

Inferring microscale properties of interacting systems from macroscale observations

Nazareno Campioni ¹, Dirk Husmeier ¹, Juan Morales,² Jennifer Gaskell,¹ and Colin J. Torney ¹

¹*School of Mathematics & Statistics, University of Glasgow, University Avenue, Glasgow, G12 8QQ, United Kingdom*

²*Grupo de Ecología Cuantitativa, INIBIOMA-CONICET, Universidad Nacional del Comahue, Quintral 1250, CP: 8400, San Carlos de Bariloche, Río Negro, Argentina*



(Received 1 September 2020; accepted 25 June 2021; published 26 October 2021)

Emergent dynamics of complex systems are observed throughout nature and society. The coordinated motion of bird flocks, the spread of opinions, fashions and fads, or the dynamics of an epidemic, are all examples of complex macroscale phenomena that arise from fine-scale interactions at the individual level. In many scenarios, observations of the system can only be made at the macroscale, while we are interested in creating and fitting models of the microscale dynamics. This creates a challenge for inference as a formal mathematical link between the microscale and macroscale is rarely available. Here, we develop an inferential framework that bypasses the need for a formal link between scales and instead uses sparse Gaussian process regression to learn the drift and diffusion terms of an empirical Fokker-Planck equation, which describes the time evolution of the probability density of a macroscale variable. This gives us access to the likelihood of the microscale properties of the physical system and a second Gaussian process is then used to emulate the log-likelihood surface, allowing us to define a fast, adaptive MCMC sampler, which iteratively refines the emulator when needed. We illustrate the performance of our method by applying it to a simple model of collective motion.

DOI: [10.1103/PhysRevResearch.3.043074](https://doi.org/10.1103/PhysRevResearch.3.043074)

I. INTRODUCTION

Complex systems are characterised by multiscale dynamics, with a high-dimensional microstate that describes the state of the individual components, and a reduced dimension macrostate that emerges from interactions at the lower level [1,2]. Connecting these two scales is the canonical challenge of complex systems science [3,4]. In certain cases a formal mathematical derivation of equations describing the macrostate may be obtained based on the properties of the lower-level components [5–7]; however this often requires simplifying assumptions that cannot be justified in most scenarios. In the absence of a formal mathematical link between scales, the inference of microscale dynamics from macroscale observations is challenging. While forward simulations of complex computer models are able to link microscale parameters with coarse grained observables, the inverse problem of statistical inference remains largely intractable [8]. This is due to the unavailability of the probability density, or likelihood, for an observation. Several simulation-based approaches have been proposed that approximate the intractable likelihood but these are often computationally expensive and lack a formal quantification of uncertainty [9–11].

Here, we approach the problem of multiscale inference by assuming the existence of an empirical Fokker-Planck equation (FPE) that describes the changes with time of the

probability density of a macroscale variable. We consider a scenario where we are able to efficiently run forward simulations of the model but do not have access to a likelihood function that provides the probability density of the empirical observations for a given parameter set, i.e., we cannot derive the drift and diffusion functions of the FPE from the microscale parameters. Instead, through the application of sparse Gaussian process (GP) regression [12,13] we learn the drift and diffusion functions of the FPE from simulations. This allows us to estimate the likelihood of microscale parameters given a set of empirical macroscale observations.

Our approach presents several advances over existing methods for inference in complex systems. Firstly, we incorporate several concepts from equation-free modeling [14] into the inference process. Equation-free modeling offers an efficient numerical method for investigating the macroscale dynamics of microscale models. By integrating aspects of this approach with sparse Gaussian process regression of the drift and diffusion functions we are able to formally quantify the uncertainty inherent in simulations of stochastic microscale models. This allows us to connect the equation-free framework with an MCMC sampler that directs simulation effort to refining regions of parameter space with high likelihood.

A conceptual overview of our framework is provided in Fig. 1. This figure illustrates the three main components of our framework. Firstly, we employ a microscale simulator to generate the macroscale dynamics of our physical model for a given microscale parameter set. Secondly, we use sparse GP regression to link the two scales and learn the macroscale drift and diffusion functions from simulation output. This allows us to estimate the likelihood of the microscale parameters given the observed data, and further provides a formal quantification of the uncertainty in the estimate. Finally, the

Published by the American Physical Society under the terms of the [Creative Commons Attribution 4.0 International](https://creativecommons.org/licenses/by/4.0/) license. Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI.

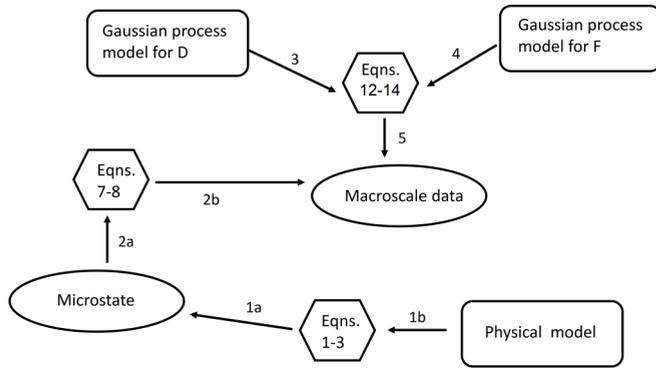


FIG. 1. Graphical summary of the proposed inference scheme. If the microstates, corresponding to the particles' positions $x_i(t)$ and velocities $u_i(t)$ in Eq. (1) were observable, we could infer the parameters of the physical model directly from the corresponding microstate data (arrows 1a and 1b). However, such high-resolution data is usually not available, and the challenge therefore is to infer the physical model parameters from macroscale features—the distribution of average velocities U in our case (arrows 2a and 2b). This distribution is in principle defined by the physical model and its parameters via Eq. (7), giving rise to the likelihood of the physical model given the macroscale data [Eq. (8)]. However, the mathematical expression of this physical model likelihood depends on two functions—the diffusion function D and the drift function F —which are not analytically tractable. We therefore approximate these functions by two Gaussian process models fitted to simulated macroscale output, based on Eqs. (12–14) (arrows 3–5). Note that the inference of these Gaussian processes is based on the probability of the simulation output given the Gaussian process, which is independent of the physical model. Inserting the Gaussian process approximations of F and D back into Eq. (7) then leads to an approximation of the physical model likelihood [Eq. (8)], which is used for inference of the physical model parameters. Note that in order to make the inference computationally efficient, the physical model likelihood is approximated (or emulated) by another Gaussian processes, which is not included in the present figure.

likelihood estimate and its associated uncertainty are passed to an adaptive MCMC algorithm that samples from the posterior distribution of the microscale parameters. The adaptive MCMC sampler employs a second, independent implementation of sparse GP regression to emulate the log-likelihood surface of the microscale parameters and uses the emulated surface when uncertainty is low but triggers further forward simulations when uncertainty is high.

The remainder of this paper is structured as follows. In Sec. II, our simulation model is introduced. Section III describes the theoretical foundation for our framework, which is explained in detail in Sec. IV. In Sec. V we give details about the parameter settings we adopt to produce the results presented in Sec. VI. Lastly, we discuss the effectiveness and potential applications of our method in Sec. VII.

II. THE MODEL

We demonstrate our framework on a simple model of collective animal movement adapted from [15,16]. The model is a one-dimensional, self-propelled particles model with the

following equations describing the evolution of the positions $x_i(t)$ and velocities $u_i(t)$ of an individual that moves along a line of dimensionless length with periodic boundary conditions,

$$\begin{aligned} x_i(t + \Delta t) &= x_i(t) + \Delta t v_0 u_i(t), \\ u_i(t + \Delta t) &= u_i(t) + \alpha(G(\bar{u}_i(t, \delta)) - u_i(t)) + \xi_i, \end{aligned} \quad (1)$$

where v_0 is a constant scaling of each particle's velocity, the quantity $\bar{u}_i(t, \delta)$ is the average velocity of all individuals, excluding individual i , within a metric interaction range of length δ , the parameter α represents the relative weight that an individual assigns to its own velocity and those of its neighbours when updating its velocity, ξ_i is a random noise term taken from a normal distribution $\mathcal{N}(0, \eta^2 \Delta t)$, and the function G represents a social interaction term, which causes an individual to adopt a similar velocity to its observed neighbours,

$$G(z) = \begin{cases} (z + 1)/2, & z > 0 \\ (z - 1)/2, & z < 0. \end{cases} \quad (2)$$

The model was developed for the study of locust moving in an annular arena [15,17] and is characterised by a double-well potential with intermittent switches occurring between metastable states representing clockwise and counterclockwise motion [see Fig. 2(a) for an example time series]. The dynamics of the model are governed by three parameters: the interaction radius, the strength of the social force, and the noise level, which we define as our macroscale parameter vector:

$$\theta = (\alpha, \delta, \eta). \quad (3)$$

Ideally, we would like to infer these parameters from detailed measurements or observations of the microstates $\{x_i(t), u_i(t)\}$. However, such high-resolution data are rarely available. In the present paper we therefore pursue an approach that focuses on the emergent macroscale properties of the system using equation-free modeling, to be explained in Sec. IV.

As in [17] our coarse macroscale variable is taken to be the global average velocity

$$U(t) = \frac{1}{N} \sum_{i=1}^N u_i(t). \quad (4)$$

Yates *et al.* [17] show that in the case of infinite interaction radius δ the evolution of U can be described by a stochastic differential equation (SDE) of the following form

$$dU = F(U, \theta)dt + \sqrt{D(U, \theta)}dW_t, \quad (5)$$

where dW_t is a Wiener process, $F(U, \theta)$ is the drift function, and $D(U, \theta)$ is the diffusion function, both of which are available in closed form. Following [17] we assume that for finite interaction radius δ , an FPE of the same form can be assumed to exist. However, in that case, $F(U, \theta)$ and $D(U, \theta)$ are no longer available analytically and have to be empirically inferred from the data.

Equation (5) gives rise to an associated FPE [18] that describes the evolution of $\rho(U, t)$ the probability density

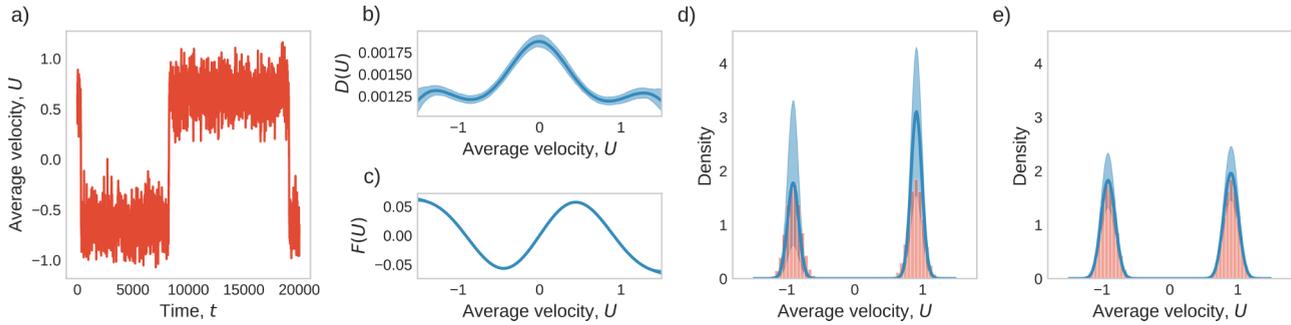


FIG. 2. (a) Sample time series from the simulation model for 20 individuals moving along a line of dimensionless length 36 with parameters $v_0 = 1$ and $\Delta t = 1$. The switches between the two metastable states $U = 1$ and $U = -1$ represent cohesive movement in clockwise or counterclockwise direction. (b) Inferred diffusion function $D(U)$ (dark blue line) and 95% posterior credible interval (light blue shaded region) learnt from simulation results using sparse GP regression, as a function of the average velocity U . (c) Idem for the drift function $F(U)$. (d) Stationary probability density mean and uncertainty calculated based on 50 000 simulation outputs. (e) Stationary probability density from 500 000 simulation outputs. Note that by increasing the number of simulations the uncertainty has greatly reduced.

function of $U(t)$:

$$\frac{\partial \rho}{\partial t} = \frac{1}{2} \frac{\partial^2 (D(U, \theta) \rho)}{\partial U^2} - \frac{\partial (F(U, \theta) \rho)}{\partial U}. \quad (6)$$

For known drift and diffusion functions, the stationary probability density (SPD) $\rho_s(U|\theta)$ can be calculated as [19]

$$\rho_s(U|\theta) = \frac{1}{Z} \exp \left(2 \int_0^U \frac{F(s, \theta)}{D(s, \theta)} ds - \ln (D(U, \theta)) \right) \quad (7)$$

where Z is a normalising constant.

Hence if the drift and diffusion functions can be derived from the microscale parameters, Eq. (7) gives us access to

$$\mathcal{L}(\theta; U_{\text{data}}) := \rho_s(U_{\text{data}}|\theta) \quad (8)$$

the likelihood of model parameters θ given empirical observations of the group average velocity $U_{\text{data}} = \{U_1, U_2, \dots, U_n\}$. With this likelihood, one can pursue parameter inference in a classical sense via maximum likelihood estimation or within a Bayesian framework by sampling from the θ posterior distribution, after defining an appropriate prior distribution.

However in most scenarios, the drift and diffusion functions are intractable and cannot be derived from microscale parameters. A variety of likelihood-free methods have been developed recently (see e.g., [9,20,21]), but they are intrinsically approximate and depend on various heuristics and intuition. To overcome this difficulty, we propose to employ sparse Gaussian process regression to learn these functions from fine-scale simulations of the model.

III. BACKGROUND

In this section we summarize the inferential machinery, which forms the foundation of our framework. Specifically, we review the concepts of Gaussian process regression [22] and sparse Gaussian process regression [23–25]. We use the letter x to denote the input or explanatory variable of the function of interest, and the letter y to denote the response or output variable. Depending on the application, x may represent the macroscale velocity U , with y representing the corresponding outputs of $F(U, \theta)$ and $D(U, \theta)$, or x may represent the microscale parameter vector θ , with y representing

the corresponding log-likelihood. Please note that x is not to be confused with the spatial coordinate in Eq. (1).

Gaussian processes are stochastic processes for which any finite collection of random variables follows a multivariate normal (MVN) distribution. Gaussian processes are therefore determined by a mean function m and a covariance matrix K , often defined as a covariance kernel $K_{ij} = k(x_i, x_j)$ that is a function of the location of the random variables. In Gaussian process regression we seek to learn a latent function f based on a set of N observations $\mathbf{y} = \{y_i\}_{i=1}^N$ at locations $\mathbf{x} = \{x_i\}_{i=1}^N$, where

$$y_i = f(x_i) + v, \quad (9)$$

and v is an additive Gaussian white noise term. The latent function f is a realisation of a Gaussian process and is modelled with a GP prior,

$$p(\mathbf{f}|\mathbf{x}) \sim \mathcal{N}(\mathbf{m}, \mathbf{K}).$$

GP regression may also be extended to consider the case of heteroscedastic noise where the variance of the observation noise v is itself a function of x [26]. GPs inherit all properties from MVNs; hence, performing GP regression when the likelihood is also Gaussian involves calculating the conditional distribution of a joint MVN. Given a set of training observations \mathbf{y} at locations \mathbf{x} , and assuming a zero-mean process, it follows that the posterior distribution of \mathbf{f}^* at a set of N^* test locations \mathbf{x}^* is given by [27]

$$p(\mathbf{f}^*|\mathbf{x}^*, \mathbf{x}, \mathbf{y}) \sim \mathcal{N}(\boldsymbol{\mu}^*, \boldsymbol{\Sigma}^*),$$

where

$$\begin{aligned} \boldsymbol{\mu}^* &= \mathbf{K}(\mathbf{x}^*, \mathbf{x})(\mathbf{K}(\mathbf{x}, \mathbf{x}) + \sigma^2 \mathbf{I})^{-1} \mathbf{y} \\ \boldsymbol{\Sigma}^* &= \mathbf{K}(\mathbf{x}^*, \mathbf{x}^*) \\ &\quad - \mathbf{K}(\mathbf{x}^*, \mathbf{x})(\mathbf{K}(\mathbf{x}, \mathbf{x}) + \sigma^2 \mathbf{I})^{-1} \mathbf{K}(\mathbf{x}, \mathbf{x}^*) \end{aligned}$$

Here \mathbf{I} is the identity matrix and $\mathbf{K}(\mathbf{x}^*, \mathbf{x})$ is the covariance matrix defined at all pairs of train and test points (i.e., an $N^* \times N$ matrix) with similar definitions for the other covariance terms. As these two equations suggest, inference of the latent function defined on the test set \mathbf{f}^* only depends on the kernel function and the data. The kernel function defines the correlation structure of the GP. In our paper we employ

one of the most commonly used kernels, the exponentiated quadratic kernel, also known as the radial basis function (RBF) kernel:

$$k(x_i, x_j) = \tau^2 \exp\left(-\frac{\|x_i - x_j\|^2}{\vartheta}\right).$$

The τ^2 and ϑ kernel hyperparameters represent the scale, or amplitude, of the process and its lengthscale and $\|\cdot, \cdot\|$ is the Euclidean norm. For a review of alternative kernels, see [22].

One of the major drawbacks of GP regression is the $\mathcal{O}(N^3)$ computational cost associated with inverting the covariance matrix. The idea behind sparse GP regression [23] is to define so called inducing locations \mathbf{z} , with corresponding latent function values \mathbf{f}_z , which can summarize the training set. The key assumption is that the latent function \mathbf{f}^* at any test inputs and the latent function at the training locations \mathbf{f} are conditionally independent given \mathbf{f}_z [25], i.e.,

$$p(\mathbf{f}^*|\mathbf{f}, \mathbf{f}_z) = p(\mathbf{f}^*|\mathbf{f}_z).$$

Given this assumption the posterior distribution of the latent function at any test locations given \mathbf{y} is

$$p(\mathbf{f}^*, \mathbf{f}_z|\mathbf{y}) = p(\mathbf{f}^*|\mathbf{f}_z)p(\mathbf{f}_z|\mathbf{y}).$$

Variational inference proceeds by introducing a variational approximation to this posterior,

$$q(\mathbf{f}^*, \mathbf{f}_z) = p(\mathbf{f}^*|\mathbf{f}_z)\phi(\mathbf{f}_z),$$

where $\phi(\mathbf{f}_z)$ is a Gaussian distribution with mean $\boldsymbol{\mu}_q$ and covariance $\boldsymbol{\Sigma}_q$. To determine the optimal variational parameters $\boldsymbol{\mu}_q$ and $\boldsymbol{\Sigma}_q$, we can maximise a lower bound on the marginal log-likelihood, which is equivalent to minimising the Kullback-Leibler (KL) divergence between the true posterior and the variational distribution. Following [13], the evidence lower bound can be obtained via Jensen's inequality:

$$\begin{aligned} \log p(\mathbf{y}) &= \log \iint p(\mathbf{y}|\mathbf{f}, \mathbf{f}_z)p(\mathbf{f}, \mathbf{f}_z)d\mathbf{f}d\mathbf{f}_z \\ &= \log \iint p(\mathbf{y}|\mathbf{f})p(\mathbf{f}, \mathbf{f}_z)\frac{q(\mathbf{f}, \mathbf{f}_z)}{q(\mathbf{f}, \mathbf{f}_z)}d\mathbf{f}d\mathbf{f}_z \\ &\geq \iint \log\left(\frac{p(\mathbf{y}|\mathbf{f})p(\mathbf{f}, \mathbf{f}_z)}{q(\mathbf{f}, \mathbf{f}_z)}\right)q(\mathbf{f}, \mathbf{f}_z)d\mathbf{f}d\mathbf{f}_z \\ &\geq \int \log p(\mathbf{y}|\mathbf{f})q(\mathbf{f})d\mathbf{f} - \mathbf{KL}(\phi(\mathbf{f}_z)\|p(\mathbf{f}_z)), \end{aligned} \quad (10)$$

where $q(\mathbf{f}) = \int q(\mathbf{f}, \mathbf{f}_z)d\mathbf{f}_z$ and the **KL** term denotes the KL divergence between the prior distribution over \mathbf{f}_z and the variational posterior.

Since $\phi(\mathbf{f}_z)$ is a multivariate Gaussian, $q(\mathbf{f})$ and the KL term in Eq. (10) are available in closed form. As the likelihood factorises across the data, i.e.,

$$p(\mathbf{y}|\mathbf{f}) = \prod_{i=1}^N p(y_i|f_i),$$

the integral $\int \log p(\mathbf{y}|\mathbf{f})q(\mathbf{f})d\mathbf{f}$ can be decomposed into N one dimensional integrals that are tractable when the likelihood is Gaussian [28].

In [13] the sparse variational method is extended to incorporate likelihoods that depend on multiple latent functions,

this is termed a chained or multitlatent GP. Considering the case when the likelihood depends on two latent functions, \mathbf{f} and \mathbf{g} , the lower bound on the marginal log-likelihood is now [13]

$$\begin{aligned} \log p(\mathbf{y}) &\geq \iint \log p(\mathbf{y}|\mathbf{f}, \mathbf{g})q(\mathbf{f})q(\mathbf{g})d\mathbf{f}d\mathbf{g} \\ &\quad - \mathbf{KL}(\phi(\mathbf{f}_z)\|p(\mathbf{f}_z)) - \mathbf{KL}(\phi(\mathbf{g}_z)\|p(\mathbf{g}_z)), \end{aligned} \quad (11)$$

where $q(\mathbf{g}) = \int p(\mathbf{g}|\mathbf{g}_z)\phi(\mathbf{g}_z)d\mathbf{g}_z$ and we have introduced two Gaussian distributions that are variational approximations to the posterior at the inducing point locations, $\phi(\mathbf{f}_z) \sim \mathcal{N}(\boldsymbol{\mu}_q^f, \boldsymbol{\Sigma}_q^f)$ and $\phi(\mathbf{g}_z) \sim \mathcal{N}(\boldsymbol{\mu}_q^g, \boldsymbol{\Sigma}_q^g)$.

In this paper we consider the case where the second latent GP determines an input dependent heteroscedastic noise term, such that

$$\begin{aligned} y_i &\sim \mathcal{N}(f(x_i), e^{g(x_i)}) \\ p(\mathbf{f}|\mathbf{x}) &\sim \mathcal{N}(\mathbf{m}_f, \mathbf{K}_f) \\ p(\mathbf{g}|\mathbf{x}) &\sim \mathcal{N}(\mathbf{m}_g, \mathbf{K}_g). \end{aligned}$$

Hence, the latent function f determines the mean of y_i at location x_i , while the latent function g is exponentiated so that it is constrained positive and is then the location dependent variance in y_i . In this case the integral in Eq. (11) is analytic [12] and a closed-form lower bound can be obtained. This lower bound can then be maximised using stochastic optimisation [28] in order to find the optimal variational distributions $\phi(\mathbf{f}_z)$ and $\phi(\mathbf{g}_z)$.

IV. METHODS

A. Microscale simulations

Simulations of the microscale model are implemented in Python using the machine learning library TensorFlow [29] and run in parallel on a GPU. Using this approach, we are able to run multiple independent instances of the model for each set of values of the parameter vector $\boldsymbol{\theta}$.

In equation-free modeling [14] short bursts of simulations are run and used to learn about the macroscale dynamics. The approach involves moving from the microscale to the macroscale, termed restriction, and moving from the macroscale to the microscale, termed lifting. The first process, restriction, is straightforward and for our model involves applying Eq. (4) to calculate the macrostate variable from the microstate. The lifting process is more involved and requires mapping a macrostate to a specific microstate. Initialising a microstate at random with a defined macrostate will introduce a lifting error as not all configurations with the same macrostate variable are equally likely.

In our application we seek to obtain values of $U_{t+\Delta t} - U_t$ from simulations that are distributed evenly across the domain of U . If we allow the microstate to evolve and record U_t throughout the simulations we will inevitably end up with simulation outputs focused in regions of high probability density, and few measurements from areas of low probability density, which in our model corresponds to $U_t \simeq 0$.

To overcome this issue, we simulate the dynamics for an initial period of time and then successively perturb the microstates of each parallel simulation to a desired set of macrostate locations. We then run several time steps of the

simulations from the perturbed microstates and record the output to use in estimating the drift and diffusion functions. For example, if we are running 1000 simulations in parallel, we run the simulations for an initial number of timesteps. Next, we define a desired set of 1000 macrostate variables U_i that are uniformly distributed across the domain. We then map each desired macrostate to the microscale simulation with the closest macrostate variable. Each simulation is subsequently perturbed by altering the velocity of each individual, so that the macrostate of the simulation matches the desired macrostate, and run forward for a number of time steps. By repeating these steps, we are able to accelerate the coverage of the whole configuration space without having to wait for the system to evolve to particular locations.

B. Inferring the drift and diffusion functions

As closed-form expressions for the drift and diffusion functions are unavailable in our application, we employ sparse Gaussian process regression to learn these functions from fine-scale simulations of the model. For notational simplicity, we will omit the explicit dependence on θ and from here on use $F(U)$, $D(U)$ and $\rho_s(U)$ to indicate, respectively, the drift and diffusion functions and the SPD.

Following [30], we define an analogous, discretized version of Eq. (5) as a stochastic difference equation

$$U_{t+\Delta t} - U_t = F(U_t)\Delta t + \epsilon\sqrt{D(U_t)\Delta t}, \quad (12)$$

where Δt is a discrete time step and $\epsilon \sim \mathcal{N}(0, 1)$. By comparing Eqs. (9) and (12) we can observe that learning the drift and diffusion functions is an example of heteroscedastic Gaussian process regression where

$$y = \frac{U_{t+\Delta t} - U_t}{\Delta t}, \quad (13)$$

$F(U_t)$ is the latent function, and the variance of the heteroscedastic noise term is

$$\sigma^2 = \frac{D(U_t)}{\Delta t}. \quad (14)$$

It is possible to learn the drift and diffusion functions sequentially from measurements of $U_{t+\Delta t} - U_t$ as in [30,31] by employing the following equation to infer $D(U)$,

$$D(U) = \lim_{\Delta t \rightarrow 0} \frac{1}{\Delta t} E[(U_{t+\Delta t} - U_t)^2 | U_t = U]. \quad (15)$$

However, this approach leads to a systematic bias for finite Δt [32]. To overcome this issue, we employ sparse GP regression within a variational framework [13]. This allows the drift and the diffusion functions to be learnt simultaneously from simulations of $U_{t+\Delta t} - U_t$ and enables us to deal with large numbers of simulation outputs ($N \sim 10^5$). Note that inferring the drift and diffusion functions involves maximising a lower bound on a second, different likelihood that depends on simulation output and not empirical observations. This second likelihood is a standard GP likelihood as defined in Eq. (10) and is separate from the physical model likelihood defined in Eq. (8). Hence, the variational approximation to the posterior for the drift and diffusion functions, along with the kernel parameters and inducing point locations, are optimised by maximising a variational lower bound on the marginal

GP log-likelihood extended to multiple latent functions [13] given in Eq. (11).

Our implementation uses the GPflow library [33], a package for building Gaussian process models using TensorFlow [29,34]. We specify Gaussian process priors on the latent drift and diffusion functions $F(U)$ and $D(U)$ with separate independent RBF kernels.

Once optimised, we obtain the diffusion function posterior distribution, shown in Fig. 2(b), and the drift function posterior, shown in Fig. 2(c). These distributions capture the inherent uncertainty in the functions due to the finite number of microscale simulations that can be performed. By drawing multiple samples from the posterior drift and diffusion functions we are able to propagate this uncertainty into the stationary probability density function of the macroscale observations U_{data} by calculating Eq. (7) for each sample using numerical quadrature. Thus, given a sequence of empirical macroscale observations, we are able to calculate an estimate of the data likelihood $\mathcal{L}(\theta; U_{\text{data}})$ as well as formally quantifying the uncertainty in the estimate.

Figures 2(d) and 2(e) show $\rho_s(U)$, the steady-state probability density of the global average velocity U calculated for different numbers of simulation outputs. As we would expect, increasing the number of simulation outputs decreases the uncertainty in $\rho_s(U)$ and will subsequently lead to reduced uncertainty in the likelihood $\mathcal{L}(\theta; U_{\text{data}})$.

The uncertainty in the likelihood is propagated from the uncertainty in the drift and diffusion functions; as our model is stochastic, this variability is intrinsic. Indeed, in the context of our framework, where simulations are stochastic and closed-form expressions for $F(U)$ and $D(U)$ are unavailable, this variability will be present for any finite number of simulation outputs.

C. The sampling algorithm

The stationary probability density defined by Eq. (7) is of central importance in our framework as it provides the tools to access the likelihood of microscale parameters θ given observations of our macroscale variable U . We perform inference of θ in a Bayesian framework and aim to sample from the posterior distribution of the microscale parameters through an adaptive MCMC sampler based on the method proposed in [35].

In a standard Metropolis-Hastings (MH) framework [36], a candidate sample θ^+ is generated from an initial point in parameter space θ^- through a specified transition kernel. The sampler evaluates the likelihood L and priors π of the two points and then either accepts or rejects the candidate location via an acceptance function. For a symmetric transition kernel, a commonly used acceptance function is

$$a = \min \left\{ 1, \frac{L(\theta^+)\pi(\theta^-)}{L(\theta^-)\pi(\theta^+)} \right\} \quad (16)$$

and the candidate location is accepted with probability a . A naive implementation of the MH sampler would involve a computationally expensive forward simulation at each step and would base acceptance on a point estimate of the data likelihood. Instead, we accelerate the sampling using a sparse GP emulator [37] that approximates the log-likelihood

surface. Note, we therefore introduce sparse Gaussian process regression at two points in our framework, firstly to learn a drift and diffusion function from microscale simulations, then secondly to emulate the log-likelihood surface when running the sampler.

For our sampling procedure, we need to choose a design for the locations of a set of points to initialize the emulator. The aim is to pick input parameters to cover the whole parameter domain efficiently. Naive designs include specifying a regular grid of parameter values, which suffers from the curse of dimensionality, or drawing samples from a uniform distribution in the parameter domain, which is inefficient due to random clustering of points and large gaps. As our initial set of points needs to span a high-dimensional microscale parameter space, we employ a space-filling design, which is a computationally efficient way to place points in a high-dimensional space such that there is a measure of uniformity in how they fill the space, i.e., they do not leave large gaps. In our work, we use a Sobol sequence [38] to create an initial log-likelihood map over a pre-specified region. For each set of initial values, we run forward simulations that yield a distribution over the drift and diffusion functions. We then generate k samples of the log-likelihood given empirical observations U_{data} by sampling the drift and diffusion functions and inserting them into Eq. (7), effectively yielding k log-likelihood evaluations for the same parameter values. The initial points form a training set for the emulator used by the sampler.

Starting from the highest log-likelihood initial location, the sampler then uses the emulated log-likelihood surface as a surrogate for the full forward simulations [39]. We employ sparse GP regression again using the GPflow package [33]. We maximise the lower bound on the marginal GP log-likelihood from Eq. (10) to optimize the variational approximation and the GP hyperparameters. As we have intrinsic stochasticity in the simulation outputs and k samples of the log-likelihood at each simulation location, this is mathematically equivalent to fitting a Gaussian process with observation noise, or nonzero nugget term [22]. By letting the GP learn the nugget parameter and covariance kernel lengthscale, the sampler is able to accurately estimate the log-likelihood surface at a candidate location along with its associated uncertainty.

At each step of the sampler, a candidate location θ^+ is generated. The sampler assesses the uncertainty in the log-likelihood surface at the proposal location θ^+ and the current location θ^- . If the uncertainty is sufficiently low then the candidate is accepted according to Eq. (16), otherwise further forward simulations are triggered to refine the emulator [35].

The refinement criterion is based on the uncertainty associated with the log-likelihood surface. Given proposed and current locations with log-likelihoods l^+ and l^- and associated uncertainties σ^+ , σ^- , we define an uncertainty indicator P^\pm as

$$P^\pm = \begin{cases} P(\zeta^+ \leq \zeta^-) & \text{if } l^+ \leq l^- \\ P(\zeta^+ > \zeta^-) & \text{otherwise,} \end{cases}$$

where ζ^+ and ζ^- are random variables drawn from $\mathcal{N}(l^+, \sigma^{+2})$ and $\mathcal{N}(l^-, \sigma^{-2})$ respectively.

The uncertainty in the log-likelihood is therefore quantified by comparing the expected relationship between two random samples from the emulator surface with the mean values. P^\pm

ranges from 0.5, meaning there is no discernible difference between the two locations, to 1, indicating complete certainty in the relationship between l^+ and l^- .

The refinement probability is then defined as

$$\gamma = 2\bar{\gamma}(s)(1 - P^\pm),$$

where s is the number of steps since the last refinement and $\bar{\gamma}(s)$ is a logistic function with a specified slope and midpoint that acts as a memory and prevents refinement at every step. As there is intrinsic uncertainty in the likelihood (unlike in [35]) forward simulations could in principle be triggered at every step; hence, our refinement probability takes on smaller values whenever a microscale simulation has just been triggered and increases with the number of steps taken since refinement. If refinement is triggered, forward simulations are run at the location with largest uncertainty. Along with triggered refinement, we also include random refinement at each step for ensuring asymptotic convergence and note that our uncertainty indicator is invariant to relabelling of θ^+ and θ^- meaning the refinement process does not impact the reversibility of the transition kernel [35]. When refinement occurs, the log-likelihood samples are added to the emulator GP training set and further optimisation is performed.

Our final comment concerns the theoretical guarantee of convergence to the true posterior distribution. A related proof is provided in [35], but for deterministic systems. Our system is stochastic, and the likelihood of Eqs. (7)–(8), estimated by fitting GPs for $F(U)$ and $D(U)$ to finite numbers of forward simulations from Eq. (1), is itself subject to uncertainty. However, convergence to the true posterior distribution is guaranteed by combining the proof in [35] with the following three well-established facts (all subject to adequate regulatory conditions): (i) that a neural network with a sufficiently large number of hidden nodes is a universal approximator and, thus, unbiased [40,41]; (ii) that a Gaussian process is the limiting case of a neural network with an infinite number of hidden nodes [42]; and that (iii) replacing the true likelihood by an unbiased estimate does not affect the limiting distribution of an MCMC sampling scheme [43,44].

V. EMPIRICAL STUDY

The drift and diffusion functions inference is performed using sparse GP regression and we specify $M = 20$ inducing points, which we found to provide an appropriate trade-off between accuracy and computational costs.

For the inference of model parameters θ , we first create synthetic data by running multiple parallel simulations of the model with the given parameter set. After relaxation of transients, we randomly select 200 simulated macroscale outputs as our empirical data. Denoting the synthetic data set as $U_{\text{data}} = \{U_1, U_2, \dots, U_{200}\}$, a sample of the log-likelihood of the parameter set given the data is given by

$$l^k = \sum_{n=1}^{200} \log \rho_s^k(U_n)$$

where $\rho_s^k(U_n)$ is the stationary probability density from Eq. (7), calculated from the k -th sample from the posterior of the drift and diffusion functions.

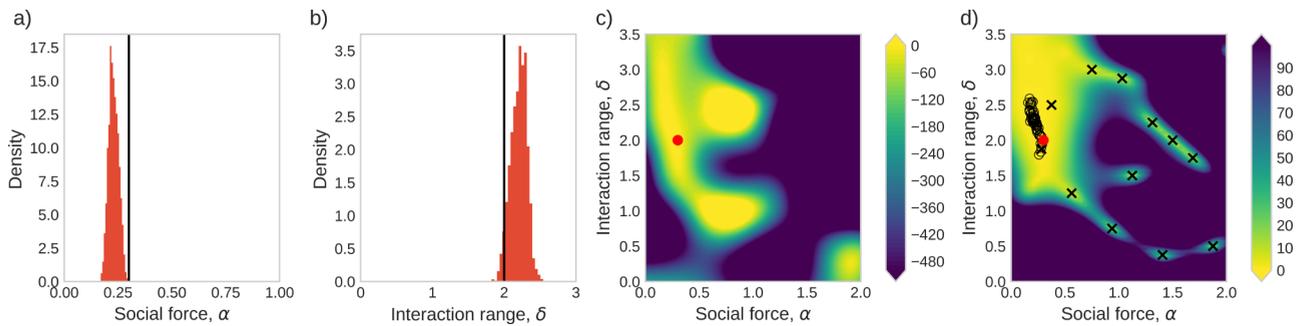


FIG. 3. Inference results for $N = 20$, $\alpha = 0.3$, and $\delta = 2$. Results shown for 10 000 samples after burn-in of 40 000 (Geweke’s diagnostic [45] was used to test convergence: highest absolute z score was 0.73 for α and 0.89 for δ). (a) Posterior distribution of the weighting given to social cues (α). Vertical bar represents the true parameter value. (b) Posterior distribution for the interaction range (δ). Vertical bar represents the true parameter value. (c) Refined surrogate log-likelihood; the true parameter value is indicated by the red, full circle. (d) Variance associated with the refined surrogate log-likelihood. Black crosses show the first initial points from our space-filling design; black open circles show every tenth refinement. The more explored regions have lower uncertainty.

Next, we employ sparse GP regression to build our surrogate log-likelihood surface. We arrange the M inducing points in a fixed uniform grid and set $M = 16$ per parameter dimension; this reduction in the number of inducing points reflects the additional computational burden arising from performing GP regression in more dimensions.

Lastly, for our MCMC scheme we choose uniform priors $\pi \sim U(0, 10)$ on all parameters to infer and we set the random refinement criterion to 10^{-4} .

VI. RESULTS

We demonstrate the performance of our method for both two-dimensional and three-dimensional inference using three parameter sets and show that we can accurately infer the interaction radius δ of the model, along with the interaction strength α and, in three dimensions, the level of noise η .

The first parameter set is defined on a group of $N = 20$ individuals with parameter values $\alpha = 0.3$, $\delta = 2$, and a fixed (assumed known) value of $\eta = 0.25$. The second parameter set is specified on a group of $N = 100$ individuals with parameters $\alpha = 0.8$, $\delta = 1$, and $\eta = 1$ (again η is held fixed). For the third parameter set we infer all three parameters for

a group size of $N = 30$. The true parameters are $\alpha = 0.6$, $\delta = 1.5$, and $\eta = 0.5$.

We then use our framework to infer the posterior distributions for α and δ , which are shown in Figs. 3(a) and 3(b) and Figs. 4(a) and 4(b) as well as for all three parameters α , δ , and η , shown in Figs. 5(a), 5(b), and 5(d). The refined, surrogate log-likelihood surface for the parameter sets is shown in Figs. 3(c) and 4(c) and the associated uncertainty in Figs. 3(d) and 4(d). By incorporating the uncertainty in the log-likelihood into the emulator the sampler is able to run forward simulations only where needed and focus on refinement in regions of high likelihood.

VII. DISCUSSION

We have presented a new statistical framework for inference of microscale parameters from macroscale measurements of interacting systems. By employing sparse Gaussian process regression we have bypassed the need for a formal link between scales and obtained approximations of the probability density of macroscale observations, simultaneously calculating the associated uncertainty caused by the use of a finite number of microscale simulations. This allows us

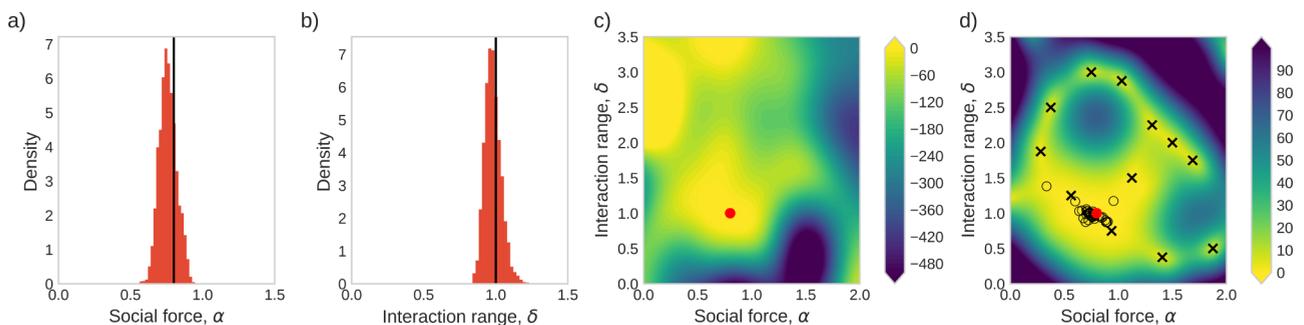


FIG. 4. Inference results for $N = 100$, $\alpha = 0.8$, and $\delta = 1$. Results shown for 10 000 samples after burn-in of 20 000 (Geweke’s diagnostic [45] was used to test convergence: highest absolute z score was 0.43 for α and 0.34 for δ). (a) Posterior distribution of the weighting given to social cues (α). Vertical bar represents the true parameter value. (b) Posterior distribution for the interaction range (δ). Vertical bar represents the true parameter value. (c) Refined surrogate log-likelihood; the true parameter value is indicated by the red, full circle. (d) Variance associated with the refined surrogate log-likelihood. Black crosses show the first initial points from our space-filling design; black open circles show every tenth refinement. For larger population size there is lower uncertainty in the simulation output so less refinement is required.

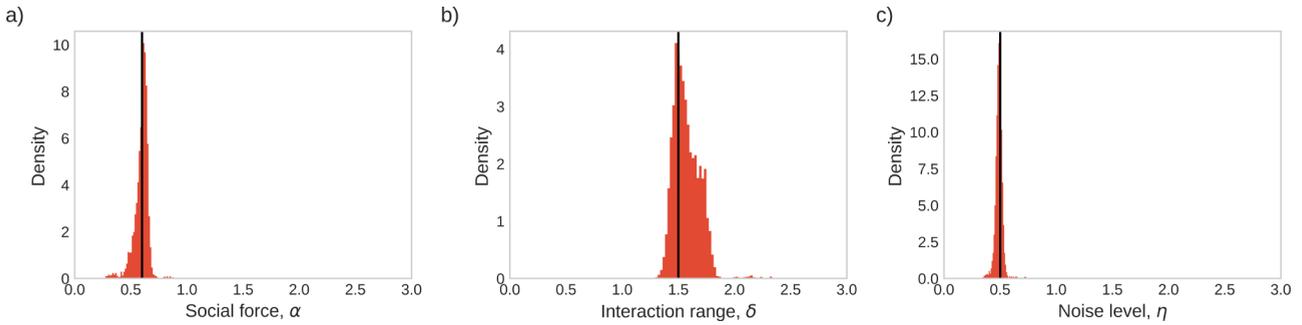


FIG. 5. Inference results for $N = 30$, $\alpha = 0.6$, $\delta = 1.5$, and $\eta = 0.5$. Results shown for 20 000 samples after burn-in of 30 000 (Geweke’s diagnostic [45] was used to test convergence: highest absolute z score was 0.67 for α , 0.59 for δ and 0.39 for η). (a) Posterior distribution of the weighting given to social cues (α). (b) Posterior distribution for the interaction range (δ). (c) Posterior distribution for the noise term (η). The vertical bars show the true parameter values.

to construct a fast, adaptive MCMC sampler that employs a second Gaussian process to emulate the log-likelihood surface.

While Gaussian process regression has been shown to be an effective method for learning drift and diffusion functions of stochastic differential equations [30], our approach presents an application of multitask Gaussian processes [13] in this context. Previous work in this area [30,31] has employed a separate Gaussian process to learn the diffusion function directly from simulation outputs as specified in Eq. (15), or by using a parametric function for the diffusion [30]. As stated earlier, the first approach leads to a bias in the posterior distribution of the diffusion function due to finite sampling rates, while the second approach neglects uncertainty. As the diffusion function appears in the denominator of the stationary probability density defined in Eq. (7), both these effects will significantly impair the estimate of the likelihood of microscale parameters given macroscale observations. We have overcome these issues by employing a multitask variational approach [13] that learns the drift and diffusion functions simultaneously and is able to quantify uncertainty in the diffusion as well as providing an unbiased estimator of both functions.

For effective sampling from the posterior, we employ an adaptive Metropolis-Hastings algorithm that is based on [35] with several modifications. Notably, we replace the local Gaussian process approximation with a sparse Gaussian process that allows us to use multiple samples from the SPD posterior for each parameter set where microscale simulations are run. By passing these multiple samples into the algorithm, the emulator GP is able to learn an effective observation noise in the simulator that arises due to the stochastic nature of the microscale model.

While we have applied our framework to a simple one-dimensional simulation model, our approach can be applied to any multiscale system that can be modelled at the microscale

but can only be easily observed at the macroscale. Collective animal movement presents one example of such a system, where individual trajectories are often difficult to observe while microscale models are straightforward to simulate [46]. However, many models of complex systems, such as models of voter behavior [47], opinion dynamics [48], or epidemics [49], share these characteristics and their microscale dynamics could be inferred from static observations using our proposed method. As for the physical sciences, our method can find applications in the context of ferromagnetism, e.g., inverse Ising problem, where the aim is to infer the coupling strength between spins given observed spin correlations, magnetizations or other data [8], as well as fluid dynamics, e.g., the inverse problem of identifying unknown flow conditions from an observed response of the free surface [50] or the non-invasive estimation of physiological parameters determining the systemic and pulmonary blood flow [51,52]. Modelling molecular, cellular, and autocatalytic pattern formation [53] is another application area of the method proposed in the present paper.

Future work will extend the framework to higher-dimensional macroscale systems, for which the presented ideas of using GPs to approximate the unavailable stationary probability density of interacting systems should in principle hold.

Source code is available on GitHub [54].

ACKNOWLEDGMENTS

This work was supported by the Leverhulme Foundation (Grant No. RPG-2018-398). J.M. was supported by a Leverhulme Visiting Professorship (VP2-2018-0630). C.J.T. is supported by a James S. McDonnell Foundation Studying Complex Systems Scholar Award. D.H. is supported by the Engineering and Physical Sciences Research Council (EP-SRC), Grant No. EP/T017899/1.

[1] P. W. Anderson, More is different, *Science* **177**, 393 (1972).
 [2] J. Sethna, *Statistical Mechanics: Entropy, Order Parameters, and Complexity* (Oxford University Press, Oxford, 2006), Vol. 14.

[3] C. R. Shalizi, Methods and techniques of complex systems science: An overview, *Complex Systems Science in Biomedicine* (Springer, Berlin, 2006), pp. 33–114.

- [4] M. Prokopenko, F. Boschetti, and A. J. Ryan, An information-theoretic primer on complexity, self-organization, and emergence, *Complexity* **15**, 11 (2009).
- [5] G. Demirel, F. Vazquez, G. Böhme, and T. Gross, Moment-closure approximations for discrete adaptive networks, *Physica D* **267**, 68 (2014).
- [6] N. Bellomo, A. Bellouquid, Y. Tao, and M. Winkler, Toward a mathematical theory of Keller-Segel models of pattern formation in biological tissues, *Math. Models Methods Appl. Sci.* **25**, 1663 (2015).
- [7] J. Toner and Y. Tu, Flocks, herds, and schools: A quantitative theory of flocking, *Phys. Rev. E* **58**, 4828 (1998).
- [8] H. C. Nguyen, R. Zecchina, and J. Berg, Inverse statistical problems: from the inverse Ising problem to data science, *Adv. Phys.* **66**, 197 (2017).
- [9] K. Cranmer, J. Brehmer, and G. Louppe, The frontier of simulation-based inference, *Proc. Natl. Acad. Sci. USA* **117**, 30055 (2020).
- [10] S. N. Wood, Statistical inference for noisy nonlinear ecological dynamic systems, *Nature (London)* **466**, 1102 (2010).
- [11] R. Wilkinson, Accelerating ABC methods using Gaussian processes, *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics* (PMLR, 2014), pp. 1015–1023.
- [12] M. Lázaro-Gredilla and M. K. Titsias, Variational heteroscedastic Gaussian process regression, *International Conference on Machine Learning (ICML)* (ICML, Bellevue, Washington, 2011).
- [13] A. D. Saul, J. Hensman, A. Vehtari, and N. D. Lawrence, Chained Gaussian processes, *Proceedings of the Nineteenth International Workshop on Artificial Intelligence and Statistics*, edited by A. Gretton and C. Robert (PMLR, Cadiz, Spain, 2016), Vol. 51, pp. 1431–1440.
- [14] C. W. Gear, J. M. Hyman, P. G. Kevrekidis, I. G. Kevrekidis, O. Runborg, C. Theodoropoulos, Equation-free, coarse-grained multiscale computation: Enabling microscopic simulators to perform system-level analysis, *Commun. Math. Sci.* **1**, 715 (2003).
- [15] J. Buhl, D. J. T. Sumpter, I. D. Couzin, J. J. Hale, E. Despland, E. R. Miller, and S. J. Simpson, From disorder to order in marching locusts, *Science* **312**, 1402 (2006).
- [16] A. Czirók, A.-L. Barabási, and T. Vicsek, Collective Motion of Self-Propelled Particles: Kinetic Phase Transition in One Dimension, *Phys. Rev. Lett.* **82**, 209 (1999).
- [17] C. A. Yates, R. Erban, C. Escudero, I. D. Couzin, J. Buhl, I. G. Kevrekidis, P. K. Maini, and D. J. T. Sumpter, Inherent noise can facilitate coherence in collective swarm motion, *Proc. Natl. Acad. Sci. USA* **106**, 5464 (2009).
- [18] C. Gardiner, *Stochastic Methods* (Springer, Berlin, 2009), Vol. 4.
- [19] H. Risken, *The Fokker-Planck Equation: Methods of Solution and Applications*, Springer Series in Synergetics (Springer, Berlin, 2012).
- [20] A. A. King, D. Nguyen, and E. L. Ionides, Statistical inference for partially observed Markov processes via the R package pomp, *J. Stat. Software* **69**, 1 (2016).
- [21] J. Owen, D. J. Wilkinson, and C. S. Gillespie, Likelihood free inference for Markov processes: a comparison, *Stat. Appl. Genetics Molecular Biology* **14**, 189 (2015).
- [22] C. E. Rasmussen and C. K. Williams, *Gaussian Processes for Machine Learning* (MIT Press, Cambridge, 2006).
- [23] E. Snelson and Z. Ghahramani, Sparse Gaussian processes using pseudo-inputs, in *Advances in Neural Information Processing Systems*, edited by Y. Weiss, B. Schölkopf, and J. Platt (MIT Press, 2006), Vol. 18.
- [24] M. Opper and C. Archambeau, The variational Gaussian approximation revisited, *Neural Comput.* **21**, 786 (2009).
- [25] M. Titsias, Variational Learning of Inducing Variables in Sparse Gaussian Processes, in *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics*, edited by D. Dyk and M. Welling (PMLR, 2009), Vol. 5, pp. 567–574.
- [26] P. W. Goldberg, C. K. Williams, and C. M. Bishop, Regression with input-dependent noise: A Gaussian process treatment, *Advances in Neural Information Processing Systems* (1998), pp. 493–499.
- [27] K. P. Murphy, *Machine learning: A Probabilistic Perspective* (MIT Press, Cambridge, 2012).
- [28] J. Hensman, N. Fusi, and N. D. Lawrence, Gaussian processes for big data, [arXiv:1309.6835](https://arxiv.org/abs/1309.6835).
- [29] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, Tensorflow: A system for large-scale machine learning, *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)* (USENIX, Savannah, GA, 2016), pp. 265–283.
- [30] P. Batz, A. Ruttner, and M. Opper, Approximate Bayes learning of stochastic differential equations, *Phys. Rev. E* **98**, 022109 (2018).
- [31] N. Campioni, D. Husmeier, J. M. Morales, J. Gaskell, and C. J. Torney, Modelling multiscale collective behavior with Gaussian processes, *Proceedings of the Second International Conference on Statistics: Theory and Applications (ICSTA'20)* (2020), pp. 124–1.
- [32] M. Ragwitz and H. Kantz, Indispensable Finite Time Corrections for Fokker-Planck Equations from Time Series Data, *Phys. Rev. Lett.* **87**, 254501 (2001).
- [33] A. G. d. G. Matthews, M. van der Wilk, T. Nickson, K. Fujii, A. Boukouvalas, P. León-Villagrà, Z. Ghahramani, and J. Hensman, GPflow: A Gaussian process library using TensorFlow, [arXiv:1610.08733](https://arxiv.org/abs/1610.08733).
- [34] J. V. Dillon, I. Langmore, D. Tran, E. Brevdo, S. Vasudevan, D. Moore, B. Patton, A. Alemi, M. Hoffman, and R. A. Saurous, TensorFlow distributions, [arXiv:1711.10604](https://arxiv.org/abs/1711.10604).
- [35] P. R. Conrad, Y. M. Marzouk, N. S. Pillai, and A. Smith, Accelerating asymptotically exact MCMC for computationally intensive models via local approximations, *J. Am. Stat. Assoc.* **111**, 1591 (2016).
- [36] W. K. Hastings, Monte Carlo sampling methods using Markov chains and their applications, *Biometrika* **57**, 97 (1970).
- [37] P. Gardner, T. J. Rogers, C. Lord, and R. J. Barthorpe, Sparse Gaussian Process Emulators for Surrogate Design Modelling, *Applied Mechanics and Materials*, Vol. 885 (Trans Tech Publications Ltd., Switzerland, 2018), pp. 18–31.
- [38] J. T. Santner, J. B. Williams, and I. W. Notz, *The Design and Analysis of Computer Experiments*, Springer Series in Statistics (Springer, New York, 2003).

- [39] R. B. Gramacy, *Surrogates: Gaussian Process Modeling, Design, and Optimization for the Applied Sciences* (CRC Press, Boca Raton, FL, 2020).
- [40] G. Cybenko, Approximation by superpositions of a sigmoidal function, *Math. Control, Signals Systems* **2**, 303 (1989).
- [41] K. Hornik, Approximation capabilities of multilayer feedforward networks, *Neural Networks* **4**, 251 (1991).
- [42] R. M. Neal, *Bayesian Learning for Neural Networks*, Lecture Notes in Statistics, Vol. 118 (Springer Verlag, Berlin, 1996).
- [43] C. Andrieu and G. O. Roberts, The pseudo-marginal approach for efficient Monte Carlo computations, *Ann. Statist.* **37**, 697 (2009).
- [44] C. Andrieu, A. Doucet, and R. Holenstein, Particle Markov chain Monte Carlo methods, *J. R. Stat. Soc. Series B* **72**, 269 (2010).
- [45] J. F. Geweke, *Evaluating the Accuracy of Sampling-Based Approaches to the Calculation of Posterior Moments*, Tech. Rep. 148 Federal reserve bank of minneapolis, 1991.
- [46] T. Vicsek and A. Zafeiris, Collective motion, *Phys. Rep.* **517**, 71 (2012).
- [47] V. Sood, T. Antal, and S. Redner, Voter models on heterogeneous networks, *Phys. Rev. E* **77**, 041121 (2008).
- [48] C. J. Torney, S. A. Levin, and I. D. Couzin, Decision accuracy and the role of spatial interaction in opinion dynamics, *J. Stat. Phys.* **151**, 203 (2013).
- [49] G. Pokharel and R. Deardon, Gaussian process emulators for spatial individual-level models of infectious disease, *Canadian J. Statistics* **44**, 480 (2016).
- [50] M. Sellier, Inverse problems in free surface flows: a review, *Acta Mech.* **227**, 913 (2016).
- [51] M. J. Colebank, L. M. Paun, M. U. Qureshi, N. Chesler, D. Husmeier, M. S. Olufsen, and L. E. Fix, Influence of image segmentation on one-dimensional fluid dynamics predictions in the mouse pulmonary arteries, *J. R. Soc., Interface* **16**, 20190284 (2019).
- [52] L. M. Paun, M. J. Colebank, M. S. Olufsen, N. A. Hill, and D. Husmeier, Assessing model mismatch and model selection in a Bayesian uncertainty quantification analysis of a fluid-dynamics model of pulmonary blood circulation, *J. R. Soc., Interface* **17**, 20200886 (2020).
- [53] H. Meinhardt, *Models of Biological Pattern Formation* (Academic Press, New York, 1982).
- [54] <https://github.com/ctorney/eqn-free-gp-inference>.