CSE 8803 EPI: Data Science for Epidemiology, Fall 2023

Lecturer: B. Aditya Prakash
Scribe: Afra Nawar, Aravind Vengarai

Month Sept 19, 2023
Lecture # : 8

1 Lecture 8: Inference I – Summary

It is often the case with real world data that the whole network will not be known a priori. There can be many missing aspects of networks such as missing nodes/edges, missing states, and unknown parameters. Because of this network inference is a nontrivial task. Different methods have been proposed towards this goal including contact tracing, deep learning models, modeling through infection cascades, and statistical models. Tools statistics such as maximum likelihood estimation have been used in conjunction with these models to understand more about patterns in the data that the models are fit to. The parameters of these networks are oftentimes also unknown and must be inferred. To this end, many calibration techniques have been proposed. In this lecture, we learn about calibration of an agent based model used in the GT WiFi mobility study in which calibration was performed using the Nelder-Mead method and was tested for sensitivity. Other methods of calibration include grid search, ODE-based methods, beam search particle filtering, model initialization using digital sources of data, physics-informed neural networks, other end-to-end learning methods.

2 If Data is Plenty

There is often plenty of data, and some questions cannot be answered just by collecting more data, such as on social media—web cascades. However, one can infer the underlying propagation network from a set of observed cascades, such as using Machine Learning methods or incorporating more general surveillance information.

3 Network Inference

Network Inference Problems involve the challenging task of uncovering hidden or partially known aspects of networks. In these scenarios, the nodes and edges of a network are not entirely observable, making the reconstruction of graph models a complex endeavor [**?**]. Maximum Likelihood Estimation (MLE) formulations have become a valuable tool to tackle such problems. MLE allows us to estimate the unknown parameters of a network model based on the observed data, offering a probabilistic framework for network inference.

One critical aspect of network inference problems is determining 'how much data' is enough for reliable and accurate estimations. This question delves into the trade-off between data volume and the precision of parameter estimates. Balancing the need for more data with the practical constraints of data collection is a fundamental challenge in this field.

3.1 Contact Tracing

Contact tracing is a crucial public health process involving systematically identifying individuals who may have come into contact with an infected person. The primary objective of contact tracing is to interrupt the transmission of infectious diseases by promptly identifying and notifying individuals at risk of infection.

This process relies on the inference of a contact network based on traces or transmission chains. To initiate contact tracing, someone initially needs to report their infection, which is the starting point for identifying potential contacts. Once this information is available, health authorities and professionals work diligently to trace and notify individuals who may have been exposed, helping to contain the spread of the disease and protect public health.

3.2 Other Goals of Contact Tracing

Contact tracing serves dual objectives. At the individual level, it provides swift diagnosis, counseling, and treatment for infected individuals, preventing complications and reinfections when possible. Simultaneously, at the public health level, it plays a crucial role in containing disease transmission by identifying and notifying exposed contacts, ultimately safeguarding community health and well-being.

3.3 Infer AIDS Contact Network

In 1984, a seminal study by Auerbach and colleagues, published in the American Journal of Medicine, was pivotal in advancing our understanding of the AIDS epidemic. This study marked a crucial turning point in our knowledge about disease transmission. Through contact tracing efforts, it provided the first direct evidence that AIDS might be spread through contact with an infectious agent during sexual encounters. This groundbreaking discovery laid the foundation for further research and intervention strategies to combat the AIDS epidemic. Subsequently, in 1988, this initial finding was confirmed, solidifying the understanding that AIDS transmission indeed occurred through sexual contact. These early insights into the AIDS contact network played a vital role in shaping public health policies and education initiatives to reduce the spread of the virus.

3.4 Digital Contact Tracing

Amid the COVID-19 pandemic, contact tracing apps emerged as crucial tools for monitoring and curbing the virus's spread. Two main approaches were used: centralized systems like TraceTogether in Singapore, and decentralized systems like Exposure Notifications, a Google and Apple collaboration. In centralized systems, the government manages Bluetooth handshakes between devices, uploading encounter histories to health authorities, possibly including phone numbers. In contrast, decentralized systems prioritize privacy, exchanging random keys via Bluetooth and periodically cross-checking them with lists of keys linked to positive COVID-19 cases.

The effectiveness of digital contact tracing has shown mixed results. Notably, countries like South Korea and Singapore reported success with their Bluetooth-based contact tracing

apps, which played a significant role in containing the virus's spread. However, the joint efforts of Apple and Google to develop Exposure Notifications faced challenges in widespread adoption. In the United States, each state implemented its own contact tracing approach. For instance, Washington, D.C., had a contact tracing system in place, while states like Georgia did not. This lack of coordination across states highlighted disparities in efforts to combat the pandemic.

The adoption of contact tracing apps also raised questions about privacy, cost, and compliance. Assessing whether the privacy trade-offs and the costs of implementation were justified became a subject of debate. The diversity of approaches and the varying levels of success in contact tracing underscored the complexity and multifaceted nature of managing the COVID-19 pandemic.

3.5 Proactive Contact Tracing for COVID

Proactive contact tracing for COVID-19 has gained traction as a response to the limitations of waiting for testing results, which can take several days. The motivation behind this approach is to take timely action to curb the spread of the virus, even before official test results are available. Instead of relying solely on testing outcomes, proactive contact tracing systems operate by propagating anonymized, non-binary risk 'scores' messages between users. These risk scores are typically computed based on factors such as recent interactions, locations visited, and potential exposure to infected individuals. Importantly, this approach estimates an individual agent's risk or expected infectiousness directly on their own device. By doing so, it empowers users with real-time information to make informed decisions about their interactions and take appropriate precautions, ultimately contributing to more effective and timely disease containment strategies.

3.6 Controlling Epidemics: How Much Tracing is Needed?

In situations where all infectious individuals exhibit symptoms, and every individual has precisely n contacts through which infection can spread, analytical results provide valuable insights into the possibility of eradicating the disease. By examining specific parameter values that lead to zero equilibrium prevalence and initial infection growth, we can determine the extent of contact tracing needed for infection elimination. The proportion of contacts that needs to be successfully traced and treated, i.e. the critical level of tracing can be defined as follows [**?**].

r = rate of trasmission

$$t_c = \frac{rn^2 + 1 - 2rn - 2n + \sqrt{r^2n^4 + 2rn^2 + 1 - 4r^2n^2(n-1)}}{2(n-1)}$$

Figure 1: Critical Tracing Efficiency

Therefore,

$$t_c = \frac{t_c}{1 + t_c}$$

The critical tracing efficiency necessary to controlan SIS-type epidemic in an unstructured network where everyone has an equal number of contact is shown below. It increases with both the transmission probability and the number of contacts.

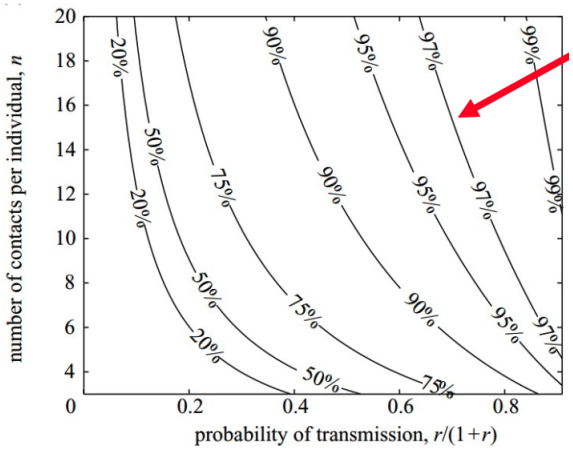


Figure 2: Critical Tracing Efficiency

3.7 Deep Learning Model for Individual Risk

We have previously looked at models, such as ODE systems, which look at population level changes as a whole. However, in real life we cannot assume that populations interact homogeneously. There are many individual differences that may change a person's risk of infection. To determine individual risk for contracting a disease, a deep learning model for proactive contact tracing was developed to ascertain the risk/infectiousness of each individual person. A few factors that contribute to higher risk were used as inputs to the deep learning model. These factors included the day to day health status (yellow) of each individual, their pre-existing conditions (teal), their encounters and graded risk messages (green/red) and day offsets (blue). To account for privacy considerations, contact data was required to be unordered (i.e. there was no information about which contacts occurred before others).

A deep learning choice that the authors made in this model was to implement a neural network for sets using set transformers and deep sets so that distributed inference using deep learning models was possible. The authors also made the choice to use positional encoding. They were able to show that their proactive contact tracing model improves in average mobility over binary contact tracing (which recommends that either all or none of an individuals contacts should quarantine). [**?**]

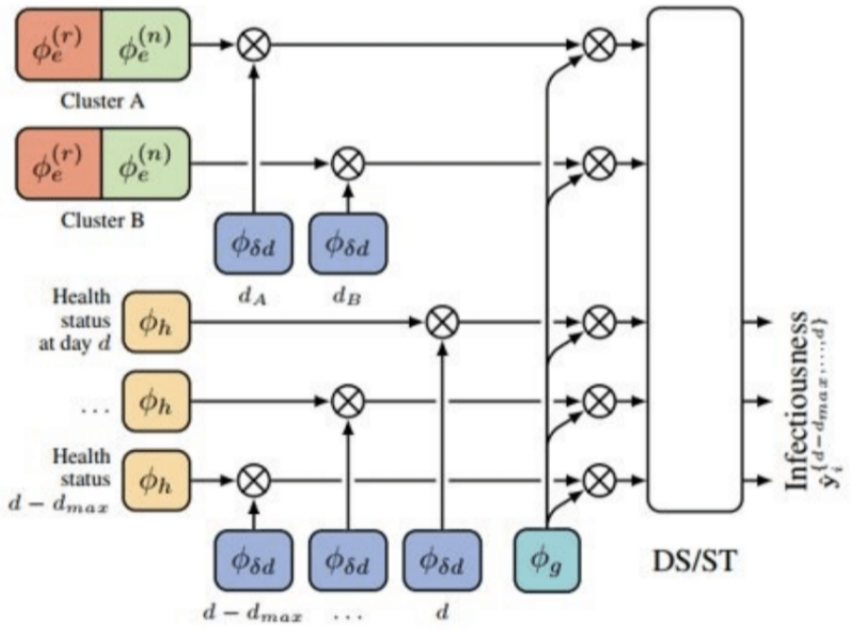


Figure 3: Proactive Contact Tracing Deep Learning Model

3.8 Network Inference via Infection Cascades

Many times, it is difficult to observe how an infection spreads through a network and even what the true underlying network should be. In this case, it is possible to infer properties of the network through observation of data for infected individuals. In the independent cascades model, data is provided which inform about the agent that is infected, type of infection, and time of infection. Using this information, we seek to infer the underlying network.

We can model all of the agents in the data as nodes of the network. Because we do not know the connectivity of the network a priori, we must *infer* the edges of the network. To do so, we use the intuition that if a node, v, gets infected soon after node u then we expect that an edge between the two nodes is present.

As a contagion spreads through the network, it will create a trace called a *cascade*. By aggregating information from multiple contagions, we use similarities in each individual cascade that allow us to infer the hidden network. [**?**]

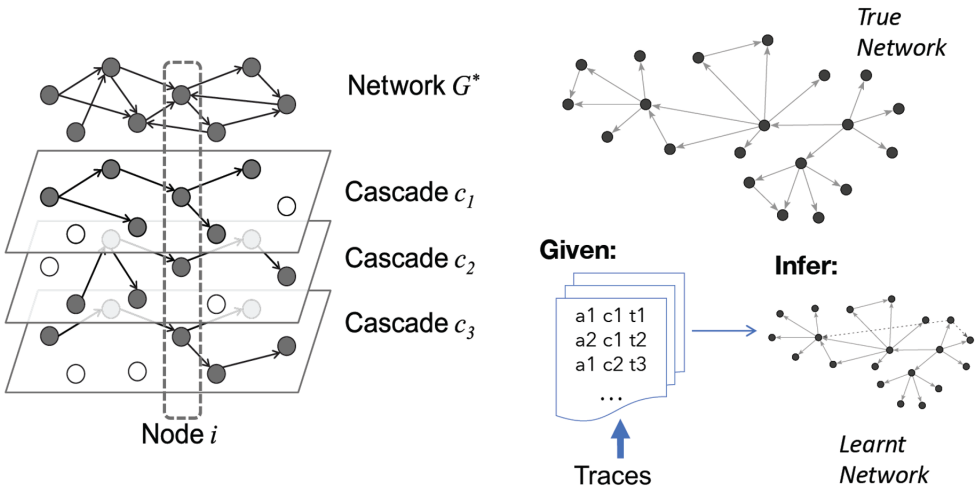


Figure 4: Information Cascades

3.9 Probabalistic Model

To infer the most optimal model, we must build a cascade transmission model $P_c(u, v)$ that gives us the probability that a cascade, c, propagated from node u to v.

If we assume that every node v can only be influenced by one other node u, it is necessarily true that the cascade's influence structure will be that of a directed tree T. We define a couple of probabilities of interest: $P(c|T)$ is the probability that the cascade followed pattern T, $P(c|G)$ is the probability that cascade occurs in graph, G, and probability $P(C|G)$ is the probability that cascades C occur in G.

Using these probability models, we can define the optimal graph, $G* = argmax(P(C|G))$. Just using this equation and iterating through the possible graphs, however, is intractible since there are many different candidates. Thus, we must find efficient algorithms to find the optimal graph. [**?**]

3.10 Extensions

There have been many extensions to the above mentioned work. Some include using dynamic networks instead of static networks. By using a dynamic network, we are able to capture information about changes in contact patterns which may affect the model [**?**]. There have been different cascade models and metapopulation models that have been proposed which branch off of the IC model [**?**, **?**]. Finally innovations have been made to analyze and improve on the performance of the IC model. Some of this work includes theoretical analysis of the amount of samples that are required to infer the edge set. For example, it was concluded that an in an SIR model $O(n\Delta logn)$ traces are enough to infer the edge set [**?**, **?**]. Finally, many studies seek to make improvements by developing more accurate and efficient algorithms for modeling.

3.11 Framework for Statistical Network Inference in Epidemiology

There are multiple different types of data which can be useful in building a epidemiological model. A contact network can help to describe the patterns of connectivity between nodes in a network. Transmission networks can describe the path which a disease has spread from node to node. Some of these networks can be improved with information about friendships and individual characteristics. Data must not be limited to just the nodes involved. Information such as a phylogenetic tree which captures disease mutations can provide directional information about transmission (e.g. someone that had a strain of contagion that evolved later could not have spread the disease to someone who has a strain that evolved earlier).

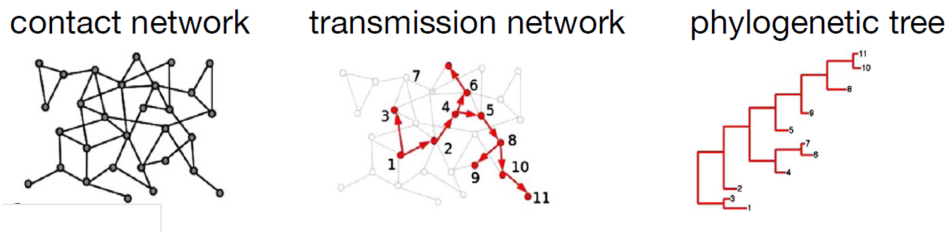


Figure 5: Types of Data

A real world example of a contact network might be a network where students are nodes and edges join students that have classes together. A transmission network could be used to represent Amazon's delivery service where nodes represent physical locations and and the directed edges indicate which location to which location a package travelled. A phylogenetic tree could be made for COVID-19 where branches might show the evolution of different strains from others.

When using these types of data, there are some important questions to keep in mind. We might consider:

- What is the best network model for a particular disease?
- How should parameter values be estimated?
- How precisely the data allows us to estimate params

In real life, we would analyze a problem and first determine what the best network model for a particular disease might be. In the case of COVID, if we are interested in the cascade of the disease from person to person, we might choose to use a transmission network because it contains directional information (unlike a contact network) and information about the patients (unlike a phylogenetic tree). We then have to estimate parameter values such as determining which transmission probabilities best explain the observed data. We must finally figure out how precisely the data allows us to estimate parameters. For example, if we are missing information about people that were infected we might arrive at a less precise model.

Statistical frameworks can help us start to address these questions. Where simulation and probability studies use a model and parameters to learn more about data, statistical inference uses data and a model to understand more about parameters.

3.12 Real World Example of Network Modeling and Inference

To illustrate, we look at the following example. We study a social relationship network of a population of students from grade 7-12 in a particular school. Figure 6 depicts the network below:

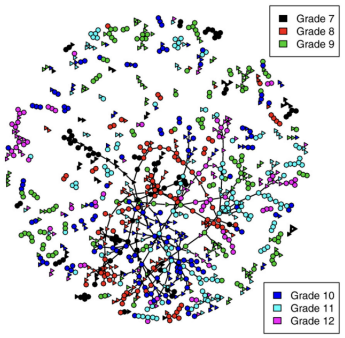


Figure 6: Social Network of High School Students

We want to test families of models that would best explain the creation of such a network and the parameters that fit such a model. We use an exponential family random graph model for this example. In this example, if we consider three parameters to affect the social network graph; grade, race, and sex, the probability, $p_{i,j}$ of an edge between nodes i and j is given by:

$$\begin{aligned} log(p_{ij}/1-p+ij = \beta_1 + \beta_2 * I(i \text{ and } j \text{ are in the same grade}) \\ +\beta_3 * I(i \text{ and } j \text{ are the same race}) \\ +\beta_4 * I(i \text{ and } j \text{ are the same sex}) \end{aligned} \tag{1}$$

where I is an indicator function that produces a 1 if the condition is met and a 0 if not.

The probability that a random network Y equals a fixed network y is given by the following equation:

$$PY = y = \kappa(\beta_1 * E(Y) + \beta_2 * G(Y) + \beta_3 * R(Y) + \beta_4 * S(Y)) \tag{2}$$

where E(Y) denotes the total edges of Y and G(Y), R(Y) and S(Y) denotes the number of edges between nodes of the same grade, race and sex respectively. κ represents a normalizing constant that ensures that the function represents a valid probability distribution.

Using observed values from the network E(y)=974, G(y)=820, R(y)=787, and S(y)=689, we can utilize MLE to solve the above equation and find the parameters that best explain the observed data. The results are shown below in Fig 7.

	Estimate	Std. error	*p*-Value
β_1 (edges)	−10.01277	0.11526	<1e-04
β_2 (nodematch.Grade)	3.23105	0.08788	<1e-04
β_3 (nodematch.Race)	1.19646	0.08147	<1e-04
β_4 (nodematch.Sex)	0.88438	0.07057	<1e-04

Figure 7: MLE Results for Exponential Random Graph Model

From this statistical inference, we can learn something interesting about the parameters. Here we can see that the grade, race, and age are all significant factors that affect the connectivity patterns in the network. The estimate values are also useful to gain more insight into these patterns. For example $exp(3.23) = 25.3$ shows us that individuals are 25.3% more likely to connect with peers in their grade. [**?**]

4 Parameter Inference

4.1 Model Based Reasoning

To infer relationships in the real-world, we can use simulations of models using data and constraints/assumptions. We have an observed dataset but usually have a choice between multiple different potential models. To choose a model, we find which one is closest to the observed data. Usually, a number of models are simulated with different parameters and stored in a database. We can then use the chosen model that is closest to the observed data for calibration.

4.2 Parameter Estimation Using Independent Cascades (IC) Model

In the independent cascades model, we would like to estimate the parameter p_{vu}, the influence probability, for all neighboring nodes. To do so, we define A_u as the number of actions taken by node u and we define A_{v2u} as the number of actions taken by node v that have also been taken by node u at a later time. We can then use maximum likelihood estimation to uncover the p_{vu} that is most probable given the action data. [**?**]

4.3 Maximum Likelihood Estimation (MLE)

Maximum likelihood estimation is a process of finding the model parameters that make an observed dataset most probable. To find these parameters, we maximize a likelihood function that is defined using the chosen model.

For example, see the below three clique network. We have assumed all probabilities of transmission on each of the edges are equal. The infected nodes are given in red and the transmission pattern is noted for three timesteps. We wish to find the parameter R, the proportion of neighbors of an infected node that will become infected in the next time step. Maximum likelihood estimation will find this value to be 0.5 because it makes the pattern in the data most likely (half of an infected nodes neighbors are infected in the next timestep).

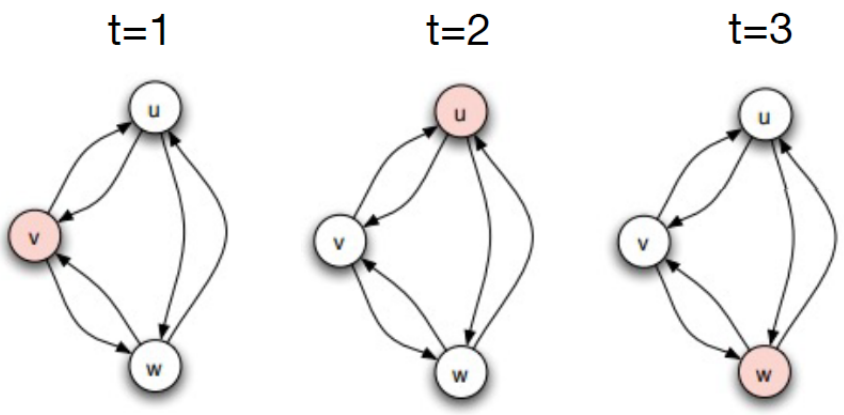


Figure 8: 3 Clique Network

4.4 Maximum Likelihood Estimation in IC Model

We now look at maximum likelihood estimation in the context of the information cascades (IC) model. We define L to be the *likelihood* of observing some data given the chosen model. Recall that the parameter that we desire to optimize is p_{vu}, the influence probability of node v on node u. This means that there is a p_{vu} chance that any action that node v takes node u will take. Because data for the IC model is in the form of a cascade, we can thus model the likelihood according to a binomial random variable:

$$L = \binom{A_v}{A_{v2u}} p_{vu}^{A_{v2u}} (1 - p_{vu})^{A_v - A_{v2u}} \tag{3}$$

Out of A_v actions, there are exactly A_{v2u} successes. The probablility of $A_v 2u$ successes is $p_{vu}^{A_{v2u}}$. Additionally, if there are $A_v 2u$ successes, there must be $A_v - A_{v2u}$ failures. The probability of this occurring can be given by $(1 - p_{vu})^{A_v - A_{v2u}}$. We must then account for these successes and failures occurring in any order by multiplying with $\binom{A_v}{A_{v2u}}$.

We can thus find the influence probabilities that best describe the data by maximizing the likelihood function

$$*p_{vu} = argmax(L) \tag{4}$$

In practice, it is not easy to maximize this function. We therefore look to minimize the log likelihood function instead.

$$*p_{vu} = argmax(L) = argmin(LL) \tag{5}$$

The log likelihood equation is as follows:

$$LL = A_{vu} log(p_{vu}) + (A_v - A_{v2u}) log(1 - p_{vu}) \tag{6}$$

To minimize this equation, we set the derivative of the LL funtion to zero and solve:

$$\delta LL/\delta p_{vu} = A_{v2u} * 1/p_{vu} + (A_v - A_{v2u}) * -1/(1 - p_{vu}) \tag{7}$$

$$p*_{vu} = A_{v2u}/A_v \tag{8}$$

Thus, the optimal influence probabilities are given by taking the fraction of actions taken by node v that have been taken by node u (successes) to the total actions taken by node u (tries). [**?**]

4.5 Generalizing the MLE Solution to SIR Model

Maximum likelihood estimation can be used to estimate the optimal parameters in an SIR model as well. We use the following example to illustrate this process. We are given a graph that is a chain with nodes A, B, and C. We observe that at the second time step, B and C are infected. There are two ways in which this could have occurred. Either B was infected at the first time point and infected C but didn't infect A, or C was infected at the first time step and infected B.

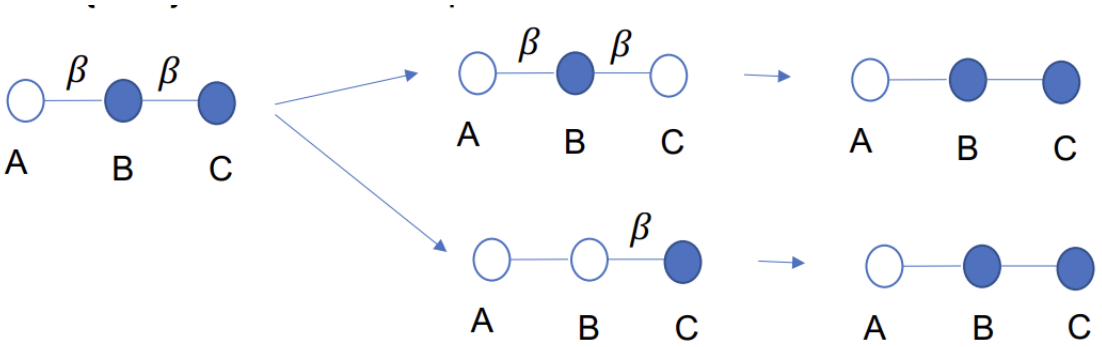


Figure 9: Example Contagion Spread through a Chain

Thus we can write the likelihood of seeing this data as the sum of the likelihoods of either path:

$$L = \beta * (1 - \beta) + \beta \tag{9}$$

We can then maximize this function to find the optimal value of β, $\beta*$ that makes the observation most probable.

In general, for the SI model, $\beta*$ can be given by finding the β that maximizes the probability of seeing a particular set I of infected nodes in graph G given by the observed data:

$$\beta* = argmax(Pr(Set\ of\ I\ that\ is\ infected\ in\ G|\beta)) \tag{10}$$

4.6 Real World Example of Calibration

The GT WiFi mobility project is a real life example of modeling where calibration was needed. The data used to build the model came from more than one source. In this project, the network was built using mobility information collected from localized wifi connection points around campus. This data was combined with time series data from GT about positive test rates and time series data about infection rates in Fulton county. [**?**]

4.7 Agent Based Model (ABM) and Parameters

For this problem, the model that was used was a agent-based model with SEIR disease progression. The SEIR model differs from the SIR model because it includes an additional *Exposed* state. Transitions are made from the *Susceptible* state to the *Exposed* state rather than directly to the *Infected* state. Exposed individuals also have their own probability of transition into the *Infected* state. [**?**]

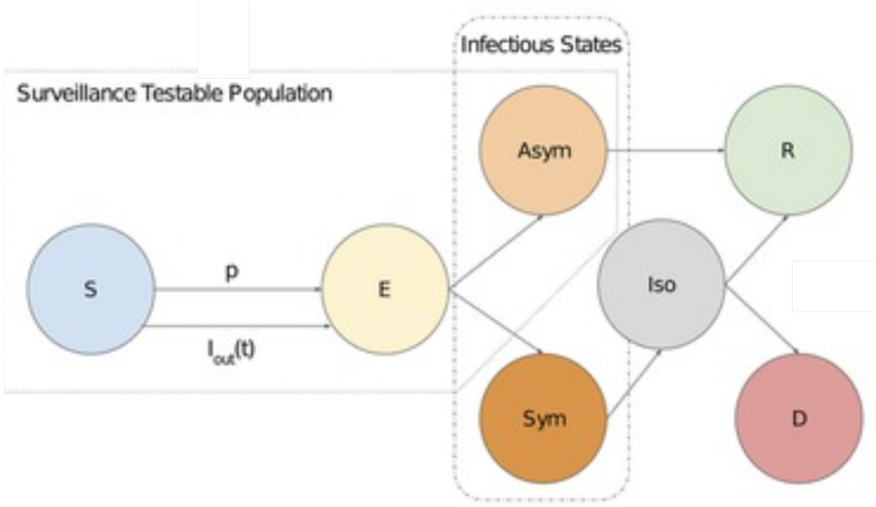


Figure 10: SEIR Network Construction

The mobility data was used to predict how people would travel and how the disease would spread as part of the agent-based model. This data was used to build the underlying contact network.

To build the model, the parameters that were estimated included the proportion of the population that were asymptomatically infected initially (I_o), the transmission probability (p), and a scaling factor which determined the probability of transmission due to contact outside of the WiMob network, α.

In this problem maximum likelihood estimation couldn't be used since the likelihood is intractable. Because of this other techniques are needed.

4.8 Setting Up the Calibration Problem

The first step to setting up the calibration problem is to specify an objective function. We use the equation:

$$f(I_o, \alpha, p) = \sqrt{(\frac{1}{W}[\sum_{w=1}^{W}(\frac{\sum_{i=1}^{N} S(I_o, \alpha, p)}{N} - R_w)^2])}$$

where S is the new asymptomatic cases in a week, w, out of those who could be tested and R is the aggregated result from surveillance testing. An assumption that we make is that each population that gets tested is tested at the same rate.

We can then view the optimization problem as:

$$argmin(f(I_o, \alpha, p))$$

The model was then trained and validated. To do so, the model was fit to data collected from the first 5 weeks of Fall 2020 and the results were validated on the remaining weeks.

4.9 Calibration

Calibration was performed using the Nelder Mead method to minimize the objective function. This is a gradient free method so gradient descent is not used. Instead, 40 different sets of parameters that lie within 40% of the minimum root mean square error are sampled and their means and standard deviations are determined.

Calibration was tested for sensitivity by studying different calibration time frames (e.g. using weeks 0-4 or weeks 5-9):

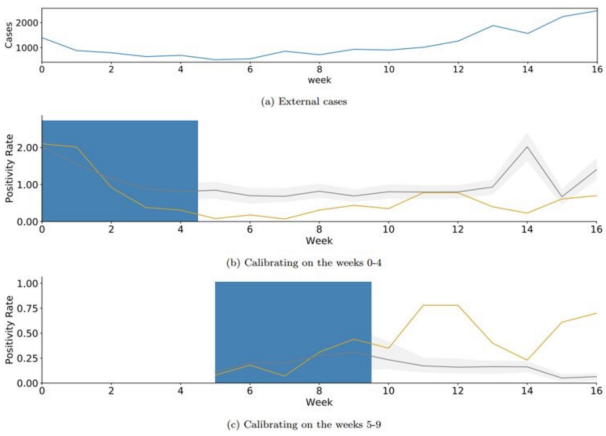


Figure 11: Calibration using Different Weeks

Other sensitivity tests included testing on other datasets which were similar (such as different campuses and counties):

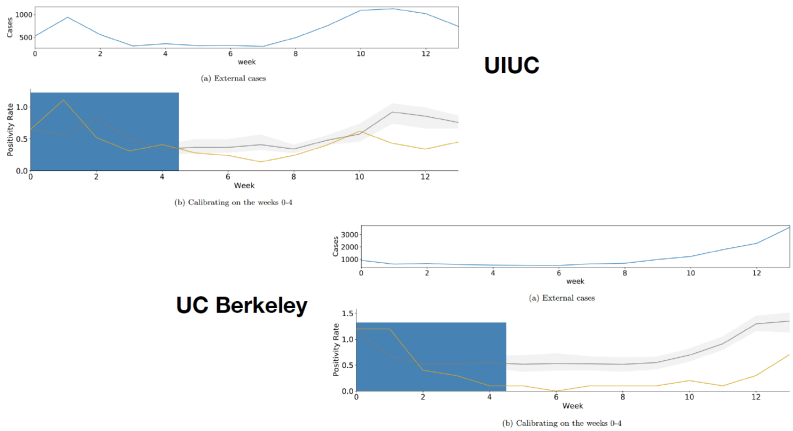


Figure 12: Calibration Using other Countries and Counties

However, calibrating an agent based model is still very challenging and remains an open question. Some of the methods to do so are discussed in following sections.

4.10 Grid Search

To use grid search to calibrate, the first step is to determine the ranges for each parameter in the parameter search. Samples are then taken evenly throughout the range. Certain points will show the best goodness of fit and are chosen as the final calibration values.

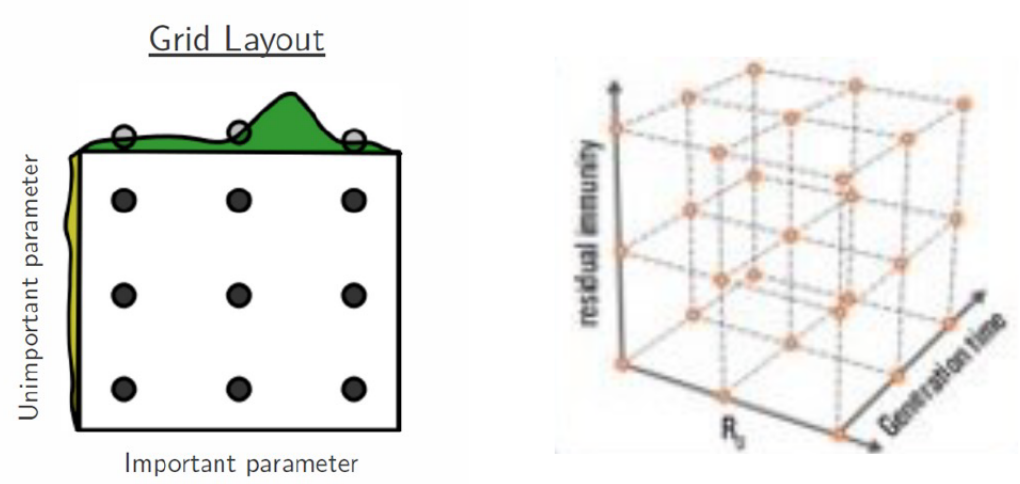


Figure 13: Grid Search Calibration

4.11 ODE-Based

To use an ODE-Based model to calibrate, the problem is first set up using simple ordinary differential equations or a compartmental model. The time series data is then observed and the ODE model is calibrated. After this, the calibrated parameters can be transferred over to the agent-based model [**?**].

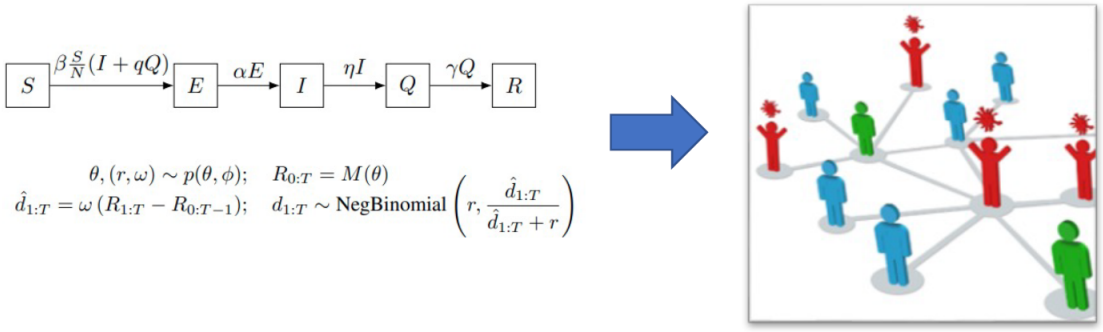


Figure 14: ODE Based ABM Calibration

4.12 Beam Search Particle Filtering

BSPF unites particle filtering with beam search and integrates intelligent adjustments to parameter perturbations, resulting in an efficient exploration of the parameter space to discover the most suitable model parameters for forecasting epidemic dynamics [**?**]. This strategy helps surmount the computational challenges associated with agent-based models and yields more precise predictions.

Initialization: In the first step of BSPF, an initial set of particles is created, each representing a possible scenario for the epidemic's progression. These particles are generated by sampling from a prior distribution, essentially capturing a range of potential outcomes.

Prediction: During the prediction phase, these particles are used to simulate how the epidemic might evolve based on the current state and historical observations. This process generates new particles that represent different future scenarios for the epidemic's progression.

Likelihood Calculation: In this step, the model compares the predicted epidemic curve with the observed data. By calculating the likelihood of the observed data given the model's parameters, it quantifies how well the model's predictions align with real-world observations.

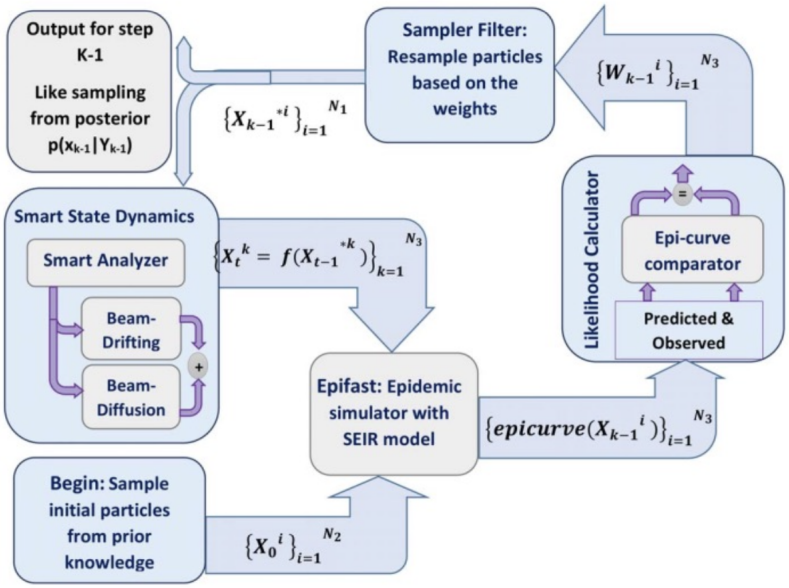
Weight Update: Based on the calculated likelihood scores, weights are assigned to each particle. Particles that closely match the observed data receive higher weights. The weights are then normalized so that they sum up to one, creating a probability distribution over the particles.

Resampling: Resampling is a crucial step to ensure that particles representing better scenarios are preserved for the next iteration. Particles are selected for the next round with a probability proportional to their weights, so particles with higher weights have a higher chance of being retained.

Smart State Dynamics: After resampling, the model assesses the differences between predicted and observed epidemic curves. It categorizes state vectors (particles) based on these differences and then adjusts the parameters of these vectors intelligently to improve predictions. The perturbations applied to parameters are carefully determined.

Adaptive Tuning: This step involves adapting the step size for parameter perturbations based on the history of particle evolution. It ensures that the model efficiently explores the parameter space, leading to convergence towards optimal parameter values.

Repeat: The entire process, from prediction to resampling and adaptive tuning, is repeated for multiple iterations. With each iteration, the model refines its understanding of the epidemic and narrows down parameter values to make more accurate predictions. This iterative approach allows BSPF to continuously improve its forecasting capabilities for infectious disease outbreaks.



4.13 Model Initialization via Digital Sources of Data

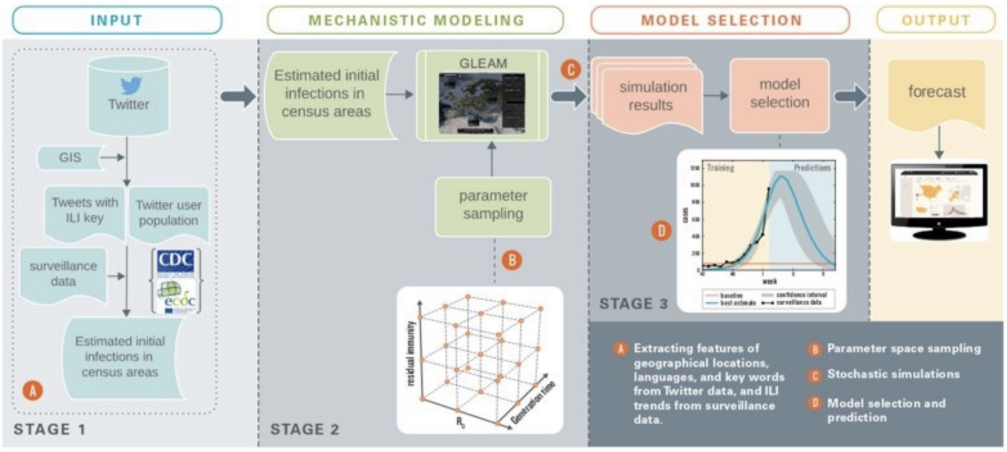
In epidemiological modeling, digital data sources like geotagged Twitter data and GIS are integrated to kickstart the modeling process [**?**]. Twitter data is filtered for ILI keywords, combined with GIS for geospatial analysis, and used alongside real-time ILI surveillance data. This data synergy forms the basis for accurate disease spread models, crucial for outbreak forecasting and mitigation.

Input: Data collection begins with geotagged tweets from Twitter, filtered for keywords related to Influenza-Like Illness (ILI). Geographic Information System (GIS) is used to associate tweets with census areas. Twitter user population data is gathered, and real-time surveillance data on ILI cases is obtained. They use these inputs to lay the groundwork for modeling.

Mechanistic Modeling: Estimated initial infections in census areas serve as the starting point. They employ the Global Epidemic and Mobility Model (GLEAM) for simulating disease transmission and mobility patterns. Key parameters affecting disease spread, such as transmission rates, are systematically sampled.

Model Selection: Multiple GLEAM simulations produce various disease spread scenarios. Model selection involves evaluating simulations based on fit with surveillance data and predictive accuracy. They rigorously apply criteria to choose the simulations that best match real-world observations.

Output: The modeling process's culmination is accurate ILI spread forecasts. They synthesize selected simulation results to predict ILI propagation in census areas over time. These forecasts aid decision-makers and health authorities in preparedness and response, helping to mitigate ILI outbreaks and safeguard public health.



4.14 Recent idea: End-to-end learning

End-to-end learning, popular in deep learning, is gaining traction in mechanistic modeling. It utilizes differentiable modules for gradient-based optimization, streamlining model development. This approach finds connections in physics and biology, enhancing model fidelity. For example, a 2019 study demonstrated improved simulations by combining data-driven and mechanistic approaches. In systems biology, end-to-end learning augments mechanistic models, potentially enhancing accuracy and prediction.

4.15 Physics-informed Neural Networks

Physics-informed Neural Networks (PINNs) represent a groundbreaking approach to solving partial differential equations (PDEs) while seamlessly integrating deep learning and fundamental physics principles [**?**]. The workflow of PINNs involves several key steps. Initially, the problem is formally defined, specifying the PDE governing the physical phenomenon under investigation, alongside any accompanying boundary conditions and initial values. Next, a neural network architecture is designed, tailored to the specific complexity of the problem, comprising layers and neurons.

The heart of PINNs lies in their unique loss function, which combines traditional data-fitting components, such as Mean Squared Error (MSE), with a physics-based residual loss. This residual loss ensures that the neural network adheres to the underlying PDE, reinforcing the physical constraints. During the training phase, optimization techniques

like Stochastic Gradient Descent (SGD) are employed to fine-tune the network's parameters—weights and biases. As the training progresses, PINNs become adept at generalization, adapting to various scenarios and offering high-resolution predictions even at unobserved data points.

In summary, PINNs are a pioneering methodology that harnesses the power of deep learning to accurately solve complex PDEs, paving the way for versatile applications across numerous domains where understanding and modeling physical phenomena are paramount.

4.16 PINNs need more epidemiological knowledge

To tackle the challenge of enhancing the accuracy of predicting epidemic dynamics within neural networks, they are confronted with the limitations of traditional methods, including the calibration of mechanistic models and the utilization of ordinary differential equations (ODEs) [**?**]. In response, their proposed solution leverages the power of transfer learning by combining a Physics-Informed Neural Network (PINN) as a "time module" with an RNN serving as a "feature module." This innovative integration enables them to tap into diverse data sources and synergize their insights, ultimately leading to more precise predictions of epidemic dynamics. Their ultimate goal is to significantly improve prediction accuracy and strengthen the correlation between their model's output and real-world data.

The time module, represented as $N_{\text{time}}(t)$, is a multi-layer perceptron that takes time t as input and predicts ODE states s_t at time t. The goal is to make $N_{\text{time}}(t)$ produce predictions that align with the epidemic dynamics described by the ODEs f_{ODE}, which means $N_{\text{time}}(t) = s_t$, while satisfying the ODE constraint $\frac{ds_t}{dt} = f_{\text{ODE}}(s_t, \Omega_t)$, where Ω_t represents learned ODE parameters at time t. The optimization involves minimizing the ODE loss (unsupervised loss $L_{\text{ODE-T}}$) and fitting the observed data (supervised loss $L_{\text{Data-T}}$). The aim is to discover the latent dynamics by obtaining s_t and Ω_t.

1. ODE Loss (Unsupervised Loss):

$$L_{ODE-T} = \frac{1}{N+1} \sum_{t=t_0}^{t_N} \left(\frac{dst}{dt} - f_{ODE}(st, \Omega_t) \right)^2$$

2. Data Loss (Supervised Loss):

$$L_{Data-T} = \frac{1}{N+1} \sum_{t=t_0}^{t_N} \left(M_t^{\wedge} - M_t \right)^2$$

In contrast to most prior work in Physics-Informed Neural Networks (PINNs), where many system dynamics states are observed, in their SEIRM model, most states are latent. They address this challenge by adding epidemiological domain knowledge in the form of monotonicity constraints. These constraints adapt monotonicity losses from Muralidhar et al. (2018), penalizing non-monotonic consecutive predictions. The differences between consecutive predictions approximate derivatives, which they generalize to continuous derivatives by taking the limit as time (t) approaches zero, allowing direct computation with autograd. They use domain knowledge to impose these constraints. For example, in SEIRM, they ensure that the Susceptible state (S_t) monotonically decreases and the Recovered state (R_t) monotonically increases by adding penalties when $\frac{dS_t}{dt}$ is positive and $\frac{dR_t}{dt}$ is negative:

$$L_{\text{Mono}} = \frac{1}{N+1}\left(\sum_{t=t_0}^{t_N}\frac{dS_t}{dt}\text{ReLU}(\frac{dS_t}{dt}) - \sum_{t=t_0}^{t_N}\frac{dR_t}{dt}\text{ReLU}(-\frac{dR_t}{dt})\right)$$

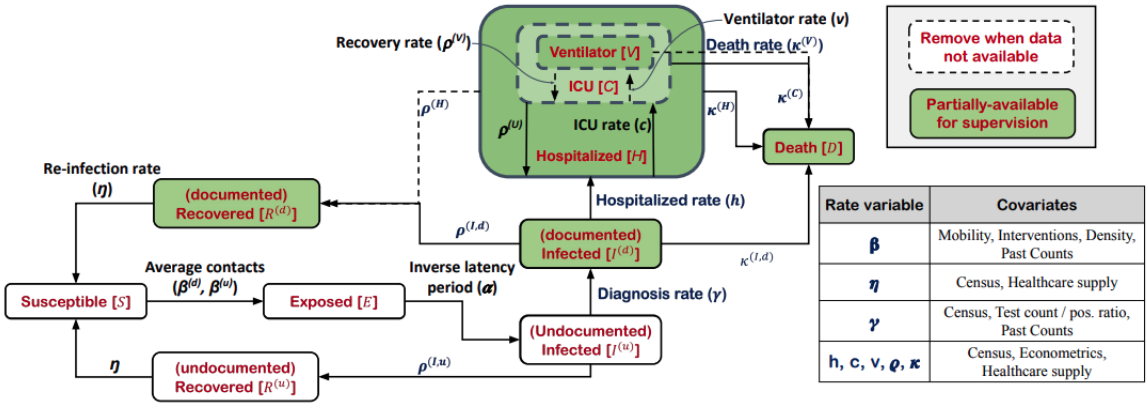
Here, $\text{ReLU}(x) = \max(0, x)$. S_t and R_t are part of st, the output of the time module, enabling computation of $\frac{dS_t}{dt}$ and $\frac{dR_t}{dt}$ with autograd.

One of the central issues in fitting Physics-Informed Neural Networks (PINNs) is the spectral bias of neural networks, which is their tendency to fit low-frequency signals (Wang et al., 2021b). To address this, neural networks are often given more flexibility to fit high-frequency systems. In this work, we employ Gaussian Random Fourier feature mappings (Tancik et al., 2020): $\Gamma(v) = [\cos(2\pi Bv), \sin(2\pi Bv)]^T$, where each entry in $B \in \mathbb{R}^{d\times 1}$ is sampled from $N(0, \sigma^2)$, with σ being a hyperparameter.

4.17 End-to-end Differentiable Learning with ODEs

End-to-end Differentiable Learning with ODEs represents a cutting-edge methodology for COVID-19 forecasting, bringing together the power of differential equations and machine learning within a single, integrated framework [**?**]. This approach builds upon the traditional SEIR model by introducing additional compartments to account for undocumented cases, hospital resource usage, and other vital aspects of disease dynamics.

One of its standout features is the interpretable encoding of covariates, enabling users, including healthcare professionals and policymakers, to grasp how various factors, such as population density, mobility patterns, and intervention measures, influence the disease's trajectory. This transparency fosters trust in the model's forecasts. Incorporating constraints and regularization terms ensures that the model's predictions align with established epidemiological principles. Notably, constraints are placed on the effective reproductive number (R0), and regularization promotes smooth trends while discouraging overfitting. Ultimately, End-to-end Differentiable Learning with ODEs serves a crucial role by offering actionable insights into COVID-19 progression. These insights can aid healthcare institutions in resource allocation, support informed policymaking, guide businesses in adjusting strategies, and instill confidence in the broader population regarding the decisions made by authorities.



4.18 End-to-end Differentiable Learning with ABMs

End-to-end Differentiable Learning with ABMs introduces an innovative approach that combines the strengths of epidemiological modeling and agent-based modeling (ABM) while addressing their individual limitations [**?**]. Traditional epidemiological models, primarily based on ordinary differential equations (ODEs), often simplify the complexity of disease transmission by assuming population homogeneity. In contrast, ABMs offer a more detailed representation of individual behavior and contact interactions but lack differentiability, making them less scalable and challenging to integrate with data sources.

To overcome these challenges, the authors present "GRADABM," a novel design for ABMs. GRADABM is designed to be scalable, fast, and differentiable, allowing for rapid simulations on standard hardware. Its key innovation lies in its ability to support both fast forward simulations and differentiable inverse simulations. This differentiability makes it compatible with deep neural networks (DNNs) and enables the incorporation of diverse data sources for calibration, forecasting, and policy evaluation. The approach encompasses several critical components, including a Transmission Model that calculates infection probabilities, a Progression Model that simulates disease stage transitions, a Calibration process that optimizes model parameters using auxiliary data, and a Training phase that iteratively refines the model through gradient-based optimization. This training process aims to minimize the disparity between model predictions and observed data, ultimately improving the model's accuracy. Once trained, the model can be employed for various purposes, such as forecasting disease spread in new scenarios or providing valuable insights for informed policy decision-making. GRADABM's combination of differentiability and scalability positions it as a versatile and powerful tool for epidemiological modeling, offering the potential to bridge the gap between agent-based modeling and data-driven approaches in epidemiology.

4.19 Benefits (vs Other Calibration Methods)

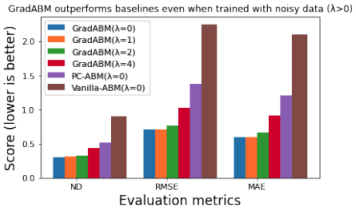
GRADABM outperforms traditional agent-based modeling (ABM) calibration methods, enhancing forecasting accuracy for infectious diseases, such as COVID-19 and Influenza [**?**]. It achieves a significant reduction in error metrics when compared to Vanilla-ABM and other methods, thanks to its gradient-based calibration, data integration capabilities through CALIBNN, and transfer learning.

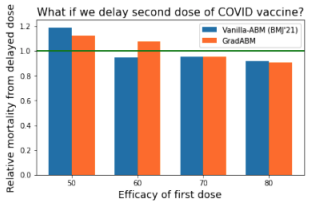
CALIBNN's integration is crucial, enabling the use of diverse data sources and flexible optimization through neural networks. GRADABM's ability to personalize parameters between counties while leveraging shared knowledge sets it apart.

GRADABM scales efficiently, running simulations rapidly on large populations. Its resistance to noisy data makes it reliable for public health scenarios, and it aids in policy decision-making, as demonstrated in a vaccine delay analysis.

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 [48]	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





4.20 Other methods

Bayesian Calibration: This approach utilizes Bayesian statistics to estimate model parameters by updating a prior distribution with observed data. It is effective in handling uncertainty and provides probabilistic parameter estimates.

Iterative Methods: Iterative techniques involve repeatedly adjusting model parameters until the simulation outputs match observed data. These methods are often computationally intensive but can yield accurate results.

Rejection ABC (Approximate Bayesian Computation): ABC methods involve generating numerous simulations with different parameter values and accepting those that closely match observed data. It's a computationally expensive approach but can be applied when the likelihood function is difficult to compute.

Markov Chain Monte Carlo (MCMC): MCMC techniques, like the Metropolis-Hastings algorithm, are used for sampling from probability distributions to estimate parameters. They are suitable for complex models and can provide posterior parameter distributions.

References

[1] B. Abrahao, F. Chierichetti, R. Kleinberg, and A. Panconesi. Trace complexity of network inference. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 491–499, 2013.

[2] 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):49, 2021.

[3] Aleta. Modelling the impact of testing, contact tracing and household quarantine on second waves of covid-19. *Nature Human Behavior*, pages 964–971, 2020.

[4] 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.

[5] 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.

[6] A. Chopra, A. Rodríguez, J. Subramanian, B. Krishnamurthy, B. A. Prakash, and R. Raskar. Differentiable agent-based epidemiological modeling for end-to-end learning. 2022.

[7] H. Daneshmand, M. Gomez-Rodriguez, L. Song, and B. Schoelkopf. Estimating diffusion network structures: Recovery conditions, sample complexity & soft-thresholding algorithm. In *International conference on machine learning*, pages 793–801. PMLR, 2014.

[8] V. Das Swain, J. Xie, M. Madan, S. Sargolzaei, J. Cai, M. De Choudhury, G. D. Abowd, L. N. Steimle, and B. A. Prakash. Empirical networks for localized covid-19 interventions using wifi infrastructure at university campuses. *Frontiers in Digital Health*, 5:1060828, 2023.

[9] K. T. D. Eames and M. J. Keeling. Contact tracing and disease control. *Proceedings: Biological Sciences*, 270(1533):2565–2571, 2003.

[10] M. Gomez-Rodriguez, J. Leskovec, and A. Krause. Inferring networks of diffusion and influence. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 5(4):1–37, 2012.

[11] P. Netrapalli and S. Sanghavi. Learning the graph of epidemic cascades. *ACM SIGMETRICS Performance Evaluation Review*, 40(1):211–222, 2012.

[12] M. Raissi, P. Perdikaris, and G. E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019.

[13] M. G. Rodriguez, D. Balduzzi, and B. Schölkopf. Uncovering the temporal dynamics of diffusion networks. *arXiv preprint arXiv:1105.0697*, 2011.

[14] A. Rodríguez, J. Cui, N. Ramakrishnan, B. Adhikari, and B. A. Prakash. Einns: Epidemiologically-informed neural networks. 2023.

[15] F. Tabataba, B. L. Lewis, M. Hosseinipour, F. S. Tabataba, S. Venkatramanan, J. Chen, D. M. Higdon, and M. V. Marathe. Epidemic forecasting framework combining agent-based models and smart beam particle filtering. pages 1099–1104, 2017.

[16] J. Wang, X. Wang, and J. Wu. Inferring metapopulation propagation network for intra-city epidemic control and prevention. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 830–838, 2018.

[17] D. Welch, S. Bansal, and D. R. Hunter. Statistical inference to advance network models in epidemiology. *Epidemics*, 3(1):38–45, 2011.

[18] K. Zhang, R. Arablouei, and R. Jurdak. Predicting prevalence of influenza-like illness from geo-tagged tweets. page 1327–1334, 2017.