Bulletin of Mathematical Biology (2024) 86:130
https://doi.org/10.1007/s11538-024-01357-2

Society for Mathematical Biology

ORIGINAL ARTICLE

Forecasting and Predicting Stochastic Agent-Based Model Data with Biologically-Informed Neural Networks

John T. Nardini[1]

Received: 11 April 2024 / Accepted: 2 September 2024 / Published online: 23 September 2024

Abstract

Collective migration is an important component of many biological processes, including wound healing, tumorigenesis, and embryo development. Spatial agent-based models (ABMs) are often used to model collective migration, but it is challenging to thoroughly predict these models' behavior throughout parameter space due to their random and computationally intensive nature. Modelers often coarse-grain ABM rules into mean-field differential equation (DE) models. While these DE models are fast to simulate, they suffer from poor (or even ill-posed) ABM predictions in some regions of parameter space. In this work, we describe how biologically-informed neural networks (BINNs) can be trained to learn interpretable BINN-guided DE models capable of accurately predicting ABM behavior. In particular, we show that BINN-guided partial DE (PDE) simulations can (1) forecast future spatial ABM data not seen during model training, and (2) predict ABM data at previously-unexplored parameter values. This latter task is achieved by combining BINN-guided PDE simulations with multivariate interpolation. We demonstrate our approach using three case study ABMs of collective migration that imitate cell biology experiments and find that BINN-guided PDEs accurately forecast and predict ABM data with a one-compartment PDE when the mean-field PDE is ill-posed or requires two compartments. This work suggests that BINN-guided PDEs allow modelers to efficiently explore parameter space, which may enable data-driven tasks for ABMs, such as estimating parameters from experimental data. All code and data from our study is available at https://github.com/johnnardini/Forecasting_predicting_ABMs.

Keywords Agent-based modeling · Machine learning · Differential equations · Data-driven modeling

✉ John T. Nardini
nardinij@tcnj.edu

[1] Department of Mathematics and Statistics, The College of New Jersey, Ewing, NJ 08628, USA

1 Introduction

Many population-level patterns in biology arise from the actions of individuals. For example, predator–prey interactions determine ecological population dynamics; individuals' adherence to public health policies limit disease spread; and cellular interactions drive wound healing and tumor invasion. Mathematical modeling is a useful tool to understand and predict how such individual actions scale into collective behavior (Anguige and Schmeiser 2009; Brauer et al. 2019; Theo et al. 2022; Huppert and Katriel 2013; Nardini et al. 2016; Shahzeb et al. 2023; Xiao and Chen 2001). In particular, stochastic agent-based models (ABMs) are a widely-used modeling framework where autonomous agents mimic the individuals of a population (Baker and Simpson 2010; Volker et al. 2005; Marshall and Galea 2015; Melissa et al. 2018). ABMs are advantageous because they capture the discrete and stochastic nature of many biological processes (Chappelle and Yates 2019). However, ABMs are computationally intensive, and their simulations can become time-consuming to perform when the population is comprised of many individuals (Simpson et al. 2022; Nardini et al. 2021). This computational restraint prevents modelers from efficiently exploring how model parameters alter model outputs. As such, there is a need for the development of methods to efficiently and accurately predict ABM behavior (Nardini et al. 2021; Kieu et al. 2020; Larie et al. 2021).

Modelers often perform ABM prediction by coarse-graining ABM rules into continuous differential equation (DE) models (Baker and Simpson 2010; Simpson et al. 2022). Ordinary DEs (ODEs) describe how a quantity (e.g., agent density) changes over time, and Partial DEs (PDEs) describe how spatially-varying ABMs change with time (Simpson et al. 2022). Such DE models are useful surrogates for ABMs because they are cheap and efficient to simulate. Mean-field DE models, which assume agents respond to the average behavior of their neighbors, have been shown to accurately predict ABM behavior at some parameter values. Unfortunately, these models can poorly predict ABM outputs when the mean-field assumption is violated (Baker and Simpson 2010; Thompson et al. 2012). For example, Baker and Simpson (2010) demonstrated that the mean-field DE model for a population growth ABM only accurately predict ABM data when agents proliferate slowly. A further complication of mean-field DEs is that they may be ill-posed at certain parameter values. Anguige and Schmeiser (2009) used a stochastic space-jump model to study how cell adhesion impacts collective migration and found that the resulting mean-field PDE model is ill-posed (and thus cannot predict ABM behavior) for large adhesion values.

Despite the inability of mean-field DE models to predict ABM behavior at all parameter values, ABM simulations do obey conservation laws (e.g., conservation of mass for spatial ABMs) (VandenHeuvel et al. 2024). Alternative DE models may thus be capable of accurately describing ABM behavior. Equation learning (EQL) is a new area of research on the development and application of algorithms to discover the dynamical systems model that best describes a dataset (Brunton et al. 2016; Kaiser et al. 2018; Rudy et al. 2019; Champion et al. 2019; Mangan et al. 2016, 2017; Messenger and Bortz 2021a, b; Lagergren et al. 2020a, b; Nardini et al. 2020). Brunton et al. 2016 Brunton et al. (2016) introduced a sparse regression-based EQL approach to learn DE models from data with a user-specified library of candidate terms. This method has

proven very successful in recovering informative models from simulated and experimental data (Rudy et al. 2017). There is a growing understanding that EQL methods can aid the prediction of ABM data (Nardini et al. 2021; Messenger et al. 2022; Messenger and Bortz 2022; Supekar et al. 2023). For example, we recently demonstrated that the least squares EQL approach learns ODE equations that accurately describe simulated ABM data, even when the collected data is incomplete or sparsely sampled (Nardini et al. 2021). Supekar et al. (2023) coupled this method with spectral basis representation data to discover PDE models that capture the emergent behavior found in active matter ABMs. Another popular EQL approach includes physics-informed neural networks (PINNs), where modelers embed physical knowledge (in the form of a known PDE framework) into the training procedure for artificial neural networks (ANNs) (Cai et al. 2021; Kaplarević-Malisić et al. 2023; Linka et al. 2022; Raissi et al. 2019; Shin et al. 2020). Trained PINN models can predict complex, sparse, and noisy data while also obeying known physical principles. Lagergren et al. (2020b) extended the PINNs framework by replacing physics-based mechanistic terms with function-approximating multi-layer perceptions (MLPs) to develop the biologically-informed neural network (BINN) methodology. As a result, BINN models can learn PDE models from data with terms that depend on space, time, or agent density. Training the BINN to simulated ABM data ensures that a realization of this PDE that best matches the data is learned. Standard methods of DE analysis, including bifurcation analysis and pattern formation, can be used to understand the ABM's behavior. BINNs thus present a promising and interpretable tool for ABM forecasting and prediction. However, determining how BINNs can be used to learn predictive DE models for ABMs remains an open area of research.

In this work, we demonstrate how to combine BINNs and PDE model simulations to forecast and predict ABM behavior. Our approach leverages BINNs' vast data and modeling approximation capability with the computational efficiency of PDE models to develop a potent ABM surrogate modeling tool. In particular, we demonstrate how to use trained BINN models to (1.) forecast future ABM data at a fixed parameter value, and (2.) predict ABM data at previously-unexplored parameter values. This latter task is achieved using multivariate interpolation, which provides a straightforward approach for inferring PDE modeling terms. We demonstrate that visually inspecting the BINN modeling terms over a range of ABM parameter values allows us to interpret how ABM parameters impact model behavior.

We apply the BINNs methodology to three case study ABMs in this work. Each case study models collective migration in cell biological experiments, such as barrier and scratch assays (Nardini et al. 2016; Simpson et al. 2022; Johnston et al. 2012; Lagergren et al. 2020b; Decaestecker et al. 2007). In a barrier assay, a two-dimensional layer of cells is cultured inside a physical boundary. Microscopy is used to image how the cell population migrates outwards once the barrier has been removed (Decaestecker et al. 2007; Das et al. 2015). Cells are closely packed in these experiments and thus interact with their neighbors. Our case study ABMs simulate how two stimuli, namely, cell pulling and adhesion, impact collectively migrating cell populations. These processes are ubiquitous in cell biology. For example, leader cells pull their followers into the wound area to heal wounded epithelial tissue, and cell adhesions in embryonic cells ensures the self organization of the different germ layers (Kashef and Franz 2015;

Venhuizen and Zegers 2017; Vishwakarma et al. 2020). ABMs provide a promising avenue to model the impacts of these stimuli on collectively migrating cell populations.

We begin this work in Sect. 2 by presenting the case study ABMs and notation. In Sect. 3, we discuss our methodologies to forecast and predict ABM behavior. In Sect. 4, we detail our results on using these methods to forecast and predict data from the three case study ABMs; this section concludes with a brief discussion on the computational expenses of each method. We conclude these results and suggest areas for future work in Sect. 5.

2 The Case Study ABMs

We consider three case study ABMs that imitate collective migration during cell biological experiments, including scratch and barrier assays (Nardini et al. 2016; Simpson et al. 2022; Johnston et al. 2012; Lagergren et al. 2020b; Decaestecker et al. 2007). Each case study ABM models how cell pulling and adhesion impact collective cell migration during these experiments (Janiszewska et al. 2020; Rothenberg et al. 2023). The ABMs are two-dimensional cellular automata with pulling agents that perform cell pulling rules and/or adhesive agents that perform rules on cell adhesion. Each model is an exclusion process, meaning that each agent can only occupy one lattice site at a time, and each lattice site is occupied by at most one agent. The first model is borrowed from Chappelle and Yates (2019) and consists only of pulling agents; the second model is inspired by the stochastic space jump model from Anguige and Schmeiser (2009) and consists only of adhesive agents; to the best of our knowledge, we are the first to study the third model, which consists of both pulling and adhesive agents.

In this section, we briefly introduce our case study ABMs and their rules on agent pulling and adhesion in Sect. 2.1; we then detail our ABM notation and simulation in Sect. 2.2. Additional details on the ABM rules and implementation can be found in electronic supplementary materials S1 and S2, respectively.

2.1 Brief Introduction to the Case Study ABMs and Their Model Rules

Rules A-F governing agent pulling and adhesion are visually depicted in Fig. 1, and the parameters for each rule are described in Table 1. In all rules, a *migrating agent* chooses one of its four neighboring lattice site to move into with equal probability (Fig. 1a). A migration event is aborted if the lattice site in the chosen direction is already occupied (Fig. 1b). We refer to a *neighboring agent* as an agent located next to the migrating agent in the direction opposite of the chosen migration direction.

Rules A, B, and E are initiated when a pulling agent attempts to migrate, which occurs with rate r_m^{pull}. Migratory pulling agents pull their neighboring agents along with them with probability p_{pull}. Rules C, D, and F are initiated when an adhesive agent attempts to migrate, which occurs with rate r_m^{adh}. Neighboring adhesive agents adhere to migrating agents and abort the migration event with probability p_{adh}. The parameter α corresponds to the proportion of adhesive agents in the simulation. Even

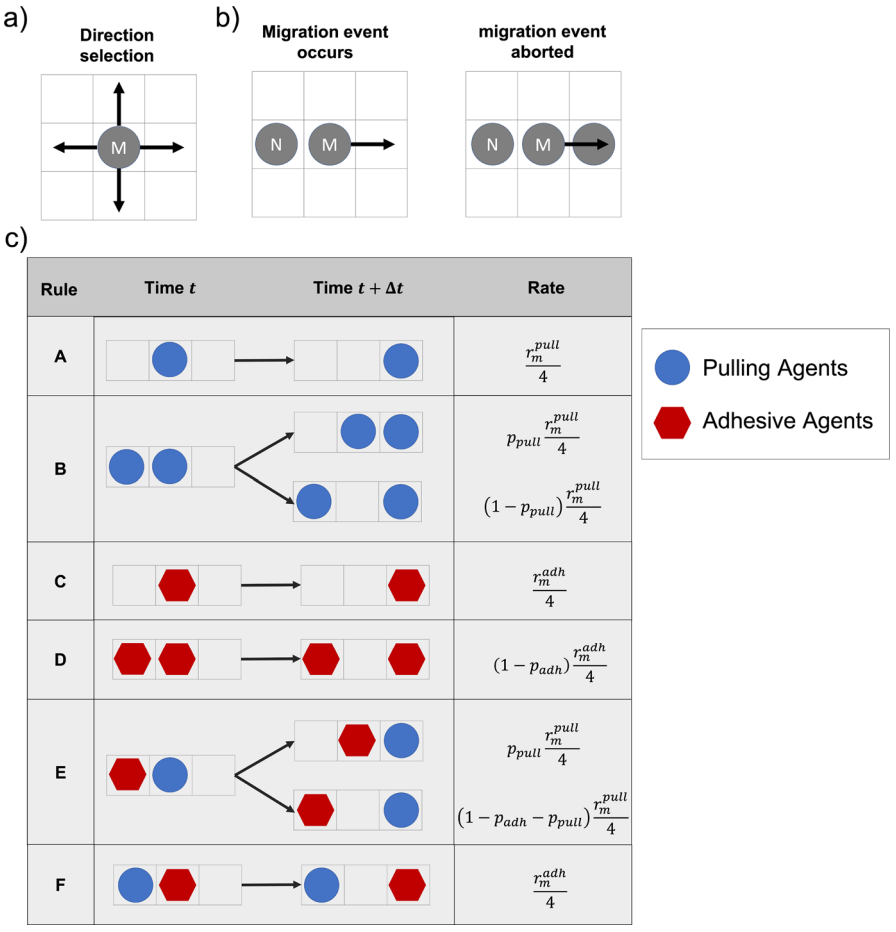

Fig. 1 ABM rules on migration, pulling, and adhesion. **a** When an agent performs a migration event, it chooses one of the four cardinal directions to move towards with equal probability; migration can also lead to a pulling or adhesion event in the chosen direction. The migrating agent is referred to as a migrating agent (M) **b** A migration event requires the lattice site in the chosen migration direction to be empty; otherwise, the migration event is aborted. A neighboring agent (N) is an agent located in the direction opposite the chosen migration direction. **c** Rules A-F dictate the rules on agent migration, pulling, and adhesion. Here, we show each rule when an agent chooses to move rightwards. Rule A prescribes how a pulling agent (blue circle) migrates when there is no neighboring agent. Rule B prescribes how a pulling agent migrates and attempts to pull a neighboring pulling agent with it. Rule C prescribes how an adhesive agent (red hexagon) migrates when there is no neighboring agent. Rule D prescribes how a neighboring adhesive agent attempts to adhere to a migrating adhesive agent and abort its migration event. Rule E prescribes how a migrating pulling agent attempts to pull its neighboring adhesive agent, while the adhesive agent attempts to adhere to the pulling agent. Rule F prescribes how a migrating adhesive agent and neighboring pulling agent do not interact with each other. The last column documents the rate at which each lattice site configuration at time t changes to the updated lattice site configuration at time $t + \Delta t$

Table 1 ABM model parameters

Variable	Description	Range
r_m^{pull}	Pulling agent migration rate	$[0, \infty)$
r_m^{adh}	Adhesive agent migration rate	$[0, \infty)$
p_{pull}	Probability of successful pulling event	[0, 1]
p_{adh}	Probability of successful adhesion event	[0, 1]
α	Proportion of adhesive agents	[0, 1]

We describe each model parameter and present their range of possible values

though we eventually summarize each ABM simulation along the x-direction, all rules on migration, pulling, and adhesion occur in all four cardinal directions.

Our three case study ABMs are:

1. **The Pulling ABM**, which consists of rules A and B. This model has parameters $\boldsymbol{p} = (r_m^{pull}, p_{pull})^T$.
2. **The Adhesion ABM**, which consists of rules C and D. This model has parameters $\boldsymbol{p} = (r_m^{adh}, p_{adh})^T$.
3. **The Pulling & Adhesion ABM**, which consists of rules A-F. This model has parameters $\boldsymbol{p} = (r_m^{pull}, r_m^{adh}, p_{pull}, p_{adh}, \alpha)^T$.

2.2 ABM Notation

All parameters used to configure ABM simulations are summarized in Table 2. Each model is simulated in the spatial domain $(x, y) \in [0, X] \times [0, Y]$. We represent this space with a two-dimensional lattice with square lattice sites of length $\Delta = 1$ to imitate a typical cell length. Let $N_P^{(r)}(x_i, t_j)$ and $N_H^{(r)}(x_i, t_j)$ denote the number of pulling and adhesive agents, respectively, in the ith column at the jth timepoint for $i = 1, \ldots, X$ and $j = 1, \ldots, T_f$ from the rth of R identically prepared ABM simulations (the input model parameters are fixed but the R model initializations and subsequent agent behaviors are stochastic). Here, X and T_f denote the number of spatial columns and temporal grid points, respectively. To estimate the spatiotemporal pulling and adhesive agent densities from the r^{th} simulation, we compute

$$P^{(r)}(x_i, t_j) = \frac{N_P^{(r)}(x_i, t_j)}{Y} \text{ and } H^{(r)}(x_i, t_j) = \frac{N_H^{(r)}(x_i, t_j)}{Y}, \text{ for } i = 1, \ldots, X, \text{ and } j = 1, \ldots, T_f,$$

respectively. The *total* agent density in the r^{th} simulation is then estimated by

$$T^{(r)}(x_i, t_j) = P^{(r)}(x_i, t_j) + H^{(r)}(x_i, t_j).$$

Table 2 ABM configuration parameters

Variable	Description	Value
R	Number of averaged ABM simulations per dataset	25
t_f	Ending simulation time	1000
Δt	Spacing between temporal gridpoints	10
T_f	Number of total timepoints	100
T_f^{train}	Number of training timepoints	75
T_f^{test}	Number of testing timepoints	25
X	Number of horizontal lattice sites	200
Y	Number of vertical lattice sites	40
Δx	Spacing between spatial points	1

We describe each parameter used for ABM configuration and present the values used throughout this study

To estimate the averaged pulling, adhesive, and total agent density in the i^{th} column from R identically prepared ABM simulations over time, we compute:

$$\langle P^{ABM}(x_i, t_j)\rangle = \frac{1}{R}\sum_{r=1}^{R} P^{(r)}(x_i, t_j);$$

$$\langle H^{ABM}(x_i, t_j)\rangle = \frac{1}{R}\sum_{r=1}^{R} H^{(r)}(x_i, t_j); \text{ and}$$

$$\langle T^{ABM}(x_i, t_j)\rangle = \frac{1}{R}\sum_{r=1}^{R} T^{(r)}(x_i, t_j), \quad \text{for } i = 1, \ldots, X \text{ and } j = 1, \ldots, T_f.$$

3 Methods to Forecast and Predict ABM Data

In this section, we outline our methodologies for forecasting future ABM data and predicting ABM data at new parameter values. This begins with a description of how we generate ABM data in Sect. 3.1 followed by an overview of the four methods we use for ABM forecasting in Sect. 3.2. We then describe our approaches for ABM forecasting and prediction in Sects. 3.3 and 3.4, respectively. We visualize how BINNs can be used for these processes in Fig. 2. All methods are implemented using Python (version 3.9.12) with code available on GitHub at https://github.com/johnnardini/Forecasting_predicting_ABMs.

3.1 Simulating ABM Data

The process of simulating ABM data is illustrated in Part 1 of Fig. 2. At the parameter value $\boldsymbol{p}$, we calculate $\langle T^{ABM}(x, t; \boldsymbol{p})\rangle = \{\langle T^{ABM}(x_i, t_j; \boldsymbol{p})\rangle\}_{i=1,\ldots,X}^{j=1,\ldots,T_f}$. For subse-

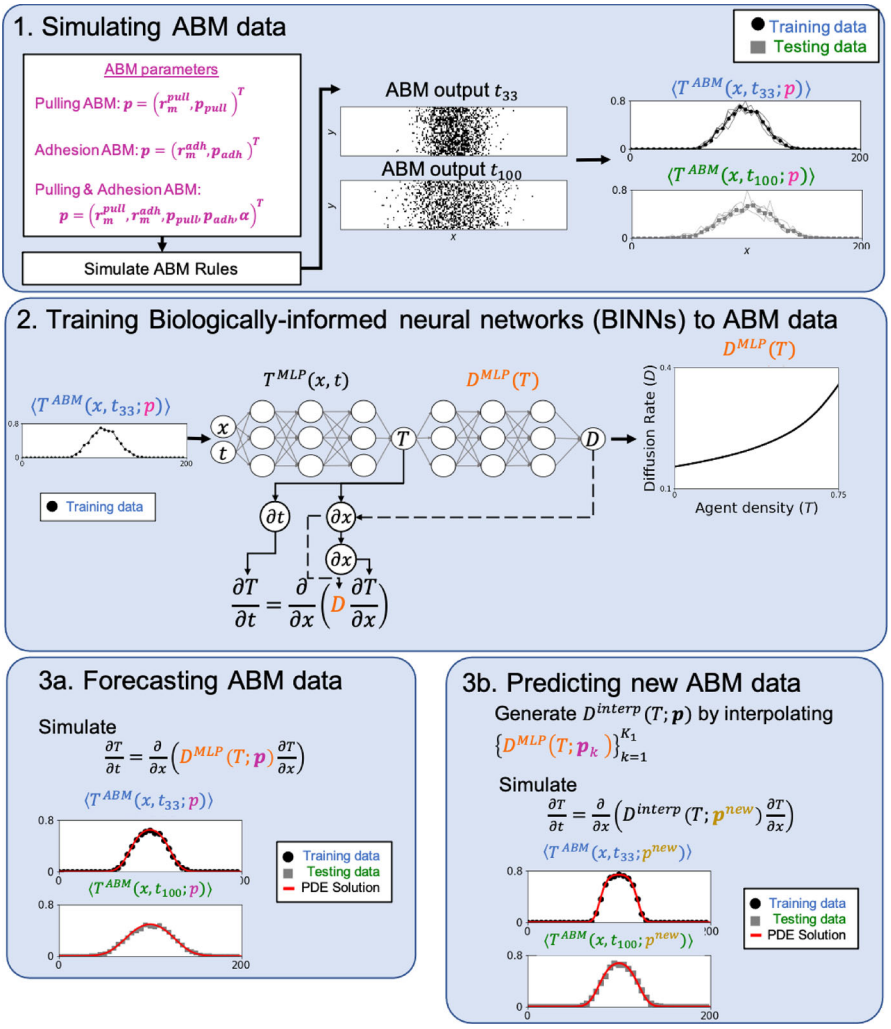

Fig. 2 Forecasting and predicting ABM data with BINNs. 1. Simulating ABM data. For a given parameter, $\boldsymbol{p}$, we simulate the Pulling, Adhesion, or Pulling & Adhesion ABM. Each model outputs snapshots of agent locations over time; we summarize this data by estimating the average total agent density along the x-direction for each snapshot. We perform R total ABM simulations (shown as thin lines) for each $\boldsymbol{p}$ and average the total spatiotemporal agent density to obtain $\langle T^{ABM}(x,t;\boldsymbol{p})\rangle$; in this figure, $R = 5$. The first T_f^{train} timepoints are placed into a training ABM dataset, and the final T_f^{test} timepoints are placed into a testing ABM dataset. 2. Training biologically-informed neural networks (BINNs) to ABM data. Each BINN model consists of a data-approximating MLP, $T^{MLP}(x,t)$, and a diffusion-rate-approximating MLP, $\mathcal{D}^{MLP}(T)$. BINN models are trained so that $T^{MLP}(x,t) \approx \langle T^{ABM}(x,t;\boldsymbol{p})\rangle^{train}$ while T^{MLP} and $\mathcal{D}^{MLP}$ satisfy Eq. (7). After model training, the inferred $\mathcal{D}^{MLP}(T)$ estimates the agent diffusion rate. 3a. Forecasting ABM data. Simulating the diffusion PDE framework with $\mathcal{D}^{MLP}(T)$ allows us to forecast the ABM training and testing data. 3b. Predicting new ABM data. We predict the rate of agent diffusion at a new parameter, $\boldsymbol{p}^{new}$, by interpolating $\mathcal{D}^{MLP}(T;\boldsymbol{p})$ over several $\boldsymbol{p}$ values to create $\mathcal{D}^{interp}(T;\boldsymbol{p})$. Simulating the diffusion PDE framework with $\mathcal{D}^{interp}(T;\boldsymbol{p}^{new})$ allows us to predict the new ABM training and testing data

quent model training and validation purposes, we split $\langle T^{ABM}(x,t;\boldsymbol{p})\rangle$ into training and testing datasets by setting

$$\langle T^{ABM}(x,t;\boldsymbol{p})\rangle^{train} = \left\{\langle T^{ABM}(x_i,t_j;\boldsymbol{p})\rangle\right\}_{i=1,\ldots,X}^{j=1,\ldots,T_f^{train}}, \text{ and}$$
$$\langle T^{ABM}(x,t;\boldsymbol{p})\rangle^{test} = \left\{\langle T^{ABM}(x_i,t_j;\boldsymbol{p})\rangle\right\}_{i=1,\ldots,X}^{j=T_f^{train}+1,\ldots,T_f^{train}+T_f^{test}}. \quad (1)$$

Here, T_f^{train} and T_f^{test} denote the number of training and testing timepoints, respectively, and $T_f = T_f^{train} + T_f^{test}$.

3.2 Models to Forecast ABM Data

We now describe the four models we use to forecast future ABM data. Namely, these models are the mean-field PDE, ANN, BINN, and BINN-guided PDE models.

The mean-field and BINN-guided PDE models consist of simulating a PDE of the form[1]:

$$\frac{\partial T}{\partial t} = \frac{\partial}{\partial x}\left(\mathcal{D}(T)\frac{\partial T}{\partial x}\right), \quad (2)$$

where $T = T(x,t) = P(x,t) + H(x,t)$ denotes the total agent density over space and time. The form of $\mathcal{D}(T)$ in Eq. (2) changes based on the ABM and the modeling approach being used. For the mean-field PDE, we determine the form of $\mathcal{D}(T)$ by converting discrete ABM rules into their continuous counterparts and invoking the mean-field assumption, which may be invalid at some parameter values. BINNs, on the other hand, are a data-driven approach to infer $\mathcal{D}(T)$ from the data without any such *a priori* assumptions.

The ANN and BINN models consist of training a prescribed neural network to ABM data and then using the trained neural network to forecast future data.

3.2.1 Mean-Field PDE Models

Here, we present the mean-field PDE models for each case study ABM. More detailed information on how the ABM rules are coarse-grained into these models are provided in electronic supplementary material S4. Our numerical method to numerically integrate these PDE models is provided in electronic supplementary material S6.

The Pulling ABM: The Pulling ABM includes only pulling agents and consists of Rules A-B from Fig. 1. In electronic supplementary material S4.1, we show that these rules can be coarse grained into the Pulling ABM's mean-field PDE model:

$$\frac{\partial P}{\partial t} = \nabla\cdot\left(\mathcal{D}^{pull}(P)\nabla P\right), \quad \mathcal{D}^{pull}(P) = \frac{r_m^{pull}}{4}\left(1+3p_{pull}P^2\right) \quad (3)$$

[1] with the exception of the mean-field PDE for the Pulling & Adhesion ABM, which requires simulating the two-compartment PDE given by Eq. (5) in Sect. 3.2.1

where $P = P(x, y, t)$ denotes the spatiotemporal pulling agent density.

The Adhesion ABM: The Adhesion ABM includes only adhesive agents and consists of Rules C-D from Fig. 1. In electronic supplementary material S4.2, we show that these rules can be coarse grained into the Adhesion ABM's mean-field PDE model:

$$\frac{\partial H}{\partial t} = \nabla \cdot \left(\mathcal{D}^{adh}(H) \nabla H \right), \quad \mathcal{D}^{adh}(H) = \frac{3 r_m^{adh}}{4} \left(p_{adh} \left(H - \frac{2}{3} \right)^2 + 1 - \frac{4 p_{adh}}{3} \right) \tag{4}$$

where $H = H(x, y, t)$ denotes the spatiotemporal adhesive agent density.

Notice that $\mathcal{D}^{adh}(H)$ from Eq. (4) becomes negative for some density values when $p_{adh} > 0.75$. This PDE thus fails to provide an ABM prediction at these parameter values because negative diffusion is ill-posed (Anguige and Schmeiser 2009).

The Pulling & Adhesion ABM: The Pulling & Adhesion ABM includes both pulling and adhesive agents, and consists of Rules A-F from Fig. 1. In electronic supplementary material S4.3, we show that these rules can be coarse-grained into the Pulling & Adhesion ABM's mean-field PDE model:

$$\begin{aligned}
\frac{\partial P}{\partial t} =& \frac{r_m^{pull}}{4} \nabla \cdot \left((1 - T) \nabla P + P \nabla T \right) \\
&+ p_{adh} \frac{r_m^{pull}}{4} \nabla \cdot \left(-3P(1 - T) \nabla H - H(1 - T) \nabla P - HP \nabla T \right) \\
&+ p_{pull} \frac{r_m^{pull}}{4} \nabla \cdot \left(3P^2 \nabla T \right) \\
\frac{\partial H}{\partial t} =& \frac{r_m^{adh}}{4} \nabla \cdot \left((1 - T) \nabla H + H \nabla T \right) \\
&+ p_{adh} \frac{r_m^{adh}}{4} \nabla \cdot \left(-4(1 - T) H \nabla H - H^2 \nabla T \right) \\
&+ p_{pull} \frac{r_m^{pull}}{4} \nabla \cdot \left(-(1 - T) H \nabla P + (1 - T) P \nabla H + 3HP \nabla T \right).
\end{aligned} \tag{5}$$

This two-compartment PDE describes the spatiotemporal densities of pulling agents, $P(x, y, t)$, and adhesive agents, $H = H(x, y, t)$. The total agent density is given by $T = T(x, y, t) = H(x, y, t) + P(x, y, t)$. To the best of our knowledge, it is not possible to convert Rules A-F into a single-compartment PDE model describing $T(x, y, t)$

3.2.2 The ANN Model

ANNs have recently gained traction as surrogate models for ABMs (Larie et al. 2021; Angione et al. 2022). Here, we consider a simple multilayer perceptron (MLP) model, $T^{MLP}(x, t)$, to predict the total agent density at the spatiotemporal point (x, t). We provide a brief description of the model architecture and training procedure in this

section; more detailed information can be found in electronic supplementary material S5.

The ANN architecture: $T^{MLP}(x,t)$ has a two-dimensional input, (x,t), and one-dimensional output, $T(x,t)$. This model has three hidden layers, each with 128 neurons. The hidden layers all have sigmoidal activation functions, and the output layer has a softplus activation function.

ANN model training: The ANN model is trained to minimize

$$\mathcal{L}_{ANN} = \mathcal{L}_{WLS}, \tag{6}$$

where $\mathcal{L}_{WLS}$ is given by Equation (S21) in electronic supplementary material S5 and computes a weighted mean-squared error (MSE) between $T^{MLP}(x,t)$ and $\langle T^{ABM}(x,t)\rangle^{train}$. Here, extra weight is assigned to data from the first timepoint to ensure that T^{MLP} closely agrees with the ABM's initial data.

We use the ADAM optimizer with default hyperparameter values to minimize Eq. (6). We perform 10^4 epochs with an early stopping criterion of 10^3 epochs.

3.2.3 The BINN Model

We provide a brief overview of our BINN model architecture and training procedure, which closely follows the implementation from the original BINN model study in Lagergren et al. (2020b). More detailed information can be found in electronic supplementary material S5.

The BINN architecture: We construct BINN models that consist of two sequential MLP models: $T^{MLP}(x,t)$ predicts the total agent density at the point (x,t), and $\mathcal{D}^{MLP}(T)$ predicts the agent diffusion rate at the density value T (Part 2 of Fig. 2). The architecture for $T^{MLP}(x,t)$ here is identical to the ANN architecture. The architecture for $\mathcal{D}^{MLP}(T)$ also has three hidden layers (each with 128 neurons), and the same hidden and output activation functions. However, this model has a one-dimensional input, T, and one-dimensional output, $D(T)$.

BINN model training: The two MLPs comprising the BINN model are trained to concurrently fit the given dataset, $\langle T^{ABM}(x,t)\rangle^{train}$, and solve the PDE given by

$$\frac{\partial}{\partial t}T^{MLP} = \frac{\partial}{\partial x}\left(\mathcal{D}^{MLP}(T^{MLP})\frac{\partial}{\partial x}T^{MLP}\right). \tag{7}$$

This is achieved by minimizing the following multi-term loss function:

$$\mathcal{L}_{BINN} = \mathcal{L}_{WLS} + \epsilon\mathcal{L}_{PDE} + \mathcal{L}_{constr}. \tag{8}$$

The equation for $\mathcal{L}_{WLS}$ is identical to Eq. (6), $\mathcal{L}_{PDE}$ computes the MSE between the left- and right-hand sides of Eq. (7) to ensure both MLPs satisfy this diffusion framework, and $\mathcal{L}_{constr}$ penalizes the two MLPs for violating user-defined criteria (such as lower and upper bounds on $\mathcal{D}^{MLP}$). The equations for these three terms are provided in Equations (S21), (S22), and (S23) from electronic supplementary material

S5. The ϵ parameter is chosen to ensure the $\mathcal{L}_{WLS}$ and $\mathcal{L}_{PDE}$ terms are weighted equally.

Following (Linka et al. 2022), we minimize Eq. (8) in a two-step process. In the first step, we minimize Eq. (6) over 10^4 epochs with an early stopping criterion of 10^3 epochs. In the second step, we minimize Equation (8) over 10^6 epochs with an early stopping criterion of 10^5 epochs. The ADAM optimizer is used during both steps with its default hyperparameter values.

3.2.4 The BINN-Guided PDE Model

BINN models are trained to satisfy Eq. (7). The *BINN-guided PDE model* computes this learned equation by simulating Eq. (2) with $\mathcal{D}(T) = \mathcal{D}^{MLP}(T)$. Our numerical method to numerically integrate this PDE is provided in electronic supplementary material S6.

3.3 Forecasting Future ABM Data

We use the four models introduced in Sect. 3.2 to forecast future ABM data (Part 3a of Fig. 2). In *forecasting*, we assess the ability of a model to compute future ABM data at a fixed parameter value from previous ABM data. This could correspond to inferring the future behavior of a computationally-intensive ABM simulation or an expensive experimental procedure.

We perform ABM forecasting by training each model to the training ABM dataset and then computing the model prediction over all space- and timepoints. The mean-field PDE model does not require any model training because we can directly compute it from the ABM parameter values. We then partition each model's prediction into training and testing datasets to match the ABM training and testing datasets from Equation (1). We report the training MSE from each model prediction as:

$$\frac{1}{XT_f^{train}} \sum_{i=1}^{X} \sum_{j=1}^{T_f^{train}} \left(T^{model}(x_i, t_j) - \langle T^{ABM}(x_i, t_j) \rangle \right)^2,$$

and the testing MSE as:

$$\frac{1}{XT_f^{test}} \sum_{i=1}^{X} \sum_{j=T_f^{train}+1}^{T_f} \left(T^{model}(x_i, t_j) - \langle T^{ABM}(x_i, t_j) \rangle \right)^2.$$

3.4 Predicting New ABM Data Using BINN-Guided PDE Models

We combine BINN modeling, multivariate interpolation, and numerical integration of PDEs to predict new ABM data (Part 3b of Fig. 2). In *predicting*, we assess the

ability of our proposed approach to compute ABM data at a parameter value that has not been seen previously. This could correspond to exploring an ABMs' parameter space, or predicting the output of an experimental procedure for different experimental conditions, such as drug concentration or the initial number of agents.

We perform multivariate interpolation using BINNs' computed diffusion rates to predict density-dependent diffusion rates for new ABM data. We define a prior parameter collection and a new parameter collection as

$$\mathcal{P}^{prior} = \{\boldsymbol{p}_k\}_{k=1}^{K_1} \text{ and } \mathcal{P}^{new} = \{\boldsymbol{p}_k^{new}\}_{k=1}^{K_2}.$$

Our workflow for predicting ABM data from $\mathcal{P}^{new}$ proceeds as follows:

1. Generate the prior and new ABM data collections by simulating the ABM at all parameters from the prior and new parameter collections:

$$\mathcal{T}^{prior} = \left\{\langle T^{ABM}(x,t;\boldsymbol{p}_k)\rangle\right\}_{k=1}^{K_1} \text{ and } \mathcal{T}^{new} = \left\{\langle T^{ABM}(x,t;\boldsymbol{p}_k^{new})\rangle\right\}_{k=1}^{K_2}.$$

2. Train a BINN model to each kth training ABM dataset from $\mathcal{T}^{prior}$ and extract $\mathcal{D}^{MLP}(T;\boldsymbol{p}_k)$ from the trained BINN model.
3. Perform multivariate interpolation on $\{\mathcal{D}^{MLP}(T;\boldsymbol{p}_k)\}_{k=1}^{K_1}$ to create an interpolant, $\mathcal{D}^{interp}(T;\boldsymbol{p})$, that matches the concatenated vector $[T,\boldsymbol{p}_k]$ to the diffusion rate $\mathcal{D}^{MLP}(T;\boldsymbol{p}_k)$ for $k=1,\ldots,K_1$.
4. Predict the new ABM dataset, $\langle T^{ABM}(x,t;\boldsymbol{p}_k^{new})\rangle$, by simulating Eq. (7) with $\mathcal{D}=\mathcal{D}^{interp}(T;\boldsymbol{p}_k^{new})$ to create $T^{interp}(x,t;\boldsymbol{p}_k^{new})$. Partition $T^{interp}(x,t;\boldsymbol{p}_k^{new})$ into its training and testing datasets to match the ABM data's training and testing datasets.
5. Compute the training and testing MSEs between $T^{interp}(x,t;\boldsymbol{p}_k^{new})$ and $\langle T^{ABM}(x,t;\boldsymbol{p}_k^{new})\rangle$ to summarize the predictive performance of $T^{interp}(x,t;\boldsymbol{p}_k^{new})$ for $k=1,\ldots,K_2$.

We implement multi-dimensional radial basis function interpolation using Sci-kit Learn's (version 0.24.2) **RBFInterpolator** command to create $\mathcal{D}^{interp}(T;\boldsymbol{p})$.

4 Results

4.1 Mean-Field and BINN-Guided PDEs Accurately Forecast Baseline ABM Simulations

We simulated the three case study ABMs using the configuration values provided in Table 2. These values were chosen to match previous studies (Chappelle and Yates 2019; Simpson et al. 2022). For ABMs of collective migration, one often chooses a large spatiotemporal domain to ensure ample ABM behavior is observed (e.g., the population spreads) while ensuring the boundary does not affect this behavior. In Table 3, we provide baseline model parameter values for each case study ABM; these

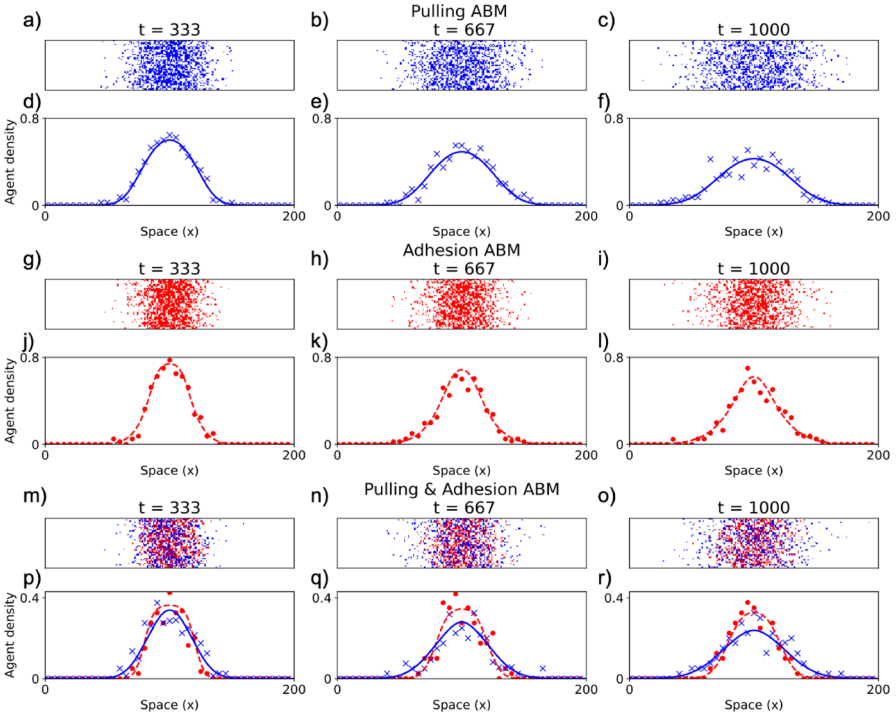

Fig. 3 Baseline ABM simulation snapshots and the mean-field PDE models for the Pulling, Adhesion, and Pulling & Adhesion ABMs. Blue pixels denote pulling agents and red pixels denote adhesive agents. All ABMs were simulated on rectangular 200×40 lattices. **a–c** Snapshots of the Pulling ABM for $r_m^{pull} = 1.0$, $p_{pull} = 0.5$. **d–f** The output spatiotemporal pulling agent density (blue 'x' marks) is plotted against the solution of the mean-field PDE (solid blue line) given by Eq. (3). **g–i** Snapshots of the Adhesion ABM for $r_m^{adh} = 1.0$, $p_{adh} = 0.5$. **j–l** The output spatiotemporal adhesive agent density (red dots) is plotted against the solution of the mean-field PDE (dashed red line) given by Eq. (4). **m–o** Snapshots of the Pulling & Adhesion ABM for $r_m^{pull} = 1.0, r_m^{adh} = 0.25, p_{pull} = 0.33, p_{adh} = 0.33, \alpha = 0.5$. **p–r** The output spatiotemporal pulling and adhesive agent densities are plotted against the solution of the mean-field PDE given by Eq. (5)

values were arbitrarily chosen to demonstrate typical ABM behavior characterized by moderate population spread. The ABM outputs are depicted against each ABM's mean-field PDE in Fig. 3. The mean-field PDE models accurately describe the baseline simulations for all three ABMs.

We investigate the performance of the mean-field PDE, ANN, BINN, and BINN-guided PDE models in forecasting Pulling ABM data from the baseline parameter values provided in Table 3. Visual inspection suggests that all four models match the ABM training data well (Fig. 4a, b). The computed training MSE values reveal that the mean-field and BINN-guided PDEs outperform the neural networks in describing this data (Table 3). The BINN, BINN-guided PDE, and mean-field PDE all accurately forecast the testing data (Fig. 4c), but the two PDE models achieve smaller testing MSE values than the BINN model (Table 3). The ANN's prediction for the testing data has a protrusion that overpredicts all data for $x > 125$ (Fig. 4c inset), which causes this

Table 3 Computed training and testing MSE values

Forecasting model	Training MSE	Testing MSE
The Pulling ABM with baseline parameters		
$\boldsymbol{p} = (r_m^{pull}, p_{pull})^T = (1.0, 0.5)^T$		
ANN	1.17×10^{-4}	9.36×10^{-4}
BINN	9.32×10^{-5}	1.47×10^{-4}
Mean-field PDE	7.45×10^{-5}	1.00×10^{-4}
BINN-guided PDE	7.64×10^{-5}	1.02×10^{-4}
The Adhesion ABM with baseline parameters		
$\boldsymbol{p} = (r_m^{adh}, p_{adh})^T = (1.0, 0.5)^T$		
ANN	1.55×10^{-4}	1.84×10^{-3}
BINN	8.54×10^{-5}	1.50×10^{-4}
Mean-field PDE	7.18×10^{-5}	9.21×10^{-5}
BINN-guided PDE	7.43×10^{-5}	1.02×10^{-4}
The Pulling & Adhesion ABM with baseline parameters		
$\boldsymbol{p} = (r_m^{pull}, r_m^{adh}, p_{pull}, p_{adh}, \alpha)^T = (1.0, 0.25, 0.33, 0.33, 0.5)^T$		
ANN	1.25×10^{-4}	2.67×10^{-3}
BINN	9.65×10^{-5}	9.96×10^{-5}
Mean-field PDE	7.50×10^{-5}	8.55×10^{-5}
BINN-guided PDE	6.55×10^{-5}	9.11×10^{-5}

Computed MSE values when forecasting $\langle T^{ABM}(x,t)\rangle^{train}$ and $\langle T^{ABM}(x,t)\rangle^{test}$ from the three ABMs at their baseline parameter values. We used an ANN, BINN, mean-field PDE, and BINN-guided PDE to forecast each baseline ABM dataset

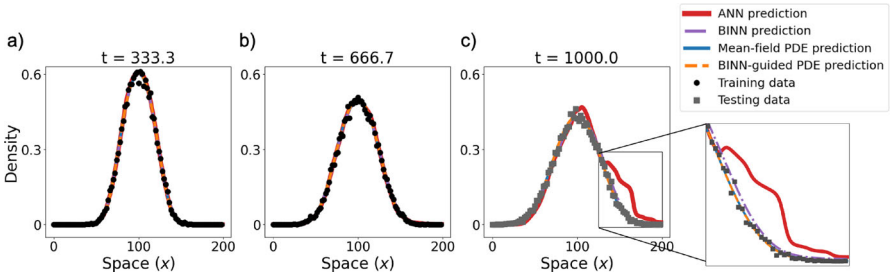

Fig. 4 Forecasting Pulling ABM data with neural networks and PDEs. ANN and BINN models were trained to fit $\langle T^{ABM}(x,t)\rangle^{train}$ from the Pulling ABM with $\boldsymbol{p} = (r_m^{pull}, p_{pull})^T = (1.0, 0.5)^T$. These two neural networks and the mean-field and BINN-guided PDE simulations were then used to forecast (**a**, **b**) $\langle T^{ABM}(x,t)\rangle^{train}$ and **c** $\langle T^{ABM}(x,t)\rangle^{test}$

model's computed testing MSE value to be almost an order of magnitude higher than all others. We obtain similar results when using the four models to predict data from the Adhesion ABM and Pulling & Adhesion ABM at their baseline parameter values (Table 3 and Supplementary Figure S1).

4.2 Forecasting ABM Data for Many Parameter Values with BINN-Guided and Mean-Field PDE Simulations

We now investigate the performance of BINN-guided and mean-field PDE simulations in forecasting ABM datasets over a wide range of parameter values for all three case study ABMs. We only consider the two PDE models (and exclude the neural network models) in this section due to their strong forecasting performance in Sect. 4.1.

4.2.1 The BINN-Guided and Mean-Field PDEs Both Accurately Forecast Pulling ABM Data

The parameters for the Pulling ABM are $\boldsymbol{p} = (r_m^{pull}, p_{pull})^T$. To evaluate the BINN-guided and mean-field PDE models' performances in forecasting Pulling ABM data over a range of agent pulling parameter values, we computed eleven ABM datasets by varying $p_{pull} = 0.0, 0.1, 0.2, \ldots, 1.0$ while fixing r_m^{pull} at its baseline value of 1.0. The inferred rates of agent diffusion from both models propose that agents diffuse slower for low densities and faster for high densities (Fig. 5a). While the mean-field diffusion rate at $p_{pull} = 0$ is constant, BINNs do not use this *a priori* information. Instead, their flexible nature leads to them learning a different diffusion rate from the data. The two PDE models achieve comparable training and testing MSE values for all values of p_{pull}, though the mean-field PDE usually attains slightly smaller values (Fig. 5b). Snapshots of both simulated PDE models against data shows that their ABM predictions are visually indistinguishable (Supplementary Figure S2(a-c)).

To evaluate both PDE models' performances over a range of pulling agent migration values, we computed 10 Pulling ABM datasets with $r_m^{pull} = 0.1, 0.2, \ldots, 1.0$ while fixing p_{pull} at its baseline value of 0.5. We find close agreement between both models' inferred diffusion rates for all values (Fig. 5c). Both models achieve similar computed training and testing MSE values (Fig. 5d). Snapshots of both simulated PDE models against data reveals that their ABM predictions are visually indistinguishable (Supplementary Figure S2(d-f)).

4.2.2 BINN-Guided PDEs Accurately Forecast Adhesion ABM Data When the Mean-Field PDE is Ill-Posed

The parameters for the pulling ABM are $\boldsymbol{p} = (r_m^{adh}, p_{adh})^T$. To evaluate the BINN-guided and mean-field PDE models' performances over a range of agent adhesion parameter values, we computed eleven ABM datasets by varying $p_{adh} = 0.0, 0.1, 0.2, \ldots, 1.0$ while fixing r_m^{adh} at its baseline value of 1.0. The inferred rates of agent diffusion from both models decrease with agent density for most values of p_{adh} (Fig. 6a). When $p_{adh} = 0$, the BINN-guided diffusion rate is slightly increasing and the mean-field model's diffusion rate is constant. The BINN-guided diffusion rates decline faster with agent density than the corresponding mean-field diffusion rates for low density values. We computed the training and testing MSEs for both models for all values of p_{adh} (Fig. 6b) and partition the results as follows:

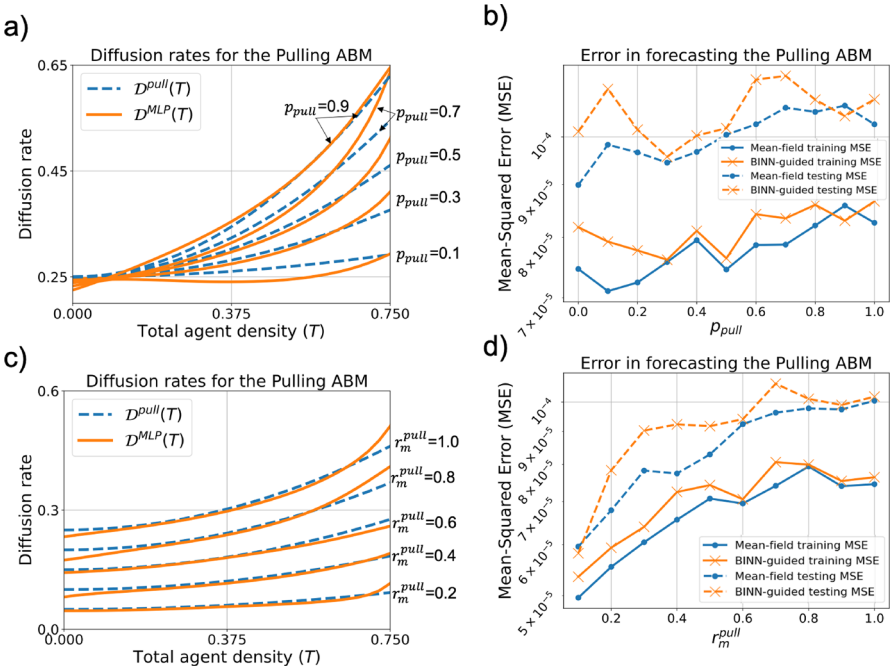

Fig. 5 Forecasting Pulling ABM data with the mean-field (MF) and BINN-guided PDEs. **a** Plots of the mean-field diffusion rate, $\mathcal{D}^{pull}(T)$, from Eq. (3) and the BINN-guided diffusion rate, $\mathcal{D}^{MLP}(T)$, for $p_{pull} = 0.1, 0.3, \ldots, 0.9$ (results not shown for $p_{pull} = 0.0, 0.2, \ldots, 1.0$ for visual ease) while fixing r_m^{pull} at its baseline value of 1.0. The horizontal axis ends at 0.75 instead of 1.0 because the ABM simulations begin with a density of 0.75 and will rarely exceed this initial value. The BINN cannot reliably predict the diffusion rate for densities outside the values observed in the data. **b** Plots of the mean-field and BINN-guided PDEs' computed training and testing MSE values while varying p_{pull} and fixing $r_m^{pull} = 1.0$. **c** Plots of $\mathcal{D}^{pull}(T)$ and $\mathcal{D}^{MLP}(T)$ for $r_m^{pull} = 0.2, 0.4, \ldots, 1.0$ while fixing p_{pull} at its baseline value of 0.5. **d** Plots of the mean-field and BINN-guided PDEs' computed training and testing MSE values while varying r_m^{pull} and fixing $p_{pull} = 0.5$

- **When $p_{adh} < 0.5$**: both models achieve similar training MSE values near 7×10^{-5} and testing MSE values around 10^{-4}.
- **When $0.5 \leq p_{adh} \leq 0.75$**: the mean-field PDE models' training and testing MSE values increase with p_{adh}, with a maximum computed value above 3×10^{-4}. The BINN-guided PDE model's training and testing MSE values remain near 7×10^{-5} and 10^{-4}, respectively.
- **When $p_{adh} > 0.75$**: the mean-field PDE model is ill-posed and cannot forecast this ABM data. The BINN-guided PDE model's computed training and testing MSE values increase with p_{adh} and have a maximum computed value of 2×10^{-4}.

Close inspection of snapshots from both PDE model simulations against ABM data from $p_{adh} = 0.7$ reveals that the mean-field PDE model slightly overpredicts the data at high densities above 0.5 and low densities below 0.1, whereas the BINN-guided PDE closely matches the data (Supplementary Figure S3(a-c)).

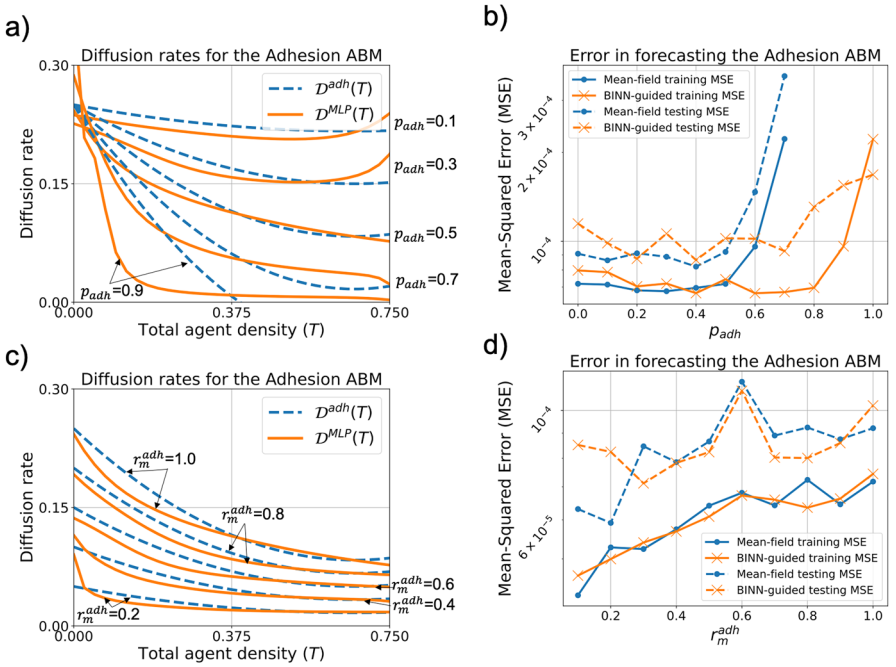

Fig. 6 Forecasting Adhesion ABM data with the mean-field (MF) and BINN-guided PDEs. **a** Plots of the mean-field diffusion rate, $\mathcal{D}^{adh}(T)$, from Eq. (4) and the BINN-guided diffusion rate, $\mathcal{D}^{MLP}(T)$, for $p_{adh} = 0.1, 0.3, \ldots, 0.9$ (results not shown for $p_{adh} = 0.0, 0.2, \ldots, 1.0$ for visual ease) while fixing r_m^{adh} at its baseline value of 1.0. **b** Plots of the mean-field and BINN-guided PDEs' computed training and testing MSE values while varying p_{adh} and fixing $r_m^{adh} = 1.0$. **c** Plots of $\mathcal{D}^{adh}(T)$ and $\mathcal{D}^{MLP}(T)$ for $r_m^{adh} = 0.2, 0.4, \ldots, 1.0$ while fixing p_{adh} at its baseline value of 0.5. **d** Plots of the mean-field and BINN-guided PDEs' computed training and testing MSE values while varying r_m^{adh} and fixing $p_{adh} = 0.5$

To evaluate both PDE models' performances over a range of adhesive agent migration values, we computed ten ABM datasets with $r_m^{adh} = 0.1, 0.2, \ldots, 1.0$ while fixing p_{adh} at its baseline value of 0.5. Both PDEs achieve similar computed training and testing MSE values for most values of r_m^{adh} (Fig. 6d). When $r_m^{adh} = 0.1$, however, the BINN-guided PDE's testing MSE value is close to 10^{-4}, whereas the mean-field PDE attains a lower testing MSE value near 6×10^{-5}. Despite these differences, the two model simulations appear similar at these parameter values (Supplementary Figure S3(d-f)).

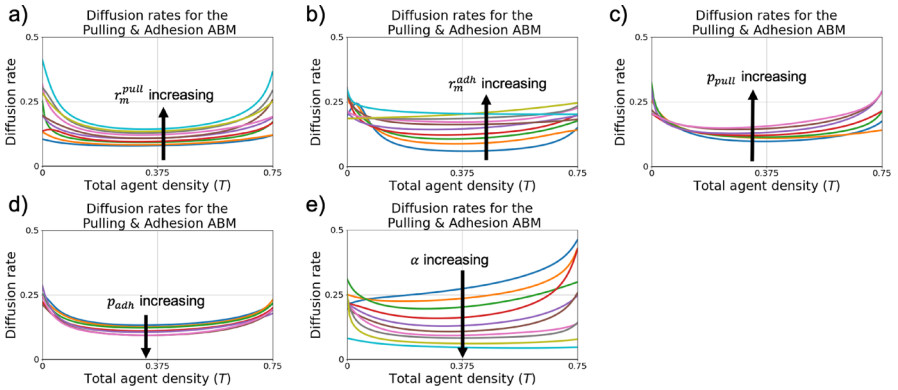

Fig. 7 The BINN-guided diffusion rates for the Pulling & Adhesion ABM data. Plots of the BINN-guided diffusion rate, $\mathcal{D}^{MLP}(T)$, when varying **a** r_m^{pull}, **b** r_m^{adh}, **c** p_{pull}, **d** p_{adh}, and **e** α

4.2.3 BINN-Guided PDEs Accurately Forecast Pulling & Adhesion ABM Data with a One-Compartment Model

The parameters for the Pulling & Adhesion ABM are $\boldsymbol{p} = (r_m^{pull}, r_m^{adh}, p_{pull}, p_{adh}, \alpha)^T$. We evaluate the performance of the BINN-guided and mean-field DE models in forecasting data from the Pulling & Adhesion ABM. We created 48 ABM datasets by fixing the baseline parameter values at $\boldsymbol{p}_{base} = (1.0, 0.25, 0.33, 0.33, 0.5)^T$ and then varying each parameter individually. We vary $r_m^{pull} = 0.5, 0.6, \ldots, 1.5$; $r_m^{adh} = 0.0, 0.1, \ldots, 1.0$; $p_{pull} = 0.1, 0.2, \ldots, 0.6, 0.67$; $p_{adh} = 0.1, 0.2, \ldots, 0.6, 0.67$; and $\alpha = 0.0, 0.1, \ldots, 1.0$. These parameter values were chosen to always satisfy $p_{pull} + p_{adh} \leq 1$.

The BINN models' inferred diffusion rates, $\mathcal{D}^{MLP}(T; \boldsymbol{p})$, are often U-shaped with larger diffusion values at low and high agent densities and smaller values at intermediate densities (Fig. 7). This U-shape tends to increase for larger values of r_m^{pull}, r_m^{adh}, and p_{pull} and decrease for larger values of p_{adh} and α. The inferred diffusion rates appear most sensitive to changes in the α parameter: at $\alpha = 0.0$, $\mathcal{D}^{MLP}(T; \boldsymbol{p})$ strictly increases with agent density and attains an average value of 0.289; at $\alpha = 1.0$, $\mathcal{D}^{MLP}(T; \boldsymbol{p})$ is strictly decreasing and has an average value of 0.051. The inferred diffusion rate is also sensitive to the r_m^{adh} and r_m^{pull} parameters: varying r_m^{adh} primarily alters the BINN diffusion rate at intermediate agent density values, whereas varying r_m^{pull} changes the BINN diffusion rate at low and high agent density values.

The BINN-guided PDE computes a single compartment to forecast the total agent density, $T(x, t)$, whereas the mean-field PDE computes two compartments forecasting the Pulling and Adhesive agent densities, $P(x, t)$ and $H(x, t)$, respectively. We forecast the total agent density with the mean-field PDE by setting $T(x, t) = P(x, t) + H(x, t)$. The two PDE models achieve similar training MSE values for most parameter values that we considered (Fig. 8). The mean-field model's testing MSE values are often smaller than the BINN-guided testing MSE values, though the

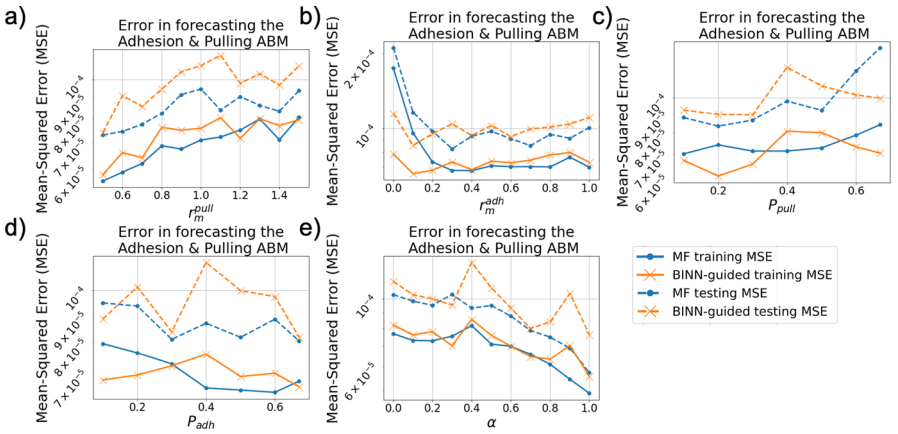

Fig. 8 Forecasting Pulling & Adhesion ABM data with the mean-field and BINN-guided PDEs. Plots of the mean-field and BINN-guided PDEs' computed training and testing values while varying **a** r_m^{pull}, **b** r_m^{adh}, **c** p_{pull}, **d** p_{adh}, and **e** α

BINN-guided PDE also achieves small testing MSE values. For example, both PDE simulations accurately predict ABM data when p_{adh} is set to 0.4, but visualizing both PDE simulations shows that the mean-field PDE better matches the elbow of the data than the BINN-guided PDE (Supplementary Figure S4(a-c)). The BINN-guided PDE outperforms the mean-field PDE in forecasting data for small values of r_m^{adh}: plotting both PDE simulations against data from $r_m^{adh} = 0.1$ shows that the mean-field PDE underpredicts the largest agent density values, while the BINN-guided PDE accurately matches this data (Supplementary Figure S4(d-f)).

4.3 Predicting ABM Data at New Parameter Values

We now examine how performing multivariate interpolation on several BINN-guided diffusion rates, $\mathcal{D}^{MLP}(T; \boldsymbol{p})$, can aid the prediction of previously-unseen ABM data at new parameter values (see Sect. 3.4 for implementation details).

We predict new data from the Adhesion and Pulling & Adhesion ABMs in this section. We do not include the Pulling ABM in this work because the mean-field PDE model accurately forecasted ABM data for all parameter values that we considered in Sect. 4.2.1.

4.3.1 Predicting Adhesion ABM Data

The parameters for the Adhesion ABM are $\boldsymbol{p} = (r_m^{adh}, p_{adh})^T$. We perform ABM data prediction for $p_{adh} \geq 0.5$ in this section because we found that the mean-field PDE model accurately forecasted ABM data for $p_{adh} \leq 0.5$ in Sect. 4.2.2.

We first predict ABM data when varying p_{adh} and fixing r_m^{adh}. The prior data collection consists of $K_1 = 6$ ABM datasets generated by varying $p_{adh} = 0.5, 0.6, 0.7, \ldots, 1.0$ while fixing r_m^{adh} at its baseline value of 1.0; the new data

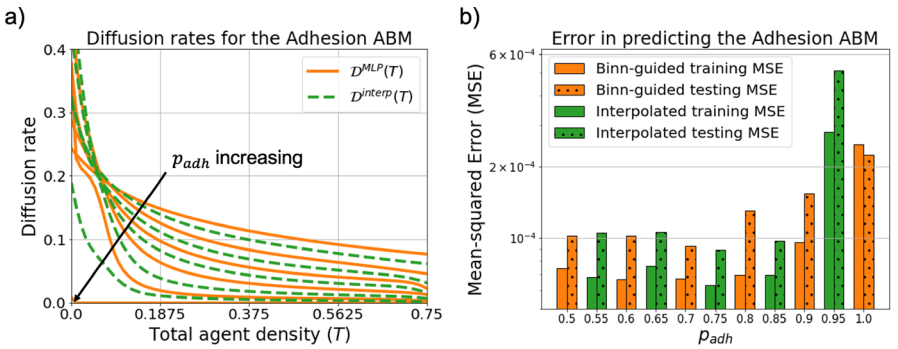

Fig. 9 Predicting Adhesion ABM data with BINN-guided PDEs and multivariate interpolation for new p_{adh} values. The parameters for the Adhesion ABM are given by $\boldsymbol{p} = (r_m^{adh}, p_{adh})^T$. Here, we vary p_{adh} while fixing r_m^{adh} at its baseline value of 1.0. The prior data collection consists of $p_{adh} = 0.5, 0.6, \ldots, 1.0$ and the new data collection consists of $p_{adh} = 0.55, 0.65, \ldots, 0.95$ **a** Plots of the learned $\mathcal{D}^{MLP}(T; \boldsymbol{p})$ diffusion rates for the prior data collection. We performed multivariate interpolation on these rates to obtain $\mathcal{D}^{interp}(T; \boldsymbol{p})$, which we plot for the new data collection. **b** Plots of the BINN-guided PDEs' computed training and testing values on the prior data collection, and the interpolated PDE's training and testing values on the new data collection

collection consists of $K_2 = 5$ ABM datasets generated by varying $p_{adh} = 0.55, 0.65, 0.75, 0.85,$ and 0.95 while fixing r_m^{adh} at its baseline value of 1.0. We performed multivariate interpolation over the six inferred $\mathcal{D}^{MLP}(T; \boldsymbol{p})$ terms from the prior data collection to generate $\mathcal{D}^{interp}(T; \boldsymbol{p})$. We use this interpolant to predict the diffusion rates for all parameters from the new data collection (Fig. 9a). All interpolated diffusion rates decrease with agent density and tend to fall with larger p_{adh} values. Most of the computed training and testing MSE values on the new data collection are comparable to their counterparts from the prior data collection (Fig. 9b). The lone exception occurs at $p_{adh} = 0.95$, where the testing MSE exceeds 5×10^{-4} while the testing MSEs at $p_{adh} = 0.9$ and 1.0 do not exceed 2.5×10^{-4}. Visual inspection of the simulated PDE prediction against ABM data at $p_{adh} = 0.95$ reveals that it matches the data well but slightly mispredicts the data's heel at later time points (Supplementary Figure S5(a-c)).

We next predict ABM data when varying both r_m^{adh} and p_{adh}. The prior data collection consists of $K_1 = 18$ ABM datasets generated by varying $r_m^{adh} = 0.1, 0.5, 1.0$ and $p_{adh} = 0.5, 0.6, \ldots, 1.0$; the new data collection consists of $K_2 = 10$ ABM datasets generated from a latin hypercube sampling of $(r_m^{adh}, p_{adh}) \in [0.1, 1.0] \times [0.5, 1.0]$ (Fig. 10a and Supplementary Table S2). We performed multivariate interpolation over each $\mathcal{D}^{MLP}(T; \boldsymbol{p})$ from the prior data collection to generate $\mathcal{D}^{interp}(T; \boldsymbol{p})$. The predicted diffusion rates for the new data collection decrease with agent density, rise for larger r_m^{adh} values, and decrease faster for larger p_{adh} values (Fig. 10b). We order the parameters from the new data collection by increasing training MSE values (Fig. 10c). The four lowest training and testing MSE values are all below 1×10^{-4}, the eight lowest are all below 2×10^{-4}, and the highest testing MSE value reaches 1.6×10^{-3}. Visual inspection of the interpolated PDE prediction with the highest testing MSE value reveals that this simulation mispredicts the data's heel but otherwise matches

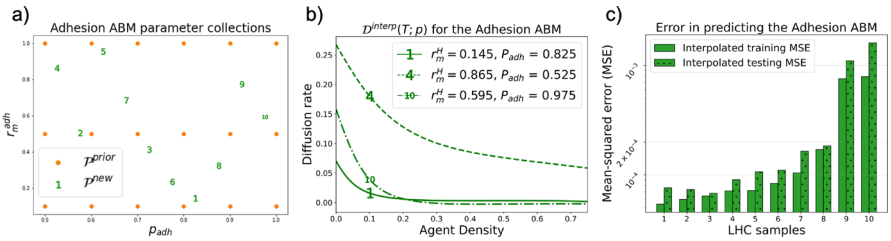

Fig. 10 Predicting Adhesion ABM data with BINN-guided PDEs and multivariate interpolation for new r_m^{adh} and p_{adh} values. The parameters for the Adhesion ABM are given by $\boldsymbol{p} = (r_m^{adh}, p_{adh})^T$. Here, we vary both parameters. **a** The prior data collection consists of $r_m^{adh} = 0.1, 0.5, 1.0$ and $p_{adh} = 0.5, 0.6, \ldots, 1.0$ and the new data collection consists of a Latin hypercube (LHC) sampling of $\boldsymbol{p} \in [0.1, 1.0] \times [0.5, 1.0]$ with $K_2 = 10$ samples. **b** We performed multivariate interpolation on the $\mathcal{D}^{MLP}(T; \boldsymbol{p})$ rates on the prior data collection to obtain $\mathcal{D}^{interp}(T; \boldsymbol{p})$. We plot three illustrative $\mathcal{D}^{interp}(T; \boldsymbol{p})$ values from the new data collection. **c** Plots of the interpolated PDE's training and testing values on the new data collection

the ABM data well (Supplementary Figure S6(a-c)). Visual inspection of the interpolated PDE prediction with the third-highest MSE value shows that this simulation accurately matches the ABM data (Supplementary Figure S6(d-f)).

4.3.2 Predicting Adhesion & Pulling ABM Data

The parameters for the Pulling & Adhesion ABM are $\boldsymbol{p} = (r_m^{pull}, r_m^{adh}, p_{pull}, p_{adh}, \alpha)^T$. We perform ABM data prediction over a large range of parameter values to determine if the one-compartment BINN-guided PDE simulations can predict this ABM's data, which results from two interacting subpopulations.

We perform multivariate interpolation over the p_{pull}, p_{adh}, and α parameters while fixing r_m^{pull} and r_m^{adh} at their baseline values of 1.0 and 0.25, respectively. The prior and new data collections consist of $K_1 = 40$ and $K_2 = 20$ ABM parameter combinations, respectively, that were generated from Latin hypercube samplings of $(p_{pull}, p_{adh}, \alpha) \in [0, 0.67] \times [0, 0.67] \times [0, 1]$ (Fig. 11a and Supplementary Tables S3 and S4). We chose samplings where $p_{pull} + p_{adh} \leq 1.0$ for all samples. The computed training and testing MSE values for the new parameter collection suggest all simulated PDE predictions accurately match the ABM data at those parameters (Fig. 11b). Of the $K_2 = 20$ computed testing MSE values in the new data collection, four are below 1×10^{-4}, 16 are below 2×10^{-4}, and all are below 5×10^{-4}. The highest and third highest testing MSE value results from $(p_{pull}, p_{adh}, \alpha) = (0.218, 0.553, 0.675)$ and $(0.251, 0.486, 0.975)$, respectively. Visually inspecting the interpolated PDE predictions from these parameter values against ABM data reveals that both match the data well, though the worst prediction overpredicts the largest ABM density values (Supplementary Figure S7).

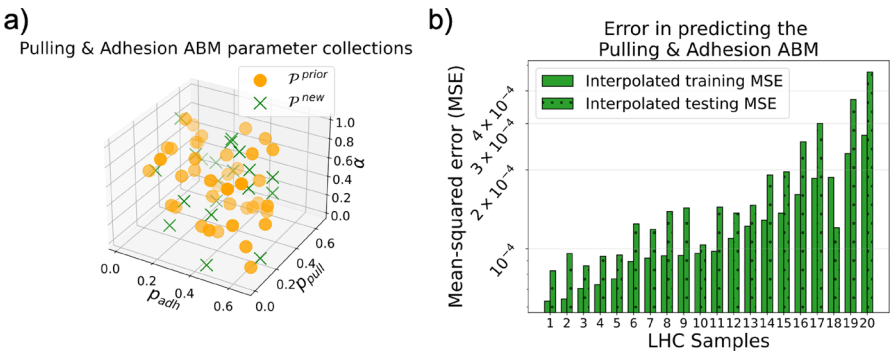

Fig. 11 Predicting Pulling & Adhesion ABM data for new p_{pull}, p_{adh}, and α values. The parameters for the Adhesion ABM are given by $\boldsymbol{p} = (r_m^{adh}, r_m^{pull}, p_{adh}, p_{pull}, \alpha)^T$. Here, we vary p_{pull}, p_{adh}, and α while fixing r_m^{pull} and r_m^{adh} at their baseline values of 1.0 and 0.25, respectively. **a** The prior data consists of a Latin hypercube (LHC) sampling of $(p_{pull}, p_{adh}, \alpha) \in [0, 0.67] \times [0, 0.67] \times [0, 1]$ with $K_1 = 40$ samples and the new data consists of a LHC sampling of the same domain with $K_2 = 20$ samples. **b** Plots of the interpolated PDE's training and testing values on the new data, arranged by increasing training MSE values

Table 4 Computational expenses of each modeling approach

ABM Name	ABM simulation	MF PDE simulation	BINN training	BG PDE simulation
Adhesion	37.5 (15.4) min	0.5 (0.15) s	10.6 (4.44) h	16.9 (23.65) s
Pulling	39.9 (15.8) min	0.6 (0.20) s	10.0 (3.99) h	164.8 (156.9) s
Pulling & Adhesion	42.5 (15.52) min	4.7 (1.20) s	13.1 (4.54) h	66.9 (50.81) s
Average	40.0 min	1.9 s	11.2 h	82.9 s

The mean wall time computations (standard deviation in parentheses) for ABM simulations, BINN training, mean-field (MF) PDE simulations, and BINN-guided (BG) PDE simulations for all three ABMs. The last row depicts the average mean computation time across all three ABMs

4.4 Comparing the Computational Expense of Each Modeling Approach

We finish with a discussion on the computational expense of all approaches discussed in this work (Table 4 and Supplementary Figure S8). We recorded the computed wall times to simulate each ABM, train each BINN model, and simulate each PDE from Sect. 4.2. Averaging across all ABMs suggests that the average ABM dataset took 40.0 min to generate with a standard deviation of 15.6 min. The average mean-field PDE model simulations for the Pulling ABM and the Adhesion ABM took 0.6 and 0.5 s to complete, respectively, which are about 4,000 and 4,500 times faster than the average ABM simulation time. The average mean-field PDE model simulation time for the Pulling & Adhesion ABM was 4.7 s, which is 542 times faster than the average ABM simulation time. Training a BINN model is the most time-consuming task with an average time of 11.2 h across all ABMs with a standard deviation of 4.32 h. The average BINN-guided PDE simulation takes 82.9 s with a standard deviation of 77.12 s, which is approximately 28 times faster than simulating the ABM.

5 Discussion and Future Work

In this work, we introduced how BINNs can be used to learn BINN-guided PDE models from simulated ABM data. BINN-guided PDE model simulations provide a new approach for forecasting and predicting ABM data. This methodology works by training a BINN model to match simulated ABM data while also obeying a pre-specified PDE model framework. After model training, future ABM data can be forecasted by simulating the BINN-guided PDE. Predicting ABM data at new parameters can be performed by simulating the pre-specified PDE framework with an interpolated modeling term. This model term is computed by interpolating over several learned BINN model terms and the parameter values that led to these terms.

It is challenging to predict how model parameters affect ABMs' output behavior due to their heavy computational nature. Mathematical modelers often address this limitation by coarse-graining ABM rules into computationally-efficient mean-field DE models. Unfortunately, these DE models may give misleading ABM predictions; furthermore, they can be ill-posed for certain parameter values (Anguige and Schmeiser 2009; Baker and Simpson 2010). Here, we demonstrated that BINN-guided PDE models accurately forecast future ABM data and predict ABM data from new parameter values. One benefit of this BINN-guided approach for ABM prediction is that BINNs can, in theory, be trained to simulated data from complex ABMs because BINN models are agnostic to the ABM rules. This is in contrast to the coarse-graining approach, which is limited to ABMs with simple rules to ensure a final PDE model can be recovered.

A limitation of the BINN-guided approach for ABM forecasting and prediction is the computational expense of BINN model training. The average BINN training procedure in this study took 11.2 h, which is 17 times longer than the average ABM data generation time of 40 min. Once a BINN model has been trained, however, the average BINN-guided PDE simulation took 83s, which is 28 times faster than the average time to generate an ABM dataset. One possible source of these long BINN training times is our chosen BINN model architecture, which consists of over over 50,000 parameters to train. Kaplarevi-Malii et al. Kaplarević-Malisić et al. (2023) proposed a genetic algorithm to identify the optimal model archictecture for PINN models. In future work, we plan to implement this algorithm to identify simpler BINN model architectures that can be efficiently trained to learn predictive PDE models for ABMs.

This work was purely computational, as we applied all prediction methodologies to simulated ABM data. It will be interesting in the future to validate the BINN-guided methodology on experimental data. Performing data-driven modeling techniques, such as parameter estimation, is challenging for ABMs due to their long simulation times. Our results suggest that BINN-guided PDE models may advance parameter estimation for ABMs by providing an accurate and efficient ABM surrogate model. For example, a typical approximate Bayesian computation (ABC) for parameter estimation requires performing 10,000 ABM simulations (Nguyen et al. 2024), which would require more than 6,600 computational hours. If we instead simulate the ABM at 10 parameter combinations, train BINN models to these data, and then use 10,000 interpolated BINN-guided PDE model simulations for ABC, then this total process would take

Table 5 Highlighting the ability of mean-field (MF) and BINN-guided (BG) PDEs to accurately forecast simulated ABM data with interpretable PDE models

	ABM prediction	Interpretability
Pulling ABM	MF PDE accurate for all parameters	MF PDE is interpretable
	BG PDE accurate for all parameters	BG PDE is interpretable
Adhesion ABM	MF PDE accurate for $p_{adh} \leq 0.5$	MF PDE is interpretable
	BG PDE accurate for $p_{adh} \leq 0.9$	BG PDE is interpretable
Pulling & Adhesion ABM	MF PDE accurate for all parameters	MF PDE not interpretable
	BG PDE accurate for all parameters	BG PDE is interpretable

349 h, a 19-fold reduction in time. This process will become even more efficient with new methodologies to expedite BINN model training.

Case study: collective migration. We studied three case study ABMs that are applicable to cell biological experiments, such as barrier and scratch assays. Each ABM consists of rules governing how key cellular interactions (namely, pulling and adhesion) impact the collective migration of cell populations during these experiments (Nardini et al. 2016; Thompson et al. 2012). Table 5 summarizes the predictive and interpretative capabilities of the mean-field and BINN-guided PDE models for the three case study ABMs. For the Pulling ABM, both models use interpretable one-compartment PDEs that accurately predict ABM behavior for all parameter values. For the Adhesion ABM, the mean-field PDE predictions become less accurate for $p_{adh} \in [0.5, 0.75]$ and are ill-posed for $p_{adh} > 0.75$, whereas the BINN-guided PDEs make accurate predictions for $p_{adh} \leq 0.9$. For the Pulling & Adhesion ABM, both PDE models accurately forecast the total ABM data for most parameter values considered. The mean-field PDE model is not interpretable, as it contains two compartments that consist of many terms. The BINN-guided PDE, on the other hand, achieves similar accuracy to the mean-field PDE with an interpretable one-compartment PDE model.

We compared the performance of the mean-field and BINN-guided PDE models throughout this work. We emphasize, however, that these two approaches are complementary, and our thorough investigation highlights the strengths and limitations of each model. The mean-field PDE is fast to simulate but can provide inaccurate, ill-posed, and/or uninterpretable ABM predictions. The BINN-guided PDE accurately predicts ABM behavior with an interpretable PDE, but current BINN model training times are lengthy. We encourage modelers to refer to these guidelines when deciding which approach to use for their future applications.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s11538-024-01357-2.

Acknowledgements The author thanks R. Baker, J. Gevertz, and S. Nardini for helpful discussion and commentary.

Funding The author acknowledges use of the ELSA high performance computing cluster at The College of New Jersey for conducting the research reported in this paper. This cluster is funded in part by the National Science Foundation under Grant numbers OAC-1826915 and OAC-2320244.

Data Availability All code and simulated data for this work is publicly available at https://github.com/johnnardini/Forecasting_predicting_ABMs.

Declarations

Conflict of interest The author declares no Conflict of interest.

References

Angione C, Silverman E, Yaneske E (2022) Using machine learning as a surrogate model for agent-based simulations. PLoS ONE 17(2):e0263150

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(3):395

Baker RE, Simpson MJ (2010) Correcting mean-field approximations for birth-death-movement processes. Phys Rev E 82(4):041905

Brauer F, Castillo-Chavez C, Feng Z (2019) Mathematical models in epidemiology, Texts in applied mathematics, vol 69. Springer, New York

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

Cai S, Wang Z, Wang S, Perdikaris P, Karniadakis GE (2021) Physics-informed neural networks for heat transfer Problems. J Heat Transf 143(6):060801

Champion K, Lusch B, Kutz JN, Brunton SL (2019) Data-driven discovery of coordinates and governing equations. Proc Natl Acad Sci 116(45):22445–22451

Chappelle G, Yates CA (2019) Pulling in models of cell migration. Phys Rev E 99(6):062413

Das AM, Eggermont AMM, Ten Hagen TLM (2015) A ring barrier-based migration assay to assess cell migration in vitro. Nat Protocols 10(6):904–915

Decaestecker C, Debeir O, Van Ham P, Kiss R (2007) Can anti-migratory drugs be screened in vitro? A review of 2D and 3D assays for the quantitative analysis of cell migration. Med Res Rev 27(2):149–176

Gibbs T, Levin SA, Levine JM (2022) Coexistence in diverse communities with higher-order interactions. Proc Natl Acad Sci 119(43):e2205063119

Grimm V, Revilla E, Berger U, Jeltsch F, Mooij WM, Railsback SF, Thulke H-H, Weiner J, Wiegand T, DeAngelis DL (2005) Pattern-oriented modeling of agent-based complex systems: lessons from ecology. Science 310(5750):987–991

Huppert A, Katriel G (2013) Mathematical modelling and prediction in infectious disease epidemiology. Clin Microbiol Infect 19(11):999–1005

Janiszewska M, Primi MC, Izard T (2020) Cell adhesion in cancer: beyond the migration of single cells. J Biol Chem 295(8):2495–2505

Johnston ST, Simpson MJ, Baker RE (2012) Mean-field descriptions of collective migration with strong adhesion. Phys Rev E 85(5):051922

Kaiser E, Nathan Kutz J, Brunton Steven L (2018) Sparse identification of nonlinear dynamics for model predictive control in the low-data limit. Proc R Soc A Math Phys Eng Sci 474(2219):20180335

Kaplarević-Malisić A, Andrijević B, Bojović F, Nikolić S, Krstić L, Stojanović B, Ivanović M (2023) Identifying optimal architectures of physics-informed neural networks by evolutionary strategy. Appl Soft Comput 146:110646

Kashef J, Franz CM (2015) Quantitative methods for analyzing cell-cell adhesion in development. Dev Biol 401(1):165–174

Kieu L-M, Malleson N, Heppenstall A (2020) Dealing with uncertainty in agent-based models for short-term predictions. R Soc Open Sci 7(1):191074

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 Math Phys Eng Sci 476(2234):20190800

Lagergren JH, Nardini JT, Baker RE, Simpson MJ, Flores KB (2020) Biologically-informed neural networks guide mechanistic modeling from sparse experimental data. PLoS Comput Biol 16(12):e1008462

Larie D, An G, Cockrell RC (2021) The use of artificial neural networks to forecast the behavior of agent-based models of pathophysiology: an example utilizing an agent-based model of sepsis. Front Physiol 12:716434
Linka K, Schäfer A, Meng X, Zou Z, Karniadakis GE, Kuhl E (2022) Bayesian physics informed neural networks for real-world nonlinear dynamical systems. Comput Methods Appl Mech Eng 402:115346
Mangan NM, Brunton SL, Proctor JL, Kutz JN (2016) Inferring biological networks by sparse identification of nonlinear dynamics. IEEE Trans Mol Biol Multi-Scale Commun 2(1):52–63
Mangan NM, Kutz JN, Brunton SL, Proctor JL (2017) Model selection for dynamical systems via sparse regression and information criteria. Proc R Soc A Math Phys Eng Sci 473(2204):20170009
Marshall BDL, Galea S (2015) Formalizing the role of agent-based modeling in causal inference and epidemiology. Am J Epidemiol 181(2):92–99
Messenger DA, Bortz DM (2021) Weak SINDy: galerkin-based data-driven model selection. Multiscale Model Simul 19(3):1474–1497
Messenger DA, Bortz DM (2021) Weak SINDy for partial differential equations. J Comput Phys 443:110525
Messenger DA, Bortz DM (2022) Learning mean-field equations from particle data using WSINDy. Phys D 439:133406
Messenger DA, Wheeler GE, Liu X, Bortz DM (2022) Learning anisotropic interaction rules from individual trajectories in a heterogeneous cellular population. J R Soc Interface 19(195):20220412
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
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(9):119
Nardini JT, Baker RE, Simpson MJ, Flores KB (2021) Learning differential equation models from stochastic agent-based model simulations. J R Soc Interface 18(176):20200987
Nguyen KC, Jameson CD, Baldwin SA, Nardini JT, Smith RC, Haugh JM, Flores KB (2024) Quantifying collective motion patterns in mesenchymal cell populations using topological data analysis and agent-based modeling. Math Biosci 370:109158
Raissi M, Perdikaris P, Karniadakis GE (2019) Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. J Comput Phys 378:686–707
Raja Noureen S, Owen JP, Mort RL, Yates CA (2023) Swapping in lattice-based cell migration models. Phys Rev E 107(4):044402
Rothenberg KE, Chen Y, McDonald JA, Fernandez-Gonzalez R (2023) Rap1 coordinates cell-cell adhesion and cytoskeletal reorganization to drive collective cell migration in vivo. Curr Biol 33(13):2587-2601.e5
Rudy SH, Brunton SL, Proctor JL, Kutz JN (2017) Data-driven discovery of partial differential equations. Sci Adv 3(4):e1602614
Rudy S, Alla A, Brunton SL, Kutz JN (2019) Data-driven identification of parametric partial differential equations. SIAM J Appl Dyn Syst 18(2):643–660
Shin Y, Darbon J, Karniadakis GE (2020) On the convergence of physics informed neural networks for linear second-order elliptic and parabolic type PDEs. Commun Comput Phys 28(5):2042–2074
Simpson MJ, Baker RE, Buenzli PR, Nicholson R, Maclaren OJ (2022) Reliable and efficient parameter estimation using approximate continuum limit descriptions of stochastic models. J Theor Biol 549:111201
Supekar R, Song B, Hastewell A, Choi GPT, Mietke A, Dunkel J (2023) Learning hydrodynamic equations for active matter from particle simulations and experiments. Proc Natl Acad Sci 120(7):e2206994120
Thompson RN, Yates CA, Baker RE (2012) Modelling cell migration and adhesion during development. Bull Math Biol 74(12):2793–2809
Tracy M, Cerdá M, Keyes KM (2018) Agent-based modeling in public health: current applications and future directions. Annu Rev Public Health 39(1):77–94. https://doi.org/10.1146/annurev-publhealth-040617-014317
VandenHeuvel DJ, Buenzli PR, Simpson MJ (2024) Pushing coarse-grained models beyond the continuum limit using equation learning. Proc R Soc A Math Phys Eng Sci 480(2281):20230619
Venhuizen J-H, Zegers MM (2017) Making heads or tails of it: cell-cell adhesion in cellular and supracellular polarity in collective migration. Cold Spring Harb Perspect Biol 9(11):a027854
Vishwakarma M, Spatz JP, Das T (2020) Mechanobiology of leader-follower dynamics in epithelial cell migration. Curr Opin Cell Biol 66:97–103

Xiao Y, Chen L (2001) Modeling and analysis of a predator-prey model with disease in the prey. Math Biosci 171(1):59–82