MWEPI-VIZ: AN INTERACTIVE VISUALIZATION DASHBOARD FOR ENSEMBLE SIMULATIONS IN COMPUTATIONAL EPIDEMIOLOGY

BY

HO SHAN LAM

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Computer Science
in the Graduate College of the
University of Illinois Urbana-Champaign, 2023

Urbana, Illinois

Adviser:

Professor John Hart

ABSTRACT

The COVID-19 pandemic has prompted the development of numerous visualization tools to facilitate understanding and communicating on epidemiological research findings. However, existing visualization work for agent-based epidemiology simulations often focuses on macro-scale phenomena, while overlooking the connection between macro-scale trends and micro-scale agent behaviors. Portraying this connection accurately for ensemble simulations can bring substantial insights into both the epidemiology of infectious diseases and public health intervention design and evaluation. To address this gap, we present MWEpi-Viz, an interactive dashboard that facilitates exploration of ensemble datasets generated via epidemiological agent-based models. In this thesis, we describe the development process, system design, and efficacy of MWEpi-Viz, which draws upon months of participatory research with a computational epidemiologist. We illustrate the utility of MWEpi-Viz using an ensemble dataset from a counterfactual COVID-19 scenario in the Champaign-Urbana community, August 2020.

ACKNOWLEDGMENTS

I am extremely grateful to my advisor, Professor John Hart, for his guidance since the beginning of my journey on visualization for epidemiology. He is always available to give invaluable insights and recommendations to my research.

I am also deeply indebted Dr. Santiago Núñez-Corrales, who mentored me throughout the development of MWEpiz, for his belief in my abilities and his dedication to the project. The completion of my thesis would not have been possible without his help and vision.

I would like to thank the Department of Computer Science and the College of Engineering for the assistantships and resources I received.

I would also like to thank Stuart Levy and Kalina Borkiewicz from the Adavanced Visualization Lab at NCSA who introduced me to this research opportunity and offered feedback to my work. I learned a lot from them during our weekly visualization journal club.

Many thanks to my family for their support throughout my studies. Most importantly, I thank my partner Elfarouk Harb, for his love and encouragement.

TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION

Adequately visualizing the outputs from computational epidemiology ensemble simulations built using agent-based models (ABMs) remains an unavoidable and challenging task. These models were successfully applied across multiple geographical scales during the COVID-19 pandemic from entire nations [1] to small communities [2], showcasing advantages otherwise unattainable with other simulation methods. During the pandemic, the significance of visually conveying such information to scientists, decision makers and the general public became apparent. Developing correct intuitions about how a pandemic operates in terms of how individual behaviors shape collective outcomes determines the effectiveness of interventions on public health and society at large, the ability to adapt these measures as predictions are obtained an updated, and compliance with specific measures as individuals connect their behavior and individual responsibilities to the larger panorama of the unfolding situation.

ABM introduces visualization challenges different from those found in other simulation paradigms [3, 4, 5]. The first one arises as a consequence of why they are used in the first place: problems where *agency* shapes the space of collective decisions –e.g., contrary to passive forces operating on particles- tend to go beyond what can be expressed with analytical models conveniently, and even when that occurs, there are no closed solutions available to use in a convenient manner. This poses the need to map the heterogeneity not only of the composition of a social system undergoing an epidemic process, but of the rules governing individual agent decisions. Second, agents not only operate individually according to rules of varying complexity while traversing environments that modulate their behavior –e.g., how individuals may spend different amounts of time depending on the establishment they visit during a pandemic- but they interact in non-trivial ways. Visualizations intended to help experts and non-experts make sense of such highly diverse landscape of possible interactions and their consequences are met with practical limits of comprehensibility and spatial information density.

Visualizing the connection between properties derived from population-level observables –i.e., a quantity of the macroscale model- and individual agent actions in clear and intellectually profitable ways for multiple classes of users remains an open problem in visualization of ensemble simulations within computational epidemiology; doing so constitutes a fundamental task for crisis communication and management [6]. This gap appears to be a major source behind model misinterpretation for decision-makers and decision uncertainty for individuals experiencing interventions, as evidenced by the wide spectrum public attitudes toward

non-pharmaceutical interventions during the early days of COVID-19 [7] and patterns of distrust [8]. More broadly, the lack of visual tools depicting this connection in an interactive manner may contribute to increasing difficulties of the general population to understand how their individual behavior contributes to an ongoing epidemic. This opens, in turn, future opportunities to expose more clearly through visual metaphors how seemingly independent biological, economic and social mechanisms work together and alter the course of actions and decisions across time in ways that may be counter-intuitive and hard to predict.

In this manuscript, we address this challenge through the design and implementation of MWEpi-Viz, the Many-World Epidemiological Visualization platform for simulation data obtained from the Epidemiology Workbench [2], an ensemble simulation platform for infectious diseases implemented in Python using Mesa [9]. Our platform makes use of interactive multi-level information selection and scenario re-enacting by presenting curated views of agents, events, and trajectories computed across individual scenarios and entire ensembles. To better engage and benefit from the human visual system, we carefully designed an interactive dashboard capable of threading a single narrative, that of an epidemic across time at multiple conceptual levels while still maintaining a coherent and unified view of the entire phenomenology driving these processes.

CHAPTER 2: RELATED WORK

2.1 VISUALIZING PANDEMIC OUTCOMES FROM SIMULATION ENSEMBLES

ABMs are used to uncover possible presence of *emergent behavior* in complex systems, epidemics being a paradigmatic example. By emergence we mean the rise of collective properties (i.e., the macroscale) that could not be anticipated from the dynamics of individual agents (i.e., the microscale), a consequence of their patterns of interaction across time. These are generative effects, surprises that bear significance at the level of the system, which often entail a substantial exploration of the space of parameters determining the individual rules in the system. ABMs are a powerful tool in the arsenal of generative social science for that reason [10]. However, capturing the bridge that connects events in the microscale to changes in the macroscale is not trivial. The non-linearity observed in macroscale behavior arises due to the compositional effect of interactions, which are transient events but can cooperate to endow the system with new dynamics. For instance, the standard epidemiological SIR model produces an infected peak corresponding to a fraction of the population [11], which can be explained analytically by the product of infected and susceptible individuals given β contacts per day and mean recovery rate γ,

$$\frac{dS}{dt} = -\beta SI, \tag{2.1}$$

$$\frac{dI}{dt} = \beta SI - \gamma I, \tag{2.2}$$

$$\frac{dR}{dt} = \gamma I. \tag{2.3}$$

Visualization-wise, the peak for the infected portion of individuals is informative for several public health issues, and became a staple during COVID-19 pandemic science and media presentations, but offered no mechanistic insights into the processes responsible for it. To uncover how this particular curve evolves, though, it is necessary to resort to the statistical physics of networks [12, 13] or to interacting particle systems theory [14, 15] and reconstruct the probabilistic processes corresponding to the interactions mentioned above. Observe that each interaction between agents is a discrete event, that agents can choose among a finite number of possible decision alternatives at each step, and that we now need to introduce an *ensemble* to sample the space of possible trajectories may follow in any given simulation, even if the rules they follow are deterministic. By computing various moments of the distri-

butions obtained across the multitude of possible worlds in an ABM ensemble simulation, whether network or an interacting particle system contagion dynamics are used, we approximate the continuous SIR model expressed through the ordinary differential equations (Eqs. 2.1-2.3) above. These only portray how fractions of the population move across multiple compartments that represent stages of the disease in question. Incidentally, the reasoning above implies that ABMs intended for limited time forecasting of disease spread and public policy design must introduce probability distributions at the agent level, and ensemble simulation at the macroscale level.

Despite the fact that the mathematical reasoning needed to uncover the connection between infectious interaction processes is well known, it is by no means trivial to extend it and include non-pharmaceutical and pharmaceutical interventions [16, 17]. Moreover, the connection has become even more inscrutable for epidemiologists, public health experts, decision makers and the general public. Due to the conceptual and analytic complexity behind recovering the the bridge connecting microscale dynamics to macroscale observables, computational epidemiologists –and by extension, visualization experts- are often faced with hard choices to communicate their results, often leading to presenting only one perspective of the epidemic process. One of them is to restrict oneself to ODE-based compartment models, simple to interpret and modify at the macroscale level to account for various interventions [18], but uninformative of fine-grained individual dynamics that matter when devising public policy measures to maximize compliance. ODE models are suitable for building dashboards, since their numerical solution is inexpensive enough to embed in web pages, and moving parameters with sliders allow users to quickly explore what the parameters do to build intuitions.

However, as the models gain compartments to compensate for more complex diseases such as COVID-19, the behavior of the system become less intuitive through visual inspection. Not only we lose clarity on the response of the model as parameter changes, but now multiple curves are needed to make sense of the process. Even worse, adding confidence intervals requires a costly estimation process that is somewhat artificial. Another alternative is to resort to a stochastic differential equations (SDE) version of SIR models, that attempts to recover part of the microscale behavior through probability distributions [19] and gain realism. User controls become more complex, leading to intermediate visualization of noise functions and probability distributions modulating disease-related parameters. Neither the ODE or the SIR admit an intuitive introduction of spatial elements, critical for realistic simulation of geography-dependent effects. Finally, using ABMs or network models reintroduce spatial dependencies and open the black box in terms of the microscale at the expense of greater visual complexity were the micro-to-macro bridge were to be exposed, particularly in large

ensemble simulations.

Though visualizations of pandemic modelling results have existed for a long time, it has received tremendous attention with the emergence of COVID-19. In this section, we discuss the recent works, the characteristics of ensemble data and epidemiology data, and their related visualization challenges.

2.2 EXISTING VISUALIZATIONS FOR COVID-19 PREDICTION MODELS

As COVID-19 drastically altered the lives of billions around the world during its peak, interest in epidemiology research and the need for better visuals to understand computational epidemiology models have been on the rise since. Traditionally, visualizations are used for reporting statistical summaries from epidemic simulation outcomes, and are instrumental to better interpret and communicate complex and intricate results. While many online COVID dashboards from governmental and public entities have swiftly appeared to inform public about the actual recorded COVID-19 incidents [20] thanks to the maturity of web frameworks and commercial business intelligence software, which support quickly assembly of classical statistical graphs, e.g. line chart, map, data tables, and pictograms. These tools are helpful for providing informative insights at a glance, but are not sophisticated enough to explain and explore computational epidemiology results. Hence, custom-made interactive dashboards have been becoming more popular for decision-makers or epidemiologists to conveniently generate and analyse scenario-based forecasting.

The design of custom-made dashboards are highly dependant on the usage of the model and are geared towards reducing time and domain expertise to execute simulations. One example is COVIs [21], a dashboard that allows journalists to comprehend the impact of different time-sensitive COVID-19 policies with a prediction model, by visualizing the epidemic curves in various countries under different scenarios defined by the user. Afzal et al. [22] proposed a map-view dashboard for running and viewing results from their COVID-19 simulation model on state-level communities, where the interface has a control panel for tuning model parameters and a map-view indicating the severity of COVID-19 across regions. Mahmood et al. [23] built a dashboard for their dengue fever agent-based simulation to report the geolocations of humans and mosquitoes and the epidemic curves. These viusalization tools have improved accesibility and interpretibility of epidemiological models, but were not adequate to uncover the full picture of ABM ensemble outcomes.

ABM simulates interactions between agents and environment to facilitate understanding of complex phenomena [3] for fields such as ecology, geography, and urban planning. ABM data is typical spatial-temporal consisting of an environment –e.g. grids or geographic area– and

agents, and animation is common for displaying changes of agent behaviors in an environment through time. Recent breakthroughs in ABM visualizations [24] greatly reduced the work of translating simulation states to visual outputs, but existing visualizations are still inherently descriptive - in a sense that they display an animation of agents moving in an environment accompanied by statistical summaries. Neither time-persistent information of an agent nor the environmental impact of agents can be efficiently explored. In Figure 2.1 we show a classical visualization paradigm of ABM simulation using the Python package - Mesa [9]. Our original implementation of the ensemble simulation also uses this platform, and it offers a real-time visualization tool while running an ABM simulation.

2.3 CHALLENGES IN VISUALIZING ENSEMBLE DATA

An ensemble data is a collection of outputs from rule-based simulations with slight perturbations of initial settings. Ensemble modelling is commonly used in scientific fields such meteorology, physics, biology, for predicting complex and uncertain outcomes. Due the nature of the simulations, the resulting datasets are often spatial-temporal which record the attributes of different members at each timestamp.

There are several overarching challenges in ensemble visualizations for all types of data [25]. Here we name two that are particularly relevant to our COVID-19 ABM. The first one is data management – depending on the model complexity and the number of simulations, an ensemble can be computationally expensive to run and to store. Off-the-shelve interactive visualization tools are normally insufficient for handling a great volume of data because responsive interactions require prior data cleaning and aggregation. Therefore, efficient data retrieval is an indispensable part to interacting with large volume of epidemiological data. The second one is dimensionality reduction – ensembles are used for observing many possible evolution of a complex system at a given space, and none of the essential goals is to compare the different worlds so to draw meaningful inferences. However, because of the highly dimensional nature of ensemble datasets, they require careful aggregation and novel visualizations for enabling pattern recognition.

Displaying statistically aggregated views of all simulations of an ensemble is a common technique to flatten ensemble data and illustrate uncertainty. Spaghetti plot is a classical technique for arranging a large volume of ensemble members simultaneously in the form of lines, so to accentuate the overall trend and outliers, but it is prone to the problem of visual clutters. Many ensemble visualization techniques extend the spaghetti plot. For instance, contour Boxplots [26] fuses spaghetti plots and box plots to visualize a large ensemble of spatial-temporal curves pressure field generated by fluid simulations, by displaying only con-

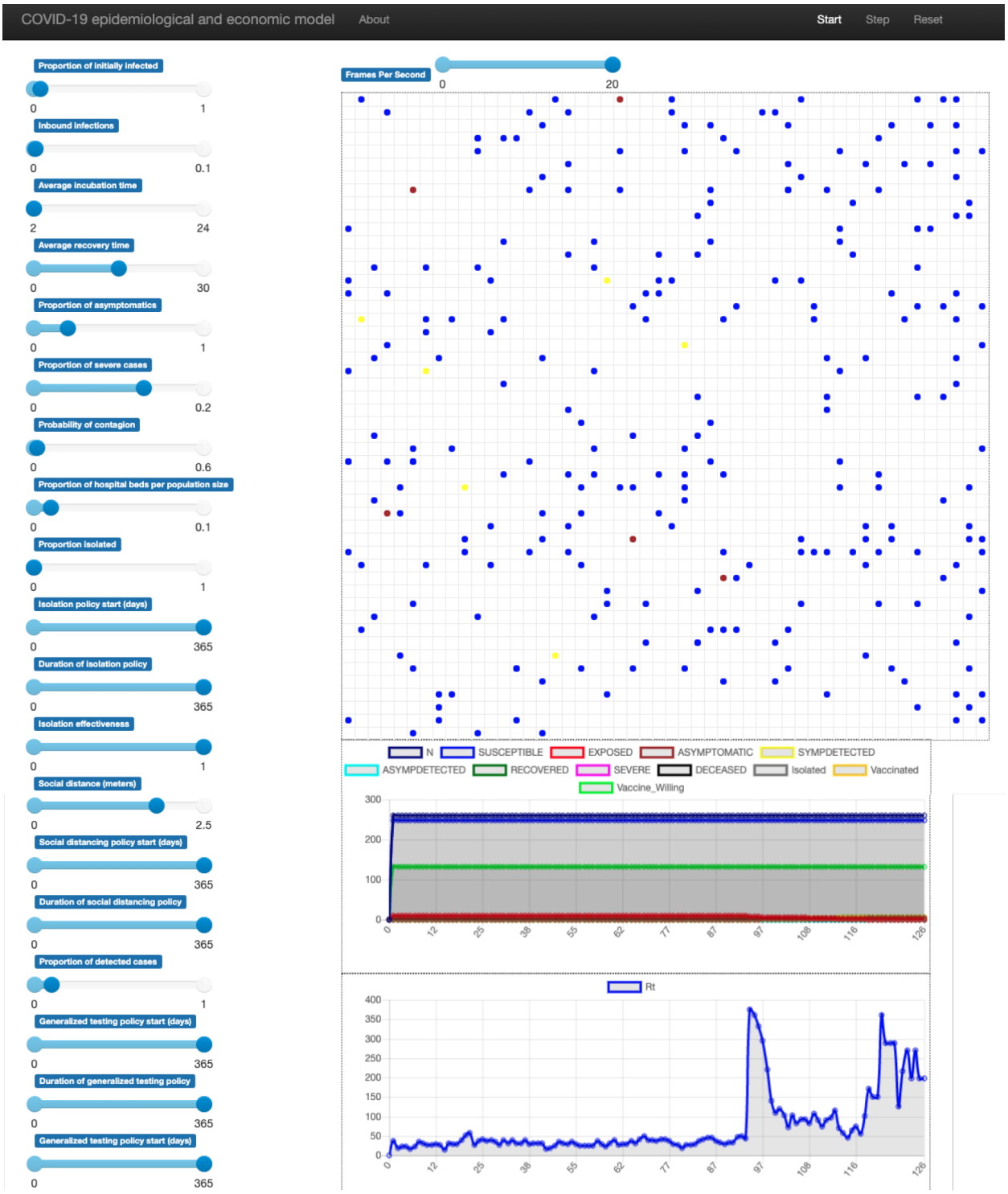


Figure 2.1: Real-time ABM visualization using Python Mesa package for simulating COVID-19 in a 50x50 grid. The dashboard constructed using Mesa displays a control panel for simulation runs (top right), a panel of sliders to tune simulation parameters (left), a grid visualization with each agent colored by their respective states at each timestamp (center), and multiple components with statistical plots (bottom).

tour lines on the quartile and extremes while coloring the interquartile region like box plots to indicate the dispersion and skewness. Curve Boxplot [27] is another similar method to characterize 2D and 3D ensemble curves efficiently using non-parametric statistical analysis to extract representative curves from the ensemble. In order to support static ensemble visualization, Hao et al. [28] proposed a cluster tree method to organize similar ensemble members. Parallel coordinates plot is another popular choice [29], [30], [31] for uncovering correlations between input parameters and model outcomes of all simulations, which displays multiple feature axes in parallel and plots all ensemble members as lines across them. Filtering and coloring are two common techniques to reduce of visual clutters and enhance readability.

Interactive visual explorers coupled with novel visualizations is another popular approach to organize ensemble datasets because interactivity allows a multi-level organization for visualizers to encode more information than static graphs. In particular, it is popular to cluster similar outcomes in ensembles to support pattern discovery and enable drill-down exploration through the cluster visualizations. [32] proposed an interactive visualization for exploring isocontours of weather forecast ensembles, in which it displays a box-plot-inspired simplified spaghetti plot of isocontours and enables users to select groups of similar isocontours with a node-link diagram generated by a clustering algorithm. [33] proposed an interactive visual analytics tool to compare an ensemble of real ocean currents across time against simulated currents; they applied clustering algorithm to both ensembles and highlight the geospatial regions of the real and model currents by their clusters, for users to reason the accuracy of their simulations. [30] designed a three-view interactive visualization tool for 2D functions ensemble data to facilitate a drill-down exploration pattern. They visualized each member function as a glyph and placed all the members in accordance with their clustering membership.

The visualizations of ensembles have always been highly tailored in accordance to the underlying data structure. [25] argued that ensemble data is so complex that there is no single visualization that could satisfy all the competing goals. Hence, they advocated for a multi-view visualization containing traditional charts, novel visualizations and interactivity would be the best approach. Most prior ensemble visualizations indeed adopted this approach and we followed this design methodology in our work.

2.4 PARTICIPATORY DESIGN

Participatory design has long been a design method before the term was coined, where designers incrementally collect design insights by observing users in their natural environ-

ment, developing prototypes, and collecting feedback from users [34]. This method does not limit itself to only software development. Richard Drew, an engineer from 3M, invented the Scotch Tape in 1923 by working closely with painters from an automobile repair shop for two years to create a secure, waterproof masking tape with the right adhesiveness perfect for masking surfaces for painting cars. Spinuzzi in 2005 first established that participatory design as a research methodology rather than a mere design approach [35]. She constructed three stages in participatory research design that researchers cycle through during development: (1) initial exploration of work, (2) discovery processes, (3) prototyping. She argued that compared with traditional research, where the objective is to extract and abstract knowledge transferable to other domains, participatory research aims at discovering "tacit knowledge" from users that are difficult to formalize and express. Since users and researchers collaborate closely, researchers are able to draw out hidden knowledge and create user-centered design with a higher success in adoption. Considering the diverse needs from different stakeholders in computational epidemiology research, in order to better scope our project, we chose participatory design as our research methodology and collaborated with one of the computational epidemiologists on the Epidemiology Workbench [2].

CHAPTER 3: REQUIREMENTS

Our goal is to build an interactive dashboard that surfaces the dynamics between individual agent behaviors, public health policies, and viral transmission. MWEpi-Viz visualizes ensemble data from the Epidemiology Workbench and is the result of a participatory design with a computational epidemiologist, who provides the ensemble data and engages in weekly design meetings. In this section, we explain the driving questions guiding the design of our dashboard for exploring ABM epidemiology ensemble data.

3.1 SCIENTIFIC REQUIREMENTS

Visualization in ensemble simulations of epidemic processes addresses three separate concerns. First, identifying the main points that are significant in terms of dynamical evolution of the interaction between pathogen and the population viewed as a system. Second, once these moments become salient, the work of computational epidemiology is to connect individual behaviors to potential interventions to measures accompanied by indicators obtainable from public health data. We are interested for this part on the ability to qualify –and quantify- the performance of interventions by translating them into the language of biological and populational mechanisms. For instance, the role of mask wearing can be interpreted at the level of its mechanical effect on droplets (i.e., reducing the magnitude of transferable viral loads across individuals), it individual effect on the probability of infection, and its populational effect of the spread of the virus given by drops in the count of newly exposed individuals. Third and most significant, the role of scientific visualization is to help construct and challenge hypotheses connecting the biology of the pathogen with attributes of individuals in order to obtain a clearer picture of how, where and when risks arises the most. This is pressingly relevant during early days of a pandemic when clinical and public health impacts are being just understood. In consequence, visualization tools must accomplish five main tasks.

(S1) Differentiate individual attributes that drive collective dynamics. Realistic epidemic models for decision-making often introduce demographic complexity by explicitly endowing agents with features associated with clinical health factors (e.g., presence of co-morbidities, age range) and well as societal risk factors (e.g., employment status, family cohabitation, transportation patterns). These may be explicitly stated as parameters of the simulation, or indirectly given by changes in accessible parameters under a given interpretation. In some cases, the attributes map directly onto

observable properties of individuals, while others correspond to abstract properties connected to the mathematical formulation of the model. Selectively visualizing simulation outcomes at the individual and collective level greatly simplifies the differential study of separate populations as they undergo an epidemic regime.

(S2) Connect parameters of the ensemble to statistical moments. Ensemble simulations of dynamical epidemic processes constitute a mean-field approximation of compartment models as indicated above. Then, given outcomes of an ensemble simulation, visualization tool need to provide access to the standard curves used in the interpretation of retrospective studies and forecast analyses. Contrary to ODE models, ensemble simulations enable calculation of uncertainties, bound by worst and best case scenarios across each variable of interest. Being able to associate a parameter set with a visual representation of moments computed from these curves provides critical insights about how reliable the resulting scenarios.

(S3) Constrain attention selectively to regions of epidemic interest. Applying ensemble models for advanced research and decision-making often result in long time series data that either approximately reproduce the past or enact various possible futures. Traditional visual representations tend to concentrate statically on one of three specific choices: 1) showing the entire epidemic process at once, 2) showing the most recent trends to then design future modeling exercises containing new public health interventions, and 3) focus only on certain time ranges where significant changes occur. Even when all three are present, these are disconnected from other visual elements that can help clarify and focus the user's attention on changes that may become imperceptible across the entire lifespan of the simulation.

(S4) Make relations between space and contagion processes explicit. A fundamental limitation of ODE and SDE models in contrast to ABM simulations is the ability to recover spatial patterns derived from the movement of the agents that further drive disease spread. ABMs are naturally multimodal in this regard. While visualizing this at the level of an ensemble does not provide specific information when the spatial representation does not include real data (e.g., based on GIS data), portraying selected ensemble elements –individual runs- remains informative. In particular, connecting specific curves from the ensemble to its specific realization in simulation space provides information about the distribution and reach of each agent, and thus about its infectious potential. More specifically, reasoning about agent trajectories as infective traces with finite effects can clarify why some policy measures may be easier to enforce

than others.

(S5) Connect statistics from collective outcomes to disease transitions. Given the statistical character of ensemble simulations, visualizing overall moments of the resulting distributions is generally uninformative. With the particular case of the Epidemiology Workbench in mind, events in realistic simulation frameworks depend on particular probability distributions to which adequate parameters are supplied. These distributions, in turn, model specific biological, clinical or social mechanisms tied to the spread of a disease and its consequences. Hence, the ability to design appropriate public health interventions translates into the ability to detect and predict changes in these distributions using data from simulations. In disease contagion models, discrete stages encode periods in which clinical outcomes vary sharply and distinguishably (e.g., changing from symptomatic to severe stages); each stage transition bears significance for specific stakeholders during the unfolding of an epidemic. Research-wise, understanding whether it is possible to intervene in ways that challenge the robust universality of contagion processes constitutes a fundamental and pressing question. To the best of our knowledge, there is no visualization that connects temporally constrained distributions for key observables in ensemble simulations of epidemic processes to disease stages despite it evident advantages.

3.2 PUBLIC HEALTH REQUIREMENTS

During the COVID-19 pandemic, several requirements for public health communication became apparent. In an unfolding situation, the main goal is to communicate with a broad range of stakeholders each of whom may need different levels of detail regarding existing information and possible future situational changes. For instance, the general public may interested in the individual risk they experience and the impacts of public health interventions on daily life in the short term, epidemiologists seek to understand the long term dynamics that impact the evolution trends of the pandemic, healthcare systems try to adapt to a changing demand altering the availability of hospital beds, and decision makers are tasked with integrating all the prior information into policy interventions that are effective.

Effectiveness in this context means that proper communication of the state of an epidemic should map into a common set of questions answerable across varying levels of resolution depending on available information [36]. Thus, any visualization tool that addresses unfolding scenarios, should cater to the following concerns:

(P1) Does the analysis provide graphical representation of the outcome over

time? As mentioned before, ensemble ABM simulations are simultaneously most reliable and hardest to convey due to their complexity. Not only the standard trends used in standard epidemiology practice need to be present, but their collective and individual properties as events in multiple possible worlds produce distributions that speak of the probabilities governing the evolution of spread.

(P2) Is there sufficient pre-intervention data to characterize pre-trends in the data? Simulations of the sort investigated here require careful calibration and accounting of past events in order to become useful for forecasting purposes. The latter entails the need to have mechanics in the visualization that help convey differences between pre-intervention, within intervention and post-intervention data trends across relevant state variables and compartment changes. It is worth noting that, due to the properties of an epidemic process, the outcome of public health interventions manifests within a delay, which also needs proper characterization. Another use case is public health intervention imputation from simulation data. In order to ensure objectivity, teams of modelers and public health experts not involved in devising public health intervention design simulation can be given the outcome of ensemble ABM simulation to evaluate whether the data shows specific changes of significance. The ability to visually narrow down time periods in this manner and evaluate differences in the resulting distributions constitutes a key tool for unbiased policy evaluation.

(P3) Is the pre-trend stable? Calibration requires choosing conditions in which the epidemic observables are predictable, and where the most significant intervening factors have been identified and characterized. To this extent, providing visualization mechanisms for ensemble runs allows generating and quickly testing the similarity between past data and generated data. We note that, to do that effectively regardless of specifics of each community, it is customary to display trends using population fractions rather than absolute population counts.

(P4) Is the functional form of the counterfactual (e.g. linear) well-justified and appropriate? A counterfactual scenario in computational epidemiology comprises a parameterization that corresponds to a reality in which a particular set of policy measures are absent, used to contrast against other scenarios where some or all of them may be implemented. Comparing between counterfactual and alternative scenarios constitutes a complex task. Visually, counterfactual simulations will approximate the behavior of an ODE compartment model without measures being applied to it. Hence, providing the ability to capture images of trends per scenario and visualizing simul-

taneously the trajectories will facilitate comparison against real data. In particular, with emphasis on data prior to an intervention, doing so helps identify potential issues with the calibration and thus with the quality of the parameterization of the model as a whole.

(P5) Is the date or time threshold set to the appropriate date or time (e.g. is there lag between the intervention and outcome)? Time lag effects exist between the formal issuing of public health intervention and its effect on the dynamics of the epidemic process due to aspects related to the clinical progression of the disease in exposed individuals. For example, the initial strain of SARS-CoV-2 had an average incubation time of seven days following a Poisson distribution [37]. Being able to convey the existence of such delay to decision makers and the general public can help increase individual compliance with policy measures only when the expected time at which effects will manifest is known. If simulations are adequately constructed, these lags should arise *in silico* as well, and visualizations generated from their outcomes should clearly show changes in the trends associated with changes in individual behavior driven by such interventions. The latter is, in general, a non-trivial challenge. In terms of simulation data analysis by teams that evaluate the quality of models, effective interventions will appear consistently across the entire ensemble and, thus, both the mean trends and trends per run will display these changes when visualized.

(P6) Is this policy the only uncontrolled or unadjusted-for way in which the outcome could have changed during the measurement period? Pandemic control is, in general, a concurrent decision process. Multiple interventions are implemented at once, often exhibit synergistic effects [2, 38], making the imputation of outcomes a complex task. One way in which ABM simulation can be used to understand each effect in an isolated manner, and then their synergies, is by creating differential scenarios following an adequate design of experiments that aims to maximize differences across combinations of measures and reduce the total number of scenarios to be run. Having a tool where ensemble outcomes can be visually interrogated in terms of trends and average, individual model runs from the perspective of collections of agents, and distributions arising from the time-dependent stage change process is indispensable to fully explore the consequences of measures. As an example, modeling toward campus reopening in Fall 2020 for UIUC [2] showed the synergistic effect between testing and mask wearing as a function of changes in the proportion of asymptomatics in the population. Even with limited visualization capabilities, the difference suggested that the testing regime would suffice to regain control after the epidemic peak produced by

mass ingress of students. However, standard visualization techniques do not provide information about the spatial consequences of interventions on disease spread, despite the fact that these can change the infective potential of individual by restricting the effectiveness of contagion or the average distance they cover during each infectious stage. Hence, visualizing and annotating sample runs becomes useful to portray clearly how individuals are impacted by top-level decisions.

3.3 VISUALIZATION REQUIREMENTS

To support efficient navigation of ABM ensembles at different levels of details, the visualization interface should organize the data such that the macro-scale model and individual agent actions can be reasonably linked and examined while catering the scientific and public health requirements mentioned above. As a result of the discussions with the domain expert, we crystallize the usability requirements and three major levels of resolutions of the ensemble data that would benefit greatly from visualizations for communication and analysis purposes.

In terms of usability, the visual explorer should be responsive to interactions that the latency between query and response should be minimized. While Nielson [39] suggested that $< 1s$ is ideal for an uninterrupted user experience for general events such as mouse click, a previous study on benchmarking interactive database visual query response time[40] suggested that the response time for continuous interactions should be $< 100ms$. Moreover, the system should display all visualizations reasonably large for comfortable viewing and interactions on smaller screen-space such as laptops. All visualizations should also be downloadable for the ease of sharing. Below are the three major levels of details:

(V1) Relationship between the ensemble average and individual runs. An overview of the trends of the disease spread in an ensemble is essential to characterize an ABM ensemble outcome, e.g. the number of agents at different stages across time of the ensemble average or in each individual run. Therefore, a visualization of an ensemble showing the time evolution viral spread through the ensemble average and of the individual runs is needed. At the same time, it should have a high degree of interactivity to serve as an anchor for further drill-down explorations. For example, displaying interactive elements connecting to other aggregated views, filtering subset of an ensemble data, or fixing the shared variables amongst the views.

(V2) Evolution of each agent in individual runs. Tracing the development of agents of an ABM model "in the wild" greatly helps building an intuitive understanding of

the model mechanics, agent behaviors, and the impact of critical events more than pure statistical summaries as users can observe actual interactions between agents in an environment. Standard ABM visualizations already achieved this by showing a naive animation of agents at their respective location at each discrete timestamp, but the discrete location recorded often results in an animation of agents hopping around, rendering users difficult to visually track the movement of an agent. Therefore, viewing an ABM animation where agents are moving from location to location smoothly becomes useful. Moreover, to create more useful depictions of agents while preserving their linkage with their environment, showing continuous evolution of an agent and its interactions with historical neighbors in the environment is needed. Hence, the visual explorer should show traces of agents such as visited locations, pathways, exposure risks.

(V3) Distributions at the agent level. The visual explorer should enable users to dissect the ensemble results at three different moments: before an intervention, during the change in behavior triggered by the intervention, after the intervention; with a flexibility to adjust the intervention start period and the duration to become fully in effect, and visualize distribution the number of days it takes for a state transition (e.g. from becoming infected to showing symptoms) to take place. Also, to summarize the state transitions of an ABM model succinctly, the visual explorer should have a state transition diagram with nodes as stages and edges as possible transitions, which summarizes an ensemble through a lens other than the classical epidemic curve.

CHAPTER 4: METHODOLOGY

4.1 COVID-19 SCENARIO SELECTION

To assess the functionality and effectiveness of MWEpi-Viz, we made use of an existing scenario used to calibrate and forecast disease spread trends in Champaign-Urbana during Fall 2020 reopening at UIUC [2]. Multiple reasons motivated this choice. First, the aforementioned scenario was used in a real-life situation to inform decision makers and public health policy experts. Second, it traces closely the local history of public policy measures at the time, with evidence of providing adequate initial calibration. Third, simulation data after August 9, 2020 in this particular scenario contains the description of a counterfactual –i.e., a scenario with not new policy measures after the indicated evaluation period- with marked differences between initial policies and then an unconstrained epidemic.

We focus on the period between May 1-7, 2020, which corresponds to the establishment of the mask mandate by the State of Illinois and gradual impact of the policy measure given the etiology of COVID-19. Concentrating our attention in this period presents two advantages: the effect of the mask mandate was visible both in real data and simulated scenarios, and mask wearing exhibits distinguishable patterns due to its effectiveness and scalability [41]. Moreover, given the compute intensity involved in reproducing this scenario, bounding the simulation period also makes this task feasible. It is worth noting that the Epidemiology Workbench was extended to record events of individual agents, and not only values from collective observables. Our selection satisfies requirements **P2** and **P3**. Since we do not perform visual comparisons across different scenarios, **P4** does not need to be satisfied here.

4.2 SOFTWARE IMPLEMENTATION

Since ensemble data can become too large for any modern web browser to process efficiently while preserving interactivity, prior data pre-processing on the backend is necessary. Our MWEpi-Viz explorer consists of three components: a dashboard, a server, and a database as shown in Fig. 4.1. The user first submits an ensemble data containing individual attributes of each agent at each timestep generated by the Epidemiology Workbench to a server, then the server generates the aggregated summaries needed by the visualizer and loads them into a database. Thanks to the availability of pre-processed data, the web dashboard only needs to render a small amount of data requested from the server at any given moment.

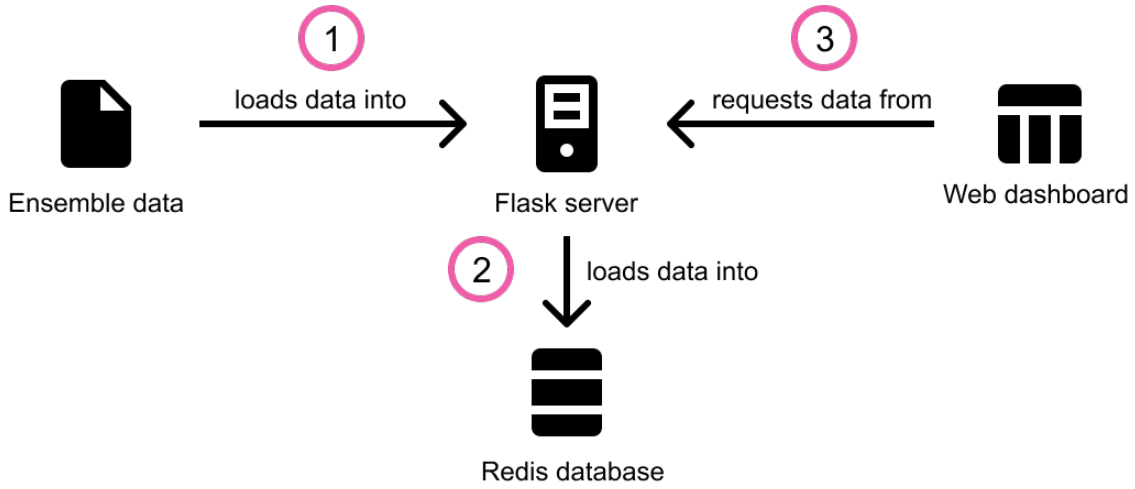


Figure 4.1: MWEpi-Viz software architecture. Ensemble data from ABM simulations with the Epidemiology Workbench are ingested into a Redis database via a Flask server. Finally, the web dashboard consumes pre-processed data for display purposes.

4.2.1 Data Specification

The Champaign-Urbana counterfactual ensemble data is a scenario of 1000 agents, undergoing a random walk in a torus environment structured as a 190×255 grid for 153 days. The agents have a probability to move to a new location every 15 minutes, resulting in total 14688 steps per simulation run. The simulation is repeated 5 times resulting in a ensemble of >77 million rows of data. A final dataset size of 2.3 GB reinforces the necessity for a data processing pipeline to achieve efficient visual queries in interactive web applications.

4.2.2 Implementation Tools

For the server, we used Python Flask [42] to create endpoints because of its simplicity. Code development was performed using Python v3.7 in order to match the version used in the Epidemiology Workbench [2]. Outcome datasets were processed using Python Pandas v1.3.5 [43] for its ease of data wrangling. For the database, we use Redis because its in-memory nature can support fast queries and its ability to store JSON strings enables seamless communication with the web dashboard. The dashboard is implemented with Svelte [44], a light-weight component-based front-end web framework. For most of the charts, the dashboard uses d3.js [45], a popular graphing library that enables direct DOM manipulations on web pages to generate interactive data visualizations. When visualizing a simulation animation in which all the agents are simultaneously moving on a grid, it is impractical to create a new DOM element per each agent and timestep with d3.js. Therefore, so we opted

for THREE.js [46], a JavaScript library that animates 2D or 3D graphics with WebGL.

4.3 ITERATIVE DESIGN

We conducted weekly meetings with the computational epidemiologist throughout the development process. At the initial exploration stage (stage 1), the scientist gave presentation of their work in which we learned the significance of their work on COVID-19 ABM simulations for the Champaign-Urbana community and the audience that would benefit from visualizations of epidemiology research. We then entered an incremental alternating design process between the discovery process (stage 2) and prototyping (stage 3). Before the first prototype, we prepared various existing visualization tools for ABM simulations or other types of epidemic models and discussed their merits and drawbacks during the weekly meetings. We learned what is needed to support intuitive connections between macro-scale changes and agent behaviors, and how the needs of different audience (e.g. policy makers, research scientists, general public) differ. These discussions helped us narrow down the focus of the visual explorer. We wanted to focus on empowering researchers to articulate the impact of the simulations, and so we excluded the user flow of comparing historical data with generated ensemble data for calibrating model parameters. At the same time, we ideated several visualization ideas such as coloring historical paths of an agent by its stage, displaying dwell-time and exposure risk to COVID-19 in the environment as a heatmaps. Conversely, the discussion also benefited the scientist to understand the technical possibility of the visualizations we brainstormed and learn about existing visualization techniques from other domains to visualize large-scale data.

After a few weeks of discussions, we began the process of prototyping. The first prototype in Figure 4.2 addressed a major usability concern of the scientist with the original visualization interface of the Python Mesa package - speed, where the visualizer suffered from significant framerate drops very early on in the simulation (at the end of day 1). Since the prototype was implemented with PIXI.js [47], a light-weight WebGL-based 2-dimensional visualization library, it rendered a snapshot for each timestep almost instantly, whereas Python Mesa was slower for it rendered the agents with the CanvasRenderingContext2D interface. This high-fidelity prototype led to the development of a low-fidelity prototype of the dashboard, which consolidated the dashboard layout where we grouped the standalone visualization ideas into hierarchically organized components that are coordinated through interactivity. When deciding on the dashboard composition, we referenced a recent study on dashboard design patterns [48]. It suggested that designing dashboard as an art of arranging information in a single screen-space by making trade-offs between level of details and the

amount of effort to explore required through interactions. Considering the complexity of ensemble datasets, the comfort of users on personal device, and the ease of extending the dashboard with new levels of aggregation, we opted for a multi-page layout instead of a single-page layout dashboard coordinated by navigation pane and interactivity.

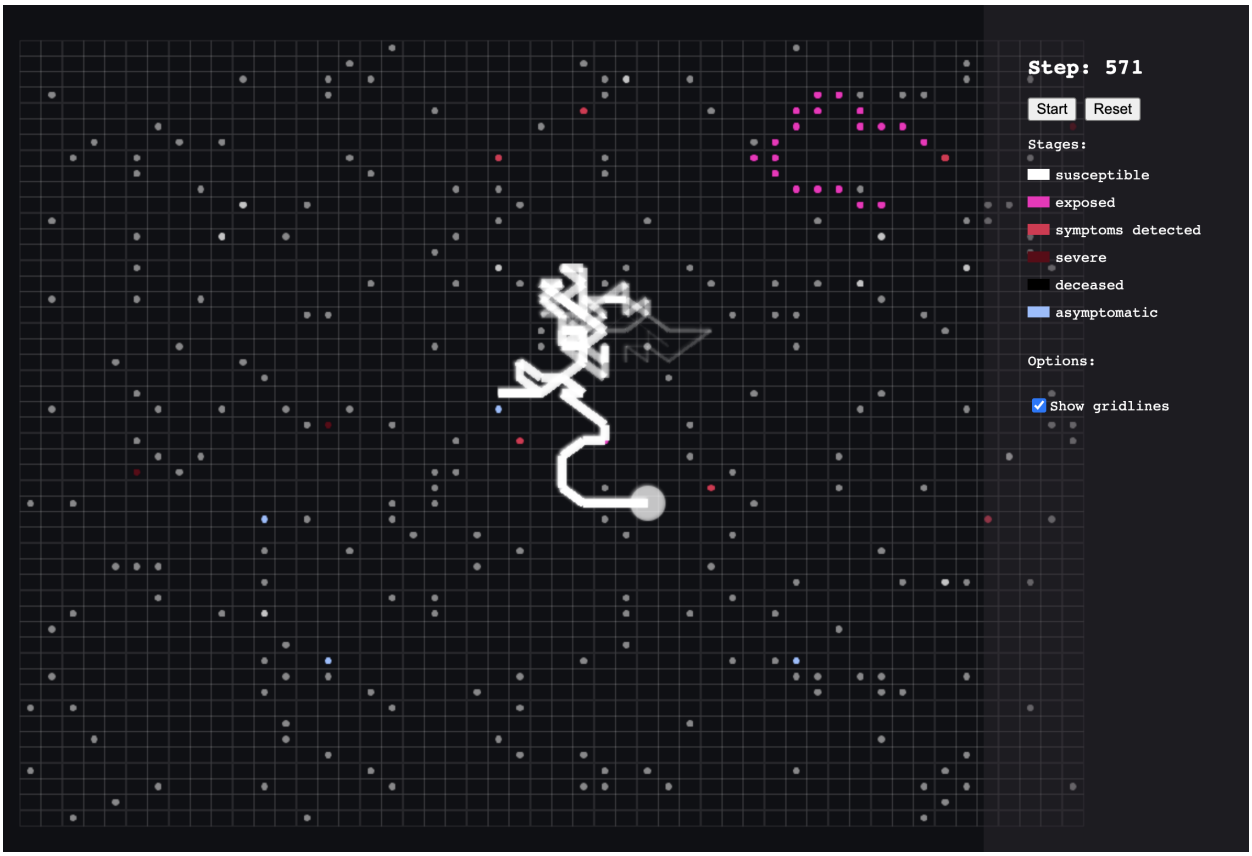


Figure 4.2: The first prototype developed in the early phase of the project. A user first starts a real-time ABM simulation server and the dashboard queries the server to obtain agent data per simulation timestep. This grid visualization resembles the design of the original visualization dashboard in Figure 2.1, but with two additional features: (1) trail of an agent of interest, (2) fast rendering. It is a milestone at the discovery stage (stage 2) through which we understood the data structure of the epidemic model and ensemble output and browsers' capabilities in rendering large volume of data. In addition, the prototype sparked fruitful inputs from the user, such as in-depth visualization ideas, color schemes, and drawbacks of existing ABM visualizations.

CHAPTER 5: RESULTS

In this section, we describe the interface design and how each component addresses the requirements in section 2. The dashboard is organized into three views, namely **ensemble view**, **simulation view**, and **agent distributions view**, addressing the needs for **S1**, **S2**, and **S3**. When the dashboard is first loaded, the top navigation bar defaults to showing the **ensemble view** and an ensemble data set pre-loaded into the server is selected. The top navigation bar in Figure 5.1.a enables navigation between the three views, the numbering on the views suggests the order of navigation for drill-down exploration, and any interactive selections made in the views are preserved during switching. Below the navigation bar is a panel for selecting any available ensembles on the server.

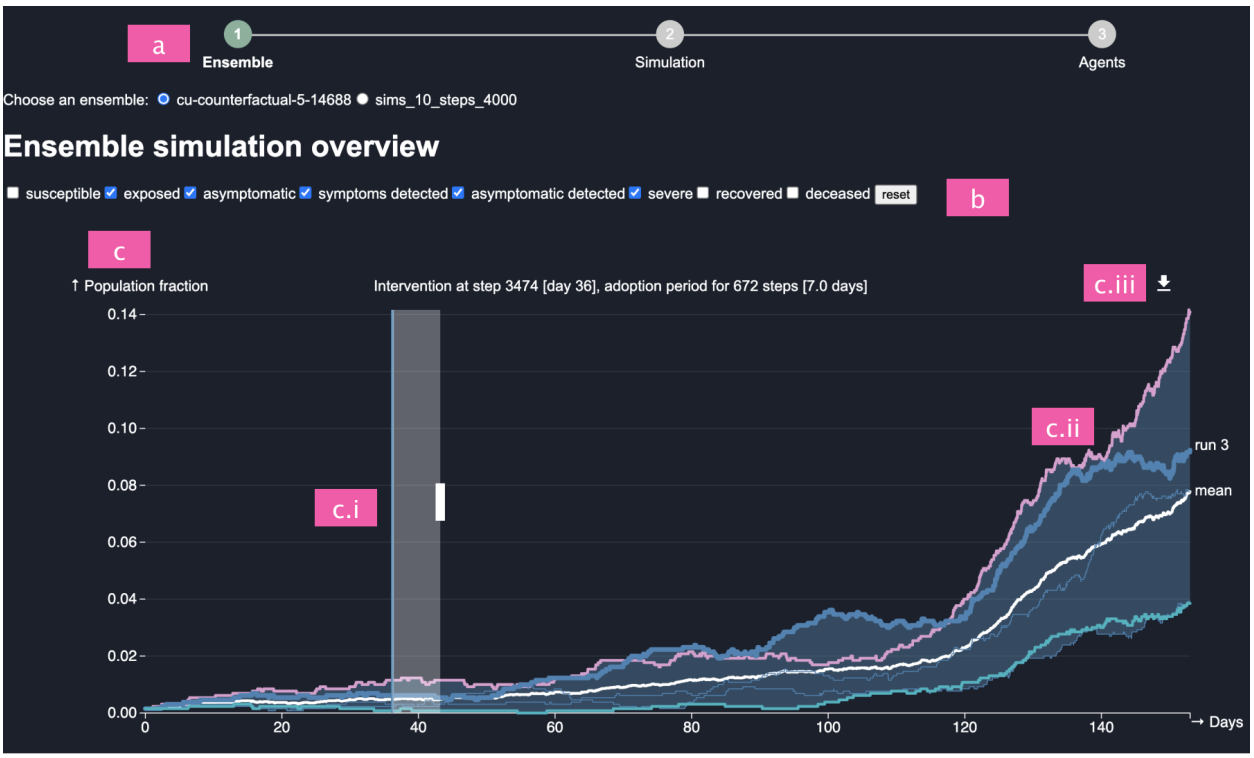


Figure 5.1: Ensemble view in MWEpi-Viz. Upon loading, simulation data undergoes aggregation to extract individual and collective agent metrics. By default, stages that map into the *infected* compartment of the SIR model are pre-selected and the resulting values requested to the Redis database. Each trend line representing an individual run is displayed in conjunction with an area defined by upper and lower runs, alongside a mean trend. All trend lines other than the mean can be selected. A user-adjustable rectangular region is used to bound moments of interest across the entire ensemble.

The **ensemble view** in Figure 5.1 is the default view when an ensemble dataset is first

loaded; it serves as an overview of the disease evolution of all the runs. It consists of a panel of checkboxes (Figure 5.1.b) for selecting stages of agents, and an ensemble chart (Figure 5.1.c) that is a spaghetti plot depicting the fraction of agents at the chosen stages for all runs across time **P1**. The spaghetti plot helps address **S3** and **V1**, as it provides an familiar graphical representation of the disease evolution over the number of days. To reduce visual clutter and highlight the points of interests, upon hover on the plot, the user sees a rectangular region (Figure 5.1.c.i) with a blue bar to the left that controls the selected step and the simulation runs which have the maximum or minimum fraction of agents with the chosen stages at the selected step are highlighted in pink and green respectively (Figure 5.1.c.ii). In addition, a white line indicates the mean across all runs to illustrate the overall trend. To facilitate sharing of the visualizations, we provide download button (Figure 5.1.c.iii) for exporting the ensemble chart as a .png file.

The ensemble chart also connects the later two views for further inspection. User can click to anchor the rectangular region in Figure 5.1.c.i and click on any runs of interest to switch to the **simulation view** of a particular run. To separate the ensemble data into pre-intervention, during intervention, and post-intervention periods for viewing their distributions in the **agent distributions view**, users can drag the white handle in Figure 5.1.c.i to the left to set the end of the intervention, and the anchored blue line is set the be the start of the intervention.

The **simulation view** provides a replay of agent behaviors in the environment for a chosen run from the **ensemble view** to satisfy **S1**, **S3**, **S4** and **V2**. The control panel in Figure 5.2.a allows users to control the replay progress with a slider and a play/pause button, and right below indicates all possible agent stages in the simulation in the order of infection stages. In the center (Figure5.2.b) is a snapshot of a simulation run at a selected time. Visual grid size is commensurate to that in the simulation, and each agent is colored by their stage. Note that multiple agents can exist in one cell at the same time. This view offers three additional features compared with traditional ABM visualizations. First, users can click on any agent to reveal the path that an agent up to the point in time on the progress slider which allows users to investigate the spatial influence an agent has and its intersection of pathways between neighbors. In particular, agent paths can be connected formally to the theory of random walks, from which several properties are derivable [49].

Second, we used 3D space to display a 2D environment for two reasons: (1) 3D space offers a vertical axis to encode more information for future work, e.g. vertical bars or stacks of agents to represent intensity of agent concentration, (2) 3D view offers a more fluid camera control to view the agents. Third, the agents will move smoothly between their locations instead of doing discrete jumps because we apply linear interpolation to the discrete locations

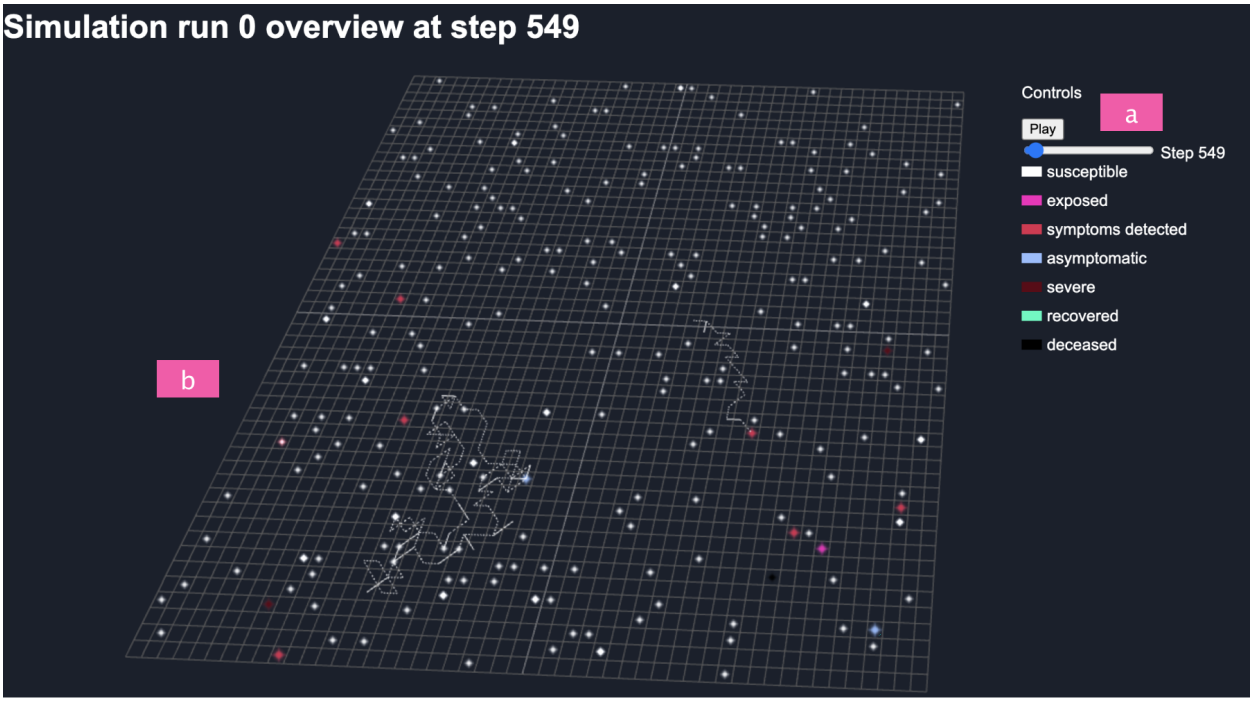


Figure 5.2: **Simulation (i.e., individual run) view** in MWEpi-Viz. A user switches to the simulation view either by clicking on the navigation bar or by selecting a run on the ensemble chart. Then, the dashboard requests agent-level data of the run from the server. A 3D visualization of agent positions across the environment is displayed, and a button controls the playback of agent dynamics per simulation step. Various controls modify the appearance of agents and their movements across space.

of the agents. Not only does it make any agent movement easier to track visually, it also separates overlapping agents visually once the agents move unless the grid is very saturated.

The **agent distributions view** in Figure 5.3 showcases the distribution of the number of days it takes for agents to undergo state transitions in the three periods of intervention set in the **ensemble view**. Though there are possible 8 stages in our example simulation, not all state transitions are possible. Hence, we designed the state transition diagram at the bottom as show in Figure 5.3.c. It displays all the possible transitions as edges: the white edges indicate the existence of a transition in ensemble while dashed edges indicate the absence of the transition. Currently, this transition diagram requires prior knowledge of the model to generate the network. The selected transition is shown on Figure 5.3.a.

For each state transition, we are interested in evaluating the effectiveness of health intervention and testing the time lag between intervention and outcome, to address **P2, P5, P6, S3, S5**, and **V3**. Each chart from Figure 5.3.b.i - 5.3.b.ii uses the pre-intervention, during intervention, and post-intervention data respectively. The y-axis shows the accumulated number of agents who experienced the transition at the said period, the x-axis shows the

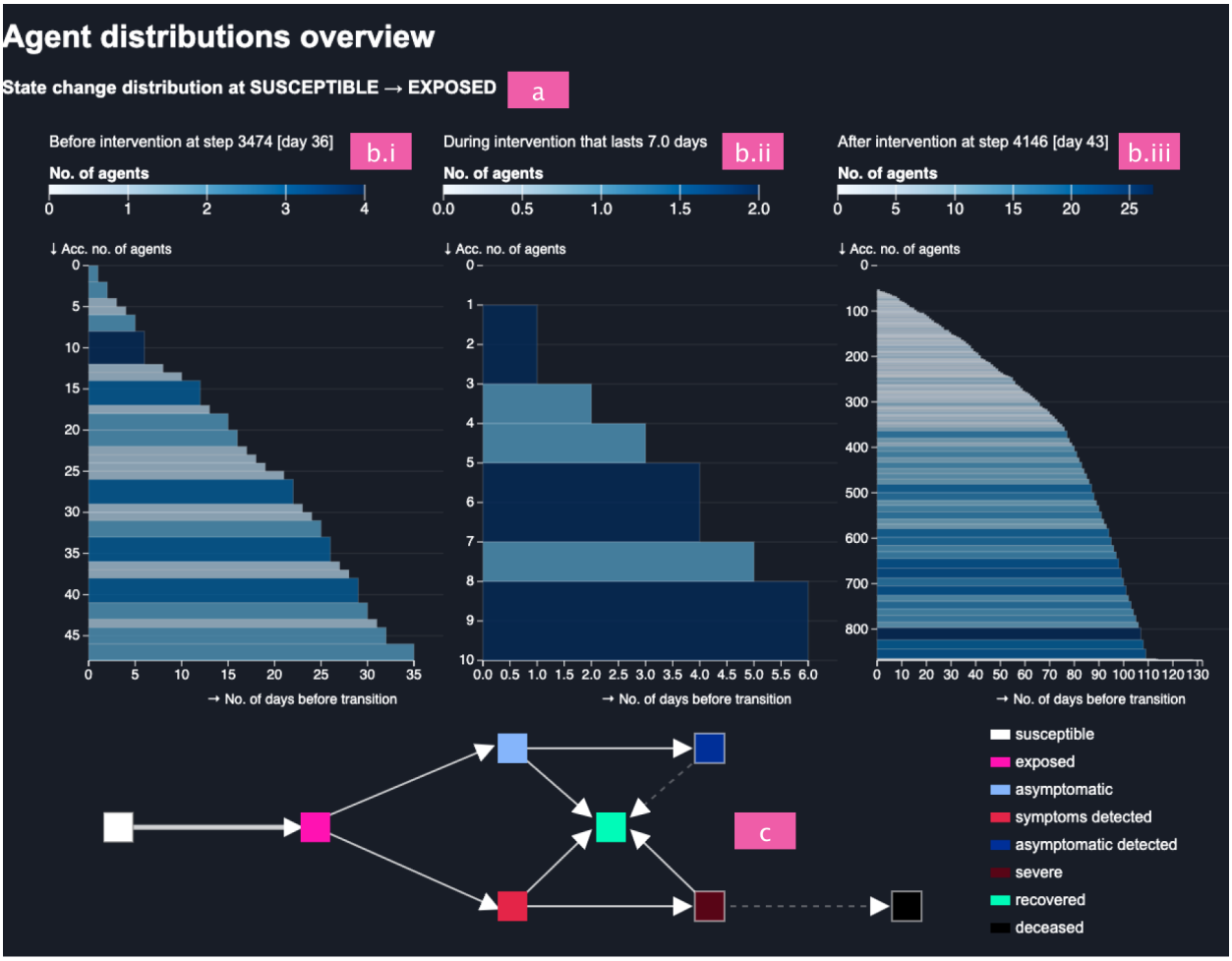
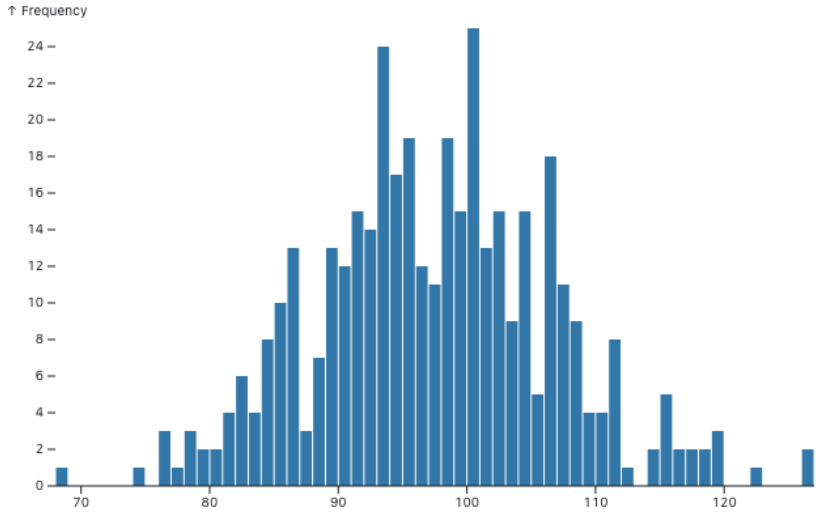


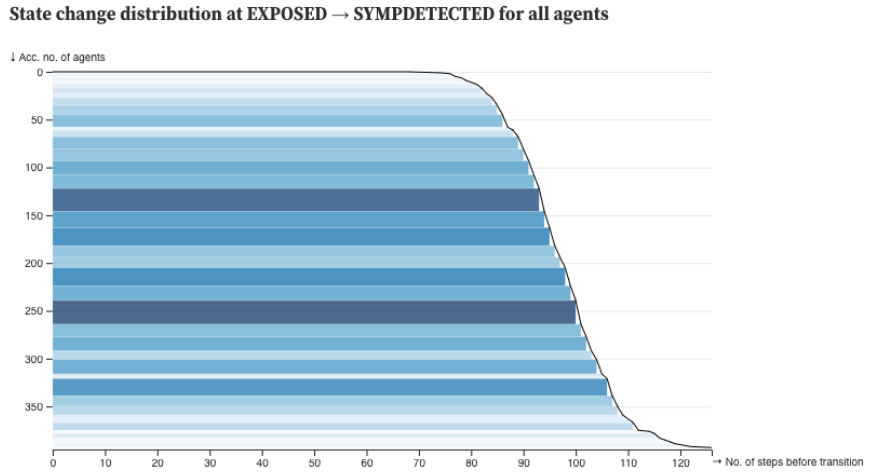
Figure 5.3: **Agent distributions view** in MWEpi-Viz. A user first selects an intervention period on the ensemble chart, then switch to the **agent distributions view** via the navigation bar. This view presents the distribution of the number of days it takes for a state transition against the accumulated number of agents in the three different phases of an intervention. By default, user views the first transition of the ABM model - Susceptible to Exposed, which represents the transition from healthy to infected. Users can examine the distributions other transitions by selecting an edge in the state transition diagram. Dashed arrows are not clickable when the corresponding state transition does not happen in the ensemble.

number of days it took an agent to undergo the transition. If there are 8 agents that took 5 days to undergo a state transition, then there is a corresponding horizontal bar of height 8 and width 5. We color the bars according to their number of agents, the more agents a bar has the darker the color is. On hovering over each bar, there is a tooltip box indicating the number of days and the number of agents. Note that we aggregate step-wise data into day-wise data, because it is more meaningful to consider the transition time from a health

intervention point of view in terms of days than granular simulation intervals.



(a) Histogram



(b) **Agent view** chart.

Figure 5.4: Histogram vs. **agent distributions view** chart. To avoid presenting a large number of histograms per each stage transition, we focused on reconceptualizing the visual portrayal of distributions across the ensemble. Conceptually, clicking each arrow may be though of as revealing the dynamical structure of the underlying distribution.

At the initial development stage, we used a classical histogram to represent the distribution for state transitions, where the x-axis is the no. of days before transition, and the y-axis is the number of agents that underwent it. However, capturing changes in the entire simulation would take a large amount of such plots, which in turn may produce information overload. Moreover, detecting differences before, during and after a given policy intervention requires organizing available information in manner that rapidly satisfies **V1** and **V3**, by visualizing

features of interest in **S2** and **S4**. After substantial analysis, giving priority to transition times in the x axis reinforces the dynamical aspect of changes due to interventions, and the width in the vertical stacking helps users reason about histogram bins in a simpler, homogeneous context of the entire ensemble simulation (Figure 5.4).

Regarding extensibility, the source code for the dashboard and the data generation scripts are released on GitHub together with our Epidemiology Workbench project [2]. Users can modify our scenario files to fine tune parameters to generate new epidemic ABM ensemble data, then feed the ensemble output to our dashboard by running our server code locally. Each of the three views and visualizations are organized into separate Svelte components in hierarchical order, users can add or remove views or visualizations accordingly.

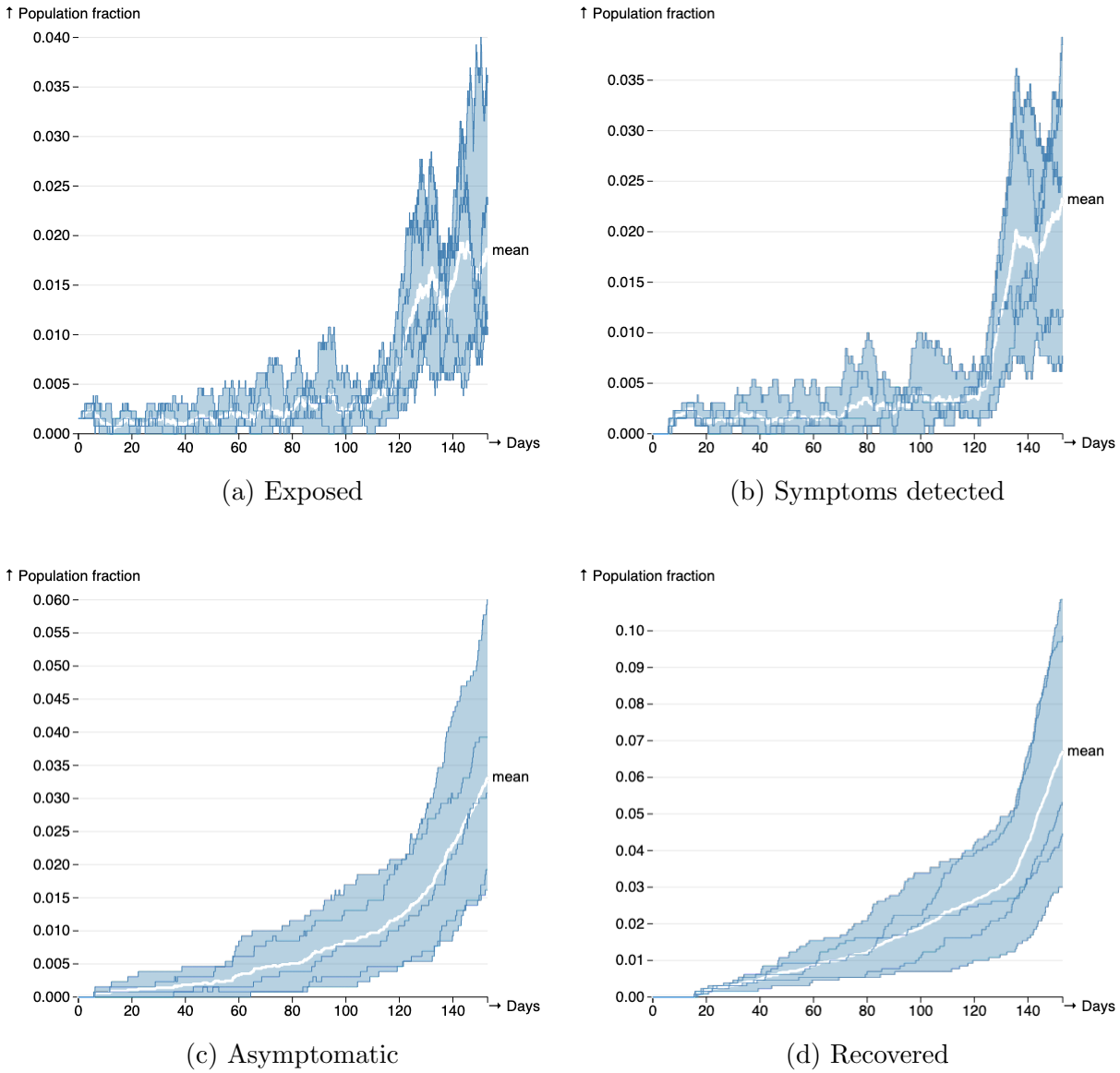
CHAPTER 6: USE CASES

In this section, we illustrate the usage MWEpi-Viz on understanding the Champaign-Urbana counterfactual ensemble dataset. In this scenario, asymptomatic agents has a 25% probability of being tested and enters isolation immediately afterwards, while symptomatic agents enters quarantine immediately once they transition from the exposed stage at the end of the virus incubation period. Compared with other scenarios in [2], the counterfactual scenario lifts the shelter-at-home policy in early August and does not perform massive testing on inbound students before the Fall semester starts. Below is a sequence of events in all of the counterfactual simulation runs:

1. April 15 (simulation day 0) - Exposure of the first representative agent, shelter-at-home policy
2. April 21 (simulation day 6) - First symptomatic representative agent
3. May 1 (simulation day 16) - Mask order from the State of Illinois
4. August 9 (simulation day 116) - Lifted shelter-at-home policy

Using agent stage selection panel on the **ensemble view**, we obtained four spaghetti plots describing the infected population fraction in various stages of infection across time (Figure 6.1) satisfying **P1**. Around day 16 when the mask order was executed, there was a drop in exposed and symptoms detected population, and the trend remained steady until day 116 – an observation that is consistent across all simulation runs. Meanwhile, asymptomatic population remained unaffected and continues to grow exponentially until the end of the simulations, exceeding the symptomatic population. This observation echoes with the finding in [2] that asymptomatic patients is a major contributor to the contagion process. At the same time, we can see a sharp growth in infected population around day 120 caused by a large volume of inbound student population.

To satisfy **S4** using the agent path visualization in **simulation view**, we selected a run with the highest number of infected population at the end and traced a symptomatic agent and an asymptomatic agent throughout their journey (Figure 6.2). Initially, both agents were in the exposed stage (Figure 6.2a). As time progressed, the top left agent turned into symptoms detected (red) and entered quarantine immediately; whereas the mid-right agent turned asymptomatic (blue) and continued doing random walk in its proximity. The path length of the blue agent is significantly longer than that of the red agent (Figure 6.2c). By the time the agents were recovered, the blue agent has travelled a much greater area than the



(a) Exposed

(b) Symptoms detected

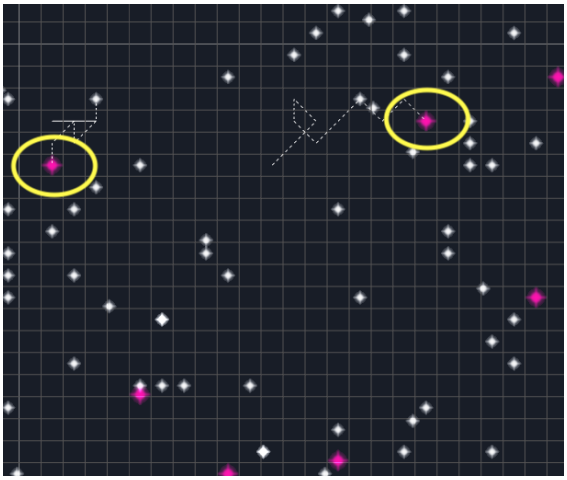
(c) Asymptomatic

(d) Recovered

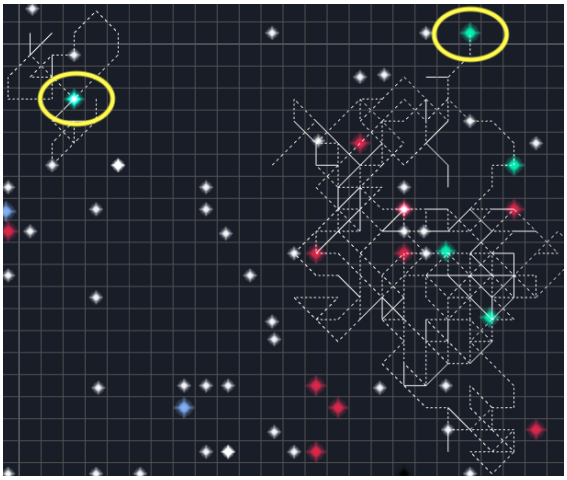
Figure 6.1: Spaghetti plots from **ensemble view** displaying the overall epidemic evolution across four infection stages using the Champaign-Urbana counterfactual scenario.

red agent, and along the path where the blue agent has travelled there were more infected agents (Figure 6.2b). This visual story could help communicate the importance of massive testing to limit the spatial influence of asymptomatic patients.

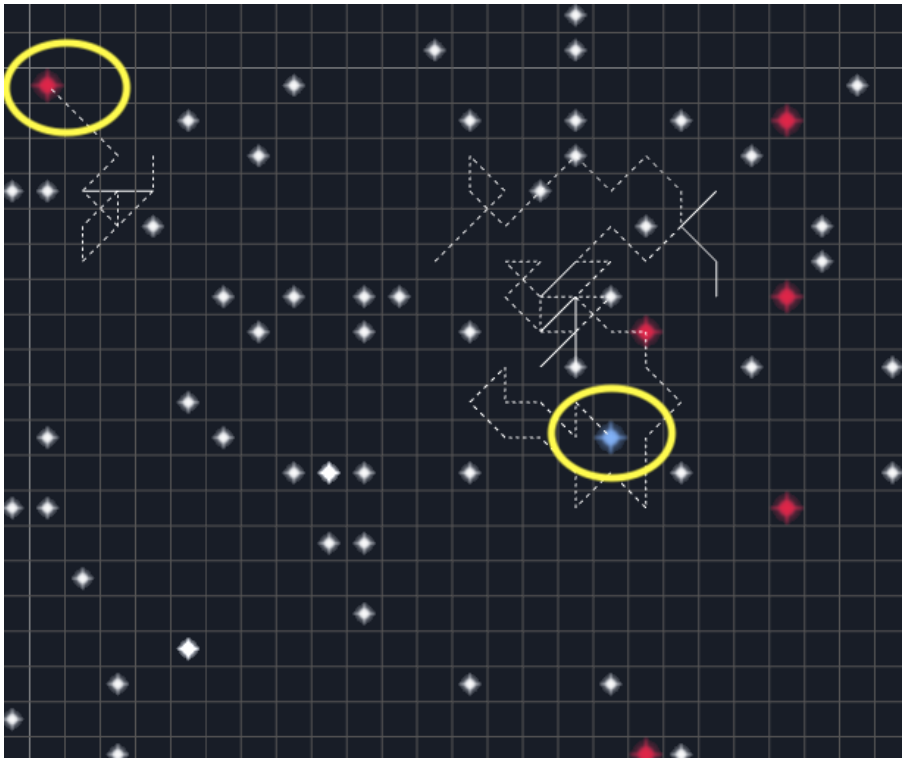
Finally, when examining the **agent distributions view** charts we should avoid doing direct comparisons between the before, during, and after intervention charts because these three charts do not share the same range in their axes. The left chart (pre-intervention) on Figure 6.3a indicates that before the mask mandate the transition from susceptible to exposed was likely to happen between 6 to 12 days since the start of the simulation, because



(a) Exposed



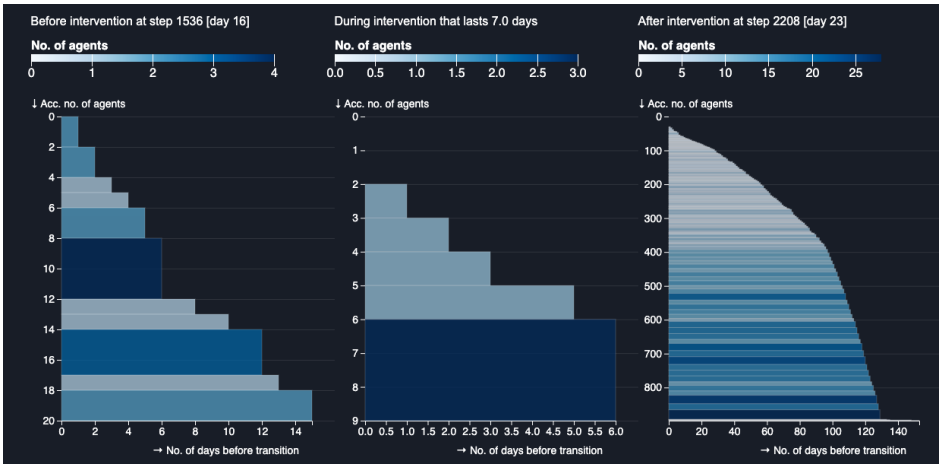
(b) Recovered



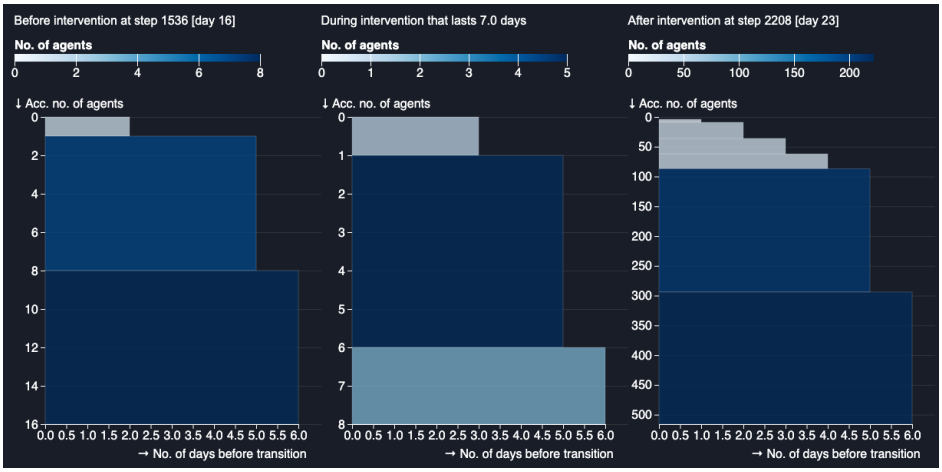
(c) Symptomatic vs. Asymptomatic

Figure 6.2: Visual comparison of reachability between an symptomatic and an asymptomatic agent.

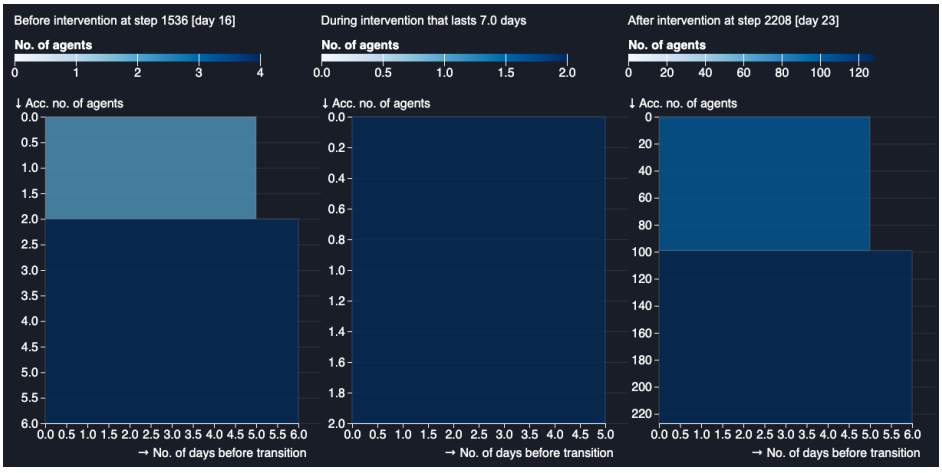
there are the dark rectangular bars concentrated at the lower half of the chart. Meanwhile after the mask mandate time lag period (Figure 6.3a, post-intervention) the lower half region is darker than the upper half, indicating that most agents who experienced the state transition around 3 months after the mask mandate. This observation is consistent with the fact that no massive testing was conducted towards the end of summer break where a large number of inbound travellers arrived. As for Figure 6.3b and Figure 6.3c, the symptomatic agents had a small likelihood of having a shorter incubation period (fewer than 4 days), but asymptomatic agents experienced the transition between 5 to 6 days regardless of the mask mandate well in line with the clinical expectations of the disease.



(a) Susceptible to Exposed.



(b) Exposed to Symptoms Detected.



(c) Exposed to Asymptomatic.

Figure 6.3: State transition charts of the Champaign-Urbana counterfactual ensemble dataset in **agent distributions view** with the intervention day set at day 16 and intervention adoption duration set at 7 days.

CHAPTER 7: CONCLUSION

In this article, we demonstrated the design and implementation of MWEpi-Viz, a visualization platform for ensemble simulations in computational epidemiology. In doing so, we identified a series of requirements from the scientific, public health and decision making perspectives connected to a central concern: mapping the behavior of each agent to population-level trends used to devise interventions. Existing visualization tools tend to focus only on one of these perspectives due to the complexity of the underlying individual and collective dynamics involved. By taking into account the probabilistic nature of ensemble simulations for infectious diseases, we were able to produce a visualization that connects events at the macro level to the distributions resulting a the agent level; to solidify this connection, we also allow users to explore individual runs visualized within the entire ensemble, and to select time slices of it to explore the distributions before, during and after an policy intervention. Each run is visualized by re-enacting agent paths using simulation data, with a set of controls intended to increase salience of of various attributes as needed. We summarize how MWEpi-Viz address the scientific, public health, and visualization requirements in Table 7.1. To the best of our knowledge, this is the first tool to achieve an integrated view of the entire chain of events in the phenomenology of a simulated epidemic.

Requirement	Dashboard	Ensemble view	Simulation view	Agent view
P1		✓	✓	✓
P2				✓
P3				✓
P4				
P5				✓
P6				✓
S1	✓		✓	
S2	✓			
S3	✓	✓	✓	✓
S4			✓	
S5				✓
V1		✓		
V2			✓	
V3				✓

Table 7.1: A summary of dashboard requirements and their corresponding visualization components. Since **P4** requires a comparison between scenarios, it is out of the scope of MWEpi-Viz.

In the future, we plan to improve the scalability our dashboard in order to sustain a long-term development of our project. Currently, our system architecture suffers from a data management issue. For example, our server expects a plain text ensemble data file recording the positions and states of all agents at each step, but the volume of plain text data a local server can process in memory is very limited and any Redis database can at most store 4GB of data, making the system inadequate to process any larger ensemble. We plan to build a more robust and memory-efficient data processing pipeline to store agent-level data in a database instead of plain text. We also plan to extend our dashboard to accommodate more model variants and auto-generate state transition diagrams by standardizing the data format to record epidemiological ABM simulations. To evaluate our system objectively, we plan conduct user studies with active users of Python Mesa and computational epidemiologists.

REFERENCES

[1] N. Ferguson, D. Laydon, G. Nedjati Gilani, N. Imai, K. Ainslie, M. Baguelin, S. Bhatia, A. Boonyasiri, Z. Cucunuba Perez, G. Cuomo-Dannenburg et al., "Report 9: Impact of non-pharmaceutical interventions (npis) to reduce covid19 mortality and healthcare demand," 2020.

[2] S. Núñez-Corrales and E. Jakobsson, "The epidemiology workbench: A tool for communities to strategize in response to covid-19 and other infectious diseases," *MedRxiv*, pp. 2020–07, 2020.

[3] D. Kornhauser, U. Wilensky, and W. Rand, "Design guidelines for agent based model visualization," *Journal of Artificial Societies and Social Simulation*, vol. 12, no. 2, p. 1, 2009.

[4] A. Dorin and N. Geard, "The practice of agent-based model visualization," *Artificial life*, vol. 20, no. 2, pp. 271–289, 2014.

[5] A. Grignard and A. Drogoul, "Agent-based visualization: A real-time visualization tool applied both to data and simulation outputs." in *AAAI Workshops*, 2017.

[6] Y. Zhang, Y. Sun, J. D. Gaggiano, N. Kumar, C. Andris, and A. G. Parker, "Visualization design practices in a crisis: Behind the scenes with covid-19 dashboard creators," *IEEE Transactions on Visualization and Computer Graphics*, vol. 29, no. 1, pp. 1037–1047, 2022.

[7] C. Doogan, W. Buntine, H. Linger, and S. Brunt, "Public perceptions and attitudes toward covid-19 nonpharmaceutical interventions across six countries: a topic modeling analysis of twitter data," *Journal of medical Internet research*, vol. 22, no. 9, p. e21419, 2020.

[8] W. Jennings, G. Stoker, V. Valgarsson, D. Devine, and J. Gaskell, "How trust, mistrust and distrust shape the governance of the covid-19 crisis," *Journal of European Public Policy*, vol. 28, no. 8, pp. 1174–1196, 2021.

[9] J. Kazil, D. Masad, and A. Crooks, "Utilizing python for agent-based modeling: The mesa framework," in *Social, Cultural, and Behavioral Modeling*, R. Thomson, H. Bisgin, C. Dancy, A. Hyder, and M. Hussain, Eds. Cham: Springer International Publishing, 2020, pp. 308–317.

[10] J. M. Epstein, "Agent-based computational models and generative social science," *Complexity*, vol. 4, no. 5, pp. 41–60, 1999.

[11] H. H. Weiss, "The sir model and the foundations of public health," *Materials matematics*, pp. 0001–17, 2013.

[12] B. Armbruster and E. Beck, "Elementary proof of convergence to the mean-field model for the sir process," *Journal of mathematical biology*, vol. 75, pp. 327–339, 2017.

[13] C. Durón and A. Farrell, "A mean-field approximation of sir epidemics on an erdös–rényi network model," *Bulletin of Mathematical Biology*, vol. 84, no. 7, p. 70, 2022.

[14] M. Capanna, "Critical asymptotic behaviour in the sir model," *arXiv preprint arXiv:1708.03905*, 2017.

[15] E. Fausti and A. Sojmark, "An interacting particle system for the front of an epidemic advancing through a susceptible population," *arXiv preprint arXiv:2210.09286*, 2022.

[16] S. Cho, "Mean-field game analysis of sir model with social distancing," *arXiv preprint arXiv:2005.06758*, 2020.

[17] J. Doncel, N. Gast, and B. Gaujal, "A mean field game analysis of sir dynamics with vaccination," *Probability in the Engineering and Informational Sciences*, vol. 36, no. 2, pp. 482–499, 2022.

[18] T. de Camino-Beck, "A modified seir model with confinement and lockdown of covid-19 for costa rica," *medRxiv*, pp. 2020–05, 2020.

[19] Y. Maki and H. Hirose, "Infectious disease spread analysis using stochastic differential equations for sir model," in *2013 4th International Conference on Intelligent Systems, Modelling and Simulation*. IEEE, 2013, pp. 152–156.

[20] E. Dong, H. Du, and L. Gardner, "An interactive web-based dashboard to track covid-19 in real time," *The Lancet infectious diseases*, vol. 20, no. 5, pp. 533–534, 2020.

[21] R. A. Leite, V. Schetinger, D. Ceneda, B. Henz, and S. Miksch, "Covis: supporting temporal visual analysis of covid-19 events usable in data-driven journalism," in *2020 IEEE Visualization Conference (VIS)*. IEEE, 2020, pp. 56–60.

[22] S. Afzal, S. Ghani, H. C. Jenkins-Smith, D. S. Ebert, M. Hadwiger, and I. Hoteit, "A visual analytics based decision making environment for covid-19 modeling and visualization," in *2020 IEEE Visualization Conference (VIS)*. IEEE, 2020, pp. 86–90.

[23] I. Mahmood, M. Jahan, D. Groen, A. Javed, and F. Shafait, "An agent-based simulation of the spread of dengue fever," in *Computational Science–ICCS 2020: 20th International Conference, Amsterdam, The Netherlands, June 3–5, 2020, Proceedings, Part III 20*. Springer, 2020, pp. 103–117.

[24] P. Taillandier, B. Gaudou, A. Grignard, Q.-N. Huynh, N. Marilleau, P. Caillou, D. Philippon, and A. Drogoul, "Building, composing and experimenting complex spatial models with the gama platform," *GeoInformatica*, vol. 23, pp. 299–322, 2019.

[25] A. T. Wilson and K. C. Potter, "Toward visual analysis of ensemble data sets," in *Proceedings of the 2009 Workshop on Ultrascale Visualization*, 2009, pp. 48–53.

[26] R. T. Whitaker, M. Mirzargar, and R. M. Kirby, "Contour boxplots: A method for characterizing uncertainty in feature sets from simulation ensembles," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 12, pp. 2713–2722, 2013.

[27] M. Mirzargar, R. T. Whitaker, and R. M. Kirby, "Curve boxplot: Generalization of boxplot for ensembles of curves," *IEEE transactions on visualization and computer graphics*, vol. 20, no. 12, pp. 2654–2663, 2014.

[28] L. Hao, C. G. Healey, and S. A. Bass, "Effective visualization of temporal ensembles," *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 1, pp. 787–796, 2015.

[29] K. Matkovic, D. Gracanin, B. Klarin, and H. Hauser, "Interactive visual analysis of complex scientific data as families of data surfaces," *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, no. 6, pp. 1351–1358, 2009.

[30] H. Piringer, S. Pajer, W. Berger, and H. Teichmann, "Comparative visual analysis of 2d function ensembles," in *Computer Graphics Forum*, vol. 31, no. 3pt3. Wiley Online Library, 2012, pp. 1195–1204.

[31] T. Nocke, M. Flechsig, and U. Bohm, "Visual exploration and evaluation of climate-related simulation data," in *2007 Winter Simulation Conference*. IEEE, 2007, pp. 703–711.

[32] B. Ma and A. Entezari, "An interactive framework for visualization of weather forecast ensembles," *IEEE transactions on visualization and computer graphics*, vol. 25, no. 1, pp. 1091–1101, 2018.

[33] P. Köthur, M. Sips, H. Dobslaw, and D. Dransch, "Visual analytics for comparison of ocean model output with reference data: Detecting and analyzing geophysical processes using clustering ensembles," *IEEE transactions on visualization and computer graphics*, vol. 20, no. 12, pp. 1893–1902, 2014.

[34] H. Karen and J. Sandra, "Contextual inquiry: A participatory technique for system design," in *Participatory design*. CRC Press, 2017, pp. 177–210.

[35] C. Spinuzzi, "The methodology of participatory design," *Technical communication*, vol. 52, no. 2, pp. 163–174, 2005.

[36] N. A. Haber, E. Clarke-Deelder, J. A. Salomon, A. Feller, and E. A. Stuart, "Covid-19 policy impact evaluation: A guide to common design issues," 2021.

[37] S. B. Polak, I. C. Van Gool, D. Cohen, J. H. von der Thüsen, and J. van Paassen, "A systematic review of pathological findings in covid-19: a pathophysiological timeline and possible mechanisms of disease progression," *Modern Pathology*, vol. 33, no. 11, pp. 2128–2138, 2020.

[38] J. Li and P. J. Giabbanelli, "Identifying synergistic interventions to address covid-19 using a large scale agent-based model," in *Computational Science–ICCS 2021: 21st International Conference, Krakow, Poland, June 16–18, 2021, Proceedings, Part III*. Springer, 2021, pp. 655–662.

[39] J. Nielsen, *Usability engineering*. Morgan Kaufmann, 1994.

[40] L. Battle, P. Eichmann, M. Angelini, T. Catarci, G. Santucci, Y. Zheng, C. Binnig, J.-D. Fekete, and D. Moritz, "Database benchmarking for supporting real-time interactive querying of large data," in *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*, 2020, pp. 1571–1587.

[41] J. Howard, A. Huang, Z. Li, Z. Tufekci, V. Zdimal, H.-M. Van Der Westhuizen, A. Von Delft, A. Price, L. Fridman, L.-H. Tang et al., "An evidence review of face masks against covid-19," *Proceedings of the National Academy of Sciences*, vol. 118, no. 4, p. e2014564118, 2021.

[42] Armin Ronacher, "Flask." [Online]. Available: https://flask.palletsprojects.com/

[43] Wes McKinney, "Pandas." [Online]. Available: https://pandas.pydata.org/

[44] Rich Harris, "Svelte." [Online]. Available: https://svelte.dev/

[45] M. Bostock, V. Ogievetsky, and J. Heer, "D3: Data-driven documents," *IEEE Trans. Visualization & Comp. Graphics (Proc. InfoVis)*, 2011. [Online]. Available: http://idl.cs.washington.edu/papers/d3

[46] THREE.js Authors, "Three.js." [Online]. Available: https://threejs.org/

[47] Mat Groves, "Pixi.js." [Online]. Available: https://pixijs.com/

[48] B. Bach, E. Freeman, A. Abdul-Rahman, C. Turkay, S. Khan, Y. Fan, and M. Chen, "Dashboard design patterns," *IEEE Transactions on Visualization and Computer Graphics*, vol. 29, no. 1, pp. 342–352, 2022.

[49] C. Pu, S. Li, and J. Yang, "Epidemic spreading driven by biased random walks," *Physica A: Statistical Mechanics and its Applications*, vol. 432, pp. 230–239, 2015.