Differentiable agent-based epidemiological modeling for end-to-end learning

Ayush Chopra [*1] **Alexander Rodríguez** [*2] **Jayakumar Subramanian** [3] **Balaji Krishnamurthy** [3]
B Aditya Prakash [2] **Ramesh Raskar** [1]

Abstract

Mechanistic simulators are an indispensable tool for epidemiology to explore the behavior of complex, dynamic infections under varying conditions and navigate uncertain environments. ODE-based models are the dominant paradigm that enable fast simulations and are tractable to gradient-based optimization, but make simplifying assumptions about population homogeneity. Agent-based models (ABMs) are an increasingly popular alternative paradigm that can represent the heterogeneity of contact interactions with granular detail and agency of individual behavior. However, conventional ABM frameworks are not differentiable and present challenges in scalability; due to which it is non-trivial to connect them to auxiliary data sources easily. In this paper we introduce GRADABM which is a new scalable, fast and differentiable design for ABMs. GRADABM runs simulations in few seconds on commodity hardware and enables fast forward and differentiable inverse simulations. This makes it amenable to be merged with deep neural networks and seamlessly integrate heterogeneous data sources to help with calibration, forecasting and policy evaluation. We demonstrate the efficacy of GRADABM via extensive experiments with real COVID-19 and influenza datasets. We are optimistic this work will bring ABM and AI communities closer together.

1. Introduction

Mechanistic simulation models explicitly model mechanisms and causal connections to explore empirically observed variables. They are an indispensable tool for epidemiology to explore the behavior of complex, dynamic infections and guide forecasting and decision making. Here, compartmental ODE-based models are the dominant paradigm which use a system of differential equations to describe disease transmission by modeling the flow rate of individuals from one disease stage (compartment) to another. Classical examples include the 'so-called' S-I-R (susceptible-infected-recovered) model with 3 compartments (32; 24); with some works which further stratify the population via separate compartmental models for each sub-population (eg: Age-SEIR (41), Geo-SEIR (9)). These SIR-like models are simple to design, very fast to execute and amenable to gradient-based optimization of simulation parameters (44; 8). Hence they have also been used in hybrid models (with deep neural networks) for forecasting influenza (25), Ebola (31) and also COVID pandemic (28). This tractability is achieved through simplifying assumptions about homogeneous transmission and perfect mixing (each agent interacts with every other agent) in the population, which may often not be desirable. Specifically, these SIR-like models do not represent the underlying contact graphs which prevents them from recreating heterogeneity of the interacting populations and modeling the adaptability of individual behavior, in the real world (10; 41). Agent-based models (ABMs) have emerged as an alternate paradigm which can alleviate these concerns.

ABMs are discrete simulators which comprise a collection of agents which can act and interact within a computational world (35; 15; 42; 63; 51). They can explicitly represent heterogeneity via the underlying contact networks and also model the adaptability of individual behavior to enable more realistic simulations. However, they are conventionally slow, difficult to scale to large population sizes (43) and tough to calibrate with real-world data (27). This is a challenge since simulation results (emergent behavior) can be highly sensitive to the scale of the input population and calibration of the input parameters. In addition, incorporating novel sources of data that could inform calibration and other downstream tasks (e.g., forecasting) is often laborious and adds overhead complexity to the ABM (e.g., incorporating digital exposure data to ABMs (34)). In this paper, we introduce GRADABM to alleviate these concerns and realize the potential of ABMs for practical decision making in epidemiology.

The key insight of GRADABM is to enable fast forward and

[*]Equal contribution [1]Massachusetts Institute of Technology [2]Georgia Institute of Technology [3]Media and Data Science Research Lab, Adobe. Correspondence to: Ayush Chopra <ayushc@mit.edu>, Alexander Rodríguez <arodriguezc@gatech.edu>.

Proceedings of the 39th International Conference on Machine Learning Workshops, Online, PMLR 119, 2022. Copyright 2022 by the author(s).

Simulator	Input (data)		Process (modeling)			Output (utility)	
	Granularity	Heterogeneity	Fast	Scalable	Differentiable	*What-if* Interventions	Forecasting
Compartmental models [35, 27, 44, 10, 47, 9]	medium	low	yes	no	yes	medium	yes
ABM (conventional) [22, 37, 42, 53]	high	medium	no	yes	no	high	no
Differentiable ABM (GradABM)	**high**	**high**	**yes**	**yes**	**yes**	**high**	**yes**

Figure 1. We introduce an end-to-end differentiable GradABM which can: i) ingest multi-granular (citizen, block, county, state) and heterogenous (clinical, behavioral, policy, census, survey) data sources; ii) simulate realistic populations *quickly and reliably* with gradient-based calibration; iii) facilitate flexible interventions for policy decisions and efficient forecasting with end-to-end ML-ABM pipelines. The key idea of GradABM is a general-purpose differentiable sparse tensor-calculus design which we validate by demonstrating utility for epidemiology here.

differentiable inverse simulations. We realise this through a differentiable sparse tensor-calculus based implementation. GradABM can simulate millions-size networked populations in few seconds on commodity hardware, run on both CPUs and GPUs and is end-to-end differentiable. Conventional ABM frameworks follow an object-oriented (agent-centered) design where the agents are modeled as objects. While conceptually appealing and extensible, this is often inefficient to represent inter-agent infection transmission (over contact networks) and within-agent disease progression as the size of agent populations and interactions scale. Our framework GradABM follows a network-centric design (21) where agents are represented as tensors, their interaction networks as (sparse) adjacency matrices and a continuous relaxation of the stochastic disease transmission model is used to produce gradient estimates with automatic differentiation. We use this scalable, fast and differentiable design of GradABM to merge with deep neural networks (DNNs) and seamlessly integrate heterogeneous data sources to help with calibration, forecasting and policy evaluation. Particularly, we use COVID-19 and Influenza modeling as examples to show the benefit of GradABM for (time-space) personalized and robust forecasting; and evaluating efficacy of pharmaceutical interventions for decision making. We believe that differentiable agent-based epidemiology can open new venues for epidemic understanding and aid practical decision making by learning from millions of agents in few seconds.

Our Contributions: (i) We present GradABM, a differentiable ABM that can simulate million-scale populations in a few seconds on commodity hardware and be merged with DNNs for end-to-end learning. The key idea of GradABM is a general-purpose differentiable sparse tensor-calculus based implementation which we validate for epidemiology here. (ii) We demonstrate the utility of GradABM for robust forecasting and analysis of COVID-19 and Influenza. (iii) We show the use of GradABM in evaluating pharmaceutical interventions for policy decision making.

2. Background and Related Work

First, we briefly mention work in Agent-based Models (ABMs) and with relevance to epidemiology. Then, we summarize relevant work on learning in ABMs by integrating with deep neural networks, methods to calibrate ABM parameters and scalability of simulation parameters. We conclude a brief mention of a useful insight we leverage for our solution (permutation invariance). We refer the reader to (24; 32; 40) for a more extensive discussion. ABMs define systems (35; 15) as collections of agents with can act and interact with each other within a computational world. They have been used to model cells in a tumor micro-environment to diagnose cancers (42), humans in a physical environment to study economic policies (63), infectious diseases (51; 5) and avatars in a digital world to study misinformation (16). **ABMs in Epidemiology** (EpiABM) are used to understand how disease spreads and evaluate efficacy of health interventions for mitigating them. EpiABMs build simulations where agents host and transmit virus within their multiple contact networks (there is a within-host disease model e.g. SIR); and (43; 26) discuss methods and assumptions for designing such simulators. Briefly, population contact networks are assumed to resemble a random graph with a specific degree distribution; and an epidemiological model (34) is used to describe between-host transmission and within-host progression of infection. Recently, such EpiABMs are used to evaluate the benefit of delaying 2nd dose of the vaccine (51), deploying mobile apps for digital contact tracing (5), prioritizing test speed over accuracy (39) to contain the spread of COVID-19. The utility of such simulators for practical decision making depends upon their ability to recreate the population with great detail, integrate with real-world data streams and analyze the sensitivity of results. **Learning** from data in ABMs by integrating with DNNs is being explored across multiple fields. The dominant method is to treat it as a multi-agent reinforcement learning task where the ABM defines the environment and a DNN is used to parameter-

ize agent action policies. Similar ideas has been used by (63) to learn equitable economic policies, (54) to analyze societal segregation dynamics, (45) to learn oil and gas macro strategies. An alternate strategy has focused on using DNNs to emulate agent-based models (11; 7; 46) by using the ABM to generate training dataset for DNNs. Recently, (11) used an epidemiological ABM to generate synthetic datasets and trained DNN models to predict infectiousness for proactive contact tracing. (46) used a similar strategy (of ABM as surrogate) to identify what simulation parameters to calibrate. In contrast, we propose to make the ABM differentiable instead so that it can be *trained end-to-end* with the DNN modules using gradient-based optimization. This makes it possible to integrate heterogeneous data to infer latent micro variables and improve forecasting power of ABMs through hybrid DNN-ABM pipelines. This also has interesting implications for calibrating ABMs, as we discuss next. **Calibration** of ABMs involves identifying appropriate values for micro variables and is essential to ensure reliability of results. Generally, a hybrid strategy is used where some parameters are sourced from control trials conducted offline by clinical experts and then others are calibrated in-silico by achieving a goodness-of-fit between emergent ABM output and real-world macro data (e.g. number of deaths). The conventional technique is to use grid search (18; 51) or beam search (58) by running several forward simulations and finding the best-fit. This is often slow and tough to scale to many parameters. Recent works is using optimization methods for faster and better calibration. (46) uses an ML model to identify what parameters to calibrate but values are identified with random search. (5) used compartmental (SEIR-like) models where parameters are learned by gradient-based optimization in the SEIR-like model and are then used in the ABM. However, SEIR-like models have few tunable parameters which limits the extent and correctness of calibration with real-world data for the ABM. In contrast, the differentiable design of GRADABM allows calibrating parameters with gradient-based optimization inside the ABM itself. This helps learn personalized parameters resulting in better (and robust) forecasting and policy decision making. Such gradient-based calibration is iterative and hence requires fast forward simulations, as is reviewed next. **Scalability** of ABMs is a key consideration since modeling granular population details is computationally expensive. Conventional frameworks like Mesa follow an object-oriented design and while ease to use (Python-API) are prohibitively slow to scale (require few hours for a single iteration). There have been attempts to reduce this burden and simulate realistic scale with distributed HPC systems and GPU-optimized implementations. EpiFast (14) demonstrated simulating large contact networks in a few minutes on distributed systems. However, the required compute for such executions is expensive and non-trivial to access. OpenABM (34) presents an optimized C++/Cuda API to build an epidemiological ABM for fast execution on commodity hardware but is tough to extend and is not amenable to gradient-based learning. Motivated by recent work in molecular dynamics (53), GRADABM is implemented using highly optimized sparse tensor APIs of auto-diff packages (eg: pytorch, JAX) to accelerate and scale simulations while preserving ease of use.

Invariances, Computation and ABMs Efficacy of DNN architectures results from overcoming the curse of dimensionality by leveraging the pre-defined invariances arising from the underlying low-dimensionality and structure of the physical world (17). This is key to efficient computation on grids with CNNs (translation invariance) and graphs with GNNs via neural message passing (permutation invariance). We posit that while useful for learning DNNs, utilizing these invariances can make computation tractable in physical systems, where they exist naturally. Specifically, we observe that epidemiological models (21; 34; 5; 51) also adhere to permutation invariance wherein the order of infectious interactions (pair-wise message passing) within a step does not matter when estimating the probability of infection from those interactions. We utilize this to make GRADABM differentiable where the transmission model can be implemented as a variant of message passing parameterized with physical equations instead of neural networks. More details in appendix.

3. Differentiable Agent-based Modeling for Epidemiology

The pipeline is summarized in Fig. 2, where the inner loop is comprised of the epidemiological model GRADABM and the outer loop of the calibration model CALIBNN. Training is executed over 4 stages: 1) generating epidemiological input parameters using a parameterized neural network (CALIBNN), 2) epidemiological forward simulation with the generated parameters (inner loop), 3) computing a loss/error between output of the epidemiological model and ground truth data, and 4) compute the gradient of this loss using back-propagation through all the previous 3 stages and then update parameters of CALIBNN using gradient descent. In the rest of this section, we first describe the epidemiological model, followed by the calibration and training pipeline.

3.1. Epidemiological Model

Here, we describe the micro loop shown in Fig. 2. We consider a K step discrete-event simulation with a networked population of n stateful agents, where A^t is the underlying contact network at given step t. We build upon a standard epidemiological model used in several clinical papers (21; 34; 51), which is comprised of a Transmission Model and a Progression Model. These two components

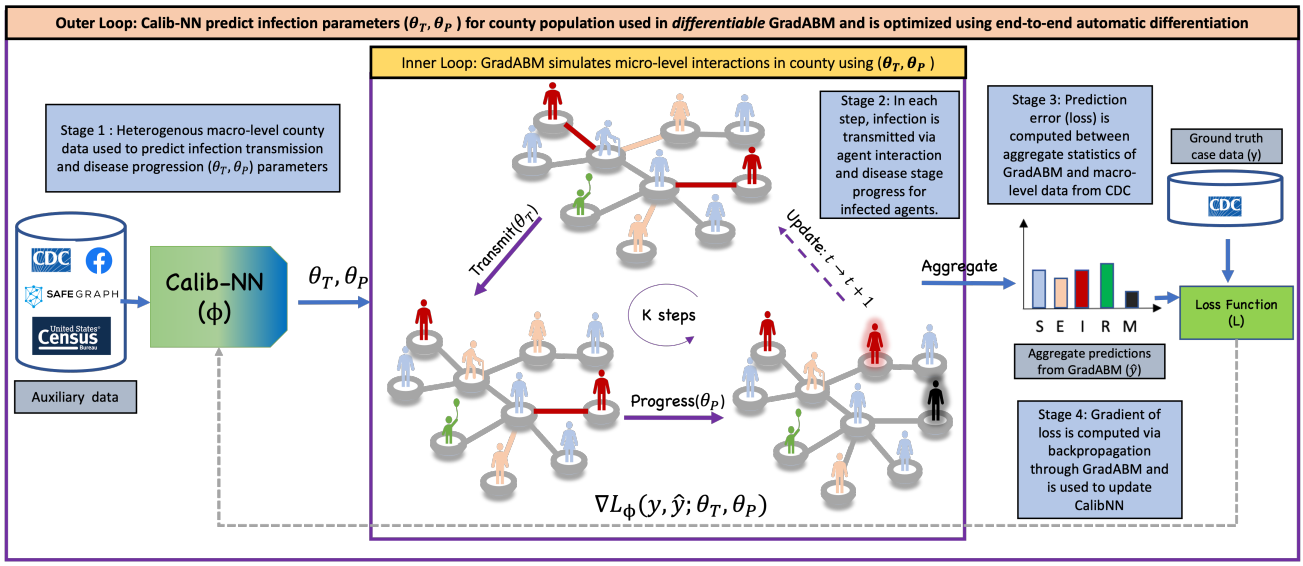

Figure 2. **Differentiable Agent-based Epidemiology** involves 4 stages i) heterogenous macro-level population data (CDC, census, behavioral, survey) is input to a calibration model (CALIBNN) to predict epidemiological parameters (θ_T, θ_P). ii): (θ_T, θ_P) are used run K forward steps of the fully-differentiable epidemiological model (GRADABM) which simulates micro-level infection transmission (`Transmit`) and disease progression (`Progress`) over individual contact networks. Disease statistics are aggregated (`Aggregate`) at end of K steps to obtain the macro-level simulation output ($\hat{y}$). iii) Error between predicted $\hat{y}$ and real-world case statistics (y) is used to define a loss ($L(\hat{y}; \hat{y})$). iv) Gradient of this loss is computed by automatic differentiation through the micro-level GRADABM to update weights of macro-level GRADABM using gradient descent.

recursively update the state of all agents, which is described next.

Agent State is denoted by a 3-D tensor X_j^t for an agent $j \in \{1, \ldots, n\}$ at time step $t \in \{0, \ldots, K\}$. $X_j^t = \{a_j, d_j^t, e_j^t\}$ where $a_j \in \{0-10, 11-20, 21-30, \ldots, 71-80, 80+\}$ is the age of the agent, which can take one of nine discrete values; d_j^t is the current disease stage of agent j and for example in COVID-19, it can take values in $\in \{S, E, I, R, M\}$ where S, E, I, R, M denote susceptible, exposed, infected, recovered and dead (mortality) stages respectively, $e_j^t \in \{-1, \ldots, t-1\}$ denotes the time step of last exposure. In some simulations, we augment the state using two additional controlled dynamic variables, which are: $v_j^t \in \{0, 1, 2\}$ that denotes the current vaccination status and $q_j^t \in \{Y, N\}$ that denotes the current quarantine status.

Transmission Model is a parameterized model that computes the probability of infection transmission as a result of an interaction between susceptible and infected (or exposed) agents. Interaction Networks are the sites for contact between agents that transmit infection. Following (51; 5), we separately model contact graphs within a county and represent agent interactions in multiple scenarios: household, random and work related interactions. The graph is generated using demographic-stratified mobility information obtained from (52), evolves at every time step (different at each step). Consider an interaction at time t, between susceptible and infected agents with states $X_i^t = [a_i, d_i^t, e_i^t]$ $X_j^t = [a_j, d_j^t, e_j^t]$, where $d_i^t = E$, $e_i^t = -1$, $d_j^t = I$ and $d_j^t \in \{0, \ldots t-1\}$ respectively. The rate of transmission at any step t primarily depends upon: (i) infectiousness of the pathogen at time t (R^t), (ii) susceptibility of infectee i to transmission (S_i), and (iii) transmissibility of the infector j (T_j) and (iv) time since exposure for infector ($\Delta E_j^t = t - e_j^t$). The infectiousness varies over time, starting at zero when the agent is infected, peaks at an intermediate time and eventually tends to zero and is modeled using a gamma distribution (5). The susceptibility of the infectee is age-stratified as described in (34). The probability of transmission ($q(.;.)$) from the interaction is represented as: $q(d_i^t, d_j^t) = 1 - e^{-\lambda(R, S_i, T_j, \Delta E_j^t)}$, where,

$$\lambda(R, S_i, T_j, \Delta E_j^t) = \frac{R S_i T_j}{\hat{I}_i} \int_{\Delta E_j^t - 1}^{\Delta E_j^t} G_\Gamma(u; \mu_j, \sigma_j^2) du$$

models the rate of transmission in the interaction, $\hat{I}_i$ is the expected number of interactions for agent i, $q(d_i^t, d_j^t)$ is then aggregated over all interactions for any agent i and then the aggregated value is used to parameterize a Bernoulli distribution and sample a discrete transmission event. A successful exposure then causes an update in X_i^t (d_i^t and e_j^t).

Progression Model Once successfully exposed (as explained in the Transmission Model) or when in an infected state, an agent j enters a hierarchy of disease stage progression which triggers subsequent changes in the state of the agent. We follow a slightly modified SEIRM (61) progression model. Susceptible (S) agents may become exposed (E) as explained in the Transmission model above, and ex-

posed agents progress to infection (I) and eventually end in a stage of recovery or mortality (R, M). The Progression Model is parametrized by stage transition times and mortality rate. The key difference between our model and the SEIRM-like models is that we track states of individual agents rather than just the aggregate numbers of agents in various disease stages. Furthermore, we model the $S \to E$ exposure step, using interaction networks which take into account local agent interactions, which is different from the standard SEIRM-like models. This model can be parameterized to study multiple infectious diseases, and through experiments we validate generality for COVID-19 and influenza, with possible changes in the type and number of disease stages.

Forward Simulation with GRADABM We combine these components of the epidemiological model to simulate the disease dynamics over a horizon of K steps. The input to the simulator are the time-dependent parameters which govern the transmission and progression models, denoted as $\theta_T^t = [R^t, S, T]$, $\theta_P^t = [m, \tau_{EI}, \tau_{IR}, \tau_{IM}]$ respectively, where $t \in \{0, \ldots, K\}$. Here R^t is the time-dependent disease reproduction number (measure of infectiousness), S is the age-stratified susceptibility, T is the age-stratified infection transmissibility, m is the mortality rate (fraction of expired agents amongst all agents in the $\{R, M\}$ disease stage), τ_{EI}, τ_{IR}, and τ_{IM} are the stage transition times. At each step t, every agent i interacts with a set of neighbors $\{j : j \in \mathcal{N}_i\}$ and may accumulate or transmit infection. This disease stage evolution, at any step t, is given by:

$$d_i^{t+1} = \texttt{Update}(X_i^t, \mathcal{N}_i, (X_j^t)_{j\in\mathcal{N}_i}, \theta_T^t, \theta_P^t), \quad (1)$$

where $\texttt{Update}(X_i^t, \mathcal{N}_i, (X_j^t)_{j\in\mathcal{N}_i}, \theta_T^t, \theta_P^t)$ is

$$= \begin{cases} \texttt{Transmit}(X_i^t, \mathcal{N}_i, (X_j^t)_{j\in\mathcal{N}_i}, \theta_T^t), & \text{if } d_i^t = S, \\ \texttt{Progress}(X_i^t, \theta_P^t), & \text{if } d_i^t \in \{E, I\}. \end{cases} \quad (2)$$

The transmission model is **Transmit**$(X_i^t, \mathcal{N}_i, (X_j^t)_{j\in\mathcal{N}_i}, \theta_T)$ where R^t, S_i, T_j are obtained from θ_T and e_j^t from X_t^j is written as:

$$d_i^{t+1} = \hat{Q}(\bigcup_{j\in\mathcal{N}_i} (\lambda(R^t, S_i, T_j, e_j^t))), \quad (3)$$

where, λ denotes the rate of transmission from a single interaction, $\bigcup$ is an aggregation function which accumulates transmission over multiple interactions and $\hat{Q} = \text{Bernoulli}(1 - e^{q(t)})$ maps this transmission to a probability of infection. Following the invariance of infection transmission noted in section 2 (more in appendix), $\bigcup$ is the summation ($\sum$) function in our implementation. The progression model $\texttt{Progress}(X_i^t, \theta_P)$ updates the disease stage from $E \to I$ or from $I \to \{R, M\}$ at times determined by the stage transition time parameters. Finally, the aggregate cumulative deaths is determined as: $\hat{y} = \texttt{Aggregate}(d_i^T) = m * (d_i^T \in \{R, M\})_{i\in\{1,\ldots,n\}}$, where $m \in \theta_P$.

3.2. Calibration and Forecasting

Training In order to use GRADABM for real-world applications, we need to be able to calibrate its parameters (θ_T^t, θ_P^t) in such a way that the aggregate predicted quantities from the model such as cumulative deaths match the observed values from real-world data. This is done using the calibration and training setup (the outer loop in Fig. 2). This is a W step loop, with each step w comprising the following four stages as shown in the figure. **Stage 1**: Generating epidemiological parameters $(\theta_T^t, \theta_P^t)_{t\in\{0,\ldots,K\}}^w$ using CALIBNN (f) with neural network parameters ϕ^w at step w and input as the auxiliary data (D) from sources as described in section 4. For ease of exposition, we drop the $t \in \{0, \ldots, K\}$ subscript and refer to this input as $(\theta_T^t, \theta_P^t)^w$. Then, we have $(\theta_T^t, \theta_P^t)^w = f(D; \phi^w)$ **Stage 2** Performing a K-step simulation using GRADABM (the inner loop in Fig. 2) with $(\theta_T^t, \theta_P^t)^w$ from CALIBNN in the previous stage. **Stage 3**: The output of GRADABM is aggregated to compute cumulative deaths. This macro value is compared with corresponding ground truth values and the following loss is computed: $\mathcal{L}(\hat{y}^w, y; (\theta_T^t, \theta_P^t)^w) = \texttt{MSE}(\hat{y}^w, y)$, where y is the ground truth data representing cumulative deaths, $\hat{y}^w$ is the cumulative deaths predicted by the epidemiological model at the the wth training step with input parameters $(\theta_T^t, \theta_P^t)^w$) and $\texttt{MSE}$ denotes the mean-squared error function. **Stage 4**: Once this loss is computed, we use automatic differentiation to estimate gradient for minimizing the loss. The gradient is computed over the K-steps of the epidemiological model GRADABM (which recursively calls $\texttt{Update}(\cdot;\cdot)$) and calibration model CALIBNN which is parameterized with a neural network. In practice, auto-diff packages such as pytorch implement back-propagation through such a 'for loop' by unrolling this loop in the internal computational graph that the library tracks to enable end-to-end differentiation. The only requirement is that all steps within the loop be differentiable, which we establish by showing that the components of our epidemiological model - Transmission model and Progression model are differentiable.

The Transmission Model is differentiable as the functions comprising it, which are: λ is a smooth function, $\bigcup$ is a permutation invariant function ($\sum$) and is linear. In order to make the sampling from $\hat{Q}$ differentiable, we utilize a continuous relaxation with Gumbel softmax reparametrization (36). Next, the Progression Model is differentiable as it is effectively a linear deterministic model, with the parameters being the transition times. Thus, the entire K step epidemiological model is differentiable.

The final step of this gradient computation involves back-propagation through the parameters of the CALIBNN neural network (ϕ) as well. Once the gradient is thus determined, the parameters of CALIBNN are then updated as given below:

$$\phi^{w+1} = \phi^{w} - \alpha \frac{\partial \mathcal{L}(\hat{y}^{w}, y; (\theta_T^t, \theta_P^t)^w)}{\partial \phi}, \tag{4}$$

where α is the learning rate, $(\theta_T^t, \theta_P^t)^w = f(D; \phi^w)$ and $\hat{y}^w$ is computed by calling `Update` K times with same parameters $(\theta_T^t, \theta_P^t)^w$. This training is continued till an acceptable level of aggregate prediction error is achieved. This process of learning via gradient descent requires multiple runs of the K-step epidemiological model. Hence, for this training to be feasible, our epidemiological model should be fast and also scalable to enable real-world relevant simulations. Thus, we note that the two key properties of GRADABM which make such iterative gradient-based optimization tractable are: i) fast simulations at realistic scale - with sparse tensor calculus; ii) differentiable simulations for gradient-based optimization.

Inference Once trained, the hybrid pipeline with CALIBNN and GRADABM can be used to forecast in unseen scenarios as well as aid in policy decision making. These unseen scenarios may include predicting future the evolution of multiple infections for a county/region, or it may involve making predictions for a new county/region for which historical data for calibration is not available, or it may involve making predictions even when the data (inputs to CALIBNN) for a county/region is noisy. In each of these scenarios, the data available for the county/region is fed to CALIBNN, which then outputs time-space personalized parameters to be used in the GRADABM. GRADABM is then run for K steps using these parameters and the simulation results are used to make the desired predictions.

4. Experimental Setup

We conduct experiments for COVID-19 and influenza on multiple counties of the state of Massachusetts, USA, and we learn personalized parameters for each county. To predict county-specific parameters, we design CALIBNN (Sec. 3.2) which ingests diverse data sources and is optimized using gradients from the prediction error of GRADABM. Next, we describe the data sources used to build the simulator and training pipeline. • **Heterogeneous Data Sources** GRADABM allow bringing together data across multiple granularities (individual, census-block, county, city, country) and modalities (mobility, policy, clinical, digital surveys) due to flexibility of agent-based simulators. We collected important publicly available[1] datasets from a variety of trusted sources. Here we introduce our data sources in the order of our training pipeline in Fig. 2. i) We use county-level time-varying features that are ingested by CALIBNN. Specifically, for COVID-19 we use 5 signals including insurance claims data, online symptoms surveys from Facebook, and line-list data. For flu, we use the 14 signals from the Google symptom dataset. Flu data is reported at state-level, so we utilize the same data for all counties. ii) GRADABM leverages individual-level data to set agents' states created based on demographic information from the US Census (1). iii) Our interaction networks are created using census block level mobility data from Safegraph. Specifically, we use MCMC sampling to generate synthetic contact graphs inside each census-block using demographic information (e.g., age). iv) To set some of the parameters of our transmission and progression models, we utilize clinical data from reliable sources such as CDC (2) and clinical papers (34; 29). The target variable for COVID-19 is COVID-associated mortality, while in flu we use influenza-like-illness (ILI) counts, which is collected by the CDC. The ground truth data for the target variables is obtained from JHU and CDC. Additional details are in the appendix. • **Baselines** We compare GRADABM against multiple popular approaches for calibrating an ABM - **Vanilla-ABM:** Following (34; 51), it uses parameters set with expert and/or authoritative information with randomized search. For COVID-19, we get R_0 and case-fatality rate from (13; 3) and R_0 for flu from (22). **PC-ABM:** Following (19), it uses R_0 and initial infections learned using a compartmental model (SEIRM and SIRS for COVID-19 and flu, respectively). We also perform studies to compare different configurations of GRADABM. GRADABM (w/o TL, w/o CALIBNN): Removes CALIBNN and its data inputs, and instead uses scalar learnable parameters. GRADABM (w/o TL): trains with only data from one county and one ABM; instead we use transfer learning in GRADABM to learn a single CALIBNN that can predict for all counties. • **Metrics** For forecasting error metrics, we adopt standard metrics used by previous work for evaluating epidemic predictions (57; 6). Specifically, we use normal deviation (ND), root mean squared error (RMSE) and mean absolute error (MAE) which are detailed in our appendix. Following CDC forecasting guidelines (12; 23), we make weekly predictions for 1 to 4 weeks ahead in the future. Our evaluation for both diseases is of at least 4 months in 10 counties. Specific counties and evaluation periods are in the appendix. • **Implementation Details** *Data preparation:* We process the time series data into padded sequences which are normalized per region. *Architecture details*: CALIBNN is an encoder-decoder neural architecture based on GRUs (20) and self-attention (59). *ABM parameters* (θ_P, θ_T): Our time-varying ABM parameters have the

[1]Data links: delphi.cmu.edu; goo.gle/covid19symptomdataset; safegraph.com; coronavirus.jhu.edu; gis.cdc.gov/grasp/fluview/fluportaldashboard.html

flexibility to be changed on each time step of the simulator (daily), but we found this to be challenging for learning. Thus, we opted to use the same set of ABM parameters for every 7 days (i.e., we learn a new set of parameters for every week). Details can be found in the appendix. Our code and data are publicly available[2].

5. Results

First, we show that GRADABM integrates with CalibNN to improve ABM forecasting performance on multiple infectious diseases. *Next*, we show that GRADABM is robust to noisy data, can seamlessly scale to simulate large populations and be used to evaluate policy interventions for decision making.

5.1. Forecasting multiple infectious diseases

Results presented in Table 1 shows that GRADABM consistently outperforms typical ABM calibration methods when forecasting infection spread of COVID-19 and Influenza, across all metrics (rows 1-3 in Table 1)[3]. For instance, GRADABM improves ND to 0.97 from 8.75 in Vanilla-ABM and 2.21 in PC-ABM on COVID-19; and to 0.41 from 0.57 in Vanilla-ABM and 0.59 in PC-ABM on Influenza. Specifically, we observe that GRADABM achieves a consistent gain of 8x to 12x over Vanilla-ABM on all metrics in COVID-19. This performance benefit is attributed to three key features of GRADABM which allow us to learn personalized time-varying parameters: gradient-based calibration, integration of heterogeneous data sources via CALIBNN, and transfer learning from joint county training. We also conduct ablation studies to analyse these features (rows 3-5 in Table 1) and note some observations. *First*, using CALIBNN is significant to performance gain as it allows: i) integration of heterogeneous data sources for learning parameters and ii) rich and flexible optimization by coupling the ABM with a neural network. *Second*, training counties independently result in marginal fall in performance (e.g.: ND goes from 0.97 to 1.26 in COVID) when compared to joint training which utilizes transfer learning. This indicates that GRADABM can automatically take advantage of what is being learned in one county can be useful for another county, but, in contrast to other methods that directly copy the learnt values, our method personalizes appropriately. Finally, we note that transfer learning does not help in Influenza like in the COVID-19 case. As noted in Sec. 4, this results from shared state-level influenza data for all counties and hence CALIBNN cannot personalize parameters across counties.

[2] https://github.com/AdityaLab/GradABM

[3] We performed the unpaired t-test test ($\alpha = 0.05$) over 5 runs to verify GRADABM performance gains w.r.t other ABM calibration methods are statistically significant

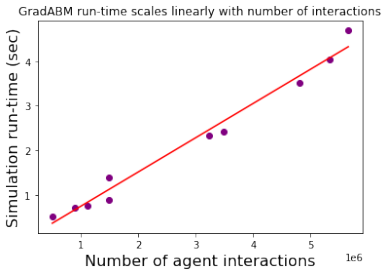

Figure 3. GRADABM run-time scales linearly with the number of interactions and is roughly 300x faster than prior-art. This benefit is due to the sparse-tensor calculus based design.

5.2. Robustness, Scalability and Decision Making

Here, we study the effectiveness of deploying GRADABM to real-world scenarios by focusing on three key considerations: i) scalability to a large real-world population, ii) robustness to noisy data, iii) utility for practical decision making by evaluating policy interventions.

GRADABM runs fast forward simulations on large populations Figure 3 shows that the run-time for GRADABM scales linearly with number of interactions (and agents) in the population and executes very quickly (even as the adjacency matrix scales quadratically). For instance, GRADABM executes a simulation with 800,000 agents (5 million interactions) over 133 steps in 4 seconds on a GPU (and 60 seconds on CPU). This is roughly 300x faster than the equivalent Mesa implementation. This performance efficiency can be attributed to the design of a sparse tensor-calculus based implementation of GRADABM. We note that the fast execution of forward simulations provides the following benefits for practical use: i) makes it possible to conduct sensitivity analyses of results through multiple runs with different hyperparameters, ii) essential for integrating with DNN pipelines for parameter calibration with requires a simulation runs in each optimization step.

GRADABM is robust to noisy data Figure 4 shows that GRADABM outperforms baselines even when trained with noise data. Data quality issues are very common in public health data (38; 49) and we evaluate performance of GRADABM in such scenarios. To study this, we introduce Gaussian noise to the ground truth target with $\mu_{\text{noise}} = 0$ and several scales of standard deviation s_{noise}. For each county, we first obtain the standard deviation of the target and multiply it by the factor λ to get the s_{noise}. We utilize 4 different values of λ. Specifically, we observe that even with large amount of noise ($\lambda = 4$) our error on all metrics is lower than the other baselines that were trained without noise. This is attributed to the differentiable design of GRADABM which allows integrating with CALIBNN such that: i) heterogeneous data sources are used to learn

Table 1. Forecasting results for COVID-19 and influenza over 5 runs. GRADABM is the only one consistently among the best performing for all (lower error metrics is better). TL - Transfer Learning

	COVID-19			Influenza		
Model	ND	RMSE	MAE	ND	RMSE	MAE
Vanilla-ABM (51)	8.75	689.92	270.13	0.57	2.03	1.72
PC-ABM (5)	2.21 ± 1.36	121.87 ± 63.97	68.20 ± 41.84	0.59 ± 0.02	2.17 ± 0.05	1.77 ± 0.05
GRADABM	$\mathbf{0.97 \pm 0.18}$	$\mathbf{50.99 \pm 12.12}$	$\mathbf{30.02 \pm 5.60}$	$\mathbf{0.41 \pm 0.02}$	$\mathbf{1.47 \pm 0.06}$	$\mathbf{1.22 \pm 0.06}$
GRADABM (w/o TL)	1.26 ± 0.43	78.22 ± 78.22	38.74 ± 13.35	0.41 ± 0.02	1.47 ± 0.06	1.22 ± 0.06
GRADABM (w/o TL, w/o CALIBNN)	2.39 ± 0.35	205.14 ± 42.56	73.66 ± 10.88	0.88 ± 0.14	2.97 ± 0.44	2.64 ± 0.43

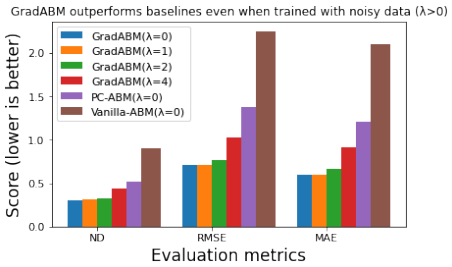

Figure 4. GRADABM is robust to noisy data. GRADABM achieves lower forecasting error than all baselines even when it is trained with noisy data ($\lambda > 0$) while the baselines receive original data. This is achieved due to differentiable design which allows encoding simulation parameters with CALIBNN and learning using heterogeneous data sources.

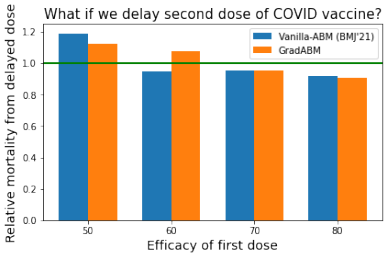

Figure 5. GRADABM helps in evaluating policy interventions. We analyse the decision of delaying the second dose of COVID-19 vaccine and observe that it is sensitive to both efficacy of first dose and calibration mechanism. GRADABM learns personalized calibration parameters which provide better forecasts and also helps in better policy decision making. See sec 5.2 to interpret this figure.

simulation parameters, ii) simulation parameters are not represented as scalar variables but the outputs of a neural network.

GradABM helps in evaluating policy interventions: *What if* we delay the second dose of COVID-19 vaccine? Here, we investigate the utility of GradABM for policy decision making and focus on the question of delaying the second dose. We follow the clinical setup as in (51) and study sensitivity of the decision to protection of first dose ($50\%, 60\%, 70\%, 80\%$) and calibration mechanism (Vanilla-ABM vs GRADABM). We compare cumulative deaths for Franklin County in MA under the two prioritization schedules — S1: Prioritized second dose, S2: Delayed second dose. The sensitivity of this prioritization decision is shown in Fig. 5, where the Y-axis shows the ratio of cumulative deaths under strategy S1 to S2. Whenever this ratio is below 1 (referenced by the green line in the figure), S1 is recommended over S2 and vice-versa. It is evident from our experiment that at vaccine first dose protection of 60%, GRADABM and Vanilla-ABM provide different recommendations, where the main difference is the calibration approach. Since, GRADABM learns personalized calibration parameters which provide better forecasts, it also helps in better policy decision making. More details in the appendix.

6. Discussion

We introduce GRADABM, a differentiable ABM that can simulate million-scale populations in a few seconds on commodity hardware and be merged with DNNs for end-to-end learning. The key idea of GRADABM is a general-purpose differentiable sparse tensor-calculus based implementation which we validate for epidemiology here. We demonstrate the utility of GRADABM to learn personalized time-varying parameters using heterogeneous data sources to better inform forecasting and policy decision making. Future work could explore other benefits like the incorporation of multiple hierarchies of data (also known as macro-micro predictions (62)). In our experiments, we found that even with sparse optimization, GRADABM utilizes a high amount of CPU/GPU memory (especially in the backward computation) which could be an issue for simulations of 6+ months. Addressing this limitation could be an interesting future direction. Our ABM used a linear deterministic model for the disease progression. Future work could explore how to incorporate more complex and stochastic disease progression models (e.g., (51)).

References

[1] U.S. Census Bureau - data U.S. Census, 2021.

[2] CDC – Key facts about influenza (flu), 2022. Accessed: 04.28.2022.

[3] E. Abdollahi, D. Champredon, J. M. Langley, A. P. Galvani, and S. M. Moghadas. Temporal estimates of case-fatality rate for covid-19 outbreaks in canada and the united states. *Cmaj*, 192(25):E666–E670, 2020.

[4] M. Abueg, R. Hinch, N. Wu, L. Liu, W. Probert, A. Wu, P. Eastham, Y. Shafi, M. Rosencrantz, M. Dikovsky, Z. Cheng, A. Nurtay, L. Abeler-Dörner, D. Bonsall, M. V. McConnell, S. O'Banion, and C. Fraser. Modeling the effect of exposure notification and non-pharmaceutical interventions on COVID-19 transmission in Washington state. *npj Digital Medicine*, 4(1):49, Dec. 2021.

[5] M. Abueg, R. Hinch, N. Wu, L. Liu, W. Probert, A. Wu, P. Eastham, Y. Shafi, M. Rosencrantz, M. Dikovsky, et al. Modeling the effect of exposure notification and non-pharmaceutical interventions on covid-19 transmission in washington state. *NPJ digital medicine*, 4(1):1–10, 2021.

[6] B. Adhikari, X. Xu, N. Ramakrishnan, and B. A. Prakash. EpiDeep: Exploiting Embeddings for Epidemic Forecasting. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining - KDD '19*, pages 577–586, Anchorage, AK, USA, 2019. ACM Press.

[7] C. Angione, E. Silverman, and E. Yaneske. Using machine learning as a surrogate model for agent-based simulations. *Plos one*, 17(2):e0263150, 2022.

[8] S. Arik, C.-L. Li, J. Yoon, R. Sinha, A. Epshteyn, L. Le, V. Menon, S. Singh, L. Zhang, M. Nikoltchev, et al. Interpretable sequence learning for covid-19 forecasting. *Advances in Neural Information Processing Systems*, 33:18807–18818, 2020.

[9] D. Balcan, V. Colizza, B. Gonçalves, H. Hu, J. J. Ramasco, and A. Vespignani. Multiscale mobility networks and the spatial spreading of infectious diseases. *Proceedings of the National Academy of Sciences*, 106(51):21484–21489, 2009.

[10] F. Ball, T. Britton, T. House, V. Isham, D. Mollison, L. Pellis, and G. S. Tomba. Seven challenges for metapopulation models of epidemics, including households models. *Epidemics*, 10:63–67, 2015.

[11] Y. Bengio, P. Gupta, T. Maharaj, N. Rahaman, M. Weiss, T. Deleu, E. Muller, M. Qu, V. Schmidt, P.-L. St-Charles, et al. Predicting infectiousness for proactive contact tracing. *arXiv preprint arXiv:2010.12536*, 2020.

[12] M. Biggerstaff, D. Alper, M. Dredze, S. Fox, I. C.-H. Fung, K. S. Hickmann, B. Lewis, R. Rosenfeld, J. Shaman, M.-H. Tsou, P. Velardi, A. Vespignani, and L. Finelli. Results from the centers for disease control and prevention's predict the 2013–2014 Influenza Season Challenge. *BMC Infectious Diseases*, 16(1):357, Dec. 2016.

[13] M. A. Billah, M. M. Miah, and M. N. Khan. Reproductive number of coronavirus: A systematic review and meta-analysis based on global level evidence. *PloS one*, 15(11):e0242128, 2020.

[14] K. R. Bisset, J. Chen, X. Feng, V. A. Kumar, and M. V. Marathe. Epifast: a fast algorithm for large scale realistic epidemic simulations on distributed memory systems. In *Proceedings of the 23rd international conference on Supercomputing*, pages 430–439, 2009.

[15] E. Bonabeau. Agent-based modeling: Methods and techniques for simulating human systems. *Proceedings of the national academy of sciences*, 99(suppl 3):7280–7287, 2002.

[16] J. Brainard, P. Hunter, and I. R. Hall. An agent-based model about the effects of fake news on a norovirus outbreak. *Revue d'epidemiologie et de sante publique*, 68(2):99–107, 2020.

[17] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst. Geometric deep learning: going beyond Euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42, July 2017. arXiv: 1611.08097.

[18] S. Chang, M. L. Wilson, B. Lewis, Z. Mehrab, K. K. Dudakiya, E. Pierson, P. W. Koh, J. Gerardin, B. Redbird, D. Grusky, et al. Supporting covid-19 policy response with large-scale mobility-based modeling. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 2632–2642, 2021.

[19] S. Chang, M. L. Wilson, B. Lewis, Z. Mehrab, K. K. Dudakiya, E. Pierson, P. W. Koh, J. Gerardin, B. Redbird, D. Grusky, et al. Supporting covid-19 policy response with large-scale mobility-based modeling. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 2632–2642, 2021.

[20] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, 2014.

[21] A. Chopra, R. Raskar, J. Subramanian, B. Krishnamurthy, E. S. Gel, S. Romero-Brufau, K. S. Pasupathy, and T. C. Kingsley. Deepabm: scalable and efficient agent-based simulations via geometric learning frameworks-a case study for covid-19 spread and interventions. In *2021 Winter Simulation Conference (WSC)*, pages 1–12. IEEE, 2021.

[22] G. Chowell, M. Miller, and C. Viboud. Seasonal influenza in the united states, france, and australia: transmission and prospects for control. *Epidemiology & Infection*, 136(6):852–864, 2008.

[23] E. Y. Cramer, E. L. Ray, V. K. Lopez, J. Bracher, A. Brennen, A. J. Castro Rivadeneira, A. Gerding, T. Gneiting, K. H. House, Y. Huang, et al. Evaluation of individual and ensemble probabilistic forecasts of covid-19 mortality in the united states. *Proceedings of the National Academy of Sciences*, 119(15):e2113561119, 2022.

[24] N. B. Dimitrov and L. A. Meyers. Mathematical approaches to infectious disease prediction and control. In *Risk and optimization in an uncertain world*, pages 1–25. INFORMS, 2010.

[25] X. Du, A. A. King, R. J. Woods, and M. Pascual. Evolution-informed forecasting of seasonal influenza a (h3n2). *Science translational medicine*, 9(413), 2017.

[26] K. Eames, S. Bansal, S. Frost, and S. Riley. Six challenges in measuring contact networks for use in modelling. *Epidemics*, 10:72–77, 2015.

[27] W. Edeling, H. Arabnejad, R. C. Sinclair, D. Suleimenova, K. Gopalakrishnan, B. Bosak, D. Groen, I. Mahmood, Q. Hashmi, D. Crommelin, et al. Model uncertainty and decision making: Predicting the impact of covid-19 using the covidsim epidemiological code. 2020.

[28] N. M. Ferguson, D. Laydon, G. Nedjati-Gilani, N. Imai, K. Ainslie, M. Baguelin, S. Bhatia, A. Boonyasiri, Z. Cucunubá, G. Cuomo-Dannenburg, et al. Impact of non-pharmaceutical interventions (npis) to reduce covid-19 mortality and healthcare demand. imperial college covid-19 response team. *Imperial College COVID-19 Response Team*, page 20, 2020.

[29] L. Ferretti, C. Wymant, M. Kendall, L. Zhao, A. Nurtay, L. Abeler-Dörner, M. Parker, D. Bonsall, and C. Fraser. Quantifying sars-cov-2 transmission suggests epidemic control with digital contact tracing. *Science*, 368(6491):eabb6936, 2020.

[30] C. for Disease Control and Prevention. U.s. influenza surveillance system: Purpose and methods., 2020.

[31] R. H. Gaffey and C. Viboud. Application of the cdc ebolaresponse modeling tool to disease predictions. *Epidemics*, 22:22–28, 2018.

[32] H. W. Hethcote. The mathematics of infectious diseases. *SIAM review*, 42(4):599–653, 2000.

[33] R. Hinch, W. J. Probert, A. Nurtay, M. Kendall, C. Wymant, M. Hall, K. Lythgoe, A. Bulas Cruz, L. Zhao, A. Stewart, et al. Openabm-covid19—an agent-based model for non-pharmaceutical interventions against covid-19 including contact tracing. *PLoS computational biology*, 17(7):e1009146, 2021.

[34] R. Hinch, W. J. M. Probert, A. Nurtay, M. Kendall, C. Wymant, M. Hall, K. Lythgoe, A. Bulas Cruz, L. Zhao, A. Stewart, L. Ferretti, D. Montero, J. Warren, N. Mather, M. Abueg, N. Wu, O. Legat, K. Bentley, T. Mead, K. Van-Vuuren, D. Feldner-Busztin, T. Ristori, A. Finkelstein, D. G. Bonsall, L. Abeler-Dörner, and C. Fraser. OpenABM-Covid19—An agent-based model for non-pharmaceutical interventions against COVID-19 including contact tracing. *PLOS Computational Biology*, 17(7):e1009146, July 2021.

[35] J. H. Holland and J. H. Miller. Artificial adaptive agents in economic theory. *The American economic review*, 81(2):365–370, 1991.

[36] E. Jang, S. Gu, and B. Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

[37] H. Kamarthi, L. Kong, A. Rodríguez, C. Zhang, and B. A. Prakash. CAMul: Calibrated and Accurate Multi-view Time-Series Forecasting. *arXiv:2109.07438 [cs, stat]*, Sept. 2021. arXiv: 2109.07438.

[38] H. Kamarthi, A. Rodríguez, and B. A. Prakash. Back2future: Leveraging backfill dynamics for improving real-time predictions in future. *International Conference on Learning Representations*, 2022.

[39] D. B. Larremore, B. Wilder, E. Lester, S. Shehata, J. M. Burke, J. A. Hay, M. Tambe, M. J. Mina, and R. Parker. Test sensitivity is secondary to frequency and turnaround time for covid-19 screening. *Science advances*, 7(1):eabd5393, 2021.

[40] A. Lavin, H. Zenil, B. Paige, D. Krakauer, J. Gottschlich, T. Mattson, A. Anandkumar, S. Choudry, K. Rocki, A. G. Baydin, et al. Simulation intelligence: Towards a new generation of scientific methods. *arXiv preprint arXiv:2112.03235*, 2021.

[41] J. Mossong, N. Hens, M. Jit, P. Beutels, K. Auranen, R. Mikolajczyk, M. Massari, S. Salmaso, G. S. Tomba, J. Wallinga, et al. Social contacts and mixing patterns relevant to the spread of infectious diseases. *PLoS medicine*, 5(3):e74, 2008.

[42] K.-A. Norton, C. Gong, S. Jamalian, and A. S. Popel. Multi-scale agent-based and hybrid modeling of the tumor immune microenvironment. *Processes*, 7(1):37, 2019.

[43] L. Pellis, F. Ball, S. Bansal, K. Eames, T. House, V. Isham, and P. Trapman. Eight challenges for network epidemic models. *Epidemics*, 10:58–62, 2015.

[44] Z. Qian, A. M. Alaa, and M. van der Schaar. When and how to lift the lockdown? global covid-19 scenario analysis and policy assessment using compartmental gaussian processes. *Advances in Neural Information Processing Systems*, 33, 2020.

[45] D. Radovic, L. Kruitwagen, C. Schroeder de Witt, B. Caldecott, S. Tomlinson, and M. Workman. Revealing robust oil and gas company macro-strategies using deep multi-agent reinforcement learning. *Available at SSRN 3933996*.

[46] T. Reiker, M. Golumbeanu, A. Shattock, L. Burgert, T. A. Smith, S. Filippi, E. Cameron, and M. A. Penny. Machine learning approaches to calibrate individual-based infectious disease models. *medRxiv*, 2021.

[47] A. Rodríguez, J. Cui, N. Ramakrishnan, B. Adhikari, and B. A. Prakash. EINNs: Epidemiologically-Informed Neural Networks. *arXiv:2202.10446 [physics, q-bio, stat]*, Feb. 2022. arXiv: 2202.10446.

[48] A. Rodríguez, N. Muralidhar, B. Adhikari, A. Tabassum, N. Ramakrishnan, and B. A. Prakash. Steering a Historical Disease Forecasting Model Under a Pandemic: Case of Flu and COVID-19. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 4855–4863, 2021. Issue: 6.

[49] A. Rodríguez, A. Tabassum, J. Cui, J. Xie, J. Ho, P. Agarwal, B. Adhikari, and B. A. Prakash. DeepCOVID: An Operational Deep Learning-driven Framework for Explainable Real-time COVID-19 Forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 15393–15400, 2021. Issue: 17.

[50] S. Romero-Brufau, A. Chopra, A. J. Ryu, E. Gel, R. Raskar, W. Kremers, K. S. Anderson, J. Subramanian, B. Krishnamurthy, A. Singh, et al. Public health impact of delaying second dose of bnt162b2 or mrna-1273 covid-19 vaccine: simulation agent based modeling study. *bmj*, 373, 2021.

[51] S. Romero-Brufau, A. Chopra, A. J. Ryu, E. Gel, R. Raskar, W. Kremers, K. S. Anderson, J. Subramanian, B. Krishnamurthy, A. Singh, K. Pasupathy, Y. Dong, J. C. O'Horo, W. R. Wilson, O. Mitchell, and T. C. Kingsley. Public health impact of delaying second dose of BNT162b2 or mRNA-1273 covid-19 vaccine: simulation agent based modeling study. *BMJ*, page n1087, May 2021.

[52] SafeGraph. Reopening the economy: Foot traffic patterns across the u.s, 2021.

[53] S. Schoenholz and E. D. Cubuk. Jax md: a framework for differentiable physics. *Advances in Neural Information Processing Systems*, 33:11428–11441, 2020.

[54] E. Sert, Y. Bar-Yam, and A. J. Morales. Segregation dynamics with reinforcement learning and agent based modeling. *Scientific reports*, 10(1):1–12, 2020.

[55] J. Shaman, A. Karspeck, W. Yang, J. Tamerius, and M. Lipsitch. Real-time influenza forecasts during the 2012–2013 season. *NATURE COMMUNICATIONS*, page 10, 2013.

[56] J. Shaman, V. E. Pitzer, C. Viboud, B. T. Grenfell, and M. Lipsitch. Absolute humidity and the seasonal onset of influenza in the continental united states. *PLoS biology*, 8(2):e1000316, 2010.

[57] F. S. Tabataba, P. Chakraborty, N. Ramakrishnan, S. Venkatramanan, J. Chen, B. Lewis, and M. Marathe. A framework for evaluating epidemic forecasts. *BMC Infectious Diseases*, 17(1):345, Dec. 2017.

[58] F. S. Tabataba, B. Lewis, M. Hosseinipour, F. S. Tabataba, S. Venkatramanan, J. Chen, D. Higdon, and M. Marathe. Epidemic forecasting framework combining agent-based models and smart beam particle filtering. In *2017 IEEE international conference on data mining (ICDM)*, pages 1099–1104. IEEE, 2017.

[59] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, \. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[60] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[61] J. T. Wu, K. Leung, and G. M. Leung. Nowcasting and forecasting the potential domestic and international spread of the 2019-ncov outbreak originating in wuhan, china: a modelling study. *The Lancet*, 395(10225):689–697, 2020.

[62] Y. Yang, N. Kiyavash, L. Song, and N. He. The Devil is in the Detail: A Framework for Macroscopic Prediction via Microscopic Models. *Advances in Neural Information Processing Systems*, 33, 2020.

[63] S. Zheng, A. Trott, S. Srinivasa, D. C. Parkes, and R. Socher. The ai economist: Taxation policy design via two-level deep multiagent reinforcement learning. *Science Advances*, 8(18):eabk2607, 2022.

A. Extended related work

Permutation invariance: key insights and assumptions

We begin by highlighting some key assumptions that are followed by epidemiologists and then leverage them to motivate the specific implementation of GRADABM. The key insight of GRADABM is to utilize differentiable sparse-tensor computation in interaction networks and symmetries of the transmission model for fast and end-to-end differentiable simulations. Furthermore, we note that: i) clinical data shows that the natural infection takes multiple days to incubate, ii) CDC policy decisions are guided by 4-week ahead forecasting and analysis. These considerations motivate the following standard assumptions in epidemiological simulation work: **Assumptions**: Several works in epidemiological ABMs (21; 34; 4; 51) ratify the following assumptions: **A1**: the granularity of each step of the simulator is one day, **A2**: infectious interactions "do not collide" i.e. on any given day, an agent can not also infect other agents the day it is exposed, **A3**: an agent cannot be reinfected while already infected, **A4**: an agent cannot be infected when it is in the RM, i.e. recovered or mortality stage, i.e., there are no reinfections (this assumption can be relaxed based on the nature of the disease being modelled) **A5**: cumulative transmission over multiple interactions can cause infection, **A6**: infection does not accumulate over days. It is important to note that real-world interaction networks routinely have low degree and high centrality and are modeled as small-world graphs as in (5; 50). This allows use of sparse tensors to represent and process interaction networks; and scale to realistic populations (larger n) which improves performance when outputs are scale dependent. Also, Assumptions **A2-A5** together imply that the order of interactions within a step does not matter. This enables us to invoke permutation invariance in interactions and makes the transmission model differentiable.

B. CALIBNN details

For CALIBNN, we encode the feature time series until t_N by passing it through a Gated Recurrent Unit (GRU) (20) to obtain a condensed representation for each time step:

$$\{\mathbf{h}^t\}_{t=t_0}^{t_N} = \text{GRU}(\{\mathbf{x}^t\}_{t=t_0}^{t_N}) \tag{5}$$

where $\mathbf{h}_t$ is the hidden state of the GRU for time step t. To capture long-term relations and prevent over-emphasis on last terms of sequence we use self-attention layer:

$$\{\lambda^t\}_{t=t_0}^{t_N} = \text{Self-Atten}(\{\mathbf{h}^t\}_{t=t_0}^{t_N}), \tag{6}$$

where Self-Atten (60) involves passing the embeddings into linear layers to extract meaningful similarities before normalizing the similarities using Softmax. Then, we use the attention weights to combine the latent representations and obtain a single embedding representing the time series of features from t_0 to t_N: $\mathbf{u}^{t_0:t_N} = \sum_{t=t_0}^{t_N} \lambda_t \mathbf{h}_t$,

$$\mathbf{U} = \text{Softmax}\Big(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}}\Big)\mathbf{V} \qquad \mathbf{u}_{t_0:t_N} = \sum_{t=t_0}^{t_N} \mathbf{u}^t \tag{7}$$

therefore, our vector $\mathbf{u}^{t_0:t_N}$ summarizes the input sequence and represents the context to be given to the decoder. Next, we use another GRU as a decoder, which takes the context $\mathbf{u}^{t_0:t_N}$ and a positional encoding τ_k that informs the model of how many days in the future we want to predict. We simply use τ_k to be a float between 0 and 1.

$$\{\mathbf{o}^t\}_{t=t_1}^{t_K} = \text{GRU}(\{\tau^k\}_{k=1}^{K}; \mathbf{u}^{t_0:t_N}) \tag{8}$$

We want to unroll our decoder to predict not only the values of the simulator, but also the future ones. For this, we pass the decoder output through a feedforward network FFN($\mathbf{o}^t$). We found the optimization can be challenging if we directly use the output of the neural network as parameters of our simulator. Therefore, we bound the output in a similar manner as (8): $\theta^t = \theta_L + (\theta_U - \theta_L) \cdot \sigma(\text{FFN}(\mathbf{o}^t))$, where θ_L and θ_U are the lower and upper bounds of θ^t for all t, and σ is the Sigmoid function.

C. More details on experimental setup

Code and data are publicly available[4]. Please refer to the README in code repository for more details.

Computational setup. All experiments were conducted using a 4 Xeon E7-4850 CPU with 512GB of 1066 Mhz main memory and 4 GPUs Tesla V100 DGXS 32GB. Our method implemented in PyTorch trains on GPU in 15 mins. Inference is takes only a few seconds.

Real-time forecasting. We follow the literature on evaluating epidemic forecasting methodologies (55; 37; 6) and use the *real-time forecasting* setup. We simulate real-time forecasting by making models train *only* using data available until each of the prediction weeks and make predictions for 1 to 4 weeks ahead in the future. Data revisions in public health data are large and may affect evaluation and conclusions (38; 23), therefore, we utilize fully revised data following previous papers on methodological advances (6; 48).

Evaluation. As evaluation is on weekly predictions, we aggregate daily predictions to obtain weekly predictions (sum for COVID-19 and average for flu). As stated in the main paper, we opted to use the same set of ABM parameters for every 7 days (i.e., we learn a new set of parameters for every week). This is a reasonable assumption as it has been often found that the parameters of mechanistic models do not change much from one day to the other (47). In

[4] https://github.com/AdityaLab/GradABM

our evaluation, we work with the following counties in Massachusetts: 25001, 25003, 25005, 25009, 25011, 25013, 25015, 25021, 25023, 25027. The specific evaluation period is determined with epidemic weeks[5] which is the standard in CDC's epidemic prediction initiatives[6]. For COVID-19 these are 202014, 202016, 202018, 202020, 202022, 202024, 202026, 202028, 202030. For flu, we evaluate in epidemic weeks 201746, 201748, 201750, 201752, 201802, 201804, 201806, 201808, 201810. As noted before, this means that for each epidemic week, we make 4 predictions in the future.

Metrics. Let $\hat{y}_{w,\tau}$ be the prediction week w for τ-weeks ahead in the future, $y_{w,\tau}$ be the corresponding ground truth value, and $e_{w,\tau} = \hat{y}_{w,\tau} - y_{w,k}$. In this paper τ takes from 1 to $T = 4$. Then, we define our metrics as follows: ND $= \sum |e_{w,\tau}| / \sum |y_{w,\tau}|$, RMSE $= \sum e^2_{w,\tau}/WT$ and MAE $= \sum |e_{w,\tau}|/WT$.

Implementation details. • *Data preparation*: All our time series data that enter to CALIBNN are padded with a minimum sequence length of 20 for COVID and 5 for influenza. We also normalize each features with mean 0 and variance 1. To inform CALIBNN of the county to predict, we use a one hot encoding for counties. • *Architecture details*: In CALIBNN, the encoder is a 2-layer bidirectional GRU and decoder is a 1-layer bidirectional GRU, both with hidden size of 32. Output layer has two linear layers of size 32x16xD with ReLU activation function and D is ABM parameter dimensions dimensions: $D = 3$ for COVID-19 and $D = 2$ for flu. • *Hyperparameters*: We found a learning rate of 10^{-3}, Xavier initialization, and the Adam optimization algorithm work best. • *ABM parameters* (θ_P, θ_T): For COVID-19, we have three parameters: R_0, mortality rate, and initial infections percentage. These are bounded with $\theta_L = [1.0, 0.001, 0.01]$ and $\theta_U = [8.0, 0.02, 1.0]$. For flu, we have two parameters: R_0 and initial infections percentage. These are bounded with $\theta_L = [1.05, 0.1]$ and $\theta_U = [2.6, 5.0]$. The initial infections percentage is the percentage of the population that is infected at time step $t = 0$ of the simulation. • *ABM clinical parameters*: To set some of the parameters of our transmission and progression models, we utilize clinical data from reliable sources. Specifically, for COVID-19 we use age-stratified susceptibility and parameters of a scaled gamma distribution to represent infectiousness as a function of time as per (33; 29). For influenza, we set those parameters based on CDC flu facts (2).

Target variables. The target variable for COVID-19 is COVID-associated mortality, while in flu we use influenza-like-illness (ILI) counts, which is collected by the CDC. ILI measures the percentage of healthcare seekers who exhibit influenza-like-illness symptoms, defined as "fever (temperature of 100°F/37.8°C or greater) and a cough and/or a sore throat without a known cause other than influenza" (30). The ground truth data for the target variables is obtained from JHU CSSE COVID-19 data repository and ILI from CDC influenza dashboard.

Details on baseline implementation. Vanilla-ABM: As noted in the main paper, R_0 and case-fatality rate are obtained from authoritative sources. To set the initial infections percentage for this baseline, we set it to the mean value of the search range. **PC-ABM:** We present details on each of the ODE models we used for this baseline.

(COVID-19) SEIRM (61): The SEIRM model consists of five compartments: Susceptible (S), Exposed (E), Infected (I), Recovered (R), and Mortality (M). It is parameterized by four variables $\Omega = \{\beta, \alpha, \gamma, \mu\}$, where β is the infectivity rate, $1/\alpha$ is the mean latent period for the disease, $1/\gamma$ is the mean infectious period, and μ is the mortality rate. The basic reproductive number $R_0 = \beta/(\gamma + \mu)$.

$$\frac{dS_t}{dt} = -\beta_t \frac{S_t I_t}{N} \qquad \frac{dE}{dt} = \beta_t \frac{S_t I_t}{N} - \alpha_t E_t \qquad (9)$$
$$\frac{dI_t}{dt} = \alpha_t E_t - \gamma_t I_t - \mu_t I_t \qquad \frac{dR_t}{dt} = \gamma_t I_t \qquad \frac{dM_t}{dt} = \mu_t I_t$$

(Flu) SIRS (56): This model consists of three compartments: Susceptible (S_t), Infected (I_t), and Recovered (R_t). It is parameterized by three variables $\Omega = \{\beta, D, L\}$, where β is the infectivity rate, D is the mean duration of immunity, and L is the mean duration of the immunity period. The basic reproductive number $R_0 = \beta D$.

$$\frac{dS_t}{dt} = \frac{N - S_t - I_t}{L_t} - \frac{\beta_t I_t S_t}{N} \qquad (10)$$
$$\frac{dI_t}{dt} = \frac{\beta_t I_t S_t}{N} - \frac{I_t}{D_t}$$

D. More details on evaluating policy interventions

We reproduce the experimental setup in (51). For interventions, we simulate standard covid-19 vaccination versus delayed second dose vaccination prioritizing the first dose. Sensitivity analyses included first dose vaccine efficacy of 50%, 60%, 70%, 80%, and 90% after day 12 post-vaccination with a vaccination rate of 0.3% population per day; assuming the vaccine prevents only symptoms but not asymptomatic spread (that is, non-sterilizing vaccine). We measure cumulative covid-19 mortality, cumulative SARS-CoV-2 infections, and cumulative hospital admissions due to covid-19 over 74 days. We explicitly modeled the confirmation of infections with polymerase chain reaction testing and

[5] https://ndc.services.cdc.gov/wp-content/uploads/MMWR_Week_overview.pdf

[6] https://predict.cdc.gov/

Name of Strategy	Description	Priority list under the strategy will be
Standard dosing (two doses on schedule)	Age prioritized vaccination with second dose at 21 days	Betty, David, Frank, Adam, Charlie, Eleanor
Delayed second dose	Age prioritized vaccination prioritizing vaccination of first-dose eligible individuals	Adam, Charlie, Eleanor, Betty, David, Frank

Table 2. Visualization of different vaccination prioritization strategies we compare using GRADABM

quarantining of known infected agents with imperfect compliance over time. To simulate a natural pattern of infection at the point vaccinations begin, we started our simulation with 10 agents infected and ran the simulation for 20 days before starting vaccinations, which corresponds to a cumulative infection rate of 1%, similar to the one in the US, UK, and most of Europe when vaccinations were started and corresponding to the time horizon used in our analysis. In both our vaccination strategies, we started administering vaccines on the basis of age, starting with people over 75, then those over 65, and so on.

We compare different vaccine regimens and prioritization schedules. Consider the following six hypothetical individuals with their age and dose eligibility on any given step of the simulations: i) Adam - first dose eligible 78 yr old; ii) Betty - second dose eligible 78 yr old; iii) Charlie - first dose eligible 68 year old; iv) David - second dose eligible 68 year old; v) Eleanor - first dose eligible 40 year old; vi) Frank - second dose eligible 40 year old. Table 2 shows the order in which these individuals are prioritized for vaccine administration under the strategies we compare.