Private Agent-Based Modeling

Ayush Chopra
Massachusetts Institute of Technology
Cambridge, USA
ayushc@mit.edu

Arnau Quera-Bofarull
University of Oxford
Oxford, UK
arnau.quera-bofarull@cs.ox.ac.uk

Nurullah Giray-Kuru
Massachusetts Institute of Technology
Cambridge, USA
ngkuru@mit.edu

Michael Wooldridge
University of Oxford
Oxford, UK
michael.wooldridge@cs.ox.ac.uk

Ramesh Raskar
Massachusetts Institute of Technology
Cambridge, USA
raskar@media.mit.edu

ABSTRACT

The practical utility of agent-based models in decision-making relies on their capacity to accurately replicate populations while seamlessly integrating real-world data streams. Yet, the incorporation of such data poses significant challenges due to privacy concerns. To address this issue, we introduce a paradigm for private agent-based modeling wherein the simulation, calibration, and analysis of agent-based models can be achieved without centralizing the agents' attributes or interactions. The key insight is to leverage techniques from secure multi-party computation to design protocols for decentralized computation in agent-based models. This ensures the confidentiality of the simulated agents without compromising on simulation accuracy. We showcase our protocols on a case study with an epidemiological simulation comprising over 150,000 agents. We believe this is a critical step towards deploying agent-based models to real-world applications.

KEYWORDS

Differentiable Agent-based Modeling; Privacy; Multi-party Computation; Automatic Differentiation; Epidemiology

ACM Reference Format:
Ayush Chopra, Arnau Quera-Bofarull, Nurullah Giray-Kuru, Michael Wooldridge, and Ramesh Raskar. 2024. Private Agent-Based Modeling. In *Proc. of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024), Auckland, New Zealand, May 6 – 10, 2024*, IFAAMAS, 10 pages.

1 INTRODUCTION

Agent-based modeling (ABM) is a bottom-up simulation technique wherein a system is modeled through the interaction of autonomous decision-making entities referred to as agents, which may represent individuals, companies, or other decision-making entities. Due to their granular approach, ABMs are a promising tool for real-world decision-making and policy design and constitute an active field of research across economics [6, 12, 15, 34], biology [28, 44], and epidemiology [7, 33, 37, 54]. The wider adoption of ABMs, however, is hindered by (1) the need for microdata to generate the underlying agent population and (2) the often substantial computational resources required to run, calibrate, and analyze an ABM. Recently, there has been significant progress towards developing new design patterns for ABMs, which exploit tensorization [17, 18] and differentiability [4, 19] of simulators. This has alleviated the computational burdens associated with ABM simulation [18], calibration [19, 52], and analysis [51] by granting access to techniques such as GPU computing and differentiable programming, allowing ABMs to scale to million-size populations [11, 53].

Unfortunately, improvements in the computational efficiency of ABMs is of little value if the quality of the underlying population microdata is poor. Currently, prevalent approaches involve the construction of synthetic populations designed to align with a predefined set of summary statistics derived from real-world observations. For instance, in epidemiological ABMs, the population is crafted to replicate summary statistics obtained from census data [7, 13, 16, 46, 47]. However, it is essential to recognize that the limited granularity of census data arises primarily from privacy considerations rather than the actual scarcity of available data. As ABMs continue to scale towards one-to-one representations of real-world systems, there remains a fundamental limitation in their modeling potential as long as privacy guarantees are not in place. Previous attempts to augment ABM data with additional information, such as mobility or health data, have resulted in data leaks that exposed agents' personal information [1, 22, 36]. These incidents underscore the need for a decentralized approach to ABM, where each agent's sensitive information is kept confidential throughout the modeling process.

Motivated by this, we introduce a new paradigm for agent-based simulation that ensures the confidentiality of each agent's sensitive information. Leveraging techniques drawn from secure multi-party computation [40], we develop privacy-preserving protocols for the simulation, calibration, and analysis of ABMs. These protocols offer robust security guarantees to agents while preserving the ability of ABMs to model complex systems effectively. Moreover, our methodology enables secure ABMs to take advantage of differentiable programming, allowing them to be integrated into machine learning pipelines, further boosting their modeling capabilities. We demonstrate the capabilities of this new methodology by running a case study in the city of Oxford, UK. We showcase how our approach can provide the same level of insight and analysis as traditional ABMs while guaranteeing the agents' privacy.

Proc. of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024), N. Alechina, V. Dignum, M. Dastani, J.S. Sichman (eds.), May 6 – 10, 2024, Auckland, New Zealand. © 2024 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org).

In summary, this work constitutes, to the best of our knowledge, the first framework for privacy-preserving ABMs. The framework supports their simulation, calibration, and analysis. We hope this development will pave the way for the secure and practical utilization of ABMs as valuable tools for policy-making in real-world settings.

2 AGENT-BASED MODELS

In this section, we formalize the processes of simulation, calibration, and analysis of ABMs. In doing so, we lay the foundations for the introduction of privacy-preserving protocols in Section 4.

2.1 Simulation of ABMs

Consider an ABM with N agents $A = \{1, 2, \ldots, N\}$. We denote by $\mathbf{z}_i(t)$ the state of agent i at time t, which encapsulates both fixed and time-evolving properties of the simulation agents. For example, $\mathbf{z}$ might represent the age and disease status of human agents in epidemiological models, or the account balance of firms in a financial auction model. As the simulation proceeds, an agent i updates their state $\mathbf{z}_i(t)$ by interacting with their neighbors $\mathcal{N}_i(t)$ and the environment $\mathcal{E}(t)$. We assume that the interaction of agents with their neighbors can be conceived as message passing on a graph $\mathcal{G} = (V, E)$, where the vertices V of the graph correspond to the agents, edges $e_{ij} \in E$ connect neighboring agents, and interactions are represented as messages $M_{ij}(t) = M(\mathbf{z}_j(t), e_{ij}(t), \boldsymbol{\theta}, t)$, where $\boldsymbol{\theta}$ are the ABM structural parameters. This is the case for a diverse class of social and biological contagion models [23]. For example, $M_{ij}(t)$ may represent the transmission of infection from agent j to agent i, which may depend on the susceptibility of agent i ($\mathbf{z}_i$), the infectivity of j ($\mathbf{z}_j$), the properties of the virus ($\boldsymbol{\theta}$), and the nature of the interaction (e_{ij}) [19, 33, 54]; or transmission of information from agent i to j which depends on the opinion of agent i ($\mathbf{z}_i$), and the assimilation and rejection thresholds of agent j ($\mathbf{z}_j$) [21, 21, 25]. Thus, at step t, each agent updates its state as follows:

$$\mathbf{z}_i(t+1) = f\left(\mathbf{z}_i(t), \bigoplus_{j \in \mathcal{N}_i(t)} M_{ij}(t), \boldsymbol{\theta}\right), \tag{1}$$

where $\oplus$ denotes an aggregation function over all received messages. The specific form of f can be tailored to capture the unique dynamics of the system under investigation. For instance, the diversity of contagion models can be encapsulated by different functional forms of f [23].

During the simulation of an ABM, a central agent (the modeler) collects a time-series of aggregate statistics over agent states, $\mathbf{x}_t = h(\{\mathbf{z}_i(t) \mid i \in A\})$, which can be used to compare the output of the model to ground-truth data. For instance, in epidemiological ABMs, h may correspond to counting the number of infected agents so that $\{\mathbf{x}_t\}_t$ is a time series of daily infections.

As we can see, both Equation (1) and the collection of the summary statistics require agents to communicate their state to other agents. In the following sections, we introduce a methodology that enables these operations to take place while preserving the privacy of individual agents.

2.2 Calibration of ABMs

Calibration refers to the process of tuning the set of structural parameters $\boldsymbol{\theta}$ so that ABM outputs $\mathbf{x}$ are compatible with given observational data $\mathbf{y}$. In epidemiological ABMs, for instance, this entails determining values for parameters like the reproduction number R_0 and mortality rates to align with the observed daily infection or mortality data.

It is important to recognize that due to the stochasticity of the model and its partial observability, multiple sets of parameter values $\boldsymbol{\theta}$ may be compatible with the observed data $\mathbf{y}$. Consequently, it becomes essential to have an accurate estimate of uncertainty around the calibrated parameters. Likewise, it is also important to be able to incorporate expert knowledge that may indicate a preference for certain regions of the parameter space over others into the calibration procedure. Both of these requirements can be met by adopting a Bayesian framework, wherein parameter inference corresponds to determining the posterior distribution over the parameters, $\pi(\boldsymbol{\theta} \mid \mathbf{y})$ using Bayes' theorem,

$$\pi(\boldsymbol{\theta} \mid \mathbf{y}) = \frac{p(\mathbf{y} \mid \boldsymbol{\theta})\, \pi(\boldsymbol{\theta})}{p(\mathbf{y})}, \tag{2}$$

where $\pi(\boldsymbol{\theta})$ is the prior distribution, $p(\mathbf{y} \mid \boldsymbol{\theta})$ is the likelihood function and $p(\mathbf{y})$ is the marginal likelihood. For ABMs, the likelihood function is typically intractable; thus, we need to consider likelihood-free calibration algorithms.

While many Bayesian calibration methods exist for ABM (see, e.g., [24, 30, 49]), we focus on methods that exploit the differentiability of the ABM. Differentiable ABMs [3, 19] are ABMs implemented in frameworks that allow computing the gradient of the ABM output respect to the structural parameters, $\nabla_{\boldsymbol{\theta}}\mathbf{x}$, in an efficient way using techniques like automatic differentiation [8]. Gradient-assisted calibration methods that take advantage of the differentiability of the ABM have been shown to be more efficient, scaling to larger parameter spaces than their gradient-free counterparts [19, 53], without requiring the use of surrogate models. In most applications, the ABM output $\mathbf{x}$ is an aggregate over agent states through time, as is the case in epidemiology, where we are typically interested in infection curves. The gradient can then be computed as an aggregation

$$\nabla_{\boldsymbol{\theta}} x_t = \bigoplus_{i \in A} \nabla_{\boldsymbol{\theta}}(h(\mathbf{z}_i(t))). \tag{3}$$

A suitable likelihood-free approach to conduct gradient-assisted Bayesian calibration in ABMs is generalized variational inference [39] (GVI). In GVI, we employ a variational approach to target the generalized posterior,

$$\pi_w(\boldsymbol{\theta} \mid \mathbf{y}) \propto e^{-w \cdot \ell(\mathbf{y}, \boldsymbol{\theta})} \pi(\boldsymbol{\theta}), \tag{4}$$

where $\ell(\mathbf{y}, \boldsymbol{\theta})$ is a loss function capturing the compatibility between the observed data $\mathbf{y}$ and the behaviour of the ABM at parameter vector $\boldsymbol{\theta}$, and $w > 0$ is a hyperparameter. This posterior can then be approximated by finding a distribution q in some variational family $\boldsymbol{Q}$ of distributions that minimises the Kullback-Liebler (KL) divergence to the generalised posterior given in Equation (4),

$$q^* = \arg\min_{\boldsymbol{\phi}} \mathcal{L}(\boldsymbol{\phi}), \tag{5}$$

where

$$\mathcal{L}(\boldsymbol{\phi}) = \mathbb{E}_{q_{\boldsymbol{\phi}}}\left[\ell(\mathbf{x}(\boldsymbol{\theta}), \mathbf{y})\right] + w\ \mathrm{KL}(q_{\boldsymbol{\phi}} \mid\mid \pi(\boldsymbol{\theta})). \tag{6}$$

This optimization problem can be tackled by estimating the gradient $\nabla_{\boldsymbol{\phi}}\mathcal{L}(\boldsymbol{\phi})$ and using an appropriate optimization technique such as stochastic gradient descent to minimize $\mathcal{L}(\boldsymbol{\phi})$. Given the differentiablity of the ABM, the gradient of the expected loss can be computed through a pathwise Monte Carlo gradient estimator via the reparameterization trick (see [43] for a comprehensive review),

$$\nabla_{\boldsymbol{\phi}}\mathbb{E}_{q_{\boldsymbol{\phi}}}\left[\ell(\mathbf{x}(\boldsymbol{\theta}), \mathbf{y})\right] \approx \frac{1}{N}\sum_{i=1}^{N} \nabla_{\boldsymbol{\phi}}\ell(x(\boldsymbol{\theta}_{\boldsymbol{\phi}}(u^{(i)}), \mathbf{y}), \tag{7}$$

where $u^{(i)} \sim p(u)$ is a sample from the base density $p(u)$ and $\boldsymbol{\theta}_{\phi}(u^{(j)})$ is the transformed sample from the candidate posterior q_{ϕ}. Finally, the gradient of the loss can be related to Equation (3) via the chain-rule,

$$\nabla_{\boldsymbol{\phi}}\ell(\mathbf{x}(\boldsymbol{\theta}, \mathbf{y})) = \nabla_{\boldsymbol{\phi}}\boldsymbol{\theta}_{\boldsymbol{\phi}} \cdot \nabla_{\boldsymbol{\theta}}\mathbf{x}. \tag{8}$$

In practice, the variational family $\boldsymbol{Q}$ will be parameterized by a deep neural network (i.e, a normalizing flow) that is trained using the gradient in Equation (7).

2.3 Analysis of ABMs

One of the core strengths of ABMs is their granularity, which enables ABM to address research questions that are beyond reach for coarser methodologies. For instance, epidemiological ABMs can help analyze the observed disparities of COVID-19 infections among demographic groups [38, 42, 51], plan effective vaccination rollout plans [59], or assess the role of latent transmission for a certain network structure [41]. All these studies require examination of the distribution of state values of the agents and their responsiveness to changes in the model's structural parameters. Specifically, we are interested in collecting a series of summary statistics over the agent's population, $\xi_k = h_k(\{\mathbf{z}_i(t) \mid i \in A\})$, and their sensitivity, $\nabla_{\boldsymbol{\theta}}\xi_k$. For instance, in an epidemiological ABM, we might investigate the distribution of infections among different age groups and assess how these infections react to variations in the reproduction number R_0. Sensitivity analysis can be a challenging task due to the high computational cost of running large ABMs and the high dimensionality of the parameter space. As with the calibration process, differentiable ABMs may help mitigate this computational bottleneck [51].

3 CHARACTERIZING PRIVACY

First, we formalize a threat model to setup constraints for a privacy-preserving solution. Then, we provide background on secure multi-party computation and describe the GMW protocol which we use to design algorithms for privacy-preserving simulation, calibration and analysis of agent-based models.

3.1 Threat Model

We assume an honest-but-curious (a.k.a. semi-honest) attacker [32] which aims to learn private information about participating agents without altering the protocol. This private information is included in an agent's state $\mathbf{z}_j(t)$, interaction trace $\{\mathcal{N}_i(t) \mid \forall t\}$, and neighborhood messages $\{M_{ij}(t) \mid i \in A, j \in \mathcal{N}(i)\}$. For instance, in epidemiological models, this can correspond to the health and demographic traits and mobility patterns of individual agents. Such an attacker can manifest itself as the coordinating server that wants to surveil agents using the mobility trace or a (sub-group) of adversarial agents, which may be incentivized to steal the personal health information of agent cohorts. In the context of agent-based modeling, this information can be leaked during message passing over per-step neighborhoods (Equation (1)) and during the collection of summary statistics over the population. The goal of this work is to alleviate such challenges and design a privacy-preserving mechanism that can compute functions over agents' states without revealing private information.

3.2 Secure Multi-party Computation

Secure multi-party computation enables a set of agents to interact and compute a joint function of their private inputs while revealing nothing but the output [40]. MPC protocols are coordinated with a server (MPC server) and are designed to protect against behavior of adversarial participants. These participants, either an agent or the server, aim to learn private information (of other entities) or cause the computation result to be incorrect. The idea was first introduced by Yao for the two-party case [58] and generalized to multiparty settings by Goldreich, Micali and Wigderson (GMW) [27]. Among other properties, GMW protocols guarantee (1) *privacy*: so that no entity can learn anything more than its prescribed output and, (2) *correctness*: so that agents receive the correct output. For instance, in an epidemiological ABM, this would ensure both that the personal disease status of agents is not leaked and that agents receive the correct transmission probability as in a centralized simulation.

3.3 The GMW protocol

The GMW protocol uses additive secret sharing to communicate (or aggregate) private inputs across the participant entities. The key insight is to divide a secret input into multiple shares in such a way that the secret can be reconstructed only when a sufficient number of shares are combined together. The scheme supports diverse aggregation queries such as secure addition or secure multiplication [9] of the secrets held by the participating agents. Here we focus on the addition case and we assume that all participating agents are required to compute the secret, usually denoted by $t = N$, but the same methodology can be extended to multiplication and composite queries (see, e.g., [40]).

Consider N agents holding private values s_i. We want to compute the sum $\sum_i s_i$ without any agent j acquiring knowledge about $s_{\{k\neq j\}}$. To setup the protocol, the agents agree on an integer $n > \sum_i s_i$ defining the finite group $\mathbb{Z}_n$ on which all computations will be carried [1]. Each agent i then samples $N-1$ random numbers, $r_{ij} \sim \mathcal{U}\{0, n-1\}$, such that the input is divided into N shares, s_{ij} defined by

$$s_i = \sum_{j=1}^{N} s_{\mathrm{ij}} \pmod n = \sum_{j=1}^{N-1} r_{ij} + \left(s_i - \sum_{j=1}^{N-1} r_{ij}\right) \pmod n. \tag{9}$$

Each agent then sends each share of their secret to each corresponding agent; agent i sends s_{i1} share to agent 1, s_{i2} share to agent 2,

[1]The choice to perform finite group arithmetics is so that no information about the secret can be gained by holding $< N$ shares.

etc. Locally, each agent performs the sum

$$\sigma_k = \sum_{i=1}^{N} s_{ik} \pmod{n}. \tag{10}$$

Finally, all values σ_k are shared so that the reconstructed sum, $S = \sum_k \sigma_k \pmod{n}$, can be computed, corresponding to the sum of the agent inputs s_i by construction. Typically, this reconstruction may be conducted by a central MPC server or a trusted agent. We summarize the protocol in Algorithm 1, and we provide an illustrating example below.

3.3.1 Additive secret sharing example. Consider $N = 3$ agents — Alice, Bob, and Carol — holding private values $s_A = 2$, $s_B = 3$, and $s_C = 5$. They wish to compute the sum of these values without disclosing their individual inputs. They agree on an integer $n = 11$, defining a finite group $\mathbb{Z}_n$. First, the agents generate 3 shares each by sampling 2 random numbers from $\mathbb{Z}_n$. For instance, Alice generates random numbers 7 and 5, so that

$$s_A = s_{AA} + s_{AB} + s_{AC} = 7 + 5 + 1 \pmod{11} = 2, \tag{11}$$

and similarly for Bob and Carol with $s_B = 2 + 0 + 1 \pmod{11}$, and $s_C = 3 + 1 + 1 \pmod{11}$. Second, the agents communicate with each other to keep one of the shares and send the other two to the other two agents and perform the sum of the received shares. For example, Alice receives s_{BA} from Bob and s_{CA} from Carol and computes

$$\sigma_A = s_{AA} + s_{BA} + s_{CA} = 7 + 2 + 3 \pmod{11} = 1 \pmod{11}, \tag{12}$$

and similarly for Bob and Carol with $\sigma_B = 5 + 0 + 1 \pmod{11} = 6 \pmod{11}$ and $\sigma_C = 1 + 1 + 1 \pmod{11} = 3 \pmod{11}$. Finally, the secret can be reconstructed by doing $S = \sigma_A + \sigma_B + \sigma_C = 10 \pmod{11}$ as expected.

In the following section, we apply the GMW protocol to generalize the above insight to share information containing agent's private information to other agents or a central MPC server, providing protocols for the computation of agent updates (Equation (1)), and gradients (Equation (3)) in a secure way, enabling privacy-preserving simulation, calibration, and analysis of ABMs.

Algorithm 1: SecureSum

Data: Agents $\{1, \ldots, N\}$ with secret inputs $s_1, \ldots, s_n$, integer $n > \sum_i s_i$.
Result: The sum of all shares $S = s_1 + \cdots + s_n$.

Splitting secret into shares and distributing:
Each party i generates N shares $s_{i1}, \ldots, s_{iN} \in \mathbb{Z}_n$ which sum up to s_i.
Each party i distributes all their shares $s_{i1}, \ldots, s_{iN} \in \mathbb{Z}_n$ to $1, \ldots, N$, including themselves.
Secure Computation (Addition):
To add the inputs securely, parties simply add their respective shares $\sigma_i = s_{1i} + \cdots s_{Ni} \mod n$.
Reconstruction:
To reveal the final result of the computation, parties collaborate by summing their shares:
$S = (\sigma_1 + \sigma_2 + \cdots + \sigma_n) \mod n$.

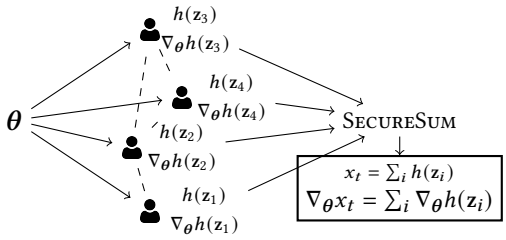

Figure 1: Diagram illustrating the SecureSimulation protocol for ABM parameters θ.

4 PRIVATE AGENT-BASED MODELS

With the MPC background introduced in the preceding section, we formulate protocols to conduct the simulation, calibration, and analysis of ABMs in a privacy-preserving way.

4.1 Secure Simulation

In order to update the state of an agent during a simulation, Equation (1) requires an aggregation over the agent's neighbors, revealing their private information. As described in the previous section, MPC is well-equipped to perform this kind of calculation without revealing the individual parties' data. Without loss of generality, we present our protocols for the case where the aggregation function $\oplus$ is a summation Σ, so that we can make use of the SecureSum protocol introduced in Algorithm 1. Furthermore, as long as the agent's update function f is differentiable respect to the structural parameters $\boldsymbol{\theta}$, which is indeed the case for many ABMs [19], each agent can store $\nabla_{\boldsymbol{\theta}} f$ for use during the calibration step. With all this in mind, we present in Algorithm 2, a privacy-preserving protocol for updating agent states.

Algorithm 2: SecureAgentUpdate

Data: Agent i with state $\mathbf{z}_i(t)$, Neighboring agent's messages $\{M_{ij}(t) \mid j \in \mathcal{N}(i)\}$, Integer n, State update rule f, ABM parameters $\boldsymbol{\theta}$
Result: New state $\mathbf{z}_i(t+1)$

Agent i calls the SecureSum protocol with neighbors $\{j \mid j \in \mathcal{N}(i)\}$ and integer n to obtain the sum $M_i(t) = \sum_{j \in \mathcal{N}(i)} M_{ij}(t)$.
Agent i updates its state $\mathbf{z}_i(t+1) = f(\mathbf{z}_i(t), M_i(t), \boldsymbol{\theta})$ and stores the gradient $\nabla_{\boldsymbol{\theta}} f$.

It is worth noting that, in contrast to general applications of the GMW protocol, only the agent who starts the protocol receives the result of the computation since there is no need for the neighboring agents to have access to that information.

Next, we introduce the SecureSimulation protocol in Algorithm 3, where, in addition to performing agent updates, we collect a time series of aggregate statistics over the agent's population and its gradient with respect to the ABM structural parameters $\boldsymbol{\theta}$.

4.2 Secure Calibration

During the calibration of an ABM, the modeler (central MPC server) requires the ability to evaluate the ABM at different values of $\boldsymbol{\theta}$, and,

Algorithm 3: SecureSimulation

Data: MPC server C, Agents $\{1, \dots, N\}$ with states $\{\mathbf{z}_1, \dots, \mathbf{z}_N\}$, ABM parameters $\boldsymbol{\theta}$, State update rule f, Number of time-steps T

Result: Aggregate statistics $\mathbf{x} = x_1, \dots, x_T$ and gradients $\nabla_{\boldsymbol{\theta}}\mathbf{x}$.

C generates a large enough prime number P and the requested statistics collecting function h; and sends them to all agents along ABM parameters $\boldsymbol{\theta}$.

for *t = 1, …, T* **do**

  for *i = 1, …, N* **do**

    Agent i calls the SecureAgentUpdate protocol (Algorithm 2) to compute $\mathbf{z}_i(t+1)$.

    Agent i gathers its information of interest $h(\mathbf{z}_i(t+1))$ and gradient $\nabla_{\boldsymbol{\theta}} h(\mathbf{z}_i(t+1))$.

  C calls the SecureSum protocol with all agents to collect the aggregate statistics x_t and their gradients $\nabla_{\boldsymbol{\theta}} x_t$.

C returns the accumulated $\mathbf{x}$ and $\nabla_{\boldsymbol{\theta}}\mathbf{x}$.

in the case of gradient-assisted calibration, the gradient of the outputs with respect to $\boldsymbol{\theta}$. The retrieval of these quantities is enabled by the SecureSimulation protocol and so classical calibration algorithms [30, 49] can be seamlessly applied to conduct a privacy-preserving calibration. Here, as outlined in Subsection 2.2, we focus on conducting GVI with a deep neural network trained to approximate the generalized posterior. This neural network seats on the central MPC server and it is trained using the collected gradients. To this end, what remains to be detailed is the transition from the gradient of aggregate statistics, $\nabla_{\boldsymbol{\theta}}\mathbf{x}$, to the loss gradient required for GVI (Equation (7)). This process is detailed in Algorithm 4.

4.3 Secure Analysis

As outlined in Subsection 2.3, we require a secure protocol to retrieve summary statistics over the agent's population, $\xi_k = h_k(\{\mathbf{z}_i(t) \mid i \in A\})$, and their sensitivity, $\nabla_{\boldsymbol{\theta}}\xi_k$. We note that we have already addressed this issue in the SecureSimulation protocol since retrieving the time-series $\mathbf{x}$ is a particular case of this more general problem. In fact, we formulate a more general approach wherein a summary statistic ξ over the population states can be obtained by aggregating over the entire population the outputs of an indicator function $\mathbb{1}_\Omega(\mathbf{z})$ acting on the agent's state,

$$\xi = \bigoplus_{i \in A} \mathbb{1}_\Omega(\mathbf{z}_i), \tag{13}$$

where Ω denotes the set of characteristics we want to aggregate on, and $\oplus$ denotes a kind of aggregation, usually a sum. For instance, Ω may correspond to the property of being infected, in which case taking $\oplus$ to be a sum would return the number of infected agents. The algorithm is formally described in Algorithm 5, which holds the same security guarantees as the one exposed in Subsection 4.1.

Likewise, sensitivity analysis can be performed securely using Algorithm 6, where each agent computes the sensitivity of their state change to the model parameters, and the central agent retrieves the aggregation by employing the SecureSum protocol.

Algorithm 4: SecureCalibration

Data: MPC server C, Agents $\{1, \dots, N\}$ with states $\{\mathbf{z}_1, \dots, \mathbf{z}_N\}$, State update rule f, Prior $\pi(\boldsymbol{\theta})$, Observed time-series $\mathbf{y}$, Loss function $\ell(\mathbf{x}, \mathbf{y})$, Number of epochs N_e, Number Monte-Carlo samples N, Number of time-steps T, Learning rate η

Result: Trained candidate posterior q^*

C initializes candidate posterior $q_{\boldsymbol{\phi}}$.

for *1, …, N_e* **do**

  for *1, …, N* **do**

    C samples ABM parameters $\boldsymbol{\theta} \sim q_{\boldsymbol{\phi}}(\boldsymbol{\theta})$.

    C executes SecureSimulation protocol (Algorithm 3) to obtain $\mathbf{x}$ and $\nabla_{\boldsymbol{\theta}}\mathbf{x}$.

    C uses the chain rule to compute

$$\nabla_{\boldsymbol{\theta}} \ell(\mathbf{x}(\boldsymbol{\theta}, \mathbf{y})) = \nabla_{\mathbf{x}} \ell(\mathbf{x}, \mathbf{y}) \cdot \nabla_{\boldsymbol{\theta}} \mathbf{x}$$

$$\nabla_{\boldsymbol{\phi}} \ell(\mathbf{x}(\boldsymbol{\theta}), \mathbf{y}) = \nabla_{\boldsymbol{\theta}} \ell(\mathbf{x}(\boldsymbol{\theta}), \mathbf{y}) \cdot \nabla_{\boldsymbol{\phi}} \boldsymbol{\theta}$$

    using the reparameterization trick (Equation (7)).

  C accumulates gradient to estimate

$$\nabla_{\boldsymbol{\phi}} \mathbb{E}_{q_{\boldsymbol{\phi}}} [\ell(\mathbf{x}(\boldsymbol{\theta}), \mathbf{y})] \approx \frac{1}{N} \sum_{i=1}^{N} \nabla_{\boldsymbol{\phi}} \ell(x(\theta), y).$$

  C computes divergence $\mathrm{KL}(q_{\boldsymbol{\phi}} \mid\mid \pi(\boldsymbol{\theta}))$ and derivative $\nabla_{\boldsymbol{\phi}} \mathrm{KL}(q_{\boldsymbol{\phi}} \mid\mid \pi(\boldsymbol{\theta}))$.

  C updates $\boldsymbol{\phi} \to \boldsymbol{\phi} - \eta \nabla_{\boldsymbol{\phi}} \mathcal{L}(\boldsymbol{\phi})$, where

$$\mathcal{L}(\boldsymbol{\phi}) = \mathbb{E}_{q_{\boldsymbol{\phi}}} [\ell(\mathbf{x}(\boldsymbol{\theta}), \mathbf{y})] + \mathrm{KL}(q_{\boldsymbol{\phi}} \mid\mid \pi(\boldsymbol{\theta})).$$

C returns trained candidate posterior q^*.

Algorithm 5: SecureSummaryStatistic

Data: MPC server C, Agents $\{1, \dots, N\}$ with states $\{\mathbf{z}_1, \dots, \mathbf{z}_N\}$, Indicator function $\mathbb{1}_\Omega(\mathbf{z})$.

Result: Aggregate quantity $\bigoplus_{i \in A} \mathbb{1}_\Omega(\mathbf{z}_i)$.

C generates prime number P and sends them to all agents alongside indicator function $\mathbb{1}_\Omega(\mathbf{z})$.

Each agent i computes result of $\mathbb{1}_\Omega(\mathbf{z}_i)$.

C executes SecureSum protocol to securely retrieve $\bigoplus_i^N \mathbb{1}_\Omega(\mathbf{z}_i)$.

5 CASE STUDY: PRIVACY-PRESERVING EPIDEMIOLOGY

In this section, we illustrate how our methodology may be deployed in practice by describing a decentralized privacy-preserving agent-based SIR model, which supports simulation, calibration, and analysis.

The model follows a standard parameterization where agents' interactions are specified through a contact graph $\mathcal{G}$, which in this case is only locally defined by each agent having access to their neighbors. Each agent has 3 possible states: 0 (Susceptible), 1 (Infected), and 2 (Recovered). We initialize the simulation by infecting a fraction I_0 of agents sampled uniformly from the population, while the remaining agents are considered susceptible. Following the notation introduced in Section 2, at each time-step, agent i

Algorithm 6: SecureSensitivityAnalysis

Data: MPC server C, Agents $\{1, \ldots, N\}$ with states $\{\mathbf{z}_1, \ldots, \mathbf{z}_N\}$, Indicator function $\mathbb{1}_\Omega(\mathbf{z})$, State update rule f

Result: Sensitivity $\nabla_{\boldsymbol{\theta}} \xi$.

1 C generates prime number P and sends them to all agents alongside indicator function $\mathbb{1}_\Omega(\mathbf{z})$.

2 Each agent i executes SecureAgentUpdate with parameters $\boldsymbol{\theta}$ to obtain $\nabla_{\boldsymbol{\theta}} f(\mathbf{z}_i, M_i, \boldsymbol{\theta})$ so that

$$\nabla_{\boldsymbol{\theta}} \xi_i = \nabla_{\boldsymbol{\theta}} f(\mathbf{z}_i, M_i, \boldsymbol{\theta}) \cdot \mathbb{1}_\Omega(\mathbf{z})$$

3 C executes SecureSum protocol to securely retrieve $\nabla_{\boldsymbol{\theta}} \xi = \bigoplus_i^N \nabla_{\boldsymbol{\theta}} \xi_i$.

updates its state following Equation (1) with

$$M_{ij}(t) = I_j(t) \tag{14}$$

where $I_j(t)$ is the infected status of the neighbor (0 or 1), so that

$$\begin{aligned} z_i(t+1) = {} & \mathbb{1}_{\{z_i=0\}} \cdot \text{Bernoulli}\left(p_{\text{inf}}^{(i)}(t)\right) + \\ & \mathbb{1}_{\{z_i=1\}} \cdot \left(1 + \text{Bernoulli}\left(p_{\text{rec}}^{(i)}\right)\right) + \\ & \mathbb{1}_{\{z_i=2\}} \cdot 2 \end{aligned} \tag{15}$$

with

$$p_{\text{inf}}^{(i)}(t) = 1 - \exp\left(-\frac{\beta S_i \Delta t}{n_i} \sum_{j \in \mathcal{N}(i)} I_j(t)\right), \tag{16}$$

where $\mathcal{N}(i)$ is the set of neighbors of agent i, S_i is the susceptibility of agent i, $n_i = \#\mathcal{N}(i)$ is the total number of neighbors, Δt is the duration of the time-step, and β is a structural parameter of the ABM called the effective contact rate. Infected agents can recover at each time step with recovery rate γ, so that

$$p_{\text{rec}}^{(i)} = 1 - \exp\left(-\gamma \Delta t\right). \tag{17}$$

For the case of a complete graph, the model reduces to the standard ODE-based SIR model with $R_0 = \beta/\gamma$ as the basic reproduction number. The model is run for n_t time steps.

To ground the example on real data, we consider the contact graph of the city of Oxford, extracted from the June ABM model [7] to determine the neighborhood of each agent, $\mathcal{N}(i)$. This contact graph includes agents' interactions in households, companies, and schools, and it is based on English census data. The choice of parameter values for the experiment is given in Table 1.

Parameter	Value
β	0.5 day^{-1}
γ	0.1 day^{-1}
I_0	0.01
Δt	1 day
n_t	60
$\mathcal{G}$	Oxford

Table 1: Parameter values for the agent-based SIR model.

5.1 Private policy assessment with ABMs

We first consider the application of the SecureSimulation protocol (Algorithm 3). Let us pose a situation where a policymaker wants to study the efficacy of mask-wearing at different compliance levels using agent-based simulation. We introduce a slight modification to Equation (16) to incorporate a reduction in the infection probability due to mask-wearing with certain compliance α,

$$p_{\text{inf}}^{(i)}(t) = 1 - \exp\left(-\frac{\beta S_i \Delta_t}{n_i} \sum_{j \in \mathcal{N}(i)} I_j(t)(1 - c_j)\right), \tag{18}$$

where $c_j \sim \text{Bernoulli}(\alpha)$, so that $\alpha_i = 1$ corresponds to full compliance where there is no transmission. Note that we are assuming complete protection against infection when wearing a mask. We proceed to execute 3 simulations for 3 different values of α. At each simulation, α is sent to the agents, where they locally compute their own compliance to the measure. The SecureSimulation protocol is then used to run the simulation and retrieve the aggregate statistic of interest, $\mathbf{x}$, which in this case is the number of infections over time. The results are shown in Figure 2, where we observe that little transmission occurs when compliance is above $\gtrapprox$ 75%.

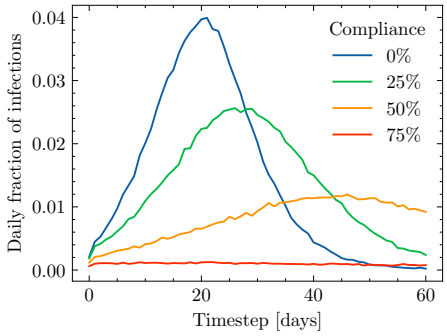

Figure 2: Infection curves for different levels of compliance: 0% (blue), 25% (green), 50% (orange), 75% (red). The number of infections has been normalized to the number of agents N. These plots are generated without releasing the infection status or compliance decision of any individual agent.

Thus, we observe that within our methodology, the policymaker could still have access to the same level of insight as a traditional ABM, all while protecting the individual agent's privacy.

5.2 Private calibration of ABMs

Next, we pose a situation where we want to calibrate our ABM with structural parameters $\boldsymbol{\theta} = (\beta, \gamma)$ to observed ground-truth data. For simplicity, we present the calibration of the β parameter given an observed curve of infections ($\mathbf{y}$), obtained by running the ABM model with the baseline parameters in Table 1.

The first step is to compute the gradient $\nabla_{\boldsymbol{\theta}} \mathbf{x}$, where $\mathbf{x}$ is the number of daily infections and $\theta = \beta$. We note that this gradient can be approximated by the gradient of the average number of new infections with respect to β,

$$\frac{\partial x_t}{\partial \beta} \approx \frac{\partial\, \mathbb{E}[\Delta I(t)]}{\partial \beta} = \sum_{i=1}^{N} \chi_i(t) \exp(-\chi_i(t)/\beta), \tag{19}$$

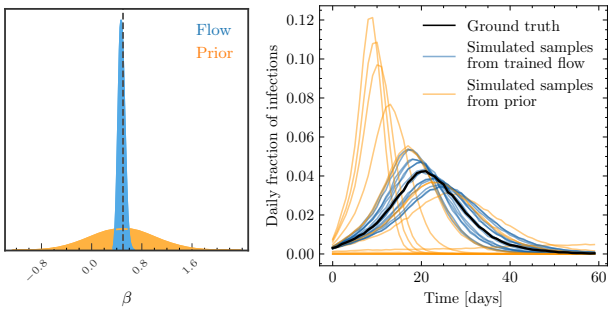

Figure 3: Left: Probability density plot for the trained normalizing flow (blue) against the prior distribution (orange). Ground-truth value is marked as a dashed black line. Right: Results from simulating β samples from the trained flow (blue) and prior (orange) compared to the ground-truth data (black). The number of infections has been normalized to the number of agents N.

where

$$\chi_i(t) = \exp\left(-\frac{\beta S_i \Delta_t}{n_i} \sum_{j \in \mathcal{N}(i)} I_j(t)\right). \tag{20}$$

A central agent can safely retrieve the gradient by performing the SecureSum protocol across all agents as described in Algorithm 3 and Algorithm 4. We thus conduct GVI by considering Q to be a masked-autoregressive normalizing flow [48] and assume the prior is a normal distribution with $\mu = 0.7$ and $\sigma = 0.5$. Figure 3 (left) shows the trained normalizing flow, which correctly assigns high probability mass to the ground-truth value. To further evaluate the goodness of the fit, we plot simulated runs from ABM parameters sampled from the trained flow in Figure 3 (right), where we compare it to runs simulated from prior samples.

This experiment highlights how privacy-preserving ABMs can be integrated into differential and probabilistic programming pipelines. This opens the door to integrating ABM insight into more complex ML pipelines leveraging heterogeneous data streams to boost the model's insight capabilities.

5.3 Private demographic study with ABMs

In this section, we apply the SecureSummaryStatistic protocol to analyze our calibrated ABM. In particular, we study the distribution of infections by age, ethnicity, and geographical location (ZIP code). This analysis may be relevant to, for instance, understanding the causes of asymmetric distribution of infections among different demographic groups [38, 42, 51].

For this exercise, we import a synthetic population of Oxford from the June model, as we did for the contact graph, so that we have access to a population with realistic age, sex, and ethnicity distributions and geographical location. Note that the ethnicity categorization follows the English census [29].

We can construct histograms that highlight the distribution of infection among different ethnic groups by considering the relevant indicator functions so that we can apply the SecureSummaryStatistic protocol. For instance, for the case of age distribution, we have

$$\mathbb{1}_\Omega(\mathbf{z}) = \mathbb{1}_{\{(z_{\text{age}} \in a_k) \wedge (z_{\text{inf}}=1)\}}(\mathbf{z}), \tag{21}$$

where z_{age} and z_{inf} are the age and infected status of the agent, and a_k are each of the histogram age bins. The number of counts in the particular age bin a_k can then be obtained upon executing SecureSum over the entire population following Algorithm 5. Figure 4 shows the age and ethnicity histogram of infections for the calibrated simulation obtained in Subsection 5.2, where the ethnicity histogram has been computed analogously to age one by considering the indicator function

$$\mathbb{1}_\Omega(\mathbf{z}) = \mathbb{1}_{\{(z_{\text{ethnicity}}=e_i) \wedge (z_{\text{inf}}=1)\}}(\mathbf{z}), \tag{22}$$

for each ethnicity category e_i. The results are in agreement with the particular demographics of the city of Oxford, dominated by students in the 20-30 age bin with a significant presence of non-White groups.

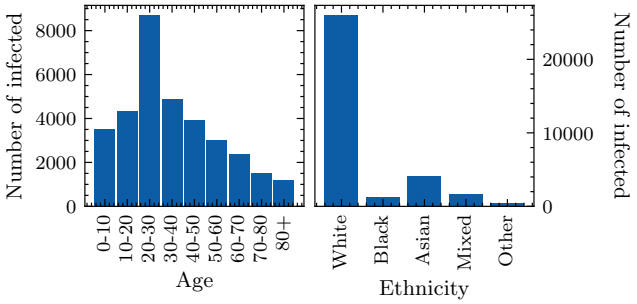

Figure 4: Age (left) and ethnicity (right) histogram of the infected population. These statistics are computed without leaking the infection or demographic properties of any agent.

Finally, a similar analysis can be done at the geographical level, where infections can be collected by ZIP code sector by executing the SecureSummaryStatistic protocol with $\mathbb{1}_{\Omega(\mathbf{z})} = \mathbb{1}_{\{z_{\text{ZIPcode}}=r_i\}}$ for each ZIP code sector r_i. In Figure 5, we show the obtained distribution of infections across the city of Oxford.

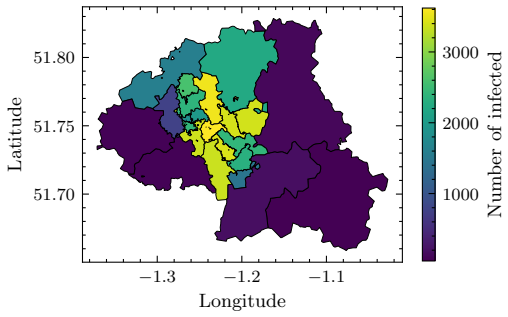

Figure 5: Geographical distribution of infections by ZIP code sector within the city of Oxford. These statistics are computed without leaking the infection status or geo-location of any individual agent.

Summarising, we have illustrated how a detailed analysis of agent properties can be performed in a privacy-preserving way by following Subsection 4.3.

6 RELATED WORK

Before concluding, we survey recent work on data-driven ABMs, with a particular focus on recent advances that aim to deploy ABMs to real-world populations with millions of agents. We also highlight the privacy challenges that arise from extending such ABMs.

6.1 Data-driven Agent-based Modeling

The push to create highly realistic ABMs has led to a growing demand for more granular heterogeneous data to construct the synthetic populations that underlie these models. This demand is particularly evident in fields such as epidemiology (e.g., [7, 18, 37]) and economics (e.g., [15, 50]), where the population is constructed from census data, using techniques such as iterative proportional fitting [20, 55] and deep generative modelling [13]. Furthermore, the integration of dynamic data streams, including sources like SafeGraph mobility data, CDC genomic data, and Facebook survey data, into the ABM calibration process has become essential for generating real-time insights. In addressing this need, the combination of differentiable ABMs and deep neural networks (DNN) stands out as an effective approach to seamlessly incorporate heterogeneous data sources during the calibration process [19]. Extending beyond the current resolution of ABM populations requires the development of new methodologies that safeguard the privacy of stakeholders, which is the objective of this work.

6.2 Data Privacy for Modeling

Recent improvements in ABM scalability have been matched by an increased risk to individual privacy [5], which has already manifested in the form of data leaks [1, 22, 36]. Statistical privacy methods such as K-anonymity and differential privacy have been employed [10, 57] to protect individual information on an aggregate level. Such methods have been used to release US census data [14] whilst protecting individual information at an aggregate level. Similar methods were also employed by Google and SafeGraph in the release of mobility trace data [2]. Although statistical hiding methods are good at hiding identifiable information in aggregate statistics, they have limited capability of achieving privacy when applied at the level of individuals. This creates a difficulty in simulating population networks, as state propagation relies on both accumulating individual agents' states, whilst evaluating policy interventions relies on access to mobility data for each agent. As a result, statistical hiding methods provide poor privacy-utility trade-offs in the context of ABM.

In contrast, cryptographic protocols have been used to provide strong privacy guarantees at the level of individuals. Recently, MPC has been explored for secure federated learning [45] and training of distributed graph neural networks [56]. These are constrained by high computation and communication cost of executing operations on deep neural networks, implementation complexity of secure aggregation, and intractability of secure backpropagation through non-linear mechanisms like graph attention. This requires hybrid approaches that leverage MPC with trusted execution environments for training DNNs [35]. In contrast, the simplicity of mechanistic models, like ABMs, allows leveraging MPC in fully decentralized scenarios. RIPPLE[31] introduces a Private Information Retrieval (PIR) based method to collect aggregate statistics on contact-tracing systems while preserving user privacy. [26] uses encrypted personal information, adding a layer of anonymity between the simulator and the agents. However, there is no mechanism for privacy-preserving calibration and interventions on ABMs [5]. To our knowledge, our work constitutes the first protocol that enables simulation, calibration, and analysis of ABMs while preserving agent privacy at every step.

7 DISCUSSION AND CONCLUSION

In this paper, we introduced a paradigm that enables the simulation, calibration, and analysis of agent-based simulations on real-world data, while safeguarding the privacy of the involved agents. Our approach leverages MPC techniques to develop robust privacy-preserving protocols, without compromising the accuracy of the ABM output. We demonstrated the efficacy of our paradigm by presenting a case study in the city of Oxford, where we evaluated mask-wearing policies, performed gradient-assisted calibration to ground-truth data, and analyzed simulation outcomes using a privacy-preserving ABM.

The presented concept model can be further developed by extending the MPC protocols in multiple ways, including

(1) Generalizing to higher-order networks to capture more complex contagion models. This could help understand the influence of group dynamics on social behavior.
(2) Addressing practical engineering challenges, such as minimizing communication overheads, supporting asynchronous message passing, and leveraging distributed computing to accelerate computation.
(3) Combining MPC with federated learning. While MPC enabled simulating with individual agents, federated learning can help incorporate siloed institutional agents. For example, this would help privately integrate CDC genomic data or FED insurance claims into epidemiological and economics models.

In the long term, private ABMs can be deployed through mobile devices, enabling passive and secure monitoring of actions and interactions within large populations. This would enable conducting digital experiments on the real-time behavior of complex systems with actual populations while ensuring robust security measures.

ACKNOWLEDGMENTS

We are grateful to N. Bishop for his very insightful comments on the paper. This research was supported by a UKRI AI World Leading Researcher Fellowship awarded to M Wooldridge (grant EP/W002949/1). M. Wooldridge acknowledges funding from Trustworthy AI - Integrating Learning, Optimisation and Reasoning (TAILOR) (https://tailor-network.eu/), a project funded by European Union Horizon2020 research and innovation program under Grant Agreement 952215.

REFERENCES

[1] 2021. Indonesia Probes Suspected Data Breach on COVID-19 App. *Reuters* (Aug. 2021).

[2] Ahmet Aktay, Shailesh Bavadekar, Gwen Cossoul, John Davis, Damien Desfontaines, Alex Fabrikant, Evgeniy Gabrilovich, Krishna Gadepalli, Bryant Gipson, Miguel Guevara, et al. 2020. Google COVID-19 Community Mobility Reports: Anonymization Process Description (Version 1.1). *arXiv preprint arXiv:2004.04145* (2020). arXiv:2004.04145

[3] Philipp Andelfinger. 2021. Differentiable Agent-Based Simulation for Gradient-Guided Simulation-Based Optimization. *arXiv:2103.12476 [cs, eess]* (March 2021). arXiv:2103.12476 [cs, eess]

[4] Gaurav Arya, Moritz Schauer, Frank Schäfer, and Chris Rackauckas. 2022. Automatic Differentiation of Programs with Discrete Randomness. arXiv:2210.08572 [cs, math]

[5] Samuel A. Assefa, Danial Dervovic, Mahmoud Mahfouz, Robert E. Tillman, Prashant Reddy, and Manuela Veloso. 2021. Generating Synthetic Data in Finance: Opportunities, Challenges and Pitfalls. In *Proceedings of the First ACM International Conference on AI in Finance (ICAIF '20)*. Association for Computing Machinery, New York, NY, USA, 1–8. https://doi.org/10.1145/3383455.3422554

[6] Robert L. Axtell and J. Doyne Farmer. [n.d.]. Agent-Based Modeling in Economics and Finance: Past, Present, and Future. *Journal of Economic Literature* ([n. d.]). https://doi.org/10.1257/jel.20221319

[7] Joseph Aylett-Bullock, Carolina Cuesta-Lazaro, Arnau Quera-Bofarull, Miguel Icaza-Lizaola, Aidan Sedgewick, Henry Truong, Aoife Curran, Edward Elliott, Tristan Caulfield, Kevin Fong, Ian Vernon, Julian Williams, Richard Bower, and Frank Krauss. 2021. June: Open-Source Individual-Based Epidemiology Simulation. *Royal Society Open Science* 8, 7 (July 2021), 210506. https://doi.org/10.1098/rsos.210506

[8] Atilim Gunes Baydin, Barak A. Pearlmutter, Alexey Andreyevich Radul, and Jeffrey Mark Siskind. 2018. Automatic Differentiation in Machine Learning: A Survey. *Journal of Machine Learning Research* 18, 153 (2018), 1–43.

[9] Donald Beaver. 1992. Efficient Multiparty Protocols Using Circuit Randomization. In *Advances in Cryptology—CRYPTO'91: Proceedings 11*. Springer, 420–432.

[10] Claudio Bettini, Sergio Mascetti, X Sean Wang, Dario Freni, and Sushil Jajodia. 2009. Anonymity and Historical-Anonymity in Location-Based Services. *Privacy in location-based applications: research issues and emerging trends* (2009), 1–30.

[11] Parantapa Bhattacharya, Jiangzhuo Chen, Stefan Hoops, Dustin Machi, Bryan Lewis, Srinivasan Venkatramanan, Mandy L Wilson, Brian Klahn, Aniruddha Adiga, Benjamin Hurt, et al. 2023. Data-Driven Scalable Pipeline Using National Agent-Based Models for Real-Time Pandemic Response and Decision Support. *The International Journal of High Performance Computing Applications* 37, 1 (2023), 4–27.

[12] Eric Bonabeau. 2002. Agent-Based Modeling: Methods and Techniques for Simulating Human Systems. *Proceedings of the National Academy of Sciences of the United States of America* 99, 10 (2002), 7280–7287. arXiv:3057854

[13] Stanislav S. Borysov, Jeppe Rich, and Francisco C. Pereira. 2019. How to Generate Micro-Agents? A Deep Generative Modeling Approach to Population Synthesis. *Transportation Research Part C: Emerging Technologies* 106 (Sept. 2019), 73–97. https://doi.org/10.1016/j.trc.2019.07.006

[14] US Census Bureau. [n.d.]. Why the Census Bureau Chose Differential Privacy. https://www.census.gov/library/publications/2023/decennial/c2020br-03.html.

[15] Adrian Carro, Marc Hinterschweiger, Arzu Uluc, and J Doyne Farmer. 2023. Heterogeneous Effects and Spillovers of Macroprudential Policy in an Agent-Based Model of the UK Housing Market. *Industrial and Corporate Change* 32, 2 (April 2023), 386–432. https://doi.org/10.1093/icc/dtac030

[16] Kevin Chapuis, Patrick Taillandier, and Alexis Drogoul. 2022. Generation of Synthetic Populations in Social Simulations: A Review of Methods and Practices. *Journal of Artificial Societies and Social Simulation* 25, 2 (2022), 6.

[17] Ayush Chopra. 2022. *Decision Making for Populations*. Ph.D. Dissertation. Massachusetts Institute of Technology.

[18] Ayush Chopra, Ramesh Raskar, Jayakumar Subramanian, Balaji Krishnamurthy, Esma S Gel, Santiago Romero-Brufau, Kalyan S Pasupathy, and Thomas C Kingsley. 2021. DeepABM: Scalable and Efficient Agent-Based Simulations via Geometric Learning Frameworks-a Case Study for COVID-19 Spread and Interventions. In *2021 Winter Simulation Conference (WSC)*. IEEE, 1–12.

[19] Ayush Chopra, Alexander Rodríguez, Jayakumar Subramanian, Arnau Quera-Bofarull, Balaji Krishnamurthy, B. Aditya Prakash, and Ramesh Raskar. 2023. Differentiable Agent-based Epidemiology. In *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems (AAMAS '23)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 1848–1857.

[20] Abdoul-Ahad Choupani and Amir Reza Mamdoohi. 2016. Population Synthesis Using Iterative Proportional Fitting (IPF): A Review and Future Research. *Transportation Research Procedia* 17 (Jan. 2016), 223–233. https://doi.org/10.1016/j.trpro.2016.11.078

[21] Adam Coates, Liangxiu Han, and Anthony Kleerekoper. 2018. A Unified Framework for Opinion Dynamics. In *Proceedings of the 17th International Conference on Autonomous Agents and Multiagent Systems*. International Foundation for Autonomous Agents and Multiagent Systems.

[22] Joseph Cox. 2019. T-Mobile 'Put My Life in Danger' Says Woman Stalked With Black Market Location Data.

[23] Peter Sheridan Dodds and Duncan J. Watts. 2004. Universal Behavior in a Generalized Model of Contagion. *Physical Review Letters* 92, 21 (May 2004), 218701. https://doi.org/10.1103/PhysRevLett.92.218701

[24] Joel Dyer, Patrick Cannon, J. Doyne Farmer, and Sebastian Schmon. 2022. Black-Box Bayesian Inference for Economic Agent-Based Models. https://doi.org/10.48550/arXiv.2202.00625 arXiv:2202.00625 [cs, econ, stat]

[25] Dimitris Fotakis, Dimitris Palyvos-Giannas, and Stratis Skoulakis. 2016. Opinion Dynamics with Local Interactions.. In *IJCAI*. 279–285.

[26] Enrique Frias-Martinez, Graham Williamson, and Vanessa Frias-Martinez. 2011. An Agent-Based Model of Epidemic Spread Using Human Mobility and Social Network Information. In *2011 IEEE Third International Conference on Privacy, Security, Risk and Trust and 2011 IEEE Third International Conference on Social Computing*. IEEE, 57–64.

[27] Oded Goldreich, Silvio Micali, and Avi Wigderson. 2019. How to Play Any Mental Game, or a Completeness Theorem for Protocols with Honest Majority. In *Providing Sound Foundations for Cryptography: On the Work of Shafi Goldwasser and Silvio Micali*. 307–328.

[28] Chang Gong, Oleg Milberg, Bing Wang, Paolo Vicini, Rajesh Narwal, Lorin Roskos, and Aleksander S. Popel. 2017. A Computational Multiscale Agent-Based Model for Simulating Spatio-Temporal Tumour Immune Response to PD1 and PDL1 Inhibition. *Journal of The Royal Society Interface* 14, 134 (Sept. 2017), 20170320. https://doi.org/10.1098/rsif.2017.0320

[29] Gov.uk. [n.d.]. List of Ethnic Groups. https://www.ethnicity-facts-figures.service.gov.uk/style-guide/ethnic-groups.

[30] Jakob Grazzini, Matteo G. Richiardi, and Mike Tsionas. 2017. Bayesian Estimation of Agent-Based Models. *Journal of Economic Dynamics and Control* 77 (April 2017), 26–47. https://doi.org/10.1016/j.jedc.2017.01.014

[31] Daniel Günther, Marco Holz, Benjamin Judkewitz, Helen Möllering, Benny Pinkas, Thomas Schneider, and Ajith Suresh. 2022. Privacy-Preserving Epidemiological Modeling on Mobile Graphs. arXiv:2206.00539 [cs]

[32] Carmit Hazay and Yehuda Lindell. 2010. A Note on the Relation between the Definitions of Security for Semi-Honest and Malicious Adversaries. *Cryptology ePrint Archive* (2010).

[33] Robert Hinch, William J. M. Probert, Anel Nurtay, Michelle Kendall, Chris Wymant, Matthew Hall, Katrina Lythgoe, Ana Bulas Cruz, Lele Zhao, Andrea Stewart, Luca Ferretti, Daniel Montero, James Warren, Nicole Mather, Matthew Abueg, Neo Wu, Olivier Legat, Katie Bentley, Thomas Mead, Kelvin Van-Vuuren, Dylan Feldner-Busztin, Tommaso Ristori, Anthony Finkelstein, David G. Bonsall, Lucie Abeler-Dörner, and Christophe Fraser. 2021. OpenABM-Covid19—An Agent-Based Model for Non-Pharmaceutical Interventions against COVID-19 Including Contact Tracing. *PLOS Computational Biology* 17, 7 (July 2021), e1009146. https://doi.org/10.1371/journal.pcbi.1009146

[34] John H. Holland and John H. Miller. 1991. Artificial Adaptive Agents in Economic Theory. *The American Economic Review* 81, 2 (1991), 365–370. arXiv:2006886

[35] Yixin Jie, Yixuan Ren, Qingtao Wang, Yankai Xie, Chi Zhang, Lingbo Wei, and Jianqing Liu. 2022. Multi-Party Secure Computation with Intel SGX for Graph Neural Networks. In *ICC 2022 - IEEE International Conference on Communications*. 528–533. https://doi.org/10.1109/ICC45855.2022.9839282

[36] Bennett Cyphers and Jason Kelley. 2021. Illinois Bought Invasive Phone Location Data From Banned Broker Safegraph. https://www.eff.org/deeplinks/2021/08/illinois-bought-invasive-phone-location-data-banned-broker-safegraph.

[37] Cliff C. Kerr, Robyn M. Stuart, Dina Mistry, Romesh G. Abeysuriya, Katherine Rosenfeld, Gregory R. Hart, Rafael C. Núñez, Jamie A. Cohen, Prashanth Selvaraj, Brittany Hagedorn, Lauren George, Michał Jastrzębski, Amanda S. Izzo, Greer Fowler, Anna Palmer, Dominic Delport, Nick Scott, Sherrie L. Kelly, Caroline S. Bennette, Bradley G. Wagner, Stewart T. Chang, Assaf P. Oron, Edward A. Wenger, Jasmina Panovska-Griffiths, Michael Famulare, and Daniel J. Klein. 2021. Covasim: An Agent-Based Model of COVID-19 Dynamics and Interventions. *PLOS Computational Biology* 17, 7 (July 2021), e1009149. https://doi.org/10.1371/journal.pcbi.1009149

[38] Kamlesh Khunti, Awadhesh Kumar Singh, Manish Pareek, and Wasim Hanif. 2020. Is Ethnicity Linked to Incidence or Outcomes of Covid-19? *BMJ* 369 (April 2020), m1548. https://doi.org/10.1136/bmj.m1548

[39] Jeremias Knoblauch, Jack Jewson, and Theodoros Damoulas. 2022. An Optimization-Centric View on Bayes' Rule: Reviewing and Generalizing Variational Inference. *The Journal of Machine Learning Research* 23, 1 (2022), 5789–5897.

[40] Yehuda Lindell. 2020. Secure Multiparty Computation (MPC). *Cryptology ePrint Archive* (2020).

[41] Yanir Marmor, Alex Abbey, Yuval Shahar, and Osnat Mokryn. 2023. Assessing Individual Risk and the Latent Transmission of COVID-19 in a Population with an Interaction-Driven Temporal Model. *Scientific Reports* 13, 1 (Aug. 2023), 12955. https://doi.org/10.1038/s41598-023-39817-9

[42] Christopher A. Martin, David R. Jenkins, Jatinder S. Minhas, Laura J. Gray, Julian Tang, Caroline Williams, Shirley Sze, Daniel Pan, William Jones, Raman Verma, Scott Knapp, Rupert Major, Melanie Davies, Nigel Brunskill, Martin Wiselka, Chris Brightling, Kamlesh Khunti, Pranab Haldar, and Manish Pareek. 2020. Socio-Demographic Heterogeneity in the Prevalence of COVID-19 during Lockdown Is Associated with Ethnicity and Household Size: Results from an Observational Cohort Study. *eClinicalMedicine* 25 (Aug. 2020). https://doi.org/10.1016/j.eclinm.2020.100466

[43] Shakir Mohamed, Mihaela Rosca, Michael Figurnov, and Andriy Mnih. 2020. Monte Carlo Gradient Estimation in Machine Learning. *Journal of Machine Learning Research* 21, 132 (2020), 1–62.

[44] Alexander Mordvintsev, Ettore Randazzo, and Craig Fouts. 2022. Growing Isotropic Neural Cellular Automata. https://doi.org/10.48550/arXiv.2205.01681 arXiv:2205.01681 [cs, q-bio]

[45] Vaikkunth Mugunthan, Antigoni Polychroniadou, David Byrd, and Tucker Hybinette Balch. [n.d.]. SMPAI: Secure Multi-Party Computation for Federated Learning. ([n. d.]).

[46] Kirill Müller and Kay W. Axhausen. 2010. Population Synthesis for Microsimulation: State of the Art. *Arbeitsberichte Verkehrs- und Raumplanung* 638 (Aug. 2010). https://doi.org/10.3929/ethz-a-006127782

[47] Marco Pangallo, Alberto Aleta, R. Maria del Rio Chanona, Anton Pichler, David Martín-Corral, Matteo Chinazzi, François Lafond, Marco Ajelli, Esteban Moro, Yamir Moreno, Alessandro Vespignani, and J. Doyne Farmer. 2022. The Unequal Effects of the Health-Economy Tradeoff during the COVID-19 Pandemic. https://doi.org/10.48550/arXiv.2212.03567 arXiv:2212.03567 [physics, q-fin]

[48] George Papamakarios, Theo Pavlakou, and Iain Murray. 2017. Masked Autoregressive Flow for Density Estimation. In *Advances in Neural Information Processing Systems*, Vol. 30. Curran Associates, Inc.

[49] Donovan Platt. 2020. A Comparison of Economic Agent-Based Model Calibration Methods. *Journal of Economic Dynamics and Control* 113 (April 2020), 103859. https://doi.org/10.1016/j.jedc.2020.103859

[50] Sebastian Poledna, Michael Gregor Miess, Cars Hommes, and Katrin Rabitsch. 2023. Economic Forecasting with an Agent-Based Model. *European Economic Review* 151 (Jan. 2023), 104306. https://doi.org/10.1016/j.euroecorev.2022.104306

[51] Arnau Quera-Bofarull, Ayush Chopra, Joseph Aylett-Bullock, Carolina Cuesta-Lazaro, Anisoara Calinescu, Ramesh Raskar, and Michael Wooldridge. 2023. Don't Simulate Twice: One-shot Sensitivity Analyses via Automatic Differentiation. In *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems (AAMAS '23)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 1867–1876.

[52] Arnau Quera-Bofarull, Ayush Chopra, Anisoara Calinescu, Michael Wooldridge, and Joel Dyer. 2023. Bayesian Calibration of Differentiable Agent-Based Models. *arXiv preprint arXiv:2305.15340* (2023). arXiv:2305.15340

[53] Arnau Quera-Bofarull, Joel Dyer, Anisoara Calinescu, and Michael Wooldridge. 2023. Some Challenges of Calibrating Differentiable Agent-Based Models. https://doi.org/10.48550/arXiv.2307.01085 arXiv:2307.01085 [cs, q-fin, stat]

[54] Santiago Romero-Brufau, Ayush Chopra, Alex J Ryu, Esma Gel, Ramesh Raskar, Walter Kremers, Karen S Anderson, Jayakumar Subramanian, Balaji Krishnamurthy, Abhishek Singh, et al. 2021. Public Health Impact of Delaying Second Dose of BNT162b2 or mRNA-1273 Covid-19 Vaccine: Simulation Agent Based Modeling Study. *BMJ (Clinical research ed.)* 373 (2021).

[55] Ludger Ruschendorf. 1995. Convergence of the Iterative Proportional Fitting Procedure. *The Annals of Statistics* 23, 4 (1995), 1160–1174. arXiv:2242759

[56] Songlei Wang, Yifeng Zheng, and Xiaohua Jia. 2023. SecGNN: Privacy-Preserving Graph Neural Network Training and Inference as a Cloud Service. *IEEE Transactions on Services Computing* 16, 4 (July 2023), 2923–2938. https://doi.org/10.1109/TSC.2023.3241615

[57] Zhibo Wang, Xiaoyi Pang, Yahong Chen, Huajie Shao, Qian Wang, Libing Wu, Honglong Chen, and Hairong Qi. 2018. Privacy-Preserving Crowd-Sourced Statistical Data Publishing with an Untrusted Server. *IEEE Transactions on Mobile Computing* 18, 6 (2018), 1356–1367.

[58] Andrew C Yao. 1982. Protocols for Secure Computations. In *23rd Annual Symposium on Foundations of Computer Science (Sfcs 1982)*. IEEE, 160–164.

[59] Shuli Zhou, Suhong Zhou, Zhong Zheng, and Junwen Lu. 2021. Optimizing Spatial Allocation of COVID-19 Vaccine by Agent-Based Spatiotemporal Simulations. *GeoHealth* 5, 6 (2021), e2021GH000427. https://doi.org/10.1029/2021GH000427