

# Towards complexity reduction of large-scale epidemic simulation in two-scale networks

Yulian Kuryliak<sup>1,\*</sup>, Michael Emmerich<sup>2</sup>

<sup>1</sup>*Institute of Computer Science and Information Technologies, ICSIT of Lviv Polytechnic National University, 12 Stepan Bandera street, 79000 Lviv, Ukraine*

<sup>2</sup>*Faculty of Information Technology, University of Jyväskylä, P.O.Box 35 (Agora) FI-40014, Finland*

## Abstract

We propose a two-scale epidemic model that distinguishes intra-community infection dynamics from inter-community transmission. At the micro-layer, an epidemic outbreak within a community is simulated with the Gillespie SIR engine with additional infections. At the macro scale each community collapses to a single node in a mobility-weighted meta-population graph; stochastic transmissions between nodes are timed with a hazard-integral implementation of Gillespie's SSA allows dynamically varying infectivity and susceptibility parameters to account for behavior changes, interventions, or emerging variants. We aim to reduce the complexity of visual analysis and simulation by introducing techniques to capture key aspects of two-scale network dynamics. For visualization, we propose Sankey diagrams to depict virus strain infection rates across communities. Our two-scale simulation approach considers cumulative infection exposure, allowing nodes to continue receiving infection pressure from external sources after initial outbreaks, capturing the effect of ongoing inter-community interactions. By abstracting macro-edges from point-to-point contacts a complexity reduction by a multiplicative factor proportional to the squared average size of a community can be achieved.

## Keywords

Large Scale Epidemic Simulation, Hazard-Integral Timing, Multi-Scale Complex Networks, Multi-Scale Networks, Multi-Scale Visualization, Flow Graphs

## List of Abbreviations

BA	Barabási–Albert model of complex scale-free network
CTMC	Continuous-Time Markov Chain
ICU	Intensive-Care Unit (special Infectious state)
SIR	Susceptible–Infectious–Removed epidemic model
SSA	Stochastic Simulation Algorithm (Gillespie)

## 1. Introduction

The aim of the paper is to develop a stochastic simulation and visual analysis technique for epidemic outbreaks on complex networks that would be possible to scale up to a country with millions of inhabitants. We condense a community into a single node (metanode) to model transmission between metanodes and use an agent-based simulation with infections from the outside to depict infection dynamics within communities.

Pandemic preparedness depends on our ability to anticipate how quickly and in what way an infection will spread, so that intensive care capacity, test-and-trace teams, and vaccination logistics can be mobilized in time and in the right places. In practice, the most critical policy indicators are how high the simultaneous case load climbs and how long the authorities have before that peak arrives. Classical compartmental models capture long-term averages, but they smooth out exactly those local surges that overwhelm hospitals. A finer-grained lens is, therefore, essential.

*MoMLeT-2025: 8th International Workshop on Modern Machine Learning Technologies, June, 14, 2025, Lviv-Shatsk, Ukraine*

\*Corresponding author.

† These authors contributed equally.

✉ yulian.a.kuryliak@lpnu.ua (Y. Kuryliak); michael.t.m.emmerich@jyu.fi (M. Emmerich)

ORCID 0000-0002-6600-2637 (Y. Kuryliak); 0000-0002-7342-2090 (M. Emmerich)



© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

One principled way to incorporate such fine-grained detail is to represent every individual as a node in a contact network and let infection propagate along its edges. Our previous work [1] presented a continuous-time Markov chain simulator (CTMC) based on an efficient customized Gillespie algorithm that realizes this idea **without** mean-field approximations, i.e., simulating the stochastic trajectory of an epidemic outbreak with realistic timing of events. The method works well with tens of thousands of individuals and accurately resolves the timing and magnitude of the infection peak. The complexity of single infection of modeling an infection event is  $O(kN)$ ,  $k$  - average degree of a node,  $N$  - number of nodes in the network. Taking into account the complexity, scaling the same agent-based engine from a 10 thousand-person community suburb to a country of 40 million multiplies the state space by three orders of magnitude per each iteration. Even if the contact graph remains sparse (average degree  $k \approx 10$ ) a single stochastic iteration grows from seconds to hours and modeling of one full iteration grows from minutes to weeks. Therefore, even with aggressive algorithmic optimization and access to powerful hardware, the agent-based engine is not an option for policy studies that require hundreds of replications across multiple parameter sets.

To overcome this limitation, we introduce a two-layer simulation model. By isolating the infection process within each community (micro layer) from the inter-community spread (macro layer), we effectively eliminate a large fraction of the network edges from the simulation space. This reduces overall complexity: the simulation time scales roughly linearly with the number of communities, in contrast to the quadratic growth in a full agent-based model. Moreover, representing each community as a node on the macro layer opens up new avenues for localized intervention. Policies such as mobility restrictions, targeted vaccination, or school closures can be applied to individual communities and their impact on national-level epidemic dynamics can be efficiently simulated and analyzed.

## 2. Related Work: Multi-Scale Network Simulation in Epidemics

Our work is mainly inspired by Colizza et al. [2], who describe a metapopulation system using transport flow data as city connections.

Multi-scale epidemic models extend classical compartmental models by embedding them in *metapopulation frameworks*, where *nodes* (metanodes) represent spatially distinct populations (e.g., cities or regions), and *edges* (metalinks) represent interactions such as travel or migration. The foundational work by Colizza and Vespignani [2] demonstrated how global disease spread can be modeled by coupling intra-population SIR dynamics with inter-population mobility networks.

Recent developments focus on integrating *high-resolution mobility data*, *heterogeneous contact patterns*, and *agent-based models* at the local level while maintaining coarse-grained population-level transmission dynamics between communities. Notable approaches include the GLEaM (Global Epidemic and Mobility model) [3], which combines stochastic compartment models with global airline and commuting networks, and patch-based models for integrating real-world commuting and transportation data [4].

Advances in computational tools have also enabled scalable simulations across multiple levels of granularity. For instance, EpiGraph [5] and Covasim [6] allow for nested simulations where community-level dynamics inform inter-community transmission rates.

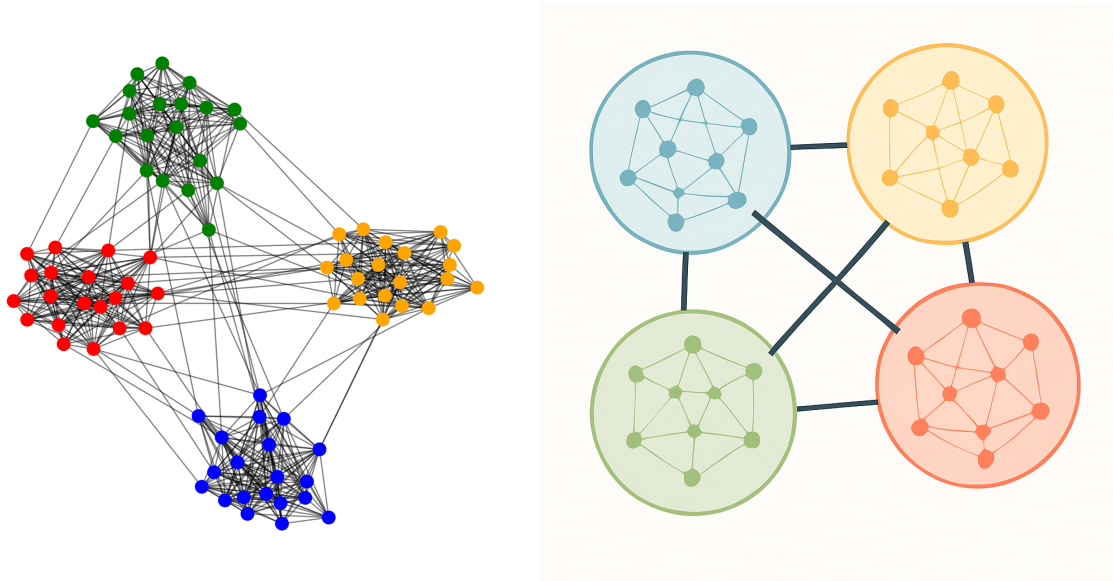
One more work that describes multiscale agent-based model is [7].

Ongoing challenges include aligning model granularity with data availability, handling non-Markovian transitions, and calibrating heterogeneous models under uncertainty.

## 3. Two-Scale Epidemic Model: Conceptual Framework

### 3.1. Overview of the Two-Scale Epidemic Model

National-level epidemic simulations must reconcile two conflicting requirements: (i) *microscopic realism* inside each locality and (ii) *computational tractability* across millions of individuals. Our Two-Scale



(a) **Microscopic view.** Agent-level contact graph comprising four densely connected communities (colours) and a sparse set of inter-community links. (b) **Two-layer abstraction.** The inter-community links have been abstracted to appear as single links (black edges) instead of point-to-point connections.

**Figure 1:** Community-structured network at two resolutions.

Epidemic Model achieves that balance by running two inter-connected layers whose time lines advance in perfect synchrony (Fig. 1b):

**Micro layer.** Every community is treated as an independent population and simulated with a stochastic Gillespie-based engine. At the end of each micro step  $\tau_{\text{micro}}$  (typically a few hours) the model outputs two scalars: contagiousness  $C_i(t)$  and susceptibility  $S_i(t)$ .

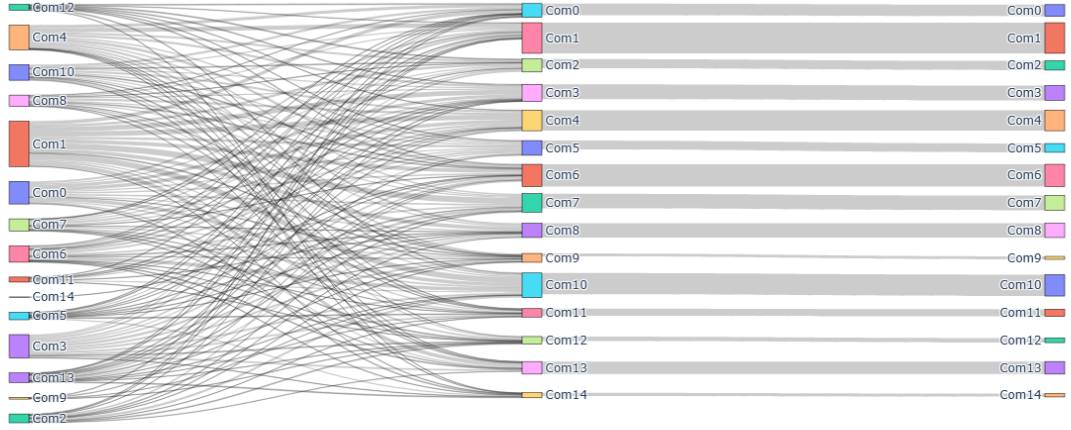
**Macro layer.** Communities are the nodes of a mobility-weighted graph. A continuous-time Markov chain (CTMC) whose time-dependent rates  $\lambda_{ij}(t) = \beta w_{ij} C_i(t) S_j(t)$  capture cross-community mobility generates the moments at which one community seeds a new infection in another. Event timing is obtained with the hazard-integral method described in Section 3.2.

**Layer synchronisation.** The two layers are updated in strict lock-step:

1. During each micro interval  $\tau_{\text{micro}}$  every community integrates its own SIR dynamics and refreshes  $(C_i, S_i)$ ; those numbers are immediately passed to the macro layer to update all  $\lambda_{ij}(t)$ .
2. The macro CTMC accumulates hazard  $H(t) = \int \Theta(s) ds$  on the same grid  $s_n = t_0 + n\tau_{\text{micro}}$ ; when  $H(t)$  reaches its random budget  $E \sim \text{Exp}(1)$  it triggers *exactly one* inter-community infection, seeds an infectious individual in the destination micro model, sets  $t_0 \leftarrow t^*$ , draws a new  $E'$ , and resumes.

This bidirectional handshake guarantees that the two layers remain numerically consistent at every discrete step.

**Sankey Flow diagram.** The diagram in Fig. 2 summarizes the simulation output of a simulated epidemic outbreak on the 'network of communities' scale. Figure 2 arranges the communities in two vertical bands. The bigger a node that represents a community is, the more infections the community transmitted or received. Nodes on the (left) represent the source community of an infection event, while those on the central and right marks the recipient communities. The flow from left to central column counts **all inter-community** (macro-layer) infection events, while the flow from central to



**Figure 2:** Sankey diagram of an epidemic outbreak in a two-layer visualization of flows of infections. The diagram is built on a median simulation in Barabási–Albert network model of 1000 nodes and 3 initial edges per node. The total number of inter infections is 502, intra infections is 474. The communities were detected by Louvain method.

right column tallies the accumulated **intra-community** (micro-layer) infections. The thickness of each ribbon is scaled as

$$w_{ij} = k \sqrt{T_{ij}},$$

where  $T_{ij}$  is the total number of transmissions and  $k$  a global scale factor chosen so that the thinnest visible edge remains  $\approx 0.4$  pt. This square-root mapping preserves small but epidemiologically relevant flows without allowing the largest outbreaks to dominate the canvas.

Start by following the thickest ribbons leaving the macro-layer. The most prominent stream pinpoints the community that exports the bulk of infections. Next, compare the width of its return flow: a narrow link back to itself but broad outgoing links to others signals a net *exporter* of infection pressure. In contrast, symmetric back-and-forth ribbons between two nodes indicate reciprocal mixing. The micro-layer further reveals where those imports amplify, because the right-hand node width is proportional to the *cumulative* intra-community burden. Thus, a modest import flow that swells into a thick micro-layer node flags a vulnerable community with intense local spread. quick visual estimate of the proportional contribution of each external source to a given outbreak.

Because the underlying contact graph (in which the simulation 2 was performed) is generated with the Barabási–Albert (BA) model, it contains high-degree hubs that contact many nodes in almost every community. As a result, the in the network are not strongly separated communities, therefore the totals of inter-community and intra-community infections are similar. Real-world contact networks are also approximately scale-free but typically have hubs of *lower* degree—households, workplaces, or schools rather than global “super-connectors.” We therefore expect real data to show a larger share of intra-community transmission than the BA benchmark depicted here.

**Complexity advantage.** Let  $N$  be the total number of agents (nodes) split into  $M$  communities. Denote by  $N_i$  their local sizes, by  $E_i$  the number of edges in the *intra-community* contact graph of community  $i$ , and by  $E_{\text{inter}}$  the number of *inter-community* mobility edges in the macro graph. A naïve agent-level SSA stores and updates one channel per susceptible–infectious edge, *i.e.*  $O(E_{\text{tot}})$  memory and  $O(E_{\text{tot}} T)$  run time, with  $E_{\text{tot}} = \sum_i E_i + E_{\text{inter}}$ ,  $T$  - total number of events. In dense networks with  $N$  nodes, up to  $O(N^2)$  edges may occur which can significantly strain the memory and time

resources, complicating simulations. Assuming the population can be partitioned into  $M$  equally sized communities, and that all inter-community connections between the same pair of communities are aggregated into a single weighted macro-link, the space complexity of the model can be reduced to  $O\left(M \left(\frac{N}{M}\right)^2\right)$ . Simplifying the expression, we obtain that the space complexity reduces by a multiplicative factor  $O\left(\frac{N^2}{M}\right)$ . The time complexity can be precisely determined using specific community sizes and Gillespie techniques as discussed in [1]. With the two-scale split:

- each Gillespie micro engine costs  $O(E_i)$  memory and  $O(I_i \langle k_i \rangle)$  time per infection, where  $I_i$  and  $\langle k_i \rangle$  are the current number of infectious nodes and their average intra-degree;
- the macro layer needs only  $O(E_{\text{inter}})$  channels and advances them in  $O(E_{\text{inter}})$  per macro event by the hazard–integral rule.

Overall storage scales as  $O(\sum_i E_i + E_{\text{inter}}) \ll O(E_{\text{tot}})$  and the run time per global step as  $O(\sum_i I_i \langle k_i \rangle + E_{\text{inter}})$ , yielding near-linear scaling in the number of communities.

**Targeted–intervention capability.** Because each community evolves in its *own* micro simulator, the platform naturally supports spatially targeted measures—lock-downs, surge vaccination or travel restrictions—are applied by simply altering that community’s parameters. An intervention applied to community  $k$  simply modifies its micro parameters (e.g. infection and recovery rates, edge weights or community structure), which in turn alters  $C_k(t)$  and  $S_k(t)$ . Those updated scalars automatically feed into the macro rates  $\lambda_{kj}(t)$  and  $\lambda_{jk}(t)$ , letting the model capture knock-on effects such as reduced export of cases from a locked-down community or heightened import risk for under-vaccinated community.

### 3.2. Macro-Layer: Inter-Community Transmission Model

**Network representation.** The national (or regional) population is partitioned into  $M$  discrete communities—cities, districts, university campuses, *etc.*— which become the nodes of a directed, weighted meta-population network. The weight  $w_{ij}$  on edge  $(i, j)$  measures how easily infection can pass from community  $i$  to community  $j$ ; it can be estimated from commuting statistics, transportation schedules, mobile-phone mobility data, or other social-interaction proxies.

**Node attributes.** Every node carries two time-dependent scalars, obtained from a within-community (micro-layer) simulator:

- **Contagiousness**  $C_i(t)$  – proportional to the current number of infectious individuals in the community  $i$ ;
- **Susceptibility**  $S_j(t)$  – proportional to the remaining susceptible individuals in the community  $j$ .

**Inter-community transmission rate.** The instantaneous hazard (rate) that one infection is seeded from community  $i$  to community  $j$  is

$$\lambda_{ij}(t) = \beta w_{ij} C_i(t) S_j(t) T,$$

where  $\beta$  is infection rate or the biological transmission rate of the pathogen and  $T$  is a constant calibration factor that matches the macro layer infection rate to the micro layer one. Only one kind of stochastic event exists at the macro layer: “*one additional infection appears in community  $j$* ”. After such an event, both  $C_i(t)$  and  $S_j(t)$  are updated by the micro-layer models and all  $\lambda_{ij}(t)$  are refreshed.



### 3.2.1. Modelling the Waiting Time Between Events

**Why the classical exponential draw is insufficient.** With a *constant* total rate  $\Theta = \sum_{i \neq j} \lambda_{ij}$ , the Gillespie algorithm samples the waiting time as

$$\Delta t = -\frac{\ln U}{\Theta}, \quad U \sim \mathcal{U}(0, 1).$$

This relies on the density  $f(t) = \Theta e^{-\Theta t}$  of the  $\text{Exp}(\Theta)$  distribution. In our application the rate is *not* constant because  $C_i(t)$  and  $S_j(t)$  evolve continuously. The correct total rate is therefore a function of time:

$$\Theta(t) = \sum_{i \neq j} \lambda_{ij}(t).$$

**Non-homogeneous Poisson formulation.** For a time-dependent rate, the probability that the next macro event has *not* occurred by real time  $t$  is

$$P(T_{\text{event}} > t) = \exp\left(-\underbrace{\int_{t_0}^t \Theta(s) ds}_{H(t)}\right),$$

where  $t_0$  is the moment the previous macro event happened and  $H(t)$  is the **cumulative hazard**. This survival function generalises the familiar  $e^{-\Theta t}$  of the homogeneous case. For more information see [8].

**Hazard–integral sampling rule.** Let  $t_0$  be the calendar time of the last macro-layer event. Define the *cumulative hazard*

$$H(t) = \int_{t_0}^t \Theta(s) ds,$$

where  $\Theta(t) = \sum_{i \neq j} \lambda_{ij}(t)$  is the instantaneous network rate. Draw one uniform deviate  $U \sim \mathcal{U}(0, 1)$  and convert it through the inverse cumulative distribution function (CDF) of the unit-rate exponential:

$$E = F^{-1}(U) = -\ln U \quad \implies \quad E \sim \text{Exp}(1).$$

Thus  $E$  is a sample obtained from the CDF of the  $\text{Exp}(1)$  distribution. The next inter-community infection occurs at the first calendar time  $t^* > t_0$  satisfying

$$H(t^*) = E.$$

Intuitively,  $E$  acts as a random “risk budget”; the event fires when the accumulated hazard reaches this threshold. If  $\Theta(t)$  is constant,  $H(t) = \Theta(t - t_0)$  and the rule collapses to the familiar Gillespie formula  $\Delta t = -\ln U / \Theta$ .

**Numerical scheme for a black-box  $\Theta(t)$ .** Because  $\Theta(t)$  is available only at discrete micro-layer update times, we approximate the integral in rectangles of width  $\tau$  (e.g.  $\tau = 0.1$  day):

$$H_{k+1} = H_k + \Theta(t_k + \frac{\tau}{2}) \tau, \quad t_{k+1} = t_k + \tau.$$

We accumulate these rectangles until  $H_{k+1} \geq E$ . The event must occur inside the last interval  $[t_k, t_{k+1})$ ; assuming  $\Theta$  is nearly constant there, we interpolate to obtain the exact calendar time

$$t^* = t_k + \frac{E - H_k}{\Theta(t_k + \frac{\tau}{2})}.$$

**Selecting the transmitting edge.** At  $t^*$  the probability that edge  $(i, j)$  is the one that fires is

$$\Pr\{(i, j) \text{ fires at } t^*\} = \frac{\lambda_{ij}(t^*)}{\Theta(t^*)}.$$

We seed one infection in community  $j$ , update the micro models for  $i$  and  $j$ , set  $t_0 \leftarrow t^*$ , draw a new  $E' \sim \text{Exp}(1)$ , and repeat the cycle.

**Accuracy and step-size choice.** The local error of the midpoint rectangle is  $O(\tau^3)$ ; choosing  $\tau$  one order of magnitude smaller than the reporting interval (e.g. 0.1 day when epidemiological data are daily) keeps timing errors below a few minutes—negligible relative to epidemiological uncertainty.

**Constant-rate sanity check.** If  $\Theta(t) \equiv \theta$  is constant, the integral collapses to  $\theta(t - t_0)$  and the method reproduces the classical formula  $\Delta t = -\ln U/\theta$ . Thus the hazard-integral procedure is a true generalisation of Gillespie’s original algorithm.

### 3.3. Micro-Layer: Intra-Community Infection Dynamics

This layer runs an **exact continuous-time SIR process** on the contact graph of a single community. It is implemented with the *incremental Gillespie algorithm* described in our earlier work [1] and extended here to cooperate with occasional *imported* (macro-driven) infections.

**Reaction channels.**

#	Transition	Propensity $a_k$
1	$S + I \xrightarrow{\beta_{\text{micro}}} I + I$	$a_1 = \beta_{\text{micro}} \frac{SI}{N}$
2	$I \xrightarrow{\gamma} R$	$a_2 = \gamma I$
3	$\emptyset \xrightarrow{\rho_{\text{imp}}(t)} I$	$a_3 = \rho_{\text{imp}}(t)$

Channel 1 captures within-community contagion, channel 2 recovery, and channel 3 the *import process*: every macro-layer jump seeds one new infectious individual, drawn uniformly at random from the current susceptible list (alternative heuristics—e.g. degree-weighted seeding— are possible).

**Synchronising with the macro layer.** At the end of each micro step of length  $\tau_{\text{micro}}$  the engine exports

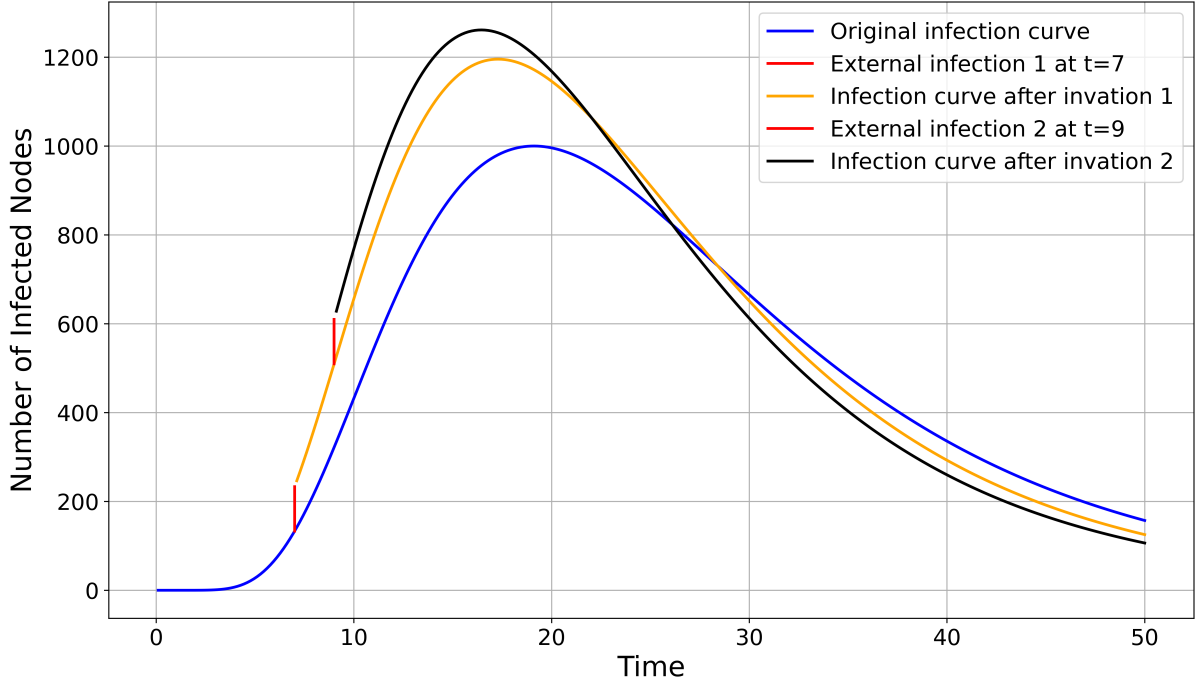
$$C_i(t) = I(t), \quad S_i(t) = S(t),$$

which refresh the inter-community rates  $\lambda_{ij}(t)$ . Conversely, when the macro CTMC fires an inter-community infection at time  $t^*$ , the micro model of the destination community receives one additional  $I$  via channel 3, after which all propensities and the next-event clock are recomputed. Hence the two layers remain *fully synchronised at every discrete tick*.

**Impact of imports on epidemic curves.** Figure 3 illustrates how two external seeding events (red arrows) alter the virus dynamics within a single community: the first import at  $t = 7$  (orange curve) raises and advances the peak, while a second import at  $t = 9$  (black curve) produces an even higher and earlier epidemic crest. The plot confirms that the micro–macro simulator is supposed to accurately capture the compound effect of imported infections on local dynamics.

**Challenge: waiting-time modelling with imports.** External arrivals break the memoryless property that underpins the classic Gillespie exponential draw:

- The internal hazard  $\Theta_{\text{int}}(t) = \beta_{\text{micro}} SI/N + \gamma I$  evolves *continuously*.



**Figure 3:** The impact of external infections on infected curve

- The import hazard  $\rho_{\text{imp}}(t)$  changes *asynchronously* whenever the macro layer injects a case, producing instantaneous jumps in the total hazard.

Two technical issues follow:

1. **Interrupted exponentials.** If an import lands before the next internally scheduled event, the pending exponential draw becomes invalid and must be discarded; a new draw is generated from the updated state.
2. **Cumulative-hazard evaluation.** A draw-once strategy keeps a running integral  $H(t) = \int_{t_0}^t \Theta_{\text{int}}(s) ds$  and fires when  $H(t)$  first exceeds  $E \sim \text{Exp}(1)$ , but the integral must be *reset* each time  $\rho_{\text{imp}}(t)$  jumps.

These complications motivate a dedicated time-estimation scheme, presented in the following subsection, that merges internal CTMC dynamics with macro imports without biasing event times.

### 3.3.1. Modelling Waiting Time Between Events at Micro Layer

The intra-community clock must decide which happens first:

1. an **internal** infection or recovery generated by the SIR Gillespie channels (propensity  $\Theta_{\text{int}}(t) = \beta_{\text{micro}} SI/N + \gamma I$ ), or
2. an **import** scheduled by the macro layer at a deterministic calendar time  $T_{\text{imp},m}$  ( $m = 1, 2, \dots$ ).

Because the SIR propensities are *constant between events*,  $\Theta_{\text{int}}$  does not vary on  $[t_0, \min\{T_{\text{imp},1}, t^*\})$ . This permits an exact,  $\mathcal{O}(1)$  waiting-time draw without numerical integration.

**Residual-budget algorithm for deterministic imports.**



1. **Initial draw.** At time  $t_0$  sample one exponential budget

$$E = -\ln U, \quad U \sim \mathcal{U}(0, 1).$$

If no import were pending, the internal waiting time would be  $\Delta t_{\text{int}} = E/\Theta_{\text{int}}$ .

2. **Compare with next import.** Let  $T_{\text{bar}}$  be the next deterministic import time (or  $+\infty$  if the queue is empty).

$$T_{\text{bar}} - t_0 \begin{cases} < \Delta t_{\text{int}} & \text{import arrives first,} \\ \geq \Delta t_{\text{int}} & \text{internal event wins.} \end{cases}$$

3. **Case A: internal event wins.** Fire the infection or recovery at  $t^* = t_0 + \Delta t_{\text{int}}$ , update  $(S, I, R)$ , set  $t_0 \leftarrow t^*$ , and return to step 1.
4. **Case B: import arrives first.** Advance the clock to  $T_{\text{bar}} = t_0 + \delta$  with  $\delta < \Delta t_{\text{int}}$ ; insert one infectious node (channel 3), which changes the rate to  $\Theta'_{\text{int}}$ .

Compute the *residual hazard budget*

$$E_{\text{res}} = E - \Theta_{\text{int}}\delta,$$

and resume the clock with

$$\Delta t_{\text{after}} = \frac{E_{\text{res}}}{\Theta'_{\text{int}}}.$$

Pop the next import from the queue (if any) and go back to step 2.

## 4. Summary and Outlook

We introduced a two-scale modeling framework for large-scale epidemic simulation that decouples intra-community infection dynamics from inter-community transmission. The core idea is to model each community independently at the micro-layer – using stochastic agent-based algorithm – and aggregate inter-community interactions via a macro-level meta-population graph. This separation of scales enables significant computational savings while preserving the essential structure of disease spread across regions.

Our analysis shows that, assuming a balanced partition into  $M$  equally sized communities, simulation and storage complexity can be reduced by a factor of  $M$ , from  $O(N^2)$  to  $O((\frac{N}{M})^2 \cdot M)$ , where  $N$  is the total population. This reduction is achieved both through an optimized two-level Gillespie algorithm and by learning intra-community dynamics via machine learning surrogates, enabling near-quadratic gains in some configurations. Similar principles apply to visualization: abstracting macro-transmissions and aggregating infection curves allows epidemic flow diagrams to remain interpretable even at national scale.

The important next step would be a thorough analysis of the approximation error introduced by our abstraction to macro-edges and the approximation of inter-infection error introduced by infection of a random node inside a community. We could use Gillespie's model on the full network as a reference model, at least for moderate size graphs, and use empirical data or parallel computing clusters for sampling micro-simulation reference data for large scale reference networks.

Looking ahead, several extensions are promising. First, improving the fidelity of intra-community dynamics through hybrid machine learning–mechanistic models could boost both accuracy and speed. Second, using data from real-world mobility and contact data for macro-layer weighting will enhance policy relevance. Finally, we envision this framework as a foundation for real-time decision support systems that simulate interventions interactively across spatial and social layers.

## Declaration on Generative AI

During the preparation of this work, one of the authors (Y. Kuryliak) used ChatGPT-4o and ChatGPT-o3 in order to improve grammar and text readability. Further, the author used ChatGPT to generate figure 1(b). After using these tool, the author reviewed and edited the content as needed and take full responsibility for the publication's content.

## References

- [1] Y. Kuryliak, M. T. Emmerich, D. Dosyn, Simulating epidemic peak dynamics on complex networks using efficient gillespie algorithms, *Infection, Genetics and Evolution* (2025) 105768.
- [2] V. Colizza, A. Vespignani, Epidemic modeling in metapopulation systems with heterogeneous coupling pattern: Theory and simulations, *Journal of theoretical biology* 251 (2008) 450–467.
- [3] D. Balcan, V. Colizza, B. Gonçalves, H. Hu, J. J. Ramasco, A. Vespignani, Modeling the spatial spread of infectious diseases: The global epidemic and mobility computational model, *Journal of Computational Science* 1 (2010) 132–145.
- [4] M. Ajelli, B. Gonçalves, D. Balcan, V. Colizza, H. Hu, J. J. Ramasco, S. Merler, A. Vespignani, Comparing large-scale computational approaches to epidemic modeling: Agent-based versus structured metapopulation models, *BMC Infectious Diseases* 10 (2010) 190.
- [5] N. Hoertel, M. Blachier, C. Blanco, M. Olsson, M. Massetti, M. S. Rico, F. Limosin, A stochastic agent-based model of the sars-cov-2 epidemic in france, *Nature Medicine* 26 (2020) 1417–1421.
- [6] C. C. Kerr, R. M. Stuart, D. Mistry, R. G. Abeyasuriya, G. Hart, K. Rosenfeld, et al., Covasim: an agent-based model of covid-19 dynamics and interventions, *PLoS Computational Biology* 17 (2021) e1009149.
- [7] L. Kou, X. Wang, Y. Li, X. Guo, H. Zhang, A multi-scale agent-based model of infectious disease transmission to assess the impact of vaccination and non-pharmaceutical interventions: The covid-19 case, *Journal of Safety Science and Resilience* 2 (2021) 199–207.
- [8] C. L. Vestergaard, M. Géniois, Temporal gillespie algorithm: Fast simulation of contagion processes on time-varying networks, *PLoS Computational Biology* 11 (2015) e1004579. doi:10.1371/journal.pcbi.1004579.