PHILOSOPHICAL TRANSACTIONS A

royalsocietypublishing.org/journal/rsta

Research

Cite this article: Böttcher L, Fonseca LL, Laubenbacher RC. 2025 Control of medical digital twins with artificial neural networks. *Phil. Trans. R. Soc. A* **383**: 20240228. https://doi.org/10.1098/rsta.2024.0228

Received: 29 April 2024
Accepted: 22 August 2024

One contribution of 13 to a theme issue 'Uncertainty quantification for healthcare and biological systems (Part 1)'.

Subject Areas:
artificial intelligence, computational biology, computer modelling and simulation, biomathematics, mathematical modelling

Keywords:
medical digital twins, control, artificial neural networks, systems biology

Author for correspondence:
Lucas Böttcher
e-mail: l.boettcher@fs.de

†These authors contributed equally to the study.

THE ROYAL SOCIETY PUBLISHING

Control of medical digital twins with artificial neural networks

Lucas Böttcher[1,2,†], Luis L. Fonseca[2,†] and Reinhard C. Laubenbacher[2]

[1]Department of Computational Science and Philosophy, Frankfurt School of Finance and Management, Frankfurt am Main 60322, Germany
[2]Department of Medicine, Laboratory for Systems Medicine, University of Florida, Gainesville, FL, USA

LB, 0000-0003-1700-1897; LLF, 0000-0002-7902-742X; RCL, 0000-0002-9143-9451

The objective of precision medicine is to tailor interventions to an individual patient's unique characteristics. A key technology for this purpose involves medical digital twins, computational models of human biology that can be personalized and dynamically updated to incorporate patient-specific data. Certain aspects of human biology, such as the immune system, are not easily captured with physics-based models, such as differential equations. Instead, they are often multi-scale, stochastic and hybrid. This poses a challenge to existing control and optimization approaches that cannot be readily applied to such models. Recent advances in neural-network control methods hold promise in addressing complex control problems. However, the application of these approaches to biomedical systems is still in its early stages. This work employs dynamics-informed neural-network controllers as an alternative approach to control of medical digital twins. As a first use case, we focus on the control of agent-based models (ABMs), a versatile and increasingly common modelling platform in biomedicine. The effectiveness of the proposed neural-network control methods is illustrated and benchmarked against other methods with two widely used ABMs. To account for the inherent stochastic nature of the ABMs we aim to control, we quantify uncertainty in relevant model and control parameters.

This article is part of the theme issue 'Uncertainty quantification for healthcare and biological systems (Part 1)'.

1. Introduction

The ultimate goal of precision medicine [1,2] is to identify interventions that can preserve or restore an individual's health by taking into account their unique personal characteristics. Computational models known as medical digital twins play an important role in realizing this goal [3–8]. Medical digital twins are designed to incorporate the most recent personal health data, offering guidance for the application of optimal interventions.

Developing medical digital twins is challenging as the underlying models must capture biological mechanisms operating at various spatial and temporal scales, such as the impact of drugs on both the intracellular scale and the larger organ or organism scale. Depending on the specific application, digital twins may also need to account for stochastic effects. Consequently, developing high-fidelity medical digital twins often necessitates the incorporation of high-dimensional, multi-scale, stochastic computational models. Given the difficulties associated with representing these intricacies using equation-based approaches, alternative model types, such as agent-based models (ABMs), frequently serve as the foundation for medical digital twins [8–16].

In biomedical applications, agents in an ABM represent biological entities such as cells in a tissue or microbes in a biofilm [17–20]. The behaviour of agents is usually described by stochastic rules, which allow them to navigate heterogeneous spatial environments and interact with each other. Since ABMs are intuitive and easily implementable computational models, they are accessible to domain experts without extensive computational modelling knowledge. They find applications in various medical scenarios, including studies on the immune system, tumour growth and treatment development [8–16]. A major drawback of using ABMs and other non-equation-based models is that the technology underpinning their analysis and use is largely missing, including parameter identifiability approaches, practical sensitivity analysis methods, forecasting algorithms and control and optimization tools. The results presented in this paper can be viewed as a contribution to the development of mathematical tools appropriate for the model types likely to be used for medical digital twins, as the use of this technology continues to expand.

Designing effective treatments using ABM-based digital twins is computationally challenging, due to the usually large state space of ABMs and the associated 'curse of dimensionality'. Additionally, while optimal control theory methods are well-established for ordinary differential equation (ODE) models in engineering [21,22], they are not readily applicable to complex hybrid models. Hence, identifying optimal controls using ABMs often relies on ad hoc methods. This issue extends beyond medicine to digital twin systems in various domains. As highlighted in the 2024 report on fundamental research gaps for digital twins by the National Academies of Engineering, Science and Medicine, there are currently no general solutions available to address this challenge [23].

While progress has been made in connecting approaches from control theory with biomedical ABMs [24], these methods are often only applicable to ODE surrogate models [25,26] and not to the original ABMs. Already in 1971, Alexey Ivakhnenko had commented in his work on polynomial neural networks on the challenges associated with the application of control theory to complex systems [27]: 'Modern control theory, based on differential equations, is not an adequate tool for solving the problems of complex control systems. It is necessary to construct differential equations to trace the input-output paths, that is, to apply a deductive deterministic approach. But it is impossible to use this approach for complex systems because of the difficulty

in finding these paths'. His work is nowadays regarded as a foundational contribution to the field of deep learning [28].

In parallel to the development of ABMs that provide the core components of many medical digital twins, artificial neural networks (ANNs) have become a common approach for deploying general function approximators in various machine-learning tasks, including dynamical system identification [29–32] and control [33–38]. Building upon these recent advances in neural-network control and automatic differentiation, in this paper, we develop ANN control approaches to effectively steer ABMs towards desired target states. We evaluate the effectiveness of these controllers against recently developed ODE-based surrogate modelling methods that do not employ ANNs [26]. (Related surrogate models have also been used to reconstruct the parameter spaces of ABMs [39].) Given the inherent stochastic nature of the ABMs we aim to control, we quantify uncertainty in relevant model and control parameters.

To illustrate the effectiveness of the proposed control approaches, we apply them to two paradigmatic ABMs: (i) a resource-dependent predator–prey system [40,41], and (ii) a regulated metabolic network. The predator–prey system that we consider was originally used to model the interaction between grass, sheep and wolves, but these types of models also find widespread applications in biomedicine [42–45]. In this context, certain immunological processes resemble those observed in ecological systems, where pathogens act as predators, preying on host cells or consuming host resources. Conversely, host cells can also function as predators and pathogens as prey. Unlike typical predators that consume prey to sustain their populations, immune cells do not rely on pathogen phagocytosis for growth. However, contact with pathogens tends to attract more immune cells, a phenomenon encoded similarly using mass-action kinetics. For example, when SARS-CoV-2 enters the human airways, it infects epithelial cells, while neutrophils and macrophages target the pathogen. Similarly, in cases of fungal infections like Aspergillosis [9,11], the pathogen scavenges iron from the host to support its growth. Meanwhile, host immune cells target the fungus and lock iron intracellularly, prompting the fungus to invade nearby blood vessels to access iron in haemoglobin. This scenario mirrors a predator–prey dynamic, with iron serving as the resource exploited by the fungus, acting as the prey.

The metabolic-pathway model that we consider in this paper describes the synthesis of two end products originating from a shared precursor. One of these end products serves a dual role: it inhibits the initial reaction in the pathway while stimulating metabolic flux towards the second end product. Such metabolic processes are prevalent in branched pathways, such as amino acid biosynthesis.

For controlling the metabolic-pathway model, we use neural ODE surrogates [29,30] and neural ODE controllers [33–35,38]. In contrast, for the predator–prey control example, we employ a model-based reinforcement learning approach. Many existing libraries rely on model-free reinforcement learning, which tends to converge more slowly and carries the risk of exploring unfavourable actions [46,47]. When an appropriate model is available to explore the action space, model-based approaches are beneficial as they converge more rapidly towards the desired solution.

To account for discrete actions, such as removing integer-valued quantities of predators and prey, in our predator–prey controller, we use a problem-tailored straight-through estimator [48,49]. This estimator has been previously applied in recommender systems [50] and inventory management [37], and a similar approach has also been considered in the context of financial markets [51].

Overall, our work contributes to expanding the applicability of ABMs across fields and offers a new perspective on controlling complex biomedical systems. Our source codes are publicly available at https://gitlab.com/ComputationalScience/abm-control.

2. Predator–prey model

We first consider a predator–prey ABM with three species, A, B and C [40–42,52]. We use a_k, b_k and c_k to denote the population sizes of species A, B and C at time k, respectively. In ecology, this model can describe the interactions between grass, sheep and wolves or between plankton, forage fish and predatory fish. In the case of Aspergillosis, the three species may represent iron (nutrient supply), Aspergillus (prey) and macrophages (predators) [9,11]. Generalized predator–prey models with even more species have found applications in studies of microbial communities [43].

We simulate the three-species predator–prey dynamics on an $L \times L$ periodic grid using an ABM parameterized as follows. Each grid cell in the predator–prey ABM can be in one of two different states: (i) 'nutrient-rich' and (ii) 'nutrient-poor'. Prey perform a random walk with a directional bias in positive x-direction and consume nutrients to stay alive. We use λ_1 to denote the energy gain per unit nutrient. Prey consume nutrients from the nearest available grid cell if it is in a 'nutrient-rich' state. This grid cell is then switched to a 'nutrient-poor' state and regenerates nutrients after τ periods.

Predators also perform a random walk with a directional bias identical to that of prey, and they consume prey when both are located within the same grid cell. The energy gain per prey is λ_2. In each period, all predators and prey lose one unit of energy to sustain their metabolism. Predators and prey die if their energy levels fall below 0. They reproduce at rates α_1 and α_2, respectively.

In figure 1, we show the output of a three-species predator–prey ABM with 51×51 grid cells at a specific moment in time. Green and light brown grid cells represent nutrient-rich and nutrient-poor regions, respectively. For more information on this model, we refer the reader to [40].

To control the predator–prey dynamics, we need to define suitable inputs and outputs for the ANN. Potential inputs are the population sizes a_k, b_k and c_k of species A, B and C at time k. In the two control problems that we consider in this paper, we aim at directly managing the numbers of predator and prey, so there are two integer-valued outputs u_1 and u_2 (figure 2*a*). If $u_1 > 0$ ($u_2 > 0$), a new prey (predator) will be added to the ABM at a position on the grid selected uniformly at random. In contrast, if $u_1 < 0$ ($u_2 < 0$), a prey (predator) will be selected uniformly at random and removed from the ABM. In the first control problem, our objective is to steer the dynamics towards a new steady state characterized by more prey and fewer predators. We use this basic control example to explain how ANN controllers can be integrated with ABMs. In the second control problem, we consider a more complex scenario involving the control of ABM dynamics during a transient phase. The ABM that we use in our simulations has 255×255 grid cells.

(a) Steady-state control

Before controlling the considered ABM, we let it evolve for 1000 time steps to estimate the steady-state numbers of predators and prey. For the parameters that we use in our simulations, the mean numbers of predators and prey over the final 100 time steps are about 1896 and 4159, respectively. When controlling the dynamics, we allow it to evolve for an additional 1000 time steps. We use N_t to denote the total number of time steps, which is 2000 in this example.

Our control target is to increase the mean number of prey in the last N_t^* time steps of the controlled time horizon (i.e. for times $k \in \{N_t - N_t^* + 1, \ldots, N_t\}$) by 10% while simultaneously reducing the corresponding mean number of predators by 50% (figure 2*b*). Intuitively, such a large reduction in the mean number of predators is associated with a large increase in the number of prey. Hence, the control function that we wish to identify has to reduce both the number of prey and predators in the steady state of the ABM. This can be achieved with a two-node ANN controller in which the two outputs

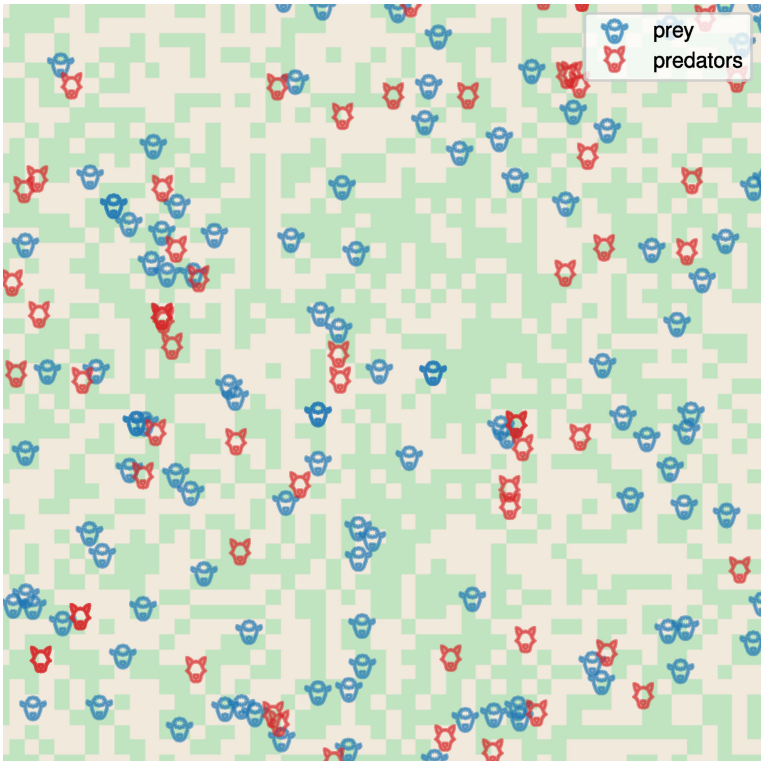

Figure 1. Predator–prey ABM dynamics. A snapshot of a three-species predator–prey ABM simulation with 51×51 grid cells. Green and light brown grid cells represent nutrient-rich and nutrient-poor regions, respectively.

are used to control the population sizes of both predators and prey (see appendix). To train the ANN controller, we use the quadratic loss function

$$J_1(\theta) = \left(\bar{b}(\theta) - \bar{b}^*\right)^2 + \left(\bar{c}(\theta) - \bar{c}^*\right)^2, \tag{2.1}$$

where $\theta \in \mathbb{R}^N$ denote ANN parameters, and $\bar{b}^*$ and $\bar{c}^*$ are the desired target states (i.e. the desired mean number of prey and predators over the last N_t^* time steps). In this first, steady-state control example, we have $N = 2$ ANN parameters (i.e. $\theta = (\theta_1, \theta_2)^\top$), $\bar{b}^* = 4575$, $\bar{c}^* = 948$ and $N_t^* = 100$. The quantities

$$\bar{b}(\theta) = \frac{1}{N_t^*} \sum_{k=1}^{N_t^*} b_{(N_t - N_t^* + k)}(\theta) \tag{2.2}$$

and

$$\bar{c}(\theta) = \frac{1}{N_t^*} \sum_{k=1}^{N_t^*} c_{(N_t - N_t^* + k)}(\theta) \tag{2.3}$$

are the corresponding reached states. We parameterize the integer-valued control function $\mathbf{u}(b_k, c_k; \theta)$ according to

$$\mathbf{u}(b_k, c_k; \theta) = \begin{pmatrix} -([b_k\theta_1]^+ - \{[b_k\theta_1]^+\}) \\ -([c_k\theta_2]^+ - \{[c_k\theta_2]^+\}) \end{pmatrix}, \tag{2.4}$$

where $[x]^+ = \mathrm{ReLU}(x) = \max(0, x)$. The notation $\{x\}$ denotes the fractional part of x. That is, $\{x\} = x - \lfloor x \rfloor$ if $x > 0$ and $\lfloor \cdot \rfloor$ denotes the floor function.[1]

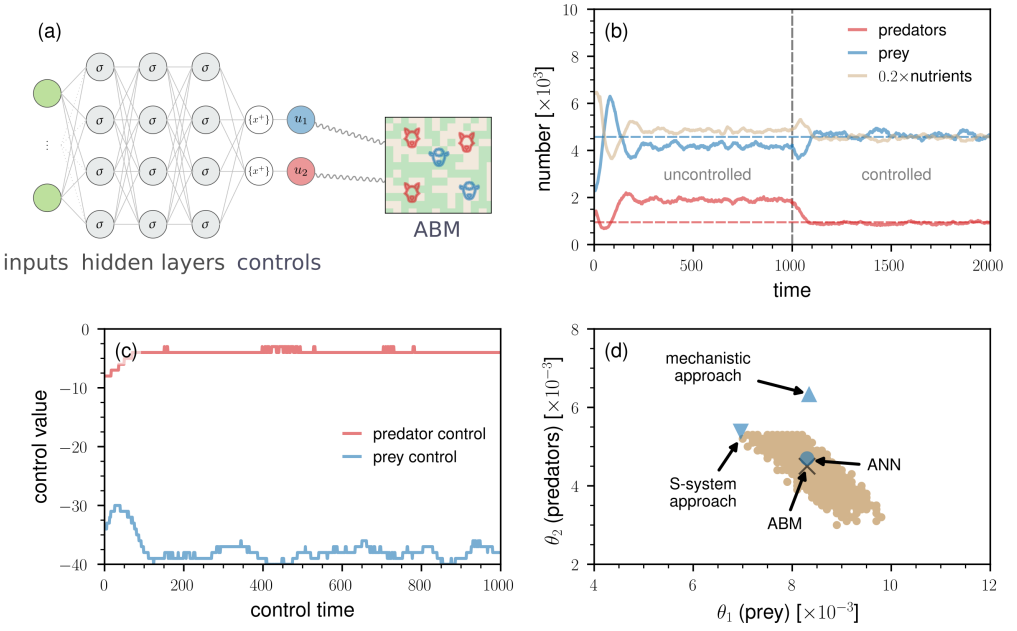

Figure 2. Control of predator–prey dynamics with an ANN. (a) To control the given predator–prey ABM, we first need to define suitable inputs and outputs for the ANN. Potential inputs are the population sizes a_k, b_k and c_k of species A, B and C at time k. We aim at directly managing the numbers of predator and prey, so there are two outputs u_1 and u_2. Using a problem-tailored straight-through estimator, the ANN outputs integer-valued controls u_1, u_2 after subtracting the fractional part $\{[\,\cdot\,]^+\}$ of the positive part of the hidden-layer outputs. We use σ and $\{x^+\}$ to indicate hidden-layer activations and the straight-through estimator, respectively. (b) The evolution of nutrient-rich lattice sites, prey, and predators based on a single instantiation of the predator–prey ABM. The vertical dashed grey line indicates the time at which the ANN controller is switched on. The controller aims at increasing the mean number of prey by 10% and reducing the mean number of predators by 50%. The dashed blue and red lines indicate the target levels of prey and predators (i.e. $\bar{b}^* = 4575$ and $\bar{c}^* = 948$), respectively. We used a 255×255 grid and set $b_0 = 2500$, $c_0 = 1250$, $\alpha_1 = 4.0$, $\alpha_2 = 5.0$, $\lambda_1 = 4.0$, $\lambda_2 = 20.0$ and $\tau = 30$ [40]. The initial proportion of nutrient-rich lattice sites is 50%. (c) The corresponding control outputs $u_1(b_k; \theta_1)$ (i.e. prey control) and $u_2(c_k; \theta_2)$ (i.e. predator control) as a function of the control time. (d) The values of θ_1 and θ_2 learned by different control methods (blue disk: ANN controller; blue triangle: mechanistic approach; blue inverted triangle: S-system approach). The black cross indicates the optimal values of the control parameters θ_1, θ_2 found via a grid search. All pairs (θ_1, θ_2) associated with controlled model trajectories were found to be one s.d. away from the optimum are coloured in brown. This brown region therefore represents the uncertainty associated with the optimal control parameters, arising from the stochastic dynamics of the underlying ABM.

We use the two control signals $u_1(b_k; \theta_1) = -([b_k\theta_1]^+ - \{[b_k\theta_1]^+\})$ and $u_2(c_k; \theta_2) = ([c_k\theta_2]^+ - \{[c_k\theta_2]^+\})$ to manage the population sizes of prey and predators, respectively. The control function (2.4) is set up such that it outputs negative integer-valued controls, meaning that a certain number of prey and predators will be removed from the ABM at each time step. More details on the training of this controller are provided in the Appendix.

The smallest loss $J_1(\theta)$ of about 74.09 achieved during training is associated with the parameters $\theta_1 = 0.0083$ and $\theta_2 = 0.0047$. The corresponding numbers of reached prey and predators are $\bar{b}(\theta) \approx 4573$ and $\bar{c}(\theta) \approx 956$.

In figure 2*c*, we show the evolution of $u_1(b_k; \theta_1)$ (i.e. prey control) and $u_2(c_k; \theta_2)$ (i.e. predator control) within the control horizon. At each time step, around 3–4 predators and between 35 and 40 predators are removed.

The learned ANN parameters $(\theta_1, \theta_2) = (0.0083, 0.0047)$ (blue disk) are close to the optimal ones (0.0083,0.0045) (black cross) (figure 2*d*). We determined the optimal parameters for prey (θ_1) and predators (θ_2) by performing a grid search over the underlying parameter space. All

pairs (θ_1, θ_2) associated with controlled model trajectories found to be one s.d. away from the optimum are coloured in brown in figure 2*d*. This brown region therefore represents the uncertainty associated with the optimal control parameters, arising from the stochastic dynamics of the underlying ABM. To quantify the uncertainty in the target quantities (i.e. the numbers of prey and predators), we evaluated the performance of the steady-state ANN controller on 50 unseen test instances of the ABM. The resulting mean values of the reached states (see equations (2.2) and (2.3)) were 4587(±71) for prey and 950(±40) for predators, which are both very close to the desired target values of $\bar{b}^* = 4575$ and $\bar{c}^* = 948$, respectively.[2] These results indicate that the controller performs well on unseen samples.

For comparison, in figure 2*d*, we also show the values of θ_1, θ_2 as identified by two ODE surrogate-modelling approaches that have been recently proposed in [26]. One surrogate model is based on a mechanistic Lotka–Volterra approximation of the ABM while the second surrogate model uses an S-system approach that is rooted in biochemical systems theory [53–55]. All ODE surrogate models that we consider in this work were trained on ABM data without assuming any knowledge on the exact parameter values. In the appendix, we provide further details on the two surrogate models. The solutions associated with both the mechanistic surrogate model (blue triangle) and the S-system approach (blue inverted triangle) are more distant from the optimum compared with the solution based on the ANN control approach. This observation holds true for the other surrogate models studied in [26].

There are two steps involved in controlling an ABM with a surrogate model. First, the surrogate model has to be trained to approximate ABM dynamics. Second, a solution to a given control problem that has been found using a surrogate model will be transferred back to the ABM. Depending on their parameterization, surrogate models may not fully capture the complex behaviour of an ABM; as a result, control solutions based on surrogate models may deviate from the desired ABM control. In contrast, the employed ANN controller directly operates on the ABM and can achieve better solutions than the ODE surrogate models considered in [26]. While the ANN controller we used in this example can achieve better solutions than surrogate-based control, not all ABM control problems can be directly addressed with this method. As an example, we will consider a control problem associated with a metabolic-network ABM [26] in §3. Instead of directly controlling the metabolic network with an ANN, we will show how to solve the control problem with a neural ODE surrogate-modelling and control approach [29,30,33–35,38].

(b) Transient control

Before focusing on the metabolic-pathway model, we briefly examine the application of the direct ANN control method that we employed in the prior section to transient dynamics. Our goal with this example is to demonstrate that the employed ANN control method can be easily adapted to different control objectives. As in the steady-state control example, we wish to increase the mean number of prey by 10% and reduce the mean number of predators by 50%. However, instead of considering a long control time horizon of 1000 time steps during which the dynamics is steered into a new steady state, we now want to adjust the mean numbers of the two species over a shorter time period of 100 time steps. The steady-state controller from the first example would not be capable of achieving this objective because it removes more prey as their numbers increase. Since increasing the number of prey is necessary to meet the control objective, the steady-state controller is unsuitable for this task. Therefore, we used a more flexible parameterization of the ANN to learn a suitable control function. To equip the ANN controller with a higher representational capacity, we use an ANN with three hidden layers

[1]Using the control function specified in equation (2.4) involves employing a problem-tailored straight-through estimator [37,50], which enables the training of ANN controllers with integer-valued outputs through backpropagation.

[2]Values in parentheses indicate the unbiased sample standard deviation.

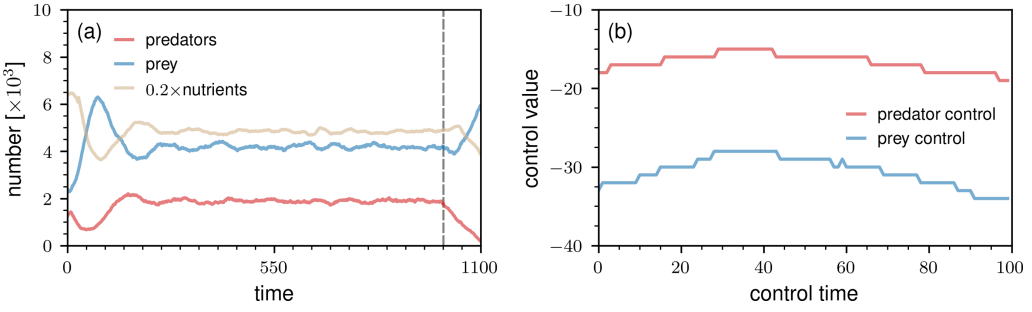

Figure 3. Control of transient predator–prey dynamics with an ANN. (a) The evolution of nutrient-rich lattice sites, prey, and predators based on a single instantiation of the predator–prey ABM. The vertical dashed grey line indicates the time at which the ANN controller is switched on. The controller aims at increasing the mean number of prey by 10% and reducing the mean number of predators by 50%. We used a 255×255 grid and set $b_0 = 2500$, $c_0 = 1250$, $\alpha_1 = 4.0$, $\alpha_2 = 5.0$, $\lambda_1 = 4.0$, $\lambda_2 = 20.0$ and $\tau = 30$. The initial proportion of nutrient-rich lattice sites is 50%. (b) The corresponding control outputs $u_1(b_k; \theta)$ (i.e. prey control) and $u_2(c_k; \theta)$ (i.e. predator control) as a function of the control time.

and between 16 and 64 neurons per layer. More details on the ANN structure and training are provided in the appendix.

In figure 3*a*, we show an example of controlled transient dynamics. The loss $J_1(\theta)$ associated with the underlying controller is about 28.00. The corresponding numbers of reached prey and predators are $\bar{b}(\theta) \approx 4576$ and $\bar{c}(\theta) \approx 953$, respectively. Examining figure 3*b*, we observe that the magnitude of the predator control signal $u_2(c_k; \theta)$ is substantially larger than in the earlier steady-state control example (figure 2*c*). On the other hand, the magnitude of the prey control signal $u_1(c_k; \theta)$ is smaller than in the prior example.

To measure the uncertainty in the target quantities (i.e. the numbers of prey and predators), we also examined the performance of the transient-dynamics ANN controller on 50 unseen test instances of the ABM. The corresponding mean numbers of prey and predators were found to be 4485(±98) and 966(±46), respectively. These results indicate that the controller performs well on unseen samples.

3. Metabolic-pathway model

We will now focus on a second ABM control problem for which the direct control approach used in the previous example cannot be implemented straightforwardly. The control problem under consideration involves a metabolic network that is based on four reactions associated with five metabolites (figure 4*a*). We model all interactions between enzymes, metabolites and their respective complexes at the microscale level.

In this ABM, we employ three types of agents: (i) metabolites, (ii) enzymes and (iii) enzymatic complexes. In total, there are five metabolites, four enzymes and 12 enzyme–metabolite complexes. The inputs to the ABM include reaction parameters as well as the initial concentrations of metabolites, enzymes and complexes.[3] The ABM outputs the concentrations of these components at each time step.

Metabolites in our model move 10 times faster than enzymes and complexes. When metabolites are in proximity to enzymes or complexes, they may bind. Additionally, complexes may dissociate into their components at any time. Enzymes have the capability to create complexes with their corresponding substrates, products and regulators. Furthermore, enzyme–

[3]The ABM incorporates more than 30 parameters, and their values are provided in the code repository accompanying this submission.

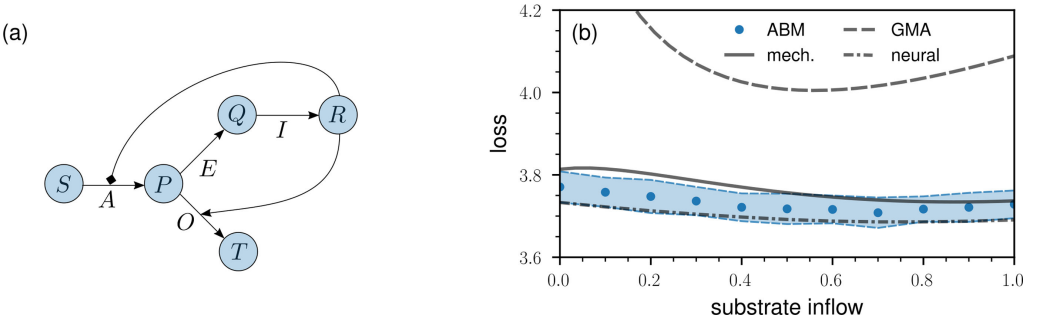

Figure 4. Learning and controlling metabolic-pathway dynamics with neural ODEs. (a) Overview of the reactions in the metabolic-pathway model. There are four reactions associated with five metabolites (S, P, Q, R and T) and four enzymes (A, E, I and O). The two arrows originating from metabolite R indicate that it inhibits enzyme A and increases the rate of enzyme O. In our ABM simulations, all reactions are modelled at the microscale level. The initial amounts of metabolites S, P, Q, R and T are 8×10^4, 2×10^4, 2×10^4, 10 and 10, respectively. The initial amount of each of the four enzymes is 200. (b). The loss of the metabolic pathway control problem (equation 3.1) as a function of the inflow of substrate S (blue dots: ABM; solid line: mechanistic surrogate model; dashed line: generalized mass action (GMA) surrogate model; dash-dotted line: neural ODE surrogate model). The blue-shaded regions represent intervals of one s.d., which are based on 100 instances of the metabolic-pathway ABM. Minimizing the loss function means minimizing substrate depletion and maximizing the production of reaction products.

substrate complexes can undergo catalysis, resulting in the formation of a complex between the enzyme and the product. We treat all four enzymatic reactions as irreversible.

The control problem that we wish to solve aims at identifying the optimal substrate inflow to minimize substrate waste and maximize the generation of reaction products. Mathematically, our objective is to determine the constant inflow of substrate, $q \in [0,1]$ per time step, that minimizes the loss function

$$J_2(q) = \frac{\sum_{k=1}^{N_t} S_k}{\sum_{k=1}^{N_t} R_k + T_k}, \tag{3.1}$$

where S_k denotes the concentration of substrate at time step k, while R_k and T_k denote the concentrations of the corresponding end products of the pathway. In all simulations, we set $N_t = 5 \times 10^4$. The initial amounts of metabolites S, P, Q, R and T are 8×10^4, 2×10^4, 2×10^4, 10 and 10, respectively. The initial amount of each of the four enzymes is 200.

(a) Neural ODE surrogate model and controller

A direct application of neural-network controllers as in the predator–prey ABM is challenging because of the various reactions that one would have to consider when keeping track of the effect of control inputs on the metabolic dynamics during training. An alternative is provided by the surrogate-modelling approach that has been proposed in [24,26]. The basic idea is to first identify control signals in ODE surrogate models and then transfer them back to an ABM. For the metabolic-pathway model, both a mechanistic Michaelis–Menten approximation and a generalized mass action (GMA) model [53–55] provide good descriptions of the underlying reactions. We define both surrogate models and describe their training in the appendix. As with the ODE surrogates discussed in the previous section, the metabolic-pathway surrogates were trained on ABM data without requiring any knowledge of the exact parameter values used in the model.

Based on the data that we show in figure 4*b*, we conclude that both surrogate models are valuable for obtaining estimates of the optimal substrate inflow. The mechanistic surrogate

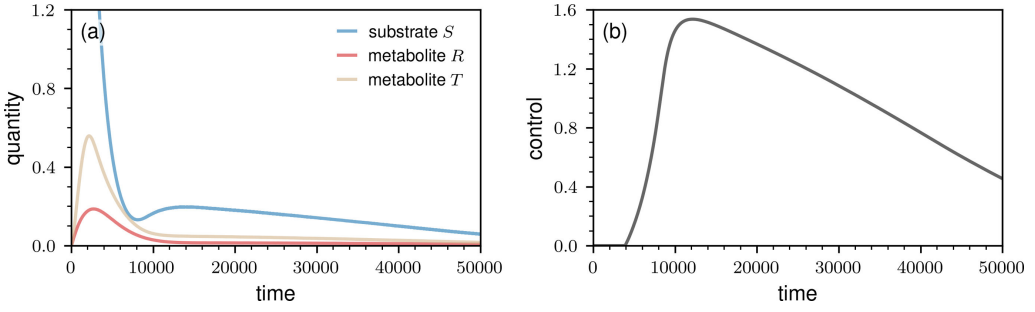

Figure 5. Controlling metabolic-pathway dynamics with a neural ODE controller. (a) We show the evolution of the amount of substrate S and the amounts of metabolites R and T. The shown amounts of S, R and T are averages over 100 instantiations of the metabolic-pathway ABM. The error bars are smaller than the line thickness. The initial amounts of metabolites S, R and T are 8×10^4, 10 and 10, respectively. In the shown plot, we rescaled these quantities by 10^{-4}. (b) The evolution of the neural ODE control signal. We trained the neural ODE controller using a mechanistic Michaelis–Menten surrogate model and the loss as defined in equation (3.1).

model and the ABM exhibit closely aligned loss values, whereas the GMA surrogate model shows substantially larger losses than those of the ABM. Despite this, the GMA approximation provides a slightly more accurate estimate of approximately 0.6 for the optimal ABM substrate inflow, which is around 0.7. In contrast, the mechanistic surrogate model suggests an optimal substrate inflow of about 0.9, a value slightly farther from the ABM optimum compared to the GMA estimate.

A complementary approach that does not require one to manually set up ODE surrogate models is based on neural ODEs, which have found applications in several tasks related to system identification and control [29–38]. Still, if mechanistic information is available, it can be easily incorporated into neural ODEs. We train a semi-mechanistic neural ODE surrogate model on ABM instances that are based on the same values of substrate inflow as in the two other surrogate models (see appendix for further details). We then used the trained neural ODE model to determine the optimal substrate inflow that minimizes the loss $J_2(q)$ (equation 3.1). The neural ODE identifies an optimal substrate inflow of 0.7, which coincides with the optimum of the ABM.[4] Neural ODE surrogate models can thus provide a valuable alternative to other surrogate models when no or only very little mechanistic information is available about the underlying ABM.

(b) Combining a mechanistic surrogate model with a neural ODE controller

Neural ODEs can also be used to directly parameterize control functions. As an example, we combine the mechanistic surrogate model of the metabolic-pathway dynamics with a neural ODE controller that outputs a time-dependent substrate inflow. The corresponding loss in the surrogate model is 3.61. This value is smaller than the minimum loss of 3.73 obtained with the optimal constant substrate inflow (figure 4*b*). To test if these loss improvements are achievable in the metabolic-pathway ABM, we fed the output of the time-dependent neural ODE controller into the ABM. The corresponding ABM loss is 3.60(±0.03), which is smaller than the minimum loss of 3.71(±0.04) achieved with a constant substrate inflow (figure 4*b*). We quantified the uncertainty in both loss values using 100 instantiations of the underlying metabolic-pathway ABM.

In figure 5, we show the evolution of the amount of substrate S, the amounts of metabolites R and T and the time-dependent neural ODE controller output $q_k(\theta)$. The initial amount of S

[4]The number of parameters in the neural ODE surrogate model is 120. In the mechanistic and GMA surrogate models, the number of parameters are 17 and 24, respectively.

is high compared with those of R and T. Minimizing the loss defined in equation (3.1) means that we have to minimize the total amount of substrate, $\sum_k S_k$, divided by the total amount of reaction products, $\sum_k R_k + T_k$. Instead of adding substrate to the system in the beginning, as was done with a constant inflow rate q, the neural ODE controller learned that adding substrate in the first few thousand time steps is not needed to achieve good loss values (figure 5*b*). The inflow of substrate $q_k(\theta)$ gets larger as the concentration of S approaches values close to those of R and T.

In summary, neural ODEs are valuable not only as surrogate models but also as control functions that can be seamlessly integrated with the former.

4. Conclusions and discussion

Control problems commonly arise in various biomedical contexts such as treatment design and pharmacology. However, conducting direct comparisons and optimizations of different treatments *in vivo* is often impractical. Rather than relying primarily on laboratory and clinical trials, medical digital twins offer a complementary approach for testing and developing treatments through *in silico* optimization.

While a diverse array of models can be integrated into medical digital twins, ABMs are among the most commonly used model types for simulating heterogeneous, multi-species biomedical systems. Despite their increasing use in biomedicine, there currently exists no general methodology to control ABMs. For example, traditional control theory methods, which have been developed for ODE systems, are not directly applicable to ABMs.

Surrogate models provide a useful approach in connecting traditional control theory with ABMs [24,26]. The main idea behind surrogate modelling involves training mechanistically inspired or more general ODE systems on ABM dynamics and solving control problems within these ODE systems. Once an appropriate control solution is identified, it is then applied back to the ABM. Similar to other function approximators, surrogate models are also subject to the bias-variance tradeoff. High-bias mechanistic approximations may be beneficial when detailed information about the inner workings of a given ABM is available, whereas lower-bias models, such as S-system approaches [53–55], may be more suitable for ABMs for which such details are unknown.

In this paper, we developed complementary ABM-control methods that use ANNs as control functions. We first considered a predator–prey ABM, where we managed the number of predators and prey by directly controlling the ABM with an ANN controller. We addressed two control tasks: (i) steady-state control and (ii) transient control. In both tasks, the ANN controller successfully identified suitable control signals. For the steady-state control task, a comparison between the surrogate-based controls proposed in [26] and the ANN controls revealed that the ANN can identify control solutions much closer to the optimum because it directly operates on the ABM without using any approximations.

In a second ABM describing metabolic-pathway dynamics, we addressed a control problem aimed at determining the optimal substrate inflow to minimize substrate depletion and maximize the generation of reaction products. For this system, we employed neural ODE surrogate models which we then used to identify suitable control signals. We found that the neural ODE approach was able to compete favourably with the best surrogate models that have been proposed in [26]. While it is possible to approximate time-series data using neural ODEs without integrating any mechanistic knowledge, considering such knowledge can improve learning performance (see e.g. [56,57]). We integrated mechanistic information into the neural surrogate model and showed that this approach could identify the optimal level of substrate inflow.

In all control approaches, the high computational costs associated with an ABM are an important factor to consider. Whether using surrogate models or operating directly on an ABM,

both approaches usually require a substantial amount of training data. The specific trade-offs involved should be carefully evaluated for each application.

Our findings suggest that ANN controllers are valuable in addressing different ABM control problems. The ability of ANNs to act as universal function approximators [58–60], combined with advancements in automatic differentiation and optimizer development, renders them well-suited for solving biomedical control problems. However, it is important to emphasize that our work should be seen as just an initial step towards solving intricate optimization and control problems associated with medical digital twins. More research is necessary to connect the proposed and related control approaches to medical digital twins, especially concerning models that are dynamically updated with patient data.

Another worthwhile direction for future work is to provide further insights into the generalization behaviour of neural ODE surrogate models [57,61,62]. For instance, in a mechanistic surrogate model, we can anticipate effective generalization across a relatively broad parameter range. However, neural-ODE and non-mechanistic surrogate models may face limitations in capturing the behaviour of a medical digital twin under parameter changes. Additionally, unlike mechanistic surrogate models, those based on neural ODEs may exhibit a higher susceptibility to overfitting. Therefore, integrating mechanistic information into a neural ODE is useful to introduce an inductive bias into the learning process.

Data accessibility. Our source codes are publicly available at [63].

Declaration of AI use. We have not used AI-assisted technologies in creating this article.

Authors' contributions. L.B.: conceptualization, formal analysis, funding acquisition, investigation, methodology, resources, software, validation, visualization, writing—original draft, writing—review and editing; L.L.F.: conceptualization, formal analysis, funding acquisition, investigation, methodology, resources, software, validation, visualization, writing—review and editing; R.C.L.: conceptualization, funding acquisition, investigation, methodology, resources, writing—review and editing.

All authors gave final approval for publication and agreed to be held accountable for the work performed therein.

Conflict of interest declaration. We declare we have no competing interests.

Funding. L.B. acknowledges financial support from hessian.AI and the Army Research Office (grant number W911NF-23-1-0129). L.L.F. and R.L. acknowledge financial support from the Defense Advanced Research Projects Agency (grant HR00112220038), and the National Institutes of Health (grants R01 GM127909 and R01 AI135128). R.L. also acknowledges financial support from the National Institutes of Health (grant R01 HL169974).

Acknowledgements. L.B. thanks Thomas Asikis for helpful discussions.

Appendix A

A.1. Neural-network architectures and training

A.1.1. Steady-state controller (predator–prey dynamics)

The ANN architecture of the steady-state controller has two inputs: (i) the number of species of type B at time k, b_k, and (ii) the number of species of type C at time k, c_k. It has two outputs $u_1(b_k;\theta_1)$ and $u_2(c_k;\theta_2)$, which are used to control the population sizes of species B and C, respectively. The controller consists of two neurons with ReLU activation function and two weights θ_1, θ_2.

We train the ANN for 50 epochs using RMSProp and a learning rate of 10^{-4}. Each training epoch takes about 3.5 min one Ryzen threadripper 3970× CPU core with a 3.7 GHz clock speed. The initial weights are both set to 5×10^{-3}.

To train the ANN using backpropagation and still output integer-valued quantities, we employ a problem-tailored straight-through estimator [37,50]. In a straight-through estimator, a mathematical operation that is applied in a forward pass is treated as an identity operation

in the backward pass. In accordance with [37,50], we implement a straight-through estimator that rounds neural-network outputs by subtracting from the positive parts $[y]^+$ of the outputs y of the last hidden layer the corresponding fractional parts $\{[y]^+\}$. That is, in a forward pass, the neural network output is $[y]^+ - \{[y]^+\}$, where $\{y\} = y - \lfloor y \rfloor$ if $y > 0$ and $\lfloor \cdot \rfloor$ denotes the floor function. While updating neural-network weights by backpropagating gradients, we detach the fractional part $\{[y]^+\}$ from the underlying computational graph such that the neural-network output is treated as $[y]^+$.

A.1.2. Transient controller (predator–prey dynamics)

The ANN architecture that we use to control transient predator–prey dynamics consists of three fully connected hidden layers with 64, 32 and 16 continuously differentiable exponential linear unit (CELU) activations each. Using CELU functions offers an advantage over traditional rectified linear units (ReLUs) because the former are continuously differentiable and produce non-zero outputs for negative arguments. This feature helps prevent the occurrence of the 'dead ReLU' problem, associated with near-zero outputs and vanishingly small gradients. Mathematically, the CELU activation is

$$\mathrm{CELU}(x, \alpha) = [x]^+ - [\alpha(1 - \exp(x/\alpha))]^+, \tag{A1}$$

which approaches $\mathrm{ReLU}(x) = [x]^+$ in the limit $\alpha \to 0^+$[64]. In this work, we set $\alpha = 1$.

The transient-dynamics ANN controller has two inputs: (i) the number of species of type B at time k, b_k, and (ii) the number of species of type C at time k, c_k. It has two outputs $u_1(b_k, c_k; \theta)$ and $u_2(b_k, c_k; \theta)$, which are used to control the population sizes of species B and C, respectively.

We train the ANN using RMSProp and a learning rate of 10^{-4}. The initial weights and biases are set to 10^{-2}. As in the steady-state controller, we use the same straight-through estimator. Loss values of about 10–100 can be achieved after 50–100 training epochs. One training epoch takes about 20 s on one Ryzen threadripper 3970× CPU core with a 3.7 GHz clock speed.

To avoid that the controller produces too large control signals in the first few training epochs, we pre-train the ANN, minimizing

$$\begin{aligned}\tilde{J}_1(\theta) = &\left[\left(\textstyle\sum_{k=1}^{10} u_1(b_k, c_k; \theta)\right)^2 - K_1\right]^2 \\ &+\left[\left(\textstyle\sum_{k=1}^{10} u_2(b_k, c_k; \theta)\right)^2 - K_2\right]^2,\end{aligned} \tag{A 2}$$

where we set $K_1 = 10^4$ and $K_2 = 10^5$. As controller inputs in the pre-training phase, we use the steady-state values of the uncontrolled dynamics (i.e. $b_k = 4159$ and $c_k = 1896$ for $k \in \{1, \ldots, 10\}$).

A.2. ODE surrogate models

As baselines for the ANN controllers that we study in this work, we use controls identified by two ODE surrogate models that have been proposed in [26]. The first surrogate-modelling approach uses a mechanistic approximation while the second one is based on an S-system approximation that is rooted in biochemical systems theory [53–55].

In the following sections, we will provide a brief overview of the mathematical forms of the surrogate models considered in this study. For further details on the training of ODE surrogate models that are not based on neural ODEs, we refer the reader to [26].

A.2.1. Mechanistic approach (predator–prey dynamics)

For the predator–prey ABM, the mechanistic surrogate model is given by

$$\begin{pmatrix}\dot{X}\\ \dot{Y}\\ \dot{Z}\end{pmatrix}=\begin{pmatrix}p_1X-p_2X^2-p_3XY\\ p_4XY-p_5Y-p_6YZ\\ p_7YZ-p_8Z\end{pmatrix}-\begin{pmatrix}0\\ \theta_1Y\\ \theta_2Z\end{pmatrix},\tag{A 3}$$

where $X \equiv X(t)$, $Y \equiv Y(t)$ and $Z \equiv Z(t)$ denote the numbers of nutrients, prey and predators at time t, respectively. It is a three-species Lotka–Volterra model with an additive control term. The model parameters are p_i ($i \in \{1, \ldots, 8\}$) and θ_1, θ_2 are the control parameters that we employ to manage the number of prey and predators. We included the model parameters in the code repository accompanying this submission.

A.2.2. S-system approach (predator–prey dynamics)

While the three-species Lotka–Volterra model in equation (A 3) has eight parameters (without counting the two control parameters θ_1, θ_2), the S-system surrogate model involves 24 parameters p_i ($i \in \{1, \ldots, 24\}$). It is given by

$$\begin{pmatrix}\dot{X}\\ \dot{Y}\\ \dot{Z}\end{pmatrix}=\begin{pmatrix}p_1X^{p_7}Y^{p_{13}}Z^{p_{19}}-p_2X^{p_8}Y^{p_{14}}Z^{p_{20}}\\ p_3X^{p_9}Y^{p_{15}}Z^{p_{21}}-p_4X^{p_{10}}Y^{p_{16}}Z^{p_{22}}\\ p_5X^{p_{11}}Y^{p_{17}}Z^{p_{23}}-p_6X^{p_{12}}Y^{p_{18}}Z^{p_{24}}\end{pmatrix}-\begin{pmatrix}0\\ \theta_1Y\\ \theta_2Z\end{pmatrix}.\tag{A 4}$$

The control term is the same as in the mechanistic approach. We included the model parameters in the code repository accompanying this submission.

A.2.3. Mechanistic approach (metabolic-pathway dynamics)

In the mechanistic surrogate model of the metabolic-pathway dynamics, we approximate the dynamics of the five metabolites $S \equiv S(t)$, $P \equiv P(t)$, $Q \equiv Q(t)$, $R \equiv R(t)$ and $T \equiv T(t)$ using a Michaelis–Menten type rate law. The surrogate model is

$$\begin{pmatrix}\dot{S}\\ \dot{P}\\ \dot{Q}\\ \dot{R}\\ \dot{T}\end{pmatrix}=\begin{pmatrix}q\\ 0\\ 0\\ 0\\ 0\end{pmatrix}+\begin{pmatrix}-1&0&0&0\\ 1&-1&0&-1\\ 0&1&-1&0\\ 0&0&1&0\\ 0&0&0&1\end{pmatrix}\begin{pmatrix}F_A\\ F_E\\ F_I\\ F_O\end{pmatrix}-\lambda\begin{pmatrix}S\\ P\\ Q\\ R\\ T\end{pmatrix},\tag{A 5}$$

where

$$\begin{aligned}F_A&=\frac{p_1\frac{S}{p_2}}{1+\frac{S}{p_2}+\frac{P}{p_3}+\frac{R}{p_4}},\\ F_E&=\frac{p_5\frac{P}{p_6}}{1+\frac{P}{p_6}+\frac{Q}{p_7}},\\ F_I&=\frac{p_8\frac{Q}{p_9}}{1+\frac{Q}{p_9}+\frac{R}{p_{10}}},\\ F_O&=\frac{p_{11}\frac{P}{p_{12}}+p_{13}\frac{R}{p_{14}}\frac{P}{p_{15}}}{1+\frac{P}{p_{12}}+\frac{T}{p_{16}}+\frac{R}{p_{14}}\left(1+\frac{P}{p_{15}}+\frac{T}{p_{17}}\right)},\end{aligned}\tag{A 6}$$

denote the fluxes through the reactions catalysed by the enzymes A, E, I and O. The quantity q is the rate of inflow of substance Q and λ is the metabolite-removal rate. The Michaelis–Menten

approximation has 17 parameters p_i ($i \in \{1, \ldots, 17\}$). The parameters are included in the code repository accompanying this submission.

A.2.4. Generalized mass action approach (metabolic-pathway dynamics)

The generalized mass action (GMA) approach [53–55] has the same mathematical structure as the mechanistic surrogate model (equation A 5), but it allows for a more straightforward determination of fluxes. Instead of the mechanistically determined fluxes (equation A 6) with 17 parameters, we use the fluxes

$$\begin{pmatrix} F_A \\ F_E \\ F_I \\ F_O \end{pmatrix} = \begin{pmatrix} p_1 S^{p_5} P^{p_9} Q^{p_{13}} R^{p_{17}} T^{p_{21}} \\ p_2 S^{p_6} P^{p_{19}} Q^{p_{14}} R^{p_{18}} T^{p_{22}} \\ p_3 S^{p_7} P^{p_{11}} Q^{p_{15}} R^{p_{19}} T^{p_{23}} \\ p_4 S^{p_8} P^{p_{12}} Q^{p_{16}} R^{p_{20}} T^{p_{24}} \end{pmatrix} \tag{A 7}$$

with 24 parameters p_i ($i \in \{1, \ldots, 24\}$) in the GMA surrogate model. The parameters are included in the code repository accompanying this submission.

A.2.5. Neural ODE surrogate model (metabolic-pathway dynamics)

In the neural ODE surrogate model, we integrate mechanistic information concerning the substrate influx at a rate of q and the outflow of all five metabolites at a rate of k. We model the remaining interactions between metabolites, enzymes, and complexes through a vector field $\mathbf{f}(\mathbf{x}; \theta)$ generated by an ANN with parameters $\theta \in \mathbb{R}^N$ and input $\mathbf{x} = (S, P, Q, R, T)^\top$. In this surrogate model, the evolution of the metabolites is described by

$$\begin{pmatrix} \dot{S} \\ \dot{P} \\ \dot{Q} \\ \dot{R} \\ \dot{T} \end{pmatrix} = \begin{pmatrix} q \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} + \mathbf{f}(\mathbf{x}; \theta) - \lambda \begin{pmatrix} S \\ P \\ Q \\ R \\ T \end{pmatrix}. \tag{A 8}$$

The ANN associated with $\mathbf{f}(\mathbf{x}; \theta)$ consists of two fully connected hidden layers with five CELU activations each. Further details on the training are reported in the code repository accompanying this submission.

A.3. Combining a mechanistic surrogate model with a neural ODE controller

Instead of using a constant substrate inflow q in the mechanistic metabolic-pathway surrogate model (equation A 5), another option is to consider a neural ODE controller that takes the current time t as an input and outputs a time-dependent substrate inflow $q(t; \theta)$, where θ are the parameters of the ANN. The resulting evolution equation is

$$\begin{pmatrix} \dot{S} \\ \dot{P} \\ \dot{Q} \\ \dot{R} \\ \dot{T} \end{pmatrix} = \begin{pmatrix} q(t; \theta) \\ 0 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} -1 & 0 & 0 & 0 \\ 1 & -1 & 0 & -1 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} F_A \\ F_E \\ F_I \\ F_O \end{pmatrix} - \lambda \begin{pmatrix} S \\ P \\ Q \\ R \\ T \end{pmatrix}. \tag{A 9}$$

The ANN associated with $q(t; \theta)$ consists of five fully connected hidden layers with five exponential linear units (ELUs) each. We train the neural ODE controller using the loss function

defined in equation (3.1). Subsequently, we use the trained controller as input in the metabolic-pathway ABM. Further details on the ODE controller are reported in the code repository accompanying this submission.

References

1. Kosorok MR, Laber EB. 2019 Precision Medicine. *Annu. Rev. Stat. Its Appl.* **6**, 263–286. (doi:10.1146/annurev-statistics-030718-105251)
2. Tsiatis AA, Davidian M, Holloway ST, Laber EB. 2019 Boca Raton, FL, UA: Chapman and Hall/CRC. (doi:10.1201/9780429192692-4)
3. Björnsson B *et al.* 2020 Digital twins to personalize medicine. *Genome Med.* **12**, 2. (doi:10.1186/s13073-019-0701-3)
4. Goecks J, Jalili V, Heiser LM, Gray JW. 2020 How Machine Learning Will Transform Biomedicine. *Cell* **181**, 92–101. (doi:10.1016/j.cell.2020.03.022)
5. Coorey G, Figtree GA, Fletcher DF, Redfern J. 2021 The health digital twin: advancing precision cardiovascular medicine. *Nat. Rev. Cardiol.* **18**, 803–804. (doi:10.1038/s41569-021-00630-4)
6. Hernandez-Boussard T, Macklin P, Greenspan EJ, Gryshuk AL, Stahlberg E, Syeda-Mahmood T, Shmulevich I. 2021 Digital twins for predictive oncology will be a paradigm shift for precision *cancer* care. *Nat. Med.* **27**, 2065–2066. (doi:10.1038/s41591-021-01558-5)
7. Laubenbacher R, Sluka JP, Glazier JA. 2021 Using digital twins in viral infection. *Science* **371**, 1105–1106. (doi:10.1126/science.abf3370)
8. Masison J *et al.* 2021 A modular computational framework for medical digital twins. *Proc. Natl. Acad. Sci. USA* **118**, e2024287118. (doi:10.1073/pnas.2024287118)
9. Oremland M, Michels KR, Bettina AM, Lawrence C, Mehrad B, Laubenbacher R. 2016 A computational model of invasive aspergillosis in the lung and the role of iron. *BMC Syst. Biol.* **10**, 34. (doi:10.1186/s12918-016-0275-2)
10. An G, Cockrell RC. 2021 Agent-based modeling of systemic inflammation: a pathway toward controlling Sepsis. In *Sepsis: methods and protocols*, pp. 231–257. New York, NY: Springer. (doi:10.1007/978-1-0716-1488-4_20)
11. Ribeiro HA, Vieira LS, Scindia Y, Adhikari B, Wheeler M, Knapp A, Schroeder W, Mehrad B, Laubenbacher R. 2022 Multi-scale mechanistic modelling of the host defence in invasive aspergillosis reveals leucocyte activation and iron acquisition as drivers of infection outcome. *J. R. Soc. Interface* **19**, 20210806. (doi:10.1098/rsif.2021.0806)
12. Joslyn LR, Linderman JJ, Kirschner DE. 2022 A virtual host model of *Mycobacterium tuberculosis* infection identifies early immune events as predictive of infection outcomes. *J. Theor. Biol.* **539**, 111042. (doi:10.1016/j.jtbi.2022.111042)
13. Joslyn LR, Flynn JL, Kirschner DE, Linderman JJ. 2022 Concomitant immunity to M. tuberculosis infection. *Sci. Rep.* **12**, 20731. (doi:10.1038/s41598-022-24516-8)
14. Ribeiro HAL, Scindia Y, Mehrad B, Laubenbacher R. 2023 COVID-19-associated pulmonary aspergillosis in immunocompetent patients: a virtual patient cohort study. *J. Math. Biol.* **87**, 6. (doi:10.1007/s00285-023-01940-6)
15. Budak M *et al.* 2023 Optimizing tuberculosis treatment efficacy: Comparing the standard regimen with Moxifloxacin-containing regimens. *PLoS Comput. Biol.* **19**, e1010823. (doi:10.1371/journal.pcbi.1010823)
16. West J, Robertson-Tessi M, Anderson ARA. 2023 Agent-based methods facilitate integrative science in *cancer*. *Trends Cell Biol.* **33**, 300–311. (doi:10.1016/j.tcb.2022.10.006)
17. Weston B, Fogal B, Cook D, Dhurjati P. 2015 An agent-based modeling framework for evaluating hypotheses on risks for developing autism: Effects of the gut microbial environment. *Med. Hypotheses* **84**, 395–401. (doi:10.1016/j.mehy.2015.01.027)
18. Bauer E, Zimmermann J, Baldini F, Thiele I, Kaleta C. 2017 BacArena: Individual-based metabolic modeling of heterogeneous microbes in complex communities. *PLoS Comput. Biol.* **13**, e1005544. (doi:10.1371/journal.pcbi.1005544)

19. Lin C, Culver J, Weston B, Underhill E, Gorky J, Dhurjati P. 2018 GutLogo: Agent-based modeling framework to investigate spatial and temporal dynamics in the *gut microbiome*. *PLoS one* **13**, e0207072. (doi:10.1371/journal.pone.0207072)
20. Archambault L, Koshy-Chenthittayil S, Thompson A, Dongari-Bagtzoglou A, Laubenbacher R, Mendes P. 2021 Understanding *Lactobacillus paracasei* and *Streptococcus oralis* Biofilm Interactions through Agent-Based Modeling. *mSphere* **6**, e00875-21. (doi:10.1128/msphere.00875-21)
21. Ogata K. 2009 Modern Control Engineering. 5th edn. London, UK: Pearson.
22. Åström KJ, Murray RM. 2021 Feedback Systems: An Introduction for Scientists and Engineers. Princeton, MA, USA: Princeton University Press.
23. National Academies of Sciences, Engineering, and Medicine. 2024 *Foundational research gaps and future directions for digital twins*. Washington, DC, USA: National Academies Press. (doi:10.17226/26894)
24. 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)
25. 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. (doi:10.1098/rsif.2020.0987)
26. Fonseca LL, Böttcher L, Mehrad B, Laubenbacher RC. 2025 Optimal control of agent-based models via surrogate modeling. *PLOS Comput. Biol* **21**, e1012138. (doi:10.1371/journal.pcbi.1012138)
27. Ivakhnenko AG. 1971 Polynomial Theory of Complex Systems. *IEEE Trans. Syst. Man Cybern.* **SMC-1**, 364–378. (doi:10.1109/tsmc.1971.4308320)
28. Schmidhuber J. 2015 Deep learning in neural networks: An overview. *Neural Networks* **61**, 85–117. (doi:10.1016/j.neunet.2014.09.003)
29. Wang YJ, Lin CT. 1998 Runge-Kutta neural network for identification of dynamical systems in high accuracy. *IEEE Trans. Neural Netw.* **9**, 294–307. (doi:10.1109/72.661124)
30. Chen TQ, Rubanova Y, Bettencourt J, Duvenaud D. 2018 Neural ordinary differential equations (eds S Bengio, HM Wallach, H Larochelle, K Grauman, N Cesa-Bianchi, R Garnett). In *Advances in neural information processing systems 31: annual conference on neural information processing sstems, neurIIPS 2018, December 3-8, 2018*, pp. 6572–6583.
31. 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**, e1008462. (doi:10.1371/journal.pcbi.1008462)
32. Fronk C, Petzold L. 2023 Interpretable polynomial neural ordinary differential equations. *Chaos* **33**, 043101. (doi:10.1063/5.0130803)
33. Böttcher L, Antulov-Fantulin N, Asikis T. 2022 AI Pontryagin or how *artificial* neural networks learn to control dynamical systems. *Nat. Commun.* **13**, 9. (doi:10.1038/s41467-021-27590-0)
34. Asikis T, Böttcher L, Antulov-Fantulin N. 2022 Neural ordinary differential equation control of dynamics on graphs. *Phys. Rev. Res.* **4**, 013221. (doi:10.1103/physrevresearch.4.013221)
35. Böttcher L, Asikis T. 2022 Near-optimal control of dynamical systems with neural ordinary differential equations. *Mach. Learn.* **3**, 045004. (doi:10.1088/2632-2153/ac92c3)
36. Mowlavi S, Nabi S. 2023 Optimal control of PDEs using physics-informed neural networks. *J. Comput. Phys.* **473**, 111731. (doi:10.1016/j.jcp.2022.111731)
37. Böttcher L, Asikis T, Fragkos I. 2023 Control of Dual-Sourcing Inventory Systems Using Recurrent Neural Networks. *INFORMS J. Comput.* **35**, 1308–1328. (doi:10.1287/ijoc.2022.0136)
38. Böttcher L. 2023 Gradient-free training of neural ODEs for system identification and control using ensemble Kalman inversion. In *ICML workshop on new frontiers in learning, control, and dynamical systems*. Honolulu, HI, USA.
39. Bergman DR, Norton KA, Jain HV, Jackson T. 2023 Connecting Agent-Based Models with High-Dimensional Parameter Spaces to Multidimensional Data Using SMoRe ParS: A Surrogate Modeling Approach. *Bull. Math. Biol.* **86**, 11. (doi:10.1007/s11538-023-01240-6)
40. Wilensky U. 1997 *NetLogo wolf sheep predation model*. Evanston, IL: Center for Connected Learning and Computer-Based Modeling, Northwestern University.
41. Wilensky U. 1999 *NetLogo*. See http://ccl.northwestern.edu/netlogo/.

42. May RM, Leonard WJ. 1975 Nonlinear Aspects of Competition Between Three Species. *SIAM J. Appl. Math.* **29**, 243–253. (doi:10.1137/0129022)
43. Faust K, Raes J. 2012 Microbial interactions: from networks to models. *Nat. Rev. Microbiol.* **10**, 538–550. (doi:10.1038/nrmicro2832)
44. Cortez MH, Weitz JS. 2014 Coevolution can reverse predator–prey cycles. *Proc. Natl. Acad. Sci. USA* **111**, 7486–7491. (doi:10.1073/pnas.1317693111)
45. Joseph TA, Shenhav L, Xavier JB, Halperin E, Pe'er I. 2020 Compositional Lotka-Volterra describes microbial dynamics in the simplex. *PLoS Comput. Biol.* **16**, e1007917. (doi:10.1371/journal.pcbi.1007917)
46. Jin C, Allen-Zhu Z, Bubeck S, Jordan MI. 2018 Is Q-learning provably efficient? (eds S Bengio, HM Wallach, H Larochelle). In *Annual Conference on Neural Information ProcessingSystems 2018, NeurIPS 2018, December 3-8*, Montréal, Canada, pp. 4868–4878. https://proceedings.neurips.cc/paper/2018/hash/d3b1fb02964aa64e257f9f26a31f72cf-Abstract.html.
47. Yarats D, Zhang A, Kostrikov I, Amos B, Pineau J, Fergus R. 2021 Improving Sample Efficiency in Model-Free Reinforcement Learning from Images. pp. 10674–10681. (doi:10.1609/aaai.v35i12.17276)
48. Bengio Y, Léonard NCourvilleAC2015 Estimating or propagating gradients throughstochastic neurons for conditional computation. *CoRR* 1308–3432. (doi:10.3390/computation3010001)
49. Yin P, Lyu J, Zhang S, J.Osher S, Qi Y, Xin J. 2019 Understanding straight-through estimator in training activation quantized neural nets. In *7th International Conference on Learning Representations, ICLR 2019*. New Orleans, LA, USA. 6-9, https://openreview.net/forum?id=Skh4jRcKQ.
50. Asikis T. 2023 *"Towards recommendations for value sensitive sustainable consumption."NeurIPS 2023 Workshop on Tackling Climate Change with Machine Learning: Blending New and Existing Knowledge Systems*. See https://nips.cc/virtual/2023/76939.
51. Dyer J, Quera-Bofarull A, Chopra A, Farmer JD, Calinescu A, Wooldridge M. 2023 Gradient-Assisted Calibration for Financial Agent-Based Models. In *ICAIF '23*, Brooklyn NY USA. New York, NY, USA: ACM. (doi:10.1145/3604237.3626857). https://dl.acm.org/doi/proceedings/10.1145/3604237.
52. Pekalski A, Stauffer D. 1998 Three Species Lotka–Volterra Model. *Int. J. Mod. Phys. C* **09**, 9. (doi:10.1142/s0129183198000674)
53. Savageau MA. 1969 Biochemical systems analysis. I. Some mathematical properties of the rate law for the component enzymatic reactions. *J. Theor. Biol.* **25**, 69. (doi:10.1016/s0022-5193(69)80026-3)
54. Savageau MA. 1970 Biochemical systems analysis. 3. Dynamic solutions using a power-law approximation. *J. Theor. Biol.* **26**, 215–226. (doi:10.1016/s0022-5193(70)80013-3)
55. Voit EO. 2013 Biochemical Systems Theory: A Review. *ISRN Biomath.* **2013**, 1–53. (doi:10.1155/2013/897658)
56. Chee KY, Jiahao TZ, Hsieh MA. 2022 KNODE-MPC: A Knowledge-Based *Data*-Driven Predictive Control Framework for Aerial Robots. *IEEE Robot. Autom. Lett.* **7**, 2819–2826. (doi:10.1109/lra.2022.3144787)
57. Chee KY, Hsieh MA, Pappas GJ. 2023 Uncertainty Quantification for Learning-based MPC using Weighted Conformal Prediction. In *2023 62nd IEEE Conference on Decision and Control (CDC)*, Singapore, pp. 342–349. IEEE. (doi:10.1109/CDC49753.2023.10383587)
58. Hornik K. 1991 Approximation capabilities of multilayer feedforward networks. *Neural Netw.* **4**, 251–257. (doi:10.1016/0893-6080(91)90009-T)
59. Hanin B, Sellke M. 2017 Approximating Continuous Functions by ReLU Nets of Minimal Width. *arXiv Preprint*. See https://arxiv.org/abs/1710.11278.
60. Park S, Yun C, Lee J, Shin J. 2020 Minimum Width for Universal Approximation. *arXiv Preprint*. See https://arxiv.org/abs/2006.08859.
61. Hammouri ZAA, Mier PR, Félix P, Mansournia MA, Huelin F, Casals M, Matabuena M. 2023 Uncertainty Quantification in Medicine Science: The Next Big Step. *Arch. De Bronconeumol.* **59**, 760–761. (doi:10.1016/j.arbres.2023.07.018)

62. Matabuena M, Díaz-Louzao C, Ghosal R, Gude F. 2024 Personalized Imputation in Metric Spaces via Conformal Prediction: Applications in Predicting Diabetes Development with Continuous Glucose Monitoring Information. *arXiv Preprint*. See https://arxiv.org/abs/2403.18069.
63. Böttcher L, Fonseca LL, Laubenbacher R. 2024 Control of medical digital twins with artificial neural networks. Preprint https://gitlab.com/Computational Science/abm-control
64. Barron JT. 2017 Continuously Differentiable Exponential Linear Units. *arXiv Preprint*. See https://arxiv.org/abs/1704.07483.