Learning black- and gray-box chemotactic PDEs/closures from agent based Monte Carlo simulation data

Seungjoon Lee1 · Yorgos M. Psarellis2 · Constantinos I. Siettos3 · Ioannis G. Kevrekidis2,4,5

Received: 28 November 2022 / Revised: 29 April 2023 / Accepted: 20 May 2023 / Published online: 21 June 2023
© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2023

Abstract
We propose a machine learning framework for the data-driven discovery of macroscopic chemotactic Partial Differential Equations (PDEs)—and the closures that lead to them—from high-fidelity, individual-based stochastic simulations of *Escherichia coli* bacterial motility. The fine scale, chemomechanical, hybrid (continuum—Monte Carlo) simulation model embodies the underlying biophysics, and its parameters are informed from experimental observations of individual cells. Using a parsimonious set of collective observables, we learn effective, coarse-grained “Keller–Segel class” chemotactic PDEs using machine learning regressors: (a) (shallow) feedforward neural networks and (b) Gaussian Processes. The learned laws can be black-box (when no prior knowledge about the PDE law structure is assumed) or gray-box when parts of the equation (e.g. the pure diffusion part) is known and “hardwired” in the regression process. More importantly, we discuss data-driven corrections (both additive and functional), to analytically known, approximate closures.

Keywords Inverse problems · Partial differential equations · Machine learning · Stochastic simulations · Chemotaxis · Numerical analysis · Multiscale methods

Seungjoon Lee and Yorgos M. Psarellis contribute equally to this paper.

Ioannis G. Kevrekidis
yannisk@jhu.edu

1 Department of Applied Data Science, San José State University, San Jose, USA
2 Department of Chemical and Biomolecular Engineering, Johns Hopkins University, Baltimore, USA
3 Dipartimento di Matematica e Applicazioni “Renato Caccioppoli” and Scuola Superiore Meridionale, Università degli Studi di Napoli Federico II, Naples, Italy
4 Department of Applied Mathematics and Statistics, Johns Hopkins University, Baltimore, USA
5 Department of Medicine, Johns Hopkins University, Baltimore, USA
1 Introduction

Since the pioneering work of Adler (1969), the chemotactic motility of bacteria, i.e., their movement in response to changes in the surrounding environment has been extensively studied, thus decoding complex mechanisms ranging from the biochemistry and molecular genetics (Parkinson 1976, 1980) to the inter- (Boyd et al. 1981; Ho et al. 2023) and intra-cellular signaling (Liu and Parkinson 1989; Heit et al. 2002), and from the sensory adaptation in response to external stimuli (Segel et al. 1986; Othmer and Schaap 1998) to the motor structure and the flagellum-related motility (Cluzel et al. 2000) and scaling up to the emergent collective behaviour (Wu et al. 2006). Depending on the level of information and spatio-temporal scale of analysis, a vast number of mathematical models have been proposed, ranging from the molecular/individual (Emonet et al. 2005; Coburn et al. 2013; Othmer et al. 2013; Rouset and Samaey 2013; Yasuda 2017) to the continuum/macroscopic scale (Patlak 1953; Keller and Segel 1971; Erban and Othmer 2004; Bellomo et al. 2010; Othmer et al. 2013; Franz and Erban 2013; Bellomo et al. 2022) (for an extensive review of both modelling approaches see Tindall et al. 2008a, b; Othmer et al. 2013; Bellomo et al. 2022).

The celebrated Keller and Segel (1971) PDE derived for the macroscopic description of the microorganism population density evolution, coupled with the concomittant chemoattractant field, constitutes the cornerstone in the field. In its simplest form, the dependence of the cell density $b(x, t)$ in space and time evolves according to

$$\frac{\partial b}{\partial t} = \nabla \cdot (D \nabla b - \chi(s)b\nabla s),$$

coupled with appropriate boundary conditions; the diffusion term in general depends on the non-chemotactic random motility of the bacteria; here, $D$ is the diffusion coefficient and $\chi$ is the chemotactic coefficient. In general, the effect of the substrate’s distribution and the population density on both $D$ and $\chi$ are not known. In order to obtain an expression in closed-form and then attempt to fuse experimental observations and bio-physical insight, several assumptions are made, resulting to different closed-form approximations. For example, in the original paper of Keller and Segel (1971) it is assumed that $D = D(s)$ and $\chi = \chi(s)$, i.e., that both depend on the distribution of the substrate $s$ (in general $s = s(x, t)$). In their more general forms, the diffusion and chemotactic coefficients depend on both the density $b$ and the substrate profile $s$, i.e. $D = D(b, s)$, $\chi = \chi(b, s)$. Assuming $D = D(b)$, $\chi = \chi(b)$, the Keller–Segel PDE reduces to a Fokker–Planck-type equation, while for constant diffusivity $D$ and constant chemotactic $\chi$ coefficient, we obtain a Smoluchowski-type equation (for a review of different closures and models refer to Othmer and Schaap 1998; Chavanis 2008; Erban and Othmer 2004; Othmer et al. 2013; Painter 2019; Dsilva et al. 2018). Obtaining an accurate constitutive relation (a closure) for the diffusivity $D$ and the chemotactic coefficient $\chi$, one that matches a specific experimental setup and/or uses
statistical mechanics, starting from cell-based models, remains both a challenging and open-ended research task. The goal is to obtain closed-form PDEs, such as those based on the Patlak–Keller–Segel (PKS) model (Patlak 1953; Erban and Othmer 2004; Kim and Yao 2012; Othmer et al. 2013) that can efficiently describe the observed collective dynamics.

Numerical “closure on demand” approaches, based on the Equation-free multiscale framework, that bypass the need to extract explicit macroscopic PDEs in a closed form have been also proposed for the scientific computation of the collective motility dynamics (Setayeshgar et al. 2005; Erban et al. 2006; Siettos 2014). For an early review, see Gorban et al. (2006). Here, starting from high-fidelity data, produced by a detailed realistic biophysical Monte-Carlo model for the motility of Escherichia coli bacteria, whose parameters are calibrated from experimental data (Berg and Brown 1972; Larsen et al. 1974; Maeda et al. 1976; Block et al. 1983, 1982; Ishihara et al. 1983; Spiro et al. 1997; Othmer and Schaap 1998; Cluzel et al. 2000; Othmer et al. 2013), and building on previous efforts (Rico-Martinez et al. 1994; Krischer et al. 1993; Gonzalez-Garcia et al. 1998; Bertalan et al. 2019; Lee et al. 2020; Kemeth et al. 2022), we propose a data-based, machine-learning-assisted framework to learn the law of the underlying macroscopic PDE. In particular, based on automatic relevance determination (ARD) (MacKay 1992; Sandhu et al. 2017) within the Gaussian Process Regression framework (Rasmussen and Williams 2005; Lee et al. 2020) for feature extraction, and on Gaussian Processes and feedforward neural networks to learn the collective dynamics, we (see Fig. 1):
a. Identify the right-hand-side (RHS) of an effective Keller–Segel-class PDE, thus obtaining a black-box PDE model;

b. Reconstruct the chemotactic term only, assuming that the diffusion term can be estimated by the high-fidelity simulations and/or knowledge of the physics, thus constructing a gray box Keller–Segel-class PDE model; and importantly,

c. Discover data-driven corrections to established approximate closure approximations of the chemotactic term, which have been derived analytically from kinetic theory/statistical mechanics based on series of assumptions (Erban and Othmer 2004).

Our methodologies belong to the continuously expanding field of nonlinear system identification using data-driven, machine-learning-assisted surrogate models. Gaussian Processes have been extensively used as surrogate models (Kocijan et al. 2005; Lee et al. 2020; Wan and Sapsis 2017; Raissi et al. 2017). Learning dynamical systems with feedforward neural networks has also been remarkably successful (Rico-Martinez et al. 1994; Lee et al. 2020; Gonzalez-Garcia et al. 1998) as well as machine learning models with more advanced architectures (PINNs Raissi et al. 2019; Raissi and Karniadakis 2018; Yang et al. 2021, Neural ODEs/PDEs Chen et al. 2019, sparse regression Wu and Xiu 2019; Rao et al. 2022; Brunton et al. 2016, recurrent networks Vlachas et al. 2018, 2022, residual networks Chen et al. 2022; Wu and Xiu 2020; Qin et al. 2019 and more Galaris et al. 2022; LeCun and Bengio 1998). Such machine learning algorithms have been also utilized in learning corrections/closures, in the context of turbulence (Duraisamy et al. 2019; Zhang and Duraisamy 2015; Beck et al. 2019) or in a more general context (Ansumali et al. 2005; Pan and Duraisamy 2018; Parish and Duraisamy 2016; Iskhakov et al. 2021; Pathak et al. 2020; Sheriffdeen et al. 2019; Jiang et al. 2019).

The novelty of our work lies, we believe, in the following: (a) When the complexity of behavioral response increases, the extraction of closed form Chemotactic PDEs is hard to obtain analytically: thus, it is difficult to describe chemotactic motility with quantitative yet generalizable models, owing to its inherent complexity. For example, in the presence of large signal gradients, the higher-order moments of the cell distribution may not be negligible, and thus cannot be discarded as independent variables (for a discussion on the difficulties in finding PDEs in a closed form for such problems see Erban and Othmer 2007). What’s more, chemotactic phenomena are highly dependent on bacterial type and strain, chemoattractant type as well as distribution and environmental conditions (to mention a few). Our data-driven models enable non-parametric learning of chemotactic behaviors within the variability seen in the training data. This benefit is more pronounced in our gray-box models, where we focus our regressor to the chemotactic term, for which a generalizable description is, in general, hard or impossible to derive. (b) Chemotactic motility is a multiscale phenomenon: Chemotaxis is an emergent behavior of large bacterial populations. It is especially non-trivial to connect phenomena and parameters at the microscale (cell-level chemical signaling and motion, as e.g. described by the chemomechanical agent-based model) to those in the macroscale (bacterial density traveling waves, as e.g. described by Keller–Segel-type PDEs). The algorithmic pipeline presented here, effectively manages to connect such disparate scales with minimum assumptions and quantitative predictive power.
(c) Calibration of existing models: We present various closure/correction models and emphasize their significance. Such models can incorporate already existing knowledge from not-so-accurate PDEs. We believe that such a data-driven calibration (and the different varieties thereof presented here) can prove useful beyond the field of chemotaxis. They outline a way to conveniently combine established qualitatively correct models (possibly from first principles) with high-fidelity (computational or real-world) data. Such models offer computational researchers “the best of both worlds”: the explainability of existing analytical models and the versatility of machine learning algorithms.

The remainder of the paper is organized as follows: in Sect. 2.1, we briefly present the bio-mechanical based Monte-Carlo model used and discuss its parametrization based on experimental studies. In Sect. 3, we present an overview of our proposed numerical framework and briefly review theoretical concepts of Gaussian Process Regression, ARD feature selection, and Artificial Neural Networks. In Sect. 4, we demonstrate the effectiveness of our proposed framework by constructing different PDE models, namely (1) a black-box, (2) a gray-box, and (3) a corrected closure model for a Keller–Segel-class macroscopic PDE. In Sect. 5, we summarize our results and discuss open issues for further development of the data-driven discovery of the underlying macroscopic PDE from data generated by microscopic simulations and/or experiments.

2 Chemotactic motility of E. coli: the bio-mechanical based Monte-Carlo model and the closed form Keller–Segel (Fokker–Planck-type) PDE

2.1 Monte-Carlo Model

Here, we used a previously derived chemo/bio-mechanical-based Monte-Carlo dynamical model (Othmer et al. 2013) to generate data for the chemotactic motility of E. Coli bacteria in response to a fixed chemoattractant substrate profile (see also Siettos 2014). A pseudo-code of the MC model is given in Appendix A. Below, we provide details about its implementation. The model calculates the probability of rotational directionality of each one of the six flagellae that extend from the surface of the cell based on changes of the concentration of the CheY-P protein, which binds to the protein FliM at the base of the rotor. The changes in the concentration of the CheY-P protein control the direction of the flagellar rotation (Othmer et al. 2013; Sarkar et al. 2010), which in turn governs the motility of the cells. When the majority of the flagellar filaments rotate in the counter-clockwise (CCW) direction the cell swims; otherwise it tumbles. The change in time of the concentration of CheY-P protein, say \( C(t) \) is represented by a simple algebraic relation reading (Setayeshgar et al. 2005; Siettos 2014): \( C(t) = \tilde{C} - gu_1(t) \), where the dynamics of \( u_1(t) \) are modeled by a simple two dimensional excitation/adaptation cartoon model given by:

\[
\begin{align*}
\frac{du_1}{dt} &= \frac{f(s) - (u_1 + u_2)}{\tau_e}, & \frac{du_2}{dt} &= \frac{f(s) - u_2}{\tau_a}, & f(s) &= k \frac{s}{K_s + s}.
\end{align*}
\]
In the above model, $\bar{C} = 2.95\mu M$ is the baseline concentration corresponding to the non-excited state (Cluzel et al. 2000; Setayeshgar et al. 2005), $g = 5$ is the amplification response to excitation (Setayeshgar et al. 2005; Othmer et al. 2013), $s$ represents the external stimulus (here, a chemoattractant substrate), $\tau_e = 0.1$ and $\tau_a = 20$ represent the excitation and adaptation time constants, respectively, and $f$ is the function encoding the signal transduction reflecting the fraction of receptors that are occupied (Othmer et al. 2013; Erban and Othmer 2004); $k = 15$ is a constant that amplifies the input signal and $K_s = 1 \mu M$ is the dissociation constant for the enzyme-substrate complex (Erban and Othmer 2004; Block et al. 1983; Setayeshgar et al. 2005).

The (mean) swimming speed ($\bar{v}$) (see the pseudo-code given in Appendix B) also depends on various parameters such as the bacteria strain, the substrate, temperature and density of cells, and may vary from $\sim 10$ to $\sim 55$ $\mu m/s$ (Berg and Brown 1972; Maeda et al. 1976). For our simulations, we have set $v = 30 \mu m/s$. Finally, we note that in the MC model (that is based on the kinetics presented in Erban and Othmer 2004), the movement of the bacteria is not affected by their density (they are “noninteracting”). Many organisms use a random walk strategy to determine their movement. In this case, the movement of organisms released at a point in a uniform dilute environment can be described as an uncorrelated, unbiased random walk of noninteracting particles on a sufficiently long time scale (Erban and Othmer 2004). More details on the MC model and its pseudo-code implementation can be found in the Appendix and in Setayeshgar et al. (2005), Siettos (2014).

### 2.2 The closed form Keller–Segel-type PDE

For a microscopic description of the $E. coli$ motility model analogous to the above, Erban and Othmer (2004), with the aid of statistical mechanics/kinetic theory, and assuming that the the signal $s(x)$ is a time independent scalar function, have derived the following parabolic Keller–Segel-class PDE in closed form in 1D:

$$\frac{\partial b}{\partial t} = \frac{\partial}{\partial x} \left( \bar{v}^2 \frac{\partial b}{\partial x} - \frac{df}{ds} \frac{c\bar{v}^2\tau_a}{\lambda_0(1 + 2\lambda_0\tau_a)(1 + 2\lambda_0\tau_e)} \frac{ds}{dx} b \right). \quad (3)$$

Here, $b = b(x, t)$ is the density of the population at $x$ and time $t$, $\bar{v}$ is the mean speed of the bacterium’s motion, $\lambda_0$ represents the basal turning frequency (frequency of tumbles) in the absence of excitation, and $c$ is a positive constant parameter that amplifies the excitation signal $u_1$ that governs the switching frequency in the presence of a stimulus (for a detailed description of the derivation of the above PDE and its parameters please refer to Erban and Othmer (2004)). Based on the above, we define the generalized CHemotactic term $C H_g$ as

$$C H_g \equiv \frac{\partial}{\partial x} \left( \frac{df}{ds} \frac{c\bar{v}^2\tau_a}{\lambda_0(1 + 2\lambda_0\tau_a)(1 + 2\lambda_0\tau_e)} \frac{ds}{dx} b \right); \quad (4)$$
the subscript $g$ refers to the generalized Keller–Segel PDE. The main assumptions made for the above closed form PDE are the following: (a) all bacteria are running with a constant velocity $\bar{v}$, without colliding, (b) the tumble phase is neglected, (c) the internal excitation–adaptation dynamics are described by the 2D cartoon model, (d) the distribution of the substrate $s(x)$ is a time-independent scalar function, (e) the turning frequency (frequency of tumbles in the presence of stimulus), say, $\lambda(t)$ is a linear function of $u_1(t)$ given by

$$\lambda(t) = \lambda_0 - cu_1(t), \quad (5)$$

Note that $\lambda(t)$ does not appear explicitly in the closed form PDE given in Eq. (3), but implicitly via the parameter $c$. In Appendix B2, we explain in detail how one can compute $c$ by measuring $\lambda$ and $\lambda_0$ from microscopic simulations.

Finally, it is assumed (f) that the gradient $s'(x)$ is shallow, so that for all practical purposes, the second order moment of the microscopic flux is zero, i.e. that Erban and Othmer (2004):

$$j_2(x, t) = \int_\mathbb{R} \bar{v} \cdot (p_+(x, z_2, t) - p_-(x, z_2, t))dz_2 = 0. \quad (6)$$

$p_{\pm}(x, z_2, t)$ is the density function of the bacteria at $(x, t)$ with the internal state $z_2(x, t) = u_2(t) - s(x)$ that run right $(+)$ or left $(-)$. The above assumptions result to the following closed form for the chemotactic coefficient $\chi$ (Erban and Othmer 2004):

$$\chi = \frac{df}{ds} \frac{c\bar{v}^2\tau_a}{\lambda_0(1 + 2\lambda_0\tau_a)(1 + 2\lambda_0\tau_e)}. \quad (7)$$

3 Machine-learning and the discovery of chemotactic laws: data-driven identification of coarse-scale PDEs

3.1 Overview

Before discovering effective PDE laws from microscopic simulations, there is a crucial prerequisite: we need to know the macroscopic observables whose field evolution laws we want to discover. There are cases for which this knowledge is given a priori: in our case we know we want to derive parabolic evolution laws for the bacterial density field. Yet this knowledge is not always a priori given, based on domain knowledge: for the chemotaxis problem itself, we know that in different parameter regimes one needs a hyperbolic (higher order) equation for the density field (Erban and Othmer 2004; Dsilva et al. 2018). Discovering sets of macroscopic observables in terms of which an evolutionary PDE can be closed is a nontrivial task; often this task can be performed using data mining/manifold learning techniques, and has been named “variable-free computation” (in the sense that the relevant variables are identified through, say, PCA or Diffusion Map processing of the fine scale simulations (Erban et al. 2007; Arbabi and Kevrekidis 2021).
In our case (with the chemoattractant field \( s(x) \) fixed, and with Keller–Segel-class equations in mind) we know that we want to identify a parabolic evolutionary PDE in terms of the evolution of a normalized bacterial density \( b(x, t) \) in one spatial dimension. That is, we expect \( \frac{\partial b}{\partial t} = F \left( b, \frac{\partial b}{\partial x}, \frac{\partial^2 b}{\partial x^2}, \ldots, s, \frac{ds}{dx}, \frac{d^2 s}{dx^2}, \ldots \right) \). The existence of such a relation between the local bacterial density time derivative and the local values and first few spatial derivatives of this field, as well as of the chemoattractant field, is our working hypothesis. The question then becomes: how many spatial derivatives of the \( b(x, t), s(x, t) \) fields are required in order to infer a useful data-driven closure? Starting with an assumed highest order of possibly influential spatial derivatives, we resolve this here using a feature selection method. More specifically, we use the automatic relevance determination (ARD) in the Gaussian framework (Rasmussen and Williams 2005), which has been widely used to identify dominant input features for a certain target output using sensitivity analysis (Liu et al. 2019; Lee et al. 2020, 2021), see Sect. 3.2.1. This feature selection also provides not only computational cost reduction but also a better physical understanding of the underlying PDEs.

### 3.2 Machine learning regression methods and feature selection

#### 3.2.1 Gaussian process (GP) regression and feature selection via ARD

In Gaussian process regression, we describe probability distributions of target (hidden) functions under the premise that these target functions are sampled from a (hidden) Gaussian process, which is a collection of random variables, such that the joint distribution of every finite subset of random variables is itself a multivariate Gaussian. This assumption is reasonable when the target functions, \( \frac{\partial b}{\partial t} = F \left( b, \frac{\partial b}{\partial x}, \frac{\partial^2 b}{\partial x^2}, \ldots, s, \frac{ds}{dx}, \frac{d^2 s}{dx^2}, \ldots \right) \), are continuous and sufficiently smooth in a target domain (Rasmussen and Williams 2006; Lee et al. 2019, 2020). To identify the hidden Gaussian process from the data \( \{ x, y = F(x) = \frac{\partial b}{\partial t} \} \) (where \( x \) represent the input vectors), we need to approximate a mean, \( m(x) = \mathbb{E}[F(x)] \) and a covariance function, \( K(x, x') = \mathbb{E}[(F(x) - m(x))(F(x') - m(x'))] \) from the given data:

\[
F \sim GP(m(x), K(x, x')).
\]

Generally, we assign a zero mean function for \( m(x) \) and approximate the covariance function, \( K(x, x') \), based on the training data set. To approximate the covariance matrix, we employ a radial basis kernel function (RBF, denoted \( \kappa(\cdot, \cdot) \)), which is the default kernel function in Gaussian process regression:

\[
K_{ij} = \kappa(x_i, x_j; \theta) = \theta_0 \exp \left( -\frac{1}{2} \frac{(x_i - x_j)^2}{\theta_1^2} \right),
\]

\( \theta = [\theta_0, \theta_1]^T \) is a vector of hyperparameters to be optimized. The optimal hyperparameter set \( \theta^* \) can be obtained by minimizing a negative log marginal likelihood over the training data set (of \( n_{\text{train}} \) data points). Using the kernel function with optimal
hyperparameters, we represent a \((n_{\text{train}} + n_{\text{test}})\)-dimensional Gaussian distribution as

\[
\begin{bmatrix}
y \\
y^*\end{bmatrix} \sim N \left( 0, \begin{bmatrix} K + \sigma^2 I & K_* \\ K^T_* & K_{**} \end{bmatrix} \right),
\]

(10)

where \(y^*\) is our prediction of the distribution at the test conditions, \(x^*, K_*\) represents the covariance matrix between training and test data, while \(K_{**}\) represents the covariance matrix between test data. Also, \(\sigma^2\) and \(I\) represent the variance of the (Gaussian) observation noise and the \(n \times n\) identity matrix, respectively.

The predicted distribution (posterior) of \(n_{\text{test}}\) data, \(y^*\), is a Gaussian distribution conditioned on the training data:

\[
y^* \sim N \left( K_* (K + \sigma^2 I)^{-1} y, K_{**} - K^T_* (K + \sigma^2 I)^{-1} K_* \right);
\]

(11)

we use the predicted mean (\(\bar{y}^*\)) as an estimate of our Quantity of Interest, e.g. of the local time derivative of the bacterial density.

To identify the salient input features (feature selection), we modify the kernel function from the default in Eq. (9). The ARD modification consists of assigning an individual hyperparameter \(\theta_l\) to each input feature (dimension) in the new covariance kernel, leading to a higher overall dimensional hyperparameter set as

\[
K_{ij} = \kappa(x_i, x_j; \theta) = \theta_0 \exp \left( -\frac{1}{2} \sum_{l=1}^{k} \frac{(x_{i,l} - x_{j,l})^2}{\theta_l} \right),
\]

(12)

where \(\theta = [\theta_0, \ldots, \theta_k]^T\) is a \(k + 1\) dimensional vector of hyperparameters and \(k\) is the number of dimensions of the input data domain.

After hyperparameter optimization, we check the magnitude of the optimal hyperparameters. As shown in Eq. (12), a large magnitude of the hyperparameter \(\theta_l\) nullifies the contribution along that direction to the distance metric (in the numerator), leading to the relative insignificance (low sensitivity) of that input feature towards the predictions. Hence, we select only a few input features that have “relatively” small magnitudes of their hyperparameters \(\theta_l\). After selecting a few salient features, we reconstruct the reduced GP model with only these selected input features, resulting in computational cost reduction. In our case this will lead to learning an approximate equation using less overall spatial derivatives in the Right-Hand-Side (RHS) than the ones retained in the physical model.

### 3.2.2 Feedforward neural networks (FNN)

Here, for learning the right-hand-side of the effective PDE, we have used an FNN with two hidden layers and a linear output layer. For each point in space and time \((x, t)\), the input to the FNN consists of the values of the cell density at the point, \(b(x, t)\), its spatial derivatives, the chemoattractant profile \(s(x)\) and its spatial derivatives, say, \(u = [b, \frac{\partial b}{\partial x}, \frac{\partial^2 b}{\partial x^2}, \ldots, \frac{\partial^m b}{\partial x^m}, s, \frac{d s}{d x}, \frac{d^2 s}{d x^2}, \ldots, \frac{d^k s}{d x^k}]^T \in \mathbb{R}^{k+m+2}\), while its output \(F(u)\):
\( \mathbb{R}^{m+k+2} \to \mathbb{R} \) is the associated time derivative \( \frac{\partial b(x, t)}{\partial t} \) at the same \((x, t)\) point. Thus, the FNN reads:

\[
F(u) = W^o \Phi_2(W_2 \Phi_1(W_1 u + \beta_1) + \beta_2) + \beta_0. \tag{13}
\]

\( W^o \) is the matrix containing the weights connecting the second hidden layer to the linear output layer, \( \Phi_1, \Phi_2 \) denote the activation functions of the first and second hidden layers respectively, \( W_1 \) is the matrix containing the weights from the input to the first hidden layer, \( W_2 \) is the matrix with the weights connecting the first hidden to the second hidden layer, \( \beta_1, \beta_2 \) are the vectors containing the biases of the nodes in the first and second layers, respectively, and \( \beta_0 \) is the bias of the output node. As has been proved by Chen and Chen (1995), such a structure (with sufficient neurons) can approximate, to any accuracy, non-linear laws of the time evolution of dynamical systems.

### 3.3 Extraction of coarse-scale PDEs from the microscopic Monte Carlo simulations

We learn three different types of data-driven PDE models, namely, (1) a black-box, (2) a gray-box, and, (3) PDE RHS’s whose closures are learned (based on GPs and FNNs) as corrections of the analytically available Keller–Segel PDE closure; how many spatial derivatives are kept in the identified RHS can be found via the ARD process above.

In what follows, we describe the basic steps of our numerical scheme.

#### 3.3.1 Black-box model

We start by learning a black-box model for the local dependence of the time derivative of the bacterial density, \( \frac{\partial b}{\partial t} \), on the (local) density and its spatial derivatives as well as on the local chemoattractant concentration and its derivatives, \( \text{1} \) (i.e. the chemotactic PDE RHS operator) as:

\[
\frac{\partial b}{\partial t} = B \left( b, \frac{\partial b}{\partial x}, \frac{\partial^2 b}{\partial x^2}, \ldots, \frac{\partial^m b}{\partial x^m}, \frac{d s}{d x}, \frac{d^2 s}{d x^2}, \ldots, \frac{d^k s}{d x^k} \right), \tag{14}
\]

The orders \( m \) and \( k \) can be are either qualitatively known a priori (as part of the problem physics) or can be identified by the ARD feature selection algorithm (see Table 1).

#### 3.3.2 Gray-box model

In several cases, we may know some component of the macroscopic dynamic behavior that comes from intuition, previous studies and/or experiments. For example, one can compute through simulations and/or the aid of statistical mechanics the diffusion

---

1 We note here that the identification is performed in Euclidean space; in the case of spherical or cylindrical geometries we may need to express the right-hand-side not in terms of derivatives wrt. the independent variables, but rather in their coordinate-invariant form Psarellis et al. (2022).
### Table 1: Selected groups of input features and the corresponding predicted quantity (output) for different data-driven PDE law correction approaches

| Data-driven model          | Input features                                                                 | Output |
|----------------------------|-------------------------------------------------------------------------------|--------|
| Black-box (Full)           | \( b, \frac{\partial b}{\partial x}, \frac{\partial^2 b}{\partial x^2}, s, \frac{ds}{dx}, \frac{d^2 s}{dx^2} \) | \( \frac{\partial b}{\partial t} \) |
| Black-box (Reduced)        | \( b, \frac{\partial b}{\partial x}, \frac{\partial^2 b}{\partial x^2}, s, \frac{ds}{dx} \) | \( \frac{\partial b}{\partial t} \) |
| Gray-box (Full)            | \( b, \frac{\partial b}{\partial x}, \frac{\partial^2 b}{\partial x^2}, s, \frac{ds}{dx}, \frac{d^2 s}{dx^2} \) | \( CH \) |
| Gray-box (Reduced)         | \( b, \frac{\partial b}{\partial x}, \frac{\partial^2 b}{\partial x^2}, s, \frac{ds}{dx} \) | \( CH \) |
| Functional correction (Full) | \( \partial CH_g, \frac{\partial CH_g}{\partial b}, \frac{\partial CH_g}{\partial bx}, \frac{\partial CH_g}{\partial s}, \frac{\partial CH_g}{\partial sx}, \frac{\partial CH_g}{\partial sxx} \) | \( CH \) |
| Functional correction (Reduced) | \( \frac{\partial CH_g}{\partial b}, \frac{\partial CH_g}{\partial bx}, \frac{\partial CH_g}{\partial s}, \frac{\partial CH_g}{\partial sx}, \frac{\partial CH_g}{\partial sxx} \) | \( CH \) |
| Correction (No derivatives) | \( CH_g, b, s \) | \( CH \) |
| Additive correction        | \( b, \frac{\partial b}{\partial x}, \frac{\partial^2 b}{\partial x^2}, s, \frac{ds}{dx}, \frac{d^2 s}{dx^2} \) | \( CH - CH_g \) |

“Reduced” represents models constructed using the (fewer) input features selected via ARD within the Gaussian process framework. We only reduce GP models through ARD; the corresponding NN reduction via autoencoders was not attempted. For the definitions of CHg, CH see Eq. (4) and Sect. 3.3.2 respectively.

The coefficient of the bacterial density. One can then write a gray-box model which contains a Known Term—such as the diffusion term—\( KT \), and Unknown Terms (the chemotactic terms) \( UT \):

\[
\frac{\partial b}{\partial t} = UT \left( b, \frac{\partial b}{\partial x}, \frac{\partial^2 b}{\partial x^2}, \ldots, \frac{\partial^m b}{\partial x^m}, s, \frac{ds}{dx}, \frac{d^2 s}{dx^2}, \ldots, \frac{d^k s}{dx^k} \right) + KT. \tag{15}
\]

Here, we assume that we know (or we can infer) the diffusion term, including the corresponding diffusion coefficient, \( D \). i.e., \( KT_{\text{Known}} = D \frac{\partial^2 b}{\partial x^2} \). In fact, here \( D \) is computed from appropriately designed microscopic random-walk/Brownian motion Monte Carlo simulations in the absence of a chemical gradient, (see Appendix B.1 for more details). Therefore, in our case, the Unknown Terms include just the chemotactic term, i.e. \( \frac{\partial b}{\partial t} = D \frac{\partial^2 b}{\partial x^2} + CH \) or \( CH = \frac{\partial b}{\partial t} - D \frac{\partial^2 b}{\partial x^2} \).

### 3.3.3 Learning closures and their corrections

If some partial information is known (e.g. some of the terms in the RHS of the PDE), we can apply the gray-box approach discussed. However, sometimes, we may have a closed form, analytical, qualitatively good (but less accurate quantitatively) PDE model/closure, capable of describing in a qualitative manner similar macro-scale dynamics. For example, we may have a closed-form PDE RHS that is capable of predicting qualitative trends, but is not suitable for accurate predictions in our particular experimental setup (e.g., one obtained for different types of chemoattractants, different types of bacteria, etc). This can be thought of as a “low-fidelity” model and it can be used in the same spirit it would be used in a multifidelity data fusion context (see Lee et al. 2019; Perdikaris et al. 2017). Exploiting such existing approximate models,
we propose a different type of gray-box machine learning scheme, to calibrate the model to observation data, so as to match our specific experimental set-up/our detailed numerical simulations.

We propose four types of data-driven closure corrections to enhance the accuracy of an effective “low-fidelity” PDE. In particular, we used the “generalized PDE” model for the chemotactic dynamics introduced in Erban and Othmer (2004) (please refer to Sect. 2.2 for more details) as the low-fidelity reference model we want to correct. The four types of closure correction models are detailed in the schematic Fig. 2 and in Table 1. First, we used a machine-learning scheme (based on GP or an FNN), to correct the chemotactic term $CH_g$ of the generalized PDE so as to learn the true chemotactic term $CH$. It may be that the quantitative closure is a simple, smooth function $F$ of the analytical approximate closure $CH_g$ in the general form of:

$$CH\left(b, \frac{\partial b}{\partial x}, \ldots, \frac{\partial^m b}{\partial x^m}, s, \frac{ds}{dx}, \ldots, \frac{d^ks}{dk^2}\right) = F\left(CH_g\left(b, \frac{\partial b}{\partial x}, \ldots, \frac{\partial^m b}{\partial x^m}, s, \frac{ds}{dx}, \ldots, \frac{d^ks}{dk^2}\right)\right).$$

(16)

More often than not, this does not suffice, and more information/more observations/more variables are necessary for quantitative prediction. Our first approach is to
exploit data-driven embedding theories (in the spirit of Whitney/Takens embeddings Whitney 1936; Nash 1966; Takens 1981; Lee et al. 2019) to discover corrections from the known \( CH_g \) to the unknown \( CH \) using the first functional derivatives of \( CH_g \) wrt. its variables (Lee et al. 2019):

\[
CH = H \left( CH_g, \frac{\partial CH_g}{\partial b}, \frac{\partial CH_g}{\partial b_x}, \frac{\partial CH_g}{\partial s}, \frac{\partial CH_g}{\partial s_x}, \frac{\partial CH_g}{\partial s_{xx}} \right). \tag{17}
\]

Within the GP framework, we also identified a second “version” of this closure correction, using fewer, dominant such derivatives via ARD analysis as:

\[
CH = H \left( \frac{\partial CH_g}{\partial b}, \frac{\partial CH_g}{\partial b_x}, \frac{\partial CH_g}{\partial s}, \frac{\partial CH_g}{\partial s_x}, \frac{\partial CH_g}{\partial s_{xx}} \right). \tag{18}
\]

As a second idea, also conceptually based on embedding theories, we considered an additional closure correction approach, in which the equation RHS was not just a function of the approximate \( CH_g \), but also included additional local inputs (in our first attempt we included the local bacterial density \( b \) and the local chemoattractant \( s \)) as additional information, so that the corrected closure is a learned function of the form

\[
CH = h(CH_g, b, s). \tag{19}
\]

Finally, we also tried a simple “additive” correction; we learned the bias term between the observed chemotactic term and the approximated chemotactic term of the generalized PDE in the form:

\[
CH - CH_g = H \left( b, \frac{\partial b}{\partial x}, \frac{\partial^2 b}{\partial x^2}, s, \frac{ds}{dx}, \frac{d^2 s}{dx^2} \right). \tag{20}
\]

Derivatives up to second order in space are sufficient to provide a closed right-hand-side. This is a qualitative physics-informed decision. Data-driven approaches to making such qualitative decisions are discussed in Li et al. (2003).

### 4 Results

#### 4.1 Monte Carlo simulations and data collection

For our computations, we run a Monte Carlo simulation of \( n = 5000 \) bacteria initially located at \( x = 5.5 \), from \( t = 0 \) to \( t = 5000 \) s with \( dt = 2s \) as reporting horizon, and collect the training data. For training, we collected data from four (fixed in time) different chemo-nutrient concentration profiles, all in the form of a Gaussian distribution \( s(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2} \left( \frac{x-\mu}{\sigma} \right)^2 \right) \): (1) \( \mu = 6, \sigma = 1 \); (2) \( \mu = 6, \sigma = 1.5 \); (3) \( \mu = 7, \sigma = 1.5 \); (4) \( \mu = 7, \sigma = 1.25 \). Specifically, from 5000 individual trajectories of the bacterial motion, we estimated the normalized bacterial density \( b(x, t) \) on a
uniform grid from \( x = 3 \) to \( x = 9 \) with \( dx = 0.05 \) at every time step using kernel smoothing (Bowman and Azzalini 1997) as:

\[
 b(x, t) = \frac{1}{nh} \sum_{i=1}^{n} K \left( \frac{x - x_i(t)}{h} \right), \tag{21}
\]

where \( K(\cdot) \) represents the kernel smoothing function (here, a Gaussian function), and \( h \) is the bandwidth (here, set to \( h = 0.3 \)). For the approximation of the first \( \partial b/\partial x \) and \( \partial b/\partial t \) and the second \( \partial^2 b/\partial x^2 \) partial derivatives of the density profile, we used central finite differences. Thus, our data-driven models are constructed based on six input features \((b, \partial b/\partial x, \partial^2 b/\partial x^2, s, ds/dx \) and \( d^2 s/dx^2)\) that are used for learning the corresponding time derivative \( \partial b/\partial t \).

### 4.2 Numerical details: training and integration

Gaussian Process learning was performed in Matlab using a RBF kernel with ARD. For feature selection, the cut-off is \( 10^3 \). That is, if the optimal hyperparameter value is higher than \( 10^3 \), we eliminate the corresponding input features. Neural Network learning was performed in Tensorflow (Abadi et al. 2015) with two hidden layers with \([9, 8, 8]\) neurons each (Black-box, Gray-box, Correction model respectively), equipped with a hyperbolic tangent activation function. The Neural Network was trained using an Adam optimizer (Kingma and Ba 2014) and MSE loss while the training hyperparameters were tuned empirically (Epochs \([2560, 10240, 2560]\), Batches \([800000, 750000, 300000]\), Initial Learning Rate 0.02 with a plateau learning rate scheduler with patience 1200 epochs and factor 0.5).

After learning the PDE right-hand-side we first tested whether the laws identified could reproduce the trajectories from which the training data were collected. For illustration, we performed numerical integration with the data-driven learned PDEs (using both GP and FNN) from \( t = 20 \) s to \( t = 4020 \) s with \( dt = 2 \) s using a 4-th order Runge–Kutta scheme, as well as a commercial package integrator (explicit Runge–Kutta method of order 5(4) Dormand and Prince 1980 as implemented by `solve_ivp` in Python, resulting in a maximum absolute error of \( 4 \times 10^{-6} \) and the corresponding integrator (and tolerances) in Matlab with maximum absolute error of \( 9 \times 10^{-4} \). High spatial frequency Fourier modes of the bacterial density profile were consistently filtered (a procedure analogous to adding hyperviscosity in hydrodynamic models, Thiem et al. 2021).

### 4.3 Discussion

The ground truth spatiotemporal evolution of the bacterial density is shown in Fig. 3a and the corresponding relative errors are shown in Fig. 3b while in Fig. 3c, we show the reconstructed profile at \( t = 1000 \) s for one of our training chemo-nutrient profiles:

\[
 s(x) = \frac{1}{\sqrt{2\pi 1.25^2}} \exp\left(-\frac{1}{2} \left(\frac{x-7}{1.25}\right)^2\right)
\]

The performance of our several different data-driven PDE closure corrections was assessed in terms of the relative approximation error.
Fig. 3  a Training/reconstruction case. Ground Truth (GT) evolution for $\mu = 7, \sigma = 1.25$: smoothened profiles of bacterial density derived from post-processing agent-based simulations. b Quantitative performance of representative data-driven models (DD) trained by Gaussian Process Regression (GP), reduced Gaussian Process Regression with ARD (GP+ARD) or a Neural Network (NN): relative error (%) based on maximum density for black-box model, gray-box model, and correction model (first, second and third rows respectively). c Qualitative comparison of profiles of bacterial density at $t = 1000$ s: PDEs learned through (left) Gaussian process; (middle) Reduced Gaussian processes (with ARD); and (right) Neural Networks. Profiles are colored as follows: blue—ground truth, orange—black-box, green—gray-box, red—correction, purple—PDE Eq. (3). The bottom row is a blowup of the profile’s peak. Note that for the Monte-Carlo simulations the initial state ($t = 0$ s) is at $x = 5.5$ cm for all agents, while all PDE simulations (DD models, PDE in 3) begin at $t = 20$ s. No flux boundary conditions were used, consistent with Erban and Othmer (2004)

between the ground truth density profiles ($b_{GT}^{(x,t)}$) and the profiles ($b_{DD}^{(x,t)}$) resulting from numerical integration of the learned approximate PDE right-hand-sides. This error was defined as $E_r = 100\frac{|b_{GT}^{(x,t)} - b_{DD}^{(x,t)}|}{\max b_{GT}^{(x,t)}}$; a comparison to the density profile of the full Monte Carlo simulations at $t = 1000$ s is also provided.

After that, we tested the performance of the data-driven PDE closure corrections with the test data from a new chemoattractant concentration profile (not included in the
Fig. 4  

(a) Testing case. Ground Truth (GT) evolution for $\mu = 6.5, \sigma = 1.35$: smoothened profiles of bacterial density derived from post-processing agent-based simulations.  

(b) Quantitative performance of representative data-driven models (DD) trained by Gaussian Process Regression (GP), reduced Gaussian Process Regression with ARD (GP+ARD) or a Neural Network (NN): relative error (%) based on maximum density for black-box model, gray-box model, and correction model (first, second and third rows respectively). 

(c) Qualitative comparison of profiles of bacterial density at $t = 1000$ s: PDEs learned through (left) Gaussian process; (middle) Reduced Gaussian processes (with ARD); and (right) Neural Networks. Profiles are colored as follows: blue—ground truth, orange—black-box, green—gray-box, red—correction, purple—PDE Eq. (3). The bottom row is a blowup of the profile’s peak. Note that for the Monte-Carlo simulations the initial state ($t = 0$ s) is at $x = 5.5$ cm for all agents, while all PDE simulations (DD models, PDE in Eq. 3) begin at $t = 20$ s. No flux boundary conditions were used, consistent with Erban and Othmer (2004)

training data set): $s(x) = \frac{1}{\sqrt{2\pi}1.35^2} \exp(-\frac{1}{2} \frac{(x-6.5)^2}{1.35^2})$. The ground truth of the testing case is plotted in Fig. 4a. The predicted profile at $t = 1000$ s and the corresponding relative errors are shown in Fig. 4b and c, respectively. Table 1, summarizes the different data-driven models with respect to (1) machine learning techniques, (2) selected features, and (3) the corresponding predicted quantity.

A benefit of reduced feature selection is that the computational cost in the training phase is reduced, while (as shown in Figs. 3 and 4) the predictive accuracy is retained.
Regarding the black-box trajectory reconstruction relative error for the training data set, the GP data-driven models never exceed 14% relative error when integrated for a long time (15% for the ARD-reduced GP), while the FNN data-driven models never exceed 4% relative error. For the test data-set, the respective maximum relative errors are 20% and 12% for the GP (full or reduced) and FNN, respectively.

Interestingly, the largest errors observed for the GP are concentrated during the fast, initial transient, while the trajectories become more accurate at later times as they approach steady state. FNNs seem to perform better in capturing this initial transient.

As shown in Figs. 3 and 4, the gray-box models have the potential to provide comparable, or improved accuracy, compared to the black-box ones. Since gray-box models are learning, in principle, different functions than black-box models, the actual computational benefit can (and will) vary. It can also vary across different learning algorithms: we observe that neural networks more prone to overfitting (Kamath et al. 2018), especially around input values underrepresented in the training dataset (such as, in this case, inputs with large gradients). Conversely, Gaussian Process Regression is more robust and smooth by design (see Eq. 11). This can be also rationalized in terms of the number of hyperparameters optimized for each learning algorithm.

Specifically, the maximum reconstruction relative errors for the training data set are 9% for the GP (12% for reduced GP) and 10% for the FNN models, while for the testing case these are 12% for the GP (19% for reduced GP) and 10% for the FNN models. These results rationalize the capabilities of gray-box models, which combine partial physical knowledge (exact, or even approximate) with data-driven information, towards accurate and efficient data-assisted modelling of complex systems.

There are, of course no guarantees here for the accurate generalization of the predictions beyond the training data; yet the performance of our models over the test set, and also for chemoattractant profiles not included in the training (see second paragraph of Sect. 4.3), appears promising. Our expectation is that the closure correction models, by always making use of the “low-fidelity” (qualitative) information at every time step, may generalize better.

Here, results are presented for two of the four closure correction approaches, i.e. the ones described in Eqs. (17, 18). In particular, the maximum relative errors for the training case were 10% for the GP (8% for reduced GP) and 6% for the FNN models, while for the testing case they were 5% for the GP (4% for reduced GP) and 4% for the FNN models. Thus, all different closure correction approaches (even though the figures for the “additive” and the “no derivatives” corrections are not included for economy of space), provide reasonable accuracy for validation as well as test profiles qualitatively and even quantitatively. Detailed performance of all models (in terms of maximum and average relative error) can be found in Tables 2 and 3 of the Appendix.

Some interesting observations can be made from Figs. 3 and 4. First, all models presented here, seem to capture the qualitative characteristics of the bacterial density traveling wave (e.g. direction, speed, height) as seen in the middle and bottom panels of Figs. 3 and 4. Note that this is achieved at a tremendously reduced computational cost compared to the full MC simulation (order $10^2$). Direct performance comparison
between ANN-based and GPR-based models is not the focus of this work; here, our object was to establish that both approaches can identify models that are macroscopically accurate. All models seem to perform better on the reconstruction case, than the testing case. In both reconstruction and testing cases, the functional correction model seems to be performing consistently better. This could be attributed to its more meaningful, physics-informed inputs: the nonlinear combinations of spatial derivatives relevant to the RHS are already part of the input, while the black- and gray-box models, needed to learn them.

5 Conclusions

Machine learning has long been used to solve the inverse problem, i.e. the identification of nonlinear dynamical systems models from data (Rico-Martinez et al. 1992; Krischer et al. 1993; Masri et al. 1993; Rico-Martinez et al. 1994; Chen and Chen 1995; Gonzalez-Garcia et al. 1998; Siettos et al. 2002; Siettos and Bafas 2002; Alexandridis et al. 2002; Raissi et al. 2017, 2019). Recently, due to technological and theoretical advances, there has been a renewed interest, especially for the case of multiscale/complex systems (Raissi et al. 2019; Kemeth et al. 2022; Lee et al. 2020; Vlachas et al. 2020; Chen et al. 2021; Karniadakis et al. 2021). We presented a machine-learning framework for the numerical solution of the inverse problem in chemotaxis. In particular, we showed how one can learn black-box and gray-box parabolic PDEs for the emergent dynamics of bacterial density evolution (and, importantly, unknown closures and their corrections) directly from high-fidelity microscopic/stochastic simulations. Specifically, we introduced a computational data-driven framework for nonlinear PDE/closure identification and correction; the framework consisted of three progressively more physics-informed processes: (a) learning a black-box PDE (learning the right-hand-side of an coarse-scale PDE including the diffusion term), (b) learning a gray-box PDE (an entire unknown closure, with a known Diffusion term), and (c) obtaining closure corrections (providing a correction of an analytically available closure in a low-fidelity, approximate PDE model). Within this framework, we exploited the Automatic Relevance Determination (ARD) algorithm for feature selection, in order to reduce the number of variables on which the closure depends. We note that a discussion of the possibility (in the spirit of Takens embeddings) of using short term measurement histories to mitigate partial observations is the subject of a forthcoming manuscript. Psarellis et al. (2022). Our overall approach forms a bridge between analytical/mechanistic/physical understanding, and data-driven “black-box” or “gray-box” learning of physical process dynamics, allowing for a synergy between varying types of physical terms/models and data-driven terms/models.

Acknowledgements This work was partially supported by the US Department of Energy, by the US Air Force Office of Scientific Research and by DARPA. C. S. was partially supported by INdAM, through GNCS and the Italian research fund FISR2020IP - 02893.

Data availability The datasets generated during and/or analysed during the current study are available from the corresponding author on reasonable request.
Declarations

Conflict of interest The authors have no competing interests to declare that are relevant to the content of this article.

Appendix A: Details on the Monte Carlo chemotaxis model

Our microscopic, agent-based model is based on the work of Othmer and Schaap (1998); Othmer et al. (2013). Each bacterium is modelled as having six flagellae; special care has been taken in modelling the direction of the rotation of the flagellar filaments, as this constitutes the basis of chemotaxis (Larsen et al. 1974; Spiro et al. 1997). Following Scharf et al. (1998), the motor dynamics are described by a two-state system modelling the transition rates (transition probabilities per unit time) between CCW and CW (counter-clockwise and clockwise, respectively) rotation for each flagellum. These are characterized by an exponential distribution of time intervals in each state (Turner et al. 1996). Let us denote by $k^+ (k^-)$ the transition rate from CCW to CW (CW to CCW). Then, the bias of CW, i.e. the fraction of time that a flagellum rotates CW is $p_{CW} = k^+ / (k^+ + k^-)$. The reversal frequency in the direction of rotation of the flagellar motors is Turner et al. (1996):

$$\rho = p_{CCW} k_+ + p_{CW} k_- = \frac{2k^+k^-}{k^+ + k^-}.$$ (A.1)

and the rate constants are given by:

$$k_+ = \frac{\rho}{2p_{CCW}}, \quad k_- = \frac{\rho}{2p_{CW}}$$ (A.2)

For a cell with $N$ flagellae, the total CCW bias of the cell is given by Spiro et al. (1997):

$$P_{CCW} = \sum_{j=0}^{N} \binom{N}{j} p_{CCW}^j (1 - p_{CCW})^{N-j}.$$ (A.3)

For $N = 6$, $p_{CCW} = 0.64$ and $\theta = N/2$, we get $P_{CCW} \sim 0.87$ suggesting that the cell spends around 90% of the time running (Spiro et al. 1997). This result is in line with experimental observations for the motility of wild-type E. coli in the absence of changes of the substrate, where the mean run (swimming) periods are $\sim 1$ s and the tumble periods $\sim 0.1$ s (for the strain AW405 in dilute phosphate buffer at 32° C) (Ishihara et al. 1983).

Experimental studies have shown that these rates depend on the CheY-P concentration, say $C$. In particular, Cluzel et al. (2000) have shown that the dependence of CW bias (between the values 0.1 and 0.9) to $C$ can be approximated by a Hill function with a coefficient $H \sim 10.3 \pm 1.1$, with a dissociation constant $K_d = 3.1$ mM/s. Thus, the CW bias reads:

$$p_{CW} = \frac{C^H}{K_d^H + C^H}.$$ (A.4)
Based on the above findings, the transition rates \( k^+ \), \( k^- \) are given by Setayeshgar et al. (2005):

\[
\begin{align*}
    k^+ &= \frac{HC^{H-1}}{K_d^H + C^H}, \\
    k^- &= \frac{1}{C} \frac{HK_d^H}{K_H^{d+} + C^H}.
\end{align*}
\]

Thus, based on the model formulation and nominal values of the parameters, the expected fraction of time spent in the CCW state in the absence of stimulus for each cell from kMC simulations is \( \sim 0.855 \), close enough to the one observed experimentally.

Thus, based on the model formulation and nominal values of the parameters, the expected fraction of time spent in the CCW state in the absence of stimulus for each cell from kMC simulations is \( \sim 0.855 \), close enough to the one observed experimentally.

In the absence of spatial variations in the chemoattractant (or repellent) profile, the rotation of the flagellar filament is biased towards the CCW direction (that is, the probability of CCW rotation of a flagellum is higher than that of CW rotation), when viewed along the helix axis towards the point of insertion in the cell (Larsen et al. 1974). This bias depends on the type of bacterial strain and the temperature; for the wild-type strain AW405, it has been found that the average value of the CCW bias is 0.64 at 32°C (Larsen et al. 1974; Block et al. 1982). When the majority of the flagellar filaments rotate CCW (CW) the cell swims (tumbles).

### Algorithm 1: Monte Carlo Model

**Initialize** Set positions \( x(t = 0) \), speed \( \bar{v}(t = 0) \) for all bacteria, the rotation direction of all flagellae, the states of the cartoon model \( u_1(t = 0), u_2(t = 0) \) and the base sampling time \( dt \).

**Update** positions \( x(t + dt) \) of the bacteria at the next time step \( (t + dt) \) as follows.

For each one of the 6 flagellae, compute the probabilities of switching the direction of the rotation, i.e. \( k_+ \cdot dt \) if the flagellum rotates CCW and \( k_- \cdot dt \) if the flagellum rotates CW. If \( k_+ \cdot dt > 1 \) or \( k_- \cdot dt > 1 \) halve the base sampling time.

For each one of the flagellae of each bacterium compare \( r \) with \( k_\pm \cdot dt \).

- if \( r > k_\pm \cdot dt \) then keep the same rotating direction
- else change rotating direction

if the number of flagella that rotate CW > 3 then tumble

else
    - if flagella previously was running then continue to run to the same direction
    - else start running to the left (\( dir = -1 \)) or to the right (\( dir = +1 \)) with equal probability

\[
\begin{align*}
    x(t + dt) &= x(t) + \bar{v} \cdot dt \\
    u_1(t + dt) &= u_1(t) + dt \cdot \frac{f(s) - u_1(t) - u_2(t)}{t_\sigma} \\
    u_2(t + dt) &= u_2(t) + dt \cdot \frac{f(s) - u_2(t)}{t_\sigma}
\end{align*}
\]
Fig. 5  The average square displacement (over all 1000 cells) as a function of time in the absence of stimulus (gradient of chemoattractant), when all cells are initialized in the tumbling phase with $u_1(0) = 0$, i.e. without excitation and fully adapted, i.e. $u_2(0) = f(s)$, and $\bar{v} = 0.003 \text{ cm/s}$.

Appendix B: Determination of the parameters of the macroscopic PDE for bacterial density evolution

Appendix B.1: Determination of the diffusion coefficient

An estimation of the diffusion coefficient for cell motility in the absence of stimulus can be attempted following two paths. From a microscopic point of view, considering a random walk simulation, the mean free path i.e., the swimming distance without any change in the direction is given by $\delta r = \tau \cdot \bar{v}$, where $\tau$ is the mean time of swimming in one direction. Considering $n$ such time steps in time $t$ (i.e. $n = t/\tau$), the total mean-squared displacement $\Delta r(t)^2$ at a certain time ($t$) is given by the Einstein relation (Berg and Turner 1990):

$$\langle \Delta r^2(t) \rangle = 2D_m t \approx 2n\delta r^2 = 2\tau \bar{v}^2 t,$$

which is valid for $t >> \tau$, where, $\tau$ is the characteristic time scale. Here, the value of $D_m$ is estimated from our Monte Carlo simulations in the absence of stimulus (we have set $s(x) = 1, \forall x$), by tracking the trajectories of 1000 cells for a time period of 2000 s. The cells are initially positioned at the middle of the domain, all initialized at the tumbling phase, with $u_1(0) = 0$ (no excitation), and adapted with $u_2(0) = f(s(x))$, with a constant velocity of $\bar{v} = 0.003 \text{ cm/s}$ (as in Berg and Turner 1990). Figure 5 depicts the average of the square distance as a function of time. By least-squares, we get $\hat{D}_m \approx 9 \cdot 10^{-6} \text{ cm}^2/\text{s}$. This is in good agreement with experimental observations for the E. coli motility (see Berg and Brown 1972; Berg and Turner 1990; Spiro et al. 1997; Cluzel et al. 2000).

From a macroscopic point of view, one can estimate the diffusion coefficient $D_M$ from a linear curve fitting between $\frac{\partial b}{\partial t}$ and $\frac{\partial^2 b}{\partial x^2}$ with finite difference approximations of
temporal and spatial derivatives at the coarse-scale. Thus, by fixing a spatial gradient of chemo-nutrient profile to zero ($\nabla c = 0$), we can consider a simple diffusion equation with a constant diffusion coefficient, $D$:

$$\frac{\partial b}{\partial t} = D \nabla^2 b.$$  \hfill (B.2)

Finally, we note that the Einstein relation for the diffusion coefficient given by Eq. (B.1) can be approximated on average over a run and tumble period as:

$$\langle \Delta r^2 \rangle \approx \bar{v}^2 \bar{T}_{\text{run}}^2 = 2\bar{D}_m (\bar{T}_{\text{run}} + \bar{T}_{\text{tumb}}),$$  \hfill (B.3)

where $\bar{T}_{\text{run}}$, $\bar{T}_{\text{tumb}}$ denote the average duration of swimming and tumbling periods, respectively, and $\bar{v}$ is the average swimming speed. Thus, based on Eq. (B.3) and assuming that the tumbling duration is negligible compared to the swimming duration (as assumed for the derivation of the generalized Keller–Segel theory embodied in Eq. (3)), an approximation of the diffusion coefficient is given by:

$$\bar{D}_m = \frac{\bar{v}^2}{2\lambda_0}, \quad \lambda_0 = \bar{T}_{\text{run}}^{-1}. \hfill (B.4)$$

Hence, setting $\bar{D}_m = \hat{D}_m \approx 9 \cdot 10^{-6} \text{ cm}^2/\text{s}$, $\lambda_0 = 1 \text{ s}^{-1}$ (in agreement with experimental findings), we get $\bar{v} = \sqrt{2}v = 3\sqrt{2} \cdot 9 \cdot 10^{-3} \text{ cm/s}$ as the average velocity appearing in Eq. (3).

**Appendix B.2: Determination of the parameter $c$ of the macroscopic PDE**

As stated in Sect. 2, one of the assumptions for the derivation of the closed-form Keller–Segel Eq. (3) is the linear relation between the turning frequency $\lambda$, and the basal frequency $\lambda_0 \sim 1 \text{ s}^{-1}$, i.e. for each cell at position $x$ at time $t$, we have (see Eq. (5)):

$$\lambda(x, t) = \lambda_0 - cu_1(x, t). \hfill (B.5)$$

For initial values $u_1(x, 0), u_2(x, 0)$ for all cells (i.e. $\forall x \in \mathbb{R}$), the analytical solution of the cartoon model (Eq. 2) is given by:

$$u_1(x, t) = \frac{e^{-\frac{t}{\tau_e}}(K_s^2 \tau_a - K_s^2 \tau_e + s^2 \tau_a - s^2 \tau_e + 2 K_s s \tau_a - 2 K_s s \tau_e)}{(K_s + s)^2 (\tau_a - \tau_e)} u_1(0, x) + \frac{e^{-\frac{t}{\tau_a}}(s^2 \tau_a u_2 - k s \tau_a + K_s^2 \tau_a u_2(0, x) + 2 K_s s \tau_a u_2(0, x)) + k s \tau_a}{(K_s + s)^2 (\tau_a - \tau_e)} - \frac{\tau_a e^{-\frac{t}{\tau_a}}(K_s^2 u_2(0, x) - k s + s^2 u_2(0, x) + 2 K_s s u_2)}{(K_s + s)^2 (\tau_a - \tau_e)}, \hfill (B.6)$$

$$u_2(x, t) = \frac{e^{-\frac{t}{\tau_a}}(K_s^2 u_2(0, x) - k s + s^2 u_2)}{(K_s + s)^2} - \frac{ks(e^{-\frac{t}{\tau_a}} - 1)}{(K_s + s)^2}. \hfill (B.7)$$
Note that, if one sets as initial value \( u_2(x, 0) = f(s) \), then the second equation of the cartoon model (see Eq. (2)) gives \( u_2(x, t) = f(s), \forall t \) and the analytical solution for \( u_1(x, t) \) is reduced to:
\[
  u_1(x, t) = u_1(x, 0)e^{-t/\tau_a}.
\]  
(B.8)

To this end, the parameter \( c \) in Eq. (5) appearing in the Keller–Segel-class PDE given by Eq. (3) can be found with the aid of Monte Carlo simulations, by fixing \( u_1(x, t), \forall t \) to different relatively small values, say \( u_1 \forall x \), and measuring, the number of turning events \( \lambda(u_1) \); then the value of the parameter \( c \) can be estimated by least-squares. A different way would be to set an initial value for \( u_1(x, 0) \) (setting also as initial value \( u_2(x, 0) = f(s) \)), run the Monte Carlo simulator, measure the turning frequencies \( \lambda(u_1(t)) \) and based on the above, the value of \( c \) can be again estimated with least-squares.

Here, to estimate \( c \), we have fixed \( u_1 \) to the following values: \(-0.02, -0.015, -0.01, -0.005, 0, 0.005, 0.01, 0.015, 0.02\), where the linear relation between \( \lambda \) and \( \lambda_0 \) is valid, and we computed \( \lambda(u_1) \) based on Monte Carlo simulations with 1000 cells for a time period of 2000s. For these values, \( \lambda_0^* \sim 1s^{-1} \) (in a good agreement with the experimental findings) and \( \hat{c} \sim 19.5 \) (99% CI 18–21). We note that this value is consistent with what has been reported in other studies (Xue 2015). For our simulations with the macroscopic PDE, we have set \( \hat{c} = 20 \).

Appendix C: Numerical details

See the Tables 2, 3 and 4.

Table 2 Relative % errors, for the reproduction case (corresponding to Fig. 3)

| ML Algorithm | Mean % relative error | Maximum % relative error |
|--------------|----------------------|-------------------------|
|              | Black-box | Gray-box | Correction | Black-box | Gray-box | Correction |
| GP           | 0.33854893 | 0.47118269 | 0.41778762 | 13.55415189 | 8.62675113 | 10.09110753 |
| GP-reduced   | 0.47537462 | 0.45665887 | 0.47132172 | 14.38729724 | 11.02033208 | 7.68339349  |
| NN           | 0.2502552  | 1.00663959 | 0.55650038 | 3.07947682  | 9.96246123  | 5.3239486   |

Table 3 Relative % error, for the testing case (corresponding to Fig. 4)

| ML Algorithm | Mean % relative error | Maximum % relative error |
|--------------|----------------------|-------------------------|
|              | Black-box | Gray-box | Correction | Black-box | Gray-box | Correction |
| GP           | 1.92548545 | 1.21342122 | 0.3875892 | 20.08370186 | 11.9252548 | 5.62689763  |
| GP-reduced   | 1.30973397 | 1.76352319 | 0.453134  | 20.07753367 | 18.99650573 | 3.69253464  |
| NN           | 0.85932661 | 1.19168495 | 0.57287602 | 2.05789847  | 9.49309879  | 3.59098562  |
Table 4  ARD weights

| Feature | Black-box | Gray-box | Feature | Correction |
|---------|-----------|----------|---------|------------|
| $b$     | 1.04E+00  | 1.47E+00 | $CH_b$  | 3.78E+03   |
| $b_x$   | 2.24E+00  | 2.05E+00 | $CH_{b_x}$ | 1.60E+01  |
| $b_{xx}$| 1.80E+01  | 5.23E+00 | $CH_{b_{xx}}$ | 6.16E+00 |
| $s$     | 1.33E+00  | 1.13E+00 | $CH_s$  | 1.07E+01   |
| $s_x$   | 6.25E−01  | 3.06E−01 | $CH_{s_x}$ | 2.04E+02   |
| $s_{xx}$| 1.72E+03  | 2.85E+03 | $CH_{s_{xx}}$ | 5.65E+00 |

The cut-off is 3 orders of magnitude

References

Abadi M, Agarwal A, Barham P, Brevdo E, Chen Z, Citro C, Corrado GS, Davis A, Dean J, Devin M, Ghemawat S, Goodfellow I, Harp A, Irving G, Isard M, Jia Y, Jozefowicz R, Kaiser L, Kudlur M, Levenberg J, Mané D, Monga R, Moore S, Murray D, Olah C, Schuster M, Shlens J, Steiner B, Sutskever I, Talwar K, Tucker P, Vanhoucke V, Vasudevan V, Viégas F, Vinyals O, Warden P, Wattenberg M, Wicke M, Yu Y, Zheng X (2015) TensorFlow: large-scale machine learning on heterogeneous systems, software available from tensorflow.org. https://www.tensorflow.org/

Adler J (1969) Chemoreceptors in bacteria. Science 166(3913):1588–1597

Alexandridis A, Siettos C, Sarimveis H, Boudouvis A, Bafas G (2002) Modelling of nonlinear process dynamics using Kohonen’s neural networks, fuzzy systems and Chebyshev series. Comput Chem Eng 26(4–5):479–486

Ansumali S, Frouzakis CE, Karlin IV, Kevrekidis IG (2005) Exploring Hydrodynamic Closures for the Lid-driven Micro-cavity. arXiv: Statistical Mechanics

Arbabi H, Kevrekidis IG (2021) Particles to partial differential equations parsimoniously. Chaos Interdiscip J Nonlinear Sci 31(3):033137

Beck A, Flad D, Munz C-D (2019) Deep neural networks for data-driven LES closure models. J Comput Phys 398:108910

Bellomo N, Bellouquid A, Nieto J, Soler J (2010) Multiscale biological tissue models and flux-limited chemotaxis for multicellular growing systems. Math Models Methods Appl Sci 20(07):1179–1207

Bellomo N, Outada N, Soler J, Tao Y, Winkler M (2022) Chemotaxis and cross-diffusion models in complex environments: models and analytic problems toward a multiscale vision. Math Models Methods Appl Sci 1–80

Berg HC, Brown DA (1972) Chemotaxis in Escherichia coli analysed by three-dimensional tracking. Nature 239(5374):500–504

Berg HC, Turner L (1990) Chemotaxis of bacteria in glass capillary arrays, Escherichia coli, motility, microchannel plate, and light scattering. Biophys J 58(4):919–930

Bertalan T, Dietrich F, Mezić I, Kevrekidis IG (2019) On learning Hamiltonian systems from data. Chaos Interdiscip J Nonlinear Sci 29(12):121107

Block SM, Segall JE, Berg HC (1982) Impulse responses in bacterial chemotaxis. Cell 31(1):215–226

Block SM, Segall JE, Berg HC (1983) Adaptation kinetics in bacterial chemotaxis. J Bacteriol 154(1):312–323

Bowman AW, Azzalini A (1997) Applied smoothing techniques for data analysis: the kernel approach with S-Plus illustrations, vol 18. OUP, Oxford

Boyd A, Krikos A, Simon M (1981) Sensory transducers of E. coli are encoded by homologous genes. Cell 26(3):333–343

Brunton SL, Proctor JL, Kutz JN (2016) Discovering governing equations from data by sparse identification of nonlinear dynamical systems. Proc Natl Acad Sci 113(15):3932–3937

Chavanis P-H (2008) Nonlinear mean field Fokker–Planck equations. Application to the chemotaxis of biological populations. Eur Phys J B 62(2):179–208

Chen RTQ, Rubanova Y, Bettencourt J, Duvenaud D (2019) Neural ordinary differential equations. arXiv:1806.07366
Chen T, Chen H (1995) Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems. IEEE Trans Neural Netw 6(4):911–917
Chen Y, Hosseini B, Owhadi H, Stuart AM (2021) Solving and learning nonlinear PDEs with Gaussian processes. J Comput Phys 447:110668
Chen Z, Churchill V, Wu K, Xiu D (2022) Deep neural network modeling of unknown partial differential equations in nodal space. J Comput Phys 449:110782
Cluzel P, Surette M, Leibler S (2000) An ultrasensitive bacterial motor revealed by monitoring signaling proteins in single cells. Science 287(5458):1652–1655
Coburn L, Ceroni L, Torney C, Couzin ID, Neufeld Z (2013) Tactile interactions lead to coherent motion and enhanced chemotaxis of migrating cells. Phys Biol 10(4):046002
Dormand J, Prince P (1980) A family of embedded Runge–Kutta formulae. J Comput Appl Math 6(1):19–26
Dsilva CJ, Talmon R, Coifman RR, Kevrekidis IG (2018) Parsimonious representation of nonlinear dynamical systems through manifold learning: a chemotaxis case study. Appl Comput Harmon Anal 44(3):759–773. https://doi.org/10.1016/j.acha.2015.06.008
Duraisamy K, Iaccarino G, Xiao H (2019) Turbulence modeling in the age of data. Annu Rev Fluid Mech 51(1):357–377
Emonet T, Macal CM, North MJ, Wickersham CE, Cluzel P (2005) Agentcell: a digital single-cell assay for bacterial chemotaxis. Bioinformatics 21(11):2714–2721
Erban R, Othmer HG (2004) From individual to collective behavior in bacterial chemotaxis. SIAM J Appl Math 65(2):361–391
Erban R, Othmer HG (2007) Taxis equations for amoeboid cells. J Math Biol 54(6):847–885. https://doi.org/10.1007/s00285-007-0070-1
Erban R, Kevrekidis IG, Othmer HG (2006) An equation-free computational approach for extracting population-level behavior from individual-based models of biological dispersal. Physica D 215(1):1–24
Erban R, Frewen TA, Wang X, Elston TC, Coifman R, Nadler B, Kevrekidis IG (2007) Variable-free exploration of stochastic models: a gene regulatory network example. J Chem Phys 126(15):04B618
Franz B, Erban R (2013) Hybrid modelling of individual movement and collective behaviour. In: Dispersal, individual movement and spatial ecology. Springer, pp 129–157
Galaris E, Fabiani G, Gallos I, Kevrekidis I, Siettos C (2022) Numerical bifurcation analysis of PDEs from lattice Boltzmann model simulations: a parsimonious machine learning approach. J Sci Comput 92(2):34
Gonzalez-Garcia R, Rico-Martinez R, Kevrekidis I (1998) Identification of distributed parameter systems: a neural net based approach. Comput Chem Eng 22:S965–S968
Gorban AN, Kevrekidis IG, Theodoropoulos C, Kazantzis NK, Öttinger HC (Eds.) (2006) Model reduction and coarse-graining approaches for multiscale phenomena. Springer, Berlin https://doi.org/10.1007/3-540-35888-9
Heit B, Tavener S, Raharjo E, Kubes P (2002) An intracellular signaling hierarchy determines direction of migration in opposing chemotactic gradients. J Cell Biol 159(1):91–102
Ho KKY, Srivastava S, Kinnunen PC, Garikipati K, Luker GD, Luker KE (2023) Oscillatory ERK signaling and morphology determine heterogeneity of breast cancer cell chemotaxis via MEK-ERK and p38-MAPK signaling pathways. Bioengineering 10(2). https://doi.org/10.3390/bioengineering10020269
Ishihara A, Segall JE, Block SM, Berg HC (1983) Coordination of flagella on filamentous cells of Escherichia coli. J Bacteriol 155(1):228–237
Iskhakov AS, Dinh NT, Chen E (2021) Integration of numerical solutions of PDEs for closure models development. Phys Lett A 406:127456
Jiang Y, Kolehmainen J, Gu Y, Kevrekidis YG, Ozel A, Sundaresan S (2019) Neural-network-based filtered drag model for gas-particle flows. Powder Technol 346:403–413
Kamath A, Vargas-Hernández RA, Krems RV, Carrington T, Manzhos S (2018) Neural networks vs Gaussian process regression for representing potential energy surfaces: a comparative study of fit quality and vibrational spectrum accuracy. J Chem Phys 148(24):241702. https://doi.org/10.1063/1.5003074
Karniadakis GE, Kevrekidis IG, Lu L, Perdikaris P, Wang S, Yang L (2021) Physics-informed machine learning. Nat Rev Phys 3(6):422–440
Keller EF, Segel LA (1971) Model for chemotaxis. J Theor Biol 30(2):225–234
Kemeth FP, Bertalan T, Thiem T, Dietrich F, Moon SJ, Laing CR, Kevrekidis IG (2022) Learning emergent partial differential equations in a learned emergent space. Nat Commun 13(1):3318
Kim I, Yao Y (2012) The Patlak–Keller–Segel model and its variations: properties of solutions via maximum principle. SIAM J Math Anal 44(2):568–602
Kingma DP, Ba J (2014) Adam: a method for stochastic optimization. https://doi.org/10.48550/ARXIV.1412.6980
Kocijan J, Girard A, Banko B, Murray-Smith R (2005) Dynamic systems identification with Gaussian processes. Math Comput Model Dyn Syst 11(4):411–424
Krischer K, Rico-Martinez R, Kevrekidis I, Rotermund H, Erlt G, Hudson J (1993) Model identification of a spatiotemporally varying catalytic reaction. AIChE J 39(1):89–98
Larsen SH, Reader RW, Kort EN, Tso W-W, Adler J (1974) Change in direction of flagellar rotation is the basis of the chemotactic response in Escherichia coli. Nature 249(5452):74–77
LeCun Y, Bengio Y (1998) Convolutional Networks for Images, Speech, and Time Series. MIT Press, Cambridge, pp 255–258
Lee S, Dietrich F, Karniadakis GE, Kevrekidis IG (2019) Linking Gaussian process regression with data-driven manifold embeddings for nonlinear data fusion. Interface Focus 9(3):20180083
Lee S, Kooshkbaghi M, Spiliotis K, Siettos CI, Kevrekidis IG (2020) Coarse-scale PDEs from fine-scale observations via machine learning. Chaos Interdiscip J Nonlinear Sci 30(1):013141
Lee K, Hernández AM, Stewart DS, Lee S (2021) Data-driven blended equations of state for condensed-phase explosives. Combust Theory Modell 1–23
Li J, Kevrekidis PG, Gear CW, Kevrekidis IG (2003) Deciding the nature of the coarse equation through microscopic simulations: the baby-bathwater scheme. Multiscale Model Simul 1(3):391–407
Liu J, Parkinson JS (1989) Role of chew protein in coupling membrane receptors to the intracellular signaling system of bacterial chemotaxis. Proc Natl Acad Sci 86(22):8703–8707
Liu K, Li Y, Hu X, Lucu M, Widanage WD (2019) Gaussian process regression with automatic relevance determination kernel for calendar aging prediction of lithium-ion batteries. IEEE Trans Industr Inf 16(6):3767–3777
MacKay DJ (1992) Bayesian interpolation. Neural Comput 4(3):415–447
Maeda K, Imae Y, Shioi J-I, Oosawa F (1976) Effect of temperature on motility and chemotaxis of Escherichia coli. J Bacteriol 127(3):1039–1046
Masri SF, Chassiakos AG, Caughey TK (1993) Identification of nonlinear dynamic systems using neural networks. J Appl Mech 60(1):123–133
Nash J (1966) Analyticity of the solutions of implicit function problems with analytic data. Ann Math 84(3):345–355
Othmer HG, Schaar P (1998) Oscillatory camp signaling in the development of Dictyostelium discoideum. Comments Theor Biol 5:175–282
Othmer HG, Xin X, Xue C (2013) Excitation and adaptation in bacteria-a model signal transduction system that controls taxis and spatial pattern formation. Int J Mol Sci 14(5):9205–9248
Painter KJ (2019) Mathematical models for chemotaxis and their applications in self-organisation phenomena. J Theor Biol 481:162–182
Parish EJ, Duraisamy K (2018) Data-driven discovery of closure models. SIAM J Appl Dyn Syst 17(4):2381–2413
Pathak J, Mustafa M, Damianou A, Lawrence ND, Karniadakis GE (2017) Nonlinear information fusion algorithms for data-efficient multi-fidelity modelling. Proc R Soc A Math Phys Eng Sci 473(2198):20160751
Psarellis YM, Lee S, Bhattacharjee T, Datta SS, Bello-Rivas JM, Kevrekidis IG (2022) Data-driven discovery of chemotactic migration of bacteria via machine learning. https://doi.org/10.48550/ARXIV.2208.11853
Qin T, Wu K, Xiu D (2019) Data driven governing equations approximation using deep neural networks. J Comput Phys 395:620–635
Raissi M, Karniadakis GE (2018) Hidden physics models: machine learning of nonlinear partial differential equations. J Comput Phys 357:125–141
Raissi M, Perdikaris P, Karniadakis GE (2017) Machine learning of linear differential equations using Gaussian processes. J Comput Phys 348:683–693
Raissi M, Perdikaris P, Karniadakis GE (2019) Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. J Comput Phys 378:686–707
Rao C, Ren P, Liu Y, Sun H (2022) Discovering nonlinear PDEs from scarce data with physics-encoded learning. arXiv:2201.12354
Rasmussen CE, Williams CKI (2005) Gaussian processes for machine learning (adaptive computation and machine learning). The MIT Press
Rasmussen CE, Williams CKI (2006) Gaussian processes for machine learning. MIT Press
Rico-Martinez R, Krischer K, Kevrekidis I, Kube M, Hudson J (1992) Discrete-time nonlinear signal processing of cu electrodissolution data. Chem Eng Commun 118(1):25–48
Rico-Martinez R, Anderson J, Kevrekidis I (1994) Continuous-time nonlinear signal processing: a neural network based approach for gray box identification. In: Proceedings of IEEE workshop on neural networks for signal processing. IEEE, pp 596–605
Rousset M, Samaey G (2013) Simulating individual-based models of bacterial chemotaxis with asymptotic variance reduction. Math Models Methods Appl Sci 23(12):2155–2191
Sandhu R, Pettit C, Khalil M, Poirol D, Sarkar A (2017) Bayesian model selection using automatic relevance determination for nonlinear dynamical systems. Comput Methods Appl Mech Eng 320:237–260
Sarkar MK, Paul K, Blair D (2010) Chemotaxis signaling protein CheY binds to the rotor protein FliN to control the direction of flagellar rotation in *Escherichia coli*. Proc Natl Acad Sci 107(20):9370–9375
Scharf BE, Fahrner KA, Turner L, Berg HC (1998) Control of direction of flagellar rotation in bacterial chemotaxis. Proc Natl Acad Sci 95(1):201–206
Segel LA, Goldbeter A, Devreotes PN, Knox BE (1986) A mechanism for exact sensory adaptation based on receptor modification. J Theor Biol 120(2):151–179
Setayeshgar S, Gear CW, Othmer HG, Kevrekidis IG (2005) Application of coarse integration to bacterial chemotaxis. Multiscale Model Simul 4(1):307–327
Sherifdeen S, Ragusa JC, Morel JE, Adams ML, Bui-Thanh T (2019) Accelerating PDE-constrained inverse solutions with deep learning and reduced order models. arXiv:1912.08864
Siettos C (2014) Coarse-grained computational stability analysis and acceleration of the collective dynamics of a Monte Carlo simulation of bacterial locomotion. Appl Math Comput 232:836–847
Siettos CI, Bafas GV (2002) Semiglobal stabilization of nonlinear systems using fuzzy control and singular perturbation methods. Fuzzy Sets Syst 129(3):275–294
Siettos CI, Bafas GV, Boudouvis AG (2002) Truncated Chebyshev series approximation of fuzzy systems for control and nonlinear system identification. Fuzzy Sets Syst 126(1):89–104
Spiro PA, Parkinson JS, Othmer HG (1997) A model of excitation and adaptation in bacterial chemotaxis. Proc Natl Acad Sci 94(14):7263–7268
Takens F (1981) Detecting strange attractors in turbulence. In: Rand D, Young L-S (eds) Dynamical Systems and Turbulence, Warwick 1980. Springer, Berlin, pp 366–381
Thiem TN, Kemeth FP, Bertalan T, Laing CR, Kevrekidis IG (2021) Global and local reduced models for interacting, heterogeneous agents. Chaos Interdiscip J Nonlinear Sci 31(7):073139
Tindall MJ, Porter S, Maini P, Gaglia G, Armitage JP (2008) Overview of mathematical approaches used to model bacterial chemotaxis I: the single cell. Bull Math Biol 70(6):1525–1569
Tindall MJ, Maini PK, Porter SL, Armitage JP (2008) Overview of mathematical approaches used to model bacterial chemotaxis II: bacterial populations. Bull Math Biol 70(6):1570
Turner L, Caplan SR, Berg HC (1996) Temperature-induced switching of the bacterial flagellar motor. Biophys J 71(4):2227–2233
Vlachas PR, Byeon W, Han ZY, Sapsis TP, Koumoutsakos P (2018) Data-driven forecasting of high-dimensional chaotic systems with long short-term memory networks. Proc R Soc A Math Phys Eng Sci 474(2213):20170844
Vlachas PR, Pathak J, Hunt BR, Sapsis TP, Girvan M, Ott E, Koumoutsakos P (2020) Backpropagation algorithms and reservoir computing in recurrent neural networks for the forecasting of complex spatiotemporal dynamics. Neural Netw 126:191–217
Vlachas PR, Arampatzis G, Uhler C, Koumoutsakos P (2022) Multiscale simulations of complex systems by learning their effective dynamics. Nat Mach Intell 4(4):359–366
Wan ZY, Sapsis TP (2017) Reduced-space Gaussian process regression for data-driven probabilistic forecast of chaotic dynamical systems. Physica D 345:40–55
Whitney H (1936) Differentiable manifolds. Ann Math 37(3):645–680
Wu K, Xiu D (2019) Numerical aspects for approximating governing equations using data. J Comput Phys 384:200–221
Wu K, Xiu D (2020) Data-driven deep learning of partial differential equations in modal space. J Comput Phys 408:109307
Wu M, Roberts JW, Kim S, Koch DL, DeLisa MP (2006) Collective bacterial dynamics revealed using a three-dimensional population-scale defocused particle tracking technique. Appl Environ Microbiol 72(7):4987–4994
Xue C (2015) Macroscopic equations for bacterial chemotaxis: integration of detailed biochemistry of cell signaling. J Math Biol 70(1):1–44
Yang L, Meng X, Karniadakis GE (2021) B-PINNs: Bayesian physics-informed neural networks for forward and inverse PDE problems with noisy data. J Comput Phys 425:109913
Yasuda S (2017) Monte Carlo simulation for kinetic chemotaxis model: an application to the traveling population wave. J Comput Phys 330:1022–1042
Zhang ZJ, Duraisamy K (2015) ‘Machine learning methods for data-driven turbulence modeling’. 22nd AIAA computational fluid dynamics conference, american institute of aeronautics and astronautics. https://doi.org/10.2514/6.2015-2460. AIAA AVIATION Forum

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.