Data-Driven Model Selections of Second-Order Particle Dynamics via Integrating Gaussian Processes with Low-Dimensional Interacting Structures *

Jinchao Feng†, Charles Kulick‡, and Sui Tang§

Abstract.
In this paper, we focus on the data-driven discovery of a general second-order particle-based model that contains many state-of-the-art models for modeling the aggregation and collective behavior of interacting agents of similar size and body type. This model takes the form of a high-dimensional system of ordinary differential equations parameterized by two interaction kernels that appraise the alignment of positions and velocities. We propose a Gaussian Process-based approach to this problem, where the unknown model parameters are marginalized by using two independent Gaussian Process (GP) priors on latent interaction kernels constrained to dynamics and observational data. This results in a nonparametric model for interacting dynamical systems that accounts for uncertainty quantification. We also develop acceleration techniques to improve scalability. Moreover, we perform a theoretical analysis to interpret the methodology and investigate the conditions under which the kernels can be recovered. We demonstrate the effectiveness of the proposed approach on various prototype systems, including the selection of the order of the systems and the types of interactions. In particular, we present applications to modeling two real-world fish motion datasets that display flocking and milling patterns up to 248 dimensions. Despite the use of small data sets, the GP-based approach learns an effective representation of the nonlinear dynamics in these spaces and outperforms competitor methods.

Key words. Particle-based system, data-driven methods, Gaussian process, kernel ridge regression, inverse problems, randomized numerical linear algebra

1. Introduction.

Interacting particle/agent systems are a broad spectrum of complex systems with multiple components interacting with each other and co-evolving with time. Individual interactions yield a wide variety of collective behaviors at different scales and levels of complexity such as clustering, alignment, swarming, synchronization, or dancing equilibrium. There are numerous real-world examples of such systems, including the orbits of planets, motion of self-propelled particles, flocking of birds, schooling of fish, aggregation of cells, consensus of opinions, and synchronization of oscillators over networks. Understanding the link between individual interactions and global-scale collective behaviors is one of the most fundamental problems in various disciplines.

Modeling interacting agents by differential equations has played a crucial role in exploring the emergence of collective behaviors from individual interactions. However, such systems are often high-dimensional and exhibit many possible dynamical couplings of components that contribute to the dynamics, making them challenging to study [1, 2, 3, 4]. Despite these challenges, recent work has made impressive progress in developing a general physical model derived from Newton's second law that can capture a wide range of collective behaviors [5, 6, 7, 8, 9]. This model describes a system of N agents interacting according to a set of ODEs, where each agent's motion is influenced by self-propulsion, friction, and interactions

*This work was partially supported by NSF DMS-2111303.
†School of Sciences, Great Bay University, Dongguan, Guangdong, China (jcfeng@gbu.edu.cn).
‡Departments of Mathematics, University of California, Santa Barbara, Isla Vista, CA (charles@math.ucsb.edu).
§Departments of Mathematics, University of California, Santa Barbara, Isla Vista, CA (suitang@math.ucsb.edu).

with other agents, represented by energy and alignment-based radial interaction kernels: for $i = 1, \cdots, N$

$$m_i \ddot{\boldsymbol{x}}_i = F_i(\boldsymbol{x}_i, \dot{\boldsymbol{x}}_i, \boldsymbol{\alpha}_i) + \sum_{i'=1}^{N} \frac{1}{N} \Big[\phi^E(||\boldsymbol{x}_{i'} - \boldsymbol{x}_i||)(\boldsymbol{x}_{i'} - \boldsymbol{x}_i) + \phi^A(||\boldsymbol{x}_{i'} - \boldsymbol{x}_i||)(\dot{\boldsymbol{x}}_{i'} - \dot{\boldsymbol{x}}_i) \Big], \tag{1.1}$$

where $m_i \geq 0$ is the mass of the agent i; $\ddot{\boldsymbol{x}}_i \in \mathbb{R}^d$ is the acceleration, $\dot{\boldsymbol{x}}_i \in \mathbb{R}^d$ is the velocity, and $\boldsymbol{x}_i \in \mathbb{R}^d$ is the position of agent i; the first term F_i is a parametric function of position and velocities, modeling self-propulsion and frictions of agent i with the environment with scalar parameters $\boldsymbol{\alpha}_i$ describing their strength; $||\boldsymbol{x}_j - \boldsymbol{x}_i||$ is the Euclidean distance; and the 1D functions $\phi^E, \phi^A : \mathbb{R}^+ \to \mathbb{R}$ are called the *energy* and *alignment-based radial interaction kernels* respectively. The ϕ^E term describes the alignment of positions based on the difference of positions; the ϕ^A term describes the alignment of velocities based on the difference of velocities. We summarize the relevant notations in Table 1.

Particular examples of (1.1) include the first-order systems ($m_i \equiv 0, \phi^A \equiv 0$) that model clustering and aggregation of agents with application to opinion dynamics [10], the second-order Cucker-Smale model ($\phi^E \equiv 0$) [11] for the flocking behavior of animals and robots, the second-order self-propelling particle model ($\phi^A \equiv 0$) that is shown to reproduce (double) milling, ring, escaping or swarming behaviors of biological motors [12], and the anticipation dynamics [13] ($\phi^A, \phi^E \neq 0$) that describes the velocity alignment and spatial concentration of animal groups. For simplicity of description, we assume the masses of all agents are the same and equal to $\boldsymbol{m}$. We write the second-order model (1.1) in a compact form:

$$\boldsymbol{m}\boldsymbol{Z}(t) = F_{\boldsymbol{\alpha}}(\boldsymbol{Y}(t)) + \mathbf{f}_{\boldsymbol{\phi}}(\boldsymbol{Y}(t)) \tag{1.2}$$

where $\boldsymbol{Y}(t) := \begin{bmatrix} \boldsymbol{X}(t) \\ \boldsymbol{V}(t) \end{bmatrix} \in \mathbb{R}^{2dN}$ represents the state variable for the system, $\boldsymbol{Z}(t) = \dot{\boldsymbol{V}}(t) = \ddot{\boldsymbol{X}}(t)$, and $\mathbf{f}_{\boldsymbol{\phi}}(\boldsymbol{Y}(t)) = \mathbf{f}_{\phi^E, \phi^A}(\boldsymbol{Y}(t))$ represents the sum of energy and alignment-based interactions as in (1.1).

Table 1
Notations for second-order systems

Variable	Definition
N	number of agents
m_i	mass of agent i
$\boldsymbol{x}_i(t) \in \mathbb{R}^d$	position vector of agent i at time t
$\dot{\boldsymbol{x}}_i(t) \in \mathbb{R}^d$	velocity vector of agent i at time t
$\ddot{\boldsymbol{x}}_i(t) \in \mathbb{R}^d$	acceleration vector of agent i at time t
F	non-collective force
$\boldsymbol{\alpha}$	parameters of F
ϕ^E, ϕ^A	energy and alignment-based interaction kernels respectively
$\|\cdot\|$	Euclidean norm in $\mathbb{R}^d$

1.1. Data-driven model selection problem. Recent advancements in data information technology, such as digital imaging, high-resolution lightweight GPS devices, and particle tracking methods, have allowed for the gathering of high-resolution trajectory data of individual particles in various applications. However, a significant issue that remains scarcely addressed is how to select models that match the observational data. For example, while there are many theoretical models known to reproduce flocking patterns, it is challenging to determine which one generates the pattern observed in the data. Previous theoretical and numerical studies cannot address this problem, as predetermined governing equations are needed, and the aim is often to reproduce qualitative rather than quantitative dynamics.

To address this issue, we consider the data-driven model selection problem, aiming to select possible models from a general form to match the observational data. For instance, given the motion data of a school of fish, we aim to determine whether to use first-order or second-order models and which types of interactions, such as alignment versus energy-based or both, contribute to collective patterns. These are challenging questions that practitioners typically address based on their expertise in the field. In this paper, we seek to develop data-driven methods to automate this step by considering a general model that incorporates many classical models as special cases.

Mathematically, we formulate the problem as follows. Given approximate observations of multiple trajectory data $\mathcal{D}_{M,L} := \{\boldsymbol{Y}^{(m)}(t_l), \boldsymbol{Z}^{(m)}(t_l)\}_{m,l=1}^{M,L}$, where the observation time instances are denoted by $0 = t_1 < \cdots < t_L = T$ and m denotes the trial number of experiments starting from different initial conditions, the goal is to infer the interaction kernels $\boldsymbol{\phi} = \phi^E, \phi^A$ as well as the unknown scalar parameters $\boldsymbol{\alpha}$ and possibly $\boldsymbol{m}$ from the trajectory data $\mathcal{D}_{M,L}$. Subsequently, we use the learned governing equations to make predictions about future events or simulate new datasets.

1.2. Scalable Model Selection by Gaussian processes. The field of data-driven model selection faces two primary practical challenges. Firstly, there is often limited information available on the parametric forms of interaction kernels, making it difficult to select a suitable approximation dictionary. Secondly, datasets may be scarce and noisy. Gaussian process (GP) based approaches in machine learning offer a solution to these challenges, as they are known for their ability to learn a rich class of nonlinear functions without making assumptions about their parametric form and for quantifying the associated uncertainty. However, the challenge of scalability to large-scale problems remains a significant hurdle for specific applications.

This paper proposes a novel approach to address these challenges by leveraging the inference power of Gaussian processes and developing efficient techniques to improve scalability. Computationally,

- We propose a novel method by modeling interaction kernels as two independent Gaussian processes to learn (1.1) from data with uncertainty quantification. We investigate whether types of interaction kernels and order information (first versus second order) of the system can be learned from scarce noisy data. We conduct intensive numerical experiments on various prototypical systems exhibiting clustering, milling, and flocking behaviors that demonstrate the effectiveness.
- We propose effective acceleration techniques based on the recent progress from randomized numerical linear algebra.

- Our method is applied to modeling two real-world fish motion sets that display flocking and milling patterns up to 248 dimensions and outperforms competitor methods that use SINDy and feed-forward neural networks.

Theoretically,

- We derive a Representer theorem that connects the GP-based estimators with the kernel ridge regression estimators, shedding light on the role of the hyperparameters in learning. It also provides a basis representation for the estimators of interaction kernels, which enables efficient trajectory prediction using learned models over larger time intervals.
- We study the well-posedness of the inverse problem for learning interaction kernels in a statistical setting.

1.3. Relevant works. Integrating machine learning techniques into the data-driven discovery of dynamical systems (see e.g. [14, 5, 15, 16, 6, 17, 18, 7, 8, 9, 19, 20, 21]) has become a hot topic in scientific machine learning, as it provides powerful models to represent the complex functional data. In terms of parametric methods, one can refer to [22] (and references therein) for the most recent survey on deep learning techniques and [15, 23, 24, 25] for sparse regression techniques.

Gaussian process regression (GPR) is a non-parametric Bayesian machine learning technique for supervised learning with a built-in quantification of uncertainty framework. As such, GPs have been applied to learn ODEs, SDEs, and PDEs [26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36] and lead to more accurate and robust models of dynamical systems. Because of the distinctive nature of dynamical data, it necessitates novel ideas and nontrivial efforts tailored to particular types of dynamical systems and data regimes. We model the latent interaction kernels as GPs and imbue them with the structure of our governing equations (translation and rotational invariance). This makes our work distinguishable from most works, which model state variables as GPs.

In the context of interaction kernel learning in interacting particle systems, least square estimators derived from the maximum likelihood method are the most frequently used, where a challenge lies in the selection of the basis to represent the interaction kernels. One can refer to [1, 2, 3, 4] for the usage of a piecewise polynomial basis. The random feature method together with sparse regression techniques is recently proposed in [37]. One can also refer to the recent methodology development of interaction kernel/potential learning in mean-field systems such as [38, 39, 40, 41, 42].

In particular, [3, 4] considered learning theory for heterogeneous systems and showed that learning multiple interaction kernels simultaneously is challenging and regularization is necessary.

For scarce noisy data, our GP method leverages the underlying statistical inference power to select the best basis to represent the observational dynamics and provides effective regularization. It yields accurate recovery of the *governing equation* beyond learning only interaction kernels. It is well-known that the non-collective force also plays an important role in determining the collective behaviors. The governing equation recovery makes our method more practical than previous work that only focuses on interaction kernels. Further, we analyze the well-posedness of the inverse problems, which complements the missing analysis in [4].

This work is an extension of our recent work [43] on a single kernel case where we assumed $\phi^A \equiv 0$ and the focus was the theoretical framework for error analysis. Here, we consider a more generalized model involving two types of kernels and consider the model selection problems. The focus is shifted to the computational aspects concerning scalability and uncertainty quantification, and real data applications.

1.4. Notation and preliminaries.

Notation. Let ρ be a Borel positive measure on D dimensional Euclean space $\mathbb{R}^D$. We use $L^2(\mathbb{R}^D;\rho;\mathbb{R}^n)$ to denote the set of $L^2(\rho)$-integrable vector-valued functions that map $\mathbb{R}^D$ to $\mathbb{R}^n$. For a function $\boldsymbol{f} \in L^2(\mathbb{R}^D;\rho;\mathbb{R}^n)$, and a vector $\boldsymbol{X} = [\boldsymbol{x}_1^\top, \cdots, \boldsymbol{x}_m^\top]^T \in \mathbb{R}^{mD}$ with $\boldsymbol{x}_i \in \mathbb{R}^D$, we use the notation $\boldsymbol{f}(\boldsymbol{X})$ to represent the image of the vector under the function of $\boldsymbol{f}$ componentwisely, namely, $\boldsymbol{f}(\boldsymbol{X}) = [\boldsymbol{f}(\boldsymbol{x}_1)^\top, \cdots, \boldsymbol{f}(\boldsymbol{x}_m)^\top]^\top \in \mathbb{R}^{mn}$. Let $\mathcal{S}_1$ be a measurable subset of $\mathbb{R}^m$, the restriction of the measure ρ on $\mathcal{S}_1$, denote by $\rho\llcorner\mathcal{S}_1$, is defined as $\rho\llcorner\mathcal{S}_1(\mathcal{S}_2) = \rho(\mathcal{S}_1 \cap \mathcal{S}_2)$ for any measurable subset $\mathcal{S}_2$ of $\mathbb{R}^D$. We used $\mathcal{N}(0, I_{d\times d})$ to denote the standard multivariate Gaussian distribution in $\mathbb{R}^d$.

Preliminaries on operator algebras. Let $\mathcal{H}_1, \mathcal{H}_2$ be Hilbert spaces. We use $\langle\cdot,\cdot\rangle_{\mathcal{H}_1}$ to denote the inner product over $\mathcal{H}_1$, and still use $\langle\cdot,\cdot\rangle$ to denote the inner product on the Euclidean space. We denote by $\mathcal{B}(\mathcal{H}_1, \mathcal{H}_2)$ the set of bounded linear operators mapping $\mathcal{H}_1$ to $\mathcal{H}_2$. Let $A \in \mathcal{B}(\mathcal{H}_1, \mathcal{H}_2)$, we use $\mathrm{Im}(A)$ to denote its range and $\|A\|$ to denote its operator norm. A is a compact operator if A maps bounded subsets of $\mathcal{H}_1$ to relatively compact subsets of $\mathcal{H}_2$ (subsets with compact closure in $\mathcal{H}_2$). We use $A^* : \mathcal{H}_2 \to \mathcal{H}_1$ to denote the adjoint operator of A, that is, $\forall f \in \mathcal{H}_1$, $g \in \mathcal{H}_2$, $\langle Af, g\rangle_{\mathcal{H}_2} = \langle f, A^*g\rangle_{\mathcal{H}_1}$.

For $d, N, M, L \in \mathbb{N}^+$, let $\boldsymbol{w} = (\boldsymbol{w}_{m,l,i})_{m,l,i=1}^{M,L,N}$, $\boldsymbol{z} = (\boldsymbol{z}_{m,l,i})_{m,l,i=1}^{M,L,N} \in \mathbb{R}^{dNML}$ with $\boldsymbol{w}_{m,l,i}, \boldsymbol{z}_{m,l,i} \in \mathbb{R}^d$, we define

$$\langle \boldsymbol{w}, \boldsymbol{z}\rangle = \frac{1}{MLN}\sum_{m,l,i=1}^{M,L,N} \langle \boldsymbol{w}_{m,l,i}, \boldsymbol{z}_{m,l,i}\rangle, \tag{1.3}$$

where $\langle \boldsymbol{w}_{m,l,i}, \boldsymbol{z}_{m,l,i}\rangle$ is the canonical inner product on $\mathbb{R}^d$.

Then for vectors $\boldsymbol{y} \in \mathbb{R}^{mdN}$ and functions $\boldsymbol{g} : \mathbb{R}^{mdN} \to \mathbb{R}^{dn}$ for some $m, n \in \mathbb{N}^+$, and let ρ be a measure at a Borel subset $\mathcal{Y}$ of $\mathbb{R}^{mdN}$, we have the norm:

$$\|\boldsymbol{g}(\boldsymbol{y})\|_{L^2(\rho)}^2 = \frac{1}{n}\int_{\mathcal{Y}} \sum_{i=1}^{n} \|\boldsymbol{g}_i(\boldsymbol{y})\|^2 \rho(d\boldsymbol{y}) \tag{1.4}$$

where $\boldsymbol{g}(x)$ is componentwise denoted by $\boldsymbol{g}_i(\boldsymbol{y}) : \mathbb{R}^{mdN} \to \mathbb{R}^d$.

For two Borel positive measures ρ_1, ρ_2 defined on $\mathbb{R}^D$, ρ_1 is said to be absolutely continuous with respect to ρ_2, $\rho_1 \ll \rho_2$, if $\rho_1(\mathcal{S}) = 0$ for every set $\rho_2(\mathcal{S}) = 0$, $\mathcal{S} \subset \mathbb{R}^D$. ρ_1 and ρ_2 are called equivalent iff $\rho_1 \ll \rho_2$ and $\rho_2 \ll \rho_1$. The product measure $\rho_1 \times \rho_2$ is defined to be a measure on $\mathbb{R}^{2D}$ satisfying the property $(\rho_1 \times \rho_2)(\mathcal{S}_1 \times \mathcal{S}_2) = \rho_1(\mathcal{S}_1)\rho_2(\mathcal{S}_2)$ for all subsets $\mathcal{S}_i \subset \mathbb{R}^D$, $i = 1, 2$.

Preliminaries on GPs (Gaussian Processes) Prior . We say $\phi \sim \mathcal{GP}(u, K)$ to denote our prior on ϕ. In particular, this means that for any $r \in \mathbb{R}$, the random variable $\phi(r)$ is Gaussian: $\phi(r) \sim \mathcal{N}(u(r), K(r,r))$, where $\mathcal{N}$ denotes the normal or multivariable normal

distributions, $u : \mathbb{R} \to \mathbb{R}$ is the mean function, and $K : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ is the covariance function. Similarly, for any $(r, r') \in \mathbb{R}^2$, the joint distribution of $\begin{bmatrix} \phi(r) \\ \phi(r') \end{bmatrix}$ is multivariate Gaussian: $\begin{bmatrix} \phi(r) \\ \phi(r') \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} (u(r) \\ (u(r') \end{bmatrix}, \begin{bmatrix} K(r,r) & K(r,r') \\ K(r',r) & K(r',r') \end{bmatrix} \right)$. This extends in a natural way to any finite set $(r_1, \ldots, r_N) \in \mathbb{R}^N$.

Preliminaries on RKHSs. Let $\mathcal{D}$ be a compact subset of $\mathbb{R}^D$. We say that $K : \mathcal{D} \times \mathcal{D} \to \mathbb{R}$ is a Mercer kernel if it is continuous, symmetric, and positive semidefinite, i.e., for any finite set of distinct points $\{x_1, \cdots, x_M\} \subset \mathcal{D}$, the matrix $(K(x_i, x_j))_{i,j=1}^M$ is positive semidefinite. For $x \in \mathbb{R}^D$, K_x is a function defined on $\mathcal{D}$ such that $K_x(y) = K(x, y)$, $y \in \mathcal{D}$. The Moore–Aronszajn theorem proves that there is an RKHS $\mathcal{H}_K$ associated with the kernel K, which is defined to be the closure of the linear span of the set of functions $\{K_x : x \in \mathcal{D}\}$ with respect to the inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}_K}$ satisfying $\langle K_x, K_y \rangle_{\mathcal{H}_K} = K(x, y)$. Let K be a Mercer kernel that is defined on $[0, R] \times [0, R]$ and use $\mathcal{H}_K$ to denote the RKHS associated with K. For two RKHS $\mathcal{H}_{K_1}, \mathcal{H}_{K_2}$, with $K_1, K_2 : \mathcal{D} \times \mathcal{D} \to \mathbb{R}$, the product RKHS $\mathcal{H}_{K_1} \times \mathcal{H}_{K_2}$ is defined to be the closure of the linear span of the set of functions $\{(K_{1,x_1}, K_{2,x_2}) : x_1, x_2 \in \mathcal{D}\}$ with respect to the inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}_{K_1} \times \mathcal{H}_{K_2}}$ satisfying $\langle (K_{1,x_1}, K_{2,x_2}), (K_{1,y_1}, K_{2,y_2}) \rangle_{\mathcal{H}_{K_1} \times \mathcal{H}_{K_2}} = K_1(x_1, y_1) + K_2(x_2, y_2)$.

2. Methodology.

In this section, we propose a learning approach based on GPs for the model selection problem.

2.1. Two independent Gaussian process priors.

We start by modeling the interaction kernel functions ϕ^E and ϕ^A with the priors as two independent Gaussian processes

$$\phi^E \sim \mathcal{GP}(0, K_{\theta^E}(r, r')), \qquad \phi^A \sim \mathcal{GP}(0, K_{\theta^A}(r, r')), \tag{2.1}$$

where K_{θ^E}, K_{θ^A} are covariance functions with hyperparameters $\boldsymbol{\theta} = (\theta^E, \theta^A)$. $\boldsymbol{\theta}$ can either be chosen by the modeler or tuned via a data-driven procedure discussed later.

2.2. Training of hyperparameters via maximum likelihood estimation.

In real-world modeling, it is possible that some other parameters such as $\boldsymbol{\alpha}$ and noise level in the data are unknown. In this section, we detail how to perform the estimation of these physical parameters in the governing equation via a data-driven hyperparameter tuning process induced by the Gaussian process. This flexible training procedure distinguishes the Gaussian process from other kernel-based methods [44, 45, 46] and regularization-based approaches [47, 48, 49].

We organize the training data into the vector format $\mathbb{Y} = [\boldsymbol{Y}^{(1,1)}, \ldots, \boldsymbol{Y}^{(M,L)}]^T \in \mathbb{R}^{dNML}$, and $\mathbb{Z} = [\boldsymbol{Z}^{(1,1)}, \ldots, \boldsymbol{Z}^{(M,L)}]^T \in \mathbb{R}^{dNML}$ where

$$\boldsymbol{Y}^{(m,l)} = \boldsymbol{Y}^{(m)}(t_l), \quad \boldsymbol{Z}^{(m,l)} = \boldsymbol{Z}^{(m)}(t_l). \tag{2.2}$$

To model the noise, we assume $\mathbb{Z} = [\boldsymbol{Z}_{\sigma^2}^{(1,1)}, \ldots, \boldsymbol{Z}_{\sigma^2}^{(M,L)}]^T \in \mathbb{R}^{dNML}$ where

$$\boldsymbol{m}\boldsymbol{Z}_{\sigma^2}^{(m,l)} = F_{\boldsymbol{\alpha}}(\boldsymbol{Y}^{(m,l)}) + \mathbf{f}_{\boldsymbol{\phi}}(\boldsymbol{Y}^{(m,l)}) + \epsilon^{(m,l)}, \tag{2.3}$$

with i.i.d (independent and identically distributed) noise $\epsilon^{(m,l)} \sim \mathcal{N}(0, \sigma^2 I_{dN})$ that is also independent of the Gaussian processes. Later, we will show the role of σ in the prediction step is equivalent to the role of the regularization constant in a Tikhonov regularization problem.

Therefore, based on the properties of Gaussian processes, with the priors of ϕ^E, ϕ^A, we have

$$\boldsymbol{m}\mathbb{Z} \sim \mathcal{N}(F_{\boldsymbol{\alpha}}(\mathbb{Y}), K_{\mathbf{f}_{\boldsymbol{\phi}}}(\mathbb{Y}, \mathbb{Y}; \theta) + \sigma^2 I_{dNML}), \tag{2.4}$$

with the mean vector $F_{\boldsymbol{\alpha}}(\mathbb{Y}) = \mathrm{Vec}(\{F_{\boldsymbol{\alpha}}(\boldsymbol{Y}^{(m,l)})\}_{m,l=1}^{M,L}) \in \mathbb{R}^{dNML}$, and $K_{\mathbf{f}_{\boldsymbol{\phi}}}(\mathbb{Y}, \mathbb{Y}; \theta) \in \mathbb{R}^{dNML \times dNML}$ is the covariance matrix between $\mathbf{f}_{\boldsymbol{\phi}}(\mathbb{Y})$ and $\mathbf{f}_{\boldsymbol{\phi}}(\mathbb{Y})$, which can be computed elementwise based on the covariance functions K_{θ^E}, K_{θ^A}, see Appendix section A for detailed formulas.

Thus, for the hyperparameters $\boldsymbol{\alpha}$, $\boldsymbol{\theta}$, and σ, we can train by maximizing the probability of the observational data, which is equivalent to minimizing the negative log marginal likelihood (NLML) (see Chapter 4 in [50])

$$\begin{aligned} -\log p(\boldsymbol{m}\mathbb{Z}|\mathbb{Y}, \boldsymbol{\alpha}, \boldsymbol{\theta}, \sigma^2) = &\frac{1}{2}(\boldsymbol{m}\mathbb{Z} - F_{\boldsymbol{\alpha}}(\mathbb{Y}))^T (K_{\mathbf{f}_{\boldsymbol{\phi}}}(\mathbb{Y}, \mathbb{Y}; \boldsymbol{\theta}) + \sigma^2 I_{dNML})^{-1} (\boldsymbol{m}\mathbb{Z} - F_{\boldsymbol{\alpha}}(\mathbb{Y})) \\ &+ \frac{1}{2}\log |K_{\mathbf{f}_{\boldsymbol{\phi}}}(\mathbb{Y}, \mathbb{Y}; \boldsymbol{\theta}) + \sigma^2 I_{dNML}| + \frac{dNML}{2}\log 2\pi. \end{aligned} \tag{2.5}$$

Note here the marginal likelihood does not simply favor the models that fit the training data best, but induces an automatic trade-off between data-fit and model complexity. To solve for the hyperparameters $(\boldsymbol{\alpha}, \boldsymbol{\theta}, \sigma)$, we can apply the conjugate gradient (CG) optimization (see Chapter 5 in [50]) to minimize the negative log marginal likelihood. More details are shown in Appendix section A.

Table 2
Notations for covariances

Variable	Definition
$K_{\boldsymbol{\theta}}(\cdot,\cdot)$	covariance kernel function with parameters $\boldsymbol{\theta}$
$K_{\theta^E}(\cdot,\cdot), K_{\theta^A}(\cdot,\cdot)$	covariance kernels for modeling ϕ^E, ϕ^A
$K_{\mathbf{f}_\phi}(\cdot,\cdot)$	covariance matrix between $\mathbf{f}_\phi(\cdot)$ and $\mathbf{f}_\phi(\cdot)$
$K_{\mathbf{f}_\phi,\phi^E}(\cdot,\cdot) := K_{\phi^E,\mathbf{f}_\phi}(\cdot,\cdot)^T$	covariance matrix between $\mathbf{f}_{\boldsymbol{\phi}}(\cdot)$ and $\phi^E(\cdot)$
$K_{\mathbf{f}_\phi,\phi^A}(\cdot,\cdot) := K_{\phi^A,\mathbf{f}_\phi}(\cdot,\cdot)^T$	covariance matrix between $\mathbf{f}_{\boldsymbol{\phi}}(\cdot)$ and $\phi^A(\cdot)$

2.2.1. Parameter m- Model selection of the order for the dynamical system. When modeling real-world dynamics, sometimes we are not sure whether to use first-order or second-order systems. From the parameter estimation perspective, it is equivalent to determining if the mass of particles is equal to zero. We can train $\boldsymbol{m}$ via minimizing (2.5). If the estimation of $\boldsymbol{m}$ is close to zero, we can consider identifying the dynamics as a first-order system. One can refer to subsection 4.3.

Table 3
Notations for second-order systems

Variable	Definition
$\boldsymbol{X} \in \mathbb{R}^{dN}$	vectorization of position vectors $(\boldsymbol{x}_i)_{i=1}^N$
$\boldsymbol{V} \in \mathbb{R}^{dN}$	vectorization of velocity vectors $(\boldsymbol{v}_i)_{i=1}^N = (\dot{\boldsymbol{x}}_i)_{i=1}^N$
$\boldsymbol{Y} \in \mathbb{R}^{2dN}$	$\boldsymbol{Y} = (\boldsymbol{X}, \boldsymbol{V})^T$
$\boldsymbol{Z} \in \mathbb{R}^{dN}$	vectorization of $(\ddot{\boldsymbol{x}}_i)_{i=1}^N$
$\boldsymbol{r}_{ij}^{\boldsymbol{x}}, \boldsymbol{r}_{ij}^{\boldsymbol{x}'} \in \mathbb{R}^d$	$\boldsymbol{X}(t)_j - \boldsymbol{X}(t)_i$, $\boldsymbol{X}(t')_j - \boldsymbol{X}(t')_i$
$\boldsymbol{r}_{ij}^{\boldsymbol{v}}, \boldsymbol{r}_{ij}^{\boldsymbol{v}'} \in \mathbb{R}^d$	$\boldsymbol{V}(t)_j - \boldsymbol{V}(t)_i$, $\boldsymbol{V}(t')_j - \boldsymbol{V}(t')_i$
$r_{ij}^{\boldsymbol{x}}, r_{ij}^{\boldsymbol{x}'} \in \mathbb{R}^+$	$r_{ik}^{\boldsymbol{x}} = \|\boldsymbol{r}_{ik}^{\boldsymbol{x}}\|, r_{ij}^{\boldsymbol{x}'} = \|\boldsymbol{r}_{ij}^{\boldsymbol{x}'}\|$
$\mathbf{f}_{\phi^E}, \mathbf{f}_{\phi^A}$	energy and alignment-based interaction force field
$\mathbf{f}_{\boldsymbol{\phi}}$	interaction force field with $\boldsymbol{\phi} = (\phi^E, \phi^A)$

2.3. Learning interaction kernels. We plug the estimators of hyperparameters obtained in subsection 2.2 into the system and assume they are known. In this subsection, we show how to learn interaction kernels. For any $r^* \in \mathbb{R}$ and the corresponding values of the kernel functions, $\phi^{\text{type}}(r^*)$, type $= E$ or A, since we have

$$\begin{bmatrix} \boldsymbol{m}\mathbb{Z} - F_\alpha(\mathbb{Y}) \\ \phi^{\text{type}}(r^*) \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} K_{\mathbf{f}_{\boldsymbol{\phi}}}(\mathbb{Y},\mathbb{Y}) + \sigma^2 I_{dNML} & K_{\mathbf{f}_{\boldsymbol{\phi}},\phi^{\text{type}}}(\mathbb{Y}, r^*) \\ K_{\phi^{\text{type}},\mathbf{f}_{\boldsymbol{\phi}}}(r^*,\mathbb{Y}) & K_{\theta^{\text{type}}}(r^*, r^*) \end{bmatrix}\right), \tag{2.6}$$

where $K_{\mathbf{f}_{\boldsymbol{\phi}},\phi^{\text{type}}}(\mathbb{Y}, r^*) = K_{\phi^{\text{type}},\mathbf{f}_{\phi}}(r^*,\mathbb{Y})^T$ denotes the covariance matrix between $\mathbf{f}_{\boldsymbol{\phi}}(\mathbb{Y})$ and $\phi^{\text{type}}(r^*)$. Conditioning on $\mathbf{f}_{\boldsymbol{\phi}}(\mathbb{Y})$, we obtain the posterior/predictive distribution for the kernel function value at r^*, $\phi^{\text{type}}(r^*)$ (see Lemma D.3 in Appendix for detailed derivation), i.e.

$$p(\phi^{\text{type}}(r^*)|\mathbb{Y}, \mathbb{Z}, r^*) \sim \mathcal{N}(\bar{\phi}^{\text{type}}, var(\bar{\phi}^{\text{type}})), \tag{2.7}$$

where

$$\bar{\phi}^{\text{type}} = K_{\phi^{\text{type}},\mathbf{f}_{\boldsymbol{\phi}}}(r^*,\mathbb{Y})(K_{\mathbf{f}_{\boldsymbol{\phi}}}(\mathbb{Y},\mathbb{Y}) + \sigma^2 I_{dNML})^{-1}(\boldsymbol{m}\mathbb{Z} - F_{\boldsymbol{\alpha}}(\mathbb{Y})), \tag{2.8}$$

$$var(\bar{\phi}^{\text{type}}) = K_{\theta^{\text{type}}}(r^*, r^*) - K_{\phi^{\text{type}},\mathbf{f}_{\boldsymbol{\phi}}}(r^*,\mathbb{Y})(K_{\mathbf{f}_{\boldsymbol{\phi}}}(\mathbb{Y},\mathbb{Y}) + \sigma^2 I_{dNML})^{-1} K_{\mathbf{f}_{\boldsymbol{\phi}},\phi^{\text{type}}}(\mathbb{Y}, r^*). \tag{2.9}$$

The posterior variance $var(\bar{\phi}^{\text{type}})$ can be used as a good indicator for the uncertainty of the estimation $\bar{\phi}^{\text{type}}$ based on our Bayesian approach.

2.4. Prediction of trajectories and its uncertainty quantification. We use the posterior mean estimators of $\boldsymbol{\phi}$ in trajectory prediction by performing numerical simulations of the equations

$$\boldsymbol{m}\hat{\boldsymbol{Z}}(t) = F_{\hat{\boldsymbol{\alpha}}}(\boldsymbol{Y}(t)) + \hat{\mathbf{f}}_{\bar{\boldsymbol{\phi}}}(\boldsymbol{Y}(t)). \tag{2.10}$$

We can also perform uncertainty quantification for the trajectory prediction via the uncertainty band of $\hat{\boldsymbol{\phi}}$. We adopted a Monte Carlo method, where we used $\boldsymbol{\phi}$ sampled from the

Algorithm 2.1 Learning kernels

Input: $(\mathbb{Y}, \mathbb{Z})$ (training data), r^* (test point), $K_{\boldsymbol{\theta}}$ (covariance functions), $\mathbf{f}_{\boldsymbol{\phi}}$ (interaction function), $F_{\boldsymbol{\alpha}}$ (force function)

1: $(\hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\theta}}, \hat{\sigma}^2) = \underset{\boldsymbol{\alpha},\boldsymbol{\theta},\sigma^2}{\arg\min} \; -\log p(\boldsymbol{m}\mathbb{Z}|\mathbb{Y}, \boldsymbol{\alpha}, \boldsymbol{\theta}, \sigma^2)$
{solve for parameters by minimizing NLML (2.5) using CG}

2: $L := \text{cholesky}(K_{\mathbf{f}_{\boldsymbol{\phi}}}(\mathbb{Y}, \mathbb{Y}) + \hat{\sigma}^2 I)$

3: $\gamma := L^T \backslash (L \backslash (\boldsymbol{m}\mathbb{Z} - F_{\hat{\boldsymbol{\alpha}}}(\mathbb{Y})))$

4: $K_E^* := K_{\mathbf{f}_{\boldsymbol{\phi}},\phi^E}(\mathbb{Y}, r^*)$
$K_A^* := K_{\mathbf{f}_{\boldsymbol{\phi}},\phi^A}(\mathbb{Y}, r^*)$ {compute covariances between $\mathbf{f}_{\boldsymbol{\phi}}(\mathbb{Y})$ and $\phi^E(r^*)$, $\phi^A(r^*)$}

5: $\bar{\phi}^{E*} := (K_E^*)^T \gamma$
$\bar{\phi}^{A*} := (K_A^*)^T \gamma$ {predictive mean (2.8)}

6: $\boldsymbol{v}_E = L \backslash K_E^*$, $\boldsymbol{v}_A = L \backslash K_A^*$

7: $var(\phi^{E*}) := K_{\hat{\theta}^E}(r^*, r^*) - \boldsymbol{v}_E^T \boldsymbol{v}_E$
$var(\phi^{A*}) := K_{\hat{\theta}^A}(r^*, r^*) - \boldsymbol{v}_A^T \boldsymbol{v}_A$ {predictive variance (2.9)}

Output: $\bar{\phi}^{E*}, \bar{\phi}^{A*}$ (mean), $var(\phi^{E*}), var(\phi^{A*})$ (variance)

posterior distribution in each simulation. Then the predictions of the trajectories are given by the mean of the trajectories' samples and the uncertainty band of each trajectory is given by the standard deviation, with the results of experiments shown in section 4. Another possible alternative is to use step-wise uncertainty quantification based on the numerical integrator scheme such as the one-step Euler method. In this case, it is easy to compute the variance of the solution from the posterior distribution of $\boldsymbol{\phi}$, since the vector field $\hat{\mathbf{f}}_{\bar{\boldsymbol{\phi}}}(\boldsymbol{Y}(t))$ is a linear combination of $\hat{\boldsymbol{\phi}}$ by its definition, which suggests it also follows a Gaussian distribution and the uncertainty band can be derived from its covariance matrix.

2.5. Acceleration of the Computation. While the full GP methods described above yield extremely accurate predictions in our empirical examples, a well-known limitation is the computational complexity; calculating the log determinant of $K_{\mathbf{f}_{\boldsymbol{\phi}}}$ and inverting the kernel matrices in the maximum likelihood estimation and prediction steps scales cubically with the matrix dimension, which is $\mathcal{O}((NdML)^3)$. Therefore, the naive approach can quickly become infeasible for large-scale problems. Below, we describe our integrated approach to the scalable estimation of hyperparameters in maximum likelihood estimation and scalable kernel prediction.

2.5.1. Efficient Hyperparameter Optimization. There are many recent advancements in accelerating the hyperparameter learning computations in the full GP methods for regression tasks. Our problem, however, presents many numerical difficulties that dampen runtime gains from traditional computational methods and must be addressed:

- **Lack of sparsity**. Many classical acceleration techniques rely on the sparsity of the kernel matrix $K_{\mathbf{f}_{\boldsymbol{\phi}}}$. As our kernel depends on pairwise distance and our modeling is nonlocal, we do not have a sparse kernel matrix in our formulation. Our method must be able to operate on dense $K_{\mathbf{f}_{\boldsymbol{\phi}}}$.
- **Extreme ill-conditioning and higher accuracy requirements**. The L^2 condition

number of a matrix is the ratio of its maximum and minimum singular values. When much larger than 1, the condition number indicates that a matrix is nearly singular, and thus accuracy-reducing errors in computation will occur. For many problems, such as those addressed in section 5, the kernel matrix $K_{\mathbf{f}_\phi}$ has observed L^2 condition number above 10^{15}. These extremely high condition numbers result in slow and inaccurate computation when using traditional methods. Our problem is also an inverse problem while learning our hyperparameters for $K_{\mathbf{f}_\phi}$ (see subsection 3.1). This is very sensitive to perturbations, especially as our optimization problem for the hyperparameters is generally not convex. We must carefully balance the tradeoff between computational time and accuracy.

We empirically observed the *approximately low-rank structure* of $K_{\mathbf{f}_\phi}$ in various examples. This motivated us to adapt two main classes of algorithms in [51] for acceleration (see pseudocode and additional details in Appendix section B):

- **Preconditioned conjugate gradient (PCG) algorithm**. The PCG algorithm allows us to avoid explicit computation of the inverse matrix $(K_{\mathbf{f}_\phi}(\mathbb{Y},\mathbb{Y};\theta)+\sigma^2 I)^{-1}$ in both MLE and prediction, as well as compute the coefficients needed in the stochastic Lanczos quadrature below. Using preconditioners, a classical numerical technique to lower condition numbers, is a necessity for variance reduction. In addition, we must maintain a low error tolerance for PCG to preserve our accuracy throughout learning. Finding effective preconditioners that are suitable to the unique structure of our kernel matrices is a challenge. We propose using the Random Gaussian Nystrom preconditioner [52] to ensure favorable tradeoffs in running time and accuracy. In our practical implementation, this preconditioner outperformed other low-rank approximation preconditioners and has low construction and inversion costs, see section 5.
- **Stochastic trace estimation** for log determinant acceleration. We utilize the identity:

$$\log\det(K_{\mathbf{f}_\phi}(\mathbb{Y},\mathbb{Y};\theta)+\sigma^2 I) = \log\det(P) + \log\det(P^{-\frac{1}{2}}(K_{\mathbf{f}_\phi}(\mathbb{Y},\mathbb{Y};\theta)+\sigma^2 I)P^{-\frac{1}{2}})$$

When P is chosen to be a preconditioner, this identity can prove highly useful. The Random Gaussian Nystrom preconditioner allows us to efficiently compute $\log\det(P)$, and for the remainder, we use the recently developed variance reduced Hutchinson's Estimator [53] combined with stochastic Lanczos quadrature [51].

Analysis of new computational complexity. PCG can reduce explicit inversion complexity from $\mathcal{O}((NdML)^3)$ to $\mathcal{O}(t(NdML)^2)$, where t is the number of iterations. The stochastic Lanczos quadrature improves log determinant complexity from $\mathcal{O}((NdML)^3)$ to $\mathcal{O}(t\ell(NdML)^2)+\mathcal{O}(\log\det(P))$, where t is both the number of eigenvalues and the number of iterations, ℓ is the number of runs of stochastic Lanczos, and $\mathcal{O}(\log\det(P))$ is the complexity of computing the log determinant of the preconditioner. In practice, we chose $\ell, t << NdML$. This lowers the theoretical complexity of these steps to the quadratic $\mathcal{O}(t\ell(NdML)^2)$. For the Random Gaussian Nystrom preconditioner P with rank r, we have construction in $\mathcal{O}(r^2(NdML)+r^3)$ time, inversion in $\mathcal{O}(r^3)$ time and log determinant in $\mathcal{O}(r)$ time.

3. Theoretical analysis. In this section, we are concerned with two theoretical problems regarding learning interaction kernels in the prediction step. The first one is to understand the role of hyperparameters in the prediction step of the Gaussian process, i.e., θ^E, θ^A, and the Gaussian noise σ. The second one is to study well-posedness as an inverse problem.

As in the prediction step, interaction kernels are the only unknown terms in the equations. We make the following simplification on the form of equations to avoid unnecessary technical hurdles:

$$\ddot{\boldsymbol{X}}(t) = \mathbf{f}_{\boldsymbol{\phi}}(\boldsymbol{Y}(t)) = \mathbf{f}_{\phi^E}(\boldsymbol{X}(t)) + \mathbf{f}_{\phi^A}(\boldsymbol{Y}(t)), \tag{3.1}$$

where the masses of the agents are assumed to be one and non-collective forces are assumed to be zero. Our analysis can be extended to general second-order systems (1.1) with known mass and non-collective force terms with slight modifications.

3.1. The Representer theorem. In the classical regression setting [50], there is an interesting link between GP regression and kernel ridge regression (KRR), where the posterior mean can be viewed as a KRR estimator to solve a regularized least square empirical risk functional. In our setting, we have noisy functional observations of the interaction kernels, i.e., the $\{r_{\mathbb{X}_M}, r_{\mathbb{V}_M}, \mathbb{Z}_{\sigma^2,M}\}$ instead of the pairs $\{r_{\mathbb{X}_M}, \phi^E(r_{\mathbb{X}_M}), \phi^A(r_{\mathbb{X}_M})\}$, where $r_{\mathbb{X}_M}, r_{\mathbb{V}_M} \in \mathbb{R}^{MLN^2}$ are the sets contains all the pairwise distances in $\mathbb{X}_M$, and $\mathbb{V}_M$, i.e.

$$r_{\mathbb{X}_M} = \{r_{ij}^{\boldsymbol{X}^{(m,l)}}\}_{i,j,m,l=1}^{N,N,M,L}, \quad r_{\mathbb{V}_M} = \{r_{ij}^{\boldsymbol{V}^{(m,l)}}\}_{i,j,m,l=1}^{N,N,M,L}, \quad \mathbb{Z}_{\sigma^2,M} = \{\boldsymbol{Z}_{\sigma^2}^{(m,l)}\}_{m,l=1}^{M,L}, \tag{3.2}$$

so we face an inverse problem here, instead of a classical regression problem. Thanks to the linearity of the inverse problem, we can still derive a Representer theorem [54] that helps clarify the role of the hyperparameters.

Assumption 3.1. We assume that K^E and K^A are two Mercer kernels defined on $[0, R] \times [0, R]$ for some $R > 0$. The true interaction functions $\phi^E \in \mathcal{H}_{K^E}$, $\phi^A \in \mathcal{H}_{K^A}$, and

$$\kappa_E^2 = \sup_{r\in[0,R]} K^E(r, r) < \infty,$$

$$\kappa_A^2 = \sup_{r\in[0,R]} K^A(r, r) < \infty.$$

Theorem 3.2 (Representer theorem). *Let K^E and K^A be two Mercer kernels that satisfy Assumption (3.1). Given the training data $\{\mathbb{Y}_M, \mathbb{Z}_{\sigma^2,M}\}$, if the priors $\phi^E \sim \mathcal{GP}(0, \tilde{K}^E)$, $\phi^A \sim \mathcal{GP}(0, \tilde{K}^A)$ with $\tilde{K}^E = \frac{\sigma^2 K^E}{MNL\lambda^E}$, $\tilde{K}^A = \frac{\sigma^2 K^A}{MNL\lambda^A}$ for some $\lambda^E, \lambda^A > 0$, then the posterior mean $\bar{\boldsymbol{\phi}} = (\bar{\phi}^E, \bar{\phi}^A)$ in (2.8) coincides with the minimizer of the regularized empirical risk functional $\mathcal{E}^{\lambda,M}(\cdot)$ on $\mathcal{H}_{K^E} \times \mathcal{H}_{K^A}$ where $\mathcal{E}^{\lambda,M}(\cdot)$ is defined by*

$$\mathcal{E}^{\lambda,M}(\boldsymbol{\varphi}) := \frac{1}{LM} \sum_{l,m=1}^{L,M} \|\mathbf{f}_{\boldsymbol{\varphi}}(\boldsymbol{Y}^{(m,l)}) - \boldsymbol{Z}_{\sigma^2}^{(m,l)}\|^2 + \lambda^E \|\varphi^E\|_{\mathcal{H}_{K^E}}^2 + \lambda^A \|\varphi^A\|_{\mathcal{H}_{K^A}}^2. \tag{3.3}$$

where $\lambda = \{\lambda^E, \lambda^A\}$ and the estimator $\bar{\boldsymbol{\phi}} \in \mathcal{H}_{K^E} \times \mathcal{H}_{K^A}$ can also be represented by

$$\bar{\boldsymbol{\phi}} = \Big(\sum_{r^x \in r_{\mathbb{X}_M}} \hat{c}_{r^x} K_{r^x}^E, \sum_{(r^x, r^v) \in (r_{\mathbb{X}_M} \times r_{\mathbb{V}_M})} \hat{c}_{r^v} K_{r^x}^A\Big), \tag{3.4}$$

with

$$\hat{\mathbf{c}}_{r^x} = \frac{1}{N}\boldsymbol{r}_{\mathbb{X}_M}^T \cdot (K_{\mathbf{f}_{\boldsymbol{\phi}}}(\mathbb{Y}_M, \mathbb{Y}_M) + \lambda^E NMLI_{dNML})^{-1}\mathbb{Z}_{\sigma^2,M},$$
$$\hat{\mathbf{c}}_{r^v} = \frac{1}{N}\boldsymbol{r}_{\mathbb{V}_M}^T \cdot (K_{\mathbf{f}_{\boldsymbol{\phi}}}(\mathbb{Y}_M, \mathbb{Y}_M) + \lambda^A NMLI_{dNML})^{-1}\mathbb{Z}_{\sigma^2,M}, \tag{3.5}$$

where $\hat{\mathbf{c}}_{r^x}$, $\hat{\mathbf{c}}_{r^v}$ *are the vectorizations of* $(\hat{c}_{r^x})_{r^x \in r_{\mathbb{X}_M}}$ *and* $(\hat{c}_{r^v})_{r^v \in r_{\mathbb{V}_M}}$ *respectively,* $r_{\mathbb{X}_M} \times r_{\mathbb{V}_M} \in \mathbb{R}^{MLN^2 \times MLN^2}$ *is the set containing all the pairwise distances in* $\mathbb{X}_M$ *and their associated pairwise distances in* $\mathbb{V}_M$ *as defined in* (3.2), $\boldsymbol{r}_{\mathbb{X}_M}$ *is the block-diagonal matrix defined by* $\mathrm{diag}(\boldsymbol{r}_{\boldsymbol{X}^{(m,l)}}) \in \mathbb{R}^{MLdN \times MLN^2}$ *and* $\boldsymbol{r}_{\boldsymbol{X}^{(m,l)}} = \mathrm{diag}(\{[\boldsymbol{r}_{i1}^{\boldsymbol{X}^{(m,l)}}, \ldots, \boldsymbol{r}_{iN}^{\boldsymbol{X}^{(m,l)}}]\}_{i=1}^N) \in \mathbb{R}^{dN \times N^2}$, *similarly for* $\boldsymbol{r}_{\mathbb{V}_M}$.

Detailed proof of Theorem 3.2 is shown in Appendix section D. From the theorem, it is clear how hyperparameters affect the prediction of interaction kernels: θ^E, θ^A, and σ jointly affect the choice of Mercer kernels and regularization constant, which becomes quite crucial in real data applications (see Figure 5). In (3.5), we also see that the posterior mean estimator ϕ^E lies in the span of basis functions with indices determined by the pairwise distances, and their coefficients are correlated with the basis functions. This is an effect imposed by the structure of the governing equation encoded in $\mathbf{f}_{\boldsymbol{\phi}}$.

3.2. Well-posedness. We are concerned with the nonparametric learning of interaction kernels. That is, we do not assume the parametric form of interaction kernels. In this case, one can not expect to recover the true interaction kernels from finite data as they live in infinite dimensional spaces. Therefore, it is important to ensure one can asymptotically identify the true interaction kernels as the number of observational data snapshots goes to infinity. Otherwise, the empirical estimators from finite data will have limited value as a scientific and predictive tool. Mathematically, we study the well-posedness under a statistical inverse problem setting. We introduce a linear operator $A : \mathcal{H}_{K^E} \times \mathcal{H}_{K^A} \to L^2(\mathbb{R}^{2dN}; \rho_{\boldsymbol{Y}}; \mathbb{R}^{dN})$ defined by

$$A\boldsymbol{\varphi} = \mathbf{f}_{\boldsymbol{\varphi}}, \tag{3.6}$$

where $\mathbf{f}_{\boldsymbol{\varphi}}$ is the right hand side of system (3.1) by replacing $\boldsymbol{\phi}$ with $\boldsymbol{\varphi}$, and $\rho_{\boldsymbol{Y}}$ is the limiting measure on $\mathbb{R}^{2dN}$ that we assume the observational data are sampled i.i.d from. For example, if we assume that the initial condition of each trial is sampled i.i.d from a measure, then

$$\rho_{\boldsymbol{Y}}(S) = \lim_{M\to\infty} \frac{1}{M} \sum_{m,l=1}^{M,L} \mathbb{1}_{\boldsymbol{Y}^{(m,l)} \in S} \tag{3.7}$$

for any Borel set $S \subset \mathbb{R}^{2dN}$ and the limit does exist in the weak sense by the law of large numbers. We denote the marginal probability measures for $\boldsymbol{X}$ and $\boldsymbol{V}$ by $\rho_{\boldsymbol{X}}$, $\rho_{\boldsymbol{V}}$ respectively.

Then the well-posedness of (3.6) is reduced to studying under which conditions A has a bounded inverse.

3.2.1. Well-posedness on an L^2 space. We first consider the embedding of $\mathcal{H}_{K^E} \times \mathcal{H}_{K^A}$ to a suitable L^2 space and consider the well-posedness in a weaker L^2-norm. Motivated by (3.4) in the Representer theorem, we consider the measures $\tilde{\rho}_r^E$, $\tilde{\rho}_r^A$ for ϕ^E, ϕ^A based on the structure of $\mathbf{f}_{\boldsymbol{\varphi}}$,

$$\tilde{\rho}_r^E(Q) = \int_Q \int_{\mathbb{R}^{dN}} \frac{1}{N(N-1)} \sum_{i \neq j} \delta_{r_{ij}^{\boldsymbol{x}^*}}(r) \cdot (r_{ij}^{\boldsymbol{x}^*})^2 d\rho_{\boldsymbol{X}}(\boldsymbol{X}^*) dr \tag{3.8}$$

$$\tilde{\rho}_r^A(Q) = \int_Q \int_{\mathbb{R}^{2dN}} \frac{1}{N(N-1)} \sum_{i \neq j} \delta_{r_{ij}^{\boldsymbol{x}^*}}(r) \cdot (r_{ij}^{\boldsymbol{v}^*})^2 d\rho_{\boldsymbol{Y}}(\boldsymbol{X}^*, \boldsymbol{V}^*) dr \tag{3.9}$$

for any set $Q \subset [0, R]$, and $\delta(\cdot)$ is the Dirac δ distribution. By the continuity, $\mathcal{H}_{K^E} \times \mathcal{H}_{K^A}$ can be naturally embedded as a subspace of $L^2([0, R] \times [0, R]; \tilde{\rho}_r; \mathbb{R} \times \mathbb{R})$ with $\tilde{\rho}_r = \tilde{\rho}_r^E \times \tilde{\rho}_r^A$. One can follow the proof of Proposition 9 in [43] to show that A is a bounded linear operator from $L^2([0, R] \times [0, R]; \tilde{\rho}_r; \mathbb{R} \times \mathbb{R})$ to $L^2(\mathbb{R}^{2dN}; \rho_{\boldsymbol{Y}}; \mathbb{R}^{dN})$.

Now we can introduce a sufficient condition to guarantee the existence of a bounded inverse of A on $L^2([0, R] \times [0, R]; \tilde{\rho}_r; \mathbb{R} \times \mathbb{R})$, called the coercivity condition:

Definition 3.3. *We say that the system* (3.1) *satisfies the coercivity condition if* $\forall \boldsymbol{\varphi} \in \mathcal{H}_{K^E} \times \mathcal{H}_{K^A}$,

$$\|A\boldsymbol{\varphi}\|^2_{L^2(\rho_{\boldsymbol{Y}})} = \|\mathbf{f}_{\boldsymbol{\varphi}}\|^2_{L^2(\rho_{\boldsymbol{Y}})} \geq c_{\mathcal{H}_{K^E}} \|\varphi^E\|^2_{L^2(\tilde{\rho}_r^E)} + c_{\mathcal{H}_{K^A}} \|\varphi^A\|^2_{L^2(\tilde{\rho}_r^A)} \tag{3.10}$$

for some constants $c_{\mathcal{H}_{K^E}}, c_{\mathcal{H}_{K^A}} > 0$.

Here we show one example to support the coercivity condition.

Theorem 3.4. *Consider* $\rho_{\boldsymbol{Y}} = \begin{bmatrix} \rho_{\boldsymbol{X}} \\ \rho_{\boldsymbol{V}} \end{bmatrix}$, *where* $\rho_{\boldsymbol{X}}$ *is the product of* N *independent and identical measures with compact support on* $\mathbb{R}^d$, *and* $\rho_{\boldsymbol{V}}$ *is defined in the same way and is independent of* $\rho_{\boldsymbol{X}}$. *Then we have*

$$\|\mathbf{f}_{\boldsymbol{\varphi}}\|^2_{L^2(\rho_{\boldsymbol{Y}})} \geq \frac{N-1}{N^2} \|\varphi^E\|^2_{L^2(\tilde{\rho}_r^E)} + \frac{N-1}{N^2} \|\varphi^A\|^2_{L^2(\tilde{\rho}_r^A)} \tag{3.11}$$

Detailed proof of Theorem 3.4 is shown in Appendix section C. In [4], the identifiability of a structured sum of ϕ^E and ϕ^A is studied. Here we consider a stronger version of identifiability as we want to individually recover ϕ^E and ϕ^A. Note that it is also possible for distributions on $\mathbb{R}^{dN}$ with non-i.i.d $\mathbb{R}^d$ components that satisfy the coercivity condition. Finally, we remark that the coercivity condition (3.10) holds on measure pairs (ρ_1, ρ_2) equivalent to $(\tilde{\rho}_r^E, \tilde{\rho}_r^A)$. This can provide us with many nontrivial examples from the special case in Theorem 3.4. We conjecture that the coercivity condition is generally satisfied and leave further investigation as future work.

3.2.2. Well-posedness on $\mathcal{H}_{K^E} \times \mathcal{H}_{K^A}$ and the convergence analysis. Now we turn to study the well-posedness on $\mathcal{H}_{K^E} \times \mathcal{H}_{K^A}$ with the stronger RKHS norm, and we make the following assumption.

Assumption 3.5. We assume that $\tilde{\rho}_r$ is non-degenerate on $[0, R] \times [0, R]$.

We remark that the above assumption is mild. For example, we can pick $\rho_{\boldsymbol{Y}}$ to be a uniform measure supported on a large enough cube, then $\tilde{\rho}_r$ satisfies the assumption.

It is straightforward to see that the coercivity condition implies injectivity of A on $\mathcal{H}_{K^E} \times \mathcal{H}_{K^A}$: $\boldsymbol{\varphi} = 0$ everywhere on $[0, R]$ when $A\boldsymbol{\varphi} = 0$ for $\boldsymbol{\varphi} \in \mathcal{H}_{K^E} \times \mathcal{H}_{K^A}$. This is due to the non-degeneracy of $\tilde{\rho}_r$ on $[0, R] \times [0, R]$ and the continuity of $\boldsymbol{\varphi}$. Therefore, A is injective. However, showing A has a bounded inverse on $\mathcal{H}_{K^E} \times \mathcal{H}_{K^A}$ is impossible when it is infinitely dimensional, as A is a compact operator. Suppose the coercivity condition (3.10) holds, then following the theoretical framework developed in [43], one could prove the well-posedness on a suitable subspace determined by the source conditions on ϕ^E, ϕ^A following inverse problem literature. In this case, it is possible to prove one could recover both kernels with a statistically optimal rate under the corresponding RKHS norm. We obtained the result for the single-kernel case in our recent work [43], and we leave the work for the double-kernel case for the future investigation.

4. Numerical examples. In this section, we investigate the performance of the algorithm proposed in section 2 to show the effectiveness of model selection in (1.1). Specific instances of (1.1) have found many applications in modeling the clustering, swarming, and alignment behaviors of collective agents. The examples include (1) Cucker-Smale dynamics (CS) with friction force ($m_i \equiv 1$, $\phi^E \equiv 0$, $\phi^A \neq 0$) in subsection 4.2.1, (2) fish milling dynamics (FM) with friction force ($m_i \equiv 1$, $\phi^E \neq 0$, $\phi^A \equiv 0$) in subsection 4.2.2, (3) anticipation dynamics (AD) ($m_i \equiv 1$, $\phi^E, \phi^A \neq 0$) in subsection 4.2.3 and (4) opinion dynamics (OD) with stubborn agents ($m_i \equiv 0$, $\phi^E \neq 0$, $\phi^A \equiv 0$) in subsection 4.3. In (1)-(3), the mass of agents is known in advance, i.e, they are second-order systems. We are interested in learning ϕ^E, ϕ^A, and other hyperparameters $\boldsymbol{\alpha}$ from data, resulting in the selection of types of interactions (energy versus alignment interactions). In (4), we used the prior knowledge that $\phi^A \equiv 0$ and investigate if the true zero mass of the opinions and ϕ^E can be learned from data, resulting in the selection of the order of the system (first versus second order).

The detailed setups of each dynamic are shown in Table 4. We applied the strategies proposed in section 2 to learn $\boldsymbol{\alpha}$ in $F_{\boldsymbol{\alpha}}$, and the interaction kernels $\phi^E(r)$, $\phi^A(r)$. We initialize the parameters in $\boldsymbol{\alpha}$ randomly from the uniform distribution $\mathcal{U}([0, 1])$, and the same for σ in the cases with noisy data. In each experiment, we run 10 independent trials and report the errors of the estimations for $\boldsymbol{\alpha}$, the estimation errors for ϕ^E, ϕ^A in the (relative) $L^\infty([0, R])$-norm, and compare the discrepancy between the true trajectories (evolved using α, ϕ^E, ϕ^A) and predicted trajectories (evolved using $\hat{\alpha}$, $\hat{\phi^E}$, $\hat{\phi^A}$) on both the training time interval $[0, T]$ and on the future time interval $[T, T_f]$, over two different sets of initial conditions (IC) – one taken from the training data, and one consisting of new samples from the same initial distribution.

Real data application. We also apply our method to two real datasets of fish in subsection 4.4, where one shows a flocking behavior and another shows a milling behavior. We fit

them into the Cucker-Smale and fish milling dynamics respectively and perform comparisons with two other classical approaches: SINDy [55] and feed-forward neural networks.

Numerical Setup.. We simulate the trajectory data $(\mathbb{Y}, \mathbb{Z})$ on the time interval $[0, T]$ with given i.i.d initial conditions generated from the probability measures specified for each system as shown in Table 4. For the training data sets, we generate M trajectories and observe each trajectory at L equidistant times $0 = t_1 < t_2 < \cdots < t_L = T$ and add Gaussian noise to $\mathbb{Z}$ with level σ. We construct an empirical approximation to the probability measure $\tilde{\rho}_r$, with 2000 trajectories and let $[0, R]$ be its support. All ODE systems are evolved using ode15s in MATLAB® with a relative tolerance at 10^{-5} and absolute tolerance at 10^{-6}. For noise-free training data, we add a jitter constant $\approx 10^{-6}$ as a way of regularization. We apply the *minimize* function in the GPML package* to train the parameters using conjugate gradient optimization with the partial derivatives shown in section 2, and set the maximum number of function evaluations to 400.

In almost all examples, we use the full GP methods, as we use scarce data and there is no need for acceleration. However, we show the effectiveness of our acceleration techniques in Fish milling dynamics in section 5 when we have a larger scale of data.

Table 4
System parameters in the dynamics

System	CS	FM	AD	OD
d	2	2	2	1
N	10	10	10	5
m_i	1	1	1	0
$[0; T; T_f]$	$[0; 10; 20]$	$[0; 5; 10]$	$[0; 10; 20]$	$[0; 2; 20]$
$\mu_0^{\boldsymbol{x}}$	$\mathrm{Unif}([-2, 2]^2)$	$\mathrm{Unif}([-0.5, 0.5]^2)$	$\mathrm{Unif}([0, 5]^2)$	$\mathrm{Unif}([-1, 1])$
$\mu_0^{\boldsymbol{v}}$	$\mathrm{Unif}([-1, 1]^2)$	$\mathrm{Unif}([0, 0]^2)$	$\mathrm{Unif}([0, 5]^2)$	-
ϕ^E	0	$\frac{1}{r}\left[-e^{-2r} + e^{-\frac{r}{4}}\right]$,	$\frac{0.1}{(1+r)^{2.5}} + \frac{1}{(1+r)^{0.5}}$	(4.4)
ϕ^A	$\frac{1}{(1+r^2)^{1/4}}$	0	$\frac{0.1}{(1+r^2)^{0.5}}$	0
$F(\boldsymbol{x}_i, \dot{\boldsymbol{x}}_i, \boldsymbol{\alpha})$	$\kappa\dot{\boldsymbol{x}}_i(1 - \|\dot{\boldsymbol{x}}_i\|^p)$	$(\gamma - \beta\|\dot{\boldsymbol{x}}_i\|^2)\dot{\boldsymbol{x}}_i$	0	(4.5)
$\boldsymbol{\alpha}$	$(\kappa, p) = (1, 2)$	$(\gamma, \beta) = (1.5, 0.5)$	-	$(P_1, \kappa) = (1, 10)$

Choice of the covariance function.. We choose the Matérn covariance function defined on $[0, R] \times [0, R]$ for the Gaussian process priors in our numerical experiments, i.e.,

$$K_\theta(r, r') = s_\phi^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu}\|r - r'\|}{\omega_\phi}\right)^\nu B_\nu\left(\frac{\sqrt{2\nu}\|r - r'\|}{\omega_\phi}\right), \tag{4.1}$$

where the parameter $\nu > 0$ determines the smoothness; $\Gamma(\nu)$ is the Gamma function; B_ν is the modified Bessel function of the second kind; and the hyperparameters $\theta = \{s_\phi^2, \omega_\phi\}$ quantify the amplitude and scale. In our numerical examples, we choose $\nu = p + 1/2$ with $p = 0$ or 1.

*Carl Edward Rasmussen & Hannes Nickisch (http://gaussianprocess.org/gpml/code)

The Reproducing Kernel Hilbert Space (RKHS), $\mathcal{H}_{Matérn}$, associated with this Matérn kernel is norm-equivalent to the Sobolev space $W_2^{\nu+1/2}([0,R])$ defined by

$$W_2^{\nu+1/2}([0,R]) := \Big\{ f \in L_2([0,R]) : \|f\|^2_{W_2^{\nu+1/2}} := \sum_{\beta \in \mathbb{N}_0^1 : |\beta| \le \nu+1/2} \|D^\beta f\|^2_{L_2} < \infty \Big\}. \tag{4.2}$$

That is to say, $\mathcal{H}_{Matérn} = W_2^s([0,R])$ as a set of functions, and there exists constants $c_1, c_2 > 0$ such that

$$c_1 \|f\|_{W_2^{\mu+\frac{1}{2}}} \le \|f\|_{\mathcal{H}_{Matérn}} \le c_2 ||f||_{W_2^{\mu+\frac{1}{2}}}, \quad \forall f \in \mathcal{H}_{Matérn}. \tag{4.3}$$

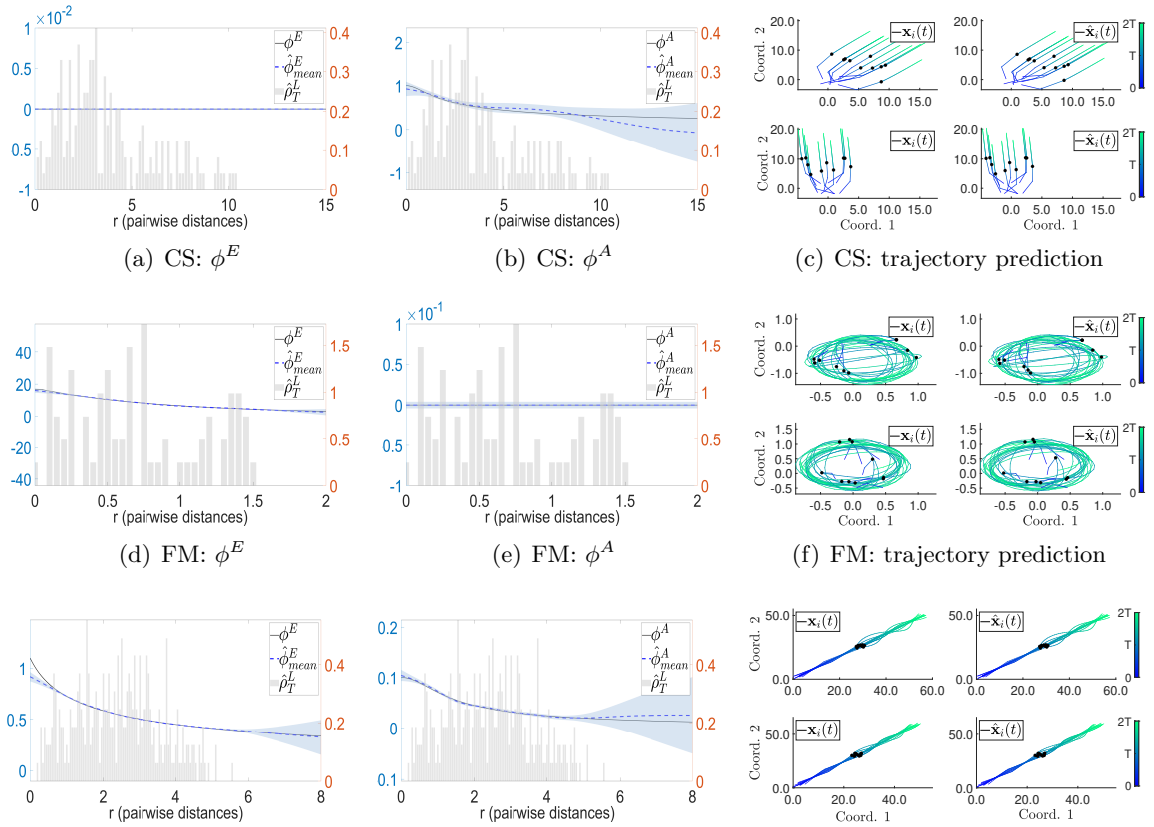
In other words, $\mathcal{H}_{Matérn}$ consists of functions that are differentiable up to order ν and weak differentiable up to order $s = \nu + \frac{1}{2}$.

4.1. Summary of the numerical experiments.

- The proposed learning approach performs *simultaneous* precise model selections from *small* amounts of *noisy* observation data. The numerical results in all different dynamics show that the algorithm can accurately identify the existence of energy-based/alignment-based interactions and can learn order information of dynamics between agents in the systems.
- The GP method selects a kernel basis to represent the underlying sparse dynamics that generalizes remarkably well in larger time prediction with new initial conditions. The occasional larger prediction errors that occur in a larger time interval may be caused by the propagation of estimation errors. We believe the performance is satisfactory since we only have very limited and noisy training data. Even in cases where the prediction errors are relatively large, the estimators can predict remarkably accurate collective behaviors of the agents, e.g. the consensus in the opinion dynamics, the flocking behavior in the Cucker-Smale dynamics, and the milling pattern in the fish milling dynamics.
- In synthetic experiments, the uncertainty quantification band for the trajectories is rather small ($\mathcal{O}(10^{-3})$), resulting from the narrow uncertainty bands of $\boldsymbol{\phi}$. In real data experiments, we found models using interaction kernels sampled from uncertainty bands all reproduced the true dynamics very well.
- The real data experiments show that the proposed GP approach combined with the particle-based models is practically applicable, and outperformed two other competitors in preserving the physics of the true dynamics.

4.2. Model selection for types of interaction kernels.

4.2.1. Cucker-Smale dynamics with friction force. The Cucker-Smale system [56, 57, 58] is used to model collective behaviors in a system of agents that follow a prescribed protocol of communication, such as wedges of bird flocks, lattices in cell organization, or bee hives [59, 60, 61]. We consider the system of N agents in the form (1.1) with components defined in Table 4, where ϕ^A is a communication kernel, or influence function, that makes the agents flock, and $F_{\boldsymbol{\alpha}}$ a Rayleigh-type friction force that pushes all magnitudes of the velocities $\|v_i\|$



(a) CS: ϕ^E (b) CS: ϕ^A (c) CS: trajectory prediction

(d) FM: ϕ^E (e) FM: ϕ^A (f) FM: trajectory prediction

(g) AD: ϕ^E (h) AD: ϕ^A (i) AD: trajectory prediction

Figure 1. *Results of learning different dynamics using the Matérn kernel. Top: Learning CSF* ($\{N, M, L, \sigma\} = \{10, 6, 3, 0.1\}$)*; Middle: Learning FM* ($\{N, M, L, \sigma\} = \{10, 3, 3, 0.1\}$)*; Bottom: Learning AD* ($\{N, M, L, \sigma\} = \{10, 12, 3, 0.01\}$)*. Left, Center: Predictive mean* $\hat{\phi^E}$ *and* $\hat{\phi^A}$ *of the true kernels, and two-standard-deviation band (light blue color) around the means. The grey bars represent the empirical density of the* $\tilde{\rho}_r$*; Right: the true (left) versus predicted (right) trajectories using* $\hat{\boldsymbol{\alpha}}$ *and* $\hat{\phi}$ *with initial conditions of training data (top) and testing data (bottom)*

towards the same value 1 and counteracts the directional alignment forces governed by ϕ^A to produce a rich variety of collective dynamics depending on the relative strengths of the involved forces.

In this example, the unknown parameters $\boldsymbol{\alpha} = (\kappa, p)$ are nonlinear with respect to the system. We show the errors of our estimation for $\boldsymbol{\alpha}$ and ϕ^E, ϕ^A in Table 5. Note that for this model, ϕ^A is in the RKHS generated by the Matérn kernel we pick. The estimated interaction kernel $\hat{\phi}^A$ can recover the true $\phi^A(r)$ almost perfectly in the region within the support of the empirical $\tilde{\rho}_r$ from both noise-free and noisy training data. Moreover, the true interaction kernel $\phi^A(r)$ is fully covered in the uncertainty region we constructed using the posterior variances. Table 5 also shows that our method can identify the nonexistence of the energy-based interaction well with small errors (at most $O(10^{-4})$) from zero in $L^\infty([0, R])$-norm. See also in Figure 1(a),(b). The errors for the predicted trajectories are shown in Table 6. We

can see that in both the training time interval $[0, 10]$ and future time interval $[10, 20]$, the estimators can produce accurate approximations of the true trajectories and the performance becomes better when we increase the size of training data (M or L).

Table 5

Means and standard deviations of the errors of $\hat{\boldsymbol{\alpha}}$ (including $\hat{\sigma}$ when noise exists) and $\hat{\phi}$ for different settings of the CS dynamics.

$\{N, M, L, \sigma\}$	$\|\hat{\boldsymbol{\alpha}} - \boldsymbol{\alpha}\|_\infty$	$\|\hat{\phi^E} - 0\|_\infty$	$\|\hat{\phi^A} - \phi^A\|_\infty / \|\phi^A\|_\infty$
$\{10, 1, 3, 0\}$	$1.9 \cdot 10^{-3} \pm 1.0 \cdot 10^{-3}$	$2.1 \cdot 10^{-5} \pm 4.0 \cdot 10^{-5}$	$5.6 \cdot 10^{-2} \pm 1.5 \cdot 10^{-2}$
$\{10, 3, 3, 0\}$	$\mathbf{1.1 \cdot 10^{-3} \pm 7.9 \cdot 10^{-4}}$	$2.6 \cdot 10^{-5} \pm 6.5 \cdot 10^{-5}$	$4.5 \cdot 10^{-2} \pm 2.0 \cdot 10^{-2}$
$\{10, 6, 3, 0\}$	$1.3 \cdot 10^{-3} \pm 2.5 \cdot 10^{-3}$	$\mathbf{1.1 \cdot 10^{-5} \pm 1.3 \cdot 10^{-5}}$	$\mathbf{3.2 \cdot 10^{-2} \pm 1.0 \cdot 10^{-2}}$
$\{10, 6, 3, 0.05\}$	$1.1 \cdot 10^{-1} \pm 1.1 \cdot 10^{-1}$	$1.2 \cdot 10^{-4} \pm 1.6 \cdot 10^{-4}$	$1.6 \cdot 10^{-1} \pm 8.6 \cdot 10^{-2}$
$\{10, 6, 3, 0.1\}$	$2.3 \cdot 10^{-1} \pm 2.3 \cdot 10^{-1}$	$1.4 \cdot 10^{-4} \pm 2.9 \cdot 10^{-4}$	$1.8 \cdot 10^{-1} \pm 8.0 \cdot 10^{-2}$

Table 6

The trajectory prediction errors for different settings.

$\{N, M, L, \sigma\}$	Training IC $[0, 10]$	Training IC $[10, 20]$	new IC $[0, 10]$	new IC $[10, 20]$
$\{10, 1, 3, 0\}$	$4.9 \cdot 10^{-4} \pm 4.2 \cdot 10^{-4}$	$6.7 \cdot 10^{-4} \pm 1.3 \cdot 10^{-3}$	$1.8 \cdot 10^{-3} \pm 4.4 \cdot 10^{-3}$	$1.4 \cdot 10^{-2} \pm 4.2 \cdot 10^{-2}$
$\{10, 3, 3, 0\}$	$2.5 \cdot 10^{-4} \pm 2.0 \cdot 10^{-4}$	$1.5 \cdot 10^{-4} \pm 1.3 \cdot 10^{-4}$	$4.9 \cdot 10^{-4} \pm 4.9 \cdot 10^{-4}$	$8.7 \cdot 10^{-3} \pm 1.7 \cdot 10^{-2}$
$\{10, 6, 3, 0\}$	$\mathbf{1.5 \cdot 10^{-4} \pm 1.2 \cdot 10^{-4}}$	$\mathbf{9.4 \cdot 10^{-5} \pm 9.2 \cdot 10^{-5}}$	$\mathbf{2.7 \cdot 10^{-4} \pm 4.1 \cdot 10^{-4}}$	$\mathbf{2.3 \cdot 10^{-4} \pm 4.6 \cdot 10^{-4}}$
$\{10, 6, 3, 0.05\}$	$2.3 \cdot 10^{-2} \pm 1.3 \cdot 10^{-2}$	$1.9 \cdot 10^{-2} \pm 1.3 \cdot 10^{-2}$	$2.7 \cdot 10^{-2} \pm 1.9 \cdot 10^{-2}$	$2.5 \cdot 10^{-2} \pm 2.0 \cdot 10^{-2}$
$\{10, 6, 3, 0.1\}$	$4.2 \cdot 10^{-2} \pm 2.6 \cdot 10^{-2}$	$3.8 \cdot 10^{-2} \pm 2.8 \cdot 10^{-2}$	$4.9 \cdot 10^{-2} \pm 3.4 \cdot 10^{-2}$	$4.5 \cdot 10^{-2} \pm 3.9 \cdot 10^{-2}$

4.2.2. Fish-Milling dynamics with friction force. In this subsection, we consider another type of cohesive collective system that produces milling patterns [62, 63]. A special instance of such systems is the D'Orsogna model [12, 61, 64], which describes the motion of N self-propelled particles powered by biological or mechanical motors, that experience a frictional force, and can produce a rich variety of collective patterns. We consider the system of N agents of the form (1.1) with components defined in Table 4, where the interaction kernel ϕ^E is derived from the Morse-type potential. Since it is singular at $r = 0$, we truncate it at $r_0 = 0.05$ with a function of the form ae^{-br} to ensure the new function has a continuous derivative. The force function $F^{\boldsymbol{v}}$ includes self-propulsion with strength γ and nonlinear drag with strength β.

The errors of the estimations for $\boldsymbol{\alpha}$ after our training procedure and the learned ϕ^E, ϕ^A are shown in Table 7. In this model, ϕ^E is in the RKHS generated by the chosen Matérn kernel. We can see that our estimators produced faithful approximations to the true kernel based on the results we report in Table 7 and Figure 1(d),(e). They also show that we can identify the nonexistence of the alignment-based interaction with very small errors and select the correct

model. The discrepancy between the true trajectories and the predicted trajectories on both the training time interval $[0, 5]$ and future time interval $[5, 10]$ are shown in Table 8. Even if the trajectory prediction errors can go up to $O(10^{-1})$ with the presence of a relatively large noise for the systems with $N = 10$, our estimators provided faithful predictions to most of the agents in the system and the milling pattern as shown in Figure 1(f).

Table 7

Means and standard deviations of the errors of $\hat{\boldsymbol{\alpha}}$ (including $\hat{\sigma}$ when noise exists) and $\hat{\phi}$ for different settings of FM dynamics.

$\{N, M, L, \sigma\}$	$\|\hat{\boldsymbol{\alpha}} - \boldsymbol{\alpha}\|_\infty$	$\|\hat{\phi^E} - \phi^E\|_\infty / \|\phi^E\|_\infty$	$\|\hat{\phi^A} - 0\|_\infty$
$\{10, 1, 3, 0\}$	$7.9 \cdot 10^{-4} \pm 1.0 \cdot 10^{-3}$	$3.6 \cdot 10^{-2} \pm 4.3 \cdot 10^{-3}$	$6.6 \cdot 10^{-4} \pm 6.9 \cdot 10^{-4}$
$\{10, 1, 9, 0\}$	$6.4 \cdot 10^{-5} \pm 6.2 \cdot 10^{-5}$	$3.9 \cdot 10^{-2} \pm 2.7 \cdot 10^{-3}$	$1.6 \cdot 10^{-4} \pm 1.3 \cdot 10^{-4}$
$\{10, 3, 3, 0\}$	$\mathbf{4.7 \cdot 10^{-5} \pm 5.0 \cdot 10^{-5}}$	$3.8 \cdot 10^{-2} \pm 5.4 \cdot 10^{-3}$	$1.2 \cdot 10^{-4} \pm 1.7 \cdot 10^{-4}$
$\{10, 3, 3, 0.01\}$	$3.4 \cdot 10^{-3} \pm 1.9 \cdot 10^{-3}$	$\mathbf{2.9 \cdot 10^{-2} \pm 5.7 \cdot 10^{-3}}$	$2.9 \cdot 10^{-3} \pm 4.3 \cdot 10^{-3}$
$\{10, 3, 3, 0.05\}$	$1.4 \cdot 10^{-2} \pm 8.5 \cdot 10^{-3}$	$4.9 \cdot 10^{-2} \pm 1.5 \cdot 10^{-2}$	$\mathbf{4.6 \cdot 10^{-5} \pm 7.0 \cdot 10^{-5}}$
$\{10, 3, 3, 0.1\}$	$3.5 \cdot 10^{-2} \pm 7.2 \cdot 10^{-2}$	$7.1 \cdot 10^{-2} \pm 2.0 \cdot 10^{-2}$	$2.9 \cdot 10^{-2} \pm 9.0 \cdot 10^{-2}$

Table 8

The trajectory prediction errors for different settings of FM dynamics.

$\{N, M, L, \sigma\}$	Training IC $[0, 5]$	Training IC $[5, 10]$	new IC $[0, 5]$	new IC $[5, 10]$
$\{10, 1, 3, 0\}$	$2.1 \cdot 10^{-3} \pm 2.0 \cdot 10^{-3}$	$1.0 \cdot 10^{-2} \pm 8.7 \cdot 10^{-3}$	$1.9 \cdot 10^{-3} \pm 1.9 \cdot 10^{-3}$	$5.4 \cdot 10^{-3} \pm 4.4 \cdot 10^{-3}$
$\{10, 1, 9, 0\}$	$\mathbf{3.4 \cdot 10^{-4} \pm 2.9 \cdot 10^{-4}}$	$\mathbf{1.4 \cdot 10^{-3} \pm 1.2 \cdot 10^{-3}}$	$\mathbf{4.7 \cdot 10^{-4} \pm 4.2 \cdot 10^{-4}}$	$\mathbf{1.3 \cdot 10^{-3} \pm 1.2 \cdot 10^{-3}}$
$\{10, 3, 3, 0\}$	$8.1 \cdot 10^{-4} \pm 8.0 \cdot 10^{-4}$	$2.2 \cdot 10^{-3} \pm 2.0 \cdot 10^{-3}$	$8.8 \cdot 10^{-4} \pm 8.8 \cdot 10^{-4}$	$3.5 \cdot 10^{-3} \pm 2.8 \cdot 10^{-3}$
$\{10, 3, 3, 0.01\}$	$8.3 \cdot 10^{-3} \pm 3.8 \cdot 10^{-3}$	$1.8 \cdot 10^{-2} \pm 1.2 \cdot 10^{-2}$	$6.6 \cdot 10^{-3} \pm 3.2 \cdot 10^{-3}$	$1.4 \cdot 10^{-2} \pm 9.3 \cdot 10^{-3}$
$\{10, 3, 3, 0.05\}$	$3.4 \cdot 10^{-2} \pm 2.1 \cdot 10^{-2}$	$7.1 \cdot 10^{-2} \pm 4.7 \cdot 10^{-2}$	$3.7 \cdot 10^{-2} \pm 1.9 \cdot 10^{-2}$	$7.0 \cdot 10^{-2} \pm 4.7 \cdot 10^{-2}$
$\{10, 3, 3, 0.1\}$	$8.0 \cdot 10^{-2} \pm 9.8 \cdot 10^{-2}$	$1.5 \cdot 10^{-1} \pm 1.9 \cdot 10^{-1}$	$9.5 \cdot 10^{-2} \pm 1.3 \cdot 10^{-1}$	$1.5 \cdot 10^{-1} \pm 2.3 \cdot 10^{-1}$

4.2.3. Anticipation Dynamics. In this subsection, we consider a more complicated model where the interactions depend on both the pairwise distance and the differences in velocities, i.e. both ϕ^E and ϕ^A are nonzero. The anticipation dynamics (AD) models in [13] are suitable candidates, and we consider the system of N agents in the form (1.1) with components defined in Table 4.

The errors of the estimations for $\boldsymbol{\alpha}$ and the learned ϕ^E, ϕ^A are shown in Table 9. In this model, both ϕ^E and ϕ^A are in the RKHS generated by the chosen Matérn kernel. We can see that our estimators produced faithful approximations to both true kernels based on the results we report in Table 9 and Figure 1(g),(h). The comparisons between the true trajectories and the predicted trajectories on both the training time interval $[0, 10]$ and future time interval $[10, 20]$ are shown in Table 10. The estimators can produce accurate approximations of the

true trajectories with errors at most $O(10^{-1})$, see also Figure 1(i).

Table 9

Means and standard deviations of the errors of $\hat{\sigma}$ (when noise exists) and $\hat{\phi}$ for different settings of AD dynamics.

$\{N, M, L, \sigma\}$	$\|\hat{\sigma} - \sigma\|_\infty$	$\|\hat{\phi^E} - \phi^E\|_\infty / \|\phi^E\|_\infty$	$\|\hat{\phi^A} - \phi^A\|_\infty / \|\phi^A\|_\infty$
$\{10, 3, 3, 0\}$	-	$9.2 \cdot 10^{-2} \pm 7.4 \cdot 10^{-3}$	$4.5 \cdot 10^{-2} \pm 1.0 \cdot 10^{-2}$
$\{10, 6, 3, 0\}$	-	$7.9 \cdot 10^{-2} \pm 6.7 \cdot 10^{-3}$	$4.3 \cdot 10^{-2} \pm 5.1 \cdot 10^{-3}$
$\{10, 12, 3, 0\}$	-	$\mathbf{7.4 \cdot 10^{-2} \pm 6.1 \cdot 10^{-3}}$	$\mathbf{3.6 \cdot 10^{-2} \pm 7.0 \cdot 10^{-3}}$
$\{10, 12, 3, 0.005\}$	$8.8 \cdot 10^{-5} \pm 5.1 \cdot 10^{-5}$	$1.3 \cdot 10^{-1} \pm 1.7 \cdot 10^{-2}$	$7.3 \cdot 10^{-2} \pm 3.2 \cdot 10^{-2}$
$\{10, 12, 3, 0.01\}$	$1.8 \cdot 10^{-4} \pm 9.9 \cdot 10^{-5}$	$1.6 \cdot 10^{-1} \pm 1.9 \cdot 10^{-2}$	$9.3 \cdot 10^{-2} \pm 4.1 \cdot 10^{-2}$

Table 10

The trajectory prediction errors for different settings of AD dynamics.

$\{N, M, L, \sigma\}$	Training IC $[0, 10]$	Training IC $[10, 20]$	new IC $[0, 10]$	new IC $[10, 20]$
$\{10, 3, 3, 0\}$	$\mathbf{4.2 \cdot 10^{-4} \pm 3.8 \cdot 10^{-4}}$	$\mathbf{2.3 \cdot 10^{-4} \pm 2.1 \cdot 10^{-4}}$	$6.1 \cdot 10^{-4} \pm 8.4 \cdot 10^{-4}$	$3.5 \cdot 10^{-4} \pm 5.0 \cdot 10^{-4}$
$\{10, 6, 3, 0\}$	$6.6 \cdot 10^{-4} \pm 7.4 \cdot 10^{-4}$	$3.8 \cdot 10^{-4} \pm 4.1 \cdot 10^{-4}$	$7.1 \cdot 10^{-4} \pm 9.2 \cdot 10^{-4}$	$3.9 \cdot 10^{-4} \pm 5.2 \cdot 10^{-4}$
$\{10, 12, 3, 0\}$	$6.2 \cdot 10^{-4} \pm 6.8 \cdot 10^{-4}$	$3.3 \cdot 10^{-4} \pm 3.7 \cdot 10^{-4}$	$\mathbf{3.7 \cdot 10^{-4} \pm 5.2 \cdot 10^{-4}}$	$\mathbf{2.1 \cdot 10^{-4} \pm 3.1 \cdot 10^{-4}}$
$\{10, 12, 3, 0.005\}$	$1.9 \cdot 10^{-3} \pm 2.1 \cdot 10^{-3}$	$1.1 \cdot 10^{-3} \pm 1.2 \cdot 10^{-3}$	$1.1 \cdot 10^{-3} \pm 1.2 \cdot 10^{-3}$	$6.8 \cdot 10^{-4} \pm 7.1 \cdot 10^{-4}$
$\{10, 12, 3, 0.01\}$	$3.4 \cdot 10^{-3} \pm 4.3 \cdot 10^{-3}$	$1.9 \cdot 10^{-3} \pm 2.4 \cdot 10^{-3}$	$1.9 \cdot 10^{-3} \pm 2.1 \cdot 10^{-3}$	$1.2 \cdot 10^{-3} \pm 1.3 \cdot 10^{-3}$

4.3. Model selection for the order of systems. An example of opinion dynamics is shown below to test the validity of our method for identifying the order of dynamic systems. This is a first-order system of N interacting agents, and each agent i is characterized by a continuous opinion variable $x_i \in \mathbb{R}$. The dynamics of opinion exchange are governed by the first-order equation mentioned in subsection 2.2.1 with

$$\phi^E(r) = \begin{cases} 25r & \text{if } 0 \le r < 0.4, \\ 10 & \text{if } 0.4 \le r < 0.6, \\ 25 - 25r & \text{if } 0.6 \le r < 1, \\ 0 & \text{if } r \ge 1. \end{cases} \tag{4.4}$$

The interaction kernel ϕ^E encodes the non-repulsive interactions between agents: all agents aim to align their opinions to their connected neighbors according to distance-based attractive influences. We consider the case where there is no non-collective force, i.e. $F_i(\boldsymbol{x}_i, \boldsymbol{\alpha}) \equiv 0$. We also consider a more complicated case where there exist stubborn agents, i.e.

$$F_i(\boldsymbol{x}_i, \boldsymbol{\alpha}) = \begin{cases} -\kappa(\boldsymbol{x}_i - P_i) & \text{if agent } i \text{ is stubborn with bias } P_i \\ 0 & \text{otherwise} \end{cases} \tag{4.5}$$

where $F_i(\boldsymbol{x}_i, \boldsymbol{\alpha})$ describes the additional influence induced by the stubbornness: the stubborn agents have strong desires to follow their bias P_i, and κ controls the rate of convergence towards their bias. The stubborn agents may cause a major effect on the collective opinion formation process. If $\kappa = 0$, then stubborn agents do not follow their biases and behave as regular agents.

Table 11 shows the errors of the estimations for m, $\boldsymbol{\alpha}$, and $\phi^E(r)$ in 10 independent trails of experiments. It shows our method can identify the order of dynamics with the estimation of $m \approx 0$ and learn the interaction kernel ϕ simultaneously, see also Figure 2.

Table 11

Means and standard deviations of the errors of $\hat{m}$ (including $\hat{\sigma}$ when noise exists) and $\hat{\phi}$ for different settings of OD dynamics.

Model	$\{N, M, L, \sigma\}$	$\|\hat{m} - 0\|_\infty$	$\|\hat{\boldsymbol{\alpha}} - \boldsymbol{\alpha}\|_\infty$	$\|\hat{\phi} - \phi\|_\infty / \|\phi\|_\infty$
OD	$\{5, 6, 3, 0\}$	$8.5 \cdot 10^{-4} \pm 9.0 \cdot 10^{-4}$	-	$3.8 \cdot 10^{-3} \pm 1.1 \cdot 10^{-3}$
OD	$\{5, 6, 3, 0.1\}$	$4.8 \cdot 10^{-3} \pm 5.2 \cdot 10^{-4}$	$3.2 \cdot 10^{-2} \pm 1.6 \cdot 10^{-2}$	$1.1 \cdot 10^{-2} \pm 5.6 \cdot 10^{-3}$
ODS	$\{10, 3, 3, 0\}$	$5.5 \cdot 10^{-4} \pm 2.8 \cdot 10^{-4}$	$7.2 \cdot 10^{-2} \pm 4.1 \cdot 10^{-2}$	$5.2 \cdot 10^{-2} \pm 4.4 \cdot 10^{-2}$
ODS	$\{10, 3, 3, 0.1\}$	$3.8 \cdot 10^{-3} \pm 1.8 \cdot 10^{-3}$	$9.0 \cdot 10^{-1} \pm 1.1 \cdot 10^{0}$	$3.3 \cdot 10^{-2} \pm 1.9 \cdot 10^{-2}$

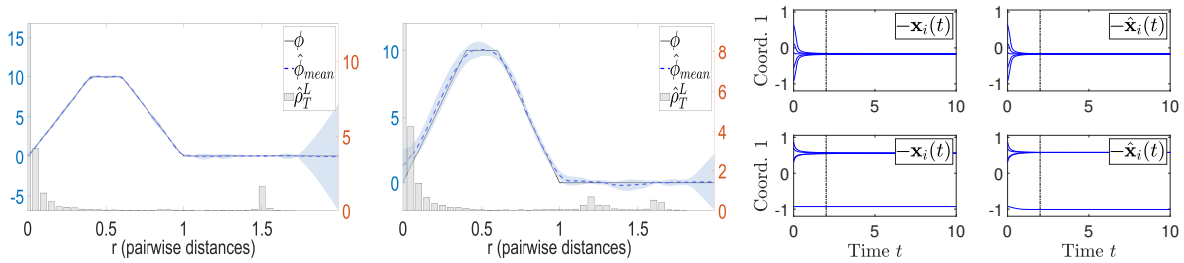


Figure 2. *Model selection of OD ($\{N, M, L\} = \{5, 6, 3\}$) and $\sigma = 0, 0.1$ using the Matérn kernel. Left: Predictive mean $\hat{\phi}$ of the true kernel, and two-standard-deviation band (light blue color) around the mean. The grey bars represent the empirical density of the $\tilde{\rho}_r$. Right: the true (left) versus predicted (right) trajectories using $\hat{\boldsymbol{\alpha}}$ and $\hat{\phi}$ with initial conditions of training data (top) and testing data (bottom) when $\sigma = 0.1$.*

4.4. Real fish data. Finally, we test the performance of our method using two real datasets of swimming fish by Couzin et al., which are available at ScholarsArchive of Oregon State University[†]. The experimental arena consisted of a white shallow tank of size 2.1×1.2 m (7×4 ft) surrounded by a floor-to-ceiling white curtain. Water depth was chosen to be 4.5-5 cm so the schools would be approximately 2D. We consider two data sets, one is from

[†]Katz, Yael, Kolbjorn Tunstrom, Christos C Ioannou, Cristian Huepe, and Iain D Couzin, 2021. The URL address is https://ir.library.oregonstate.edu/concern/datasets/zk51vq07c

frame 2201 to frame 2296 which consists of 50 fish and forms a flocking behavior, and another one is from frame 4601 to frame 4798 which consists of 124 fish and forms a milling behavior. We relabel them as frame 0 to frame 95 and frame 0 to frame 198 respectively, refer to more details of the dataset in the supplementary information of [65]. We first normalize the position data into the region [0,1], and then we smooth the data by using a moving window average with a window size of 10 frames and apply the finite difference method to calculate the velocities and accelerations.

Flocking behavior example. For the first data set, as shown in Figure 3(e), the fish will eventually follow approximately the same direction as time evolves. In this case, the magnitude of the velocity data of fish is relatively small. The velocities can be considered the same, as long as their normalized direction vectors are very close. So the fish exhibit approximate flocking behavior (i.e. $\|\boldsymbol{v}_i - \boldsymbol{v}_c\| \approx 0$ for all i and some common velocity $\boldsymbol{v}_c$). Therefore, we use the Cucker-Smale system shown in subsection 4.2.1 to model the flocking behavior, i.e. considering the governing equation (1.1) with corresponding interaction kernel ϕ^A and force $F(\boldsymbol{x}_i, \dot{\boldsymbol{x}}_i, \boldsymbol{\alpha})$, $\boldsymbol{\alpha} = (\kappa, p)$ shown in Table 4 for CS dynamics.

The training data consists of frame 0 and frame 28. In the training procedure, we first use a subset of data with two selected agents, the initialization of hyperparameters for $\theta^E, \theta^A, \sigma$, and $\boldsymbol{\alpha}$ are $(1,1), (1,1), 0.001, (1,1)$, and we set the length of runs in the minimizer solver to be 100. The results shown in Figure 3(a),(b) suggest there only exist alignment-based interactions since the estimated energy-based interaction $\hat{\phi}^E \equiv 0$. Therefore, we use all data to learn the system with only ϕ^A. After we obtain the estimators, we run the learned dynamical system on the time interval [0,20] with frame 0 as the initial condition. We find that the simulated position data at $t = 19$ matches the position data at frame 95 very well. We then compare the original position data set with the simulated ones at $t = 0 : 0.2 : 19$.

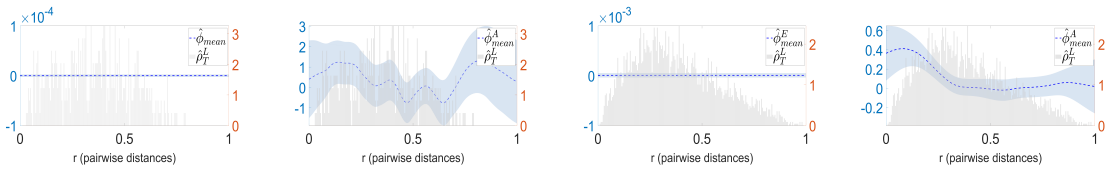
Milling behavior example. For the second data set, as shown in Figure 3(g), the fish will eventually follow approximately a milling pattern. Therefore, we use the Fish-Milling system shown in subsection 4.2.2 to model the milling behavior, i.e. considering the governing equation (1.1) with interaction kernel ϕ^E and force $F(\boldsymbol{x}_i, \dot{\boldsymbol{x}}_i, \boldsymbol{\alpha})$, $\boldsymbol{\alpha} = (\gamma, \beta)$ shown in Table 4 for FM dynamics.

The training data consists of frame 0 and frame 28. In the training procedure, we first use a subset of data with two selected agents, the initialization of hyperparameters for $\theta^E, \theta^A, \sigma$, and $\boldsymbol{\alpha}$ are $(1,1), (1,1), 0.001, (1,1)$, and we set the length of runs in the minimizer solver to be 100. With the estimated parameters, we obtained the estimators for ϕ^E and ϕ^A using all data, and run the learned dynamical system at the time interval [0,38] with the frame 0 as the initial condition.

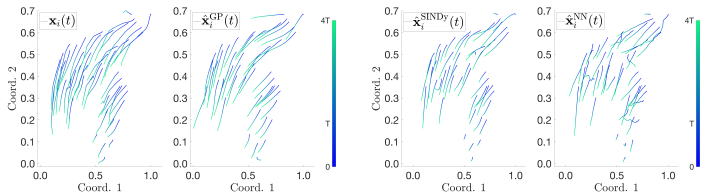
Measure of Performance. To evaluate the performance at the group level, we consider the group polarisation $M(t)$ [66], which is a vector order parameter that encapsulates both the direction and degree of the fish alignment, which is defined by

$$M(t) = \frac{1}{N} \sum_{i=1}^{N} \hat{\boldsymbol{v}}_i(t) \tag{4.6}$$

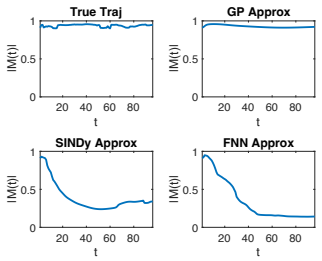
where $\hat{\boldsymbol{v}}_i(t) = \boldsymbol{v}_i(t)/\|\boldsymbol{v}_i(t)\|$ is the direction of motion of the i-th fish (at time t). When $|M|$ is close to 1, the fish are moving in a coherent direction, whereas when $|M|$ is close to zero,



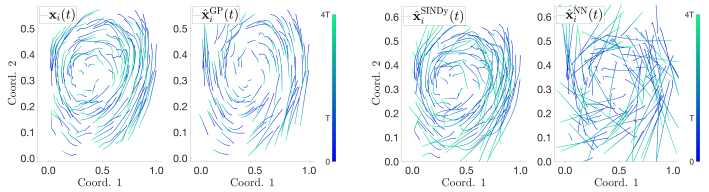
(a) Real data CS: ϕ^E (b) Real data CS: ϕ^A (c) Real data FM: ϕ^E (d) Real data FM: ϕ^E



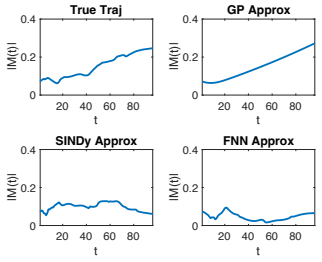
(e) Real data CS trajectory prediction



(f) Real data CS comparisons



(g) Real data FM trajectory prediction



(h) Real data FM comparisons

Figure 3. *(a)(b): Fitting into a Cucker-Smale system (dim=100), estimated ϕ^E and ϕ^A with all data; (c)(d): Fitting into a Fish-Milling system (dim=248), estimated ϕ^E and ϕ^A with data of 2 agents; (e)(g): true dynamics (left) v.s. the predicted dynamics using our proposed approach, SINDy model, and FNN model, with frame 0 as the initial condition; (f)(h): baseline comparisons using the group polarisation parameter $M(t)$.*

there is no prevailing direction and individual motion is effectively isotropic.

Baseline Comparisons. We perform comparisons with approaches that learn the right-hand side function of (1.1) directly from trajectory data: the first one is SINDy [55], which aims at finding a sparse representation for each row of governing equations in a (typically large) dictionary; the second one is regression using feed-forward neural networks, for which we use the MATLAB® 2021a Deep Learning Toolbox™.

For the SINDy model, we apply a reasonably large dictionary consisting of monomials up to order 2, sines, and cosines of frequencies $\{k\}_{k=1}^{10}$. For the neural network model, we consider a three-layer FNN (Feed-Forward Neural Network) with $[50, 50, 25]$ hidden units for the flocking behavior example, and a two-layer FNN with $[40, 20]$ hidden units for the milling behavior example.

The predictive trajectories for the flocking behavior example using different models are shown in Figure 3(e). We compare the performances in terms of group polarisation $M(t)$

in Figure 3(f), and the order-1 Wasserstein distances between the empirical distributions of $M(t)$ in true data and the predicted dynamics are $W(p_M^{true}, p_M^{gp}) = 0.0112$, $W(p_M^{true}, p_M^{SINDy}) = 0.5497$, and $W(p_M^{true}, p_M^{NN}) = 0.5869$, see Figure 4 (Left). The results for the milling behavior example are shown in Figure 3(g). We compare the performances in Figure 3(h), and the order-1 Wasserstein distances between the empirical distributions of $M(t)$ in the true data and the predicted dynamics are $W(p_M^{true}, p_M^{gp}) = 0.0087$, $W(p_M^{true}, p_M^{SINDy}) = 0.0530$, and $W(p_M^{true}, p_M^{NN}) = 0.1055$, see Figure 4 (Right). In both examples, we can see that although both predictions using the SINDy and FNN models look similar to the true trajectories, based on the group polarisation parameter $M(t)$ and comparing the changes of $M(t)$ in t or the empirical distributions of $M(t)$, only our model using GP captures the group behaviors.

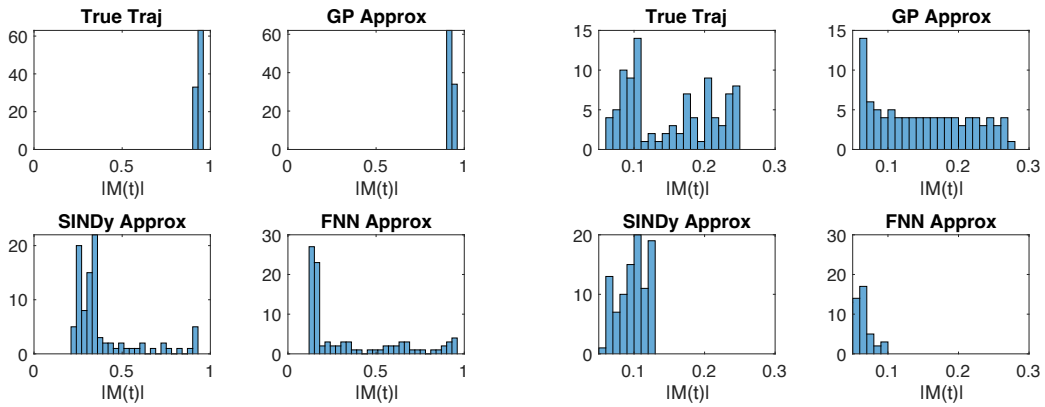


Figure 4. *Baseline comparisons using the group polarisation parameter $M(t)$. Left: the empirical distribution of $M(t)$ in the flocking behavior example, the order-1 Wasserstein distances between true data and the predicted dynamics are $W(p_M^{true}, p_M^{gp}) = 0.0112$, $W(p_M^{true}, p_M^{SINDy}) = 0.5497$, and $W(p_M^{true}, p_M^{NN}) = 0.5869$. Right: the empirical distribution of $M(t)$ in the milling behavior example, the order-1 Wasserstein distances between true data and the predicted dynamics are $W(p_M^{true}, p_M^{gp}) = 0.0087$, $W(p_M^{true}, p_M^{SINDy}) = 0.0530$, and $W(p_M^{true}, p_M^{NN}) = 0.1055$.*

We also compare our result with two other GP models in the milling behavior example, where the parameters $\boldsymbol{\alpha}$ are not estimated properly: (1) we use the initial values of hyperparameters, i.e. let $\theta^E, \theta^A, \sigma$ and $\boldsymbol{\alpha}$ equal $(1,1), (1,1), 0.001, (1,1)$, and do not train those hyperparameters; (2) we apply the noise-free model, i.e. do not consider noise and let $\sigma \equiv 0$. The results of these two GP models are shown in Figure 5.

5. Acceleration Result Comparison. We now present our acceleration (see subsection 2.5) results for a 20-dimensional Fish Milling (FM) system with increasing observational data. When we have a larger amount of observational data, we will focus on learning σ, γ, and β, and use $\theta^E = \theta^A = 1$ for a default prior with the Matérn kernel. We will show the impact on kernel predictions is minimal.

We use $\nu = \frac{3}{2}$ for all examples below. All results shown are averaged over 10 complete runs with standard deviation included where we used the same training data but with initialized hyperparameters uniformly at random from an interval centered at the ground truth with

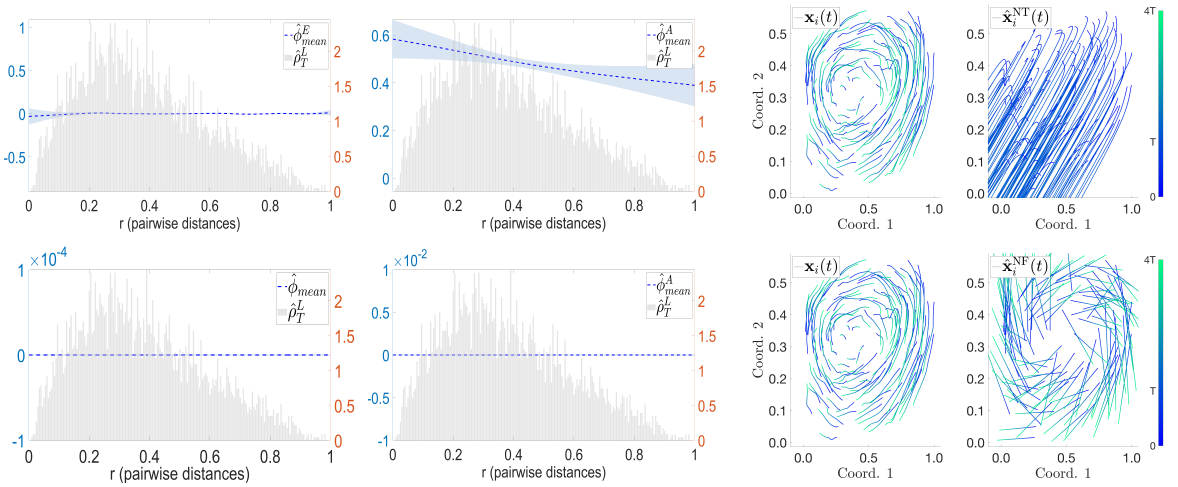


Figure 5. *Fitting into a Fish-Milling system (dim=248). Top: estimated ϕ^E, ϕ^A, and predictive trajectory with initial parameters, i.e. no training (NT) for hyperparameters $\theta^E, \theta^A, \sigma$, and $\boldsymbol{\alpha}$; Bottom: estimated ϕ^E, ϕ^A, and predictive trajectory using noise-free (NF) model, i.e. $\hat{\sigma} = 0$.*

radius 0.5 in each trial. We use the randomized Gaussian Nystrom preconditioner [52] for all tests with rank the floor of $\frac{30}{\log(12)} \cdot \log(\frac{NdML}{10})$. While we would ideally use the effective rank of our kernel matrix, this is expensive to compute in practice and we resort to empirical approximation for our trials.

Table 12
System parameters in Fish Milling above

d	N	$[0; T; T_f]$	$\alpha = (\gamma, \beta)$	$\mu_0^{\boldsymbol{x}}$	$\mu_0^{\boldsymbol{v}}$
2	20	$[0, 5, 10]$	$(1.5, 0.5)$	$\mathrm{Unif}([-1, 1]^2)$	$\mathrm{Unif}([0, 0]^2)$

Figure 6 shows that our hyperparameter learning method is able to accurately recover the hyperparameters σ, γ, β with greatly improved runtime compared to the full GP method. Once these hyperparameters are learned, our acceleration can also be utilized for the prediction of the kernel, which also has a low observed error, see Figure 7. Most errors of the kernel prediction occur away from the support of observed data in the FM system and do not affect the trajectory prediction of our system. This is quantified in the very low relative L^2 error of the predicted trajectories of the FM system using our predicted kernel as shown in Figure 8.

These results provide clear evidence of successful acceleration options while maintaining highly acceptable accuracy. While the running time of accelerated MLE can still be expensive for prohibitively large data, the accelerated method scales much better than the fully explicit method and opens up exciting possibilities in modeling large datasets. We note that prediction also scales quite well and relies only upon PCG and preconditioner choice, allowing the usage

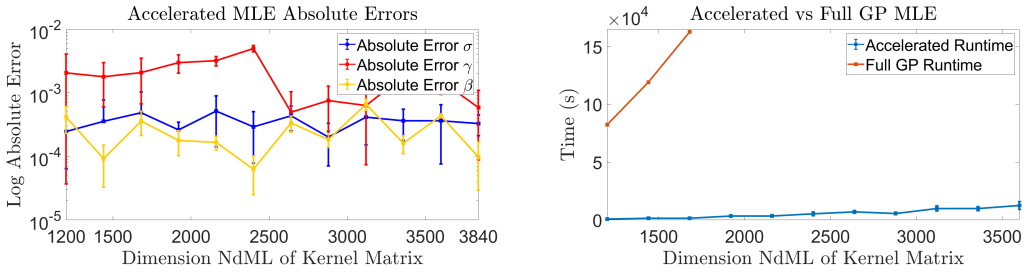


Figure 6. *Log plot of absolute error of learned hyperparameters γ (red), β (yellow), and σ (blue) for the FM system ($\{N, M, L\} = \{20, M, 6\}$) with varying M. True values are $\sigma = 0.01, \gamma = 1.5, \beta = 0.5$. Shown also is a runtime comparison with Full GP.*

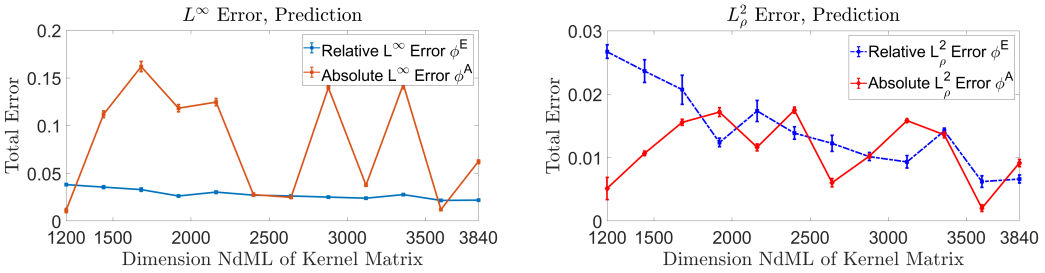


Figure 7. *Accelerated kernel prediction error for the experiments above. Left is the L^∞ error, relative for ϕ^E and absolute for ϕ^A as $\phi^A = 0$ is the ground truth. Right is the L^2_ρ error, relative for ϕ^E and absolute for ϕ^A.*

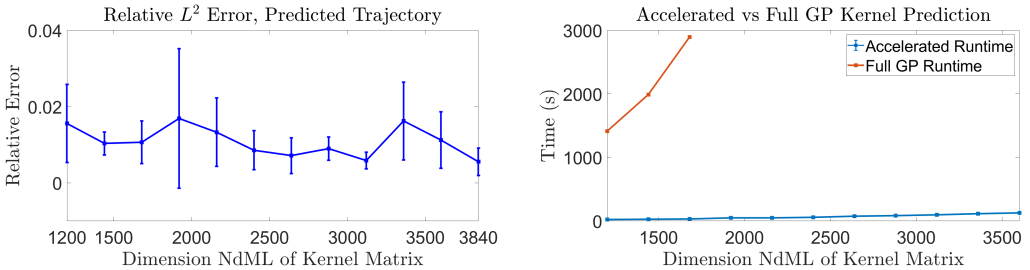


Figure 8. *Relative trajectory prediction error on testing data, FM system ($\{N, M, L\} = \{20, M, 6\}$) with varying M. This plot uses test error on the full interval of $[0, 10]$. Prediction error quickly goes to zero in testing. Shown also is a runtime comparison with full GP.*

of efficient cross-validation techniques for hyperparameter choice in certain classes of problems.

Our central findings are the following:

- We have discovered that accurate hyperparameter recovery can be achieved using a small set of observational data, and using more training data does not necessarily improve the accuracy. This is due to the lack of consistency in the training of MLE, which is a well-known result in Gaussian process regression. We recommend that one should split a small subset for hyperparameter tuning and then use the full dataset for kernel learning. We have seen empirical success with this method.
- In 10 trials with small M, we often observed one or two trials with relatively large recovery errors in hyperparameters. We removed these outliers from our data before

plotting above. We attribute this to the instability of the Lanczos algorithm or the non-convexity of the optimization problem, as in these cases, we observed that the minimization of MLE stopped very early. Nonetheless, we would like to point out that even in these cases, we obtained very satisfying performance in kernel learning and trajectory prediction.

- There are additional opportunities for acceleration in kernel learning that depend on the specific problem and infrastructure available. For instance, in the case of $\nu = \frac{1}{2}$, we may exploit sparsity in a decomposition of the kernel matrix $K_{\mathbf{f}_{\boldsymbol{\phi}}}(\mathbb{Y}, \mathbb{Y}; \theta)$, as developed in [67], while maintaining desired exactness. We leave the extension of this method to all half-integer ν values for future work. Furthermore, there are avenues for accelerating GP learning using modern hardware. With access to GPUs, one can parallelize the explicit construction of kernel matrices and the Lanczos algorithm calculations. These steps are embarrassingly parallel and allow for demanding much greater accuracy.

6. Final remarks and future work. In this work, we present an approach based on Gaussian processes to perform the model selection of particle/agent-based models from scarce and noisy data. We propose efficient acceleration techniques to improve the scalability. The methodology is extendable to cover heterogeneous systems with multiple types of agents and external potentials. It is also possible to extend the learning approach to the mean-field limits of the particle models. Another line of future work is to apply the quantitative framework developed in this paper to design a data acquisition plan (active learning). The goal is to optimize the kernel learning using the least amount of trajectory data by looking at their marginal pairwise distance distributions. We leave it as future work.

Acknowledgments. Charles Kulick was partially supported by NSF DMS-2111303. S.T. was partially supported by Hellman Family Faculty Fellowship, and the NSF DMS-2111303. S.T. would like to thank Hengrui Luo and Didong Li for their helpful discussions.

7. Appendix.

A. Learning approach for model selection. Our learning approach is a generalization of the methodology proposed in [43]; to be self-contained, we state the detailed formulation here.

Lemma A.1. *Let $\boldsymbol{\phi} = (\phi^E, \phi^A)$ be two Gaussian processes with mean zero and covariance function $K_{\theta^E}, K_{\theta^A} : [0, R] \times [0, R] \to \mathbb{R}$ respectively, i.e., $\phi^{\text{type}} \sim \mathcal{GP}(0, K_{\theta^{\text{type}}}(r, r'))$, type $=$ E or A, and $\boldsymbol{mZ}(t) = F_{\boldsymbol{\alpha}}(\boldsymbol{Y}(t)) + \mathbf{f}_{\boldsymbol{\phi}}(\boldsymbol{Y}(t))$ as defined in (1.2). Then for any $t, t' \in [0, T]$, we have that,*

$$\begin{bmatrix} \boldsymbol{mZ}(t) \\ \boldsymbol{mZ}(t') \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} F_{\boldsymbol{\alpha}}(\boldsymbol{Y}(t)) \\ F_{\boldsymbol{\alpha}}(\boldsymbol{Y}(t')) \end{bmatrix}, K_{\mathbf{f}_{\boldsymbol{\phi}}}(\boldsymbol{Y}(t), \boldsymbol{Y}(t')) \right), \tag{A.1}$$

where $K_{\mathbf{f}_\phi}(\boldsymbol{X}(t), \boldsymbol{X}(t')))$ is the covariance matrix $\mathrm{Cov}(\mathbf{f}_\phi(\boldsymbol{X}(t)), \mathbf{f}_\phi(\boldsymbol{X}(t')))$ with (i, j)th block

$$\mathrm{Cov}([\mathbf{f}_{\boldsymbol{\phi}}(\boldsymbol{Y})]_i, [\mathbf{f}_{\boldsymbol{\phi}}(\boldsymbol{Y}')]_j) = \frac{1}{N^2} \sum_{k \neq i, k' \neq j} \left(K_{\theta^E}(r^{\boldsymbol{x}}_{ik}, r^{\boldsymbol{x}'}_{jk'}) \boldsymbol{r}^{\boldsymbol{x}}_{ik} \boldsymbol{r}^{\boldsymbol{x}'\,T}_{jk'} + K_{\theta^A}(r^{\boldsymbol{x}}_{ik}, r^{\boldsymbol{x}'}_{jk'}) \boldsymbol{r}^{\boldsymbol{v}}_{ik} \boldsymbol{r}^{\boldsymbol{v}'\,T}_{jk'} \right), \tag{A.2}$$

see Table 3 for the definitions.

Proof. For $\phi \sim \mathcal{GP}(0, K_\theta(r, r'))$, and any $r, r' \in [0, R]$, we have that,

$$\mathbb{E}[\phi(r)] = 0, \tag{A.3}$$

$$\mathrm{Cov}[\phi(r), \phi(r')] = K_\theta(r, r'). \tag{A.4}$$

Therefore, for any collection of states $\{r_i\}_{i=1}^n \subset [0, R]$, and $\{a_i\}_{i=1}^n, \{b_i\}_{i=1}^n \subset \mathbb{R}$, the linear operator on function values $\mathcal{L}(\{\phi(r_i)\}_{i=1}^n) := (a_i\phi(r_i) + b_i)_{i=1}^n$ satisfies

$$\mathcal{L}(\{\phi(r_i)\}_{i=1}^n) \sim \mathcal{N}(\mathrm{vec}(\{b_i\}_{i=1}^n), \Sigma_{\mathcal{L}(\phi)}), \tag{A.5}$$

where $\mathcal{N}$ denotes the Gaussian distribution, $\mathrm{vec}(\{b_i\}_{i=1}^n) \in \mathbb{R}^n$ is the vectorization of $\{b_i\}_{i=1}^n$, and the covariance matrix $\Sigma_{\mathcal{L}(\phi)} = \{a_i a_j K_\theta(r_i, r_j)\}_{i,j=1}^n \in \mathbb{R}^{n\times n}$.

Therefore, since ϕ^E, ϕ^A are independent, and $\mathbf{f}_{\boldsymbol{\phi}}(\boldsymbol{Y}(t))$ is linear in $\boldsymbol{\phi}$, for any t, t', we have that

$$\begin{bmatrix} \mathbf{f}_{\boldsymbol{\phi}}(\boldsymbol{Y}(t)) \\ \mathbf{f}_{\boldsymbol{\phi}}(\boldsymbol{Y}(t')) \end{bmatrix} \sim \mathcal{N}(\mathbf{0}, K_{\mathbf{f}_{\boldsymbol{\phi}}}(\boldsymbol{Y}(t), \boldsymbol{Y}(t'))), \tag{A.6}$$

where $K_{\mathbf{f}_{\boldsymbol{\phi}}}(\boldsymbol{Y}(t), \boldsymbol{Y}(t')))$ is the covariance matrix

$$\mathrm{Cov}(\mathbf{f}_{\boldsymbol{\phi}}(\boldsymbol{Y}(t)), \mathbf{f}_{\boldsymbol{\phi}}(\boldsymbol{Y}(t'))) = \big(\mathrm{Cov}([\mathbf{f}_{\boldsymbol{\phi}}(\boldsymbol{Y}(t))]_i, [\mathbf{f}_{\boldsymbol{\phi}}(\boldsymbol{Y}(t')]_j))\big)_{i,j=1}^{N,N}, \tag{A.7}$$

with (i, j)th block

$$\mathrm{Cov}([\mathbf{f}_{\boldsymbol{\phi}}(\boldsymbol{Y})]_i, [\mathbf{f}_{\boldsymbol{\phi}}(\boldsymbol{Y}')]_j) = \frac{1}{N^2} \sum_{k\neq i, k'\neq j} \big(K_{\theta^E}(r_{ik}^{\boldsymbol{x}}, r_{jk'}^{\boldsymbol{x}'}) \boldsymbol{r}_{ik}^{\boldsymbol{x}} \boldsymbol{r}_{jk'}^{\boldsymbol{x}'\,T} + K_{\theta^A}(r_{ik}^{\boldsymbol{x}}, r_{jk'}^{\boldsymbol{x}'}) \boldsymbol{r}_{ik}^{\boldsymbol{v}} \boldsymbol{r}_{jk'}^{\boldsymbol{v}'\,T}\big),$$

Thus, by (1.2), the observation $\boldsymbol{Z}$ in the model follows the Gaussian distribution

$$\begin{bmatrix} \boldsymbol{m}\boldsymbol{Z}(t) \\ \boldsymbol{m}\boldsymbol{Z}(t') \end{bmatrix} \sim \mathcal{N}\Big(\begin{bmatrix} F_{\boldsymbol{\alpha}}(\boldsymbol{Y}(t)) \\ F_{\boldsymbol{\alpha}}(\boldsymbol{Y}(t')) \end{bmatrix}, K_{\mathbf{f}_{\boldsymbol{\phi}}}(\boldsymbol{Y}(t), \boldsymbol{Y}(t'))\Big). \tag{A.8}$$

■

Then suppose that the training data consists of

$$\{\mathbb{Y}_M, \mathbb{Z}_{\sigma^2,M}\} = \{\mathbb{X}_M, \mathbb{V}_M, \mathbb{Z}_{\sigma^2,M}\} \tag{A.9}$$

with

$$\begin{aligned}
\mathbb{X}_M &= \mathrm{Vec}\big(\{\boldsymbol{X}^{(m,l)}\}_{m,l=1}^{M,L}\big) \in \mathbb{R}^{dNML}, \\
\mathbb{V}_M &= \mathrm{Vec}\big(\{\boldsymbol{V}^{(m,l)}\}_{m,l=1}^{M,L}\big) = \mathrm{Vec}\big(\{\dot{\boldsymbol{X}}^{(m,l)}\}_{m,l=1}^{M,L}\big) \in \mathbb{R}^{dNML}, \\
\mathbb{Z}_{\sigma^2,M} &= \mathrm{Vec}\big(\{\boldsymbol{Z}_{\sigma^2}^{(m,l)}\}_{m,l=1}^{M,L}\big) = \mathrm{Vec}\big(\{\ddot{\boldsymbol{X}}^{(m,l)} + \sigma^2\boldsymbol{\epsilon}^{(m,l)}\}_{m,l=1}^{M,L}\big) \in \mathbb{R}^{dNML}
\end{aligned}$$

where we observe the dynamics at $0 = t_1 < t_2 < \cdots < t_L = T$; m indexes trajectories corresponding to different initial conditions at $t_1 = 0$; $\boldsymbol{X}^{(m,1)} \overset{i.i.d}{\sim} \mu_0^{\boldsymbol{x}}$, $\boldsymbol{V}^{(m,1)} \overset{i.i.d}{\sim} \mu_0^{\boldsymbol{v}}$, $(\mu_0^{\boldsymbol{x}}, \mu_0^{\boldsymbol{v}})$ are two independent probability measure on $\mathbb{R}^{dN}$; the noise term $\boldsymbol{\epsilon}^{(m,l)} \overset{i.i.d}{\sim} \mathcal{N}(\mathbf{0}, I_{dN})$; we assume that $\mu_0 = (\mu_0^{\boldsymbol{x}}, \mu_0^{\boldsymbol{v}})$ is independent of the distribution of noise.

Applying Lemma A.1, we now derive the negative log marginal likelihood for training parameters $\boldsymbol{\alpha}$, $\boldsymbol{\theta}$, and σ, with given observational data as specified above.

Proposition A.2. *Denote* $\boldsymbol{Y}^{(m,l)} = \boldsymbol{Y}^{(m)}(t_l)$ *and* $\boldsymbol{Z}_{\sigma^2}^{(m,l)} = \boldsymbol{Z}^{(m)}(t_l) + \epsilon^{(m,l)}$ *with i.i.d noise* $\epsilon^{(m,l)} \sim \mathcal{N}(0, \sigma^2 I_{dN\times dN})$. *Suppose we are given the training data set* $(\mathbb{Y}_M, \mathbb{Z}_{\sigma^2,M}) := \{(\boldsymbol{Y}^{(m,l)}, \boldsymbol{Z}_{\sigma^2}^{(m,l)})\}_{m,l=1}^{M,L}$ *for* $M, L \in \mathbb{N}$, *such that*

$$\boldsymbol{Z}_{\sigma^2}^{(m,l)} = F_{\boldsymbol{\alpha}}(\boldsymbol{Y}^{(m,l)}) + \mathbf{f}_{\boldsymbol{\phi}}(\boldsymbol{Y}^{(m,l)}) + \epsilon^{(m,l)}, \tag{A.10}$$

with $F_{\boldsymbol{\alpha}}$, $\mathbf{f}_{\boldsymbol{\phi}}$ *defined in Table 3. Then the negative log marginal likelihood of* $\mathbb{Z}_{\sigma^2,M}$ *given* $\mathbb{Y}_M$ *and parameters* $\boldsymbol{\alpha}$, θ, σ *satisfies*

$$\begin{aligned} &-\log p(\boldsymbol{m}\mathbb{Z}_{\sigma^2,M}|\mathbb{Y}_M, \boldsymbol{\alpha}, \boldsymbol{\theta}, \sigma^2) \\ &= \frac{1}{2}(\boldsymbol{m}\mathbb{Z}_{\sigma^2,M} - F_{\boldsymbol{\alpha}}(\mathbb{Y}_M))^T (K_{\mathbf{f}_{\boldsymbol{\phi}}}(\mathbb{Y}_M, \mathbb{Y}_M; \boldsymbol{\theta}) + \sigma^2 I_{dNML})^{-1} (\boldsymbol{m}\mathbb{Z}_{\sigma^2,M} - F_{\boldsymbol{\alpha}}(\mathbb{Y}_M)) \end{aligned} \tag{A.11}$$

$$+ \frac{1}{2}\log|K_{\mathbf{f}_{\boldsymbol{\phi}}}(\mathbb{Y}_M, \mathbb{Y}_M; \boldsymbol{\theta}) + \sigma^2 I_{dNML}| + \frac{dNML}{2}\log 2\pi. \tag{A.12}$$

where I_{dNML} *is the identity matrix of consistent size.*

Proof. Using Lemma A.1, since $\epsilon^{(m,l)}$ is i.i.d Gaussian noise and is independent of the initial distributions, we have that

$$\boldsymbol{m}\mathbb{Z}_{\sigma^2,M} \sim \mathcal{N}(F_{\boldsymbol{\alpha}}(\mathbb{Y}_M), K_{\mathbf{f}_{\boldsymbol{\phi}}}(\mathbb{Y}_M, \mathbb{Y}_M; \boldsymbol{\theta}) + \sigma^2 I_{dNML}), \tag{A.13}$$

where the mean vector $F_{\boldsymbol{\alpha}}(\mathbb{Y}_M) = \mathrm{Vec}((F_{\boldsymbol{\alpha}}(\boldsymbol{Y}^{(m,l)}))_{m,l=1}^{M,L}) \in \mathbb{R}^{dNML}$, and the covariance matrix $K_\phi(\mathbb{Y}_M, \mathbb{Y}_M; \theta) = \left(\mathrm{Cov}(\mathbf{f}_{\boldsymbol{\phi}}(\boldsymbol{Y}^{(i,j)}), \mathbf{f}_{\boldsymbol{\phi}}(\boldsymbol{Y}^{(i',j')}))\right)_{i,i',j,j'=1}^{M,M,L,L} \in \mathbb{R}^{dNML\times dNML}$ can be computed by using (A.7). According to the properties of the Gaussian distribution, given $\mathbb{Y}_M$ and parameters $\boldsymbol{\alpha}$, θ, σ, we have the negative log marginal likelihood function as shown in (A.12). ∎

As mentioned in the main text, we can apply the gradient-based method [68], to minimize the negative log marginal likelihood and solve for the hyperparameters $(\boldsymbol{\alpha}, \boldsymbol{\theta}, \sigma)$.

Proposition A.3. *Let* $\boldsymbol{\gamma} = (K_{\mathbf{f}_{\boldsymbol{\phi}}}(\mathbb{Y}_M, \mathbb{Y}_M; \boldsymbol{\theta}) + \sigma^2 I)^{-1}(\boldsymbol{m}\mathbb{Z}_{\sigma^2,M} - F_{\boldsymbol{\alpha}}(\mathbb{Y}_M))$. *The partial derivatives of the marginal likelihood w.r.t. the parameters* $\boldsymbol{\alpha}$, $\boldsymbol{\theta}$, *and* σ *can be computed as follows:*

$$\frac{\partial}{\partial \boldsymbol{\alpha}_i}\log p(\boldsymbol{m}\mathbb{Z}_{\sigma^2,M}|\mathbb{Y}_M, \boldsymbol{\alpha}, \boldsymbol{\theta}, \sigma^2) = \boldsymbol{\gamma}^T \frac{\partial F_{\boldsymbol{\alpha}}(\mathbb{Y}_M)}{\partial \boldsymbol{\alpha}_i}. \tag{A.14}$$

$$\frac{\partial}{\partial \boldsymbol{\theta}_j}\log p(\boldsymbol{m}\mathbb{Z}_{\sigma^2,M}|\mathbb{Y}_M, \boldsymbol{\alpha}, \boldsymbol{\theta}, \sigma^2) = \frac{1}{2}\mathrm{Tr}\left((\boldsymbol{\gamma}\boldsymbol{\gamma}^T - (K_{\mathbf{f}_{\boldsymbol{\phi}}}(\mathbb{Y}_M, \mathbb{Y}_M; \boldsymbol{\theta}) + \sigma^2 I)^{-1})\frac{\partial K_{\mathbf{f}_{\boldsymbol{\phi}}}(\mathbb{Y}_M, \mathbb{Y}_M; \boldsymbol{\theta})}{\partial \boldsymbol{\theta}_j}\right). \tag{A.15}$$

$$\frac{\partial}{\partial \sigma}\log p(\boldsymbol{m}\mathbb{Z}_{\sigma^2,M}|\mathbb{Y}_M, \boldsymbol{\alpha}, \boldsymbol{\theta}, \sigma^2) = \mathrm{Tr}\left((\boldsymbol{\gamma}\boldsymbol{\gamma}^T - (K_{\mathbf{f}_{\boldsymbol{\phi}}}(\mathbb{Y}_M, \mathbb{Y}_M; \boldsymbol{\theta}) + \sigma^2 I)^{-1})\right)\sigma. \tag{A.16}$$

With the updated prior ϕ from $\boldsymbol{\theta}$, and the parameters $\boldsymbol{\alpha}$, σ, we show the detailed derivation of our estimators for the prediction $\phi(r^*)$ at $r^* \in [0, R]$.

Theorem A.4. *Suppose we are given the training data set* $(\mathbb{Y}_M, \mathbb{Z}_{\sigma^2,M}) := \{(\boldsymbol{Y}^{(m,l)}, \boldsymbol{Z}_{\sigma^2}^{(m,l)})\}_{m,l=1}^{M,L}$ *defined in Proposition A.2, and the hyperparameters* $(\boldsymbol{\alpha}, \boldsymbol{\theta}, \sigma)$ *are known. Then for any* $r^* \in [0, R]$, *type* $= E$ *or* A, $\phi^{\text{type}}(r^*)$ *satisfies*

$$p(\phi^{\text{type}}(r^*)|\mathbb{Y}_M, \mathbb{Z}_{\sigma^2,M}) \sim \mathcal{N}(\bar{\phi}^{\text{type}}, var(\bar{\phi}^{\text{type}})), \tag{A.17}$$

where

$$\bar{\phi}^{\text{type}} = K_{\phi^{\text{type}},\mathbf{f}_{\boldsymbol{\phi}}}(r^*, \mathbb{Y}_M)(K_{\mathbf{f}_{\boldsymbol{\phi}}}(\mathbb{Y}_M, \mathbb{Y}_M) + \sigma^2 I_{dNML})^{-1}(\boldsymbol{m}\mathbb{Z}_{\sigma^2,M} - F_{\boldsymbol{\alpha}}(\mathbb{Y}_M)), \tag{A.18}$$

$$var(\bar{\phi}^{\text{type}}) = K_{\theta^{\text{type}}}(r^*, r^*) - K_{\phi^{\text{type}},\mathbf{f}_{\boldsymbol{\phi}}}(r^*, \mathbb{Y}_M)(K_{\mathbf{f}_{\boldsymbol{\phi}}}(\mathbb{Y}_M, \mathbb{Y}_M) + \sigma^2 I_{dNML})^{-1} K_{\mathbf{f}_{\boldsymbol{\phi}},\phi^{\text{type}}}(\mathbb{Y}_M, r^*). \tag{A.19}$$

and $K_{\mathbf{f}_{\boldsymbol{\phi}},\phi^{\text{type}}}(\mathbb{Y}_M, r^*) = K_{\phi^{\text{type}},\mathbf{f}_{\boldsymbol{\phi}}}(r^*, \mathbb{Y}_M)^T$ *denotes the covariance matrix between* $\mathbf{f}_{\boldsymbol{\phi}}(\mathbb{Y}_M)$ *and* $\phi^{\text{type}}(r^*)$.

Proof. Since $\mathbf{f}_{\boldsymbol{\phi}}(\mathbb{Y}_M)$ is defined componentwisely as in (1.2), for any $r^* \in [0, R]$, we have that

$$\begin{bmatrix} \mathbf{f}_{\boldsymbol{\phi}}(\mathbb{Y}_M) \\ \phi^{\text{type}}(r^*) \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} K_{\mathbf{f}_{\boldsymbol{\phi}}}(\mathbb{Y}_M, \mathbb{Y}_M) & K_{\mathbf{f}_{\boldsymbol{\phi}},\phi^{\text{type}}}(\mathbb{Y}_M, r^*) \\ K_{\phi^{\text{type}},\mathbf{f}_{\boldsymbol{\phi}}}(r^*, \mathbb{Y}_M) & K_{\theta^{\text{type}}}(r^*, r^*) \end{bmatrix}\right), \tag{A.20}$$

where $K_{\mathbf{f}_{\boldsymbol{\phi}}}(\mathbb{Y}_M, \mathbb{Y}_M)$ is the covariance matrix between $\mathbf{f}_{\boldsymbol{\phi}}(\mathbb{Y}_M)$ and $\mathbf{f}_{\boldsymbol{\phi}}(\mathbb{Y}_M)$ as we defined in Proposition A.2, and $K_{\mathbf{f}_{\boldsymbol{\phi}},\phi^{\text{type}}}(\mathbb{Y}_M, r^*) = K_{\phi^{\text{type}},\mathbf{f}_{\boldsymbol{\phi}}}(r^*, \mathbb{Y}_M)^T$ is the covariance matrix between $\mathbf{f}_{\boldsymbol{\phi}}(\mathbb{Y}_M)$ and $\phi^{\text{type}}(r^*)$, i.e., $K_{\mathbf{f}_{\boldsymbol{\phi}},\phi^{\text{type}}}(\mathbb{Y}_M, r^*) = (\text{Cov}(\mathbf{f}_{\boldsymbol{\phi}}(\boldsymbol{Y}^{(m,l)}), \phi^{\text{type}}(r^*)))_{m,l=1}^{M,L}$ and the i-th component of $\text{Cov}(\mathbf{f}_{\boldsymbol{\phi}}(\boldsymbol{Y}^{(m,l)}), \phi^{\text{type}}(r^*))$ is computed by

$$\text{Cov}([\mathbf{f}_{\boldsymbol{\phi}}(\boldsymbol{Y}^{(m,l)})]_i, \phi^E(r^*)) = \frac{1}{N}\sum_{k\neq i} K_{\theta^E}(r_{ik}^{\boldsymbol{X}^{(m,l)}}, r^*)\boldsymbol{r}_{ij}^{\boldsymbol{X}^{(m,l)}}, \tag{A.21}$$

$$\text{Cov}([\mathbf{f}_{\boldsymbol{\phi}}(\boldsymbol{Y}^{(m,l)})]_i, \phi^A(r^*)) = \frac{1}{N}\sum_{k\neq i} K_{\theta^A}(r_{ik}^{\boldsymbol{X}^{(m,l)}}, r^*)\boldsymbol{r}_{ij}^{\boldsymbol{V}^{(m,l)}}. \tag{A.22}$$

Note that $\boldsymbol{m}\boldsymbol{Z}_{\sigma^2}^{(m,l)} = F_{\boldsymbol{\alpha}}(\boldsymbol{Y}^{(m,l)}) + \mathbf{f}_{\phi}(\boldsymbol{X}^{(m,l)}) + \epsilon^{(m,l)}$ with i.i.d noise $\epsilon^{(m,l)} \sim \mathcal{N}(0, \sigma^2 I_{dN})$ for all (m, l), so we have

$$\begin{bmatrix} \boldsymbol{m}\mathbb{Z}_{\sigma^2,M} - F_{\boldsymbol{\alpha}}(\mathbb{Y}_M) \\ \phi^{\text{type}}(r^*) \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} K_{\mathbf{f}_{\phi}}(\mathbb{Y}_M, \mathbb{Y}_M) + \sigma^2 I_{dNML} & K_{\mathbf{f}_{\phi},\phi^{\text{type}}}(\mathbb{Y}_M, r^*) \\ K_{\phi^{\text{type}},\mathbf{f}_{\boldsymbol{\phi}}}(r^*, \mathbb{Y}_M) & K_{\theta^{\text{type}}}(r^*, r^*) \end{bmatrix}\right), \tag{A.23}$$

Therefore, based on the properties of the joint Gaussian distribution (see Lemma D.3), conditioning on $(\mathbb{Y}_M, \mathbb{Z}_{\sigma^2,M})$, we have that

$$p(\phi^{\text{type}}(r^*)|\mathbb{Y}_M, \mathbb{Z}_{\sigma^2,M}, r^*) \sim \mathcal{N}(\bar{\phi}^{\text{type}}, var(\bar{\phi}^{\text{type}})), \tag{A.24}$$

where $\bar{\phi}^{\text{type}}$ and $var(\bar{\phi}^{\text{type}})$ are defined as in (A.18) and (A.19). ∎

B. Psuedocode for Acceleration. In this section, we discuss in detail the acceleration of the computations used in our GP framework. We first review the bottleneck in the computation: our likelihood function evaluation is very slow, as it involves inverting the kernel matrix $(K_{\mathbf{f}_{\boldsymbol{\phi}}}(\mathbb{Y},\mathbb{Y};\theta)+\sigma^2 I)^{-1}$ and computing the log determinant $\log\det(K_{\mathbf{f}_{\boldsymbol{\phi}}}(\mathbb{Y},\mathbb{Y};\theta)+\sigma^2 I)$. We also require evaluation of the gradient for exact optimization, which further requires evaluation of the trace $\mathrm{Tr}((K_{\mathbf{f}_{\boldsymbol{\phi}}}(\mathbb{Y},\mathbb{Y};\theta)+\sigma^2 I)^{-1}\frac{\partial K_{\mathbf{f}_{\boldsymbol{\phi}}}(\mathbb{Y},\mathbb{Y};\theta)}{\partial\theta_i})$ for each parameter θ_i as shown in Proposition A.3.

Our primary goal is to avoid explicit inversion of the kernel matrix entirely by utilizing the Preconditioned Conjugate Gradient (PCG) algorithm, see Algorithm B.1. PCG is an iterative method that can solve systems $Ax=b$ for x without explicitly inverting A through clever choices of update at each step. This algorithm is central for scalability when solving large-scale linear systems with positive definite matrices in the numerical linear algebra literature. Note that the standard CG method is unlikely to work, as our kernel matrix is likely to be very ill-conditioned. An efficient preconditioner will be necessary to avoid extremely slow convergence. As mentioned in the main paper, we recommend the Randomized Gaussian Nystrom preconditioner for improving performance.

Algorithm B.1 Preconditioned CG for solving $(K_{\mathbf{f}_{\boldsymbol{\phi}}}(\mathbb{Y},\mathbb{Y};\theta)+\sigma^2 I)\boldsymbol{x}=\boldsymbol{b}$

Input: $K_{\mathbf{f}_{\boldsymbol{\phi}}}(\mathbb{Y},\mathbb{Y};\theta)+\sigma^2 I$ (matrix-vector multiplication of kernel), P (preconditioner), $\boldsymbol{b}$ (target vector), $\boldsymbol{x_0}$ (initial guess), errorTol (error tolerance), t (iterations)

$r_0 := \boldsymbol{b}-(K_{\mathbf{f}_{\boldsymbol{\phi}}}(\mathbb{Y},\mathbb{Y};\theta)+\sigma^2 I)\boldsymbol{x_0}$
$z_0, d_0 := Pr_0$
while $\|r_n\| >$ errorTol and $n < t$
  $v_n := (K_{\mathbf{f}_{\boldsymbol{\phi}}}(\mathbb{Y},\mathbb{Y};\theta)+\sigma^2 I)d_{n-1}$
  $\alpha_n := \frac{r_{n-1}^T z_{n-1}}{d_{n-1}^T v_n}$
  $x_n := x_{n-1}+\alpha_n d_{n-1}$
  $r_n := r_{n-1}-\alpha_n v_n$
  $z_n := Pr_n$
  $\beta_n := \frac{z_n^T r_n}{z_{n-1}^T r_{n-1}}$
  $d_n := z_n+\beta_n d_{n-1}$

Output: x_n (solution to $(K_{\mathbf{f}_{\boldsymbol{\phi}}}(\mathbb{Y},\mathbb{Y};\theta)+\sigma^2 I)\boldsymbol{x}=\boldsymbol{b}$),
$\{(\alpha_i,\beta_i)$ for all $i\le n\}$ (exclusively for constructing Lanczos weights)

Now we can solve the problem of slow likelihood function evaluations. Instead of inversion, our proper preconditioner will allow us to apply PCG and reduce the computational complexity from cubic for inversion to quadratic, see Algorithm B.1. Note that PCG is only limited by the runtime of matrix-vector multiplication, and in the presence of sparsity or other structural features that allow for linear time matrix-vector multiplication, the complexity of PCG will also reduce to linear time. This can be accomplished in the $\nu=\frac{1}{2}$ case using [69].

Then we consider the log determinant evaluation. Using stochastic Lanczos quadrature,

we can instead compute an estimator for $\mathrm{Tr}(\log((K_{\mathbf{f}_\phi}(\mathbb{Y},\mathbb{Y};\theta)+\sigma^2 I)))$, see Algorithm B.2. This algorithm requires quadrature weights, but these can be efficiently recovered by running the PCG algorithm and arranging α,β in a tridiagonal matrix, as seen in [70]. Then we apply stochastic trace estimation, as developed in [51]. These methods also extend to gradient calculations.

Algorithm B.2 Stochastic Trace Estimation with Lanczos Quadrature

Input: $(K_{\mathbf{f}_\phi}(\mathbb{Y},\mathbb{Y};\theta)+\sigma^2 I)$ (matrix-vector multiplication for kernel matrix), P (preconditioner), n (number of test vectors), m (number of Lanczos coefficients)

for i from 1 to n
  $v_n \sim$ Rademacher {draw from Rademacher distribution}
  $T := PCG((K_{\mathbf{f}_\phi}(\mathbb{Y},\mathbb{Y};\theta)+\sigma^2 I), P, v_n, \ell = m)$ {get Lanczos coefficients}
  $[W,\lambda] := eig(T)$
  for j from 1 to m
    $\gamma_i := \gamma_i + W_{1,j}^2 \log(\lambda_j)$
$tr_{est} := \log\det(P) + \frac{NdML}{n}\sum_{i=1}^n \gamma_i$

Output: tr_{est} (estimated trace of $\log((K_{\mathbf{f}_\phi}(\mathbb{Y},\mathbb{Y};\theta)+\sigma^2 I))$)

When choosing a preconditioner, we must have a method for fast and accurate computation of matrix-vector multiplication by P^{-1} and evaluation of $\log\det(P)$, as these are necessary operations in the above algorithms.

One widely applicable class of preconditioners for positive semi-definite matrices is the low-rank Nystrom approximation. The central idea is to create a low-rank approximation P of a matrix A of interest, with the expectation that $P^{-1}A$ will have a condition number close to 1. One common implementation is to subsample r columns of the matrix and use these to construct an approximation for the missing entries with rank at most r.

The randomized Gaussian Nystrom preconditioner builds on this idea. Written in a general form, we have $P = A\Omega(\Omega^T A\Omega)^\dagger \Omega^T A^T$ for a chosen matrix $\Omega \in \mathbb{R}^{NdML\times r}$. Column subsampling is a special case where columns of the matrix Ω have a single non-zero entry of the unit 1. However, Ω can also be populated with randomized Gaussian entries. This idea, developed in [52], has resulted in better empirical performance and enjoys theoretical support. For an implementation see Algorithm B.3.

C. Proof of the Coercivity condition in subsection 3.2.

Theorem C.1. *Consider* $\rho_{\boldsymbol{Y}} = \begin{bmatrix}\rho_{\boldsymbol{X}}\\ \rho_{\boldsymbol{V}}\end{bmatrix}$, *where* $\rho_{\boldsymbol{Y}}$ *is the product of* N *independent and identical measures with compact support on* $\mathbb{R}^d$ *and* $\rho_{\boldsymbol{V}}$ *is defined in the same way and is independent of* $\rho_{\boldsymbol{X}}$. *Then we have*

$$\|\mathbf{f}_{\boldsymbol{\varphi}}\|^2_{L^2(\rho_{\boldsymbol{Y}})} \geq \frac{N-1}{N^2}\|\varphi^E\|^2_{L^2(\tilde{\rho}_r^E)} + \frac{N-1}{N^2}\|\varphi^A\|^2_{L^2(\tilde{\rho}_r^A)} \tag{C.1}$$

Algorithm B.3 Randomized Gaussian Nystrom Preconditioner

Input: $K_{\mathbf{f}_\phi}(\mathbb{Y},\mathbb{Y};\theta)$ (matrix-vector multiplication for kernel matrix), σ^2 (noise hyperparameter), r (rank of preconditioner)

$\Omega \sim$ Standard Gaussian $\in \mathbb{R}^{NdML\times r}$
$R = \mathrm{QR}(\Omega)$ using economy QR
$Y = K_{\mathbf{f}_\phi}(\mathbb{Y},\mathbb{Y};\theta)R$
$\nu = \mathrm{eps}(||Y||_F)$
$Y_\nu = Y + \nu R$
$C = \mathrm{chol}(R^T Y_\nu)$
$B = Y_\nu / C$
$U, \Sigma = \mathrm{svd}(B)$ using economy svd
$\Lambda = \max(0, \Sigma^2 - \nu I)$
$P^{-1} = (\Lambda(-1) + \sigma^2)U(\Lambda + \sigma^2 I)^{-1}U^T + I - UU^T$
$\log\det(P) = \mathrm{sum}(\mathrm{sum}(\log(\frac{1}{\Lambda+\sigma^2})))$

Output: P^{-1}, $\log\det(P)$ (needed preconditioner quantities)

Proof. Following the definition of measure $\rho_{\boldsymbol{Y}}$ and the norm in (1.4), we have

$$\begin{aligned}\|\mathbf{f}_{\boldsymbol{\varphi}}\|^2_{L^2(\rho_{\boldsymbol{Y}})} &= \frac{1}{N}\sum_{i=1}^N \left\| \sum_{i'=1}^N \frac{1}{N}\Big[\varphi^E(|\boldsymbol{x}_{i'}-\boldsymbol{x}_i|)(\boldsymbol{x}_{i'}-\boldsymbol{x}_i) + \varphi^A(|\boldsymbol{x}_{i'}-\boldsymbol{x}_i|)(\boldsymbol{v}_{i'}-\boldsymbol{v}_i)\Big]\right\|^2_{L^2(\rho_{\boldsymbol{Y}})} \\ &= \frac{1}{N^3}\sum_{i=1}^N \left(\left(\sum_{j=k=1}^N + \sum_{j\neq k=1}^N \right) C^E_{i,j,k} + C^A_{i,j,k} + D_{i,j,k} \right) \\ &= \frac{N-1}{N^2}(\|\varphi^E\|^2_{L^2(\tilde\rho^E_r)} + \|\varphi^A\|^2_{L^2(\tilde\rho^A_r)}) + \mathcal{R} \qquad \text{(C.2)}\end{aligned}$$

where

$$\begin{aligned} C^E_{i,j,k} &= \langle \varphi^E(\|\boldsymbol{x}_j-\boldsymbol{x}_i\|)(\boldsymbol{x}_j-\boldsymbol{x}_i), \varphi^E(\|\boldsymbol{x}_k-\boldsymbol{x}_i\|)(\boldsymbol{x}_k-\boldsymbol{x}_i)\rangle_{L^2(\rho_{\boldsymbol{Y}})}, \\ C^A_{i,j,k} &= \langle \varphi^A(\|\boldsymbol{x}_j-\boldsymbol{x}_i\|)(\boldsymbol{v}_j-\boldsymbol{v}_i), \varphi^A(\|\boldsymbol{x}_k-\boldsymbol{x}_i\|)(\boldsymbol{v}_k-\boldsymbol{v}_i)\rangle_{L^2(\rho_{\boldsymbol{Y}})}, \\ D_{i,j,k} &= \langle \varphi^E(\|\boldsymbol{x}_j-\boldsymbol{x}_i)\|(\boldsymbol{x}_j-\boldsymbol{x}_i), \varphi^A(\|\boldsymbol{x}_k-\boldsymbol{x}_i\|)(\boldsymbol{v}_k-\boldsymbol{v}_i)\rangle_{L^2(\rho_{\boldsymbol{Y}})} \\ &\quad + \langle \varphi^A(\|\boldsymbol{x}_j-\boldsymbol{x}_i\|)(\boldsymbol{v}_j-\boldsymbol{v}_i), \varphi^E(\|\boldsymbol{x}_k-\boldsymbol{x}_i\|)(\boldsymbol{x}_k-\boldsymbol{x}_i)\rangle_{L^2(\rho_{\boldsymbol{Y}})} = 0, \\ \mathcal{R} &= \frac{1}{N^3}\sum_{i=1}^N \sum_{j\neq k, j\neq i, k\neq i} (C^A_{ijk} + C^E_{ijk}). \end{aligned}$$

By the property of $\rho_{\boldsymbol{Y}}$, when i,j,k are distinct, we have

$$\begin{aligned} C^E_{ijk} &= \mathbb{E}\big[\varphi^E(\|X_1-X_2\|)\varphi^E(\|X_1-X_3\|)\,\langle X_2-X_1, X_3-X_1\rangle\,\big] \\ C^A_{ijk} &= \mathbb{E}\big[\varphi^A(\|X_1-X_2\|)\varphi^A(\|X_1-X_3\|)\big]\mathbb{E}\big[\,\langle V_2-V_1, V_3-V_1\rangle\,\big], \end{aligned}$$

for all (i,j,k), where X_is and V_is are identical copies of the position and velocity variables $\boldsymbol{x}_i, \boldsymbol{v}_i$s. From the Lemma C.2 below,

$$C^E_{ijk} \geq 0, C^A_{ijk} \geq 0$$

and we used the fact

$$\mathbb{E}\big[\,\langle V_2 - V_1, V_3 - V_1\rangle\,\big] = \mathbb{E}(\|V_1\|^2) - \|\mathbb{E}(V_1)\|^2 \geq 0$$

Therefore,

$$\|\mathbf{f}_{\boldsymbol{\varphi}}\|^2_{L^2(\rho_{\boldsymbol{Y}})} \geq \frac{N-1}{N^2}(\|\varphi^E\|^2_{L^2(\tilde{\rho}^E_r)} + \|\varphi^A\|^2_{L^2(\tilde{\rho}^A_r)})$$ ■

The proof of Theorem C.1 uses the following lemma.

Lemma C.2. *If X, Y, Z are i.i.d random vectors, then for any measurable function g on $\mathbb{R}^d$, we have that*

$$\mathbb{E}[g(X-Y)g(X-Z)\langle X-Y, X-Z\rangle] \geq 0,$$
$$\mathbb{E}[g(X-Y)g(X-Z)] \geq 0,$$

provided the expectation exists.

Proof. Without loss of generality, suppose the probability density function of X is $p(x)$. (The discrete distribution case follows from the same argument). Let $(U,V) = (X-Y, X-Z)$. By the independence of X, Y, Z, the pdf of (U,V) is

$$p(u,v) = \int p(x)p(x-u)p(x-v)dx.$$

Since

$$\sum_{i=1}^N \sum_{j=1}^N c_i \bar{c}_j p(u_i, u_j) = \int p(x)|\sum_{i=1}^N c_i p(x-u_i)|^2 dx \geq 0,$$

which means $p(u,v)$ is positive definite (p.d.) As $\langle u, v\rangle$ is p.d and $g(u)g(v)$ is p.d. [71], we get $g(u)g(v)\langle u, v\rangle p(u,v)$ is p.d.. Note that

$$\mathbb{E}[g(X-Y)g(X-Z)\langle X-Y, X-Z\rangle] = \int_{\mathbb{R}^{2d}} g(u)g(v)\langle u, v\rangle p(u,v)dudv, \tag{C.3}$$

if the function $g(u)g(v)\langle u, v\rangle p(u,v)$ is measurable and integrable. Then the inequality holds by p.d. property. Similarly, one can prove the second inequality. ■

D. Proof of Representer Theorem. We prove the Representer Theorem (Theorem 3.2 in main text subsection 3.1) by using an operator-theoretic approach.

Proposition D.1. *Given the empirical noisy trajectory data $(\mathbb{Y}_M, \mathbb{Z}_{\sigma^2,M}) = \{\mathbb{X}_M, \mathbb{V}_M, \mathbb{Z}_{\sigma^2,M}\}$. We define the sampling operator $A_M : \mathcal{H}_{K^E} \times \mathcal{H}_{K^A} \to \mathbb{R}^{dNML}$ by*

$$A_M\boldsymbol{\varphi} = \mathbf{f}_{\boldsymbol{\varphi}}(\mathbb{X}_M) := \mathrm{Vec}(\{\mathbf{f}_{\boldsymbol{\varphi}}(\boldsymbol{Y}^{(m,l)})\}^{M,L}_{m,l=1}) = \mathrm{Vec}(\{\mathbf{f}_{\varphi^E}(\boldsymbol{Y}^{(m,l)}) + \mathbf{f}_{\varphi^A}(\boldsymbol{Y}^{(m,l)})\}^{M,L}_{m,l=1}), \tag{D.1}$$

where $\mathbb{R}^{dNML}$ is equipped with the inner product defined in (1.3).

1. *The adjoint operator A_M^* is a finite rank operator. For any noise vector $\mathbb{W}$ in $\mathbb{R}^{dNML}$, let $\mathbb{W}_{m,l,i} \in \mathbb{R}^d$ denote the i-th component of (m,l)th block of $\mathbb{W}$ as the same way in $\mathbb{Y}_M$, then we have*

$$A_M^*\mathbb{W} = \Big(\frac{1}{LM} \sum_{l,m=1}^{L,M} \sum_{i=1,i'\neq i}^{N} \frac{1}{N^2} K^E_{r^{\boldsymbol{X}^{(m,l)}}_{ii'}} \langle \boldsymbol{r}^{\boldsymbol{X}^{(m,l)}}_{ii'}, \mathbb{W}_{m,l,i} \rangle, \tag{D.2}$$
$$\frac{1}{LM} \sum_{l,m=1}^{L,M} \sum_{i=1,i'\neq i}^{N} \frac{1}{N^2} K^A_{r^{\boldsymbol{X}^{(m,l)}}_{ii'}} \langle \boldsymbol{r}^{\boldsymbol{V}^{(m,l)}}_{ii'}, \mathbb{W}_{m,l,i} \rangle \Big).$$

For any function $\boldsymbol{\varphi} \in \mathcal{H}_{K^E} \times \mathcal{H}_{K^A}$, we have that

$$B_M\boldsymbol{\varphi} := A_M^* A_M \boldsymbol{\varphi} = \Big(\frac{1}{LM} \sum_{l,m=1}^{L,M} \sum_{i=1,i',i''\neq i}^{N} \frac{1}{N^3} K^E_{r^{\boldsymbol{X}^{(m,l)}}_{ii'}} (\langle \varphi^E, K^E_{r^{\boldsymbol{X}^{(m,l)}}_{ii''}} \rangle_{\mathcal{H}_{K^E}} \langle \boldsymbol{r}^{\boldsymbol{X}^{(m,l)}}_{ii'}, \boldsymbol{r}^{\boldsymbol{X}^{(m,l)}}_{ii''} \rangle \tag{D.3}$$
$$+ \langle \varphi^A, K^A_{r^{\boldsymbol{X}^{(m,l)}}_{ii''}} \rangle_{\mathcal{H}_{K^A}} \langle \boldsymbol{r}^{\boldsymbol{X}^{(m,l)}}_{ii'}, \boldsymbol{r}^{\boldsymbol{V}^{(m,l)}}_{ii''} \rangle),$$
$$\frac{1}{LM} \sum_{l,m=1}^{L,M} \sum_{i=1,i',i''\neq i}^{N} \frac{1}{N^3} K^A_{r^{\boldsymbol{X}^{(m,l)}}_{ii'}} (\langle \varphi^A, K^A_{r^{\boldsymbol{X}^{(m,l)}}_{ii''}} \rangle_{\mathcal{H}_{K^A}} \langle \boldsymbol{r}^{\boldsymbol{V}^{(m,l)}}_{ii'}, \boldsymbol{r}^{\boldsymbol{V}^{(m,l)}}_{ii''} \rangle \tag{D.4}$$
$$+ \langle \varphi^E, K^E_{r^{\boldsymbol{X}^{(m,l)}}_{ii''}} \rangle_{\mathcal{H}_{K^E}} \langle \boldsymbol{r}^{\boldsymbol{V}^{(m,l)}}_{ii'}, \boldsymbol{r}^{\boldsymbol{X}^{(m,l)}}_{ii''} \rangle) \Big). \tag{D.5}$$

2. *If $\boldsymbol{\lambda} = (\lambda^E, \lambda^A) > 0$, a unique minimizer $\phi^{\boldsymbol{\lambda},M}_{\mathcal{H}_{K^E}\times\mathcal{H}_{K^A}}$ that solves*

$$\operatorname*{arg\,min}_{\boldsymbol{\varphi}\in\mathcal{H}_{K^E}\times\mathcal{H}_{K^A}} \mathcal{E}^{\boldsymbol{\lambda},M}(\boldsymbol{\varphi}) := \|A_M\boldsymbol{\varphi} - \mathbb{Z}_{\sigma^2,M}\|^2 + \|\sqrt{\boldsymbol{\lambda}} \cdot \boldsymbol{\varphi}\|^2_{\mathcal{H}_{K^E}\times\mathcal{H}_{K^A}}$$

exists and is given by

$$\phi^{\boldsymbol{\lambda},M}_{\mathcal{H}_{K^E}\times\mathcal{H}_{K^A}} = (B_M + \boldsymbol{\lambda})^{-1} A_M^* \mathbb{Z}_{\sigma^2,M}. \tag{D.6}$$

where we interpret the map $\boldsymbol{\lambda}(\boldsymbol{\phi})$ by $\boldsymbol{\lambda} \cdot \boldsymbol{\phi} = (\lambda^E\phi^E, \lambda^A\phi^A)$.

Proof. The part 1 of Proposition D.1 can be derived by using the identity $\langle A_M\boldsymbol{\varphi}, \boldsymbol{w}\rangle = \langle \boldsymbol{\varphi}, A_M^*\boldsymbol{w}\rangle_{\mathcal{H}_{K^E}\times\mathcal{H}_{K^A}}$. Part 2 of Proposition D.1 is straightforward by solving the normal equation. ∎

Now we derive a basis representation formula for the empirical minimizer of (D.6)

Theorem D.2. *If $\boldsymbol{\lambda} > 0$, then the minimizer of the regularized empirical risk functional $\mathcal{E}^{\boldsymbol{\lambda},M}(\cdot)$ has the form*

$$\phi^{\lambda,M}_{\mathcal{H}_{K^E}\times\mathcal{H}_{K^A}} = \Big(\sum_{r^x \in r_{\mathbb{X}_M}} \hat{c}_{r^x} K^E_{r^x}, \sum_{(r^x,r^v)\in(r_{\mathbb{X}_M}\times r_{\mathbb{V}_M})} \hat{c}_{r^v} K^A_{r^x} \Big), \tag{D.7}$$

where $r_{\mathbb{X}_M} \in \mathbb{R}^{MLN^2}$ *is the set contains all the pair distances in* $\mathbb{X}_M$*, i.e.*

(D.8)
$$r_{\mathbb{X}_M} = \left[r_{11}^{(1,1)}, \dots, r_{1N}^{(1,1)}, \dots, r_{N1}^{(1,1)}, \dots, r_{NN}^{(1,1)}, \dots, r_{11}^{(M,L)}, \dots, r_{1N}^{(M,L)}, \dots, r_{N1}^{(M,L)}, \dots, r_{NN}^{(M,L)}\right]^T,$$

and $r_{\mathbb{X}_M} \times r_{\mathbb{V}_M} \in \mathbb{R}^{MLN^2 \times MLN^2}$ *is the set contains all the pair distances in* $\mathbb{X}_M$ *and their associated pair distances in* $\mathbb{V}_M$*.*
Moreover, we have

$$\begin{aligned}\hat{c}_{r^x} &= \frac{1}{N}\boldsymbol{r}_{\mathbb{X}_M}^T \cdot (K_{\mathbf{f}_{\boldsymbol{\phi}}}(\mathbb{Y}_M, \mathbb{Y}_M) + \lambda^E NMLI)^{-1}\mathbb{Z}_{\sigma^2,M},\\ \hat{c}_{r^v} &= \frac{1}{N}\boldsymbol{r}_{\mathbb{V}_M}^T \cdot (K_{\mathbf{f}_{\boldsymbol{\phi}}}(\mathbb{Y}_M, \mathbb{Y}_M) + \lambda^A NMLI)^{-1}\mathbb{Z}_{\sigma^2,M},\end{aligned} \tag{D.9}$$

where the block-diagonal matrix $\boldsymbol{r}_{\mathbb{X}_M} = \mathrm{diag}(\boldsymbol{r}_{\boldsymbol{X}^{(m,l)}}) \in \mathbb{R}^{MLdN \times MLN^2}$ *and* $\boldsymbol{r}_{\boldsymbol{X}^{(m,l)}} \in \mathbb{R}^{dN \times N^2}$ *defined by*

$$\boldsymbol{r}_{\boldsymbol{X}^{(m,l)}} = \begin{bmatrix} \boldsymbol{r}_{11}^{(m,l)}, \dots, \boldsymbol{r}_{1N}^{(m,l)} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \boldsymbol{r}_{21}^{(m,l)}, \dots, \boldsymbol{r}_{2N}^{(m,l)} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \boldsymbol{r}_{N1}^{(m,l)}, \dots, \boldsymbol{r}_{NN}^{(m,l)} \end{bmatrix}, \tag{D.10}$$

and same for $\boldsymbol{r}_{\mathbb{V}_M}$*.*

Proof. Let $\mathcal{H}_{K^E,M}$ be the subspace of $\mathcal{H}_{K^E}$ spanned by the set of functions $\{K_r^E : r \in r_{\mathbb{X}_M}\}$, and similarly for $\mathcal{H}_{K^A,M}$. By Proposition Proposition D.1, we know that $B_M(\mathcal{H}_{K,M}^E \times \mathcal{H}_{K,M}^A) \subset \mathcal{H}_{K,M}^E \times \mathcal{H}_{K,M}^A$. Since B_M is self-adjoint and compact, by the spectral theory of self-adjoint compact operator (see [72]), $\mathcal{H}_{K,M}^E \times \mathcal{H}_{K,M}^A$ is also an invariant subspace for the operator $(B_M + \lambda I)^{-1}$. Then by (D.6), there exists vectors $\hat{c}_{r^x}$, $\hat{c}_{r^v}$ such that

$$\phi^{\lambda,M}_{\mathcal{H}_{K^E \times K^A}} = \Big(\sum_{r^x \in r_{\mathbb{X}_M}} \hat{c}_{r^x} K_{r^x}^E, \sum_{(r^x,r^v) \in (r_{\mathbb{X}_M} \times r_{\mathbb{X}_M})} \hat{c}_{r^v} K_{r^x}^A\Big). \tag{D.11}$$

Then, multiplying $(B_M + \boldsymbol{\lambda})$ on both sides of (D.6) and plugging in (D.11), we can obtain

$$\begin{cases} \big(\boldsymbol{r}_{\mathbb{X}_M}^T \boldsymbol{r}_{\mathbb{X}_M} K^E(r_{\mathbb{X}_M}, r_{\mathbb{X}_M}) + \lambda^E N^3 MLI\big)\hat{c}_{r^x} + \boldsymbol{r}_{\mathbb{X}_M}^T \boldsymbol{r}_{\mathbb{V}_M} K^A(r_{\mathbb{X}_M}, r_{\mathbb{X}_M})\hat{c}_{r^v} &= N\boldsymbol{r}_{\mathbb{X}_M}^T \mathbb{Z}_{\sigma^2,M} \\ \big(\boldsymbol{r}_{\mathbb{V}_M}^T \boldsymbol{r}_{\mathbb{V}_M} K^A(r_{\mathbb{X}_M}, r_{\mathbb{X}_M}) + \lambda^A N^3 MLI\big)\hat{c}_{r^v} + \boldsymbol{r}_{\mathbb{V}_M}^T \boldsymbol{r}_{\mathbb{X}_M} K^E(r_{\mathbb{X}_M}, r_{\mathbb{X}_M})\hat{c}_{r^x} &= N\boldsymbol{r}_{\mathbb{V}_M}^T \mathbb{Z}_{\sigma^2,M} \end{cases}$$

using the matrix representation of $(B_M + \boldsymbol{\lambda})$ with respect to the spanning sets $\{K_r^E : r \in r_{\mathbb{X}_M}\}$ and $\{K_r^A : r \in r_{\mathbb{X}_M}\}$.

Recall that we have $K^E(r_{\mathbb{X}_M}, r_{\mathbb{X}_M}) = (K^E(r_{ij}, r_{i'j'}))_{r_{ij}, r_{i'j'} \in r_{\mathbb{X}_M}}$, $K^A(r_{\mathbb{X}_M}, r_{\mathbb{X}_M}) = (K^A(r_{ij}, r_{i'j'}))_{r_{ij}, r_{i'j'} \in r_{\mathbb{X}_M}}$ and $K_{\mathbf{f}_{\boldsymbol{\phi}}}(\mathbb{Y}_M, \mathbb{Y}_M) = \mathrm{Cov}(\mathbf{f}_{\boldsymbol{\phi}}(\mathbb{Y}_M), \mathbf{f}_{\boldsymbol{\phi}}(\mathbb{Y}_M))$, so using the identity

$$\boldsymbol{r}_{\mathbb{X}_M} K^E(r_{\mathbb{X}_M}, r_{\mathbb{X}_M})\boldsymbol{r}_{\mathbb{X}_M}^T + \boldsymbol{r}_{\mathbb{V}_M} K^A(r_{\mathbb{X}_M}, r_{\mathbb{X}_M})\boldsymbol{r}_{\mathbb{V}_M}^T = N^2 K_{\mathbf{f}_{\boldsymbol{\phi}}}(\mathbb{Y}_M, \mathbb{Y}_M) \tag{D.12}$$

and the fact that the matrices $\left(\boldsymbol{r}_{\mathbb{X}_M}^T \boldsymbol{r}_{\mathbb{X}_M} K^E(r_{\mathbb{X}_M}, r_{\mathbb{X}_M}) + \lambda^E N^3 MLI\right)$, $\left(\boldsymbol{r}_{\mathbb{V}_M}^T \boldsymbol{r}_{\mathbb{V}_M} K^A(r_{\mathbb{X}_M}, r_{\mathbb{X}_M}) + \lambda^A N^3 MLI\right)$ are invertible, one can verify that

$$\begin{cases} \hat{c}_{r^x} &= \frac{1}{N} \boldsymbol{r}_{\mathbb{X}_M}^T \cdot (K_{\mathbf{f}_{\boldsymbol{\phi}}}(\mathbb{Y}_M, \mathbb{Y}_M) + \lambda^E NMLI)^{-1} \mathbb{Z}_{\sigma^2, M}, \\ \hat{c}_{r^v} &= \frac{1}{N} \boldsymbol{r}_{\mathbb{V}_M}^T \cdot (K_{\mathbf{f}_{\boldsymbol{\phi}}}(\mathbb{Y}_M, \mathbb{Y}_M) + \lambda^A NMLI)^{-1} \mathbb{Z}_{\sigma^2, M}, \end{cases} \tag{D.13}$$

is the solution. ∎

Now we are ready to finish the proof of the Representer theorem.

Proof. Let $\tilde{K}^E = \frac{\sigma^2 K^E}{MNL\lambda^E}$, $\tilde{K}^A = \frac{\sigma^2 K^A}{MNL\lambda^A}$.
Since $\phi^E \sim \mathcal{GP}(0, \tilde{K}^E)$, $\phi^A \sim \mathcal{GP}(0, \tilde{K}^A)$, the posterior mean in (2.8) will then become

$$\begin{aligned} \bar{\phi}_M^E(r^*) &= \tilde{K}_{\phi^E, \mathbf{f}_{\boldsymbol{\phi}}}(r^*, \mathbb{X}_M)(\tilde{K}_{\mathbf{f}_{\boldsymbol{\phi}}}(\mathbb{Y}_M, \mathbb{Y}_M) + \sigma^2 I)^{-1} \mathbb{Z}_{\sigma^2, M} \\ &= \frac{1}{N} \tilde{K}^E_{r^T_{\mathbb{X}_M}}(r^*) \boldsymbol{r}_{\mathbb{X}_M}^T (\tilde{K}_{\mathbf{f}_{\boldsymbol{\phi}}}(\mathbb{Y}_M, \mathbb{Y}_M) + \sigma^2 I)^{-1} \mathbb{Z}_{\sigma^2, M} \\ &= \frac{1}{N} K^E_{r^T_{\mathbb{X}_M}}(r^*) \boldsymbol{r}_{\mathbb{X}_M}^T (K_{\mathbf{f}_{\boldsymbol{\phi}}}(\mathbb{Y}_M, \mathbb{Y}_M) + NML\lambda^E I)^{-1} \mathbb{Z}_{\sigma^2, M} \\ &= K_{\phi^E, \mathbf{f}_{\boldsymbol{\phi}}}(r^*, \mathbb{X}_M)(K_{\mathbf{f}_{\boldsymbol{\phi}}}(\mathbb{Y}_M, \mathbb{Y}_M) + NML\lambda^E I)^{-1} \mathbb{Z}_{\sigma^2, M} \\ &= \sum_{r \in r_{\mathbb{X}_M}} \hat{c}_r K_r^E, \end{aligned}$$

where $\hat{c}_r$ is defined in (D.9) and we used the identity $K_{\phi^E, \mathbf{f}_{\boldsymbol{\phi}}}(r^*, \mathbb{X}_M) = \frac{1}{N} K^E_{r^T_{\mathbb{X}_M}}(r^*) \boldsymbol{r}_{\mathbb{X}_M}^T$ (also for $\tilde{K}$) in the proof. Similarly, we can get the posterior mean for $\bar{\phi}_M^A(r^*)$. ∎

Lemma D.3. *Let $\boldsymbol{x}$ and $\boldsymbol{y}$ be jointly Gaussian random vectors*

$$\begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{y} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu_{\boldsymbol{x}} \\ \mu_{\boldsymbol{y}} \end{bmatrix}, \begin{bmatrix} A & C \\ C^T & B \end{bmatrix} \right), \tag{D.14}$$

then the marginal distribution of $\boldsymbol{x}$ and the conditional distribution of $\boldsymbol{x}$ given $\boldsymbol{y}$ are

$$\boldsymbol{x} \sim \mathcal{N}(\mu_{\boldsymbol{x}}, A), \quad \text{and } \boldsymbol{x}|\boldsymbol{y} \sim \mathcal{N}(\mu_{\boldsymbol{x}} + CB^{-1}(\boldsymbol{y} - \mu_{\boldsymbol{y}}), A - CB^{-1}C^T). \tag{D.15}$$

Proof. See, e.g. [50], Appendix A. ∎

REFERENCES

[1] Fei Lu, Ming Zhong, Sui Tang, and Mauro Maggioni. Nonparametric inference of interaction laws in systems of agents from trajectory data. *Proceedings of the National Academy of Sciences*, 116(29):14424–14433, 2019.

[2] Fei Lu, Mauro Maggioni, and Sui Tang. Learning interaction kernels in stochastic systems of interacting particles from multiple trajectories. *arXiv preprint arXiv:2007.15174*, 2020.

[3] Fei Lu, Mauro Maggioni, and Sui Tang. Learning interaction kernels in heterogeneous systems of agents from multiple trajectories. *Journal of Machine Learning Research*, 22(32):1–67, 2021.

[4] Jason Miller, Sui Tang, Ming Zhong, and Mauro Maggioni. Learning theory for inferring interaction kernels in second-order interacting agent systems. *arXiv preprint arXiv:2010.03729*, 2020.

[5] M. Schmidt and H. Lipson. Distilling free-form natural laws from experimental data. *Science*, 324(5923):81–85, 2009.

[6] S. Brunton, N. Kutz, and J. Proctor. Data-driven discovery of governing physical laws. *SIAM News*, 50(1), 2017.

[7] Sheng Zhang and Guang Lin. Robust data-driven discovery of governing physical laws with error bars. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 474(2217):20180305, 2018.

[8] M. Ballerini, N. Cabibbo, R. Candelier, A. Cavagna, E. Cisbani, I. Giardina, V. Lecomte, A. Orlandi, G. Parisi, A. Procaccini, M. Viale, and V. Zdravkovic. Interaction ruling animal collective behavior depends on topological rather than metric distance: Evidence from a field study. *Proc Natl Acad Sci USA*, 105(4):1232–1237, 2008.

[9] W. Bialek, A. Cavagna, I. Giardina, T. Mora, E. Silvestri, M. Viale, and A. M. Walzak. Statistical mechanics for natural flocks of birds. *Proc Natl Acad Sci USA*, 109:4786 – 4791, 2012.

[10] Sebastien Motsch and Eitan Tadmor. Heterophilious dynamics enhances consensus. *SIAM review*, 56(4):577–621, 2014.

[11] Felipe Cucker and Steve Smale. On the mathematics of emergence. *Japanese Journal of Mathematics*, 2(1):197–227, 2007.

[12] Maria R D'Orsogna, Yao-Li Chuang, Andrea L Bertozzi, and Lincoln S Chayes. Self-propelled particles with soft-core interactions: patterns, stability, and collapse. *Physical review letters*, 96(10):104302, 2006.

[13] Ruiwen Shu and Eitan Tadmor. Anticipation breeds alignment. *Archive for Rational Mechanics and Analysis*, 240(1):203–241, 2021.

[14] J. Bongard and H. Lipson. Automated reverse engineering of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences of the United States of America*, 104(24):9943–9948, 2007.

[15] S. Brunton, J. Proctor, and J. Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences of the United States of America*, 113(15):3932–3937, 2016.

[16] S. Rudy, S. Brunton, J. Proctor, and N. Kutz. Data-driven discovery of partial differential equations. *Science Advances*, 3(4):e1602614, 2017.

[17] X. Han, Z. Shen, W. Wang, and Z. Di. Robust reconstruction of complex networks from sparse data. *Physical Review Letters*, 114(2):028701, 2015.

[18] S. Kang, W. Liao, and Y. Liu. Ident: Identifying differential equations with numerical time evolution. *arXiv preprint arXiv:1904.03538*, 2019.

[19] M. Raissi. Deep hidden physics models: Deep learning of nonlinear partial differential equations. *The Journal of Machine Learning Research*, 19(1):932–955, 2018.

[20] M. Raissi and G. Karniadakis. Hidden physics models: Machine learning of nonlinear partial differential equations. *Journal of Computational Physics*, 357:125–141, 2018.

[21] Z. Long, Y. Lu, X. Ma, and B. Dong. PDE-net: Learning PDEs from data. *arXiv preprint arXiv:1710.09668*, 2017.

[22] Lu Lu, Xuhui Meng, Zhiping Mao, and George Em Karniadakis. Deepxde: A deep learning library for solving differential equations. *SIAM Review*, 63(1):208–228, 2021.

[23] G. Tran and R. Ward. Exact recovery of chaotic systems from highly corrupted data. *Multiscale Modeling and Simulation*, 15(3):1108–1129, 2017.

[24] H. Schaeffer, G. Tran, and R. Ward. Extracting sparse high-dimensional dynamics from limited data. *SIAM Journal on Applied Mathematics*, 78(6):3279–3295, 2018.

[25] L. Boninsegna, F. Nüske, and C. Clementi. Sparse learning of stochastic dynamical equations. *The Journal of Chemical Physics*, 148(24):241723, 2018.

[26] Markus Heinonen, Cagatay Yildiz, Henrik Mannerström, Jukka Intosalmi, and Harri Lähdesmäki. Learning unknown ODE models with Gaussian processes. In *International Conference on Machine Learning*, pages 1959–1968. PMLR, 2018.

[27] Cedric Archambeau, Dan Cornford, Manfred Opper, and John Shawe-Taylor. Gaussian process approximations of stochastic differential equations. In *Gaussian Processes in Practice*, pages 1–16. PMLR, 2007.

[28] Cagatay Yildiz, Markus Heinonen, Jukka Intosalmi, Henrik Mannerstrom, and Harri Lahdesmaki. Learn-

ing stochastic differential equations with Gaussian processes without gradient matching. In *2018 IEEE 28th International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6. IEEE, 2018.

[29] Zheng Zhao, Filip Tronarp, Roland Hostettler, and Simo Särkkä. State-space Gaussian process for drift estimation in stochastic differential equations. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5295–5299. IEEE, 2020.

[30] Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. Machine learning of linear differential equations using Gaussian processes. *Journal of Computational Physics*, 348:683–693, 2017.

[31] Jiuhai Chen, Lulu Kang, and Guang Lin. Gaussian process assisted active learning of physical laws. *Technometrics*, pages 1–14, 2020.

[32] Hongqiao Wang and Xiang Zhou. Explicit estimation of derivatives from data and differential equations by Gaussian process regression. *International Journal for Uncertainty Quantification*, 11(4), 2021.

[33] Yifan Chen, Bamdad Hosseini, Houman Owhadi, and Andrew M. Stuart. Solving and learning nonlinear PDEs with Gaussian processes. *Journal of Computational Physics*, 447, 2021.

[34] Seungjoon Lee, Mahdi Kooshkbaghi, Konstantinos Spiliotis, Constantinos I Siettos, and Ioannis G Kevrekidis. Coarse-scale PDEs from fine-scale observations via machine learning. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 30(1):013141, 2020.

[35] Jean-Luc Akian, Luc Bonnet, Houman Owhadi, and Éric Savin. Learning "best" kernels from data in Gaussian process regression. with application to aerodynamics. *arXiv preprint arXiv:2206.02563*, 2022.

[36] Matthieu Darcy, Boumediene Hamzi, Jouni Susiluoto, Amy Braverman, and Houman Owhadi. Learning dynamical systems from data: a simple cross-validation perspective, part ii: nonparametric kernel flows. *preprint*, 2021.

[37] Fanghui Liu, Xiaolin Huang, Yudong Chen, and Johan AK Suykens. Random features for kernel approximation: A survey on algorithms, theory, and beyond. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(10):7128–7148, 2021.

[38] Quanjun Lang and Fei Lu. Learning interaction kernels in mean-field equations of 1st-order systems of interacting particles. *arXiv preprint arXiv:2010.15694*, 2020.

[39] Quanjun Lang and Fei Lu. Identifiability of interaction kernels in mean-field equations of interacting particles. *arXiv preprint arXiv:2106.05565*, 2021.

[40] Yuchen He, Sung Ha Kang, Wenjing Liao, Hao Liu, and Yingjie Liu. Numerical identification of nonlocal potential in aggregation. *arXiv preprint arXiv:2207.03358*, 2022.

[41] Felix P Kemeth, Tom Bertalan, Thomas Thiem, Felix Dietrich, Sung Joon Moon, Carlo R Laing, and Ioannis G Kevrekidis. Learning emergent partial differential equations in a learned emergent space. *Nature Communications*, 13(1):1–13, 2022.

[42] Sui Tang, Malik Tuerkoen, and Hanming Zhou. On the identifiablility of nonlocal interaction kernels in first-order systems of interacting particles on riemannian manifolds. *arXiv preprint arXiv:2305.12340*, 2023.

[43] Jinchao Feng, Charles Kulick, Yunxiang Ren, and Sui Tang. Learning particle models of swarming from data with Gaussian processes. *arXiv preprint arXiv:2106.02735*, 2022.

[44] Michael E Tipping. Sparse Bayesian learning and the relevance vector machine. *Journal of machine learning research*, 1(Jun):211–244, 2001.

[45] Bernhard Schölkopf, Alexander J Smola, Francis Bach, et al. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2002.

[46] Vladimir Vapnik. *The nature of statistical learning theory*. Springer science & business media, 2013.

[47] Andrei Nikolajevits Tihonov. Solution of incorrectly formulated problems and the regularization method. *Soviet Math.*, 4:1035–1038, 1963.

[48] Andrei Nikolaevich Tikhonov, AV Goncharsky, VV Stepanov, and Anatoly G Yagola. *Numerical methods for the solution of ill-posed problems*, volume 328. Springer Science & Business Media, 2013.

[49] Tomaso Poggio and Federico Girosi. Networks for approximation and learning. *Proceedings of the IEEE*, 78(9):1481–1497, 1990.

[50] Christopher KI Williams and Carl Edward Rasmussen. *Gaussian processes for machine learning*, volume 2. MIT press Cambridge, MA, 2006.

[51] Jonathan Wenger, Geoff Pleiss, Philipp Hennig, John Cunningham, and Jacob Gardner. Preconditioning

for scalable Gaussian process hyperparameter optimization. *arXiv preprint arXiv:2017.00243*, 2022.
[52] Zachary Frangella, Joel Tropp, and Madeleine Udell. Randomized Nyström preconditioning. *arXiv preprint arXiv:2110.02820*, 2021.
[53] Raphael A. Meyer, Cameron Musco, et al. Hutch++: Optimal stochastic trace estimation. *arXiv preprint arXiv:2010.09649*, 2020.
[54] Houman Owhadi and Clint Scovel. *Operator-adapted wavelets, fast solvers, and numerical homogenization: from a game theoretic approach to numerical approximation and algorithm design*, volume 35. Cambridge University Press, 2019.
[55] Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 113(15):3932–3937, 2016.
[56] Felipe Cucker and Steve Smale. Emergent behavior in flocks. *IEEE Transactions on automatic control*, 52(5):852–862, 2007.
[57] Felipe Cucker and Jiu-Gang Dong. A general collision-avoiding flocking framework. *IEEE Transactions on Automatic Control*, 56(5):1124–1129, 2011.
[58] Roman Shvydkoy et al. *Dynamics and analysis of alignment models of collective behavior*. Springer, 2021.
[59] Shin Mi Ahn, Heesun Choi, Seung-Yeal Ha, and Ho Lee. On collision-avoiding initial configurations to Cucker-Smale type flocking models. *Communications in Mathematical Sciences*, 10(2):625–643, 2012.
[60] Young-Pil Choi, Seung-Yeal Ha, and Zhuchun Li. Emergent dynamics of the Cucker-Smale flocking model and its variants. In *Active Particles, Volume 1*, pages 299–331. Springer, 2017.
[61] Yao-Li Chuang, Maria R D'orsogna, Daniel Marthaler, Andrea L Bertozzi, and Lincoln S Chayes. State transitions and the continuum limit for a 2d interacting, self-propelled particle system. *Physica D: Nonlinear Phenomena*, 232(1):33–47, 2007.
[62] Nicole Abaid and Maurizio Porfiri. Fish in a ring: spatio-temporal pattern formation in one-dimensional animal groups. *Journal of The Royal Society Interface*, 7(51):1441–1453, 2010.
[63] Ryan Lukeman, Yue-Xian Li, and Leah Edelstein-Keshet. A conceptual model for milling formations in biological aggregates. *Bulletin of mathematical biology*, 71(2):352, 2009.
[64] Dhananjay Bhaskar, Angelika Manhart, Jesse Milzman, John T Nardini, Kathleen M Storey, Chad M Topaz, and Lori Ziegelmeier. Analyzing collective motion with machine learning and topology. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 29(12):123125, 2019.
[65] Yael Katz, Kolbjørn Tunstrøm, Christos C Ioannou, Cristián Huepe, and Iain D Couzin. Inferring the structure and dynamics of interactions in schooling fish. *Proceedings of the National Academy of Sciences*, 108(46):18720–18725, 2011.
[66] Jitesh Jhawar, Richard G Morris, UR Amith-Kumar, M Danny Raj, Tim Rogers, Harikrishnan Rajendran, and Vishwesha Guttal. Noise-induced schooling of fish. *Nature Physics*, 16(4):488–493, 2020.
[67] Mengyang Gu, Xubo Liu, et al. Scalable marginalization of correlated latent variables with applications to learning particle interaction kernels. *arXiv preprint arXiv:2203.08389*, 2022.
[68] Dong C Liu and Jorge Nocedal. On the limited memory BFGS method for large scale optimization. *Mathematical programming*, 45(1):503–528, 1989.
[69] Mengyang Gu, Xubo Liu, Xinyi Fang, and Sui Tang. Scalable marginalization of latent variables for correlated data. *arXiv preprint arXiv:2203.08389*, 2022.
[70] Jacob Gardner, Geoff Pleiss, et al. GPyTorch: Blackbox matrix-matrix Gaussian process inference with GPU acceleration. *arXiv preprint arXiv:1809:11165v6*, 2021.
[71] Zhongyang Li, Fei Lu, Mauro Maggioni, Sui Tang, and Cheng Zhang. On the identifiability of interaction functions in systems of interacting particles. *Stochastic Processes and their Applications*, 132:135–163, 2021.
[72] Jiri Blank, Pavel Exner, and Miloslav Havlicek. *Hilbert space operators in quantum physics*. Springer Science & Business Media, 2008.