Developing Surrogates for Epidemic Agent-Based Models via Scientific Machine Learning

Sharv Murgai
Monta Vista High School
murgai.sharv@gmail.com

Utkarsh
MIT CSAIL
utkarsh5@mit.edu

Kyle C. Nguyen
Sandia National Laboratories
kcnguye@sandia.gov

Alan Edelman
MIT CSAIL
edelman@mit.edu

Erin C. S. Acquesta
Sandia National Laboratories
eacques@sandia.gov

Christopher Vincent Rackauckas
MIT CSAIL
crackauc@mit.edu

Abstract

Forecasting epidemic trajectories is challenging due to nonlinear dynamics, abrupt interventions, and limited data. Classical compartmental models are interpretable but rigid, while purely data-driven methods often lack physical consistency. We propose a hybrid framework based on Universal Differential Equations (UDEs), which augment epidemiological Ordinary Differential Equations (ODEs) with neural networks to capture unresolved dynamics. To overcome instability in standard single-shooting training, we introduce two advances: (i) a multiple-shooting (MS) scheme that segments trajectories and enforces continuity across intervals, and (ii) a prediction error method that iteratively corrects forward simulations. Applied to epidemic datasets, our approach reduces symptomatic prediction error by 77.8% with multiple shooting and 82.1% with the prediction error method (PEM), particularly around intervention periods such as lockdowns. Multiple shooting improves stability and efficiency, while the prediction error method achieves the most accurate ground-truth alignment at a higher computational cost. These results demonstrate how MS and PEM methods could improve the fitting of UDEs, which will help bridge the gap between purely mechanistic models and agent-based models, and enable interpretable and accurate epidemic forecasting.

1 Introduction

Accurately simulating and forecasting the trajectory of an epidemic remains a central challenge in public health, directly shaping policy decisions that affect millions of people. The task is complicated by nonlinear transmission dynamics, abrupt interventions such as lockdowns, and sparse or noisy data. Agent-based models (ABMs) offer a powerful framework by representing heterogeneous agents and their interactions, enabling the emergence of complex behaviors such as variable compliance, localized outbreaks and interventions, and community-level heterogeneity [23, 8, 7]. However, ABMs are computationally intensive and challenging to calibrate at scale. As a result, there is ongoing interest in developing surrogate or approximate models for ABMs that preserve key dynamics while being more computationally scalable [14, 15, 5]. Traditional ODE models, such as the SIR model and its variations, are often derived as approximate models of ABMs. While being easier to calibrate and can provide interpretability and physical grounding, ODE models often assume homogeneous mixing, are less flexible, and may fail to capture complex local dynamics of ABMs [14, 15]. In contrast, purely data-driven machine learning approaches can adapt to complex signals, but they risk overfitting and frequently produce predictions that violate known biological principles [25].

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: Machine Learning and the Physical Sciences.

To bridge these extremes, Universal Differential Equations (UDEs) embed neural networks within mechanistic ODEs [19]. Compared to the ODEs framework, this hybrid modeling paradigm retains interpretability while allowing unresolved dynamics, such as behavioral changes, waning immunity, or reporting delays, to be learned directly from data. In addition, the UDE framework enables computation of the time-varying reproduction number and global sensitivity analysis of disease parameters, providing insights for policy makers that are often infeasible with purely data-driven machine learning models or ABMs. However, training UDEs is notoriously difficult [19, 4]. Standard single-shooting optimization, which fits neural parameters across full trajectories, is prone to instability, exploding gradients, and poor convergence [9, 26]. These issues are particularly acute in epidemic systems, where abrupt shifts in behavior or interventions induce discontinuities [15].

In this work, we introduce two complementary methodological advances that address these challenges. First, we develop a multiple-shooting (MS) scheme [21], which segments trajectories into shorter intervals and enforces continuity across boundaries, improving stability and robustness in the presence of sharp transitions. Second, we adapt the prediction error method (PEM) from system identification [13, 12], which iteratively corrects forward simulations to better align short-term predictions with observed data.

Together, these approaches stabilize training and improve the fidelity of hybrid epidemiological models. More broadly, our work demonstrates how physics-informed machine learning can enhance epidemic forecasting by combining mechanistic epidemiology with modern neural network methods.

We summarize our contributions as follows:

1. **Universal Differential Equations for epidemic modeling**: We develop UDEs as surrogates for exascale agent-based epidemic simulations, bridging mechanistic epidemiology with neural components to capture complex local dynamics while preserving interpretability.
2. **Stabilizing training for UDEs**: We adapt multiple shooting and the prediction error method (PEM) to the UDE setting, reducing symptomatic prediction error by 77.8% through MS, and 82.1% through PEM, compared to single shooting and improving alignment of critical compartments such as deaths.
3. **Systematic evaluation on ABM data**: Using ExaEpi [1], we present the first comparative study of vanilla UDEs, MS-UDEs, and PEM-UDEs on epidemic ABM outputs, quantifying stability–accuracy trade-offs and demonstrating the potential of scientific machine learning surrogates in epidemiology.

Why ABM surrogates are needed. Operational calibration of agent-based models (ABMs) requires repeated stochastic simulations over large, realistic contact networks. Even with HPC scaling, full ABM runs are costly; for example, Loimos simulates a 200-day California digital twin in $\sim$42 s on 4096 Perlmutter cores (4.6B TEPS) [11, 10], and CityCOVID reports $\sim$100 CPU-hours for a single 70-day run [20]. Routine tasks such as parameter estimation, sensitivity analysis, and "what-if" sweeps require *many* such runs, making near real-time iteration impractical [2, 16]. Our UDE surrogates reduce wall-clock by orders of magnitude while preserving population-scale constraints (mass balance, nonnegativity) and enabling downstream analyses [18].

2 Methodology

2.1 Baseline UDE Training

As a baseline, we consider a UDE model in which a neural network (NN), with an input size of 6 neurons, and 3 hidden layers of 10 neurons each, augments the mechanistic S, E, I_p, I_s, I_a, D, and R dynamics (see Appendix A.1 and Appendix A.2). The UDEs training and NN inputs utilized all the state variables except for the S trajectory. Training is performed end-to-end using single-shooting optimization, where gradients are backpropagated across the full trajectory. While conceptually straightforward and shown to outperform its fully mechanistic version [15], this approach is unstable in practice: small errors accumulate over long horizons, leading to exploding or vanishing gradients and poor convergence. These issues are especially severe in epidemic systems where interventions induce abrupt shifts in transmission behavior.

We benchmark against a vanilla UDE trained by single shooting over the full trajectory. This is the canonical single–shooting baseline in system identification and UDE training [13, 19], and it reflects the approach practitioners would try first. We also considered purely data-driven baselines (e.g., GP surrogates, direct neural sequence models), but these lack mechanistic interpretability and do not support epidemiological analyses such as sensitivity studies or $\mathcal{R}_t$ estimation. Thus, the single–shooting UDE is the most appropriate mechanistic baseline; our contributions (MS, PEM) are plug-in training strategies that improve that baseline.

2.2 Multiple Shooting (MS)

To address instability in single shooting, we adapt a multiple-shooting (MS; see Appendix A.3) scheme for UDE training. The time horizon is divided into shorter segments, each solved independently. Continuity is enforced at the boundaries to ensure global consistency. This segmentation reduces the effective integration horizon for gradient backpropagation, mitigating numerical instability and improving robustness to abrupt regime changes such as lockdowns or behavioral shifts. Although multiple shooting is a classical technique in numerical analysis, our contribution lies in formulating it for neural-augmented epidemiological models, where continuity constraints propagate through neural network parameters embedded in mechanistic dynamics.

2.3 Prediction Error Method (PEM)

The prediction error method (PEM) is a classical system identification technique for fitting dynamical models to data [13, 12]. It augments the dynamics with a correction term that incorporates the instantaneous mismatch between predictions and observations, thereby stabilizing training and reducing long-horizon error accumulation. This structure is closely related to observer formulations such as the extended Kalman filter [24], but PEM typically uses a simpler, fixed-gain correction that avoids the difficulty of tuning time-varying gains. The resulting optimization minimizes the accumulated prediction error, often using mean squared error as the loss. Further details on the observer formulation and adaptation to UDEs from ABMs are provided in the Appendix A.4.

3 Results

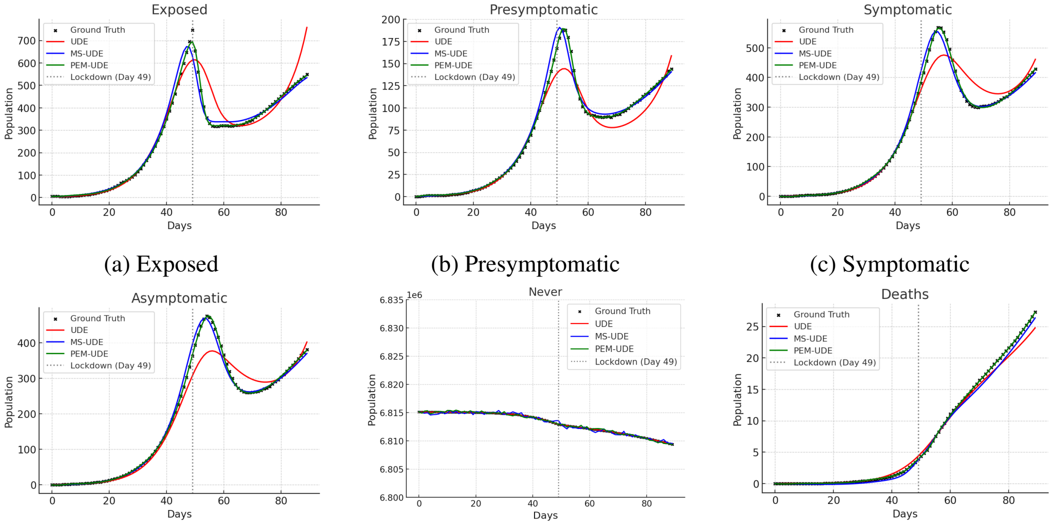
The implementation details are available in Appendix A.6 and the data setup is discussed in Appendix A.7. We discuss our insights on the results in the following sections.

3.1 Baseline Results

The baseline UDE trained with single-shooting optimization provides a mixed quality of fit across compartments. As shown in Figure 1, the model captures the general rising and falling trends of the Exposed, Presymptomatic, and Symptomatic compartments, but it fails to reproduce the precise peak magnitudes and timing. The Asymptomatic state exhibits noticeable deviations in both amplitude and curvature, while the Presymptomatic compartment drifts away from the ground-truth trajectory, reflecting accumulated errors over long horizons. Although the Deaths and Never compartments show better agreement than other states, the overall performance highlights the inherent limitations of single-shooting UDE training: small discrepancies early in the trajectory propagate forward, leading to unstable optimization and systematic mismatches as noted complementary in Nguyen et al. [15]. These shortcomings motivate the introduction of more robust training strategies such as multiple shooting and the prediction error method.

3.2 Multiple Shooting Results

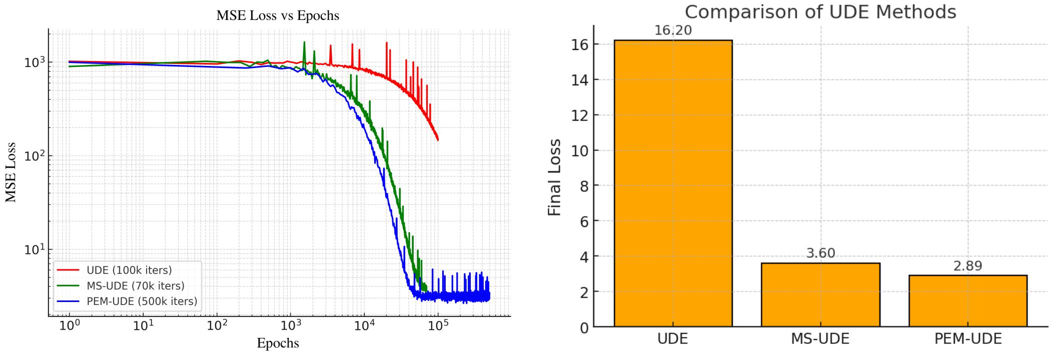
Figure 1 presents results obtained with the multiple-shooting (MS) framework. Compared to the baseline, MS-UDE achieves substantially improved alignment across all compartments, including the previously unstable Presymptomatic and Asymptomatic states. The Exposed population, which was poorly fitted under single shooting, is captured more faithfully when continuity across shooting intervals is enforced. These results confirm that segmenting long trajectories into shorter intervals stabilizes gradient propagation and mitigates the accumulation of integration errors.



(a) Exposed (b) Presymptomatic (c) Symptomatic

(d) Asymptomatic (e) Never (f) Deaths

Figure 1: Comparison of ExaEpi ground truth trajectories with UDE-based reconstructions for the E, $I_{\sim s}$, I_s, I_a, S, and D compartments. Black markers denote ground truth averages, while colored lines represent Baseline UDE, MS-UDE, and PEM-UDE predictions. The plots highlight model accuracy across different states under different training strategies.



(a) (b)

Figure 2: (a) Convergence of MSE loss and (b) final losses for UDE, MS-UDE and PEM-UDE.

3.3 Prediction Error Method Results

Figure 1 illustrates the results of training with the prediction error method (PEM). By anchoring the model repeatedly to short-horizon observations, PEM-UDE achieves the most accurate reconstructions overall. This is especially evident in the Symptomatic and Deaths compartments, which are critical for evaluating public health outcomes. The method also improves fidelity in the Never compartment compared to both baseline UDE and MS-UDE, showing that PEM helps reduce long-term drift while retaining short-term alignment.

3.4 Loss Comparison

Figure 2b summarizes the final losses for all methods, and Figure 2a shows the corresponding learning curves. The baseline single-shooting UDE converges slowly and plateaus at a relatively high error (16.20). In contrast, MS-UDE achieves a markedly lower final loss of 3.60, and PEM-UDE further improves this result to 2.89. The learning curves confirm that both MS and PEM not only converge faster but also yield more accurate and stable training outcomes. These quantitative improvements are consistent with the qualitative trajectory reconstructions shown in the compartment plots.

3.5 Practical guidance: MS vs. PEM

Both MS and PEM improve stability and accuracy, but they favor different regimes. Table 1 summarizes trade-offs we observed.

Table 1: Guidance for choosing MS vs. PEM (empirical trends from our study). H = measurement operator mapping states to observed components.

Method	Strengths	Limitations	Recommended use
MS-UDE	Stable gradients on long horizons; mitigates drift; robust to regime shifts via continuity constraints	Slightly higher compute per epoch; windowing hyperparameters; in this work we assume full-state observability at window boundaries; with partial observations MS works by estimating latent window states and using a measurement model H.	Multi-wave or long-window scenarios; scarce/noisy observations
PEM-UDE	Highest short-horizon fidelity; strong alignment to observed bands; fastest apparent convergence	Requires tuning observer gain; can over-anchor if gains too large	Noisy data with frequent observations; rapid regime changes

4 Conclusion and Future Work

In this work, we investigated the challenges of training UDEs for epidemiological ABMs and showed that single-shooting optimization often fails due to instability and long-horizon error accumulation. We introduced two complementary strategies, MS-UDE, which improves stability by segmenting trajectories, and PEM-UDE, which anchors forecasts to observations: both of which significantly enhance reconstruction accuracy, with PEM-UDE yielding the best alignment.

Future work will evaluate these methods on broader epidemic datasets and explore richer noise models, scalability to larger populations, and GPU-parallelized implementations [22], which are compatible but require detailed investigation. In addition, coupling UDEs with symbolic recovery of terms offers a promising path for recovering interpretable governing equations [3, 6]. More broadly, these techniques extend beyond epidemic modeling to other areas of scientific machine learning where stability and interpretability are essential.

Limitations. Current experiments use averaged synthetic trajectories from a single ABM scenario. Broader external validation (multiple settings and real hospitalization/ICU/case datasets), energy/time accounting across methods, and automated tuning of PEM gains are left to future work. Our MS implementation assumes full-state observability at window boundaries (or an explicit latent-state estimation scheme); general MS can handle partial observations with latent-state estimation, which we leave to future work.

References

[1] GitHub - AMReX-Agent/ExaEpi: An agent-based epidemiological simulation code using AMReX — github.com. `https://github.com/AMReX-Agent/ExaEpi`. [Accessed 29-08-2025].

[2] Keith R. Bissett, Jose Cadena, Maleq Khan, and Chris J. Kuhlman. Agent-based computational epidemiological modeling. *Computing in Science & Engineering*, 23(6):26–38, 2021. Discusses need for many runs for sensitivity/analysis at large scales.

[3] Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the national academy of sciences*, 113(15):3932–3937, 2016.

[4] Anthony G Chesebro, David Hofmann, Vaibhav Dixit, Earl K Miller, Richard H Granger, Alan Edelman, Christopher V Rackauckas, Lilianne R Mujica-Parodi, and Helmut H Strey. Scientific machine learning of chaotic systems discovers governing equations for neural populations. *arXiv preprint arXiv:2507.03631*, 2025.

[5] Maria-Veronica Ciocanel, John T. Nardini, Kevin B. Flores, Erica M. Rutter, Suzanne S. Sindi, and Alexandria Volkening. Enhancing generalizability of model discovery across parameter space with multi-experiment equation learning (ME-EQL), 2025. URL `https://arxiv.org/abs/2506.08916`. Version Number: 1.

[6] Miles Cranmer. Interpretable machine learning for science with pysr and symbolicregression. jl. *arXiv preprint arXiv:2305.01582*, 2023.

[7] Timothy C Germann, Kai Kadau, Ira M Longini Jr, and Catherine A Macken. Mitigation strategies for pandemic influenza in the united states. *Proceedings of the National Academy of Sciences*, 103(15):5935–5940, 2006.

[8] Volker Grimm and Steven F Railsback. Agent-based models in ecology: patterns and alternative theories of adaptive behaviour. In *Agent-based computational modelling: Applications in demography, social, economic and environmental sciences*, pages 139–152. Springer, 2006.

[9] Victor Isakov. *Inverse problems for partial differential equations*. Springer, 2006.

[10] Joy Kitson, Ian Costello, Jiangzhuo Chen, Diego Jiménez, Stefan Hoops, Henning Mortveit, Esteban Meneses, Jae-Seung Yeom, Madhav V. Marathe, and Abhinav Bhatele. Pandemics in silico: Scaling an agent-based simulation on realistic social contact networks, 2024. Preprint version with the same performance figure.

[11] Joy Kitson, Ian Costello, Jiangzhuo Chen, Diego Jiménez, Stefan Hoops, Henning Mortveit, Esteban Meneses, Jae-Seung Yeom, Madhav V. Marathe, and Abhinav Bhatele. Pandemics in silico: Scaling an agent-based simulation on realistic social contact networks. In *2025 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, 2025. doi: 10.1109/IPDPS64566.2025.00050. Loimos: 200-day CA digital twin in ~42 s on 4096 Perlmutter cores; 4.6B TEPS.

[12] Roger Larsson, Zoran Sjanic, Martin Enqvist, and Lennart Ljung. Direct prediction-error identification of unstable nonlinear systems applied to flight test data. *IFAC Proceedings Volumes*, 42(10):144–149, 2009.

[13] Lennart Ljung. Prediction error estimation methods. *Circuits, Systems and Signal Processing*, 21(1):11–21, 2002.

[14] John T. Nardini, Ruth E. Baker, Matthew J. Simpson, and Kevin B. Flores. Learning differential equation models from stochastic agent-based model simulations. *Journal of The Royal Society Interface*, 18(176):rsif.2020.0987, 20200987, March 2021. ISSN 1742-5662. doi: 10.1098/rsif.2020.0987. URL `https://royalsocietypublishing.org/doi/10.1098/rsif.2020.0987`.

[15] Kyle Cuongthe Nguyen, Kyle Thomas Ritscher, Jaideep Ray, and Erin Carolyn Solfiell Acquesta. Model-form error correction using universal differential equations for an agent-based model of infectious disease. Technical report, Sandia National Lab.(SNL-CA), Livermore, CA (United States), 2024.

[16] Arnau Quera-Bofarull, Bryan McWilliams, et al. One-shot sensitivity analyses via automatic differentiation. In *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 1867–1875, 2023. States that sensitivity analyses typically require re-running ABMs many times.

[17] Christopher Rackauckas and Qing Nie. Differentialequations. jl–a performant and feature-rich ecosystem for solving differential equations in julia. *Journal of open research software*, 5(1): 15–15, 2017.

[18] Christopher Rackauckas, Yingbo Ma, Julius Martensen, Collin Warner, Kirill Zubov, Rohit Supekar, Dominic Skinner, Ali Ramadhan, and Alan Edelman. Universal differential equations for scientific machine learning. *arXiv preprint arXiv:2001.04385*, 2020.

[19] Christopher Rackauckas, Yingbo Ma, Julius Martensen, Collin Warner, Kirill Zubov, Rohit Supekar, Dominic Skinner, Ali Ramadhan, and Alan Edelman. Universal differential equations for scientific machine learning. *arXiv preprint arXiv:2001.04385*, 2020.

[20] Connor Robertson, Jaideep Ray, and Cosmin Safta. Bayesian calibration of stochastic agent based model via pca-based surrogate modeling. Technical Report SAND2024-01892C, Sandia National Laboratories, 2024. CityCOVID ∼100 CPU-hrs per 70-day simulation; calibration requires hundreds to thousands of ABM runs.

[21] Jasem Tamimi and Pu Li. Nonlinear model predictive control using multiple shooting combined with collocation on finite elements. *IFAC-PapersOnLine*, 48(8):241–246, 2015.

[22] Utkarsh Utkarsh, Valentin Churavy, Yingbo Ma, Tim Besard, Prakitr Srisuma, Tim Gymnich, Adam R Gerlach, Alan Edelman, George Barbastathis, Richard D Braatz, et al. Automated translation and accelerated solving of differential equations on multiple gpu platforms. *Computer Methods in Applied Mechanics and Engineering*, 419:116591, 2024.

[23] Zhihui Wang, Joseph D Butner, Vittorio Cristini, and Thomas S Deisboeck. Integrated pk-pd and agent-based modeling in oncology. *Journal of pharmacokinetics and pharmacodynamics*, 42(2):179–189, 2015.

[24] Greg Welch, Gary Bishop, et al. An introduction to the kalman filter. 1995.

[25] Xiaojun Wu, MeiLu McDermott, and Adam L MacLean. Data-driven model discovery and model selection for noisy biological systems. *PLOS Computational Biology*, 21(1):e1012762, January 2025. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1012762. URL `https://dx.plos.org/10.1371/journal.pcbi.1012762`.

[26] Nan Ye, Farbod Roosta-Khorasani, and Tiangang Cui. Optimization methods for inverse problems. In *2017 MATRIX annals*, pages 121–140. Springer, 2019.

A Appendix

A.1 Baseline Compartmental ODEs for Epidemiology

We begin from the extended SEIR-type model used in ExaEpi [1]. The population is stratified into seven compartments:

- S: Susceptible individuals who have not yet been infected.
- E: Exposed individuals in the latent period, infected but not yet infectious.
- $I_{\sim s}$: Presymptomatic infectious individuals who will eventually develop symptoms.
- I_s: Symptomatic infectious individuals who may recover or die.
- I_a: Asymptomatic infectious individuals who never develop symptoms but can transmit.
- D: Deceased individuals due to COVID-19.
- R: Recovered individuals with immunity after infection.

The governing ODEs are:

$$\frac{dS}{dt} = -\lambda(t)S, \tag{1}$$

$$\frac{dE}{dt} = \lambda(t)S - \frac{1}{T_E}E, \tag{2}$$

$$\frac{dI_{\sim s}}{dt} = \frac{1-f_a}{T_E}E - \frac{1}{T_{I_{\sim s}}}I_{\sim s}, \tag{3}$$

$$\frac{dI_s}{dt} = \frac{1}{T_{I_{\sim s}}}I_{\sim s} - \frac{1}{T_s}I_s, \tag{4}$$

$$\frac{dI_a}{dt} = \frac{f_a}{T_E}E - \frac{1}{T_{\text{inf}}}I_a, \tag{5}$$

$$\frac{dD}{dt} = \frac{1-f_r}{T_s}I_s, \tag{6}$$

$$\frac{dR}{dt} = \frac{f_r}{T_s}I_s + \frac{1}{T_{\text{inf}}}I_a. \tag{7}$$

Here,

$$\lambda(t) = p_t\,\kappa(t)\,\frac{\eta_a I_a + I_s + \eta_a I_{\sim s}}{N - D},$$

is the force of infection, where p_t is the probability of transmission per contact, $\kappa(t)$ is the time-dependent contact rate (reduced after lockdown), η_a is the relative infectiousness of asymptomatic or presymptomatic cases, and N is the total population.

The dependent time constants are:

$$T_{I_{\sim s}} = T_{\text{inc}} - T_E, \tag{8}$$

$$T_s = T_E + T_{\text{inf}} - T_{\text{inc}}. \tag{9}$$

The contact rate $\kappa(t)$ is modeled as a piecewise function to account for reduced mixing after lockdown:

$$\kappa(t) = \begin{cases} \kappa_1, & t < t_{ld}, \\ \kappa_2, & t \geq t_{ld}, \end{cases}$$

where t_{ld} is the start day of lockdown.

A.2 Universal Differential Equations (Vanilla UDEs)

While ordinary differential equation (ODE) compartmental models capture average population dynamics, they cannot fully represent the heterogeneous, local-level behaviors present in agent-based models (ABMs). Universal Differential Equations (UDEs) provide a hybrid modeling framework that preserves the mechanistic ODE structure while replacing unknown or poorly captured terms with data-driven neural network (NN) approximations [19]. This approach allows UDEs to act as surrogates for ABMs: interpretable at the population scale while flexible enough to capture dynamics that the baseline equations.

Formally, given the baseline system:

$$\frac{d\mathbf{u}}{dt} = f_\theta(\mathbf{u}, t),$$

where $\mathbf{u}(t) \in \mathbb{R}^n$ are the compartmental states and f_θ is the mechanistic epidemiological model, a UDE replaces selected functions or parameters inside f_θ with neural network outputs. For example, the time-dependent contact rate $\kappa(t)$ can be parameterized as

$$\kappa_\phi(\mathbf{u}, t) = \xi_\kappa\,\left|\mathcal{N}_\phi\Big(\tfrac{E}{N_0}, \tfrac{I_{\sim s}}{N_0}, \tfrac{I_s}{N_0}, \tfrac{I_a}{N_0}, \tfrac{D}{N_0}, \tfrac{R}{N_0}\Big)\right|_1, \tag{10}$$

where $\mathcal{N}_\phi$ is a neural network with weights ϕ, normalized inputs are the current states, and ξ_κ is an optional scaling factor. This learned contact rate then enters into the infection force

$$\lambda_\phi(t) = p_t\, \kappa_\phi(\mathbf{u}, t)\ \frac{\eta_a I_{\sim s} + I_s + \eta_a I_a}{S + E + I_{\sim s} + I_s + I_a + R},$$

which replaces the fixed form $\lambda(t)$ in the ODE system.

In more general UDE designs, other parameters such as the recovery fraction f_r or asymptomatic fraction f_a can also be replaced by neural outputs, leading to different architectures (1NN, 2NN, 3NN) depending on how many terms are learned by the network

Loss function. The UDE is trained against observed data $\mathbf{y}(t_k)$ by minimizing a trajectory loss, typically mean-squared error (MSE):

$$\mathcal{L}_{\text{UDE}}(\theta, \phi) = \sum_k \big\| \hat{\mathbf{u}}(t_k; \theta, \phi) - \mathbf{y}(t_k) \big\|^2, \tag{11}$$

where $\hat{\mathbf{u}}(t_k; \theta, \phi)$ denotes the simulated solution at observation times t_k.

Thus, in this "vanilla" UDE setting, neural networks serve as replacements for key epidemiological functions (e.g., $\kappa(t)$), correcting model-form errors while retaining the mechanistic backbone of the compartmental system.

A.3 Multiple Shooting for UDEs (MS-UDE)

To reduce long-horizon error accumulation and improve numerical stability, we train the UDE with a *multiple shooting* [21] objective. Let $0 = t_0 < t_1 < \cdots < t_T$ be observation times and partition the index set $\{0, \ldots, T\}$ into M contiguous windows $\mathcal{W}_i = \{s_i, \ldots, e_i\}$ of fixed length m ("group_size" in code). For each window we solve a short IVP and penalize discontinuities between windows. In our implementation, we assume all compartment states are observed (or reliably reconstructed) at window boundaries so that continuity constraints are well-posed. In general, MS accommodates partial observations by optimizing latent z_i and applying H to the simulated states in the data term.

Dynamics (replacement-style UDE). The mechanistic RHS f_θ uses neural replacements for selected terms (e.g., contact rate):

$$\kappa_\phi(\mathbf{u}, t) = \xi_\kappa\, \big|\mathcal{N}_\phi(\tilde{\mathbf{u}})\big|_1, \qquad \tilde{\mathbf{u}} = \left[\tfrac{E}{N_0}, \tfrac{I_{\sim s}}{N_0}, \tfrac{I_s}{N_0}, \tfrac{I_a}{N_0}, \tfrac{D}{N_0}, \tfrac{R}{N_0}\right], \tag{12}$$

$$\lambda_\phi(t) = p_t\, \kappa_\phi(\mathbf{u}, t)\ \frac{\eta_a I_{\sim s} + I_s + \eta_a I_a}{S + E + I_{\sim s} + I_s + I_a + R}, \tag{13}$$

$$\frac{d\mathbf{u}}{dt} = f_\theta\big(\mathbf{u}, t; \kappa = \kappa_\phi(\mathbf{u}, t)\big). \tag{14}$$

Windowed simulations. For window $\mathcal{W}_i = \{s_i, \ldots, e_i\}$ we introduce a (possibly free) initial state $\mathbf{z}_i \in \mathbb{R}^n$ at time t_{s_i} (in practice, $\mathbf{z}_1 = \mathbf{u}_0$; subsequent $\mathbf{z}_i$ can be optimized or anchored to data). We simulate

$$\hat{\mathbf{u}}_i(t; \mathbf{z}_i, \theta, \phi) \quad \text{for } t \in [t_{s_i}, t_{e_i}] \quad \text{via the UDE IVP.}$$

Data misfit. With observations $\mathbf{y}(t_k)$, the per-window fit is

$$\mathcal{L}_{\text{data}} = \sum_{i=1}^{M} \sum_{k \in \mathcal{W}_i} \big\| \hat{\mathbf{u}}_i(t_k; \mathbf{z}_i, \theta, \phi) - \mathbf{y}(t_k) \big\|_2^2. \tag{15}$$

Continuity penalty. To enforce global consistency we penalize jumps at window boundaries:

$$\mathcal{L}_{\text{cont}} = \lambda_{\text{MS}} \sum_{i=1}^{M-1} \big\|\ \hat{\mathbf{u}}_i\big(t_{e_i}; \mathbf{z}_i, \theta, \phi\big) - \mathbf{z}_{i+1}\ \big\|_2^2, \tag{16}$$

where λ_{MS} is the continuity weight ("`continuity_term`" in code).

MS-UDE objective.

$$\min_{\theta,\phi,\{\mathbf{z}_i\}} \quad \mathcal{L}_{\text{MS}} := \mathcal{L}_{\text{data}} + \mathcal{L}_{\text{cont}}. \tag{17}$$

Remark (Observability). In our implementation, the continuity penalty couples the terminal state of window i to the initial state of window $i+1$. This implicitly assumes full-state observability at window boundaries; with partial observations, one must introduce latent-state decision variables and additional constraints/priors.

Training procedure

1. **Partition:** Build windows with fixed length m (“`group_size`”) using `group_ranges`.
2. **Forward solves:** For each window i, solve the UDE IVP on $[t_{s_i}, t_{e_i}]$ from $\mathbf{z}_i$.
3. **Loss:** Compute $\mathcal{L}_{\text{data}}$ with MSE (function `loss_function`) and add $\lambda_{\text{MS}}\,\mathcal{L}_{\text{cont}}$ (“`continuity_term`”).
4. **Optimize:** ADAM warm start ($\alpha = 10^{-3}$, many iters) → LBFGS refinement, as in the script. Parameters include NN weights ϕ, any learnable ODE parameters θ, and optionally the window initial states $\{\mathbf{z}_i\}$ if treated as decision variables.

A.4 Prediction Error Method for UDEs (PEM-UDE)

The prediction error method (PEM) minimizes a loss based on *short-horizon predictions* rather than full simulation error. By introducing a feedback term that anchors trajectories to observed data, PEM stabilizes optimization, smooths otherwise chaotic loss landscapes, and improves robustness to drift [13, 4]. This makes PEM particularly valuable when training UDEs on unstable or chaotic systems, where direct trajectory fitting is ill-posed.

Data interpolation. Let $\{(t_k, \mathbf{y}_k)\}_{k=0}^{T}$ be observations. We define a continuous, vector-valued interpolant

$$\mathbf{y}(t) \approx \text{LinearInterp}\big(\{t_k, \mathbf{y}_k\}\big),$$

so the predictor can access measurements at arbitrary solver steps. This mirrors prior PEM implementations that interpolate observational data to align with solver integration steps [4].

Predictor (observer-augmented UDE). Following the PEM-UDE formulation, we add an observer correction term to the mechanistic-plus-neural dynamics. In our epidemiological replacement-style UDE, the predictor takes the form

$$\frac{d\mathbf{u}}{dt} = f_\theta\big(\mathbf{u}, t; \kappa = \kappa_\phi(\mathbf{u}, t)\big) \;+\; K(t)\big(\mathbf{y}(t) - \mathbf{u}\big), \tag{18}$$

where f_θ denotes the mechanistic ODE with neural replacements (e.g., learned κ_ϕ), and the feedback term $K(\mathbf{y}(t) - \mathbf{u})$ injects the instantaneous prediction error. In Chesebro et al. [4], K is introduced as a hyperparameter to tune the steepness of the loss landscape; in our implementation we treat K as a constant, $K = \text{diag}(k_1, \ldots, k_n)$ is treated as a set of *trainable parameters* learned jointly with (θ, ϕ).

PEM loss. Given observation times $\{t_k\}$, the PEM objective is an MSE on the predicted trajectory, with a mild Tikhonov prior on the gains:

$$\mathcal{L}_{\text{PEM}}(\theta, \phi, K) = \frac{1}{T+1}\sum_{k=0}^{T} \big\|\hat{\mathbf{u}}(t_k; \theta, \phi, K) - \mathbf{y}(t_k)\big\|_2^2 \;+\; \lambda_K \|K\|_F^2, \tag{19}$$

where $\hat{\mathbf{u}}$ is the solution of (18). The regularization prevents K from becoming excessively large and trivially tracking the interpolated data.

Training procedure.

1. **Build interpolant** $\mathbf{y}(t)$ from the data (piecewise-linear).
2. **Form predictor** (18) by augmenting the UDE with $K(\mathbf{y}(t) - \mathbf{u})$.
3. **Optimize** $\min_{\theta,\phi,K} \mathcal{L}_{\text{PEM}}$ with an ADAM warm start (e.g., $\alpha = 10^{-3}$) followed by LBFGS refinement.

Remark. This formulation is consistent with the general PEM-UDE approach proposed for chaotic systems [4], while our adaptation applies it to epidemiological surrogates of ABMs. The key benefit remains the same: PEM smooths the loss landscape and suppresses long-term divergence, enabling tractable gradient-based training of UDEs.

A.5 Adaptation to Agent-Based Models

Multiple shooting, prediction error methods, and universal differential equations are established techniques in numerical analysis and system identification. Multiple shooting has long been used to stabilize trajectory optimization and parameter estimation by breaking long horizons into shorter subproblems; prediction error methods are classical in control theory and system identification for ensuring well-posed optimization landscapes; and UDEs have been recently formalized in the scientific machine learning literature as a way to combine mechanistic models with neural networks.

Our contribution lies not in proposing these techniques anew, but in how we *adapt* them to the agent-based model (ABM) setting. Specifically, we demonstrate how:

- UDEs [19] can be used as surrogates for exascale ABM simulations, with neural replacements for key parameters (e.g., contact rate $\kappa(t)$) to capture emergent, heterogeneous effects not expressible in closed-form ODEs.
- Multiple shooting [21] can be reformulated for neural-augmented epidemiological models, ensuring stability and robustness in the presence of abrupt interventions (e.g., lockdowns).
- PEM [4] can be extended to this context by interpolating averaged ABM trajectories and learning observer gains K jointly with UDE parameters, anchoring the surrogate to stochastic ABM outputs while preserving differentiability for gradient-based training.

In this way, our work provides a bridge between the rich local dynamics captured by ABMs and the tractable, interpretable structure of compartmental ODEs, by leveraging and adapting existing methodologies from numerical optimization and system identification. This adaptation allows for scalable surrogate modeling of ABMs while maintaining interpretability and fidelity to the original agent-based dynamics.

A.6 Implementation Details

All models are trained using a two-stage procedure: an ADAM optimizer phase for exploration, followed by LBFGS for refinement. The loss function is the mean squared error across the six normalized compartments. Exact formulations of the SEInsIsIaDR model, UDE dynamics, multiple-shooting constraints, and PEM updates are provided in the Appendix. All the training was conducted on an 8-core Intel Broadwell CPU instance (e2–standard–8, 32 GB RAM) using the x86_64 architecture. We use the `DifferentialEquations.jl` [17] ecosystem in the Julia programming language for our code implementation.

A.7 Data and Model Setup

We train and evaluate our models using data generated from ExaEpi, an exascale high-performance epidemiological simulator [1]. The simulator implements stochastic realizations of autonomous interactions between agents and their environments that produce compartmental dynamics, which we average across multiple runs to reduce noise and provide smooth training targets. The dataset includes seven epidemiological compartments: Never (Susceptible, denoted as S), Exposed (E), Presymptomatic ($I_{\sim s}$), Symptomatic (I_s), Asymptomatic (I_a), Deaths (D), and Recovered (R). Each compartment is normalized by the initial population size N_0 to ensure stability during optimization. The UDEs is trained to reproduce the temporal trajectories of these compartments over the specified horizon.