INTERFACE

royalsocietypublishing.org/journal/rsif

Review

Cite this article: Nardini JT, Baker RE, Simpson MJ, Flores KB. 2021 Learning differential equation models from stochastic agent-based model simulations. *J. R. Soc. Interface* **18**: 20200987.
https://doi.org/10.1098/rsif.2020.0987

Received: 4 December 2020
Accepted: 22 February 2021

Subject Category:
Reviews

Subject Areas:
biomathematics, computational biology, systems biology

Keywords:
agent-based models, differential equations, equation learning, population dynamics, disease dynamics

Author for correspondence:
John T. Nardini
e-mail: jtnardin@ncsu.edu

Learning differential equation models from stochastic agent-based model simulations

John T. Nardini[1], Ruth E. Baker[2], Matthew J. Simpson[3] and Kevin B. Flores[1]

[1]North Carolina State University, Mathematics, Raleigh, NC, USA
[2]Mathematical Institute, University of Oxford, Oxford, UK
[3]School of Mathematical Sciences, Queensland University of Technology, Brisbane 4001, Australia

JTN, 0000-0002-5503-1934; REB, 0000-0002-6304-9333; MJS, 0000-0001-6254-313X; KBF, 0000-0002-4000-6971

Agent-based models provide a flexible framework that is frequently used for modelling many biological systems, including cell migration, molecular dynamics, ecology and epidemiology. Analysis of the model dynamics can be challenging due to their inherent stochasticity and heavy computational requirements. Common approaches to the analysis of agent-based models include extensive Monte Carlo simulation of the model or the derivation of coarse-grained differential equation models to predict the expected or averaged output from the agent-based model. Both of these approaches have limitations, however, as extensive computation of complex agent-based models may be infeasible, and coarse-grained differential equation models can fail to accurately describe model dynamics in certain parameter regimes. We propose that methods from the equation learning field provide a promising, novel and unifying approach for agent-based model analysis. Equation learning is a recent field of research from data science that aims to infer differential equation models directly from data. We use this tutorial to review how methods from equation learning can be used to learn differential equation models from agent-based model simulations. We demonstrate that this framework is easy to use, requires few model simulations, and accurately predicts model dynamics in parameter regions where coarse-grained differential equation models fail to do so. We highlight these advantages through several case studies involving two agent-based models that are broadly applicable to biological phenomena: a birth–death–migration model commonly used to explore cell biology experiments and a susceptible–infected–recovered model of infectious disease spread.

1. Introduction

Complex interactions between individuals are a crucial aspect of many biological processes: honeybees dance to direct others to food sources [1], cells push their neighbours to promote invasion during tumorigenesis [2] and animal herds aggregate together to deter predation [3]. Agent-based models (ABMs) are invaluable tools to simulate how such interactions between individuals scale to population-wide phenomena [4]. In an ABM, the states and decisions of individual agents are simulated using pre-defined rules to govern the agents' interactions and behaviour [5]. The ease of construction of ABMs by domain experts and modellers allows for complex models that can capture rich dynamical behaviour [5,6].

There are many approaches to predicting the emergent behaviour of stochastic ABMs, each of which presents its own advantages and challenges. The most straightforward and commonly used approach to interrogate ABMs is extensive Monte Carlo simulation using well-established computational algorithms [5] (Arrow 1 of figure 1). Average ABM behaviours at fixed parameter values can be inferred from many simulations from the central limit

THE ROYAL SOCIETY PUBLISHING

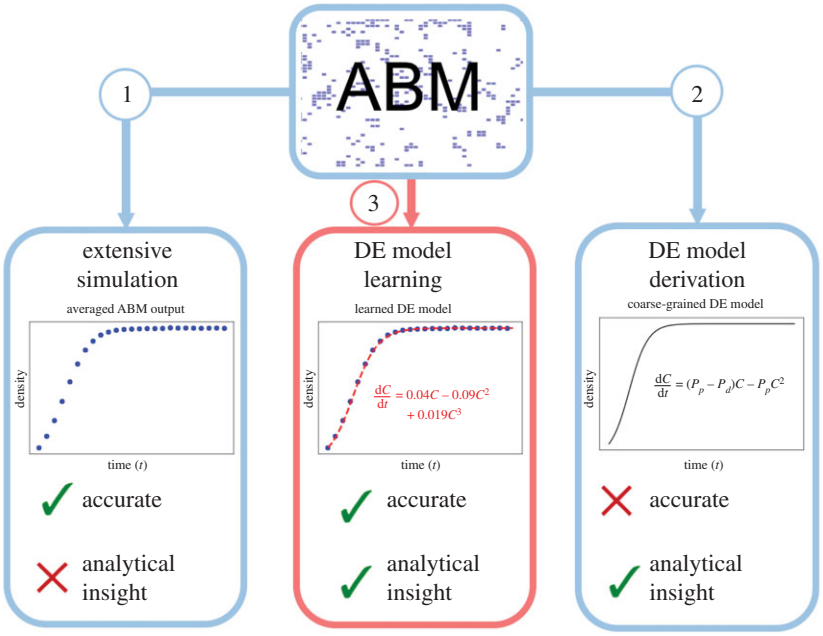

Figure 1. An illustration of current (blue) and proposed (red) methods to predict emergent ABM behaviour. Extensive simulation (Arrow 1) is performed by running many ABM simulations over a range of parameter values, and then using Monte Carlo techniques to average ABM output. While this approach will accurately predict ABM dynamics, it can be computationally intensive to perform. DE models derived using model coarse graining approaches (Arrow 2) can be analysed (e.g. using bifurcation analysis or perturbation methods). This technique is advantageous because such analytical methods do not require any computation. Unfortunately, coarse-grained models will provide inaccurate predictions in many parameter regimes. We propose that DE models can be learned from ABM simulation data using techniques from equation learning (Arrow 3). This method is advantageous because it may only require a small number of ABM simulations, will lead to a DE model that can predict ABM dynamics accurately, and can be informed with analytical techniques.

theorem [7], and the inverse problem of inferring model parameter distributions can be done with Markov chain Monte Carlo samplers [8]. Unfortunately, such extensive simulation of many ABMs may not be feasible due to significant computational costs involved. An alternative method to predict the emergent behaviour of an ABM consists of deriving differential equation (DE) models to approximate ABM output (Arrow 2 of figure 1). Each ABM has a master equation that can be derived directly from the model rules. There are many approaches to simplify this master equation and approximate its dynamics with more tractable DE models. The most commonly used DE model approximations for ABMs are *mean-field* models [9–12]. Alternative formulations to mean-field models are also possible, see [13] for an extensive tutorial. Mean-field models describe the evolution of population density over time (and possibly space) and can be derived by approximating agent–agent interactions with locally averaged agent densities [12]. Mean-field DE models are often simple to solve (either analytically or numerically), so they provide an advantageous alternative to extensive simulation of the ABM. Furthermore, such DE models are amenable to analytical techniques (including bifurcation, travelling wave, perturbation analysis), which can be used to predict how ABM output will change in response to variations in parameter values [14,15].

Previous ecological studies have demonstrated some of the advantages of both extensive simulation and model coarse graining for ABM analysis [16]. For example, Bernoff *et al.* [17] model the foraging behaviour of the Australian plaque locust with a discrete and stochastic ABM. In the model, individual locusts forage and feed on a given resource (representative of food) and, in turn, create a spatial gradient of this resource. The model robustly shows that individual locust behaviour drives the formation of this resource gradient and, in turn, determines how the averaged profile of locust density migrates and forms over time. The authors derive the mean-field partial differential equation (PDE) model for this ABM and perform a travelling wave analysis to quantify how the locust population's invasion speed depends on the total mass of locusts. The mean-field PDE model is shown to match the ABM output well in biologically consistent parameter regimes. In addition, non-mean-field models have been considered to approximate other ABMs of locust behaviour. For example, the ABMs in [18] describe self-organizing locust behaviours through rules governing locust attraction, repulsion, and alignment during foraging and invasion. By simulating the ABM over many different parameter values, Dkhili *et al.* [18] discovered three distinct population patterns (spot, band and ribbon formations). Topaz *et al.* [19] analysed a continuous partial integro-differential equation as a representation of this locust flocking behaviour and used a linear stability analysis to provide analytical insights into which parameter values governing agent interactions lead to the formation of such spatial patterns. There are thus many scenarios in which DE models supplement ABM simulations to aid in our understanding of emergent behaviour.

Despite their wide use, coarse-grained models can provide misleading predictions of ABM dynamics in regions of parameter space in which the assumptions made during the coarse graining process do not hold [9,12,20]. Furthermore, it can be challenging to determine informative DE models for more complex ABMs. As one such example, Gallaher *et al.* [21] constructed an ABM in which thousands of cells with different phenotypes compete for space during tumour growth. Each agent in the simulation is given an

internal set of dynamic traits dictating how fast the agent moves in space and how frequently it divides. The intricate dynamics of this model allow for interesting findings of biological relevance, including how transmission of proliferation rates from parent to daughter cells alters the final trait landscape of the population and, in turn, the eventual physical clustering of the population. Formulating a DE model for this process from ABM rules would be challenging, however, due to the many different cell phenotypes and complicated rules between such cells. Instead, methods to directly infer DE models from a small number of ABM simulations may provide a useful tool for modellers to determine the salient features necessary for modelling complex ABM dynamics. ABMs with evolving trait landscapes are becoming increasingly common to study tumour dynamics, so such learned DE models will be widely applicable to this growing field of research [22].

Equation learning (EQL) is a recent field from data science that aims to infer the dynamical systems model that best describes a given dataset [23]. The learned models can, in turn, be used to understand the system under study by providing a mechanistic description of observed dynamics or predicting how dynamics will change in response to different conditions. There has been much progress in this field over the past 5 years largely thanks to increases in computational power, and many EQL methods can accurately recover DE models from artificially simulated noisy data from DE models [24–27]. There have been some recent studies showing that equations can be learned from noisy experimental data [28], and the EQL field is now in a position where EQL can, in principle, be used to aid in the development of DE models to approximate the dynamics of complex ABMs (Arrow 3 of figure 1). In this review article, we will detail how a commonly used EQL methodology [23] can be used to learn DE models that accurately describe ABM dynamics. In doing so, we also explore how EQL provides insight into ABM behaviours when traditional modelling approaches (e.g. coarse-graining) fail to capture ABM complexity, as well as EQL performance in practical situations, such as those with sparse data samples.

Similar to coarse-grained DE models, learned DE models can be analysed using both computational algorithms and analytical techniques to infer the emergent behaviour that results from a given set of ABM, or estimate mechanistic parameters from data. Furthermore, such learned equations may have fewer computational requirements than ABMs if they can learn ABM dynamics from a small number of simulations. As a result, EQL methods may be tractable for learning DE models for a broad range of ABM simulations. There are many ways in which learned DE models may be useful for ABM analysis, including the discovery of novel DE models, predicting unobserved dynamics from complex ABMs and enabling accurate parameter estimation. Furthermore, as ABMs imitate many key features of biological systems (including stochasticity and heterogeneity), inferring DE models from ABM data is an intermediate step towards developing algorithms to aid the discovery of models from experimental data.

This article is intended to serve as a review and tutorial on three separate but synergistic methods (extensive simulation, coarse-grained model derivation, and EQL) to infer the emergent behaviour of ABMs. The first two are frequently used for ABM analysis, and we propose that analysis of a learned DE model from ABM data provides increased understanding of the mechanisms driving observed behaviour. We will showcase each of these methods and highlight the advantages and limitations of each. In §2, we discuss ABM set-up and implementation as well as how simple ABM rules can be coarse grained directly into DE models. In §3, we discuss how methods from EQL can be used to learn DE models from ABM data directly. Our goal in §4 is to present six questions (Q1–Q6) relating to how EQL can aid in the analysis of ABM behaviour, and we address each question with case study examples. We use two representative ABMs throughout: a birth–death–migration (BDM) model of population dynamics [9], and a susceptible–infected–recovered (SIR) model of infectious disease dynamics. We note, however, that the approaches discussed within this tutorial are broadly applicable to many social and biological phenomena that have been modelled by ABMs previously [5,29–33]. We make final conclusions, summarize the advantages and limitations of each method, and suggest future avenues for research in §5. Python code for all tutorials and case studies shown in this study are publicly available at https://github.com/johnnardini/Learning-DE-models-from-stochastic-ABMs.

2. Coarse graining agent-based models into differential equation models

Coarse-grained DE models are now frequently used to investigate how rules governing individual behaviours translate to emergent behaviour at the population level [9,12,34]. In this section, we illustrate this approach by introducing two simple ABMs and coarse graining them to give DE models. We consider two ABMs: (i) a BDM process in §2.1, and (ii) a SIR model in §2.2. While we focus on ordinary differential equation (ODE) models throughout this article, the derivation of PDE models in the presence of spatial heterogeneity can also be performed using extensions of the methods presented herein, as discussed in [12,35,36].

2.1. A birth, death and migration agent-based model

We consider the BDM process, a lattice-based ABM in which agents are able to give birth, die, and move [9]. This ABM is representative of many biological phenomena, for example agents may represent cells during the wound healing process [35] or the invasion of animals in ecology [16]. We begin by introducing the ABM rules in §2.1.1 and then coarse grain these rules into DE models and compare to ABM output in §2.1.2.

2.1.1. Agent-based model rules

We use a two-dimensional square lattice with a lattice spacing of Δ. We arbitrarily set $\Delta = 1$ and assume the lattice has X lattice sites in each spatial dimension. Each lattice site is indexed by $\alpha = (i, j) \in \mathbb{N}^2$, $i, j = 1, \ldots, X$. For each interior lattice site, α, we define its neighbouring sites, $\mathcal{B}(\alpha)$, using the Von Neumann neighbourhood $\mathcal{B}(\alpha) = \{(i, j+1), (i, j-1), (i+1, j), (i-1, j)\}$ and adjust this definition at boundary sites to enforce no-flux conditions. We designate the occupancy of each lattice site α as $\mu_\alpha(t) = 0$ if α is unoccupied at time t or by $\mu_\alpha(t) = A$ if α is occupied by an agent at time t. For simplicity, we will also use the notation $0_\alpha(t)$ and $A_\alpha(t)$ when α is unoccupied and occupied, respectively.

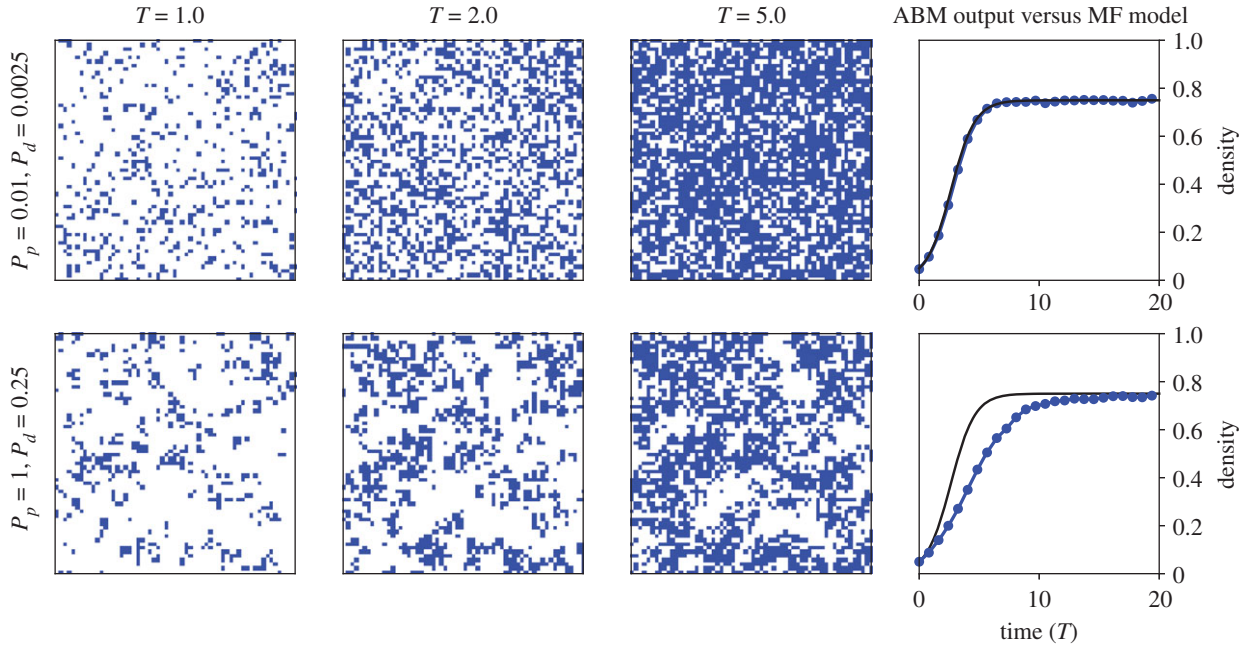

Figure 2. ABM simulation snap shots for the BDM ABM. Blue pixels denote $A_\alpha(t)$ and white pixels denote $0_\alpha(t)$. Simulations were computed with agent migration rate $P_m = 1$. The right-most column depicts the output agent density from one ABM simulation (blue line and dots) against the solution of the logistic equation (2.7) (solid black line). Quantities are plotted against non-dimensionalized time $T = (P_p - P_d)t$ for ease of interpretation. The ABM was computed on a square lattice of length $X = 120$.

To represent volume exclusion, or crowding, we assume that each lattice site can be occupied by a maximum of one agent at a time.

We next define how agents proliferate, migrate and die. For proliferation events, we assume that agents proliferate with rate P_p (formally, an agent will attempt to proliferate over an infinitesimal timestep of duration dt with probability P_pdt). If an agent chooses to proliferate, then it will attempt to place its daughter cell into a neighbouring site $\beta \in \mathcal{B}(\alpha)$ with the choice of β made uniformly at random. If the chosen site is occupied, the event is aborted. This process may be written as a bimolecular reaction with rate $P_p/4$:

$$A_\alpha + 0_\beta \xrightarrow{P_p/4} A_\alpha + A_\beta, \quad \beta \in \mathcal{B}(\alpha). \tag{2.1}$$

The reaction rate here is divided by four because proliferating agents randomly choose one of their four neighbouring sites to place their daughter cell into. For death events, lattice sites transition from occupied to unoccupied without any explicit crowding effects. We assume agents die with rate P_d, and write death events as a monomolecular reaction with rate P_d:

$$A_\alpha \xrightarrow{P_d} 0_\alpha. \tag{2.2}$$

Agents attempt to move with rate P_m. During migration events, agents randomly choose a neighbouring lattice site $\beta \in \mathcal{B}(\alpha)$ to attempt to move to. If the chosen site is already occupied, then the migration event is aborted. This process may be written as a bimolecular reaction with rate $P_m/4$:

$$A_\alpha + 0_\beta \xrightarrow{P_m/4} 0_\alpha + A_\beta, \quad \beta \in \mathcal{B}(\alpha). \tag{2.3}$$

The reaction rate here is divided by four because migrating agents randomly choose one of their four neighbouring sites to place their daughter cell into. Following previous ABM studies, we have chosen to include empty space as an interacting agent in equations (2.1) and (2.2) to incorporate the effects of volume exclusion. This choice converts migration into a bimolecular reaction instead of a monomolecular reaction [9,36–40].

The BDM model can be simulated using the Gillespie algorithm [41], which is provided for the BDM process in algorithm 1 in appendix C. Each ABM simulation is initialized by placing agents uniformly at random throughout the lattice so that 5% of lattice sites are occupied. Note that each ABM simulation begins with a new initial configuration of agent locations throughout the lattice. Reflecting boundary conditions are used at the boundaries of the lattice, which enforces a no-flux condition in the spatial domain. We use the following notation to summarize the output from an ABM simulation. To estimate the total agent density from the n^{th} of N identically prepared ABM simulations we compute

$$C^{(n)}_{\text{ABM}}(t) = \frac{C^{(n)}(t)}{X^2}, \tag{2.4}$$

where $C^{(n)}(t)$ is the number of occupied sites at time t. To estimate the averaged agent density over time from N identically prepared ABM simulations, we compute

$$\langle C_{\text{ABM}}(t) \rangle = \frac{1}{N} \sum_{n=1}^{N} C^{(n)}_{\text{ABM}}(t). \tag{2.5}$$

We depict snapshots of two simulations of the BDM process in figure 2. The blue dots in the right-most column correspond to $C^{(1)}_{\text{ABM}}(t)$ from these individual simulations.

2.1.2. Model coarse graining

ABM rules are often coarse grained into continuous DE models to aid in their analysis. Many previous studies [9,12,42] have demonstrated that the mean-field DE model for the BDM process described in §2.1 is given by the logistic DE model

$$\frac{\mathrm{d}}{\mathrm{d}t} C(t) = P_p C(t)(1 - C(t)) - P_d C(t). \tag{2.6}$$

This model is advantageous in that it can be solved analytically to give

$$C(t) = \frac{KC(0)e^{rt}}{K + C(0)(e^{rt} - 1)}, \tag{2.7}$$

where $r = P_p - P_d$, $K = (P_p - P_d)/P_p$, and $C(0)$ denotes the initial condition. The full derivation of this model is provided in appendix A. To arrive at equation (2.6), we use the *mean-field assumption* that the occupancies of neighbouring lattice sites are independent, i.e. for all sites α, $\mathbb{P}[A_\alpha(t), A_\beta(t)] = \mathbb{P}[A_\alpha(t)]\mathbb{P}[A_\beta(t)]$, $\beta \in \mathcal{B}(\alpha)$, where $\mathbb{P}[A_\alpha(t)]$ is the probability that lattice site α is occupied at time t and $\mathbb{P}[A_\alpha(t), A_\beta(t)]$ is the joint occupancy probability of neighbouring lattice sites α and β at time t.

Mean-field models are widely used to predict ABM dynamics; however, they fail to accurately predict dynamics in regions of parameter space in which the mean-field assumption is violated [9,12,20,43–46]. We depict ABM snapshots for two simulations of the BDM process in figure 2. The mean-field assumption seems to be satisfied during the simulation with $(P_p, P_d) = (0.01, 0.0025)$: agents appear uniformly distributed, which indicates that neighbouring site occupancies are independent of each other. As expected, we observe close agreement between $C(t)$ and $\langle C_{\text{ABM}}(t)\rangle$ for this parameter combination. For the simulation with $(P_p, P_d) = (1, 0.25)$, however, the ABM simulation exhibits strong clustering. In this case, neighbouring site occupancies will be dependent within and outside of cluster regions, which violates the mean-field assumption. As a result, we observe poor agreement between the mean-field model and the ABM simulation output, $\langle C_{\text{ABM}}(t)\rangle$. Similar results have been documented previously for this model over a wide range of parameter values: the mean-field model matches ABM output well for small values of P_p/P_m and P_d/P_m (P_m is fixed at unity in all simulations in this study), but this agreement worsens as either of these ratios increase [9].

2.2. A susceptible–infected–recovered agent-based model

Susceptible, infected and recovered (SIR) models are used in epidemiology to model and predict the emergence of infectious [47] and waterborne [48] diseases. In this section, we detail how the previous modelling framework can be extended to derive DE models for such ABMs of disease spread. We introduce the model rules in §2.2.1 and then derive the mean-field DE model and compare it to ABM output in §2.2.2.

2.2.1. Agent-based model Rules

We use an equivalent lattice to that presented in §2.1. Each lattice site, α, can now take one of four states over time: $0_\alpha(t)$, $S_\alpha(t)$, $I_\alpha(t)$ and $R_\alpha(t)$ denote that α is unoccupied, or occupied by a *susceptible*, *infected* or *recovered* agent at time t, respectively. We assume three rules governing how agents move, infect, and recover in our SIR model. For agent movement, we assume that each agent moves with rate P_m. When an agent attempts to migrate, the agent chooses a neighbouring lattice site $\beta \in \mathcal{B}(\alpha)$ randomly to move to. If the chosen site is already occupied, then the migration event is aborted. We write this process as a bimolecular reaction with rate $P_m/4$:

$$Y_\alpha + 0_\beta \xrightarrow{P_m/4} 0_\alpha + Y_\beta, \quad \beta \in \mathcal{B}(\alpha), \tag{2.8}$$

for $Y \in \{S, I, R\}$.

The second rule governs infection of agents, which occurs with rate P_I. During an infection event, an infected agent at lattice site α will randomly infect an agent at a neighbouring lattice site $\beta \in \mathcal{B}(\alpha)$. If the chosen site β is occupied by a susceptible agent, then the susceptible agent becomes infected. Otherwise, the infection does not alter the state of lattice site β. We model this rule using a bimolecular reaction with rate $P_I/4$:

$$I_\alpha + S_\beta \xrightarrow{P_I/4} I_\alpha + I_\beta, \quad \beta \in \mathcal{B}(\alpha). \tag{2.9}$$

The final rule concerns the recovery of infected agents: infected agent recover with rate P_R. We model this process using a monomolecular reaction with rate P_R:

$$I_\alpha \xrightarrow{P_R} R_\alpha. \tag{2.10}$$

Simulation of the SIR ABM proceeds as follows. We begin each simulation by randomly placing susceptible agents in 49% of the lattice sites, infected agents in 1% of the lattice sites and leaving the remaining lattice sites unoccupied. Note that each ABM simulation begins with a new initial configuration of agent locations throughout the lattice. Because there is no death or birth in the model, the proportion of occupied lattice sites is fixed at $M = 0.5$ for all time. We use reflecting boundary conditions, which model a no-flux condition in the spatial domain. The Gillespie algorithm is used to simulate the model [41]. From the n^{th} of N identically prepared simulations, we let $S^{(n)}_{\text{ABM}}(t)$, $I^{(n)}_{\text{ABM}}(t)$ and $R^{(n)}_{\text{ABM}}(t)$, denote the fractions of susceptible, infected and recovered agents in the model over time, respectively (e.g. $S^{(n)}_{\text{ABM}}(t)$ is equal to the number of susceptible agents at time t divided by MX^2). We then estimate the averaged ABM fraction for each subpopulation by averaging over all N simulations

$$\begin{aligned}\langle S_{\text{ABM}}(t)\rangle &= \frac{1}{N}\sum_{n=1}^{N} S^{(n)}_{\text{ABM}}(t);\\ \langle I_{\text{ABM}}(t)\rangle &= \frac{1}{N}\sum_{n=1}^{N} I^{(n)}_{\text{ABM}}(t);\\ \langle R_{\text{ABM}}(t)\rangle &= \frac{1}{N}\sum_{n=1}^{N} R^{(n)}_{\text{ABM}}(t).\end{aligned} \tag{2.11}$$

We depict snapshots of two simulations of the SIR model in figure 3. In both cases, we observe that the small initial proportion of infected agents causes an outbreak of infection. The majority of agents have become infected and then recovered by the end of both simulations.

2.2.2. Model coarse graining

We show in appendix B that the mean-field model for the SIR process is given by the frequently used system of equations:

$$\frac{dS}{dt} = -MP_I SI; \quad \frac{dI}{dt} = MP_I SI - P_R I; \quad \frac{dR}{dt} = P_R I. \tag{2.12}$$

In equation (2.12), the variables $S(t)$, $I(t)$, $R(t)$ denote the fraction of susceptible, infected and recovered agents at time t, respectively. In figure 3, we depict snapshots from two simulations of the SIR ABM together with evolution of the

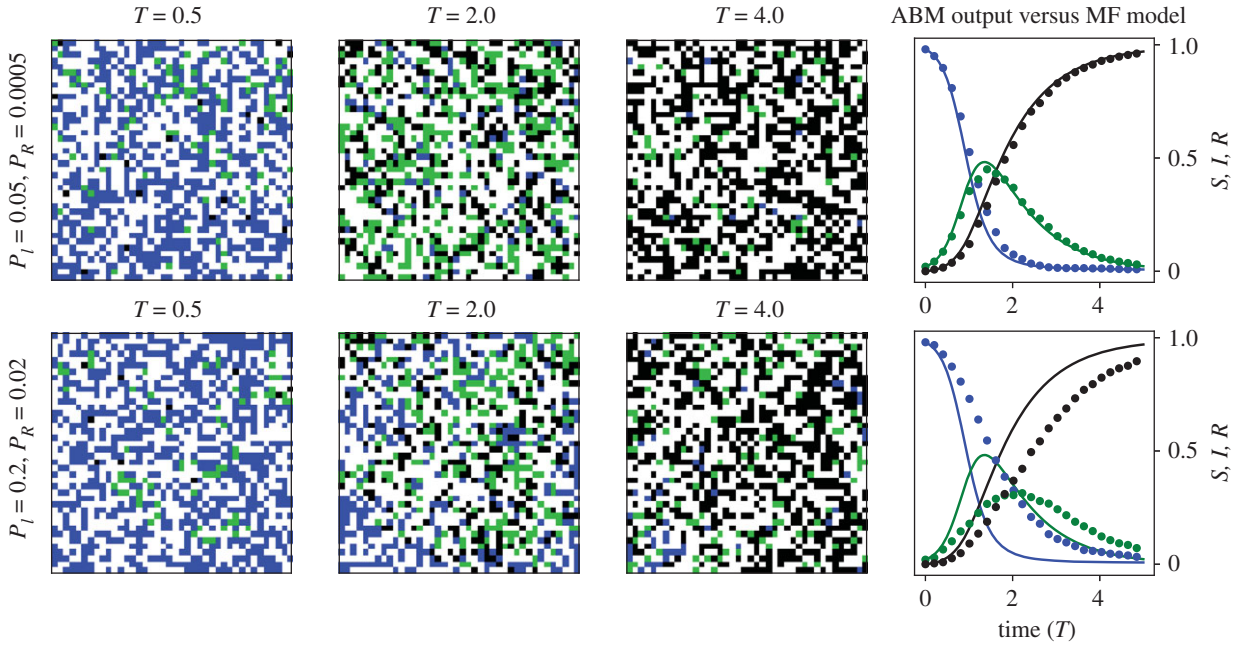

Figure 3. Simulation snap shots for the SIR ABM. Blue pixels denote $S_\alpha(t)$, green pixels denote $I_\alpha(t)$, black pixels denote $R_\alpha(t)$ and white pixels denote $0_\alpha(t)$. In the right-most column, we compare predictions of the mean-field SIR model (equation (2.12)) to the computed ratios of S, I and R from one ABM simulation. Solid lines correspond to the mean-field (MF) model and dots correspond to ABM simulation output. Quantities are plotted against non-dimensionalized time $T = P_R t$ for ease of interpretation. The ABM was computed on a square lattice of length $X = 40$.

corresponding ABM densities. The ABM simulation with $(P_I, P_R) = (0.005, 0.0005)$ appears well-mixed for all agent-types, which would satisfy the mean-field assumption. As a result, simulations of the mean-field model predict the ABM density well. If we increase the infection and recovery rates to $(P_I, P_R) = (0.2, 0.02)$, however, then the resulting ABM simulation has separate patches comprised primarily of infected agents or susceptible agents. These patches of single agent types decrease the population-wide average infection rate because infected agents in the middle of an infected cluster are unable to infect any susceptibles. As a result, the mean-field assumption is violated within these patches, and the mean-field model cannot accurately predict ABM dynamics.

For both of the BDM and SIR models, we have seen that some parameter regimes lead to close agreement between the ABM output and mean-field model predictions, whereas other parameters lead to poor agreement between the two. There is thus a need to develop methods that can determine when mean-field modes will accurately predict ABM dynamics and find novel DE models that accurately predict ABM dynamics when the mean-field models fail to do so.

3. Equation learning

For many EQL studies, the goal is to infer a dynamical systems model, written as $C(t)$, from a time-varying dataset, $C_d(t)$. This dynamical system can broadly be written as

$$\frac{\mathrm{d}C(t)}{\mathrm{d}t} = \mathcal{F}, \tag{3.1}$$

where $\mathcal{F}$ describes the dynamics of $C(t)$. When $C_d(t)$ is a time-varying scalar quantity (or a vector of scalar quantities), then an ODE model is relevant, in which case $\mathcal{F} = \mathcal{F}(t, C)$. When $C_d(t)$ varies over time and a one-dimensional spatial dimension, x, then a PDE model may be more relevant, in which case $\mathcal{F} = \mathcal{F}(t, x, C, C_x, C_{xx}, \ldots)$. In the following sections, we exemplify how methods from EQL may be used to learn a form of $\mathcal{F}$ from output ABM data.

3.1. Model learning example

In this section, we outline the steps one may take to learn a DE model from ABM data for the BDM model (figure 4). Code is provided for the results presented in this section in the file EQL Tutorial.ipynb.

3.1.1. Equation learning pipeline

We illustrate how to use EQL methods using data from the BDM process described in §2.1. In the first step, we simulate the ABM 50 times with parameter values $(P_p, P_m, P_d) = (0.01, 0.005, 1.0)$ and average the output to acquire $C_d(t) = \langle C_{\mathrm{ABM}}(t) \rangle$. The time vector, t, is sampled on an equispaced grid such that $t_i = (i-1)\Delta t$, $i = 1, \ldots, 100$ for some small $\Delta t > 0$.

In the second step of this process, we estimate the numerical derivative of $C_d(t)$. Finite difference computations are a simple method to approximate derivatives [49]. We use centred differences at the internal time points and forward and backward differences at the first and final time points, respectively:

$$\left.\begin{aligned} \frac{\mathrm{d}C_d(t_1)}{\mathrm{d}t} &= \frac{C_d(t_2) - C_d(t_1)}{\Delta t}, \\ \frac{\mathrm{d}C_d(t_i)}{\mathrm{d}t} &= \frac{C_d(t_{i+1}) - C_d(t_{i-1})}{2\Delta t}, \quad i = 2, \ldots, n-1, \\ \frac{\mathrm{d}C_d(t_n)}{\mathrm{d}t} &= \frac{C_d(t_n) - C_d(t_{n-1})}{\Delta t}. \end{aligned}\right\} \tag{3.2}$$

The resulting computation is plotted in Step 2 of the EQL pipeline depicted in figure 4.

The third step of the EQL pipeline requires the construction of a library of potential terms for inclusion in the inferred DE model. We saw that polynomials in $C(t)$ can describe the ABM output in §2.1.2, so we assume that the underlying

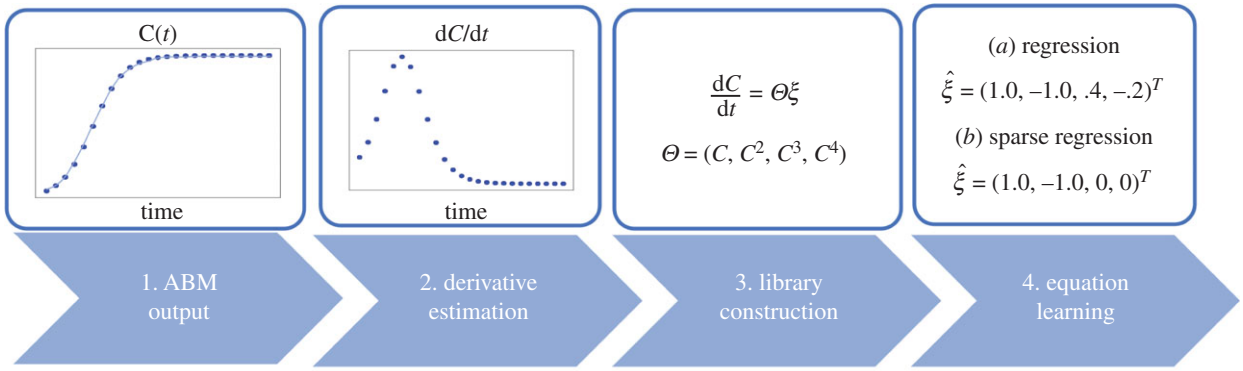

Figure 4. EQL pipeline. Step 1: Generate averaged ABM output; Step 2: Estimate the temporal derivative of the ABM output; Step 3: Library construction; Step 4: Equation inference. At Step 4, one can either perform (*a*) regression or (*b*) sparse linear regression to learn an equation for the ABM output. We will consider both the Lasso and Greedy sparse regression algorithms to perform EQL.

model here is a polynomial in $C(t)$. Recall from the rules of the BDM ABM that each agent interacts with its four neighbouring sites, so we further assume that this polynomial is up to fourth order. See reference [50] for scenarios where fourth or higher-order polynomials are needed to match ABM output for reactions involving two agents. As a non-zero constant in $\mathcal{F}$ would represent a constant source or sink of agents (which is not present in the ABM), we set the constant in this polynomial to be zero. Altogether, we propose the following possible model for the BDM process

$$\frac{dC}{dt} = \sum_{i=1}^{4} \xi_i C^i, \tag{3.3}$$

for unknowns $\xi_i \in \mathbb{R}$. Given data $C_d(t)$, we substitute into equation (3.3) to arrive at the following linear system of equations satisfied by the unknowns $\xi_1, \ldots, \xi_4$:

$$\begin{bmatrix} \frac{dC_d(t_1)}{dt} \\ \frac{dC_d(t_2)}{dt} \\ \vdots \\ \frac{dC_d(t_n)}{dt} \end{bmatrix} = \begin{bmatrix} C_d(t_1) & C_d^2(t_1) & C_d^3(t_1) & C_d^4(t_1) \\ C_d(t_2) & C_d^2(t_2) & C_d^3(t_2) & C_d^4(t_2) \\ \vdots & \vdots & \vdots & \vdots \\ C_d(t_n) & C_d^2(t_n) & C_d^3(t_n) & C_d^4(t_n) \end{bmatrix} \begin{bmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \\ \xi_4 \end{bmatrix}. \tag{3.4}$$

For convenience, we re-write equation (3.4) as

$$\frac{dC_d}{dt} = \boldsymbol{\Theta}\xi, \tag{3.5}$$

where the columns of the $n \times 4$ matrix $\boldsymbol{\Theta}$ contain the library terms evaluated at the data points $C_d(t_i)$, $i = 1, \ldots, n$.

The fourth step in the EQL pipeline is to infer the form of the DE model that best approximates the dynamics of $C_d(t)$. We do so by finding the least-squares solution of equation (3.5), given by

$$\hat{\xi} = \arg\min_{\xi \in \mathbb{R}^4} \left\{ \frac{1}{n} \left\| \frac{dC(t)}{dt} - \boldsymbol{\Theta}\xi \right\|_2^2 \right\}. \tag{3.6}$$

We solve equation (3.6) using numpy's **lstsq** command from the linear algebra package and find $\xi = [0.0048, -0.0105, 0.0031, -0.0030]^T$. This solution suggests that the following model best describes the ABM dynamics:

$$\frac{dC(t)}{dt} = 0.0048C - 0.0105C^2 + 0.0031C^3 - 0.0030C^4. \tag{3.7}$$

We numerically simulate equation (3.7) with initial condition $C(0) = C_d(0)$ using a fourth-order Runge–Kutta method [49]. The resulting output, $C(t)$, is depicted against $C_d(t)$ in figure 5 and we observe that this model accurately predicts the ABM output.

By solving equation (3.5) directly, we are likely to find forms of $\mathcal{F}$ with many terms because the system is overdetermined when n, the number of data points, satisfies $n \gg 4$. We may wonder if a simpler form of $\mathcal{F}$ can also accurately describe the ABM dynamics. A logistic model such as that presented in equation (2.6) has only two terms and may describe the data accurately, for example. Instead of solving equation (3.6), we can use sparse regression methods to find a sparse vector, ξ, to solve equation (3.5). There are many approaches to sparse regression, including the *least absolute shrinkage and selection operator* (Lasso), ridge regression, and the Greedy algorithm [25,51,52]. We use the Lasso algorithm in this section, but will return later to a discussion of whether alternative methods should also be considered. The Lasso method solves the regularized system

$$\hat{\xi} = \arg\min_{\xi \in \mathbb{R}^4} \left\{ \frac{1}{n} \left\| \frac{dC(t)}{dt} - \boldsymbol{\Theta}\xi \right\|_2^2 + \lambda \|\xi\|_1 \right\}, \tag{3.8}$$

for some $\lambda > 0$, which is called a *regularization parameter* [51]. Regularization is used to avoid extreme values in ξ and to prevent overfitting. There are many approaches to solve the Lasso problem; here we use the Fast iterative Shrinking-Thresholding Algorithm (FISTA) [53]. We provide the pseudo-code for this algorithm in algorithm 2 of appendix D. Using a regularization strength of $\lambda = 0.0004$ (see appendix E for a discussion of hyperparameter selection) we find the resulting vector to be $\hat{\xi} = [0.0047, -0.0095, 0, 0]$. This estimate suggests that the model equation

$$\frac{dC}{dt} = 0.0047C - 0.0095C^2, \tag{3.9}$$

should be a more parsimonious model for the BDM process than equation (3.7). We depict the solution to equation (3.9) (with initial condition $C(0) = C_d(0)$) against the ABM output in figure 5 and observe that this DE model accurately approximates $C_d(t)$. Furthermore, we observe that the form of equation (3.9) is similar to the logistic DE,

$$\frac{dC}{dt} = P_p C(1 - C) - P_d C = (P_p - P_d)C - P_p C^2. \tag{3.10}$$

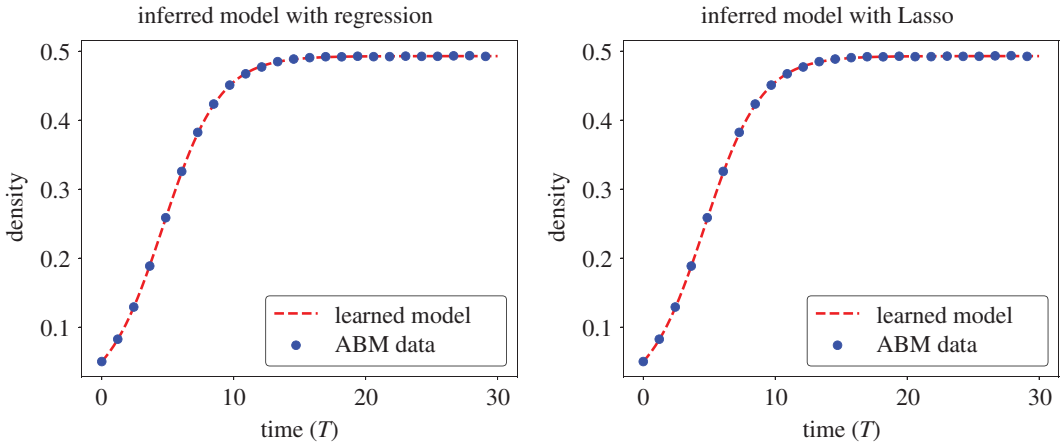

Figure 5. Simulations of the inferred models for the BDM ABM data. (left) The inferred DE model $dC/dt = 0.0100C - 0.0182C^2 + 0.0037C^3 - 0.0019C^4$ (determined by solving equation (3.5) using linear regression) is depicted against output ABM data, $C_d(t)$. (Right) The inferred DE model $dC/dt = 0.0047C - 0.0095C^2$ (determined by solving equation (3.5) using Lasso) is depicted against $C_d(t)$.

By comparing coefficients between equations (3.9) and (3.10) we can estimate the mechanistic ABM parameters P_p and P_d as $\hat{P}_p = 0.0095$ and $\hat{P}_d = 0.0048$. These estimates are very close to the true underlying values of $P_p = 0.01$ and $P_d = 0.05$. The proposed EQL methodology is thus able to simultaneously infer a DE model that accurately predicts ABM output and provides realistic parameter estimates when combined with the mean-field model (we note that more common forms of parameter estimation, such as maximum likelihood, may need to be used after the equation form has been determined). We have thus shown that concepts from EQL can be used to determine simple DE models that accurately describe ABM dynamics.

3.1.2. Different forms of sparse regression

As mentioned previously, there are many approaches to find sparse solutions to equation (3.5). The Greedy algorithm is another popular algorithm for sparse regression and solves a similar problem to the Lasso problem from equation (3.8). In the Greedy algorithm, we solve

$$\hat{\xi} = \arg\min_{\xi \in \mathbb{R}^4} \left\{ \frac{1}{n} \left\| \frac{dC(t)}{dt} - \boldsymbol{\Theta}\xi \right\|_2^2 + \lambda \|\xi\|_0 \right\}, \tag{3.11}$$

for some regularization parameter $\lambda > 0$. Here, $||\xi||_0$ counts the number of non-zero terms in ξ. We use the forward–backward approach to solve this system, which converts the regularization parameter λ into a tolerance hyperparameter. Pseudo-code for this algorithm is provided in [52]. We use this algorithm on the ABM data with a tolerance of 0.0001 and find the resulting vector to be $\hat{\xi} = [0.0047, -0.0095, 0.0001, 0]$, which suggests that the model equation

$$\frac{dC}{dt} = 0.0047C - 0.0095C^2 + 0.0001C^3, \tag{3.12}$$

is able to describe the ABM dynamics. We note that this learned equation is similar in form to the learned equation from Lasso (equation (3.9)), although there is an extra cubic term with a small coefficient. We will consider both the Lasso and Greedy algorithms for EQL in future case studies.

4. Case studies for equation learning in agent-based model analysis

We use this section to explore how methods from EQL can aid modellers in performing ABM analysis with EQL methods. We will do so through five case studies pertaining to: how learned equations change with ABM parameters in §4.1, how learned equations are affected by the number of performed ABM simulations in §4.2, the performance of learned equations in predicting unobserved ABM dynamics in §4.3, the performance of EQL methods for DE model selection from ABM data in §4.4, and the performance of EQL methods for learning systems of equations in §4.5. Through these case studies, we introduce and address six questions on the use and efficacy of EQL for ABM analysis.

4.1. Case study 1: comparing differential equation models in describing agent-based model dynamics

Coarse-grained models are advantageous for ABM analysis because they are easy to interpret, formulate and solve. Unfortunately, coarse-grained DE models only provide accurate ABM analysis in parameter regimes where ABM simulations adhere to their underlying assumptions [12,46]. Previous studies have defined criteria to aid modellers in determining when to rely on mean-field models, but these approaches are often only valid for simple ABMs and heuristic in nature for more complex scenarios, such as bistable systems [9,12,46,54,55]. The purpose of this case study is to determine if EQL methods can be used as a simple test to determine when mean-field models accurately predict ABM dynamics, and to propose novel models for more accurate inference. These goals are summarized in the following questions:

(Q1) Can EQL aid in determining when mean-field models accurately approximate ABM dynamics?
(Q2) Can methods from EQL discover novel DE models for accurate ABM analysis when the mean-field assumption is invalid?

To address both Questions (Q1) and (Q2), we will test the ability of the mean-field and learned DE models to predict ABM dynamics for the BDM model over a range of

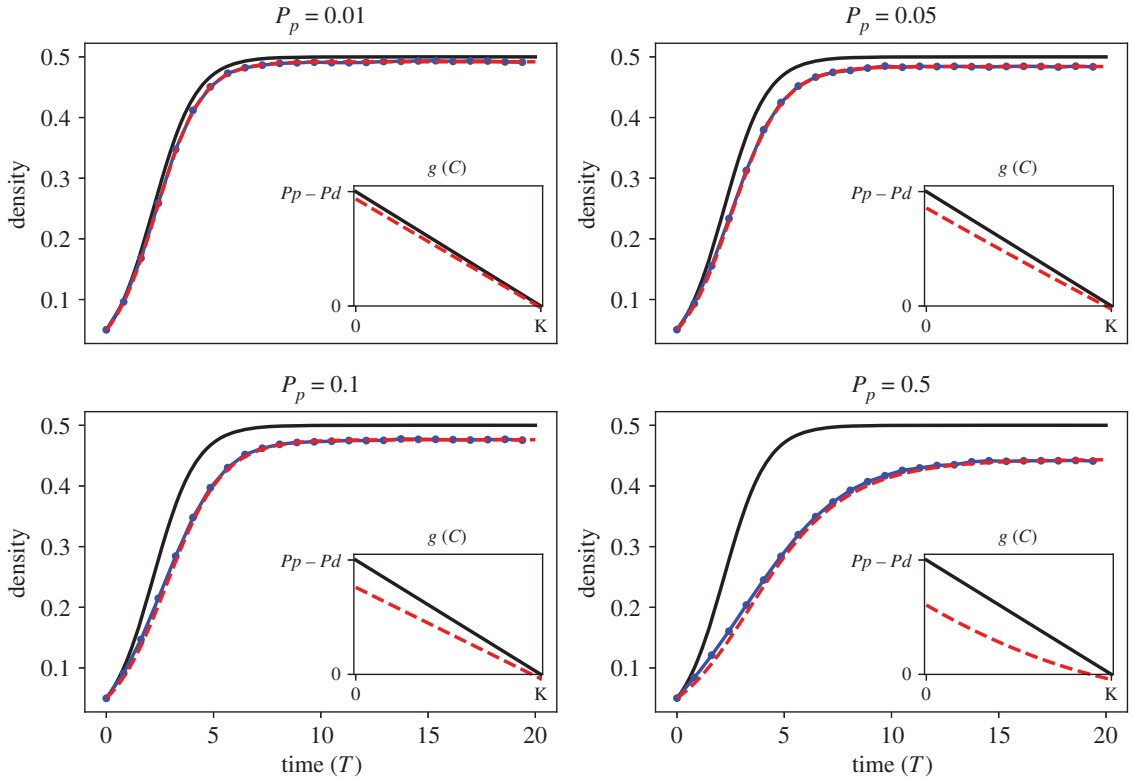

Figure 6. Case study 1. Comparing mean-field and learned model predictions to ABM data from the BDM process. In each figure, we depict $\langle C_{\mathrm{ABM}}(t)\rangle$ (blue dots and line) against the corresponding mean-field model (solid black line) and learned equation (red dashed line). All simulations are depicted as a function of non-dimensionalized time $T = t(P_p - P_d)$. The insets in all frames depict the predicted *percapita* growth rates, $\mathcal{G}(C)$, from both models where $\mathrm{d}C/\mathrm{d}t = C\mathcal{G}(C)$.

Table 1. Case study 1. Mean-field and learned DE models for the BDM process for various values of (P_p, P_d). MSE denotes the mean squared error between the model solution and $\langle C_{\mathrm{ABM}}(t)\rangle$.

P_p	P_d	mean-field model (MSE)	learned model (MSE)
0.01	0.005	$\mathrm{d}C/\mathrm{d}t = 0.005C - 0.01C^2$ (0.0011)	$\mathrm{d}C/\mathrm{d}t = 0.00468C - 0.0095C^2 - 0.0C^3$ (0.0001)
0.05	0.025	$\mathrm{d}C/\mathrm{d}t = 0.025C - 0.05C^2$ (0.0026)	$\mathrm{d}C/\mathrm{d}t = 0.02134C - 0.04572C^2 + 0.00343C^3$ (0.0002)
0.1	0.05	$\mathrm{d}C/\mathrm{d}t = 0.05C - 0.1C^2$ (0.004)	$\mathrm{d}C/\mathrm{d}t = 0.03962C - 0.09057C^2 + 0.01561C^3$ (0.0003)
0.5	0.25	$\mathrm{d}C/\mathrm{d}t = 0.25C - 0.5C^2$ (0.01)	$\mathrm{d}C/\mathrm{d}t = 0.15671C - 0.49984C^2 + 0.33125C^3$ (0.0005)

mechanistic ABM parameter values. The SIR model is considered in a later case study.

4.1.1. Varying proliferation rates for the birth–death–migration process

We now investigate the performance of the mean-field and learned DE models in describing ABM output from the BDM process over a range of proliferation rates. The proliferation rates are varied as $P_p = 0.01, 0.05, 0.1, 0.5$. For each value of P_p, we set the death rate, P_d, to be half of P_p and fix the migration rate at $P_m = 1$. The ABM output is comprised $C_d(t) = \langle C_{\mathrm{ABM}}(t)\rangle$, which is averaged over $N = 50$ ABM simulations to ensure convergence to mean behaviour and reduce the impact of noise. For model learning, we use the library of right-hand-side terms $\boldsymbol{\Theta} = [C, C^2, C^3, C^4]$, and use the Greedy algorithm [52] to sparsely solve the linear system $\mathrm{d}C_d/\mathrm{d}t = \boldsymbol{\Theta}\xi$. The code for this case study is provided in the file Case study 1. Varying parameters for BDM.ipynb.

We compute predictions of the mean-field and learned DE models with $\langle C_{\mathrm{ABM}}(t)\rangle$ in figure 6 as well as the model equations and their mean-squared error (MSE) in approximating $\langle C_{\mathrm{ABM}}(t)\rangle$ in table 1. Both the mean-field model and learned DE model provide accurate predictions of the ABM output for $P_p = 0.01$, with a MSE between the simulated model and $\langle C_{\mathrm{ABM}}(t)\rangle$ of 0.0011 and 0.0001, respectively. As P_p increases to 0.05, 0.1 and 0.5, the mean-field model does an increasingly poor job in predicting the ABM data, overpredicting agent density for all time. The MSE of the mean-field DE model increases with P_p. The learned DE models, on the other hand, accurately predict the ABM data and maintain MSE values below 0.0005 for all values of P_p. The learned DE model form is similar to the mean-field model for $P_p = 0.01$, but for $P_p = 0.05, 0.1, 0.5$ the learned model also recovers cubic terms. When the learned model resembles the mean-field model, then the mean-field model accurately predicts $\langle C_{\mathrm{ABM}}(t)\rangle$. On the other hand, when the learned model deviates from the mean-field model, the mean-field model poorly predicts the ABM data.

We suggest that the mean-field model can make accurate predictions of ABM behaviours when the learned equation closely resembles the mean-field model (including both

equation form and parameter estimates); otherwise, the mean-field model can lead to inaccurate predictions of ABM behaviours. Consider the *per capita growth rate* as one illustrative example. For a DE model of the form $dC/dt = \mathcal{F}(C)$, the *per capita* growth rate is defined by $\mathcal{F}(C) = C\mathcal{G}(C)$, and it quantifies the average contribution of each individual to population growth over time. We plot $\mathcal{G}(C)$ for the mean-field model and each learned model in the insets of figure 6. The mean-field model predicts that the *per capita* growth is a linear decreasing function connecting $(0,P_p - P_d)$ and $(K,0)$, where $K = (P_p - P_d)/P_p$ is the carrying capacity predicted by the mean-field model. The learned model predictions of $\mathcal{G}(C)$ closely resemble the mean-field model predictions of $\mathcal{G}(C)$ for $P_p = 0.01$ and 0.05. At the larger proliferation rates of $P_p = 0.1$ and 0.5, however, the learned model *per capita* growth rates are much lower than the mean-field model rates. Recall that higher rates of proliferation lead to spatial clustering of agents in the ABM: this clustering reduces the averaged *per capita* growth rate, which the learned model can account for but the mean-field model does not. The effective carrying capacity of the ABM reduces as P_p increases (with $P_d = P_p/2$), which is again likely due to increased spatial clustering. All learned models accurately capture this reduction in the carrying capacity, whereas the mean-field models do not.

The EQL pipeline results for these four simulated datasets suggests that the mean-field model will accurately describe ABM data when the learned equation form matches that of the mean-field model. When the mean-field models are not able to accurately predict ABM dynamics, learned models of the form

$$\frac{dC}{dt} = \alpha C + \beta C^2 + \gamma C^3, \quad \alpha, \beta, \gamma \in \mathbb{R}, \tag{4.1}$$

can accurately predict ABM data instead. This form of learned equation was able to provide more accurate ABM analysis than the mean-field model over a range of parameter values.

4.2. Case study 2: altering the number of agent-based model simulations

ABMs are inherently noisy due to the random updating of agent states that occurs during simulation. When using EQL methods to analyse such ABMs, one should take care to ensure they learn the mean dynamics and do not overfit to small trends in the data. We averaged ABM data over a large number of ABM simulations in previous sections of this study to ensure the data had converged to its mean value. Performing such extensive simulations may not be feasible for computationally intensive ABMs, however, so we now investigate how the learned DE model changes with the number of ABM simulations. This can be summarized with the following question:

(Q3) How can we determine when enough ABM simulations have been performed for accurate DE model learning?

To investigate (Q3), we consider datasets from the BDM ABM that have been averaged over different numbers of ABM simulations. All data in this case study are simulated using the parameter values $P_p = 0.01$, $P_d = 0.005$, and $P_m = 1$. ABM data are comprised $\langle C_{\mathrm{ABM}}(t)\rangle$, which is computed over $N = 1, 5, 10$ or 25 simulations, and we used 10 separate datasets for each value of N to investigate how stochastic fluctuations affect the final results. For DE learning, we use the Greedy algorithm to solve $dC/dt = \boldsymbol{\Theta}\xi$ for $\boldsymbol{\Theta} = [C, C^2, C^3, C^4]$. We denote $\hat{\xi}^N$ as the estimate of ξ that results for each value of N. The code for this case study is provided in the file Case study 2: varying number of ABM Simulations.ipynb.

For each value of N considered, we learned ten separate DE models from ten separate realizations of $\langle C_{\mathrm{ABM}}(t)\rangle$ and then computed the average learned DE model by averaging the coefficients from each of the 10 learned equations. Model predictions from the averaged learned DE model and mean-field model are depicted against all ABM data in figure 7. The solution profile for each average learned DE model does not change too much as N increases, but the ABM data fluctuations decrease with larger values of N. We depict the distributions of each coefficient for each value of N in fig. 11 in appendix F. As expected, the variance of each distribution decreases with N, so we can be more certain about learned DE models with more ABM simulations. We present the averaged learned DE models for each value of N in table 2 as well as the averaged mean-field and learned model MSEs. The mean-field MSE maintains a nearly constant value for all N, but the learned model MSE decreases with N. The learned equation is cubic for all values of N, and the cubic coefficient appears to approach zero as N increases. The differences in MSE between successive averaged ξ^N estimates (i.e. $\|\hat{\xi}^5 - \hat{\xi}^1\|_2$, etc.) is low for $N > 10$. The insensitivity of the learned equation above $N = 10$ suggests that $N = 10$ or 25 simulations are sufficient to accurately capture the mean BDM ABM dynamics for the parameter values we used. As discussed in §3.1.1, we used finite differences for numerical differentiation in these cases. Recent studies have demonstrated how the use of polynomial spline interpolation or artificial neural networks (ANNs) to improve EQL performance in the presence of noise levels on the magnitude of that observed in experimental data [24,25].

4.3. Case study 3: learning agent-based model dynamics from sparse time samples

A current challenge for modellers is to develop EQL methods that are able to learn DE models from real experimental or clinical data. ABMs are a useful intermediate step to test the predictions of mathematical methods because ABMs emulate the stochastic and discrete nature of many biological processes and allow researchers to alter aspects of the data. Biological data present many practical challenges for modellers, including only partial observations of the process under consideration or sparse sampling of the data [56]. We will use this case study to consider the performance of the EQL methods in the face of both limited data sampling and partial data observations. In turn, we address the following questions:

(Q4) How can we determine the resolution needed for accurate DE model learning?
(Q5) Which time scales are informative for learning predictive DE models for unobserved data?

4.3.1. Case study 3a: learning birth–death–migration dynamics from sparsely sampled agent-based model data

We applied the EQL methodology to ABM data where $n = 13, 25, 50$ and 100 time samples were collected. For all

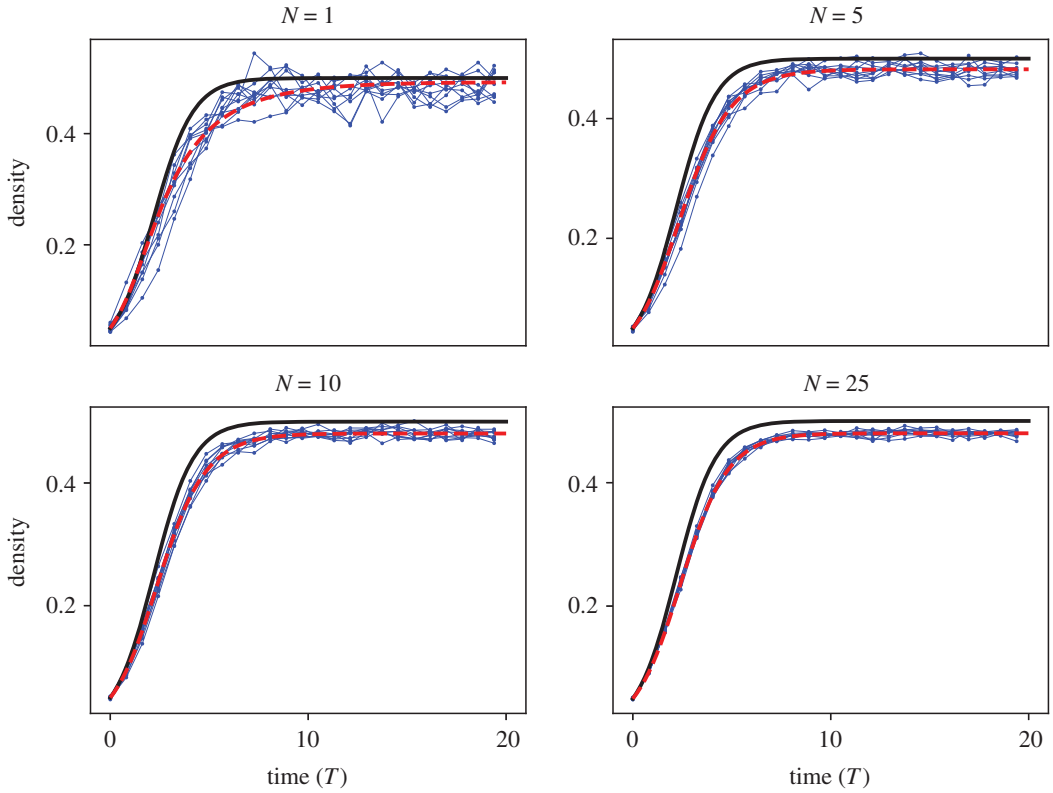

Figure 7. Case study 2. Learning equations from varying numbers of ABM simulations. In each figure, we depict ten realizations of $\langle C_{\mathrm{ABM}}(t)\rangle$ (blue dots and line) against the corresponding mean-field model (solid black line) and averaged learned equation, (red dashed line), which depicts the final learned equation whose coefficients were averaged over all the learned DE models for each realization of $\langle C_{\mathrm{ABM}}(t)\rangle$. Each realization of $\langle C_{\mathrm{ABM}}(t)\rangle$ was averaged over $N = 1$ (top left), $N = 5$ (top right), $N = 10$ (bottom left), and $N = 25$ (bottom right) simulations of the BDM model. Each individual simulation was initialized by placing agents uniformly at random throughout the lattice so that 5% of lattice sites were occupied. All simulations are depicted against non-dimensionalized time $T = t(P_p - P_d)$. The ABM parameters used in these simulations are $P_p = 0.01$, $P_d = 0.005$, $P_m = 1$.

Table 2. Case study 2. Learned DE models for the BDM process for various numbers of ABM simulations. We fixed $P_p = 0.01$, $P_d = 0.005$ and $P_m = 1$ in each scenario and averaged ABM output over the given value of N ABM simulations ten separate times to investigate the EQL method's performance in the presence of stochastic ABM fluctuations. The presented learned DE models depicts the final learned equation whose coefficients were averaged over all the learned DE models for each realization of $\langle C_{\mathrm{ABM}}(t)\rangle$. The right-most column corresponds to the MSE between successive $\hat{\xi}$ estimates: e.g. for $N = 5$, we compute $\|\hat{\xi}^5 - \hat{\xi}^1\|_2$.

N	mean-field model (MSE)	learned model (MSE)	$\hat{\xi}$ MSE
1	$dC/dt = 0.005C - 0.01C^2$ (0.0037)	$dC/dt = 0.00568C - 0.01953C^2 + 0.01624C^3$ (0.0025)	—
5	$dC/dt = 0.005C - 0.01C^2$ (0.0031)	$dC/dt = 0.00482C - 0.01299C^2 + 0.00622C^3$ (0.0012)	0.012
10	$dC/dt = 0.005C - 0.01C^2$ (0.0028)	$dC/dt = 0.00472C - 0.01193C^2 + 0.00439C^3$ (0.0008)	0.002
25	$dC/dt = 0.005C - 0.01C^2$ (0.0027)	$dC/dt = 0.00453C - 0.01054C^2 + 0.00232C^3$ (0.0005)	0.003

values of n, the data were collected at equispaced time intervals between the same starting and ending time points. All ABM simulations were computed with mechanistic parameters $P_p = 0.05$, $P_d = 0.0125$, and $P_m = 1$. The ABM data, $\langle C_{\mathrm{ABM}}(t)\rangle$, was averaged over $N = 50$ simulations. The code for this case study is provided in the file Case study 3a. Varying data resolution.ipynb.

We depict model predictions from the learned equation against ABM data in figure 8. The learned DE models accurately predict the ABM output for $n = 100$, 50 and 25 time samples. With $n = 13$ time samples, however, the learned equation predicts the same carrying capacity as the data but fails to accurately predict the ABM dynamics before plateauing (excluding the initial time sample, which is used as the initial condition). The learned model equations for these different scenarios, and the MSE between the learned model prediction and ABM data from each model, are summarized in table 3. The MSE is less than or equal to 10^{-3} for $n \geq 13$ but rises to 0.0154 for $n = 0.13$. From this case study, we suggest that $n = 25$ time samples provides sufficient time resolution to accurately infer the underlying dynamics for parameter values $P_p = 0.05$, $P_d = 0.0125$, and $P_m = 1$. If we analysed experimental data we had reason to believe resulted from similar underlying parameter values, we would trust our analysis of the experimental data if we had 25 or more equispaced time points over this same time interval. We note alternative methods of numerical differentiation can be used to improve EQL results with limited time samples here. As an example, we have recently introduced an ANN approach that has proven successful in learning

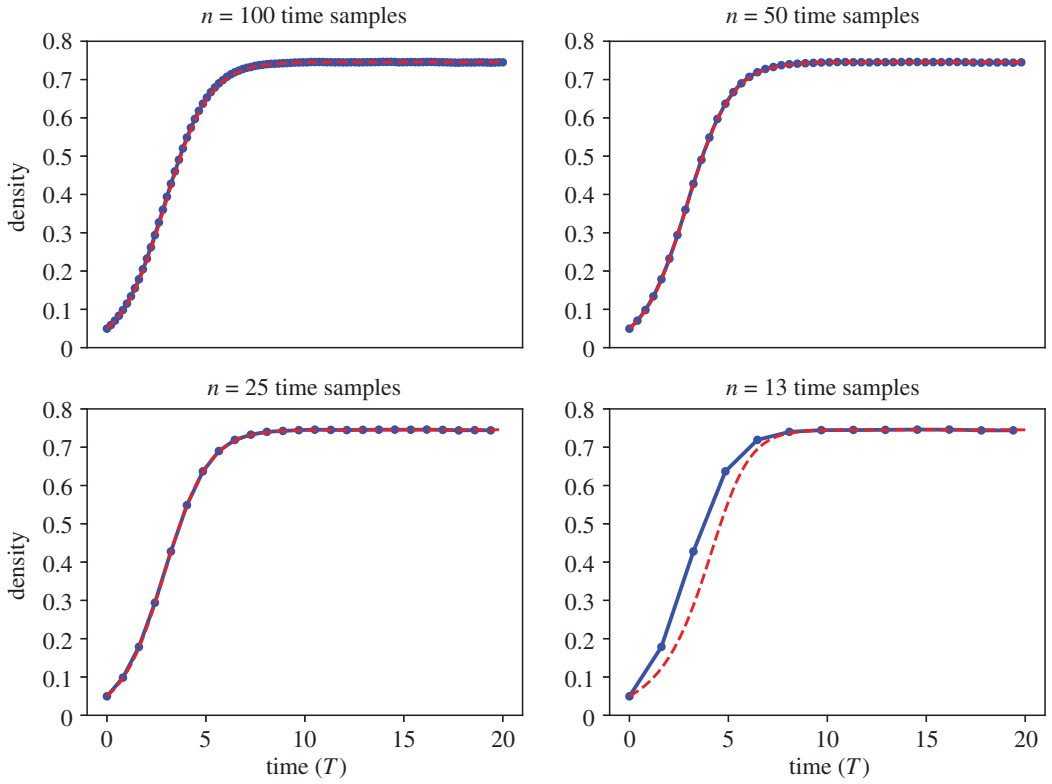

Figure 8. Case study 3a. Learning equations with varying time resolution. We applied the EQL methodology for ABM data with $n = 13$, 25, 50 or 100 time samples and depict the learned model (red dashed line) against the ABM data (blue solid line and dots). All simulations are depicted against non-dimensionalized time $T = t(P_p - P_d)$. We fixed $P_p = 0.05$, $P_d = 0.0125$, and $P_m = 1$.

Table 3. Case study 3a. Learned model equations for case study 3a with varying numbers of time samples, n, from the data. MSE denotes the mean squared error between the learned model prediction and $\langle C_{\mathrm{ABM}}(t)\rangle$.

% of data	learned model (MSE)
100	$dC/dt = 0.03374C - 0.04522C^2$ (0.0002)
50	$dC/dt = 0.03379C - 0.04531C^2 - 0.0C^3$ (0.0002)
25	$dC/dt = 0.03372C - 0.0452C^2$ (0.0004)
13	$dC/dt = 0.02073C - 0.03733C^3$ (0.0154)

equations from sparse PDE data [56]. This method would likely outperform the finite difference scheme used for differentiation throughout this study. We point the interested reader to [24,56] for more information on this ANN approach.

4.3.2. Case study 3b: predicting unobserved birth–death–migration dynamics

We applied the EQL methodology to ABM data where only the first 10%, 20%, 25% or 50% of the ABM data are used for training and the remaining data are held out for testing how well the learned DE model predicts unobserved dynamics. ABM data are computed with $P_p = 0.01$, $P_d = 0.005$ and $P_m = 1$ and $\langle C_{\mathrm{ABM}}(t)\rangle$ is averaged over 50 ABM simulations with $n = 100$ data points until $t = 15(P_p - P_d)$. The code for this case study is provided in the file Case study 3b: predicting unobserved dynamics.ipynb.

We depict model predictions against the testing and training data in figure 9. When trained on 10% of data, the inferred model grows without bound for all simulated time and does not accurately describe the testing data. When trained on 20% of data, the learned model plateaus above the carrying capacity of the data, approximately $C = 0.75$. For training data comprised 25% and 50% of the available data, the learned models closely predict the test data. We suggest that, for this case study, data should be sampled beyond the inflection point in order to accurately predict the unobserved dynamics and carrying capacity. One possible explanation for this observation is that all data before the inflection point is concave up and all data after the inflection point is concave down. Including data past the inflection point appears necessary to capture how $C_{\mathrm{ABM}}(t)$ becomes concave down before reaching the carrying capacity. Further investigation into informative datasets for EQL training have been explored elsewhere [56].

The learned DE model for these different data percentages and their MSEs on the test data set are summarized in table 4. As expected, the testing data MSE decreases as more of the data are used to learn the DE model.

4.4. Case study 4: using equation learning methods for model selection

Our first case study used EQL methods to demonstrate that the mean-field model may not be valid for predicting the dynamics of the BDM ABM when $P_p > 0.1$ (with other parameters fixed at $P_m = 1$ and $P_d = P_p/2$). If one wants to use a DE model to analyse such ABM simulations, then an alternative model may be required. In appendix A, we show the DE model given by

$$\frac{dC}{dt} = P_pC(1 - FC) - P_dC, \tag{4.2}$$

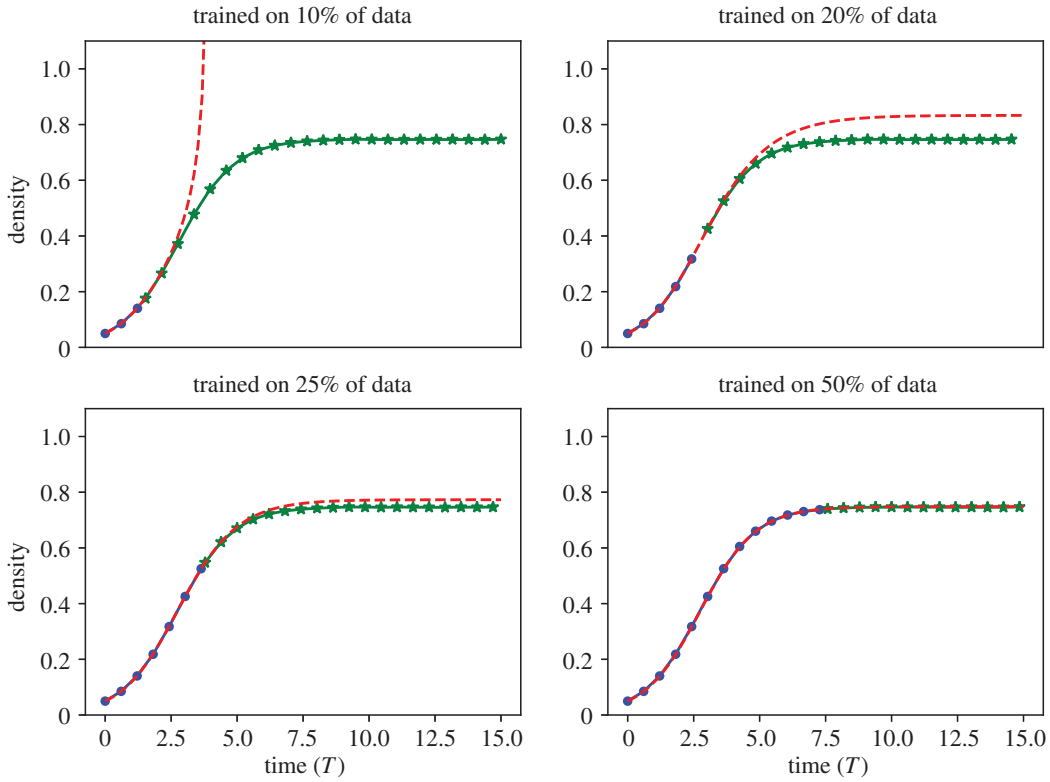

Figure 9. Case study 3b. Predicting BDM dynamics from partial ABM data. We applied the EQL methodology on the first 10% (top left), 20% (top right), 25% (bottom left) and 50% (bottom right) of $\langle C_{\text{ABM}}(t)\rangle$ to investigate the learned equations performance in predicting unobserved ABM dynamics. The blue dots correspond to ABM output that was used to infer the learned model, the green stars denote ABM output that was used for model testing, and the red dashed line denotes the solution of the learned model. All simulations are depicted against non-dimensionalized time $T = t(P_p - P_d)$. We fixed $P_p = 0.01$, $P_d = 0.005$ and $P_m = 1$ in each scenario and averaged ABM output over $N = 50$ simulations.

Table 4. Case study 3b. Summary of models learned with the first 10%, 20%, 25% and 50% of $\langle C_{\text{ABM}}(t)\rangle$. MSE on the remaining data is given in parentheses. Data were simulated with the BDM model using $P_p = 0.01$, $P_d = 0.005$ and $P_m = 1$, averaged over $N = 50$ ABM simulations. Note that our implementation of the learned DE model trained on 10% of the data fails to converge because this model grows faster than exponential growth, which is not realistic of the ABM data. We depict the MSE of this learned equation against $\langle C_{\text{ABM}}(t)\rangle$ as nan (not a number) due to this numerical instability.

%	learned model (testing MSE)
10	$dC/dt = 0.00802C - 0.02098C^2 + 0.03837C^3$ (nan)
20	$dC/dt = 0.00745C - 0.01154C^2 + 0.00305C^3$ (0.0071)
25	$dC/dt = 0.00737C - 0.01081C^2 + 0.00159C^3$ (0.0021)
50	$dC/dt = 0.00721C - 0.00975C^2 + 0.00016C^3$ (0.0002)

can be used to model output from the BDM ABM. In equation (4.2), F is the occupancy correlation between neighbouring lattice sites [9]. For the lattice-based BDM model, this value is defined between two neighbouring lattice sites α, β as

$$F(t) = \frac{\mathbb{P}[A_\alpha(t), A_\beta(t)]}{\mathbb{P}[A_\alpha(t)]\mathbb{P}[A_\beta(t)]}. \tag{4.3}$$

Note that if the occupancy probabilities of these sites are independent, then $F(t) \equiv 1$, and equation (4.2) simplifies to the mean-field DE model.

Model selection studies [57] are suited to determine which model most parsimoniously describes a given dataset from several plausible models. We may now be interested in determining which of our two models, the mean-field model in equation (2.6), or the DE model in equation (4.2), best describes ABM output for the BDM process. A typical model selection study for these two models may be problematic, however, as deriving and computing the DE model for F in equation (4.2) is complicated even for a scenario as simple as the BDM model, yielding an additional set of auxiliary DEs needed to describe F [9]. Since the derivation of such auxiliary equations is effectively intractable for more complex ABM models, we will investigate the following question:

(Q6) Can methods from EQL be used for DE model selection for ABM analysis?

In doing so, we propose an alternative strategy for model selection, i.e. for selecting additional time-dependent variables, such as the occupation correlation, $F(t)$, to increase DE model accuracy with concepts from EQL.

4.4.1. Model selection with equation learning for the birth–death–migration process

We use this section to demonstrate how EQL methods can be used for model selection between the logistic equation given by equation (2.6) and the modified logistic equation given by equation (4.2). The code for this case study is provided in the file Case study 4: model selection with EQL.ipynb. The rate of proliferation, P_p, varies over the values $P_p = 0.005$, 0.01,

0.05, 0.1, 0.5. For each value of P_p, we set $P_d = P_p/2$ and fix $P_m = 1$. The occupancy correlation value from equation (4.3) is estimated from the n^{th} of N ABM simulations by computing

$$F^{(n)}(t) = \frac{C^{(n,2)}_{\text{ABM}}(t)X^4}{\chi^2(C^{(n)}_{\text{ABM}}(t))^2}, \tag{4.4}$$

where $C^{(n,2)}_{\text{ABM}}(t)$ is the number of jointly occupied neighbouring pairs of lattice sites over time, X is the length of the lattice and χ^2 is the total number of adjacent lattice-site pairs [12]. The average occupancy correlation values is then averaged over all $N = 50$ simulations:

$$\langle F(t)\rangle = \frac{1}{N}\sum_{k=1}^{N} F^{(n)}(1, t). \tag{4.5}$$

The model selection methodology using EQL concepts proceeds as follows. From each ABM dataset, we compute both $C_d = \langle C_{\text{ABM}}(t)\rangle$ and $F_d = \langle F(t)\rangle$ and substitute these values into two separate $n \times 2$ matrices of right-hand-side terms given by

$$\boldsymbol{\Theta}_1 = [C_d(1 - C_d), C_d] \quad \text{and} \quad \boldsymbol{\Theta}_2 = [C_d(1 - FC_d), C_d],$$

where any multiplication is performed element-wise. Note that $\boldsymbol{\Theta}_1$ corresponds to the library of terms for equation (2.6) while $\boldsymbol{\Theta}_2$ corresponds to the library of terms for equation (4.2). We uniformly at random place half of the elements comprising $\mathrm{d}C_d(t)/\mathrm{d}t$ and the corresponding rows from $\boldsymbol{\Theta}_1$, $\boldsymbol{\Theta}_2$ into training sets given by the vector $\mathrm{d}C^{\text{train}}_d(t)/\mathrm{d}t$ and the $n/2 \times 2$ matrices $\boldsymbol{\Theta}^{\text{train}}_1$, $\boldsymbol{\Theta}^{\text{train}}_2$. The remaining elements of $\mathrm{d}C_d(t)/\mathrm{d}t$ and rows of $\boldsymbol{\Theta}_1$, $\boldsymbol{\Theta}_2$ are placed into testing sets given by $\mathrm{d}C^{\text{test}}_d(t)/\mathrm{d}t$ and matrices $\boldsymbol{\Theta}^{\text{test}}_1$, $\boldsymbol{\Theta}^{\text{test}}_2$. The training set is used to estimate $\hat{\xi}_1$ from $\boldsymbol{\Theta}^{\text{train}}_1$ and $\hat{\xi}_2$ from $\boldsymbol{\Theta}^{\text{train}}_2$ by solving the two linear regression problems

$$\frac{\mathrm{d}C^{\text{train}}_d}{\mathrm{d}t} = \boldsymbol{\Theta}^{\text{train}}_i \xi_i, \quad i = 1, 2. \tag{4.6}$$

Note that the vector $\hat{\xi}_1$ parametrizes equation (2.6), and $\hat{\xi}_2$ parametrizes equation (4.2). We then use the testing set to select the best model for each dataset by

$$\hat{\xi}_i = \arg\min_i \left\| \frac{\mathrm{d}C^{\text{test}}_d}{\mathrm{d}t} - \boldsymbol{\Theta}^{\text{test}}_i \hat{\xi}_i \right\|_2. \tag{4.7}$$

We use 100 randomly sampled training and validation sets and select whichever of the two models minimizes equation (4.7) more often in these 100 testing-validation realizations.

In table 5, we present the final selected models for various values of P_p. The mean-field model is selected for $P_p = 0.005$ and 0.01 with 57 and 77 of the 100 total votes, respectively. Equation (4.2) is selected for all larger proliferation values with at least 93 of the 100 total votes. These results are in agreement with Case study 1, where the mean-field model predicted ABM output well at $P_p = 0.01$, but was unable to predict these data for larger values of P_p (see, e.g. figure 6). In addition to more accurately matching the ABM output than the mean-field model, equation (4.2) also provides accurate parameter estimates for P_p and P_d. By matching the forms of the modified logistic equation to the learned equation (i.e. we can estimate P_p using the coefficient in front of $C(1 - FC)$ and P_d using the negative of the coefficient in front of C), we observe that these estimates appear very close to their true underlying values.

Table 5. Case study 4. Model selection with EQL. We present the average selected model equations for $\langle C_{\text{ABM}}(t)\rangle$ over various values of P_p. For each value of P_p, we set $P_d = P_p/2$ and $P_m = 1$. ABM output is averaged over $N = 50$ ABM simulations of the BDM model to ensure convergence to mean behaviour. The right-most column lists how many votes the selected equation received out of 100 total.

P_p	P_d	selected model	votes (out of 100)
0.005	0.0025	$\mathrm{d}C/\mathrm{d}t = -0.00245C + 0.00485C(1 - C)$	57
0.01	0.005	$\mathrm{d}C/\mathrm{d}t = -0.00483C + 0.00952C(1 - C)$	77
0.05	0.025	$\mathrm{d}C/\mathrm{d}t = -0.02482C + 0.04936C(1 - FC)$	93
0.1	0.05	$\mathrm{d}C/\mathrm{d}t = -0.04966C + 0.09874C(1 - FC)$	100
0.5	0.25	$\mathrm{d}C/\mathrm{d}t = -0.25248C + 0.50271C(1 - FC)$	100

4.5. Case study 5: varying infection probability rates for the susceptible–infected–recovered model

The EQL methods considered in this work are applicable to many ABMs, including the SIR model introduced in §2.2. We will now learn models for this ABM over several parameter regimes and test the performance of the mean-field and learned models in predicting ABM output [47,58,59].

We let the agent infection rate take the values $P_I = 0.005$, 0.01, 0.05, 0.1, set the agent recovery rate, P_R, to be one-tenth of the infection rate for each value of P_I, and fix $P_m = 1$ for all scenarios. The ABM output comprises $S_d = \langle S_{\text{ABM}}(t)\rangle$, $I_d = \langle I_{\text{ABM}}(t)\rangle$ and $R_d = \langle R_{\text{ABM}}(t)\rangle$, all of which are averaged over $N = 25$ ABM simulations from a square lattice with length $X = 40$. Because these three quantities will always sum to unity in the model, we focus on learning equations for $S(t)$ and $I(t)$ and note that $R(t) = 1 - S(t) - I(t)$. For model learning, we use the matrix of potential right-hand-side terms given by $\boldsymbol{\Theta} = [S, S^2, I, I^2, SI]$. The Lasso algorithm is used here to solve for the unknown vectors ξ_1 and ξ_2 from the linear systems $\mathrm{d}S/\mathrm{d}t = \boldsymbol{\Theta}\xi_1$ and $\mathrm{d}I/\mathrm{d}t = \boldsymbol{\Theta}\xi_2$. Note that even for a two-dimensional system with five library terms, the total number of possible models for the S and I variables is $\sum_{i=0}^{5}\binom{5}{i} = 32$ each, resulting in a total combination of $32^2 = 1024$ possible DE models, which highlights the difficulty in finding a predictive model from this large suite of possibilities. The code for this case study is provided in the file Case study 5. SIR Varying params.ipynb.

We depict the predictions of the mean-field and learned DE models over time against ABM output in figure 10. The mean-field and learned DE model equations with corresponding MSEs are presented in table 6. For $P_I = 0.005$ the mean-field model predicts the ABM output well, but as P_I increases the mean-field model predictions worsen, as evidenced by increases in the MSE. The mean-field model underpredicts the susceptible agent density at all times and overpredicts the infected agent density at early times. The learned model, on the other hand, is able to predict ABM

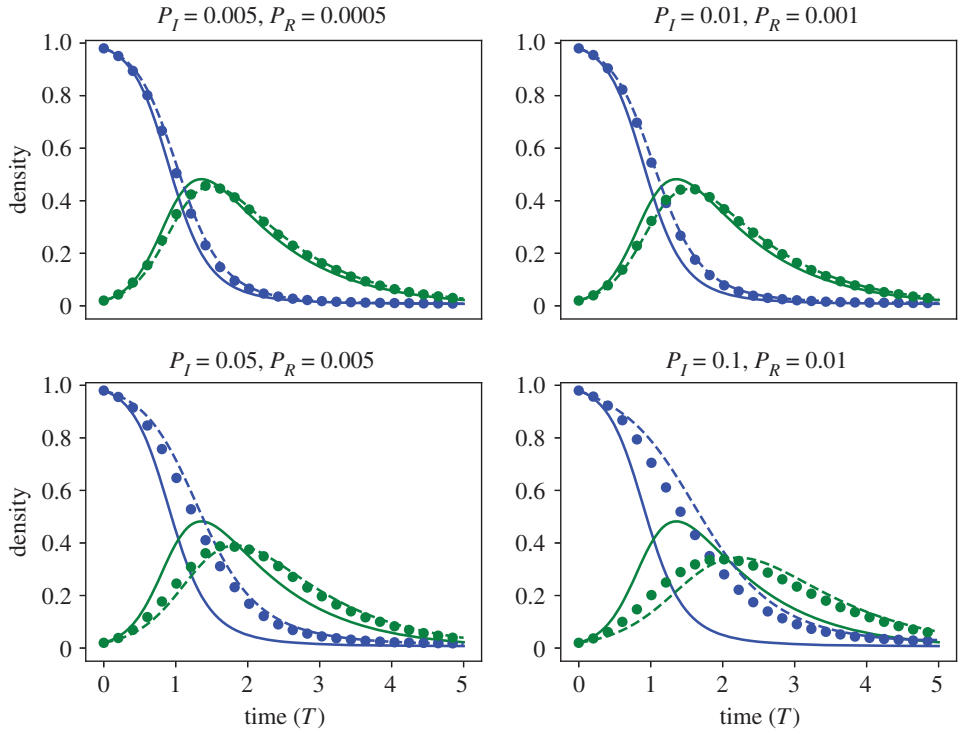

Figure 10. Case study 5. Comparing mean-field and learned model predictions to ABM data from the SIR model. In each figure, we depict the predictions of the mean-field model (solid blue curve for $S(t)$ and solid green curve for $I(t)$), against the predictions of the learned DE models (blue dashed curve for $S(t)$ and dashed green line for $I(t)$), as well as $\langle S_{\mathrm{ABM}}(t)\rangle$ (blue dots) and $\langle I_{\mathrm{ABM}}(T)\rangle$ (green dots). All simulations are depicted against non-dimensionalized time $T = tP_R$. We fixed $P_m = 1$ for each simulation. All ABM simulations were computed on a square lattice of length $X = 40$.

Table 6. Case study 5: mean-field and learned DE models for the SIR ABM for various values of (P_I, P_R), along with the computed MSE values between model and ABM output, and model R_0 calculations. We fix $P_m = 1$ in each scenario.

P_I	P_R	mean-field model (MSE)	R_0	learned model (MSE)	R_0
0.005	0.0005	$dS/dt = -0.0025IS$ (0.0027)	5.0	$dS/dt = -0.00229IS$ (0.0012)	4.68
		$dI/dt = 0.0025IS - 0.0005I$ (0.002)		$dI/dt = -0.00049I + 0.00225IS$ (0.0009)	
0.01	0.001	$dS/dt = -0.005IS$ (0.0044)	5.0	$dS/dt = -0.00447IS$ (0.0006)	4.52
		$dI/dt = 0.005IS - 0.001I$ (0.0029)		$dI/dt = -0.00098I + 0.00443IS$ (0.0005)	
0.05	0.005	$dS/dt = -0.025IS$ (0.0107)	5.0	$dS/dt = -0.01881IS$ (0.003)	3.72
		$dI/dt = 0.025IS - 0.005I$ (0.0066)		$dI/dt = -0.00472I + 0.01833IS$ (0.0021)	
0.1	0.01	$dS/dt = -0.05IS$ (0.0164)	5.0	$dS/dt = 0.01569S - 0.01608S^2 - 0.00545I - 0.0459IS$ (0.0055)	3.41
		$dI/dt = 0.05IS - 0.01I$ (0.0097)		$dI/dt = -0.00906I + 0.03101IS$ (0.0033)	

output accurately for all values considered and achieves low MSE values (with the lowest value at $P_I = 0.01$). The learned equation forms are similar to the mean-field model for $P_I = 0.005$, 0.01 and 0.05. For $P_I = 0.1$, the learned equations have additional S, S^2 and I terms in the model equations for dS/dt. These terms may be used to capture effects that are not described by the mean-field model.

The basic reproductive number, or R_0, is defined as the expected number of secondary infections that result from a single infection in a population comprised solely of susceptible agents [47,58,59]. A disease may spread rapidly when $R_0 > 1$, and will die out when $R_0 < 1$. R_0 is now a commonly used criterion to determine when a disease will continue to spread through a population and possibly cause an outbreak. More details of how to computer R_0 are provided in [47], but we note here that R_0 can be found for the mean-field SIR model by determining critical parameter values where $dI(t = 0)/dt$ will exceed zero at the initial condition. From equation (2.12), the mean-field model shows that $dI/dt > 0$ when $MP_IS > P_R$. Recalling that $S(t) \approx 1$ at the start of the disease spread, this implies that $dI/dt > 0$ when $MP_I/P_R > 1$, i.e. $R_0 = MP_I/P_R$ for the mean-field model. The values of R_0 can be computed using similar methods for the learned equations detailed in table 6. Here we observe that the mean-field model predicts that $R_0 = 5.0$ for all considered scenarios (recall, $M = 0.5$ in all simulations), yet as P_I increases from 0.005 to 0.01, 0.1, 0.25 (with $P_I/P_R = 10.0$), the learned equations predict $R_0 = 4.68, 4.52, 3.72, 3.41$, respectively.

Table 7. Highlighting the strengths (✓) and limitations (✗) of the three approaches in this article for understanding the resulting behaviour from ABMs.

	extensive simulation	DE model derivation	equation learning
amenable for ABMs of varying complexity	✓	✗	✓
computationally efficient	✗	✓	✓
amenable to analytical forms of exploration	✗	✓	✓
provides accurate estimates of emergent ABM behaviour throughout parameter space	✓	✗	✓
extrapolates to unobserved parameter values and data	✓	✓	✗
interpretable	✗	✓	✓

Thus, while the mean-field model suggests that a single infected individual will cause five secondary infections (in a population full of susceptible agents) at each of these infection rates, the learned models suggest that as P_I increases (with the ratios P_I/P_R and P_I/P_m fixed), the number of secondary infections will decrease in this scenario. Recall from figure 3 that infected agents tend to cluster with large values of P_I (while P_m and P_I/P_R are fixed). The mean-field model does not consider how such spatial clustering may affect the number of secondary infections that result from a single infection in a population comprised solely of susceptible agent. The learned DE models, on the other hand, implicitly account for this information when finding a predictive DE model from the ABM data.

5. Conclusion and discussion

In this work, we have considered three different, yet synergistic, approaches to study the emergent behaviour of ABMs. These approaches include extensive simulation, DE model derivation using coarse graining approaches and EQL. We demonstrated some of the strengths and weaknesses of each approach through their applications to two ABMs: a BDM model and an SIR model. We summarize some of these strengths and weaknesses in table 7.

Extensive simulation is the most commonly used and straightforward approach in understanding the behaviour of ABMs. Extensive simulation is advantageous because, in theory, it can be used to analyse any ABM: it only requires simulating the ABM on a computer. This approach quickly becomes practically challenging, however, because it can become difficult to simulate sufficiently complex ABMs with basic computing hardware. The BDM process used in this work is simple enough to be simulated on a personal laptop (2.5 GHz Intel Core i7 processor, 16 GB RAM) for all datasets generated for this study. The SIR model is more involved, however, and was run on a computer cluster to ease implementation. Computation of a single SIR simulation with $P_I = 0.005$, $P_R = 0.0005$, $P_M = 1.0$ took about 48 min to compute. Two further drawbacks of performing extensive simulation are that this technique is not compatible with analytical methods, and that its output may not be interpretable, i.e. we observe that the system behaviour changes as parameters are varied, but we may not understand *why* this happens, nor can we necessarily predict this behaviour *a priori*.

There is a wide literature demonstrating that coarse graining approaches can be used to derive DE models that approximate ABM dynamics [3,9,12,17,35,60,61]. Such DE models are advantageous because they are usually simple to solve (either analytically or numerically), interpretable, and provide insight into how changing ABM parameters can lead to emergent behaviours. Analytical techniques allow us to infer how the system will behave over many different parameter values without simulating the ABM. For example, the *per capita* growth rate for the BDM model predicts the population growth rate as a function of density, and the R_0 calculation for the SIR model predicts if a disease will outbreak or die out. However, we saw throughout the presented case studies that these analyses fail to accurately predict ABM behaviour for parameter regimes where the mean-field assumption is violated. Another limitation of this approach is that the derivation of DE models requires user-made assumptions, such as the mean-field assumption, which are often violated by many ABM simulations. As there is no universal methodology to convert ABM rules into predictive DE models for all input parameter values, EQL provides a powerful framework to learn informative DE models for ABMs and, in turn, determine when simplifying assumptions (e.g. the mean-field assumption) are reasonable.

EQL is a recent field of research that seeks to infer DE models directly from observed data. EQL combines many benefits of the two previously mentioned methodologies to analyse the emergent behaviour of ABMs. We highlighted many of the advantages of these approaches, and addressed several ways in which EQL can aid modellers in ABM analysis, through our investigation of Questions (Q1)–(Q6). While the mean-field model does not accurately predict ABM output for many parameter combinations, our exploration of (Q1) demonstrated that EQL provides a simple way to determine when the mean-field model can or cannot be trusted for such predictions. If the learned equation form matches the mean-field model, then the mean-field model should provide accurate insight; when it does not, then alternative models may be needed, as we discussed in (Q2). We also demonstrated that EQL methods can be used to infer novel DE models for ABMs. We observed that the mean-field model cannot predict output of the BDM process for large rates of agent proliferation and death relative to motility. Instead, equation (4.1) can be used to accurately predict ABM dynamics. Further investigation needs to be carried

out in order to understand how equation (4.1) may result from ABM rules.

A significant challenge of using ABMs in practice is their intensive computational nature. Through (Q3), we explored how many ABM simulations are necessary to reliably predict ABM output. We showed that the learned equation may deviate from the average ABM behaviour when only a small number of ABM simulations are used. With a sufficiently large number of ABM simulations, however, the learned equation can accurately predict ABM dynamics. For the BDM process, we found that $N = 10$ ABM simulations was sufficient. In practice, we can determine when enough simulations have been performed to capture mean ABM dynamics by considering how much the learned vector, ξ, changes with additional ABM simulations. When this vector becomes sufficiently insensitive to increases in N, then sufficient ABM dynamics have been performed.

In (Q4), we investigated the sampling resolution needed in time to reliably predict ABM dynamics. In this case study, we determined a magnitude between uniformly sampled time samples above which the EQL pipeline could not learn a DE model that accurately predicted ABM behaviour. We used Question (Q5) to determine when EQL methods can be used to predict unobserved ABM dynamics. In the presented case study, we used the BDM model to demonstrate that the learned equations can accurately predict unobserved ABM dynamics when the observed data exceeds half of the population's carrying capacity. These two investigations, Q4 and Q5, suggest important future work must be performed to determine strategic (and not necessarily uniform) samples of the ABM that are informative and capture all dynamic regimes of the data). Some preliminary work towards these questions for PDE models has been investigated in [56]. A limitation of EQL methods, as opposed to ABM simulation and mean-field models, is that it may not be able to accurately extrapolate to unobserved data and parameters, as we observed in Case study 3b. This is possible with ABM simulation, where the ABM can be run for longer time or at different parameter values. Similarly, the mean-field model can extrapolate its predictions by solving the model over a longer time period or simulating it at different parameter values.

Finally, we considered Question (Q6) to determine if EQL methods can be used to aid in model selection for ABMs. This is advantageous because DE models that are more complex than mean-field models can also be derived for ABMs. Although potentially more accurate, these models are difficult to interpret and simulate than mean-field models. Equation (4.2), as one example, can be challenging to implement numerically because the dynamical system for the occupancy correlation function, $F(t)$, requires solving a high-dimensional system of equations using user-defined closure approximations [9]. Instead, we can measure F over each ABM simulation and use these observations of F in a hybrid approach to select whether including F leads to a more accurate DE model. Applying such a hybrid approach to the simulation of DE models has been recently proposed to aid in reducing identifiability-related issues [62,63].

EQL methods are quickly growing in popularity as a means to infer DE models from noisy data [24,25,56]. We have shown in this study that such methods provide a reliable and promising tool to aid modellers in interpreting and analysing ABMs. Learning DE models from data do not require user-made assumptions on whether or not the ABM simulation satisfies certain properties, as is needed for the derivation of mean-field models. We have also demonstrated in this work how EQL methods can be used to predict ABM dynamics from limited ABM simulations, learn DE models from only a subset of data, and accurately predict dynamics over a wide range of parameter values. Such learned equations from ABM data also make ABM analysis more interpretable, as analysis of the learned equation provides insight into the underlying biological mechanisms (e.g. *per capita* growth rates, R_0, bifurcation analysis, etc.). The order of a learned equation may provide insight into how many neighbouring lattice site occupancies impact individual agent behaviour [50].

There are many areas for exciting future work in ABM analysis using EQL methods. For example, global sensitivity analysis techniques [64] could be used to determine ABM parameter thresholds where the learned equation forms change, and what insights these threshold values may provide into the ABM dynamics. We anticipate that, through this tutorial, EQL will increasingly be used to interpret complex ABM simulations. Future work should aim to address challenges that prevent the learning of DE models from real experimental data. While we focused on learning deterministic DE models from stochastic ABM simulations in this work, more recent studies have explored learning stochastic DE model forms (including both drift and diffusion estimates) from data [65]. Another recent study has shown that the dynamics from a stochastic non-Markovian model can be learned using a simpler time-inhomogeneous Markovian model framework with the aid of ANNs [66]. We also focused on simple ABMs in this study (including the BDM process and the SIR model), but future work should examine model learning for more complex ABM dynamics, such as bistable [60] and periodic behaviour [67].

Data accessibility. All data considered in this work were generated from model simulations. All code and data used are provided at the following Github link: https://github.com/johnnardini/Learning-DE-models-fromstochastic-ABMs.

Authors' contributions. J.T.N. implemented the methods and created the simulated datasets; J.T.N., R.E.B., M.J.S. and K.B.F. conceived the methodology, interpreted the results and wrote the paper.

Competing interests. We declare we have no competing interests.

Funding. R.E.B. is a Royal Society Wolfson Research Merit Award holder and would like to acknowledge BBSRC for funding through grant BB/R000816/1. M.J.S. is supported by the Australian Research Council (DP200100177). K.B.F. is supported in part by the National Science Foundation (Grant No. IOS-1838314) and in part by the National Institute of Aging (Grant No. R21AG059099)

Appendix A. Coarse graining birth–death–migration rules into a differential equation model

In this section, we will derive coarse-grained DE models of the BDM process. We begin by defining $\mathbb{P}[0_\alpha(t)]$ and $\mathbb{P}[A_\alpha(t)]$ as the probabilities that the individual lattice site α is either vacant or occupied, respectively, at time t. We simplify notation by writing: $\mathbb{P}[A_\alpha(t)] = C_\alpha(t)$ and $\mathbb{P}[0_\alpha(t)] = 1 - C_\alpha(t)$.

Similarly, let $\mathbb{P}[A_\alpha(t), A_\beta(t)]$ denotes the probability that both neighbouring sites α and β are occupied at time t; we refer to this value as the *neighbouring lattice site occupancy probability*. Along these lines, $\mathbb{P}[0_\alpha(t), A_\beta(t)]$ is the probability that α is vacant and β is occupied at time t, etc. These joint

probabilities are related to the individual occupancy probabilities through their marginal probabilities:

$$\left.\begin{aligned} C_\alpha(t) &= \mathbb{P}[A_\alpha(t), A_\beta(t)] + \mathbb{P}[A_\alpha(t), 0_\beta(t)];\\ 1 - C_\alpha(t) &= \mathbb{P}[0_\alpha(t), A_\beta(t)] + \mathbb{P}[0_\alpha(t), 0_\beta(t)];\\ C_\beta(t) &= \mathbb{P}[A_\alpha(t), A_\beta(t)] + \mathbb{P}[0_\alpha(t), A_\beta(t)];\\ 1 - C_\beta(t) &= \mathbb{P}[A_\alpha(t), 0_\beta(t)] + \mathbb{P}[0_\alpha(t), 0_\beta(t)] \end{aligned}\right\} \quad \text{(A 1)}$$

neighbouring occupancy probabilities are also related to the individual occupancy probabilities using the joint occupancy correlation function [9]:

$$F(t;\, \alpha,\, \beta) = \frac{\mathbb{P}[A_\alpha(t), A_\beta(t)]}{C_\alpha(t) C_\beta(t)}. \quad \text{(A 2)}$$

Note that if $F(t;\, \alpha,\, \beta) = 1$, then $\mathbb{P}[A_\alpha(t), A_\beta(t)] = C_\alpha(t)C_\beta(t)$, indicating that the occupancy of the neighbouring sites α and β are independent. We can combine equations (A 1) and (A 2) to write each neighbouring occupancy probability in terms of the individual occupancy probabilities and the occupancy correlation function

$$\left.\begin{aligned} &\mathbb{P}[A_\alpha(t), A_\beta(t)] = C_\alpha(t)C_\beta(t)F(t;\, \alpha,\, \beta);\\ &\mathbb{P}[A_\alpha(t), 0_\beta(t)] = C_\alpha(t)\Big(1 - C_\beta(t)F(t;\, \alpha,\, \beta)\Big);\\ &\mathbb{P}[0_\alpha(t), A_\beta(t)] = C_\beta(t)\Big(1 - C_\alpha(t)F(t;\, \alpha,\, \beta)\Big);\\ &\mathbb{P}[0_\alpha(t), 0_\beta(t)] = 1 - C_\alpha(t) - C_\beta(t) + C_\alpha(t)C_\beta(t)F(t;\, \alpha,\, \beta). \end{aligned}\right\} \quad \text{(A 3)}$$

We are now ready to convert the rules of the BDM process into a coarse-grained DE model. We begin by writing a master equation for how $C_\alpha(t)$ will change due to the effects of agent birth, death and migration:

$$\frac{\mathrm{d}C_\alpha(t)}{\mathrm{d}t} = K_{\text{birth}} + K_{\text{death}} + K_{\text{migration}}. \quad \text{(A 4)}$$

We now aim to derive expressions for K_{birth}, K_{death} and $K_{\text{migration}}$. The birth reaction from equation (2.1) specifies that the density at lattice site α may increase when α is unoccupied and $\beta \in \mathcal{B}(\alpha)$ is occupied because the agent at β may give birth and place its daughter agent in α. We can then write

$$K_{\text{birth}} = \frac{P_p}{4} \sum_{\beta \in \mathcal{B}(\alpha)} \mathbb{P}[0_\alpha(t), A_\beta(t)], \quad \text{(A 5)}$$

because any of the neighbouring lattice sites may undergo birth events. Similarly, we can convert equation (2.2) as

$$K_{\text{death}} = -P_d C_\alpha(t), \quad \text{(A 6)}$$

for agent death and convert equation (2.3) as

$$K_{\text{migration}} = \frac{P_m}{4} \sum_{\beta \in \mathcal{B}(\alpha)} (\mathbb{P}[0_\alpha(t), A_\beta(t)] - \mathbb{P}[A_\alpha(t), 0_\beta(t)]), \quad \text{(A 7)}$$

for agent migration. Substitution of these terms into equation (A 4) provides the master equation for the BDM process

$$\begin{aligned} \frac{\mathrm{d}C_\alpha(t)}{\mathrm{d}t} = &\frac{P_p}{4} \sum_{\beta \in \mathcal{B}(\alpha)} \big(\mathbb{P}[0_\alpha(t), A_\beta(t)]\big) - P_d C_\alpha(t)\\ &+ \frac{P_m}{4} \sum_{\beta \in \mathcal{B}(\alpha)} \big(\mathbb{P}[0_\alpha(t), A_\beta(t)] - \mathbb{P}[A_\alpha(t), 0_\beta(t)]\big). \end{aligned} \quad \text{(A 8)}$$

Equation (A 8) provides a DE model to describe the dynamics of $C_\alpha(t)$, however, this equation is not closed because we need to know $\mathbb{P}[A_\alpha(t), A_\beta(t)]$ in order to evaluate the right-hand side. We can use the marginal identities from equations (A 1) and (A 3) to simplify the terms in equation (A 8) and write

$$\begin{aligned} \frac{\mathrm{d}C_\alpha(t)}{\mathrm{d}t} = &\frac{P_m}{4} \sum_{\beta \in \mathcal{B}(\alpha)} (C_\beta(t) - C_\alpha(t)) + \frac{P_p}{4} \sum_{\beta \in \mathcal{B}(\alpha)} C_\beta(t)(1\\ &- C_\alpha(t)F(t;\, \alpha,\, \beta)) - P_d C_\alpha(t). \end{aligned} \quad \text{(A 9)}$$

We proceed by making a simplification in order to close this system. Since we initiate all simulations with agents distributed uniformly at random, we assume that all individual occupancy probabilities are equally distributed[1] on average so that $C_\alpha(t) = C_\gamma(t) = C(t)$ for any two lattice sites α and γ. From this assumption, we have that $F(t;\, \alpha,\, \beta) = F(t;\, |\alpha - \beta|)$, i.e. $F(t;\, \alpha,\, \beta) = F(t;\, 1)$ for $\beta \in \mathcal{B}(\alpha)$. From equation (A 2), we next write $\mathbb{P}[A_\alpha(t), A_\beta(t)] = C^2(t)F(t;\, 1)$ for $\beta \in \mathcal{B}(\alpha)$. These observations lead to the following DE model from equation (A 9):

$$\frac{\mathrm{d}}{\mathrm{d}t}C(t) = P_p C(t)(1 - C(t)F(t;\, 1)) - P_d C(t). \quad \text{(A 10)}$$

Equation (A10) is not yet closed because we still do not know $F(t;\, 1)$. Our second simplification is the *mean-field assumption*, in that the occupancy probabilities of neighbouring lattice sites are independent so that $F(t;\, 1) \equiv 1$. This assumption leads to the *mean-field model* for the ABM:

$$\frac{\mathrm{d}}{\mathrm{d}t}C(t) = P_p C(t)(1 - C(t)) - P_d C(t). \quad \text{(A 11)}$$

Note that equation (A 12) can be re-formulated as the standard logistic DE model given by

$$\frac{\mathrm{d}}{\mathrm{d}t}C(t) = rC(t)\left(1 - \frac{C(t)}{K}\right), \quad \text{(A 12)}$$

where $r = P_p - P_d$, $K = (P_p - P_d)/P_p$. This model is advantageous in that it is closed and can be solved analytically:

$$C(t) = \frac{KC(0)\mathrm{e}^{rt}}{K + C(0)(\mathrm{e}^{rt} - 1)}, \quad \text{(A 13)}$$

where $C(0)$ denotes the initial condition.

Appendix B. Coarse graining susceptible–infected–recovered rules into a differential equation model

We now derive DE models governing the dynamics for $\mathbb{P}[S_\alpha(t)]$, $\mathbb{P}[I_\alpha(t)]$ and $\mathbb{P}[R_\alpha(t)]$. As for the BDM model, because the initial agent configurations are uniformly distributed in space, we assume the probability of any type of agent occupancy (S, I or R) is independent of the lattice site and define $S(t) = \mathbb{P}[S_\alpha(t)]$, $I(t) = \mathbb{P}[I_\alpha(t)]$, $R(t) = \mathbb{P}[R_\alpha(t)]$. By converting the bimolecular reactions in equations (2.8) and (2.9) into the corresponding occupancy probability configurations that will lead to changes in S, I or R, and converting the monomolecular reaction in equation (2.10) into the individual occupancy probabilities that will lead to changes in S, I or R, we derive the master system of equations for $S(t)$, $I(t)$ and $R(t)$

to be:

$$\left.\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t}S(t) &= \sum_{\beta\in B(\alpha)}\left[\frac{P_m}{4}\mathbb{P}[0_\alpha(t), S_\beta(t)] - \frac{P_m}{4}\mathbb{P}[S_\alpha(t), 0_\beta(t)] - \frac{P_I}{4}\mathbb{P}[S_\alpha(t), I_\beta(t)]\right];\\
\frac{\mathrm{d}}{\mathrm{d}t}I(t) &= \sum_{\beta\in B(\alpha)}\left[\frac{P_m}{4}\mathbb{P}[0_\alpha(t), I_\beta(t)] - \frac{P_m}{4}\mathbb{P}[I_\alpha(t), 0_\beta(t)] + \frac{P_I}{4}\mathbb{P}[I_\alpha(t), S_\beta(t)]\right]\\
&\quad - RC_I(t);\\
\frac{\mathrm{d}}{\mathrm{d}t}R(t) &= \sum_{\beta\in B(\alpha)}\left[\frac{P_m}{4}\mathbb{P}[0_\alpha(t), I_\beta(t)] - \frac{P_m}{4}\mathbb{P}[I_\alpha(t), 0_\beta(t)]\right] + P_R I(t).
\end{aligned}\right\} \tag{B 1}$$

We then use the mean-field assumption to write $\mathbb{P}[Y_\alpha(t), Z_\beta(t)] = Y(t)Z(t)$, where $Y,\ Z \in \{S,\ I,\ R,\ 0\}$. This assumption reduces equation (B 1) to the commonly used SIR model given by:

$$\frac{\mathrm{d}S}{\mathrm{d}t} = -P_I SI; \quad \frac{\mathrm{d}I}{\mathrm{d}t} = P_I SI - P_R I; \quad \frac{\mathrm{d}R}{\mathrm{d}t} = P_R I. \tag{B 2}$$

In equation (B 2), the variables S, I and R denote the density of susceptible, infected and recovered agents over time, respectively, which cannot exceed 0.5 if only half of the simulation domain is occupied by agents. We can convert these variables to the fraction of susceptible, infected and recovered agents by computing the dimensionless variables $S^*(t) = S(t)/M$, $I^*(t) = I(t)/M$ and $R^*(t) = R(t)/M$, where M is the proportion of occupied lattice sites in the simulation domain. The system of equations for these variables are given by

$$\begin{aligned}
\frac{\mathrm{d}S^*}{\mathrm{d}t} &= -MP_I S^* I^*,\\
\frac{\mathrm{d}I^*}{\mathrm{d}t} &= MP_I S^* I^* - P_R I^*, \quad \frac{\mathrm{d}R^*}{\mathrm{d}t} = P_R I^*.
\end{aligned} \tag{B 3}$$

Appendix C. Gillespie algorithm

Algorithm 1: Gillespie algorithm for the BDM process (modified from [12]).

Algorithm 1: Gillespie algorithm for the BDM process (modified from [23]

Create $X \times X$ lattice with user-specified placement of agents;

Maximum lattice occupancy is given by $N = X^2$

Set $t = 0$; Set maximum simulation time t_{end};

Set $C(t)$ equal to to number of agents on the lattice;

while $t < t_{end}$ *and* $C(t) < N$ **do**

  Randomly choose an agent and determine its lattice site;

  Calculate the propensity function $a(t) = (P_p + P_m + P_d)C(t)$;

  Calculate the following random variables, uniformly distributed on $[0, 1] : \gamma_1, \gamma_2$;

  Calculate time step $\tau = -\ln(\gamma_1)/a(t)$;

  $t = t + \tau$;

  $C(t) = C(t - \tau)$;

  $R = a(t)\gamma_2$;

  if $R < P_p C(t)$ **then**

    Choose adjacent lattice site with equal probability 1/4;

    if *chosen lattice site is empty* **then**

      Place new agent to chosen lattice site;

      $C(t) = C(t) + 1$;

    end

  else if $R \in ((P_p C(t), (P_p + P_m)C(t)]$ **then**

    Choose adjacent lattice site with equal probability 1/4;

    if *chosen lattice site is empty* **then**

      Move agent to chosen lattice site;

    end

  else if $R \in ((P_p + P_m)C(t), (P_p + P_m + P_d)C(t)]$ **then**

    Remove agent from lattice site;

    $C(t) = C(t) - 1$;

  end

end

Appendix D. Lasso algorithm using FISTA

Algorithm 2: Lasso implementation using FISTA from [53].

Algorithm 2: Lasso implementation using FISTA from [4].

Input: Matrix A, vector b, and regularization parameter λ

Output: $x = \frac{1}{2}\arg\min \|Ax - b\|_2^2 + \lambda\|x\|_1$.

Get $d = \#$ columns in A.

for $i \leftarrow 0$ **to** d **do**

  Set $a_i = \|A[:, i]\|_2$.

  Set $A[:, i] = A[:, i]/a_i$.

end

Set $L = \|A^T A\|_2$ (the largest singular value of A), $w = \vec{0}$, $w_{\text{old}} = \vec{0}$.

while $iter < iter_{\max}$ **do**

  $z = \frac{iter}{iter + 1}(w - w_{\text{old}})$

  $w_{\text{old}} = w$

  $z = z - A^T(A^T z - b)/L$

  for $i \leftarrow 0$ **to** d **do**

    $w[i] = (z[i]) \times (\max\{|z[i] - \lambda/L|, 0\})$

  end

end

for $i \leftarrow 0$ **to** d **do**

  Set $w[i] = w[i] \times a_i$.

end

Appendix E. Hyperparameter selection for Lasso

This section provides brief notes about the practical selection of hyperparameters for the Lasso method. We used regularization parameter $\lambda = 0.0004$ for the Lasso algorithm to learn an equation for $\langle C_{\text{ABM}}(t)\rangle$ in the tutorial in §3.1.1. The optimal value of this hyperparameter is typically not known *a priori*. There are several ways to select such a hyperparameter, including a grid search [24], cross validation [69] or Bayesian optimization [70]. We discuss the grid search option here due to its simplicity. In a grid search to determine an appropriate value for λ, we specify several plausible options, given by $\{\lambda_1, \lambda_2, \ldots, \lambda_n\}$ and split the data into training ($\mathrm{d}C_d^{\text{train}}(t)/\mathrm{d}t$ and Θ^{train}) and testing ($\mathrm{d}C_d^{\text{test}}(t)/\mathrm{d}t$ and Θ^{test}) sets. The training and testing portions of Θ will contain all columns of Θ but only a subset of the rows. For a possible hyperparameter value, λ_i, we solve the Lasso problem from equation (3.8) using the training data and $\lambda = \lambda_i$ to determine the resulting ξ estimates, $\hat{\xi}_i$. The optimal value of λ is then chosen as:

$$\hat{\lambda} = \arg\min_{\lambda_i} \left\| \frac{\mathrm{d}C_d^{\text{test}}(t)}{\mathrm{d}t} - \Theta^{\text{test}}\xi_i \right\|_2. \tag{E 1}$$

The optimal value $\hat{\lambda}$ results in the estimate $\hat{\xi}$ that best generalizes to the testing data. Recent work has shown that for Lasso, $\hat{\xi}$ tends to incorporate small additional terms in the final learned equation [24]. To select the regularization parameter, λ, for the Lasso algorithm, we randomly split half of $\langle C_{\text{ABM}}(t)\rangle$ into a training set and the remaining into a testing set. We considered 100 values of λ between 10^{-5} and 10^{-3} as well as $\lambda = 0$ and solved equation (3.8) using the training set for all 101 potential values of λ. We can perform this operation several times (changing the training and testing set each time) and notice that sometimes the chosen hyperparameter is zero and sometimes it is non-zero. When the chosen hyperparameter is zero, then the EQL pipeline learns an equation of the form

$$\frac{\mathrm{d}C}{\mathrm{d}t} = 0.00488C - 0.01187C^2 + 0.00806C^3 - 0.00831C^4, \tag{E 2}$$

whereas when the chosen hyperparameter is non-zero, then the EQL pipeline learns an equation of the form

$$\frac{\mathrm{d}C}{\mathrm{d}t} = 0.00468C - 0.00951C^2. \tag{E 3}$$

To ensure the final learned equation is sensitive to changes in each non-zero library coefficient, we perform a round of pruning after learning which proceeds as follows. The jth non-zero term of $\hat{\xi}$ is included in the final inferred equation if $\|\mathrm{d}C_d^{\text{test}}(t)/\mathrm{d}t - \Theta^{\text{test}}\hat{\xi}_j\|_2^2$ increases by a given pruning percentage, where here $\hat{\xi}_j$ is the estimated parameter vector with the jth term manually set to zero.

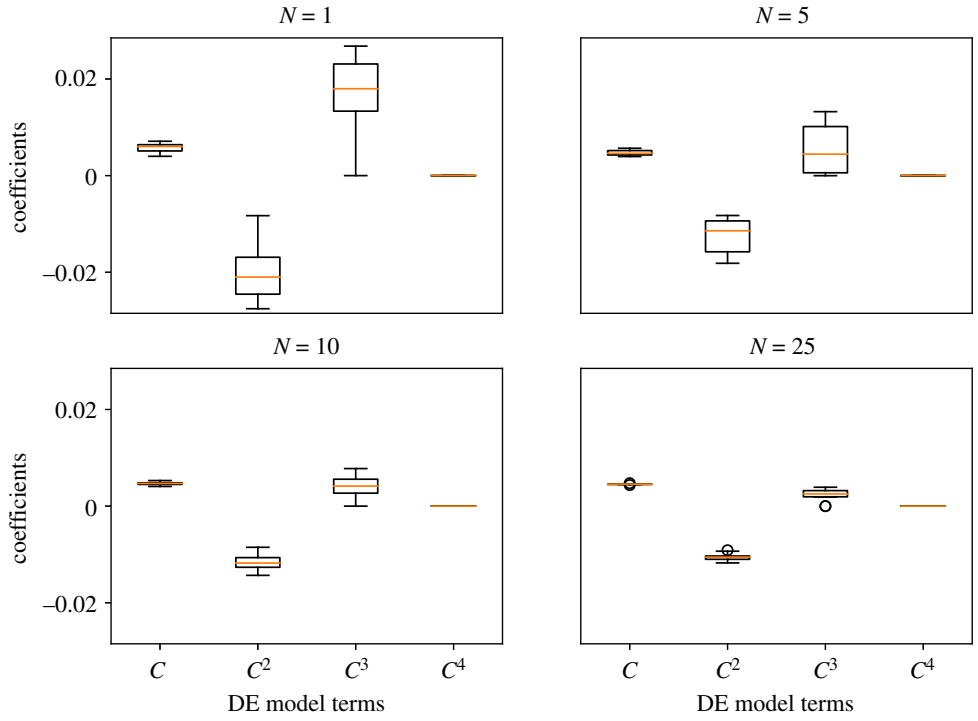

Figure 11. Summaries of the distributions of model parameters for the learned DE models from Case study 2 over various values of N. Each box and whisker plot summarizes the distribution of coefficient estimates from ten realizations of $\langle C_{\mathrm{ABM}}(t)\rangle$ for various values of N. In each box and whisker plot, the lower line of the box portion provides the 25% quartile of the data and the upper line denotes the 75% quartile. The orange line on each box plot denotes the median coefficient value. The length of the upper and lower whiskers are 1.5 times the interquartile range of the distribution, and dots denote outlier points.

To ensure that the final learned equation is not an artefact of the training and testing split, we perform this entire process for 10 randomized training and testing splits of the data and select the equation form arises most frequently. We set the parameters for each coefficient to be the mean of each coefficient for each time the final equation form was learned. If we set our pruning percentage to be 5%, then the majority of learned equations will be of the form

$$\frac{\mathrm{d}C}{\mathrm{d}t} = 0.00468C - 0.00951C^2. \tag{E 4}$$

Appendix F. Case study 2: parameter distributions

See figure 11.

Endnote

[1]This assumption is not applicable when the initial configuration is spatially heterogeneous. See [36,68] for the derivation of PDE models in this scenario.

References

1. Couzin ID, Krause J, Franks NR, Levin SA. 2005 Effective leadership and decision-making in animal groups on the move. *Nature* **433**, 513–516. (doi:10.1038/nature03236)
2. Schmidt S, Friedl P. 2009 Interstitial cell migration: integrin-dependent and alternative adhesion mechanisms. *Cell Tissue Res.* **339**, 83. (doi:10.1007/s00441-009-0892-9)
3. Binny RN, Law R, Plank. Living in groups MJ. 2020 spatial-moment dynamics with neighbour-biased movements. *Ecol. Modell.* **415**, 108825. (doi:10.1016/j.ecolmodel.2019.108825)
4. D'Orsogna MR, Chuang YL, Bertozzi AL, Chayes LS. 2006 Self-propelled particles with soft-core interactions: patterns, stability, and collapse. *Phys. Rev. Lett.* **96**, 104302. (doi:10.1103/PhysRevLett.96.104302)
5. An G *et al.* 2017 Optimization and control of agent-based models in biology: a perspective. *Bull. Math. Biol.* **79**, 63–87. (doi:10.1007/s11538-016-0225-6)
6. Mirams GR *et al.* 2013 Chaste: an open source C++ library for computational physiology and biology. *PLoS Comput. Biol.* **9**, e1002970. (doi:10.1371/journal.pcbi.1002970)
7. Seber GAF, Wild CJ. 1988 *Nonlinear regression*. Wiley Series in Probability and Statistics. Hoboken, NJ: John Wiley & Sons, Inc.
8. Beheshti R, Sukthankar G. 2013 Improving Markov chain Monte Carlo estimation with agent-based models. In *Social computing, behavioral-cultural modeling and prediction* (eds AM Greenberg, WG Kennedy, ND Bos), Lecture Notes in Computer Science, pp. 495–502. Berlin, Germany: Springer.
9. Baker RE, Simpson MJ. 2010 Correcting mean-field approximations for birth-death-movement processes. *Phys. Rev. E* **82**, 041905. (doi:10.1103/PhysRevE.82.041905)
10. Chaplain MAJ, Lorenzi T, Macfarlane FR. 2020 Bridging the gap between individual-based and continuum models of growing cell populations. *J. Math. Biol.* **80**, 343–371. (doi:10.1007/s00285-019-01391-y)
11. Cruz R d. l., Guerrero P, Spill F, Alarcón T. 2016 Stochastic multi-scale models of competition within heterogeneous cellular populations: simulation methods and mean-field analysis. *J. Theor. Biol.* **407**, 161–183. (doi:10.1016/j.jtbi.2016.07.028)
12. Fadai NT, Baker RE, Simpson MJ. 2019 Accurate and efficient discretizations for stochastic models providing near agent-based spatial resolution at low computational cost. *J. R. Soc. Interface* **16**, 20190421. (doi:10.1098/rsif.2019.0421)

13. Schnoerr D, Sanguinetti G, Grima R. 2017 Approximation and inference methods for stochastic biochemical kinetics—a tutorial review. *J. Phys. A: Math. Theor.* **50**, 093001. (doi:10.1088/1751-8121/aa54d9)
14. Murray JD. 1984 *Asymptotic analysis*. New York, NY: Science & Business Media.
15. Murray JD. 2002 *Mathematical biology I. An introduction*, vol. 17, 3rd edn. Interdisciplinary Applied Mathematics. Berlin, Germany: Springer.
16. Bernoff AJ, Topaz CM. 2013 Nonlocal aggregation models: a primer of swarm equilibria. *SIAM Rev.* **55**, 709–747. (doi:10.1137/130925669)
17. Bernoff AJ, Culshaw-Maurer M, Everett RA, Hohn ME, Strickland WC, Weinburd J. 2020 Agent-based and continuous models of hopper bands for the Australian plague locust: how resource consumption mediates pulse formation and geometry. *PLoS Comput. Biol.* **16**, e1007820. (doi:10.1371/journal.pcbi.1007820)
18. Dkhili J, Berger U, Idrissi Hassani LM, Ghaout S, Peters R, Piou C. 2017 Self-organized spatial structures of locust groups emerging from local interaction. *Ecol. Modell* **361**, 26–40. (doi:10.1016/j.ecolmodel.2017.07.020)
19. Topaz CM, D'Orsogna MR, Edelstein-Keshet L, Bernoff AJ. 2012 Locust dynamics: behavioral phase change and swarming. *PLoS Comput. Biol.* **8**, e1002642. (doi:10.1371/journal.pcbi.1002642)
20. Matsiaka OM, Penington CJ, Baker RE, Simpson MJ. 2017 Continuum approximations for lattice-free multi-species models of collective cell migration. *J. Theor. Biol.* **422**, 1–11. (doi:10.1016/j.jtbi.2017.04.009)
21. Gallaher J, Anderson ARA. 2013 Evolution of intratumoral phenotypic heterogeneity: the role of trait inheritance. *Interface Focus* **3**, 20130016. (doi:10.1098/rsfs.2013.0016)
22. West J, Hasnain Z, Macklin P, Newton PK. 2016 An evolutionary model of tumor cell kinetics and the emergence of molecular heterogeneity driving Gompertzian growth. *SIAM Rev.* **58**, 716–736. (doi:10.1137/15M1044825)
23. Brunton SL, Proctor JL, Kutz JN. 2016 Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proc. Natl Acad. Sci. USA* **113**, 3932–3937. (doi:10.1073/pnas.1517384113)
24. Lagergren JH, Nardini JT, Michael Lavigne G, Rutter EM, Flores KB. 2020 Learning partial differential equations for biological transport models from noisy spatio-temporal data. *Proc. R. Soc. A* **476**, 20190800. (doi:10.1098/rspa.2019.0800)
25. Rudy SH, Brunton SL, Proctor JL, Kutz JN. 2017 Data-driven discovery of partial differential equations. *Sci. Adv.* **3**, e1602614. (doi:10.1126/sciadv.1602614)
26. Zhang S, Lin G. 2018 Robust data-driven discovery of governing physical laws with error bars. *Proc. R. Soc. A* **474**, 20180305. (doi:10.1098/rspa.2018.0305)
27. Zhang S, Lin G. 2019 Robust subsampling-based sparse Bayesian inference to tackle four challenges (large noise, outliers, data integration, and extrapolation) in the discovery of physical laws from data. (http://arxiv.org/abs/1907.07788)
28. Lagergren JH, Nardini JT, Baker RE, Simpson MJ, Flores KB. 2020 Biologically-informed neural networks guide mechanistic modeling from sparse experimental data. (http://arxiv.org/abs/2005.13073)
29. Bonabeau E. 2002 Agent-based modeling: methods and techniques for simulating human systems. *Proc. Natl Acad. Sci. USA* **99**(Suppl. 3), 7280–7287. (doi:10.1073/pnas.082080899)
30. Cosgrove J, Butler J, Alden K, Read M, Kumar V, Cucurull-Sanchez L, Timmis J, Coles M. 2015 Agent-based modeling in systems pharmacology. *CPT Pharmacometrics Syst. Pharmacol.* **4**, 615–629. (doi:10.1002/psp4.12018)
31. Interian R, Rodríguez-Ramos R, Valdés-Ravelo F, Ramírez-Torres A, Ribeiro C, Conci A. 2017 Tumor growth modelling by cellular automata. *Math. Mech. Complex Syst.* **5**, 239–259. (doi:10.2140/memocs.2017.5.239)
32. Stevens A. 2000 A stochastic cellular automaton modeling gliding and aggregation of myxobacteria. *SIAM J. Appl. Math.* **61**, 172–182. (doi:10.1137/S0036139998342053)
33. Stevens A, Othmer H. 1997 Aggregation, blowup, and collapse: the ABC's of taxis in reinforced random walks. *SIAM J. Appl. Math.* **57**, 1044–1081. (doi:10.1137/S0036139995288976)
34. Anguige K, Schmeiser C. 2009 A one-dimensional model of cell diffusion and aggregation, incorporating volume filling and cell-to-cell adhesion. *J. Math. Biol.* **58**, 395. (doi:10.1007/s00285-008-0197-8)
35. Johnston ST, Simpson MJ, McElwain DLS. 2014 How much information can be obtained from tracking the position of the leading edge in a scratch assay?. *J. R. Soc. Interface* **11**, 20140325. (doi:10.1098/rsif.2014.0325)
36. Simpson MJ, Landman KA, Hughes BD. 2009 Multi-species simple exclusion processes. *Physica A* **388**, 399–406. (doi:10.1016/j.physa.2008.10.038)
37. Cianci C, Smith S, Grima R. 2016 Molecular finite-size effects in stochastic models of equilibrium chemical systems. *J. Chem. Phys.* **144**, 084101. (doi:10.1063/1.4941583)
38. Jin W, McCue SW, Simpson MJ. 2018 Extended logistic growth model for heterogeneous populations. *J. Theor. Biol.* **445**, 51–61. (doi:10.1016/j.jtbi.2018.02.027)
39. Johnston ST, Simpson MJ, Baker RE. 2012 Mean-field descriptions of collective migration with strong adhesion. *Phys. Rev. E* **85**, 051922. (doi:10.1103/PhysRevE.85.051922)
40. McKane AJ, Newman TJ. 2004 Stochastic models in population biology and their deterministic analogs. *Phys. Rev. E* **70**, 041902. (doi:10.1103/PhysRevE.70.041902)
41. Gillespie DT. 1977 Exact stochastic simulation of coupled chemical reactions. *J. Phys. Chem. A* **81**, 2340–2361. (doi:10.1021/j100540a008)
42. Simpson MJ, Binder BJ, Haridas P, Wood BK, Treloar KK, McElwain DLS, Baker RE. 2013 Experimental and modelling investigation of monolayer development with clustering. *Bull. Math. Biol.* **75**, 871–889. (doi:10.1007/s11538-013-9839-0)
43. Hiebeler D. 1997 Stochastic spatial models: from simulations to mean field and local structure approximations. *J. Theor. Biol.* **187**, 307–319. (doi:10.1006/jtbi.1997.0422)
44. Middleton AM, Fleck C, Grima R. 2014 A continuum approximation to an off-lattice individual-cell based model of cell migration and adhesion. *J. Theor. Biol.* **359**, 220–232. (doi:10.1016/j.jtbi.2014.06.011)
45. Newman TJ, Grima R. 2004 Many-body theory of chemotactic cell-cell interactions. *Phys. Rev. E* **70**, 051916. (doi:10.1103/PhysRevE.70.051916)
46. Simpson MJ, Sharp JA, Baker RE. 2014 Distinguishing between mean-field, moment dynamics and stochastic descriptions of birth–death–movement processes. *Physica A* **395**, 236–246. (doi:10.1016/j.physa.2013.10.026)
47. Blackwood J, Childs L. 2018 An introduction to compartmental modeling for the budding infectious disease modeler. *Lett. Biomathematics* **5**, 195–221. (doi:10.30707/LiB5.1Blackwood)
48. Eisenberg MC, Robertson SL, Tien JH. 2013 Identifiability and estimation of multiple transmission pathways in cholera and waterborne disease. *J. Theor. Biol.* **324**, 84–102. (doi:10.1016/j.jtbi.2012.12.021)
49. LeVeque RJ. 2007 *Finite difference methods for ordinary and partial differential equations: steady-state and time-dependent problems*. Philadelphia, PA: Society for Industrial and Applied Mathematics.
50. Simpson MJ, Landman KA, Hughes BD. 2010 Cell invasion with proliferation mechanisms motivated by time-lapse data. *Physica A* **389**, 3779–3790. (doi:10.1016/j.physa.2010.05.020)
51. Tibshirani R. 1996 Regression shrinkage and selection via the Lasso. *J. R. Stat. Soc.: Ser. B (Methodological)* **58**, 267–288. (doi:10.1111/j.2517-6161.1996.tb02080.x)
52. Zhang T. 2009 Adaptive forward-backward greedy algorithm for sparse learning with linear models. In *Advances in neural information processing systems* (eds D Koller, D Schuurmans, Y Bengio, L Bottou), pp. 1921–1928. Cambridge, MA: MIT Press.
53. Beck A, Teboulle M. 2009 A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imaging Sci.* **2**, 183–202. (doi:10.1137/080716542)
54. Grima R. 2010 An effective rate equation approach to reaction kinetics in small volumes: theory and application to biochemical reactions in nonequilibrium steady-state conditions. *J. Chem. Phys.* **133**, 035101. (doi:10.1063/1.3454685)
55. Smith S, Cianci C, Grima R. 2016 Analytical approximations for spatial stochastic gene expression in single cells and tissues. *J. R. Soc. Interface* **13**, 20151051. (doi:10.1098/rsif.2015.1051)
56. Nardini JT, Lagergren JH, Hawkins-Daarud A, Curtin L, Morris B, Rutter EM, Swanson KR, Flores KB. 2020

Learning equations from biological data with limited time samples. *Bull. Math. Biol.* **82**, 119. (doi:10.1007/s11538-020-00794-z)

57. Bortz DM, Nelson PW. 2006 Model selection and mixed-effects modeling of HIV infection Dynamics. *Bull. Math. Biol.* **68**, 2005–2025. (doi:10.1007/s11538-006-9084-x)
58. Diekmann O, Heesterbeek JAP, Metz JAJ. 1990 On the definition and the computation of the basic reproduction ratio R0 in models for infectious diseases in heterogeneous populations. *J. Math. Biol.* **28**, 365–382. (doi:10.1007/BF00178324)
59. Viceconte G, Petrosillo N. 2020 COVID-19 R0: magic number or conundrum? *Current Infectious Disease Reports* **12**. 8516–8517. (doi:10.4081/idr.2020.8516)
60. Johnston ST, Baker RE, McElwain DLS, Simpson MJ. 2017 Co-operation, competition and crowding: a discrete framework linking allee kinetics, nonlinear diffusion, shocks and sharp-fronted travelling waves. *Sci. Rep.* **7**, 42134. (doi:10.1038/srep42134)
61. Nardini JT, Chapnick DA, Liu X, Bortz DM. 2016 Modeling keratinocyte wound healing: cell-cell adhesions promote sustained migration. *J. Theor. Biol.* **400**, 103–117. (doi:10.1016/j.jtbi.2016.04.015)
62. Hamilton F, Lloyd AL, Flores KB. 2017 Hybrid modeling and prediction of dynamical systems. *PLoS Comput. Biol.* **13**, e1005655. (doi:10.1371/journal.pcbi.1005655)
63. Lagergren J, Reeder A, Hamilton F, Smith RC, Flores KB. 2018 Forecasting and uncertainty quantification using a hybrid of mechanistic and non-mechanistic models for an age-structured population model. *Bull. Math. Biol.* **80**, 1578–1595. (doi:10.1007/s11538-018-0421-7)
64. Smith RC. 2013 *Uncertainty quantification: theory, implementation, and applications*. Computational science and engineering series. Philadelphia, PA: Society for Industrial and Applied Mathematics.
65. Klus S, Nüske F, Peitz S, Niemann J-H, Clementi C, Schütte C. 2020 Data-driven approximation of the Koopman generator: model reduction, system identification, and control. *Physica D* **406**, 132416. (doi:10.1016/j.physd.2020.132416)
66. Jiang Q, Fu X, Yan S, Li R, Du W, Cao Z, Qian F, Grima R. 2020 Neural network aided approximation and parameter inference of stochastic models of gene expression. *bioRxiv* 12.15.422883.
67. Walker DC, Hill G, Wood SM, Smallwood RH, Southgate J. 2004 Agent-based computational modeling of wounded epithelial cell monolayers. *IEEE Trans. Nanobioscience* **3**, 153–163. (doi:10.1109/TNB.2004.833680)
68. Deutsch A, Dormann S. 2005 *Cellular automaton modeling of biological pattern formation*, 2nd edn. Boston, MA: Modeling and Simulation in Science, Engineering and Technology.
69. Mangan NM, Kutz JN, Brunton SL, Proctor JL. 2017 Model selection for dynamical systems via sparse regression and information criteria. *Proc. R. Soc. A* **473**, 20170009. (doi:10.1098/rspa.2017.0009)
70. Snoek J, Larochelle H, Adams RP. 2012 Practical Bayesian optimization of machine learning algorithms. In *Advances in neural information processing systems* (eds F Pereira, CJC Burges, L Bottou, KQ Weinberger), vol. 25, pp. 2951–2959. Cambridge, MA: MIT Press.