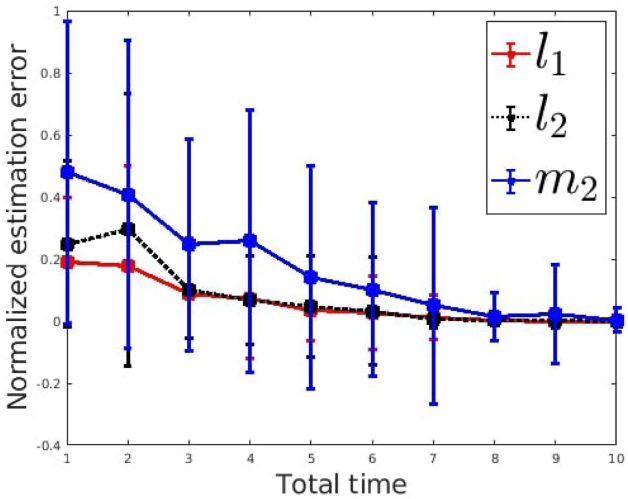

FIG. 8. Application of Algorithm 1 to the double pendulum systems with varying total run time T_f (see Sec. V B). Mean normalized estimation error (17) for the three parameters l_1, l_2, and m_2 as a function of the total length T_f, with error-bars of one standard deviation.

space, we follow the nonlinear transformation introduced in Refs. 15 and 18. Let $u_j \in \mathbb{R}^{128}$ be the jth order Legendre polynomial evaluated on 128 uniformly spaced points in $[-1, 1]$, and let[78]

$$y(t) = G(x(t)) := u_1 x_1(t) + u_2 x_2(t) + u_3 x_3(t) + u_4 x_1(t)^2 + u_5 x_2(t)^2 + u_6 x_3(t)^2, \tag{23}$$

where the ω notations were omitted for brevity [see Fig. 9(b)]. We apply Algorithm 2 to $y(t)$ to estimate ω^*. Our algorithm's estimation of these parameters leads to nearly indistinguishable low-dimensional trajectories $x(t)$ [see Fig. 9(c)].

We then consider a *noisy* observation function

$$\mathcal{G}(x(t); \zeta) = G(x(t) + \zeta), \tag{24}$$

where G is given by (23) and $\zeta \sim \mathcal{N}(0, \sigma^2 I_3)$ is three-dimensional normally distributed with $\sigma = 15$. Note that in this case, the resulting noise in the observation is neither additive nor Gaussian. The estimated parameters produce comparable trajectories [see Fig. 9(d)].

To systematically study the performance of our proposed approach, we repeat the following experiment 20 times: we set $\beta = 8/3$, draw $\sigma \in [15, 25]$ and $\rho \in [40, 80]$, the initial conditions $x_1(0)$, $x_2(0)$, and $x_3[0]$ from $[0, 1]$, all uniformly at random. For each set of parameters and initial conditions, we solve the Lorenz ODE (22) and sample the solution with $\Delta t = 0.01$ and $T_f = N \cdot \Delta t = 10$, where $N = 1000$. For each of these 20 instances we estimate the parameters σ, and ρ using Algorithms 1 and 2. For Algorithm 1, we use a grid with 50 values of each parameter, i.e., a search-grid of size $|\Omega_{\text{search}}| = 50^2 = 2500$, with $\Delta\sigma = 0.2$ and $\Delta\rho = 0.8$. The

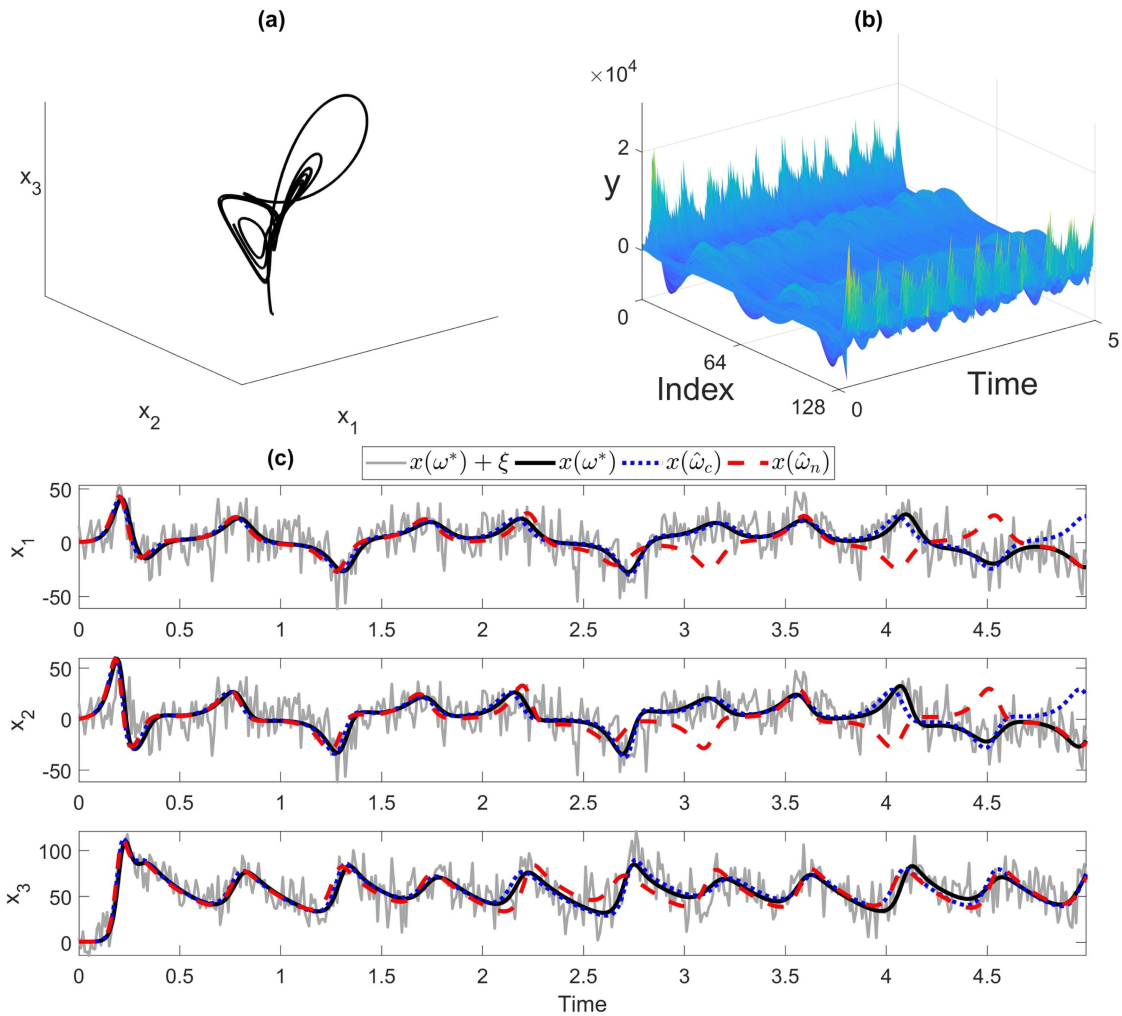

FIG. 9. (a) Lorenz system (22) with $\rho = 60$, $\sigma = 20$, and $\beta = 8/3$. (b) The observed $y(t)$ [see (23)]. (c) The coordinates of $x(t; \omega^*)$ (solid, black) and its noisy variant $x(t; \omega^*) + \zeta$ (grey) vs time for the true parameter. For each coordinate, we present the trajectory using the estimated parameters based on the clean signal $\hat{\omega}_c$ (dots, blue) and noisy signal $\hat{\omega}_n$ (dashes, red).

median normalized estimation error (17) of Algorithm 1 is 12% and 1.2% for σ and ρ, respectively. We repeat the same experiment for the optimization-based Algorithm 2 with $|\Omega_{\text{init}}| = 100$, where the overall median normalized errors are 7.5% and 1.6% for σ and ρ, respectively (see box plots in Fig. 10). To provide a baseline, we further evaluate the performance of an "Oracle" estimator. We define the Oracle estimator as the linear maximum likelihood estimator (4) when G is known. Using this estimator, we find the parameter $\omega \in \Omega_{\text{search}}$ that minimizes the square difference between y and $G(x(\omega))$. The box plots of the oracle estimation appear in the right side of Fig. 10, the median normalized errors are 6.4% and 0.3% for σ and ρ, respectively. Next, we use Algorithm 2 to estimate all three parameters of the Lorenz system σ, ρ, and β. To ensure observability, we use $\beta = 8/3$ and draw $\sigma \in [15, 25]$ and $\rho \in [40, 80]$, the initial conditions $x_1(0)$, $x_2(0)$, and $x_3(0)$ from $[0, 1]$, all uniformly at random.[79] We use $|\Omega_{init}| = 100$ (β is initialized by drawing from $[2, 3]$), and obtain an overall median normalized of 7.4%, 5.3%, and 5.1% for σ, ρ and β, respectively (see bottom right box plots in Fig. 10).

The accuracy of Algorithm 1 is limited by the grid resolution of Ω_{search}. The introduction of an optimization scheme in Algorithm 2 removes this limitation, but brings a host of other accuracy issues, e.g., the non-convexity of the optimization problem (12). However, practical considerations aside, what is the inherent accuracy of the score (15), independently of the search or optimization method? To answer this question, we repeated the test of Algorithm 1 on the noiseless observation (23) of the Lorenz system (22), but when *only* the parameter σ is unknown. This allows us to substantially refine the grid and increase the size of Ω_{grid} from 10 to 10^4. For each grid-resolution, we average over 100 simulations and observe that the median estimation error (17) decreases by more than an order of magnitude (see Fig. 11). Whether this trend saturates at some point, or conversely the error vanishes as $|\Omega_{\text{grid}}| \to \infty$, remains an interesting open question.

D. Model coordinates and phase space coordinates

The results above highlight an important distinction between the model coordinates $x(t; \omega)$ and phase space coordinates. From a dynamical systems perspective, it might seem as if our model coordinates $x(t; \omega)$ in the double pendulum example consist of only partial data—the double pendulum system (Appendix C) is a fourth-order ODE, with two angles and two angular velocities, while we use only the position coordinates. Therefore, our model coordinates $x(t; \omega)$

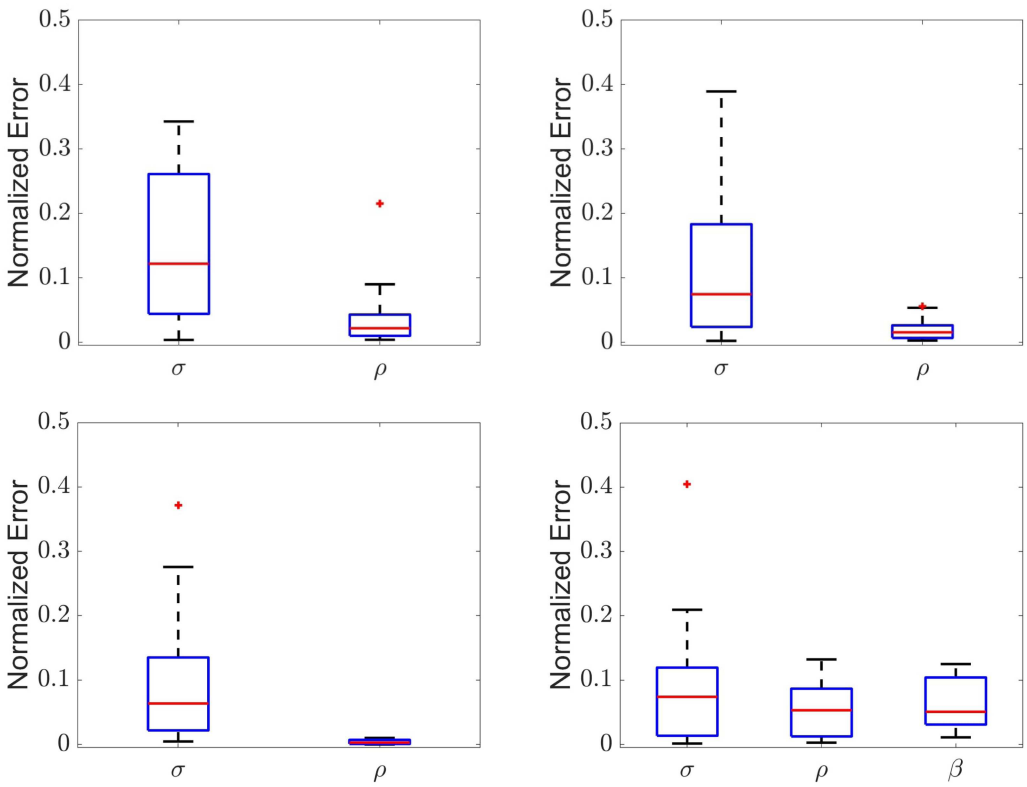

FIG. 10. Box plots of normalized errors (17) for estimated parameters σ and ρ for the Lorenz'63 model (22) given by the observation function (23). Top left: Algorithm 1. Top right: Algorithm 2. Red lines represent the medians and whisker bars indicate the 25th and 75th quantiles. Bottom left: Oracle estimator using the unknown observation function G. Bottom right: estimation of three parameters using Algorithm 2.

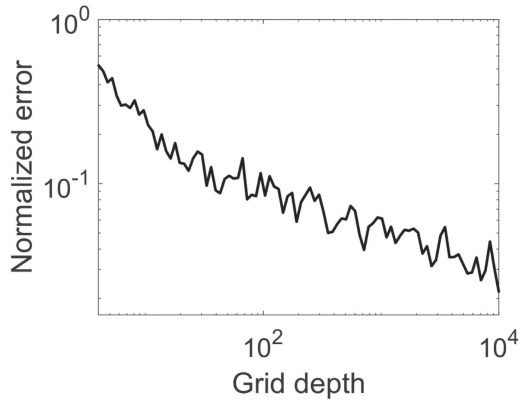

FIG. 11. Estimating σ in the Lorenz equation (22) from the observations (23) where all other parameters are fixed and known. Parameter estimation error (17) vs number of grid points $|\Omega_{\text{grid}}|$ (see Algorithm 1).

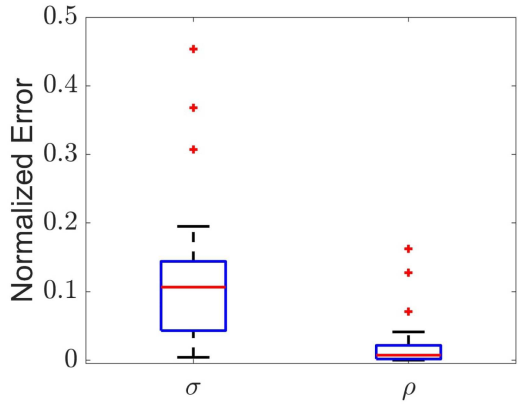

FIG. 12. Box plots of the normalized errors (17) for estimating σ and ρ in the Lorenz system (22) based on the *partial* observations function y_{partial} [see (25)].

do not specify a single point in phase space. The key observation here is that we observe a time series, and not a single point in time. We claim that temporal derivatives are implied by the time series. This can be understood heuristically, as the differences between subsequent times are indicative of the velocities. In the spirit of Taken's theorem, delay coordinates $(x(t_1;\omega),\ldots,x(t_n;\omega))$ can reconstruct the phase space, especially in this instance where we only "drop" two coordinates (see, e.g., Ref. 26).

To test the hypothesis that time series of partially observed data can be sufficient for parameter estimation purposes, we applied Algorithm 2 for the case where only two out of the three coordinates of the Lorenz model (22) are available; we use the following *partial* observation function:

$$y_{\text{partial}}(t) = G(x(t)) := u_1x_1(t) + u_2x_2(t) + u_4x_1(t)^2 + u_4x_2(t)^2, \tag{25}$$

where $x_1(t)$ and $x_2(t)$ are coordinates of the Lorenz'63 system (22) and $u_i \in \mathbb{R}^{128}$ be the ith Legendre polynomial. Next, we simulate 20 trajectories based on the same scheme described in Sec. V C. In this experiment, since $y_{\text{partial}}(t)$ only depends on $x_1(t)$ and $x_2(t)$, we only used these two coordinates to generate the kernel K^x [see (14)]. We apply Algorithm 2 with $|\Omega_{init}| = 100$, and obtain an overall median normalized of 10.65% and 0.74% for σ and ρ, respectively (see box plots in Fig. 12). This is evidence that a time series of partially observed data is sufficient to estimate the underlying parameters.

VI. DISCUSSION

A. Relevant literature—Learning and dynamics

In what follows, we discuss the relation between the main problem of this paper (parameter estimation with an unknown observation function) to notable questions at the interface of dynamical systems, statistics, and machine learning.

This study is related to the fast-growing field of model discovery and machine learning of physical systems in general.[4,8,10,12,15,18,23,25,69] Generally (notwithstanding their many differences), these works aim to discover governing equations from data using various machine learning techniques. There are three features that distinguish our study from model discovery studies. First, our approach and settings are not agnostic to the physical modeling, and this study does not aim at "physics discovery."[4,15] Rather, we use the laws of physics and known models to estimate the parameters. The second distinction is that the unknown/unspecified portion of our settings is the correspondence between the model and the observations [denoted by G; see (2)], which in model discovery studies is usually assumed to be known.

The third distinction between this study and machine learning problems in general is that we do not have ample training data, i.e., many pairs of a parameter ω and the resulting observation $y(t)$. Nor do we even observe many different signals $y(t)$, which correspond to different (unknown) parameters ω (as in, e.g., Ref. 45). Critically, even though one can generate many trajectories $x(t;\omega)$ by solving the underlying ODE, our data include only a single experiment with a single observed $y(t)$.

Another increasingly popular application of machine learning to dynamical systems is learning implicit propagation models. By observing many instances of a system's evolution, one learns the time-evolution or Koopman operator to propagate the observations in time *without* learning an underlying ODE or partial differential equation (PDE), either in a model-free fashion,[27,51,68] or using partial knowledge of the underlying system.[2,5,67,70] In this study, propagating the observations $y(t)$ (e.g., a video) in time does not seem to advance the estimation of the system's parameters ω^*.

Our problem can be considered as a novel variant of standard inverse problems.[3,61] Broadly speaking, in these problems, a known forward (perhaps noisy) map $\omega \mapsto y(t) = G(x(t;\omega);\zeta)$ is inverted in some sense to recover x or ω^*. This is typically done using the observations y and some *a priori* knowledge on the model. The key

difference between standard inverse problems and this study is that G in our case is completely unknown, and we do not attempt to recover it. A particularly relevant type of inverse problems is that where G is only known approximately, due to, e.g., modeling errors or numerical approximation.[20,57]

Also related to this study is the general problem of nonlinear dimensionality reduction and manifold learning. Since $y = G(x)$ (suppressing noise), $\{y(t;\omega) \mid t \geq 0,\ \omega \in \Omega\}$ can be viewed as a $m+1$ dimensional manifold in the ambient space $\mathbb{R}^D$. Manifold learning techniques can, therefore, be applied to recover this low-dimensional structure.[7,21,24,25,60,73] These techniques, however, do not provide a straightforward way to compare the sub-manifold resulting from $y(t)$ with the model coordinates $x(t;\omega)$. Even diffusion-based techniques, which allow one to identify the modality of $x(t;\omega^*)$ in the observation space[42,44] do not lead directly to ways to identify ω^* from the full parameter space Ω. We consider diffusion-based solutions of our problem an interesting direction of future studies.

B. Future works

In this work, we presented two examples (the double pendulum and the Lorenz'63 system), where the number of estimated parameters is relatively small. As in other inverse problems, many issues might arise from considering a high-dimensional parameter, e.g., identifiability, computational efficiency, and convergence of the optimization scheme. Adjusting and extending our scheme to high-dimensional parameters is, therefore, an interesting remaining challenge.

One class of potential applications is the deduction of homogenized constants from reduced models. Consider, for concreteness, a Hamiltonian describing optical propagation of laser beams in a waveguides array, or of a quantum-mechanical electron in two-dimensional graphene. The dynamics of the corresponding system can be approximately reduced to an effective Dirac equation, where short-time dynamics and the micro-structure of the lattice are homogenized to a single constant wave speed. It might be possible, using our method, to observe the full dynamics, and by indirect comparison to the reduced model deduce the homogenized velocity and other constants of the lattice such as topological charges. Even though this problem involves complex PDEs, our proposal is to estimate a low-dimensional parameter. This is only an example to a broad class of homogenization schemes in optics, radio-frequency arrays, fluid dynamics, and continuum mechanics. If successful, such an approach could allow measurement from complex dynamical systems by indirect and non-explicit comparison to cheaper and simpler reduced models.

Another potential application of our method is predicting hematoma expansion after intracerebral hemorrhage based on non-contrast computed tomography (CT). Over the years, pathological observations have led to the development of several parametric models of the propagation of hemorrhage, e.g., Refs. 29, 13, and 14. This is another case where low-dimensional parameters underlie high-dimensional data and requires the ability to estimate the model parameters of a dynamical system from noisy observations consisting of only a single trajectory. Here, the parameter estimation problem is to identify the dynamical regime. One can investigate the ability of our method to estimate the model parameters from a small number of CT scans from a single patient, thereby allowing us to determine whether the hematoma is likely to expand, which in turn necessitates a life-saving, yet risky medical intervention, or whether the hematoma contracts and only monitoring is required. We believe that there exists a large number of such applications from the realm of medical data analysis and related fields, which can benefit from the presented method.

The approach presented in this paper can be viewed as a general scheme (see Fig. 2) of which Algorithms 1 and 2 are two effective representatives. Our choice of Gaussian kernels in (14) and (13) is judicious, standard, and effective, but might be improved. One avenue for improvement would be considering kernels that take into account the temporal progression of the samples, as in, e.g., Refs. 25 and 62. One of the main constraints in this paper is that we observe only a single time series $y(t)$. Suppose the problem is extended to measure many such time series, either by altering ω^* or by changing the initial conditions, then an appropriate kernel might be learned (see, e.g., Refs. 41, 22, and 50).

AUTHORS' CONTRIBUTIONS

O.L. and A.S. contributed equally to this work.

APPENDIX A: DERIVATION OF (9)

We write here the details for $\bar{\Phi}$ and K^y:

$$\begin{aligned}\|\bar{\Phi}\|_F^2 &= \mathrm{Tr}(\bar{\Phi}^\top \bar{\Phi}) \\ &= \sum_{j=1}^{N} \langle \bar{\Phi}_{\cdot,j}, \bar{\Phi}_{\cdot,j} \rangle \\ &= \sum_{j=1}^{N} \langle \phi(y(t_j)) - \frac{1}{N}\sum_{i=1}^{N} \phi(y(t_i)), \phi(y(t_j)) - \frac{1}{N}\sum_{i'=1}^{N} \phi(y(t_{i'})) \rangle\end{aligned}$$

$$
\begin{aligned}
&= \sum_{j=1}^{N}\left[\langle\phi(y(t_j)),\phi(y(t_j))\rangle - \frac{2}{N}\sum_{i}\langle\phi(y(t_j)),\phi(y(t_i))\rangle + \frac{1}{N^2}\sum_{i,i'}\langle\phi(y(t_i)),\phi(y(t_{i'}))\rangle\right]\\
&= \sum_{j=1}^{N}\langle\phi(y(t_j)),\phi(y(t_j))\rangle - \frac{2}{N}\sum_{i,j}\langle\phi(y(t_j)),\phi(y(t_i))\rangle + \frac{1}{N^2}\sum_{j,i,i'}\langle\phi(y(t_i)),\phi(y(t_{i'}))\rangle\\
&= \sum_{j=1}^{N}\langle\phi(y(t_j)),\phi(y(t_j))\rangle - \frac{2}{N}\sum_{i,j}\langle\phi(y(t_j)),\phi(y(t_i))\rangle + \frac{N}{N^2}\sum_{i,i'}\langle\phi(y(t_i)),\phi(y(t_{i'}))\rangle\\
&= \sum_{j=1}^{N}k^y(y(t_j),y(t_j)) - \frac{1}{N}\sum_{i,\ell}k^y(y(t_i),y(t_\ell))\\
&= \mathrm{Tr}(K^yH), \quad H_{ij} = \delta_{ij} - \frac{1}{N},
\end{aligned}
$$

where we have used the fact that ϕ is real valued and, therefore, the inner product is symmetric.

APPENDIX B: ROOF OF LEMMA 1

Since $\alpha \cdot \mathrm{Tr}(A) = \mathrm{Tr}(\alpha A)$ for every scalar α and square matrix A, then

$$
s(\omega) = \mathrm{Tr}\left(\frac{K^x(\omega)H}{\|K^x(\omega)H\|_F}\frac{K^yH}{\|K^yH\|_F}\right) = \left\langle\frac{K^x(\omega)H}{\|K^x(\omega)H\|_F}, \frac{K^yH}{\|K^yH\|_F}\right\rangle_F,
$$

where, as before, $\langle\cdot,\cdot\rangle_F$ is the Frobenius inner product. Hence, we can restrict the analysis to the case of $\|K_xH\|_F = \|K_yH\|_F = 1$. We first prove that

$$
A = \arg\max_{\substack{B\in M_n(\mathbb{R})\\ \|B\|_F=1}}\langle A,B\rangle_F
$$

for any $A \in M_n(\mathbb{R})$ with $\|A\|_F = 1$. Define $\Delta := B - A$. Then,

$$
1 = \|B\|_F^2 = \|A\|_F^2 + \langle A,\Delta\rangle_F + \langle\Delta,A\rangle_F + \|\Delta\|_F^2.
$$

Since A and B are real, $\langle A,\Delta\rangle_F = \langle\Delta,A\rangle_F$ and so $\langle A,\Delta\rangle_F = -\|\Delta\|_F^2/2$. Therefore,

$$
\begin{aligned}
\langle A,B\rangle_F &= \langle A,A\rangle_F + \langle A,\Delta\rangle_F\\
&= \|A\|_F^2 - \frac{1}{2}\|\Delta\|_F^2\\
&= 1 - \frac{1}{2}\|\Delta\|_F^2.
\end{aligned}
$$

Therefore, the maximum of this expression is attained when $\|\Delta\|_F = 0$, i.e., when $\Delta = 0$ and $A = B$.

In the case where $A = K^yH$ and $B = K^x(\omega)H$, the two centered kernels are equal if and only if we have the pairwise equalities $\|y_i(\omega^*) - y_j(\omega^*)\|_2 = \|x_i(\omega) - x_j(\omega)\|_2$ for all times t_i and t_j. Since G is an $\ell^2(\mathbb{R}^d \to \mathbb{R}^d)$ isometry, this inequality occurs if and only if $\|x_i(\omega^*) - x_j(\omega^*)\|_2 = \|x_i(\omega) - x_j(\omega)\|_2$ for all times t_i and t_j, i.e., where $x_i(\omega) = Tx_i(\omega^*)$ for an $\ell^2(\mathbb{R}^d)$ isometry.

APPENDIX C: EXPLICIT ODEs FOR THE DOUBLE PENDULUM

For completeness, we include here the Euler–Lagrange ODEs that govern the double pendulum system (see Ref. 56 for derivation and details). Denote by θ_1 and θ_2 the angles of the respective pendulums from the negative y axis, then

$$
\frac{d}{dt}(\vec{\theta}) = \frac{d}{dt}\begin{pmatrix}\theta_1\\ \theta_2\\ \theta_3\\ \theta_4\end{pmatrix} = \begin{pmatrix}\theta_3\\ \theta_4\\ g_1(\vec{\theta})\\ g_1(\vec{\theta})\end{pmatrix},
$$

where

$$
g_1(\vec{\theta}) = \frac{g(\sin\theta_2\cos\Delta\theta - \mu\sin\theta_1) - (l_2\dot{\theta_2}^2 + l_1\dot{\theta_1}^2\cos\Delta\theta)\sin\Delta\theta}{l_1(\mu - \cos^2\Delta\theta)},
$$

$$
g_2(\vec{\theta}) = \frac{g\mu(\sin\theta_1\cos\Delta\theta - \sin\theta_2) + (\mu l_1\dot{\theta_1}^2 + l_2\dot{\theta_2}^2\cos\Delta\theta)\sin\Delta\theta}{l_2(\mu - \cos^2\Delta\theta)},
$$

and where

$$
\Delta\theta = \theta_1 - \theta_2, \quad \mu = 1 + \frac{m_1}{m_2}.
$$

DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author upon reasonable request.

REFERENCES

[1] F. Abramovich and Y. Ritov, *Statistical Theory: A Concise Introduction* (CRC Press, 2013).

[2] T. Arcomano, I. Szunyogh, J. Pathak, A. Wikner, B. R. Hunt, and E. Ott, "A machine learning-based global atmospheric forecast model," Geo. Res. Lett. **47**, e2020GL087776 (2020).

[3] R. C. Aster, B. Borchers, and C. H. Thurber, *Parameter Estimation and Inverse Problems* (Elsevier, 2018).

[4] S. Atkinson, W. Subber, L. Wang, G. Khan, P. Hawi, and R. Ghanem, "Data-driven discovery of free-form governing differential equations," arXiv:1910.05117 (2019).

[5]O. Azencot, N. B. Erichson, V. Lin, and M. W. Mahoney, "Forecasting sequential data using consistent Koopman autoencoders," Proceedings of the 37th International Conference on Machine Learning (PMLR, 2020), pp. 475–485, see http://proceedings.mlr.press/v119/azencot20a.html.
[6]F. T. Bach and M. I. Jordan, "Kernel independent component analysis," J. Mach. Learn. Res. **3**, 1–48 (2002).
[7]M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," Neural Comput. **15**, 1373–1396 (2003).
[8]T. Bertalan, F. Dietrich, I. Mezic, and I. G. Kevrekidis, "On learning Hamiltonian systems from data," Chaos **29**, 121107 (2020).
[9]M. B. Blaschko and A. Gretton, "Taxonomy inference using kernel dependence measures," Max-Planck-Insitut Technical Report No. 181 (2008).
[10]J. Bongard and H. Lipson, "Automated reverse engineering of nonlinear dynamical systems," Proc. Natl. Acad. Sci. U.S.A. **104**, 9943–9948 (2007).
[11]B. Boots and G. J. Gordon, "Two-manifold problems with applications to nonlinear system identification," in *Proceedings of the International Conference on Machine Learning* (Omnipress, 2012).
[12]J. J. Bramburger and J. N. Kutz, "Poincré maps for multiscale physics discovery and nonlinear Floquet theory," Physica D **408**, 132479 (2020).
[13]B. Brouwers, A. Biffi, M. Ayres A, K. Schwab, L. Cortellini, M. Romero J, and N. Goldstein J, "Apolipoprotein E genotype predicts hematoma expansion in lobar intracerebral hemorrhage," Stroke **43**(6), 1490–1495 (2012).
[14]B. Brouwers, Y. Chang, J. Falcone, X. Cai, M. Ayres, T. Battey, A. Vashkevich, K. McNamara, V. Valant, and K. Schwab, "Predicting hematoma expansion after primary intracerebral hemorrhage," JAMA Neurol. **71**, 158–164 (2014).
[15]S. L. Brunton, J. L. Proctor, and J. N. Kutz, "Discovering governing equations from data by sparse identification of nonlinear dynamical systems," Proc. Natl. Acad. Sci. U.S.A. **113**, 3932–3937 (2016).
[16]R. H. Byrd, M. E. Hribar, and J. Nocedal, "An interior point algorithm for large-scale nonlinear programming," SIAM J. Optim. **9**, 877–900 (1999).
[17]R. H. Byrd, J. C. Gilbert, and J. Nocedal, "A trust region method based on interior point techniques for nonlinear programming," Math. Program. **89**, 149–185 (2000).
[18]K. Champion, B. Lusch, J. N. Kutz, and S. L. Brunton, "Data-driven discovery of coordinates and governing equations," Proc. Natl. Acad. Sci. U.S.A. **116**, 22445–22451 (2019).
[19]A. Charu, A. Hinneburg, and D. Keim, "On the surprising behavior of distance metrics in high dimensional space," in *International Conference on Database Theory* (Springer, Berlin, 2001).
[20]E. Clearey, A. Garbuno-Inigo, S. Lan, T. Schneider, and A. M. Stuart, "Calibrate, emulate, sample," J. Comp. Phys. **424**, 109716 (2021).
[21]R. R. Coifman and S. Lafon, "Diffusion maps," Appl. Comput. Harm. Anal. **21**, 5–30 (2006).
[22]C. Cortes, M. Mehryar, and R. Afshin, "Algorithms for learning kernels based on centered alignment, Corinna Cortes, Mehryar Mohri, Afshin Rostamizadeh" J. Mach. Learn. Res. **13**(1), 795–828 (2019).
[23]M. Cranmer, A. Sanchez-Gonzalez, P. Battaglia, R. Xu, K. Cranmer, D. Spergel, and S. Ho, "Discovering symbolic models from deep learning with inductive biases," *Advances in Neural Information Processing Systems 33* (NeurIPS 2020), see https://proceedings.neurips.cc/paper/2020/hash/c9f2f917078bd2db12f23c3b413d9cba-Abstract.html.
[24]D. L. Donoho and C. Grimes, "Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data," Proc. Natl. Acad. Sci. U.S.A. **100**, 5591–5596 (2003).
[25]C. J. Dsilva, R. Talmon, R. R. Coifman, and I. G. Kevrekidis, "Parsimonious representation of nonlinear dynamical systems through manifold learning: A chemotaxis case study," Appl. Comput. Harmon. Anal. **44**, 759–773 (2018).
[26]C. J. Dsilva, R. Talmon, C. W. Gear, R. R. Coifman, and I. G. Kevrekidis, "Data-driven reduction for a class of multiscale fast-slow stochastic dynamical systems," SIAM J. Appl. Dyn. Syst. **15**, 1327–1351 (2016).
[27]P. Dubois, T. Gomez, L. Planckaert, and L. Perret, "Data-driven predictions of the Lorenz system," Physica D **408**, 132495 (2020).
[28]N. Dunford and J. T. Schwartz, *Linear Operators, Part I* (Wiley, New York, 1958).
[29]M. Fisher, "Pathological observations in hypertensive cerebral hemorrhage," J. Neuropathol. Exp. Neurol. **30**, 536–550 (1971).
[30]K. Fukumizu, F. R. Bach, and A. Gretton, "Statistical consistency of kernel canonical correlation analysis," J. Mach. Learn. Res. **8**, 361–383 (2007).
[31]K. Fukumizu, A. Gretton, X. Sun, and B. Schölkopf, "Kernel measures of conditional dependence," in *Advances in Neural Information Processing* (Curran Associates Inc., 2008), pp. 489–496.
[32]A. Gretton, O. Bousquet, A. Smola, and B. Schölkopf, "Measuring statistical dependence with Hilbert–Schmidt norms," in *International Conference on Algorithmic Learning Theory* (Springer, Berlin, 2005), pp. 63–77.
[33]A. Gretton, R. Herbrich, A. J. Smola, O. Bousquet, and B. Schölkpf, "Kernel methods for measuring independence," J. Mach. Learn. Res. **6**, 2075–2129 (2005).
[34]A. Gretton, K. Fukumizo, C. H. Teo, K. Song, B. Schölkopf, and A. J. Smola, "A kernel statistical test of independence," in *Advances in Neural Information Processing* (Curran Associates Inc., 2008), pp. 585–592.
[35]A. Hinneburg, A. Charu, and D. Keim, "What is the nearest neighbor in high dimensional spaces?," in *26th International Conference on Very Large Databases* (Morgan Kaufmann Publishers Inc., 2000).
[36]T. Hofmann, B. Schölkopf, and A. J. Smola, "Kernel methods in machine learning," Ann. Statist. **36**, 1171–1220 (2008).
[37]A. Iserles, *A First Course in the Numerical Analysis of Differential Equations* (Cambridge University, 2009).
[38]R. T. Keller and Q. Du, "Discovery of dynamics using linear multistep methods," SIAM J. Numer. Anal. **59**, 429–455 (2020).
[39]S. Lafon, Y. Keller, and R. Coifman, "Data fusion and multicue data matching by diffusion maps," IEEE Trans. Pattern Anal. Mach. Intell. **28**(11), 1784–1797 (2006).
[40]L. D. Landau and E. M. Lifshitz, *Course of Theoretical Physics, Mechanics* (Pergamon, Oxford, 1980), Vol. 1.
[41]L. Le, J. Hao, Y. Xie, and J. Priestley, "Deep kernel: Learning kernel function from data using deep neural network," in *Proceedings of the 3rd IEEE/ACM International Conference on Big Data Computing, Applications and Technologies* (Association for Computing Machinery (ACM), 2016).
[42]R. R. Lederman and R. Talmon, "Learning the geometry of common latent variables using alternating-diffusion," Appl. Comput. Harm. Anal. **44**, 509–536 (2018).
[43]O. Lindenbaum, M. Salhov, A. Yeredor, and A. Averbuch, "Gaussian bandwidth selection for manifold learning and classification," Data Min. Knowl. Discov. **34**, 1–37 (2020).
[44]O. Lindenbaum, A. Yeredor, M. Salhov, and A. Averbuch, "Multi-view diffusion maps," Inf. Fusion **55**, 127–149 (2020).
[45]C.-H. Liu, Y. Tao, D. Hsu, Q. Du, and S. J. L. Billinge, "Using a machine learning approach to determine the space group of a structure from the atomic pair distribution function," Acta Cryst. **A75**, 633–643 (2019).
[46]E. N. Lorenz, "Deterministic nonperiodic flow," J. Atmos. Sci. **20**, 130–141 (1963).
[47]T. Michaeli, W. Wang, and K. Livescu, "Nonparametric canonical correlation analysis," in *International Conference on Machine Learning* (JMLR.org, 2016), pp. 1967–1976.
[48]V. V. Nguyen and E. F. Wood, "Review and unification of linear identifiability concepts," SIAM Rev. **24**, 34–51 (1982).
[49]J. Nocedal and S. J. Wright, *Numerical Optimization* (Springer, New York, 2006).
[50]H. Owhadi and G. R. Yoo, "Kernel flows: From learning kernels from data into the abyss," J. Comput. Phys. **389**, 22–47 (2019).
[51]J. Pathak, B. Hunt, M. Girvan, Z. Lu, and E. Ott, "Model-free prediction of large spatiotemporally chaotic systems from data: A reservoir computing approach," Phys. Rev. Lett. **120**, 024102 (2018).
[52]S. Saitoh, *Theory of Reproducing Kernels and its Applications* (Longman Scientific and Technical, Harlow, 1988).
[53]M. Salhov, O. Lindenbaum, Y. Aizenbud, A. Silberschatz, Y. Shkolnisky, and A. Averbuch, "Multi-view kernel consensus for data analysis," Appl. Comput. Harmon. Anal. **49**(1), 208–228 (2020).
[54]B. Schölkopf, A. Smola, and K.-S. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," Neural Comput. **10**, 1299–1319 (1998).

[55] J. Shawe-Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis* (Cambridge University, 2004).
[56] T. Shinbrot, C. Grebogi, J. Wisdom, and J. A. Yorke, "Chaos in a double pendulum," Am. J. Phys. **60**, 491–499 (1992).
[57] G. Shulkind, L. Horesh, and H. Avron, "Experimental design for nonparametric correction of misspecified dynamical models," SIAM/ASA J. Uncertain. Quant. **6**, 880–906 (2018).
[58] A. Singer, R. Erban, I. Kevrekidis, and R. R. Coifman, "Detecting intrinsic slow variables in stochastic dynamical systems by anisotropic diffusion maps," Proc. Natl. Acad. Sci. U.S.A. **106**, 16090–16095 (2009).
[59] A. J. Smola and B. Schölkopf, *Learning with Kernels* (GMD-Forschungszentrum Informationstechnik, 1998), Vol. 4.
[60] B. Sober and D. Levin, "Manifold approximation by moving least-squares projection (MMLS)," Construct. Approx. **52**, 433–478 (2020).
[61] A. M. Stuart, "Inverse problems: A Bayesian perspective," Acta Num. **19**, 451–559 (2010).
[62] R. Talmon and R. R. Coifman, "Empirical intrinsic geometry for nonlinear modeling and time series filtering," Proc. Natl. Acad. Sci. U.S.A. **110**, 12535–12540 (2013).
[63] M. Vestner, R. Litman, A. Rodola, M. Bronstein, and D. Cremers, "Product manifold filter: Non-rigid shape correspondence via kernel density estimation in the product space," in *Proceedings of the Computer Vision and Pattern Recognition* (IEEE, 2017).
[64] A. F. Villaverde, "Observability and structural identifiability of nonlinear biological systems," Complexity **2019**, 8497093.
[65] R. A. Waltz, J. L. Morales, J. Nocedal, and D. Orban, "An interior algorithm for nonlinear optimization that combines line search and trust region steps," Math. Program. **107**, 391–408 (2006).
[66] D. Wang, L. Shi, and J. Cao, "Fast algorithm for approximate k-nearest neighbor graph construction," in *IEEE 13th International Conference on Data Mining Workshops* (IEEE, 2013), pp. 349–356.
[67] A. Wikner, J. Pathak, B. R. Hunt, M. Girvan, T. Arcomano, I. Szunyogh, A. Pomerance, and E. Ott, "Combining machine learning with knowledge-based modeling for scalable forecasting and subgrid-scale closure of large, complex, spatiotemporal systems," Chaos **30**, 053111 (2020).
[68] M. O. Williams, I. G. Kevrekidis, and C. W. Rowley, "A data-driven approximation of the Koopman operator: Extending dynamic mode decomposition," J. Nonlinear Sci. **25**, 1307–1346 (2015).
[69] O. Yair, R. Talmon, R. R. Coifman, and I. G. Kevrekidis, "Reconstruction of normal forms by learning informed observation geometries from data," Proc. Natl. Acad. Sci. U.S.A. **114**, E7865–E7874 (2017).
[70] J. Yu and J. S. Hesthaven, "Flowfield reconstruction method using artificial neural networks," AIAA J. **57**, 482–498 (2019).
[71] L. Zelnik-Manor and P. Perona, "Self-tuning spectral clustering," in *Advances in Neural Information Processing Systems* (MIT Press, 2004), pp. 1601–1608.
[72] O. Yair, F. Dietrich, R. Mulayoff, R. Talmon, and I. G. Kevrekidis, "Spectral discovery of jointly smooth features for multimodal data," arXiv:2004.04386, 2020.
[73] A. Holiday, M. Kooshkbaghi, J. M. Bello-Rivas, C. W. Gear, A. Zagaris, and I. G. Kevrekidis, "Manifold learning for parameter reduction," J. Comput. Phys. **392**, 419–431 (2019).
[74] It is not essential that G is defined on all of $\mathbb{R}^d$, but just on $\cup_{\omega\in\Omega}\{x(t;\omega) \mid t \geq 0\}$. Furthermore, in our analysis and examples, usually $\mathcal{Z}$ is either the space of the model $\mathbb{R}^d$ or the observation space $\mathbb{R}^D$.
[75] The kernels k^x and k^y are generally *not* inner products in the original spaces $\mathbb{R}^d$ and $\mathbb{R}^D$, respectively, since they are not linear in either of their components.
[76] It can often be the case that for ω values outside of Ω, the underlying ODE does not yield a well-posed solution. For example, the harmonic oscillator $\ddot{x}(t) + \omega x = 0$ for $\omega < 0$ yields exponentially growing and ill-posed solutions.
[77] In our simulations, we used MATLAB's implementation of the IPA method.
[78] This precise form of G is not particularly important, only that it is truly nonlinear and incorporates all of x's coordinates. Other observation functions were tested to yield similar result with our algorithm (results not shown).
[79] We observe that for $\beta \neq 8/3$, many different sets of parameters produce nearly indistinguishable trajectories, thus obstructing observability (see discussion in Sec. IV C).